

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH ŘEŠENÍ INTELIGENTNÍHO DOMU S BEZDRÁTOVÝMI
SENZORY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN MACHÁČEK

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH ŘEŠENÍ INTELIGENTNÍHO DOMU S BEZDRÁTOVÝMI SENZORY

SOLUTION PROPOSAL FOR INTELIGENT HOUSE USING WIRELESS SENSORS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN MACHÁČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ KRAJSA, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jan Macháček

ID: 146895

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Návrh řešení inteligentního domu s bezdrátovými senzory

POKYNY PRO VYPRACOVÁNÍ:

S využitím platformy Raspberry-Pi navrhnete a realizujete systém pro ovládání inteligentního domu (spínání spotřebičů, ovládání motoru, měření vybraných veličin). K ovládání systému navrhnete webové rozhraní. Část ovládání a měření bude realizována bezdrátově.

DOPORUČENÁ LITERATURA:

[1] GIRLING, Gary. Raspberry Pi: a practical guide to the revolutionary small computer. Sparkford: Haynes Publishing, 2013, 169 s. ISBN 978-0-85733-295-0.

[2] HORAN, Brendan. Practical Raspberry Pi. New York: Apress, 2013, xxxii, 239 s. ISBN 978-1-4302-4971-9.

Termín zadání: 10.2.2014

Termín odevzdání: 4.6.2014

Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Obsahem této bakalářské práce je návrh systému pro řízení inteligentního domu založeného na platformě Raspberry Pi. Práce obsahuje návrh expanzní desky pro rozšíření GPIO portu Raspberry Pi. Expanzní deska zahrnuje množství periferií, které je možné ovládat ovládacím panelem nebo prostřednictvím webového rozhraní. Pomocí komplexního systému lze ovládat různá elektrická zařízení a vytvářet interakce na základě zadaných podmínek.

KLÍČOVÁ SLOVA

inteligentní dům, Raspberry Pi, expanzní deska, bezdrátové senzory

ABSTRACT

The purpose of the bachelor's thesis was design of system for management of an Intelligent House based on a platform Raspberry Pi. The paper contains design of expansion panel from GPIO port of Raspberry Pi. The expansion panel includes many controllable peripheries which can be accessed by a control panel or a web interface. The complex system can manage various electrical devices and interact based on given conditions.

KEYWORDS

intelligent house, Raspberry Pi, expansion panel, wireless sensors

MACHÁČEK, Jan *Návrh řešení inteligentního domu s bezdrátovými senzory*: bakalářská práce. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 60 s. Vedoucí práce byl Ing. Ondřej Krajsa, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh řešení inteligentního domu s bezdrátovými senzory“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Místo

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ondřeji Krajsovi, Ph.D. za odborné vedení, konzultace a podnětné návrhy k práci.

Místo

.....

(podpis autora)

OBSAH

| | |
|--|-----------|
| Úvod | 11 |
| 1 Řídící jednotka | 13 |
| 1.1 Raspberry Pi | 13 |
| 1.1.1 Operační systém | 13 |
| 1.1.2 Instalace | 14 |
| 1.1.3 Programování | 15 |
| 1.1.4 GPIO konektor | 16 |
| 1.1.5 Ovládání GPIO | 16 |
| 1.1.6 Sběrnice I2C (Inter-Integrated Circuit) | 18 |
| 1.1.7 Sběrnice SPI (Serial Peripheral Interface) | 19 |
| 1.2 Expandér MCP23017 | 19 |
| 1.2.1 Ovládání MCP23017 | 19 |
| 1.3 Blokové schéma | 22 |
| 1.4 Ovládání displeje a klávesnice | 23 |
| 1.4.1 Maticová klávesnice | 24 |
| 1.4.2 Displej | 25 |
| 1.5 Periferie | 26 |
| 1.5.1 Modul reálného času (RTC) | 26 |
| 1.5.2 Relé spínače | 27 |
| 1.5.3 Teplotní čidlo | 28 |
| 1.5.4 Krokový motor | 29 |
| 1.5.5 PWM řízení | 30 |
| 2 Bezdrátová část | 33 |
| 2.1 Arduino Nano 3.0 | 34 |
| 2.1.1 Programování | 34 |
| 2.2 Bezdrátový modul nRF24L01 | 35 |
| 2.3 Teplotní čidlo Dallas 18b20 | 35 |
| 3 Software | 36 |
| 3.1 Webové rozhraní a MySQL databáze | 37 |
| 3.1.1 Umístění | 37 |
| 3.2 Raspberry Pi | 37 |
| 3.3 Bezdrátový modul | 38 |

| | | |
|----------|---|-----------|
| 4 | Realizace a možnosti rozšíření | 39 |
| 4.1 | Realizace | 39 |
| 4.2 | Rozšíření | 39 |
| 5 | Závěr | 41 |
| | Literatura | 42 |
| | Seznam symbolů, veličin a zkratk | 44 |
| | Seznam příloh | 45 |
| A | Rozšiřující Deska | 46 |
| A.1 | Schéma zapojení | 46 |
| A.2 | Deska plošných spojů | 48 |
| A.3 | Seznam součástek | 52 |
| B | Bezdrátová část | 53 |
| B.1 | Deska plošných spojů | 53 |
| B.2 | Seznam součástek | 55 |
| C | Ukázka webových stránek | 56 |
| D | Návod k instalaci | 58 |
| E | Návod k obsluze | 59 |
| F | Obsah CD s přílohou | 60 |

SEZNAM OBRÁZKŮ

| | | |
|------|--|----|
| 1.1 | Raspberry Pi Model B | 14 |
| 1.2 | Tabulka prvotního nastavení | 15 |
| 1.3 | Rozložení GPIO portu [2] | 16 |
| 1.4 | Expandér MCP23017 | 20 |
| 1.5 | Blokové schéma | 22 |
| 1.6 | Zapojení displeje a klávesnice | 23 |
| 1.7 | Maticová klávesnice | 24 |
| 1.8 | Struktura menu | 25 |
| 1.9 | Schéma zapojení relé | 27 |
| 1.10 | Schéma zapojení teplotního čidla | 29 |
| 1.11 | Schéma zapojení krokového motoru | 30 |
| 1.12 | PWM driver | 31 |
| 2.1 | Schéma bezdrátové části | 33 |
| 2.2 | Arduino Nano 3.0 | 34 |
| 3.1 | Vývojový diagram celého zařízení | 36 |
| A.1 | Předloha desky plošných spojů spodní stana | 48 |
| A.2 | Předloha desky plošných spojů vrchní stana | 49 |
| A.3 | Osazovací plán spodní stana | 50 |
| A.4 | Osazovací plán vrchní stana | 51 |
| B.1 | Předloha desky plošných spojů spodní stana | 53 |
| B.2 | Předloha desky plošných spojů vrchní stana | 53 |
| B.3 | Osazovací plán spodní stana | 54 |
| B.4 | Osazovací plán vrchní stana | 54 |
| C.1 | Záložka ovládání | 56 |
| C.2 | Záložka statusy | 57 |

SEZNAM TABULEK

| | | |
|-----|--|----|
| 1.1 | Parametry Raspberry Pi | 13 |
| 1.2 | Adresování MCP23017 výstup | 20 |
| 1.3 | Adresování MCP23017 vstup | 21 |
| 1.4 | Adresování MCP23017 výstup | 21 |
| 1.5 | Seznam registrů a jejich parametrů | 32 |

SEZNAM UKÁZEK

| | | |
|------|--|----|
| 1.1 | Vyvolání prvotní konfigurace | 15 |
| 1.2 | Instalace ovladače. | 17 |
| 1.3 | Instalace I_2C podpory pro Python. | 17 |
| 1.4 | Spouštění I_2C po startu. | 17 |
| 1.5 | GPIO17 jako výstup a vstup. | 18 |
| 1.6 | Příkaz pro zobrazení detekovaných I_2C zařízení. | 18 |
| 1.7 | MCP23017 registr GPB jako výstup. | 20 |
| 1.8 | MCP23017 registr GPA jako vstup. | 21 |
| 1.9 | MCP23017 nastavení GPA0 na Log „1“. | 21 |
| 1.10 | Instalace RTC8563. | 26 |
| 1.11 | Zápis času do RTC. | 26 |
| 1.12 | Výpis času z RTC. | 26 |
| 1.13 | PWM řízení. | 32 |

ÚVOD

Tato práce se zabývá návrhem řídicího systému pro inteligentní dům, tedy návrhem systému, který je schopen komplexně ovládat různé periferie domu nebo bytu, s co nevyšším pohodlím uživatele. V dnešní době máme doma spoustu různých řídicích systémů např.: systém regulace topení, řízení osvětlení, dálkově ovládané brány apod. Všechny tyto systémy fungují samostatně a tak uživatel, který je ovládá je nucen mít několik různých ovládacích prvků a orientovat se v několika ovládacích systémech. Proto vznikla myšlenka navrhnout systém, který bude schopen všechny tyto ovládací prvky zahrnout pod jeden, který je dokáže ovládat všechny najednou. Ze sjednocení ovládání pak plyne spousta teoretických výhod a tím větší komfort pro uživatele. Hlavní podmínkou bylo pro řízení využít platformu Raspberry - Pi, která se v posledních letech stala velmi oblíbenou, díky své jednoduchosti a nízké ceně.

Využití

Jak už bylo popsáno v úvodu, je toto zařízení koncipováno především pro využití v rodinném domě, kde je právě množství ovládatelných periférií. Samozřejmě lze zařízení použít například v bytě, či na objekt, který běžně neobýváme např. chatu, včelín atd. Díky pokrytí 3G a EDGE sítě máme dneska téměř všude přístup k internetu a tak se nabízí ovládat „inteligentní stavbu“, právě pomocí těchto chytrých sítí. Kromě ovládání periférií je neméně důležité měření fyzikálních veličin. V tomto případě je to především měření teploty, tedy možnost v každé místnosti měřit teplotu, díky použití Raspberry - Pi a expandérů je to teoreticky až 128 čidel střídových a nespočet digitálních. Dále lze měřit i jiné veličiny, jako například atmosférický tlak, obsah CO₂ ve vzduchu, vlhkost vzduchu atd. Avšak měření těchto veličin není popsáno v této práci. Dalším prvkem jsou digitální vstupy, na které lze napojit systém pohybových čidel či optických závor a na základě jejich sepnutí vyvolat nějakou interakci. V neposlední řadě fakt, že o řízení všech periférií se stará jeden systém dává uživateli možnost tvořit si scénáře na základě času. Což znamená například ve stanovenou dobu provést určité operace, spustit kropení trávníku, ráno otevřít okna, roztáhnout žaluzie atd.

Požadavky

- Celý systém musí být navržen na platformě Raspberry Pi
- Navrhnout expanzní modul pro Raspberry – Pi
- Spínání elektrických spotřebičů
- Ovládání krokového motoru
- Měření vybraných veličin
- Navrhnout ovládání pomocí webového rozhraní
- Návrh a provedení bezdrátového senzoru

1 ŘÍDÍCÍ JEDNOTKA

1.1 Raspberry Pi

Jak je uvedeno v zadání, řídicí jednotka, tedy srdce celého systému stojí na platformě Raspberry Pi, konkrétně na její již druhé revizi Raspberry Pi model B. Model B má oproti předchozí verzi více operační paměti, o jeden USB port více a ethernetový adaptér, který je pro aplikaci klíčový. Je to malý jednočipový plnohodnotný počítač, který vyniká zejména svojí velikostí a nízkou cenou a univerzálností využití. Raspberry Pi má tyto následující hardwarové parametry:

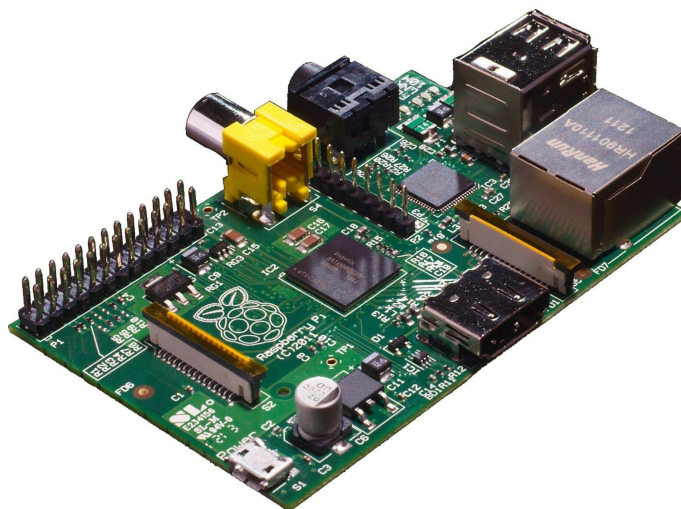
| | |
|---|----------------------|
| Processor | ARM1176JZF-S 700 Mhz |
| Grafický procesor | VideoCore IV |
| Operační paměť | 512 MB |
| 2*USB 2.0 | |
| Obrazový výstup Composite RCA, HDMI, DSI | |
| Zvukový výstup přes 3,5 mm konektor, HDMI | |
| slot pro SD nebo MMC kartu | |
| ethernetový adaptér 10/100 s konektorem RJ45 | |
| 12*GPIO, UART, I ₂ C, sběrnici SPI | |

Tab. 1.1: Parametry Raspberry Pi

Raspberry Pi není příliš výkonné, ale pro tuto aplikaci je téměř ideální. Tím, že je to plnohodnotný PC, podporuje různé verze operačních systémů (Linux, Android). Hlavní výhodou jsou tzv. GPIO porty. [1]

1.1.1 Operační systém

Výrobce a komunita nabízí nepřehledné množství operačních systémů. Na výběr jsou buď linuxové distribuce, nebo distribuce zakládající se na systému Android. Byla zvolena oficiální distribuce „Raspbian - Wheezy“, která má základ v Debianu 6. Důvody pro výběr této distribuce byly především stabilita systému a kompromis mezi svižností systému a možnostmi, které právě Debian 6 nabízí. Dalším důvodem byl také fakt, že právě tato verze upravená pro Raspberry Pi, je hojně využívaná komunitou vývojářů na internetu. Díky tomu existuje spousta internetových fór, která se zabývají popsanou tematikou, jak v anglickém tak i v českém jazyce.



