



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ROZŠIŘUJÍCÍ DESKA PRO RASPBERRYPI

RASPBERRYPI EXPANSION BOARD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tadeáš Divácký

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Lukáš Kopečný, Ph.D.

BRNO 2017



# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**  
Ústav automatizace a měřicí techniky

**Student:** Tadeáš Divácký

**ID:** 164255

**Ročník:** 3

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## Rozšiřující deska pro RaspberryPi

### POKYNY PRO VYPRACOVÁNÍ:

- 1) Seznamte se s AD převodníky, GPIO expandéry a obvody reálného času (RTC) vybavené sběrnicemi I2C nebo SPI.
- 2) Navrhněte a realizujte univerzální desku k RaspberryPi, která bude obsahovat GPIO expandér, AD převodník, RTC obvod, bzučák, LED diody, tlačítka připojená na GPIO porty expandéru a zdrojovou část pro napájení vlastní desky i RaspberryPi.
- 3) Tato deska bude obsahovat řídicí mikroprocesor, který bude zpracovávat signály z připojených periférií, ultrazvukových senzorů, generovat PWM výstupy a komunikovat s nadřazeným systémem.
- 4) Navrhněte firmware do mikroprocesoru pro komunikaci a ovládání periférií a ultrazvukových senzorů.
- 5) Volitelně navrhněte knihovnu pro práci v RaspberryPi, pomocí které budete komunikovat a ovládat periferie navržené univerzální desky.
- 6) Ověřte a vyhodnoťte výsledky Vaší práce.

### DOPORUČENÁ LITERATURA:

dle doporučení vedoucího práce

**Termín zadání:** 6.2.2017

**Termín odevzdání:** 29.5.2017

**Vedoucí práce:** Ing. Lukáš Kopečný, Ph.D.

**Konzultant:**

**doc. Ing. Václav Jirsík, CSc.**  
*předseda oborové rady*

### UPOZORNĚNÍ:

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



## **Abstrakt**

Tato práce se zabývá návrhem a následnou realizací rozšiřující desky pro RaspberryPi. Na začátku jsou vypsány všechny integrované obvody, které byly pro tuto práci vybrány. Dále je zde popsána napájecí část, sloužící jako zdroj energie pro integrované obvody a RaspberryPi, a také realizace desky.

Poslední část je zaměřena na software a to jest firmware do ATmegy a část o RaspberryPi která obsahuje vypsané knihovny které jsou použité pro funkčnost programu a napsanou knihovnu pro obsluhu integrovaných obvodů.

## **Klíčová slova**

RaspberryPi 2B, AD-převodník, AD7997BRU, GPIO expandér, MCP23017, Hodiny reálného času, RTC, DS3234, DC/DC měnič, LDO regulátor, I2C sběrnice, ATmega88PA, Ultrazvukové senzory, HC SR04

## **Abstract**

This thesis deals with the design and subsequent realization of the expansion board for RaspberryPi. At the beginning, all the integrated circuits selected for this work are listed. In addition, the power supply section for power circuits and RaspberryPi is described here, as well as the realization of the board. The last part is focused on software and firmware that is in ATmega series and part of Raspberry Pi which includes listed libraries that sum used for the functionality of the program and written library for handling integrated circuits.

## **Keywords**

RaspberryPi 2B, AD-converter, AD7997BRU, GPIO expander, MCP23017, Real time clock, RTC, DS3234, DCDC converters, LDO regulator, I2C bus, ATmega88PA, Ultrasonic sensor, HC SR04

### **Bibliografická citace:**

Divácký, T. *Rozšiřující deska pro RaspberryPi*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 51s. Vedoucí práce: Ing. Lukáš Kopečný, Ph.D.

## Prohlášení

„Prohlašuji, že svou závěrečnou práci na téma *Rozšiřující deska pro RaspberryPi* jsem vypracoval samostatně pod vedením vedoucího *bakalářské práce* a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **29. května 2017**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu *bakalářské* práce Ing. Lukáš Kopečný, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce, Ing. František Burian, Ph.D za pomoc při realizaci desky a také Andree Mrázové za pomoc s napsáním a korekturou práce.

V Brně dne **29. května 2016**

.....  
podpis autora

# Obsah

1	Úvod .....	9
2	Raspberry Pi .....	10
3	Volba Integrovaných obvodů pro rozšiřující desku .....	11
3.1	AD převodníky .....	11
3.2	GPIO expandéry .....	12
3.3	Hodiny Reálného času(RTC).....	13
3.4	Mikroprocesor .....	14
3.5	Ultrazvukový senzor.....	15
4	Zdrojová část .....	16
4.1	DC/DC měnič .....	17
4.2	Lineární měnič LDO .....	20
5	Sběrnice .....	21
5.1	I <sup>2</sup> C.....	22
5.2	SPI.....	23
6	Software .....	24
6.1	Firmware do ATmegy88 .....	24
6.1.1	Přetaktování.....	24
6.1.2	Obsluha I <sup>2</sup> C sběrnice.....	28
6.1.3	Obsluha ultrazvukového senzoru .....	29
6.2	Software do RaspberryPi.....	31
6.2.1	Připojení k RaspberryPi .....	31
6.2.2	Program v RaspberryPi .....	33
7	Realizace desky.....	35
8	Oprava chyb desky .....	37
9	Závěr .....	38
	Literatura .....	39
	Seznam zkratk.....	40
	Seznam příloh .....	41

## Seznam obrázků

Obr 1 - Raspberry Pi 2B[2] .....	10
Obr 2 - Pinout AD7997BRU [3] .....	11
Obr 3 - Pinout MCP23017[4] .....	12
Obr 4 - Pinout DS3234[5].....	13
Obr 5 - Pinout ATmega88PA-AU [6] .....	14
Obr 6 - HC-SR04 ultrazvukový senzor[7] .....	15
Obr 7 - Přívod napájení + Pojistka.....	16
Obr 8 - DC/DC měnič s LM2596.....	17
Obr 9 - Připojení zátěže .....	18
Obr 10 - Odpojení zátěže.....	18
Obr 11 - Napěťové zvlnění na DC/DC měniči.....	19
Obr 12 - LDO stabilizátor na desce.....	20
Obr 13 - Schéma I <sup>2</sup> C sběrnici na desce.....	21
Obr 14 - Tabulka adres zařízení na I <sup>2</sup> C směrnici.....	22
Obr 15 - Zapojení RTC na SPI sběrnici[5] .....	23
Obr 16 - Distribuce hodin v mikroprocesoru[6] .....	25
Obr 17 - Schéma zapojení krystalu k ATmege88 .....	25
Obr 19 - Doba startu pro krystal s nízkou spotřebou a volbou zdroje hodin[6] .....	26
Obr 18 - Provozní režim krystalů s nízkou spotřebou[6].....	26
Obr 20 - Okno programu ProgISP 1.72 pro volbu změny pojistek.....	27
Obr 21 - Naměřené impulzy na ultrazvukovém senzoru.....	29
Obr 22 - Blokované schéma celé desky .....	35
Obr 23 - Osazená deska umístěna na RaspberryPi .....	36
Obr 24 - Testování připojení LED diod .....	36

# 1 ÚVOD

Robotika a kybernetika jsou jakožto vědecké obory dnešní doby v největším rozmachu. Důležitou částí těchto věd jsou způsoby řízení, pro které jsou používány nejrůznější druhy platforem, jako např. Arduino, STM32-discovery kit, nebo RaspberryPi. Proto jsem se v této bakalářské práci zaměřil právě na jednu z těchto platforem. Problémem je, že řídicí desky neobsahují vše, co by bylo třeba, a tak jsem se snažil najít způsob, jak by se dokázaly všechny potřebné periferie vtěsnat do jedné rozšiřující desky.

Nejprve bude třeba vybrat vhodný typ AD převodníku, GPIO expandéru a stejně tak i RTC, a to jak z hlediska ceny, tak z hlediska typu sběrnice. Poté je třeba vytvořit návrh rozšiřující desky, která bude výše zmíněné integrované obvody obsahovat. Pro napájení je také třeba navrhnout zdrojovou část platformy a desky samotné. GPIO expandér bude třeba rozšířit bzučák, LED diody a tlačítka.

Do desky se poté zintegruje řídicí mikroprocesor, díky kterému se budou moci zpracovávat signály z připojených periférií, jako například z ultrazvukových senzorů vzdálenosti, dále generovat PWM výstupy a také bude umožněna komunikace s nadřazeným systémem.

Nakonec bude třeba navrhnout firmware, který bude sloužit ke komunikaci s nadřazenou platformou a bude obsluhovat připojené ultrazvukové senzory vzdálenosti.

## 2 RASPBERRY PI

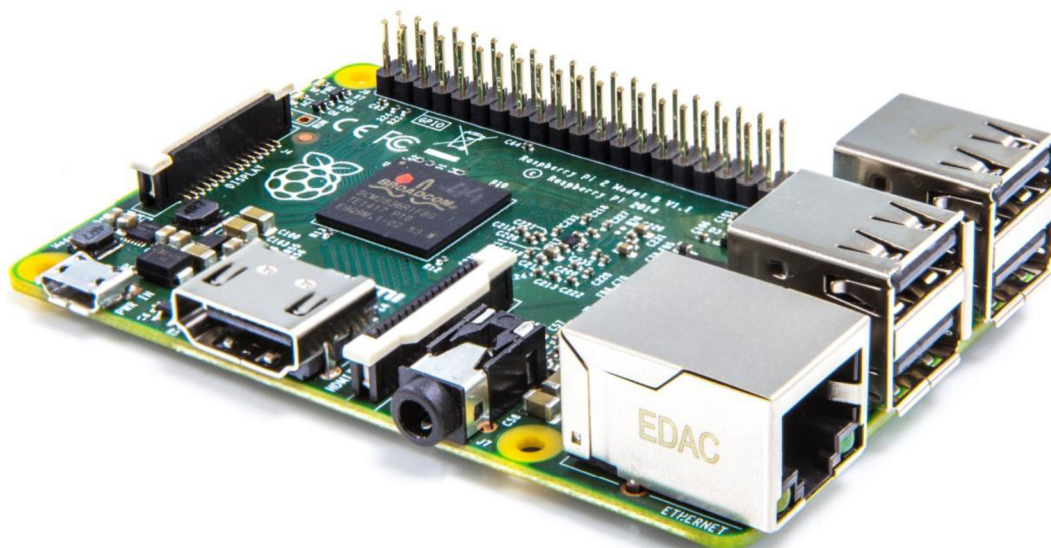
Výběr platformy pro tuto práci byl v zásadě jednoduchou volbou, jelikož jsem si práci s RaspberryPi vyzkoušel ve škole, kdy jsem měl za úkol postavit robota právě na této platformě.

Jedná se o jeden z druhů platforem o velikosti přibližně kreditní karty, který dokáže vykonávat funkce jako běžný počítač. To znamená, že na této platformě může běžet plnohodnotný operační systém, ať už Windows IoT nebo například Linux.

V této práci je vybráno RaspberryPi 2 Model B, který obsahuje čtyřjádrový procesor od firmy Broadcom s jádrem ARM Cortex-A7 a označením BCM2836. Jedná se o procesor, který pracuje na frekvenci 900MHz a obsahuje CPU, GPU, DSP, SDRAM, a jeden USB port. V této verzi RaspberryPi je zvětšená velikost paměti typu SDRAM na velikost 1 GB. [1]

Velkou výhodou této platformy oproti obyčejnému počítači je, že jsou zde vyvedeny GPIO porty, jež lze programově „realtime“ ovládat.

Během práce s RaspberryPi v předmětu BPRP, jsem stihl postřehnout pár nedostatků, které by se na tomto miniaturním počítači dali vylepšit. Je možné, že právě z důvodu nedostatku prostoru na něj výrobci nemohli přidat všechny potřebné součástky, ale jelikož mým cílem a pro mé potřeby je třeba menšího vylepšení, tak jsem se rozhodl připojit k němu svou vlastní rozšiřující desku.



Obr 1 - Raspberry Pi 2B[2]



## 3 VOLBA INTEGROVANÝCH OBVODŮ PRO ROZŠIŘUJÍCÍ DESKU

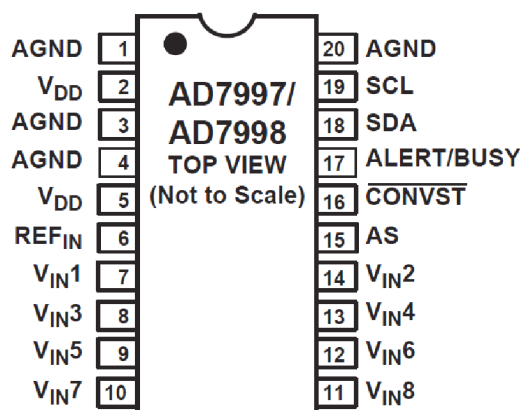
V této části je popsán výběr integrovaných obvodů, jako je AD převodník, GPIO expandér a RTC a také teorie fungování těchto obvodů.

### 3.1 AD převodníky

Jedná se o integrovaný obvod sloužící na převod analogové hodnoty napětí do digitální podoby. V dnešní době obsahuje AD převodník takřka každý mikroprocesor a lze jej najít v televizorech, telefonech, automobilech a jiných zařízeních. Existují různé druhy AD převodníků, jako například paralelní, aproximační, integrační, sigma-delta a další. Analogově digitální převodník je možné využít pro měření velikosti napětí v robotice, na senzory s analogovým výstupem, digitalizaci obrazu, atd.

Jelikož se v této práci snažím o co nejšetrnější využití prostoru, a umístit na desku další mikroprocesor obsahující AD převodník by přineslo práci a místo navíc, jsem se po konzultaci rozhodl pro externí AD převodník, který by komunikoval s RaspberryPi po sběrnici I<sup>2</sup>C. Na stránce gme.cz jsem našel čip s označením AD7997BRU, který mi vyhovoval jak z hlediska parametrů, tak z hlediska ceny.

Jedná se o aproximační 10 bitový AD převodník obsahující I<sup>2</sup>C sběrnici, jejíž adresa je určena dle specifického čísla čipu a dle způsobu zapojení pinu pro nastavení adresy, tedy zda je zapojený na kladné napájení nebo na zem. Tedy u tohoto převodníku je adresa při kladném zapojení 010 0010 (0x22), a při uzemněném zapojení 010 0001(0x21) a bez zapojeného pinu 010 0000(0x20).



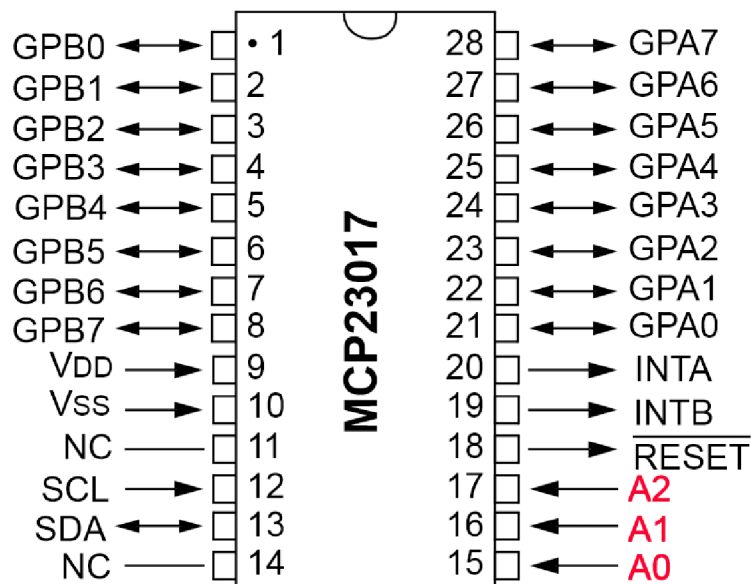
Obr 2 - Pinout AD7997BRU [3]

## 3.2 GPIO expandéry

GPIO expandéry, celým názvem general-purpose input/output, jsou piny, které se nacházejí na většině platform, a není předem nadefinované, jakým způsobem budou fungovat (pracovat). Až uživatel si určí, zda půjde o vstup (např. tlačítko, přerušeni, senzor a další), nebo o výstup (např. PWM, LED dioda, UART a další). Tato definice probíhá na základě GPIO pinů, díky nimž si můžeme určit nejen, zda jde o vstup či výstup, ale také zde lze interně nastavit pull up, nebo pull down rezistory, díky kterým je na pinu buď kladné napětí (obvykle 3.3 nebo 5V), anebo bude pin sloužit jako potenciál 0V (ZEM, GND).

