

ČESKÉ  
VYSOKÉ  
UČENÍ  
TECHNICKÉ  
V PRAZE

**Fakulta elektrotechnická**

**Katedra měření**

**Diplomová práce**

# **Universální PLC kontrolér s HDMI výstupem**

**Bc. Lukáš Líbal**

**Květen 2015**

**Vedoucí práce: Ing. Radek Sedláček, Ph.D.**





## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Lukáš Líbal**

Studijní program: **Inteligentní budovy**

Název tématu česky: **Univerzální PLC kontrolér s HDMI výstupem**

Název tématu anglicky: **General-purpose PLC Controller with HDMI Output**

### Pokyny pro vypracování:

Navrhněte a realizujte programovatelný logický automat s HDMI výstupem pro připojení externího zobrazovače pro zobrazování procesních hodnot. Pro začlenění do stávajících systémů elektroinstalace PLC osadte následujícími perifériemi: 2 analogovými vstupy (rozsah 10 V), 2 analogovými výkonovými výstupy (48V/2A) pro řízení akčních členů (např. krokových motorů), 20 galvanicky oddělenými digitálními vstupy a výstupy, komunikačním rozhraním Ethernet 10/100 a RS-485. Hardwarové řešení PLC založte na použití některé platformy mini-PC (Raspberry PI či BeagleBone). Potřebné periférie, které nebudou dostupné přímo na zvolené platformě mini-PC, realizujte formou rozšiřující karty. Vytvořte ovladače pro jednotlivé periférie a grafickou knihovnu pro HDMI výstup. Jednoduchým ukázkovým kódem demonstруйте funkčnost všech periférií včetně HDMI rozhraní.

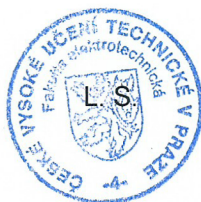
### Seznam odborné literatury:

- [1] Raspberry Pi [online] [cit. 2014-09-11]. Dostupné z: <http://www.raspberrypi.org/>
- [2] Beagleboard.org [online] [cit. 2014-09-11]. Dostupné z: <http://beagleboard.org/bone>
- [3] Dobkin, Bob Williams, Jim: Analog Circuit Design - A Tutorial Guide to Applications and Solutions. Elsevier. 2011. ISBN: 978-0-12-385185-7.

Vedoucí diplomové práce: Ing. Radek Sedláček, Ph.D. (K 13138)

Datum zadání diplomové práce: 11. září 2014

Platnost zadání do<sup>1</sup>: 29. ledna 2016



Doc. Ing. Jan Holub, Ph.D.  
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 11. 9. 2014

<sup>1</sup> Platnost zadání je omezena na dobu tří následujících semestrů.





## **Prohlášení**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 11. 5. 2015



## **Poděkování**

Na úvod této práce bych chtěl zejména poděkovat vedoucímu mé práce Ing. Radku Sedláčkovi, Ph.D. za trpělivý a vstřícný přístup.

## Abstrakt

Práce se zabývá návrhem a realizací jednoduchého programovatelného automatu s grafickým výstupem založeným na Raspberry Pi.

Vlastní programovatelný automat je založený na mikrokontroléru z řady STM32F0. Zařízení je navrženo jako zapouzdřený celek, disponující běžně používanými periferiemi. Včetně všech potřebných napájecích zdrojů.

Testování funkčnosti je provedeno na modelu Thompsonova-Lampardova etalonu kapacity. Ovládání je realizováno pomocí grafického uživatelského rozhraní.

## Abstract

This thesis deals with the design and implementation of a simple programmable controller with graphical output based on the Raspberry Pi.

Own programmable logic controller is based on a microcontroller of STM32F0 family. The device is designed as an encapsulated unit, which has a commonly used peripheries. Including all the necessary power supplies.

Functional testing is performed on a model Thompson-Lampardo's standard capacity. The control is realized by using a graphical user interface.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Průmyslová řešení</b>	<b>2</b>
2.1	PLC MPC223AC . . . . .	3
2.2	Loxone Miniserver . . . . .	4
2.3	Vision430 . . . . .	5
<b>3</b>	<b>Návrh hardware</b>	<b>7</b>
3.1	RaspberryPi . . . . .	8
3.1.1	HDMI . . . . .	9
3.2	Mikrokontrolér . . . . .	10
3.3	RS-485 . . . . .	12
3.4	Analogový výkonový výstup . . . . .	13
3.5	Analogové vstupy . . . . .	15
3.6	Digitální vstupy a výstupy . . . . .	17
3.7	Napájení . . . . .	18
3.8	Mechanická část . . . . .	20
<b>4</b>	<b>Grafický výstup</b>	<b>21</b>
4.1	Inteligentní displeje . . . . .	21
4.1.1	4D Systems PICASO . . . . .	22
4.1.2	FTDI EVE . . . . .	23
4.2	Raspberry Pi jako zobrazovač . . . . .	23
4.2.1	Instalace a konfigurace . . . . .	24
<b>5</b>	<b>Testování</b>	<b>26</b>
5.1	Ovládací aplikace . . . . .	27



5.1.1	Qt framework . . . . .	28
5.1.2	Grafická podoba . . . . .	28
5.1.3	Logika aplikace . . . . .	29
5.2	Program pro mikrokontrolér . . . . .	31
5.3	Měření . . . . .	33
<b>6</b>	<b>Návrh univerzální knihovny</b>	<b>37</b>
6.1	Rozšiřující deska . . . . .	37
6.2	RaspberryPi . . . . .	39
<b>7</b>	<b>Závěr</b>	<b>41</b>
	<b>Literatura</b>	<b>43</b>
	<b>Seznam použitých zkratk</b>	<b>45</b>
	<b>Přílohy</b>	<b>47</b>
A	Výkresy . . . . .	47
B	Mechanické provedení . . . . .	49
C	Schémata zapojení . . . . .	59

## Seznam obrázků

1	PLC (převzato z [1]) . . . . .	2
2	MICROPEL PLC MP223AC (převzato z [2]) . . . . .	4
3	Loxone Miniserver (převzato z [3]) . . . . .	5
4	Vision430 (převzato z [4]) . . . . .	6
5	Blokové schéma návrhu . . . . .	7
6	Raspberry Pi model B (převzato z [5]) . . . . .	8
7	Jádro ARM Cortex M0 (převzato z [6]) . . . . .	11
8	Schéma ADM2582E (převzato z [7]) . . . . .	12
9	Blokové schéma L6205 (převzato z [8]) . . . . .	14
10	Schéma zapojení H-můstku . . . . .	14
11	Blokové schéma A/D kanálu . . . . .	15
12	Schéma zapojení vstupních obvodů A/D kanálu . . . . .	16
13	Schéma zapojení filtru a převodníku A/D kanálu . . . . .	17
14	Zapojení izolovaných vstupů s výstupy . . . . .	17
15	Schéma napájení pro digitální část . . . . .	19
16	Schéma napájení pro analogovou část . . . . .	19
17	4D Systems inteligentní display (převzato z [9]) . . . . .	22
18	CooCox IDE . . . . .	26
19	ST discovery kit (převzato z [10]) . . . . .	27
20	Aplikace v Qt Creatoru . . . . .	29
21	Řízení krokového motorku s polovičním krokem (Převzato z [11]) . . . . .	32
22	Model elektrod etalonu (převzato z [12]) . . . . .	33
23	Testování aplikace . . . . .	36



## Kapitola 1

# Úvod

Programovatelný logický automat (PLC) je zařízení v sobě sdružující potřebnou logiku k provádění uživatelských funkcí. Jedná se o jednoduchý průmyslový počítač, který automatizuje potřebné procesy. Poskytuje všeobecně volně konfigurovatelnou programovou funkčnost, kterou je kdykoliv možno jednoduše rekonfigurovat dle aktuálních potřeb systému, které vycházejí z místa nasazení. Tedy dnes v různých podobách dnes proniká od všech míst kolem nás v každodenním životě.

Funkčnost je omezena dostupnými perifériemi, kterými daný automat disponuje. Tedy existují různé konfigurace periférií systémů do různých oblastí nasazení. Nejčastějšími perifériemi jsou obyčejné digitální vstupy, výstupy a taktéž v analogové formě. Jednotky jsou často omezeny vlastními přímými perifériemi, ale na konkrétní případ nasazení lze řešit pomocí dalších rozšíření, pokud PLC disponuje danou konektivitou.

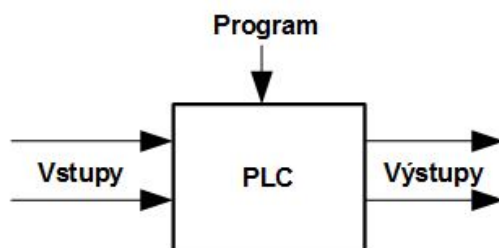
Cílem této práce je navrhnout obdobný jednoduchý univerzální programovatelný logický automat, který bude disponovat běžnými perifériemi pro všeobecné využití. Bude danou konektivitou pro jeho možné rozšíření o případné další potřebné moduly. A taktéž přímým grafickým výstupem na běžně užívané displeje disponující HDMI či DVI kompatibilním rozhraním.

## Kapitola 2

# Průmyslová řešení

Programovatelný logický automat je zmenšený průmyslový počítač specializující se na vykonávání operací v daném čase. Z toho vyplývá, že se využívá ve všech automatizovaných výrobních procesech. Setkáváme se s nimi ve také všech oblastech, kde je potřeba provádět automatické akce, tedy principiálně při měření a následné regulaci. Vyjma těch případů, kdy se jedná o jednoúčelová zařízení, která už neposkytují možnost programování.

Základem těchto automatů je speciální podoba mikrokontrolérů využívajících programovatelné paměti k uložení instrukcí a implementace funkcí např. logických, sekvencních, časových a dalších k řízení strojů a procesů. Základní model funkčnosti je na obrázku 1.



Obrázek 1: PLC (převzato z [1])

PLC má velkou výhodu, že jeden shodný základní mikrokontrolér může být použitý pro široké spektrum řídicích systémů. Je-li potřeba měnit systém a pravidla řízení, vše co je nutné udělat je nahrát jiné instrukce. Není potřeba dělat žádný zásah do zapojení. Výsledkem jsou flexibilní a cenově dostupné systémy řízení, které se dost liší v jejich povaze a složitosti [1].

Samotné automaty můžeme rozdělit podle jejich vlastní komplexnosti, která je

dána funkcionalitou jimi poskytovanou. Tato komplexnost vychází např. z logických funkcí jak moc složité mohou být sestavovány, dále jejich časové závislosti a podobně. Nejzásadnější je periferní vybavenost a jejich specializace ke konkrétnímu účelu. Nejobecnějším měřítkem může být počet vstupů, výstupů a kompaktnost provedení.

Dle těchto parametrů můžeme rozdělit PLC do těchto skupin

**Kompaktní** - Jde o provedení zařízení na DIN lištu s daným počtem I/O, který lze rozšiřovat omezeně. Všechny I/O mají pevně danou funkčnost, která nelze překonfigurovat. Obsahují základní periferie v podobě již zmíněných I/O v digitální a analogové podobě a nějaké další komunikační rozhraní typu RS-232 či Ethernetu. V této podobě disponují základní světelnou signalizací nebo jednoduchým znakovým displejem. Nejčastější uplatnění tohoto provedení se nalézá v oblasti Inteligentních elektroinstalací a všeobecně Měření a regulace.

**Modulární** - Toto provedené je vlastně variabilní obdobou kompaktních systémů. Jsou založeny na základním modulem, který je možné volitelně dle nabídky dalších různě zaměřených modulů rozšiřovat. Tím je umožněno vyvíjet různé specializované moduly samostatně a nasadit je pouze tam kde je třeba. Jejich zaměření je tedy jasně dané do automatizovaných provozů, kde jsou potřeba speciální funkce a také třeba požadavky na zvýšenou bezpečnost zálohování.

Tímto by jsme měli dané základní rozdělení podle provedení, možnosti rozšiřování a zaměření na oblast využití. Nyní se podíváme na některé vybrané zástupce s různým zaměřením.

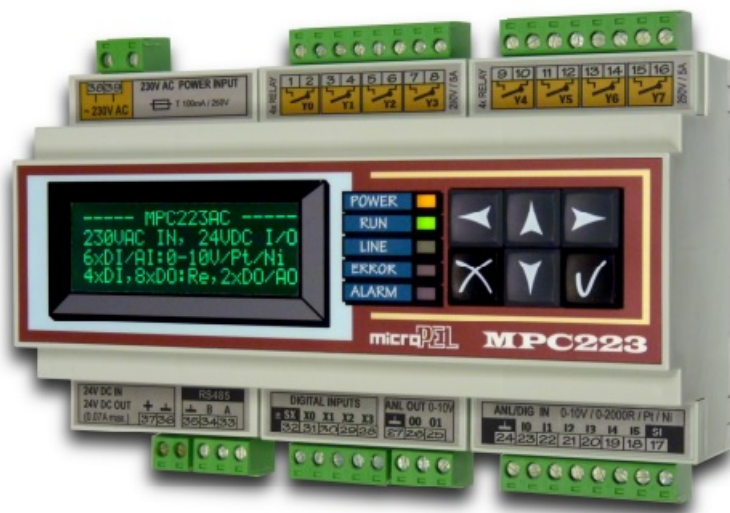
## ■ 2.1 PLC MPC223AC

V tomto případě se jedná o zástupce kompaktního provedení PLC od firmy MICRO-PEL. Kompaktní automat s pevnou konfigurací I/O s vlastnosti určenými zejména v aplikacích měření a regulace. Možnost použití tedy není vázané na jeden konkrétní typově kompatibilní regulovaný systém a tento automat je funkčně volně konfigurovatelný.

## 2.2 LOXONE MINISERVER

Mezi jeho konektivitu patří následující [2].

- 6x univerzální vstup digitální/analogový
  - pro měření napětí 0-10 V
  - pro měření odporu 0..2500 R
  - pro čidla Pt100: -200 ° ..+850 °C
  - pro čidla Pt1000: -200 ° ..+400 °C
  - pro čidla Ni1000: -150 ° ..+250 °C
- 4x digitální výstup, galvanicky oddělený
- 8x reléový výstup
- 2x univerzální výstup digitální/analogový 0-10 V
- alfanumerický displej 4x20 znaků
- 6x ovládací tlačítka



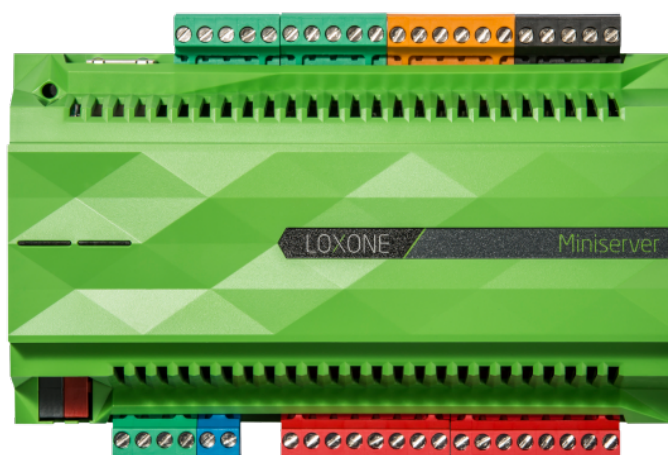
Obrázek 2: MICROPEL PLC MP223AC (převzato z [2])

## 2.2 Loxone Miniserver

Jedná se o modulární PLC od firmy Loxone Electronics určený pro inteligentní elektroinstalace. Vedle hlavního řídicího automatu nazvaného Miniserver existují rozšiřující

moduly, které lze nezávisle přidávat. Jejich zaměření je cílené do elektroinstalace, multimédií ale taktéž v sobě zahrnuje měření a regulaci nebo jen čistě digitální I/O. Pro zobrazování a řízení jsou dostupné služby webového serveru a aplikací na chytrá zařízení. Základní vlastnosti miniserveru jsou následující [3].

- 8 digitálních vstupů a výstupů
- 4 analogové vstupy a výstupy
- Ethernet
- KNX/EIB
- mnoho rozšiřujících modulů



Obrázek 3: Loxone Miniserver (převzato z [3])

## ■ 2.3 Vision430

Vision430 je řada OPLC (PLC + HMI) integrované v jednom pouzdře od firmy Uni-tronics. Jde o řadu systému disponujícím dotykovým displejem o velikosti 4,3 palce a rozlišením 490 x 272 pixelů. Dle konkrétního modelu je dostupná konektivita k běžně užívaným komunikačním rozhraním a konfigurace vstupů snímačů. Je zde možnost všechny modely rozšířit o modulové prvky. Tvorba ovládacího rozhraní je prováděna pomocí nástroje, který ulehčuje jeho tvorbu [4].

