

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MĚŘENÍ VELIČIN VE STAVEBNÍCH KONSTRUKCÍCH

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN VLADAŘ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MĚŘENÍ VELIČIN VE STAVEBNÍCH KONSTRUKCÍCH

MEASSURING SYSTEM FOR BUILDING CONSTRUCTIONS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MARTIN VLADAŘ

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. ZDENĚK BRADÁČ, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Martin Vladař

ID: 120514

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Měření veličin ve stavebních konstrukcích

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte elektronický decentralizovaný systém vybavený rozhraním Ethernet pro měření parametrů (teplota, vlhkost atd.) ve stavebních konstrukcích.

1. Proved'te literární rešerši.
2. Navrhněte a realizujte koncepci systému jako modulárního mikropočítačového systému s funkcností měření fyzikálních parametrů ve stavebních konstrukcích.
3. Vytvořte vhodné komunikační rozhraní a připojení do nadřazeného systému tak, aby bylo možné připojit řádově až stovky čidel.
4. Navrhněte a realizujte elektroniku systému, realizujte DPS, osad'te je součástkami a oživte.
5. Vytvořte komplexní programové vybavení, které umožní ovládání a administrování systému.
6. Otestujte funkčnost a demonstруйте ji.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6
Dle pokynů vedoucího práce.

Termín zadání: 9.2.2015

Termín odevzdání: 18.5.2015

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové 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

Diplomová práce se zabývá měřením fyzikálních veličin ve stavebnictví za použití decentralizovaného systému pro sběr naměřených dat. Systém tvoří Raspberry Pi jako řídicí člen systému. K němu jsou na třech větvích připojeny moduly pro koncentraci dat ze snímačů fyzikálních veličin. Na každé z větví může být připojeno až 32 zařízení. V Raspberry Pi je implementováno webové rozhraní pro zobrazování naměřených hodnot a administraci systému.

KLÍČOVÁ SLOVA

Raspberry Pi, PHP, HTML, MySQL, ATmega8, mikrokontrolér, WWW, C, Jazyk C, LAMP Stack, Linux, Makefile, CSS, DPS, LM75A, ISP, LED, wiringPi, RS-485, Ethernet, UART, Eagle, Kate, I2C, 1-wire, SN75176A

ABSTRACT

This thesis deals with the measurement of physical quantities in building using decentralized system for data collecting. The system consists of Raspberry Pi as a master device of the system. There are three branches connected to this master device. Each branch consists of up to 32 modules for data-concentration from sensors of physical quantities. The Raspberry Pi has implemented web interface for displaying measured values and system administration.

KEYWORDS

Raspberry Pi, PHP, HTML, MySQL, ATmega8, microcontroller, WWW, C, Language C, LAMP Stack, Linux, Makefile, CSS, DPS, LM75A, ISP, LED, wiringPi, Ethernet, RS-485, UART, Eagle, Kate, I2C, 1-wire, SN75176A

VLADAŘ, Martin *Měření veličin ve stavebních konstrukcích*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2015. 76 s. Vedoucí práce byl doc. Ing. Zdeněk Bradáč, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Měření veličin ve stavebních konstrukcích“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové 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(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) 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.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Zdeňku Bradáčovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	10
1 Literární řešerše	11
1.1 Jednoduché stanice	11
1.2 Decentralizované stanice	11
1.3 Měření fyzikálních veličin ve stavebnictví	11
2 Koncepce systému	13
2.1 Topologie	13
2.2 Funkce	15
2.3 Raspberry Pi	16
2.4 ATmega8	17
2.5 Převodník I2C na 1-wire	18
2.6 Převodník UART na RS-485	18
2.7 Teplotní snímač LM75A	19
3 Návrh DPS	21
3.1 Napájení	21
3.2 RS-485	21
3.3 I2C a 1-wire	22
3.4 DPS Raspberry Pi	23
3.4.1 Schéma	23
3.4.2 Deska plošných spojů	24
3.4.3 Chyby a úprava návrhu DPS Raspberry Pi	25
3.5 DPS koncentrátoru dat	27
3.5.1 Schéma	27
3.5.2 Deska plošných spojů	29
3.5.3 Chyby a úprava návrhu DPS koncentrátoru dat	31
3.6 DPS snímače LM75A	31
4 Komunikační protokol RS-485	32
4.1 Funkce komunikace	32
4.2 SET řetězec	33
4.2.1 Příkaz s parametrem	33
4.2.2 Příkaz bez parametru	34
4.3 GET řetězec	35
4.4 Hardware	36

5	Koncentrátor dat	38
5.1	Programové vybavení koncentrátoru dat	38
5.1.1	Komunikační protokol	39
5.1.2	Manipulace s daty	39
5.1.3	Variabilita snímačů	41
6	Programy pro Raspberry Pi	42
6.1	LAMP Stack	42
6.2	Programy v jazyce C	44
6.2.1	Wiring Pi	44
6.2.2	Program dip_find_devices	46
6.2.3	Program dip_set_devices	47
6.2.4	Program dip_calibrate_devices	48
6.2.5	Program dip_measure	48
6.2.6	Makefile	48
7	Webové rozhraní	50
7.1	Vzhled WWW stránek	50
7.2	Struktura WWW stránek	51
7.2.1	Hlavička	51
7.2.2	Menu	52
7.3	Wizard	52
7.3.1	Krok 1	52
7.3.2	Krok 2	53
7.3.3	Krok 3	53
7.3.4	Krok 4	54
7.3.5	Krok 5	55
7.3.6	Krok 6	56
7.3.7	Krok 7	57
7.3.8	Krok 8	58
7.4	Zobrazování naměřených hodnot	59
7.4.1	Správa databáze	59
7.4.2	Odkaz ALL VALUES	61
7.4.3	Odkaz CUSTOM VALUES	62
7.4.4	Odkaz ERASE VALUES	63
8	Závěr	64
	Literatura	66

Seznam symbolů, veličin a zkratk	68
Seznam příloh	69
A DPS Raspberry Pi	70
A.1 DPS top Raspberry Pi	70
A.2 DPS bottom Raspberry Pi	70
A.3 Osazovací výkres top Raspberry Pi	71
A.4 Osazovací výkres bottom Raspberry Pi	71
A.5 Schéma zapojení Raspberry Pi	72
B DPS koncentrátor dat	73
B.1 DPS top koncentrátor dat	73
B.2 DPS bottom koncentrátor dat	73
B.3 Osazovací výkres top koncentrátor dat	74
B.4 Osazovací výkres bottom koncentrátor dat	74
B.5 Schéma zapojení koncentrátor dat	75
C Obsah přiloženého DVD	76

SEZNAM OBRÁZKŮ

2.1	Topologie logovacího systému	13
2.2	Raspberry Pi (PC velikosti kreditní karty) [1]	16
2.3	Mikrokontrolér Atmega8(L), řídící člen data-koncentrátoru [2]	17
2.4	Vnitřní zapojení DS2482-100 [3]	18
2.5	Pouzdro a logický diagram SN75176A [4]	18
2.6	Typická aplikace teplotního snímače LM75A [5]	19
2.7	Schéma zapojení teplotního snímače LM75A pro měřicí systém	20
2.8	Deska plošných spojů teplotního snímače LM75A pro měřicí systém	20
3.1	První verze schématu zapojení pro Raspberry Pi	23
3.2	DPS pro Raspberry Pi	24
3.3	Opravená verze schématu zapojení pro Raspberry Pi	25
3.4	Opravená verze schématu zapojení pro Raspberry Pi	26
3.5	Připojená DPS pro Raspberry Pi	26
3.6	DPS pro Raspberry Pi	27
3.7	DPS pro Raspberry Pi	29
3.8	Osazené DPS se snímači LM75A	31
4.1	Diagram komunikace na sběrnici RS-485	37
5.1	Koncentrátor dat v průběhu měření(napravo) se zapojeným snímačem LM75A(nalevo)	38
6.1	Rozvržení a názvy pinů na Raspberry Pi [8]	45
6.2	Část formuláře generovaná pro každý data-koncentrátor	47
6.3	Část menu generována na základě připojených datakoncentrátorů	47
7.1	Vzhled webové stránky	51
7.2	Seznam nalezených zařízení pomocí Wizardu	53
7.3	Generovaný formulář pro dva připojené data-koncentrátory	54
7.4	Seznam příkazů pro data-koncentrátory pro kontrolu	55
7.5	Status odeslání pro jednotlivé příkazy (První zařízení v pořadí pro demonstraci odpojeno)	56
7.6	Vstup pro kalibrační teplotu	57
7.7	Status kalibrace data-koncentrátorů (První v pořadí pro demonstraci odpojen)	58
7.8	Informační stránka o začátku měření	58
7.9	Zobrazení všech záznamů databáze	62
7.10	Formulář pro výběr specifických dat z databáze	63
7.11	Výpis naměřených hodnot pro snímač 5	63

ÚVOD

Cílem diplomové práce je provést literární rešerši problematiky měření fyzikálních veličin ve stavebních konstrukcích. Na základě rešerše navrhnout vlastní systém jako modulární mikropočítačový systém pro realizaci tohoto měření. Návrh se bude skládat z koncepce systému, návrhu desek plošných spojů pro jednotlivé části systému, návrh komunikačního rozhraní a komplexní softwarové vybavení pro mikropočítačové moduly i pro modul určený k ovládání a administraci systému. Do systému musí být možné zapojit řádově stovky snímačů pro měření fyzikálních veličin.

První část návrhu systému bude jeho koncepce. Bude nutné definovat jednotlivé prvky systému a vytvořit mezi nimi vazby a topologii. Od jejich vzájemné provázanosti se bude odvíjet zbytek návrhu systému. Další částí bude návrh desek plošných spojů pro jednotlivé prvky této koncepce, osazení těchto desek součástkami a oživení hardwarové části.

Systém musí mít implementován komunikační protokol pro přenos dat mezi mikropočítačovými moduly a hlavním, sběrným a řídicím modulem. Moduly by mezi sebou měly být schopny komunikovat na vzdálenost desítek až stovek metrů, což je hlavním požadavkem na řešení komunikačního protokolu.

Mikroprocesorové moduly musí obsahovat softwarové vybavení, umožňující získávání hodnot měření fyzikálních veličin osazenými snímači. Softwarové vybavení také musí dovolovat tyto záznamy udržovat v paměti a pomocí komunikační sběrnice umožňovat nastavení modulu a odesílání naměřených hodnot do nadřazeného modulu.

Modul určený k administraci, nastavování a sběru dat z mikroprocesorových modulů musí být schopen komunikace s těmito moduly, reprezentace a uložení naměřených dat získaných z těchto modulů.

1 LITERÁRNÍ REŠERŠE

1.1 Jednoduché stanice

Mezi základní teplotní dataloggery patří malé stanice obsahující snímače teploty a vlhkosti přímo s elektronikou pro ukládání, přenos a jednoduché zobrazení dat pomocí displaye. Taková zařízení jsou dostupná v každém větším obchodě s elektronikou. Jejich přesnost (kterou se chlubí) zpravidla dosahuje $\pm 1\%$ pro teplotu a $\pm 5\%$ pro relativní vlhkost. Obvyklé množství hodnot, které jsou dataloggery schopné uchovávat je kolem 32000, což je nejspíš odvozeno od omezení velikosti 16-bitového integeru, tedy 32768 hodnot. Nabízí se ale otázka, proč nebylo použito bezznaménkového integeru pro zdvojnásobení hodnot. Proto bude množství naměřených hodnot omezeno jiným faktorem. Mezi takové stanice patří například CEM DT-172 s displayem, Voltcraft DL-121TH připomínající USB klíčenku bez displaye. Většina těchto teplotních dataloggerů zaznamenává data do své interní paměti. Její čtení se realizuje připojením k PC pomocí USB rozhraní. Pro čtení je potom nutné mít nainstalovaný program pro čtení dat. Formát dat je většinou ve formátu txt, xls, csv. Některé programy dovolují vizualizaci hodnot v podobě grafu.

1.2 Decentralizované stanice

Tento druh teplotních dataloggerů je o poznání sofistikovanější. Jejich provedení často připomíná výrobky určené pro průmyslovou automatizaci. Vypadají solidně, jsou vyrobeny z odolnějších materiálů, především z hliníku a jeho slitin. Bohužel decentralizací se u mnoha výrobců snižuje možnost nastavení pro měření. Nastavení periody posílání vzorků bývá značně omezeno, nebo nejde nastavit vůbec. Další nevýhodou je zachování velikosti paměti pro ukládání naměřených hodnot. Problém potom nastává při použití více měřících stanic. Ty si celkovou paměť centrály rozdělují mezi sebe a tím se snižuje velikost použitelné paměti pro jednotlivé stanice.

1.3 Měření fyzikálních veličin ve stavebnictví

Ve stavebnictví je potřeba měřit fyzikální veličiny pro širokou škálu aplikací. Zadaním diplomové práce je navrhnout technické řešení pro samotné měření, proto se zaměřím na parametry snímačů a požadavky na systém pro tato měření. Teorii samotného měření se budu věnovat jen okrajově.

Mezi nejběžnější aplikace patří měření součinitele prostupu tepla stávajících stavebních konstrukcí [14]. Na tuto aplikaci se zaměřuje nejvíce výrobci měřících zařízení. Pro tato měření se používají snímače s přesností na desetinu °C. Měří se venkovní teplota, teplota vzduchu uvnitř místnosti a teplota zdiva ve více místech. Z těchto hodnot se počítá součinitel prostupu tepla U .

Další aplikací je například měření teploty při zkoušení reakce stavebních výrobků na oheň, včetně podlahových krytin, zkoušení požární odolnosti stavebních konstrukcí, zkoušení komínů a okmínových vložek a podobně [15]. V této aplikaci je kladen menší důraz na přesnost snímače teploty, naopak snímač musí být schopen měřit a odolat vysokým teplotám.

Další aplikací pro měření teploty může být například kontrola zahřívání betonu při tuhnutí pro betonování masivních konstrukcí [16]. V tomto případě jsou požadavky na snímače opět odlišné a to přesnost jednotek stupňů a rozsah do 100 °C.

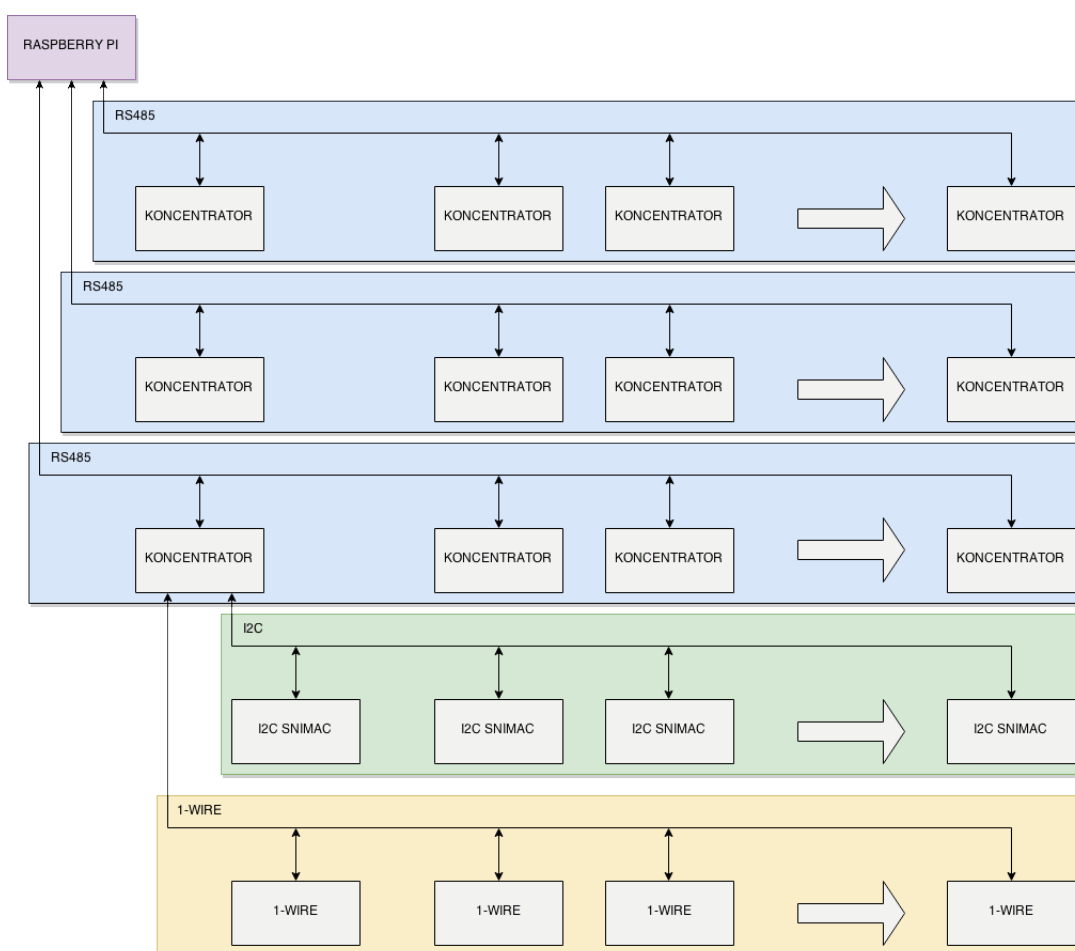
Záměrně jsem uvedl tři aplikace s odlišnými požadavky na teplotní snímače. Požadavky se liší podle zadání a typu aplikace. Proto se budu snažit návrh systému udělat univerzální a použitelný s libovolnými snímači.

Pro demonstraci funkce systému jsem se po konzultaci s vedoucím diplomové práce rozhodl použít snímače teploty LM75A od firmy Texas Instruments.

2 KONCEPCE SYSTÉMU

Prvním úkolem je vymyslet topologii celého systému. Po konzultaci s vedoucím diplomové práce jsem se rozhodl použít jako hlavní člen malý, ale plnohodnotný počítač Raspberry Pi, který je blíže popsán v kapitole 2.3. Pro mikroprocesorové moduly jsem také po konzultaci s vedoucím vybral malý, 8-mi bitový mikrokontrolér od firmy Atmel ATMega8, který svými parametry a periferiemi zapadá do návrhu a je blíže popsán v kapitole 2.4. Pro komunikaci mezi těmito prvky použiji sběrnici RS-485, která splňuje požadavek možnosti komunikace na vzdálenost desítek až stovek metrů.

2.1 Topologie



Obr. 2.1: Topologie logovacího systému

Systém pro sběr naměřených hodnot topologicky odpovídá rozvržení na obrázku 2.1. Nejvyšším prvkem hierarchie je Raspberry Pi. Přímo na Raspberry je možné připojit až 3 větve sběrnice RS-485. Na každou tuto větev je možné připojit až 32 zařízení, tedy data-koncentrátorů. Větvě jsou přímé a každý data-koncentrátor je osazen dvěma konektory pro připojení ke sběrnici RS-485. Jeden konektor je vstupem a druhý výstupem pro připojení dalšího data-koncentrátoru. Více větví je zde z důvodu možnosti potřeby zapojovat koncentrátory do hvězdy, což je s lineární topologií sběrnice RS-485 problematické.

Snažil jsem se zachovat co největší variabilitu celého systému. To však není možné bez jistých omezení. K omezením tak patří například implementace pouze rozhraní I2C a 1-wire pro připojení snímačů teploty. U snímačů má však zákazník možnost výběru a to zejména kvůli různorodosti použití systému. Některé aplikace mohou klást větší požadavky na přesnost snímačů, jiné zase na jejich rozsah a podobně.

Linka RS-485 potřebuje pro svou správnou funkci na obou svých koncích terminátory v podobě odporů. Terminátory na začátku větví jsou osazeny přímo na desce připojené k Raspberry Pi, ukončovací terminátor se pak zapíná vždy na posledním zařízení ve větvi. Pro tuto funkci slouží binární přepínač na každém data-koncentrátoru. Více o návrhu je popsáno v kapitole 3.

Každý data-koncentrátor je vybaven periferiemi pro komunikaci využívající I2C a 1-wire sběrnice. Tyto sběrnice se využívají pro komunikaci se snímači. Cílem diplomové práce je vytvořit systém, který je schopen funkce při použití libovolných snímačů za cenu minimálních úprav softwarového vybavení. Protože snímače se liší přesností, rozlišením, rozměry a cenou, je pro všestrannost systému dána volnost výběru zákazníkovi kvůli odlišným požadavkům na snímače pro odlišné aplikace. Jediným požadavkem na snímače je tedy rozhraní I2C nebo 1-wire.

Pro demonstraci systému jsem vybral I2C snímač teploty LM75A, který je blíže popsán v kapitole 2.7. Pro tento snímač jsem vytvořil i jednoduchou desku plošných spojů.

Co se týká potencionální marketingové strategie produktu, je zde možnost vytvářet zařízení "na míru" nebo realizovat systém se snímači podle přání zákazníka a tento systém přidat do nabídky. Základ systému by tedy zůstal stejný a produkt by se lišil pouze v použitých snímačích a tomu odpovídajících úpravách softwarového vybavení.

2.2 Funkce

Hlavními prvky systému jsou data-koncentrátory. Ty se starají o periodické získávání hodnot ze snímačů. Pro správu těchto zařízení slouží Raspberry Pi plnící funkci Master zařízení v systému.