Obr. 1.1: Raspberry Pi Model B

1.1.2 Instalace

Protože Raspberry Pi, neobsahuje žádnou vlastní interní paměť typu ROM, pro tento účel je osazena slotem pro paměťové karty typu SD(HC). Nejdříve bylo tedy nutné vybrat vhodnou paměťovou kartu. Na kartu nejsou kladeny žádné specifické nároky, je však vhodné aby její kapacita byla minimálně 4 GB, protože při použití 2 GB karty nezbývalo téměř žádné místo pro další aplikace. Kromě kapacity je důležitým aspektem rychlost čtení a zápisu dat, která ovlivňuje celkovou plynulost systému. V této práci byla zvolena karta o kapacitě 4 GB. Kromě SD karty je po zakoupení Raspberry Pi nutné dokoupit micro USB kabel, který slouží k napájení a není součástí balení. Dále je třeba stáhnout samotný systém například z oficiální stránky: <http://www.raspberrypi.org/downloads/>. Stažený soubor typu zip „2014-01-07-wheezy-raspbian.zip“ má po rozbalení lehce pod 2 GB. Potom je třeba soubor.img nakopírovat na paměťovou kartu, tak aby z ní šlo později nabootovat. K tomuto účelu existuje velké množství freewarových programů. V tomto případě byl použit program „Win32 Disk Imager“. Po úspěšné přípravě paměťové karty již stačí vložit kartu přímo do příslušného slotu a připojit Raspberry Pi k napájení. Při prvním spuštění je třeba připojit klávesnici, myš a monitor (2*USB, HDMI), pro prvotní konfiguraci. Po prvním spuštění se zobrazí tabulka obr: 1.2, kde je možné provést základní konfiguraci, nastavení hesla, nastavení času a ča-

| Raspi-config | |
|---------------------------------|---------------------------------------|
| <code>info</code> | Information about this tool |
| <code>expand_rootfs</code> | Expand root partition to fill SD card |
| <code>overscan</code> | Change overscan |
| <code>configure_keyboard</code> | Set keyboard layout |
| <code>change_pass</code> | Change password for 'pi' user |
| <code>change_locale</code> | Set locale |
| <code>change_timezone</code> | Set timezone |
| <code>memory_split</code> | Change memory split |
| <code>ssh</code> | Enable or disable ssh server |
| <code>boot_behaviour</code> | Start desktop on boot? |
| <code>update</code> | Try to upgrade raspi-config |

<Select>
<Finish>

Obr. 1.2: Tabulka prvotního nastavení

sové zóny, zapnutí a vypnutí grafického prostředí a především povolení serveru SSH, který slouží pro vzdálené připojení například pomocí programu „PuTTY“. Právě připojování pomocí programu „PuTTY“ se ukázalo jako velmi praktické. Tabulka pro prvotní konfiguraci se zobrazí pouze po prvním spuštění, pro pozdější vyvolání je nutné zadat následující příkaz:

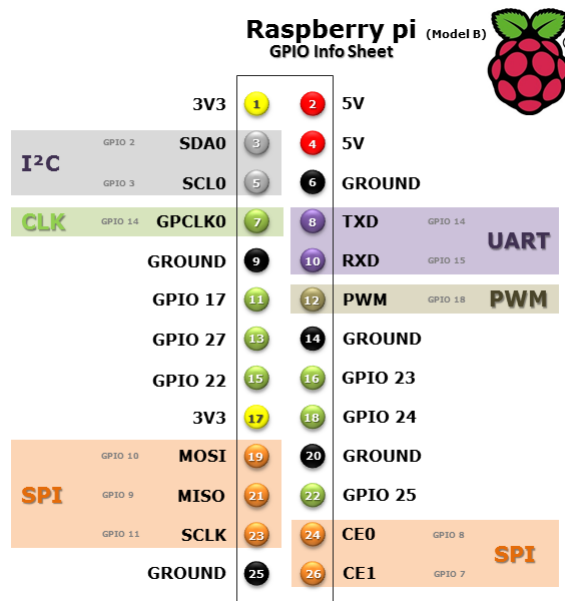
Ukázka 1.1: Vyvolání prvotní konfigurace

```
sudo raspi-config
```

Kromě programu „PuTTY“, který vzdáleně zobrazuje pouze terminál, lze využít i program VNC, který slouží pro vzdálené sdílení plochy. Avšak protože se jedná o Linux, pro všechny operace stačí pouze terminál. Pro vzdálený přístup je nutné znát IP adresu ethernetového rozhraní. To lze zjistit, buď příkazem „ifconfig“ nebo v administraci směrovače. Raspberry Pi je defaultně nastaveno, aby IP adresu získalo z DHCP serveru.

1.1.3 Programování

Při testování různých součástí a při prvních pokusech GPIO, bylo potřeba nějakým způsobem psát různé malé testovací skripty. První skripty byly psány pro bash a byly ideální pro vyzkoušení některých základních principů ovládání. Pro naprogramování celého systému se však nehodily. Pro všechny další operace byl použit skriptovací jazyk Python 2. Python je výhodný zejména díky své jednoduché syntaxi a další důvod byl existence velké vývojářské komunity, která používá Python



Obr. 1.3: Rozložení GPIO portu [2]

právě v kombinaci s Raspberry Pi. Díky tomu lze nalézt spousty informačních fór a stránek, které se zabývají danou problematikou.

1.1.4 GPIO konektor

Tento konektor je hlavním důvodem, proč je Raspberry Pi využíváno v různých hardwarových řešeních. Jsou to vlastně dvě řady po třinácti vidlicových vývodech. Na těchto pinech je vyvedeno 12 GPIO pinů. Tyto piny, jak už název napovídá, jsou plně ovladatelné digitální vstupy nebo výstupy. Mezi těmito piny jsou vyvedeny tři komunikační rozhraní UART, I2C a SPI. V mém případě jsem využil I2C v kombinaci s expandérem MCP23017. Dále GPIO výstupy obsahují napájení 3,3 V a 5 V. Tyto piny byly využity pro napájení expanzní desky. Napájecí piny mají omezení v maximálním odebíraném proudu a to asi 400 mA, záleží totiž na aktuální spotřebě Raspberry, která se při zátěži pohybuje okolo 700 mA. Dle datasheetu vstupní pojistka vyřadí Raspberry z provozu při překročení 1,1 A.

1.1.5 Ovládání GPIO

Ještě před začátkem ovládání bylo nutné pozastavit se nad několika parametry a omezeními. Bez použití expandérů máme k dispozici 12 digitální vstupů nebo výstupů. Každý z těchto I/O má při log 1 napětí 3,3 V, jako log 0 Raspberry podle

datasheetu rozlišuje napětí do 1 V. Maximální proud, který dokáže jeden GPIO dodat je 16 mA. Pokud je pin nastaven jako vstup, dochází k takzvanému hazardu, protože má takovýto vstup vysokou impedanci a nedá se jednoznačně říct, jak jej Raspberry vyhodnotí. Proto je nutné využít pull-Up rezistory. Tento problém byl nakonec vyřešen aktualizací GPIO ovladače, kde přibyla funkce připojení vnitřních rezistorů. Jak bylo již zmíněno, součástí GPIO konektoru je I2C sériová sběrnice. Tato sběrnice byla použita pro ovládání expanzní desky, tedy pro řízení MCP23017 expandérů a dalších zařízení ovládaných sériově (více v kapitole 1.2). [3]

Před samotným ovládáním bylo nutné nainstalovat ovladače pro ovládání GPIO s podporou pro Python, to lze provést pomocí následujících příkazů v terminálu:

Ukázka 1.2: Instalace ovladače.

```
wget https://raspberrypi-gpio-python.googlecode.com/files
/python-rpi.gpio_0.5.3a-1_armhf.deb
sudo dpkg -i python-rpi.gpio_0.5.3a-1_armhf.deb
```

Pro uvedení I2C do provozu, bylo třeba upravit soubor:

```
etc/modprobe.d/raspi-blacklist.conf
```

Zde je třeba zakomentovat všechny vypsané řádky a nainstalovat opět podporu pro Python pomocí příkazů:

Ukázka 1.3: Instalace I₂C podpory pro Python.

```
sudo apt-get install i2c-tools
sudo modprobe i2c-dev
sudo apt-get install python-smbus.
```

Nakonec se edituje soubor `/etc/modules` do kterého je třeba přidat řádek `i2c-dev`, což má za následek, že modul I2C se bude spouštět automaticky po startu.

Ukázka 1.4: Spouštění I₂C po startu.

```
sudo nano /etc/modules
```

Příklady programů pro ovládání GPIO vstup a výstup:

Ukázka 1.5: GPIO17 jako výstup a vstup.

```
#!/usr/bin/python
# -*- encoding: utf-8 -*-

import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT) # nastavení pinu jako výstupní

GPIO.output(17, True) # přepnutí GPIO17 do log "1"
time.sleep(2) # spoždění 2s
GPIO.output(17, False) # nastavení pinu na log "0"
time.sleep(2) # spoždění 2s

GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
#nastavení pinu GPIO 17 jako vstupní a připojení pull -up odporu

while True:
    print GPIO.input(8) #vypíše aktuální logický stav
    time.sleep(0.5)
```

1.1.6 Sběrnice I2C (Inter-Integrated Circuit)

Součástí GPIO portu je sběrnice I2C. Tato sběrnice propojuje jedno zařízení Master a více zařízení typu Slave. Propojení je realizované pomocí dvou vodičů SDA - datový vodič a SCL - hodinový signál. Jedná se tedy o polo-duplexní komunikaci, kdy je v daném okamžiku vysíláno pouze jedním směrem. Komunikace probíhá tak, že Master může vysílat vždy, Slave pouze na vyžádání Mastera. Rozlišení více zařízení typu Slave je zajištěno pomocí adresy, tedy každé takové zařízení má svoji vlastní adresu. Počet zařízení typu Slave, je limitován počtem adres a kapacitancí sběrnice. Po připojení nějakého integrovaného obvodu pomocí I2C, si lze na Raspberry Pi ověřit jeho funkčnost a správné zapojení následujícím příkazem:

Ukázka 1.6: Příkaz pro zobrazení detekovaných I2C zařízení.

```
sudo i2cdetect 1
```

Po zadání příkazu v terminálu se objeví tabulka adres připojených zařízení. Jedná se o užitečný příkaz, který se osvědčil při ožívování expanzní desky v následující kapitole.

1.1.7 Sběrnice SPI (Serial Peripheral Interface)

Je to externí sériová sběrnice sloužící k propojení dvou nebo více komunikujících uzlů. Podobně jako je tomu u I2C se zde dělí zařízení na Master a Slave. Hodinový signál (vodič SCK) je distribuován do všech uzlů sběrnice a pro posílání dat jsou zde dva vodiče MISO a MOSI. Dalším vodičem této sběrnice je CS, slouží k výběru aktivního uzlu sběrnice. Sběrnice používá fullduplexní přenos dat. Nevýhodou SPI je, že data pomocí této sběrnice, je možné přenášet pouze na kratší vzdálenost, což je způsobeno nutností synchronizace hodinového signálu. Další nevýhodou je, že přenos neobsahuje žádné potvrzení doručení. Tedy v případě, kdy některé zařízení nedokáže dostatečně rychle zpracovávat přijímaná data, dojde k nedoručení a tím i k chybě.

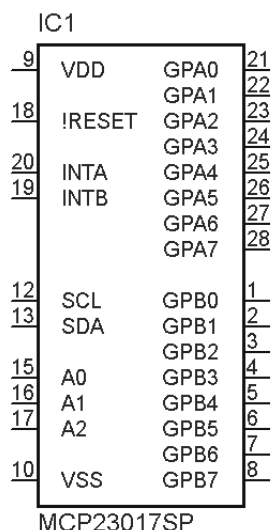
1.2 Expandér MCP23017

Vzhledem k tomu, že Raspberry Pi má na svém GPIO portu pouze dvanáct I/O pinů, bylo nutné najít nějaký způsob, kterým bylo možné tuto sadu pinů rozšířit tak, aby bylo možné připojit více periferií. K tomuto účelu byl použit integrovaný obvod MCP23017. Je to 16 bitový expander, který se připojuje pomocí komunikační sběrnice I2C. Celý obvod lze napájet od 1,8 V do 5,5 V. Stejně jako u GPIO portu na Raspberry, dokáže každý pin dodat maximální proud o hodnotě 16 mA při napájecím napětí 3,3 V. Další výhodou je, že expander již obsahuje pull-up rezistory na všech GP pinech, které lze pomocí příslušného příkazu odpojit nebo připojit. Ty jsou vhodné při vyčítání stavu z jednotlivých pinů, kdy stejně jakou u GPIO pinů, může nastat hazardní stav.[5]

Rozšiřující obvod je 16 bitový, tyto byty jsou však rozděleny do dvou registrů GPA a GPB. Komunikace přes I₂C probíhá přes piny 12 a 13. Každý takto připojený expander rozšiřuje celkový počet GPIO pinů o 16. Každý takový obvod je 3 bitově adresovatelný pomocí pinů A0, A1 a A2. Celkově tedy lze připojit až osm takových expandérů, čímž lze získat až 128 I/O vstupů, přičemž lze každý zvlášť ovládat.