## 2.3 VISION430

Parametry konkrétního modelu V430J-RA22 jsou:

- 12 digitálních vstupů
- 2 analogové vstupy a výstupy
- 1 vysokorychlostní vstup (30KHz)
- 2 vstupy čidla Pt100
- 8 digitálních reléových výstupů
- RS-232/RS-485, USB, CANbus



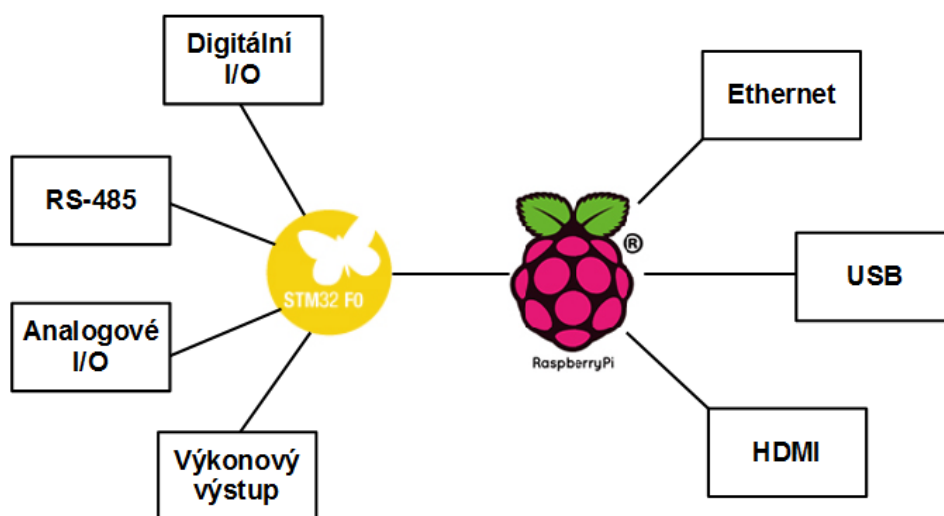
Obrázek 4: Vision430 (převzato z [4])



## Kapitola 3

# Návrh hardware

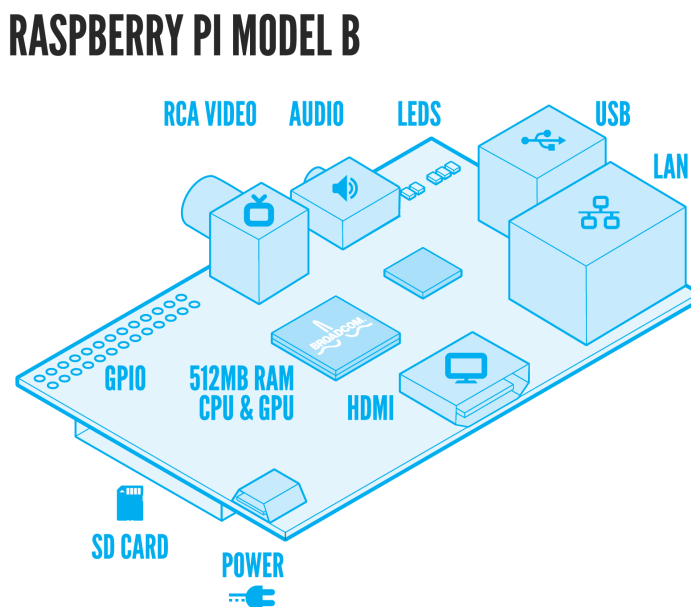
Základní požadavky na celý programovatelný automat nám poskytne blokové schéma, pomocí něhož zachytíme jednotlivé navrhované logické bloky. Na tyto abstrahované bloky uplatníme konkrétní požadavky na jejich vlastnosti, které aplikujeme na konkrétní obvody a jejich celkové zapojení ve schématu. Na obr. [5] jsou vidět jednotlivé funkční části námi navrhovaného logického programovatelného automatu, které si jednotlivě popíšeme v obvodové realizaci. Stejně tak již od počátku musíme brát zřetel na zapouzdření celého systému a od toho odvíjet konkrétní návrh plošného spoje.



Obrázek 5: Blokové schéma návrhu

## ■ 3.1 RaspberryPi

Pro realizaci jedné části našeho automatu byla vybrána platforma RaspberryPi model B. Jež bude využita zejména jako grafický procesor, ovládací prvek a poskytne nám další konektivitu, kterou disponuje. Základem celého mini-PC o velikosti o něco málo větší než je platební karta je systém na čipu (SoC) BCM2835. Obsahující mikroprocesor na bázi jádra osvědčené ARM architektury ARMv6, doplněný o grafický procesor VideoCore IV. Celý zmíněný systém disponuje sdílenou operační pamětí o kapacitě 512 MB, což je plně dostačující pro běh vhodného operačního systému s grafickým prostředím. Samotný operační systém musí být nahraný a spouštěný na vyjímatelné SD kartě. Model Raspberry Pi B je vidět na obrázku 6.



Obrázek 6: Raspberry Pi model B (převzato z [5])

Jak již bylo řečeno bude využívána zejména ve funkci grafického výstupu. Dle zvoleného systému bude možné volit možnost zobrazování, ať už na nízké úrovni v OpenGL, nebo vyšších aplikačních frameworků vhodných k tvorbě uživatelských rozhraní. Pomocí nichž bude prostředníkem v prezentaci jakýchkoli stavových či uživatelských dat na jakémkoliv standardním monitoru. Obrazová data jsou přenášena přímo po rozhraní

HDMI 1.4.

Tato platforma je taktéž vybavena i dalšími periferiemi, jež jsou určeny ke komunikaci s okolím. Mezi tato komunikační rozhraní, která využijeme, patří Ethernet a zejména USB pomocí něhož je možné připojit též klasická ovládací zařízení jako myš či klávesnici. V neposlední řadě disponuje také univerzálními vstupy a výstupy (GPIO). Některé disponují také speciální funkcí v podobě sběrnic USART, SPI, I<sup>2</sup>C apod.. Této funkcionalitě využijeme k jejich rozšíření, jelikož GPIO je s ohledem na naše požadavky nedostatek a bude potřeba jich navýšit. Také komunikační rozhraní RS-485 je možné převodem úrovní vyrobit ze sériové linky. S touto realizací se v tomto případě nepočítá, a to z důvodu nedefinovaného času vykonání programu na běžícím operačním systému nás tento fakt omezuje i u univerzálních vstupů a výstupů. Problémem je u komunikačních protokolů, kdy je potřeba odpovědět či potvrdit příjem do jasně definovaného času.

Z výše uvedených důvodů realizaci komunikačního rozhraní RS-485 a zvýšení počtu GPIO vyřešíme návrhem rozšiřující desky obsahující mikrokontrolér a další potřebné součásti k realizaci potřebných periferií. Ten bude vše řídit, využívat RaspberryPi jako grafického výstupu a přijímat příkazy k ovládání periferií, kterými bude disponovat. Nesmíme ale zapomenout na případnou komunikaci po Ethernetu, která zůstane umístěn na této platformě. Za pomoci něhož budeme mít k dispozici další způsob prezentace dat a řízení navrhovaného automatu.

### ■ 3.1.1 HDMI

HDMI (High-Definition Multi-media Interface) je rozhraní pro přenášení obrazového a zvukového signálu v nekomprimované podobě. Toto rozhraní jak název napovídá pochází z multimediální oblasti průmyslu i přes to je jeho signál kompatibilní s rozhraním DVI. Tato kompatibilita není stoprocentní, jedno omezení spočívá v omezení rozlišení přenášeného obrazu. Dalšími jsou neschopnost přenášení zvuku a realizace funkce CEC po rozhraní DVI. Z toho vyplývá že v určitých případech je možné použít přímého pasivního převodníku HDMI na DVI. A to za předpokladu že zobra-

## 3.2 MIKROKONTROLÉR

zovací zařízení dokáže správně oddělit obrazová a zvuková data, která jsou přenášena shodnou cestou.

Za dobu svého vývoje existuje více verzí, které doplňují další funkcionalitu např. doplnění o Ethernet či postupné zvětšování přenosového pásma. HDMI rozhraní nejčastěji využívá konektor typu A, mající 19 pinů a je právě kompatibilní se zmíněným DVI, signály jsou uvedeny v tabulce [1]

Tabulka 1: Signály HDMI

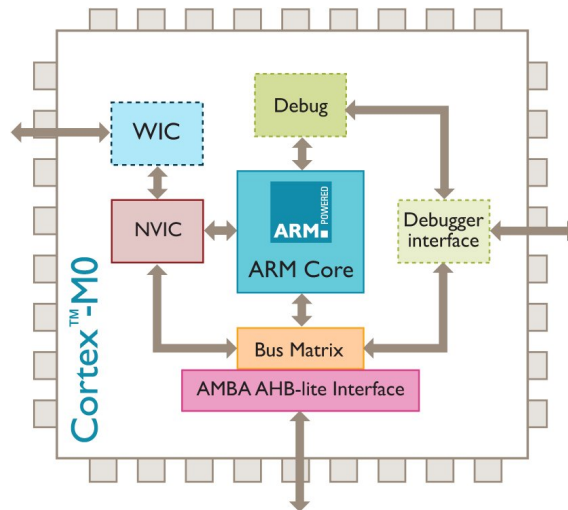
Číslo	Funkce	Číslo	Funkce
1	TMDS Data2+	11	TMDS Clock Shield
2	TMDS Data2 Shield	12	TMDS Clock-
3	TMDS Data2-	13	CEC
4	TMDS Data1+	14	Reserved (N.C. on device)
5	TMDS Data1 Shield	15	SCL
6	TMDS Data1-	16	SDA
7	TMDS Data0+	17	DDC/CEC Ground
8	TMDS Data0 Shield	18	+5 V Power (max 50 mA)
9	TMDS Data0-	19	Hot Plug Detect
10	TMDS Clock+		

## 3.2 Mikrokontrolér

Z požadavků na ostatní periférie jako jsou vstupy, výstupy, RS-485 a SPI pro komunikaci s A/D převodníky jsme zvolili mikrokontrolér STM32F051R8. Je to 32-bitový mikrokontrolér ARM Cortex M0 7 implementující ARMv6-M architekturu. Pro naši aplikaci disponuje dostatečným výkonem a má následující parametry.

- 48 MHz maximální frekvence
- 64 KB programové paměti
- 8 KB SRAM paměti

- 55 GPIO z toho 36 GPIO 5 V tolerantní
- 8 univerzálních časovačů
- 2x SPI, 2x I<sup>2</sup>C, 2x USART



Obrázek 7: Jádro ARM Cortex M0 (převzato z [6])

Jednotlivé vyjmenované periferie využijeme v našem návrhu. Časovače pro odměřování času a provádění cyklických operací s výhodou můžeme využít i přímé hardwarové podpory ke generování signálu PWM. Tento signál je vhodný ke spínání výkonového analogového výstupu. Standardní komunikační rozhraními jako je sériová linka již využijeme pro realizaci RS-485, kdy mikrokontrolér umožňuje využít i některý ze standardizovaných protokolů. Jedna sběrnice SPI bude využita pro příjem dat z A/D převodníku, který má operační napětí 5 V, tady je nutno počítat s využitím k tomu přizpůsobených vstupů.

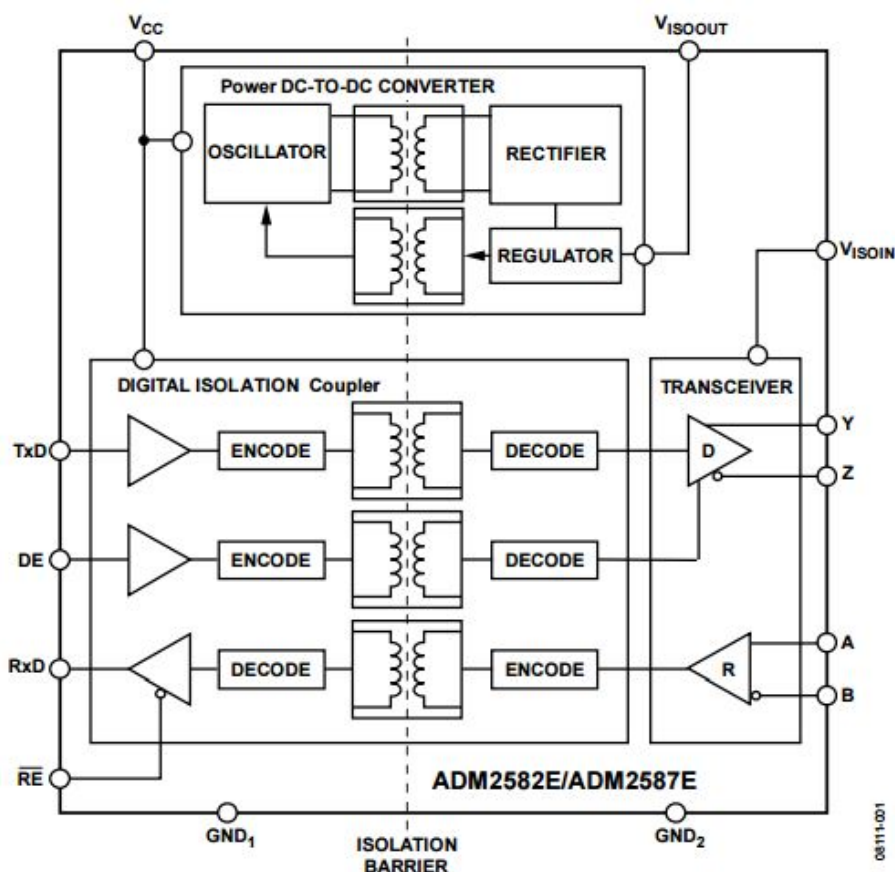
Ke komunikaci s Raspberry Pi jsou v návrhu vyhrazeny sběrnice SPI, jako rezerva navíc ještě sběrnice I<sup>2</sup>C. Z důvodu že Raspberry Pi umožňuje provozovat SPI pouze v režimu master, je přidán ještě jeden univerzální digitální signál pro indikaci potřeby komunikace směrem od mikrokontroléru. Tím se vyhneme neustálému dotazování směrem k slave zařízení, zda má potřebu komunikovat.

Veškeré další součásti jako vstupy, výstupy, povolení vysílání apod. budou využívat GPIO bez speciálních funkcí. Programování bude nevrženo standardním způsobem přes

ST-link v2, pro odladění v aplikaci jsou přidána tlačítka a signalizační LED.

### 3.3 RS-485

Sběrnice RS-485 je jedna z nejrozšířenějších standardizovaných sběrnic v průmyslové komunikaci. Zejména z důvodů jednoduché implementace a dobré odolnosti vůči rušení. Představuje nejčastější volbu jakou lze rozšiřovat modulární systémy a v tomto případě pomůže případnému začlenění, kde bude potřeba využívat možností navrhovaného systému. Ruku v ruce s tímto rozhraním jdou též komunikační protokoly jako MODBUS či jeho modifikace a další uzavřené protokoly různých výrobců zařízení.



Obrázek 8: Schéma ADM2582E (převzato z [7])

Realizace této sběrnice je provedena napojením na sériovou linku mikrokontroléru, tedy signály vysílání a příjmu TxD, RxD. Vedle nich ještě řídicí signály povolující

příjem a vysílání RE, DE. Tyto signály by principiálně nebyly potřeba, ale přinášejí možnost snížení spotřeby komunikačního rozhraní a taktéž menší zatěžování sběrnice připojenými posluchači. Převod úrovní mezi mikrokontrolérem a komunikačním rozhraním obstarává integrovaný obvod ADM2582E, jehož struktura je na obrázku 8.

Dle vnitřní blokové struktury integrovaného je zřejmé, že se jedná o verzi obvodu umožňující připojení do plně duplexní konfigurace komunikace s galvanickým oddělením obou stran s ochranou do  $\pm 15$  kV ESD. Aby toto oddělení bylo plně funkční, je též potřeba disponovat izolovaným zdrojem napájení. Výhodnou volbou tohoto obvodu je již integrovaný oddělovací DC-DC měnič pro napájení, tento fakt značně zjednodušuje návrh. Zapojení je doplněno o možnost vřazení zakončovacího odporu na vedení sběrnice a taktéž umožňuje funkčnost při připojení na poloduplexní komunikaci, jednoduchou změnou konfigurace pomocí zkratovacích propojek.

## ■ 3.4 Analogový výkonový výstup

Pro posílení výstupu, možnosti spínat napětí 48 V a proud 2 A pro řízení dvoufázového krokového motorku, nebo dvou stejnosměrných motorků jsou použity dva plně H - můstky realizované v technologii DMOS. Realizované integrovaným obvodem L6205 v pouzdře PowerSO 20, vybavenými běžnými ochranami, které jsou pro jejich použití běžné. Vnitřní blokové schéma je vidět na obrázku 9.

Každý můstek je plně oddělený a samostatně říditelný. Ovládání se tedy provádí třemi vstupními signály na každý můstek, dva spínají tranzistory a třetí signálem je enable. Enable je signál, který je vhodný pro využití pulzně šířkové modulace (PWM). Vhodně je připojen na odpovídající výstupy mikrokontroléru, kde je příslušný signál jednoduše programovatelný a je generovaný pomocí časovače. Můstek ke své funkci potřebuje minimum dalších podpurných součástek. Jediným doplněním jsou součástky k realizaci nábojové pumpy k zvýšení napětí pro řízení tranzistorů horní poloviny můstku. Schéma zapojení samotného můstku jen na obrázku 10.