GPIO port je tvořen převážně osmi piny GPIO expandéru, slouží jako prostředek k co nejlepšímu propojení řídicí jednotky a vnější periférie. Lze k němu připojit vše co má digitální vstup nebo výstup.

Pro svou práci jsem si vybral GPIO expandér MCP23017SO, který obsahuje dva osmibitové porty s označením A a B. Dále 3 piny s označením A0, A1 a A2, které můžeme zapojit na kladné napětí nebo na 0V (ZEM, GND), čímž můžeme manuálně navolit adresu tohoto slave obvodu na I<sup>2</sup>C sběrnici. Díky těmto třem pinům, které lze nastavit, je možné pospojovat k sobě až osm těchto obvodů MCP23017. To znamená, že dohromady lze RaspberryPi rozšířit až o 128 GPIO pinů.

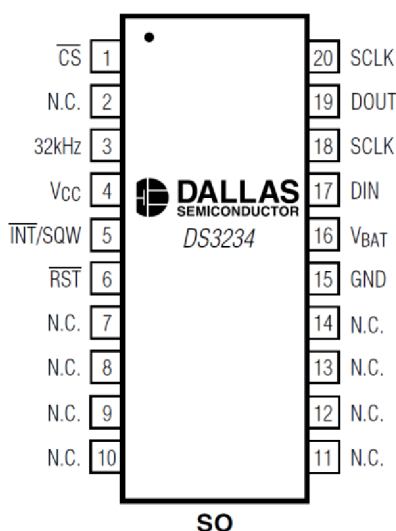


Obr 3 - Pinout MCP23017[4]

### 3.3 Hodiny Reálného času(RTC)

Jsou dva způsoby, kterými platformy, v tomto případě RaspberryPi, zjišťuje informace o aktuálním čase, které později využívá. Buď tyto informace získává z internetového připojení při každém zapojení, anebo si jej může pravidelně uchovávat a díky RTC a k němu připojené baterii si čas sám odpočítávat, podobně jako hodiny.

RTC, neboli Real Time Clock (hodiny reálného času) je pojmenování pro integrovaný obvod, sloužící k zapamatování a poskytování informací o aktuálním čase. Pro své fungování nepotřebuje nadřazenou periférii, ale bez této periférie je tato funkce RTC zcela zbytečná. RTC je tedy rozšiřující periférií pro platformu, jako je např. Arduino, nebo RaspberryPi.



Obr 4 - Pinout DS3234[5]

Pro svou práci jsem si zvolil integrovaný obvod RTC s označením DS3234 obsahující zapouzdřovaný krystal, jež ke své funkci nepotřebuje mnoho externích součástek a tudíž je vhodný k ušetření prostoru pro další potřebné integrované obvody. Komunikace s tímto čipem probíhá prostřednictvím SPI sběrnice.

Při výběru tohoto čipu jsem vycházel ze dvou kritérií, a to je typ sběrnice a kompatibilita s RaspberryPi. SPI sběrnici jsem zvolil z toho důvodu, že ke sběrnici I<sup>2</sup>C jsou připojeny zbývající obvody nacházející se na této desce.

### 3.4 Mikroprocesor

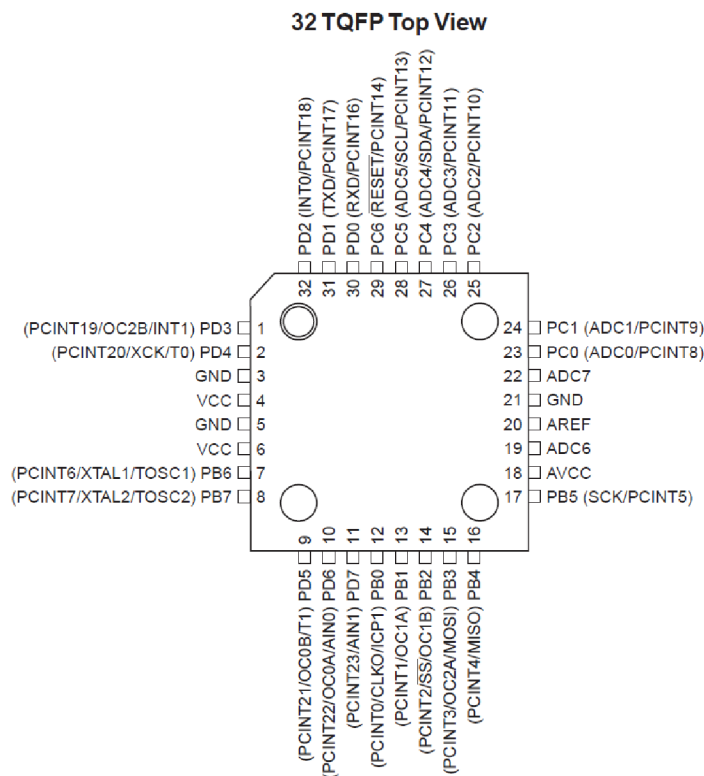
Mikroprocesory jsou integrovanými celky, které již obsahují paměť typu ROM a RAM a slouží jako řídicí jednotky.

Pro svou práci využívám mikroprocesor ATmega88PA od firmy Atmel. Tento mikroprocesor je z řady AVR a obsahuje paměti:

- EEPROM - o velikosti 512B, sloužící k dlouhodobému uchování dat
- SRAM o velikosti 1kB, která je volně přístupná na data a je energeticky závislá, což znamená, že se po odpojení napájení tato data ztrácí
- In-System-Programmed FLASH paměť o velikosti 8kB, sloužící k nahrání vlastního programu, jež se zachová i po odpojení zdroje energie.

Tento mikroprocesor pracuje na frekvenci 8 MHz, ale je možné jej přetaktovat externím krystalem až na 20MHz.[6]

Mikroprocesory tohoto typu mi vyhovují jak z hlediska jednoduchého zacházení a širokých možností využití pro různé typy aplikací, jako jsou např. generování PWM, komunikace po sériové lince, SPI sběrnice, může sloužit také jako I<sup>2</sup>C, master nebo slave. Tuto možnost jsem využil i v této práci, kde mikroprocesor slouží jako slave na I<sup>2</sup>C sběrnici.



Obr 5 - Pinout ATmega88PA-AU [6]

### 3.5 Ultrazvukový senzor

K měření vzdálenosti lze využít jak senzorů optických, tak senzorů ultrazvukových. Optické senzory využívají k měření vzdálenosti paprsek světla či už ve viditelném či neviditelném spektru. Ultrazvukové senzory však využívají zvukových vln, které nejčastěji vyše piezo-keramický měnič vůči překážce, od které se následně vlny odrazí a vrací se zpět k senzoru. Tento typ senzorů má využití v průmyslu nejen jako měřič vzdálenosti, ale také například jako měřič průtoku nebo výšky hladiny v nádrži.

Pro tuto práci byl vybrán senzor s označením HC-SR04. Existuje mnoho způsobů vyhotovení tohoto typu senzoru od různých výrobců a v různých cenových kategoriích pohybujících se od 20 do 100 korun.

Jedná se o senzor pracující na principu měření vzdálenosti pomocí ultrazvuku, tzn. na frekvenci vyšší než 20 kHz. Jeho obsluha se provádí pomocí čtyř pinů, z nichž dva jsou na napájení senzoru na 5V. Pomocí prvního pinu s označením TRIG startujeme měření a pomocí druhého pinu s označením ECHO získáváme zpětně informace o naměřené vzdálenosti, která je vyjádřena šířkou impulzů.

Deska obsahuje tři čipy s označením Max3232, LM324 a třetí čip je bez označení.

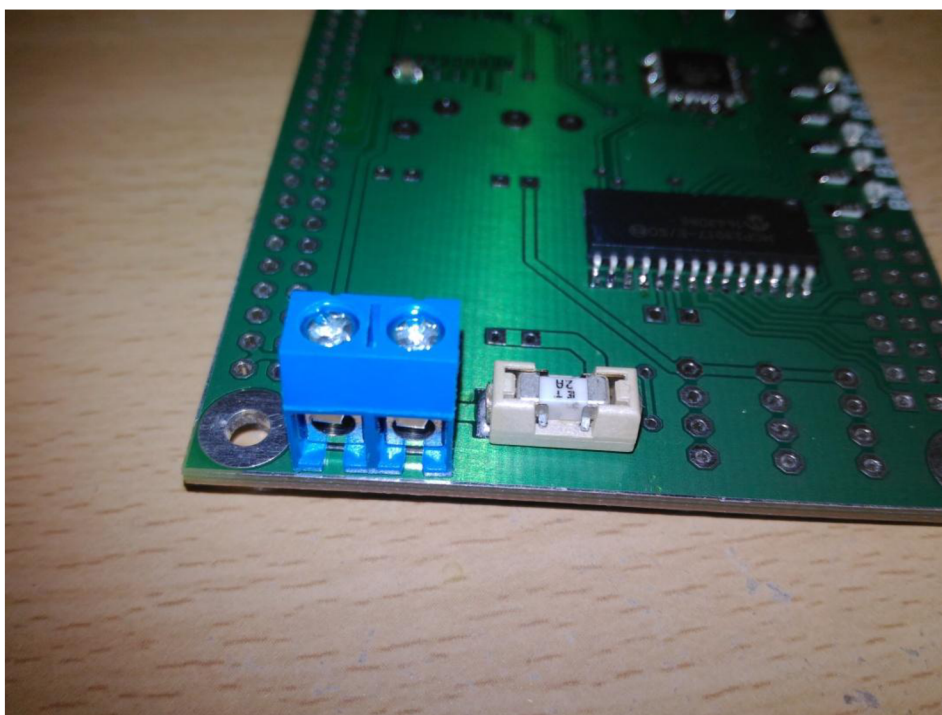


Obr 6 - HC-SR04 ultrazvukový senzor[7]

## 4 ZDROJOVÁ ČÁST

Zdrojová část této platformy je od počátku navrhovaná tak, že bude obsahovat dva napěťové potenciály. Jedná se o 5V napájení, sloužící jako zdroj energie pro RaspberryPi a ultrazvukové senzory, popřípadě by bylo možné z něj napájet také servomotor, a 3.3V napájení, které je zdrojem energie pro integrované obvody, jež jsou připojené k RaspberryPi logice.

Napájení je dovedeno prostřednictvím svorkovnice typu ARK500/2, do které je možnost upevnit drát s větším průměrem pro přivedení většího proudu. Maximální vstupní proud přivedený do desky je omezen na 2 ampéry. Z tohoto důvodu je ve vstupu umístěna pomalá vyměnitelná pojistka na 2A, sloužící na ochranu celé této platformy a RaspberryPi.



**Obr 7 - Přívod napájení + Pojistka**

Jelikož RaspberryPi potřebuje ke svému fungování 5V a integrované obvody zase 3.3V, bylo zapotřebí umístit na desku dva měniče.

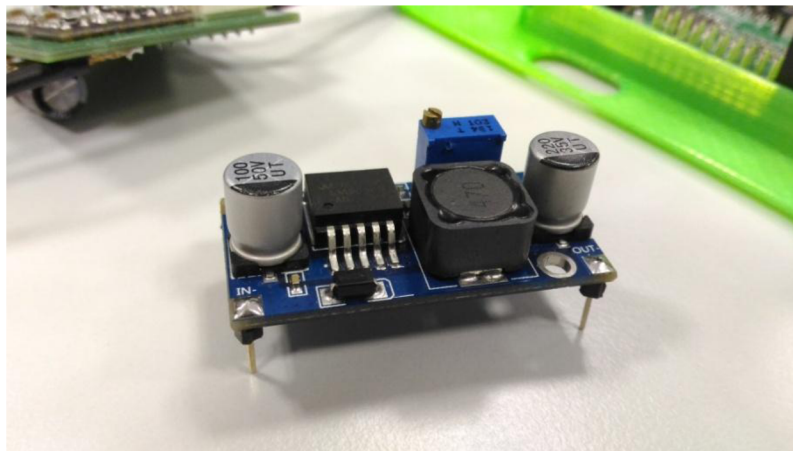
Měniče slouží ke snížení vyššího napětí na potřebnou úroveň pro fungování připojených periférií. Neznámějšími existujícími měniči stejnosměrného napětí jsou DC/DC měnič a LDO stabilizátor napětí.



## 4.1 DC/DC měnič

Pro tuto práci byl použit DC/DC Step-Down měnič na konvertování vstupního napětí na 5V pro RaspberryPi.

Při volbě DC/DC měniče bylo na výběr ze dvou typů. První obsahoval čip s označením LM2596, avšak po zkušenostech s tímto typem DC/DC měniče bylo zjevné, že pro tuto práci nebude vhodný. Po změření bylo zjištěno, že je jeho napěťového zvlnění velké, což by mohlo při poklesu nebo při vzestupu tohoto napětí vést k poškození RaspberryPi, a také bylo od pohledu zřejmé, že deska měniče je větších rozměrů, než je třeba.



**Obr 8 - DC/DC měnič s LM2596**

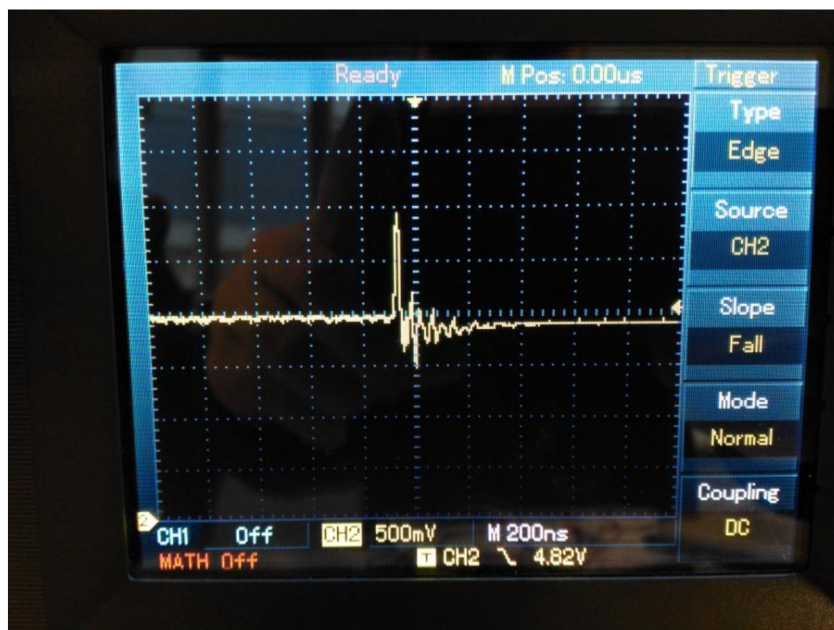
Proto byl také vybrán měnič obsahující čip s označením MP1584. Tento měnič byl zakoupen na ebay. Jedná se o čínskou verzi desky, a proto obsahuje plno nedokonalostí a nekorektností v zapojení. Po bližší kontrole desky byly nalezeny chyby, jež by mohly zásadně ovlivnit vlastnosti tohoto měniče.

Jedná se o vstupní a výstupní proudovou smyčku, která je nedokonale vytvořena, a může sama o sobě způsobovat rušení u výstupu. Dále je také nevhodně umístěn vstupní kondenzátor.