Raspberry Pi se stará o vyčítání naměřených dat z data-koncentrátorů, nastavování těchto měřících stanic a zprostředkovává přístup k naměřeným hodnotám, které jsou ukládány do databáze.

Datové koncentrátory jsou určeny k samostatnému měření a pouze na požadavek od Raspberry Pi posílají data Masteru, který je vloží do databáze a data-koncentrátory ve své paměti po odeslání mažou data pro další měření. Z toho vyplývá, že je nežádoucí volit dlouhý interval pro vyčítání hodnot z data-koncentrátorů. Hlavním důvodem je snížení nároků na paměť data-koncentrátorů a častějším vyčítáním se také dosáhne aktuálnějších hodnot v databázi.

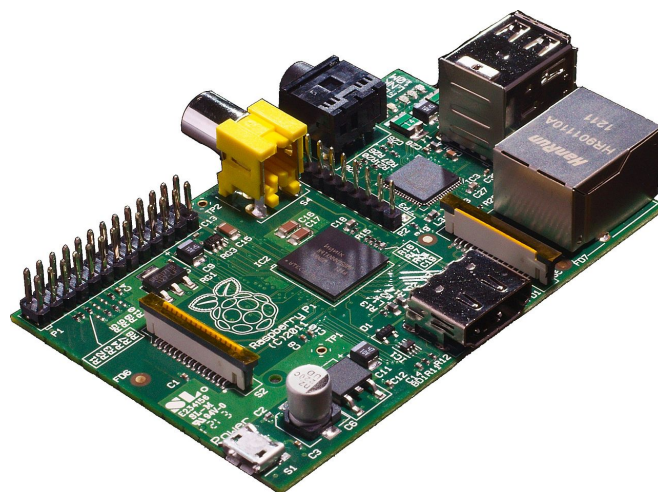
Vzhledem k tomu, že navrhuji systém pro širokou škálu snímačů a také proto, že pro demonstraci funkce jsem použil snímače LM75A (kapitola 2.7), které mají přesnost ± 3 °C, je nutné do systému implementovat kalibraci snímačů. K tomu bude využito Raspberry Pi a na něm běžící grafické uživatelské rozhraní stejně jako pro zobrazení naměřených hodnot.

Po propojení prvků systému je potřebné provést nastavení systému. V mém případě je potřeba:

- Detekovat všechny data-koncentrátory
- Nastavit tyto data-koncentrátory
- Provést kalibraci snímačů
- Periodicky vyčítat naměřené hodnoty

K tomu jsem napsal aplikace v jazyce C, které se starají o jednotlivé funkce a jsou spouštěny z PHP kódu, aby bylo možné jejich výstupy jednoduše sledovat z uživatelsky přátelského rozhraní, z webových stránek.

2.3 Raspberry Pi



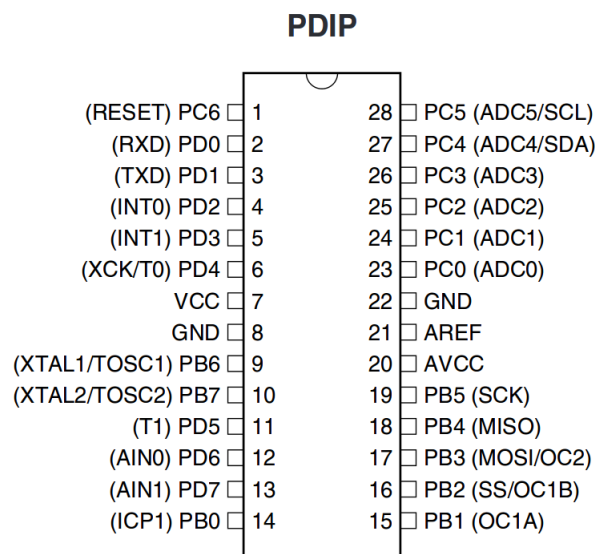
Obr. 2.2: Raspberry Pi (PC velikosti kreditní karty) [1]

Navržený systém je založen na platformě jménem Raspberry Pi. Jde o malý počítač, jehož DPS má velikost kreditní karty. Tato platforma je jako nadřazený člen v systému pro logování dat ideální volbou, už jen z důvodu maximálního příkonu kolem 3 Wattů. Díky jeho rozšířenosti je zajištěna dobrá podpora, a to jak od vývojářů, tak od uživatelů. Není tedy divu, že je určen pro použití s operačními systémy založené na Linuxovém jádře. Nejpoužívanější jsou samozřejmě ty oficiálně podporované. Patří mezi ně Raspbian založený na Debian distribuci Linuxu a Pidora jako klon Fedora Linuxu. Dnes na oficiálních stránkách další distribuci nenajdete. Mezi další distribuce, které se používají na Raspberry ale patří i Arch Linux a operační systém známý z mobilních zařízení, Android.

Raspberry je osazeno 700MHz procesorem, 512MB RAM sdílených s grafickým procesorem, slotem pro SD kartu, dvěma USB porty a ethernetovým adaptérem 10/100 s konektorem RJ45. Deska je dále osazena MicroUSB konektorem určeným výhradně k napájení, jack 3,5mm konektorem pro práci se zvukem a HDMI konektorem. Na desce je také patice s GPIO porty, které dovolují práci s UART, I2C a SPI. Raspberry bylo navrženo jako prvek kombinující mikrokontrolér a osobní počítač. [1]

Pro účel diplomové práce jsem použil operační systém Raspbian dostupný z oficiálních webových stránek [1].

2.4 ATmega8

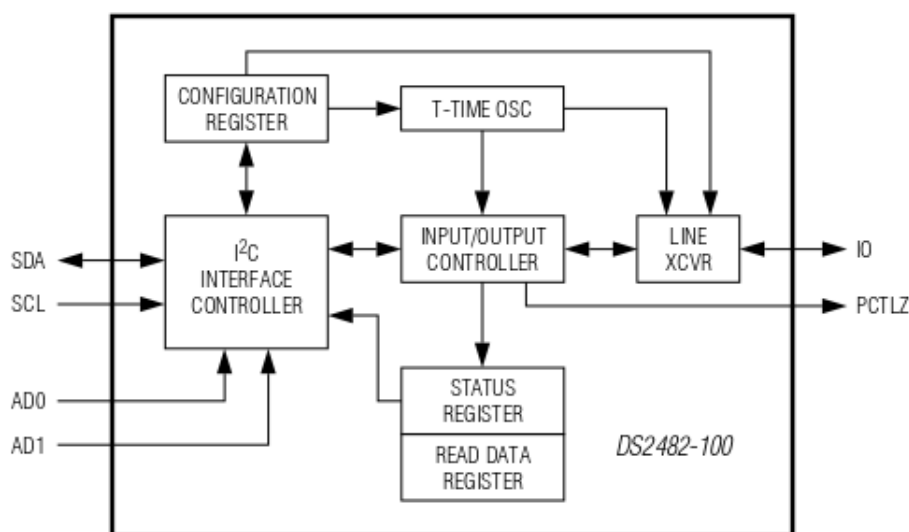


Obr. 2.3: Mikrokontrolér Atmega8(L), řídící člen data-koncentrátoru [2]

Tento kontrolér má pouze 8kB flash paměti pro program, 512B EEPROM, 1kB-SRAM, USART, SPI a A/D převodník [2]. Parametry tohoto procesoru pro logovací systém postačují. Mikrokontrolér bude obsluhovat sběrnice TWI (I2C), UART, 1-wire (pomocí převodníku I2C na 1-wire, který je popsán v kapitole 2.5). Pro dočasné ukládání dat bude sloužit paměť EEPROM, která je i přes svou malou velikost (512 B) dostatečně velká pro zálohu, protože se jedná o krátkodobou zálohu pouze do doby přenesení dat do Raspberry Pi pro jejich zařazení do databáze. K mikrokontroléru bude pro získávání dat ze senzorů připojen převodník z rozhraní I2C na rozhraní 1-wire (kapitola 2.5). Kromě 1-wire bude pro propojení se senzory využívána samotná I2C sběrnice.

Komunikace s Raspberry Pi bude realizována na sběrnici RS-485 pomocí UART, který je součástí mikrokontroléru a převodníku UART na RS-485 (kapitola 2.6).

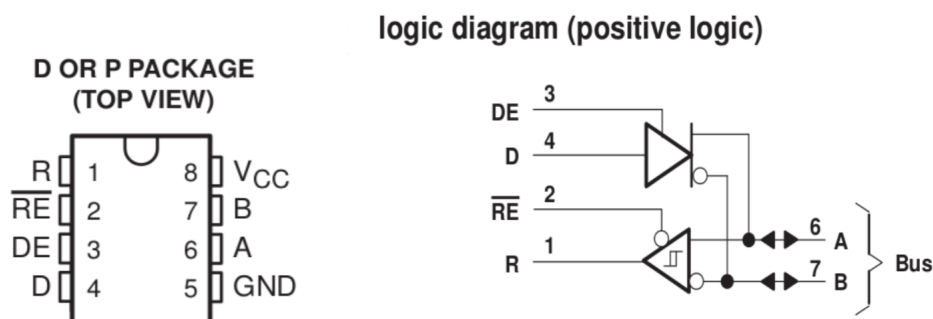
2.5 Převodník I2C na 1-wire



Obr. 2.4: Vnitřní zapojení DS2482-100 [3]

Pro komunikaci s 1-wire snímači je osazen převodník ze sběrnice I2C na 1-wire. Více informací o převodníku je možné nalézt v katalogových listech [3] a v kapitole 3 je popsáno jeho zapojení. Převodník je od firmy Maxim a nese označení DS2482-100.

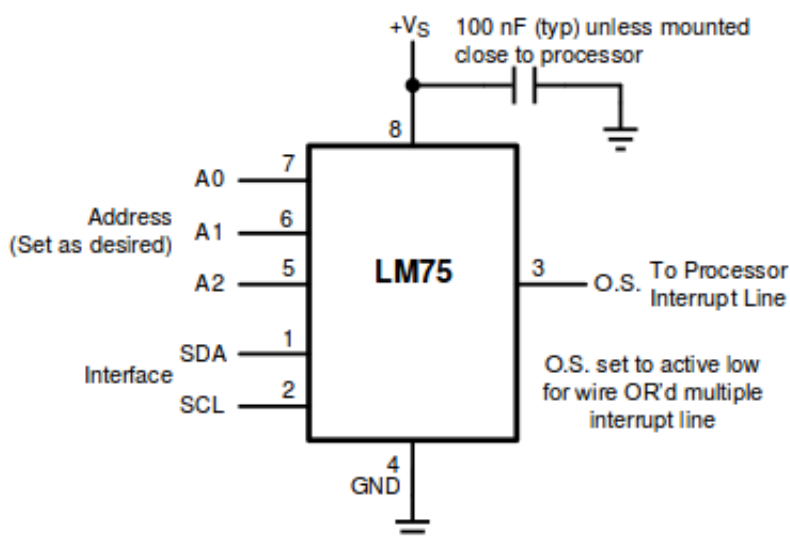
2.6 Převodník UART na RS-485



Obr. 2.5: Pouzdro a logický diagram SN75176A [4]

Pro komunikaci mezi jednotlivými data-koncentrátory se používá UART a převodník na half-duplex RS-485 sběrnici SN75176A od společnosti Texas Instruments [4]. Na straně Raspberry Pi jsou použity 3 tyto převodníky, jeden pro každou větev. Každý data-koncentrátor je osazen jedním z těchto převodníků. O řízení veškeré komunikace na RS-485 sběrnici se stará Raspberry Pi a plní funkci Master. Všechny ostatní zařízení připojené ke sběrnici naslouchají a chovají se jako Slave. Je to jediný způsob jak se vyhnout kolizím na tomto druhu sběrnice při takovém počtu zařízení.

2.7 Teplotní snímač LM75A



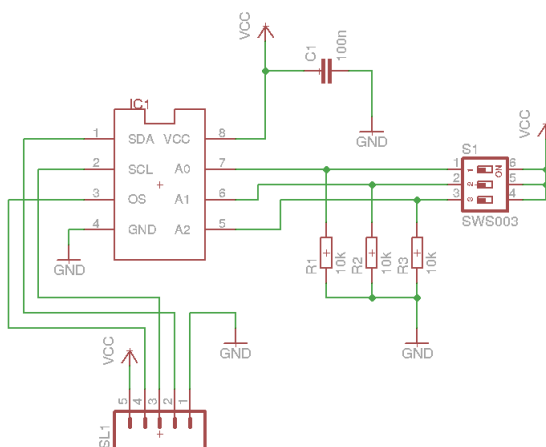
Obr. 2.6: Typická aplikace teplotního snímače LM75A [5]

Pro demonstraci systému jsem vybral měření teploty snímačem LM75A od firmy Texas Instruments. S tímto snímačem je možné komunikovat po sběrnici I2C. U snímače je možné nastavit adresu I2C a to v rozsahu tří bitů. Na stejné sběrnici tedy může být zapojeno až osm těchto snímačů. Rozlišení teploty je u tohoto snímače $0,5^{\circ}\text{C}$ a jeho přesnost je v rozmezí teplot $-25^{\circ}\text{C} - 100^{\circ}\text{C} \pm 2^{\circ}\text{C}$ a pro teploty v rozmezí $-55^{\circ}\text{C} - 125^{\circ}\text{C} \pm 3^{\circ}\text{C}$. Rozsah měření teploty je $-55^{\circ}\text{C} - 125^{\circ}\text{C}$ [5].

Kvůli velké odchylce snímače je potřebná kalibrace. Ta bude realizována uživatelsky příjemnou cestou přes webové rozhraní, stejně jako nastavení snímačů připojených ke sběrnici (více v kapitole 7.3).

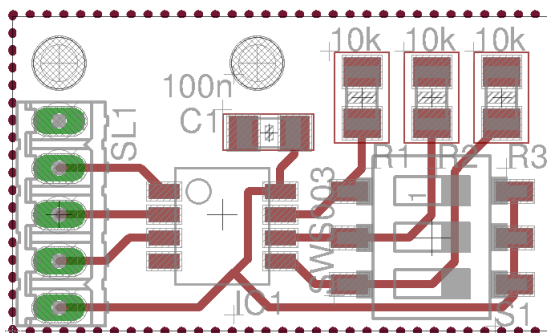
Na základě typické aplikace teplotního snímače (2.7) jsem vytvořil schéma zapojení a desku plošných spojů pro tento snímač. Hlavním důvodem je velikost pouzdra

SOIC(8), tedy pouzdro pro povrchovou montáž v kombinaci s možností nastavení adresy I2C, které je bez této desky velmi komplikované.



Obr. 2.7: Schéma zapojení teplotního snímače LM75A pro měřicí systém

Schéma je velmi jednoduché, obsahuje pouze samotný snímač, blokovací kondenzátor, pull-down rezistory, binární přepínač pro nastavení adresy a konektor pro připojení ke sběrnici I2C.



Obr. 2.8: Deska plošných spojů teplotního snímače LM75A pro měřicí systém

Deska plošných spojů je stejně jako schéma zapojení velmi jednoduchá. Snímač jsem se rozhodl umístit doprostřed desky, aby byl chráněn před mechanickým poškozením. Konektor pro připojení k I2C sběrnici je z praktických důvodů naopak na kraji stejně jako binární přepínač pro volbu adresy snímače. Pro instalaci desky na místo měření teploty jsou určeny dva otvory průměru 2,8mm na okraji desky.

3 NÁVRH DPS

Návrh DPS je realizován v programu Eagle 6.5.0 pro Linux Light Edition. Tato verze programu je s omezeními, ale zdarma a pro návrh postačuje.

Byly navrženy celkem tři desky. Jedna byla popsána v kapitole 2.7. Druhá deska slouží jako nástavba pro Raspberry Pi na jeho GPIO piny a třetí je data-koncentrátor, jehož produkce se očekává ve větším množství, jelikož se jedná o decentralizovaný systém.

3.1 Napájení

Napájení je do celé sítě distribuováno od Raspberry Pi. Samotný výkon malého počítače se při použití více koncentrátorů jeví jako nedostačující, a tak je přivedeno napájení o vyšším napětí, které je stabilizováno na +5V [7].

Stabilizátorem je také osazen každý data-koncentrátor. Do systému pro měření je od desky plošných spojů pro Raspberry Pi posíláno vyšší napětí z externího zdroje.

Přikláním se k tomuto řešení kvůli omezení proudu stabilizátoru na 1A, což je pro tak rozvětvený systém nedostačující. Dalším důvodem jsou ztráty na vedení, které by se na vedení o teoretické délce až 1200 metrů [6] mohly silně projevit a napětí na konci vedení by nedostačovalo pro napájení data-koncentrátorů.

Celkový příkon systému lze jen těžko určit, protože je závislé pouze na aplikaci a nevím tedy, kolik a jakých snímačů bude v systému nakonec použito.

Pro napájení z externího adaptéru je použit konektor WEALTHMETAL DS-241B s vidlicí šířky 2,1mm. Tento konektor je používán na velkém množství adaptérů. Napětí adaptéru je omezeno zespod pouze dropem stabilizátoru a shora maximálním napájením, které stabilizátor na svém vstupu snese, tedy 35V [7].

Raspberry samotné je napájeno obvyklým způsobem, a to Micro-USB kabelem. Propojeny jsou pouze země, kvůli možnému rozdílu jejich potenciálu, který by v lepším případě zapříčinil nefunkční UART, v horším případě by došlo ke zničení Raspberry Pi.

3.2 RS-485

Pro vedení RS-485 je zvolen ethernetový kabel CAT6 pro jeho rozšíření, dostupnost a složení ze čtyř kroucených dvojlinek, které jsou vhodné právě pro RS-485.

V systému je použit Master-Slave hierarchie pro komunikaci s koncentrátory. Raspberry je Master a řídí všechnu komunikaci. DPS na Raspberry Pi je osazena

třemi konektory RJ45 a třemi převodníky z UART na RS-485. Tento systém Master-Slave je implementován kvůli half-duplex komunikaci, pro kterou musí být použit komunikační protokol, pomocí kterého bude Raspberry Pi žádat postupně každý koncentrátor o naměřená data. Data budou obsahovat typ a adresu snímače, čas měření pro synchronizaci v databázi a samotnou hodnotu měření. Protože vyčítání hodnot zabere určitou dobu, je potřeba použít buffer. Ten bude realizován v paměti EEPROM data-koncentrátoru, tedy ATMegy8. Tato paměť má velikost pouze 512 bajtů, což není mnoho ale i pro rozvětvené systémy by měla být dostačující. V opačném případě se do systému přidá externí paměť, ideálně na sběrnici I2C, která bude i tak využívána, takže přidání paměti bude jednoduché a nebude při tom potřeba příliš zasahovat do plošného spoje.

Raspberry Pi musí být schopno postupně přepínat mezi všemi třemi větvemi a procházet postupně všechny data-koncentrátory na každé větvi. Implementováno bude zjištění počtu zařízení na každé větvi, pomocí jednoduchého echa. Adresy jednotlivých koncentrátorů se dají nastavit pomocí kódovacího přepínače, který je umístěn na všech data-koncentrátorech a je pětibitový. Na každé větvi je tedy možné provozovat maximálně 32 zařízení. Master komunikaci řídí a nepotřebuje tedy svou adresu, pomocí které by byl dosažitelný pro koncentrátory.

Master vždy pošle požadavek na jeden z data-koncentrátorů a předá mu sběrnici. Koncentrátor poté podle požadavku odešle zprávu, že na adrese existuje, pošle buffer naměřených dat, popřípadě se uspí, vypne, nebo například vymaže buffer s hodnotami. Po přijetí ukončovacího znaku nebo udaného počtu bajtů si bere sběrnici zpět Master a pokračuje na další adresu. To se opakuje pro všech 32 adres na větvi a pro všechny tři větve.