Vstupní úrovně signálu pro řízení můstku jsou vstupní o úrovni napětí 5 V, přičemž námi používaný mikrokontrolér pracuje na napětí o úrovni 3,3 V. V tomto případě je





využito pro změnu úrovně galvanického oddělení obvodů duálních optočlenů MOCD217M. Pomocí nichž je jednak umožněno provést potřebnou změnu úrovně signálu na 5 V pro řízení můstku, tak i bezpečné oddělení výkonové části od ostatních. Konkrétní zapojení oddělení je obdobné jako v případě izolovaných výstupů 14.

## ■ 3.5 Analogové vstupy

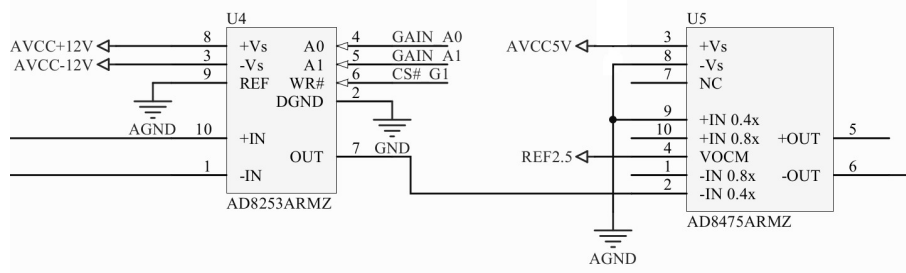
Ke sledování hodnot napětí, které jsou funkčně závislé na vyhodnocované jiné fyzikální veličině nám poslouží dva navržené analogové vstupy. Ty to vstupy poskytují dostatečný rozsah pro vstupní signál a taktéž tedy i odolnost na vstupu. Zpracovávají signál procházející „kanálem“ operačních zesilovačů, které jej dále upravují pro následný převod pomocí A/D převodníku. Převodník po komunikační sběrnici předá mikrokontroléru hodnotu již v digitální podobě, kde s ní můžeme dále pracovat, jak potřebujeme. Každý vstup se skládá z řetězce prvků, které jsou vidět na obrázku 11.



Obrázek 11: Blokové schéma A/D kanálu

Prvním obvodem na vstupu je konkrétní obvod AD8253, jedná se tedy o digitálně programovatelný přístrojový zesilovač. V naší aplikaci je napájený zdrojem pro analogovou část  $\pm 12$  V, tím je zajištěn velký vstupní a výstupní rozsah signálů. Hlavní a námi žádanou funkcionalitou tohoto obvodu je programovatelné zesílení a tím možnost přesného měření i slabých úrovní napětí na vstupu. Toto zesílení je možné využít jako 1, 10, 100 a 1000 krát, jež je jednoduše nastavitelné pomocí dvou signálů A0, A1 a platnost těchto signálů musí být potvrzena ještě jedním signálem označeným WR. Ovládací signály jsou přivedeny od mikrokontroléru, kdy nám na řízení těchto dvou obvodů stačí celkem pouze 4 signální vodiče.

### 3.5 ANALOGOVÉ VSTUPY

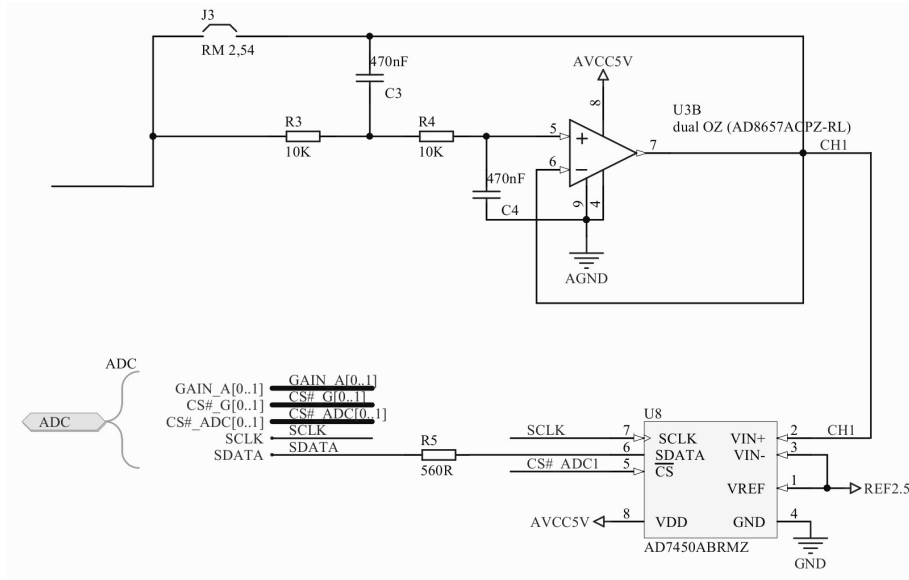


Obrázek 12: Schéma zapojení vstupních obvodů A/D kanálu

Za vstupním zesilovačem, následuje plně diferenciální zesilovač AD8475 s integrovanými precizními rezistory. U tohoto obvodu využíváme možnosti zesílit vstupní signál 0,4 krát a to z důvodu zeslabení signálu kvůli vstupu A/D převodníku. Taktéž na výstupu pracujeme už jen s unipolárním signálem, kdy využijeme „posunuté nuly“ na výstupu pomocí referenčního napětí 2,5 V. Jak již bylo řečeno obvod je plně diferenciální tedy i výstup. Avšak s ním již nepracujeme diferenciálně vůči druhému výstupu, ale vůči analogové zemi, tím dojde k dalšímu cílenému zeslabení signálu na polovinu. Schéma vstupu po tento obvod je na obrázku 12.

Ne vždy je pro nás žádoucí měřit vyšší frekvence analogového signálu, resp. ve spoustě aplikací mohou být rušivým elementem. Je tedy potřeba je odfiltrovat v tomto případě v následuje nízkofrekvenční filtr vytvořený z diskretního operačního zesilovače, zapojeného s útlumem 40 db/dek pro frekvence vyšší než 33 Hz. Obvod je možné v případě potřeby překlenout a vyšší frekvence signálu ponechat.

Posledním prvkem analogového kanálu je A/D převodník, tedy převod z analogového signálu na digitální, který je konečným výstupem z této obvodové části. Konkrétní obvodem realizujícím tento převod signálu realizuje je AD7450. Jde tedy o rychlý až 1 MSPS, 12-bitový aproximační převodník. Vstup převodníku je diferenciální. My toho využijeme přivedením napětí reference 2,5 V na invertující vstup. Tím je zajištěn převod správné hodnoty napětí, kdy je „posunutá nula“ právě posunuta do hodnoty této reference. Převod je inicializován mikrokontrolérem a to digitálním signálem CS (Chip select). Na hranu tohoto signálu je vstup ovzorkován a převeden. Samotný převod a přenos jednotlivých bitů je závislý na signálu CLK (clock) sběrnice SPI. Jelikož obvod

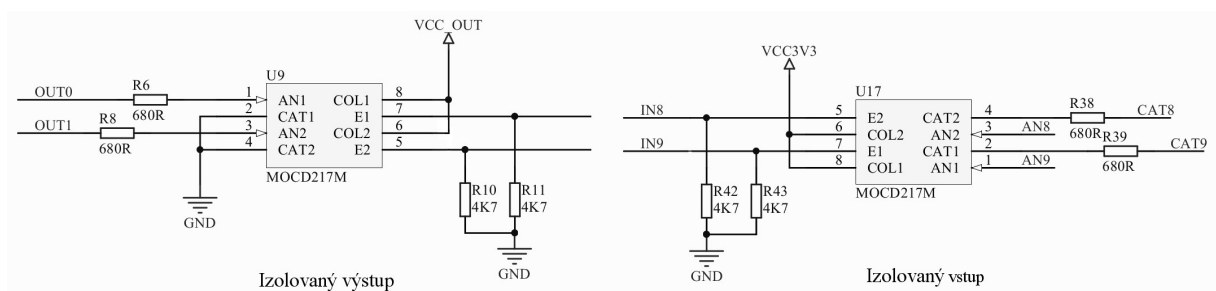


Obrázek 13: Schéma zapojení filtru a převodníku A/D kanálu

je napájen z 5 V zdroje, je k mikrokontroléru připojen na vstupy které jsou odolné vůči takové úrovni napětí. Pro omezení proudu je navržena možnost zapojit omezující rezistor. Schéma zapojení filtru a převodníku je na obrázku 13.

### 3.6 Digitální vstupy a výstupy

Jakékoliv vstupy a výstupy i v té nejzákladnější podobě zapnuto a vypnuto se využijí k signalizaci, či jednoduchému spínání vybraných prvků. Může se jednat o výstupní povely k nějakým akcím a zpětně mohou vstupovat signalizace hraničních stavů apod.



Obrázek 14: Zapojení izolovaných vstupů s výstupy

V tomto návrhu počítáme konkrétně s 10-ti vstupy a 10-ti výstupy. Z důvodu

## 3.7 NAPÁJENÍ

připojování různorodých zařízení se jedná o izolované galvanicky oddělené I/O. Toto řešení požaduje další oddělený zdroj k napájení izolovaného výstupu. Použitým oddělení jsou v obou případech oddělovací optrony v duálním řešení MOCD217M s izolační schopností 2,5 kV. Využití tohoto řešení pro oddělení vstupu nám umožní tedy izolovat a ochránit elektroniku od vnějších vlivů. Jejich výhodou je široký rozsah proudové zátížitelnosti emitující diody, to přináší možnost většího rozsahu napětí na vstupu.

Konkrétní zapojení jednoho izolovaného vstupu a výstupu je zachycené na obrázku 14. Jak je ze zapojení zřejmé na výstupu oddělovače jsou použity pull-down rezistory. To z důvodu že optron je realizovaný pomocí fototranzistoru a v tomto případě signál nebude invertovaný.

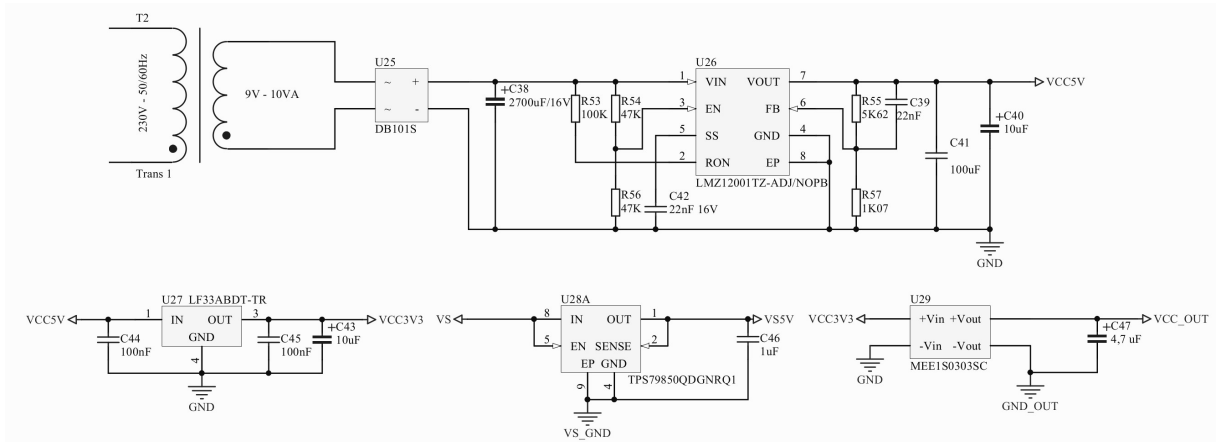
### ■ 3.7 Napájení

Potřeby kladené na napájení vycházejí z jednotlivých prvků obsažených v návrhu a jejich spotřeby. Nejprve je potřeba požadavky na napájení oddělit od sebe dle toho s jakými signály pracují, tedy analogové a digitální. A dále též podle napětí a příslušného odběru některé můžeme označit jako výkonové. Nami zvolené zdroje jsou následující:

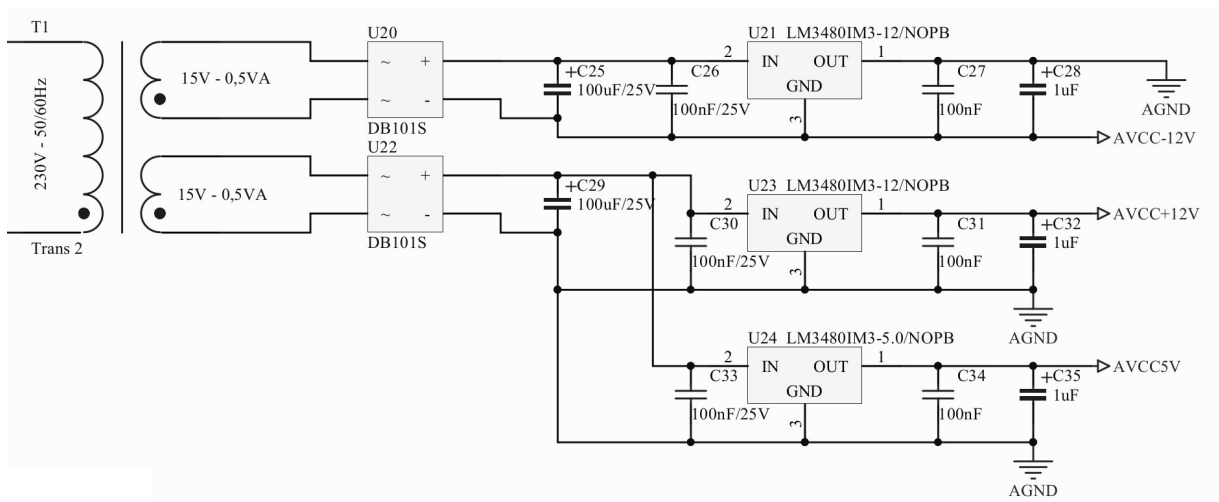
- transformátor 9 V @ 10 VA – zdroj napájení pro digitální obvody
- transformátor 2 x 15 V @ 1 VA – napájení pro analogové obvody
- spínaný zdroj 24 V @ 100 W – napájení pro výkonové výstupní obvody

Jak je vidět jedná se o různorodé zdroje odpovídající svými parametry oblasti jejich využití v naší aplikaci. Výstupní napětí je potřeba ještě stabilizovat a odvodit odpovídající úroveň napětí. Pro digitální část je nejprve stabilizováno napětí 5 V, které je na výstupu spínaného regulátoru LMZ12001. Toto napětí je použito pro Raspberry Pi platformu, která si jej na desce sama dále upravuje. V našem návrhu je ještě sníženo na 3,3 V lineárním regulátorem LF33. A následně použito pro napájení všech digitálních obvodů, vyjma výstupů. Pro napájení výstupů je použit ještě oddělující izolovaný DC/DC měnič MEE1S0303SC. Schéma části napájení pro digitální část je vyobrazeno na 15.

Pro analogovou část jsou využity dvě oddělená vinutí transformátoru k jednoduchému vytvoření napětí  $\pm 12$  V. Obě větve jsou realizovány lineárními regulátory



Obrázek 15: Schéma napájení pro digitální část



Obrázek 16: Schéma napájení pro analogovou část

### 3.8 MECHANICKÁ ČÁST

LM3480IM3-12, kdy vhodnou kombinací výstupů získáme požadovaných  $\pm 12$  V. Paralelně od jedné větve je též odvozeno napětí 5 V, pro analogové obvody, taktéž lineárním regulátorem LM3480IM3-5.0. Schéma části napájení pro analogovou část je vyobrazeno na 16.

Zbývajícím zdrojem je zdroj pro výkonovou část, který je realizovaný pomocí spínaného vestavného zdroje MEAN WELL S-100F-24. Oddělení výkonového výstupu ještě požaduje izolované napětí 5 V, které je získáno z těchto 24 V lineárním regulátorem TPS79850.

## ■ 3.8 Mechanická část

K zapouzdření řešení celistvého systému, jak desky navrhovaného plošného spoje, tak platformy Raspberry Pi i spínaného zdroje byla kovová krabička U-KK09-23251. Tato krabička je v „U“ provedení o rozměrech 235 x 257 x 83 mm do níž se všechny části vejdou.

Z volby zapouzdření nám vychází velikost plošného spoje a to hlavně jeho šíře v oblasti předního panelu, kde jsou umístěny konektory v přímém provedení na DPS. Veškeré konektory jsou umístěny v oblasti nad plošným spoje a vycházejí z jedné řady nasunovacích konektorů PTR řady STL1550 s roztečí 3,5 mm. Na zadním panelu zbývá místo pro konektor síťového napájení a další konektory HDMI, USB a Ethernet, kde jsou v panelovém provedení. To nám dává možnost volnosti v jejich umístění. Podoba předního panelu, konkrétně části ve které se nacházejí konektory a kompletní výrobní podklady jsou v příloze A.

## Kapitola 4

# Grafický výstup

Dnes mezi nejčastěji užívané řešení vizualizace je využití jednoduchých grafických displejů, které jsou přímo připojené k mikrokontroléru. Tyto displeje poskytují možnosti využívat prvky grafických primitiv, či pokročilých knihoven, které ne vždy musejí být kompatibilní napříč mikrokontroléry.

Druhým možným užívaným řešením v oblasti vizualizace je užití tzv. Inteligentního displeje. Rozdíl od klasického spočívá ve faktu, že logika vykreslování, tedy i dostupné grafické prvky zahrnuté v knihovně jsou přímo v displeji. Ovládání probíhá pomocí zasílání příkazů, po běžně užívaných periferních sběrnicích, kterým displej interpretuje. Velmi často umožňují využívat dotykových panelů.

