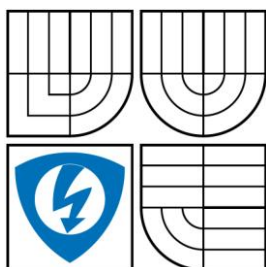


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECH-  
NOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## PŘEVODNÍK MEZI SBĚRNICEMI I2C A RS232

I2C-RS232 CONVERTER

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

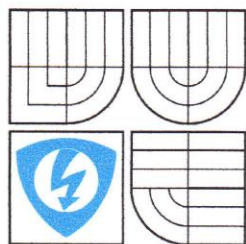
AUTOR PRÁCE  
AUTHOR

MARTIN BEDNAŘÍK

VEDOUCÍ PRÁCE  
SUPERVISOR

doc. Ing. VÁCLAV ZEMAN, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
**Teleinformatika**

**Student:** Martin Bednařík

**ID:** 73091

**Ročník:** 3

**Akademický rok:** 2008/2009

## NÁZEV TÉMATU:

**Převodník mezi sběrnicemi I2C a RS232**

## POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte převodník mezi sériovou sběrnicí RS232 a sběrnicí I2C, na které jsou zapojeny obvody s vstupně/výstupními branami PCF 8574. Navrhněte komunikační protokol. Zařízení bude osazeno mikrokontrolérem ATMEL a ovládací program bude napsán v jazyce "C". Projekt řešte ve spolupráci s pracovníky vývojového centra Honeywell v Brně.

## DOPORUČENÁ LITERATURA:

- [1] Kainka, B. : Využití rozhraní PC, HEL, 1996. ISBN 80-902059-3-3
- [2] Mann, B. : C pro mikrokontroléry, BEN, 2003. ISBN 80-7300-077-6
- [3] Kainka, B. : Elektronika s podporou PC, BEN, 2004. ISBN 80-86167-22-4

**Termín zadání:** 9.2.2009

**Termín odevzdání:** 2.6.2009

**Vedoucí práce:** doc. Ing. Václav Zeman, Ph.D.

**prof. Ing. Kamil Vrba, 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 § 152 trestního zákona č. 140/1961 Sb.

## Anotace

Cílem tohoto projektu je obeznámit se s funkcí sběrnice I<sup>2</sup>C a mikrokontroléry AVR a s principem propojení mikrokontroléru s ostatními obvody pomocí této sběrnice. Dále navrhnout takové zařízení, které bude schopno zprostředkovat komunikaci mezi osobním počítačem a ovládanými obvody.

Součástí projektu je tvorba potřebného programového vybavení pro mikrokontrolér a návrh vlastního komunikačního protokolu.

Dalším cílem projektu je získat přehled o použitých technologiích, možnostech jejich využití a případného rozšíření zařízení o další prvky.

## Abstract

The objective of this thesis is to get familiar with function of I<sup>2</sup>C bus and microcontrollers AVR and study the principle of connecting microcontroller with the other circuits via this bus. Another goal is to design a device which will enable a communication between personal computer and controlled circuits.

A part of this project is production of necessary microcontroller software equipment and to develop own communication protocol.

Another output of this project is getting an overview of applied technologies, study possibilities of their additional usage and potential enlargement of device with another components.

## Klíčová slova

Mikrokontrolér, I<sup>2</sup>C, TWI, sběrnice, Atmel, AVR, ATmega8, RS232, MAX232, USART, SDA, SCL, PCF8574, PCF8591, START podmínka, STOP podmínka

## Key words

Microcontroller, I<sup>2</sup>C, TWI, bus, Atmel, AVR, ATmega8, RS232, MAX232, USART, SDA, SCL, PCF8574, PCF8591, START condition, STOP condition

**Bibliografická citace mé práce**

Bednařík, Martin. *Převodník mezi sběrnicemi I2C a RS232*. Brno : Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 51 stran, 3 přílohy. Vedoucí bakalářské práce doc. Ing. Václav Zeman, Ph.D.

## Prohlášení

Prohlašuji, že svou bakalářskou práci na téma „Převodník mezi sběrnice I<sup>2</sup>C a RS232“ jsem vypracoval samostatně pod vedením vedoucího semestrálního projektu 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 uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 § 152 trestního zákona č. 140/1961 Sb.

V Brně dne .....

.....

podpis autora

## Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Václav Zeman, Ph.D. a panu Ing. Ondřeji Pavelkovi za poskytnutí cenných rad a připomínek při zpracování mé bakalářské práce.

## Obsah

<b>ÚVOD</b> .....	<b>8</b>
<b>1 SBĚRNICE I<sup>2</sup>C</b> .....	<b>9</b>
1.1 HISTORIE .....	9
1.2 DESIGN .....	9
1.3 LINKOVÁ VRSTVA .....	10
1.4 FYZICKÁ VRSTVA .....	11
1.5 MULTI-MASTER SYSTÉMY .....	12
<b>2 ROZHRANÍ RS232</b> .....	<b>14</b>
2.1 ÚVOD .....	14
2.2 NAPĚŤOVÉ VLASTNOSTI .....	14
2.3 KONEKTORY .....	15
2.4 SIGNÁLY .....	15
2.5 ASYNCHRONNÍ KOMUNIKACE .....	16
<b>3 ATMEL AVR</b> .....	<b>17</b>
3.1 ÚVOD .....	17
3.2 ARCHITEKTURA AVR MIKROKONTROLÉRŮ .....	17
<b>4 VÝBĚR KOMPONENT PRO ZAŘÍZENÍ</b> .....	<b>19</b>
4.1 POŽADAVKY .....	19
4.2 OVLÁDACÍ MIKROKONTROLÉR .....	19
4.2.1 Registr přenosové rychlosti (TWBR) .....	20
4.2.2 Control registr (TWCR) .....	20
4.2.3 Status registr (TWSR) .....	20
4.2.4 Datový registr (TWDR) .....	20
4.2.5 Adresový registr (TWAR) .....	21
4.3 MOŽNOSTI PROGRAMOVÁNÍ MIKROKONTROLÉRU .....	21
4.3.1 JTAG programátor .....	22
4.3.2 ISP programátor .....	22
4.4 I <sup>2</sup> C EXPANDER .....	23
4.5 AD/DA PŘEVODNÍK .....	24
4.6 PŘEVODNÍK TTL↔RS232 .....	24
4.7 ANALOGOVÝ MULTIPLEXOR .....	25
<b>5 NÁVRH ZAŘÍZENÍ</b> .....	<b>27</b>
5.1 NAPÁJECÍ ZDROJ .....	27
5.2 KOMUNIKACE S MIKROKONTROLÉREM (HW ÚROVEŇ) .....	28
5.3 ZAPOJENÍ MIKROKONTROLÉRU ATMEGA8 .....	29
5.4 ZAPOJENÍ AD/DA PŘEVODNÍKU .....	30
5.5 ZAPOJENÍ I <sup>2</sup> C EXPANDERU .....	31
5.6 CELKOVÉ SCHÉMA ZAPOJENÍ .....	32
<b>6 NÁVRH DESKY PLOŠNÝCH SPOJŮ</b> .....	<b>32</b>
<b>7 NÁVRH PROGRAMU PRO MIKROKONTROLÉR</b> .....	<b>33</b>
7.1 BLOKOVÉ SCHÉMA PROGRAMU .....	33
7.2 POSTUP PROGRAMOVÁNÍ + UKÁZKA KÓDU .....	34
<b>8 OVLÁDÁNÍ PŘEVODNÍKU</b> .....	<b>36</b>
<b>9 ZÁVĚR</b> .....	<b>38</b>
<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>39</b>
<b>SEZNAM ZKRATEK</b> .....	<b>40</b>

SEZNAM PŘÍLOH.....	40
OBSAH CD .....	41
A SCHÉMA PŘEVODNÍKU .....	42
B DESKA PLOŠNÝCH SPOJŮ .....	48
C SEZNAM SOUČÁSTEK .....	51

## Seznam obrázků

Obr. 1.1: Design I2C sběrnice [1] .....	10
Obr. 1.2: Časový diagram I2C [1] .....	10
Obr. 1.3: Opakovaný start [1] .....	11
Obr. 1.4: Validita dat [1] .....	11
Obr. 1.5: Vnitřní struktura I <sup>2</sup> C rozhraní.....	12
Obr. 1.6: Vliv délky a zakončení sběrnice na tvar impulsů .....	12
Obr. 1.7: Synchronizace SCL hodin [1] .....	13
Obr. 1.8: Arbitrace mezi dvěma mastery [1].....	13
Obr. 2.1: Napěťové úrovně RS232 [10] .....	14
Obr. 2.2: Časový diagram RS232 (přenos jednoho znaku s jedním stop bitem a bez parity) .....	16
Obr. 3.1: Blokové schéma AVR architektury [1].....	17
Obr. 4.1: Mikrokontrolér ATmega 8 [1] .....	19
Obr. 4.2: Bity registru TWBR .....	20
Obr. 4.3: Bity registru TWCR.....	20
Obr. 4.4: Bity registru TWSR .....	20
Obr. 4.5: Bity registru TWDR .....	20
Obr. 4.6: Bity registru TWAR .....	21
Obr. 4.7: Blokové schéma TWI modulu [1] .....	21
Obr. 4.8: Blokové schéma zapojení programátoru.....	22
Obr. 4.9: Schéma ISP programátoru .....	22
Obr. 4.10: Adresy I <sup>2</sup> C expanderů .....	23
Obr. 4.11: Blokové schéma PCF8574 [8] .....	23
Obr. 4.12: Adresa I <sup>2</sup> C AD/DA převodníku .....	24
Obr. 4.13: Blokové schéma PCF8591 [9] .....	24
Obr. 4.14: Vnitřní struktura MAX232 [7] .....	25
Obr. 5.1 Blokové schéma zapojení převodníku.....	27
Obr. 5.2: Schéma zdroje .....	28
Obr. 5.3: Schéma zapojení převodníku MAX232.....	28
Obr. 5.4: Schéma zapojení mikrokontroléru .....	29
Obr. 5.5: Schéma zapojení obvodu pro změnu rozsahu měření napětí .....	30
Obr. 5.6: Schéma zapojení AD/DA převodníku .....	30
Obr. 5.7: Schéma zapojení výstupního expanderu.....	31
Obr. 5.8: Schéma zapojení vstupního expanderu .....	32
Obr. 7.1: Blokové schéma programu mikrokontroléru.....	33
Obr. 8.1: Schéma ovládání převodníku.....	37

## Seznam tabulek

Tab. 2.1: Napěťové úrovně RS232 .....	14
Tab. 2.2: Signály u DCE a DTE .....	15
Tab. 3.1: Nastavení předděličky .....	20
Tab. 4.1: Pravdivostní tabulka .....	25

## Úvod

Převodník mezi sběrnicemi I<sup>2</sup>C a RS232 využívá mikropočítač, který převede slovní příkaz v předem definovaném tvaru na binární číslo, které se vyšle přes dvou vodičovou sběrnici I<sup>2</sup>C k dalším obvodům jako jsou 4 expandery, jejichž prostřednictvím se budou spínat relé a AD/DA převodník, sloužící k nastavení či zjištění hodnoty napětí.

Rozbor mé bakalářské práce jsem rozdělil do osmi kapitol. V první popisuji samotnou I<sup>2</sup>C sběrnici, která bude tvořit hlavní část komunikace s ostatními zařízeními. Ve druhé se zabírám RS232 rozhraním, přes které se bude provádět ovládání a přes kterou se budou posílat příkazy z PC. Ve třetí kapitole rozebírám mikrokontrolér AVR, který zde tvoří právě onen převodník mezi sběrnicemi I<sup>2</sup>C a RS232. Čtvrtá a následující kapitoly se zabírají už samotným návrhem převodníku, výběrem vhodných komponent a ukázkou programování mikrokontroléru.



## 1 Sběrnice I<sup>2</sup>C



I<sup>2</sup>C (Inter-Integrated Circuit) je multi-masterová sériová sběrnice, vynalezená firmou Philips a používaná k připojení nízkorychlostních periférií jako jsou základní desky, obvody, nebo mobilní telefony. Podle stavu k datu 1. října 2006, nejsou za použití této sběrnice vyžadovány žádné licenční poplatky. Nicméně poplatky jsou vyžadovány u přiřazování I<sup>2</sup>C slave-adres (adresy slave zařízení).

SMBus (System Management Bus) je jistou podmnožinou I<sup>2</sup>C, která striktně definuje elektrické vlastnosti a tvar komunikačního protokolu. Cíl SMBus je zajistit odolnost a funkceschopnost. Moderní I<sup>2</sup>C systémy si přivlastnily politiku a pravidla právě od SMBus a rozdíly mezi těmito dvěma standardy jsou často v praxi zastírány.

### 1.1 Historie

I<sup>2</sup>C sběrnice byla vynalezena firmou Philips Semiconductors. Vznikla za účelem poskytovat snadné propojení CPU k periferním obvodům.

Periferní obvody v komplexních systémech jsou často připojovány k CPU za pomoci paralelní adresy a datové sběrnice. Což v konečném důsledku vyústilo v mnoho cest na desce plošných spojů, zvláště pro adresy a zvláště pro data. V masové produkci pro televize, videorekordéry a různá audio zařízení toto není přijatelné. V těchto zařízeních každá komponenta, která může být ušetřena, znamená nárůst zisku pro společnost, menší cenu zařízení a více cenově dostupných produktů pro koncového zákazníka. Mimoto, mnoho kontrolních linek naznačuje, že systémy jsou více citlivé k poruchám způsobenými elektromagnetickou interferencí a elektrickými výboji.

Výzkum vedený společností Philips Labs překonal tyto problémy a vyvinul dvou vodičovou komunikační sběrnici nazvanou I<sup>2</sup>C. Její jméno doslova vysvětluje její účel: poskytovat komunikační vedení mezi integrovanými obvody.

Dnes je I<sup>2</sup>C sběrnice používána v mnoha oblastech než pouze u video a audio zařízeních. Sběrnice je obvykle nasazována v průmyslu jako de-facto standard. I<sup>2</sup>C byla přijata několika společnostmi zabývajícími se výrobou mikročipů, např. Xicor, ST Microelectronics, Infineon Technologies, Intel, Texas Instruments, Maxim, Atmel.

### 1.2 Design

I<sup>2</sup>C používá pouze dva obousměrné vodiče. Jeden se nazývá SDA (Serial Data) a druhý SCL (Serial Clock), na které jsou připojeny tzv. Pull-Up rezistory, sloužící k zajištění log. 1, pokud se po sběrnici nekomunikuje. Typické napětí jsou +5V a +3,3V, ale i systémy s jiným, vyšším či nižším napětím, jsou povolené také.

I<sup>2</sup>C má 7bitový adresový prostor s 16 rezervovanými adresami, a tak maximum možných počet připojených zařízení je 112 ( $2^7 - 16 = 112$ ). Nejběžnějšími rychlostmi používanými na této sběrnici jsou 100 kbit/s (standart mode) a 10kbit/s (low-speed mode). Poslední revize I<sup>2</sup>C může adresovat více zařízení (10bitové adresy) a lze po ní komunikovat rychleji, např. 400 kbit/s (Fast mode), 1Mbit/s (Fast mode+, Fm+) a 3,4Mbit/s (High-speed mode)

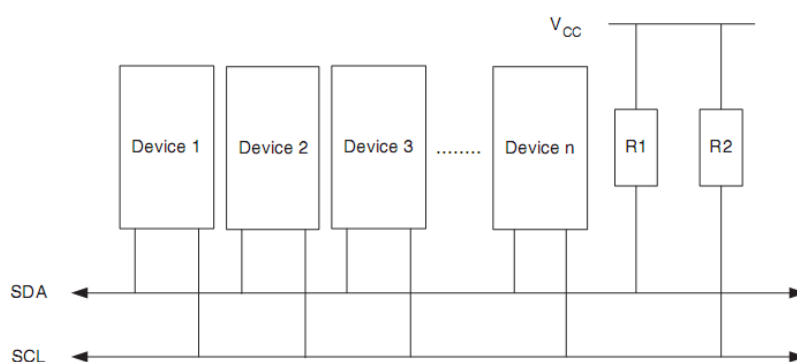
Maximální počet zařízení je limitován nejen adresovým prostorem, ale také celkovou kapacitou sběrnice, která může být maximálně 400pF, což omezuje možnost komunikace pouze na pár metrů.

