

Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta

Katedra informatiky

Bakalářská práce

Raspberry PI – soubor laboratorních úloh

7. července 2017

Vypracoval: Stanislav Hupka

Vedoucí práce: Ing. Milan Šerý, Ph.D.

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

Jméno a příjmení: **Stanislav HUPKA**

Osobní číslo: **P13807**

Studijní program: **B7507 Specializace v pedagogice**

Studijní obor: **Informační technologie a e-learning**

Název tématu: **Raspberry PI - soubor laboratorních úloh**

Zadávací katedra: **Katedra informatiky**

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

1. Teoretický úvod

- popis počítačů mini PC
- porovnání dostupných modelů
- vybavení Raspberry PI
- OS pro Raspberry PI
- obsluha periférií

2. Provést rešerši podobných řešení a porovnat s nimi svůj koncept

3. Návrh souboru úloh (12)

4. Návrh a tvorba přídavných HW modulů

5. Praktické úlohy

- instalace OS na Raspberry PI (1)
- konfigurace OS (2)
- programová obsluha periférií (3)
- možnosti zpracování a vizualizace dat (4)
- obsluha GPIO vstupy (5)
- obsluha GPIO výstupy (6)
- obsluha I2C (7)
- obsluha SPI (8)
- obsluha RS232 (9)
- obsluha ETHERNET (10)
- Raspberry PI jako web server (11)
- připojení modulu ARDUINO (12)

6. Vyhodnocení a závěr

Rozsah grafických prací: CD ROM

Rozsah pracovní zprávy: 40

Forma zpracování bakalářské práce: tištěná

Seznam odborné literatury:

1. MATOUŠEK, David. Práce s mikrokontroléry ATMEL. 2. vyd. Praha: BEN - technická literatura, 2006. ?C & praxe. ISBN 80-7300-209-4.
2. VÁŇA, Vladimír. Mikrokontroléry ATMEL AVR: programování v jazyce Bascom. 1. vyd. Praha: BEN - technická literatura, 2004. ISBN 80-7300-115-2.
3. CHMELA, František. Automatizace a robotizace I: (Automatizace). 1. vyd. Praha: České vysoké učení technické, 1992.
4. Sedláček M.: Zpracování signálů. ČVUT Praha 1995
5. KOCOUREK, Petr. Přenos informace. Vyd. 2. Praha: České vysoké učení technické, 1999. ISBN 80-01-01978-0.
6. HAASZ, Vladimír, Jiří NOVÁK a Jaroslav ROZTOČIL. Číslicové měřicí systémy. Vyd. 2. přeprac. Praha: České vysoké učení technické, 2000. ISBN 80-01-02219-6.
7. NORRIS, Donald. Raspberry Pi: projekty. 1. vyd. Brno: Computer Press, 2015, 264 s. ISBN 978-80-251-4346-9.
8. VODA, Zbyšek. Průvodce světem Arduina. Vydání první. Bučovice: Martin Stríž, 2015, 239 stran. ISBN 978-80-87106-90-7

Vedoucí bakalářské práce:

Ing. Michal Šerý, Ph.D.

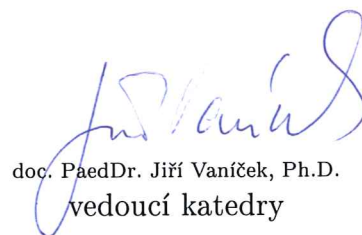
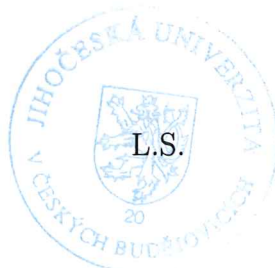
Katedra aplikované fyziky a techniky

Datum zadání bakalářské práce: 21. dubna 2016

Termín odevzdání bakalářské práce: 28. dubna 2017



Mgr. Michal Vančura, Ph.D.
děkan



doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 21. dubna 2016

Prohlášení

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně, pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 7. července 2017

Stanislav Hupka

Abstrakt

Cílem této práce je vytvořit soubor úloh pro praktickou demonstraci projektů z oboru automatizace, jejich zpracování a vizualizaci pomocí počítače Raspberry Pi. Zvolený problém jsem po porovnání dostupných modelů počítačů koncepce „mini PC“ realizoval pomocí Raspberry Pi 3 Model B. Vytvořil jsem celkem dvanáct úloh, které řeší měření fyzikálních veličin a jejich následné zpracování a vizualizaci. K tomuto účelu byl vytvořen modulární systém, kde pomocí jednoduché konfigurace bude možno úlohy bezpečně realizovat bez rizika poškození elektronických obvodů a hlubší znalosti návrhu elektronických obvodů.

Vytvořené řešení umožňuje studentům vyzkoušet si teoretické znalosti z oborů elektronických měření, automatizace a programování. Propojením jednotlivých úloh dojde k vytvoření funkčního celku, od vytěžení dat z čidel, přes SW zpracování, až po grafický výstup pomocí webového rozhraní.

Klíčová slova

Raspberry Pi, automatizace, měření, programování, vizualizace

Abstract

The goal of this work is to create a set of tasks for practical demonstration projects in the field of automation, data processing and visualization using computer Raspberry Pi. The chosen problem after comparing the available computer models "mini PC" concept I implemented using Raspberry Pi 3 Model B. I created twelve tasks that deal the measurement of physical quantities and their subsequent processing and visualization. For this purpose was created a modular system where, using a simple configuration tasks will be implemented safely without risk of damage to electronic components and deeper knowledge of the design of electronic circuits.

Made solutions allows students to test their theoretical knowledge in the fields of electronic measurement, automation and programming. Integrating individual jobs will be created functional units, from extracting data from sensors, processing software, to the graphical output using a web interface.

Keywords

Raspberry Pi, automation, measurement, programming, visualization

Poděkování

Rád bych poděkoval Ing. Milanu Šerému, Ph.D., za odborné vedení, cenné rady, věcné připomínky a vstřícnost při konzultacích mé bakalářské práce.

Obsah

1	Úvod	14
2	Teoretický úvod	15
2.1	Popis počítačů „mini PC“	16
2.2	Porovnání dostupných modelů	17
2.3	Vybavení Raspberry Pi	23
2.3.1	BCM2837[5, 6]	23
2.3.2	Napájení[7]	24
2.3.3	USB[8]	25
2.3.4	GPIO[11]	25
2.4	OS pro Raspberry Pi	26
2.4.1	Raspbian	27
2.4.2	Ubuntu MATE	27
2.4.3	Snappy Ubuntu Core	27
2.4.4	Windows 10 IoT Core	28
2.4.5	OSMC	28
2.4.6	LibreELEC	28
2.4.7	PiNet	29
2.4.8	RISC OS	30
2.4.9	Weather Station	30
2.4.10	Výběr vhodného OS	30
2.5	Obsluha periférií	31
3	Rešerše podobných projektů	32
4	Návrh souboru vzorových úloh	33
4.1	Instalace OS a základní konfigurace Raspberry Pi	33
4.2	Konfigurace systému pomocí „raspi-config“[19]	33
4.2.1	Expand Filesystem	34

4.2.2	Change User Password	34
4.2.3	Boot Options	34
4.2.4	Internationalisation Options	34
4.2.5	Enable Camera	35
4.2.6	Overclock	35
4.2.7	Advanced Options	35
4.3	Obsluha ETHERNET	36
4.4	Základní využití GPIO - vstup	37
4.5	Základní využití GPIO – výstup	38
4.6	Programová obsluha periférií	39
4.7	Využití I2C sběrnice	41
4.7.1	Obsluha I2C pomocí Raspberry Pi	42
4.7.2	Ovládání digitálních vstupů a výstupů pomocí sběrnice I2C	42
4.7.3	Využití I2C sběrnice pro měření elektrických veličin (AD převodník)	44
4.7.4	Využití I2C sběrnice pro měření neelektrických veličin	47
4.8	Obsluha SPI	50
4.8.1	Obsluha SPI pomocí Raspberry Pi[34]	51
4.8.2	Připojení TFT displeje pomocí SPI	52
4.9	Obsluha RS232	52
4.10	Připojení modulu ARDUINO	54
4.11	Raspberry Pi jako web serveru	55
4.12	Možnosti zpracování a vizualizace dat	55
4.12.1	Zpracování dat pro následnou vizualizaci	55
4.12.2	Vizualizace dat pomocí webového rozhraní	56
4.12.3	Vizualizace dat pomocí displeje HW modulu	57
5	Návrh a tvorba přídatných HW modulů	59
5.1	Návrh zapojení HW modulu	59
5.2	Návrh desky plošných spojů HW modulu	61

6	Praktické řešení vzorových úloh	64
6.1	Instalace OS a základní konfigurace Raspberry Pi	65
6.1.1	Instalace pomocí NOOBS[43]	65
6.1.2	Instalace obrazu OS pomocí MS Windows[44]	66
6.1.3	Instalace obrazu OS v GNU/Linux[45]	67
6.1.4	První spuštění	68
6.2	Konfigurace systému pomocí „raspi-config“	69
6.2.1	System	70
6.2.2	Interfaces	70
6.2.3	Performance	71
6.2.4	Localisation	71
6.3	Obsluha ETHERNET	72
6.3.1	Nastavení statické IP adresy	72
6.3.2	Připojení pomocí SSH	73
6.3.3	Nakopírování souborů do Raspberry Pi	74
6.4	Základní využití GPIO - vstup	75
6.5	Základní využití GPIO – výstup	76
6.6	Programová obsluha periférií	77
6.7	Využití I2C sběrnice	78
6.7.1	Ovládání digitálních vstupů a výstupů pomocí sběrnice I2C	79
6.7.2	Využití I2C sběrnice pro měření elektrických veličin (AD převodník)	80
6.7.3	Využití sběrnice I2C pro měření neelektrických veličin	80
6.8	Obsluha SPI	81
6.9	Obsluha RS232	82
6.9.1	Komunikace pomocí terminálu v GNU/Linux	83
6.9.2	Výstup hodnot z programu na sériový port	84
6.10	Připojení modulu ARDUINO	85
6.10.1	Nahrání programu do ARDUINA	85
6.10.2	Komunikace ARDUINO - Raspberry Pi	86

6.11 Raspberry Pi jako web server	86
6.12 Možnosti zpracování a vizualizace dat	88
6.12.1 Zpracování dat pro následnou vizualizaci	88
6.12.2 Vizualizace dat pomocí webového rozhraní	88
6.12.3 Vizualizace dat pomocí displeje HW modulu	89
7 Návrh dalších úloh k řešení	91
8 Závěr	92

1 Úvod

Dnes je možné vysledovat trend implementování inteligentních počítačových systémů do běžné elektroniky, která nás obklopuje. Ať už se jedná o autonomní systémy měření a regulace, nebo inteligentní senzory napojené na nadřazený systém, používá se stále výkonnějšího HW vybavení, mnohdy obsahující vlastní OS, umožňující pohodlnou komunikaci s těmito systémy pomocí chytrých telefonů, nebo přes webové rozhraní. Toto téma je dnes velice populární a prochází dynamickým vývojem díky technologiím jako je Internet věcí¹, moderním trendům v průmyslu a jejich implementací v praxi.

Snahou práce je vytvořit učební pomůcku, která bude rozvíjet schopnosti studentů orientovat se komplexně v této problematice. Jako základ systému je využít počítač Raspberry Pi. Porovnáním s dalšími produkty na trhu, i přes to, že nemá nejvyšší výpočetní výkon, byl vybrán jako nejvhodnější kandidát pro naše účely. Vyniká hlavně rozsáhlou komunitou a tím pádem i vysokou funkčností a velice precizním zdokumentováním funkcí.

Hlavní přínosem přípravku bude praktické odzkoušení si znalostí, jak z oblasti HW, tak i SW a propojení těchto znalostí ve funkční řešení. Pro tento účel bude připraveno dvanáct úloh, kde postupným rozvojem dojdeme od základní konfigurace Raspberry Pi, přes získání vstupních dat až po prezentaci dat na webovém rozhraní a grafickém displeji.

Cílem práce je vytvořit modulární systém (učební pomůcku) pro praktické odzkoušení úloh z výpočetní a automatizační techniky. Tento systém bude umožňovat kompletní realizaci projektů od instalace SW, přes zapojení jednotlivých modulů pro měření fyzikálních veličin až po jejich zpracování a výstup. Postupným přidáváním jednotlivých částí systému bude posouvat znalosti studentů od dílčích znalostí ke komplexnímu zvládnutí problému zpracování a vizualizace dat.

¹Internet věcí (anglicky Internet of Things, zkratka IoT) je v informatice označení pro propojení vestavěných zařízení s Internetem. Propojení zařízení by mělo být zejména bezdrátové a mělo by přinést nové možnosti vzájemné interakce nejen mezi jednotlivými systémy a též přinést nové možnosti jejich ovládání, sledování a zajištění pokročilých služeb.[1]

2 Teoretický úvod

Pokud chceme popsat počítače s koncepcí „mini PC“, musíme si nejdříve ujasnit, co si pod tímto názvem v našem pojetí představujeme a co vedlo ke vzniku těchto zařízení. Budeme-li považovat za počátek éry „mini PC“ vznik Raspberry Pi (dále jen RPi) [2], pak můžeme považovat za zrod myšlenky malého a levného počítače rok 2006, kdy Eben Upton a jeho kolegové z University of Cambridge zaznamenali znepokojivý fenomén. Zaznamenali značný meziroční pokles úrovně dovedností studentů v oblasti informačních technologií a aplikace počítačů pro vědecké účely.

Identifikovali problém ve změně přístupu k výuce informačních technologií. Starší generace studentů se učila programovat na jednoduchých počítačích typu ZX Spectrum, Commodore64 aj. Díky nedostupnosti komerčních PC, byly tyto levné počítače jedinou možností jak proniknout do světa počítačů. Z dnešního pohledu můžeme tyto počítače přirovnat ke koncepci RPi. Oproti nim dnešní generace, díky rozmachu domácích PC, berou PC jen, jako „nástroj“ a tomu se přizpůsobila i výuka. Hlavní náplní počítačové výuky se stalo programování HTML, práce ve Wordu a Excelu.

Uvědomili si, že řešením by mohl být vývoj jednoduchého finančně dostupného zařízení pro výuku informačních technologií stejně, jako tomu bylo na počátku devadesátých let u počítačů typu ZX Spectrum, Commodore64 aj. Mezi roky 2006 až 2008 Eben Upton navrhl několik verzí toho, co se později stalo základem pro RPi. V této době díky rozmachu mobilních telefonů se stávají procesory určené pro mobilní zařízení dostatečně výkonné a zároveň cenově dostupné. To vedlo ke vzniku Raspberry Pi Foundation, kde Eben Upton, Rob Mullins, Jack Lang a Alan Mycrofta začali vyvíjet RPi, jak ho známe dnes.

RPi[3] je jednodeskový počítač o velikosti kreditní karty, tj. 85×56 mm, osazený procesorem ARM. Původně byl navržen Raspberry Pi Foundation jako finančně dostupný prostředek pro výuku informatiky. Zájem o něj však předčil očekávání jeho tvůrců. První série byla vyprodána již během jedné hodiny po uvedení do prodeje. Na další série se čekací doba pro nové objednávky protáhla i na několik měsíců.

RPi umožňuje spouštět grafické aplikace jako webový prohlížeč, tabulkový procesor, textový editor, jednodušší hry, přehrávat video ve vysokém rozlišení. OS, programy a data jsou umístěny na SD kartě. Konektivita RPi obsahuje HDMI rozhraní, audio jack 3,5 mm, USB porty, připojení LAN (RJ45), WLAN, Bluetooth, rozhraní CSI pro připojení kamery, rozhraní DSI pro připojení displeje, 40 pinový konektor GPIO obsahující digitální I/O, I2C sběrnici, SPI sběrnici, UART, napájení +3,3 V a +5 V.

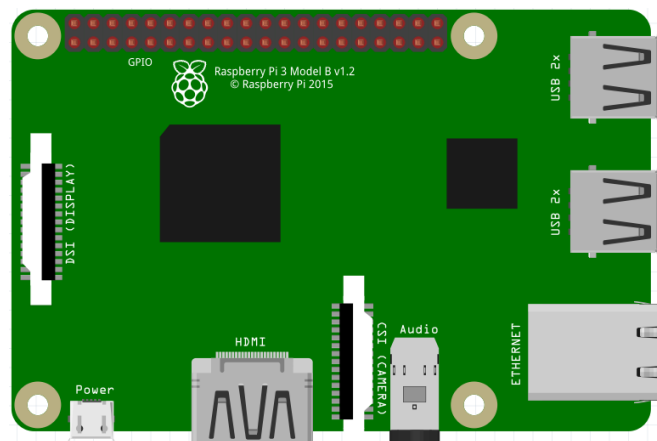
2.1 Popis počítačů „mini PC“

Porovnáme-li produkty rodiny počítačů „mini PC“, dojdeme k určitým charakteristikám, které jsou shodné nebo přinejmenším podobné pro většinu produktů od vzniku prvního RPi. Tuto koncepci převzali od Raspberry Pi Foundation i ostatní výrobci obdobných zařízení a vznikl tak jakýsi standart pro tyto počítače. Počítače „mini PC“ můžeme tedy charakterizovat těmito vlastnostmi:

- Rozměry - velikost modulu je přibližně shodná s velikostí kreditní karty (cca 85x55 mm).
- Napájení - řešeno externím napájecím zdrojem s jediným napětím +5 V připojeným konektorem micro USB nebo obdobným.
- GPIO header - GPIO je vyvedeno prostřednictvím 40 pinového konektoru. Obsahuje obvykle kromě digitální I/O, I2C sběrnici, SPI sběrnici, UART, napájení +3,3 V a +5 V.
- USB - jeden až čtyři USB porty pro připojení polohovacích zařízení, úložných zařízení atd.
- RJ45 - Ethernet port pro připojení do sítě.
- HDMI - konektor pro připojení displeje.
- DSI - „display interface“ pro připojení LCD displeje.
- CSI - „camera interface“ pro připojení CMOS senzoru kamery.
- Audio Jack 3.5 mm - pro připojení sluchátek nebo jiného analogového zvukového výstupního zařízení.
- Slot na paměťovou kartu s nainstalovaným OS.

Kromě těchto charakteristických znaků se modely mírně liší v rozložení konektorů osazením indikačních LED, nebo vybavením dalším příslušenstvím. Dále pak vznikají zařízení vymykající se této charakteristice určené pro speciální účely. Ale i přes to většina zařízení momentálně dostupných na trhu lze charakterizovat výše uvedenými vlastnostmi.

Na obrázku 1 vidíme typického zástupce „mini PC“ Raspberry Pi 3 model B, kde můžeme najít všechny výše uvedené charakteristické znaky.



Obrázek 1: Raspberry Pi 3 rozložení konektorů

2.2 Porovnání dostupných modelů

V současné době je díky popularitě projektu Raspberry Pi na trhu několik dalších výrobců, kteří vyrábějí podobná zařízení. Tito výrobci víceméně kopírují koncept Raspberry Pi, s tím, že vytvářejí další klony se specifickými vlastnostmi. Vznikají tak např. zařízení určené pro práci s multimédií, které mají vyšší výpočetní výkon, obsahují SATA porty, rychlejší síťová rozhraní. Kromě zařízení Raspberry Pi porovnáme parametry počítačů Banana Pi² a Orange Pi³.

Od každého výrobce porovnáme nejvýkonnější model, ekonomický model a variantu „mini“ určenou pro zástavbu do produkčních řešení. V tabulkách⁴ 1, 2 a 3 jsou porovnány jednotlivé typy „mini PC“ dle těchto parametrů⁵:

- CPU - typ osazeného procesory.
- GPU- typ osazeného grafického akcelérátoru.
- RAM - velikost, popřípadě typ operační paměti.
- Uložiště - datové médium, které můžeme použít přímo, bez nutnosti připojení přes USB port.

²<http://www.banana-pi.org/index.html>

³<http://www.orangepi.org/>

⁴Pro lepší orientaci jsou v tabulkách barevně odlišeny modely, které by měly být porovnávány spolu.

⁵Veškeré technické parametry jsou převzaty z prezentovaných dat jednotlivých modelů na stránkách výrobců.

- LAN - metalické síťové připojení Ethernet, rychlost.
- WiFi/BT - bezdrátové připojení, typ a rychlost.
- USB Porty - typ, rychlost a počet USB portů.
- OS - výrobcem oficiálně podporované OS, dostupné na stránce výrobce.
- Cena - průměrná cena v Kč, vypočtená průměrem nejméně pěti hodnot nalezených na internetu. U některých výrobků byla cena uvedena pouze v USD, v tom případě byla přepočtena s kurzem 25,00 Kč/1 USD.

Porovnávají jsou pouze parametry, ve kterých se modely odlišují. Jelikož vycházejí modely ze stejné koncepce, shodují se v parametrech jako např:

- Napájení
- HDMI výstup
- CSI
- DSI
- GPIO header

Porovnáním parametrů udávaných výrobcí bylo vypočteno bodové hodnocení v rozsahu $0 \div 1$ (vyšší číslo - lepší hodnocení). Podrobná metodika hodnocení, viz příloha 2. Z výsledků, jak můžeme vidět v tabulce 4, je patrné, že pokud se budeme řídit pouze parametry udávanými výrobcí, tak ve všech kategoriích dopadli nejhůře počítače Raspberry. Díky tomu, že modely Orange a Banana mají lepší HW parametry při podobné ceně, vychází jejich hodnocení lépe.