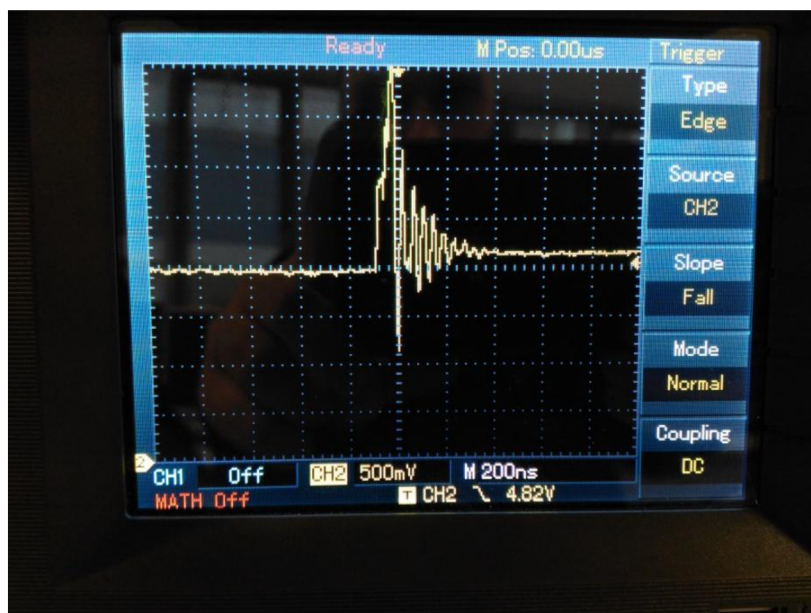
DC/DC měnič pracuje na frekvenci až 1.5MHz. Když si však vezmeme náhradní schéma elektrolytického kondenzátoru, zjistíme, že obsahuje parazitické vlastnosti, které se výrazněji projevují při vyšších frekvencích. Proto bylo následně do schématu zahrnut také keramický kondenzátor o 100nF.

Na tomto DC/DC měniči jsem provedl měření napěťového zvlnění a také připojení a odpojení zátěže, vše bez kondenzátoru.

U připojení a odpojení zátěže se jednalo o skokovou změnu z 0.5A na 1A a opačně.



Obr 9 - Připojení zátěže

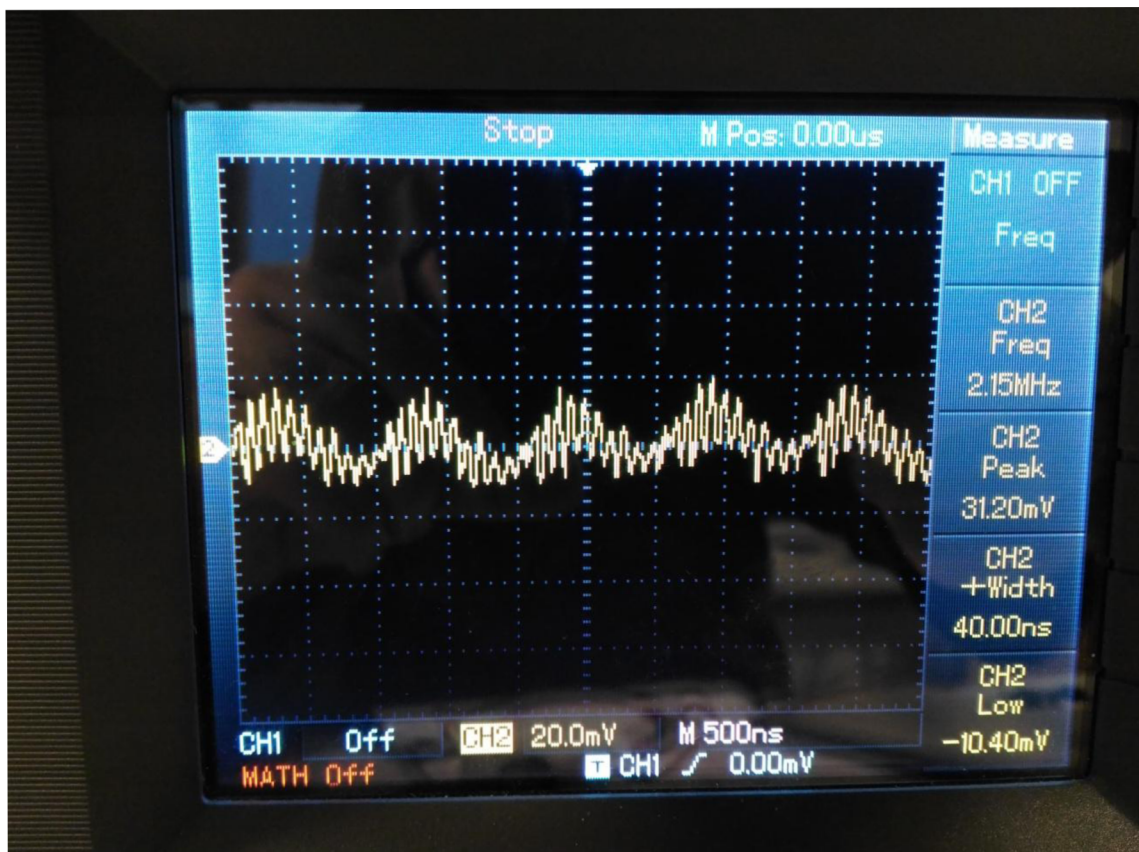


Obr 10 - Odpojení zátěže



Ve školních laboratořích bylo provedeno měření napětového zvlnění. Toto měření bylo provedeno na osciloskopu UNI-T bez výstupního filtračního kondenzátoru.

Napětové zvlnění kolísalo, ale pohybovalo se kolem hodnoty 30mV. Na Obr.11 je vidět, že byla naměřena hodnota 31.2mV.



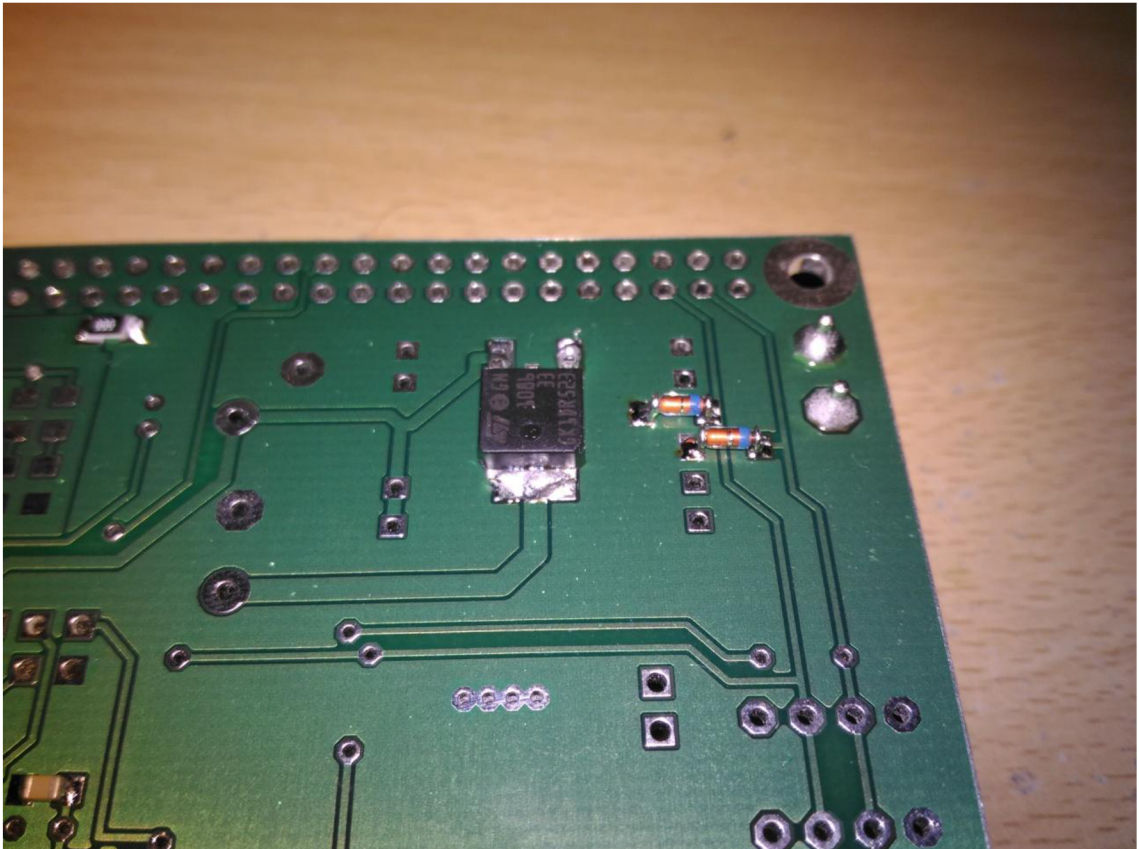
**Obr 11 - Napětové zvlnění na DC/DC měniči**

## 4.2 Lineární měnič LDO

LDO (Low drop) je stabilizátor s velmi malým úbytkem napětí. Jedná se o lineární DC regulátor napětí, který reguluje výstupní napětí, ačkoliv je vstupní napětí blízké tomu výstupnímu. Velkou výhodou tohoto typu regulátoru je, že zde není přítomnost spínacího šumu.

Další výhodou jsou jeho malé rozměry, jelikož se nachází v pouzdře DPACK. Do této práce byl zvolen stabilizátor s označením LD1086, jež je schopen poskytnout až 1.5A výstupního proudu. Při maximálním odebíraném proudu je zaručen maximální napěťový úbytek 1.2V.

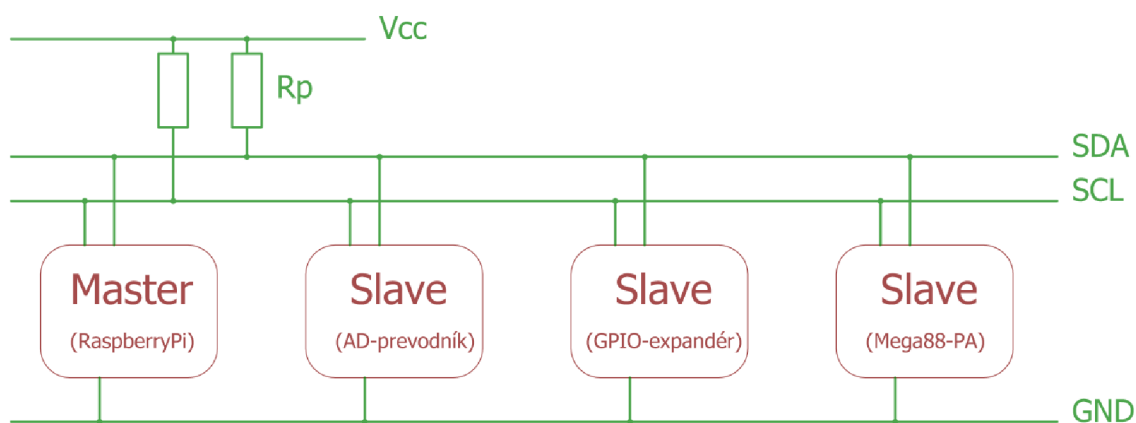
Tento stabilizátor poskytuje napěťovou úroveň 3.3V pro integrované obvody připojené k logice RaspberryPi.



Obr 12 - LDO stabilizátor na desce

## 5 SBĚRNICE

Sběrnice je buďto shromáždění vícero vodičů, nebo cestiček na plošném spoji, jejímž prostřednictvím probíhá komunikace mezi dvěma a více zařízeními. Počty vodičů se liší od typu sběrnice. V této práci jsou využívány dva typy sběrnic, a to I<sup>2</sup>C sběrnice a SPI sběrnice. Z důvodu programového využití I<sup>2</sup>C sběrnice dochází v Linuxovém jádru k omezení přístupu, a proto se pro RTC využívá SPI sběrnice.



Obr 13 - Schéma I<sup>2</sup>C sběrnici na desce

## 5.1 I<sup>2</sup>C

Tuto sběrnici s označením I<sup>2</sup>C, neboli Inter-Integrated Circuit, představila na trhu firma Phillips, a obsahuje dvojici hlavních vodičů a zem.

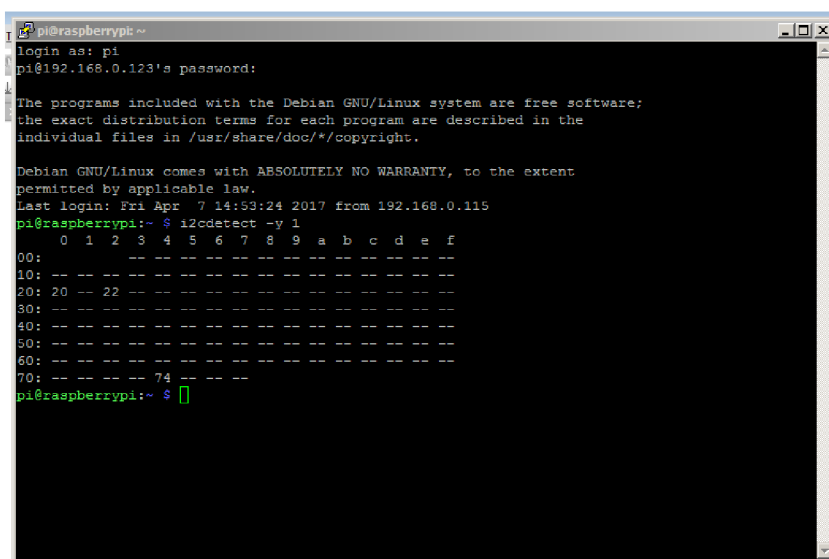
První z vodičů má označení SDA z anglického Synchronous Data. Jedná se o datový vodič, skrz který jsou obousměrně posílána data s jejich příslušnými adresami. Jelikož se jedná o synchronní posílání dat, druhý vodič s označením SCL, z anglického Synchronous Clock, zabezpečuje onu synchronizaci. Jedná se o synchronizační hodiny, jež zabezpečuje Master, jehož úlohu může vykonávat nadřazený mikroprocesor nebo počítač, v tomto případě RaspberryPi, a ten zde určuje, kdy se vysílají a přijímají data.

Je možné měnit rychlost této sběrnice, čímž je ovlivněn SCL vodič, který určuje, jak rychle bude komunikace probíhat.

Sběrnice je tedy rozdělena na master a slave obvody. Jako slave obvody lze v dnešní době použít různá zařízení, v této práci je představuje AD převodník, GPIO expandér a AT-mega88, která je programově nastavená jako již zmíněný slave obvod.

Na desce jsou také vyvedeny tři vývody na připojení dalších externích zařízení, jako například obvod KM2, sloužícího pro řízení krokových motorů. Obvody, které jsou na I<sup>2</sup>C sběrnici ovládané jako slave obvody rozeznáváme dle jejich adresy. U většiny zařízení lze volit adresy v určitém rozsahu. Část adresy je pevně zvolená z výroby, a druhá část adresy lze zvolit externě pomocí vyvedených adresovacích pinů.

AT-mega má ve svém kódu zabezpečenou možnost změny adresy v případě, že by došlo ke kolizi s jiným zařízením.