Uzly na sběrnici mohou být dvojího druhu:

- Master uzel – uzel poskytující hodinový signál a data
- Slave uzel – uzel přijímající hodinový signál a data

Jak již bylo zmíněno, tak sběrnice je multi-masterová, což znamená, že zde může být více než jeden master uzel. Role master a slave může být změněna mezi zprávami (po STOP podmínce). Existují 4 módy operací, ve kterých mohou zařízení pracovat, ale většina zařízení pracuje v jedné roli a dvou módech:

- Master transmit – master uzel odesílá data to slave,
- Master receive – master uzel přijímá data od slave,
- Slave transmit – slave uzel odesílá data do mastera,
- Slave receive – slave uzel přijímá data od mastera.



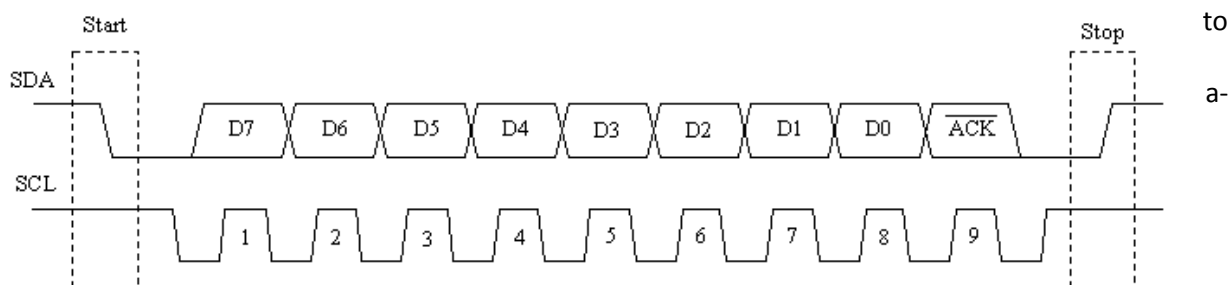
Obr. 1.1: Design I2C sběrnice [1]

### 1.3 Linková vrstva

Master je nejprve v master transmit módu. Pošle START BIT následovaný 7bitovou adresou slave zařízení, se kterou si přeje komunikovat. Za adresou odesílá ještě jeden bit (tzv. R/W) určující, zda bude do zařízení zapisovat (log. 0) nebo z něho číst (log. 1).

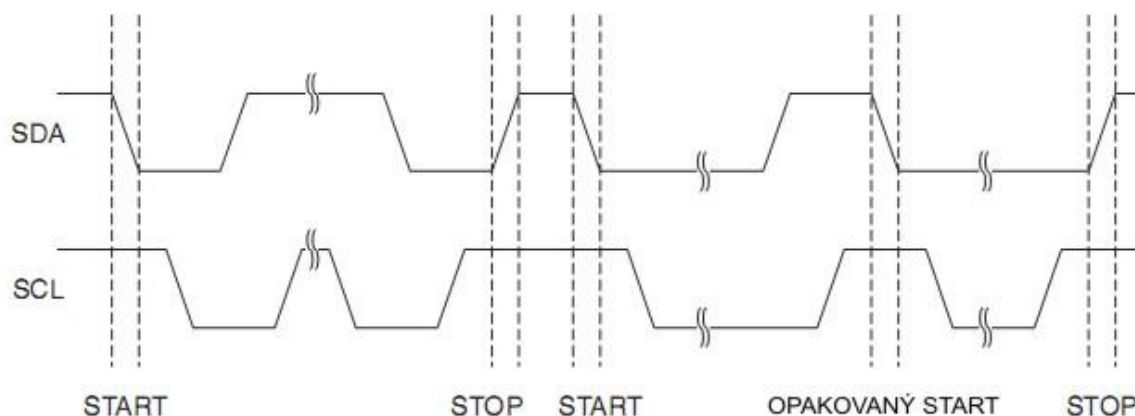
Příslušný přijímač potvrzuje příjem bajtu nízkou úrovní na SDA, dokud master nevyšle devátý hodinový impuls na SCL. Potvrzení současně znamená, že se může přijímat další bajt. Master, resp. slave po té může fungovat jako přijímač nebo jako vysílač (dle R/W bitu).

U adres a dat je prvně vyslán MSB bit. START BIT je indikován stažením SDA na nízkou úroveň (LOW), zatímco SCL je na vysoké úrovni (HIGH). STOP BIT je naopak indikován přejitím SDA z LOW (dále jen L) na HIGH, SCL zůstává na úrovni HIGH (dále jen H).



Obr. 1.2: Časový diagram I2C [1]

Master poté ukončí přenos STOP BITem nebo může odeslat další START BIT, pokud si přeje udržet kontrolu nad sběrnicí pro další přenos.



Obr. 1.3: Opakovaný start [1]

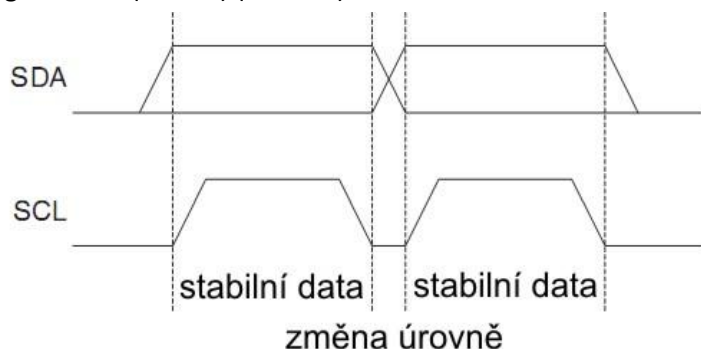
I<sup>2</sup>C definuje 3 typy zpráv:

- Jednoduchá zpráva, kde master zapisuje data do slave,
- Jednoduchá zpráva, kde master čte data ze slave,
- „Kombinovaná“ zpráva, kde master přijme alespoň dvě zprávy a/nebo zapisuje do jednoho či více slave zařízení.

V kombinované zprávě, každý zápis či čtení začíná START podmínkou a adresou slave zař. Po první START podmínce, jsou také vyslány „opakované START bity“, které ovšem nejsou předcházeny STOP bity, a tak slave zař. ví že další přenos je součástí téže zprávy.

## 1.4 Fyzická vrstva

Z elektrického hlediska jsou kanály SDA a SCL zapojeny jako otevřený kolektor, a právě proto jsou zde potřeba pull-up rezistory. Připojení sběrnice na zem je považováno za log. 0, zatímco ponechání sběrnice ve výchozím stavu je log. 1. Je to používáno jako přístup zařízení k médium. Vysokorychlostní systémy také přidávají zdroje proudu. Takováto podpora umožňuje rychlejší dobu náběhu a vyšší povolenou kapacitu sběrnice. Změna datových bitů je povolena pouze tehdy, když je hodinový signál v low (SCL v L) (obr. 1.4).



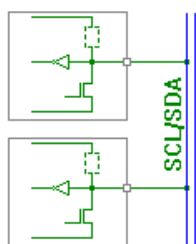
Obr. 1.4: Validita dat [1]

Změna těchto bitů, když je SCL v H, znamená podmínku START či STOP. Maximální frekvence signálu SCL je podle verze I<sup>2</sup>C, a to 100 kHz nebo 400 kHz. Pro obě frekvence je dána minimální povolená doba setrvání SCL v úrovni L a H. Při komunikaci i při přenosu dat si jednotlivé stanice synchronizují generátory hodin tak, že trvání úrovně H na SCL je odmě-

řováno vnitřním časovačem každé stanice až od okamžiku, kdy SCL

skutečně úrovně H dosáhne (protože je SCL typu otevřený kolektor, může být v úrovni L držen i v situaci, kdy se daná stanice snaží nastavit úroveň H).

Podobně je doba trvání úrovně L na SCL odměřována od sestupné hrany. Sběrnice I<sup>2</sup>C neumožňuje duplexní přenos, v jednom okamžiku vysílá jen jedno zařízení.



Obr. 1.5: Vnitřní struktura I<sup>2</sup>C rozhraní

Rozhraní sběrnice tvoří vstupní buffer a tranzistor s otevřeným kanálem nebo otevřeným kolektorem. Jak lze vidět na Obr. 1.5, tak pull-up rezistory v zařízení jsou vlastně malé zdroje proudu anebo také vůbec neexistují uvnitř obvodu.

Avšak princip otevřeného kolektoru má i stinnou stránku. Pokud bude sběrnice dlouhá, bude to mít vážný dopad na rychlost komunikace. Jak bylo řečeno v úvodu, tak maximální kapacita sběrnice může být 400pF. Tuto kapacitu vytvoří právě vodiče, a platí čím delší, tím větší kapacita. A pokud jsou připojeny pull-up rezistory (tedy pasivní prvek, zatímco zdroj proudu byl aktivní prvek), vznikne RC článek, který bude mít negativní dopad na ostrost hran obdélníkových pulsů. Čím vyšší bude hodnota RC konstanty, tím pomaleji můžeme po sběrnici komunikovat. Toto je daň za efekt, který ovlivňuje ostrost hran impulsů na sběrnici. A



Obr. 1.6: Vliv délky a zakončení sběrnice na tvar impulsů

tak od jistého bodu zařízení už nemusí rozpoznat rozdíl mezi úrovní L a H. Ba co víc, z konce sběrnice se může signál při vysokých rychlostech odrazet. Tyto odražené signály se sečtou s užitečným signálem, a tak mohou rušit přenos či poškodit samotná data. Dokonce ani Schmittův KO na vstupu sběrnice neeliminuje tento efekt.

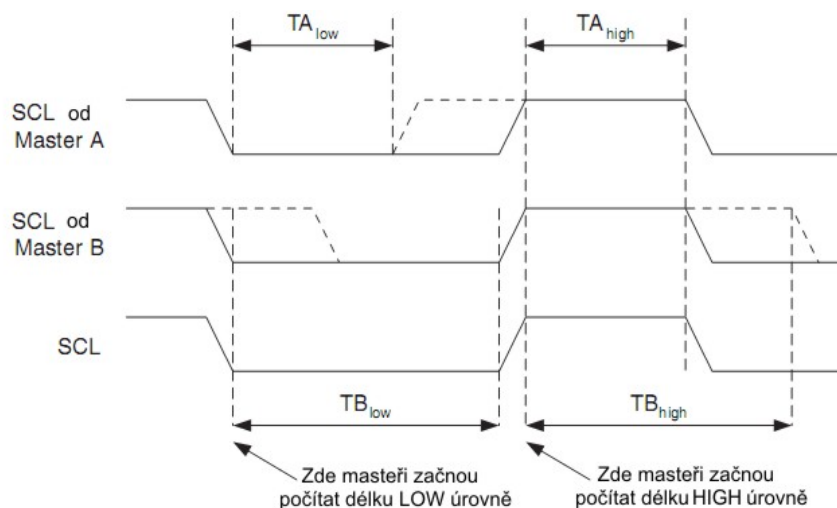
## 1.5 Multi-master systémy

Jelikož je sběrnice multi-masterová, tak v jednom momentě může přenos iniciovat více než jedno zařízení.

Tedy například pokud jeden uzel vysílá H a současně jiný vysílá L, tak právě uzel vysílající H toto pocítí, protože sběrnice není v HIGH (není připojena k Vdd). Tento mechanismus umožňuje některé ze stanic zpomalit přenos: pomalá stanice může podržet po určitou dobu signál SCL v úrovni L a tím zabránit vysílající stanici ve vyslání dalšího bitu. (angl. "Clock stretching")

Pokud je tento mechanismus použit na SDA, nazýváme to "Arbitrace" a je tím zabezpečeno, že v jednom momentě existuje pouze jeden vysílač.

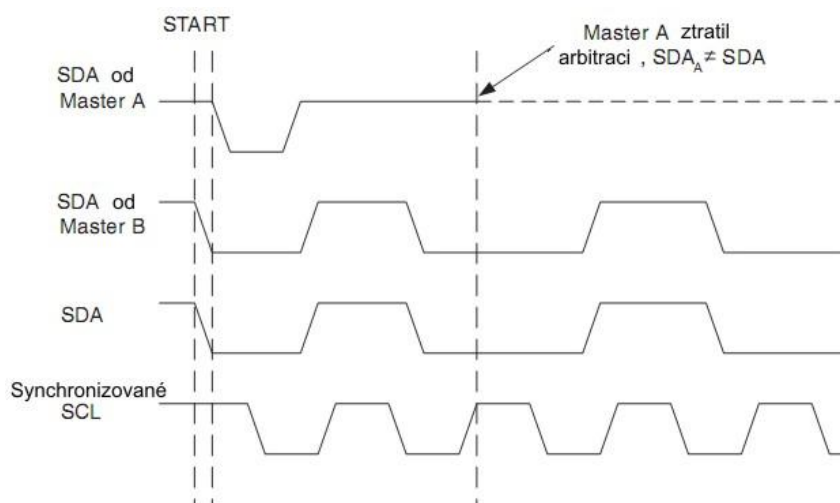
Jelikož existuje více masterů a jak bylo řečeno, tak každý master generuje hodinový signál, který u každého mastera může mít různou frekvenci, tak zde musí dojít k synchronizaci.



Obr. 1.7: Synchronizace SCL hodin [1]

Hodiny všech masterů jsou vynásobeny, což vytvoří nový hodinový signál, kde výsledná úroveň HIGH u SCL je rovna právě nejkratší periodě u všech masterů a výsledná úroveň LOW je rovna právě nejdelší periodě (Obr. 1.7).

Tedy všichni masteri musí naslouchat na SCL lince, k efektivnímu startu čítání LOW, HIGH a periody.



Obr. 1.8: Arbitrace mezi dvěma mastery [1]

Arbitrace je uskutečňována všemi mastery současně s monitorováním SDA linky po vyslání dat. Pokud hodnota přečtená na SDA nesehlasí s hodnotou, kterou master předtím vyslal, došlo ke ztrátě arbitrace. K tomu může dojít, pokud jeden master vyslal úroveň H, zatímco jiný vyslal L. Poté master, který se desynchronizoval, musí přejít do slave módu a

sledovat, zda je naadresován jiným masterem. SDA linka by měla zůstat v H úrovni, ale "ztraceným" masterům je dovoleno generovat hodinový signál, dokud nedojde k ukončení přenosu dat. Arbitrace bude pokračovat dokud nezůstane pouze jeden master, což může trvat i několik bitů. Pokud se několik masterů pokouší naadresovat jedno slave zařízení, arbitrace bude pokračovat až do vyslání dat.

## 2 Rozhraní RS232

### 2.1 Úvod

**RS232** (Recommended Standard 232) je standard pro sériový binární přenos propojující DTE (Data Terminal Equipment) a DCE (Data Circuit-terminating Equipment). Je obvykle používán v počítači, avšak v dnešní době už se od používání tohoto rozhraní již téměř definitivně ustoupilo a to bylo nahrazeno výkonnějším USB. Nicméně v průmyslu je tento standard, především jeho modifikace - standardy RS422 a RS485, velice rozšířen a pro své specifické rysy tomu tak bude i nadále. Na rozdíl od komplexnějšího USB, standard RS232 pouze definuje, jak přenést určitou sekvenci bitů a nezabývá se už vyššími vrstvami komunikace. V referenčním modelu ISO/OSI tak představuje pouze fyzickou vrstvu.