MODEL	CPU	GPU	RAM
RASPBERRY PI 3 MODEL B	1.2GHz 64-bit quad-core ARMv8 CPU	VideoCore IV 3D graphics core	1GB RAM
RASPBERRY PI 2 MODEL B	900MHz quad-core ARM Cortex-A7 CPU	VideoCore IV 3D graphics core	1GB RAM
RASPBERRY PI ZERO W	1GHz, single-core CPU		512MB RAM
Banana Pi BPI-M64	1.2 Ghz Quad-Core ARM Cortex A53 64-Bit Processor	MALI-400 MP2 @ 500MHz	2GB DDR3 SDRAM
Banana Pi BPI-M2U	quad-core cortex -A7 R40	MALI-400 MP2 @ 500MHz	2GB DDR3 SDRAM
Banana Pi BPI-M2M	Allwinner R16,Quad-code ARM Cortex-A7	Mali 400 MP2	512MB DDR3
Orange Pi Plus2	H3 Quad-core Cortex-A7	Mali400MP2 GPU @600MHz	2GB DDR3
Orange Pi PC 2	Quad-core 64-bit Cortex-A53	Mali450	1GB DDR3
Orange Pi Zero	H2 Quad-core Cortex-A7	Mali400MP2 GPU @600MHz	256MB/512MB DDR3 SDRAM

Tabulka 1: Porovnání parametrů „mini PC“ - část 1

MODEL	Uložiště	LAN	WiFi/BT
RASPBERRY PI 3 MODEL B	Micro SD card	10/100 Mbit / s Ethernet	802.11n/Bluetooth 4.1
RASPBERRY PI 2 MODEL B	Micro SD card	10/100 Mbit / s Ethernet	
RASPBERRY PI ZERO W	Micro SD card		802.11n/Bluetooth 4.1
Banana Pi BPI-M64	MicroSD Card/eMMC(8GB onboard)	10/100/1000 Mbps Ethernet	802.11n/Bluetooth 4.0
Banana Pi BPI-M2U	MicroSD Card/eMMC(8GB onboard)/SATA	10/100/1000 Mbps Ethernet	802.11n/Bluetooth 4.0
Banana Pi BPI-M2M	MicroSD (TF) card, eMMC 8G on board		802.11n/Bluetooth 4.0
Orange Pi Plus2	SD(TF) card/ MMC card slot , SATA, 16GB EMMC Flash	10/100/1000 Mbps Ethernet	802.11n
Orange Pi PC 2	MicroSD (TF) card	10/100/1000 Mbps Ethernet	
Orange Pi Zero	MicroSD (TF) card	10/100 Mbit / s Ethernet	802.11n

Tabulka 2: Porovnání parametrů „mini PC” - část 2

MODEL	USB Porty	OS	Cena
RASPBERRY PI 3 MODEL B	4 x USB	Raspbian ,Pi-dora, LibreELEC, OSMC, RISC OS ,Arch Linux, Windows IoT	1046
RASPBERRY PI 2 MODEL B	4 x USB	Raspbian ,Pi-dora, LibreELEC, OSMC, RISC OS ,Arch Linux	1029
RASPBERRY PI ZERO W	1 x MicroUSB	Raspbian	250
Banana Pi BPI-M64	2x USB 2.0, USB 1x OTG(Micro USB)	Android, Raspbian, Ubuntu, OpenSUSE, Debian, Bananian, Windows IoT	1250
Banana Pi BPI-M2U	3x USB 2.0, USB 1x OTG(Micro USB)	Debian, Ubuntu	1250
Banana Pi BPI-M2M	1x USB 2.0 host, 1x USB 2.0 OTG	Android, Raspbian, Ubuntu, OpenSUSE, Debian, Bananian	900
Orange Pi Plus2	4x USB 2.0HOST, 1x USB 2.0 OTG	Android, Ubuntu, Debian, Raspbian	1250
Orange Pi PC 2	3x USB 2.0HOST, 1x USB 2.0 OTG	Android, Ubuntu, Debian, Raspbian	600
Orange Pi Zero	1x USB 2.0HOST, 1x USB 2.0 OTG	Android, Lubuntu, Debian, Raspbian	275

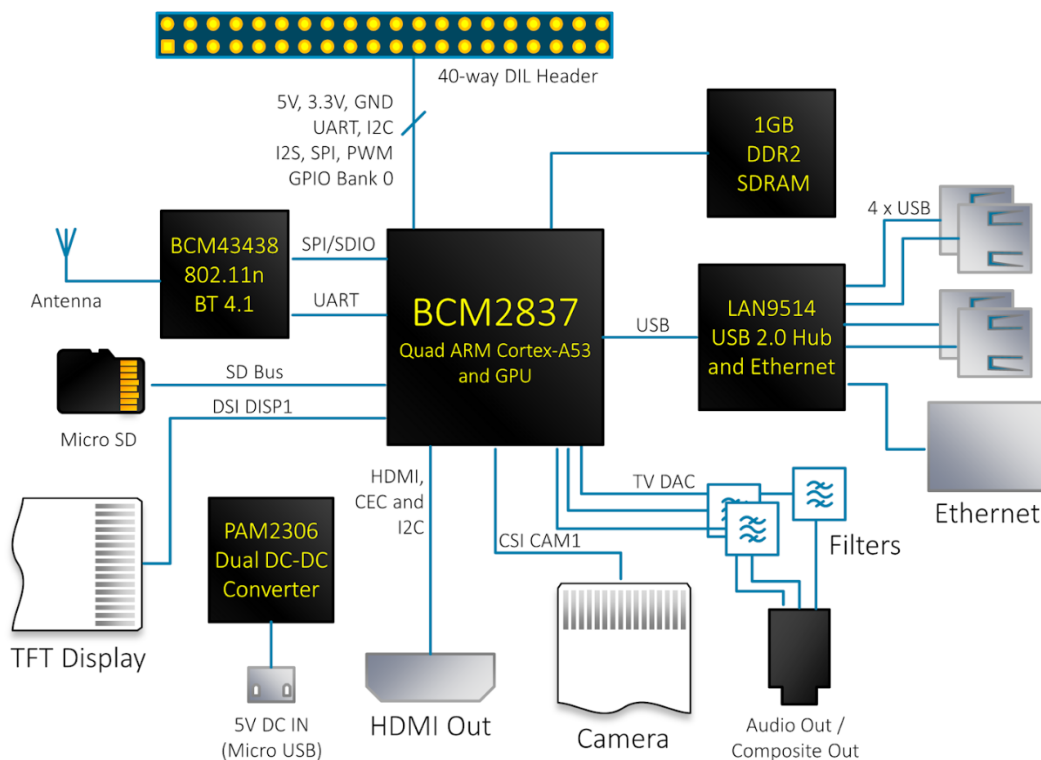
Tabulka 3: Porovnání parametrů „mini PC” - část 3

MODEL	CPU	RAM	Ulo- žiště	LAN	WiFi BT	USB	OS	Cena	Cel- kem
RASPBERRY PI 3 MODEL B	1.00	0.50	0.33	0.67	1.00	0.73	1.00	-0.84	4.39
RASPBERRY PI 2 MODEL B	0.80	0.50	0.33	0.67	0.00	0.89	1.00	-0.82	3.37
RASPBERRY PI ZERO W	0.25	1.00	0.50	0.00	1.00	0.40	0.17	-0.28	3.04
Banana Pi BPI-M64	1.00	1.00	0.67	1.00	1.00	0.82	1.00	-1.00	5.48
Banana Pi BPI-M2U	0.80	1.00	1.00	1.00	1.00	1.00	0.29	-1.00	5.09
Banana Pi BPI-M2M	1.00	1.00	1.00	0.00	1.00	1.00	1.00	-1.00	5.00
Orange Pi Plus2	0.80	1.00	1.00	1.00	0.50	1.00	0.57	-1.00	4.87
Orange Pi PC 2	1.00	0.50	0.33	1.00	0.00	0.56	0.57	-0.48	4.44
Orange Pi Zero	1.00	1.00	0.5	1.00	0.50	1.00	0.67	-0.31	5.97

Tabulka 4: Porovnání parametrů „mini PC” - souhrn

2.3 Vybavení Raspberry PI

V této kapitole si popíšeme hardwarové vybavení RPi, konkrétně pro verzi Raspberry Pi 3 MODEL B, kterou budeme využívat pro realizaci praktických úloh. Na obrázku 2 vidíme všechny důležité obvody RPi.



Obrázek 2: Raspberry Pi 3 Blokový Diagram[4]

2.3.1 BCM2837[5, 6]

Jádrem RPi 3 je čipset BCM2837, díky němuž je možno poprvé osadit RPi 64-bitovým procesorem. Konkrétně čtyř-jádrovým ARM Cortex-A53, taktovaném na 1,2 GHz. Stejně jako u předchozí generace RPi je osazen GPU Broadcom VideoCore IV, taktovaném na 400 MHz s maximálním rozlišením Full HD 1080p. Z obrázku 2 vidíme napojení dalších periférií na BCM2837. Oproti předchozím generacím RPi zde přibyl obvod BCM43438 zajišťující bezdrátovou konektivitu pomocí WiFi a Bluetooth.

Níže jsou uvedeny periferie obsluhované BCM2837:

- 1.2 GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN
- Bluetooth 4.1 (Bluetooth Low Energy)
- 1 GB RAM
- 4 USB ports
- 40 GPIO pins
- Full HDMI port
- Ethernet port
- Combined 3.5 mm audio jack and composite video
- Camera interface (CSI)
- Display interface (DSI)
- Micro SD card slot
- VideoCore IV 3D graphics core

2.3.2 Napájení[7]

RPi je napájen přes konektor micro USB napájecím napětím +5,1 V. Odběr RPi je závislý na počtu připojených periférií. Obecně lze konstatovat, že použití zdroje +5 V 2,5 A by mělo být dostatečné k rozeběhnutí RPi s připojenými perifériemi. Typicky RPi odebírá 700 až 1000 mA v závislosti na připojených perifériích. Maximální celkový proud, který je možno odebírat z RPi je 1 A. Pokud chceme připojit USB zařízení s větším odběrem, musíme použít USB rozbočovač s externím napájením. Odběr přes piny GPIO nesmí překročit celkově 50 mA a 16 mA na jediný pin. Překročení těchto hodnot může vést ke zničení obvodu BCM2837. HDMI port odebírá cca 50 mA a kamerový modul 250mA.

RPi lze taktéž napájet přes GPIO pin +5 V, nebo přes USB, pomocí USB rozbočovače. Ovšem pozor, v tomto případě jsou vyřazeny ochranné obvody napájecího okruhu a může dojít ke zničení RPi. Toto řešení není doporučováno výrobcem jako oficiální způsob napájení.

2.3.3 USB[8]

RPi je vybaven čtyřmi USB 2.0 porty, připojenými přes LAN9541 (kombinovaný USB 2.0 Hub/100 Mbit Ethernet) na jeden USB kanál BCM2837.

USB port RPi má určité odlišnosti od USB zařízení běžného PC. Je to dáno tím, že obvod BCM2837 je primárně navržený pro použití v mobilních zařízeních. Proto má RPi ve skutečnosti pouze jediný USB port, jelikož mobilní zařízení většinou využívají USB pro připojení k PC v režimu host. Tzn. rychlost všech připojených USB periférií, včetně ethernetového připojení se dělí o jeden USB port o maximální rychlosti 480 Mbps. USB port RPi je OTG⁶. Obecně poskytuje přiměřenou úroveň funkčnosti pro většinu zařízení, ale díky tomu, že má jednodušší HW, komunikaci musí z větší části zajišťovat systémový software. Lze říci, že veškeré USB zařízení, které je funkční na PC pod GNU/Linux fungují i na RPi.

2.3.4 GPIO[11]

GPIO(General-purpose input/output)[12] lze obecně popsat jako pin integrovaného (počítačového) obvodu, jehož chování a směřování (zda-li jde o vstupní, nebo výstupní pin) lze ovládat uživatelem za běhu. GPIO nejsou standardně využívány a nemají předem definovaný účel. Předpokládá se, že při vývoji zařízení může nastat požadavek na ovládání dalších obvodů nebo funkcí pomocí čipu. V tomto případě, pokud bychom neměly volné GPIO piny, by muselo dojít k re-designu chipu nebo celého zařízení.

U RPi jsou GPIO vyvedeny přímo z procesoru BCM2837. Obvod BCM2837 obsahuje 3 banky GPIO. Každá banka může mít vlastní odlišné napájení VDD. V případě RPi 3 je použito pro všechny banky napájecí napětí +3,3 V. Pokud dojde k přivedení vyššího napětí na pin GPIO, může dojít k zničení obvodu BCM2837.

Fyzicky máme přístup na konektoru P1 pouze k vybraným pinům banky 0. Některé GPIO piny mají kromě standardního vstupu/výstupu ještě další alternativní funkci.

⁶USB On-The-Go, zkracované na USB OTG nebo jen OTG, je normativní specifikace prvně použitá v roce 2001, která umožňuje zařízení s USB jako digitální zvukový přehrávač nebo mobilní telefon komunikovat jako hostitelské zařízení pro USB přístroje, jako paměť USB flash disk, digitální kamera, počítačová myš nebo klávesnice. Použití USB OTG ale dovoluje přepnout se do opačného uspořádání, do role hostujícího klienta pro jiné hostitelské zařízení, např. počítač, a naopak. Například mobilní telefon může načíst z připojeného přístroje, média jako hostitel nebo se sám hlásit jako paměťové medium po připojení k PC. Obdobně spolu naváží kabelové spojení např. dva mobily.[9]

Osazení konektoru P1 a funkce jednotlivých pinů jsou zřejmé z obrázku 3.

Raspberry Pi 3 GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power	⬛	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	⬛	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	⬛	Ground	06
07	GPIO04 (GPIO_GCLK)	⬛	(TXD0) GPIO14	08
09	Ground	⬛	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	⬛	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	⬛	Ground	14
15	GPIO22 (GPIO_GEN3)	⬛	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	⬛	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	⬛	Ground	20
21	GPIO09 (SPI_MISO)	⬛	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	⬛	(SPI_CE0_N) GPIO08	24
25	Ground	⬛	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	⬛	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	⬛	Ground	30
31	GPIO06	⬛	GPIO12	32
33	GPIO13	⬛	Ground	34
35	GPIO19	⬛	GPIO16	36
37	GPIO26	⬛	GPIO20	38
39	Ground	⬛	GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

Obrázek 3: Raspberry Pi 3 GPIO Header [10]

2.4 OS pro Raspberry Pi

RPi je na rozdíl od klasických PC, které jsou postaveny na architektuře x86/x64, postaveno na architektuře ARM. Tzn., že na RPi nemůžeme nasadit operační systém určený pro klasické PC. Z tohoto hlediska se jeví jako nejpříhodnější použít na RPi OS založený na GNU/Linux, který bývá kompilován na velké množství architektur a na jehož základě je postaven i Android, který byl speciálně vytvořen pro mobilní zařízení postavené na architektuře ARM. Tato myšlenka vedla k vytvoření oficiálního OS pro RPi, Raspbian. Jako základ pro Raspbian posloužil Debian linux, upravený pro potřeby RPi.

S rostoucí popularitou projektu RPi se začali objevovat další OS. Níže si rozebereme OS zmiňované na oficiální stránce projektu RPi.⁷

⁷<https://www.raspberrypi.org/downloads/>

2.4.1 Raspbian

Raspbian je oficiálně podporován Raspberry Pi Foundation jako primární systém pro RPi. Raspbian byl vytvořen jako nezávislý projekt Mikem Thompsonem a Peterem Greenem. První sestavení bylo dokončeno v roce 2012 a obsahovalo přes 35000 balíčků optimalizovaných pro RPi. Raspbian je stále v aktivním vývoji, jeho aktuální verze⁸ je postavena na Debian Jessie [13].

Aktuální verze obsahuje desktopové prostředí PIXEL (Pi Improved Xwindows Environment, Lightweight). PIXEL se skládá z modifikovaného prostředí LXDE a správce oken Openbox.

Obsahuje velké množství SW, což ho činí univerzální v různých oblastech nasazení. Obsahuje SW pro vzdělávání, programování, běžné desktopové aplikace, aplikace pro práci s multimédií, serverové aplikace. Díky široké komunitě kolem RPi je výborně podporován a zmapován.

2.4.2 Ubuntu MATE

Martin Wimpress a Rohith Madhavan vytvořili klon Ubuntu MATE pro RPi 2 a RPi 3.

Ubuntu MATE je linuxová distribuce odvozená od operačního systému Ubuntu, na jehož základech staví. Jako grafické prostředí používá MATE.

MATE je pokračováním desktopového prostředí GNOME starší řady 2, které bylo použito jako výchozí grafické prostředí ve starších verzích Ubuntu [14].

Díky použití odlehčené verze Ubuntu je možné spouštět tento rozšířený linuxový desktopový systém i na méně výkonném HW jako je RPi. RPi s Ubuntu MATE je použitelné jako nenáročný desktopový klient.

2.4.3 Snappy Ubuntu Core

Snappy Ubuntu Core je minimalistická odlehčená reedice Ubuntu. Je to transakčně aktualizovatelný OS určený pro běh na zařízeních IoT, cloudových zařízeních apod. Striktní oddělení jádra systému a ovladačů zařízení, OS a aplikační vrstvy, umožňuje u rozsáhlých aplikací paralelní vývoj [15].

Filozofie běhu aplikací je převzatá z Ubuntu Phone OS. Prakticky Ubuntu Core nahradil Ubuntu Phone OS, jehož vývoj je zastaven. Na zařízení je nainstalováno Ubuntu Core. Pomocí transakčně aktualizovatelného balíčkovacího systému Snap,

⁸RASPBIAN JESSIE verze 02 2017

který nahradil zastaralý systém balíčků .deb, se aplikace nahrávají do zařízení. Díky moderní filozofii, je možné spravovat různé verze, snadno mazat a aktualizovat aplikace. To je vhodné pro vývoj aplikací.

2.4.4 Windows 10 IoT Core

Windows 10 IoT Core je verze systému Windows 10 optimalizována pro běh na malých zařízeních, postavených na architektuře ARM nebo x86/x64. Pro vytváření aplikací můžeme využít technologie Microsoft, jako např. Universal Windows Platform (UWP) API nebo Visual Studio [16].

Běh aplikací je analogický s ostatními mobilními zařízeními od Microsoftu. Na RPi běží Windows 10 IoT Core a podobně jako např. na mobilní telefon, nahrajeme aplikace do zařízení a spustíme. Windows 10 IoT Core nám umožňuje přistupovat k rozhraním RPi jako je GPIO nebo displej.

2.4.5 OSMC

OSMC (Open Source Media Center) je open source multimediální centrum založené na Linuxu. Umožňuje přehrávat média z připojených zařízení, lokální síť i internetu. OSMC je založen na projektu Kodi⁹ a Debian Linuxu.

2.4.6 LibreELEC

LibreELEC je speciální linuxová distribuce vyvinutá pro běh projektu Kodi na speciálním HW určeném pro mediacentra.

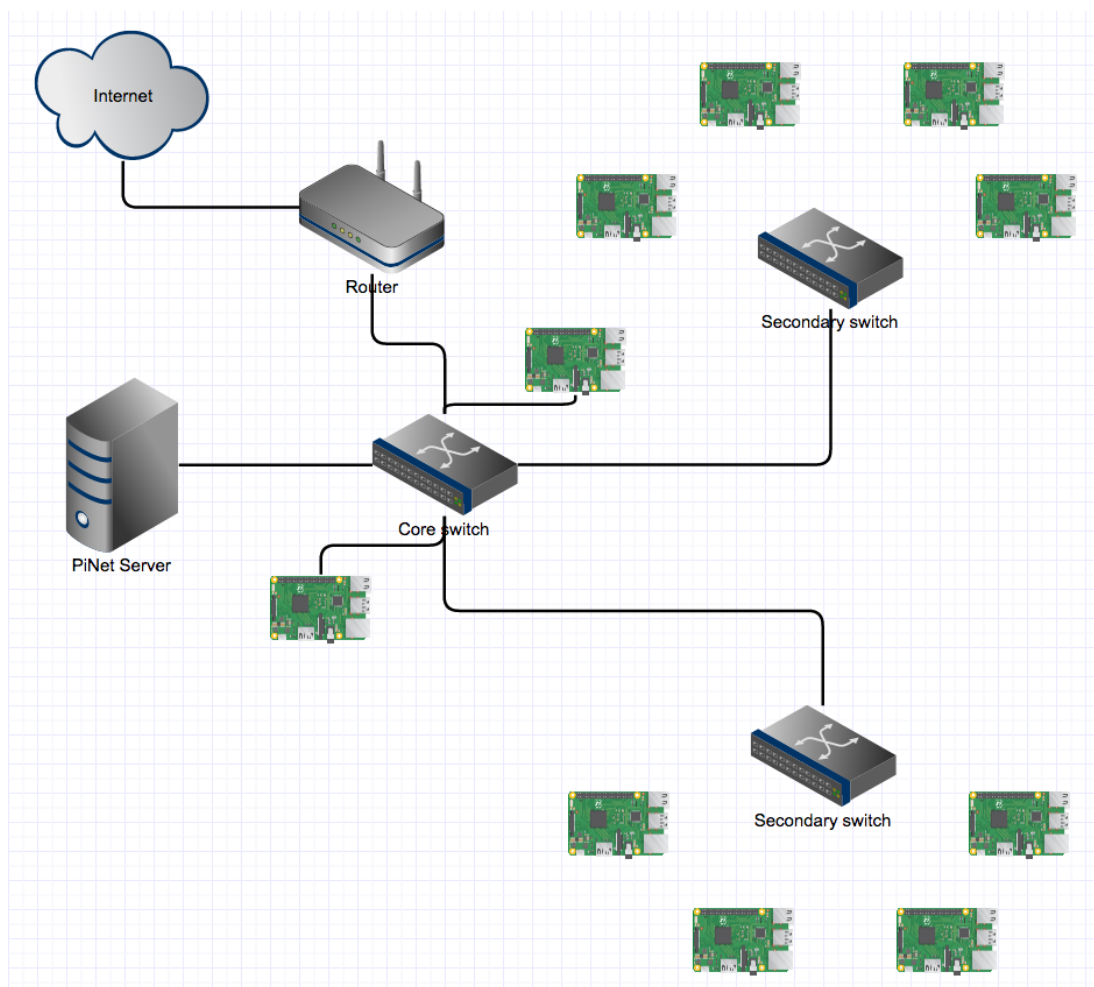
⁹<https://kodi.tv/about>

2.4.7 PiNet

PiNet je centralizovaný systém pro vytvoření školní sítě založené na RPi. Jádrem systému je výkonný server, na kterém běží Ubuntu s nainstalovaným PiNet serverem. Doporučený HW pro běh serveru je:

- dual core 1.5 GHz procesor (Intel Pentium nebo vyšší)
- 2 GB paměti RAM
- 40 GB místa na pevném disku
- Gigabitový Ethernet port (100 Mbit/s není oficiálně podporováno)

Na tomto serveru jsou uloženy uživatelské účty, sdílené složky a je z něj spouštěn po síti operační systém připojených RPi. Možné uspořádání PiNet sítě vidíme na obrázku 4.



Obrázek 4: PiNet network [17]

2.4.8 RISC OS

RISC OS je operační systém původně navržený Acorn Computers Ltd in Cambridge, England. Poprvé byl vydán v roce 1987 pro vlastní osobní počítač Archimedes založený na architektuře ARM. Po rozpadu Acorn Computers v roce 1998 se vývoj roztříštil mezi několik firem, které nezávisle pokračovali na vývoji. Od roku 2017 je jedna nezávislá větev vyvíjena RISCOS Ltd. a RISC OS Open community. Nejnovější stabilní verze lze spouštět na ARMv3/ARMv4, na ARMv5 Iyonix, ARMv7 Cortex-A8. K dispozici je vývojová verze pro Raspberry Pi [18].

2.4.9 Weather Station

Oracle Weather Station je projekt primárně určený pro školy, který umožňuje pomocí speciálního HW a SW shromažďovat pomocí různých snímačů údaje o počasí. Umožňuje zaznamenávat data nejen na místní úrovni, ale také zapisovat data do centrální databáze Oracle, takže tyto data mohou sdílet uživatelé z celého světa.

2.4.10 Výběr vhodného OS

Na základě výše uvedených vlastností OS vybereme nejvhodnější OS pro aplikaci v souboru laboratorních úloh dle kapitoly 4.

K výběru vhodného OS musíme stanovit kritéria, která jsou nezbytná pro realizaci úloh. Prostudováním zadání úloh byla zvolena tato kritéria:

- **Vzdálené síťové připojení** - možnost spouštět vzdáleně programy, vytěžovat data z čidel atd.
- **Obsluha periférií** - možnost programově přistupovat k perifériím, zejména k GPIO rozhraní RPi.
- **Zpracování a vizualizace dat** - možnost ukládat data pro pozdější zpracování a vytváření grafického výstupu.
- **Prezentace dat pomocí webu** - možnost zobrazit naměřené údaje pomocí webového rozhraní, optimálně s během webového serveru přímo na RPi.

Prověřením vybraných OS bylo zjištěno, že naše požadavky splňují Raspbian a Ubuntu Mate. S těmito OS je možné realizovat navržené úlohy pomocí nástrojů obsažených v instalaci. Konfigurace a zprovoznění potřebných nástrojů je dobře zdokumentovaná a existuje nepřehledné množství návodů, jak dosáhnout kýželného výsledku. Toto platí hlavně o Raspbianu. U ostatních OS je splnění našich kritérií buď vyloučené, nebo by jejich splnění vyžadovalo značné úsilí, protože tyto systémy jsou navrženy na speciální účely a nedají se tak snadno přizpůsobit. Porovnání vybraných OS je zhodnoceno v tabulce 5. Jako kritérium byl zvolen dvouhodnotový systém (1 - vyhovuje, 0 - nevyhovuje, N/A -nezjištěno, netestováno).

OS	vzdálené sít'ové připojení	obsluha periférií	zpracování vizualizace dat	prezentace dat pomocí webu
Raspbian	1	1	1	1
Ubuntu Mate	1	1	1	1
Snappy Ubuntu Core	0	1	1	0
Windows 10 IoT Core	0	1	1	0
OSMC	0	0	1	0
LibreELEC	0	0	0	0
PiNet	N/A	N/A	N/A	N/A
RISC OS	0	0	0	0
Oracle Weather Station	N/A	N/A	N/A	N/A

Tabulka 5: Porovnání OS pro RPi

2.5 Obsluha periférií

Obsluha periférií je cesta získávání online dat z okolí, ať už pomocí vstupů RPi, připojených čidel, nebo komunikace s navazujícími systémy. Této problematice a vlastnímu technickému řešení se budeme podrobně věnovat v kapitole 4 a 5.