1.2.1 Ovládání MCP23017

Důležitým prvkem při komunikaci s expandérem je adresování. U každého obvodu je nutné nastavit jeho adresu, pomocí které pak bude komunikovat přes sběrnici.



Obr. 1.4: Expandér MCP23017

Adresa se stanovuje zapojením. K tomuto účelu slouží piny A0, A1, A2. Tyto piny se připojí buď na napájení, nebo na zem, podle toho jakou adresu chceme zvolit.

Pokud všechny piny připojíme na zem, bude adresa 0x20. Číslo za (x) vždy začíná číslem 20, k tomuto číslu se pak přičítá námi nastavené číslo v binární hodnotě. Tedy pokud je například připojen pin A2 na VCC, bude tím reprezentovat číslo 4. Když toto číslo přičteme k číslu 20, získáme výslednou adresu 0x24.

Pomocí skriptu lze opět nastavit každý port jako vstup nebo výstup. Jelikož má expander rozdělené jednotlivé piny do dvou registrů, lze ovládat souhrnně i jednotlivé registry. Nastavování jednotlivých registrů, pro vstup nebo výstup, se provádí pomocí následujících příkazů a adres v hexadecimálním tvaru. [4]

Příklad nastavení všech pinů v registru GPB jako výstupních:

Ukázka 1.7: MCP23017 registr GPB jako výstup.

```
sudo i2cset -y 1 0x20 0x01 0x00 # nastavení GPB registru jako Výstup
```

| Adresa v Hexadecimálním tvaru | Funkce |
|-------------------------------|------------------------------------|
| 0x20 | adresa obvodu |
| 0x01 | adresa registru GPB |
| 0x00 | nastaví všechny piny jako výstupní |

Tab. 1.2: Adresování MCP23017 výstup

Příklad nastavení všech pinů v registru GPA jako vstupních:

Ukázka 1.8: MCP23017 registr GPA jako vstup.

```
sudo i2cset -y 1 0x20 0x00 0xFF # nastavení GPB registru jako Vstup
```

| Adresa v Hexadecimálním tvaru | Funkce |
|-------------------------------|-----------------------------------|
| 0x20 | adresa obvodu |
| 0x00 | adresa registru GPA |
| 0xFF | nastaví všechny piny jako vstupní |

Tab. 1.3: Adresování MCP23017 vstup

Příklad nastavení prvního pinu GPA0 jako log „1“ :

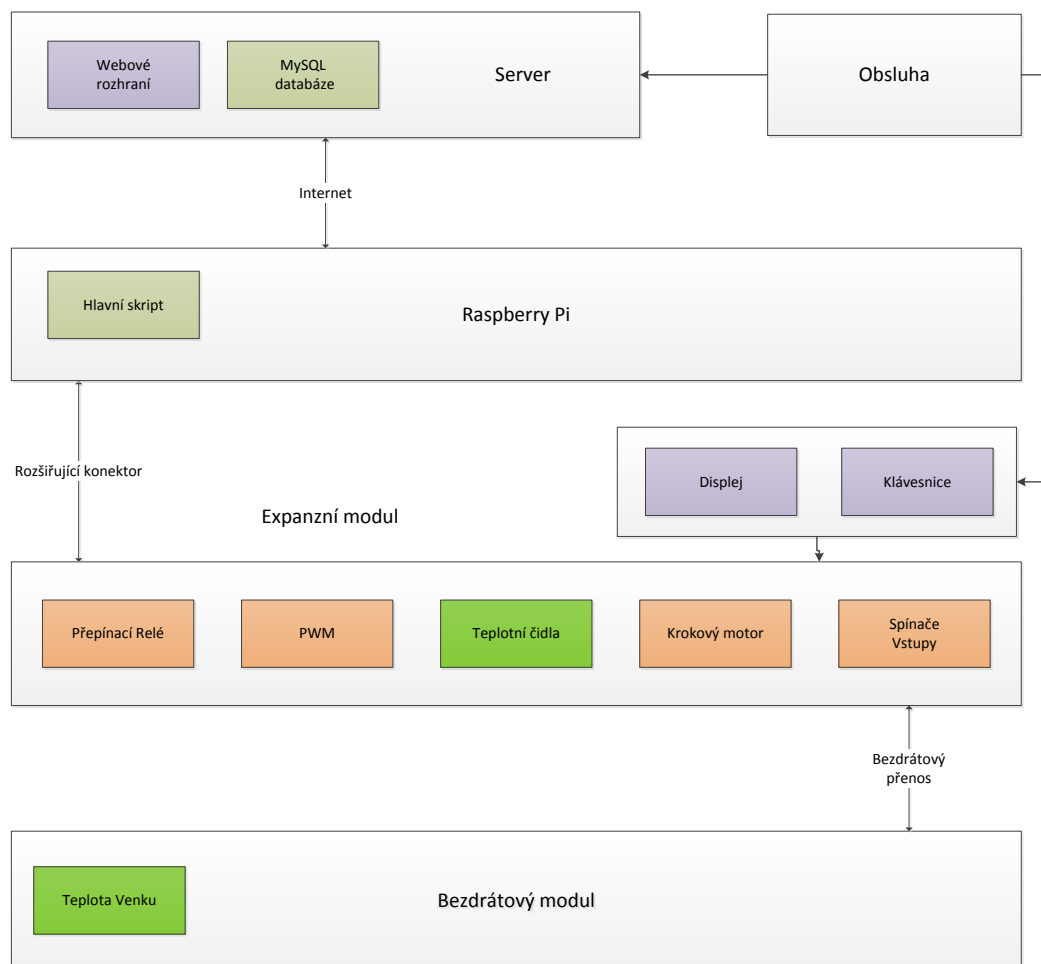
Ukázka 1.9: MCP23017 nastavení GPA0 na Log „1“.

```
sudo i2cset -y 1 0x20 0x12 0x01
```

| Adresa v Hexadecimálním tvaru | Funkce |
|-------------------------------|------------------------------------|
| 0x20 | adresa obvodu |
| 0x00 | adresa registru GPA pro nastavení |
| 0x01 | nastavení nultého bitu do log „1“. |

Tab. 1.4: Adresování MCP23017 výstup

1.3 Blokové schéma

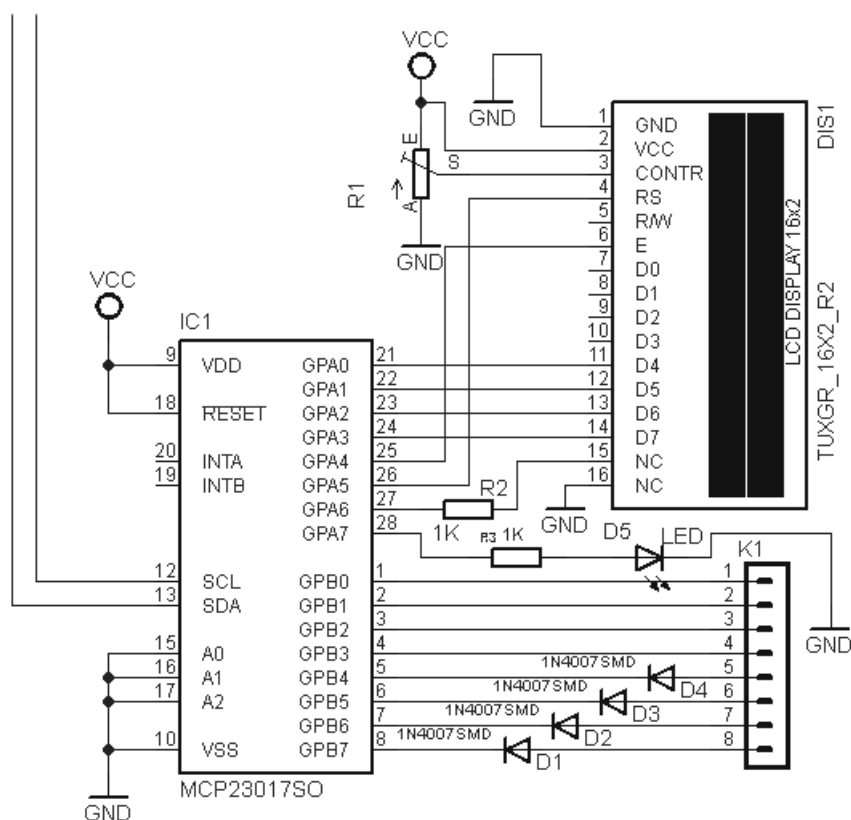


Obr. 1.5: Blokové schéma

Na obr: 1.5 lze vidět blokové schéma celého navrženého systému. Celé zařízení lze rozdělit do dvou funkčních bloků. První blok zahrnuje webové rozhraní, které slouží k obsluze, dále MySQL databáze, ve které se nachází data aktuálního nastavení celého zařízení. Druhý blok je samotné Raspberry Pi, sloužící jako řídicí jednotka, na které je spuštěn řídicí skript. Ten zajišťuje ovládání periferií a vytváří uživatelské prostředí ve spojení s displejem a klávesnicí. Dále může být na Raspberry Pi spuštěn webový server Apache (záleží, zda jsou webové stránky uloženy na serveru v Raspberry Pi, nebo na hostingu od providera), na kterém běží webové rozhraní. Blok expanzní modul je připojen k řídicí jednotce pomocí GPIO portu. Tato připojená rozšiřující deska zajišťuje přímé ovládání připojitelných periferií. Na rozšiřující desce se nachází dva ovládací prvky, klávesnice a displej, které slouží k obsluze. Dále

tři PWM výstupy, tři relé spínače, jeden výstup na krokový motor, dvě čidla teploty a trojice digitálních vstupů (3-5 V). Ve schématu se dále nachází třetí blok, bezdrátový modul, který zastupuje bezdrátové senzory. Tento bezdrátový modul obsahuje teplotní čidlo, vlastní řídicí jednotku a modul pro bezdrátovou komunikaci. Celý systém lze ovládat pomocí kombinace dvou rozhraní. Buď pomocí displeje a klávesnice, které jsou osazeny přímo na krabičce daného modulu, nebo pomocí webové stránky, která je přístupná buď z internetu nebo z lokální sítě, ke které je připojeno zařízení.

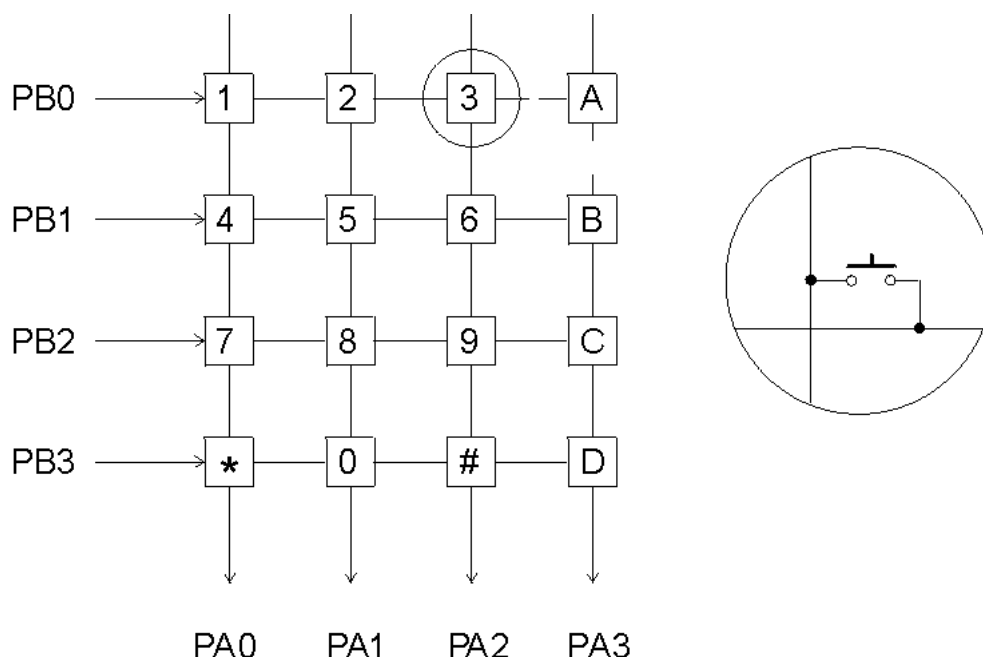
1.4 Ovládání displeje a klávesnice



Obr. 1.6: Zapojení displeje a klávesnice

Základem tohoto řídicího bloku, obr: 1.6, je dvouřádkový alfanumerický displej, po 16 znacích na jeden řádek. Jako ovládací vstup slouží maticová klávesnice 4x4. Klávesnice i displej jsou připojeny k expanderu MCP230017.

1.4.1 Maticová klávesnice



Obr. 1.7: Maticová klávesnice

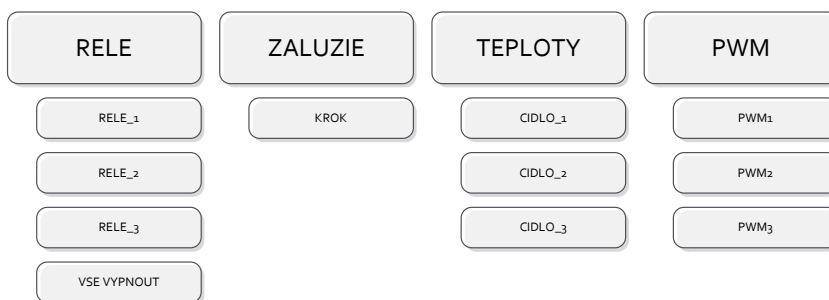
Jako vstup byla použita 4x4 maticová klávesnice. Použití maticového zapojení má několik výhod i nevýhod. Nejdříve výhody - pokud by byla standardně připojena všechna tlačítka k expandéru, využil bych tím všechny GPA a GPB porty, celkem 16 pinů. V zapojení do matice se tak redukuje počet použitých GPIO portů na polovinu. Nevýhoda je komplikovanější ovládání a neschopnost rozpoznat některé kombinace stisknutých tlačítek. Další nevýhodou je, že rozpoznání stisku jednotlivých tlačítek trvá trochu déle. Celkově ovšem převládají klady nad zápory, neboť odezva tlačítek ani v maticovém provedení není nikterak kritická a aplikace nevyužívá vícenásobné stisknutí tlačítek v jeden moment.