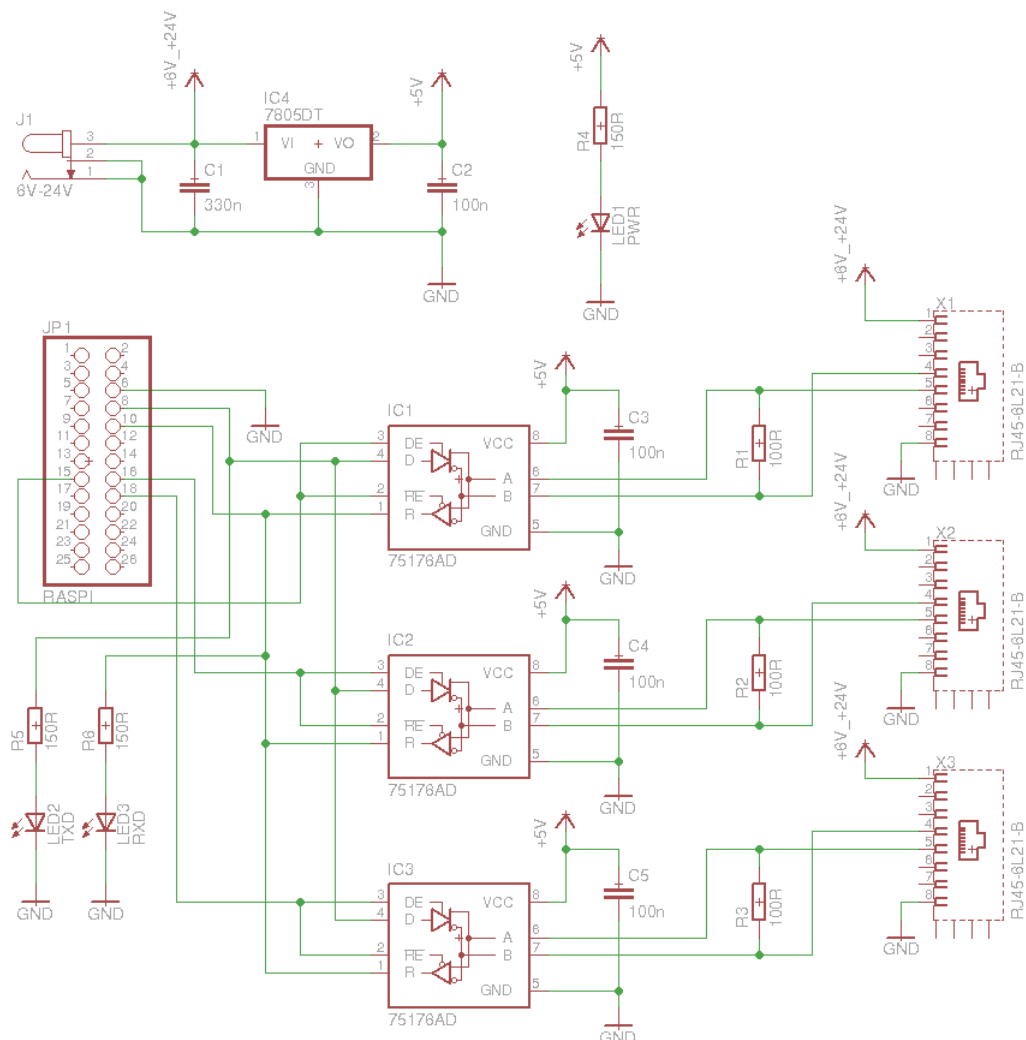
3.3 I2C a 1-wire

Na data-koncentrátoru jsou použity dvě sběrnice pro komunikaci se snímači. Jedná se o I2C a 1-wire rozhraní. Aby nebyl uživatel systému omezen na určitý typ konektorů a snímačů, ale mohl použít libovolný, jsou oba typy snímačů připojovány pomocí šroubovacích konektorů. Kromě datových vodičů je vyvedeno také +5V a GND pro napájení snímačů. I2C sběrnice je přivedena přímo na mikrokontrolér ATMegy8. K datovému i hodinovému vodiči SDA a SCL jsou připojeny pouze pull-up rezistory.

Pro komunikaci se snímači na 1-wire sběrnici je použito převodníku MAXIM DS2482-100 (více v kapitole 2.5). Tento převodník obstarává komunikaci s 1-wire snímači a mikrokontrolér komunikuje pouze s převodníkem pomocí I2C sběrnice. Pokud by nebylo použito tohoto převodníku, musela by se komunikace po 1-wire sběrnici napsat softwarově.

3.4 DPS Raspberry Pi

3.4.1 Schéma

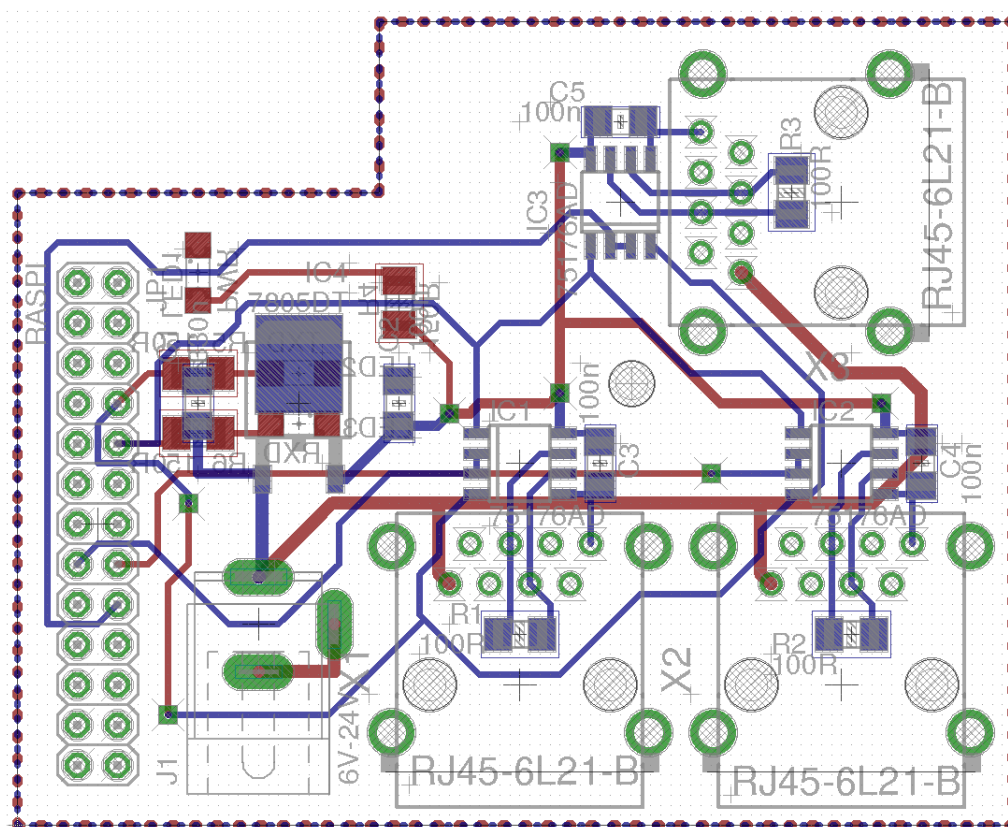


Obr. 3.1: První verze schématu zapojení pro Raspberry Pi

Schéma DPS pro Raspberry Pi je velmi jednoduché. Je zde pouze napájecí konektor s šířkou hrotu 2,1mm, který přivádí vstupní napětí na stabilizátor napětí 7805DT. Je osazena LED signalizující zařízení pod napětím. Hlavním prvkem je 26-ti pinová patice na Raspberry Pi, která je protikusem právě na GPIO Raspberry. Na zdrojích napětí Raspberry Pi a adaptéru jsou propojeny země. Schéma obsahuje trojici bloků pro jednotlivé větve RS-485. Každý blok se skládá z převodníku UART na RS-485 SN75176AD od firmy Texas Instruments [4] a konektoru RJ45. Jedná se o začátky každé větve, takže nechybí 100 Ohm terminátor pro každou větev. Vnitřní kroucený

pár konektoru RJ45 je využit pro diferenciální vodiče A a B. Dále je zapojena zem a napájení přímo z externího zdroje. Byly přidány LED pro indikaci aktivní komunikace na lince. Díky LED je možné poznat, jestli Raspberry Pi právě čte nebo zapisuje na sběrnici.

3.4.2 Deska plošných spojů

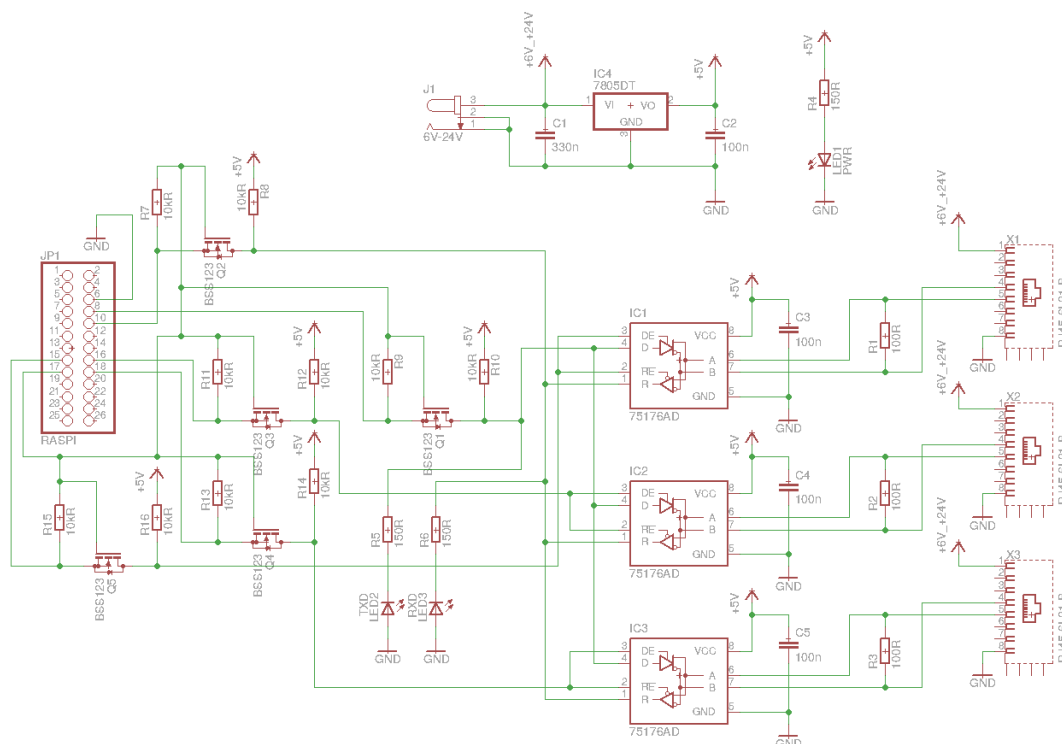


Obr. 3.2: DPS pro Raspberry Pi

Návrh plošného spoje pro Raspberry Pi byl komplikován omezenými rozměry, které musely být respektovány. Na horní straně plošného spoje jsou 3 konektory RJ45 a konektor pro připojení externího napájecího adaptéru. Každý z konektorů má u sebe převodník, terminátor a blokovací kondenzátor situovaný co nejbližší každému integrovanému obvodu. Vidět jdou 3 LED, jedna pro indikaci napájení, další dvě pro indikaci aktivity na TXD a RXD pinech Raspberry Pi, tedy indikující aktivní sériovou linku. Kromě konektorů RJ45, konektoru napájení a konektoru na 26-ti pínovém konektoru na GPIO piny Raspberry Pi je DPS realizováno pomocí součástek pro povrchovou montáž. Důvodem použití SMD součástek je požadavek na malé rozměry DPS.

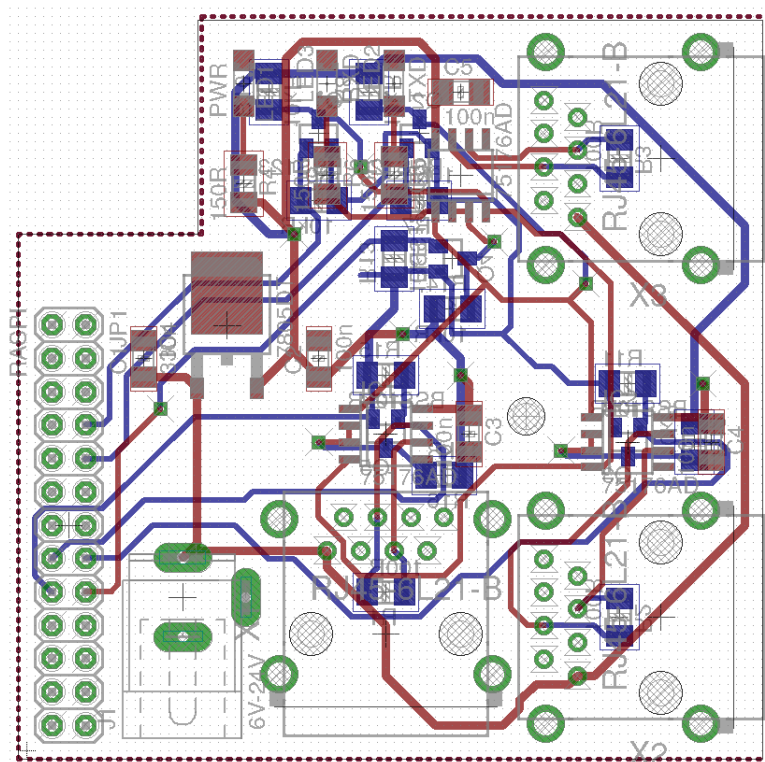
3.4.3 Chyby a úprava návrhu DPS Raspberry Pi

Během návrhu první verze schématu jsem se chybně domníval, že napěťové úrovně Raspberry Pi jsou 5 V. Skutečná hodnota je však 3,3 V, a proto jsem musel schéma upravit a přidat převodníky napěťových úrovní. Po opravě této chyby jsem již na další problém ve schématu nenarazil. Na obrázku 3.3 je zobrazena opravená verze schématu pro Raspberry Pi.



Obr. 3.3: Opravená verze schématu zapojení pro Raspberry Pi

Návrh nové desky se zkomplikoval, jelikož přibýlo relativně velké množství součástek. I novou desku jsem se snažil rozměry přizpůsobit tak, aby se dala připojit k Raspberry Pi shora, nepřesahovala za okraj Raspberry Pi a vypadala jako rozšiřující modul. Rozměr je omezen zejména konektory Raspberry Pi, kvůli kterým musí být deska v jednom rohu vykrojena. Nová deska plošných spojů je zobrazena na obrázku 3.4 a na obrázku 3.5 je zobrazena nová deska připojena na Raspberry Pi shora. Jde vidět, že ideálně pasuje i při zachování možnosti připojení kamery pomocí CSI konektoru.



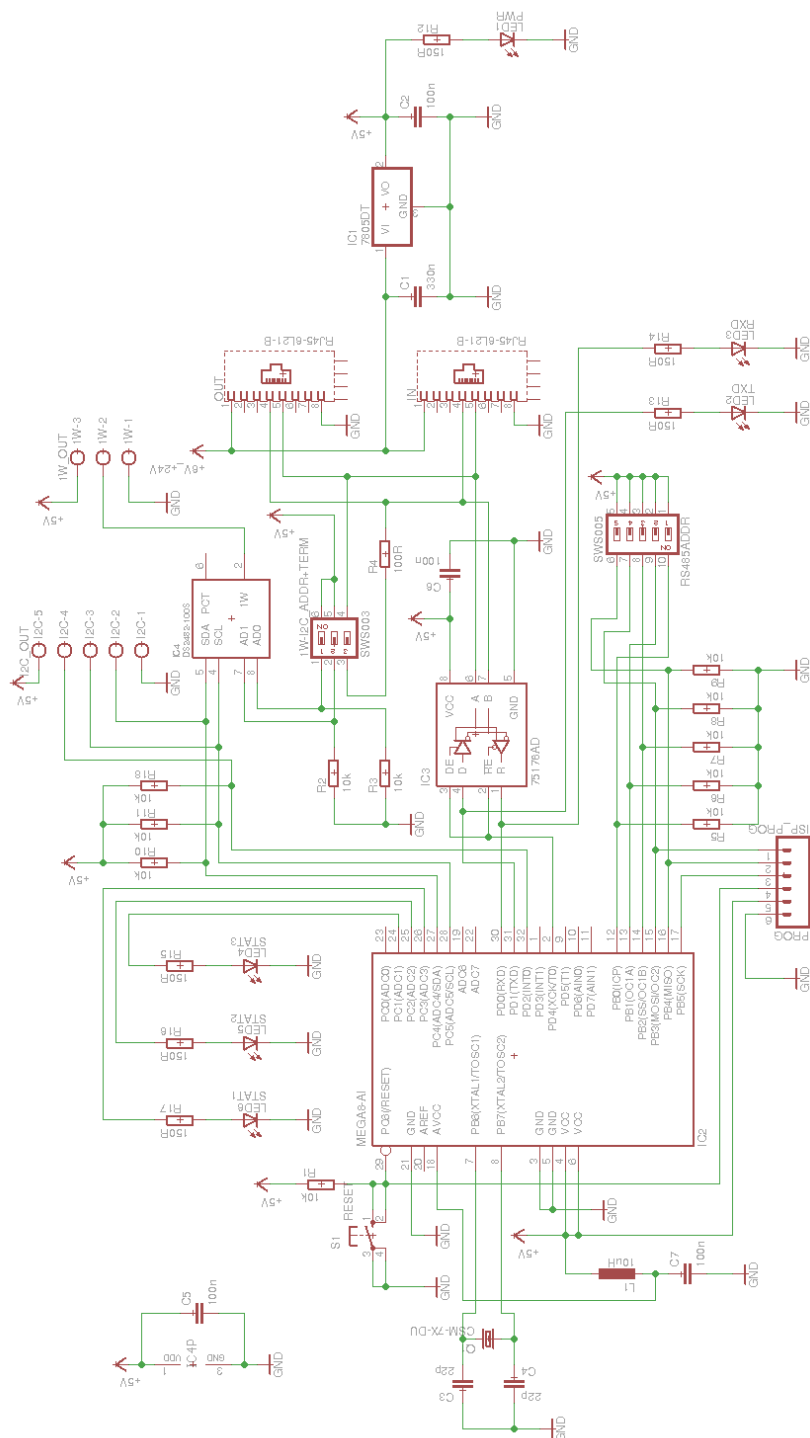
Obr. 3.4: Opravená verze schématu zapojení pro Raspberry Pi



Obr. 3.5: Připojená DPS pro Raspberry Pi

3.5 DPS koncentrátoru dat

3.5.1 Schéma



Obr. 3.6: DPS pro Raspberry Pi

Schéma koncentrátoru je o poznání složitější než schéma desky pro Raspberry Pi. Celému schématu dominuje ATmega8. Tento mikrokontrolér je osazen několika vstupně-výstupními periferiemi. Napájení je realizováno stejně jako u desky plošných spojů u Raspberry Pi, tedy stabilizátorem napětí 7805DT. Vstupní napětí se bere z konektoru RJ45, kam je posíláno z desky pro Raspberry Pi. Ani zde nechybí LED signalizace napájení. ATmega je tedy napájena +5V a je připojeno i napájení pro A/D převodník. Ten sice teď není využíván, ale jeden pin (ADC0) je volný pro možnost potřeby.

Na mikrokontrolér jako data-koncentrátor jsou kladeny vysoké nároky na přesnost času pro záznam měření. Proto je osazen externí krystal pro přesnější hodinový signál. Jeho hodnotu jsem nakonec zvolil 7 372 800 Hz, kvůli menší chybě UART při tomto taktu mikrokontroléru.

Z ATmegy jsou vyvedeny piny pro ISP SPI programátor. V případě potřeby se vyvedou piny pro JTAG kvůli debugování při tvorbě programu. Program však nebude příliš složitý, takže s touto možností nepočítám.

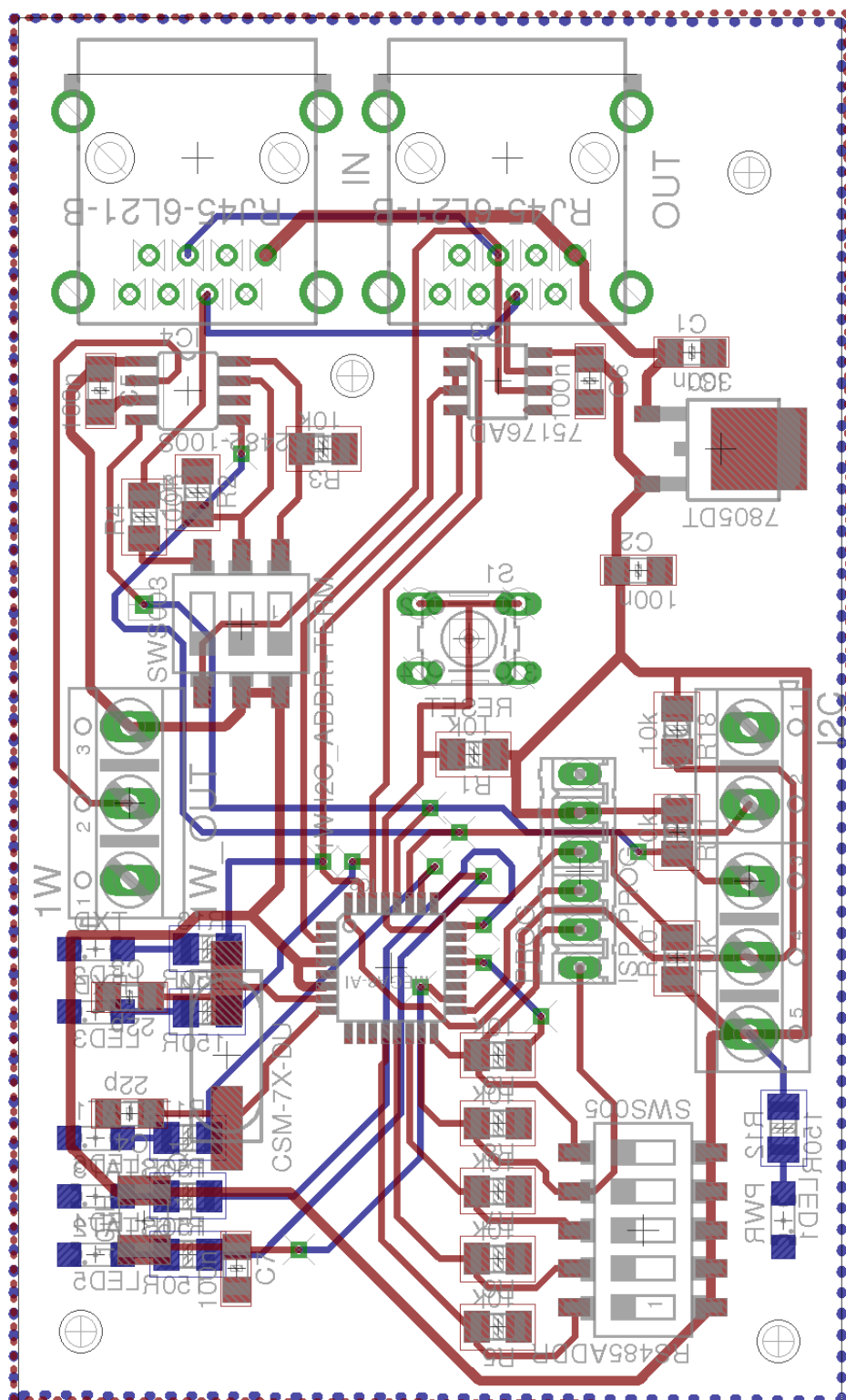
K mikrokontroléru jsou připojeny 3 LED, pro které zatím není známo jejich využití. Dá se však předpokládat, že alespoň některé stavy, do kterých se mikrokontrolér dostane, by měly být pozorovatelné jasnou signalizací.