3 Rešerše podobných projektů

Pro moji bakalářskou práci jsem našel inspiraci v několika publikacích nebo webových stránkách zabývajících se problematikou RPi. Níže si představíme publikaci **NORRIS, Donald. Raspberry Pi: projekty** a online projekt **Adafruit**¹⁰. Tyto dva projekty mi byly největší inspirací.

NORRIS, Donald. Raspberry Pi: projekty [23] - jedná se o publikaci, která komplexně popisuje práci s RPi. V úvodu je popsán HW a jednotlivé periferie RPi. Na tuto kapitolu navazuje popis SW, kde je postupně vysvětlena instalace OS do RPi, přes základní vlastnosti OS Linux až po základy programování, zejména v jazyce Python. Další kapitoly jsou návody jednotlivých projektů. Každá kapitola je jeden nezávislý projekt, kde je uvedeny požadavky na projekt (potřebné HW vybavení, způsob propojení s RPi), programová obsluha, případně potřebný SW a na konec shrnutí projektu.

Adafruit - RASPBERRY PI¹¹ - webová stránka obsahující v současné době cca 190 návodů (projektů) z mnoha odvětví postavených na RPi. Jednotlivé projekty jsou prezentovány moderní a srozumitelnou formou. Kromě textového popisu obsahují množství dalších informací jako obrázky, videa, schémata zapojení. V neposlední řadě obsahují konfigurace, zdrojové kódy, někde jsou k dispozici i celé vytvořené programy. Tato forma návodů dává možnost vyzkoušet si projekty i začátečníkům, nebo nadšencům bez hlubších znalostí dané problematiky.

¹⁰Společnost Adafruit byla založena v roce 2005 hackerkou a inženýrkou MIT, Limor "Ladyada" Fried. Jejím cílem bylo vytvořit místo pro online učení elektroniky a vytváření návrhů produktů pro tvůrce všech věkových skupin a dovedností. Adafruit se rozrostla na více než 100 zaměstnanců s továrnou 50 000 m². [24]

¹¹<https://learn.adafruit.com/category/raspberry-pi>

4 Návrh souboru vzorových úloh

V této kapitole si ukážeme návrh a způsob řešení celkem dvanácti vzorových úloh. Jedná se o základní úlohy, které slouží k vlastnímu zprovoznění RPi a dále pak k demonstraci možností RPi jako řídicího systému pro měření fyzikálních veličin, jejich zpracování a vizualizace.

4.1 Instalace OS a základní konfigurace Raspberry Pi

Prvním krokem, bez kterého není možné RPi používat je instalace operačního systému. Pokud jsme nezakoupili RPi s předinstalovaným OS, musíme umístit obraz OS na SD kartu, ze které je schopné RPi tento systém spustit. Na základě výběru dle kapitoly 2.4 jsme se rozhodli používat oficiální distribuci Raspbian. Nicméně postup instalace lze aplikovat i na ostatní dostupné distribuce.

Vlastní instalace v podstatě spočívá jenom v překopírování sestaveného obrazu disku na SD kartu. Tento krok můžeme provést ručním překopírováním bitové kopie obrazu disku. Ukážeme si, jak se tento krok provádí pomocí nástroje Win32DiskImager v MS Windows nebo pomocí standardních nástrojů GNU/Linux.

Další možností, která nevyžaduje žádné speciální nástroje, je použití instalátoru NOOBS¹². Tento instalátor obsahuje obraz distribuce Raspbian, kterou nainstaluje na SD kartu. Pokud má RPi k dispozici internetové připojení umožňuje nám instalaci dalších distribucí online. Instalace NOOBS spočívá pouze v překopírování obsahu staženého zip archivu na SD kartu, která je naformátována na souborový systém FAT32. Po nastartování RPi se otevře průvodce, kde zvolíme jaký OS chceme nainstalovat a instalátor vše automaticky nainstaluje.

Další výhodou NOOBS je možnost nainstalovat více různých OS vedle sebe, jediné omezení je velikost SD karty. Pak NOOBS funguje jako zavaděč a umožňuje spouštět libovolný nainstalovaný OS.

4.2 Konfigurace systému pomocí „raspi-config“ [19]

Raspi-config je přehledný nástroj, který slouží pro konfiguraci stěžejních vlastností RPi v distribuci Raspbian. Zjednodušuje uživateli konfiguraci systému bez hlubších znalostí linuxových nástrojů pomocí přehledného grafického rozhraní.

¹²NOOBS - New Out Of the Box Software, je instalátor sloužící pro zjednodušení procesu instalace OS na RPi. Pomocí NOOBS lze instalovat nejen Raspbian, ale i alternativní OS, umístěné na SD kartě, nebo pokud je dostupné připojení k internetu i pomocí sítě.

Jelikož raspi-config zasahuje do systémových konfiguračních souborů, je jej třeba spouštět s právy uživatele „root“ a některá nastavení se aplikují až po restartu systému.

Dále budou popsány jednotlivé vlastnosti, které raspi-config umožňuje nastavit.

4.2.1 Expand Filesystem

Tato volba se využívá v případě, že obraz Raspbian je menší než velikost SD karty, na kterou tento systém instalujeme. Takže pokud použijeme větší kartu než 4 GiB, což je přibližně velikost oficiálního obrazu Raspbianu, zbytek karty by nebyl přístupný. Volbou Expand Filesystem rozšíříme oblast na plnou velikost karty¹³.

4.2.2 Change User Password

Volba umožňuje nastavit heslo uživatele.

Ve výchozím nastavení RPi je nastaven uživatel `pi` a heslo `raspberry`. Z bezpečnostního hlediska, hlavně v případě, kdy bude RPi připojeno na internet, je záhodno toto nastavení změnit.

4.2.3 Boot Options

Tato volba umožňuje nastavit, zda se po startu spustí grafické prostředí nebo terminálové uživatelské rozhraní a zda dojde k automatickému přihlášení uživatele.

4.2.4 Internationalisation Options

Volba obsahuje tři položky přizpůsobující národní prostředí distribuce.

- **Change locale** - mění nastavení lokalizace distribuce.
- **Change timezone** - mění časovou zónu distribuce.
- **Change keyboard layout** - mění rozložení klávesnice.

¹³V případě instalace pomocí NOOBS se oblast rozšíří automaticky při instalaci

4.2.5 Enable Camera

Tato volba povoluje použití modulu kamery připojeného přes rozhraní CSI. Pokud chceme používat tento modul musíme mít rezervováno minimálně 128 MiB RAM pro GPU viz kapitola 4.2.7.

4.2.6 Overclock

Volba nabízí možnost přetaktovat frekvenci CPU RPi. Přetaktováním můžeme snížit životnost RPi. Může dojít k nestabilitě systému. Pro tento případ můžeme vyřadit přetaktování v průběhu startu systému stiskem klávesy **shift**.

4.2.7 Advanced Options

- **Overscan** - pokud je tato volba aktivní, neroztáhne obraz na celou obrazovku, ale zobrazí obraz v nominálním rozlišení a zbytek obrazovky vyplní černými okraji.
- **Hostname** - nastavuje název počítače, pod kterým je identifikován v síti.
- **Memory split** - mění velikost paměti RAM, kterou využívá GPU.
- **SSH** - povoluje vzdálené připojení k RPi přes SSH.
- **Device Tree** - povoluje správu zařízení pomocí „Device Tree“¹⁴.
- **SPI** - povoluje automatické zavedení modulu jádra pro obsluhu SPI rozhraní při startu systému.
- **I2C** - povoluje automatické zavedení modulu jádra pro obsluhu I2C rozhraní při startu systému.
- **Serial** - povoluje výstup hlášení systému přes sériovou linku.
- **Audio** - přepíná zvukový výstup mezi HDMI a analogový 3,5 mm jack.
- **Update** - aktualizace nástroje raspi-config na nejnovější verzi.

¹⁴Více informací o implementaci Device Tree v RPi <https://www.raspberrypi.org/documentation/configuration/device-tree.md>

4.3 Obsluha ETHERNET

Pokud chceme používat RPi, nebo obecně i jiný počítač, jako zařízení sloužící ke sběru dat, nebo řízení automatizačních procesů, musíme mít přístup k naměřeným údajům a možnost ovládat tento systém. Přímé ovládání pomocí klávesnice, myši a výstup na monitor v tomto případě není optimální řešení a v praxi se moc nepoužívá.

Podstatně lepší řešení je vzdálené připojení. U jednodušších zařízení s nízkým datovým tokem je plně dostačující komunikace po sériové lince. Pro zařízení přenášejících velké množství dat, je nejvhodnější síťové připojení. V průmyslových aplikacích se díky robustnosti používá metalické, nebo optické připojení. U méně kritických aplikací lze využít i bezdrátové řešení.

Konkrétně pro využití RPi jako řídicí prvek zabudovaný v HW modulu¹⁵ je důležitá možnost spravovat zařízení, bez nutnosti připojování klávesnice a myši, nebo demontování RPi z HW modulu.

Pro naše potřeby je plně vyhovující vzdálené připojení přes SSH. S tímto nástrojem jsme schopni plně ovládat RPi, obsloužit a nakonfigurovat veškeré další úlohy.

SSH (Secure Shell)[20] je program a zároveň zabezpečený komunikační protokol v počítačových sítích, které používají TCP/IP. SSH byl navržen jako náhrada za telnet a další nezabezpečené vzdálené shelly (rlogin, rsh apod.), které posílají heslo v nezabezpečené formě a umožňují tak jeho odposlechnutí při přenosu pomocí počítačové sítě. Šifrování přenášených dat, které SSH poskytuje, slouží k zabezpečení dat při přenosu přes nedůvěryhodnou síť, jako je například Internet.

Pro nastavení SSH připojení s RPi je žádoucí změnit výchozí nastavení sítě s dynamicky přidělovanou adresou na statickou IP adresu. Zjednoduší se tím hledání zařízení v síti. Dále je nutné povolit SSH připojení v rasp-config, viz kapitola 6.2.2.

Nyní se můžeme připojit ze vzdáleného počítače a využívat terminálu RPi. Pro pohodlnější práci si můžeme nainstalovat souborový manažer Midnight Commander¹⁶.

¹⁵HW modul je zařízení, na kterém budeme realizovat jednotlivé úlohy, viz kapitola 5

¹⁶<https://midnight-commander.org/>

4.4 Základní využití GPIO - vstup

Jak již bylo zmíněno v kapitole 2.3, RPi nám umožňuje využívat 25 GPIO linek, kromě jiných funkcí, hlavně jako digitální vstupy/výstupy.

Pro práci s GPIO budeme využívat přístup přes „GPIO Sysfs Interface“^[21] pomocí standardních linuxových příkazů `cat` a `echo`.

„GPIO Sysfs Interface“ nám umožňuje na uživatelské úrovni přímo ovládat směr a stav GPIO linek přes soubory v Sysfs¹⁷. Pro přístup k GPIO máme tři skupiny souborů v `/sys/class/gpio`:

- Ovládací rozhraní - slouží k získání kontroly nad GPIO.
- Vlastní soubory jednotlivých GPIO.
- GPIO ovladače (`gpio_chip` instances).

Pomocí zápisu/čtení těchto souborů jsme schopni ovládat jednotlivé GPIO linky.

Nejdříve musíme vybraný GPIO pin zpřístupnit pro další použití. Uživatel přebírá od systému kontrolu nad GPIO tím, že zapíše číslo linky do souboru `export`.

```
echo "21" > /sys/class/gpio/export # 21 = GPIO21(pin #40)
```

Následně nastavíme linku jako vstupní nebo výstupní zápisem `in` nebo `out` do souboru `direction`.

```
echo "in" > /sys/class/gpio/gpio21/direction
```

Pokud je linka nastavena jako vstupní, čtení stavu provádíme pomocí přečtení hodnoty souboru `value`.

```
cat /sys/class/gpio/gpio21/value
```

V případě nastavení výstupní linky, ovládáme hodnotu na výstupu obdobně zápisem hodnoty do souboru `value`.

```
echo 1 > /sys/class/gpio/gpio21/value
```

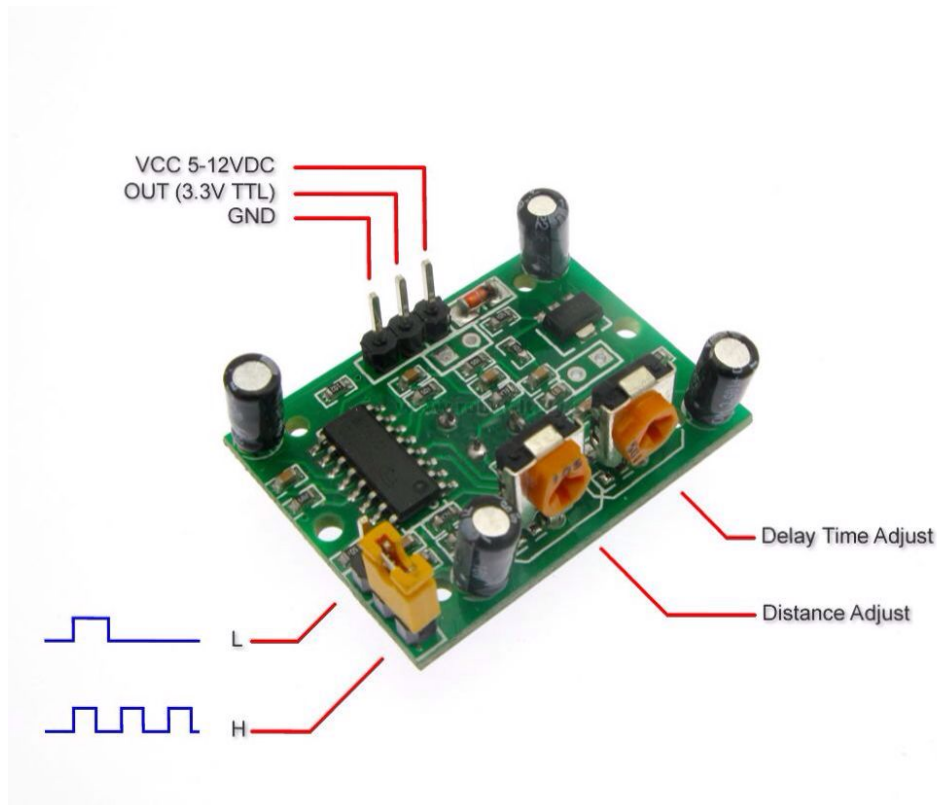
Po ukončení práce s GPIO linkou je žádoucí uvolnit linku z uživatelského prostoru, jinak až do vypnutí RPi bude linka nepřístupná systému.

```
echo "21" > /sys/class/gpio/unexport
```

¹⁷Sysfs je v informatice virtuální systém souborů, který je zahrnut v jádře Linuxu od verze 2.6. Sysfs exportuje informace o zařízeních a ovladačích z jádra systému do uživatelského prostoru pomocí adresářového stromu umístěného v adresáři `/sys` a je používán i pro konfiguraci systému.^[22]

Pro vyzkoušení tohoto řešení využijeme pohybový senzor PIR¹⁸, kde budeme detekovat jeho aktivaci. Při aktivaci senzoru soubor `.../gpio21/value` bude mít hodnotu 1.

PIR senzor je třeba použít takový, který pracuje s výstupní úrovní 3,3 V, viz obrázek 5. Pokud by se na GPIO objevilo napětí vyšší, může dojít ke zničení vstupu, potažmo celého RPi.



Obrázek 5: PIR senzor

4.5 Základní využití GPIO – výstup

Obdobně, jak je popsáno v kapitole 4.4 realizujeme s využitím sysfs ovládní stavu výstupu GPIO. Na výstupu bude připojena LED dioda¹⁹, která se bude na základě zápisu log. 0/1 aktivovat.

Kombinací s úlohou dle kapitoly 4.4 vytvoříme alarm reagující na aktivaci PIR senzoru. Po aktivaci alarmu rozblikáme LED a vypíšeme alarmové hlášení do terminálu. Pro realizaci této úlohy využijeme programovací jazyk Python a třídu `RPi.GPIO`, která je součástí balíčku PyPi. Tento balíček není třeba instalovat, je součástí instalace OS Raspbian. Jak můžeme vidět ve výpisu 1, přístup přes tuto

¹⁸<https://www.adafruit.com/product/189>

¹⁹Pro připojení přímo na GPIO výstup je možné použít LED diodu s odběrem max. 16 mA, jinak může dojít ke zničení vstupu, viz kapitola 2.3.2.

knihovnu je velmi intuitivní. Konfigurace se sestává s přiřazení módu GPIO jako vstup, nebo výstup `GPIO.setup(38, GPIO.OUT)` a dále pak ze čtení nebo zápisu tohoto vstupu `GPIO.output(38, 0)`.

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40,GPIO.IN)    #IN PIR sensor
GPIO.setup(38, GPIO.OUT) #OUT LED

x=1
while True:
    i=GPIO.input(40)
    cas=time.strftime("%d.%m.%Y_%H:%M")
    if i==0 and x==1: #PIR NORMAL
        print cas, "_NENI_ALARM"
        x=0
        GPIO.output(38,1 ) #LED OFF
        time.sleep(0.1)
    elif i==1 and x==0: #PIR ALARM
        print cas, "_ALARM_AKTIVACE!!!"
        x=1
        for count in range(0, 10):
            GPIO.output(38, 0) #LED ON
            time.sleep(0.2)
            GPIO.output(38, 1) #LED OFF
            time.sleep(0.2)
```

Výpis 1: pir.py

4.6 Programová obsluha periférií

V této kapitole využijeme GPIO linku pro připojení digitálního čidla teploty a vlhkosti DHT22(AM2302).

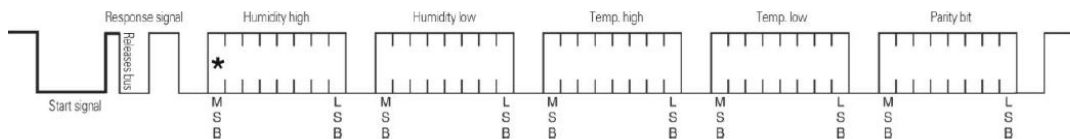
DHT22[25] obsahuje kapacitní snímač vlhkosti a polovodičový snímač teploty spojený s osmibitovým mikropočítačem. Snímač je digitálně kalibrován. DHT22 komunikuje po jednovodičové sběrnici, přenos je možný do max. vzdálenosti 20 m.

Technické parametry:

- Napájení: 3,3...6 V DC.
- Výstupní digitální signál přes single-bus(SDA).

- Kapacitní snímač pro měření relativní vlhkosti, DS18B20 pro měření teploty.
- Rozsah měření 0...100% RH, -40...125°C.
- Přesnost $\pm 2\%$ RH, $\pm 0,2^\circ\text{C}$.
- Rozlišení 0,1% RH, 0,1°C.

Komunikace mezi RPi a DHT22 začíná inicializací čidla po uvedení pod napětí. Po dobu 2 vteřin je čidlo nedostupné a není možné s ním komunikovat. Po uplynutí této doby DHT22 čeká na „start signal“ od RPi. Po obdržení signálu odešle odezvu a následně 40 bitové slovo obsahující data, viz obrázek 6.



Obrázek 6: DHT22 komunikační protokol [25]

Pro komunikaci se senzorem využijeme ovladač ze stránek projektu adafruit²⁰. Po nainstalování máme k dispozici knihovny a vzorové programy v jazyce Python potřebné pro odečtení teploty a relativní vlhkosti. Program si ještě rozšíříme o výpočet rosného bodu. Výsledný kód, viz výpis 2.

```
import sys
import Adafruit_DHT
import math

#konstanty pro vypočet dewpoint
x = 1
k~ = 237.7
z~ = 6.11
c = 7.5

# Parse command line parameters.
sensor_args = { '11': Adafruit_DHT.DHT11,
                '22': Adafruit_DHT.DHT22,
                '2302': Adafruit_DHT.AM2302 }
if len(sys.argv) == 3 and sys.argv[1] in sensor_args:
    sensor = sensor_args[sys.argv[1]]
    pin = sys.argv[2]
```

²⁰<https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging/software-install-updated>


```

else :
    print('usage: sudo ./Adafruit_DHT.py [11|22|2302] GPIOpin#')
    print('example: sudo ./Adafruit_DHT.py 2302 4 Read_from_an_
          AM2302_connected_to_GPIO_#4')
    sys.exit(1)

humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)
if humidity is not None and temperature is not None:
    #***** vypocet dewpoint dle CHMU*****
    d = c * temperature / (k~ + temperature)
    Es = z~ * 10 * d
    E = humidity * Es / 100
    f = math.log(E / z~, 10)
    dewpoint = f * k~ / (c - f)
    #*****
    print('Temp: {0:0.1f}'.format(temperature))
    print('Humidity: {0:0.1f}'.format(humidity))
    print('Dewpoint: {0:0.1f}'.format(dewpoint))
else :
    print('Failed to get reading. Try again!')
    sys.exit(1)

```

Výpis 2: *dht.py*

4.7 Využití I2C sběrnice

I2C(Inter-Integrated Circuit)[26] je multi-master, multi-slave, packet switched, single-ended, sériové sběrnice, navržena firmou Philips začátkem 80. let, ke komunikaci mezi obvody umístěnými na stejné desce plošných spojů.

I2C používá pouze dvě obousměrné linky, sériové datové vedení (SDA) a hodinový signál (SCL). Typická napětí se používají 5 nebo 3,3 V.

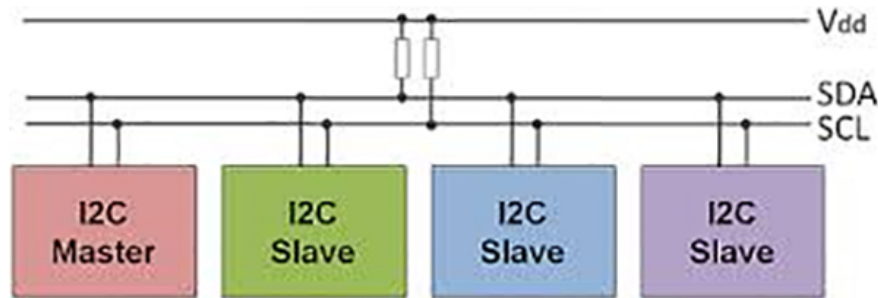
Původní rychlost komunikace byla maximálně 100 kbit/s. Od roku 1998 máme k dispozici rychlejší přenosy 400 kbit/s (fastmode) a 3,4 Mbit/s (high speed).

I2C nemusí sloužit pouze pro komunikaci mezi obvody na jedné desce, ale může být použita pro připojení obvodů přes kabel. Jednoduchost a flexibilita aplikace činí I2C použitelnou pro široké spektrum aplikací.

Mezi hlavní rysy I2C sběrnice patří:

- Pouze dva vodiče pro přenos dat.
- Přenosová rychlost nemusí být striktně definována, jako například u RS232, master generuje hodiny sběrnice.

- Existují jednoduché „master/slave“ vztahy mezi všemi obvody.
- Každé zařízení připojené ke sběrnici je softwarově adresovatelné jedinečným adresou.
- I2C je skutečná „multi-master“ sběrnice, poskytuje arbitráž a detekce kolizí.



Obrázek 7: I2C sběrnice[27]

4.7.1 Obsluha I2C pomocí Raspberry Pi

Připojení k I2C sběrnici je umožněno přes GPIO header, viz obrázek 3:

- Pin #03 SDA
- Pin #05 SCL

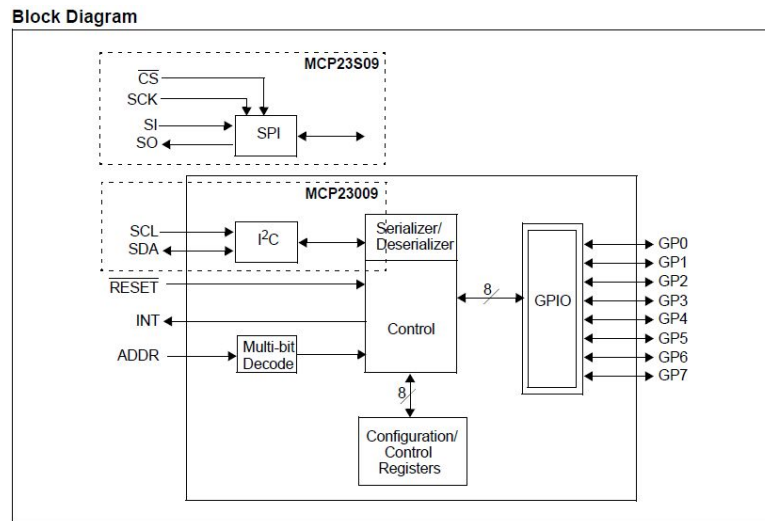
Abychom mohli začít využívat sběrnici I2C musíme mít povolenou sběrnici v rasp-config, pro načtení modulu jádra I2C, viz kapitola 6.2.2.

Nejednodušší přístup k I2C sběrnici nám umožní balíček I2C-Tools, který je standardně součástí OS Raspbian. Pomocí programu `i2cset` a `i2cget` můžeme realizovat čtení a zápis na sběrnici I2C. Sofistikovanější přístup nám umožní knihovna `python-smbus` a její aplikace v jazyce Python.

4.7.2 Ovládání digitálních vstupů a výstupů pomocí sběrnice I2C

S využitím obvodu MCP23009 (8-bitový I2C port expander) [28] budeme realizovat totožné úlohy jako v kapitolách 4.4 a 4.5. Účelem je ukázat jiný způsob práce s digitálním vstupem/výstupem. V tomto případě se jedná o daleko bezpečnější metodu, jelikož nepřipojujeme HW přímo na piny BMC2837, viz kapitola 2.3.4 a nemůže dojít k jeho zničení při vyšším proudovém zatížení, nebo napětím vyšším než 3,3 V.

Funkce obvodu MCP23009 je patrná z obrázku 8.



Obrázek 8: MCP23009 - blokový diagram[28]

Pomocí zápisu do registru IODIR , viz obrázek 9, budeme ovládat LED diodu připojenou na GP6, konkrétně zápisem log 0.

REGISTER 1-1: IODIR – I/O DIRECTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
IO7	IO6	IO5	IO4	IO3	IO2	IO1	IO0
bit 7							bit 0

Obrázek 9: MCP23009 - IODIR registr[28]

Čtení stavu PIR senzoru, připojeného na GP7, provádíme z registru GPIO, viz obrázek 10.

REGISTER 1-10: GPIO – GENERAL PURPOSE I/O PORT REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GP7	GP6	GP5	GP4	GP3	GP2	GP1	GP0
bit 7							bit 0

Obrázek 10: MCP23009 - GPIO registr[28]

Pro obsluhu úloh využijeme balíček I2C-Tools, konkrétně takto:

```
i2cset -y 1 0x20 0 0xbf #-y 1 I2C sběrnice 1, 0x20 adresa I2C zařízení, 0 adresa registru, 0xbf zapsaná hodnota(1011 1111b)
i2cget -y 1 0x20 9 #-y 1 I2C sběrnice 1, 0x20 adresa I2C zařízení, 9 adresa registru
```

A dále obsluhu pomocí programu `pir_i2c.py`, jak je patrné z výpisu 3.