Princip rozpoznání stisku je následující. Celý 8 – bitový registr GPB je rozdělen do dvou částí - 1 až 4 a 5 až 8. K první skupině pinů jsou připojeny sloupce, k druhé řádky. V první fázi běží na pinu 5 až 8 sekvence, kdy se postupně na každém pinu na malý okamžik objeví log „0“. Piny 1 až 4 jsou díky přepnutí pull - up odporu přivedeny do log „1“. Ve chvíli kdy dojde ke stisku, objeví se na příslušném pinu 1 až 4 log „0“. Na základě toho, že vždy v jeden okamžik je log „0“ přivedena pouze na jeden řádek a log „0“ se po stisku objeví pouze v jednom sloupci, lze snadno odvodit, který znak byl stisknut.

1.4.2 Displej

Pro zobrazení byl využit jednoduchý alfanumerický displej CM160224SFAYA-I1. Displej je zapojen dle obr: 1.8 a je standardně ovládán 8 + 2 datovými vodiči, ale lze jej zapojit tak, že postačí pouze 4 + 2 datové vodiče. Což v konečném důsledku znamená úsporu 4 GPIO pinů. Princip je následující. Při polovičním počtu ovládacích pinů se instrukce vysílá po dvou intervalech. Nejdříve se pošle horní polovina bytu, potom se vyšle kladný impuls na pin E a následuje spodní polovina bytu. Následuje druhý kladný pulz na pin E. Po doručení celého bytu pak následuje výpis příslušného znaku v ASCII tabulce. [6]

Kombinací maticové klávesnice a displeje byl vytvořen následující interface a menu, které odpovídají tomuto zobrazení:



Obr. 1.8: Struktura menu

Ovládání pomocí displeje a klávesnice nebylo zamýšleno jako primární obslužný systém. Lze jej do jisté míry takto využít, ovšem není to příliš komfortní. Slouží v případech, kdy uživatel nemá přístup k ovládací stránce a chce celý systém uvést bezpečně mimo provoz. Primární obslužný systém jsou internetové stránky, na kterých lze pohodlně ovládat celý systém.

1.5 Periferie

1.5.1 Modul reálného času (RTC)

Jednou z nevýhod Raspberry Pi je, že nemá vlastní modul reálného času. Pokud tedy není připojeno k internetu, nezná Raspberry Pi aktuální čas. Při připojení k internetu, lze nastavit nějaký časový interval, při kterém si Raspberry Pi srovná čas a při použití CRONu, to lze nastavit i při spuštění. Problém však nastává, když je Raspberry Pi příliš vytížené, v tu chvíli dochází ke zpoždování. Všechny tyto problémy byly vyřešeny pomocí obvodu PCF8563. Je to malý hodinový obvod, který komunikuje pomocí I2C sběrnice a je napájen plochou baterií, která udržuje obvod v chodu po odpojení napájení. Obvod si udržuje stále hodnotu aktuálního času a na požádání například nějaké aplikace hodnotu vypíše. Při instalaci modulu bylo nutné nainstalovat ovladač. K tomuto účelu byl použit ovladač naprogramovaný v jazyku Python od vývojáře jménem Milan Krúpa, který jej nabízí volně ke stažení na svých webových stránkách.[8]

Ukázka 1.10: Instalace RTC8563.

```
wget http://www.astromik.org/raspi/rtc8563.py
sudo python /home/pi/rtc8563.py -i
```

Po nainstalování je nutné si nastavit přesný čas na Raspberry Pi, protože je nejdříve třeba zapsat přesnou hodnotu aktuálního času, do RTC modulu, z které pak bude vycházet. Po jeho nastavení je třeba zadat příkaz pro zapsání času do RTC.

Ukázka 1.11: Zápis času do RTC.

```
sudo python /home/pi/rtc8563.py -pi2rtc
```

Načtení aktuálního času z RTC, například při zapnutí Raspberry se provede následujícím příkazem.

Ukázka 1.12: Výpis času z RTC.

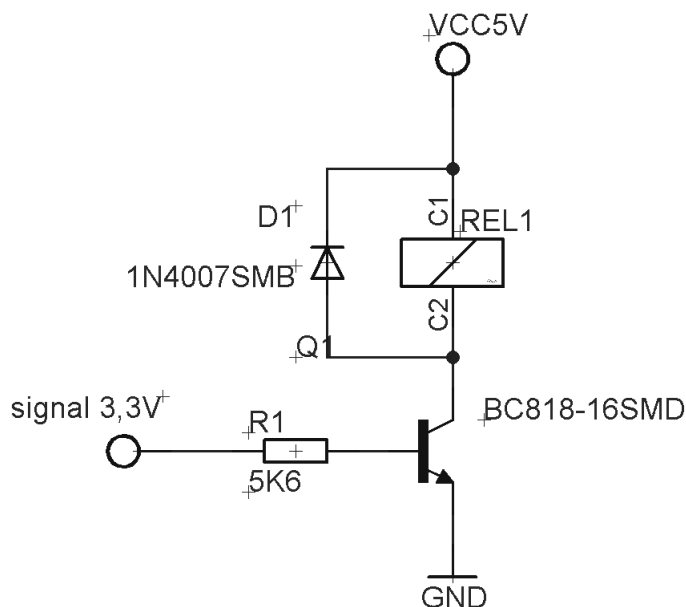
```
sudo python /home/pi/rtc8563.py -rtc2pi
```

Pomocí těchto dvou příkazů lze vytvořit skript, který po naběhnutí systému načte hodnotu času. Při detekci připojení internetu je pak možné modul seřadit. Jelikož se jedná o velmi levný modul reálného času, dochází u něj k odchýlkám. Při testovacím měření to bylo zhruba 3s za 24h. Tato odchylka je závislá na okolní teplotě.

1.5.2 Relé spínače

Jednou z nejdůležitějších vlastností celého zařízení je schopnost zapnout nebo vypnout síťově napájené domácí nebo světelné okruhy. K tomuto účelu bylo nutné do expanzní desky zakomponovat množství ovládaných relé nebo možnost připojení externího. Relé je tedy základní splnitelnou periferií, která má bezpočet využití. Například jimi lze ovládat již zmíněné osvětlení, spínat elektromotory, domácí spotřebiče nebo například různá zahradní čerpadla atd.

Při výběru relé bylo nutno dbát na řadu vstupních parametrů. Bylo nutné, aby relé bylo schopné sepnout stejnosměrným napětím, tedy aby na vstupu obsahovalo DC cívku. Dalším parametrem byl fakt, že nejvyšší napětí na desce je 5 V. Tedy muselo stačit 5 V na ovládání řídicí cívky. A nakonec výstup musel být schopen sepnout síťové napětí 230 V. Zvolil jsem relé RELRAS0515, které splnilo všechny parametry.



Obr. 1.9: Schéma zapojení relé

Protože všechny řídicí piny při log „1“ mají hodnotu napětí 3,3 V bylo nutné spínat relé pomocí tranzistoru. Zapojení jsem zvolil následující. Tedy Báze tranzistoru je připojena k GPIO výstupu expandéru MCP23017. Na R1 vzniká úbytek napětí, který zajistí bezpečné sepnutí tranzistoru. Při sepnutí tranzistoru, pak dojde k sepnutí cívky, která přepne relé. Dioda D1 zabraňuje proražení tranzistoru. Cívka relé má totiž kromě elektrického odporu ještě indukčnost, na které při zavření tranzistoru vzniká napěťová špička.

Proud nutný k sepnutí relé:

$$I_c = 72 \text{ mA} \quad (1.1)$$

Výpočet proudu báze:

$$I_b = \frac{I_c}{h_{FE}} = \frac{72 \times 10^{-3}}{160} = 0,00045 \text{ A} \quad (1.2)$$

Výpočet napětí na bázi:

$$U_{R_b} = U_{in} - U_{BE} = 3,3 - 0,7 = 2,6 \text{ V} \quad (1.3)$$

Výpočet hodnoty R1:

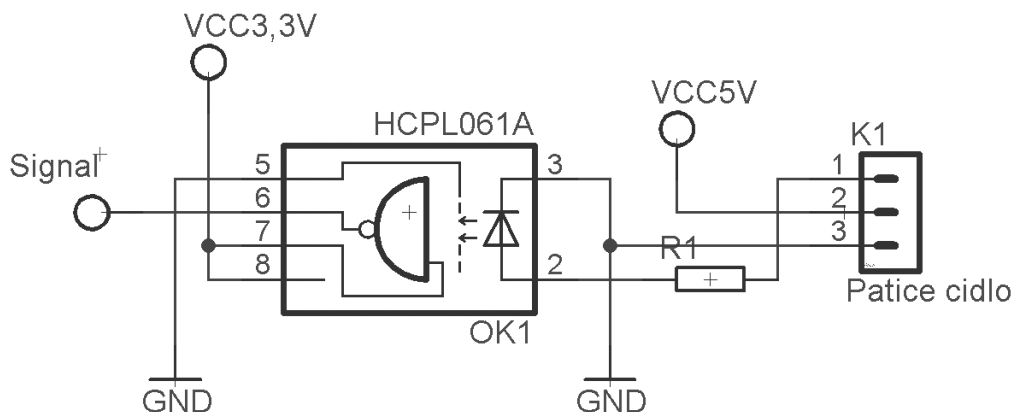
$$R_b = \frac{U_{R_b}}{I_b} = \frac{2,6}{0,00045} = 5,777 \text{ k}\Omega \quad (1.4)$$

1.5.3 Teplotní čidlo

Dalším důležitým prvkem je možnost snímání teploty. Jako čidlo teploty bylo použito SMT160. Jde o digitální čidlo teploty, které pracuje na principu změny střídy obdélníkového signálu. Frekvence pulsů je 3 kHz. Díky tomuto principu stačí k odečtení teploty pouze jeden GPIO vstup. Jak je již řečeno v úvodu, při využití osmi expandérů MCP23017 je možné připojit až 128 teplotních čidel, když zanedbáme všechny ostatní periferie. Pro výpočet teploty ze střídy platí následující vztah:

$$\frac{T_1}{T} = 0,320 + 0,00470 * t \quad (1.5)$$

Nevýhodou čidla je, že je nutné jej napájet 5 V a tak je nutné jej od obvodu oddělit. K tomuto účelu jsem využil optočlen s hradlem obvod 6N137. Nevýhodou je invertovaný výstupní signál z optočlenu a bylo třeba s tím počítat při psaní funkce pro vyčítání teploty. [9]



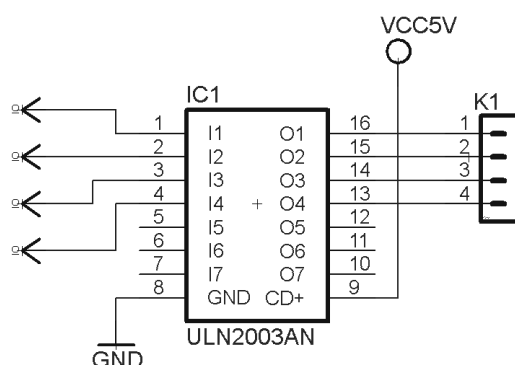
Obr. 1.10: Schéma zapojení teplotního čidla

Na obr: 1.10 je schéma zapojení teplotního čidla. Funkce je následující: teplotní čidlo má tři vývody, napájení, výstup pro střihu a zem. Výstup je připojen k diodě v optočlenu. Ta předává obdélníkové pulsy hradlu, to signál invertuje a dále posílá signál o amplitudě 3,3 V, který je přiveden na vstupní pin MCP23017. Výhodou zapojení je oddělení signálových GPIO pinů od čidla a tedy možnost čidlo prodloužit vodiči. Jak moc lze prodloužit, záleží na typu zvoleného vodiče.