Schéma dále obsahuje kódovací přepínače, pomocí kterých se nastavuje adresa zařízení na RS-485 sběrnici. Je zde 5 přepínačů, což odpovídá právě 32 různým kombinacím, tedy adresám data-koncentrátorů na sběrnici RS-485.

Signál ze vstupního konektoru sběrnice RS-485 je přeposlán na výstupní konektor RS-485 pro připojení dalšího zařízení v řadě. Pokud je zařízení poslední v řadě, i když jich není 32, je nutné zapojit terminátor. K tomu slouží kódovací přepínač. Ten kromě připojování terminátoru také nastavuje I2C adresu převodníku I2C na 1-wire (kapitola 2.5). Tedy ne celou adresu, ale pouze její spodní 2 bity.

Pro funkci sběrnice I2C jsou připojeny na vodiče SDA a SCL pull-up rezistory stejně jako na interrupt vodič, který využívají některé snímače pro signalizaci připravených dat ke čtení a podobně. Tento vodič je připojen na pin INT0, takže vyvolává hardwarové přerušování, čímž odpadá nutnost hlídat si data-ready flag ve smyčce a vytěžovat procesor. Pro 1-wire rozhraní je použit 3-pinový konektor na data, napájení a GND.

3.5.2 Deska plošných spojů



Obr. 3.7: DPS pro Raspberry Pi

Plošný spoj data-koncentrátoru byl pro návrh jednodušší než deska pro Raspberry Pi a to z toho důvodu, že jsem nemusel řešit striktní rozměry a pouze jsem se snažil DPS minimalizovat a oddělit konektory a indikační prvky každý na jednu stranu DPS.

Na jedné straně jsou situovány konektory a vysoké součástky a na druhé straně je tlačítko reset a indikační LED. Po instalaci DPS do krabičky chci mít diody blízko stěny krabičky a konektory RJ45 spuštěné dolů na dno krabičky. Konektory RJ45 budou samozřejmě přístupné z boku krabičky. Nastavení adresy koncentrátoru na sběrnici RS-485 bude přístupné pouze po otevření krabičky stejně tak jako nastavení terminátoru a I2C adresy převodníku z I2C na 1-wire DS2482-100. Tato nastavení bude provádět specializovaný pracovník při instalaci celého systému a pro obsluhu jsou k dispozici pouze indikační LED a tlačítko reset.

Další otázkou je ochrana DPS data-koncentrátoru. Řešení je najít v nabídce odpovídající krabičku nebo si krabičku vytisknout pomocí 3D tiskárny, které jsou k dispozici podobně jako klasické tiskárny v copy centrech. Stejně tak by bylo vhodné řešit obal pro Raspberry Pi s DPS.

Rozložení součástek DPS je situováno kolem mikrokontroléru. Na desce jsou situovány dva kódové přepínače. Jeden z nich slouží k nastavení adresy koncentrátoru na RS-485 sběrnici. Druhý slouží pro nastavení (zapnutí) terminátoru, pokud se jedná o poslední element sběrnice a pro nastavení adresy převodníku I2C na 1-wire. Uprostřed DPS je situováno tlačítko pro reset zařízení. Indikační diody jsou rozmístěny různě po celé DPS podle jejich funkce. Osamocená je LED pro indikaci napětí (zapnutí) data-koncentrátoru, na druhé straně jsou potom LED pro indikaci aktivity na UART (oddělené RxD a TxD) a tři LED, které indikují stav data-koncentrátoru. První indikuje detekovaný data-koncentrátor Master zařízením. Druhá LED signalizuje nastavení a kalibraci zařízení a třetí LED aktivní měření data-koncentrátoru.

Uprostřed DPS je situován konektor pro programování koncentrátoru. U něj také není žádoucí, aby byl přístupný obsluze.

Naopak na bocích DPS jsou konektory pro I2C snímače a 1-wire snímače. O napájení všech integrovaných obvodů se stará stabilizátor napětí 7805DT, který jako vstupní napětí používá přívod z RJ45 konektoru, který k němu směřuje DPS pro Raspberry Pi a DPS koncentrátoru jej zase posílá dalšímu modulu přes RJ45 konektor. Oba konektory RJ45 jsou situovány také na okraji DPS pro snadnou dostupnost a jsou vedle sebe. Po uzavření do krabičky tedy bude horní strana určena indikačním LED a tlačítku reset, jeden bok pro RJ45 konektory pro RS-485 sběrnici, druhý bok jen pro I2C konektor, třetí bok naproti pro 1-wire interface a spodní strana a jeden bok budou zaslepeny.

3.5.3 Chyby a úprava návrhu DPS koncentrátoru dat

Schéma i desku plošných spojů data-koncentrátoru se mi podařilo navrhnout dokonaleji, než tomu bylo u desky pro Raspberry Pi (kapitola 3.4). Snad jediným nedostatkem byla absence zpětné vazby od binárního přepínače pro nastavení I2C adresy převodníku I2C na 1-wire (kapitola 2.5) do mikrokontroléru ATMega8. Deska byla však upravena pouze přidáním dvou vodičů pro přidání této zpětné vazby. Zásah nebyl natolik markantní, abych pro data-koncentrátor nechal vyrábět novou DPS.

3.6 DPS snímače LM75A

Pro snímač LM75A od firmy Texas Instruments, který s data-koncentrátory komunikuje prostřednictvím sběrnice I2C jsem navrhl malou desku plošných spojů. Bez této desky by bylo připojení snímače do systému velmi komplikované. Návrh desky je popsán v kapitole 2.7. Zde je pouze přidána fotografie hotových výrobků.



Obr. 3.8: Osazené DPS se snímači LM75A

4 KOMUNIKAČNÍ PROTOKOL RS-485

Pro komunikaci mezi prvky systému po sběrnici RS-485 bylo nutné navrhnout komunikační protokol. Z důvodu možného počtu zařízení na všech sběrnicích až 96 (32 na každé ze tří sběrnic) jsem se snažil minimalizovat množství dat na sběrnici. Implementoval jsem tedy pouze nejnutnější bajty, které je nutné pro nastavení a vyčítání hodnot posílat.

Protože komunikace mezi Master a Slave zařízením se pro směry komunikace výrazně liší, jsou implementovány dva typy řetězců, které se na sběrnici mohou objevovat.

Jeden z řetězců je určen pro komunikaci od Master (Raspberry Pi) ke Slave (data-koncentrátor) zařízením. Řetězce putující tímto směrem slouží k nastavení data-koncentrátorů a ke komunikaci s nimi pomocí jednoduchých příkazů. Proto tento typ řetězce označujeme **SET**.

Druhý typ řetězce je tedy typ **GET** a slouží ke komunikaci od data-koncentrátorů k Raspberry Pi. Tento typ řetězce má větší délku a slouží k posílání naměřených dat.

Všechna komunikace mezi zařízeními má fixní rychlost 115200 BAUD, posílá se vždy 8 bitů, jeden stop bit, bez parity.

4.1 Funkce komunikace

Každá ze sběrnic je separovaná od ostatních, a přepínání a správu mezi nimi má na starosti výhradně Master (Raspberry Pi). Princip funkce jsem tak zjednodušil na obsluhu jedné sběrnice s 32-ti zařízeními a přepínání mezi sběrnicemi. Jediné zařízení v systému, které má informaci o tom na jaké sběrnici jsou jednotlivé data-koncentrátory připojeny je Master (Raspberry Pi). Samotné data-koncentrátory nesou informaci o tom na které sběrnici jsou připojeny.

Každý data-koncentrátor na sběrnici má svou unikátní adresu, která je nastavována hardwarově a případné kolize v těchto adresách musí být vyřešeny správcem měřicího systému. Komunikační protokol není schopen tuto kolizi ani detekovat, ani vyřešit, alespoň v této verzi systému.

Pro lepší přehlednost v kódech jak Master, tak i Slave zařízení jsou kódy příkazů pro příkazy s parametrem a příkazy bez parametru rozděleny. Toto rozdělení kódů jsem udělal v polovině velikosti bajtu. Kódy v rozmezí 0x00 – 0x7F jsou určeny pro kódy příkazů bez parametru a v rozmezí 0x80 – 0xFF jsou kódy příkazů s parametrem. Další výhodou tohoto rozdělení kódů je také menší možnost kolize při komunikaci systémem. Pokud koncentrátor očekává příkaz s parametrem

(0x80 - 0xFF) a obdrží bajt hodnoty kódu příkazu bez parametru (0x00 - 0x7F), vyhodnotí přijatý bajt jako chybný a tím i celý řetězec.

4.2 SET řetězec

Jak již bylo naznačeno v kapitole 4, pro komunikaci směrem od Raspberry Pi k data-koncentrátorům je určen řetězec SET. Tento řetězec slouží k posílání příkazů pro nastavení každého z data-koncentrátoru. Jsou rozlišovány dva typy příkazů, a to s parametrem a bez parametru. Posílaný řetězec má pouze čtyři bajty a obsahuje pouze nejdůležitější informace.

4.2.1 Příkaz s parametrem

SB	ADDR	CMD	PARAM
----	------	-----	-------

SB	Startovní bajt
ADDR	Adresa zařízení
CMD	Kód příkazu 0x80 - 0xFF
PARAM	Parametr příkazu

Každý příkaz začíná startovním bajtem. Tento bajt má vždy hodnotu 0xFD a je unikátní. To znamená, že se na sběrnici nemůže objevit jindy než v okamžik, kdy se jedná opravdu o start bajt. Abych zaručil tuto unikátnost, musel jsem komunikaci navrhnout tak, aby naměřené hodnoty a ostatní informace byly posílány jako řetězce znaků. Protože ASCII hodnoty základní sady mají pouze 7 bitů, je tedy možné zvolit startovní bajt libovolně od hodnoty 0x80 až po hodnotu 0xFF. Zvolil jsem tedy hodnotu při horním okraji rozsahu, a to 0xFD.

Adresa zařízení, která je posílána hned za startovním bajtem je absolutní adresa zařízení na sběrnici. Tato adresa je fyzicky nastavena na každém koncentrátoru, na sběrnici musí být unikátní pro každé zařízení, což je striktně požadováno od správce systému a komunikační protokol tuto možnost kolize neřeší. Adresa je pěti-bitová a maximální hodnota je tedy 0x1F.

Dalším bajtem v řetězci je kód příkazu. Zde je přehled implementovaných kódů příkazů s parametrem:

Kód	Funkce
0x81	Nastavení snímače LM75A s adresou danou parametrem jako aktivní
0x82	Nastavení časové periody měření na hodnotu danou parametrem
0x83	Zkalibruje všechny snímače na teplotu danou parametrem

Posledním bajtem je parametr. Ten v závislosti na příkazu, ke kterému patří, mění svou funkci. Funkce parametrů jsou popsány v tabulce výše.

4.2.2 Příkaz bez parametru

SB	ADDR	0x00	CMD
----	------	------	-----

SB	Startovní bajt
ADDR	Adresa zařízení
0x00	Nulový bajt
CMD	Kód příkazu 0x00 - 0x7F

Stejně jako u příkazů s parametrem, i příkazy bez parametru začínají startovním bajtem. Tento bajt je unikátní, konstantní a má hodnotu 0xFD.

Druhým bajtem v pořadí koncentrátor obdrží adresu. Ta určuje adresu zařízení na sběrnici, musí být unikátní a jsou u ní kladeny požadavky na správce měřicího systému. Ten musí zajistit její unikátnost, samotný komunikační protokol není schopen rozpoznat kolizi duplikovaných adres.

V případě příkazů bez parametru je na rozdíl od příkazů s parametrem třetí přijatý bajt nulový (0x00).

Je to až čtvrtý bajt, který v případě příkazu bez parametru definuje příkaz. Zde je přehled implementovaných kódů příkazů bez parametru.

Kód	Funkce
0x01	Test aktivního zařízení (ACK = ano)
0x02	Aktivuje 1-wire sběrnici
0x03	Reset koncentrátoru (mazání paměti)
0x04	Zahájí měření koncentrátoru
0x05	Ukončení měření koncentrátoru
0x06	Žádost o poslání naměřených dat

4.3 GET řetězec

GET řetězec slouží pro získávání naměřených dat z data-koncentrátorů. K tomu slouží dvojice příkazů. Aby bylo možné naměřené hodnoty z koncentrátoru vyčíst, je potřeba znát jejich počet. K tomu slouží první příkaz, který obsahuje pouze počet naměřených hodnot. Počet naměřených hodnot je tedy omezen na velikost tří bajtů v podobě ASCII znaků, tedy 999 hodnot. Řetězec je natolik jednoduchý a krátký, že není implementována žádná kontrola. Master zařízení pouze po přijetí dvou bajtů posílá acknowledge bajt (0xA5). Správnost bajtu však není nijak kontrolována, například kontrolním redundantním součtem (CRC) nebo podobně.

0xF1	COUNT_H:M:L(3)
------	----------------

0xF1 Označení příkazu poslání počtu hodnot
COUNT_H:M:L(3) Počet hodnot k odeslání z data-koncentrátoru

Dalším GET příkazem je řetězec s hodnotou jednoho měření. Tento řetězec musí obsahovat typ snímače, ze kterého byla hodnota pořízena, jeho pořadové číslo přidělené data-koncentrátorem, čas od restartu času data-koncentrátoru a samotnou teplotu. Všechna data jsou posílána jako ASCII znak z důvodu vyhnutí se kolizím s hodnotou 0xFD, což je start bajt. Pokud by za ním shodou okolností následoval řetězec pro změnu nastavení data-koncentrátoru, došlo by k závažné chybě a postižený data-koncentrátor by mohl změnit svou funkci. Je nutné si uvědomit, že při veškeré komunikaci všechny data-koncentrátory na sběrnici stále naslouchají.

0xF2	DEV_TYP	DEV_NR_H:M:L(3)	TIME_5:0(6)	TEMP_3:0(4)
------	---------	-----------------	-------------	-------------

0xF2 Označení příkazu poslání naměřené hodnoty
DEV_TYP Typ snímače (I2C, 1-wire)
DEV_NR_H:M:L(3) Pořadové číslo snímače (udává data-koncentrátor)
TIME_5:0(6) Počet sekund v době měření od restartu času
TEMP_3:0(4) Naměřená teplota

U GET řetězce je již znát zvýšení datového toku použitím ASCII znaků. Toto opatření je ale nutné kvůli jistotě, aby se na sběrnici nevyskytovaly znaky, které jsou určeny jako unikátní. Jedná se především o start bajt následovaný adresou zařízení. Při přesunu dat z data-koncentrátoru do Raspberry Pi se na sběrnici nesmí objevit tato kombinace, aby byla zajištěna absence kolizí komunikace na sběrnici.

4.4 Hardware

Hardwarové řešení komunikace pomocí RS-485 bylo realizováno pomocí převodníku z UART na RS-485 SN75176A od firmy Texas Instruments (více v kapitole 2.6). Důvodem tohoto řešení je požadavek na komunikaci na vzdálenost desítek až stovek metrů. To je nemožné realizovat pomocí UART, kterým je vybavena jak ATmega8, tak i Raspberry Pi. Navíc u ATmega8 na straně data-koncentrátoru je použita 5 V logika a na straně Raspberry Pi 3,3 V logika. Viz kapitola 3.4, 3.5. Proto jsem použil převodník SN75176A.

Převodník SN75176A umí pracovat pouze v half-duplexním módu a je potřeba řídit přepínání toku dat. Na straně data-koncentrátoru se piny DE a \overline{RE} (více kapitola 2.6) spojily a jsou přivedeny na výstupní pin 4 portu D (Více kapitola 3.5). Tímto pinem se softwarově řídí tok dat, tedy vybírá, jestli mikrokontrolér čte nebo zapisuje na sběrnici.

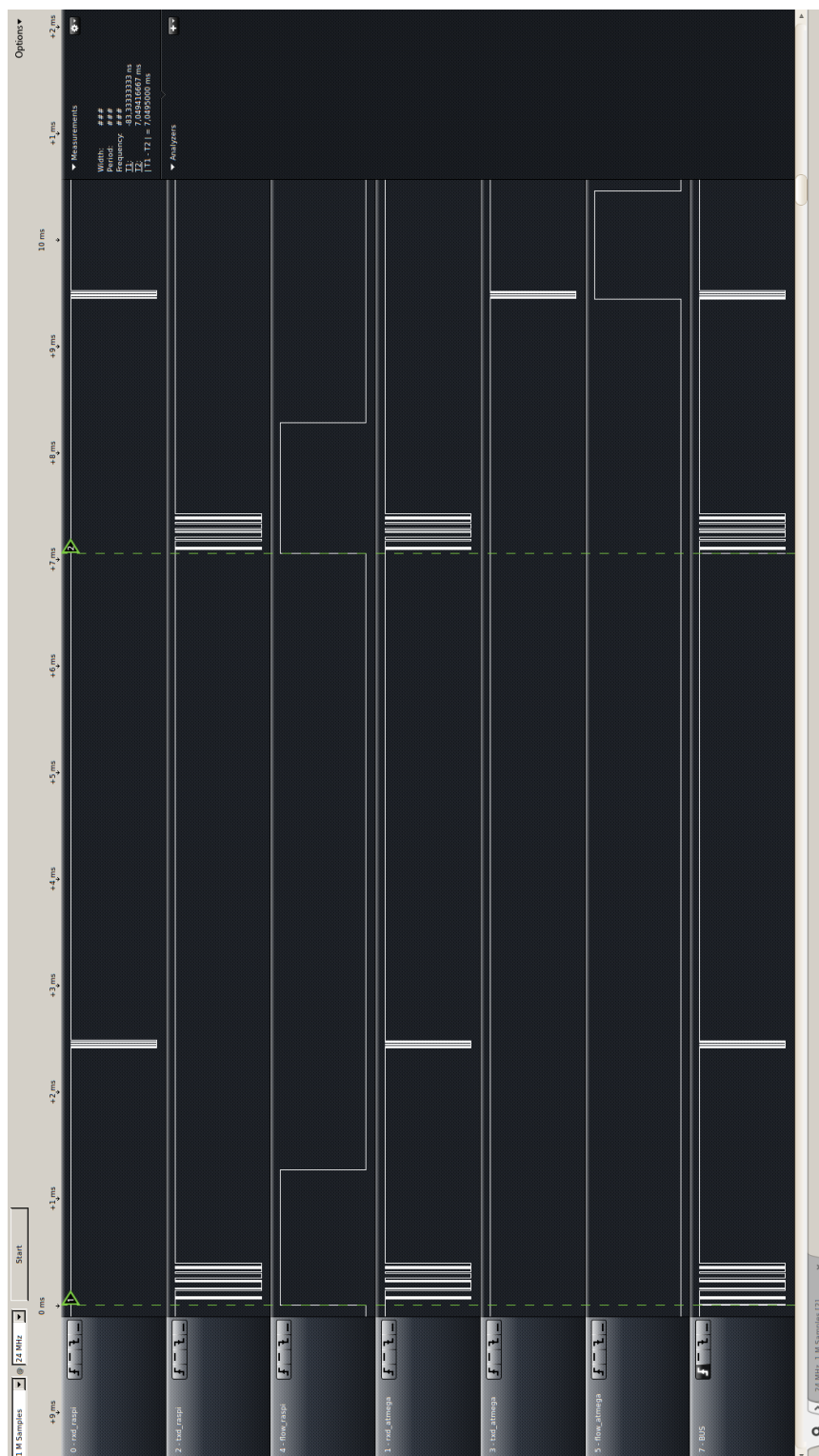
Na straně Raspberry Pi jsem postupoval obdobně. Protože Raspberry Pi využívá tři sběrnice RS-485 pro komunikaci s data-koncentrátory, využil jsem řízení toku zároveň pro výběr sběrnice.