Uživatelská data jsou posílána jako časová posloupnost bitů. Je podporován synchronní i asynchronní přenos. Kromě datového okruhu je zde ještě kontrolní okruh používaný k řízení komunikace mezi DTE a DCE. Oba dva okruhy pracují v jednom směru, signalizace od DTE do DCE nebo opačně. Jelikož vysílaná data a přijímaná pracují na oddělených okruzích, interface může pracovat ve full duplex módu. Standard nedefinuje charakter rámcování během přenosu nebo charakter kódování. Pořadí přenosu datových bitů je od nejméně významného bitu (LSB) po bit nejvýznamnější (MSB). Počet datových bitů je volitelný, obvykle se používá 8 bitů, lze se ale také setkat se 7 nebo 9 bity.

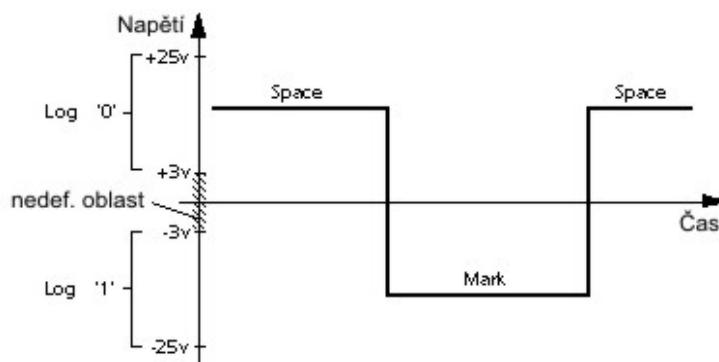
### 2.2 Napěťové vlastnosti

Logický stav „0“/„1“ přenášených dat je reprezentován pomocí dvou možných úrovní napětí, které jsou bipolární a dle zařízení mohou nabývat hodnot  $\pm 5$  V,  $\pm 10$  V,  $\pm 12$  V nebo  $\pm 15$  V. Nejčastěji se používá varianta, kdy logické hodnotě 1 odpovídá napětí  $-12$  V a logické hodnotě 0 pak  $+12$  V. Log. 1 je někdy označována jako *marking state* nebo také klidový stav, log. 0 se přezdívá *space state*. Log. 1 je indikována zápornou úrovní, zatímco logická 0 je přenášena kladnou úrovní výstupních vodičů. Povolené napěťové úrovně jsou uvedeny v tabulce 3.1.

Základní tři vodiče rozhraní (příjem RxD, vysílání TxD a společná zem GND) jsou doplněny ještě dalšími sloužícími k řízení přenosu (vstupy DCD, DSR, CTS, RI, výstupy DTR, RTS). Ty mohou a nemusí být používány (zapojeny), nebo mohou být použity pro napájení elektronických obvodů v zařízení, jako je například počítačová myš. Výstupní elektronika je vybavena ochranou proti zkratu, kdy po překročení proudu 20 mA proud již dále neroste.

Tab. 2.1: Napěťové úrovně RS232

Datové signály		
Úroveň	Vysílač	Přijímač
L	+5 V až +15V	+3 V až +25V
H	-5 V až -15V	-3 V až -25V
nedef.	-3 V do +3 V	



Obr. 2.1: Napěťové úrovně RS232 [10]

Jak můžeme vidět, tak napěťové úrovně jsou vyšší než úrovně používané u integrovaných obvodů (TTL, CMOS). Proto pokud chceme komunikovat s těmito obvody přes RS232, tak je potřeba speciální obvod ke konverzi úrovní. Protože oba konce RS232 okruhu potřebují, aby na zemním pinu bylo 0 V, tak problém může nastat v případě, pokud propojujeme dvě zařízení, kde napětí na GND vodiči na jenom konci a konci druhém není 0 V. Což může nastat v případě, pokud jsou zařízení napájena např. z jiné zásuvky, a tak napěťový potenciál na GND pinu obou zařízení nemusí být vzhledem ke společnému bodu 0 V. Zde by mohla vzniknout zemní smyčka a zničit zařízení.

## 2.3 Konektory

RS232 zařízení mohou být klasifikovány jako DTE a DCE, což říká, že každé zařízení v okruhu posílá a přijímá každý signál. Standard doporučuje, ale nenařizuje D-subminiaturní 25pinový konektor. Standard specifikuje 20 různých signálů. Ale většina zařízení používá pouze část signálů, a tak často jsou používány menší konektory. Například 9pinový DE-9 konektor.

## 2.4 Signály

Transmitted Data (TxD)

- Data posílána z DTE do DCE

Received Data (RxD)

- Data posílány z DCE do DTE.

Request To Send (RTS)

- Potvrzení (nastavení do 0) od DTE k připravení DCE na příjem dat.

Ready To Receive (RTR)

- Potvrzení od DTE, že je schopno přijímat data

Clear To Send (CTS)

- Potvrzení od DCE o příjmu RTS a povolení DTE k vysílání dat

Data Terminal Ready (DTR)

- Potvrzení od DTE k indikaci, že je připraveno být připojeno

Data Set Ready (DSR)

- Potvrzení od DCE k indikaci, že je zapnuto a připraveno přijímat příkazy nebo data od DTE

Data Carrier Detect (DCD)

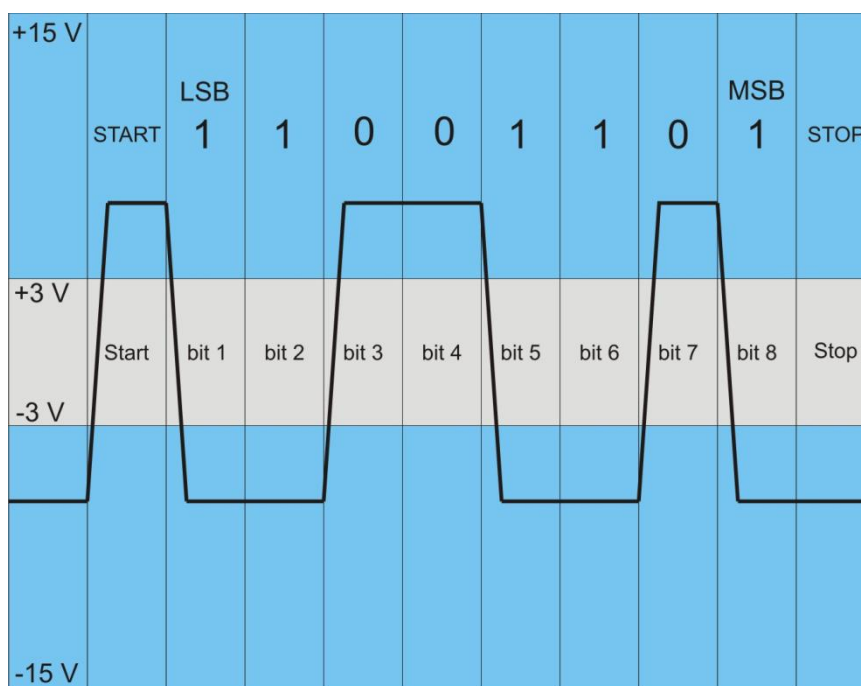
- Potvrzení od DCE, že bylo vytvořeno připojení

Tab. 2.2: Signály u DCE a DTE

Signál název	Zdroj	DTE		DCE	
Common Ground	G				
Transmitted Data	TxD	●			
Received Data	RxD			●	
Data Terminal Ready	DTR	●			
Data Set Ready	DSR			●	
Request To Send	RTS	●			
Clear To Send	CTS			●	
Carrier Detect	DCD			●	

## 2.5 Asynchronní komunikace

I když zařízení komunikující mezi sebou znají přenosovou rychlost, jakou se přenášejí data, přijímač musí vědět, kdy má začít přijímat data. Musí před samotným přenosem dat proběhnout synchronizace. V případě synchronní komunikace musí existovat s datovým vodičem existovat i synchronizační vodič, přes který vysílač oznámí přijímači, že vysílá data. Naopak u asynchronní komunikace neexistuje synchronizační vodič, protože v tomto případě synchronizace probíhá vysláním předem definovanou posloupnost bitů po datovém vodiči, čímž se přijímač zasynchronizuje. V případě RS232 každému vyslanému bajtu předchází jeden START bit, kterým logická hodnota na lince přejde do opačného stavu. V tomto případě na úroveň log. 0, což je u RS232 napětí v rozmezí +3 až +15 V. Po datových bitech následuje paritní bit a poté jeden nebo i více STOP bitů, kterým linka přejde do klidového stavu-indikováno úrovní log. 1, což je napětí v rozmezí -3 až -15 V. Toto nám dovoluje použít méně vodičů, ovšem za cenu menší přenosové rychlosti, kterou právě způsobí počítačová synchronizace.



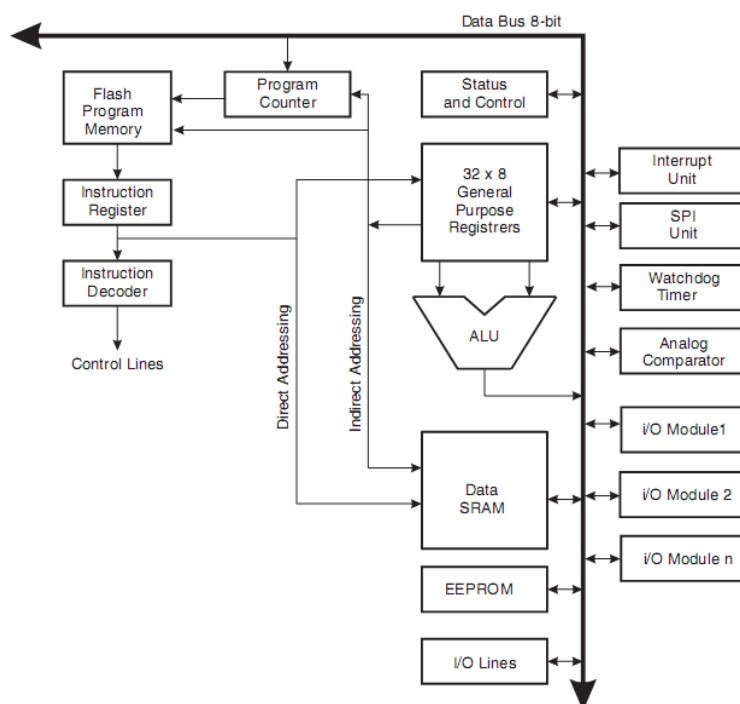
Obr. 2.2: Časový diagram RS232 (přenos jednoho znaku s jedním stop bitem a bez parity)



### 3 Atmel AVR

#### 3.1 Úvod

AVR je v podstatě modifikovanou architekturou 8bitových RISC jednočipových mikrokontrolérů, které byly vyvinuty firmou Atmel v roce 1996. AVR byl prvním z rodiny mikrokontrolérů, které používaly flash paměť k uložení obslužného programu, jako protiklad k jednoprogramovatelným pamětím ROM, EPROM či EEPROM používané u jiných mikrokontrolérů.



Obr. 3.1: Blokové schéma AVR architektury [1]

#### 3.2. Architektura AVR mikrokontrolérů

K maximalizaci výkonu a souběžnosti, AVR používá Harvardskou architekturu – paměti programu a dat jsou oddělené. Paměť programu je přístupná pomocí dvoustupňové pipeline. Když se určitá instrukce začíná vykonávat, další instrukce je připravována k zpracování. Tato konstrukce dovozuje zpracovávání instrukcí vždy v jednom hodinovém cyklu. Paměť programu je tzv. In-System Reprogrammable Flash memory (reprogramovatelná flash paměť).

AVR jádro se skládá ze 32 stejných 8bitových registrů, které mohou obsahovat jak data tak adresy a do nichž se přistupuje během jednoho cyklu. Toto umožňuje "jedno-cyklová" Aritmeticko Logická Jednotka (ALU). Zpracování operandů probíhá následovně: během jednoho taktovacího cyklu se přivedou oba zdrojové operandy z pracovních registrů, uskuteční se potřebná operace a výsledek se uloží zpátky do registrů. Při úspěšném provedení je tedy potřebný čas jeden takt. Jedná se výhradně o operace typu register - register, na kterých je AVR architektura založena. Posledních 6 registrů můžeme ve dvojici použít jako ukazatele adresy pro nepřímé adresování paměti dat. Tyto registry označované písmeny X, Y a Z dovozuji libovolné ukládací operace (Load/Store).

Programátor má například na výběr, zda ukazatel adresy bude po zpracování určité instrukce inkrementovat nebo před zpracováním této instrukce dekrementovat. Užitečné je pro adresování využít možnosti 6bitového posunu v ukazateli adresy v dvojítech registrech Y a Z.

ALU podporuje aritmetické a logické operace mezi registry nebo mezi konstantami a registry. Operace konaná v rámci jednoho registru může být také vykonána v ALU. Po vykonání aritmetické operace je aktualizován Status Register jako informace o výsledku operace.

Vykonávání programu ovlivňují podmíněné a nepodmíněné skoky a volání instrukcí. Většinu AVR instrukcí tvoří jedno 16bitové slovo. Každý adresa v paměti programu obsahuje 16 nebo 32bitové instrukce.

Programová flash paměť je rozdělena do dvou sekcí – Boot program sekce a Application program sekce. Obě sekce mají oddělené Lock Bity sloužící k ochraně proti zápisu/čtení. SPM instrukce, která zapisuje do Application flash paměti musí být umístěna v Boot program sekci.

Během přerušení nebo při volání podprogramu je návratová adresa programového čítače uložena v zásobníku. Zásobník je efektivně alokovan v SRAM a následkem toho velikost zásobníku je limitována jenom celkovou velikostí a využitím SRAM. Všechny uživatelské programy musí inicializovat Stack Pointer v „reset“ rutině. Data v SRAM mohou být jednoduše přístupná skrz pět různých adresovacích módů, která podporuje avr architektura.

Mikrokontroléry AVR obsahují celkem tři druhy pamětí, viz Obr. 3.1:

- paměť programu **FLASH**,
- paměť dat (přechodných) **RAM (SRAM)**,
- paměť dat **EEPROM**.

Paměť programu **FLASH**, slouží k uložení programu či konstant, které v programu využíváme. **Tato data lze uložit pouze za použití programátoru.**

Do paměti **RAM** se uchovávají přechodná data. Tato je rozdělena na několik desítek registrů (jeden registr = 1 bajt). Tato data mohou být pracovní nebo i řídicí – většina registrů ovládá periferie nebo samotný chod procesoru. Jiné zase mohou informovat o stavu, v jakém se procesor nachází.

Paměť **EEPROM** slouží pro data, která chceme, aby zůstala zachována i po odpojení napájení. Jako příkladem lze uvést nějaká konfigurační data nebo třeba kód v případě kódového zámku.

## 4 Výběr komponent pro zařízení

### 4.1 Požadavky

Od tohoto přípravku je požadováno především ovládání specifických obvodů pomocí počítače. K tomuto ovládání se využívá textová komunikace mezi PC a mikrokontrolérem.

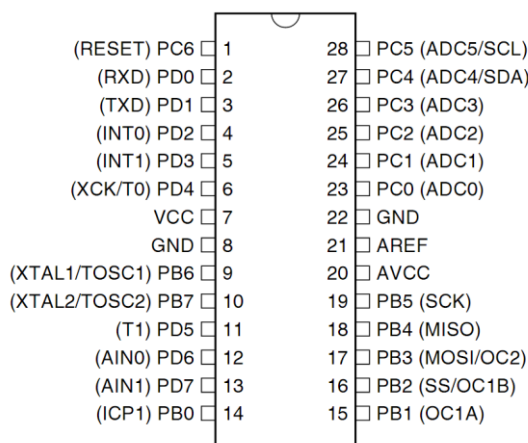
### 4.2 Ovládací mikrokontrolér

Jako hlavní mikrokontrolér jsem vybral od firmy Atmel z řady ATmega. Konkrétně **ATmega 8**, což je 8bitový mikroprocesor, který přímo v sobě obsahuje periferie pro komunikaci přes USART a I<sup>2</sup>C.