1.5.4 Krokový motor

Další možnou ovladatelnou periferií je krokový motor. Krokový motor lze využít třeba pro ovládání žaluzií, otvírání ventilačních oken. Výhodou krokových motorů je přesné nastavení pootočení. Princip krokového motoru je v postupné změně polarity na cívkách statoru, který roztáčí rotor změnou magnetického pole. Samostatný krokový motor však nelze přímo připojit k výstupům z expanderu, je nutné použít nějaký budič motoru. V tomto případě byl použit obvod ULN2003. Tento obvod je vlastně sedmi-kanálové pole s Darlingtonovými tranzistory. Každý tranzistor je tedy budič s nízkou vstupní impedancí. Každý jednotlivý kanál lze ovládat napětovou úrovní 3,3 V. [10]

Zapojení znázorňuje obr: 1.11. Na I1 až I4 jsou přivedeny GPIO piny. Krokové motorky mají obvykle 5 přívodních vodičů, jeden napájecí a čtyři řídicí. Patice na obrázku zobrazuje připojení právě čtyř řídicích vodičů. Připojení je následující. Modrá – 1, růžová – 2, žlutá – 3, oranžová – 4. Pátý červený vodič je záměrně nepřipojen, neboť jeho příkon krokového motoru je velký, je třeba je připojit k hlavní napájecí



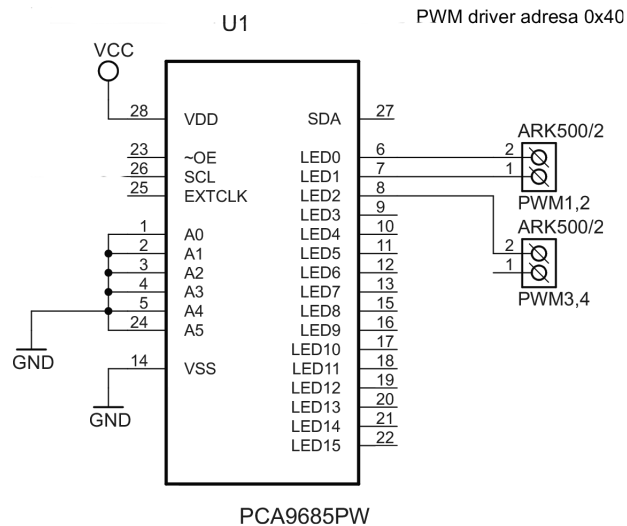
Obr. 1.11: Schéma zapojení krokového motoru

větvi 5 V. Je však třeba brát zřetel na příkon krokového motoru, společně se spotřebou celého zařízení nepřesáhl maximální výkon napájecího zdroje. V praktickém zapojení je třeba připojit napájecí vodič na jiný externí zdroj napájení.

Toto zapojení je však pouze ukázkové, protože není vhodné. Důvodem je velká spotřeba GPIO pinů, tedy 4 na každý motorek. Což je vzhledem nutnosti ovládání dalších zařízení příliš. V praxi je tedy lepší nahradit krokový motorek, tam kde je to možné, servomotorem, který lze ovládat pouze jedním GPIO výstupem. Nebo navrhnout jiné zapojení.

1.5.5 PWM řízení

Pulzně šířková modulace, je vlastně přenos analogového signálu pomocí střídavy. Je to způsob jak pomocí digitálního výstupu, něco plynule regulovat. To lze použít například k plynulé regulaci osvětlení nebo postupnému rozsvícení či zhasnutí světla. Ideální je využít PWM řízení k regulaci svitu led diodových pásek, pomocí MOS-FET transistoru. Další možností využití, je ovládání servomotorů. Pro PWM řízení byl použit obvod PCA9685. Jedná se o I/O expandér podobně jako je tomu u MCP23017, avšak primárně slouží k řízení svitu led-diod. Obvod regulaci svitu obstarává právě pomocí PWM. Jedná se tak o vhodný PWM driver, který má ještě navíc výhodu v tom, že pro připojení nepotřebuje žádný z GPIO pinů. Je připojen obdobně jako MCP23017, pomocí I2C sběrnice, připojením SDA a SCL. Rozdíl je pouze v rozsahu adres. Obvod může nabývat adres v rozsahu 0x60 až 0x67. To znamená, že celé zařízení lze rozšířit o dalších 7 stejných obvodů, což nabízí teoretických 105 samostatných PWM regulací, při 15 výstupech na obvod. Obvod má jednu nevýhodu, obsahuje pouze jeden frekvenční nastavitelný generátor. To znamená, že



Obr. 1.12: PWM driver

lze generovat PWM signál na všech výstupech obvodu pouze o jedné nastavené frekvenci v jednom okamžiku. Nelze tedy nastavit dva PWM signály o dvou různých frekvencích. Nastavení frekvence se provádí nastavením „prescaleru“ neboli děličky, což je konstanta pomocí které se dělí frekvence vnitřního generátoru (25 MHz). Výpočet byl proveden podle následujícího vztahu.

Výpočet konstanty děličky:

$$P_k = \left(\frac{f_g}{f_p} \right) + 1 \quad (1.6)$$

Celý obvod je opět napájen 3,3 V.[11] Tímto způsobem lze tedy generovat PWM signál. V zapojení jsou vyvedeny tři PWM výstupy, lze však k jednomu expandéru vyvést až 16 výstupů. Další výhodou obvodu je, že zbylé porty, které nejsou zapojeny lze využít jako standardní GPIO piny. Každý z pinu lze nastavit do 4 různých stavů:

1. Výstup logická „1“ (3,3 V)
2. Výstup logická „0“ (GND)
3. Vstup rozpoznání logické „1“ a „0“ (3,3 V a 0 V)
4. Nastavení PWM signálu (frekvence dle prescaleru, střída)

Tab. 1.5: Seznam registrů a jejich parametrů

| Číslo registru | Funkce |
|----------------|--|
| 0xFE | Konstanta (prescaler), nastavení frekvence |
| 0x00 | Start, stop, restart generátoru |
| 0x01 | Nastavení logiky pwm signálu („H“ = Vcc ; „L“ = GND) |
| 0x06 - 0x45 | Nastavení střídání, na jednotlivých pinech |

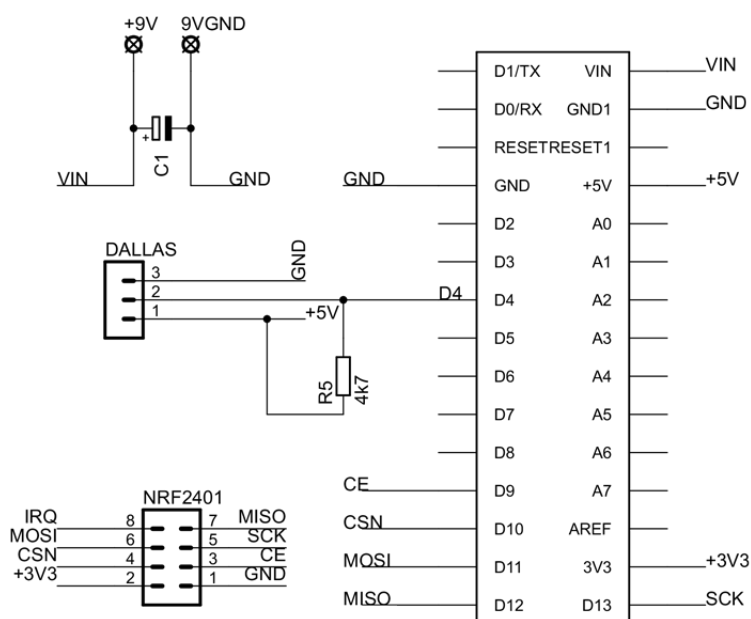
Princip ovládání, je podobný jako u MCP23017. Rozdíl je zde v adresování jednotlivých ovládacích registrů, které jsou popsány v tabulce „Register summary“, obsažené v datasheetu obvodu. Hodnoty lze zadávat v hexadecimální nebo desítkové soustavě. Nastavení střídání se provádí, pomocí 4 registrů pro každý pin. Do těchto registrů je zadána hodnota v rozsahu 0 - 4096. Tento rozsah představuje jednu periodu frekvence děličky. Pro nastavení 50% střídání je tedy třeba nastavit signál log „1“ od 0 - 2047 a log „0“ 2048 - 4096. Poměr daného času pro jednotlivé logické úrovně je stejný, což odpovídá střídání 50%. Příklad příkazu pro generování signálu o frekvenci 50Hz se střídáním 50% na pinu Led 0.

Ukázka 1.13: PWM řízení.

```
fg=25000000          # frekvence vnitřního generátoru
fp = 50              # požadovaná frekvence PWM (50Hz)
p = int((fg /fp) / 4096)+1  # výpočet konstanty prescaleru
bus.write_byte_data(0x40,0x00,0x90) # restart generátoru
bus.write_byte_data(0x40,0xFE,p)   # předání konstanty prescaleru
bus.write_byte_data(0x40,0x01,0x04) # nastavení logiky výstupu
bus.write_byte_data(0x40,0x06,0x00) # čas délky log „0“ v čase T(0)
bus.write_byte_data(0x40,0x07,0x00) # přechod z „0“ na „1“ v čase T(0)
bus.write_byte_data(0x40,0x08,0x7FF) # čas délky z „1“
bus.write_byte_data(0x40,0x09,0x800) # přechod z "1" na "0"
bus.write_byte_data(0x40,0x0A,0x00) # spuštění generátoru
```

2 BEZDRÁTOVÁ ČÁST

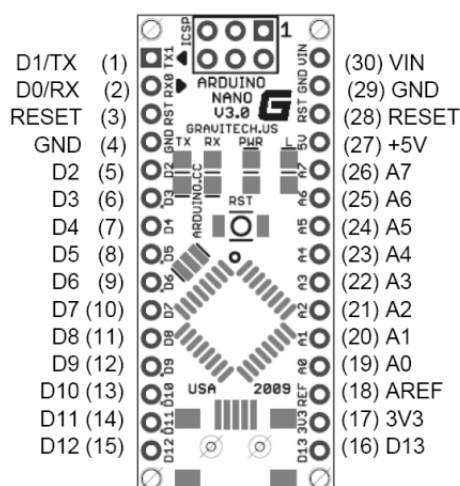
V bezdrátové části bylo původním cílem navrhnout nízkoenergetický bezdrátový modul, který bude schopen obousměrné komunikace a bude mít možnost připojení libovolných periférií, jak senzorů neelektrických veličin tak ovládacích prvků. K tomuto účelu byla použita platforma Arduino založená na mikrokontroléru ATmega. Tato platforma je často zmiňovaná jako konkurence právě pro Raspberry Pi, protože obě dvě platformy nabízejí množství podobných ovládaných periférií (I/O, PWM, komunikační sběrnice). Proto spojení těchto dvou platform skýtá velké množství možností a využití. Další částí je bezdrátový přenos, pro tento účel byl vybrán modul pro bezdrátovou komunikaci nRF24L01. Který vyniká zejména svojí nízkou energetickou náročností. Poslední součástí obsaženou v bezdrátové části je teplotní čidlo Dallas 18b20. Bezdrátová část je napájena akumulátorem a je usazena v plastovém krabici, která má z přední strany otvory pro přístup vzduchu (teplotní čidlo).



Obr. 2.1: Schéma bezdrátové části

2.1 Arduino Nano 3.0

Jak už bylo uvedeno, řídicím prvkem bezdrátové části je open-source platforma, založená na mikrokontroléru ATmega od firmy ATmel. Tento vývojový modul je nejmenším z celé rodiny vývojových desek Arduino. Jeho základem je mikrokontroler Atmel ATmega328. Celý tento modul obsahuje několik základních periférií, které umožňují jeho samostatnou činnost. Modul obsahuje 14 digitálních vstupně/výstupních pinů, z nichž 6 lze využít jako PWM výstupy, dále 6 analogových vstupů, 16 MHz krystal a mini USB konektor (napájení, programování), který slouží i k naprogramování. Celý modul pracuje s napětím 5 V. K napájení modulu je třeba napětí 7 - 12 V, stačí tedy 9 V plochá baterie. Maximální proud, který je schopen dodat jeden I/O pin je 40 mA. Dále disponuje SPI a I2C sběrnici.[12]



Obr. 2.2: Arduino Nano 3.0

2.1.1 Programování

K naprogramování Arduina bylo třeba pouze připojit modul k počítači pomocí mini USB konektoru. A následně nainstalovat ovladač z oficiálních stránek arduino.cc, konkrétně ovladač Arduino 1.0.5. Tento stažený soubor obsahuje kromě ovladače i nástroj k naprogramování, jehož součástí je minimalistické vývojové prostředí. Program obsahuje i sadu užitečných tutoriálů, ze kterých bylo vycházeno při tvorbě ovládacího programu pro bezdrátovou část.