Princip je založen na tom, že všechna zařízení naslouchají a teprve při přijetí adresy data-koncentrátoru má jedno z nich právo odpovídat. Raspberry Pi vybere sběrnici tím, že jí přepne do režimu zápisu a začne na ni posílat data. Až po odeslání dat se přepne do režimu čtení a očekává odpověď. Právě okamžik přechodu ze zápisu do čtení je kritickým místem. Musí být zajištěno, že v okamžiku otáčení směr komunikace na sběrnici, tedy že například Raspberry Pi začíná číst a data-koncentrátor naopak zapisovat a vice versa se nestane, že zařízení, které začíná na sběrnici zapisovat začne posílat data dřív, než je druhá strana přepnuta pro čtení. Poté by došlo ke ztrátě těchto dat.

Tento problém komunikace jsem vyřešil pomocí funkcí pro čekání tak, že čekání zařízení, které má zapisovat je delší než pro zařízení, které má číst. Tím je zajištěno, že nedojde ke ztrátě informace. Komunikace je fixně nastavena na rychlost 115200 BAUD. Původně jsem chtěl umožnit uživateli rychlost měnit, ale kvůli časování toku dat po sběrnici by se při změně rychlosti muselo změnit i časování změny toku dat, což by velmi zkomplikovalo implementaci této funkce.

Pro vizualizaci a ladění časů přepínání jsem použil datový analyzátor. Na obrázku 4.1 je zobrazena komunikace mezi Raspberry Pi a dvěma data-koncentrátory. Raspberry Pi postupně posílá po jednom příkazu do obou data-koncentrátorů. Jedná se o kalibraci, která je více popsána v kapitole 7.3.6. V diagramu vidíme dva téměř totožné příkazy. Důvodem, proč vidíme změnu toku dat pouze od druhého zařízení, je absence připojení pinu toku dat prvního zařízení k logickému analyzátoru. Podle časových značek jde vidět, že posílání jednoho příkazu trvá kvůli čekání při přepí-

nání toku 7 ms. Časové značky také oddělují příkazy poslané do prvního a druhého data-koncentrátoru.

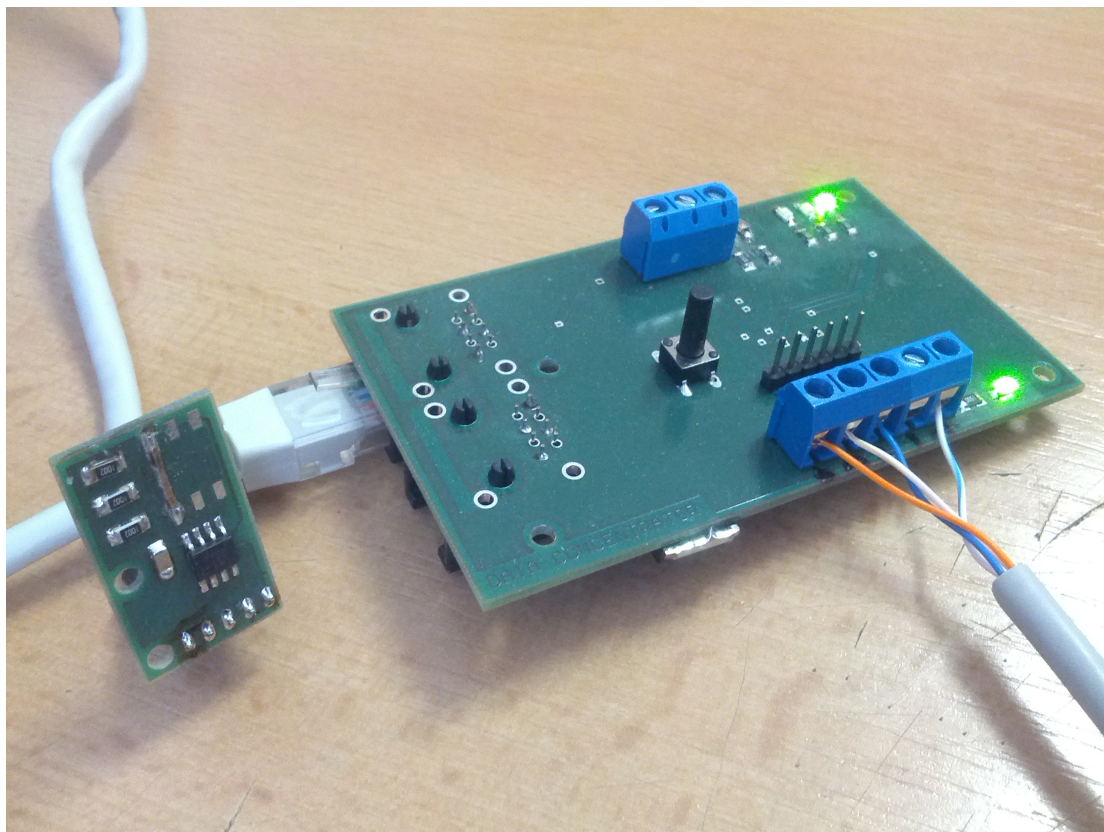


Obr. 4.1: Diagram komunikace na sběrnici RS-485

5 KONCENTRÁTOR DAT

Návrh desky plošných spojů je realizován v programu Eagle 6.5.0 Light Edition pro Linux a je popsána v kapitole 3.5.2.

Během ožívování desky jsem zjistil jediný problém, a to absenci zpětné vazby od binárního přepínače pro nastavení adresy převodníku I2C sběrnice na 1-wire sběrnici. Problém i s řešením je více popsán v kapitole 3.5.3.



Obr. 5.1: Koncentrátor dat v průběhu měření(napravo) se zapojeným snímačem LM75A(nalevo)

5.1 Programové vybavení koncentrátoru dat

Program data-koncentrátoru jsem psal v programu Kate na operačním systému Ubuntu Linux 14.10. Pro překlad a nahrávání programu do data-koncentrátoru jsem napsal Makefile, který je s ostatním softwarem obsahem DVD přiloženého k diplomové práci. Pro kompilaci programu jsem použil `avr-gcc` ve verzi 4.8.2. Pro nahrávání programu do mikrokontroléru ATmega8 jsem použil `avrdude` ve verzi 6.0.1.

Více informací o použitém softwaru je k dispozici v manuálových stránkách `man avrdude` nebo v literatuře [11] a [12]

K nahrávání programu do mikroprocesoru jsem použil programátor `STK500v2`, který jsem si vyrobil na střední škole. Jeden z důvodů, proč jsem DPS koncentrátoru neosadil JTAG konektorem je ten, že tento programátor nepodporuje JTAG, čímž jsem se připravil o možnost ladění softwaru přímo na hardware.

Program pro data-koncentrátor je napsán v jazyce C. Pro lepší přehlednost kódu jsem se jej snažil rozdělit do funkcí.

5.1.1 Komunikační protokol

Implementace komunikačního protokolu je realizována po znacích. Každý příchozí znak vyvolá přerušení, ve kterém je znak zpracován. Jediným řetězcem, který Master data-koncentrátoru posílá je řetězec příkazu. Ten má fixní délku čtyř bajtů, a proto není jeho zpracování příliš komplikované.

Data-koncentrátor na sběrnici stále naslouchá a čeká až se objeví start bajt následovaný adresou data-koncentrátoru. Pouze v tom okamžiku se v komunikaci posouvá dále a začíná zpracovávat zbytek řetězce. Funkce jednotlivých příkazů jsou popsány v kapitole 4.

5.1.2 Manipulace s daty

Stejně jako v programu pro Raspberry Pi pro měření dat `dip_measure` (kapitola 6.2.5) jsem i v programu pro mikrokontrolér definoval strukturu pro ukládání jednotlivých měření.

```
typedef struct {
    uint8_t sensor_type;
    uint32_t address;
    uint32_t time;
    uint16_t value;
} sensor_data;
```

Nahrávání dat do struktury je realizováno vždy v přerušení od časovače. 16-ti bitový časovač je nastaven na vyvolání přerušení každou vteřinu. V okamžiku přerušení se inkrementuje globální proměnná `global_seconds`, která udává počet vteřin od posledního vyčítání dat a slouží k synchronizaci času s Raspberry Pi. Inkrementuje se také statická proměnná `new_run_time`, která se po dosažení doby měřící periody data-koncentrátoru nuluje a dochází k měření teploty všemi snímači. Při měření se do záznamu ukládá čas `global_seconds`, tedy čas od synchronizace.

Ten je po odeslání dat přičten k času v Raspberry Pi, který byl také uložen při synchronizaci. Tím se dosáhne součtu času synchronizace a časového intervalu od synchronizace po měření, což dává skutečný čas měření.

Do adresy snímače `address` se ukládá adresa snímače pro který bylo měření realizováno. Měření se realizuje pouze pro snímače, které jsou pomocí webového Wizardu nastaveny jako aktivní. Tato informace se posílá pomocí příkazů z Raspberry Pi a je popsána v kapitole 7. Pokud data-koncentrátor obdrží příkaz pro aktivaci snímače, uloží hodnotu 1 do pole `lm75a_enable[index]`, ve kterém index odpovídá číslu snímače (0-7).

Získání teploty ze snímače je realizováno voláním funkce `lm75a_measure(addr)`, kde `addr` je adresa snímače, která se skládá z čísla snímače (0-7) a základu jeho I2C adresy. Tyto hodnoty jsou sečteny a výsledkem je celá adresa snímače. Základ I2C adresy snímače je v programu makro `#define LM75A_ADDR_BASE (0x48)`. Hodnota základu adresy je vyčtena z katalogového listu snímače [5].

Před spuštěním měření je provedena kalibrace. Ta probíhá tak, že po obdržení kalibrační teploty od Raspberry Pi je pro každý snímač vypočtena korekce teploty. Ta se vypočítá jako rozdíl aktuální teploty a kalibrační teploty. Tato korekce je uložena do pole `lm75_correction[index]`, kde index odpovídá číslu snímače (0-7). Při zaznamenávaném měření je hodnota korekce odečtena od změřené teploty. Tím je zajištěna kalibrace všech snímačů LM75A.

Po každém měření je inkrementován čítač hodnot k vyčtení. Tato hodnota je důležitá pro posílání dat Raspberry Pi, které očekává počet hodnot, které budou posílány.

Poslání dat do Raspberry Pi je vyřešeno velmi jednoduše. Pomocí tohoto kódu jsem přeměroval standardní výstup na UART.

```
static FILE mystdout = FDEV_SETUP_STREAM(uart_putchar, NULL,
_FDEV_SETUP_WRITE);

static int uart_putchar(char c, FILE *stream) {

    loop_until_bit_is_set(UCSRA, UDRE);
    UDR = c;
    return 0;
}

stdout = &mystdout;
```

Je nutné posílat předem definovaný řetězec obsahující hodnoty určité délky bajtů. Pokud má hodnota menší počet znaků než je definováno, je nutné zbylé znaky doplnit

nulami. Přesměrováním výstupu je k tomu snadné použít funkci `printf()`;

Díky tomu je poslání celého řetězce do Raspberry Pi otázkou jednoho řádku kódu:

```
printf("%c%d%03lu%06lu%04d", 0xF2, measure_data[i].sensor_type,  
measure_data[i].address, measure_data[i].time,measure_data[i].value);
```

Implementováno je stejně jako u Raspberry Pi přepínání toku dat po sběrnici. Zde se o přepínání mezi čtením a zápisem starají dvě funkce:

```
void uart_read_mode(void); // set rs485 bus to read mode  
void uart_write_mode(void); // set rs485 bus to write mode
```

Časování, které je rozepsáno v kapitole 4 je zahrnuto uvnitř těchto funkcí.

Jako doplňující funkce je implementováno ovládání LED.

```
void led_on(uint8_t number);  
void led_off(uint8_t number);
```

Parametrem těchto funkcí je číslo LED, která se má rozsvítit nebo zhasnout. LED jsou indexovány od nuly (0-2).

5.1.3 Variabilita snímačů

Systém je navržen pro použití s různými snímači. Pro jednodušší a přehlednější změnu snímače je i funkce pro čtení hodnoty ze snímače LM75A napsána v dalším souboru mimo `main.c`, aby byla minimalizována nutnost zásahu do hlavního souboru.

Pro LM75A se jedná o soubory `lm75a.h` a `lm75a.c`. Hlavičkový soubor obsahuje hlavičku funkce `uint16_t lm75a_measure(unsigned char lm75_address);` a makra s názvy registrů snímače. Ve zdrojovém souboru je napsáno tělo funkce `uint16_t lm75a_measure(unsigned char lm75_address);`.

6 PROGRAMY PRO RASPBERRY PI

6.1 LAMP Stack

Pro zprovoznění HTML a CSS na Raspberry Pi, tedy zpřístupnění služby http na portu 80, je nutná instalace webového serveru. Jako řešení jsem zvolil službu Apache2. Zde je její instalace:

```
pi@raspberrypi ~ \$ sudo apt-get install apache2
```

Po instalaci Apache2 je potřeba zjistit, jestli služba na Raspberry běží:

```
pi@raspberrypi ~ \$ sudo service apache2 status
Apache2 is running (pid 2555).
```

Pokud služba běží, měl by být na portu 80 dostupný http server:

```
pi@raspberrypi ~ \$ nmap localhost -p 80
```

```
Starting Nmap 6.00 ( http://nmap.org ) at 2015-05-02 01:06 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00051s latency).
Other addresses for localhost (not scanned): 127.0.0.1
PORT      STATE SERVICE
80/tcp    open  http
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.98 seconds
```

Tím je zprovozněn http server na naší platformě. Pro funkci webové aplikace je ale potřeba navíc PHP, a to kvůli volání C programů z webové aplikace a pro práci s databázemi, do kterých budou ukládány naměřené hodnoty teplot ze snímačů. Pro PHP a MySQL jsou také hotové balíčky a jejich instalace je jednoduchá. Instalace PHP:

```
pi@raspberrypi ~ \$ sudo apt-get install php5 libapache2-mod-php5
php5-mcrypt
```

Instalace MySQL:

```
pi@raspberrypi ~ \$ sudo apt-get install mysql-server php5-mysql
libapache2-mod-auth-mysql
```

Při instalaci MySQL jsem vyzván k zadání hesla pro uživatele `root`. Po instalaci je potřeba pro MySQL přidat uživatele, ať do databáze nepřistupuji jako `root`. To se provede takto:

```
pi@raspberrypi ~ $ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 51
Server version: 5.5.43-0+deb7u1 (Debian)
```

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> CREATE USER 'user'@'localhost' IDENTIFIED BY 'user';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON *.*
      -> TO 'user'@'localhost' WITH GRANT OPTION;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

Poté se přihlásím jako nový uživatel `user`, vytvořím databázi `temp` a přepnu se do ní.

```
mysql> CREATE DATABASE temp;
```

```
mysql> USE temp;
Database changed
```

Teď jsem vytvořil databázi `temp` a můžu ji začít používat pro ukládání naměřených hodnot.

Tímto jsme ukončili instalaci tak zvaného LAMP stacku. Pro celý LAMP stack jsou k dispozici balíčky. Mým úkolem byla pouze instalace těchto balíčků a jejich nastavení.

6.2 Programy v jazyce C

Programy v jazyce C jsou linkovány do systému Linux běžícího na Raspberry Pi a jsou volány jako systémové programy. Linkování je provedeno příkazem `ln`. Příklad přidání odkazu (linku) do systému:

```
pi@raspberrypi ~ $ sudo ln -s /home/pi/prog /usr/local/bin/prog
```

Protože tímto příkazem zapisujeme do systémové složky `/usr/local/bin/`, jejímž vlastníkem je `root` (to zjistím příkazem `ls -All /usr/local`):

```
drwxrwsr-x 3 root staff 4096 May  2 00:30 bin
```

je potřeba příkaz pro vytvoření linku volat jako superuser, tedy před příkazem použít `sudo`. Volat příkaz pomocí PHP skriptu jsem provedl pomocí:

```
<?php
echo shell_exec("dip_find_devices");
?>
```

Samotné volání se provádí příkazem `shell_exec()`; a příkaz `echo` pošle výstup do HTML kódu, takže výstup programu se zobrazí přímo na stránce, ve které je volán (v programu funkce `printf()`).

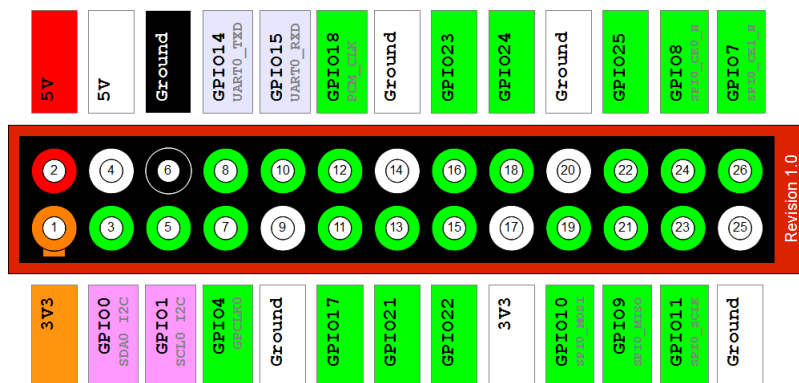
Vzhledem k tomu, že je na stránce potřeba zobrazit více informací závislých na programu, který na této stránce voláme, tento program generuje další soubory na které se na stránce odkazujeme. Proto jsou některé soubory dynamicky generovány a to buď programem psaným v jazyce C nebo PHP skriptem vycházejícím z formuláře.

6.2.1 Wiring Pi

Pro použití Raspberry Pi jako Master zařízení v diplomové práci potřebuji přistupovat k UART sběrnici. Pomocí převodníku ze sběrnice UART na diferenciální linku RS-485 je možné komunikovat s mikroprocesorovými moduly. Protože na Raspberry Pi používám tři převodníky z UART na RS-485 (jsou popsány v kapitole 2.6), které pracují v polo-duplexním módu, je potřeba řídit tok dat. To znamená, že je potřeba

jeden pin pro volbu, jestli se na převodník zapisuje nebo se z něj čte. Protože používám celkem tři tyto převodníky, potřebují také tři GPIO piny pro řízení tohoto toku dat.

Pro shrnutí tedy potřebuji UART, což jsou dva piny (RXD a TXD) a tři další piny pro řízení toku. K tomu jsem použil piny GPIO14(UART0_TXD), GPIO15(UART0_RXD) pro UART a piny GPIO22, GPIO23 a GPIO24 pro řízení toku převodníků UART na RS-485. Rozvržení pinů Raspberry Pi je na obrázku 6.1.



Obr. 6.1: Rozvržení a názvy pinů na Raspberry Pi [8]

Na Raspberry Pi ovšem běží operační systém Raspbian, což je plnohodnotný systém pro osobní počítače. Stejně jako není zvykem prostřednictvím PC ovládat GPIO piny, i zde je potřeba specifického softwaru pro přístup na tyto piny.

K tomuto účelu jsem vybral komplexní balíček s názvem `wiringPi` [9].

Balíček jsem stáhl pomocí programu GIT a sestavil pomocí příkazu `./build`. Instalace je tedy velmi jednoduchá a balík je připraven k použití. Tento balík mimo jiné obsahuje knihovny pro jazyk C pro práci s GPIO piny a hotovou knihovnu pro sériovou komunikaci využívající GPIO piny GPIO14(UART0_TXD) a GPIO15(UART0_RXD). Knihovny do programu v jazyce C stačí přidat pomocí `#include <wiringPi.h>` a `#include <wiringSerial.h>`.

Aby bylo možné přistupovat k pinům bez použití práv uživatele root, což je potřeba pro volání programu z PHP (voláme program jako uživatel `www-data`), musíme piny, se kterými chceme pracovat exportovat. To se provede pomocí příkazu `gpio export 22 output` pro nastavení pinu GPIO22 jako výstupní. To samé provedeme pro piny GPIO23 a GPIO24.

V hlavní části programu se provede inicializace pomocí `wiringPiSetupSys()`. V tomto okamžiku jsem schopen přistupovat k pinům z programu v jazyce C. Na piny zapisuji pomocí `digitalWrite(<pin>, <stav>);`,

například `digitalWrite(BUS0, HIGH);` kde `BUS0` je makro pro hodnotu 22.

Pro používání sériové sběrnice slouží tyto příkazy:

```
int serialOpen (char *device, int baud);
void serialClose (int fd);
void serialPutchar (int fd, unsigned char c);
void serialPrintf (int fd, char *message, ...);
int serialGetchar (int fd);
```

Tato část zdrojového kódu byla převzata z webových stránek [9]. Pro více informací o projektu Wiring Pi doporučuji navštívit web [9].

6.2.2 Program `dip_find_devices`

Program `dip_find_devices` je volán z PHP skriptu, který slouží pro vyhledávání data-koncentrátorů na všech sběrnicih RS-485. Po nalezení vypisuje seznam a počet data-koncentrátorů do webové stránky pro kontrolu uživatelem. Zároveň s výpisem do stránky dynamicky generuje několik souborů, které jsou používány v dalších krocích. Mezi tyto soubory patří:

- `wizard_step3.php` jako formulář pro nastavení data-koncentrátorů
- `wizard_step4.php` pro zpracování formuláře posílaného do tohoto souboru metodou POST
- `db_select.php` zobrazí generovaný seznam zařízení do menu webové stránky

Soubor `wizard_step3.php` je na základě nalezených data-koncentrátorů na sběrnicih generovaný formulář pro nastavení zařízení. Pro každý data-koncentrátor je pak vygenerována část formuláře zobrazena na obrázku 6.2. Zde uživatel zvolí I2C adresy, na kterých jsou připojeny snímače LM75A (více v kapitole 2.7), interval po kterém se opakují měření na tomto data-koncentrátoru a zapíná funkci pro detekci 1-wire snímačů.