Jiným obdobným možným řešením bývá odesílání stavových dat po sběrnicích typu RS-485 nebo po síti Ethernet. Kdy jsou přenášena data s identifikátorem a až v cílovém zařízení jsou zpracována, resp. svou grafickou interpretaci získávají až zde.

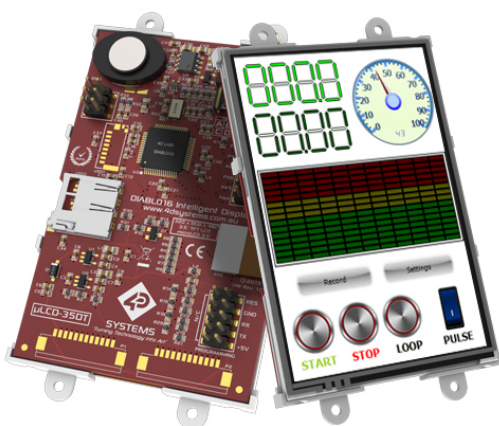
### 4.1 Inteligentní displeje

Náš celý systém je vlastně myšlenkou vytvoření jednoduchého programovatelného automatu, doplněného o inteligentní displej, který je prezentovaný libovolným displayem. Díky dostupné konektivě taktéž zůstává zachována i možnost stavová data prostě někam posílat k dalšímu zpracování. Cílem je navrhnout univerzální grafickou knihovnou, které by byla v používání obdobou inteligentních displejů. Proto se na řešení některých zástupců krátce podíváme.

## 4.1 INTELIGENTNÍ DISPLEJE

### 4.1.1 4D Systems PICASO

Výrobce nabízející celou škálu produktů z oblasti zobrazování. V našem zájmu jsou moduly inteligentních displejů a grafických procesorů. Ve většině případů se jedná o modulové řešení displeje např.  $\mu$ LCD-35DT a řešení disponující VGA např.  $\mu$ VGA-III, založené na firemním procesoru označeném jako PICASO. Možnosti těchto produktů jsou obdobné nejvíce se odlišují výstupem resp. dostupným rozlišením. Konkrétní podoba výstupu se vytváří v grafické aplikaci, kdy objekty je možno uživatelsky přizpůsobit. Tato přizpůsobení se do modulů vkládají uložené na mikro SD kartě. Výhodné je že moduly mohou pracovat samy o sobě, kdy fungují jako mikrokontroléry s příslušnými vstupy a výstupy, dle řešení.



**$\mu$ LCD-35DT**  
3.5" Diablo16 Intelligent Display Module

Obrázek 17: 4D Systems inteligentní display (převzato z [9])

Parametry  $\mu$ VGA-III

- VGA rozhraní.
- Rozlišení až 800x480.
- Podpora 4DGL grafické knihovny.
- Sériové rozhraní.
- I<sup>2</sup>C, 13 x GPIO



### ■ 4.1.2 FTDI EVE

Jedná se o řešení pro inteligentní displeje, které se snaží aplikaci oprostít od nároků na grafický výkon. Mezi jeho možnosti použití připadá též možnost přehrávání zvuku. Je nutné naprogramovat objekty se kterými se bude pracovat, resp. je ovládat. Existují provedení pro dotykové displeje na kapacitním i resistivním principu. Výhodou je dostupnost řešení v provedení modulu a nebo samostatného integrovaného obvodu pro návrh přímo v aplikaci. Nevýhodou je omezená možnost použití grafických objektů. Z následujících parametrů je patrné že jsou předně určené pro malé LCD displeje, které jsou omezené rozlišením a jejich ne příliš přívětivou cenou.

Parametry FT800

- Pro dotykové panely.
- Rozlišení až 512x512 s rozhraním RGB-6,6,6.
- Komunikace po I<sup>2</sup>C, SPI.
- Audio PCM.

## ■ 4.2 Raspberry Pi jako zobrazovač

Jak již bylo zmíněno, výstupní interface, na který budou generována výstupní obrazová data, je umístěný na Raspberry Pi v podobě HDMI/DVI. Tato platforma funguje pod operačním systémem linux a dalších jako např. RISC OS. Mezi deriváty linuxu je nejvíce komunitně podporovaný systém Raspbian, což je obdoba Debianu sestavená pro tuto platformu. Kompletní přehled a získat je lze na webu <sup>1</sup>. Pro naše potřeby zvolíme zřejmě nejrozšířenější a to Raspbian. Právě z důvodu shodného ovládání z klasické PC platformy.

Z uvedených faktů je zřejmé, že nabízí obdobné funkce, které jsou známé z klasického počítačového světa. Naším cílem je využít základní grafické funkce k vyobrazení stavových informací a taktéž zpracovávat zpětnou interakci od uživatele. Vhodnou možností k zpracování událostí je využít služeb jednoduchého GUI, a mít tak u

<sup>1</sup><https://www.raspberrypi.org/downloads/>, Platné k 8.4.2015

## 4.2 RASPBERRY PI JAKO ZOBRAZOVÁČ

vykreslování možnost volby využívat grafických primitiv, tak připravených prvků. V našem případě budeme ať až nepřímo nebo přímo k tomuto využívat OpenGL ES.

Z toho vyplývá, že budeme vytvářet grafické objekty se kterými bude možné pracovat, obdobně jako u inteligentních displejů. Zároveň tím že se ke zpracování událostí a vykreslování námi nevržených grafických objektů budeme využívat GUI. Abychom mohli využívat všech dostupných možností, které zde uvažujeme, je potřeba danou platformu nejprve nainstalovat a patřičně nakonfigurovat.

### ■ 4.2.1 Instalace a konfigurace

Jak je výše uvedeno k funkci je potřeba využívat vybraný operační systém. V našem případě ze zmíněného důvodu je zvoleného řešení na bázi linuxu a to Raspbian. Instalace je velmi jednoduchá, spočívá v získání instalačního obrazu na výše zmíněném webu, jeho přenesením na SD kartu a následným spuštěním. Přenos na SD kartu provedeme pomocí softwarového nástroje Win32DiskImager.

Při prvním spuštění je potřeba využívat připojeného displeje a ovládacích periférií, konkrétně klávesnice. Po spuštění a načtení systému proběhne krátká konfigurace, které spočívá v nastavení jazykové mutace. Dále přizpůsobení systému velikosti použité SD karty, nastavení velikosti sdílené paměti pro grafický procesor a povolení nízkoúrovňových periférií. V neposlední řadě povolení SSH pro vzdálený přístup a nastavené do jakého prostředí se bude běžně systém spouštět.

Tato instalace je obdobná i pro další dostupné distribuce, jen záleží na konfiguračních specifikách daného derivátu existují, ale jsou i výjimky kdy je potřeba instalovat plně ručně. Pro naši volbu byl právě dostupný skript konfiguratoru, který daný proces velmi ulehčil, avšak i ostatní mají podrobnou dokumentaci.

Díky početné komunitě, která stojí za projektem této platformy existuje mnoho knihoven pro práci s nízkoúrovňovými perifériemi. Jednou takovou je knihovna WiringPi, je dostupná po programovací jazyk C. Tato knihovna byla využita pro otestování námi uvažovaných periférií jako SPI, TWI a GPIO. Nejjednodušší otestování schopnosti tyto periferie ovládat bylo provedeno na odpovídajících sériových pamětech. Samotný

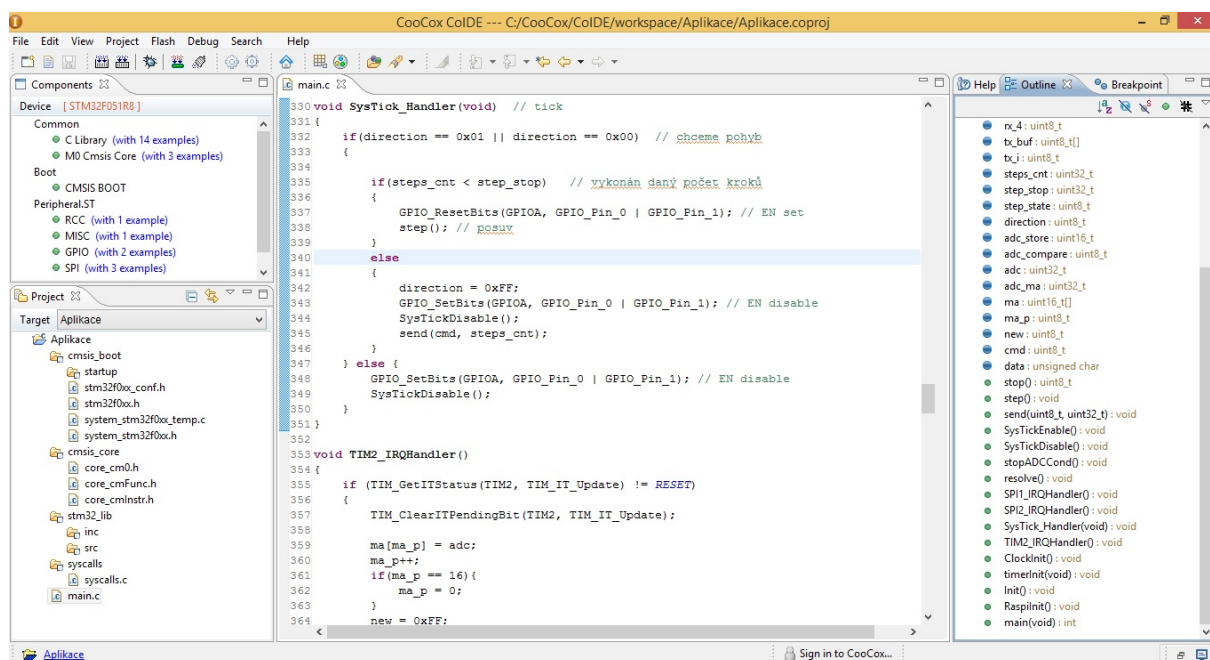
## 4.2 RASPBERRY PI JAKO ZOBRAZOVACĚ

grafický výstup funguje ihned po zapnutí a kompatibilita OpenGL ES byla ověřena pomocí přiložených demo aplikací v systému. Tedy platforma je takto připravena pro práci.

## Kapitola 5

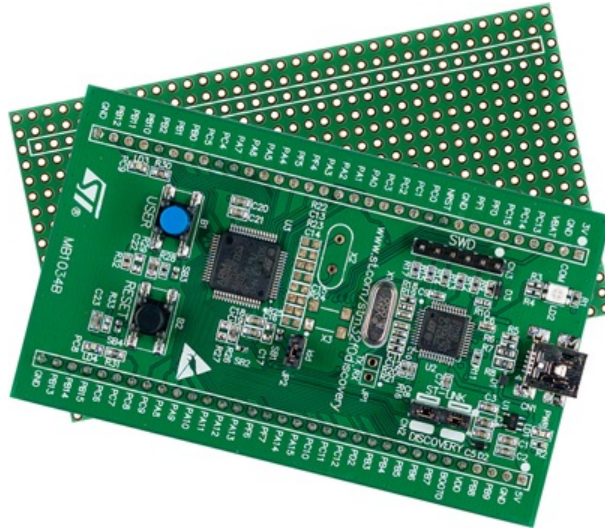
# Testování

Testování navrženého a sestaveného zařízení je nejdůležitější částí, kterou je potřeba provést pro ověření právnosti návrhu a funkčnosti celé realizace. Podstata ožívování a testování spočívá v programování samotného mikrokontroléru, který ovládá jednotlivé periferie.



Obrázek 18: CooCox IDE

K programování bylo využito vývojového prostředí CooCox, na obrázku 18, které umožňuje tvorbu aplikačních programů v programovacím jazyce C. K dispozici jsou standardní knihovny od STmicroelectronics k ovládání jednotlivých periférií, tak je vývoj jednodušší. Výhodou tohoto prostředí je také, že obsahuje debugger a je tedy možné aplikaci odladit přímo v mikrokontroléru. K programování a ladění byl využit ST-link v2, kterým disponuje ST discovery kit, je vidět na obrázku 19.



Obrázek 19: ST discovery kit (převzato z [10])

Pomocí těchto zmíněných nástrojů byl otestován navržený hardware po jednotlivých dílčích perifériích, abychom je mohli s jistotou používat jako celku v testovací aplikaci.

Z těchto dílčích částí byl vytvořen řídicí program k řízení dále zmíněnému přípravku Thompsonova-Lampardova etalonu kapacity. Tento program bude řídit měření na daném přípravku na základě příkazů, které budou posílány ovládací aplikací vytvořené na Raspberry Pi.

## ■ 5.1 Ovládací aplikace

Ovládání jednotlivých akcí automatu je prováděno pro tyto účely testování z platformy Raspberry Pi. Řídicí příkazy jsou přenášeny po komunikační sběrnici do mikrokontroléru. Aplikace je vytvořena a napsána v programovacím jazyce C++ s využitím jednoduchého grafického prostředí, které je navrženo pro tento konkrétní účel. K vytvoření ovládacího prostředí je využito multiplatformního Qt frameworku, za použití QtCreatoru.

### ■ 5.1.1 Qt framework

Jde o multiplatformní knihovnu, které je dostupná od desktopového prostředí přes embedded až po mobilní zařízení. Knihovna je zaměřená nejen na uživatelské rozhraní, ale i další multimediální a komunikační části, které jsou tedy dostupné v rámci jednoho balíku. Licenčně je dostupný jak pro komerční využití, tak i pro vývoj open source, za předpokladu splnění daných podmínek.

Tato knihovna je předně vyvíjena v jazyce C++, ale je portována i na jiné programovací jazyky, jako je například Python. Ačkoliv se jedná o programovací jazyk C++, je potřeba využít dané kompilační nástroje, které jsou běžně dostupné. A to z důvodu že je mírně rozšířený v oblasti šíření a zpracování událostí od objektů.

Toto specifikum jsou tzv. sloty a signály. V podstatě se jedná o běžně známé callbacky, v případě slotů a signálů je přidána typová kontrola, která přináší jisté bezpečí. Další výhodami je, že je mohou obsahovat jakékoli objekty zděděné od základního objektu a jejich napojování metodou connect. Konkrétní možnosti použití a specifika je možné zhlédnout na webu <sup>2</sup>.

### ■ 5.1.2 Grafická podoba

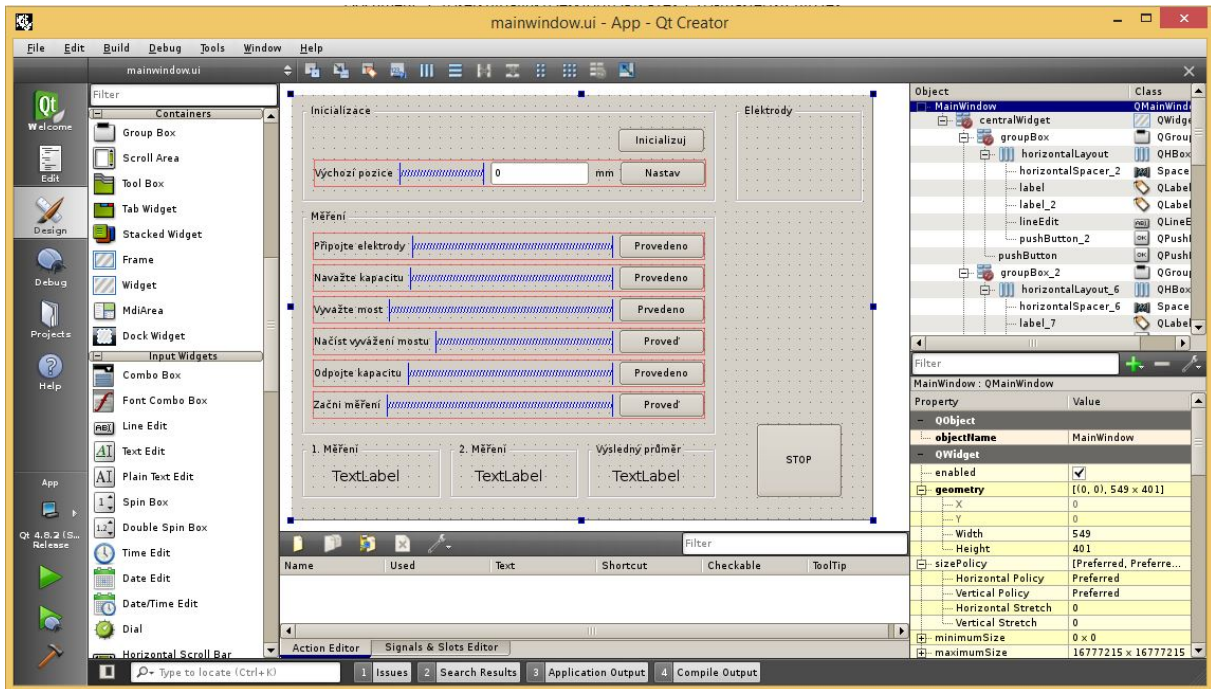
Samotná grafická podoba aplikace je navržena pomocí již zmíněného QtCreatoru, který umožňuje uživatelsky, přívětivě sestavit jednoduché GUI. Tak i sestavit samotný program, včetně napojení výše uvedených signálů, slotů a tedy zpracovávat uživatelské události.