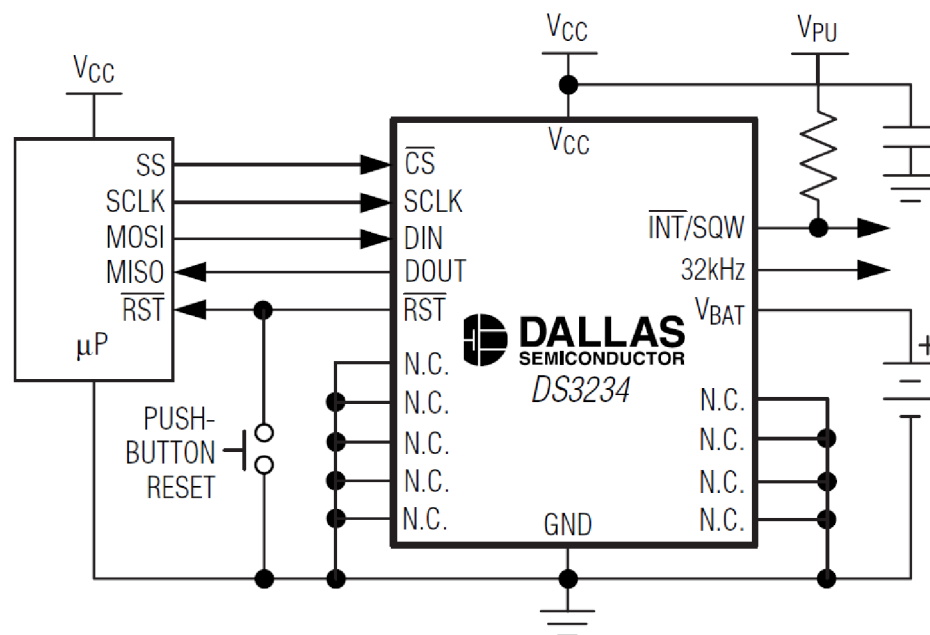
```
pi@raspberrypi ~$ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 -- 22 -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi ~$
```

Obr 14 - Tabulka adres zařízení na I<sup>2</sup>C směrnicí

## 5.2 SPI

Tato sběrnice s označením SPI, nebo také Serial Peripheral Interface, jež je použita pro RTC, je podobně jako I<sup>2</sup>C sběrnice rozdělena na master a slave obvody a také umožňuje oboustranné posílání dat, které zajišťují dva vodiče s označením MISO (v angličtině Master in, Slave out) a MOSI (Master out, slave in). Dále obsahuje vodič s označením SCK, jehož prostřednictvím je rozváděn hodinový signál generátoru hodinového signálu a SS který určuje, s kterým slave obvodem bude komunikace probíhat. V této práci je SPI sběrnice použita pro komunikaci s RTC, avšak blíže se s ní nepracuje, a proto je v této práci zmíněna pouze informativně.

Na Obr 15 je schéma zapojení integrovaného obvodu DS3234 což je RTC obvod na SPI sběrnici.



Obr 15 - Zapojení RTC na SPI sběrnici[5]

## 6 SOFTWARE

Laicky software představuje „to nehmotné“ uvnitř počítače či jiného zařízení. Software dělíme na systémový, který zajišťuje chod zařízení, a na aplikační, který může zajišťovat chod jiného zařízení. Software tedy mohou být data, či programy nacházející se v zařízení a kterých se na rozdíl od hardwaru nemůžeme dotknout.

V této práci je po softwarové stránce popsán firmware ATmegy88 a její přetaktování, I<sup>2</sup>C sběrnice, snímání ultrazvukový senzor, a RaspberryPi připojení a jeho programování.

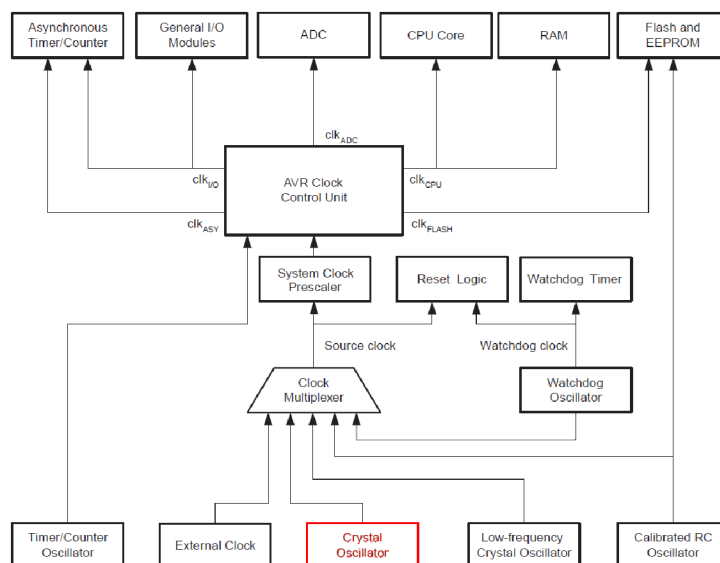
### 6.1 Firmware do ATmegy88

Firmware do AT-megy byl psán ve vývojovém prostředí AVR Studio s WinAVR knižnicemi. Po napsání kódu v jazyce C se firmware přeloží do souboru s příponou .hex, který se poté “vypálí” do flash paměti a soubor s příponou .eep, které se zase “vypálí” do eeprom paměti.

Práci s mikroprocesorem jsem rozdělil do tří kroků, a to přetaktování, obsluha I<sup>2</sup>C sběrnice a obsluha ultrazvukových senzorů.

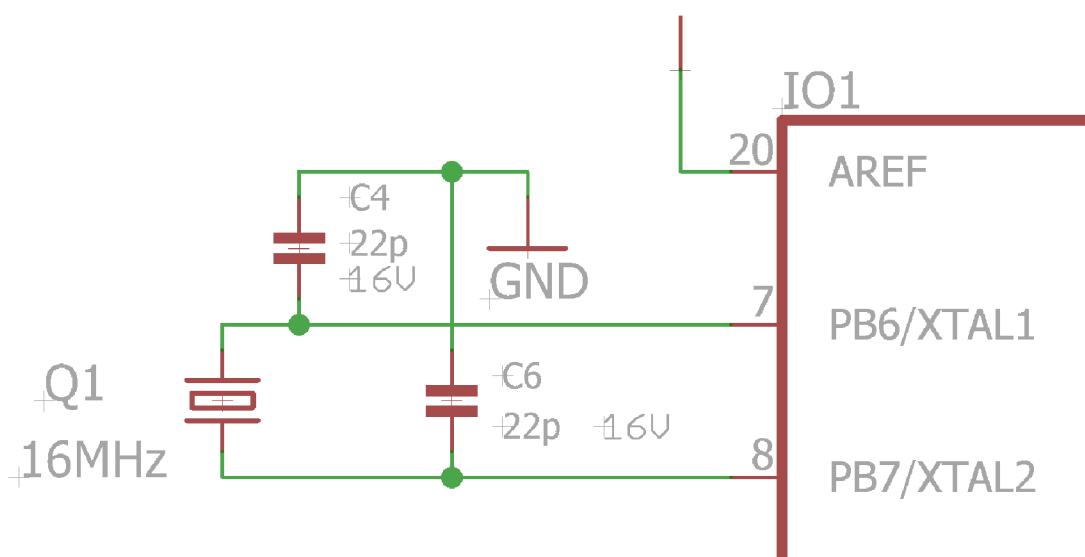
#### 6.1.1 Přetaktování

Mikroprocesor ATmega88PA-AU, který byl použit v této práci, je možné přetaktovat různými způsoby, a to externím hodinovým signálem, krystalem, krystalem s nízkou frekvencí, nebo kalibrovaným interním RC oscilátorem. Jak je vidět na Obr 16 na kterém je zobrazena distribuce hodin v mikroprocesoru, rozhodl jsem se přetaktovat tento mikroprocesor pomocí externího krystalu.



**Obr 16 - Distribuce hodin v mikroprocesoru[6]**

Tento krystal je zapojen na piny s označením PB6(XTAL1) a PB7 (XTAL2). Ke krystalu je potřebné připojit dva kondenzátory vůči zemi, viz Obr 17



**Obr 17 - Schéma zapojení krystalu k ATmege88**

Dle datasheetu[6] by kondenzátory měly mít hodnotu 12 až 22pF. Pro tuto práci byl zvolen keramický kondenzátor s hodnotou 22pF/16V v smd pouzdře o velikosti 0805.



V tabulce (viz Obr 18) s označením: Provozní režim krystalů s nízkou spotřebou (Low Power Crystal Oscillator Operating Modes ) pro 16MHz krystal, bylo potřeba změnit hodnoty v registru pojistek s označením CKSEL3, CKSEL2 a CKSEL1 na hodnotu 111.

Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 (pF)	CKSEL3...1 <sup>(1)</sup>
0.4 - 0.9	–	100 <sup>(2)</sup>
0.9 - 3.0	12 - 22	101
3.0 - 8.0	12 - 22	110
8.0 - 16.0	12 - 22	111

**Obr 18 - Provozní režim krystalů s nízkou spotřebou[6]**

Následně v tabulce (obr 19) s označením doba startu pro krystal s nízkou spotřebou a volbou zdroje hodin (Start-up Times for the Low Power Crystal Oscillator Clock Selection), byl zvolen krystal s rychlým nárůstem výkonu (Crystal Oscillator, fast rising power ) s nastavením hodnoty pojistek CKSEL0 na 1 , SUT1 na 1 a SUT0 na 0.

Oscillator Source / Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V <sub>CC</sub> = 5.0V)	CKSEL0	SUT1...0
Ceramic resonator, fast rising power	258 CK	14CK + 4.1ms <sup>(1)</sup>	0	00
Ceramic resonator, slowly rising power	258 CK	14CK + 65ms <sup>(1)</sup>	0	01
Ceramic resonator, BOD enabled	1K CK	14CK <sup>(2)</sup>	0	10
Ceramic resonator, fast rising power	1K CK	14CK + 4.1ms <sup>(2)</sup>	0	11
Ceramic resonator, slowly rising power	1K CK	14CK + 65ms <sup>(2)</sup>	1	00
Crystal Oscillator, BOD enabled	16K CK	14CK	1	01
Crystal Oscillator, fast rising power	16K CK	14CK + 4.1ms	1	10
Crystal Oscillator, slowly rising power	16K CK	14CK + 65ms	1	11

**Obr 19 - Doba startu pro krystal s nízkou spotřebou a volbou zdroje hodin[6]**

Dále bylo třeba změnit hodnotu v registru pojistek s označením CKDIV8 tak, aby bylo umožněno dělit vnitřní hodiny osmi. V datasheetu[6] je zmíněné, že jakmile se překročí interní maximální frekvence, je potřeba zapnout tuto funkci. Toto tvrzení bylo také ověřeno, když byla v mikroprocesoru použita funkce `_delay_us(10)`; pomocí které bylo vygenerováno zpoždění 10us na zdržení procesoru. Toto zdržení bylo změřeno na osciloskopu. Při prvním pokusu byly naměřeny impulzy s šířkou 80us.

Při druhém pokusu po změně hodnoty v registru CKDIV8 byla naměřena správná hodnota, tj. 10us.



Všechny tyto změny byly provedeny pomocí programátoru USBASP, a nahrávacího prostředí ProgISP 1.72. Tento program obsahuje okno, skrz které jde velmi jednoduše “naklikat” změny v pojistkách, a rovnou je nahrát do mikroprocesoru, jak je vidět na obrázku 20.



**Obr 20 - Okno programu ProgISP 1.72 pro volbu změny pojistek**

## 6.1.2 Obsluha I<sup>2</sup>C sběrnice

Táto část programu byla přebrána z projektu KM2[9]. Na počátku programu bylo potřeba nastavit registry I<sup>2</sup>C sběrnice, aby ji bylo možné zapnout jako slave. V registru s označením TWCR, jsou bity, pomocí kterých se provádí změny na této sběrnici. Pro tuto práci jsou nejdůležitější pouze dva z nich, a to bity s označením TWEA a TWEN.

- Bit TWEN – TWI Enable Bit slouží pro zapnutí I<sup>2</sup>C.
- Bit TWEA – bit sloužící ke generování potvrzovacího impulsu (ACK).

V hlavní supersmyčce se cyklicky kontroluje stav na I<sup>2</sup>C sběrnici. Podle toho, jaký je přijatý pokyn, se pomocí switch (přepínače) provede daný úkol.

Existují dvě činnosti, které lze s I<sup>2</sup>C vykonávat, a to je čtení a zápis.

- TW\_ST\_SLA\_ACK: jedná se o status, kdy mají být připravena data na čtení ze Slave do Master.
- TW\_ST\_DATA\_ACK: Tento status je nastaven v momentě, kdy se posílají data ze Slave do Master.
- TW\_SR\_SLA\_ACK: Tento status je nastaven, když budou posílána data z Master do Slave.
- TW\_SR\_DATA\_ACK: Tento status je nastaven, když se zasílají data z Master do Slave.
- TW\_SR\_STOP: Tento status je nastaven ve chvíli, kdy už byla odeslána všechna data z Master do Slave.

### 6.1.3 Obsluha ultrazvukového senzoru

Ultrazvukový senzor je obsluhován pomocí jedné funkce zvané Measure. V této funkci se pomocí přepínače (switch) vybere jeden ze čtyř senzorů, pomocí kterého bude následně provedeno čtení vzdálenosti.

Algoritmus pro práci se senzorem je následující:

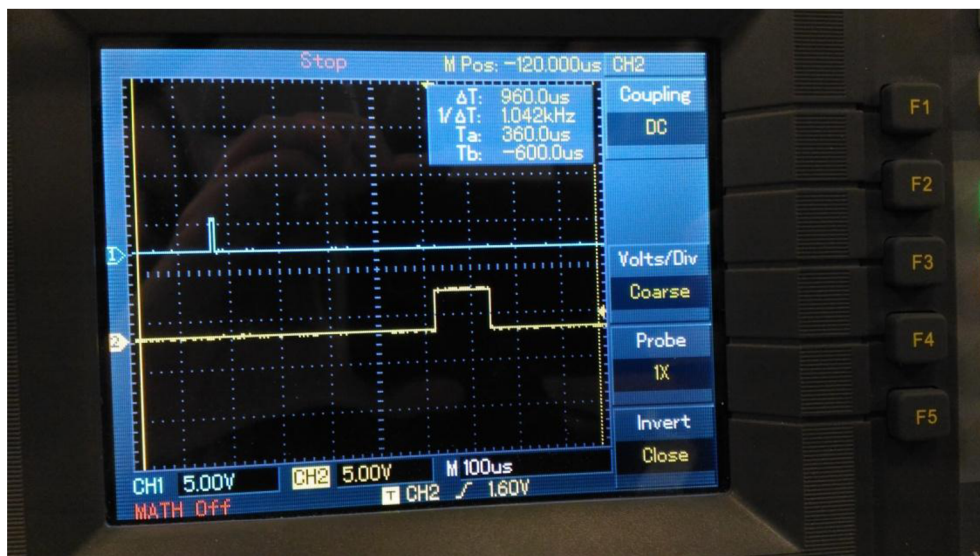
Pomocí funkce na zpoždění se vygeneruje 10us impuls, jehož prostřednictvím je zahájeno měření. Měření probíhá tak, že senzor vyšle osm impulsů o frekvenci 40kHz (proto je ultrazvukový), a následně pak nastaví výstup s označením ECHO na logickou jedničku, a čeká na odražený impuls. S této šířky impulsu pak lze vypočítat vzdálenost objektu, od kterého byla zvuková vlna odražena. Pro měření šířky impulsu je napsaná funkce `getPulse()`.

Přivolaná funkce pak měří šířku impulsu. Na měření této šířky je použit 16-bit časovač.

V této práci je však pro toto měření používáno 8 bitů. Po I<sup>2</sup>C je posíláno 16 bitů, kde horních 8 bitů je možné použít například na detekování chyb.

Po změření šířky impulsu je dané číslo uloženo do globální proměnné. Zde zatím nelze naměřený počet inkrementů považovat za vzdálenost objektu.

Na obr 21 je vyobrazen na osciloskopu naměřený 10us impuls, a následující impuls, jež představuje vzdálenost objektu.



Obr 21 - Naměřené impulsy na ultrazvukovém senzoru

Tato vzdálenost je pak v RaspberryPi nebo mikroprocesoru přepočítaná na centimetry, a s touto hodnotou se může dále pracovat.