Bus	1	Number of bus device is connected to
Address	0x01	Address of device (0x00-0x1F)
Measure interval	<input type="text" value="1"/>	Time of measure period in minutes
1-wire devices	<input type="checkbox"/>	Check if there is/are 1-wire sensor(s)
LM75A address 0	<input type="checkbox"/>	Check if LM75A connected with addr 0
LM75A address 1	<input type="checkbox"/>	Check if LM75A connected with addr 1
LM75A address 2	<input type="checkbox"/>	Check if LM75A connected with addr 2
LM75A address 3	<input type="checkbox"/>	Check if LM75A connected with addr 3
LM75A address 4	<input type="checkbox"/>	Check if LM75A connected with addr 4
LM75A address 5	<input type="checkbox"/>	Check if LM75A connected with addr 5
LM75A address 6	<input type="checkbox"/>	Check if LM75A connected with addr 6
LM75A address 7	<input type="checkbox"/>	Check if LM75A connected with addr 7

Obr. 6.2: Část formuláře generovaná pro každý data-koncentrátor

Soubor `wizard_step4.php` je určen pro zpracování vygenerovaného formuláře a na základě vstupů formuláře vytváří příkazy, které budou v dalším kroku poslány do data-koncentrátorů. Tento soubor musí být dynamicky generován proto, že nejsou známy názvy proměnných formuláře, protože i ten je generován dynamicky.

Soubor `db_select.php` je vygenerován jako seznam data-koncentrátorů, přiřazených ke sběrnicím, ke kterým jsou připojeny. Sběrnice, na kterých není připojen žádný data-koncentrátor nejsou v seznamu zobrazeny. Tento soubor je poté importován do části `menu` na levé straně webové stránky. Vzhled a provedení je znázorněno na obrázku 6.3.

```

TOPOLOGY:

BUS 1
• DEVICE 0x07

BUS 2
• DEVICE 0x01

```

Obr. 6.3: Část menu generována na základě připojených datakoncentrátorů

6.2.3 Program `dip_set_devices`

Program `dip_set_devices` má za úkol otevřít soubor `commands`, který generuje PHP skript stránky `wizard_step4.php`, který obsahuje příkazy pro nastavení data-koncentrátorů. Tyto příkazy odesílá a po přijetí každého příkazu kontroluje přijetí odpovědi od data-koncentrátoru. Pokud je odpověď `0xA5`, je příkaz přijat v pořádku.

Status odeslání všech příkazů je potom zobrazen pomocí webového rozhraní. Více o odesílání příkazů v kapitole 7.3.4.

6.2.4 Program `dip_calibrate_devices`

Po nastavení všech zařízení pomocí programu `dip_set_devices` je potřeba zkalibrovat hodnoty teplot data-koncentrátorů. K tomuto účelu jsem napsal program `dip_calibrate_devices`, kterému je zadána kalibrační teplota jako parametr a tuto hodnotu postupně posílá všem data-koncentrátorům. Stejně jako je tomu u programu `dip_set_devices` (kapitola 6.2.3) i program `dip_calibrate_devices` kontroluje přijetí kalibrační teploty data-koncentrátory a status odeslání zobrazuje prostřednictvím webového rozhraní. Více v kapitole 7.3.6.

6.2.5 Program `dip_measure`

Program `dip_measures` při svém spuštění přečte soubor `device_list`, který obsahuje časy period měření pro jednotlivé data-koncentrátory. Tyto hodnoty používá jako časovou základnu pro čtení naměřených hodnot. Čtení je prováděno každou periodu, takže hodnoty z data-koncentrátorů jsou aktualizovány velmi často. Díky tomu je sice více zatěžována sběrnice, ale výhodou je aktuální stav databáze naměřených hodnot.

Program si uloží časové konstanty a po dosažení času měření vyšle požadavek na čtení dat data-koncentrátoru. Data-koncentrátor mu pošle zpět řetězec s počtem hodnot, které má uchovány v paměti, aby věděl kolik řetězců má očekávat. Data-koncentrátor začne posílat samotné hodnoty. Více o komunikaci je popsáno v kapitole 4.

6.2.6 Makefile

Pro zjednodušení kompilace programů jsem napsal jednoduchý **Makefile**. Tento **Makefile** obsahuje v podstatě jen příkazy pro kompilaci jednotlivých programů. Protože téměř všechny zdrojové kódy jsem psal přímo na Raspberry Pi v konzolovém editoru `nano`, alespoň kompilaci jsem si tímto způsobem zjednodušil.

Pro každý program je v souboru **Makefile** pravidlo pro jeho vytvoření. Výhodou je, že je jedno jaký program jsem právě upravil, pokud došlo ke změně souboru zahrnutého v pravidle pro jeho vytvoření, dojde k jeho kompilaci. Stačí ve stejné složce kde je uložen **Makefile** napsat příkaz `make` a všechny programy, kterým byly změněny zdrojové kódy se znovu zkompilují. Další výhoda je v tom, že například balíček Wiring Pi (popsán v kapitole 6.2.1) požaduje pro kompilaci přidání parametru

-l wiringPi. Pokud ale použiji Makefile, vložím parametr do pravidla pro vytvoření programu pouze jednou a kompiluji pomocí `make`. Zde je část Makefile zobrazující pravidlo pro vytvoření `dip_find_devices`:

```
dip_find_devices : dip_find_devices.c
    gcc -Wall dip_find_devices.c -o dip_find_devices -l wiringPi
```

Pokud tedy upravím soubor `dip_find_devices.c`, příkaz `make` vykoná příkaz `gcc -Wall dip_find_devices.c -o dip_find_devices -l wiringPi -std=c99`

Popsal jsem zde pouze nejzákladnější používání souboru Makefile. Pro více informací doporučuji začít zadáním příkazu `man make` a literaturou [10].

7 WEBOVÉ ROZHRAŇÍ

Webové rozhraní sice spadá do kapitoly 6, ale řešení a jeho popis je natolik komplikované a rozsáhlé, že jsem se rozhodl věnovat mu celou kapitolu.

Pro přístup uživatele k naměřeným datům, kontrolou nad měřícím systémem, nastavení jednotlivých datových koncentrátorů a uživatelsky příjemnému zobrazení naměřených dat jsem napsal webovou aplikaci. K vytvoření jsem použil balíček LAMP. Ten se skládá z kombinace Linux, Apache, MySQL, PHP. Apache slouží k funkci HTML a CSS kódů. PHP a MySQL je potřeba kvůli databázi pro naměřené hodnoty. Mimo LAMP jsem napsal několik programů v jazyce C, které jsou volány z webové aplikace pomocí PHP skriptů a slouží především k obsluze hardwarové části systému.

Mezi programy psanými v jazyce C a v jazyce PHP jsou velmi úzké vazby. V následujících kapitolách se budu snažit popsat jejich provázanost a funkci jednotlivých částí. Pro pochopení funkce webového rozhraní je nutné pročtení a pochopení celé kapitoly 7. Hlavními prvky webového rozhraní jsou:

- Vyhledání datakoncentrátorů na sběrnících
- Nastavení nalezených datakoncentrátorů
- Kalibrace všech datakoncentrátorů
- Spuštění měření kalibrovaných datakoncentrátorů
- Zobrazení naměřených hodnot

Při tvorbě webového rozhraní a pro práci s databázemi jsem čerpal převážně z webové stránky w3schools [13].

7.1 Vzhled WWW stránek

Jedná se o stránku na které by měl pracovat správce systému. I přes snahu jsem nebyl schopen ji navrhnout dostatečně jednoduchou pro běžného uživatele. Tato nemožnost zjednodušení je zaviněna komplexností systému a vazbou na hardware, který se mění a nelze proto s jistotou popsat, co přesně má uživatel nastavit. Pro každého uživatele bude nastavení rozdílné. Hardwarová nastavení adres snímačů převyšuje znalosti laického uživatele, už jen kvůli použití binárního kódu.

Přesto jsem se snažil udělat stránku co nejjednodušší a uživatelsky přívětivou.

Hlavní stránka (`index.php`) je realizována pomocí **frames** a je rozdělena do tří částí, a to **head**, **menu** a **body**.

Webové stránky jsou napsány a odladěny pro webový prohlížeč Mozilla Firefox verze 38.0.

Decentralized temperature measuring system

- 3 busses based decentralised system
- Each bus contains up to 32 data-concentrating devices
- Each device operates with i2c and 1-wire temperature sensors

Masters degree Thesis
Martin Vladar, 2015, Version 1.3.1

[WIZARD](#)

[REFRESH](#)

[ALL VALUES](#)

[CUSTOM VALUES](#)

[ERASE VALUES](#)

TOPOLOGY:

BUS 1

- DEVICE 0x07

BUS 2

- DEVICE 0x01

PHOTOS

- [NIGHT PHOTO](#)
- [DAY PHOTO](#)
- [CUSTOM PHOTO](#)
- [PHOTO LIBRARY](#)

[FILES](#)

ID	Bus	Device address	Sensor type	Sensor address	Time	Temperature [°C]
1	2	1	1	2	2015-05-16 05:48:16	19
2	2	1	1	5	2015-05-16 05:48:16	19.5
3	1	7	1	0	2015-05-16 05:49:01	21
4	1	7	1	1	2015-05-16 05:49:01	21
5	2	1	1	2	2015-05-16 05:50:16	19
6	2	1	1	5	2015-05-16 05:50:16	19.5
7	2	1	1	2	2015-05-16 05:52:16	19
8	2	1	1	5	2015-05-16 05:52:16	19.5
9	1	7	1	0	2015-05-16 05:53:01	21
10	1	7	1	1	2015-05-16 05:53:01	21
11	2	1	1	2	2015-05-16 05:54:16	19
12	2	1	1	5	2015-05-16 05:54:16	19.5
13	2	1	1	2	2015-05-16 05:56:16	19
14	2	1	1	5	2015-05-16 05:56:16	19.5
15	1	7	1	0	2015-05-16 05:57:01	21
16	1	7	1	1	2015-05-16 05:57:01	21
17	2	1	1	2	2015-05-16 05:58:16	19
18	2	1	1	5	2015-05-16 05:58:16	19.5

Obr. 7.1: Vzhled webové stránky

7.2 Struktura WWW stránek

Stránka je rozdělena do tří částí. Mezi tyto části patří:

- Head (hlavička stránky). Jedná se o statický HTML kód
- Menu, které je z části generováno dynamicky
- Body (tělo stránky), které jak je zvykem, slouží pro zobrazení obsahu stránky

7.2.1 Hlavička

Head nebo také hlavička dokumentu je velmi jednoduchá a obsahuje pouze základní údaje o názvu práce, autorovi a velmi stručný popis. Jak název napovídá, je situována v horní části stránky, zabírá 100% šířky stránky a je vysoká 150 pixelů.

Zbytek stránky je rozdělen do dalších dvou rámců. Rámec menu je široký přesně 175 pixelů a rámec body zabírá zbytek šířky stránky.

7.2.2 Menu

Menu je situováno do levé části stránky, jak je zvykem u většiny stránek. Jsou zde přístupné nástroje pro nastavení systému, kontrolou nad systémem a přístup k naměřeným hodnotám. Rámec body je využíván k zobrazování obsahu. Je zde zobrazován například celý postup k nastavení všech zařízení krok za krokem. Další funkcí je zobrazování naměřených hodnot, ať už z jednotlivých zařízení, jednotlivých sběrnic nebo všech sběrnic.

Stránky jsou stejně jako všechny zdrojové kódy napsány v anglickém jazyce.

Část menu, která je určena pro zobrazování topologie systému, je generována dynamicky. Jedná se však pouze o část menu, která je do zbytku importována. Tím je myšlen seznam zařízení a sběrnic, na které jsou připojeny. Protože Wizard (kapitola 7.3) dokáže prohledat sběrnice a nalézt na nich data-koncentrátory, je tento seznam obsazených sběrnic a data-koncentrátorů škoda nevyužít pro přehledné zobrazení topologie systému. Po vyhledání data-koncentrátorů je proto programem `dip_find_devices` generován jednoduchý PHP skript, který je do menu importován, a topologie systému je přehledně viditelná přímo v menu webového rozhraní.

7.3 Wizard

Menu obsahuje všechny prvky potřebné pro správu systému. Hlavním a nejobsáhlejším prvkem je Wizard. Ten uživatele postupně provede sérií kroků pro nastavení měřicího systému. Z důvodu rozdílného nastavení a kombinace snímačů pro každou aplikaci systému je velká část kódu dynamicky generována. S výhodou jsem zde používal generovaný kód. Například seznam příkazů `commands` (kapitola 7.3.4) je generován pomocí PHP kódu v souboru `wizard_step4.php` a samotný soubor `wizard_step4.php` je generován pomocí programu `dip_find_devices` psaného v jazyce C na základě počtu data-koncentrátorů a jejich adres.

7.3.1 Krok 1

V prvním kroku je správce vyzván ke správné instalaci a kontrole systému. V tomto kroku je kladen důraz na správné zapojení a ukončení všech sběrnic RS-485. Jednotlivá varování jsou následující:

- Varování před připojením systému do sítě Ethernet. Varování je zde z důvodu použití kabelů CAT6, se kterými se setkáváme právě v této síti
- LED označená PWR (kapitola 3.5.2) musí svítit na všech data-koncentrátorech
- Odpor ukončující vedení musí být zapnut pouze u zařízení na konci sběrnice (Pozná se tak, že je k němu připojen pouze vstupní kabel CAT6)

- Adresu převodníku DS2482 změníme pouze v případě nefunkčnosti
- Je nutné se ujistit, že adresa každého zařízení je unikátní. Ta se nastavuje přepínačem RS485ADDR (kapitola 3.5.2)

7.3.2 Krok 2

V tomto kroku se z PHP spouští program C s názvem `dip_find_devices`, který prohledává všechny adresy (`0x00 - 0x1F`) na všech třech sběrnicích. Kvůli čekání na odpověď od připojených zařízení je toto vyhledávání časově nejnáročnějším prvkem nastavování. I v nejhorším případě však nepřesáhne dobu deseti vteřin.

Po vyhledávání je zobrazen seznam všech zařízení, která byla nalezena. V tomto kroku je uživatel vyzván ke kontrole, jestli některé zařízení v seznamu nechybí. Pokud ano, je vyzván ke kontrole zapojení a úkonům z Kroku 1. Pokud je vše v pořádku a seznam obsahuje všechna fyzicky připojená zařízení, postoupí se k dalšímu kroku.

Welcome to device recognition wizard

Heureka! There is a device on bus0 with address 0x01 alive!
Heureka! There is a device on bus1 with address 0x07 alive!

2 device(s) found!

If there is any device missing, please follow instructions and try again:

- Make sure all devices are connected to measuring system, not ethernet.
 - POWER ON LEDs have to be ON.
 - The terminator have to be ON if device ends this bus (OUT not connected).
 - Make sure address of each device is unique.
- Two or more devices with same addresses will not work together!
-

Obr. 7.2: Seznam nalezených zařízení pomocí Wizardu

7.3.3 Krok 3

Pro každý nalezený data-koncentrátor je generován formulář pro jeho následné nastavení. Tato generace je závislá na počtu a adresách nalezených data-koncentrátorů. Pro každý data-koncentrátor je vygenerován formulář obsahující proměnné s prefixem odpovídajícím sběrnici a adrese data-koncentrátoru. Pro data-koncentrátor na sběrnici 1 s adresou `0x07` mají proměnné prefix `1_7_`, takže mají název například `1_7_timeout`.

Generated form for 2 device(s)

Bus	0	Number of bus device is connected to
Address	0x01	Address of device (0x00-0x1F)
Measure interval	<input type="text" value="1"/>	Time of measure period in minutes
1-wire devices	<input type="checkbox"/>	Check if there is/are 1-wire sensor(s)
LM75A address 0	<input type="checkbox"/>	Check if LM75A connected with addr 0
LM75A address 1	<input type="checkbox"/>	Check if LM75A connected with addr 1
LM75A address 2	<input type="checkbox"/>	Check if LM75A connected with addr 2
LM75A address 3	<input type="checkbox"/>	Check if LM75A connected with addr 3
LM75A address 4	<input type="checkbox"/>	Check if LM75A connected with addr 4
LM75A address 5	<input type="checkbox"/>	Check if LM75A connected with addr 5
LM75A address 6	<input type="checkbox"/>	Check if LM75A connected with addr 6
LM75A address 7	<input type="checkbox"/>	Check if LM75A connected with addr 7

Bus	1	Number of bus device is connected to
Address	0x07	Address of device (0x00-0x1F)
Measure interval	<input type="text" value="1"/>	Time of measure period in minutes
1-wire devices	<input type="checkbox"/>	Check if there is/are 1-wire sensor(s)
LM75A address 0	<input type="checkbox"/>	Check if LM75A connected with addr 0
LM75A address 1	<input type="checkbox"/>	Check if LM75A connected with addr 1
LM75A address 2	<input type="checkbox"/>	Check if LM75A connected with addr 2
LM75A address 3	<input type="checkbox"/>	Check if LM75A connected with addr 3
LM75A address 4	<input type="checkbox"/>	Check if LM75A connected with addr 4
LM75A address 5	<input type="checkbox"/>	Check if LM75A connected with addr 5
LM75A address 6	<input type="checkbox"/>	Check if LM75A connected with addr 6
LM75A address 7	<input type="checkbox"/>	Check if LM75A connected with addr 7

Obr. 7.3: Generovaný formulář pro dva připojené data-koncentrátory

7.3.4 Krok 4

Nastavení je z formuláře pomocí metody `post` posíláno PHP skriptu, který ze správcem nastavených hodnot vygeneruje příkazy pro odeslání do jednotlivých zařízení. Jedná se o generovaný PHP kód také pomocí programu `dip_find_devices`.

Tento PHP skript jednak vypisuje přímo na stránku uživatelsky čitelný seznam příkazů, které jsou generovány pomocí nastavení formuláře a jednak generuje soubor `commands`. Tento soubor obsahuje pro uživatele již špatně čitelné řetězce pro nastavení jednotlivých data-koncentrátorů. Každý řádek reprezentuje jeden z příkazů pro nastavení (více v kapitole 4.2) a je ukončen znakem `P`. V tomto kroku je uživatel opět vyzván ke kontrole, proto je zde zobrazen pro uživatele čitelný seznam příkazů. Po kontrole je možné vrátit se k formuláři pro nastavení nebo pokračovat. Tím se jednotlivé příkazy odešlou do data-koncentrátorů kterým přísluší.

Please make sure these are commands you want to execute:

On bus0 device 0x01 set measure refresh to 20 minutes.
On bus0 device 0x01 set LM75A with address 0x00 active.
On bus0 device 0x01 set LM75A with address 0x01 active.
On bus1 device 0x07 set measure refresh to 30 minutes.
On bus1 device 0x07 set LM75A with address 0x03 active.
On bus1 device 0x07 set LM75A with address 0x05 active.

[BACK TO FORM](#)

[NEXT](#)

Obr. 7.4: Seznam příkazů pro data-koncentrátory pro kontrolu

Generovaný soubor `commands` pro srovnání vypadá takto:

```
00FD018214
00FD018100
00FD018101
01FD07821E
01FD078103
01FD078105
P
```

7.3.5 Krok 5

Z PHP skriptu je opět volán program psaný v jazyce C, který otevírá soubor `commands` a jeho obsah zpracovává, tedy odesílá příkazy v něm obsažené odpovídajícím zařízením. Pro tento účel je napsán program `dip_set_devices`. Při odeslání každého příkazu samozřejmě očekává odpověď a na základě této odpovědi vypisuje status odeslání příkazů do data-koncentrátorů na stránku. V případě chyby je z tohoto kroku opět možný návrat k nastavovacímu formuláři a hodnoty opravit nebo zkusit formulář odeslat znovu.

Výhodou provádění úkonů krok po kroku je jednak dobrá kontrola uživatelem a minimalizace chyb nastavení, které se postupně řeší velmi snadno, a jednak každý přístup do souboru a spouštění programů probíhá postupně. Postupný přístup do souborů je výhodný díky tomu, že odpadá nutnost používat mutexy nebo semaforey pro kontrolu nad přístupy do souborů.

Command send status
Please make sure all commands were accepted

FD 01 82 14 to bus 0 send FAILED
FD 01 81 00 to bus 0 send FAILED
FD 01 81 01 to bus 0 send FAILED
FD 07 82 1E to bus 1 send OK
FD 07 81 03 to bus 1 send OK
FD 07 81 05 to bus 1 send OK

[SET AGAIN](#)

[NEXT](#)

Obr. 7.5: Status odeslání pro jednotlivé příkazy (První zařízení v pořadí pro demonstraci odpojeno)

Zde jsou příkazy zobrazeny ve tvaru řetězců, které sice nejsou pro uživatele tak dobře čitelné, ale hlavní informací je zde status. Ten musí být vždy OK a pokud není, není důležité ve kterém z příkazů došlo k chybě, protože je v každém případě potřeba příkazy odeslat znovu.