Vnitřní periferie mikrokontroléru:

- Dva 8bitové čítače/časovače,
- Jeden 16bitový čítač/časovač,
- Obvod reálného času s odděleným oscilátorem,
- 3 PWM kanály,
- 6kanálový AD převodník,
- Bitově orientovaná dvou vodičová sériová sběrnice (Two-Wire),
- Programovatelné sériové USART rozhraní,
- Master/Slave SPI rozhraní,
- Programovatelný watchdog časovač,
- Analogový komparátor.

Mě z tohoto výčtu periférií zajímá především Two-wire sběrnice a USART rozhraní. Jak si lze všimnout, tak ve výčtu chybí právě ona důležitá I<sup>2</sup>C sběrnice. To je z toho důvodu, že u tohoto obvodu se jmenuje Two-wire interface (TWI), která je samozřejmě plně kompatibilní s I<sup>2</sup>C. Modul TWI je zobrazen na Obr. 4.2.

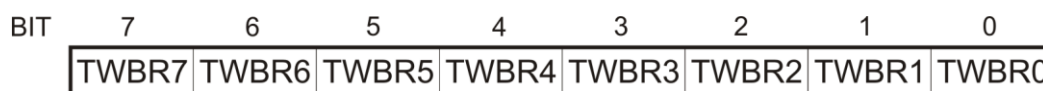


Obr. 4.1: Mikrokontrolér ATmega 8 [1]

Jak lze vidět na Obr. 4.1, mikrokontrolér se sestává ze dvou 8bitových a jednoho 7bitového vstupně/výstupního portu. V tomto případě mezi důležitou součástí patří kontakty TWI, které jsou na portu C na pinech 4(SDA) a 5(SCL).

Obr. 4.2 zobrazuje samotný TWI modul a lze vidět, že obsahuje 5 registrů (TWAR, TWBR, TWDR, TWSR, TWCR), které slouží k nastavení i k samotné komunikaci, a které rozeberu v následujícím textu.

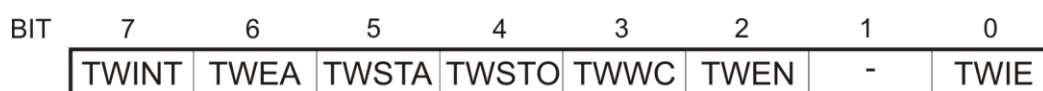
#### 4.2.1 Registr přenosové rychlosti (TWBR)



Obr. 4.2: Bity registru TWBR

TWBR společně s bity předděličky nastavuje přenosovou rychlost, tedy hodinové impulsy generátoru SCL.

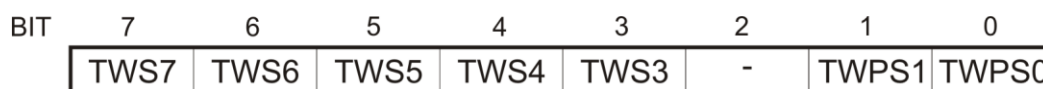
#### 4.2.2 Control registr (TWCR)



Obr. 4.3: Bity registru TWCR

TWCR je používán pro samotnou komunikaci na TWI sběrnici. Je používán k zapnutí TWI, iniciování Master přístupu vykonáním START podmínky, generování potvrzovací impulsu, generování STOP podmínky a ke kontrole zahlcení sběrnice. Registr také indikuje, zda mohou být data zapsána do TDWR registru.

#### 4.2.3 Status registr (TWSR)



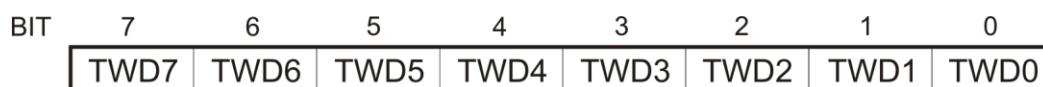
Obr. 4.4: Bity registru TWSR

Registr obsahuje 5 bitů (TWS7 až TWS3), které vyjadřují stav TWI logiky a TWI sběrnice. Bity TWPS1 a TWPS0 určují hodnotu předděličky podle Tab. 4.1.

Tab. 3.1: Nastavení předděličky

TWPS1	TWPS0	Hodnota předděličky
0	0	1
0	1	4
1	0	16
1	1	64

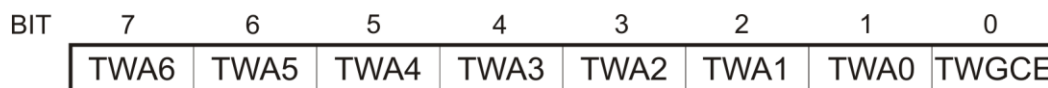
#### 4.2.4 Datový registr (TWDR)



Obr. 4.5: Bity registru TWDR

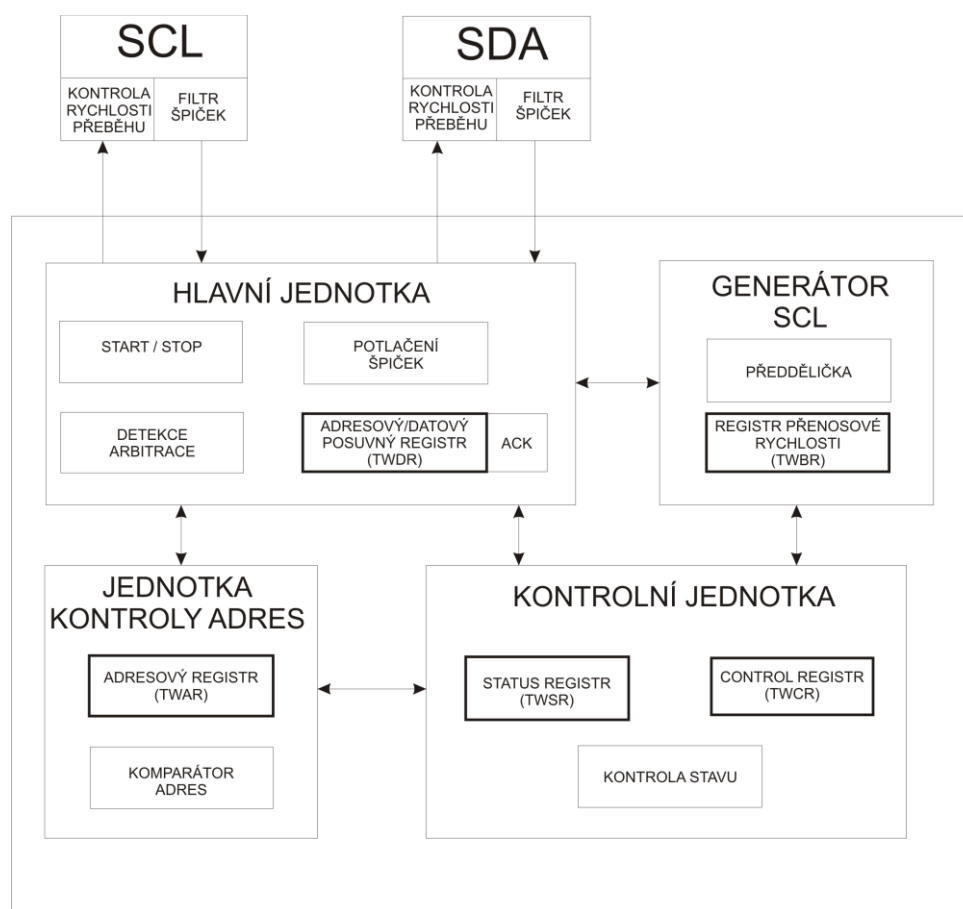
V módu vysílače registr TWDR obsahuje další bajt, který bude odeslán. V módu přijímače obsahuje poslední přijatý bajt. Registr je přístupný k zápisu, pokud TWI neprovádí přesouvání bajtu.

#### 4.2.5 Adresový registr (TWAR)



Obr. 4.6: Bity registru TWAR

Registr obsahuje 7bitovou adresu slave obvodu. LSB registru je používán k povolení rozpoznání broadcast volání slave obvodu.



Obr. 4.7: Blokové schéma TWI modulu [1]

### 4.3 Možnosti programování mikrokontroléru

Existují dvě možnosti programování mikrokontrolérů:

- JTAG programátor
- ISP programátor

Princip JTAG programátoru pouze nastíním, ale podrobněji popisovat nebudu, protože jsem ho já nevyužíval. K programování jsem využíval ISP programátor, který nyní blíže popíši.

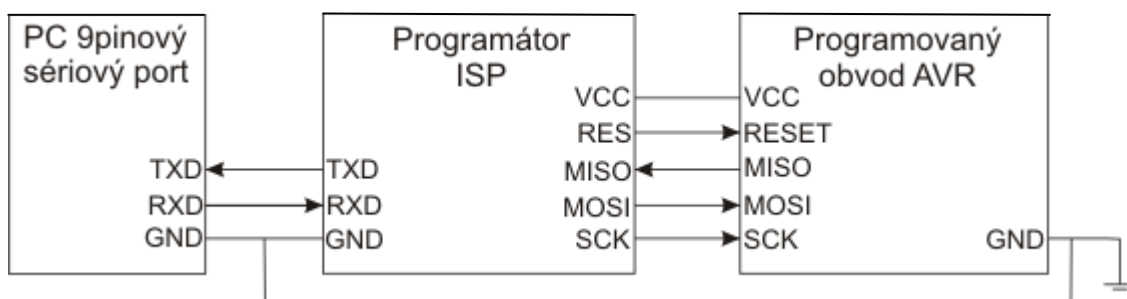
### 4.3.1 JTAG programátor

Skupina výrobců integrovaných obvodů označovaná jako JTAG (Joint Test Actoin Group) chce dosáhnout dohody, která by stanovila princip konstrukce integrovaných obvodů. Cílem je, aby metodika testování integrovaných obvodů byla použitelná i tehdy, pokud se na jedné desce vyskytují komponenty od různých výrobců. Snaží se tedy o jakousi standardizaci. Současná verze je pospána normou IEEE 1149.1 a je výsledkem sjednocení názorů výrobců integrovaných obvodů a jejich zákazníků.

### 4.3.2 ISP programátor

ISP programátor (In-System Programmer) umožňuje programování jakéhokoli AVR mikrokontroléru vsazeného v koncovém systému. Tedy samotný obvod není potřeba vyjmát a vkládat do nějakého programátoru.

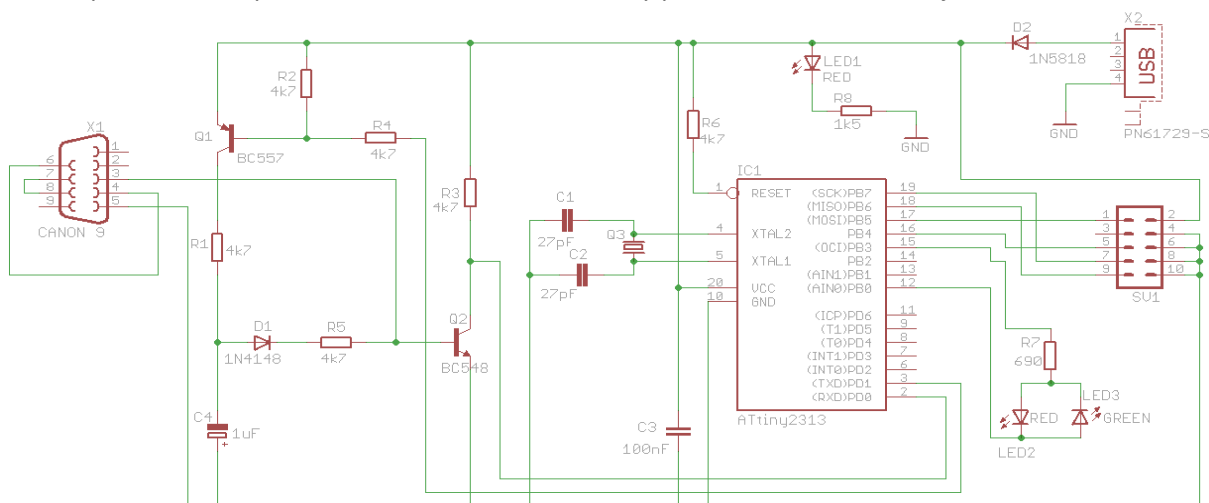
Programátor je připojen k cílovému obvodu pomocí 6 vodičů jak ukazuje Obr. 12.



Obr. 4.8: Blokové schéma zapojení programátoru

Serial Peripheral Interface (SPI) se skládá ze tří vodičů: Seriól Clock (SCK), Master In – Slave Out (MISO) a Master Out – Slave In (MOSI). Při programování mikrokontroléru, programátor pracuje jako Master a programovaný obvod jako Slave.

Programátor poskytuje hodinový signál ke komunikaci na SCK lince. Každý impuls na SCK přeneše jeden bit z programátoru (Master) do obvodu (Slave) přes MOSI linku. Analogicky, každý impuls na SCK lince přeneše jeden bit z obvodu do programátoru přes MISO linku. Využil jsem základní programátor podle Atmel specifikace AVR910 zkonstruovaný podle [5]. Jeho schéma je na Obr. 4.9.



Obr. 4.9: Schéma ISP programátoru

Základem programátoru je mikrokontrolér ATtiny2313. S počítačem se propojuje pomocí sériové linky RS232, a podle zapojení konektoru CANON-9 lze využít, že se počítá s připojením převodníku USB-RS232. Samozřejmě ale není to nutností. USB konektor zde slouží pouze k napájení, jinou funkci nemá.

Celý tento programátor je velice levný a k amatérské tvorbě dostačuje. Tento programátor jsem nijak nenavrhol, ale pouze zkonstruoval. Materiály na něj mě dodal pan Pavelka.

## 4.4 I<sup>2</sup>C expander

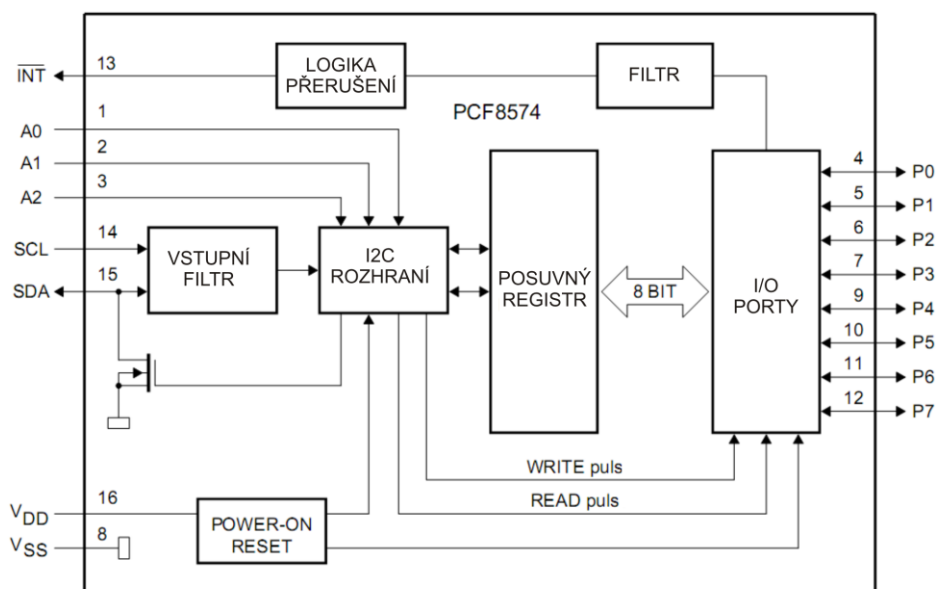
Výrobce obvodu je firma Philips a má označení PCF8574(A). Je vyroben technologií CMOS. Tento obvod obsahuje 8bitový obousměrný port a rozhraní pro I<sup>2</sup>C sběrnici. Obvod má nízký odběr a obsahuje výstupy schopné napájet LED diodu. Také je schopen ovládat linku přerušeni (INT), která může být napojena na obvody přerušeni v mikrokontroléru. Posláním přerušeni po této lince může informovat mikrokontrolér, když jsou na jeho portu nějaká příchozí data, aniž by měl navázanou komunikaci po I<sup>2</sup>C sběrnici. Tím je myšleno, že tento obvod může setrávat s režimu slave. Existují dva druhy tohoto obvodu. PCF8574 a PCF8574A, které se liší pouze prefixem v adrese.