```
import RPi.GPIO as GPIO
import time
from smbus import SMBus

bus = SMBus(1) # 1 - /dev/i2c-1
address = 0x20

data = bus.read_i2c_block_data(address, 0x09, 1)
bus.write_byte_data(0x20, 0x00, 0xff) #LED OFF
time.sleep(0.1)
bus.write_byte_data(0x20, 0x00, 0xbf) #LED ON
```

Výpis 3: pir_i2c.py

4.7.3 Využití I2C sběrnice pro měření elektrických veličin (AD převodník)

K měření napětí využijeme modulu AD(DA) převodníku osazeným obvodem PCF8591²¹.

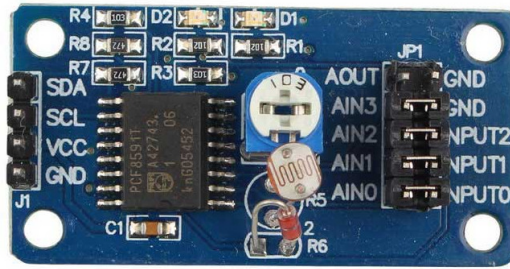
Na obrázku 11 vidíme skutečné provedení modulu. Modul je osazen 8-bitovým převodníkem se 4 analogovými vstupy, 1 analogovým výstupem a sériovým I2C rozhraním, charakterizovaný těmito vlastnostmi[29]:

- Napájecí napětí 2,5...6 V.
- Sériový vstup/ výstup přes I2C sběrnici.
- Hardwarová volba adresy pomocí 3 pinů.
- Vzorkovací frekvence dána rychlostí I2C sběrnice.
- 4 analogové vstupy nastavitelné jako jednoduché nebo diferenciální.
- 1 analogový výstup.
- Rozsah vstupního napětí od VSS do VDD.

Modul je napájen přes konektor J1, kterým je také připojen k I2C sběrnici. Analogový vstup/výstup je realizován přes konektor JP1. Na vstup AIN0 je připojen potenciometr, na AIN1 foto-odpor, na AIN2 teplotně závislý odpor. Pomocí

²¹<http://www.dx.com/cs/p/pcf8591-ad-da-converter-module-analog-to-digital-and-analog-to-digital-conversion-for-arduino-420245#.WMzutBLhBTY>

jumperů můžeme nastavit zda budeme využívat vstupy s těmito čidly, nebo je použijeme pro vstup externího napětí.



Obrázek 11: AD(DA) převodník PCF8591

Pro obsluhu modulu máme vytvořený program v jazyce Python `pcf8591.py`, který nám umožňuje číst hodnoty napětí na vstupech AIN a nastavovat napětí na výstupu AOUT. Výpis 4 nám ukazuje, jakým způsobem získáváme data ze senzoru.

```
import time
from smbus import SMBus
from sys import exit

bus = SMBus(1) # 1 - /dev/i2c-1
address = 0x48
A0 = 0x40 # D/A active select channel 0 POTENCIOMETER
A1 = 0x41 # D/A active select channel 1 FOTO R
A2 = 0x42 # D/A active select channel 2 R Teplota ???
A3 = 0x43 # D/A active select channel 3 vstup

print("Výběr_fuknce:")
print("0.....A/D_kanál_0_-_Potenciometr")
print("1.....A/D_kanál_1_-_Fotoodpor")
print("2.....A/D_kanál_2_-_Teplotně_závislý_odpor")
print("3.....D/A_převodník ,_výstup_měření_na_A/D_kanál_3")
n = int(raw_input("Zadej_volbu(1_~_3):"))

if n == 3:
    print
    out = int(raw_input("Zadej_hodnotu(0_~_100%):"))
    out = out*255/100
    bus.write_byte_data(address,A3,out)
    value = bus.read_byte(address)
    time.sleep(0.1)
    value = bus.read_byte(address)
    print("A/D_OUT:.....%1.3f_[V]_.. " %(value*3.3/255))
```

```

exit()
elif n == 0: mode = A0
elif n == 1: mode = A1
elif n == 2: mode = A2
else:
    print("NEPLATNÁ_VOLBA!!!")
    exit()

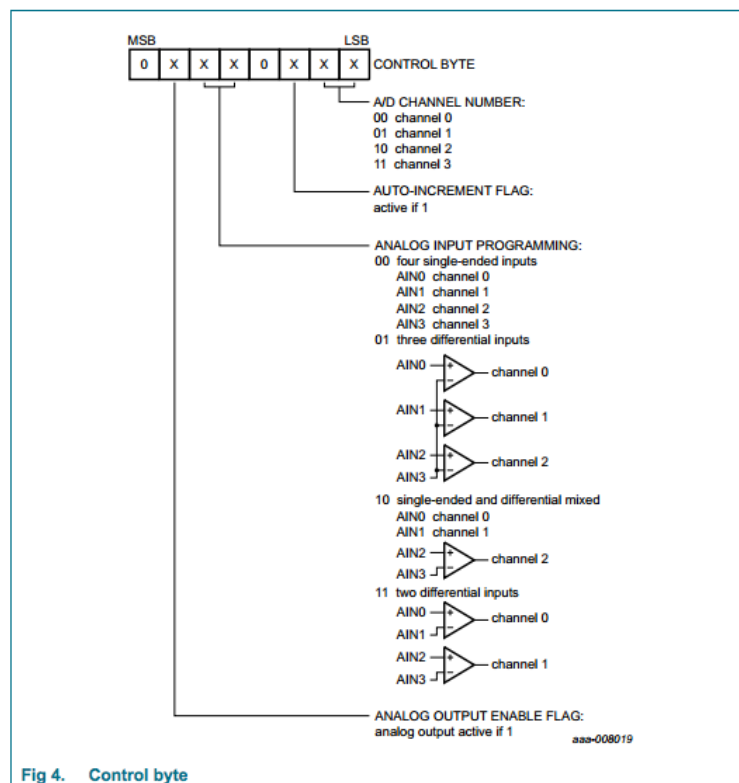
while True:
    bus.write_byte(address, mode)
    value = bus.read_byte(address)
    time.sleep(0.1)
    value = bus.read_byte(address)
    print("A/D_OUT: %1.3f [V]" %(value * 3.3 / 255))

```

Výpis 4: pcf8591.py

Nejdříve musíme do „control registru“ zapsat hodnotu určující funkci převodníku a aktivní kanál, jak je patrné z obrázku 12. V případě použití DA převodníku ještě zapíšeme požadovanou výstupní hodnotu.

Přečtení naměřené hodnoty probíhá jednoduše načtením slova se zadáním adresy čidla.

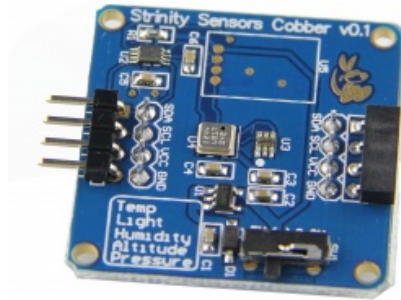


Obrázek 12: Control registr PCF8591[29]

4.7.4 Využití I2C sběrnice pro měření neelektrických veličin

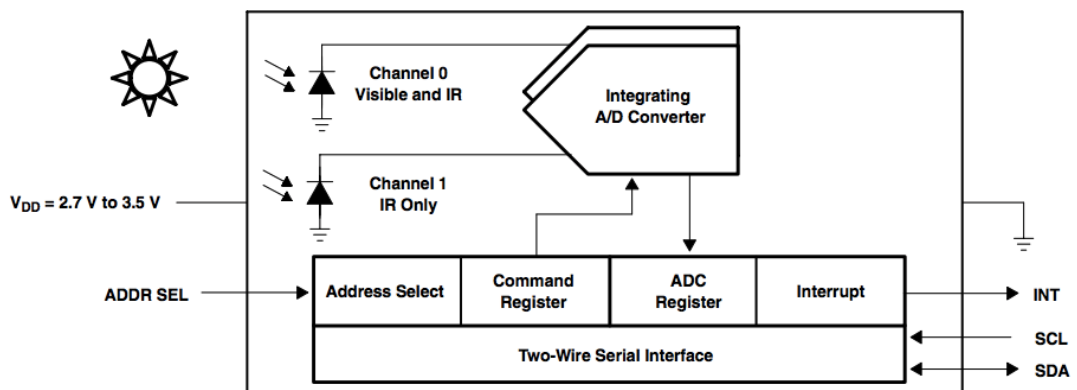
Pro ukázkou měření a zpracování neelektrických veličin použijeme sdružený senzor umožňující měřit teplotu, atmosférický tlak a intenzitu osvětlení.²²

Na obrázku 13 vidíme skutečné provedení. Modul je osazen čidlem intenzity osvětlení TLS2561 a digitálním čidlem tlaku BMP180.



Obrázek 13: Modul měření neelektrických veličin

TLS2561[30] je převodník intenzity osvětlení na digitální hodnotu s výstupem na I2C sběrnici. Je osazen širokopásmovou diodou pro snímání viditelného a infračerveného záření a diodou snímající pouze infračervené záření. Dva integrované 16-bitové AD převodníky převedou naměřený proud na digitální hodnotu reprezentující osvětlení v luxech. Na obrázku 14 vidíme funkční blokový diagram převodníku.



Obrázek 14: Funkční blokový diagram TLS2561[30]

²²<http://www.dx.com/cs/p/4-in-1-temperature-pressure-altitude-light-sensor-module-for-raspberry-pi-arduino-blue-359549#.WPt0qFOLRTY>

Pro získání dat z čidla TLS2561 máme vytvořený program `tls2561.py` v jazyce python. Výpis 5, nám ukazuje jakým způsobem získáváme data ze senzoru.

Nejdříve musíme zapsat do „control registru“ hodnotu 0x03 ,čímž zapneme čidlo, viz obrázek 15.

7	6	5	4	3	2	1	0	
0h	Resv	Resv	Resv	Resv	Resv	Resv	POWER	CONTROL
Reset Value:	0	0	0	0	0	0	0	0

FIELD	BIT	DESCRIPTION
Resv	7:2	Reserved. Write as 0.
POWER	1:0	Power up/power down. By writing a 03h to this register, the device is powered up. By writing a 00h to this register, the device is powered down. NOTE: If a value of 03h is written, the value returned during a read cycle will be 03h. This feature can be used to verify that the device is communicating properly.

Obrázek 15: Control registr TLS2561[30]

Dále zapíšeme do „timing registru“ hodnotu 0x02, čímž nastavíme čas měření na 402 ms, viz obrázek 16.

7	6	5	4	3	2	1	0	
1h	Resv	Resv	Resv	GAIN	Manual	Resv	INTEG	TIMING
Reset Value:	0	0	0	0	0	0	1	0

FIELD	BIT	DESCRIPTION
Resv	7-5	Reserved. Write as 0.
GAIN	4	Switches gain between low gain and high gain modes. Writing a 0 selects low gain (1X); writing a 1 selects high gain (16X).
Manual	3	Manual timing control. Writing a 1 begins an integration cycle. Writing a 0 stops an integration cycle. NOTE: This field only has meaning when INTEG = 11. It is ignored at all other times.
Resv	2	Reserved. Write as 0.
INTEG	1:0	Integrate time. This field selects the integration time for each conversion.

Integration time is dependent on the INTEG FIELD VALUE and the internal clock frequency. Nominal integration times and respective scaling between integration times scale proportionally as shown in Table 6. See Note 5 and Note 6 on page 5 for detailed information regarding how the scale values were obtained; see page 22 for further information on how to calculate lux.

Table 6. Integration Time

INTEG FIELD VALUE	SCALE	NOMINAL INTEGRATION TIME
00	0.034	13.7 ms
01	0.252	101 ms
10	1	402 ms
11	--	N/A

Obrázek 16: Timing registr TLS2561[30]

Po zpoždění 0,5 s, při kterém se změří hodnoty, přečteme hodnoty z registrů AD převodníků ADC0 a ADC1.

Na závěr nám zbývá spojit „low a high byte“ do jedné hodnoty.


```

from smbus import SMBus
import time

bus = SMBus(1)
bus.write_byte_data(0x39, 0x00 | 0x80, 0x03)
bus.write_byte_data(0x39, 0x01 | 0x80, 0x02)
#0x39 adresa I2C zařízení, 0x00(1) adresa registru, 0x80 start, 0x02
  (3)hodnota
time.sleep(0.5)
data = bus.read_i2c_block_data(0x39, 0x0C | 0x80, 2)
data1 = bus.read_i2c_block_data(0x39, 0x0E | 0x80, 2)
#0x39 adresa I2C zařízení, 0x0C(E) low(high) byte ADC, 0x80 start, 2
  2 byte ADC low,ADC high

ch0 = data[1] * 256 + data[0]
ch1 = data1[1] * 256 + data1[0]

```

Výpis 5: *tls2561.py*

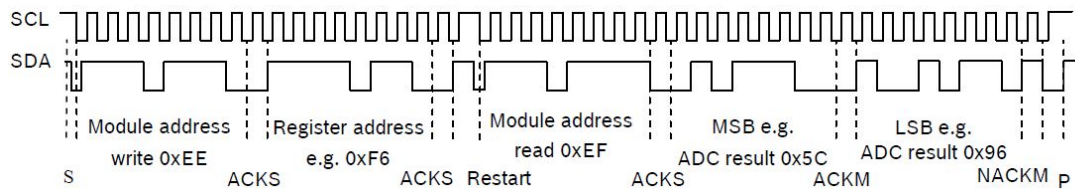
BMP180[31] je přesný digitální snímač tlaku. BMP180 s nízkým napájecím napětím a příkonem, je optimalizováno pro použití v mobilních telefonech, PDA a navigační zařízení s GPS. Rozhraní I2C umožňuje snadné ovládání a integraci s mikrokontrolérem.

BMP180 je založen na piezo-rezistivní technologii s vysokou přesností, linearitou a dlouhodobou stabilitou.

Vlastnosti převodníku BMP180:

- Měřený rozsah tlaku: 300 ... 1100 hPa (+9000 m ... -500 m nadmořské výšky)
- Napájení: 1,8 ... 3,6 V (VDD), 1,62 ... 3,6 V (VDDIO)
- Spotřeba: 5 μ A při 1 vzorku/sec. ve standardním módu
- Šum: 0,06 hPa (0,5 m) v úsporném režimu, 0,02 hPa (0,17 m) v módu rozšířené přesnosti
- Integrované teplotní čidlo
- I2C digitální výstup
- Kalibrovaná čidla

Na obrázku 17 vidíme časový diagram typické komunikace BMP180 po sběrnici I2C.



Obrázek 17: BMP180 - časový diagram čtení 16-bit. hodnoty AD [31]

Pro komunikaci se senzorem využijeme ovladač ze stránek projektu adafruit²³. Po nainstalování máme k dispozici knihovny a vzorové programy v jazyce Python potřebné pro odečtení teploty, absolutního atmosférického tlaku, atmosférického tlaku přepočteného na hladinu moře a nadmořské výšky.

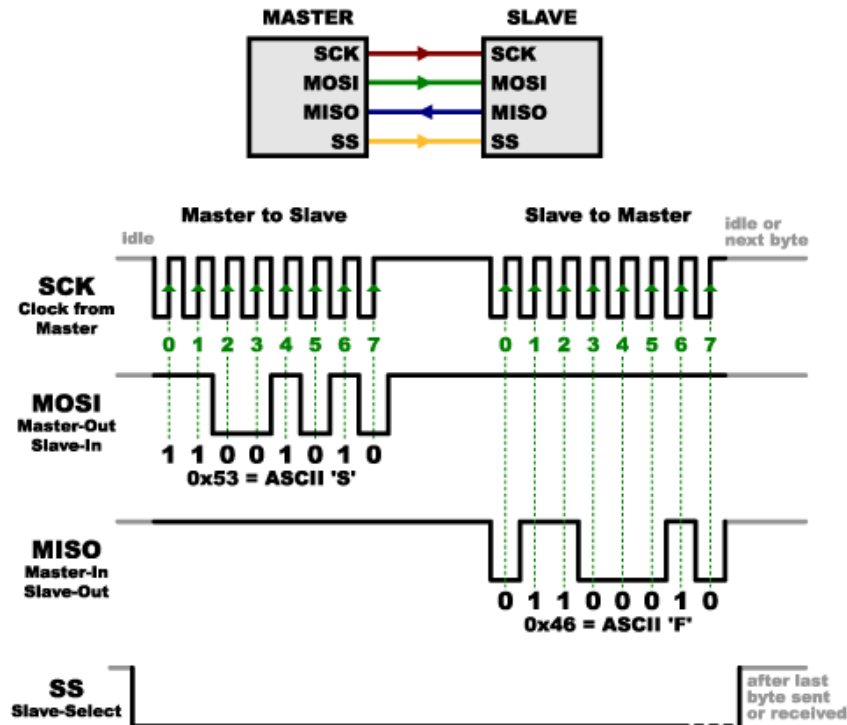
4.8 Obsluha SPI

SPI (Serial Peripheral Interface)[32] je sériová synchronní sběrnice, která se využívá pro komunikaci mikroprocesoru s perifériemi, jako jsou paměti EEPROM, AD převodníky, displeje. SPI je plně duplexní master-slave sběrnice. SPI využívá pro komunikaci čtyři vodiče:

- **SCLK** (Serial Clock) - hodinový signál generovaný Master zařízením.
- **MOSI** (Master Output Slave Input) - data směrem Master → Slave.
- **MISO** (Master Input Slave Output) - data směrem Slave → Master.
- **SS** (Slave Select) - výběr Slave zařízením, logickou nulou od Master zařízením.

Komunikaci začne Master výběrem Slave zařízením pomocí SS (nastaví log. 0 na zařízením, se kterým chce komunikovat). Poté začne generovat hodinový signál na SCLK a v té chvíli vyšlou obě zařízením současně svoje data přes MOSI a MISO. Komunikace je aktivní, dokud je generován hodinový signál SCLK a dokud je nastaven SS na log. 0. Délka vyslaných dat je buď 8, nebo 16 bitů.

²³<https://learn.adafruit.com/using-the-bmp085-with-raspberry-pi/hooks-everything-up?view=all>



Obrázek 18: SPI sběrnice[33]

4.8.1 Obsluha SPI pomocí Raspberry Pi[34]

Připojení k SPI sběrnici je umožněno přes GPIO header, viz obrázek 3.:

- Pin #19 MOSI
- Pin #21 MISO
- Pin #23 SCLK
- Pin #24 CE0(SS0)
- Pin #26 CE1(SS1)

Abychom mohli začít využívat sběrnici SPI, musíme mít povolenu sběrnici v raspiconfig pro načtení modulu jádra SPI, viz kapitola 6.2.2.

K přístupu ke sběrnici SPI můžeme použít terminál a zapisovat/číst data přímo z `/dev/spidev`.

```
echo -ne "0x01" > /dev/spidev0.0
```

Tímto zapíšeme binární hodnotu 1 na sběrnici SPI.

Nebo můžeme využít knihovnu z projektu wiringpi²⁴ k sofistikovanějšímu přístupu ke sběrnici SPI. Výpis 6 nám ukazuje příklad přístupu v jazyce Python.

```
import wiringpi

SPIchannel = 1 #SPI kanál (CE1)
SPISpeed = 500000 #frekvence SCLK v ~Hz
wiringpi.wiringPiSetupGpio()
wiringpi.wiringPiSPISetup(SPIchannel, SPISpeed)

sendData = str(42) #odešleme 2 byte, 4 a ~2
recvData = wiringpi.wiringPiSPIDataRW(SPIchannel, sendData)
#recvData uchová list [NumOfBytes, recvDataStr] např: [2, '0x9A, 0xCD
']
```

Výpis 6: SPI.py

4.8.2 Připojení TFT displeje pomocí SPI

Rozhraní SPI můžeme využít v podstatě stejně jako rozhraní I2C k připojení různých čidel a převodníků. Toto jsme již realizovali v kapitole 4.7 právě pomocí sběrnice I2C. Proto pomocí SPI rozhraní budeme řešit připojení dotykového TFT displeje, na kterém můžeme realizovat grafický výstup celého OS, potažmo našich úloh.

Použili jsme 3.5" dotykový TFT LCD displej s rozlišením 320 x 480 px, přímo navržený pro RPi. Komunikace s jádrem je realizována prostřednictvím modulu jádra BCM2835. Na stránkách výrobce²⁵ lze stáhnout příslušný ovladač, pomocí kterého pohodlně zprovozníme displej pod OS Raspbian.

4.9 Obsluha RS232

V této úloze budeme realizovat čtení dat a ovládání RPi přes sériový port z PC. Později využijeme komunikaci přes sériový port pro komunikaci s modulem Arduino.

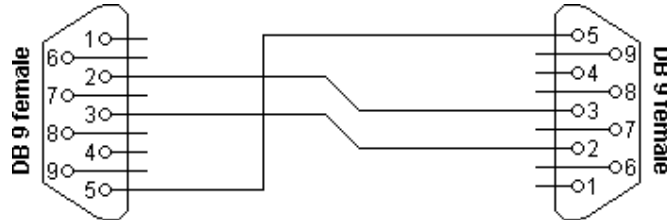
Nejprve musíme vyřešit rozdílné úrovně signálů sériového portu RPi a PC. Zatímco RPi sériový port má 3,3 V TTL signálovou logiku, PC pracuje s napětím v rozsahu +3 V až +15 V pro log. 1 a -3 V až -15 V pro log. 0 (signál TXD a RXD)[35]. K vyřešení rozdílných úrovní jsme použily obvod MAX232N²⁶ v typickém zapojení, viz obrázek 20.

²⁴<http://wiringpi.com/>

²⁵[http://www.waveshare.com/wiki/3.5inch_RPi_LCD_\(A\)](http://www.waveshare.com/wiki/3.5inch_RPi_LCD_(A))

²⁶<http://www.ti.com/lit/ds/symlink/max232.pdf>

K přímému propojení PC <-> RPi využíváme zapojení s kříženými signály TXD a RXD dle obrázku 19. Toto překřížení je realizováno na HW modulu, proto pro připojení používáme přímý kabel, nebo častěji pro novější PC bez sériového portu převodník USB <-> RS232.



Obrázek 19: RS232 zapojení[36]

Základní komunikaci budeme realizovat v terminálu OS GNU/Linux pomocí příkazů `echo` a `cat`.

Ukážeme si příklad využití sériové komunikace v jazyce Python. Modifikujeme úlohu dle kapitoly 4.5, kde budeme zprávy odesílat nejen na lokální terminál, ale i na sériový port, viz výpis 7.