Návrh aplikace vychází z potřebných řídicích událostí a akcí, které jsou potřeba k měření na modelu etalonu. S ohledem na tyto požadavky v návrhovém systému vypadá následovně 20. Kde je vidět námi vytvořený formulář, další prvky, které je možné použít a jejich nastavení.

Program zpracování událostí, musí taktéž spolupracovat s knihovnou wiringPi, která na dané platformě usnadňuje přístup k perifériím, v našem případě to je využívání SPI ke komunikaci s mikrokontrolérem.

---

<sup>2</sup><http://doc.qt.io/qt-4.8/signalsandslots.html>, Platné k 7.5.2015



Obrázek 20: Aplikace v Qt Creatoru

### 5.1.3 Logika aplikace

Tato aplikace, jak je patrné je napsána v jazyce C++ za využití Qt frameworku, bude spuštěna na platformě Raspberry Pi, ježe je součástí našeho automatu. V podobě testovací aplikace musí implementovat komunikaci pomocí periférií, které ji propojují s naší rozšiřující deskou. K tomu musí mít definovaný způsob zacházení s konkrétními příkazy a jejich významem.

Pro přístup k perifériím na RaspberryPi byla využita zmiňovaná knihovna wiringPi. Knihovna umožňuje přístup k jednotlivým pinům, tak i dostupným perifériím a poskytuje další funkcionalitu. V rámci této aplikace budeme využívat přístup k pinům a sběrnici SPI, které je potřeba pouze jednoduše inicializovat. Následujícím způsobem se načte samotná knihovna a inicializuje se SPI kanál 1 s hodinovým signálem o frekvenci 3 MHz.



## 5.1 OVLÁDACÍ APLIKACE

```
if (wiringPiSetup()==-1)
{
    printf("Nelze inicializovat knihovnu\n");
    return 1;
}

if (wiringPiSPISetup(1, 3000000)==-1)
{
    printf("Nelze inicializovat SPI\n");
    return 1;
}
```

Kompletní komunikace se tedy odehrává po sběrnici SPI, RaspberryPi má v této věci nevýhodu, že musí být při této komunikaci jedinečně master. V tomto případě bylo zvoleno řešení potřeby komunikace od rozšiřující desky periodickým dotazováním se zda jsou dostupná nějaká data pro přenos.

Tento fakt sebou obnáší konfiguraci univerzálního časovače. V tomto případě je využito systémového časovače, který je dostupný pod Qt frameworkem. To s sebou nese jisté omezení a to v minimálním časovém úseku 1 ms, který není nijak garantován. Za účelem zobrazování stavových dat nebo zpracováním uživatelských příkazů, které nejsou časově kritické, je toto řešení dostačující. Získávání a zpracování standardizovaných příkazů na RaspberryPi je následující



```
wiringPiSPIDataRW (1, &cmd, 1);

if (cmd == 0x11 || cmd == 0x22 || cmd == 0x33 ||
    cmd == 0x44 || cmd == 0x55)
{
    wiringPiSPIDataRW (1, &data0, 1);
    wiringPiSPIDataRW (1, &data1, 1);
    wiringPiSPIDataRW (1, &data2, 1);
} else {
    this->timer.start(1000);
    return;
}
```

Z uvedeného úryvku kódu se dá vyčíst, že v testovací aplikaci vystačíme s minimem příkazů. Taktéž komunikace po SPI má pevně definovaný rámec a to přidává na bezpečnosti komunikace. Když se podíváme na konkrétní příkazy, tak mají následující význam:

- 0x11 - při příjmu znamená potvrzení o inicializaci.
- 0x22 - potvrzuje výchozí posuv, následuje počet kroků které byly vykonány.
- 0x33 - po tomto příkazu přijme hodnotu z A/D převodníku, resp. vyvážení mostu.
- 0x44 - potvrzuje konec měření, je následován počtem vykonaných kroků.
- 0x55 - potvrzení, že mikrokontrolér je ve výchozím stavu.

Pomocí výše jmenovaných příkazů je v této aplikaci automat uživatelsky ovládán. Děje se tomu tak na základě uživatelských akcí z formuláře aplikace. Tedy vybrané akce způsobí vyslání příkazu a toto jsou adekvátní odpovědi po jejich zpracování a vykonání.

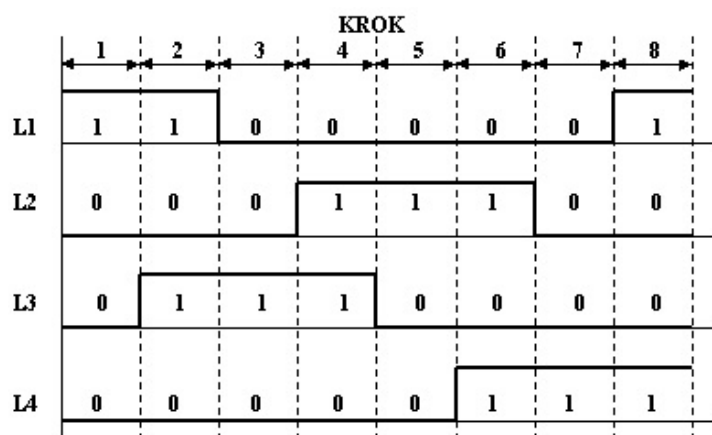
## ■ 5.2 Program pro mikrokontrolér

Tento program je zásadní pro funkcionalitu celého programovatelného automatu. V toto případě jde o modifikovanou verzi uvažovanou pro univerzální knihovnu. Ve své

## 5.2 PROGRAM PRO MIKROKONTROLÉR

upravené funkcionalitě odráží potřeby, jež jsou potřebné v testovací aplikaci, které jsou kritické. Základem tohoto programu je inicializace vstupů a výstupů, dále odpovídající konfigurace periférií, jak je dle zapojení rozšiřující desky třeba.

Další akce na straně mikrokontroléru jsou dány dle přijatých příkazů a jejich parametrů. Tyto příkazy odpovídají perifériím, kterými deska disponuje. Implementuje tedy funkčnost komunikace po sběrnici SPI k RaspberryPi, tedy tu po které přijímá příkazy a vrací zprávy o provedení či stavu. Jednou funkcí je ovládání h-můstku k řízení krokového motorku. Je realizováno posloupností stavů fází motorku, které jsou potřebné k vykonávání pohybu krokového motorku, v tomto případě s polovičním krokem 21. Tyto kroky jsou vykonávány tolikrát, podle přijatého počtu kroků za příkazem. Mezi kroky je třeba vyčkat, tedy udělat nějaké časové zpoždění. Tato vlastnost je realizována pomocí časovače SysTick, kdy se při přerušení volá příslušná rutina k vykonání kroku.



Obrázek 21: Řízení krokového motorku s polovičním krokem (Převzato z [11])

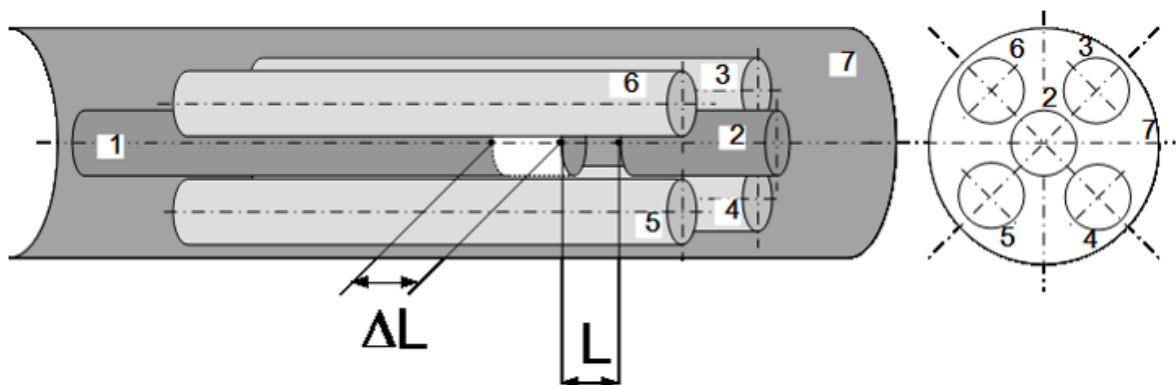
Následně implementuje funkcionalitu A/D převodníku. Tento převodník je typu SAR a je připojený na druhou sběrnici SPI. Podle přijatého příkazu je nejprve nastaveno zesílení signálu na vstupu. Následně spuštěn převod, ten z pohledu mikrokontroléru je kompletní po přenosu 16-ti bitů po sběrnici SPI. Využijeme tedy klasického přerušení a data zpracujeme. V tomto případě je s odpovídajícím příkazem pošleme k vizualizaci.

Zbývající funkcionalitou je nastavování digitálních výstupů a čtení digitálních vstupů. Této funkce je využito v tomto případě k hlídání stavu koncových spínačů. Pro účel měření pomocí testovací aplikace je zde ještě realizována komplexní funkce pro toto

měření. A to v podobě pohybu krokového motorku, souvisle s tím podmínka na čtení hodnoty A/D převodníku a její porovnání s dříve naměřenou hodnotou.

## ■ 5.3 Měření

Konečné testování jako celku bylo provedeno měřením na modelu etalonu kapacity pracujícím na principu Thompsonova-Lampardova teoremu [12]. Zjednodušeně se jedná o zařízení, které má přesně definovanou a stálou kapacitu. Resp. změnu kapacity, která je polohou stínící elektrody, která odstíňuje dvojici na které se měří. Poloha je řízena pomocí krokového motorku, kterým je realizován lineární posuv a tedy poloha, délka vysunutí stínící elektrody. Na obrázku 22 je vidět model etalonu, resp. elektrod. Kde 1 - je posuvná stínící elektroda; 2 - pevná stínící elektroda; 3, 5 a 4, 6 - páry aktivních elektrod; L - udává aktivní délku kapacity.



Obrázek 22: Model elektrod etalonu (převzato z [12])

K měření kapacity budeme využívat měřicího RLC mostu BM-539. A to tak, že budeme měřit hodnotu jeho vyvážení v podobě analogové hodnoty napětí na jeho výstupu.

Náš programovatelný automat je tedy připojený k zmíněnému modelu etalonu kapacity a využívá následujících periférií. Analogový výkonový výstup k řízení krokového motorku. Digitální výstup k „napájení“ bezpečnostních koncových spínačů a digitální vstupy k snímání signálu z těchto spínačů. Analogový vstup je využit pro snímání hodnoty vyvážení RLC mostu.

## 5.3 MĚŘENÍ

Toto zapojení je provedeno daným způsobem, který je pro tuto úlohu vyžadován. Význam jednotlivých konektorů a jejich pinů na našem řídicím automatu, je uveden v příloze B. Napájecí vodič pro koncové spínače musí být připojen na digitální výstup č.9. Signál z horního koncového spínače (bílý CINCH), je přiveden na digitální vstup č.9. a z dolního koncového spínače (žlutý CINCH) na digitální vstup č.8. Jelikož se jedná o izolované vstupy je nutno ještě propojit libovolný pin digitálních výstupů GND na piny digitálních vstupů GND8 a GND9.

Taktéž výkonový výstup musí být patřičně připojený a to jeden pár vinutí (červený CINCH) na levý výkonový výstup B1 - B2. Druhý pár vinutí (černý CINCH) na pravý výkonový výstup A1 - A2. Pro měření výstupu vyvážením RLC mostu, kdy zemní vodič je přiveden na analogový vstup IN2- a signální vodič na vstup IN2+.

Další periferie jako USB, jsou využity pro připojení ovládacích zařízení. HDMI výstup přes převodník na DVI připojen kde standardnímu LCD monitoru, tedy poskytuje nám informace o stavu a umožňuje nám interaktivně ovládat měření. Druhou možností pro řízení měření bylo využití Ethernetu a přes něj vzdáleného přístupu, tedy možnosti využít notebooku k řízení a vizualizaci stejným způsobem jako přímo na realizovaném automatu.

## Spuštění a průběh

Po zapojení etalonu a automatu výše uvedeným způsobem již zbývá připojit k automatu ovládací zařízení jako myš, klávesnici a běžný LCD displej. Jako poslední zbývá napájecí kabel, kdy posledním úkonem je spuštění celého automatu spínačem.

Ihned po zapnutí se na našem displeji okamžitě zobrazí spouštění operačního systému Raspbian. Spustí se do grafického prostředí, jaké běžně známe z počítačového světa. Kde je potřeba provést před spuštěním měření ještě jeden úkon a to načtení modulu ovladačů do jádra. Provedeme tak zadáním příkazu `"sudo gpio load spi"` do terminálu. Následně již můžeme spustit měřicí aplikaci, kterou jsme si již k tomuto účelu připravili.

Tato aplikace nás postupně provede celým měřicím procesem, po jednotlivých dílčích krocích, které je potřeba uživatelsky vykonávat. Jelikož se jedná o měření ke stanovení

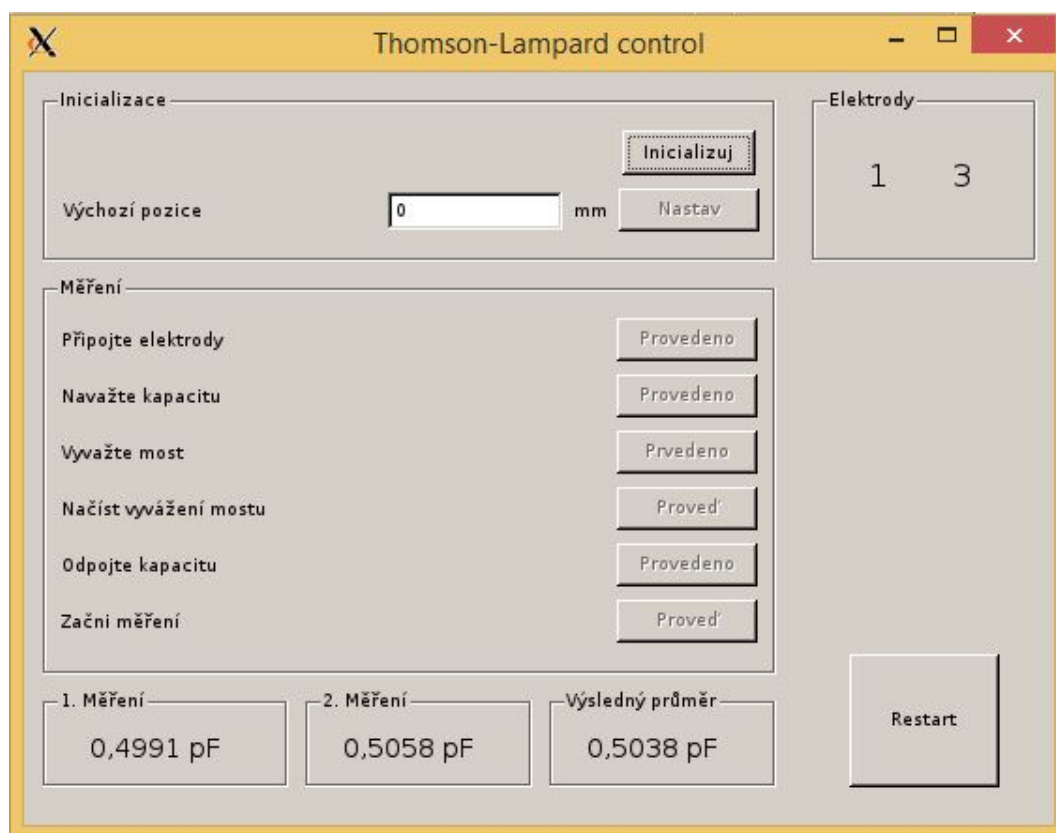
neznámé kapacity, je třeba vykonávat tyto následující kroky.

1. Provede inicializaci etalonu - stínící elektroda sjede do výchozí polohy, tj. do krajní polohy a následně malý úsek ve směru následného pohybu. Tento krok je blokující a vyžaduje potvrzení o mikrokontroléru.
2. Nastavení výchozí polohy - umožňuje uživateli posunout výchozí pozici elektrody, tím je umožněno provádět měření i v jiné části etalonu. V tomto případě se čeká až se tento úkon provede.
3. Připojení příslušných elektrod etalonu - uživatel potvrzuje připojení vyobrazeného páru elektrod pro toto měření k RLC mostu.
4. Navázání kapacity - potvrzuje se, že neznámá kapacita byla paralelně navázána k měřicímu mostu.
5. Vyvážení mostu - je třeba měřicí most ideálně vyvážit.
6. Načtení vyvážení mostu - aplikace odešle požadavek na měření výstupní hodnoty vyvážení mostu, kterou si uloží pro toto měření. Nelze pokračovat dokud tato hodnota nebude načtena.
7. Odpojení kapacity - tímto se potvrzuje, že neznámá kapacita byla odpojena a zůstal připojený pouze etalon kapacity.
8. Začátek měření - vyšle signál k započetí měření, automat pohybuje elektrodou, do té doby dokud se měřicí most opět nevyváží. Když k tomu dojde aplikace z počtu vykonaných kroků, krokového motorku vypočítá neznámou kapacitu.
9. Kroky 1. - 8. se opakují s druhým párem elektrod. Z obou měření se vypočítá průměr, tato hodnota odpovídá neznámé měřené kapacitě.