Ale především hlavním úkolem tohoto obvodu je převádět data poslaná po I<sup>2</sup>C sběrnici na paralelní tvar a opačně. Lze ho tedy nazvat jakýmsi (de)multiplexorem.

Mimo jiné obsahuje také vstupy pro nastavení fyzické adresy. První nibl je pevně nastaven od výrobce a uživatel si volí pouze bity A0, A1, A2, viz Obr. 4.11. Jelikož jsou nastavitelné pouze 3 bity, plyne z toho, že maximální počet těchto obvodů na jedné sběrnici je 2<sup>3</sup> tedy 8. Samozřejmě lze použít kombinaci s a bez přípony A, čímž se zvýší počet na 16 expanderů připadající jedné I<sup>2</sup>C sběrnici.



Obr. 4.10: Adresy I<sup>2</sup>C expanderů

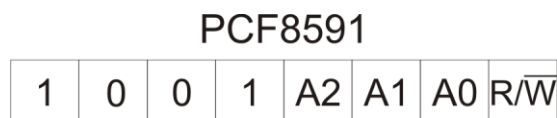


Obr. 4.11: Blokové schéma PCF8574 [8]

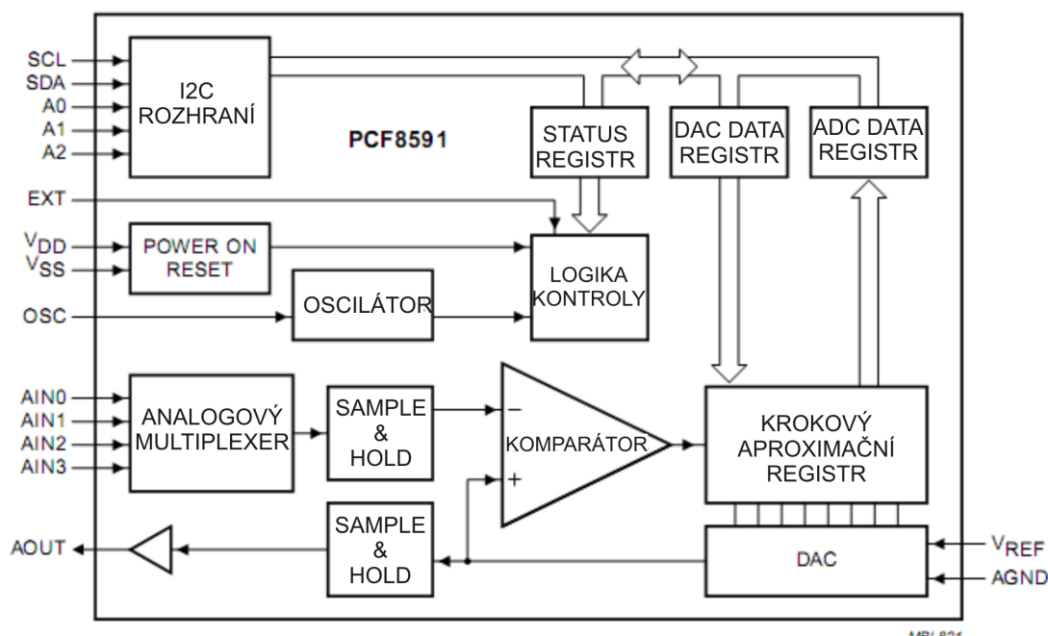
## 4.5 AD/DA převodník

Výrobce převodníku je opět firma Philips a má označení PCF8591. Vyroben je technologií CMOS. Obvod obsahuje 4 analogové vstupy, 1 analogový výstup a samozřejmě I<sup>2</sup>C rozhraní. Mezi funkce obvodu patří multiplexování analogového vstupu, sample&hold funkce, 8bitový analog/digital a digital/analog převod. Maximální rychlost převodu je dána maximální rychlostí I<sup>2</sup>C sběrnice. Převodník je 8bitový a při referenčním napětí 5 V lze tedy krokovat výstupní napětí po asi 20 mV.

Převodník má integrován oscilátor nutný k analog-digital konverzi. Jak ukazuje Obr. 4.13, tak lze k převodníku připojit externí oscilátor, pokud by frekvence integrovaného nedostačovala.



Obr. 4.12: Adresa I<sup>2</sup>C AD/DA převodníku

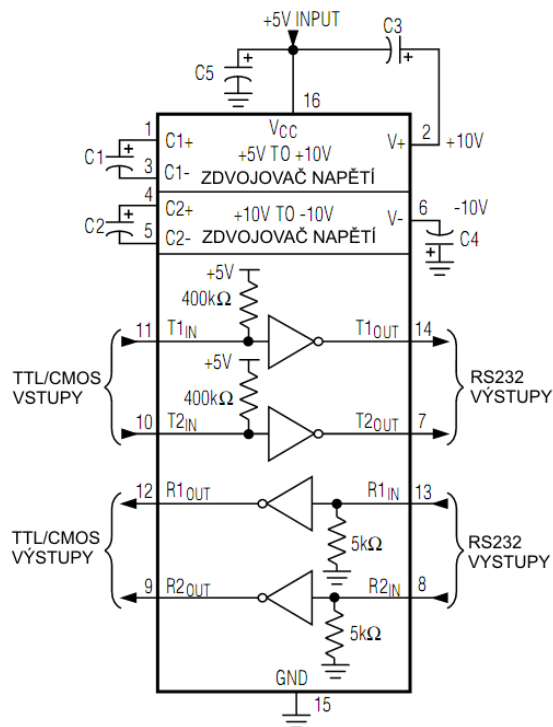


Obr. 4.13: Blokové schéma PCF8591 [9]

## 4.6 Převodník TTL↔RS232

Vlastnosti rozhraní RS232 jsem blíže popsal v kapitole 2. Z této kapitoly již vyplývá, že pokud chceme propojit obvody TTL logiky a rozhraní RS232, je nutno použít nějaký převodník úrovní. Mezi nejznámější patří obvod firmy Maxim, MAX232. Ten obsahuje dva převodníky z RS232 (+12V a -12V) na TTL (5 V a 0 V) a dva převodníky z TTL na RS232. Jelikož je tento obvod napájen 5V, tak ke konverzi na ±12 V využívá nábojové pumpy a externě připojeným kondenzátorů. A jelikož kvalita napětí značně tedy závisí na kvalitě těchto kondenzátorů, je třeba si dát pozor, nejlépe se úplně vyhnout elektrolytickým kondenzátorům, u kterých se jejich parametry s časem zhoršují. Proto nejlepší kondenzátory použitelné jako nábojová pumpa jsou kondenzátory tantalové.





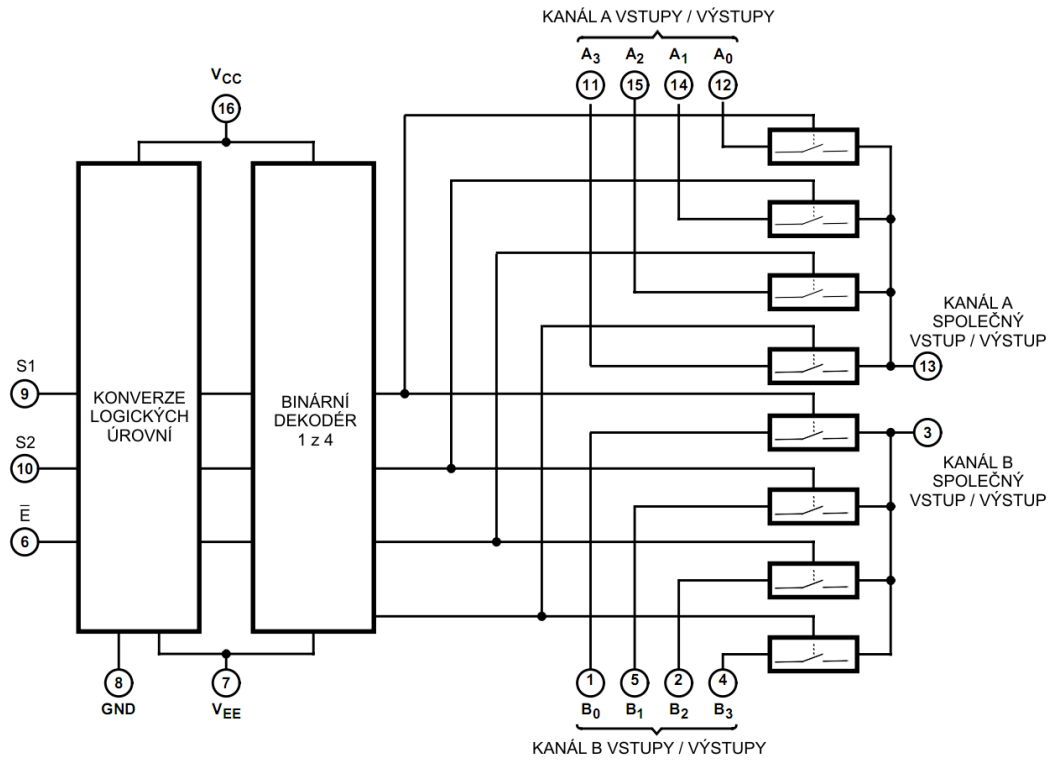
Obr. 4.14: Vnitřní struktura MAX232 [7]

## 4.7 Analogový multiplexor

Analogový multiplexor je zařízení, které v sobě obsahuje digitálně ovládané analogové spínače vyrobené polovodičovou CMOS technologií. Tento analogový multiplexor spíná analogové napětí, které může být až do velikosti napětí napájecího. Tyto spínače jsou obousměrné a dovolují, že jakýkoliv analogový vstup může být použitý i jako výstup či naopak. Spínače mají nízký odpor v sepnutém stavu a nízkou propustnost (velký odpor) ve stavu rozepnutém. Multiplexor má také „povolovací“ vstup E (enable), který v úrovni H rozepne všechny spínače nezávisle na úrovních vstupů S1 a S2, viz Tab. 4.1.

Tab. 4.1: Pravdivostní tabulka

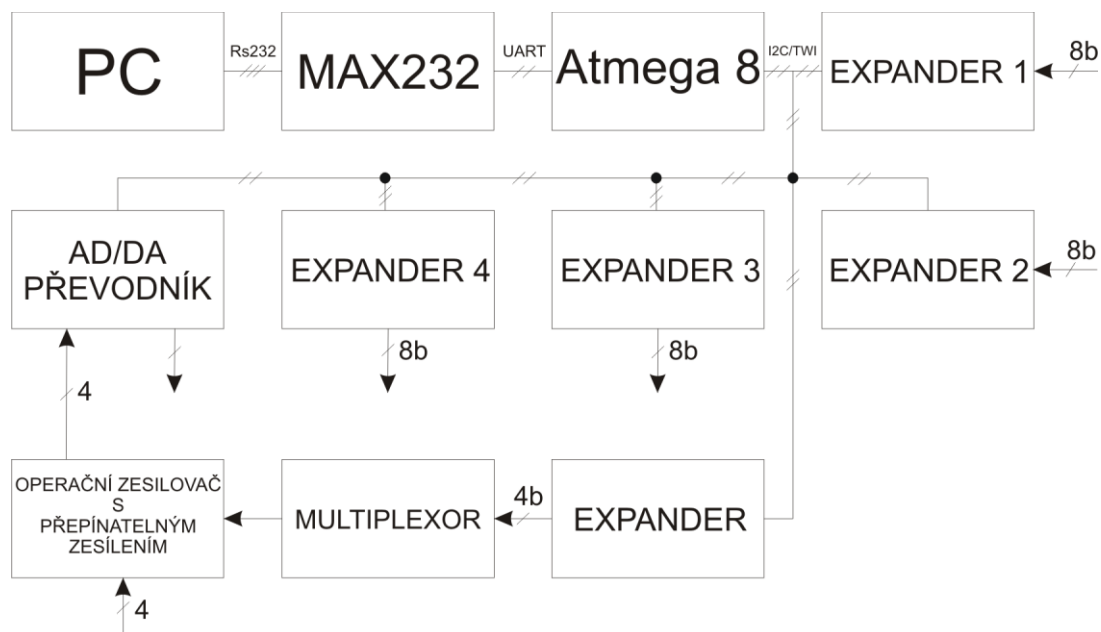
E	VSTUPY		SEPNUTÉ KANÁLY
	S1	S2	
L	L	L	A0, B0
L	L	H	A1, B1
L	H	L	A2, B2
L	H	H	A3, B3
H	X	X	žádný



Obr. 4.15: Blokové schéma multiplexoru

## 5 Návrh zařízení

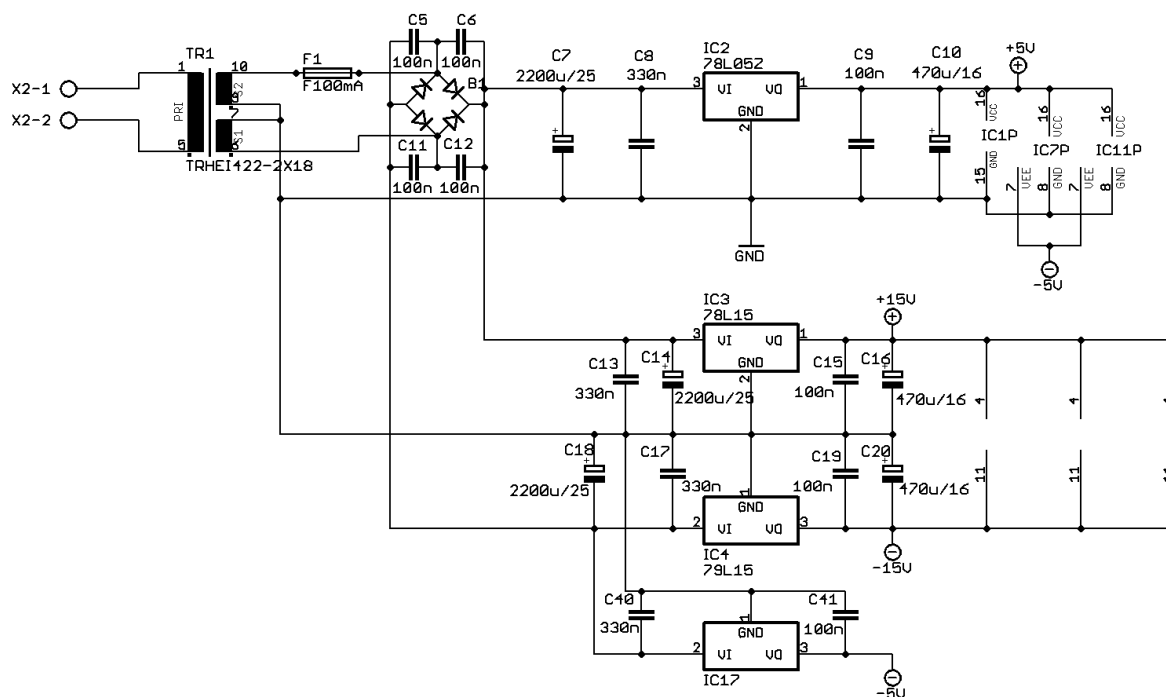
V této kapitole se budu věnovat postupnému vývoji hardware pro ovládání obvodů na TWI sběrnici. Rozeberu všechny hlavní části zařízení a popíši jejich funkci.