Pro výpočet vzdálenosti je použita frekvence, na kterou je mikroprocesor přetaktován, což je v tomto případě frekvence 16MHz.[8]

Při práci s časovačem je nastavená předdělička (Prescale) na hodnotu 256, a tak je pak možné vypočítat, za jaký čas byla hodnota inkrementována.

$$inkrement[\mu s] = \frac{prescale}{F_{CPU}} = \frac{256[-]}{16 [MHz]}$$

Z tohoto výpočtu vyplývá, že časovač je inkrementován každých 16 $\mu$ s. S touto informací je možné vypočítat šířku impulsu.

$$t_{on}[\mu s] = inkrement[\mu s] * pocet\_inkrementu[-]$$

Když máme vypočítanou délku impulsu v mikrosekundách, tak můžeme vypočítat vzdálenost objektu od snímače. Pro tento výpočet využijeme rychlost zvuku ve vzduchu. Jako demonstrativní výpočet použijeme rychlost zvuku v suchém vzduchu při 25°C.  $C = 346.3$  m/s, což si pro výpočet převedeme na 0.0346.3 cm/ $\mu$ s.

$$vzdalenost[cm] = t_{on}[\mu s] * C\left[\frac{cm}{\mu s}\right]$$

Tato vzdálenost představuje celkovou délku letu zvukové vlny. Pro reálnou vzdálenost objektu je ještě potřeba podělit dvěma. Po tomto posledním úkonu je výsledná hodnota konečná hodnota v centimetrech.

Když tedy budeme pracovat s počtem inkrementů, stačí nám násobit konstantou

$$k = 0.27704[-]$$

## 6.2 Software do RaspberryPi

Jelikož se tato práce zabývá rozšířením RaspberryPi, bylo součástí práce také napsat vzorový program, který by demonstroval funkčnost rozšířeného RaspberryPi.

Pro RaspebrryPi existuje vícero operačních systémů [10] :

- RASPBIAN
- Ubuntu MATE
- Snappy Ubuntu
- PIDORA
- Linutop
- SARPi
- Arch Linux ARM
- Gentoo Linux
- FreeBSD
- Kali Linux
- RISC OS Pi

Pro tuto práci byl zvolen Rasbian Jessie lite, založený na Debian. Jedná se o nejrozšířenější linuxovou distribuci pro RaspberryPi.

### 6.2.1 Připojení k RaspberryPi

Po nainstalování RaspberryPi na SD kartu jako .img stažený z oficiálních stránek (<https://www.raspberrypi.org/downloads/>), připojení k monitoru a spuštění systému bylo potřeba povolit SSH (Secure Shell) ovládání. RaspberryPi je pak možné na dálku ovládat z externího počítače, kde běží například operační systém Windows. S pomocí programu PuTTY se pak lze z RaspberryPi připojit a ovládat jej pomocí linuxových příkazů. SSH je v nové verzi defaultně vypnuto, proto je potřebné ho zapnout v *raspi-config*.

Pro připojení RaspberryPi s notebookem, pomocí kterého se programuje, je možné různými způsoby. V této práci byly zkoušené ke zprovoznění dvě připojení.

První je že se v RaspberryPi nastaví připojení k wifi. Nastavení provedeme zde :

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

a to tak, že použijeme existující wifi síť, na kterou provedeme konfiguraci a RaspberryPi se k síti připojí. Poté notebook připojíme ve stejné síti a musíme si zjistit IP adresu RaspberryPi, aby bylo možné se k němu připojit.

Tento soubor je pak konfiguračně defaultně povolen v souboru, který se nachází zde:

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

Tento soubor použijeme i ve druhém případě. Avšak tentokrát nepoužijeme `wpa_supplicant.conf`, jež zakomentujeme v `interfaces`, ale vlastní konfiguraci wifi připojení. Bude se jednat o AdHoc sit. [11]

```
auto wlan0  
iface wlan0 inet static  
address 192.168.1.1  
netmask 255.255.255.0  
wireless-channel 1  
wireless-ssid MyNetwork  
wireless-mode ad-hoc
```

Pak stačí, když připojíme notebook k síti, kterou vytvoří RaspberryPi. Následně manuálně nastavíme IP o jedno číslo vyšší, aby nedošlo ke kolizi IP adres. Tento krok je nezbytný z toho důvodu, že RaspberryPi nemá DHCP server pro přidělování IP adres. Každá z těchto dvou možností má svoje výhody a také nevýhody.

Při první možnosti máme připojení na internet, při druhé ne. Při první možnosti jsme fixovaní na pevný bod na nějakou třetí stranu. Při druhé možnosti nám vystačí RaspberryPi s wifi donglem a notebook.

## 6.2.2 Program v RaspberryPi

Programování probíhalo ve vývojovém prostředí NetBeans, kde bylo potřebné nastavit BuildHost. BuildHost se nastavuje podle IP adresy RaspberryPi, na kterém probíhalo přeložení a spuštění programu.

Vzorový program byl napsán v jazyce C. Nejdůležitější částí bylo zahrnout pomocí preprocesorového příkazu `#include` všechny potřebné knihovny:

```
/*Knihovny pro praci s I2C pod jadrem linuxu*/
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <linux/i2c-dev.h>

/*Standartne knihovny pro C*/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdint.h>
#include <string.h>
#include <stdarg.h>
#include <inttypes.h>
#include <alloca.h>
#include <math.h>

//nestandardni knihovny
#include "logging.h"
#include "i2c.h"
#include "ad799x.h"
#include "mcp23017.h"
```

Velká část kódu je použita z již existujícího projektu. Zdrojové kódy jsou přebrané z projektu do předmětu BPRP, kde také vznikl nápad na tuto práci, jež je primárně určena pro rozšíření KamBota.

Používám knihovnu `Logging.h`, která je vytvořena pro logování zpráv. Primárně lze použít pro vypisování informací na obrazovku konzoly.

Dále knihovny `I2C`, `I2C.h` a `AD` převodník `AD799X.h` od doktora Františka Buriana [12].

V této práci jsem se zaměřil na napsání knihovny s názvem `mcp23017.h` která bude obsahovat obsluhu GPIO expandéru a periférií připojených k tomuto expandéru.

Knižnice obsahuje definice vypsané z datasheetu a funkce pro práci s GPIO expandérem:

```
static inline int BoardIO(int bus, int chip_addr, uint8_t port, uint8_t Value )
```

Jedna se o funkci pomocí, které lze na gpio expandéru nastavit zda bude požadovaný pin vstupní nebo výstupní. Návrátová hodnota z funkce signalizuje chybu

```
static inline int SetPullUpBTN(int bus, int chip_addr, uint8_t port, uint8_t Value )
```

Jedna se o funkci primárně určenou pro tlačítka. Pomoci ní lze nastavit vnitřní PullUp 100kohm rezistor pro jednotliví pin nebo celý port najednou. Lze nastavit, pouze když je daný pin nastaven jako vstupní. Návrátová hodnota z funkce signalizuje chybu.

```
static inline int gpio_BTN(int bus, int chip_addr, uint8_t *btn)
```

Další funkce je určena pro čtení tlačítek umístěných na desce. Funkce je vytvořena tak, že pokud je tlačítko přečteno správně, tak bude hodnota vracena buď 1,2,3, kde 1 je stlačené tlačítko číslo 1. 2 je stlačené tlačítko číslo 2. A 3 jsou stlačené obě dvě tlačítka najednou.

Pokud je hodnota přečtena zle, tak je vrácené záporné číslo čili chyba.

```
static inline int SetLED_BUZZ(int bus, int chip_addr, uint8_t Value);
```

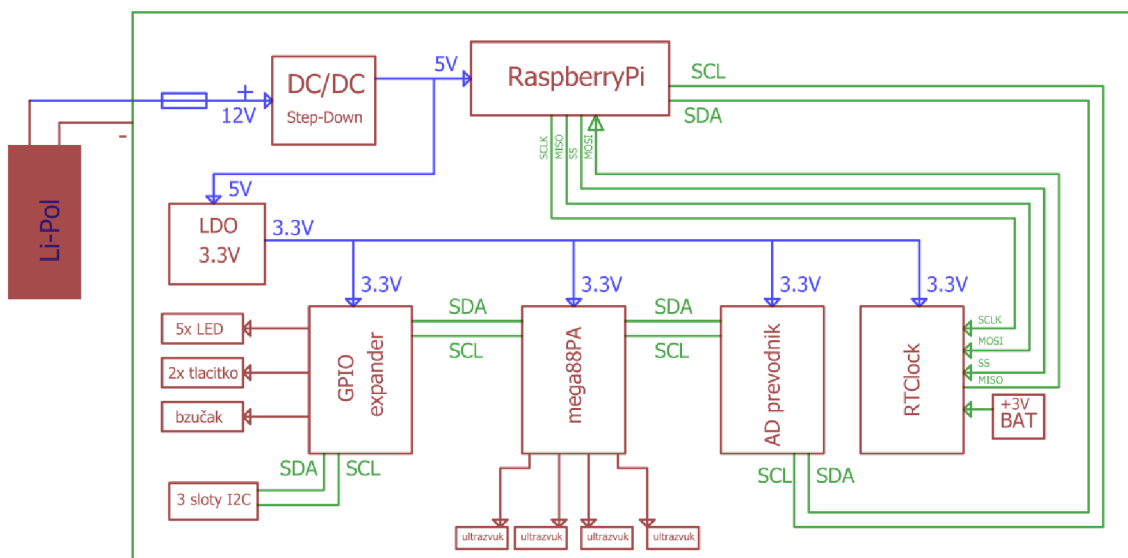
```
static inline int ClearLED_BUZZ(int bus, int chip_addr, uint8_t Value);
```

Další dvě funkce slouží pro obsluhu LED diod a bzučák. Jedna slouží na nastavení do logické jedničky a druhá na resetování do logické nuly daného pinu.



## 7 REALIZACE DESKY

Při realizaci desky jsem si navrhnul blokové schéma na obr 22, které obsahuje kompletně znázorněnou desku, co vše obsahuje.



Obr 22 - Blokové schéma celé desky

Schéma je rozděleno na 7 částí. Každou část jsem pak umístil do samostatného listu, které sou v příloze 1 A až G. Také na příloženém CD.

Realizace jednotlivých schémat byla jednoduchá. Většina zapojení integrovaných obvodu byla přebrána z datasheetu.

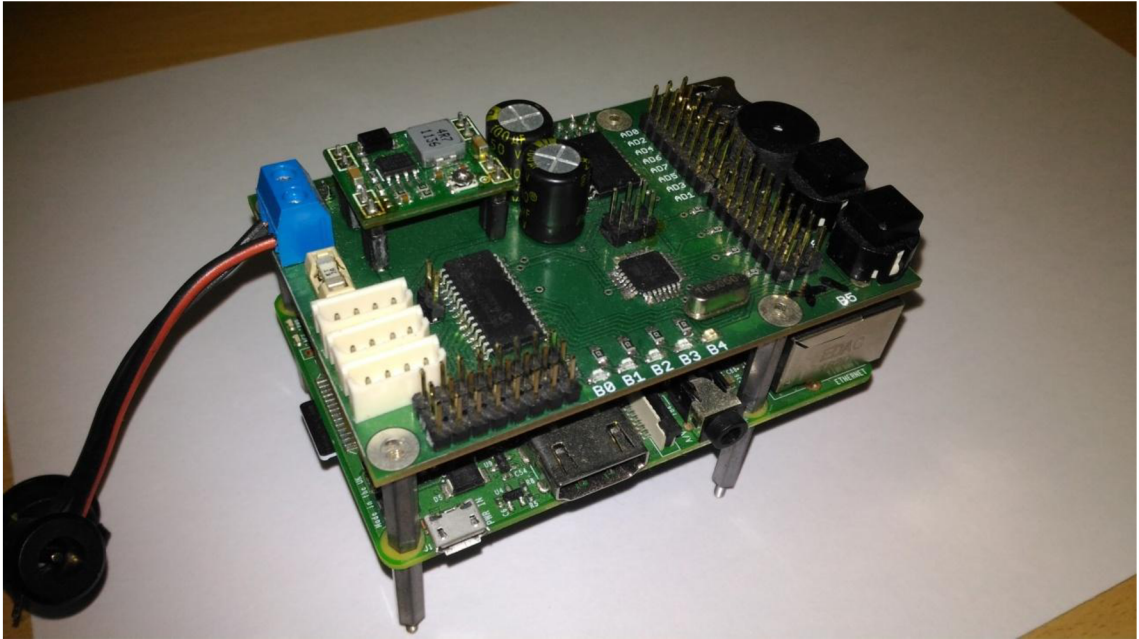
Deska byla od počátku navrhovaná jako dvouvrstvá s prokovama. Deska byla vyrobena ve firmě Apama s.r.o. Brno. V příloze 3 je zobrazena TOP a BOTTOM strana vyrobené desky.

Na obr 23 je zobrazena osazená deska připevněna na RaspberryPi. Pak na Obr 24 je vidět testování LED diod umístěných na této desce.

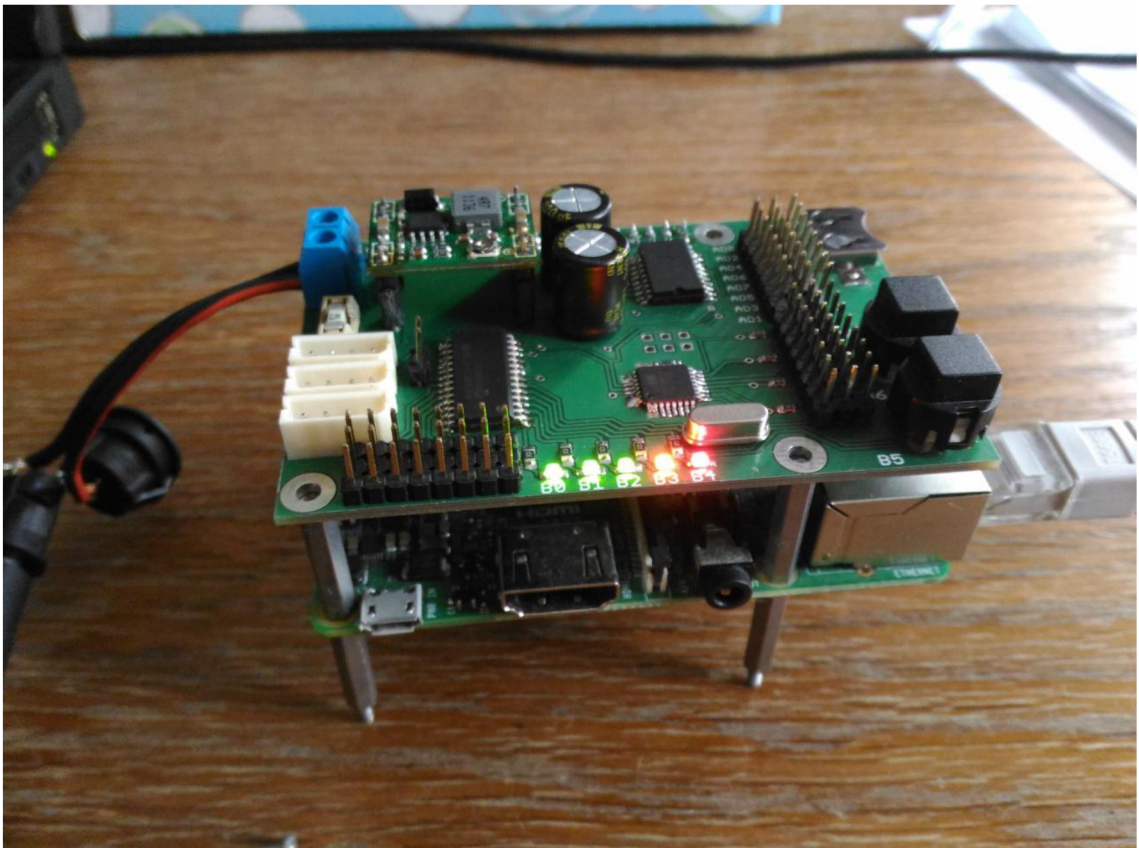
Deska byla navržena asi za týden. Nejdělsí čas zabralo shromažďování součástek, protože byly nakupovány z různých internetových e-shopu a nebyly často skladem. Následné zapájení a oživení desky zabralo jeden den. Předtím probíhalo měření na DC/DC měniči.