Z bezpečnostních důvodů nebo důvodu závažné chyby automatu aplikace obsahuje tlačítko na zastavení měření, které taktéž slouží k restartování aplikace a možnosti dalšího nového měření.

### 5.3 MĚŘENÍ

Tímto měřením jsme otestovali základní funkčnost celého automatu, výsledný stav měřicí aplikace je následující 23, kde je vidět hodnota měřené kapacity.



Obrázek 23: Testování aplikace

## Kapitola 6

# Návrh univerzální knihovny

Myšlenka vytvoření univerzální pro řízení námi navrženého etalonu vznikla za cílem vyhnout se klasickému programování a umožnit vytvářet ovládací prvky pomocí konfiguračních souborů. To si žádá standardizovat přístup k jednotlivým perifériím na rozšiřující desce. A na RaspberryPi poskytnutí knihovny, která obstará komunikaci a dle konfiguračních souborů uživatele bude spravovat akce a přijatá data.

### 6.1 Rozšiřující deska

Program pro námi navrženou desku musí být dostatečně robustní, aby zvládl bezpečně zpracovávat komunikaci mezi ní a RaspberryPi. Tato komunikace je po sběrnici SPI, kdy mikrokontrolér je v režimu slave. Měl by mít dostatečně velký buffer pro příjem příkazů k vykonávání. A to z důvodu, že ne všechny příkazy se dají vykonat okamžitě, resp. odezva na ně, například A/D převod nějakou chvíli trvá. Definované příkazy říkají co se má provést a mohou být následovány dalšími konfiguračními daty.

#### Digitální vstupy

Čtení digitálních vstupů je jeden z nejjednodušších a okamžitě proveditelných úkonů, který lze automaticky vykonat. Mikrokontrolér načte odpovídající hodnoty vstupů, kterými disponuje, vytvoří z nich datagram s příslušným příkazem a ihned jej odešle ovládacímu zařízení.

#### Digitální výstupy

Zápis digitálních výstupů je obdobou jejich čtení. Přijatá data se dekodují a zapíší na odpovídající vstupu. V tomto případě je též využito masky pro jejich zápis, aby

## 6.1 ROZŠÍŘUJÍCÍ DESKA

nedošlo nedbalému ovlivnění jiných, které chceme ponechat v původním stavu. Zápis je potvrzen odesláním příkazu o splnění.

### Analogový výstup

Řízení analogového výstupu, by mělo mít možnost pracovat ve více režimech. Nabízí se možnost řízení dvoufázového krokového motorku a nebo dvou stejnosměrných motorků. Toto je nutné rozlišit odeslaným příkazem. U krokového motorku je tedy definována posloupnost kroků které jsou vykonávány. Společně s daným příkazem musí být přijaty informace o počtu kroků, směru jejich provádění a konfigurovatelném zpoždění mezi nimi. Po provedení příslušného počtu kroků je odesláno potvrzení o jejich vykonání.

Řízení stejnosměrného motorku je jednodušší, kdy je udán směr smyslu proudu a možnost regulace otáček pomocí PWM. V tomto případě potvrzení obsahuje informaci pouze o tom že daná akce se skutečně provádí.

### Analogové vstupy

Analogové vstupy jsou další periferií, kterou musíme ovládat. Jelikož máme dostupné dva oddělené kanály, i když A/D převodníky sdílejí jednu komunikační sběrnici, příkazem si musíme příslušný kanál vybrat. Dle návrhu také musíme zvolit zesílení na vstupu a to patřičným způsobem do zesilovače zapsat. Poté teprve spustit konkrétní převodník. Když je převod dokončen, je vyvoláno přerušování a získaná hodnota s patřičným identifikátorem odeslána zpět, jako reakce na požadavek.

### RS-485

Mezi periferiemi, kterými deska též disponuje je sběrnice RS-485. Pro tuto knihovnu je navržena pro využití odesílání stavových a ladících dat. Jelikož je realizována pomocí sériové linky v mikrokontroléru, který podporuje i některé vybrané protokoly nebude problém ji jednoduše přizpůsobit daným podmínkám.



## ■ 6.2 RaspberryPi

Na tuto platformu musí být vytvořena též odpovídající a z pohledu myšlenky stěžejní knihovna, resp. celá aplikace. Musí se skládat z komunikační a vizualizační části.

### Komunikace

Musí umět zajišťovat komunikaci po SPI k rozšiřující desce s mikrokontrolérem a periferiemi. Odráží v sobě veškerou funkcionalitu, kterou nabízí deska a to v podobě dostupných funkcí, které jsou jí definovány. Obsahuje v sobě ukrytou komunikaci po SPI, kterou spravuje. Vytváří a dekoduje jednotlivé příkazy, které na základě konfiguračního souboru poskytnutého uživatelem zpracovává.

Jak bylo řečeno jsou dány funkce, které jsou zavedené v našem případě k ovládání periferií. Tedy jako nastavení digitálních výstupů načtení analogové hodnoty a dalších. Jelikož program počítá s tvorbou za pomoci Qt frameworku, jsou pro jednotlivé funkce vytvořeny sloty pro uživatelské události. Které se dle jednoduchého textového konfiguračního souboru vzájemně propojí. Tomu se tak děje jak v případě vyslání požadavku, tak i při přijetí odezvy se dle tohoto souboru odezva může promítnout do uživatelského rozhraní.

V této části jsou ukryté také časové funkce, které jsou běžně volány periodicky. Příkladem může být načítání analogové hodnoty v nějakém intervalu. Tato funkcionalita je odděleně zaznamenána též v konfiguračním souboru.

### Tvorba prostředí

Vytvoření uživatelského rozhraní je ponecháno na prostředí Qt Creatoru. Uživatel si vytvoří okenní aplikaci a do té si jednoduše pomocí myši „nakliká“ dovolené prvky, typu „labelu, inputů, buttonů“. Jednoduše si sestaví prostředí dle svých požadavků a o zpracování událostí na úrovni programovacího jazyka se vůbec nestará. Jedinou povinností zásahu do kódu je načtení námi navrhované knihovny k obstarání komunikace a propojení událostí a prvků.

## 6.2 RASPBERRYPI

Po vytvoření formuláře je na uživateli využití připraveného parseru na XML soubor, který v sobě právě obsahuje informace o vytvořeném formuláři. Ten jej zpracuje a uživateli vygeneruje jednoduchý textový konfigurační soubor, ve kterém nabídne dostupné funkce k využití. Další krok už je na uživateli, na jakou akci, kterou funkcionalitu bude požadovat a kam se případně projeví odezva o provedení. Toto přiřazení spočívá v uvedení některé z funkce do příslušného místa v propojovací tabulce konfiguračního souboru.

Vytvořenou aplikaci v Qt Creatoru, bude nutno zkompileovat. Pro její použití dále už jen spustit s parametrem, kterým bude název příslušného konfiguračního souboru. Aplikace si jej načte, zpracuje a uživateli nabídne prostředí, které si vytvořil a to s chováním, které si nadefinoval v onom souboru.

## Kapitola 7

# Závěr

Cílem této práce bylo navrhnout a vytvořit programovatelný logický automat s grafickým výstupem a danými parametry. V úvodu se práce zabývá stručným přehledem, možnostmi a zaměřením různých programovatelných logických automatů.

S ohledem na naše požadavky a existující řešení, je práce založena a realizována na platformě RaspberryPi. Na tomto řešení je založen zejména grafický výstup, doplněný o námi navrženou rozšiřující desku s mikrokontrolérem, disponující dalšími potřebnými perifériemi.

K oživení zhotoveného programovatelného automatu byly vytvořeny ovládací programy pro jednotlivé obsažené periferie a jejich řízení. Za účelem vizualizace a ovládání byla realizována komunikace mezi rozšiřující deskou a RaspberryPi. Funkčnost celého navrženého automatu byla ověřena úlohou měření na Thompson-Lampardově etalonu kapacity. Ke kterému byla vytvořena účelová řídicí aplikace k řízení tohoto měření.

Podářilo se tedy v rámci této práce zhotovit logický programovatelný automat, který přímo disponuje grafickým výstupem na běžně dostupný LCD displej. Kde je možno vytvářet vizualizační prostředí pomocí běžných programovacích prostředí. Výhodou je, že RaspberryPi je obdobou PC tak jak je známe a to včetně operačního systému. Díky disponující konektivité je možné též využívat vzdáleného přístupu k jeho správě.

Během testování navrženého systému se projevila nevýhoda řešení a to konkrétně v případě galvanického oddělení pomocí zvolených optočlenů. Ačkoliv toto řešení zajišťuje robustnost konstrukce, může v časově kritických aplikacích způsobovat problémy ve zpoždění. V těchto případech by bylo vhodnější zvolit galvanické oddělení na např. kapacitním principu.

Navržena byla také jednoduchá univerzální knihovna pro vizualizaci stavových dat a ovládání. Měla by umožnit vytvořit ovládací a vizualizační aplikaci i laikům za pomoci

## 7.0 ZÁVĚR

konfiguračních souborů. Avšak zpracování dat a událostí na straně RaspberryPi pro toto zjednodušení se už nestihlo realizovat. I bez této funkce může automat bez problémů fungovat, jen je potřeba vytvářet aplikace kompletně v programovacím jazyce.

# Literatura

- [1] William Bolton. *Programmable Logic Controllers*. Newnes, 2009.
- [2] [http://www.micropel.cz/images/stories/Dokumenty/mpc223\\_tech.pdf](http://www.micropel.cz/images/stories/Dokumenty/mpc223_tech.pdf), Platné k 4. 5. 2015.
- [3] <http://shop.loxone.com/cscz/miniserver.html>, Platné k 4. 5. 2015.
- [4] <http://www.unitronics.com/plc-hmi/plc-vision-enhanced/v430->, Platné k 4. 5. 2015.
- [5] <http://www.raywenderlich.com/44918/raspberry-pi-airplay-tutorial>, Platné k 8. 4. 2015.
- [6] <http://archive.linuxgizmos.com/smallest-lowest-power-arm-core-debuts/>, Platné k 8. 4. 2015.
- [7] [http://www.analog.com/media/en/technical-documentation/data-sheets/ADM2582E\\_2587E.pdf](http://www.analog.com/media/en/technical-documentation/data-sheets/ADM2582E_2587E.pdf), Platné k 8. 4. 2015.
- [8] <http://www.st.com/web/en/resource/technical/document/datasheet/CD00002345.pdf>, Platné k 8. 4. 2015.
- [9] [http://www.4dsystems.com.au/product/uLCD\\_35DT/](http://www.4dsystems.com.au/product/uLCD_35DT/), Platné k 4. 5. 2015.
- [10] <http://www.st.com>, Platné k 4. 5. 2015.
- [11] <http://mujweb.cz/fuksam/navrhy/krokmot/krokmot.htm>, Platné k 4. 5. 2015.
- [12] W. K. CLOTHIER. A calculable standard of capacitance. *Metrologia*, 1(2):36–55, 4 1968.