```
import RPi.GPIO as GPIO
import serial
import time

tty=serial.Serial('/dev/serial0') #serial port 0
GPIO.setup(40,GPIO.IN)           #IN PIR sensor
GPIO.setup(38, GPIO.OUT)         #OUT LED

x=1
while True:
    i=GPIO.input(40)
    cas=time.strftime("%d.%m.%Y_%H:%M")
    if i==0 and x==1:             #PIR NORMAL
        print cas, "_NENI_ALARM"
        tty.write(cas+'_NENI_ALARM\n') #serial out
        x=0
        GPIO.output(38,1)         #LED OFF
```

Výpis 7: `pir.py`

4.10 Připojení modulu ARDUINO

V této úloze využijeme RPi jako nadřazený počítač pro získávání dat a ovládání modulu Arduina²⁷. Na Arduinu si vytvoříme program, konkrétně se jedná o úlohu řízení jasu LED diody pomocí PWM na základě nastavení potenciometru. Pomocí RPi budeme přes sériovou linku komunikovat s Arduinem a odečítat hodnotu napětí na potenciometru. Dále budeme mít možnost pomocí jednoduchých příkazů rozsvěcovat a zhasínat LED diodu.

Pro napsání programu využijeme oficiálního vývojového prostředí Arduino IDE²⁸ a jazyka Wiring²⁹. Syntaxe jazyka Wiring je podobná známému jazyku C. Program obsahuje dvě hlavní funkce, setup a loop. Funkce setup se provede pouze jednou na začátku programu a typicky obsahuje počáteční nastavení. Funkce loop se provádí opakovaně a je v ní umístěn hlavní programový kód.

Program, který použijeme pro komunikaci s RPi změří napětí potenciometru a hodnotu zapíše na sériový port. Dále čte data ze sériového portu. Pokud přečte posloupnost znaků „led0“, zablokuje výstup na LED diodu, v opačném případě pomocí PWM výstupu rozsvítí LED diodu v jasu odpovídajícímu nastavení potenciometru. Výpis 8 ukazuje řešení tohoto programu.

```
byte led = 6; //pin s LED diodou
byte pot = A0; //pin s připojeným potenciometrem
int val;
float napeti;
char znak;

void setup() {
  Serial.begin(9600);
}

void loop() {
  val = analogRead(pot); //cteni hodnoty na A0 a uprava rozsahu
  napeti = ((float)val/1024) * 5;
  Serial.print("Napětí_potenciometru:");
  Serial.print(napeti,4);
  Serial.println("_V");
  delay(500);
  if (test_led()){analogWrite(led, val/4);} //generovani PWM
```

²⁷Arduino [čti Arduíno] je jednodeskový počítač založený na mikrokontrolerech ATmega od firmy Atmel. Svým návrhem se snaží podpořit výuku informatiky ve školách a seznámit studenty s tím, jak jsou pomocí počítačů řízena různá zařízení (např. mikrovlnná trouba, automatická pračka a jiné stroje).[37]

²⁸<https://www.arduino.cc/en/Main/Software>

²⁹Wiring je programovací jazyk vytvořený pro programování mikrokontroléru bez specifických znalostí hardware. V současné době je nejznámější jako součást open-source platformy Arduino.[38]

```

else {analogWrite(led , 0);}
}

boolean test_led(){
char znak;
while (Serial.available() > 0) {
znak = Serial.read();
if (znak == 'l'){ znak = Serial.read();
if (znak == 'e'){ znak = Serial.read();
if (znak == 'd'){ znak = Serial.read();
if (znak == '0'){return false;
else { return true;}}}}}}
}
}

```

Výpis 8: *Arduino_PWM.ino*

4.11 Raspberry Pi jako web serveru

Abychom mohli RPi plnohodnotně využít k vizualizaci naměřených hodnot pomocí webových stránek musíme zprovoznit lokální webový server běžící na RPi. Konkrétně si nainstalujeme webový server Apache 2³⁰.

Apache 2 je webový server s otevřeným zdrojovým kódem běžící na většině operačních systémů a platformách. Je i součástí repozitářů OS Raspbian. Instalaci, včetně všech knihoven provádíme standardně pomocí příkazu `apt-get install`.

Kromě Apache 2 ještě budeme potřebovat zprovoznit podporu skriptovacího programovacího jazyka PHP, který využijeme k dynamickému načítání naměřených dat. PHP i potřebné knihovny k propojení s Apache serverem jsou také standardní součástí repozitářů.

4.12 Možnosti zpracování a vizualizace dat

4.12.1 Zpracování dat pro následnou vizualizaci

Pro zobrazení dat získaných z připojených čidel využijeme programy vytvořené v předchozích úlohách `DHT.py`, `BMP.py` a `TSL2651.py`. Vytvoříme si skript `mereni`, pomocí něhož budeme ukládat do textových souborů výsledky měření.

³⁰<https://httpd.apache.org/>

Jak je patrné z výpisu 9, do jednotlivých textových souborů se uloží pouze číselné hodnoty bez dalších znaků. Takto upravené výstupy programů později využijeme jako data dynamicky načítaná HTML stránkami.

```
#!/bin/bash

echo `date +"%A_%d.%B_%Y,_%H:%M" ` >/var/www/html/time.txt
cd /home/pi/Mereni/Adafruit_Python_DHT/examples
echo `./DHT.py 2302 19|head -n1|cut -c 6-20` C >/var/www/html/temp.txt
echo `./DHT.py 2302 19|head -n2|tail -n1|cut -c 11-20` % >/var/www/html/humidity.txt
echo `./DHT.py 2302 19|tail -n1|cut -c 11-20` C >/var/www/html/dewpoint.txt
cd /home/pi/Mereni/Adafruit_Python_BMP/examples
echo `./BMP.py|tail -n1|cut -c 21-30` >/var/www/html/press.txt
cd /home/pi/Mereni/
echo `./tsl2561.py|head -n1|cut -c 30-40` >/var/www/html/sun.txt
```

Výpis 9: mereni

Abychom zajistili pravidelné aktualizování naměřených údajů využijeme službu Cron³¹, která bude pravidelně spouštět skript `mereni` s intervalem jedné minuty.

4.12.2 Vizualizace dat pomocí webového rozhraní

Pro vizualizaci naměřených hodnot využijeme nainstalovaného webového serveru, který jsme si zprovoznili v úloze dle kapitoly 4.11. Na tomto serveru si zprovozníme webovou stránku, která bude sloužit k zobrazení naměřených údajů z čidel a jejich prezentaci v grafické podobě pomocí vzdáleného síťového připojení.

Pro grafické zpracování stránek je využito technologií HTML5 a CSS. Kompletní zdrojová data jsou součástí příloženého CD, viz příloha 7.

Dynamické načítání měřených dat, které máme připravené v příslušných textových souborech je realizováno v jazyce PHP pomocí funkce `readfile` a obnovování (refresh) webové stránky pomocí tagu `<meta http-equiv="refresh" content="60">`.

³¹Cron je softwarový démon, který v operačních systémech automatizovaně spouští v určitý čas nějaký příkaz resp. proces (skript, program apod.). Jedná se vlastně o specializovaný systémový proces, který v operačním systému slouží jakožto plánovač úloh, jenž umožňuje opakované spouštění periodicky se opakujících procesů (např. noční běhy dávkových úloh při hromadném zpracování dat) apod.[39]

Příklad načítání dat z textových souborů můžeme vidět ve výpisu 10.

```

<br>
Aktuální teplota:
<span class="valuered"><?php readfile("temp.txt");?></span>
<br style="clear:_both">
```

Výpis 10: index.php

Po nakopírování stránek do RPi (adresář /var/www) by měly být stránky přístupné na adrese přiřazené RPi (192.168.2.210) do webového prohlížeče.

4.12.3 Vizualizace dat pomocí displeje HW modulu

Jedná se v podstatě také o zobrazení hodnot pomocí webové stránky. Použijeme již vytvořenou stránku v kapitole 4.12.2. Přizpůsobíme stránku pomocí responzivního web designu³² pro nízké rozlišení 3,5" displeje, viz výpis 11

```
<!--screen.css pro rozlišení > 1024px
      mobile.css pro rozlišení < 1024px-->
<link rel="stylesheet" type="text/css" media="screen_and_(min-width:
  _1025px)" href="screen.css" />
<link rel="stylesheet" type="text/css" media="screen_and_(max-width:
  _1024px)" href="mobile.css" />
```

Výpis 11: index.php

Pro zobrazení připravené stránky použijeme speciální režim prohlížeče chromium³³, který zobrazuje pouze obsah webové stránky v celoobrazovkovém módu, tzv. „Kiosk screen mode“.

³²Responzivní web design (anglicky Responsive web design) je pojem, se kterým přišel americký programátor Ethan Marcotte ve stejnojmenném článku na blogu A LIST Apart.[1][2][3] Jedná se o způsob stylování HTML dokumentu, které zaručí, že zobrazení stránky bude optimalizováno pro všechny druhy nejrůznějších zařízení (mobily, notebooky, netbooky, tablety atd.).[40]

³³Prohlížeč chromium je výchozí prohlížeč OS Raspbian

Aby se nám po startu RPi rovnou zobrazovala tato stránka, upravíme spouštěcí soubory grafického prostředí OS Raspbian, konkrétně soubor `~/.config/lxsession/LXDE-pi/autostart` do následující podoby:

```
@lxpanel --profile LXDE-pi
@pcmanfm --desktop --profile LXDE-pi
#@xscreensaver -no-splash
@xset s off
@xset -dpms
@xset s noblank

@sed -i 's/"exited_cleanly": false/"exited_cleanly" true/'
~/.config/chromium/Default/Preferences
@chromium-browser --noerrdialogs --kiosk http://192.168.2.210 --
incognito