Velká výhoda desky je, že její velikost odpovídá velikosti RaspberryPi. Je možné desku umístit přímo nad RaspberryPi, jak je vidět na obr 23 a 24.

Důležitým ochranným prvkem na desce jsou kromě pojistky na vstupu taky dvě zenerové diody, které jsou umístěny na I2C sběrnici. Ty slouží pro ochranu, kdyby bylo připojené zařízení s jinou napětíovou logikou.



Obr 23 - Osazená deska umístěna na RaspberryPi



Obr 24 - Testování připojení LED diod

## 8 OPRAVA CHYB DESKY

Avšak ani tento prototyp se neobešel bez chyb, které byly nalezeny při osazování a programování na desce. Nalezené chyby jsem opravil v další verzi desky, kterou příkládám jako přílohu na přiloženém CD.

Popis nalezených a opravených chyb ve verzi 2 desky:

Tranzistor jako spínací prvek pro bzučák. Nevhodně zvolená knihovna v eagle, pouzdro vývody nožiček nestandardní. Oprava provedena změnou knihovna. Na desce byl použit MOSFET IRMLM0060TRPbF.

Schéma bylo poskládáno z více zdrojů schémat. Při kontrolování LED diod v pouzdře 0805 bylo zjištěno, že jsou použity z různých knihoven. V opravené verzi jsem použil všechny LED diody z jedné knihovny.

Když už byly upravené pouzdra LED diod, tak taky všechny odpory na desce byly sjednoceny na pouzdra 0805. Na prototypu se u ultrazvukového senzoru nacházely odpory v pouzdře 0603 kvůli místu, ale je možné použít i větší pouzdro.

U bzučáku bylo potřeba přidat závěrně diodu a byl připojen na 5V.

Při osazování DC/DC měniče bylo zjištěno, že navržena knihovna má posunutou rozteč vývodů mezi vstupem a výstupem o 0,05 palce. Při měření na DC/DC měniči bylo zjištěno, že má nevhodně vyřešenou filtraci vstupního a výstupního napětí. S tohoto důvodu byly na desku přidány filtrační kondenzátory. Konkrétně 100nF/50V a 1nF/50V, které byly umístěny jak na výstup z DC/DC měniče, tak i z LDO stabilizátoru.

K AD-řevodníku musí být přidán odpor k volbě adresy na I<sup>2</sup>C sběrnici a to z důvodu, že GPIO expandér má adresové piny napevno uzemněné a tím mají pevně navolenou adresu.

Dále jsem na desku umístil informativní LED diodu, která signalizuje zapnutí napájení.

Při programování knihovny pro RaspberryPi, bylo jsem narazil na rozdílné ovládání LED diod a bzučáku. S tohoto důvodu byly na opravené desce otočeny LED diody polaritou opačně, aby bylo ovládání shodné s bzučákem a snadnější na programování. Dále také byla přidána možnost napájet hardwarově Pull-Up rezistor k tlačítkům. Aby byla možnost volby zda použít interní pull-up rezistor v GPIO expandéru nebo již zmíněné hardwarové.

## 9 ZÁVĚR

Cílem bakalářské práce bylo navrhnout a zhotovit rozšiřující desku pro platformu RaspberryPi. Nejprve bylo třeba zvolit a seznámit se se všemi integrovanými obvody, jež by měly být na rozšiřující desce. Poté byla potřeba napsat firmware pro mikrokontroler, jež je součástí desky, a také knihovnu pro RaspberryPi sloužící pro obsluhu periférií desky.

Po vhodných konzultacích byly nakonec vybrány obvody splňující stanovené podmínky. Byl použit vhodný AD převodník s označením AD7997BRU, sloužící pro čtení analogových hodnot pro další zpracování v digitální podobě. Tento převodník může být použit pro různé senzory, které se v robotice hodně používají, ať už na identifikaci vnitřních nebo identifikaci vnějších stavů robota.

Vhodně byl vybrán také GPIO-expandér, který obsahuje dva osmibitové porty. Jeden port je připojen na periférie, které jsou obsažené na desce. Jedná se o pět LED diod, dvě tlačítka a jeden bzučák. Proto byla pro RaspberryPi napsána knihovna k těmto perifériím, jejíž pomocí je možné je obsluhovat. Druhý port je vyveden do pinu header s roztečí 0.1 palce. Na ty je pak možné v logice 3.3V cokoliv připojit.

Mimoto je na desce také mikroprocesor, který primárně slouží pro obsluhu čtyř ultrazvukových senzorů.

V této práci byl napsán firmware, jež z mikroprocesoru udělá slave obvod na I<sup>2</sup>C sběrnici. V tomto mikroprocesoru pak běží program, který pomocí vnitřního čítače zabezpečuje měření šířky impulsu, který vrací senzor jakožto vzdálenost měřeného objektu.

Dále je na desce umístěn RTC(Real Time Clock, hodiny reálného času), zapojené k RaspberryPi pomocí SPI sběrnice. Dále se však v této práci s tímto obvodem nepracovalo.

Deska byla od počátku navrhována v programu Eagle 7.7.0 jako dvouvrstvá s prokovami. Výroba desky byla zabezpečena firmou AMAPA s.r.o. Brno. Po osazení, zapájení, oživení a naprogramování byly na desce objeveny různé chyby. Tyto chyby jsou popsány v poslední kapitole a následně jsou opraveny v druhé verzi této desky, která se nachází v příloženém CD.

Do budoucna by bylo možné na této desce udělat různé úpravy, které by vedly k vylepšení tohoto nápadu. Deska byla primárně navržena pro projekt KamBot na ušetření místa a lepší práci s tímto robotem pro studenta.

Tímto si dovoluji prohlásit, že jsem splnil všechny body zadání.

# Literatura

- [1] RaspberryPi. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-05-22]. Dostupné z: [https://cs.wikipedia.org/wiki/Raspberry\\_Pi](https://cs.wikipedia.org/wiki/Raspberry_Pi)
- [2] RaspberryPi2B. In: *Www.raspi.cz* [online]. online: raspi.cz, 2015 [cit. 2017-05-17]. Dostupné z: <http://www.raspi.cz/wpcontent/uploads/2015/02/raspberry-pi-2.jpeg>
- [3] Datasheet AD9779BRU[online]. b.r. [cit. 2017-05-22]. Dostupné z: [http://www.phys.hawaii.edu/~idlab/taskAndSchedule/HVprojPage/Datasheets/ADC-AD7997\\_7998.pdf](http://www.phys.hawaii.edu/~idlab/taskAndSchedule/HVprojPage/Datasheets/ADC-AD7997_7998.pdf)
- [4] Datasheet MCP23017[online]. b.r. [cit. 2017-05-22]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf>
- [5] Datasheet DS3234[online]. b.r. [cit. 2017-05-22]. Dostupné z: <https://www.sparkfun.com/datasheets/BreakoutBoards/DS3234.pdf>
- [6] Datasheet ATmega88PA[online]. b.r. [cit. 2017-05-22]. Dostupné z: [http://www.atmel.com/images/Atmel-9223-Automotive-Microcontrollers-ATmega48PA-ATmega88PA-ATmega168PA\\_Datasheet.pdf](http://www.atmel.com/images/Atmel-9223-Automotive-Microcontrollers-ATmega48PA-ATmega88PA-ATmega168PA_Datasheet.pdf)
- [7] HC SR04. In: *Https://www.upgradeindustries.com* [online]. : [cit. 2017-05-22]. Dostupné z: [https://www.upgradeindustries.com/media/img/hi\\_res/hcsr04\\_hires.jpg](https://www.upgradeindustries.com/media/img/hi_res/hcsr04_hires.jpg)
- [8] Ultrazvukovy senzor. In: *Www.tajned.cz* [online]. online: tajned.cz, 2014 [cit.2017-05-22].Dostupné z: <http://www.tajned.cz/2014/06/ultrazvukovy-modul-hc-sr04/>
- [9] KM2 firmware. In: *Github Brno UAMT* [online]. online: github, 2016 [cit. 2017-05-22]. Dostupné z: <https://github.com/uamt-brno/RaspBot/tree/master/src/KM2>
- [10] RaspberryPi OS. In: *Http://www.hongkiat.com* [online]. online: hongkiat, 2016 [cit. 2017-05-22]. Dostupné z: <http://www.hongkiat.com/blog/pi-operating-systems/>
- [11] RaspberryPi wifi AdHoc. In: *Https://www.raspberrypi.org/* [online]. online: <https://www.raspberrypi.org/>, 2016 [cit. 2017-05-22]. Dostupné z: <https://www.raspberrypi.org/forums/viewtopic.php?f=29&t=39927>
- [12] Library I2C.h a AD799X.h. In: *github -uamt-brno* [online]. online: github-uamt-brno, 2016 [cit. 2017-05-22]. Dostupné z: <https://github.com/uamt-brno/RaspBot/tree/master/src/RaspberryPi/shared>

# Seznam zkratek

AD- Analog Digital

GPIO - General-purpose input/output

RTC- Real Time Clock (Hodiny Reálného času)

LED - Light-Emitting Diode (dioda emitující světlo)

PWM - Pulse Width Modulation (Pulzní šířková modulace)

EEPROM - Electrically Erasable Programmable Read-Only Memory (Elektronicky Vymazatelná Paměť pouze pro čtení)

SRAM - Static Random Access Memory (Statická paměť s náhodným přístupem)

FLASH – Flash paměť je elektricky programovatelná paměť s libovolným přístupem.

LDO - Low drop (stabilizátor s velmi malým úbytkem napětí)

I2C - Inter-Integrated Circuit (sériová sběrnice)

SCL - Synchronous Clock (synchronizační hodiny)

SDA - Synchronous Data (synchronní data)

# Seznam příloh

Příloha 1. Schémata

Příloha 2. Deska TOP, BOTTOM a společně vygenerované.

Příloha 3. Fotografie neosazené desky

Příloha 4. Fotografie osazené desky

Příloha 5 CD, které obsahuje:

Eagle data vyrobeného prototypu desky

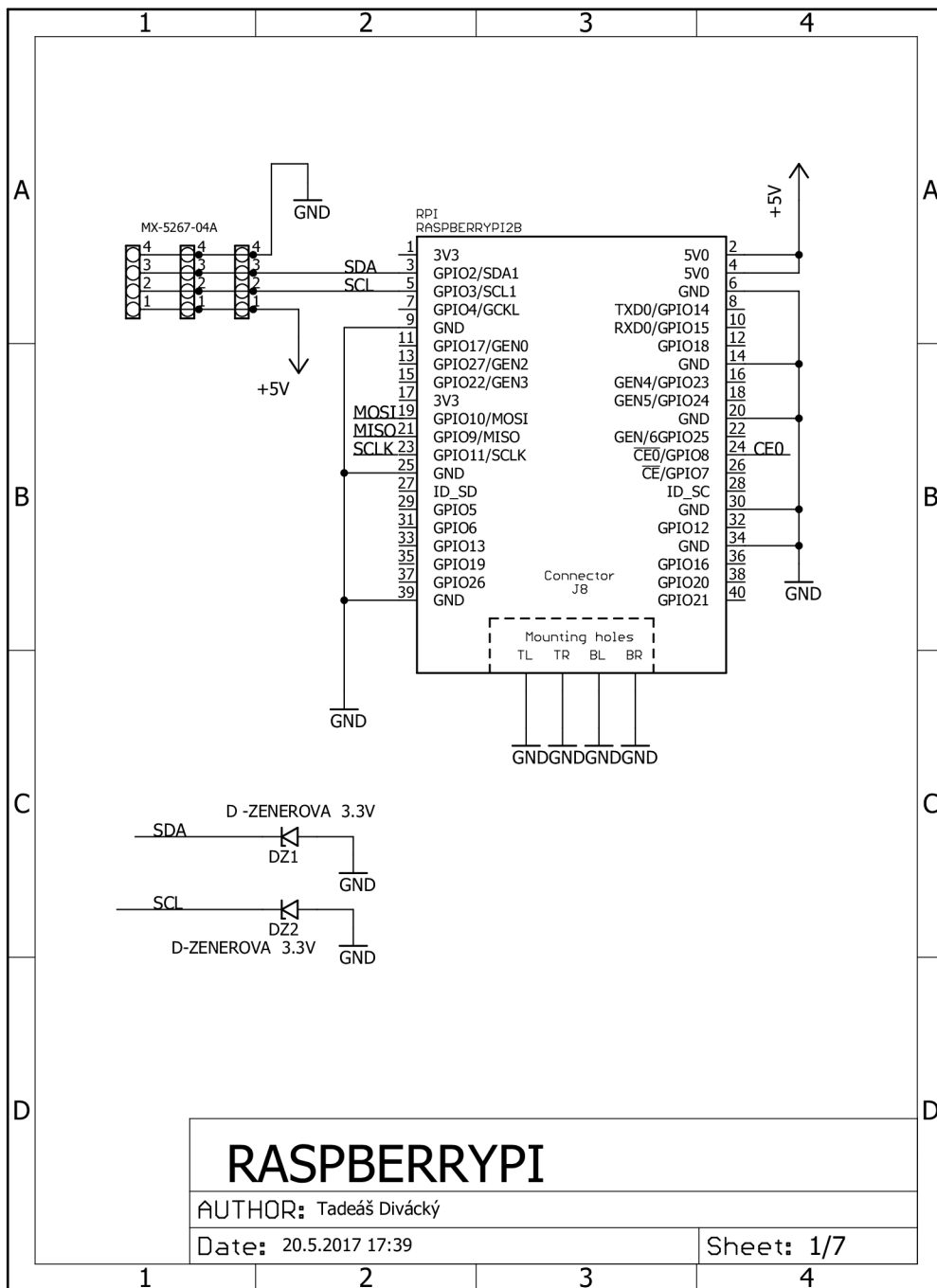
Eagle data desky s opravenými chybami

Soubor main.h s firmwarem pro ATmega88PA

Projekt pro RaspberryPi pro obsluhu desky

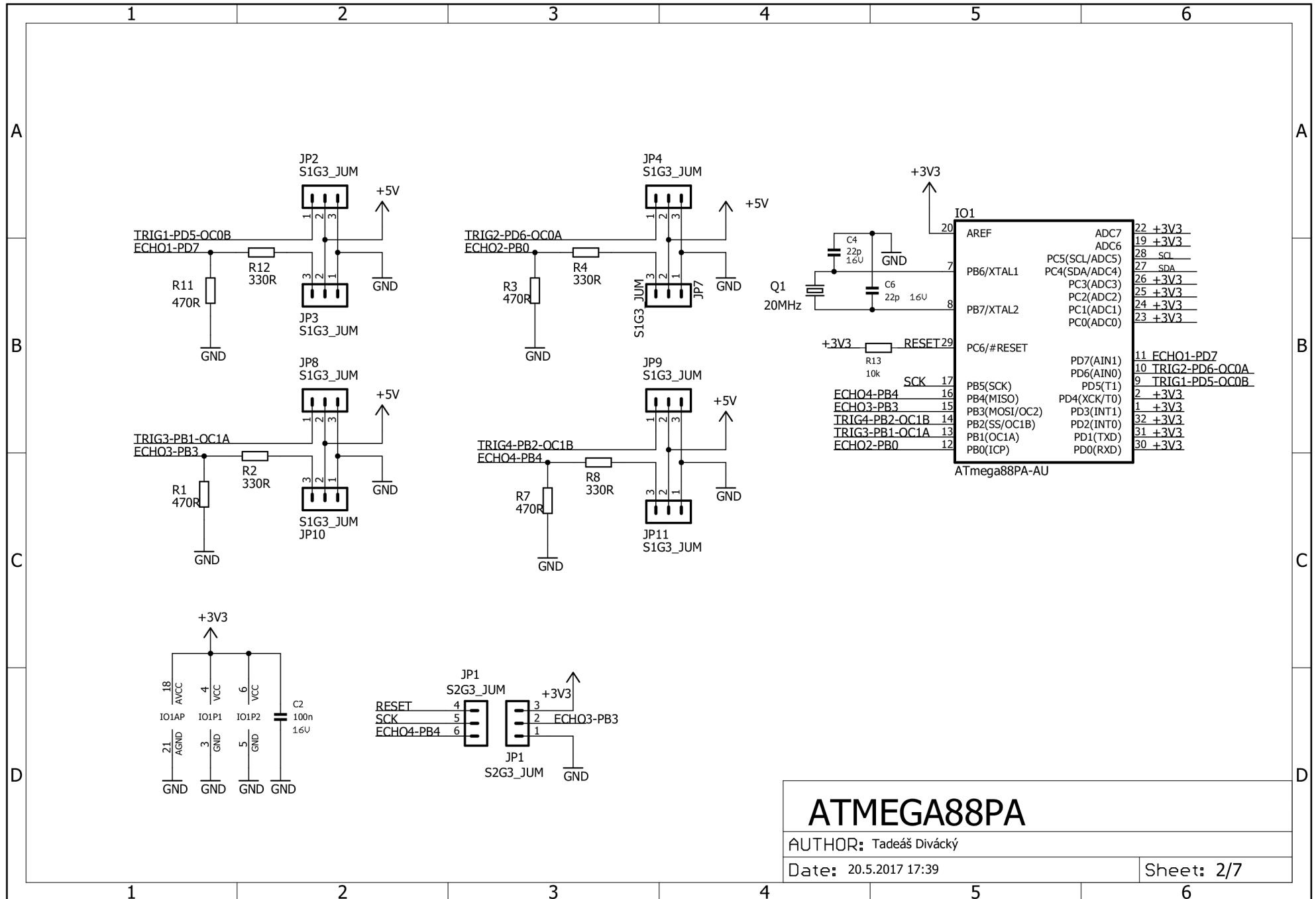
Práce v digitální podobě

# Příloha 1-A





# Příloha 1-B



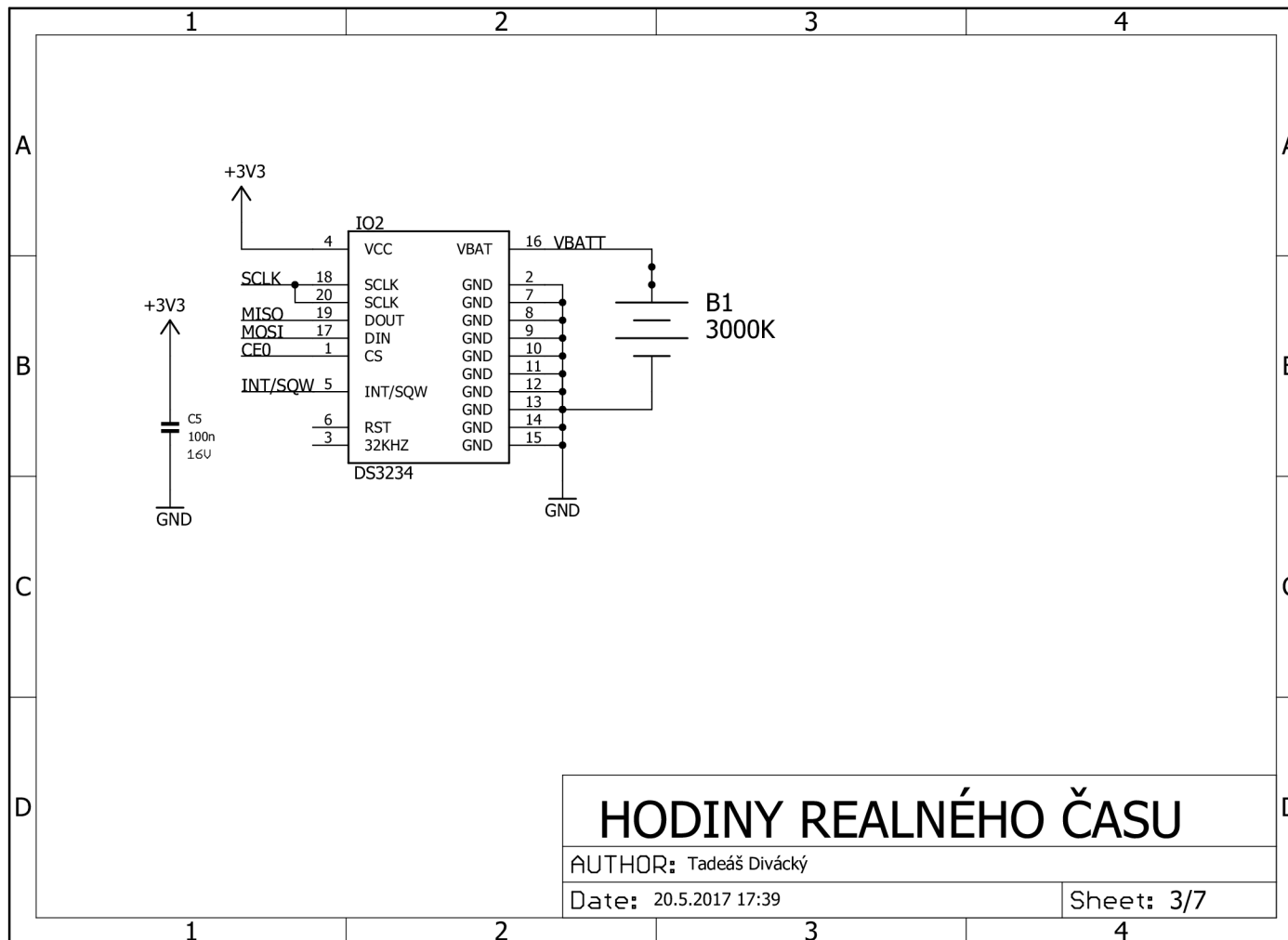
## ATMEGA88PA

AUTHOR: Tadeáš Diváček

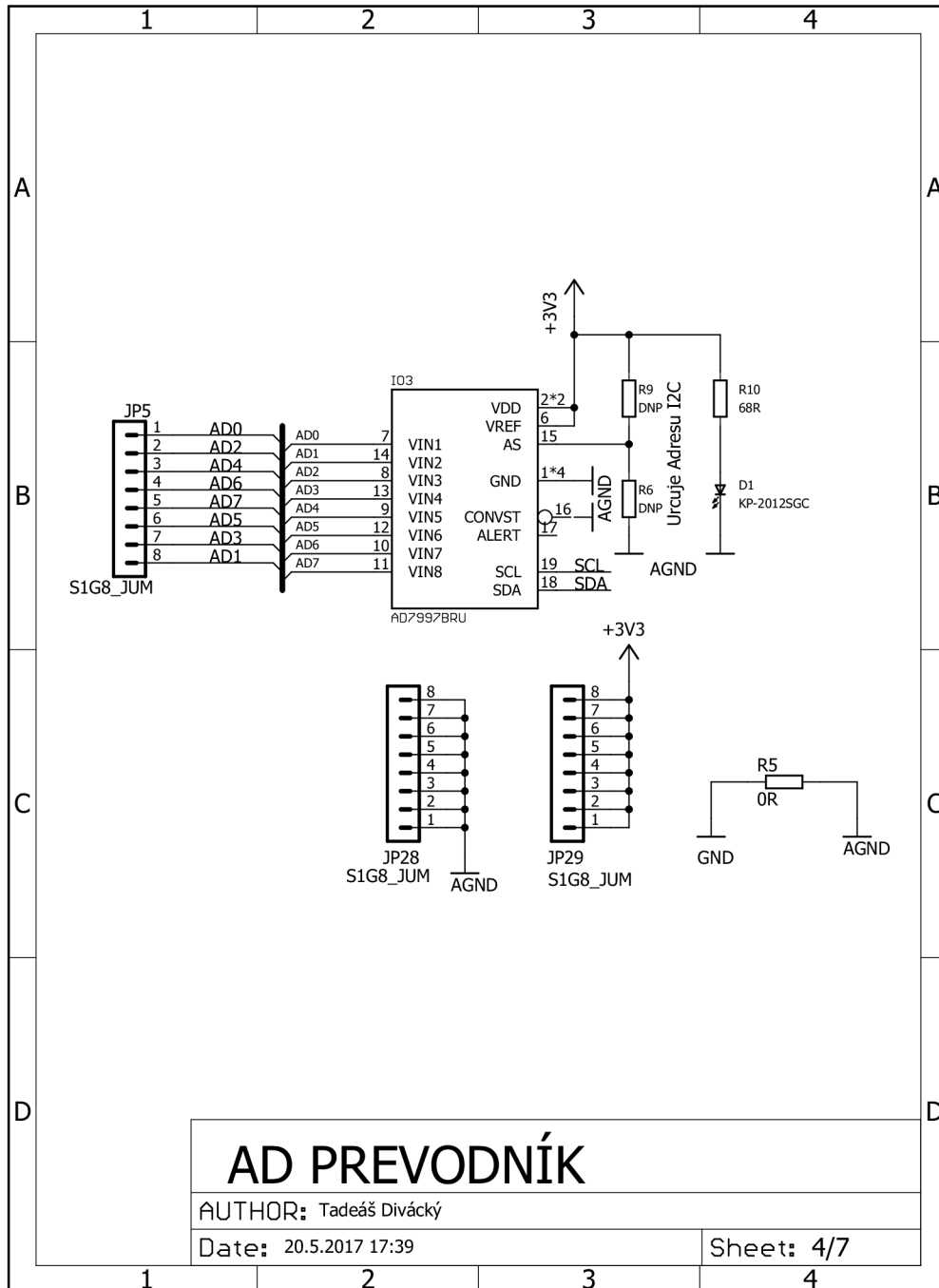
Date: 20.5.2017 17:39

Sheet: 2/7

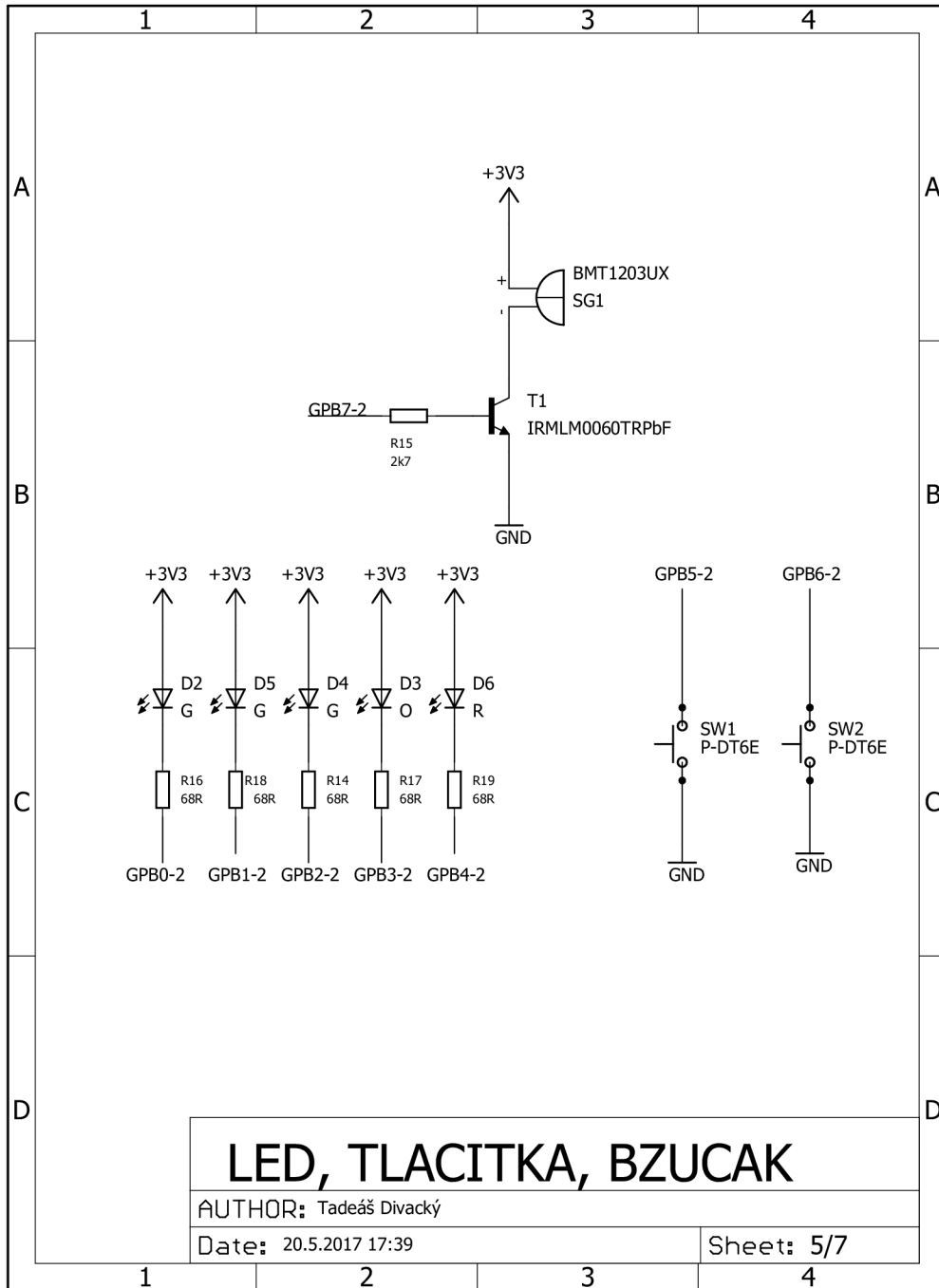
Příloha 1-C

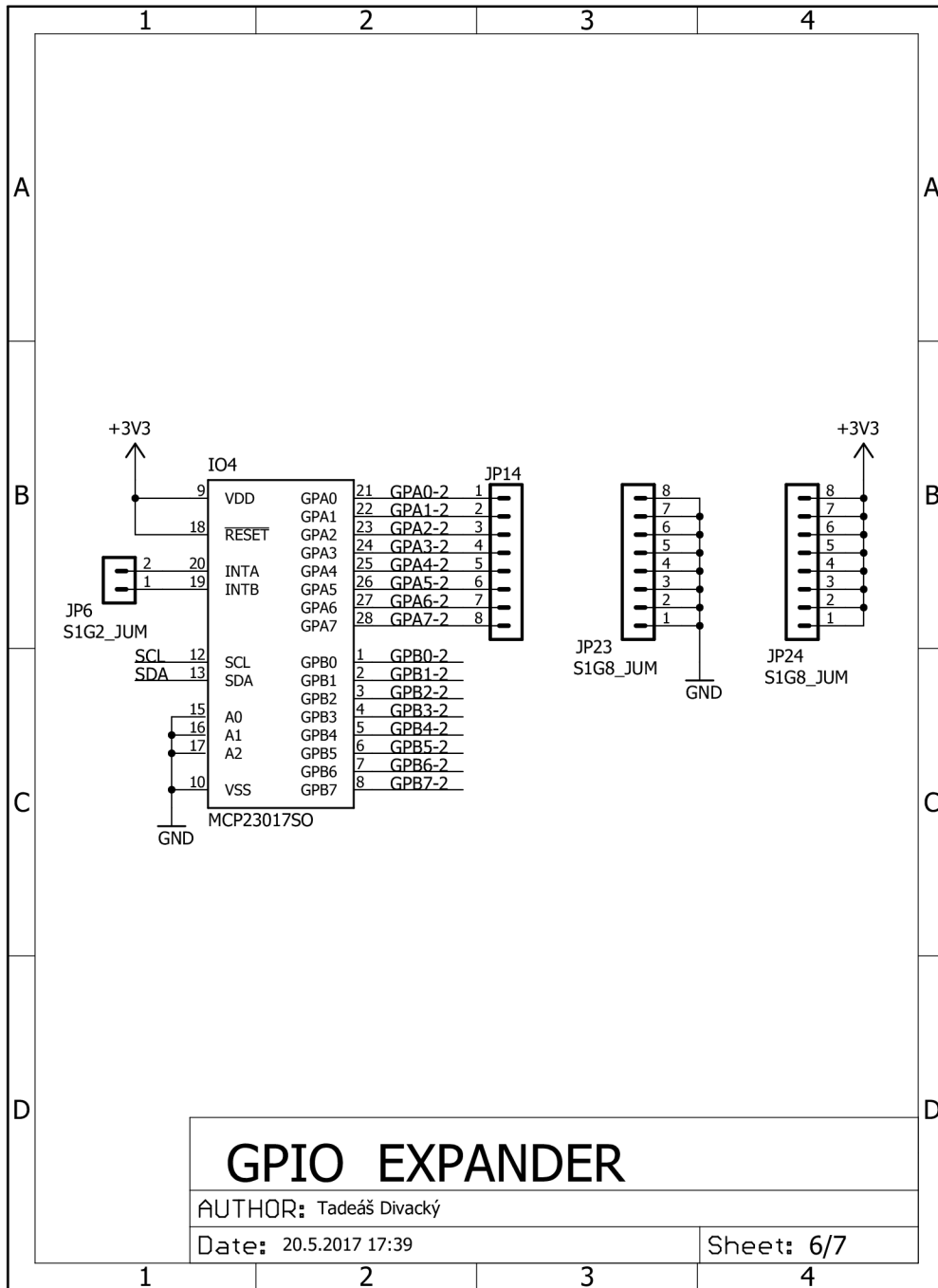


Příloha 1-D