7.3.6 Krok 6

Pokud jsou příkazy do data-koncentrátorů poslány v pořádku, je potřeba snímače na data-koncentrátorech kalibrovat. K tomu slouží šestý krok Wizardu. Zde je uživatel vyzván k zadání teploty, ve které se právě snímače nachází. Nevýhodou je potřeba mít všechny snímače na jednom místě, ale pro takto rozsáhlý a variabilní systém s proměnným počtem snímačů je to bohužel nutnost. Výhodou tohoto řešení je, že si uživatel může vybrat druh kalibrace. Pokud potřebujeme záznam o absolutní teplotě, provede kalibraci snímačů podle přesného přístroje. Snímače lze ale také zkalibrovat na libovolnou teplotu, například 0 °C a výsledkem bude odchylka od této teploty. Snímače v tomto případě nemusejí být kalibrovány na stejnou teplotu a systém bude zaznamenávat teploty relativní.

Calibration page

Please move all sensors to a place with known temperature for at least a few minutes.
In best case, use calibrated device to get this temperature.

Please write this temperature below and calibrate all sensors
to this temperature by pressing "NEXT" button.

Temperature (Celsius) to calibrate the system to:

NEXT

Obr. 7.6: Vstup pro kalibrační teplotu

7.3.7 Krok 7

V sedmém kroku následuje kontrola kalibrace. Je zde ošetřeno zadání teploty v rozsahu bezznaménkového osmibitového integeru a kontrola přijetí kalibračních dat data-koncentrátorem. V případě chyby je možné se podobně jako u kroku 2 (kapitola 7.3.2), 4 (kapitola 7.3.4) nebo 5 (kapitola 7.3.5) vrátit zpět a kalibraci opakovat.

Z této stránky se pomocí PHP skriptu opět volá program napsaný v jazyce C. Jeho název je tentokrát `dip_calibrate_devices` a slouží výhradně ke kalibraci data-koncentrátorů. Ze souboru `device_list` načítá čísla sběrnic a adresy data-koncentrátorů, které jsou nalezeny. Tento soubor je generován spolu se souborem `commands` skriptem `wizard_step4.php`, který je také generován, a to programem `dip_find_devices`.

Je zde ale malý rozdíl oproti jiným C programům, které jsou volány z PHP. Program `dip_calibrate_devices` je totiž volán s parametrem. Protože soubor `device_list` je generován již ve čtvrtém kroku Wizardu (kapitola 7.3.4), ale kalibrační teplota je známa až v sedmém kroku, rozhodl jsem se vygenerovat příkazy do souboru `device_list` bez posledního parametru (kalibrační teploty) a posílat jej jako parametr programu. Volání programu s parametrem z PHP skriptu:

```
$string = sprintf("dip_calibrate_devices %d", $_POST['calibrate']);  
echo shell_exec($string);
```

Příkaz `echo` vypisuje výstup z programu na stránku, takže téměř všechn obsah stránky pochází z programu `dip_calibrate_devices` a PHP kód má pouze pár řádků. Výstup programu vypsáný do stránky potom vypadá takto:

Calibration status
Please make sure all devices calibrated successfully

Calibrating all devices to 20 degrees Celsia...

Calibration of device 01 on bus 1 OK
Calibration of device 07 on bus 1 OK

Clicking NEXT will start measuring system

[CALIBRATE AGAIN](#)

[NEXT](#)

Obr. 7.7: Status kalibrace data-koncentrátorů (První v pořadí pro demonstraci odpojen)

7.3.8 Krok 8

V posledním kroku získáváme pouze informaci o spuštění systému. Kliknutím na odkaz NEXT se dostaneme na tabulku zobrazující databázi naměřených hodnot. Měření začíná spuštěním programu `dip_measure` na pozadí.

Congratulations!
Measure system configuration completed!

System is running
Please avoid Wizard if there are no changes in system

[NEXT](#)

Obr. 7.8: Informační stránka o začátku měření

7.4 Zobrazování naměřených hodnot

Pro ukládání naměřených hodnot slouží databáze. Zvolil jsem databázi MySQL, pro jejíž instalaci jsou ve standardních repositářích dostupné balíčky, je dobře provázána s PHP a lze s ní pracovat i z jiných programovacích jazyků pomocí `query` příkazů.

Pro zobrazení naměřených hodnot slouží odkaz `ALL VALUES` pro zobrazení všech naměřených hodnot na všech sběrnicích, nebo odkaz `CUSTOM VALUES`, který slouží k vybrání sběrnice, data-koncentrátoru nebo adresy snímače LM75A, pro které se mají zobrazit naměřené hodnoty. Databázi je možné z webového rozhraní také smazat pomocí odkazu `ERASE VALUES`

Tyto odkazy je možné vidět na obrázku 7.1.

7.4.1 Správa databáze

Instalaci MySQL serveru a vytvoření databáze jsem provedl a popsal v kapitole 6.1. V této kapitole pokročím dále a budu se věnovat samotné tabulce pro hodnoty, ukládání hodnot do této tabulky a jejich zpětnému vyčítání pro zobrazení.

Příkaz pro tvorbu tabulky `temperature` pro ukládání naměřených hodnot:

```
mysql> CREATE TABLE temperature( TempID INT PRIMARY KEY  
AUTO_INCREMENT, bus INT, dev_addr INT, sens_type INT,  
sens_addr INT, time DATETIME, value FLOAT);
```

Pro přidání hodnoty do tabulky:

```
mysql> INSERT INTO temperature (bus, dev_addr, sens_type, sens_addr,  
time, value) VALUES(bus_nr, dev_addr_nr, sens_type_nr, sens_addr_nr,  
time_stamp, value_nr);
```

Například:

```
mysql> INSERT INTO temperature (bus, dev_addr, sens_type, sens_addr,  
time, value) VALUES(1, 7, 1, 4, '2015-05-13 20:14:35', 23.5);
```

Pro hodnotu snímače na zařízení sběrnice 1, s adresou 7, typem snímače 1, adresou snímače 4 v čase '2015-05-13 20:14:35' a teplotou na snímači 23,5 °C.

Tento řetězec je potřeba generovat z programu `dip_measure`.

Měření programem probíhá tak, že se obsah každého řetězce z data-koncentrátoru přepíše do struktury, která obsahuje přehledně rozdělené informace o měření hodnoty.

Číslo sběrnice a adresa data-koncentrátoru je nutná pro správný zápis teploty do databáze. Tyto hodnoty nejsou získávány z data-koncentrátoru, ale z Raspberry Pi, které v sobě drží tuto hodnotu. Na základě toho také směřuje příkaz požadavku na posílání naměřených hodnot z data-koncentrátorů.

Typ a adresa snímače jsou první hodnoty, které jsou od data-koncentrátoru získány.

Struktura vypadá takto:

```
typedef struct {  
    uint8_t bus;           // číslo sběrnice  
    uint8_t device;        // adresa data-koncentrátoru  
    uint8_t sens_type;     // typ snímače  
    uint16_t sens_addr;    // adresa snímače (přidělena data-koncentrátorem)  
    uint32_t time;         // čas  
    float value;           // změřená hodnota  
} record;
```

Typ jsem přidal kvůli možnosti zapojení velké škály snímačů ať už na I2C nebo 1-wire sběrnici (viz kapitola 2). Snímače tedy bude pravděpodobně nutné odlišit a proto jsem přidal tento parametr. Adresa snímače nemusí odpovídat skutečné adrese (například I2C adrese) snímače. Tuto adresu přiděluje data-koncentrátor a jsou pro ni vyhrazeny tři bajty. Jak víme z kapitoly 4.3, na sběrnici RS-485 jsou posílány ASCII hodnoty znaků kvůli kolizím a proto je tedy maximální adresa snímače "999", což s rezervou převyšuje reálně odhadovaný počet (desítky snímačů).

Prvek struktury s názvem `time` je hodnota času, která uplynula od posledního vyčítání nebo od posledního restartu času v data-koncentrátoru. Pro synchronizaci času s minimální odchylkou je použito jednoduché metody.

Je důležité, aby čas odpovídal momentu měření a ne až vyčítání této hodnoty. Tato metoda spočívá v synchronizaci času při vyčtení paměti data-koncentrátoru nebo při jeho mazání. V tomto okamžiku se nuluje časovač data-koncentrátoru, který počítá vteřiny a zároveň se v programu `dip_measure` ukládá současný systémový čas separovaně pro každý data-koncentrátor. Separovaně z toho důvodu, že jednotlivé data-koncentrátory mohou mít rozdílné periody měření.

Ve stejný okamžik se tedy nuluje čas data-koncentrátoru a je uložen systémový čas Raspberry Pi.

Data-koncentrátor pak pro každé měření ukládá kromě samotné teploty i informaci o čase, který uplynul ve vteřinách od posledního vyčítání hodnot. Po odeslání řetězce do Raspberry Pi se v programu `dip_measure` přičte k uloženému systémovému času hodnota vteřin z data-koncentrátoru a výsledný čas odpovídá době měření.

Hodnota uplynulých vteřin se posílá na šesti bajtech, odpovídá tedy maximální délce "999 999" ve vteřinách, což je doba odpovídající více než jedenácti dnům. Tímto je omezena maximální doba bez vyčtení hodnot z data-koncentrátoru. Vyčtení by sice bylo možné, ale čas v databázi by byl u hodnoty posunut o zmiňovaných jedenáct dnů dozadu kvůli přetečení čísla "999 999".

Samotnou hodnotu teploty jsem se rozhodl posílat ve tvaru, který je vyčítán přímo ze snímače. Pro toto řešení jsem se rozhodl kvůli úspoře bajtů při odesílání, protože již oddělení desetinné a celé části čísla by zabralo jeden bajt navíc v podobě desetinné tečky a samotná skutečnost, že se jedná o desetinná čísla naznačuje, že počítání a manipulace s nimi bude pro mikrokontrolér bez FPU komplikovaná a časově náročná. O přepočtení na desetinné číslo interpretující teplotu se proto stará až program `dip_measure` běžící na Raspberry Pi.

Po odeslání hodnot do Raspberry Pi data-koncentrátor nuluje čítač naměřených hodnot a čítač vteřin od posledního čtení. To znamená, že nové hodnoty ukládá znovu od indexu 0 a hodnotu `time` také počítá opět od nuly. V programu `dip_measure` se opět ukládá systémový čas pro novou synchronizaci a cyklus se uzavírá.

7.4.2 Odkaz ALL VALUES

Odkazem `ALL VALUES` je možné zobrazit do frame body PHP stránku s názvem `write_db.php`, která se připojí do MySQL, vybere databázi `temp` a v ní z tabulky `temperature` vypíše všechny hodnoty. Zde je zkrácený výpis kódu, který se stará o přístup do databáze a výběr všech hodnot z tabulky `temperature`:

```
$servername = "localhost";
$username = "user";
$password = "user";
$dbname = "temp";
$conn = new mysqli($servername, $username, $password, $dbname);

$sql = "SELECT * FROM temperature";
$result = $conn->query($sql);
//zpracování $result
$conn->close();
```

Po získání `$result` pro každý řádek databáze přidám řádek tabulky. Tato tabulka je stylizována pomocí kaskádových stylů a obsahuje všechny hodnoty nacházející se v databázi naměřených hodnot. Více o této databázi v kapitole 7.4.1.

Příkaz `ALL VALUES` slouží k zobrazení všech naměřených hodnot.

Decentralized temperature measuring system

- 3 busses based decentralised system
- Each bus contains up to 32 data-concentrating devices
- Each device operates with i2c and 1-wire temperature sensors

Masters degree Thesis
Martin Vládar, 2015, Version 1.3.1

[WIZARD](#)

[REFRESH](#)

[ALL VALUES](#)
[CUSTOM VALUES](#)
[ERASE VALUES](#)

BUS 1

- DEVICE 0x07

BUS 2

- DEVICE 0x01

PHOTOS

- [NIGHT PHOTO](#)
- [DAY PHOTO](#)
- [CUSTOM PHOTO](#)
- [PHOTO LIBRARY](#)

[FILES](#)

ubuntu®
14.10
DOWNLOAD NOW ›

ID	Bus	Device address	Sensor type	Sensor address	Time	Temperature [°C]
1	2	1	1	2	2015-05-15 22:32:12	22
2	2	1	1	5	2015-05-15 22:32:12	22
3	1	7	1	0	2015-05-15 22:32:58	16.5
4	1	7	1	1	2015-05-15 22:32:58	17
5	2	1	1	2	2015-05-15 22:34:12	10
6	2	1	1	5	2015-05-15 22:34:12	12
7	2	1	1	2	2015-05-15 22:36:12	8
8	2	1	1	5	2015-05-15 22:36:12	9
9	1	7	1	0	2015-05-15 22:36:57	8.5
10	1	7	1	1	2015-05-15 22:36:57	10
11	2	1	1	2	2015-05-15 22:38:12	7.5
12	2	1	1	5	2015-05-15 22:38:12	8
13	2	1	1	2	2015-05-15 22:40:12	15.5
14	2	1	1	5	2015-05-15 22:40:12	16.5
15	1	7	1	0	2015-05-15 22:40:57	19
16	1	7	1	1	2015-05-15 22:40:57	18.5
17	2	1	1	2	2015-05-15 22:42:12	18
18	2	1	1	5	2015-05-15 22:42:12	18.5
19	2	1	1	2	2015-05-15 22:44:12	19
20	2	1	1	5	2015-05-15 22:44:12	19.5
21	1	7	1	0	2015-05-15 22:44:57	21.5
22	1	7	1	1	2015-05-15 22:44:57	21
23	2	1	1	2	2015-05-15 22:46:13	19.5
24	2	1	1	5	2015-05-15 22:46:13	20
25	2	1	1	2	2015-05-15 22:48:13	19.5
26	2	1	1	5	2015-05-15 22:48:13	20
27	1	7	1	0	2015-05-15 22:48:58	22
28	1	7	1	1	2015-05-15 22:48:58	21.5
29	2	1	1	2	2015-05-15 22:50:13	19.5
30	2	1	1	5	2015-05-15 22:50:13	20
31	2	1	1	2	2015-05-15 22:52:13	19.5
32	2	1	1	5	2015-05-15 22:52:13	20
33	1	7	1	0	2015-05-15 22:52:58	22
34	1	7	1	1	2015-05-15 22:52:58	21.5
35	2	1	1	2	2015-05-15 22:54:13	19.5
36	2	1	1	5	2015-05-15 22:54:13	20
37	2	1	1	2	2015-05-15 22:56:13	19.5
38	2	1	1	5	2015-05-15 22:56:13	20
39	1	7	1	0	2015-05-15 22:56:58	22
40	1	7	1	1	2015-05-15 22:56:58	21.5
41	2	1	1	2	2015-05-15 22:58:13	19.5
42	2	1	1	5	2015-05-15 22:58:13	20
43	2	1	1	2	2015-05-15 23:00:13	19.5

Obr. 7.9: Zobrazení všech záznamů databáze

7.4.3 Odkaz CUSTOM VALUES

Odkaz CUSTOM VALUES návštěvníka odkazuje na stránku s formulářem pro výběr specifických hodnot z databáze. Je možné měnit sběrnici, adresu data-koncentrátoru a adresu snímače LM75A pro který si uživatel přeje zobrazit naměřené hodnoty. Tlačítkem SHOW se poté na základě formuláře vygeneruje query příkaz, pomocí kterého se vyčtou data a zobrazí se.

Custom data list. Select data parameters to show.

BUS Select bus
 Device address Select device address
 Sensor address Select LM75A sensor address

Obr. 7.10: Formulář pro výběr specifických dat z databáze

Na obrázku 7.11 je zobrazená část výpisu po vybrání snímače 5 ve formuláři na obrázku 7.10. Výpis odpovídá výběru hodnot pro snímač 5 z obrázku 7.9, což je dobře patrné na hodnotě ID.

ID	Bus	Device address	Sensor type	Sensor address	Time	Temperature [°C]
2	2	1	1	5	2015-05-15 22:32:12	22
6	2	1	1	5	2015-05-15 22:34:12	12
8	2	1	1	5	2015-05-15 22:36:12	9
12	2	1	1	5	2015-05-15 22:38:12	8
14	2	1	1	5	2015-05-15 22:40:12	16.5
18	2	1	1	5	2015-05-15 22:42:12	18.5
20	2	1	1	5	2015-05-15 22:44:12	19.5
24	2	1	1	5	2015-05-15 22:46:13	20
26	2	1	1	5	2015-05-15 22:48:13	20
30	2	1	1	5	2015-05-15 22:50:13	20
32	2	1	1	5	2015-05-15 22:52:13	20
36	2	1	1	5	2015-05-15 22:54:13	20
38	2	1	1	5	2015-05-15 22:56:13	20
42	2	1	1	5	2015-05-15 22:58:13	20
44	2	1	1	5	2015-05-15 23:00:13	20
48	2	1	1	5	2015-05-15 23:02:13	20
50	2	1	1	5	2015-05-15 23:04:13	20

Obr. 7.11: Výpis naměřených hodnot pro snímač 5

7.4.4 Odkaz ERASE VALUES

Po kliknutí na odkaz ERASE VALUES se do frame **body** vypíše PHP obsah stránky `erase_db.php`. Tento příkaz slouží k mazání všech záznamů z tabulky `temperature` v databázi `temp`. Samotné mazání je realizováno `query` příkazem, který je volán po připojení do databáze. Připojení do databáze jsem popsal v kapitole 7.4.2, proto zde uvedu jen samotný příkaz pro mazání záznamů.

```
$sql = "truncate temperature";
```

Po vykonání PHP skriptu se zobrazí v části **body** webové stránky nápis `Database cleared`.

8 ZÁVĚR

Úkolem diplomové práce bylo vymyslet koncepci systému pro záznam měření fyzikálních parametrů ve stavebních konstrukcích. Byla navržena hierarchie systému, jeho struktura, komunikační protokol mezi moduly systému, distribuce napájení celým systémem, hardwarové řešení celého systému v podobě desek plošných spojů a všechny prvky byly vybaveny softwarovým vybavením.

Byl kladen důraz na všestrannost a variabilitu systému. Při návrhu systému jsem však zjistil, že je potřeba udělat několik ústupků v podobě pevně daných pravidel a s tím přicházejících omezení. Mezi tyto ústupky patří použití pouze dvou sběrnic na které lze připojovat snímače pro měření fyzikálních veličin. Těmito sběrnici jsou I2C a 1-wire. Za cenu vyšší vytíženosti sběrnice jsem se také rozhodl pro vysokou frekvenci vyčítání dat z data-koncentrátorů. Tím je dosaženo aktuálnosti dat v databázi a také jsou sníženy nároky na paměť data-koncentrátorů.

Navrhl a zrealizoval jsem tedy kompletní, komplexní systém pro měření fyzikálních veličin. Jako jádro systému je implementována síť decentralizovaných modulů s jedním hlavním prvkem pro sběr a vyčítání dat. Hlavním prvkem je malý počítač Raspberry Pi, který zprostředkovává interakci s uživatelem systému v podobě webového rozhraní. Na Raspberry běží několik služeb umožňujících funkci sběru a interpretace naměřených dat celého systému. Tyto služby jsou Apache2 server, PHP5 server a MySQL databáze, které se starají o webové rozhraní a prostřednictvím technologie ethernet dovolují uživateli jednoduché připojení a přístup k celému měřicímu systému včetně jeho administrace. Hardwarové řešení jsem se snažil co nejvíce automatizovat a zjednodušit, aby mohl systém využívat i laický uživatel.

Troufám si říci, že jsem tohoto cíle dosáhl. Systém po prvotním překonání nastavování adres snímačů a data-koncentrátorů není problém ovládat pouze díky webovému rozhraní. Wizard, nástroj na nastavení, provede i technicky málo zdatného uživatele nastavením systému krok po kroku. Poté už je vše pouze otázkou jednoho kliknutí pro zobrazení aktuálních hodnot měření.

Sběrnice RS-485, které je v systému využíváno, je určena pro zapojení zařízení na jedné větvi. Aby jsem zabránil tomuto omezení, implementoval jsem tři takové větve. Díky tomu je možné mít data-koncentrátory zapojeny na sběrnici v topologii "do hvězdy". Na každou z větví je možné připojit až 32 data-koncentrátorů, což dává celkový počet 96 zařízení pro měření, přičemž na každém mohou být zapojeny desítky až stovky snímačů. V mém demonstračním řešení jsem omezen adresací snímačů LM75A a na jeden data-koncentrátor připadá pouze 8 snímačů.

Pro demonstraci jsem navrhl a realizoval DPS pro snímače teploty LM75A, který s data-koncentrátorem komunikuje na sběrnici I2C. Jeho adresa jde nastavit v rozsahu tří bitů, takže na jeden data-koncentrátor je možné připojit maximálně 8 těchto

snímačů. I v případě tohoto omezení adresace snímačů je možné provozovat systém s 768-mi snímači, čímž jsem splnil zadání. Požadavkem bylo připojit řádově až stovky snímačů.

Zobrazování a práce s daty je intuitivní a velmi jednoduchá. Protože je systém navržen pro použití velkého množství data-koncentrátorů a snímačů, implementoval jsem formulář pro výběr zobrazovaných dat. V tomto formuláři může uživatel zvolit sběrnici, adresu data-koncentrátoru a adresu snímače pro který chce zobrazit naměřené hodnoty. Tyto tři volby je samozřejmě možné i kombinovat.