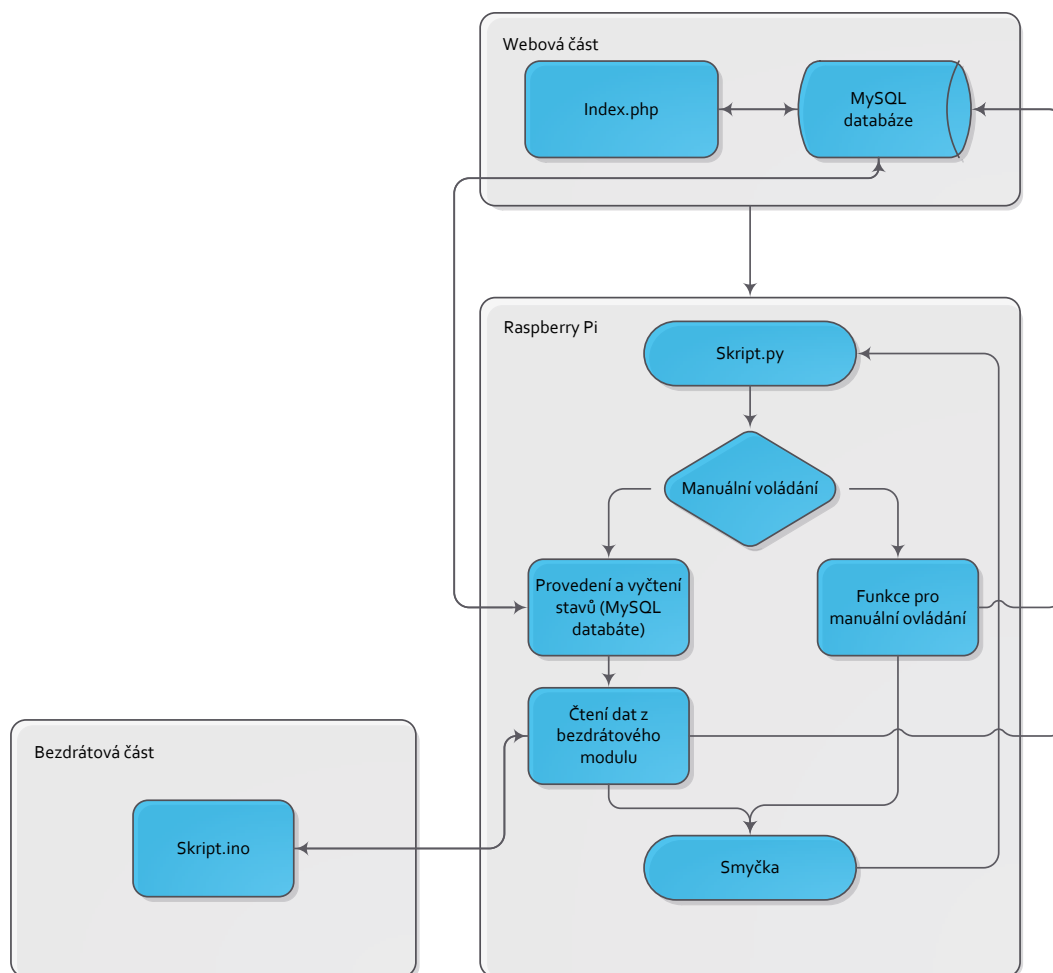
2.2 Bezdrátový modul nRF24L01

Je to nízkoenergetický modul, který obsahuje transceiver, tedy přijímač i vysílač. Dovoluje obousměrný datový tok typu half-duplex. Modul pracuje ve volném frekvenčním pásmu 2,4 GHz. Modul, je připojen k Arduinu i k Raspberry Pi přes sběrnici SPI a k napájení 3,3 V. Ve volném prostoru je jeho dosah bez připojení externí antény zhruba 100 m. V prostoru, kde je více překážek, například v budově, klesá jeho dosah téměř na třetinu. Data jsou před přenosem zabalena do packetů (Enhanced ShockBurst), které kromě zasílaných dat obsahují kontrolní součet a další režie, které slouží k bezpečnému doručení. Každý doručенý paket je potvrzen přijímací stranou. Bezpečnou komunikaci zabezpečuje samotný modul a tak nebylo nutné se touto problematikou zabývat při programování Arduina.

2.3 Teplotní čidlo Dallas 18b20

Jedná se o digitální teplotní čidlo s rozsahem od -55°C do 125°C přesností $\pm 0,5^{\circ}\text{C}$. Čidlo podporuje tzv. 1-Wire komunikaci, to znamená, že k jednomu pinu například právě Arduina lze připojit více čidel najednou. Každé čidlo obsahuje své unikátní 64-bitové číslo, toto číslo slouží jako kódové číslo k rozlišení od ostatních. Čidlo narozdíl od střídivého, vrací přesnou hodnotu teploty, není třeba hodnotu dopočítávat. Čidlo lze nastavit do 4 rozlišení přesnosti.

3 SOFTWARE



Obr. 3.1: Vývojový diagram celého zařízení

Na diagramu obr: 3.1, lze vidět jak celé zařízení funguje. Jak již bylo uvedeno hlavní výpočetní část je prováděna v Raspberry Pi. To obstarává ovládání jednotlivých periferií, vyčítání dat ze senzorů a komunikaci s bezdrátovým modulem. Nad Raspberry Pi stojí externí webový server, na kterém je umístěno webové rozhraní a MySQL databáze. Tato databáze obsahuje stavová data, hodnoty senzorů a časové hodnoty. Jednotlivé hodnoty v tabulce má právo měnit webové rozhraní a Raspberry Pi. Pod Raspberry Pi je bezdrátová část, která je připojena pomocí bezdrátového modulu.

3.1 Webové rozhraní a MySQL databáze

Tyto dvě části v obrázku: 3.1, zahrnuté v položce webové části, tvoří klíčovou roli v celkovém ovládní. Webové rozhraní je aplikace na http severu naprogramovaná v jazyku PHP, plus několik jQuery Pluginů. Dohromady tvoří uživatelský interface, jež slouží k ovládní a signalizaci zařízení. Webové rozhraní je dostupné na adrese `http://31.31.76.16/machacek/`. Webový interface je rozdělen na dvě kategorie, „Statusy,“ a „Ovládní“ (přílohy C.1 a C.2). Stránka Statusy obsahuje informace o jednotlivých částech zařízení. Stavové informace o výstupech, naměřené hodnoty teplotních čidel a aktuální nastavení dalších položek. V druhé kategorii je vlastní ovládní systému, na této stránce lze ovládat každou periférii zvlášť pomocí příslušných polí. Webové rozhraní všechna zobrazená data vyčítá z MySQL databáze, která slouží jako prostředník pro výměnu dat mezi spuštěným skriptem v Raspberry Pi a Webovým rozhraním. Pokud tedy uživatel provede nějakou změnu v nastavení, tak se tato změna pomocí MySQL příkazů zapíše do příslušné tabulky. MySQL server je spuštěn na stejné adrese jako webový server. To že je webové rozhraní a MySQL server spuštěn na externím serveru, je pouze ukázkou a je možné tyto dvě služby spustit i přímo na Raspberry Pi.

3.1.1 Umístění

Výběr externího serveru obsahuje několik následujících výhod, ale i nevýhod. Výhodou externího serveru je, že není třeba se starat o tuto část, protože stačí využít některou z nabídek webového hostingu a webové stránky uploadovat a vytvořit potřebné tabulky v MySQL databázi. Výhodou je taky bezproblémový přístup odkudkoliv. Nevýhodou je nutnost stránky zabezpečit přístupovým heslem. Další nevýhodou je nutnost, aby Raspberry Pi bylo neustále připojeno k internetu. Rovněž výpadky internetu způsobí zastavení činnosti. Pokud by byly stránky spuštěny i s MySQL serverem přímo na Raspberry Pi, vyplývala by z toho výhoda nezávislosti na internetovém připojení. Ovšem pro zajištění přístupu k webovému rozhraní odkudkoliv z internetu, je nutné aby Raspberry Pi bylo připojeno k síťovému prvku, jež má přiřazenou veřejnou IP adresu. Nevýhodou je možné bezpečnostní riziko přímého připojení k internetu a hardwarové vytížení Raspberry Pi.

3.2 Raspberry Pi

Na Raspberry Pi je spuštěn `Skript.py`. Tento skript je tvořen nekonečnou programovou smyčkou. Struktura programu je následující. Na začátku programu jsou nejdříve definovány všechny periférie a jejich adresy. Tedy označení jednotlivých pinů

expanderů, jestli se má jednat o vstup, výstup či jinak. Po této inicializaci následuje podmínka dostupnosti MySQL databáze. Pokud je databáze dostupná, začne program provádět nekonečnou smyčku, pokud ne vyčkává na její dostupnost. Nekonečná smyčka je tvořena řadou funkcí, které se postupně dotazují na stavy, nebo je naopak zapisují, v jednotlivých tabulkách MySQL databáze. Tyto úpravy jsou prováděny pomocí knihovny `ClassMySQL.py`, jež obsahuje metody pro editaci MySQL databáze. V třídě `ClassMySQL.py` jsou tyto metody prováděny pomocí standardních MySQL příkazů („Select“, „Read“, „Update“, „Insert“, „Delete“). Součástí smyčky je podmínka pro manuální ovládání. Tato podmínka je splněna po přidržení příslušného tlačítka na klávesnici. Splněním této podmínky program skočí do další nekonečné podmínky, která obstarává ovládání klávesnice a displeje. V tomto režimu lze měnit údaje v MySQL databázi. Po ukončení manuálního režimu, se v nekonečné smyčce provedou předchozí změny.

3.3 Bezdrátový modul

Bezdrátová část respektive Arduino bylo naprogramováno v Jazyce C++. Důvod užití C++ namísto již použitého jazyka Pythonu, byla absence knihoven pro ovládání modul nRF24L01. Knihovny pro ovládání modulu jsou k dispozici pouze v jazyku C++. Komunikace mezi Raspberry Pi a Arduinem probíhá následovně. Raspberry Pi řídí celou komunikaci, která probíhá systémem dotaz/odpověď.

Raspberry Pi nejdříve pošle zprávu HELLO, pokud je Arduino připraveno odpovědět, odpoví zprávou READY, v opačném případě neodpoví a Raspberry Pi za určenou dobu opět vyšle dotaz. Pokud Raspberry Pi dostane odpověď READY, následuje ze strany Raspberry Pi již konkrétní dotaz na hodnotu, kterou posílá teplotní čidlo. Arduino zašle požadovaná data, následuje ukončení spojení zprávou BEY nejdříve od Raspberry a poté potvrzení od Arduina. Po této sekvenci opět čeká Arduino na zprávu HELLO. Raspberry posléze zapíše hodnotu teploty, kterou přijal do MySQL databáze.

4 REALIZACE A MOŽNOSTI ROZŠÍŘENÍ

4.1 Realizace

Zařízení je složeno z jednotlivých samostatných zapojení, která byla spojena dohromady. Nejdříve probíhal návrh expanzní desky pro GPIO port. Výsledné schéma zobrazeno v příloze: A.1, bylo poskládáno s dílčími zapojení, která byla realizována zvlášť a později dohromady na nepájivém poli. Po vyzkoušení funkčnosti všech zapojení a vyladění jejich chyb, byla navržena deska plošných spojů (příloha: A.1 a A.2). Tato deska pak byla osazena součástkami a oživena. Oživení probíhalo pomocí malých skriptů, které vždy ovládaly pouze určitou část právě testovaného zapojení. S touto deskou byl následně vytvořen hlavní ovládací skript, kterým byly předchozí menší skripty spojeny do jednoho. Po odladění všech nalezených, problémů byla expanzní deska společně s Raspberry Pi umístěna do univerzální konstrukční krabičky. Následně byla vytvořena bezdrátová část dle schématu na obrázku 2.2, zapojení bylo opět nejdříve provedeno na nepájivém poli a následně byla vytvořena deska plošných spojů, příloha (příloha: B.1 a B.2). Hotový bezdrátový modul byl opět umístěn do univerzální konstrukční krabičky. Nakonec bylo vytvořeno webové rozhraní, ukázka v příloze: C.1 a C.2. Protože bylo třeba implementovat ovládání na webovou stránku, bylo posléze nutné přepracovat znovu celý skript v Raspberry Pi. Předchozí verze totiž ukládala aktuální nastavení a všechny statové hodnoty do textového souboru. A tak byl celý skript přepsán pro komunikaci a ukládání na MySQL server.

4.2 Rozšíření

Vhledem k charakteru práce, nelze brát hotové zařízení jako jednotku schopnou ovládat například celý dům. Je to dáno počtem periférií a celkovým rozpočtem. Lze tedy práci brát jako ukázkou navržených možností řízení a měření, byť zastoupených v malém počtu. Cele zařízení, lze velmi snadno rozšířit a aplikovat na konkrétní objekt, či dle potřeb zákazníka. Rozšířit jde každou periférii, některé až do teoretických stovek kusů (relé, spínače, vstupy, PWM, teplotní čidla). Další prostor pak nabízí bezdrátový modul. Jak bylo uvedeno, Arduino obsahuje také datovou sběrnici I2C, stejně jako Raspberry Pi. Je tedy teoreticky možné při naprogramování příslušných funkcí připojit expanzní desku pro Raspberry k Arduino.

Pro rozšíření expanzní desky například o sadu 8 relé, je třeba dodržet následující postup. Jelikož se jedná o 8 relé, je třeba využít jeden celý port expanderu MCP23017. K expanderu následně připojit relé, dle schématu zapojení (1.9). Pak je třeba, upravit `Skript.py`, kde stačí pouze pomocí copy/paste zkopírovat funkci

pro relé `relvyp()`, `relzap()`, přidat příslušný index. Dále je třeba upravit databázový příkaz ve větvi manuálního ovládání. Poté je třeba přidat v MySQL do databáze `rele` řádky, pro každé nové relé. Nakonec stačí upravit webové stránky. Přidat PHP funkce pro úpravu a vyčítání dat V MySQL databázi, daného řádku, opět pomocí copy/paste, dle zavedené logiky a propojit je s elementy na stránkách.

5 ZÁVĚR

Cílem této bakalářské práce bylo navrhnout koncepci řízení inteligentního domu, založenou na platformě Raspberry Pi. Obsahem bakalářské práce bylo především seznámení se s principy a možnostmi řízení Raspberry Pi, a na základě těchto znalostí navrhnout rozšiřující kit pro GPIO konektor a bezdrátový modul. Expanzní deskou v této podobě lze ovládat osvětlení, spínat spotřebiče, ovládat krokové motorky, servo-motorky. Celý systém lze ovládat pomocí dvouřádkového displeje a maticové klávesnice, nebo s využitím webového rozhraní. Celé stávající zařízení je však nutné chápat pouze jako testovací. Pro jeho praktické využití by bylo nutné celé zařízení značně rozšířit, proto při návrhu bylo dbáno právě na snadnou rozšiřitelnost. Všechny výše popsané principy tedy lze aplikovat. Stejně je to i s programovým řízením, kde možnost budoucího rozšíření byla hlavním parametrem.

Hlavním přínosem bakalářské práce, jsou především získané znalosti při řešení této problematiky. Hlavním stěžejním bodem bylo propojení tří různých systémů (webových stránek, Raspberry Pi, Arduina), které byly programovány každý v jiném jazyce. Dalším přínosem je fakt, že výsledné zařízení dává uživateli možnosti jistého dohledu nad nějakým objektem. Výsledné zařízení však určitě nelze brát jako hotový produkt, neboť zde nejsou vyřešeny některé otázky bezpečnosti a problematika inteligentního domu je obecně velmi rozsáhlá.