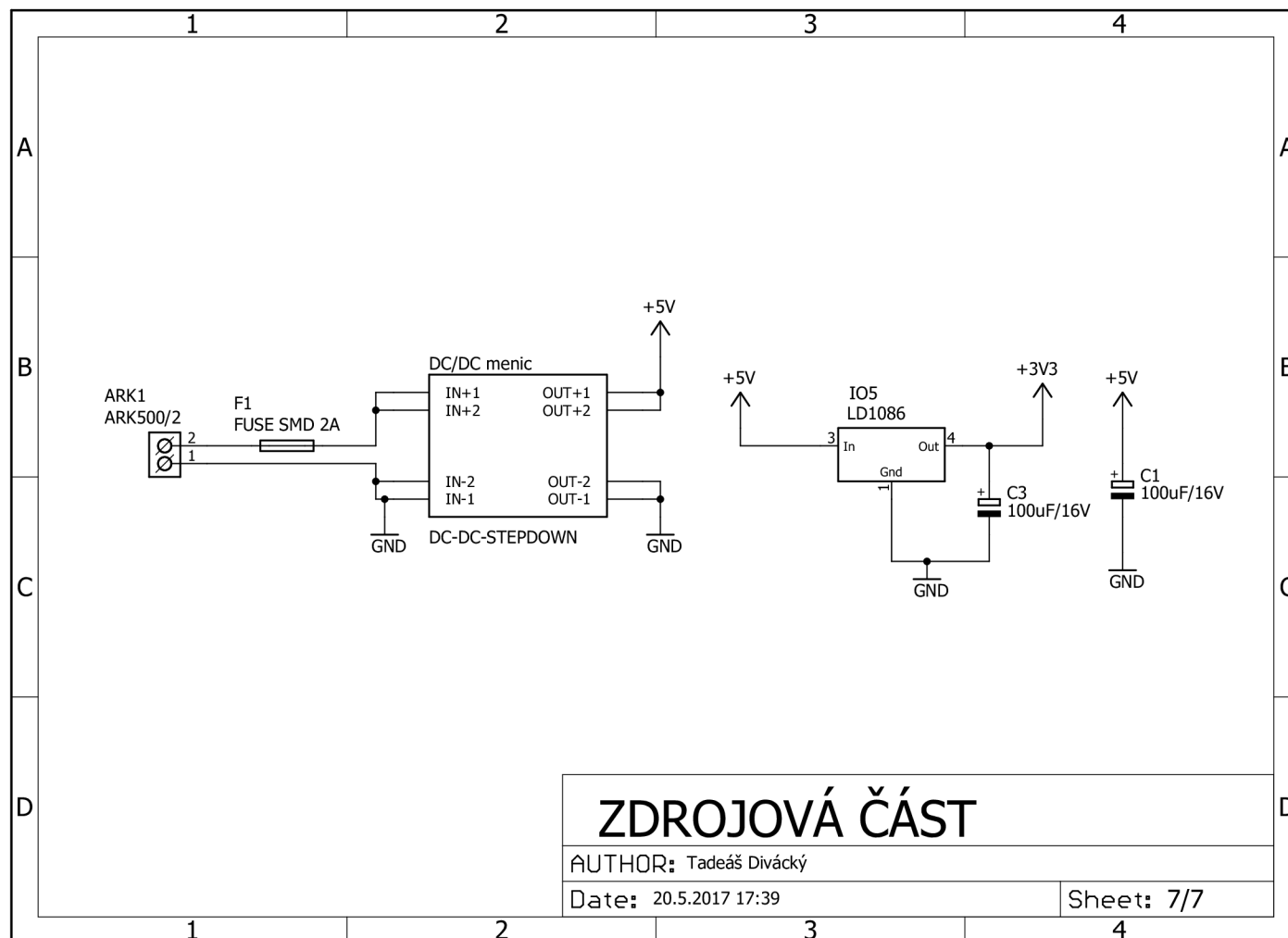


# Příloha 1-E

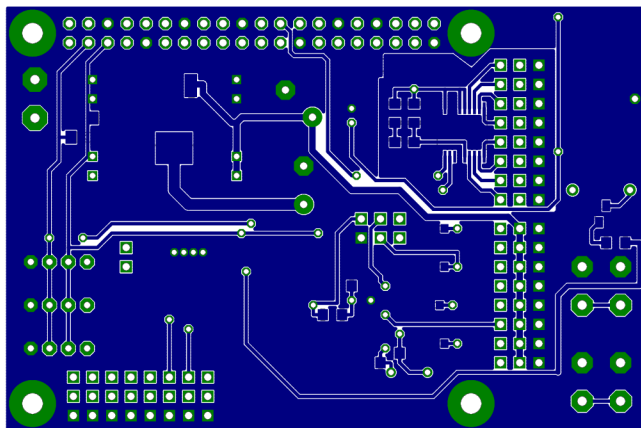




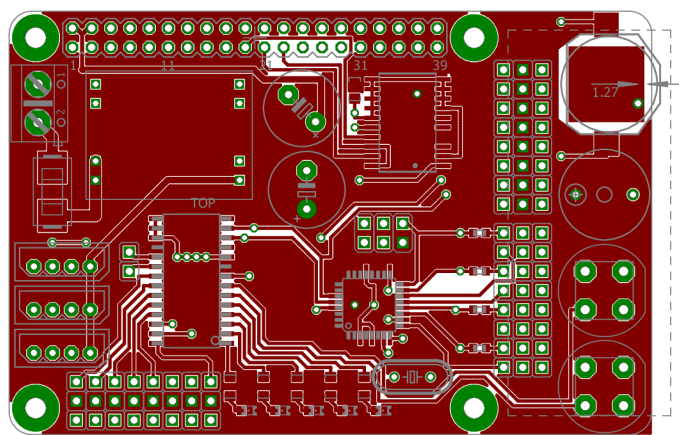
# Příloha 1-G



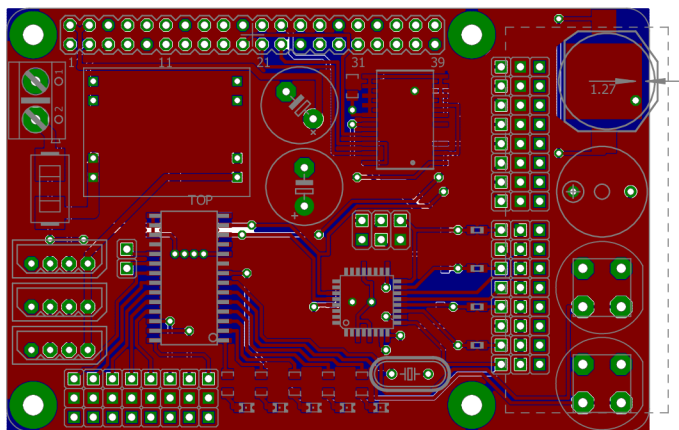
## Příloha 2



DPS BOTTOM strana

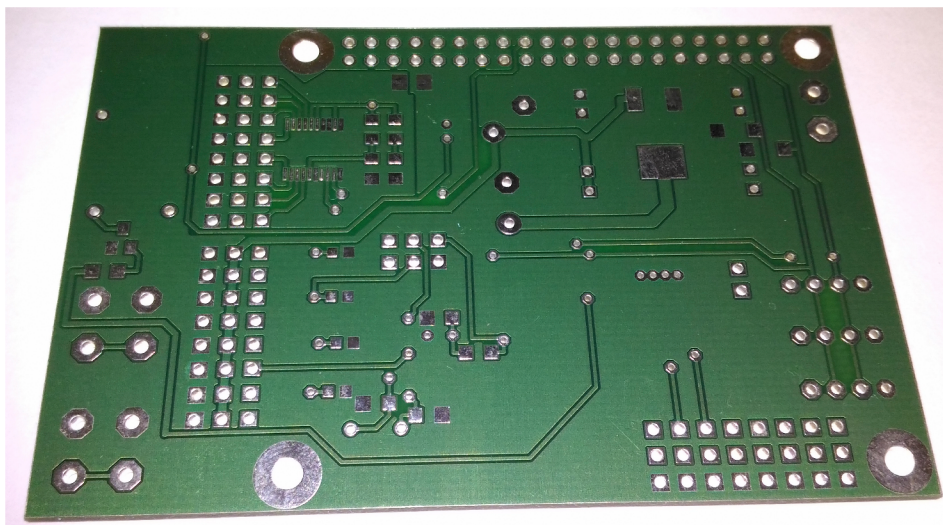


DPS TOP strana

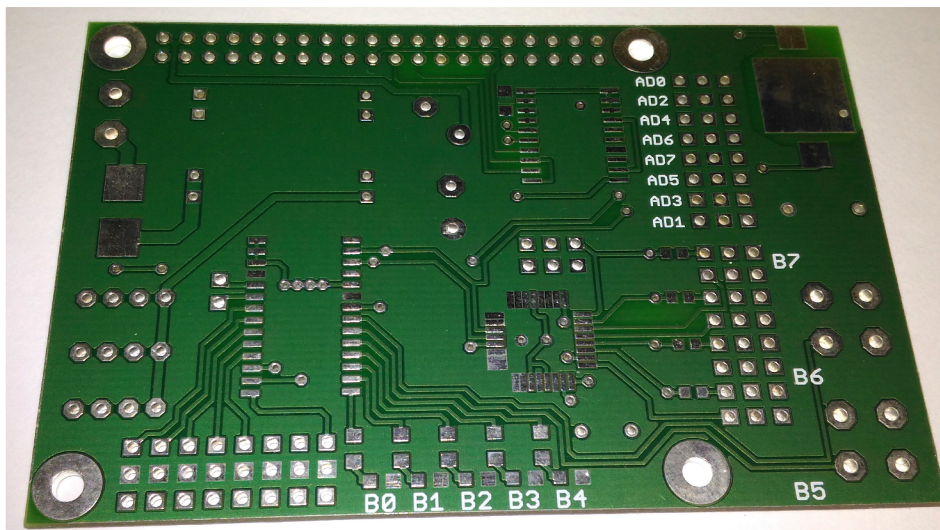


DPS TOP a BOTTOM strana

Příloha 3



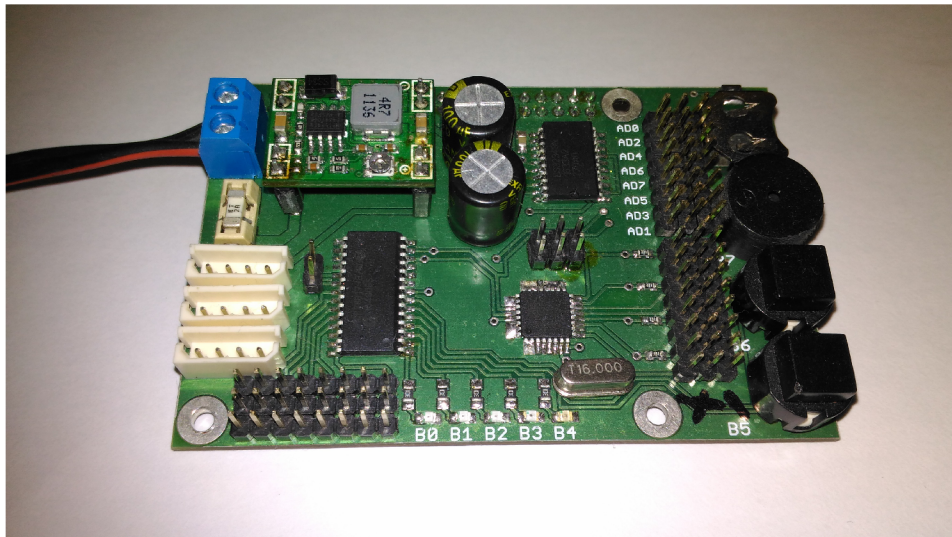
Fotografie Bottom strany desky



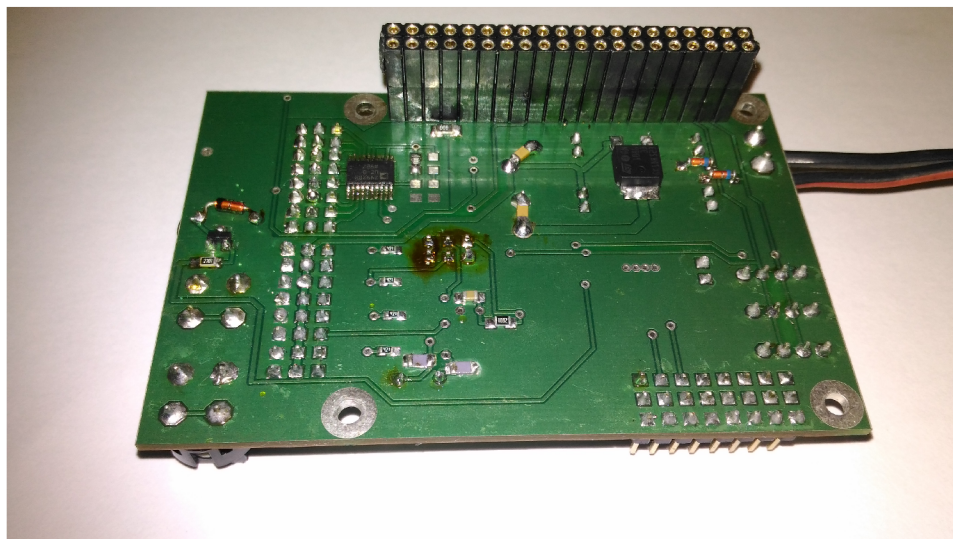
Fotografie Top strany desky



Příloha 4



Fotografie osazené Top strany desky



Fotografie osazené Bottom strany desky