# Seznam použitých zkratek

*A/D* - Analog-to-digital

*ARM* - Advanced RISC Machines

*CAN* - Controller Area Network

*CEC* - consumer electronic control

*DDC* - Display Data Channel

*DMOS* - Double-diffused metal-oxide-semiconductor

*DVI* - Digital Visual Interface

*ESD* - Electrostatic discharge

*GPIO* - General-purpose input/output

*GUI* - Graphical User Interface

*HDMI* - High-Definition Multi-media Interface

*HMI* - Human Machine Interface

*I/O* - Input/output

*I<sup>2</sup>C* - Inter-Integrated Circuit

*OpenGL* - Open Graphics Library

*OpenGL ES* - OpenGL for Embedded Systems

*OPLC* - Operator panel & Programmable Logic Controller

*PCM* - Pulse-code modulation

*PLC* - Programmable Logic Controller

*PWM* - Pulse Width Modulation

*RISC* - Reduced Instruction Set Computing

*SCL* - Serial Clock Line

*SDA* - Serial Data Line

*SDC* - Secure digital card

## 7.0 SEZNAM POUŽITÝCH ZKRATEK

*SoC* - System on chip

*SPI* - Serial Peripheral Interface

*SRAM* - Static Random Access Memory

*SSH* - Secure Shell

*TMDS* - Transition-minimized differential signaling

*TWI* - Two Wire Interface

*USART* - Universal Synchronous Asynchronous Receiver and Transmitter

*USB* - Universal Serial Bus

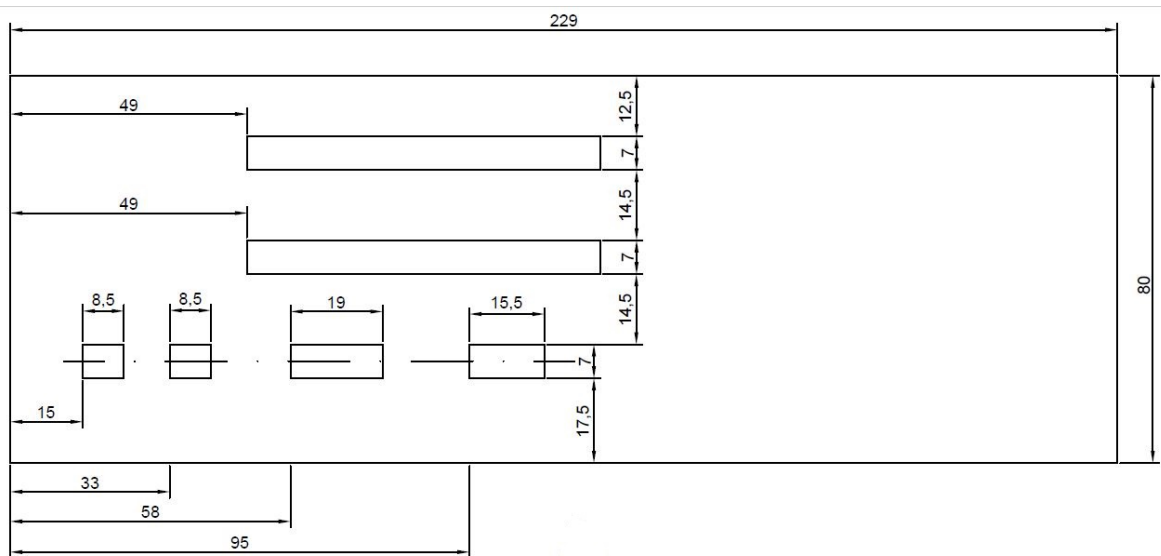
*VGA* - Video Graphics Array



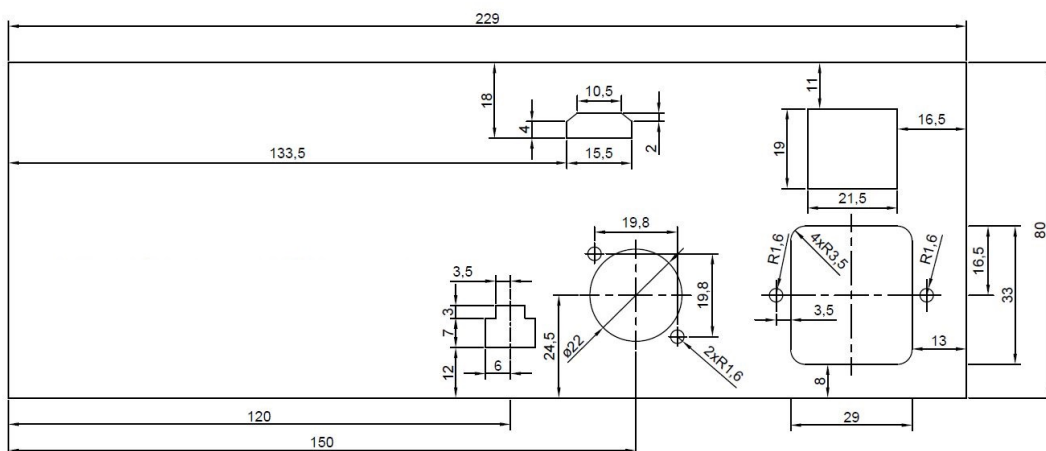
# ■ Přílohy

## ■ A Výkresy

### Přední panel



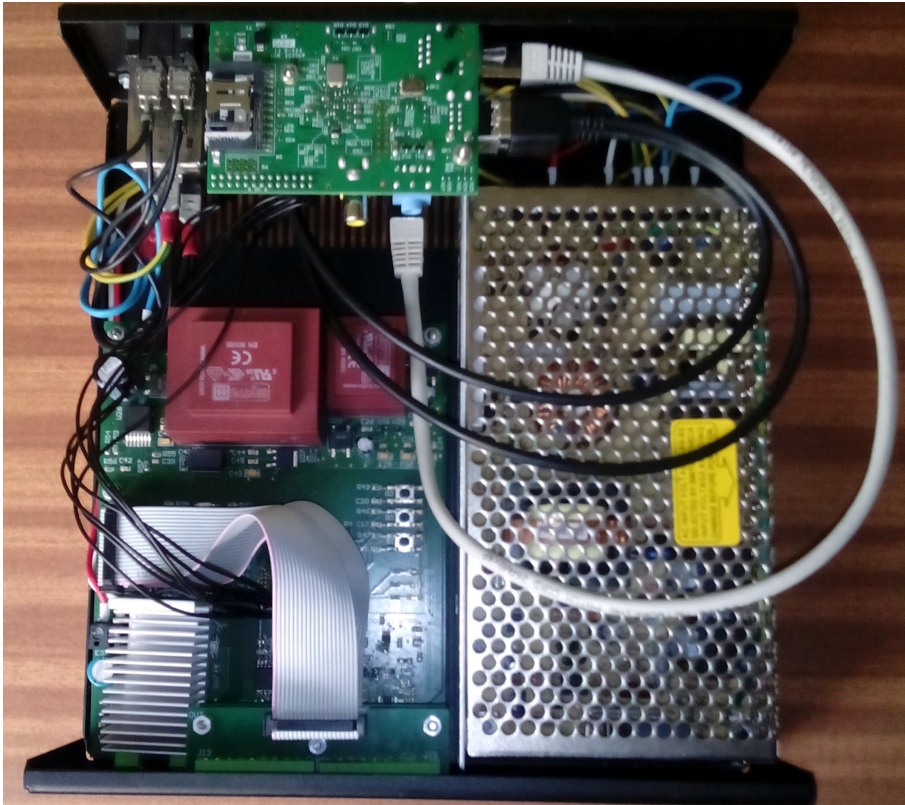
### Zadní panel



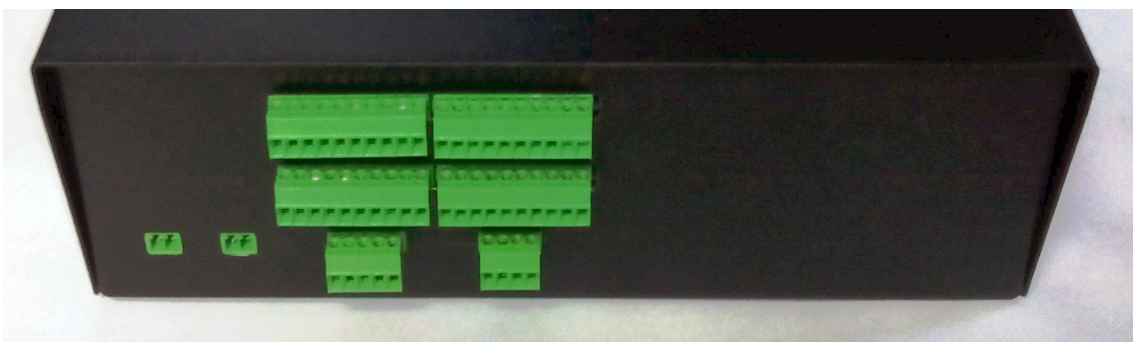


## ■ B Mechanické provedení

Vnitřní uspořádání



Pohled zepředu

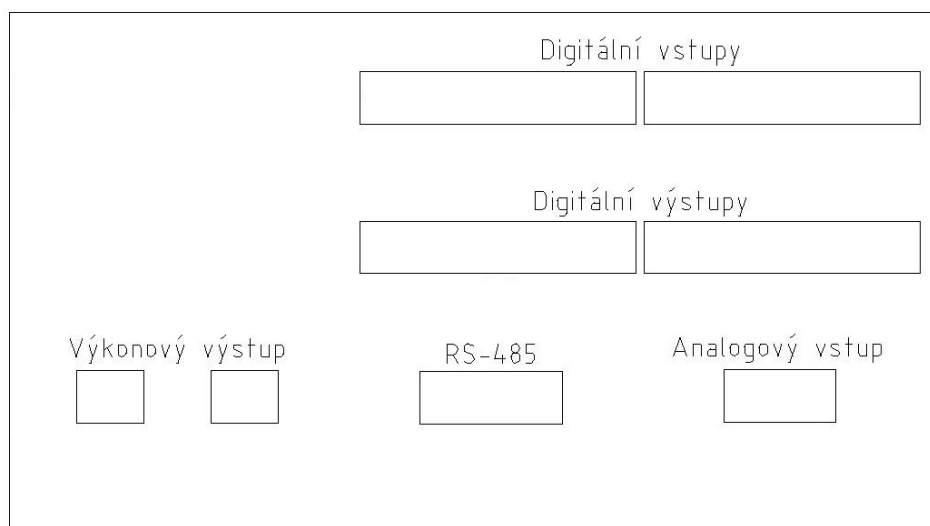


## B MECHANICKÉ PŘÍKONENÍ

### Pohled zezadu



### Rozmístění konektorů



## Popis konektorů

### Digitální výstupy

0	GND	1	GND	2	GND	3	GND	4	GND	5	GND	6	GND	7	GND	8	GND	9	GND
---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----

### Digitální vstupy

0	GND0	1	GND1	2	GND2	3	GND3	4	GND4	5	GND5	6	GND6	7	GND7	8	GND8	9	GND9
---	------	---	------	---	------	---	------	---	------	---	------	---	------	---	------	---	------	---	------

### Výkonový výstup

B 1	B 2
-----	-----

A 1	A 2
-----	-----

### RS-485

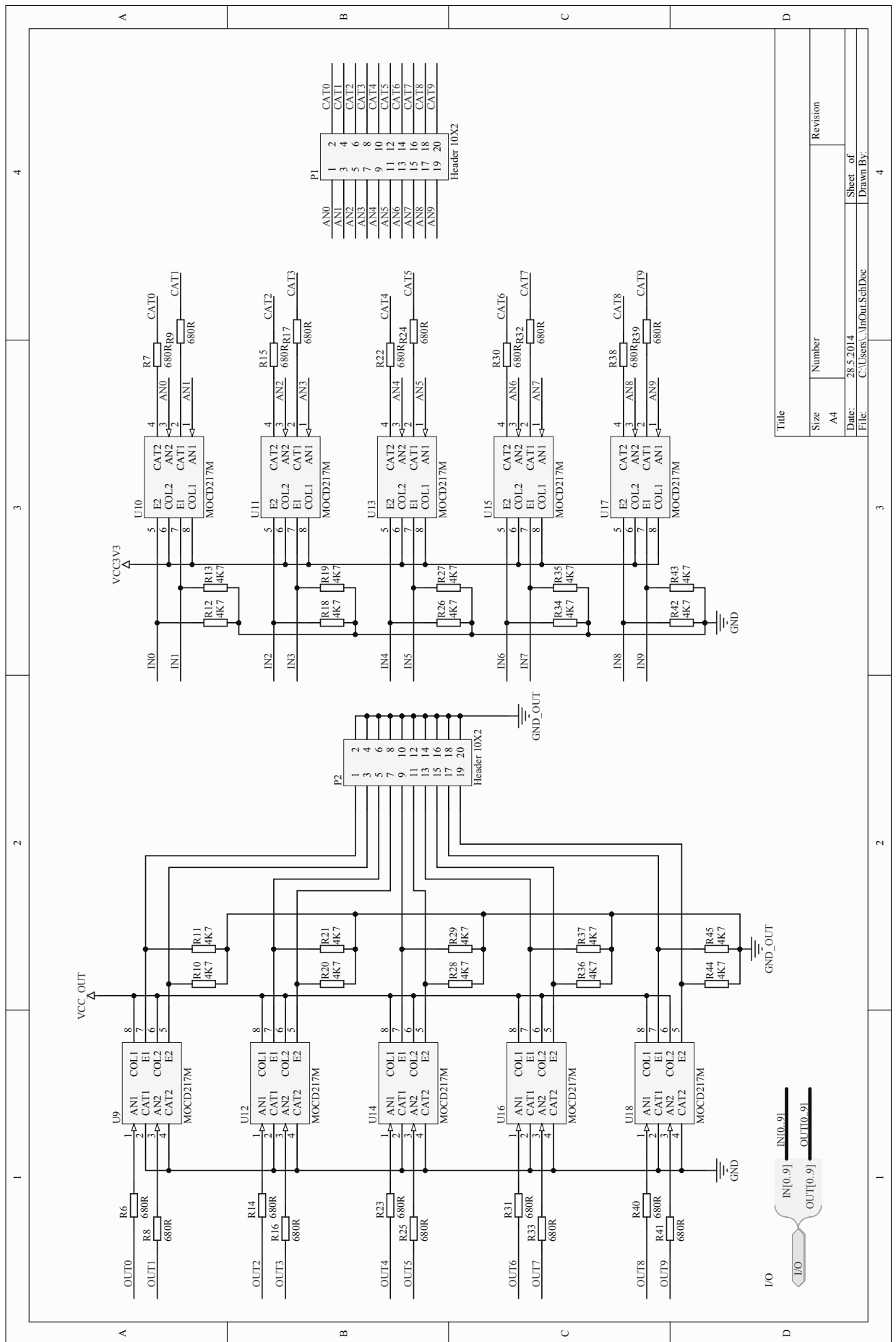
A	B	GND	Z	Y
---	---	-----	---	---

### Analogové vstupy

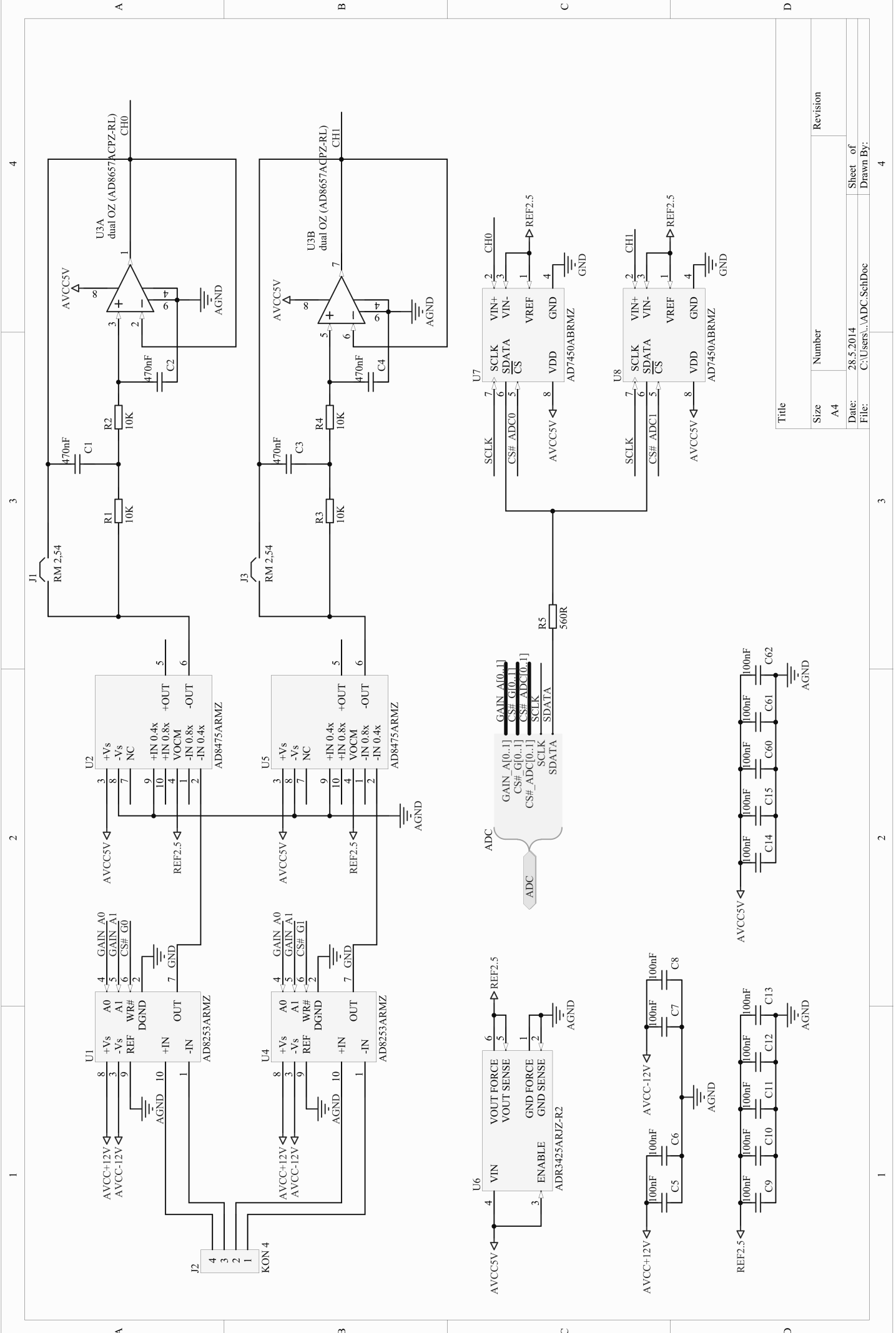
IN 1 -	IN 1 +	IN 2 -	IN 2 +
--------	--------	--------	--------



# C Schémata zapojení

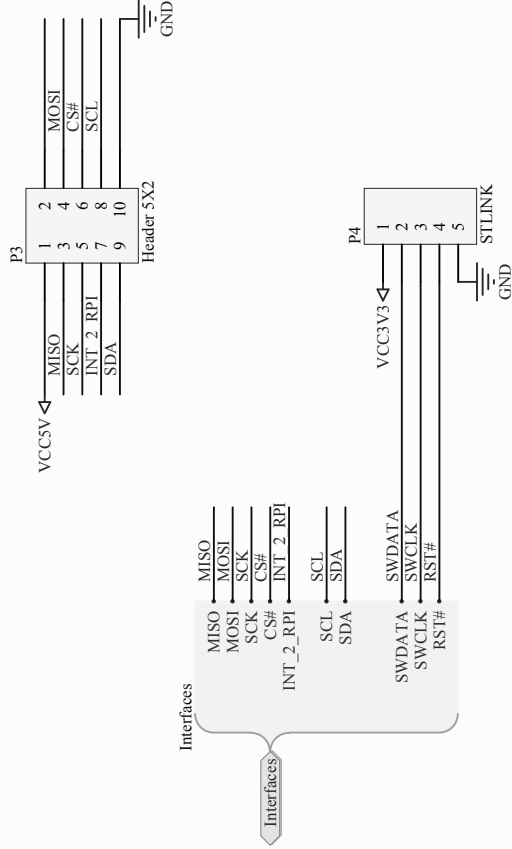


Title		Revision	
Size	A4	Number	
Date:	28.5.2014	Sheet of	
File:	C:\Users\...InOutSchDoc	Drawn By:	



Title	
Size	Number
A4	
Date:	Revision
28.5.2014	
File:	Sheet of
C:\Users\...ADC_SchDoc	Drawn By:
	4





Title

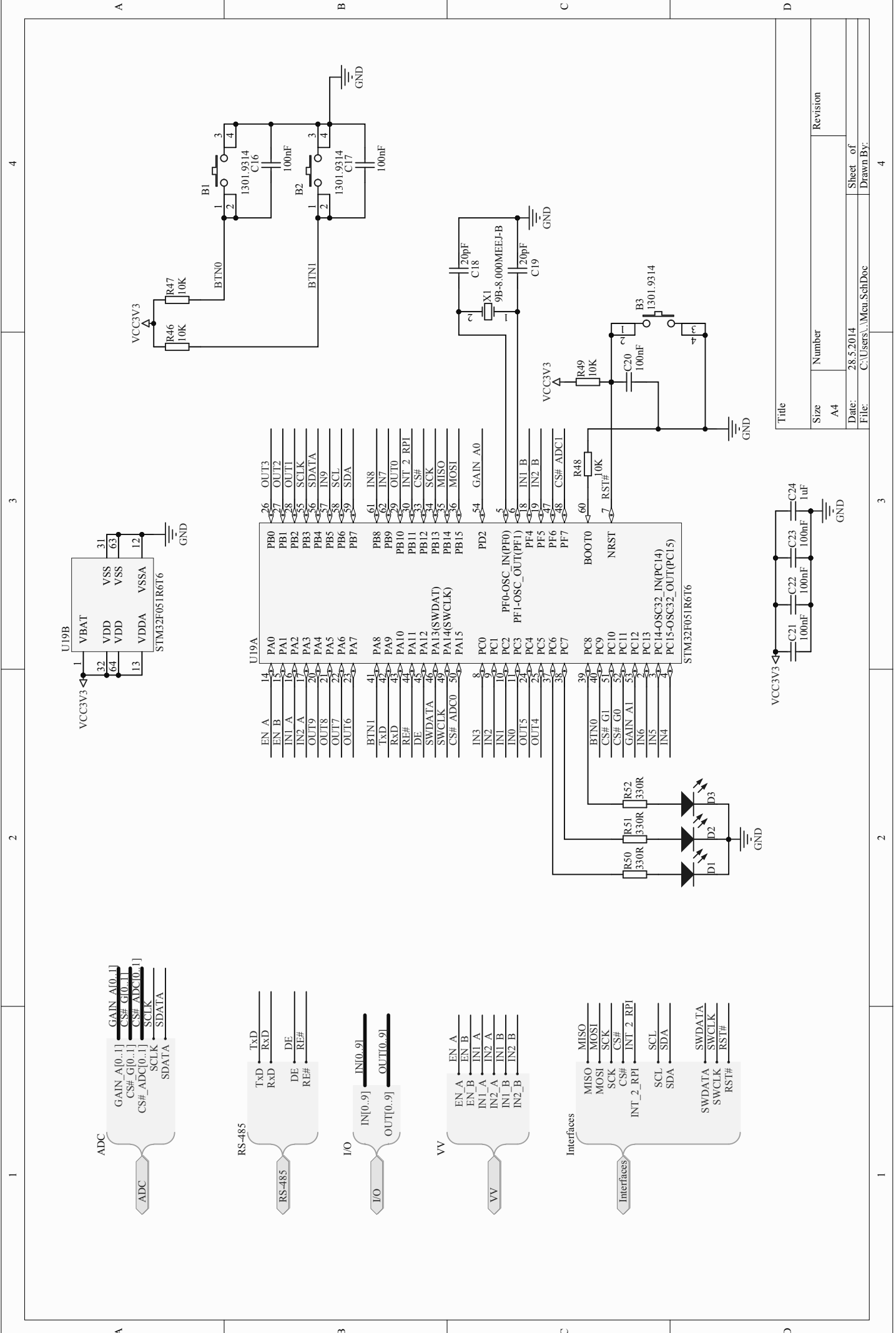
Revision

Size  
A4

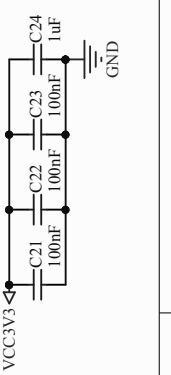
Date: 28.5.2014  
File: C:\Users\...\Interfaces\SchDoc