Obr. 5.1 Blokové schéma zapojení převodníku

### 5.1 Napájecí zdroj

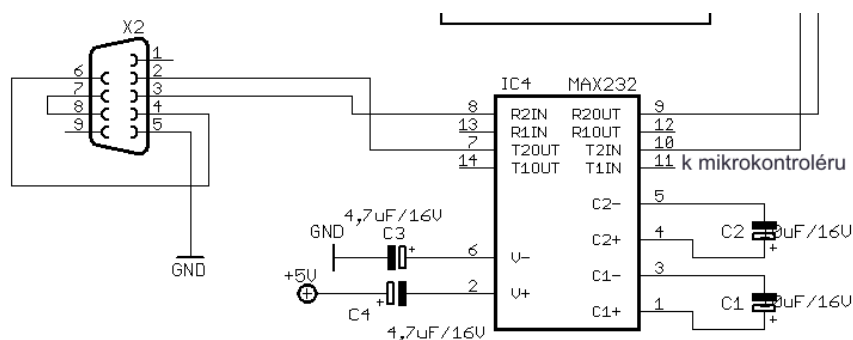
Napájecí zdroj obsahuje 4 napájecí větve. První větev je +5V, sloužící k napájení integrovaných obvodů vyjma operačních zesilovačů. Druhá větev -5V společně s +5V slouží k vytvoření napěťového rozsahu, který je možný připojit na vstupy multiplexoru. Třetí větev +15V potažmo čtvrtá větev -15V, slouží právě k napájení operačních zesilovačů. Jelikož vstupní napětí u některých těchto operačních zesilovačů je až 12V, musel jsem zvolit napájení právě  $\pm 15V$ . Zdroj je zapojen dle schématu: transformátor-usměrňovač-stabilizátor-filtr, jak lze vidět na Obr. 5.2. Transformátor má sekundární vinutí 2x15V/6VA k vytvoření záporného napětí pro operační zesilovače. Hodnota pojistky 100 mA byla zvolena z důvodu zapojení lineárních stabilizátorů, kterými může protékat maximální proud právě 100mA. Elektrolytické kondenzátory slouží k vyfiltrování zvlnění a jejich velikost byla vypočítána tak, aby maximální zvlnění napětí bylo 1%.



Obr. 5.2: Schéma zdroje

## 5.2 Komunikace s mikrokontrolérem (HW úroveň)

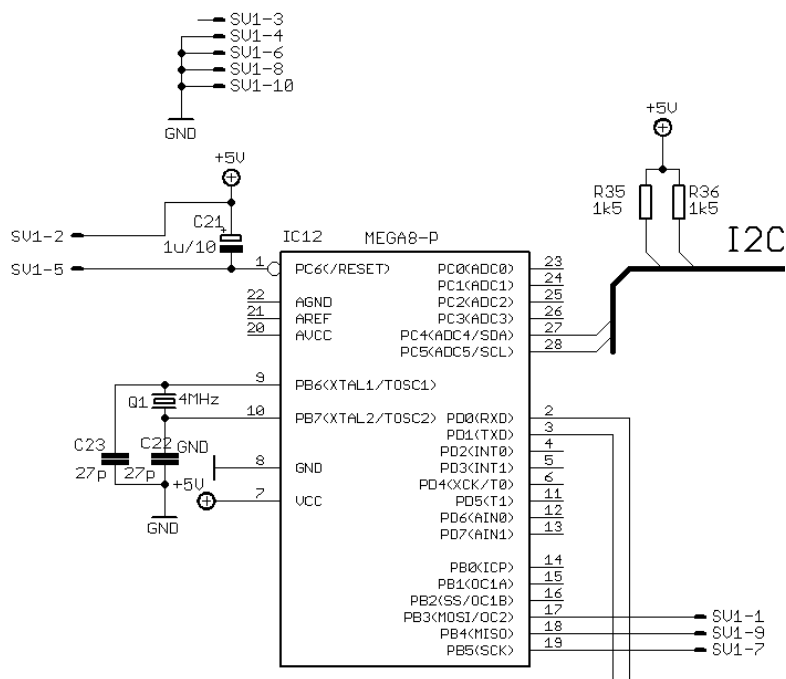
Jak již bylo několikrát zmíněno, komunikace s mikrokontrolérem probíhá pomocí sériové linky RS232. Jako převodník RS232-TTL jsem využil obvodu firmy Maxim MAX232. V zapojení jsem použil základního zapojení tohoto obvodu, které je na Obr. 5.3. Kondenzátory C1 až C4 jsou ony kondenzátory nábojové pumpy. Ze zapojení konektoru X2 lze vytušit, že jeho kontakty jsou spojeny tak, že se k připojení k zařízení používá převodník USB-RS232. Samozřejmě ale není to nutností, ale jelikož se dnes už ustoupilo od komunikace přes RS232 a toto rozhraní se už nenachází na drtivé většině dnešních počítačů, je zapojen tento konektor právě takto.



Obr. 5.3: Schéma zapojení převodníku MAX232

### 5.3 Zapojení mikrokontroléru ATmega8

Jak lze vidět na Obr. 26, tak na mikrokontroléru je využita pouhá čtvrtina kontaktů. Je to dáno především sériovou komunikací, která je zde jako jediná využita. Na vstupy XTAL1 a XTAL2 je připojen externí krystalový oscilátor s frekvencí 4,000MHz. Použití tohoto oscilátoru není nutností, protože



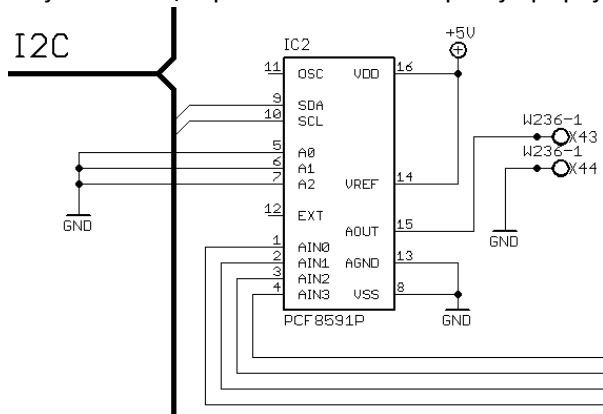
Obr. 5.4: Schéma zapojení mikrokontroléru

mikrokontrolér obsahuje i interní. Záleží pouze na naprogramování tzv. fuse bitů. Ovšem při programování mikrokontroléru je třeba využít externího oscilátoru, a jelikož se na desce nachází konektor (SV1) pro připojení programátoru, tak jsem se rozhodl právě pro připojení externího krystalu, který je vyblokován kondenzátory 27 pF proti zemi.

Kontakty PC4 (SDA) a PC5 (SCL) jsou kontakty TWI sběrnice a PD0 a PD1 jsou kontakty rozhraní USART. Kondenzátor C21 zde slouží jako jednoduchý resetovací obvod, který po připojení napájení vytvoří krátký záporný impuls na vstupu RESET, který restartuje mikrokontrolér. Rezistory R35 a R36 jsou pull-up rezistory pro TWI sběrnicí, jejichž úloha je popsána v kapitole 1.4. Podle [8] byla velikost pull-up rezistorů vypočítána na 1,5 kΩ.

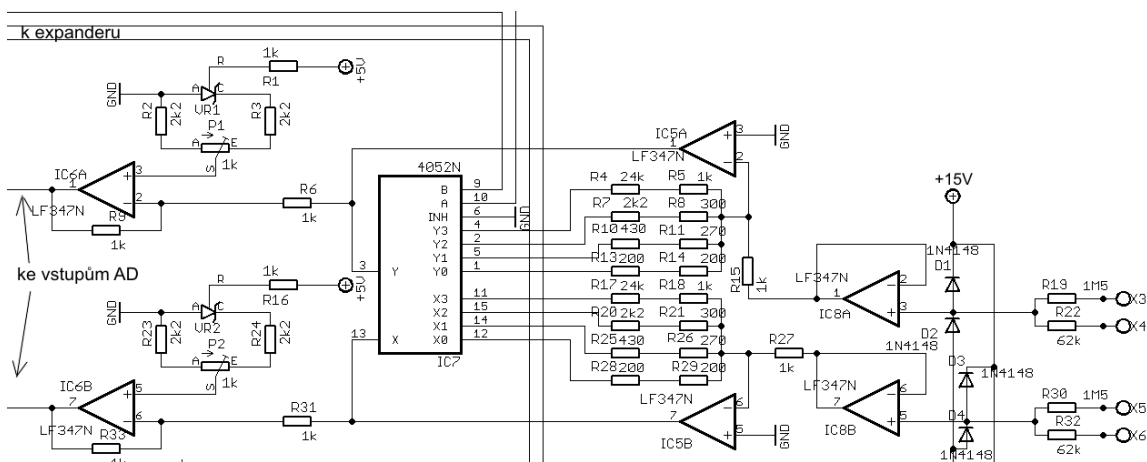
## 5.4 Zapojení AD/DA převodníku

Kontakty A0, A1, A2, sloužící k nastavení adresy, jsou všechny uzemněny, tudíž adresa tohoto obvodu je 1001 000x, kde x je bit čtení/zápisu. Referenční napětí je připojeno na vstup VREF. Podle



Obr. 5.5: Schéma zapojení AD/DA převodníku

velikosti tohoto napětí se přepočítává napětí na AD a DA převodníku. Na jeho velikosti a správném nastavení tedy velice záleží z hlediska přesného nastavení/odečtu napětí. Výstup DA převodníku je na kontaktu AOUT. Rozsah napětí je dán referenčním napětím, v tomto případě tedy 0 až 5 V. Ze zapojení vyplývá, že by AD převodník nebyl schopen změřit napětí vyšší než 5 V. Před vstupy AIN0 – AIN3 je tedy třeba zapojit obvod, který bude přepínat vstupní rozsahy napětí. K tomuto účelu je zde využito zapojení operačního zesilovače s přepínatelným zesílením. Toto zesílení je přepínáno změnou hodnoty odporu rezistoru ve zpětné vazbě. Přepínání probíhá za pomoci analogového multiplexoru 4052, řízeného I<sup>2</sup>C expandérem, viz Obr. 5.6.



Obr. 5.4: Schéma zapojení obvodu pro změnu rozsahu měření napětí

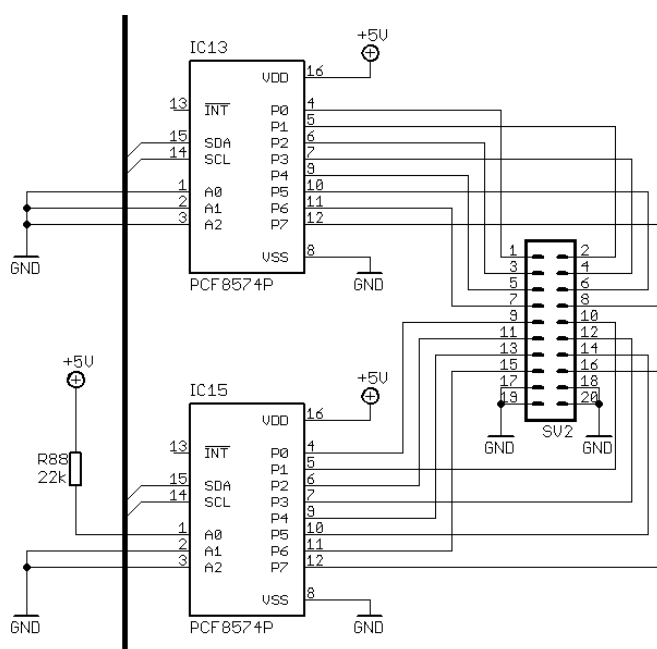
Na vstupu je zapojen dělič napětí s dělicím poměrem 1:25. Diody D1, D2, atd. tvoří obvod ochrany před velkým vstupním napětím. Operační zesilovač IC8 slouží pouze k impedančnímu přizpůsobení. Jeho vstupní odpor je v řádu  $10^{12} \Omega$  a vstupní dělič není tedy zatížen. Je zapojen jako napěťový sledovač a jeho přenos je tedy 1. Dále následuje invertující zesilovač s přepínatelným zesílením. Po TWI se pošle do expanderu bitová posloupnost, která odpovídá určitému propojení vstupu/výstupu X a Y se vstupy/výstupy X0, ..., X3 a Y0, ..., Y3, viz [12]. Dojde tak ke spojení rezistorů R4, R5 atd. s výstupem operačního zesilovače, čímž se změní zesílení tohoto zesilovače. Hodnota rezistorů ve zpětné vazbě je vypočítána tak, aby na výstupu IC5 bylo maximálního napětí 5 V, které je už připoje-

no na vstup AD převodníku. Na výstupu IC5 je ještě zapojen další invertující operační zesilovač, sloužící k nastavení stejnosměrné složky, a tudíž je možno měřit i záporné napětí. Ovšem v tomto případě by se musely změnit odpory ve zpětné vazbě IC5, aby maximální napětí na výstupech IC5 bylo 2,5 V.

Je ale dále třeba si také uvědomit, že rozsah vstupních napětí multiplexoru je dán napětím mezi kontakty Vee a Vcc multiplexoru. A jelikož operační zesilovač IC5 je zapojen jako invertující, vytvoří tedy záporné napětí na jeho výstupu a tedy i na vstupu multiplexoru je toto záporné napětí. A jestliže napájecí kontakt Vcc je připojen na +5 V, tak Vee musí být na -5 V (viz Obr. 5.2), čímž se dostáváme s rozsahem i do záporných hodnot.

## 5.5 Zapojení I<sup>2</sup>C expanderu

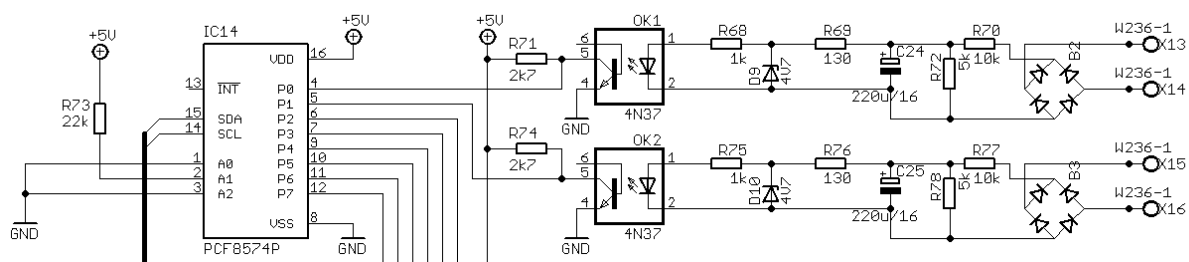
Jelikož je potřeba 16 vstupů a 16 výstupů, je potřeba použít 4 expanderu. 2 jako výstupní a 2 jako vstupní. Opět vstupy A0, A1, A2 slouží k nastavení adresy. U IC13 je zvoleno opět uzemnění těchto vstupů, což se může zdát kolizní s předchozím AD/DA převodníkem, ale jelikož je od výrobce nastaven jiný prefix adresy, tak jsem si mohl opět dovolit nastavit adresní vstupy na úroveň L. Adresa obvodu IC13 je tedy 0100 000x (ovšem pouze u PCF8574, při použití PCF8574A je adresa 0111 000x) a IC15 je 0100 001x (při použití PCF8574 je adresa 0111 001x). Jako konektor pro výstupní expander je využit 20pinový konektor pro plochý kabel, u kterého jsou 4 piny využity pro signálovou zem.