@point-rpi
```

Po restartu RPi se spustí grafické prostředí přímo do prohlížeče chromium a zobrazí se obsah naší webové stránky.

5 Návrh a tvorba přídatných HW modulů

K realizaci souboru úloh budeme potřebovat prakticky zapojovat elektronické součástky a moduly na GPIO rozhraní RPi. Toto můžeme realizovat nejjednodušeji pomocí nepájivého pole. Realizace pomocí nepájivého pole má výhodu v univerzálnosti použití, kde nám nabízí v podstatě neomezené modifikace zapojení. Avšak klade velký důraz na přesnost zapojení. V případě jednoduchých, ne příliš rozsáhlých úloh jsme schopni dosáhnout správného zapojení. Pokud ale chceme realizovat složitější úlohy, nebo kombinaci těchto úloh nastává problém s nepřehledností a možnou generací chyb. V případě připojení obvodů bez jakékoliv ochrany na vstupy GPIO, které jsou přímo napojeny na sběrnice ARM obvodů, by nesprávné zapojení mohlo mít fatální následky, s velkou pravděpodobností zničení celého RPi. Dále bychom museli u každé úlohy dodávat podrobné schéma zapojení a zatěžovat se časově náročným sestavováním obvodů.

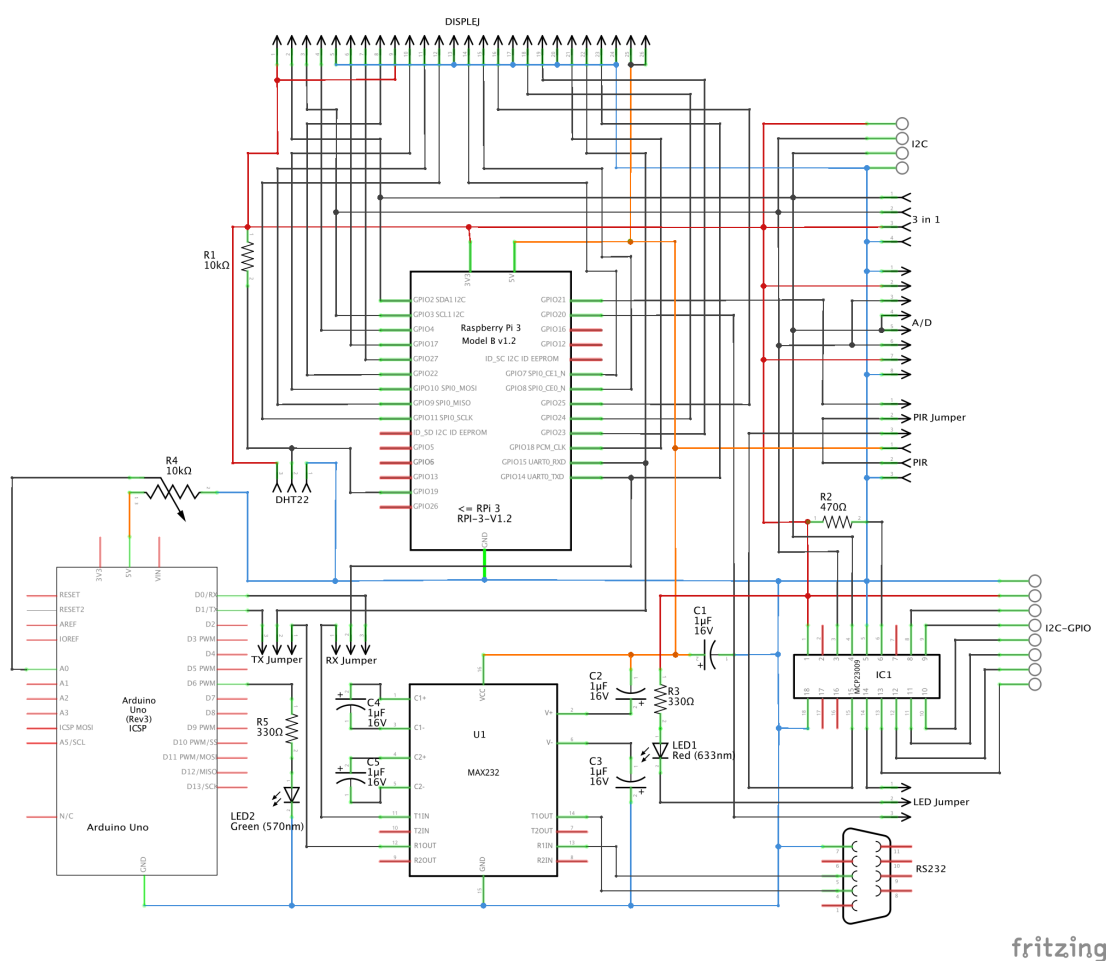
Z výše uvedených důvodů se řešení pomocí nepájivého pole jeví pro naše aplikace jako nevhodné. Z našeho pohledu se jeví jako vhodnější alternativa vývoj vlastního HW modulu zaměřeného speciálně na námi navržené úlohy. Tím sice ztratíme jistou univerzálnost a modifikovatelnost zapojení, ale naproti tomu získáme jistotu správného zapojení a vyloučíme možné chyby a následné zničení obvodů. Pokud bude třeba provádět konfigurace HW modulu, budou řešeny pomocí propojek, kde nebude možné žádnou variantou zapojení obvodu zničit.

5.1 Návrh zapojení HW modulu

Teoretický návrh zapojení HW modulu vychází z kapitoly 4. Vlastní návrh zapojení vznikl sloučením zapojení jednotlivých úloh. Každá úloha byla prakticky realizována a odzkoušena na nepájivém poli. Poté byly zapojeny všechny úlohy najednou a optimalizováno zapojení. Z tohoto zapojení vznikl podklad pro návrh HW modulu, jak ho vidíme na obrázku 20. Konkrétně byla pro návrh HW modulu použita open-source aplikace Fritzing.

Aplikace Fritzing[41] vznikla jako univerzitní projekt na Fachhochschule Potsdam University of Applied Sciences, nyní je vyvíjena Friedens-of-Fritzing foundation. Filozofie aplikace je usnadnit přenesení obvodu ze skutečného prototypu do schématu a dále možnost vytvoření desky plošného spoje. Obvod je po sestavení a odzkoušení na nepájivém poli překreslen do virtuálního modelu zapojení v pohledu nepájivého pole podle skutečného prototypu. Tento model je přenesen do pohledu elektronického schématu a následně je možno vytvořit na základě schématu desku plošných spojů. Aplikace umožňuje „automatické routování“, kdy

program automaticky navrhne optimální řešení návrhu desky plošných spojů. Aplikace je primárně navržena pro práci s moduly Arduino a Raspberry. Kromě toho knihovna programu obsahuje většinu běžně užívaných součástek. Pokud máme požadavky na další součástky, nebo moduly, můžeme si je sami vytvořit ve vestavěném editoru součástek. Fritzing není tak komplexní jako profesionální CAD nástroje, ale je plně dostačující pro práci v univerzitním nebo domácím prostředí. Aplikace má na rozdíl od profesionálních nástrojů intuitivní a jednoduché ovládání, ovšem na úkor pokročilejších funkcí.



Obrázek 20: HW modul - schéma

5.2 Návrh desky plošných spojů HW modulu

Návrh desky plošných spojů je realizován jako oboustranný plošný spoj osazený klasickými součástkami. Osazení součástkami SMD, vzhledem k velikosti modulu dané rozměry přídatných modulů, je neopodstatněné a pro prototypovou výrobu v domácích podmínkách nepřináší žádné výhody. Deska plošných spojů je taktéž navržena v aplikaci Fritzing, vychází z výše navrženého schématu. Bude opatřena z vrchní strany potiskem pro snadnější orientaci.

Napájení celého modulu je řešeno napájecím zdrojem 5 V min. 2 A, který primárně napájí RPi. Napětí je následně rozvedeno přes GPIO konektor RPi na celou desku HW modulu. Napájení modulu Arduino UNO je řešeno z konektoru RPi USB typ A, propojovacím kabelem na konektor micro USB modulu Arduino UNO. Toto řešení bylo zvoleno zejména proto, že výrobce nedoporučuje napájet Arduino UNO přímo přes pin 5 V z důvodu absence jakýkoliv ochrany[42] a vyšší napětí pro napájení přes Vin pin nemáme k dispozici. Dalším přínosem tohoto řešení je možnost použít propojovací kabel k připojení PC a programování Arduina bez nutnosti vyjmutí z HW modulu.

HW modul je koncipován modulárně, tzn. hlavní části lze demontovat bez nutnosti pájení a použít je v jiné aplikaci, nebo zaměnit za kompatibilní moduly.

To se týká modulů:

- RPi
- Arduino UNO
- TFT displeje
- AD převodníku
- sdruženého čidla 4 v 1
- PIR senzoru

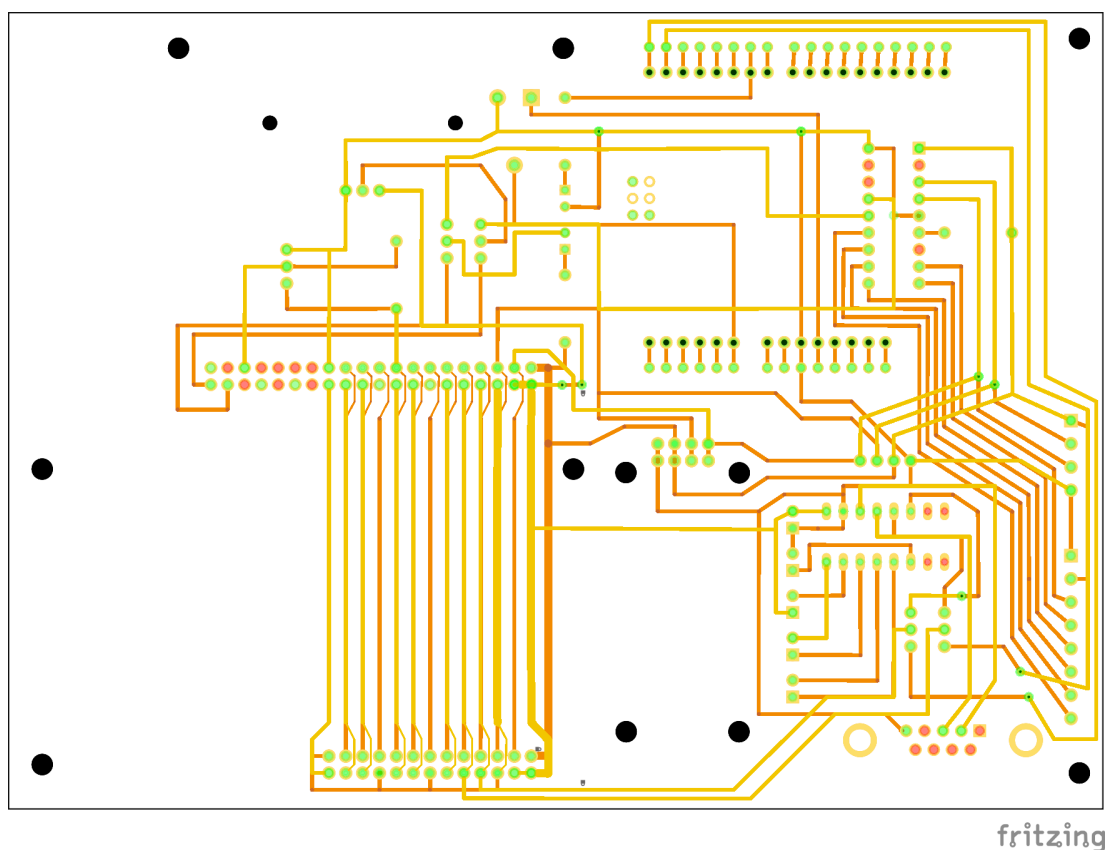
Ostatní součástky jsou pevně osazené.

Volitelné konfigurace jsou realizovány pomocí „jumperů“. Zapojení je koncipováno tak, že nelze udělat nepřipustnou kombinaci a tím poškodit HW obvody.

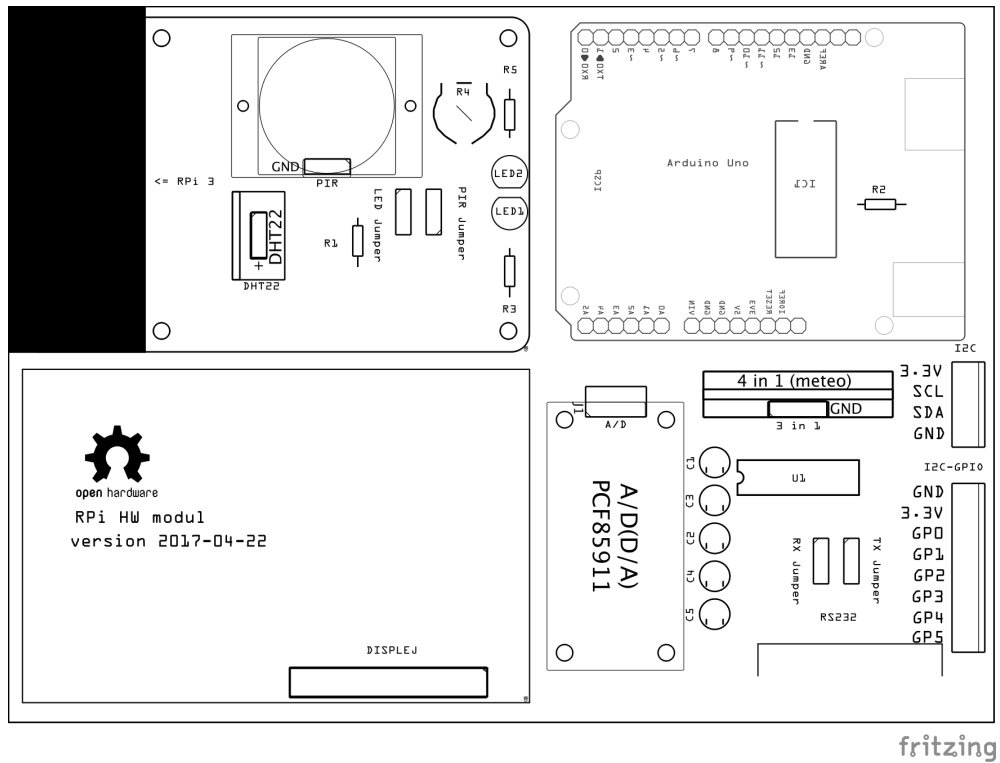
Pomocí „jumperů“ můžeme nastavit tyto možnosti:

- **PIR Jumper** - volba připojení čidla pohybu přímo na RPi pin GPIO21, nebo přes I2C budič MCP23009.
- **LED Jumper** - volba připojení nízkopříkonové LED přímo na RPi pin GPIO20, nebo přes I2C budič MCP23009.
- **TX Jumper** - volba směrování vysílací linky sériového portu přes převodník MAX232 na konektor RS232, nebo na pin RX Arduina.
- **RX Jumper** - volba směrování přijímací linky sériového portu přes převodník MAX232 na konektor RS232, nebo na pin TX Arduina.

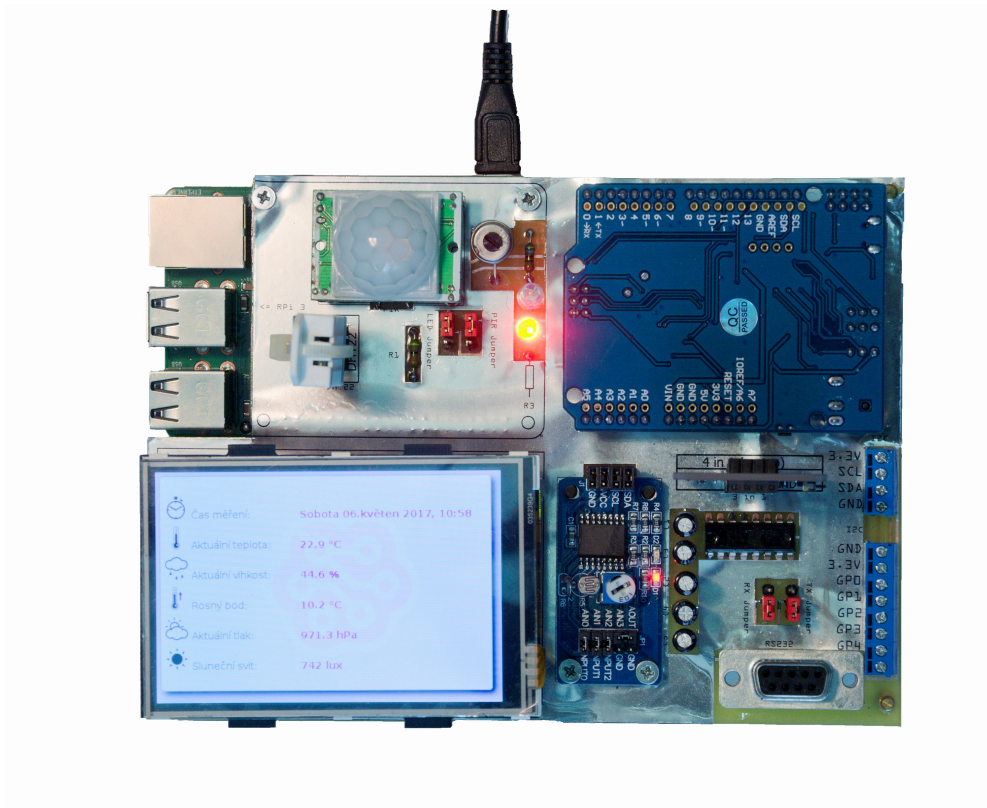
Navržený plošný spoj HW modulu a osazení součástek je patrné z obrázků 21 a 22.



Obrázek 21: HW modul - PCB



Obrázek 22: HW modul - osazení



Obrázek 23: HW modul - praktická realizace

6 Praktické řešení vzorových úloh

Máme navržen soubor úloh pro praktickou ukázkou realizace měření různých fyzikálních veličin. Pro tento účel jsme navrhli HW modul usnadňující praktickou realizaci těchto úloh bez podrobných znalostí obvodových řešení jednotlivých obvodů. Nyní budeme postupně, prakticky realizovat jednotlivé úlohy, jak jsme si je navrhli v kapitole 4. V kapitole 4 je podrobně teoreticky popsáno, co je cílem jednotlivých úloh a jakým technickým řešením k tomuto cíli dospějeme. V této kapitole si prakticky demonstrujeme řešení těchto úloh. Úlohy budou koncipovány jako laboratorní měření pro žáky, proto zde bude stručnou a přesnou formou formulován postup, jak dosáhnout cíle dané úlohy počínaje přípravou, seznamem potřebného vybavení, přes vlastní postup zprovoznění, až k ověření předpokládaného výstupu úlohy.

K realizaci praktických úloh budeme potřebovat následující HW vybavení:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.³⁴
- Čtečka karet microSD.
- USB klávesnici a myš.
- Monitor s HDMI portem kompatibilní s Raspberry Pi 3 Model B.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.^{35 36}
- Arduino UNO s výstupní úrovní 3,3 V kompatibilní.
- HW modul.
- TFT displej 3,5".
- Sériový kabel RS232, popřípadě převodník USB<=>RS232.

U jednotlivých úloh bude vždy uvedeno, které vybavení budeme přesně potřebovat. V případě potřeby specifické konfigurace, bude vždy tato operace u dané úlohy přesně definována.

Všechny odkazované soubory a programy, které budeme dále využívat, jsou součástí příloženého CD.

³⁴Konkrétně byly úlohy realizovány s verzí Windows 10 Pro X64 - verze 1607 a Debian GNU/Linux 8 (jessie) 64bit.

³⁵Napájecí zdroj +5 V min. 2 A s konektorem MicroUSB.

³⁶RPi s dostupným připojením do stejné sítě jako PC a dostupným připojením na internet přes Ethernet nebo WiFi.

6.1 Instalace OS a základní konfigurace Raspberry Pi

Postup instalace demonstrujeme na instalaci OS **Raspbian**, ale obecně lze použít pro jakýkoliv jiný OS vytvořený pro RPi.

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Čtečka karet microSD.
- USB klávesnici a myš.
- Monitor s HDMI portem kompatibilní s Raspberry Pi 3 Model B.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.

6.1.1 Instalace pomocí NOOBS[43]

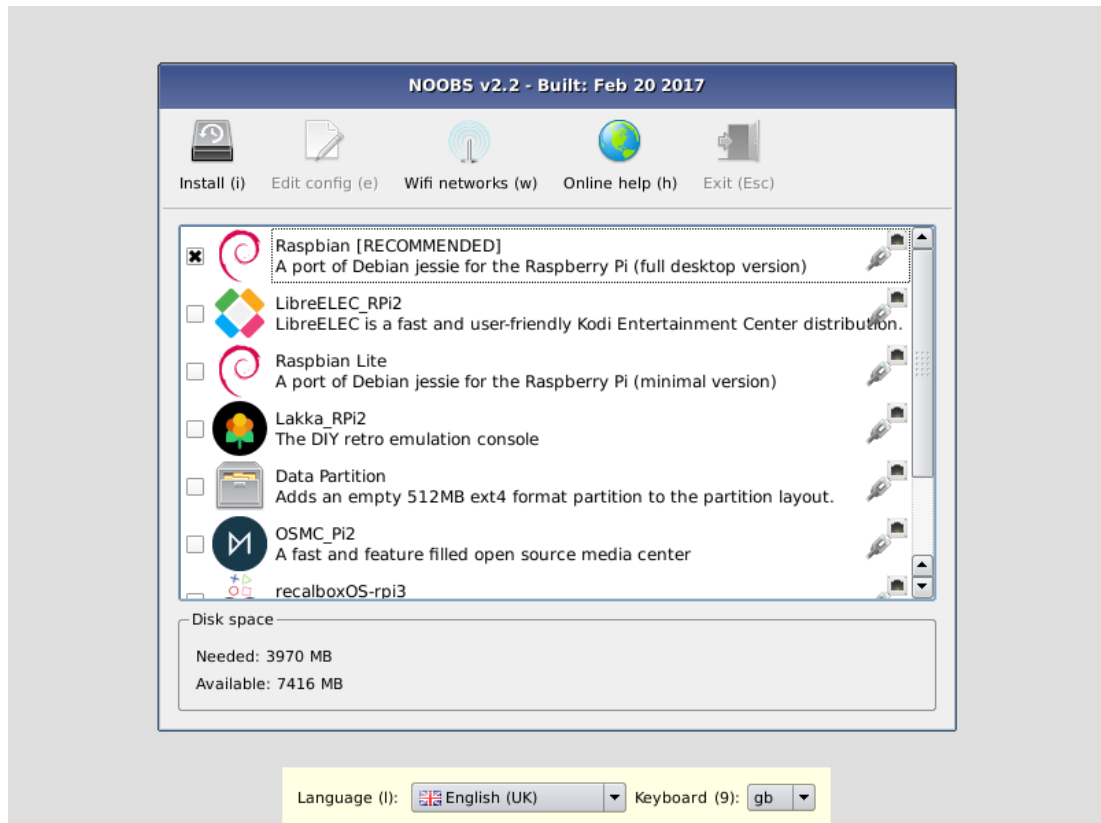
Stáhneme aktuální verzi instalátoru NOOBS ze stránek <https://www.raspberrypi.org/downloads/noobs/>.

Vložíme SD kartu do čtečky. Pokud karta není naformátována, nebo má jiný formát než FAT32, naformátujeme kartu tak, aby obsahovala pouze jednu FAT32 partition³⁷.

Rozbalíme zip archiv a nakopírujeme na SD kartu. Bezpečně odebereme kartu a vložíme do slotu RPi.

Připojíme k RPi monitor, klávesnici, myš a zapneme RPi. RPi nabojuje do NOOBS, kde zvolíme OS k instalaci (v našem případě Raspbian), viz obrázek 24. Poté NOOBS automaticky nainstaluje zvolený OS a spustí vybraný systém. Další spouštění už probíhají přímo do nainstalovaného OS.

³⁷Doporučený program na formátování SD karty je pro Windows „SD Formatter“ https://www.sdcard.org/downloads/formatter_4/index.html a pro GNU/Linux GParted (standardně součástí Debian repozitářů)



Obrázek 24: NOOBS instalátor

6.1.2 Instalace obrazu OS pomocí MS Windows[44]

Stáhneme aktuální verzi Raspbian³⁸ ze stránek <https://www.raspberrypi.org/downloads/>.

Stáhneme aktuální verzi Win32DiskImager ze stránek <https://sourceforge.net/projects/win32diskimager/>. Nainstalujeme do počítače, případně použijeme „portable verzi“.

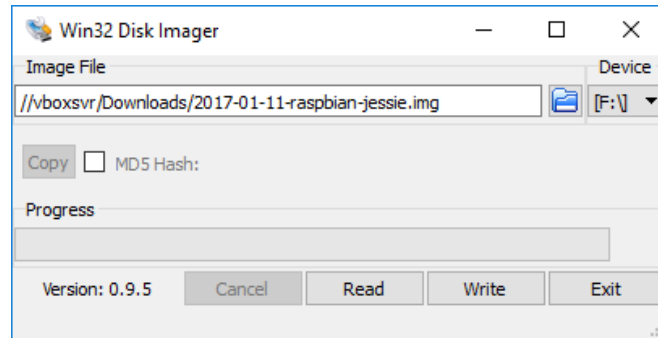
Ve Win32DiskImager zvolíme soubor s obrazem OS Raspbian, viz obrázek 25. Vložíme SD kartu do čtečky a vybereme tento disk ve Win32DiskImager jako cílové zařízení pro instalaci³⁹ a zapíšeme obraz na kartu.

Zavřeme Win32DiskImager, bezpečně odebereme kartu a vložíme ji do slotu RPi.

³⁸verze s okenním manažerem PIXEL(v dalších úlohách budeme pracovat s grafickým výstupem).

³⁹Při výběru zařízení musíme zvolit správnou jednotku, všechna data na kartě budou přepsána!

Připojíme k RPi monitor, klávesnici, myš a zapneme RPi. Po zapnutí RPi se automaticky spustí nainstalovaný Raspbian.



Obrázek 25: Win32DiskImager

6.1.3 Instalace obrazu OS v GNU/Linux[45]

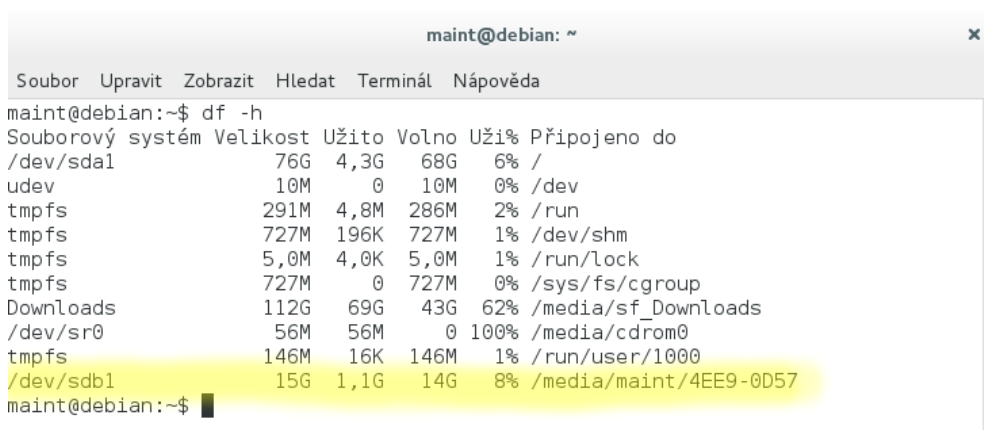
Stáhneme aktuální verzi Raspbian⁴⁰ ze stránek <https://www.raspberrypi.org/downloads/>.

Vložíme SD kartu do čtečky.

Spustíme terminál a zadáme příkaz pro zobrazení připojených disků:

```
df -h
```

V seznamu disků najdeme SD kartu. Pravděpodobně bude na konci seznamu. V pravém sloupci bude `/media/maint/XXXX-XXXX` a v levém sloupci `/dev/sdX1`. X bude v našem případě `b`, viz obrázek 26. Potom jednotka bude `/dev/sdb1`.



Obrázek 26: Terminál, výběr disku

⁴⁰verze s okenním manažerem PIXEL (v dalších úlohách budeme pracovat s grafickým výstupem).

Přihlásíme se jako root a odpojíme disk:

```
su
umount /dev/sdb1
```

Zapíšeme obraz OS na disk:

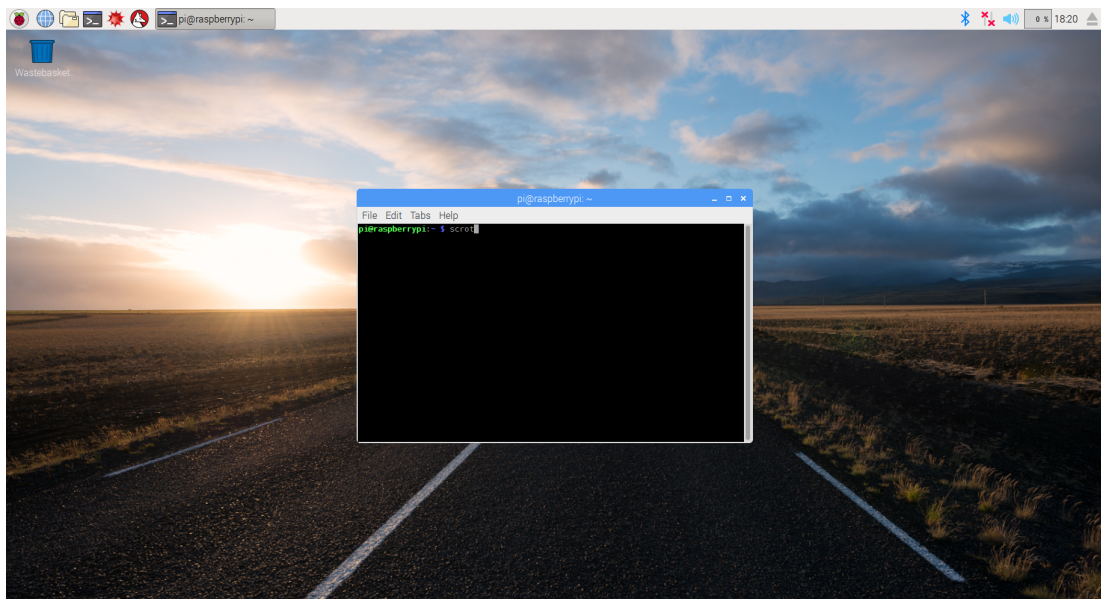
```
dd bs=4M if=2017-01-11-raspbian-jessie.img of=/dev/sdb
sync
```

Vyjmeme SD kartu ze čtečky a vložíme do slotu RPi.

Připojíme k RPi monitor, klávesnici, myš a zapneme. Po zapnutí se automaticky spustí nainstalovaný Raspbian.

6.1.4 První spuštění

Po zapnutí by nám měl Raspbian nastartovat automaticky do grafického prostředí PIXEL, viz obrázek 27.



Obrázek 27: grafické prostředí PIXEL

Jako první zprovozníme síťové připojení, standardně se nastavení nachází v horní liště u hodin. Můžeme použít jak připojení pomocí Ethernetového kabelu, tak WiFi připojení. Nastavíme síťové připojení a vyzkoušíme funkčnost.

Nyní můžeme nainstalovat dostupné aktualizace. Spustíme terminál a zadáme příkazy pro aktualizaci systému:

```
sudo apt-get update
sudo apt-get upgrade
```

Po dokončení je vhodné provést restart RPi.

Pro další práci je vhodné nainstalovat souborový manažer Midnight Commander:

```
sudo apt-get install mc
```

6.2 Konfigurace systému pomocí „raspi-config“

K realizaci této úlohy budeme potřebovat:

- USB klávesnici a myš.
- Monitor s HDMI portem kompatibilní s Raspberry Pi 3 Model B.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.

K nastavení systému můžeme použít dva nástroje. Terminálovou aplikaci `raspi-config` nebo grafický nástroj `Raspberry Pi Configuration`. Oba nástroje jsou rovnocenné a lze pomocí nich nastavit stejné vlastnosti. Podrobný popis vlastností je popsán v kapitole 4.2.

`Raspi-config` spustíme z terminálu příkazem⁴¹:

```
sudo raspi-config
```

Program `Raspberry Pi Configuration` spustíme z menu nabídky `start`:

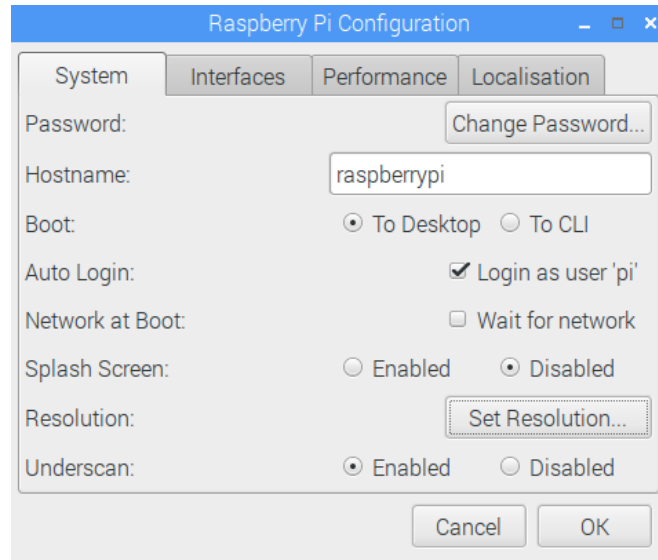
```
Preferences -> Raspberry Pi Configuration
```

Dále si popíšeme nastavení jednotlivých vlastností potřebných pro další úlohy pomocí grafického nástroje `Raspberry Pi Configuration`.

⁴¹`raspi-config` můžeme využít pokud nemáme k dispozici grafické rozhraní, například používáme-li vzdálené připojení přes SSH

6.2.1 System

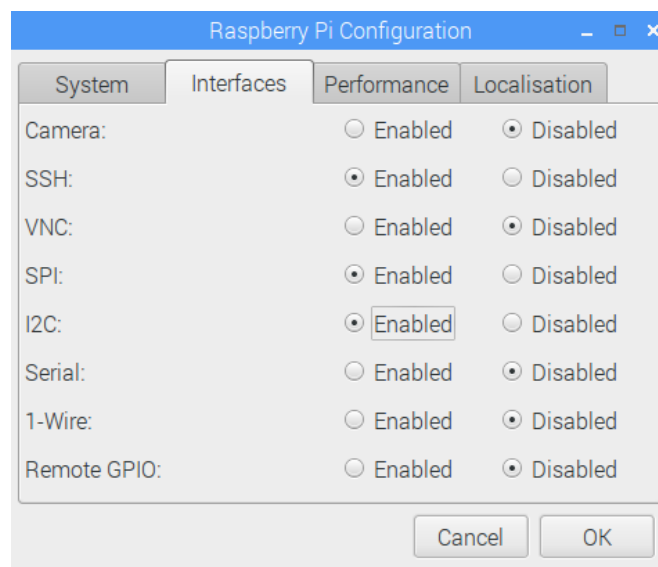
Nastavíme dle obrázku 28. Případně můžeme změnit výchozí údaje názvu zařízení a hesla⁴².



Obrázek 28: Raspberry Pi Configuration - System

6.2.2 Interfaces

Nastavíme dle obrázku 29.

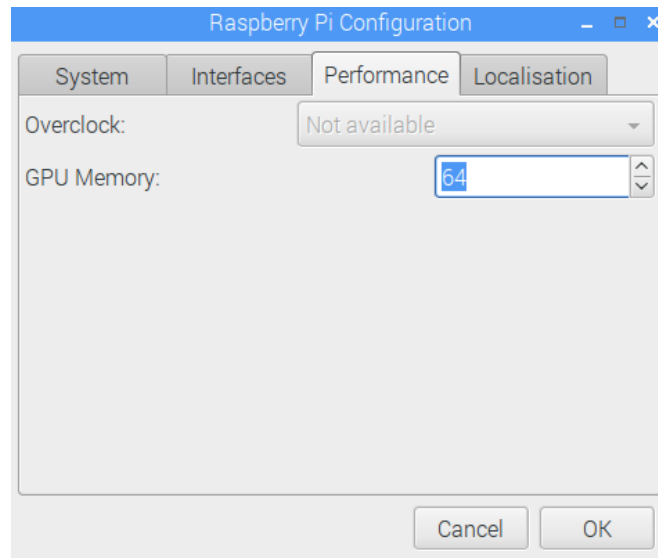


Obrázek 29: Raspberry Pi Configuration - Interfaces

⁴²V případě změny přihlašovacích údajů, musíme tyto údaje používat v dalších úlohách, kde je v postupu použito výchozích hodnot.

6.2.3 Performance

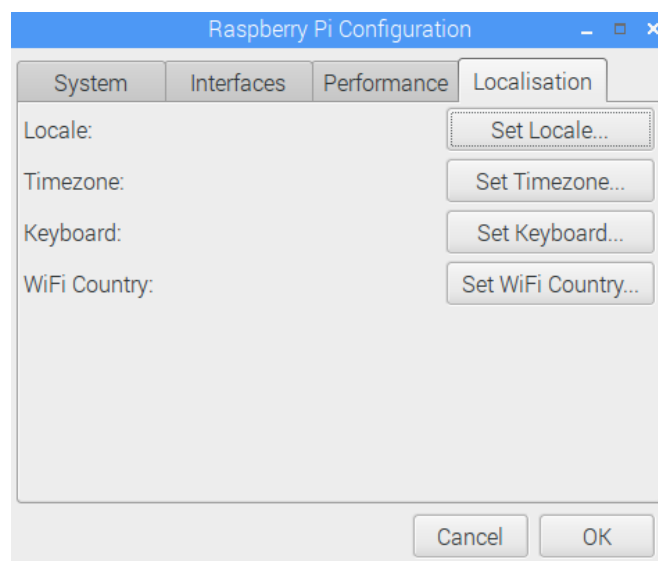
Nastavíme dle obrázku 30.



Obrázek 30: Raspberry Pi Configuration - Performance

6.2.4 Localisation

Nastavení národního prostředí, jako je jazyk, časová zóna, rozložení klávesnice. Můžeme ponechat ve výchozím anglickém nastavení, nebo pro lepší komfort ovládní změnit dle osobních preferencí. Pokud změníme konfiguraci národního prostředí, změna se projeví až po restartu RPi.



Obrázek 31: Raspberry Pi Configuration - Localisation

6.3 Obsluha ETHERNET

V této úloze se budeme věnovat primárně zprovoznění vzdáleného připojení pomocí SSH. Pro jednodušší přístup RPi přiřadíme statickou IP adresu.

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- USB klávesnici a myš.
- Monitor s HDMI portem kompatibilní s Raspberry Pi 3 Model B.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.

6.3.1 Nastavení statické IP adresy

Máme funkční připojení k síti, viz kapitola 6.1.4.

Máme povolené SSH připojení, viz kapitola 6.2.2.

Spustíme terminál a zjistíme nastavení sítě pomocí příkazu:

```
ifconfig
```

Z obrázku 32 vidíme, že používáme WiFi připojení `wlan0` s přidělenou adresou DHCP `192.168.2.210`. Tuto adresu můžeme použít jako statickou adresu, kterou přidělíme RPi, případně použijeme jinou adresu v metrice odpovídající dané síti.