LITERATURA

- [1] GIRLING, Gary. *Raspberry Pi: a practical guide to the revolutionary small computer*. Sparkford: Haynes Publishing, 2013, 169 s. ISBN 978-0-85733-295-0.
- [2] *Raspberry Pi GPIO info sheet* [online]. 2013, poslední aktualizace 18.9.2013 [cit.17.12.2013]. Dostupné z URL: <<http://www.corpshadow.com/raspberry-pi-gpio-info-sheet>>.
- [3] *Pokusy s GPIO konektorem* [online]. 2013, poslední aktualizace 4.8.2013 [cit.17.12.2013]. Dostupné z URL: <<http://www.astromik.org/raspi/07.htm>>.
- [4] *Raspberry Pi and the MCP23017 I2C I/O Expander* [online]. 2012, poslední aktualizace 19.5.2012 [cit.15.12.2013]. Dostupné z URL: <<http://nathan.chantrell.net/20120519/raspberry-pi-and-the-mcp23017-i2c-io-expander/>>.
- [5] *Raspberry Pi I/O Expander Board* [online]. 2012, poslední aktualizace 2.6.2012 [cit.15.12.2013]. Dostupné z URL: <<http://nathan.chantrell.net/20120602/raspberry-pi-io-expander-board/>>.
- [6] *Datasheet, ATM1602B* [online]. [cit.21.12.2013]. Dostupné z URL: <<http://www.hebeiltd.com.cn/lcm.datasheet/ATM1602B.pdf>>.
- [7] *RTC on Raspberry Pi* [online]. 2012, poslední aktualizace 6.7.2012 [cit.29.12.2013]. Dostupné z URL: <<http://www.raspberrypi.org/phpBB3/viewtopic.php?t=13710&p=142972>>.
- [8] *Hodiny reálného času (RTC) - PCF8563* [online]. 2014, poslední aktualizace 22.2.2014 [cit.29.3.2014]. Dostupné z URL: <<http://www.astromik.org/raspi/rtc8563.py>>.
- [9] *Datasheet, SMT16030 DIGITAL TEMPERATURE SENSOR* [online]. 2005, poslední aktualizace 23.5.2005 [cit.29.12.2013]. Dostupné z URL: <<http://www.smartec.nl/pdf/DSSMT16030.PDF>>.
- [10] *Datasheet, HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON TRANSISTOR ARRAYS* [online]. 2012, poslední aktualizace 10.2.2013 [cit.29.12.2013]. Dostupné z URL: <<http://www.ti.com/lit/ds/symlink/uln2003a.pdf>>.

- [11] *Datasheet PCA9685* [online]. 2002, poslední aktualizace 16.9.2010 [cit. 29.12.2013]. Dostupné z URL: <<http://www.adafruit.com/datasheets/PCA9685.pdf>>.
- [12] *Arduino Nano 3.0 (ATmega328)* [online]. [cit. 29.12.2013]. Dostupné z URL: <http://www.kitrobot.com.ar/Arduino_Nano.pdf>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

SD Secure Digital

SSH Secure Shell

I2C Inter-Integrated Circuit

DPS Deska plošných spojů

USB Universal Serial Bus

UART Universal Asynchronous Receiver/Transmitter

RTC Real-time clock

SPI Serial peripheral interface

GPIO General-purpose input/output

FTP File transfer protokol

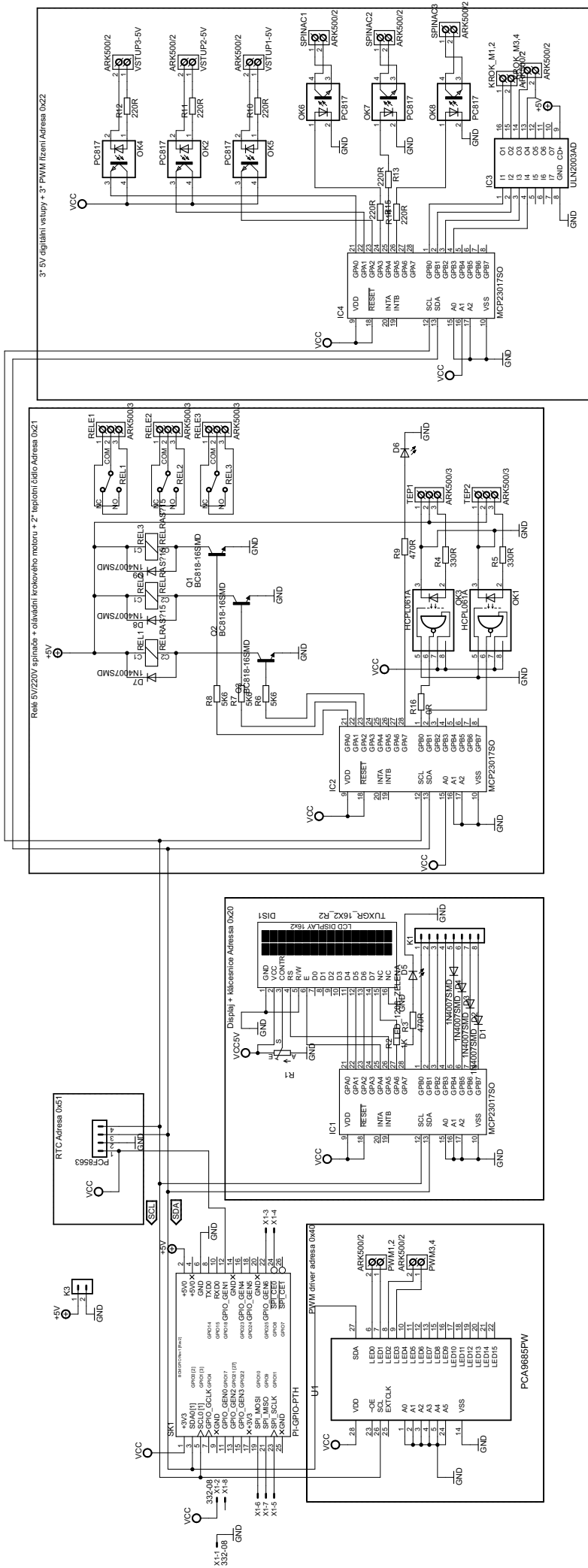
SMD Surface mount device

SEZNAM PŘÍLOH

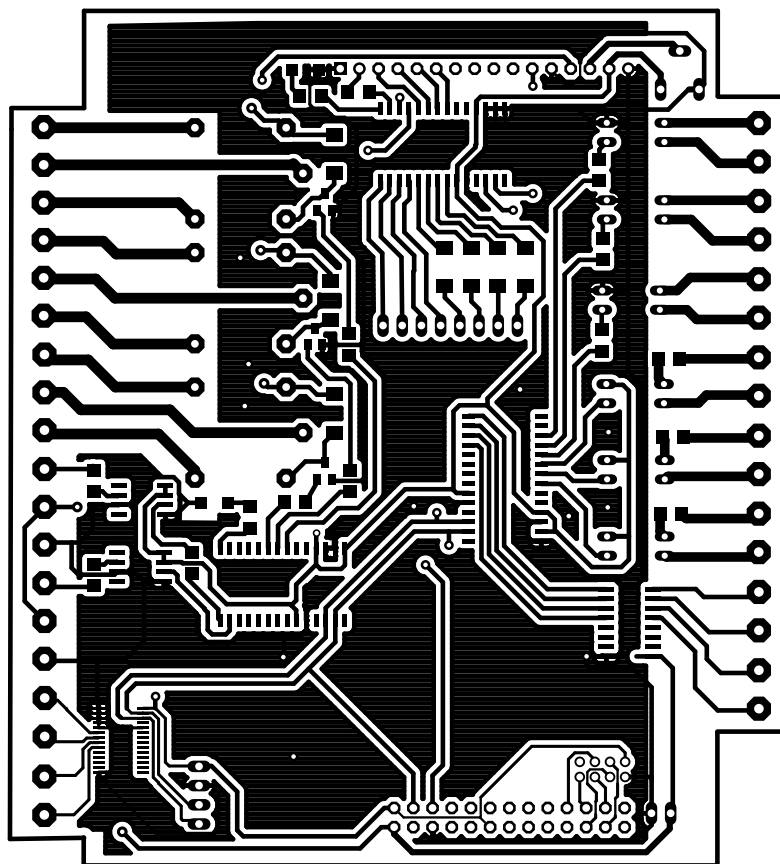
| | |
|------------------------------------|-----------|
| A Rozšiřující Deska | 46 |
| A.1 Schéma zapojení | 46 |
| A.2 Deska plošných spojů | 48 |
| A.3 Seznam součástek | 52 |
| B Bezdrátová část | 53 |
| B.1 Deska plošných spojů | 53 |
| B.2 Seznam součástek | 55 |
| C Ukázka webových stránek | 56 |
| D Návod k instalaci | 58 |
| E Návod k obsluze | 59 |
| F Obsah CD s přílohou | 60 |

A ROZŠIŘUJÍCÍ DESKA

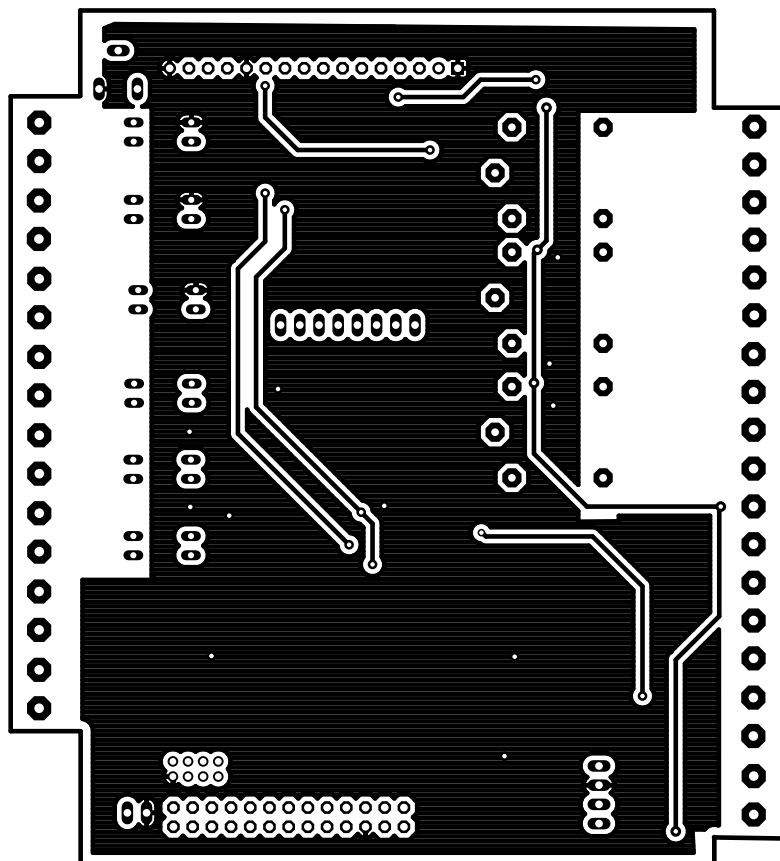
A.1 Schéma zapojení



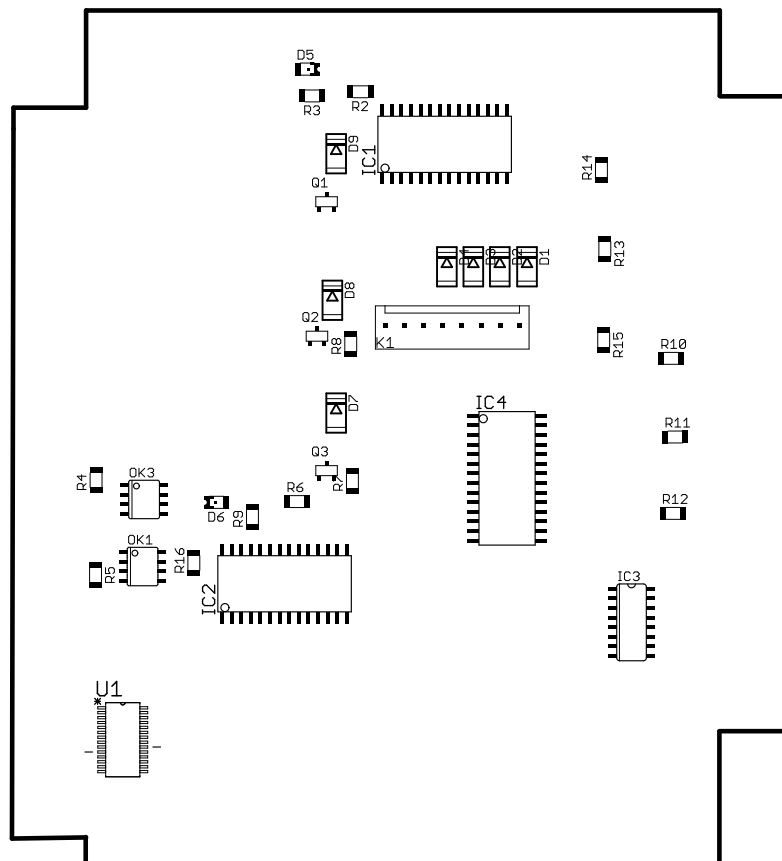
A.2 Deska plošných spojů



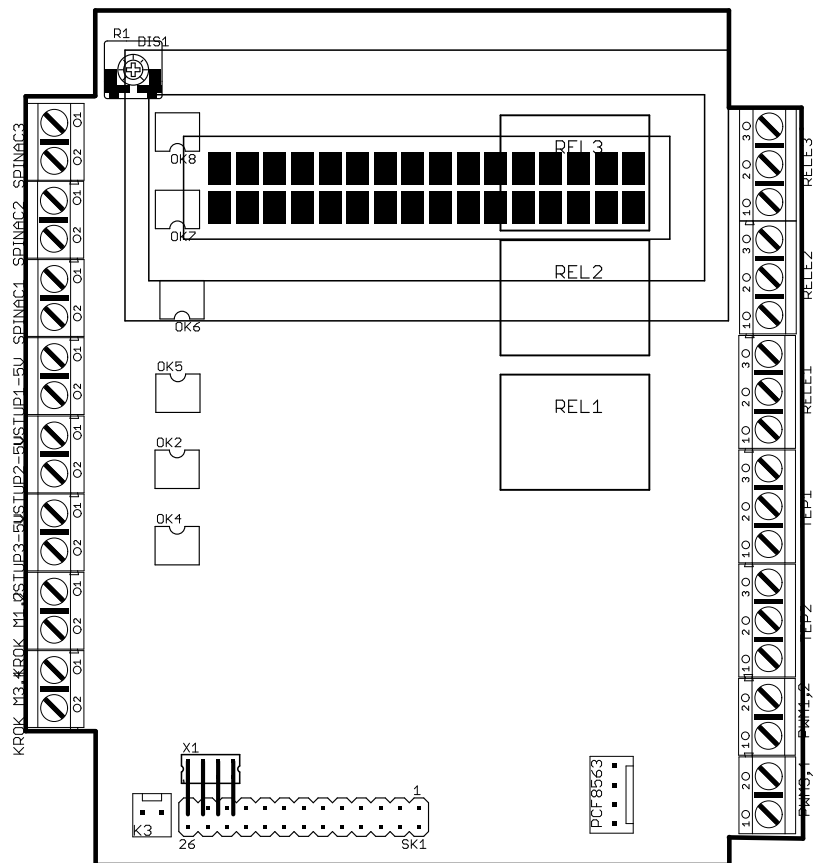
Obr. A.1: Předloha desky plošných spojů spodní stana