Obr. 5.7: Schéma zapojení výstupního expanderu

Na Obr. 5.8 lze vidět zapojení vstupního expanderu. Tento expander slouží ke zjištění přítomnosti napětí přivedeného mezi svorky X13 a X14, atd. Jeho vstupy jsou galvanicky odděleny optočleny z důvodu ochrany. Na vstupní svorky X13, X14, atd. lze připojit střídavé či stejnosměrné napětí v rozsahu 0 až 200 V. Toto napětí přivedeme na usměrňovací můstek a poté na napěťový dělič 1:3. Tento dělicí poměr je vypočítán tak, aby se úroveň L indikovala na vstupu expanderu od napětí asi 12V. Protože se jedná o zatížený dělič, tak při 12 V nebude na odporu R72 1/3 tohoto napětí, ale přibližně 1/7 vstupního napětí. Úroveň L se na expanderu bude indikovat v rozsahu vstupních napětí 12 až 200 V, zatímco úroveň H v rozmezí napětí 0 až 11 V. Účel, pro který je toto zařízení určeno je takový, že rozmezí napětí 0 až 12 V znamená „bez napětí“ a napětí vyšší než 12 V znamená „napětí je přítomno“. Tedy ze zapojení lze vytušit, že naopak při napětí menším než 12 V by se na vstupu expande-

ru indikovala úroveň H, protože tranzistor v optočlenu OK1 by nebyl sepnutý a naopak od napětí 12 V by se indikovala úroveň L. Je tedy potřeba v programu mikrokontroléru invertovat tyto úrovně, aby H znamenalo „napětí je přítomno“ a L „bez napětí“.



Obr. 5.8: Schéma zapojení vstupního expanderu

## 5.6 Celkové schéma zapojení

Jelikož se jedná o schéma velkého rozsahu, je uvedeno v příloze A.

## 6 Návrh desky plošných spojů

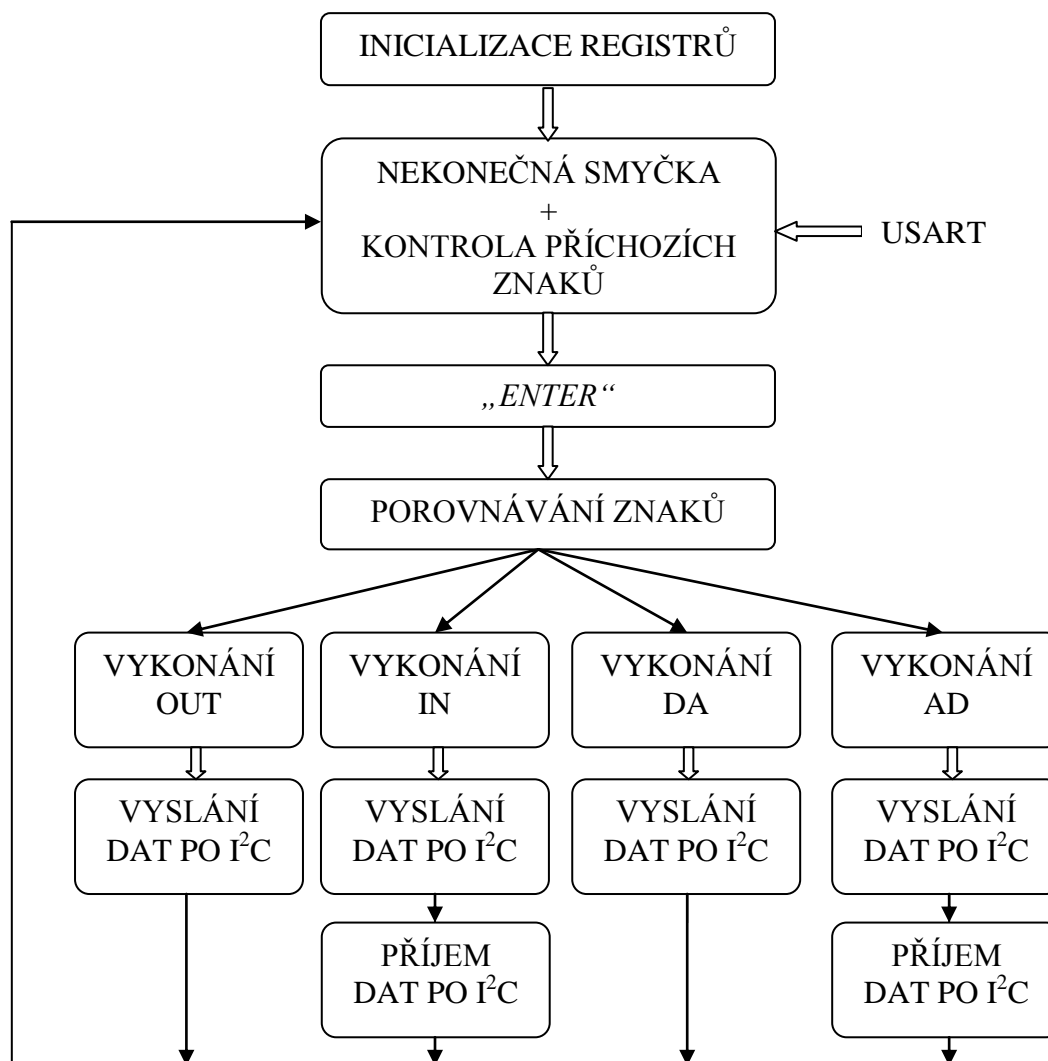
Převodník je umístěn na jednostranné desce o rozměrech 280 x 147 mm. Při použití součástek pro povrchovou montáž by se sice celkové rozměry zmenšily přibližně na polovinu, ale jelikož se nejedná o zařízení pro sériovou tovární výrobu, tudíž při ručním pájení je mnohem komfortnější použít běžné „vývodové“ součástky. Návrh schéma i desky probíhal v programu CadSoft Eagle 5.4.0. a samotná deska plošných spojů a osazení je uvedeno v příloze B.



## 7 Návrh programu pro mikrokontrolér

Programoval jsem ve vývojovém prostředí C v aplikaci CodeVisionAVR. Tato aplikace přímo v sobě obsahuje podporu na mnou používaný programátor AVR910.

### 7.1 Blokové schéma programu



Obr. 7.1: Blokové schéma programu mikrokontroléru

## 7.2 Postup programování + ukázka kódu

Jelikož se jedná o rozsáhlý kód, uvedu pouze důležité části.

Nejdříve je potřeba inicializovat samotné registry v mikrokontroléru.

### Inicializace USART

```
UCSRA=0x00;
UCSRB=0x18; // povolení přijímače a vysílače
UCSRC=0x86; // nastavení počtu datových bitů na 8
UBRRH=0x00; // společně s UBRRL nastavuje modulační rychlost na 9600,
            // viz tabulka 61 v [1]
UBRRL=0x19;
```

### Inicializace TWI

```
TWSR=0x00; // pouze počáteční inicializace, později se s ním pracuje
TWBR=0x18; // nastavení přenosové rychlosti, viz rovnice 7.1
TWAR=0x00; // pouze počáteční inicializace, později se s ním pracuje
TWCR=0x44; // povolení generování potvrzovacího impulsu ACK
```

Jednotka generátoru SCL nastavuje frekvenci hodinových impulsů podle

$$SCL = \frac{CPU}{16 + 2(TWBR) \cdot 4^{TWPS}}, \quad (7.1)$$

kde:

- CPU je frekvence mikrokontroléru,
- TWBR je hodnota registru přenosové rychlosti,
- TWPS je hodnota bitů předděličky ve status registru.

Tedy při použití 4MHz oscilátoru, nastavení registru TWBR na 24 a hodnoty bitů předděličky na 0, lze podle rovnice 7.1 vypočítat, že jsem zvolil frekvenci  $SCL = 62\,500$  kHz. Tato hodnota byla zvolena jako kompromis mezi rychlostí přenosu dat a [8], [9], kde lze vyčíst, že maximální možná hodnota SCL může být 100 kHz.

Po této počáteční inicializaci následuje hlavní smyčka programu:

```
while (1)
{
k[j]=getchar(); //fce na snimaní znaku z USART
putchar( k[j]); //fce pro vyslání znaku pres USART

if (k[j] == 13) //pokud enter
{
printf("\n\r"); //odradkuj
if ((k[0] == 111 || k[0] == 79) && (k[1] == 117 || k[1] == 85) && (k[2] ==
116 || k[2] == 84)) // pokud out/OUT
{
output1 = 0; //inicializace promenne pro vystupni expander
not_number=0; //inicializace promenne pro kontrolu prichozich znaku
for ( i=4; i<19; i++) //smycka pro kontrolu prichozich znaku
if ( k[i]<=47 || k[i]>=50) not_number++; //overeni zda parametr obsahuje
// pouze „1“ nebo „0“
if (not_number > 0) ERR(); //kdyz neobsahuje, vypis chybu
else
{
```

```

output1 = ((k[11]&1)<<7) + ((k[10]&1)<<6) + ((k[9]&1)<<5) + ((k[8]&1)<<4) +
((k[7]&1)<<3) + ((k[6]&1)<<2) + ((k[5]&1)<<1) + (k[4]&1); //prevod z ascii
do binarniho tvaru
OUT_fce(output1, output2); //fce pro vystupni expander
};
}

```

Jak lze vidět v předcházejícím výpisu, proměnná `outputx` slouží k vyslání znaků po TWI sběrnici. Ale jelikož v poli `k[]` jsou uloženy znaky v ascii tvaru, je třeba tyto znaky převést do odpovídající binární podoby. Tedy například znak `1` (v ascii 49) uložený v proměnné `k[11]` je třeba bitově vynásobit s číslem `1`, abychom dostali místo `0011 0001` (49) skutečně `0000 0001` (1). A naopak v případě funkce `IN` je třeba z binární podoby znaku `1` převést do ascii tvaru za pomoci bitového součtu s číslem `48`. Tedy v případě potřeby vyslání znaku `1` (`0000 0001`) se přičte `48` (`0011 0000`), což dává číslo `49` (`0000 0001`) v ascii tvaru odpovídající právě číslu `1`.

V případě zadání špatného znaku se lze opravit pomocí *backspace*:

```

void backspace( void)
{
putchar(32); //vysle znak mezery
putchar(8); //vysle znak backspace
j--; //dekrementuje ukazatele v poli
}

```

Deklarace vlastních funkcí zajišťujících komunikaci po TWI. Jako příklad uvedu pouze dvě:

```

void OUT_fce(unsigned char data1, unsigned char data2)
{
    /***expander***/
    //0111000x = adresa PCF8574A expander1
    //0111001x = adresa PCF8574A expander2

    i2c_startni(0x70); // 01110000 "zapisovaci" adresa expanderu 1
    i2c_zapis(data1); // posli po TWI obsah promenne data1
    i2c_stopni(); // vysli stop bit
    delay_us(10); // cekej 10us
    i2c_startni(0x72); // 01110010 "zapisovaci" adresa expanderu 2
    i2c_zapis(data2); // posli po TWI obsah promenne data2
    i2c_stopni(); // vysli stop bit
    delay_us(10); // cekej 10us
}

```

```

void IN_fce(void)
{
    unsigned char addr;

    if (k[3] == 49) addr = 0x71; // 01110001 "cteci" adresa vstupniho
                                // expanderu 1
    else addr=0x73; // 01110011 "cteci" adresa vstupniho expanderu 2
    i2c_startni(addr); // vysli start bit s adresou
    i2c_readNak(); // prijmi data přes TWI
    i2c_stopni(); // vysli stop bit

    bin2ascii(); // fce pro prevod do ascii tvaru
}

```

## 8 Ovládání převodníku

Příkazy se do převodníku posílají za pomoci sériové linky RS232. Programů, které dokážou komunikovat přes RS232 je samozřejmě celá spousta, ale v tom nejjednodušším případě je záhodno využít aplikaci, který se nachází přímo v operačním systému Windows. Jedná se o Hyperterminal. Je to komunikační terminál, sloužící k připojení přes TCP/IP, k serverům Telnet, ale především k připojení k hostitelským počítačům či jiným zařízením pomocí kabelu nulového modemu.

K fyzickému připojení k převodníku je tedy zapotřebí převodníku USB-RS232 nebo sériového kabelu zapojeného jako tzv. nulový modem – v tomto případě je ovšem potřeba si vyrobit konvertor samec-samec, protože na desce převodníku se nachází samice konektoru Canon 9.

Terminál spustíme, zvolíme přímé připojení na port, na který je převodník připojený a zadáme následující parametry:

- Bity za sekundu: **9600**
- Datové bit: **8**
- Parita: **Žádná**
- Počet stop-bitů: **1**
- Řízení toku: **Žádná**

Tyto parametry nastavení je třeba dodržet nebo nebude komunikace s převodníkem probíhat správně. V případě potřeby jiného nastavení je třeba přeprogramovat program mikrokontroléru.

Poté je spojení s převodníkem navázáno a my můžeme zadávat příkazy ve tvaru:

**Příkaz(mezera)parametr(enter)**

Převodník je schopen vykonávat 4 příkazy, viz Tab. 8.1.

Tab. 8.1: Příkazy

Příkaz	Účel příkazu	Parametr	Příklad příkazu
OUT	slouží k nastavení binární kombinace na výstupech expanderů	16bitové číslo v binárním tvaru	OUT 1111000011110000
IN	slouží ke zjištění stavů na vstupu expanderu	číslo expanderu: 1 nebo 2	IN 1
AD	ke zjištění hodnoty napětí na vstupu AD převodníku	číslo kanálu AD převodníku: 0, 1, 2 nebo 3 a rozsah napětí 1, 2, 3 nebo 4	AD 1
DA	slouží k nastavení požadované hodnoty napětí na výstupu DA převodníku	napětí ve voltech ve tvaru: X,XXX vždy zaokrouhleno na tisíce!!	DA 2,256

Pozn.: zadávané napětí u DA převodníku musí být zadáno na 3 desetinná místa, jinak může dojít k chybě!

Příkaz IN vrací 8bitové binární číslo, odpovídající logickému stavu na vstupu daného expanderu v bitové posloupnosti odpovídající posloupnosti svorek:

- X13+X14, ..., X27+X28, v případě zvolení expanderu 1,
- X29+X30, ..., X43+X44, v případě zvolení expanderu 2.

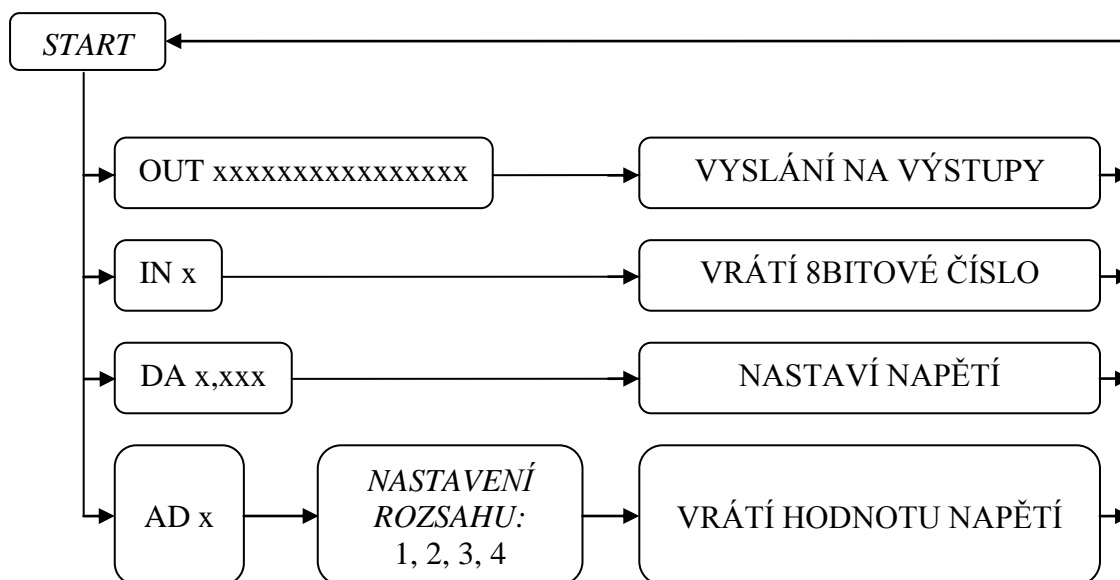
Příkaz AD vrací napětí na zvoleném vstupním kanálu AD převodníku, viz Tab. 8.2

Tab. 8.2: Vstupní svorky AD převodníku

Příkaz+parametr	Vstupní svorky
AD 0	X9, X10
AD 1	X7, X8
AD 2	X5, X6
AD 3	X3, X4

Po zadání příkazu AD x je třeba zadat rozsah měřeného napětí, jak je známo například z multimetrů. Je třeba tedy zadat maximální možné měřené napětí, které se bude připojovat na vstupní svorky. V případě připojení vyššího napětí než bylo zadáno, může dojít k nevratnému poškození AD převodníku. Lze nastavit rozsahy měření 5, 50, 150 a 300 V. Samozřejmě nejlepší je postupovat od rozsahu 300 V níže. Vidíme-li, že napětí je menší než další následující rozsah, tedy 150 V, můžeme nastavit tento rozsah a měřit přesněji.