```
pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:5a:97:f1
          inet6-adr: fe80::f02f:727f:7756:a645/64  Rozsah:Linka
          AKTIVOVÁNO VŠESMĚROVÉ_VYSÍLÁNÍ MULTICAST  MTU:1500  Metrika:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          kolizi:0 délka odchozí fronty:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Místní smyčka
          inet adr:127.0.0.1  Maska:255.0.0.0
          inet6-adr: ::1/128  Rozsah:Počítač
          AKTIVOVÁNO SMYČKA BĚŽÍ  MTU:65536  Metrika:1
          RX packets:200 errors:0 dropped:0 overruns:0 frame:0
          TX packets:200 errors:0 dropped:0 overruns:0 carrier:0
          kolizi:0 délka odchozí fronty:1
          RX bytes:16656 (16.2 KiB)  TX bytes:16656 (16.2 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:0f:c2:a4
          inet adr:192.168.2.210  Všeměr:192.168.2.255  Maska:255.255.255.0
          inet6-adr: fe80::f528:7c21:e0af:6d9d/64  Rozsah:Linka
          AKTIVOVÁNO VŠESMĚROVÉ_VYSÍLÁNÍ BĚŽÍ MULTICAST  MTU:1500  Metrika:1
          RX packets:173 errors:0 dropped:91 overruns:0 frame:0
          TX packets:122 errors:0 dropped:0 overruns:0 carrier:0
          kolizi:0 délka odchozí fronty:1000
          RX bytes:34113 (33.3 KiB)  TX bytes:17234 (16.8 KiB)
```

Obrázek 32: Raspberry Pi - ifconfig

Nastavíme statickou IP adresou editací souboru `/etc/network/interfaces` příkazem:

```
sudo nano /etc/network/interfaces
```

V souboru najdeme konfiguraci námi používaného rozhraní, v našem případě `wlan0`:

```
allow-hotplug wlan0
iface wlan0 inet manual
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Konfiguraci upravíme následovně:

```
allow-hotplug wlan0
iface wlan0 inet static
address 192.168.2.210
netmask 255.255.255.0
gateway 192.168.2.1
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Provedeme restart síťového rozhraní příkazy:

```
sudo ifdown wlan0
sudo ifup wlan0
```

Po restartu by mělo být funkční síťové připojení s námi zadanou IP adresou.

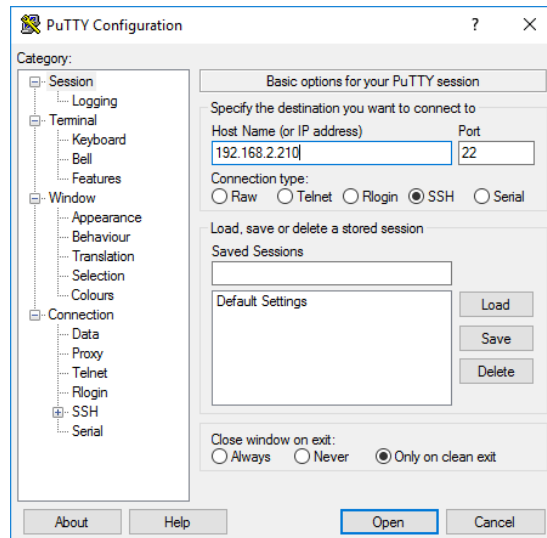
6.3.2 Připojení pomocí SSH

V tuto chvíli již nepotřebujeme monitor ani klávesnici s myší připojené k RPi. Veškeré operace budeme provádět přes vzdálený PC.

Tento postup je primárně určený pro připojení v MS Windows, kde musíme nainstalovat externí aplikaci pro připojení přes SSH. Použijeme SSH a Telnet klienta PuTTY.

Aplikaci stáhneme ze stránek <http://www.putty.org/>, nainstalujeme do počítače, případně použijeme „portable verzi“.

Spustíme PuTTY a nastavíme připojení dle obrázku 33.



Obrázek 33: PuTTY - konfigurace

Otevře se okno terminálu s výzvou k přihlášení, provedeme přihlášení:

```
login as:pi
pi@192.168.2.210's password:raspberry
```

V GNU/Linux nemusíme instalovat žádný program, pouze spustíme terminál a připojíme se k RPi příkazem:

```
ssh pi@192.168.2.210
```

a zadáním hesla:

```
pi@192.168.2.210's password:raspberry
```

Nyní jsme schopni ovládat RPi vzdáleně přes SSH připojení.

6.3.3 Nakopírování souborů do Raspberry Pi

Pro další úlohy budeme využívat před připravené programy a skripty. Překopírujeme si tyto soubory z příloženého CD (příloha 6) do RPi pomocí příkazu `scp`⁴³.

Z lokálního počítače nakopírujeme adresář `Mereni` do `~/Mereni` pomocí příkazu:

```
scp -r Mereni pi@192.168.2.210:
```

Připojíme se přes SSH, viz kapitola 6.3.2.

Provedeme změnu práv souborů, aby je bylo možno spouštět:

```
chmod -r +x Mereni
```

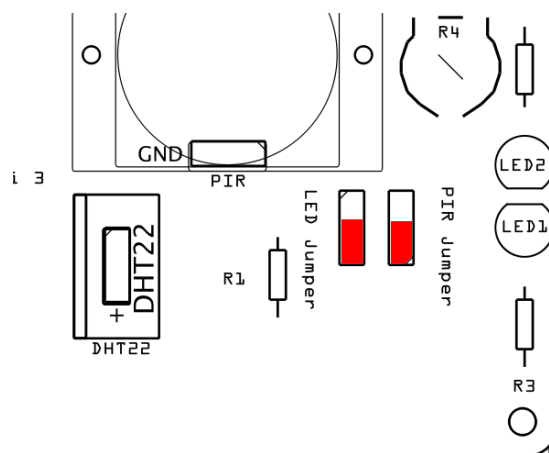
⁴³`scp` je součástí instalace GNU/Linux. V MS Windows použijeme identicky program `pscp`, který je součástí instalace programu `putty`, viz kapitola 6.3.2.

6.4 Základní využití GPIO - vstup

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.

Připojíme RPi do HW modulu a nakonfigurujeme dle obrázku 34.



Obrázek 34: GPIO/I2C konfigurace

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Nastavíme GPIO21 jako vstup:

```
echo "21" > /sys/class/gpio/export
echo "in" > /sys/class/gpio/gpio21/direction
```

Přečteme hodnotu na vstupu:

```
cat /sys/class/gpio/gpio21/value
```

Pokud bude senzor nezakrytý, bude na výstupu 0, když senzor zakryjeme a příkaz zopakujeme, měla by na výstupu být 1.

Po ukončení práce uvolníme GPIO příkazem:

```
echo "21" > /sys/class/gpio/unexport
```

Toto je nejjednodušší přístup k hodnotám z GPIO. Pro pokročilejší užití můžeme využít přístup přes knihovny programovacích jazyků C++ nebo Python.

V další úloze si popíšeme práci s GPIO jako vstupem a kombinací obou úloh si demonstrujeme jednoduchý program snímající stav PIR čidla a detekci alarmu rozblíkním LED a výstupem upozornění na terminál.

6.5 Základní využití GPIO – výstup

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.

Připojíme RPi do HW modulu a nakonfigurujeme HW modul dle obrázku 34.

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Nastavíme GPIO20 jako výstup:

```
echo "20" > /sys/class/gpio/export  
echo "out" > /sys/class/gpio/gpio20/direction
```

Zapišeme hodnotu na výstup:

```
echo 1 > /sys/class/gpio/gpio20/value
```

Ve výchozím stavu LED1 svítí (LED zapojena mezi +5 V a GPIO20). Pokud zapišeme 1, LED1 zhasne, při zápisu 0 se LED1 rozsvítí.

Po ukončení práce uvolníme GPIO příkazem:

```
echo "20" > /sys/class/gpio/unexport
```

Pro pokročilejší užití je vhodnější k ovládní GPIO využít přístup přes knihovny programovacích jazyků C++ nebo Python. Nyní realizujeme ovládní LED1 v jazyce Python. Na základě aktivace PIR senzoru, dojde k rozblíkní LED1 a výstupu upozornění na terminál. Program spustíme příkazem:

```
~/Mereni/pir.py
```

Pokud zakryjeme PIR senzor, dojde k aktivaci alarmu a rozblíká se LED1 s výstupem na terminál „ALARM AKTIVACE“. Po odeznění LED1 nesvítí s výstupem na terminál „NENI ALARM“.

6.6 Programová obsluha periférií

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.

Na HW modul připojíme modul „DHT22”, viz obrázek 22.

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Pro měření využijeme sdružené čidlo teploty a vlhkosti DHT22. Nejdříve si nainstalujeme potřebné ovladače:

```
sudo apt-get install build-essential python-dev
cd ~/Merení/
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo python setup.py install
```

K vlastnímu měření si upravíme vzorový program `simpletest.py` (viz výpis 2)⁴⁴ z tohoto projektu a spustíme:

```
~/Merení/Adafruit_Python_DHT/examples/DHT.py 2302 19
```

Program nám poskytuje hodnoty:

- Teplota
- Vlhkost vzduchu
- Rosný bod

⁴⁴Vytvořený program `dht.py` je součástí příloženého CD, viz příloha 3.

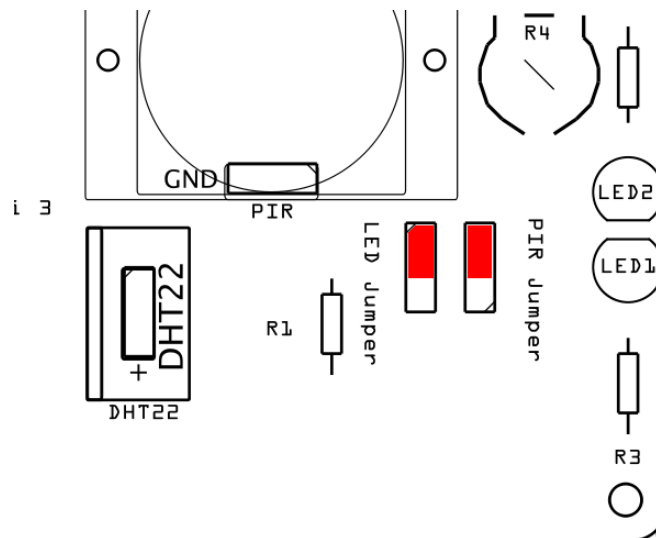
6.7 Využití I2C sběrnice

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.

Máme povolenu I2C sběrnici v raspi-config pro načtení modulu jádra I2C, viz kapitola 6.2.2.

Připojíme RPi do HW modulu a nakonfigurujeme HW modul dle obrázku 35.



Obrázek 35: GPIO/I2C konfigurace

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2 a detekujeme senzory připojené na I2C sběrnici:

```
i2cdetect -y 1
```

Pokud máme připojeny všechny moduly, měli bychom detekovat senzory s adresou 20, 39, 48 a 77 viz obrázek 36

```

[pi@raspberrypi:~ $ i2cdetect -y 1
     0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20: 20  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  -- 39  --  --  --  --  --  --
40:  --  --  --  --  --  --  -- 48  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  -- 77  --  --  --  --  --  --  --  --
pi@raspberrypi:~ $ █

```

Obrázek 36: I2C - `i2cdetect`

Tímto máme splněny výchozí podmínky pro další úlohy a můžeme přistoupit k jejich realizaci.

6.7.1 Ovládání digitálních vstupů a výstupů pomocí sběrnice I2C

Budeme realizovat úlohy dle kapitol 6.4 a 6.5 s využitím obvodu MCP23009 (8-bitový I2C port expander).

Zapneme RPi a připojíme se přes SSH, viz 6.3.2.

Přečteme hodnotu na vstupu:

```
i2cget -y 1 0x20 9
```

Pokud bude senzor nezakrytý, bude na výstupu `0x00`, když senzor zakryjeme a příkaz zopakujeme, měla by na výstupu být `0x80`, ve dvojkové soustavě 1000 0000. Čteme hodnotu GP7, tj. hodnotu nejvyššího bitu.

Obdobně můžeme nastavovat hodnotu na výstupu:

```
i2cset -y 1 0x20 0 0xbf
```

Pokud zapíšeme `0xbf` LED1 se rozsvítí. Při zápisu `0xff` se LED1 zhasne. Ovládneme GP6, tj. 7 bit, aktivaci provádíme log. 0. Zápis `0xbf` znamená v dvojkové formě zápis 1011 1111.

Pro pokročilejší užití je vhodnější k ovládání využít přístup přes knihovny programovacích jazyků C++ nebo Python. Obdobně jako v kapitole 6.5 nyní realizujeme ovládání LED1 v jazyce Python. Na základě aktivace PIR senzoru, dojde k rozblikání LED1 a výstupu upozornění na terminál. Program spustíme příkazem:

```
~/Mereni/pir_i2c.py
```

Pokud zakryjeme PIR senzor, dojde k aktivaci alarmu a rozbliká se LED1 s výstupem na terminál „ALARM AKTIVACE“. Po odeznění LED1 nesvítí s výstupem na terminál „NENI ALARM“.

6.7.2 Využití I2C sběrnice pro měření elektrických veličin (AD převodník)

Na HW modul připojíme modul „AD(DA) PCF8591“, viz obrázek 22.

Na modulu AD převodníku propojíme piny konektoru JP1 pomocí jumperů následovně:

- AIN0-INPUT0
- AIN1-INPUT1
- AIN2-INPUT2
- AIN3-AOUT

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

S převodníkem budeme pracovat v jazyce Python, spustíme program:

```
~/Mereni/pcf8591.py
```

Nyní můžeme ovládat čtyři AD vstupy a jeden DA výstup dle menu programu:

- **0 AD kanál 0 - potenciometr** napětí děliče, které nastavujeme pomocí potenciometru RW1, viz obrázek 11.
- **1 AD kanál 1 - foto-odpor** - napětí foto-odporu R5, viz obrázek 11.
- **2 AD kanál 2 - teplotně závislý odpor** - napětí teplotně závislého odporu R6, viz obrázek 11.
- **3 DA převodník, výstup měřen na AD kanál 3** - nastavení hodnoty DA převodníku a odečtení nastaveného napětí pomocí kanálu 3 AD převodníku.

6.7.3 Využití sběrnice I2C pro měření neelektrických veličin

Na HW modul připojíme modul „4 in 1(meteo)“, viz obrázek 22.

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Pro měření intenzity osvětlení pomocí obvodu **TSL2561** spustíme program:

```
~/Mereni/tsl2561.py
```

Program nám měří intenzitu osvětlení ve viditelném a infračerveném spektru.

Pro měření tlaku pomocí čidla **BMP180** musíme nejdříve nainstalovat potřebné ovladače:

```
sudo apt-get install build-essential python-dev
cd ~/Měření/
git clone https://github.com/adafruit/Adafruit_Python_BMP.git
cd Adafruit_Python_BMP
sudo python setup.py install
```

K vlastnímu měření si upravíme vzorový program `simpletest.py`⁴⁵ z tohoto projektu a spustíme:

```
~/Měření/Adafruit_Python_BMP/examples/BMP.py
```

Program nám poskytuje hodnoty:

- Teplota
- Absolutní atmosférický tlak
- Atmosférický tlak přepočtený na hladinu moře
- Nadmořská výška

6.8 Obsluha SPI

Rozhraní SPI můžeme využít v podstatě stejně jako rozhraní I2C, k připojení různých čidel a převodníků. Toto jsme již realizovali v kapitole 6.7 právě pomocí sběrnice I2C. Proto pomocí SPI rozhraní budeme řešit připojení dotykového TFT displeje, na kterém můžeme realizovat grafický výstup celého OS, potažmo našich úloh.

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.
- TFT displej 3,5"

⁴⁵Vytvořený program `BMP.py` je součástí příloženého CD, viz příloha 4.

Máme povolené SSH připojení, viz kapitola 6.2.2

Na HW modul připojíme modul „DISPLEJ”, viz obrázek 22.

Zapneme RPi a připojíme se přes SSH, viz 6.3.2.

Stáhne aktuální ovladač displeje ze stránek [http://www.waveshare.com/wiki/3.5inch_RPi_LCD_\(A\)](http://www.waveshare.com/wiki/3.5inch_RPi_LCD_(A)) do RPi[46]:

```
cd /tmp
wget http://www.waveshare.com/w/upload/4/4b/LCD-show-161112.tar.gz
```

rozbalíme a překopírujeme do domovského adresáře

```
tar xvf LCD-show-*.tar.gz
cp -r LCD-show ~/
```

a nainstalujeme ovladač:

```
cd ~/LCD-show
chmod +x LCD35-show
./LCD35-show
```

Po nainstalování se RPi automaticky restartuje, displej by měl začít zobrazovat.

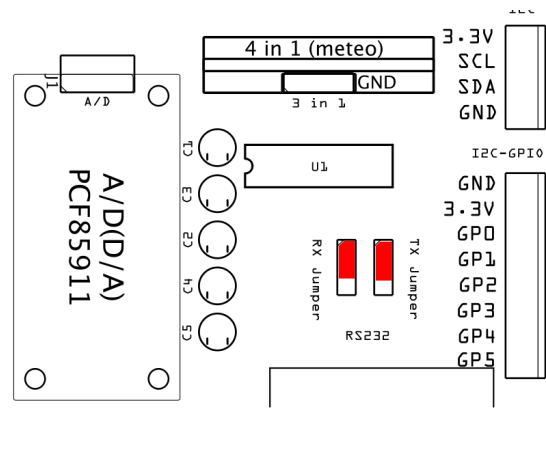
6.9 Obsluha RS232

Komunikaci přes sériový port později využijeme ke komunikaci s ARDUINO modulem. Ukážeme si jednoduchou komunikaci pomocí terminálu s PC.

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.
- Sériový kabel RS232, popřípadě převodník USB<=>RS232.

Připojíme RPi do HW modulu a nakonfigurujeme HW modul dle obrázku 37.



Obrázek 37: RS232 konfigurace

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Propojíme PC s HW modulem sériovým kabelem.⁴⁶

Jelikož RPi 3 obsahuje také Bluetooth, který je primárně využit jako sériový port a díky němuž je HW sériový port (UART) zakázán a je na místo něho použit sériový port SW emulovaný na pinech #8 a #10, zakážeme Bluetooth sériový port a budeme využívat plnohodnotný sériový port UART.

Bluetooth sériový port zakážeme editací souboru „/boot/config.txt“

```
sudo nano /boot/config.txt
```

kde na konec souboru přidáme:

```
dtoverlay=pi3-disable-bt
```

a následně restartujeme RPi:

```
sudo reboot
```

6.9.1 Komunikace pomocí terminálu v GNU/Linux

Připojíme se přes SSH, viz kapitola 6.3.2.

Spustíme příkaz pro čtení dat ze sériového portu:

```
cat /dev/serial0
```

⁴⁶V případě, že PC není osazeno v dnešní době již málo obvyklým sériovým portem, použijeme převodník USB<=>RS232.

Na PC spustíme další terminál a zadáme příkaz pro zapsání textu na sérový port:⁴⁷

```
su
echo Hello world >/dev/ttyUSB0
```

Po provedení příkazu `echo...` by se měl na terminálu RPi objevit text „Hello word“.

Nyní vyzkoušíme identicky opačnou komunikaci, tak že na PC zadáme příkaz pro čtení ze sériového portu:

```
cat /dev/ttyUSB0
```

a v terminálu RPi zadáme příkaz pro zápis na sériový port:

```
echo Hello world >/dev/serial0
```

Po provedení příkazu `echo...` by se měl text „Hello word“ tentokrát objevit na terminálu PC.

6.9.2 Výstup hodnot z programu na sériový port

Ukážeme si možnost využití sériového portu pro příjem zpráv o alarmu PIR čidla.

Připojíme se přes SSH, viz kapitola 6.3.2. Spustíme program pro obsluhu PIR čidla, kde je naprogramován výstup i na sériový port:

```
~/Mereni/pir.py
```

Spustíme na PC terminál a zadáme příkaz pro čtení ze sériového portu:

```
su
cat /dev/ttyUSB0
```

Nyní přicházejí na sériový port stejné zprávy jako na terminál RPi.

⁴⁷V GNU/Linux nemá uživatel standardně práva používat sériový port, operace musíme provádět jako root, nebo změnit nastavení práv pro dané zařízení.

6.10 Připojení modulu ARDUINO

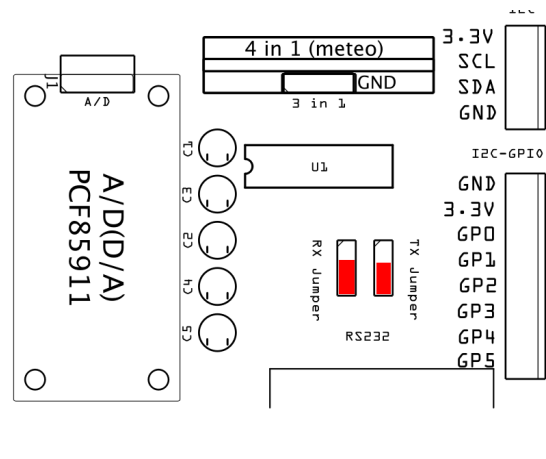
Pomocí Arduina realizujeme úlohu ovládání jasu LED2 přes PWM na základě nastavení napětí pomocí potenciometru R4. Pomocí RPi budeme přes sériovou linku odečítat napětí potenciometru a ovládat svit LED2.

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- Arduino UNO s výstupní úrovní 3,3 V kompatibilní.
- HW modul.

Na HW modul připojíme RPi a Arduino, viz obrázek 22.

Nakonfigurujeme HW modul dle obrázku 38.



Obrázek 38: RS232 konfigurace

6.10.1 Nahrání programu do ARDUINA

Nejprve si stáhneme nejnovější verzi Arduino IDE ze stránek <https://www.arduino.cc/en/Main/Software>. Vybereme si požadovaný OS a nainstalujeme. Program je multiplatformní Java aplikace, takže další postup bude pro všechny OS v podstatě identický.

Připojíme Arduino pomocí USB kabelu k PC.

Spustíme program Arduino, zkontrolujeme, že máme vybraný správný USB port (Nástroje → Port → /dev/usbserial/XXX). Zkontrolujeme že máme vybranou správnou vývojovou desku (Nástroje → Vývojová deska → Arduino UNO). Pokud je konfigurace nesprávná, upravíme.

Spustíme program Arduino a v menu(Soubor → Otevřít) vybereme soubor `Arduino_PWM.ino`⁴⁸

Nahrajeme program do Arduina(Projekt → Nahrát). Ihned po nahrání by měl být program funkční a na HW modulu by se měla rozsvítit LED2. Pokud budeme po-otáčet potenciometrem R4, měla by se měnit intenzita svitu LED2.

Ukončíme program Arduino a odpojíme Arduino od PC.

6.10.2 Komunikace ARDUINO - Raspberry Pi

Připojíme Arduino k RPi (přes USB z RPi napájíme Arduino)

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Nyní můžeme číst hodnoty napětí potenciometru přes sériový port zadáním příkazu:

```
cat /dev/serial0
```

Pokud budeme pootáčet potenciometrem R4, měla by se měnit intenzita svitu LED2 a hodnota napětí.

Dále nám program umožňuje přes sériový port ovládat svit LED2. Můžeme zapínat a vypínat LED2 posláním příkazu `led0` (LED nesvítí) nebo `led1` (LED svítí) přes sériový port, jak vidíme níže.