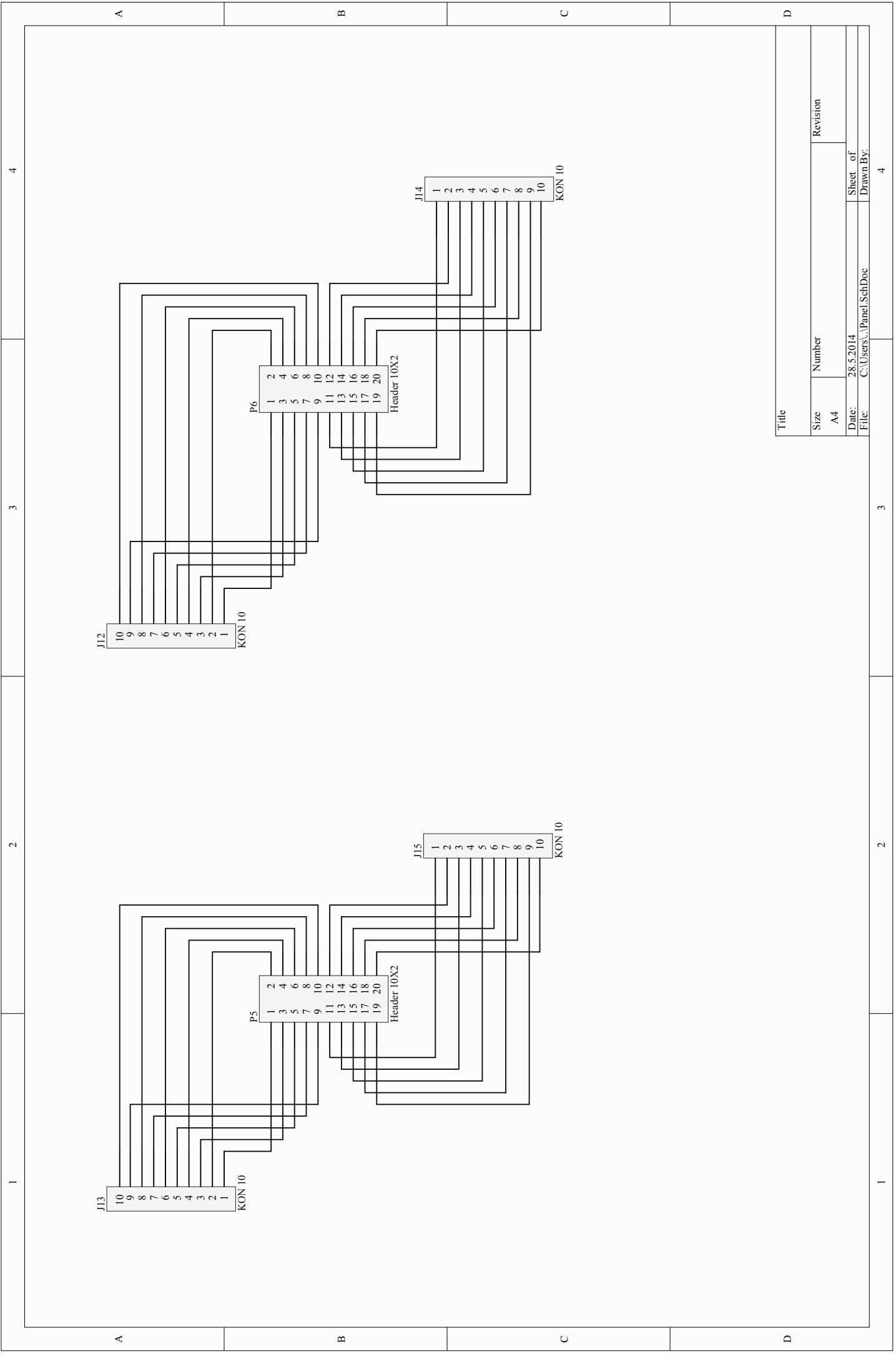
Sheet of  
4

Drawn By:



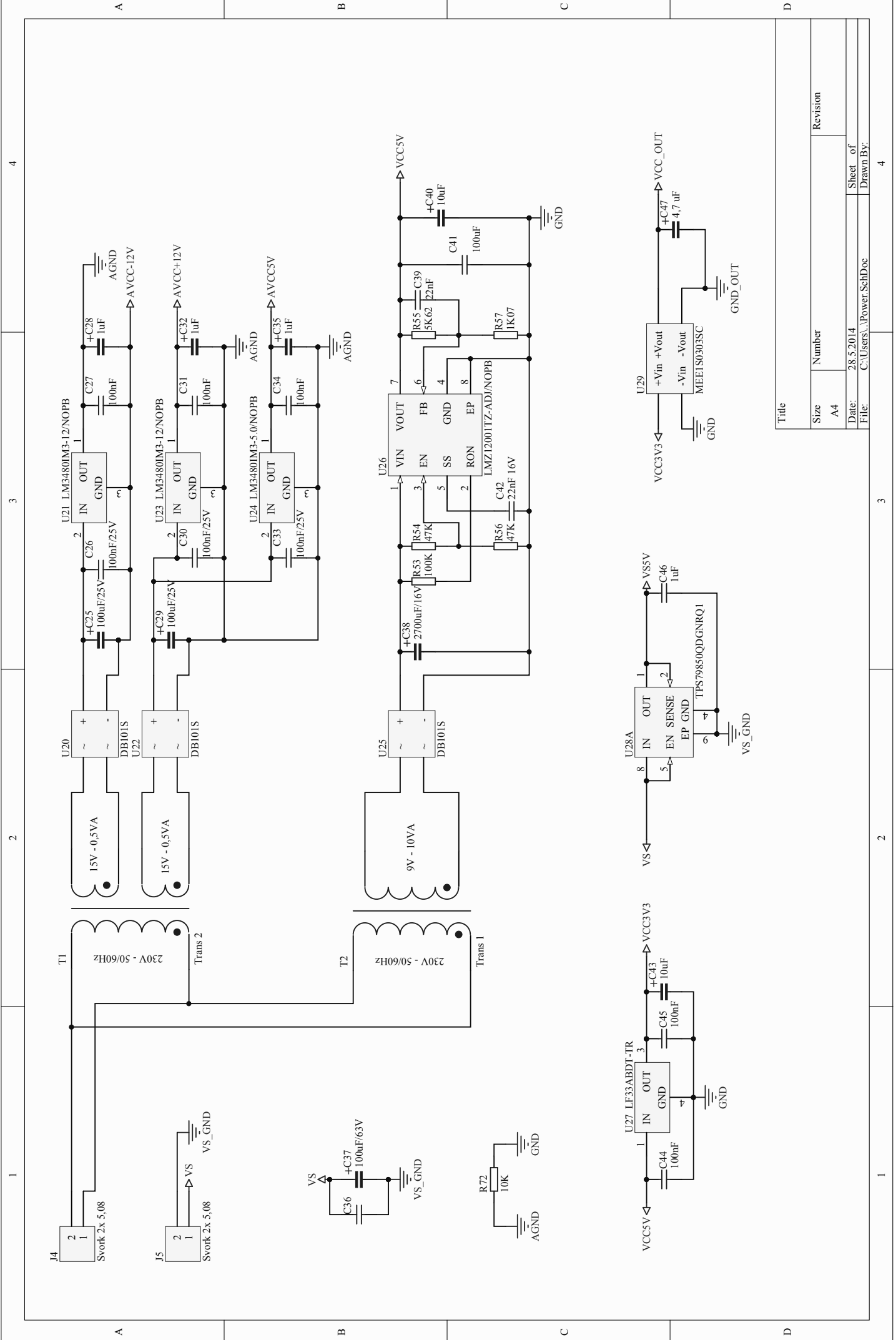
Title	
Size	Number
A4	
Date:	Revision
28.5.2014	
File:	Sheet of
C:\Users\...\Mcu_SchDoc	Drawn By:
	4



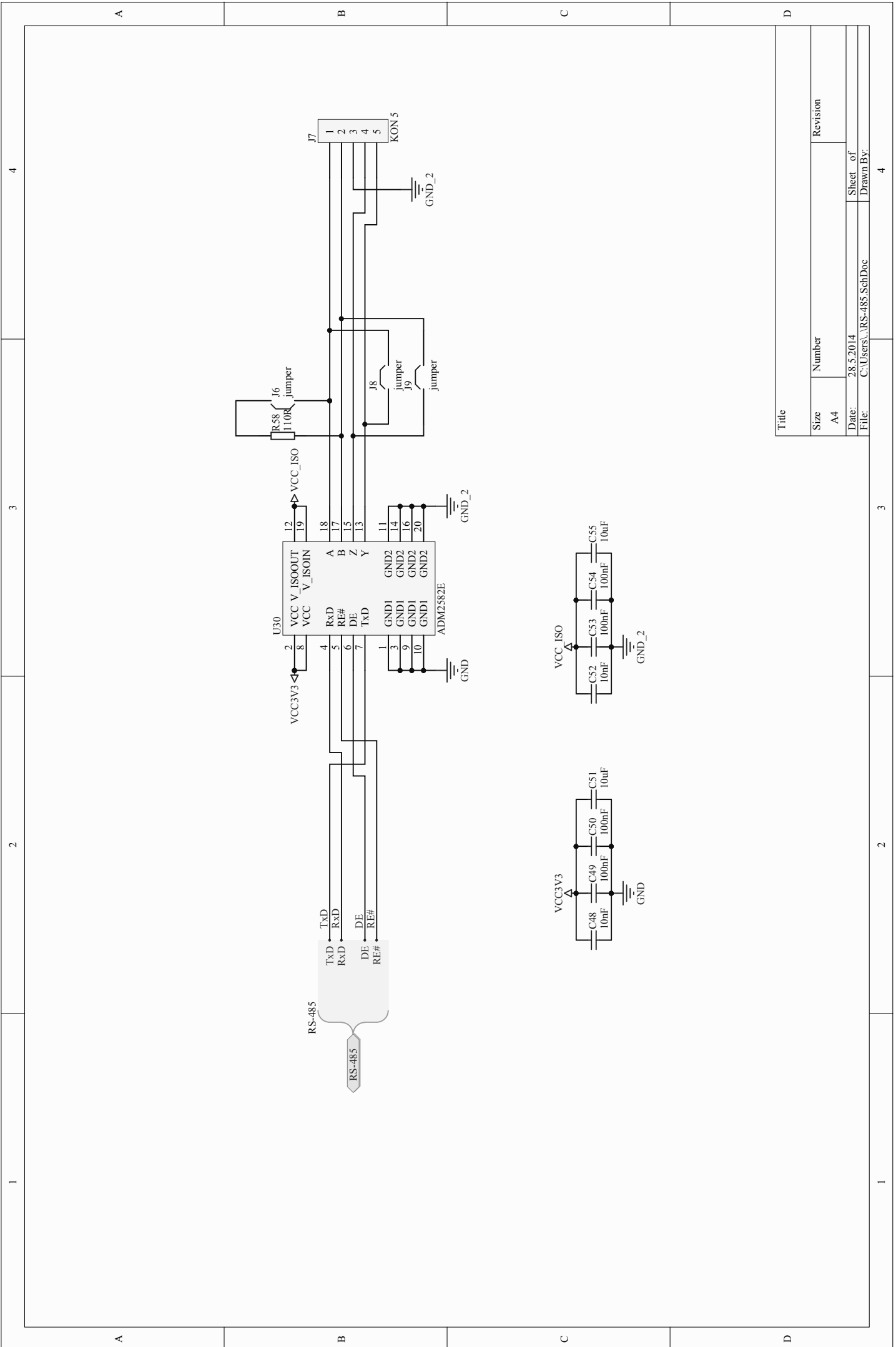


Title

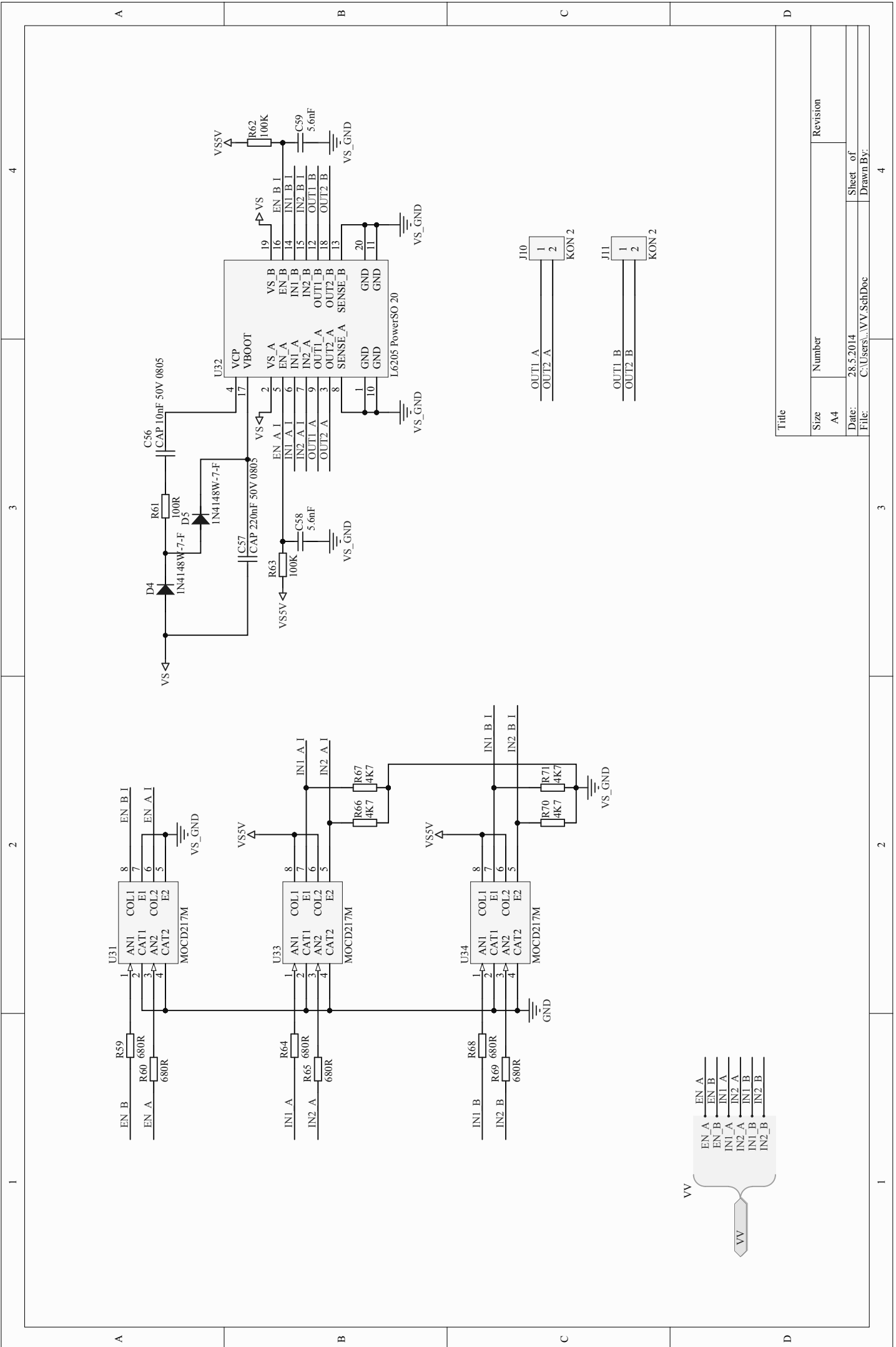
Size	Number	Revision
A4		
Date:	28.5.2014	Sheet of
File:	C:\Users\...\Panel\SchDoc	Drawn By:



Title	
Size	Number
A4	
Date:	Revision
28.5.2014	
File:	Sheet of
C:\Users\...Power.SchDoc	Drawn By:



Title	
Size	Revision
A4	
Date:	28.5.2014
File:	C:\Users\...RS-485.SchDoc



4

3

2

1

Title	
Size	Number
A4	A4
Date:	Revision
28.5.2014	
File:	Sheet of
C:\Users\...VV_SchDoc	Drawn By:
	4

3

2

1