Systém jsem testoval pro různá nastavení periody čtení hodnot ze snímačů, zapojení snímačů samostatně i společně na stejnou větev RS-485, kombinoval jsem snímače na data-koncentrátorech. Při správném nastavení podle pokynů nástroje Wizard byl systém vždy funkční a nebyl problém naměřené hodnoty z libovolného snímače okamžitě vyhledat.

U návrhu webového rozhraní jsem se snažil o co největší čistotu provedení a přehlednost. Použil jsem neutrální barvy, ale přesto jsem se snažil, aby nebyla grafika příliš strohá, jak je tomu například u webových průvodců pro nastavení routerů a podobných zařízení.

LITERATURA

- [1] *Oficiální webové stránky pro podporu Raspberry Pi* [online]. Dostupné z URL: <<https://www.raspberrypi.org/>>.
- [2] *Atmel, 8-bit Atmel with 8KBytes In-System Programmable Flash ATmega8 ATmega8L* [online]. 2013, poslední aktualizace 2.2013 [cit. 15.5.2015]. Dostupné z URL: <http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_l_datasheet.pdf>.
- [3] *Maxim Integrated, DS2482-100 Single-Channel 1-Wire Master* [online]. 2015, poslední aktualizace 1.2015 [cit. 15.5.2015]. Dostupné z URL: <<http://datasheets.maximintegrated.com/en/ds/DS2482-100.pdf>>.
- [4] *Texas Instruments, SN75176A Differential Bus Transceiver* [online]. 1984, poslední aktualizace 1.2015 [cit. 15.5.2015]. Dostupné z URL: <<http://www.ti.com/lit/ds/symlink/sn75176a.pdf>>.
- [5] *Texas Instruments, LM75A Digital Temperature Sensor and Thermal Watchdog With Two-Wire Interface* [online]. 2000, poslední aktualizace 12.2014 [cit. 15.5.2015]. Dostupné z URL: <<http://www.ti.com/lit/ds/symlink/lm75a.pdf>>.
- [6] *Wikipedie: RS-485* [online]. 2013, poslední aktualizace 5.5.2013 [cit. 15.5.2015]. Dostupné z URL: <<http://cs.wikipedia.org/wiki/RS-485>>.
- [7] *ST, L78xx L78xxC Positive voltage regulators* [online]. 2004, poslední aktualizace 18.3.2008 [cit. 15.5.2015]. Dostupné z URL: <<http://www.farnell.com/datasheets/36610.pdf>>.
- [8] *Raspberry Pi Spy, Simple Guide to the RPi GPIO Header and Pins* [online]. 2015, [cit. 15.5.2015]. Dostupné z URL: <<http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>>.
- [9] *Wiring Pi: GPIO Interface library for the Raspberry Pi* [online]. 2015, [cit. 15.5.2015]. Dostupné z URL: <<http://wiringpi.com/>>.
- [10] *GNU: GNU Make* [online]. 2014, poslední aktualizace 5.10.2014 [cit. 15.5.2015]. Dostupné z URL: <<http://www.gnu.org/software/make/>>.
- [11] *avrdude(1) - Linux man page* [online]. 2013, poslední aktualizace 4.2014 [cit. 15.5.2015]. Dostupné z URL: <<http://linux.die.net/man/1/avrdude>>.

- [12] *avr-gcc(1) - Linux man page* [online]. 1988, poslední aktualizace 2010 [cit. 15. 5. 2015]. Dostupné z URL: <<http://linux.die.net/man/1/avr-gcc>>.
- [13] *w3school.com THE WORLD'S LARGEST WEB DEVELOPER SITE* [online]. 1999, poslední aktualizace 2015 [cit. 15. 5. 2015]. Dostupné z URL: <<http://www.w3schools.com/>>.
- [14] *Ing. Renata Straková, Ing. Josef Knob, Měření součinitele prostupu tepla stávajících stavebních konstrukcí* [online]. 19. 12. 2011, [cit. 15. 5. 2015]. Dostupné z URL: <<http://stavba.tzb-info.cz/prostup-tepla-stavebni-konstrukci/8125-mereni-soucinitele-prostupu-tepla-stavajicich-stavebnich-konstrukci>>.
- [15] *POŽÁRNÍ ZKUŠEBNA 1026* [online]. 2004, [cit. 15. 5. 2015]. Dostupné z URL: <http://www.pavus.cz/index.php?lang=cz&sec=adv_redaction&r_cat=1&s_cat=34&title_string=Po%C5%BE%C3%A1rn%C3%AD+zku%C5%A1ebna+1026>.
- [16] *Technologie chlazení betonu v TBG METROSTAV* [online]. 7. 2013, [cit. 15. 5. 2015]. Dostupné z URL: <http://www.tbg-metrostav.cz/fileadmin/user_upload/napsali_o_nas/clanky_ke_stazeni/076_077_MAT07.pdf>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

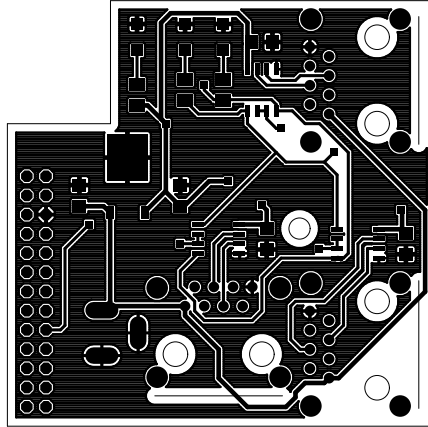
I2C	Inter-Integrated Circuit
TWI	Dvou vodičová sběrnice – Two wire interface
LAMP	Linux, Apache, MySQL, PHP
ASCII	Americký standardní kód pro výměnu informací – American standard code for information interchange
UART	Asynchronní sériové rozhraní – Universal asynchronous receiver and transmitter
WWW	Celosvětově rozšířená síť – World wide web
DPS	Deska plošných spojů – Printed circuit board (PCB)
GPIO	Programovatelné vývody zařízení – General Purpose Input/Output
GND	Uzemnění (referenční potenciál) – Ground
LED	Dioda emitující světlo – Light-emitting diode
SMD	Součástka pro povrchovou montáž – Surface mount device
CSI	Sériové rozhraní pro kameru – Camera serial interface
CRC	Cyklický redundantní součet – Cyclic redundancy check
HTML	Značkovací jazyk pro tvorbu webových stránek – HyperText Markup Language
HTTP	Hypertextový protokol pro výměnu HTML dokumentů – HyperText Transfer Protocol
PHP	PHP: Hypertextový Preprocesor – PHP: Hypertext Preprocessor

SEZNAM PŘÍLOH

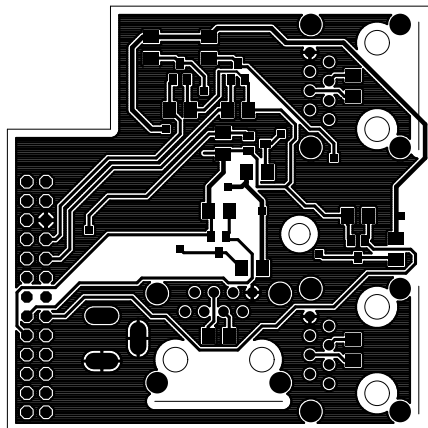
A	DPS Raspberry Pi	70
A.1	DPS top Raspberry Pi	70
A.2	DPS bottom Raspberry Pi	70
A.3	Osazovací výkres top Raspberry Pi	71
A.4	Osazovací výkres bottom Raspberry Pi	71
A.5	Schéma zapojení Raspberry Pi	72
B	DPS koncentrátor dat	73
B.1	DPS top koncentrátor dat	73
B.2	DPS bottom koncentrátor dat	73
B.3	Osazovací výkres top koncentrátor dat	74
B.4	Osazovací výkres bottom koncentrátor dat	74
B.5	Schéma zapojení koncentrátor dat	75
C	Obsah přiloženého DVD	76

A DPS RASPBERRY PI

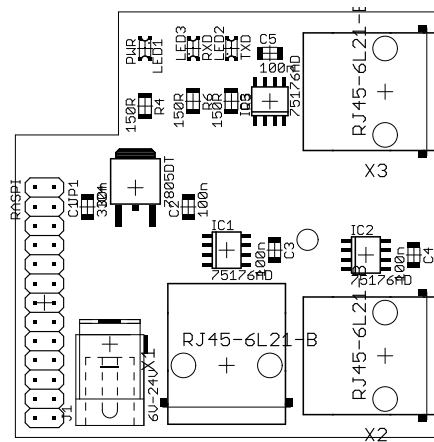
A.1 DPS top Raspberry Pi



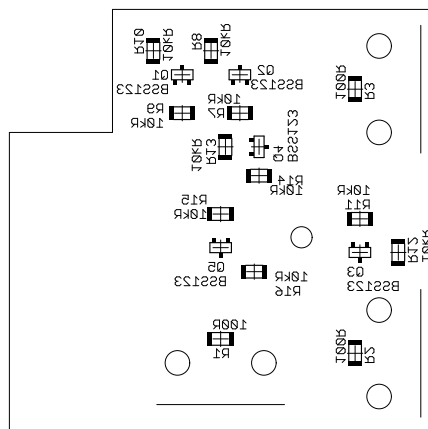
A.2 DPS bottom Raspberry Pi



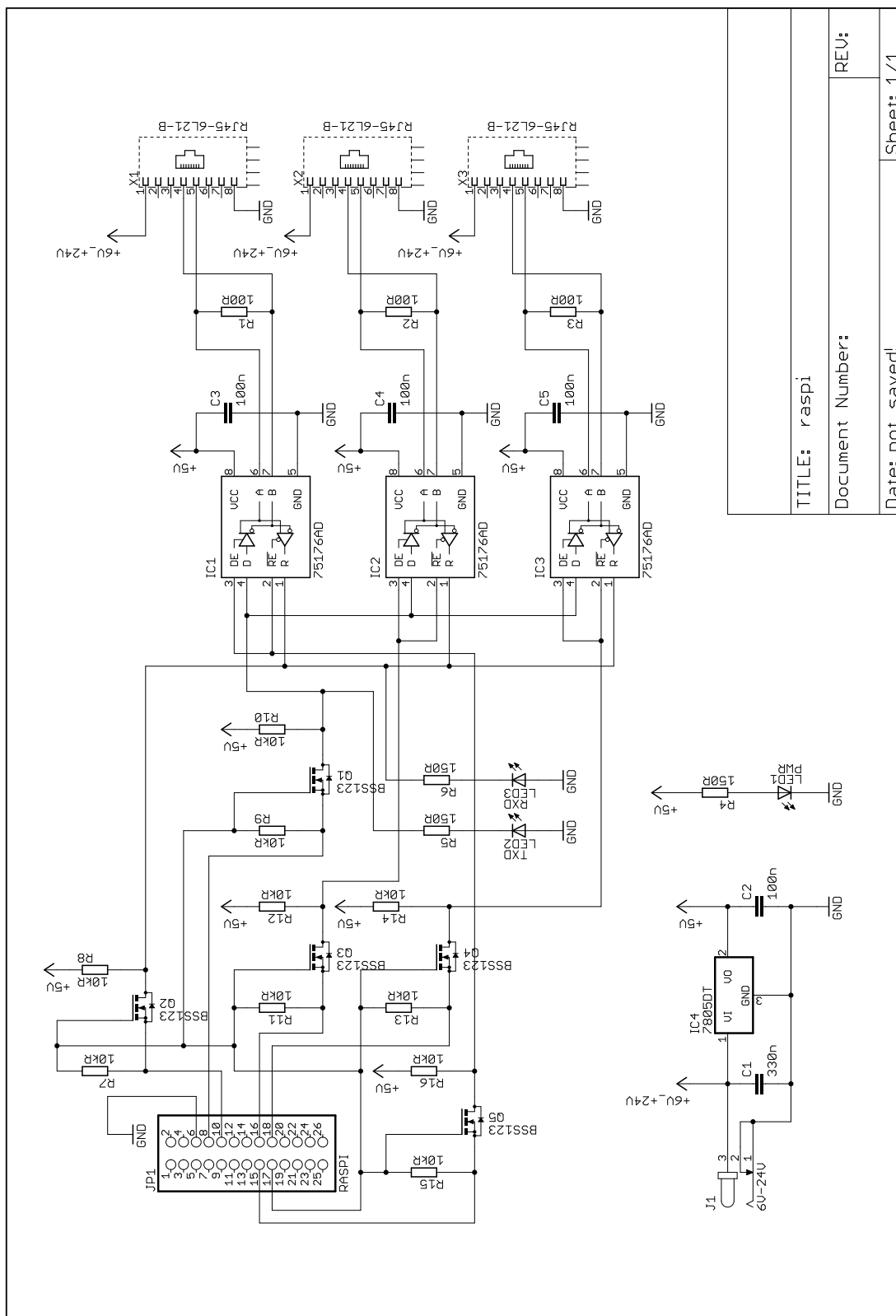
A.3 Osazovací výkres top Raspberry Pi



A.4 Osazovací výkres bottom Raspberry Pi

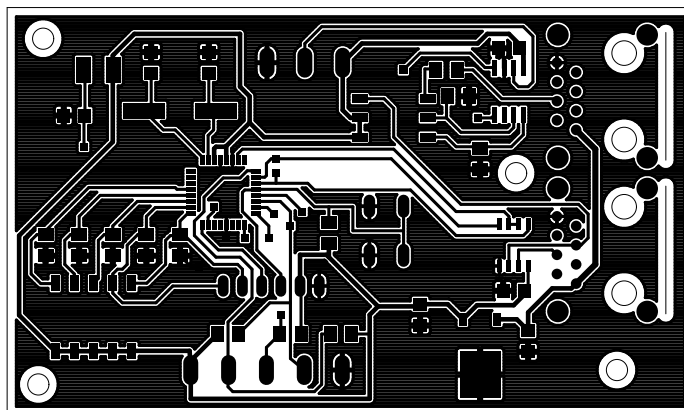


A.5 Schéma zapojení Raspberry Pi

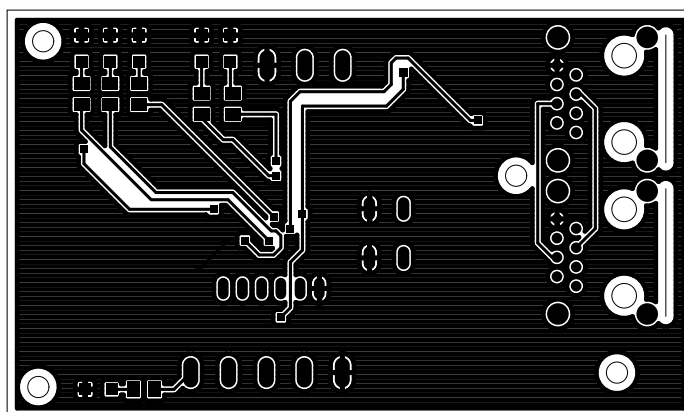


B DPS KONCENTRÁTOR DAT

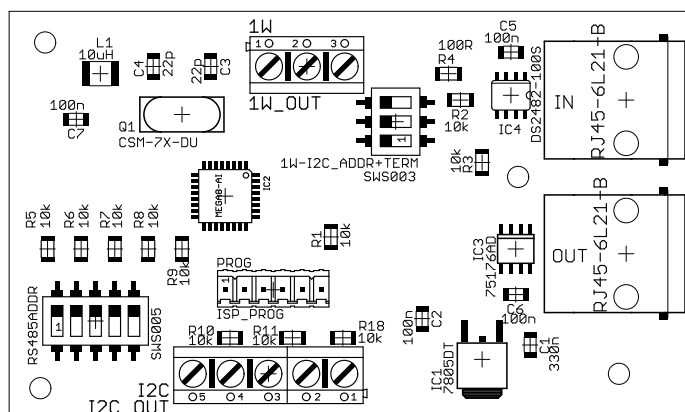
B.1 DPS top koncentrátor dat



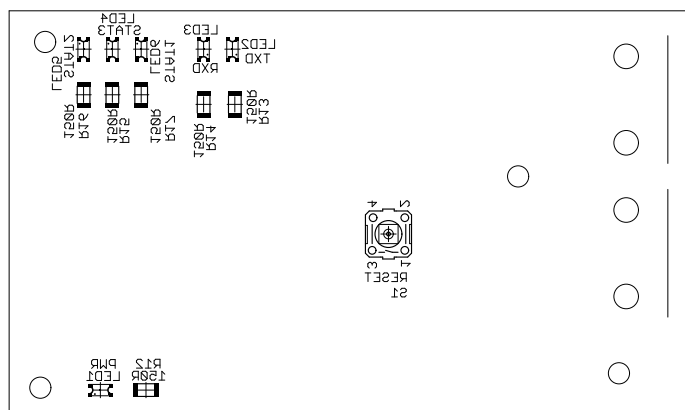
B.2 DPS bottom koncentrátor dat



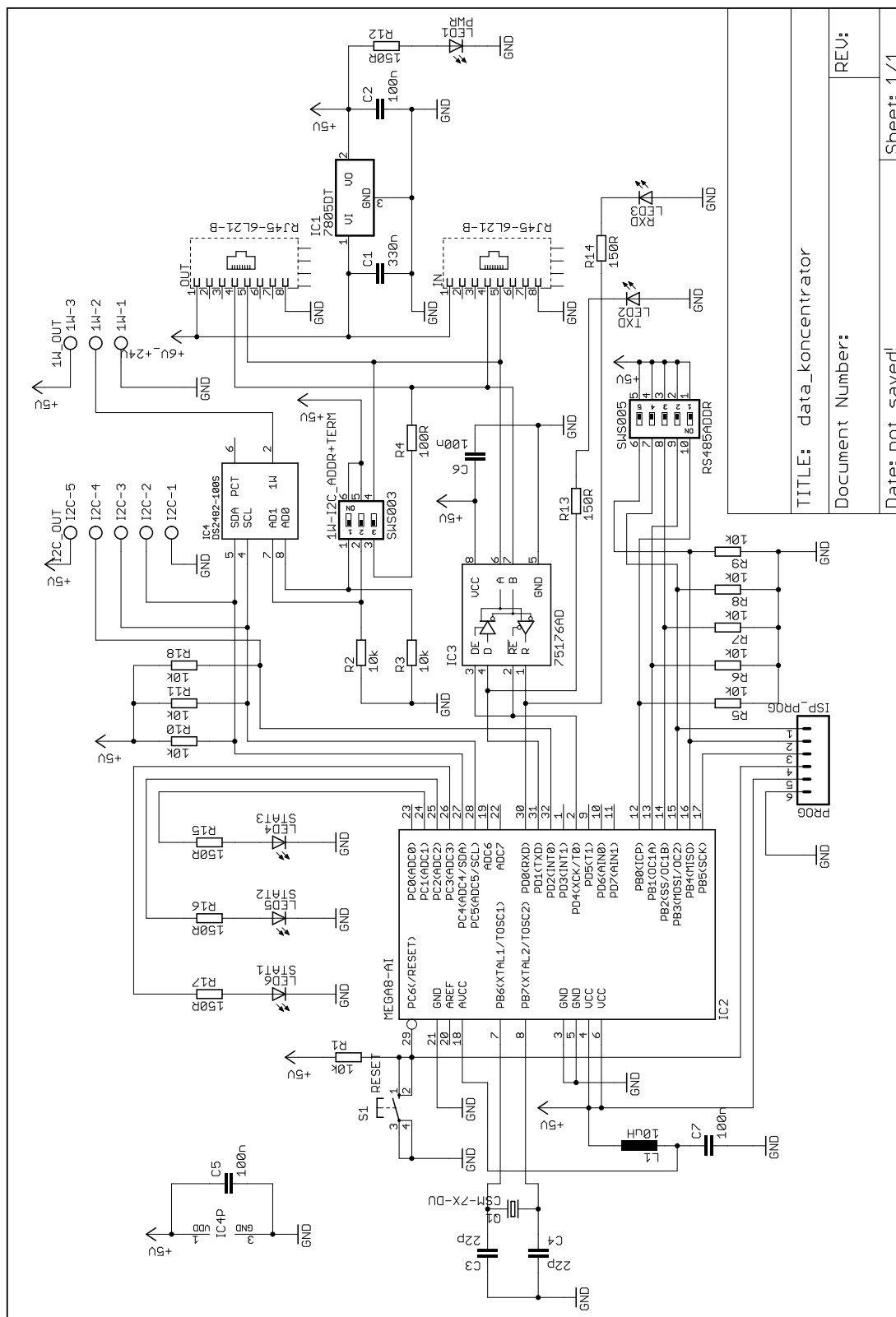
B.3 Osazovací výkres top koncentrátor dat



B.4 Osazovací výkres bottom koncentrátor dat



B.5 Schéma zapojení koncentrátor dat



TITLE: data_koncentrator

Document Number:

REV:

Date: not saved!

Sheet: 1/1

C OBSAH PŘILOŽENÉHO DVD

Obsahem DVD jsou všechny vytvořené zdrojové kódy jak pro Raspberry Pi, tak pro data-koncentrátor.

Složka `raspi` obsahuje soubory z raspberry, jak v jazyce C, tak i celé webové stránky.

Složka `koncentrator` obsahuje softwarové vybavení data-koncentrátoru.

Součástí DVD je také elektronická podoba této dokumentace.