```
echo led0 >/dev/serial0 #led0 - LED nesvítí
echo led1>/dev/serial0 #led1 - LED svítí
```

6.11 Raspberry Pi jako web server

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.

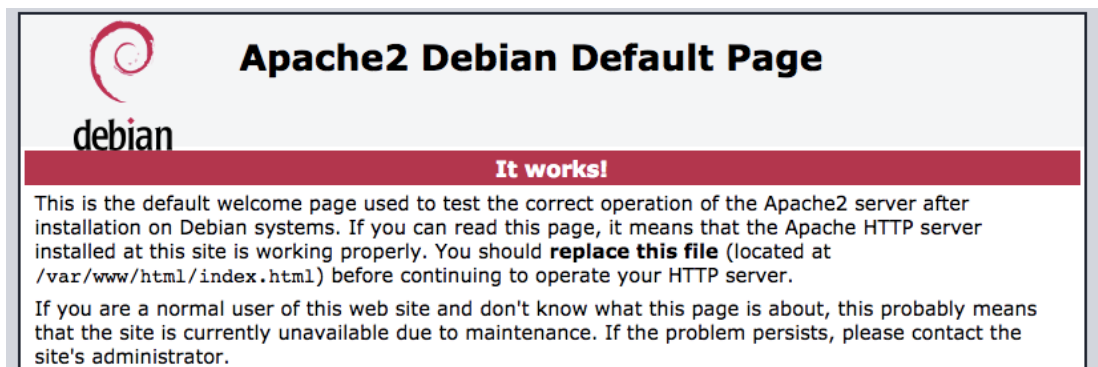
⁴⁸Vytvořený program `Arduino_PWM.ino` je součástí příloženého CD, viz příloha 5.

Zapneme RPi a připojíme se přes SSH, viz kapitola 6.3.2.

Nainstalujeme web server Apache2:

```
sudo apt-get install apache2
```

Po instalaci ověříme funkčnost nově nainstalovaného serveru, tak že na PC připojeném do stejné sítě jako RPi spustíme webový prohlížeč a zadáme IP adresu serveru 192.168.2.210 (stejnou, kterou používáme pro připojení přes SSH). Pokud vše funguje správně, zobrazí se nám „It Works!“, viz obrázek 39. Tím jsme ověřili, že server běží.



Obrázek 39: Apache2 - testovací stránka

Nainstalujeme podporu jazyka PHP. Jedná se o dva balíčky, balíček php5, tedy samotný interpret a balíček libapache2-mod-php5, který zajišťuje propojení interpreta se serverem. Instalaci provedeme příkazem:

```
sudo apt-get install libapache2-mod-php5 php5
```

Provedeme změnu vlastníka adresáře /var/www na uživatele pi. Momentálně tento adresář patří uživateli root, což by mohlo později činit potíže. Proto změníme vlastníka složky pomocí následujícího příkazu:

```
sudo chown -R pi /var/www
```

6.12 Možnosti zpracování a vizualizace dat

K realizaci této úlohy budeme potřebovat:

- Počítač s MS Windows nebo GNU/Linux a připojením na internet.
- Raspberry Pi 3 Model B s odpovídajícím napájecím zdrojem a připojením na internet.
- HW modul.
- TFT displej 3,5"

Připojíme se přes SSH, viz kapitola 6.3.2.

6.12.1 Zpracování dat pro následnou vizualizaci

Pro zobrazení dat získaných z připojených čidel využijeme programy vytvořené v předchozích úlohách `DHT.py`, `BMP.py` a `TSL2651.py`. Vytvoříme si skript `mereni`, pomocí něhož budeme ukládat do textových souborů výsledky měření. Tento skript budeme spouštět v pravidelných intervalech pomocí služby `cron`.

Vytvoříme si adresář `cron.mereni` pomocí příkazu:

```
sudo mkdir /etc/cron.mereni
```

Překopírujeme si skript `mereni` do nově vytvořeného adresáře:

```
sudo cp ~/Mereni/mereni /etc/cron.mereni
```

Editujeme `crontab` pro pravidelné spouštění skriptu `měření` příkazem

```
sudo crontab -e
```

a na konec souboru přidáme řádek:

```
*/1 * * * * /bin/run-parts /etc/cron.mereni/ >> /dev/null
```

Tímto jsme zajistili, že se bude skript měření spouštět pravidelně každou minutu a naměřené hodnoty ukládat do příslušných textových souborů v adresáři `/var/www/html`. Nyní můžeme ověřit, že existují textové soubory `dewpoint.txt`, `humidity.txt`, `press.txt`, `sun.txt`, `temp.txt`, `time.txt` a že obsahují naměřené údaje.

6.12.2 Vizualizace dat pomocí webového rozhraní

Pro zobrazení hodnot z čidel pomocí webové stránky musíme nakopírovat vytvořené stránky z lokálního počítače do RPi na kterém běží webový server.

Z lokálního počítače nakopírujeme adresář `html`⁴⁹ do `/var/www/html` pomocí příkazu `scp`:⁵⁰

```
scp -r html pi@192.168.2.210:/var/www/
```

Provedeme změnu vlastníka adresáře `/var/www` na uživatele `pi` pomocí následujícího příkazu:

```
sudo chown -R pi /var/www
```

Na vzdáleném počítači spustíme webový prohlížeč a zadáme adresu `192.168.2.210`. Měla by se zobrazit stránka a zobrazit hodnoty z čidel, viz obrázek 40.



Obrázek 40: Webová stránka - vizualizace

6.12.3 Vizualizace dat pomocí displeje HW modulu

Jedná se v podstatě také o zobrazení hodnot pomocí webové stránky. Tato stránka bude přizpůsobena pro rozlišení displeje pomocí responsivního zobrazení a spuštěna v celoobrazovkovém módu, tzv. „kiosk screen“. Abychom mohli toto realizovat musíme provést úpravy ve spuštění OS Raspbian.

Nainstalujeme si příslušný SW pomocí příkazu:

```
sudo apt-get install chromium-browser x11-xserver-utils unclutter
```

⁴⁹Vytvořené webové stránky jsou součástí přiloženého CD, viz příloha 7.

⁵⁰`scp` je součástí instalace GNU/Linux. V MS Windows použijeme identicky program `pscp`, který je součástí instalace programu `putty`, viz kapitola 6.3.2.

Editujeme soubor pro spuštění okenního manažeru:

```
sudo nano ~/.config/lxsession/LXDE-pi/autostart
```

A upravíme soubor do následující podoby:

```
@lxpanel --profile LXDE-pi
@pcmanfm --desktop --profile LXDE-pi
#@xscreensaver -no-splash
@xset s off
@xset -dpms
@xset s noblank

@sed -i 's/"exited_cleanly": false/"exited_cleanly" true/'
~/.config/chromium/Default/Preferences
@chromium-browser --noerrdialogs --kiosk http://192.168.2.210 --
incognito

@point-rpi
```

Po restartování RPi by měla nastartovat stránka s naměřenými údaji v plném rozlišení, viz obrázek 41.



Obrázek 41: Chromium - kiosk screen

7 Návrh dalších úloh k řešení

V kapitole 4 a 5 jsme na zadaných úlohách demonstrovali způsob návrhu a vlastního řešení vzorových úloh. K řešení jsme vytvořili částečně univerzální HW modul. S tímto přípravkem je možné vytvářet další úlohy, jejichž vlastní řešení by si studenti realizovali již sami, podobně jako u vzorových úloh. V návaznosti na vzorové úlohy by se dalo realizovat nepřeberné množství dalších úloh, namátkou zmíníme některé zajímavé náměty:

- Měření průtoku kapalin pomocí Hallova snímače⁵¹.
- Měření vzdálenosti pomocí ultrazvukového čidla⁵².
- Komunikace s GPS modulem pomocí RS232⁵³
- Ovládaní modulu SSRelay⁵⁴ s využitím obvodu MCP23009.
- Spínání piezo-sirény (přes SSRelay) na základě detekce pohybu (modifikace úlohy dle kapitoly 4.7.2).
- Použití nástroje RRDtool⁵⁵ pro záznam dat.
- Použití nástroje RRDtool pro vizualizaci dat.
- Využití TFT displeje pro ovládání vstupů.

⁵¹<http://www.dx.com/p/g1-2-plastic-water-flow-sensor-turbine-hall-flowmeter-for-arduino-437604#.WVecgnXyhTY>

⁵²<http://www.dx.com/p/open-smart-ultrasonic-sensor-module-4-digit-display-for-arduino-448194#.WVecUnXyhTY>

⁵³<http://www.dx.com/p/u-blox-neo-6m-gps-positioning-module-to-serial-ttl-rs232-black-312527#.WVea6HXyhTY>

⁵⁴<http://www.dx.com/p/2-ch-5v-solid-state-relay-for-arduino-382764#.WVeSKnXyhTY>

⁵⁵<http://oss.oetiker.ch/rrdtool/>

8 Závěr

Bakalářská práce měla za cíl zmapovat možnosti a prakticky vytvořit příklady laboratorních úloh zabývajících se implementací „mini PC“, konkrétně Raspberry Pi 3 Model B, jako řídicího prvku systému pro měření a vizualizaci fyzikálních veličin.

Na základě zadání laboratorních úloh byl vybrán konkrétní typ počítače „mini PC“ vhodný pro realizaci. Lze konstatovat, že po stránce HW vybavení vyhovovaly kritériím všechny základní modely testovaných výrobků. Vybraný model Raspberry Pi 3 Model B byl upřednostněn spíše díky konzervativnímu přístupu, jako model s největší uživatelskou základnou a velmi širokou komunitou vývojářů. Tím pádem byla nejlépe popsána implementace námi požadovaných funkcí a praktická realizace úloh byla možná bez zásadních problémů.

Vlastní návrh souboru úloh probíhal zmapováním teoretických možností, následně experimentální realizací na nepájivém poli. Takto byly postupně realizovány všechny úlohy a následně propojeny ve funkční celek. Na základě tohoto návrhu byl vytvořen přípravek, HW modul.

Nutnost vytvoření HW modulu se ukázala při realizaci na nepájivém poli. Díky širokému rozsahu úloh bylo velmi nepohodlné a v podstatě nerealizovatelné řešit úlohy bez nějakého přípravku přímo propojováním jednotlivých komponent. HW modul byl navržen pro realizaci konkrétních dvanácti úloh z této bakalářské práce, ale přesto byl koncipován částečně univerzálně, aby bylo možné úlohy obměňovat, rozšiřovat a vytvářet úplně nové.

Navržené úlohy byly prakticky realizované s pomocí vytvořeného HW modulu. Obdobným způsobem lze studentům zadávat další úlohy k samostatnému řešení. S pomocí HW modulu a již vyřešených úloh by měli být schopni realizovat modifikaci těchto úloh, nebo vytvářet úplně nové. Návrhy možných dalších úloh jsou obsahem kapitoly 7.

V bakalářské práci byla ověřena použitelnost počítačů „mini PC“ jako vhodných kandidátů pro širokou škálu praktických odborných předmětů zabývajících se problematikou výpočetní techniky, programování, měření a automatizace. Doplněním vhodných modulů, kterých je na trhu v současné době dostatečné množství, je možné realizovat levnou a přitom dostatečně výkonnou učební pomůcku reagující na soudobé trendy v moderní elektronice.

Na základě výše uvedených závěrů lze konstatovat, že bakalářská práce splnila stanovené cíle.

Použitá literatura a zdroje

- [1] Internet věcí. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-07-05]. Dostupné z: https://cs.wikipedia.org/wiki/Internet_v%C4%9Bc%C3%AD
- [2] RPi General History. Embedded Linux Wiki [online]. Simon Small, 2017 [cit. 2017-04-11]. Dostupné z: http://elinux.org/RPi_General_History
- [3] Raspberry Pi. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-03-10]. Dostupné z: https://cs.wikipedia.org/wiki/Raspberry_Pi#Model_.E2.80.9A.E2.80.9C
- [4] Raspberry Pi 3 Block Diagram. Element14 community [online]. 2017 [cit. 2017-04-11]. Dostupné z: <https://www.element14.com/community/community/raspberry-pi/blog/2017/01/16/raspberry-pi-3-block-diagram>
- [5] BCM2837. Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-04-17]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2837/README.md>
- [6] BCM2835 ARM Peripherals [online]. Broadcom Europe, 2012 [cit. 2017-04-17]. Dostupné z: <https://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>
- [7] POWER SUPPLY. Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-04-17]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>
- [8] USB. Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-04-17]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/usb/README.md>
- [9] USB OTG. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-17]. Dostupné z: https://cs.wikipedia.org/wiki/USB_OTG
- [10] Raspberry Pi 3 Model B GPIO 40 Pin Block Pinout. Element14 community [online]. A Premier Farnell Company [cit. 2017-04-17]. Dostupné z: <https://www.element14.com/community/docs/DOC-73950/1/raspberry-pi-3-model-b-gpio-40-pin-block-pinout>

- [11] GPIO. Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-04-17]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>
- [12] General-purpose input/output. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-17]. Dostupné z: https://en.wikipedia.org/wiki/General-purpose_input/output
- [13] Raspbian. Raspbian [online]. 2017 [cit. 2017-04-11]. Dostupné z: <https://www.raspbian.org/FrontPage>
- [14] Ubuntu MATE. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-19]. Dostupné z: https://cs.wikipedia.org/wiki/Ubuntu_MATE
- [15] Ubuntu Core. Ubuntu Developer [online]. Canonical, 2017 [cit. 2017-04-17]. Dostupné z: <https://developer.ubuntu.com/core>
- [16] Learn about Windows 10 IoT Core. Microsoft Developer [online]. © Microsoft, 2017 [cit. 2017-04-17]. Dostupné z: <https://developer.microsoft.com/en-us/windows/iot/Explore/IoTCore>
- [17] PiNet network. In: PiNet [online]. [cit. 2017-04-19]. Dostupné z: <http://pinet.org.uk/assets/images/network1.png>
- [18] RISC OS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-19]. Dostupné z: https://en.wikipedia.org/wiki/RISC_OS
- [19] RASPI-CONFIG. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-03-10]. Dostupné z: <https://www.raspberrypi.org/documentation/configuration/raspi-config.md>
- [20] Secure Shell. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-27]. Dostupné z: https://cs.wikipedia.org/wiki/Secure_Shell
- [21] GPIO Sysfs Interface for Userspace. In: The Linux Kernel Archives [online]. Linux Kernel Organization [cit. 2017-04-27]. Dostupné z: <https://www.kernel.org/doc/Documentation/gpio/sysfs.txt>

- [22] Sysfs. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-30]. Dostupné z: <https://cs.wikipedia.org/wiki/Sysfs>
- [23] NORRIS, Donald. Raspberry Pi: projekty. 1. vyd. Brno: Computer Press, 2015, 264 s. ISBN 978-80-251-4346-9.
- [24] About Adafruit. Adafruit [online]. NYC: Adafruit, 2017 [cit. 2017-07-02]. Dostupné z: <https://www.adafruit.com/about>
- [25] Temperature and humidity module AM2302 Product Manual [online]. Aosong(Guangzhou) Electronics Co. [cit. 2017-06-01]. Dostupné z: <http://akizukidenshi.com/download/ds/aosong/AM2302.pdf>
- [26] I2C – What’s That? In: I2C [online]. [cit. 2017-04-19]. Dostupné z: <http://www.i2c-bus.org/>
- [27] I2C. In: Engineers Garage [online]. EngineersGarage [cit. 2017-07-07]. Dostupné z: <https://www.engineersgarage.com/sites/default/files/I2C%28Inter%20Integrated%20Circuit%29%20TWI%28Two%20Wire%20Interface%29.jpg>
- [28] MCP23009/MCP23S09: 8-Bit I/O Expander with Open-Drain Outputs [online]. Microchip Technology, 03/26/09n. 1. [cit. 2017-04-17]. BST-BMP180-DS000-09. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/22121b.pdf>
- [29] PCF8591: 8-bit A/D and D/A converter [online]. NXP B.V, 2013 [cit. 2017-04-17]. Dostupné z: https://www.nxp.com/documents/data_sheet/PCF8591.pdf
- [30] TSL2560, TSL2561: LIGHT-TO-DIGITAL CONVERTER [online]. TAOS, 2009 [cit. 2017-04-17]. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf>
- [31] BMP180: Digital pressure sensor [online]. Bosch Sensortec, 2013 [cit. 2017-04-17]. BST-BMP180-DS000-09. Dostupné z: <https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf>
- [32] Serial Peripheral Interface Bus. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-20]. Dostupné z: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

- [33] Serial Peripheral Interface (SPI). In: SparkFun [online]. SparkFun Electronics [cit. 2017-04-19]. Dostupné z: <https://cdn.sparkfun.com/assets/c/7/8/7/d/52ddb2dcce395fed638b4567.png>
- [34] SPI [online]. In: . RASPBERRY PI FOUNDATION [cit. 2017-04-20]. Dostupné z: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/spi/README.md#driver>
- [35] , RS-232. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-06-08]. Dostupné z: <https://cs.wikipedia.org/wiki/RS-232>
- [36] RS232 serial null modem cable wiring. In: Lammert Bies [online]. Lammert Bies [cit. 2017-07-07]. Dostupné z: https://www.lammertbies.nl/comm/info/RS-232_null_modem.html
- [37] Arduino. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-06-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Arduino>
- [38] Wiring (programovací jazyk). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-06-11]. Dostupné z: [https://cs.wikipedia.org/wiki/Wiring_\(programovac%C3%AD_jazyk\)](https://cs.wikipedia.org/wiki/Wiring_(programovac%C3%AD_jazyk))
- [39] Cron. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-06-21]. Dostupné z: <https://cs.wikipedia.org/wiki/Cron>
- [40] Responzivní web design. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-06-28]. Dostupné z: https://cs.wikipedia.org/wiki/Responzivn%C3%AD_web_design
- [41] Fritzing. Fritzing [online]. Friends-of-Fritzing foundation, 2017 [cit. 2017-04-11]. Dostupné z: <http://fritzing.org/home/>
- [42] ARDUINO & GENUINO PRODUCTS > Arduino/Genuino UNO. ARDUINO [online]. 2017 [cit. 2017-04-11]. Dostupné z: <https://www.arduino.cc/en/main/arduinoBoardUno>
- [43] NOOBS - Raspberry Pi Documentation. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-02-26]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/noobs.md>

- [44] Installing operating system images using Windows - Raspberry Pi Documentation. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-02-26]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/installing-images/windows.md>
- [45] Installing operating system images on Linux - Raspberry Pi Documentation. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-02-26]. Dostupné z: <https://www.raspberrypi.org/documentation/installation/installing-images/linux.md>
- [46] 3.5inch RPi LCD (A). Waveshare Wiki [online]. 2017 [cit. 2017-02-26]. Dostupné z: [http://www.waveshare.com/wiki/3.5inch_RPi_LCD_\(A\)](http://www.waveshare.com/wiki/3.5inch_RPi_LCD_(A))
- [47] Raspberry Pi 3 is out now! Specs.. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi [online]. RASPBERRY PI FOUNDATION, 2017 [cit. 2017-03-10]. Dostupné z: <https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>
- [48] Banana Pi M64. BPI [online]. Sinovoip, 2017 [cit. 2017-03-10]. Dostupné z: <http://www.banana-pi.org/m64.html>
- [49] . Orange Pi [online]. Shenzhen Xunlong Software CO.,Limited, 2017 [cit. 2017-03-10]. Dostupné z: <http://www.orange-pi.org/>
- [50] SIEVER, Ellen. Linux v kostce: pohotová referenční příručka. Praha: Computer Press, 1999. ISBN 80-7226-227-0.
- [51] Location of connectors and main ICs on Raspberry Pi 3. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-20]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/Raspberrypi_3B.svg/300px-Raspberrypi_3B.svg.png

Seznam obrázků

1	Raspberry Pi 3 rozložení konektorů	17
2	Raspberry Pi 3 Blokový Diagram[4]	23
3	Raspberry Pi 3 GPIO Header [10]	26
4	PiNet network [17]	29
5	PIR senzor	38
6	DHT22 komunikační protokol [25]	40
7	I2C sběrnice[27]	42
8	MCP23009 - blokový diagram[28]	43
9	MCP23009 - IODIR registr[28]	43
10	MCP23009 - GPIO registr[28]	43
11	AD(DA) převodník PCF8591	45
12	Control registr PCF8591[29]	46
13	Modul měření neelektrických veličin	47
14	Funkční blokový diagram TLS2561[30]	47
15	Control registr TLS2561[30]	48
16	Timing registr TLS2561[30]	48
17	BMP180 - časový diagram čtení 16-bit. hodnoty AD [31]	50
18	SPI sběrnice[33]	51
19	RS232 zapojení[36]	53
20	HW modul - schéma	60
21	HW modul - PCB	62
22	HW modul - osazení	63
23	HW modul - praktická realizace	63
24	NOOBS instalátor	66
25	Win32DiskImager	67
26	Terminál, výběr disku	67
27	grafické prostředí PIXEL	68
28	Raspberry Pi Configuration - System	70

29	Raspberry Pi Configuration - Interfaces	70
30	Raspberry Pi Configuration - Performance	71
31	Raspberry Pi Configuration - Localisation	71
32	Raspberry Pi - ifconfig	72
33	PuTTY - konfigurace	74
34	GPIO/I2C konfigurace	75
35	GPIO/I2C konfigurace	78
36	I2C - i2cdetect	79
37	RS232 konfigurace	83
38	RS232 konfigurace	85
39	Apache2 - testovací stránka	87
40	Webová stránka - vizualizace	89
41	Chromium - kiosk screen	90

Seznam tabulek

1	Porovnání parametrů „mini PC” - část 1	19
2	Porovnání parametrů „mini PC” - část 2	20
3	Porovnání parametrů „mini PC” - část 3	21
4	Porovnání parametrů „mini PC” - souhrn	22
5	Porovnání OS pro RPi	31

Seznam algoritmů

1	pir.py	39
2	dht.py	40
3	pir_i2c.py	44
4	pcf8591.py	45
5	tls2561.py	49
6	SPI.py	52
7	pir.py	53
8	Arduino_PWM.ino	54
9	mereni	56
10	index.php	57
11	index.php	57

Přílohy

1. CD – na přiloženém CD se nachází plné znění diplomové práce pod názvem souboru `/bp/RPi-soubor_lab_úloh.pdf`, dále jsou zde umístěny všechny soubory a programy potřebné pro realizaci úloh.
2. `porovnání_modelů.xlsx` (soubor je obsažen na přiloženém CD v adresáři `/bp`)
3. `dht.py` (soubor je obsažen na přiloženém CD v adresáři `/sw/adafruit`)
4. `BMP.py` (soubor je obsažen na přiloženém CD v adresáři `/sw/adafruit`)
5. `Arduino_PWM.ino` (soubor je obsažen na přiloženém CD v adresáři `/sw/Arduino_PWM`)
6. Adresář obsahující programy a scripty (soubory jsou obsaženy na přiloženém CD v adresáři `/sw/Mereni`)
7. Adresář obsahující webové stránky (soubory jsou obsaženy na přiloženém CD v adresáři `/sw/html`)