Obr. A.2: Předloha desky plošných spojů vrchní stana



Obr. A.3: Osazovací plán spodní stana



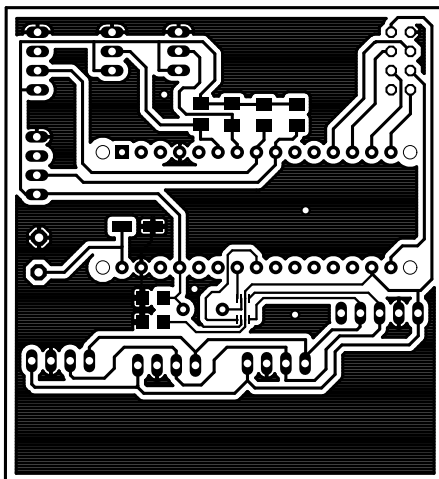
Obr. A.4: Osazovací plán vrchní stana

A.3 Seznam součástek

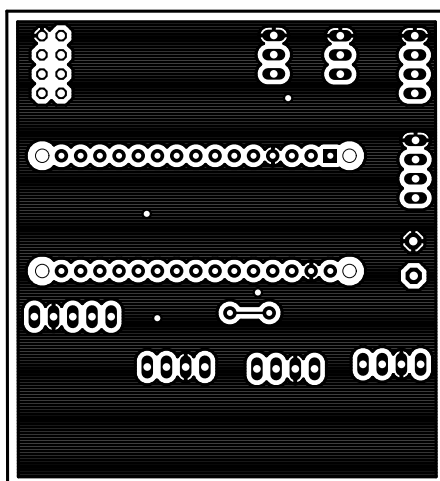
| Počet | Hodnota | Pouzdro | Reference | Popis |
|-------|-----------------|-----------------------|---------------------------------|-------------------|
| 1 | 50K | CA6V | R1 | odporový trimr |
| 2 | zelená | LED 1206 | D6, D5 | Led dioda |
| 1 | | PSH02-02P | k3 | vidlice 2 pin |
| 1 | PCF8563 | PCF8563 | PCF8563 | RTC |
| 1 | 0R | R1206 | R16 | SMD rezistor |
| 1 | | PSH02-08P | K1 | vidlice 8 pin |
| 1 | 1K | R1206 | R2 | SMD rezistor |
| 7 | 1N4007SMD | DO213AB | D1, D2, D3, D4, D7, D8, D9 | SMD dioda |
| 6 | 220R | R1206 | R10, R11, R12, R13, R14, R15 | SMD rezistor |
| 2 | 330R | R1206 | R4, R5 | SMD rezistor |
| 1 | 332-08 | 332-08 | X1 | dutinková vidlice |
| 2 | 470R | R1206 | R3, R9 | SMD rezistor |
| 3 | 5K6 | R1206 | R6, R7, R8 | SMD rezistor |
| 10 | ARK500/2 | ARK500/2 | | Svorkovnice |
| 5 | ARK500/3 | ARK500/3 | | Svorkovnice |
| 3 | BC818- 16SMD | SOT23- BEC | Q1, Q2, Q3 | NPN Transistor |
| 2 | HCPL061A | SOIC08 | OK1, OK3 | Optočlen |
| 3 | MCP23017S | SO28W | IC1, IC2, IC4 | I2C expander |
| 6 | PC817 | DIL04 | OK2, OK4, OK5, OK6, OK7, OK8 | Optočlen |
| 1 | PCA9685PW | SOP65P640X110- 28N | | PWM driver |
| 1 | MLW26G | MLW26G | SK1 | vidlice 2*13 |
| 3 | RELRAS0515 | RELE RAS | REL1, REL2, REL3 | Relé |
| 1 | ATM16*2B | ATM16*2B | DIS1 | I2C LCD displej |
| 1 | ULN2003AD | SO16 | IC3 | Budič k. |

B BEZDRÁTOVÁ ČÁST

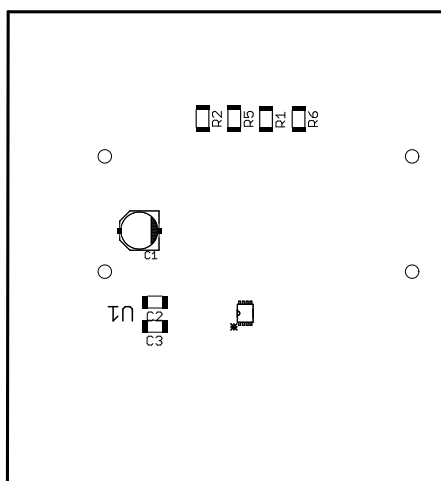
B.1 Deska plošných spojů



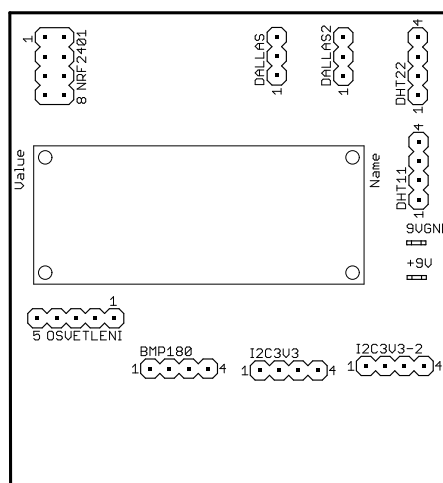
Obr. B.1: Předloha desky plošných spojů spodní stana



Obr. B.2: Předloha desky plošných spojů vrchní stana



Obr. B.3: Osazovací plán spodní stana



Obr. B.4: Osazovací plán vrchní stana

B.2 Seznam součástek

| Počet | Hodnota | Pouzdro | Reference | Popis |
|-------|--------------|---------|-----------------|------------------|
| 2 | 47nF | C1206 | C2, C3 | SMD Konden. |
| 2 | 18b20 | MA03-1 | DALLAS, DALLAS2 | Dutinková lišta |
| 1 | | MA04-2 | NRF2401 | Dutinková lišta |
| 2 | 4k7 | M1206 | R2, R5 | SMD Rezistor |
| 1 | ARDUINO-NANO | PDIP-32 | ARDUINO | Arduino Nano 3.0 |

C UKÁZKA WEBOVÝCH STRÁNEK

Statusy Ovládání

Relé

| Název | Stav | Datum sepnutí | Čas sepnutí | Datum vypnutí | Čas vypnutí | Zbývá do |
|-------|---|---|------------------------------------|---|------------------------------------|----------|
| rele1 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | offline |
| rele2 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | offline |
| rele3 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | offline |

[Nastav](#)

Spínače

| Název | Stav | Datum sepnutí | Čas sepnutí | Datum vypnutí | Čas vypnutí | Zbývá do |
|---------|---|---|------------------------------------|---|------------------------------------|----------|
| Spinac1 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | offline |
| Spinac2 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | offline |
| Spinac3 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | <input type="text" value="31.05.2014"/> | <input type="text" value="--:--"/> | offline |

[Nastav](#)

PWM

| Název | Stav | Slider | Hodnota |
|-------|---|---------------------------------|---------|
| pwm1 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="range" value="0"/> | 0 % |
| pwm2 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="range" value="0"/> | 0 % |
| pwm3 | <input type="radio"/> On <input checked="" type="radio"/> Off | <input type="range" value="0"/> | 0 % |

[Nastav](#)

Žaluzie

| Název | Slider | Hodnota |
|---------|--|---------|
| Žaluzie | <input type="range" value="0"/> roztáheno zataženo | 0 % |

Aktuální čas serveru 22:08:00 31.05.2014

[Nastav](#)

Obr. C.1: Záložka ovládání

Statusy

Ovládaní

Teploty



Statusy ovládacích prvků

Relé

| Název | Status | Čas sepnutí | Čas vypnutí |
|-------|---------|-------------|-------------|
| rele1 | offline | N/A | N/A |
| rele2 | offline | N/A | N/A |
| rele3 | offline | N/A | N/A |

Spínače

| Název | Status | Čas sepnutí | Čas vypnutí |
|---------|---------|-------------|-------------|
| Spinac1 | offline | N/A | N/A |
| Spinac2 | offline | N/A | N/A |
| Spinac3 | offline | N/A | N/A |

Vstupy

| Název | Status |
|--------|---------|
| ttlin1 | offline |
| ttlin2 | offline |
| ttlin3 | offline |

PWM

| Název | Status | Hodnota |
|-------|---------|---------|
| pwm1 | offline | N/A |
| pwm2 | offline | N/A |
| pwm3 | offline | N/A |

Žaluzie

| Název | Hodnota |
|---------|---------|
| Žaluzie | 0 % |

0 % - roztaženo | 100 % zataženo

Obr. C.2: Záložka statusy

D NÁVOD K INSTALACI

1. Nainstalovat Os na Raspbery Pi, dle popisu v kapitole 1.1.2.
2. Provést prvotní nastavení dle 1.1.2.
3. Nainstalovat ovladače pro ovládání GPIO portu, I2C podporu python, dle popisu v kapitole 1.1.5.
4. Zkopírovat složku Raspberrypython, která je obsažena v příloze na CD .
5. Připojit expanzní desku k Raspberry Pi a v terminálu pomocí příkazu 1.6 zjistit zda, se hlásí všechna zařízení připojena pomocí I2C.
6. Nainstalovat MySQLdb knihovny příkazem:

```
sudo apt-get install python-mysqldb
```
7. Nainstalovat knihovny pro bezdrátovou komunikaci:
 - `mkdir gitwork`
 - `cd gitwork`
 - `git clone https://github.com/stanleyseow/RF24.git`
 - `cd RF24/librf24-rpi/librf24`
 - `make`
 - `sudo make install`
 - `sudo ldconfig -v | grep librf`
 - `g++ -Wall -Ofast -mfpu=vfp -mfloat-abi=hard -march=armv6zk -mtune=arm1176jzfs -L/lib/ -lrf24 /home/pi/machacek/nfr.cpp -o /home/pi/machacek/nrfdir`
8. Připojit Raspberry Pi k internetu.
9. Spustit Skript.py, který se nachází ve složce Raspberrypython,

```
python Skript.py.
```
10. Po jeho spuštění, by se měli začít vypisovat stavy aktuální konfigurace, uložené v MySQL databázi.
11. Pokud systém vypíše chybu: "nastala chyba při připojení k databázi", zkontrolujte připojení k internetu příkazem `ifconfig`.

E NÁVOD K OBSLUZE

1. Nejdříve připojte zařízení pro ovládání, k příslušným svorkovnicím.
2. Poté připojte zařízení k elektrické síti, a k internetu pomocí ethernetového kabelu.
3. Po připojení stiskněte kolíbkový vypínač na boku krabičky, do polohy zapnuto.
4. Po správném naběhnutí, by se měl rozsvítit displej a zobrazit se nápis v provozu.
5. Následně je možné přejít na webové stránky:
`http://31.31.76.16/machacek/`
6. Zde je možné ovládat připojené periferie.
7. Při přidržení znaku * na klávesnici, je možné aktivovat manuální mod, ve které lze v omezené míře provádět nastavení preferíí.
8. Listovat v menu lze pomocí kláves "A", "B", pro výběr slouží "D", "C" pro ukončení a Esc.
9. Nakonfigurované změny v manuálním modu se projeví, až po jeho opuštění.

F OBSAH CD S PŘÍLOHOU

- Složka Eagle (schémata zapojení, desky plošných spojů)
- Složka Raspberrypython (Hlavní skript a použité knihovny)
- Složka Bezdrát (skript pro Arduino)
- Složka Web (zdrojové kódy webových stránek)
- Soubor Přístupy.txt (přístupové kódy na ftp server a do MySQL databáze)