Ovládání převodníku probíhá podle následujícího schématu:



Obr. 8.1: Schéma ovládání převodníku

Zadávání příkazů je ošetřeno proti chybám a to tak, že se vypíše:

*UNKNOWN COMMAND* - pokud je příkaz nerozpoznán,

*ERROR* - pokud se příkaz sice rozpozná, ale je špatný počet parametrů, popř. špatný parametr.

Každý příkaz je mikrokontrolérem vykonán až po ukončení vstupního řádku klávesou [Enter].

## 9 Závěr

V kódu mikrokontroléru bylo pro ukládání příchozích znaků využito jednoduché ukládání těchto znaků do pole. Jestliže se komunikuje s mikrokontrolérem přes USART rozhraní, pak se spíše využívá přerušení a kruhový buffer. Je to z toho důvodu, pakliže by mikrokontrolér vykonával jinou činnost a během toho by mu byly posílány znaky přes USART, pak by tyto zkrátka nebyli nikam ukládány ani poté vykonány, protože mikrokontrolér může v jednu chvíli vykonávat pouze jednu činnost. Jestliže by měl naprogramováno přerušení při příchodu znaků přes USART, pak by přerušil dosavadní činnost a přijímal tyto znaky. Ovšem v případě tohoto zařízení nebylo toto potřeba, jelikož příkazy jsou vykonány ihned a nepotřebují žádné časové prodlení po jejich příchodu. Samozřejmě vykonání příkazu trvá jistou dobu, ale než se stačí vůbec napsat další příkaz, tak předchozí je již dávno vykonán a mikrokontrolér už poté jenom čeká na příchod dalších znaků. Z toho vyplývá, že zde není implementován žádný kruhový buffer ani rutina obsluhující přerušení.

Převodník byl navrhnout s ohledem na bezpečnost. Jelikož se na vstupní expandery může připojit až napětí blízké síťovému, tak tato část je opticky oddělena od zbytku zařízení. Optočleny byly také připojeny z důvodu zamezení toků vyrovnávacích proudů. Takovéto oddělení by si zasloužila i část s AD převodníkem určená pro měření napětí. Ale jelikož implementace optočlenu do této části by byla velice komplikovaná, tak bylo od tohoto upuštěno.

Výsledkem této práce je funkční zařízení, které se připojuje k počítači pomocí rozhraní RS232 a je schopno měřit a generovat napětí požadované velikosti v jistých mezích, indikovat přítomnost napětí a generovat logické úrovně. Toto zařízení je napájeno síťovým napětím 230 V a je tedy přenosné všude tam, kde je možnost odběru tohoto napětí.

Stavba samotného zařízení obsahující všechny prvky vyjde přibližně na 1000 Kč. V případě použití součástek pro povrchovou montáž by se cena mírně snížila z důvodu menší plochy desky plošného spoje, ale jak již bylo řečeno, nejedná se o zařízení určené pro sériovou výrobu, a tedy mírně větší náklady zde nehrají tak velkou roli. Takto relativně nízká cena je dána i použitím prostorově úsporné sériové sběrnice než by tomu bylo v případě paralelní komunikace zabírající mnohem větší plochu. Jistou nevýhodou by se možná mohla zdát i pomalejší sériová komunikace oproti paralelní, ale jelikož se zde nevyužívá žádné měření v reálném čase, jako je například osciloskop, tak se tato pomalejší komunikace ani neprojeví.

Části potřebné k ověření funkčnosti každého bloku zařízení byly sestaveny a otestovány jejich funkčnost. Jedná se konkrétně o jeden výstupní expander, jeden vstupní expander a dva kanály AD převodníku se změnou rozsahu.

## Seznam použité literatury

- [1] Atmel, ATmega8 datasheet. Dostupný z WWW:  
<[http://www.atmel.com/dyn/resources/prod\\_documents/doc2486.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf)>
- [2] Atmel, ATtiny2313 datasheet. Dostupný z WWW:  
<[http://www.atmel.com/dyn/resources/prod\\_documents/DOC2543.PDF](http://www.atmel.com/dyn/resources/prod_documents/DOC2543.PDF)>
- [3] AVR [online]. Dostupný z WWW: <<http://avr.hw.cz>>
- [4] I<sup>2</sup>C sběrnice [online]. Dostupný z WWW: < <http://www.esacademy.com/faq/i2c/>>
- [5] ISP programátor AVR. Dostupný z WWW: <<http://www.mikrocontroller-projekte.de/>>
- [6] KAINKA, B. *Využití rozhraní PC*. HEL 1996. ISBN 80-902059-3-3
- [7] Maxim, MAX232 datasheet. Dostupný z WWW:  
<<http://datasheets.maxim-ic.com/en/ds/MAX220-MAX249.pdf>>
- [8] Philips, PCF8574 datasheet. Dostupný z WWW:  
<[http://www.nxp.com/acrobat\\_download/datasheets/PCF8574\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8574_4.pdf)>
- [9] Philips, PCF8591 datasheet. Dostupný z WWW:  
<[http://www.nxp.com/acrobat\\_download/datasheets/PCF8591\\_6.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8591_6.pdf)>
- [10] RS232 rozhraní [online]. Dostupný z WWW: < <http://hw.cz/rs-232>>
- [11] Texas Instrument, CD4052 datasheet. Dostupný z WWW:  
<[http://www.icbase.com/ic\\_search/just.asp?urlftp=/TI/TI38790410.pdf](http://www.icbase.com/ic_search/just.asp?urlftp=/TI/TI38790410.pdf)>
- [12] VÁŇA, Vladimír. *Mikrokontroléry Atmel AVR – Programování v jazyce C*. 1. vydání. Praha, BEN 2006. 215 s. ISBN 80-7300-102-0

## Seznam zkratk

I <sup>2</sup> C	Inter-Integrated Circuit
SMBus	System Management Bus
USART	Universal Synchronous Asynchronous Receiver Transmitter
CPU	Central Processor Unit
SDA	Serial Data
SCL	Serial Clock
KO	Klopný Obvod
RS232	Recommended Standard 232
DTE	Data Terminal Equipment
DCE	Data Circuit-terminating Equipment
LSB	Last Significant Bit
MSB	Most Significant Bit
TTL	Transistor Transistor Logic
CMOS	Complementary Metal–Oxide Semiconductor
AVR	Advanced Virtual RISC (Alf Vegard RISC)
ROM	Read Only Memory
RAM	Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
RISC	Reduced Instruction Set Computer
ALU	Arithmetic-Logic Unit
AD	Analog-Digital
DA	Digital-Analog
TWI	Two-Wire Interface
TWBR	Two-Wire Bit-rate Register
TWCR	Two-Wire Control Register
TWSR	Two-Wire Status Register
TWAR	Two-Wire Address Register
TWDR	Two-Wire Data Register
JTAG	Joint Test Action Group
ISP	In-System Programming
MOSI	Master Output Slave Input
MISO	Master Input Slave Output
SCK	Serial Clock
USB	Universal Serial Bus

## Seznam příloh

- A Schéma převodníku
- B Deska plošných spojů
- C Seznam součástek

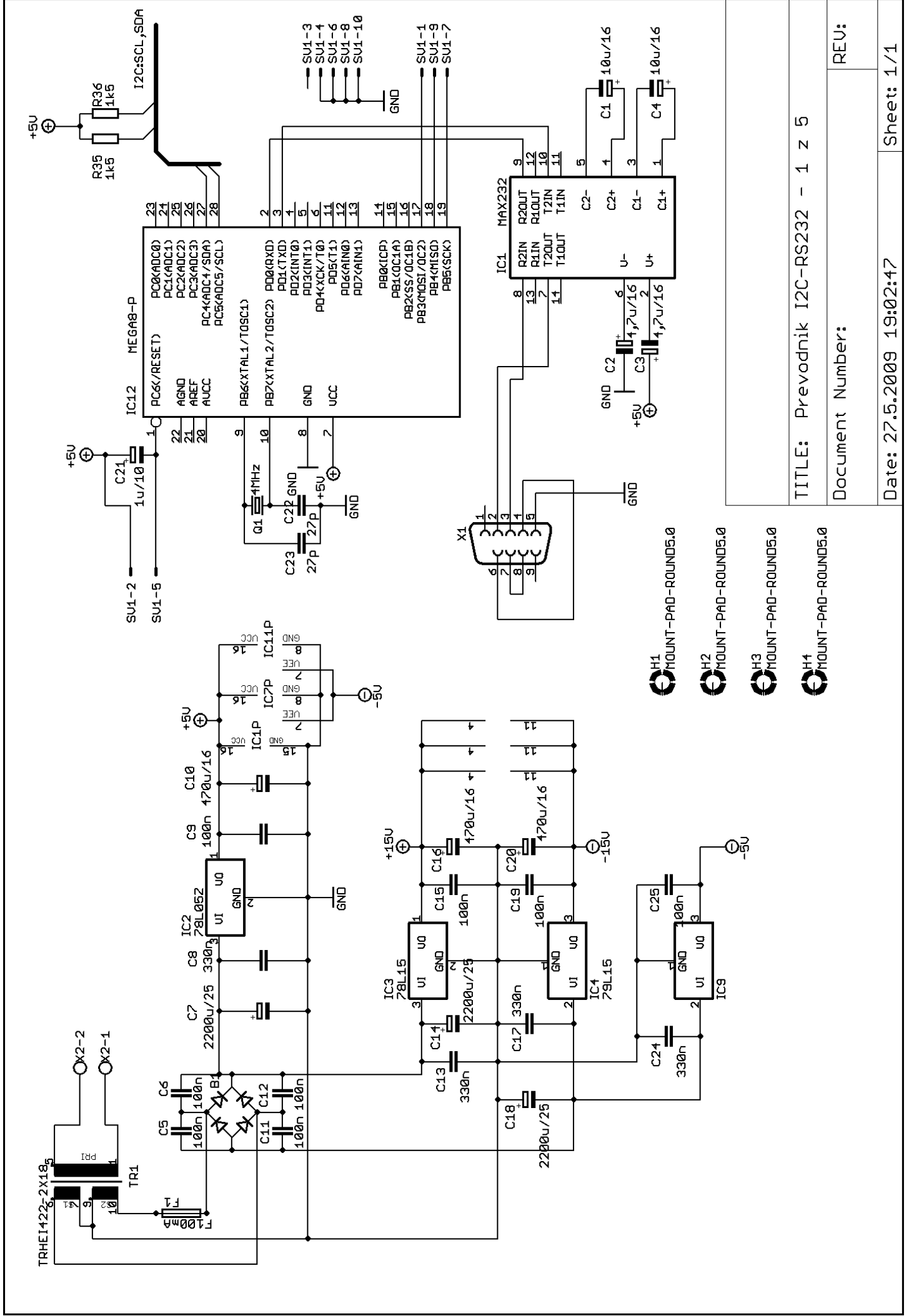


## Obsah CD

Přiložené CD obsahuje:

- Program ATmega8 – obsahuje hlavní zdrojový soubor a potřebné hlavičkové soubory
- Schema – obsahuje schéma a desku plošných spojů pro aplikaci Eagle
- Převodník mezi sběrnicemi – elektronická verze bakalářské práce
- Datasheety – datasheety použitých součástek

## **A Schéma převodníku**

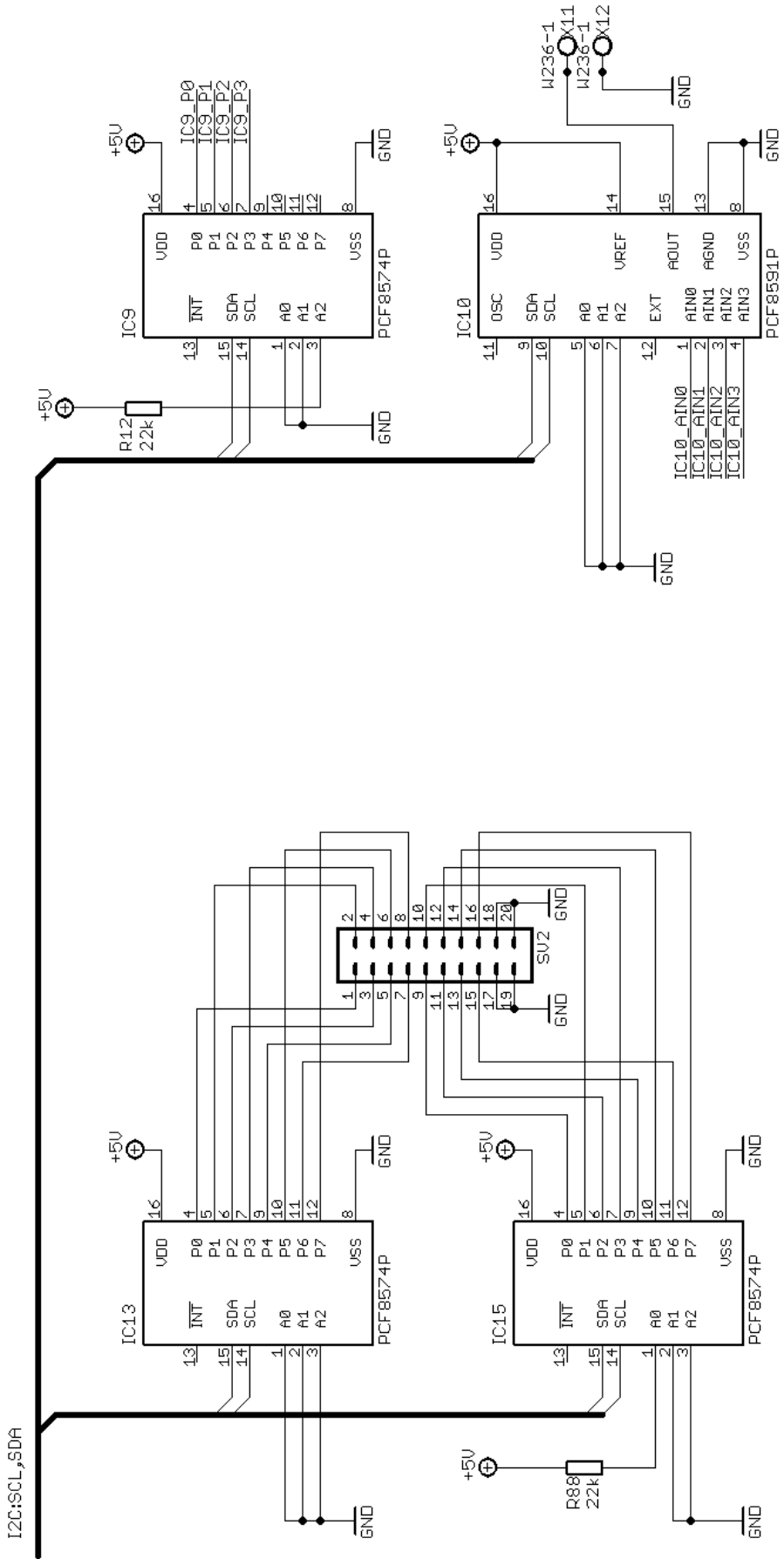


TITLE: Prevodnik I2C-RS232 - 1 z 5

Document Number: REV:

Date: 27.5.2009 19:02:47

Sheet: 1/1

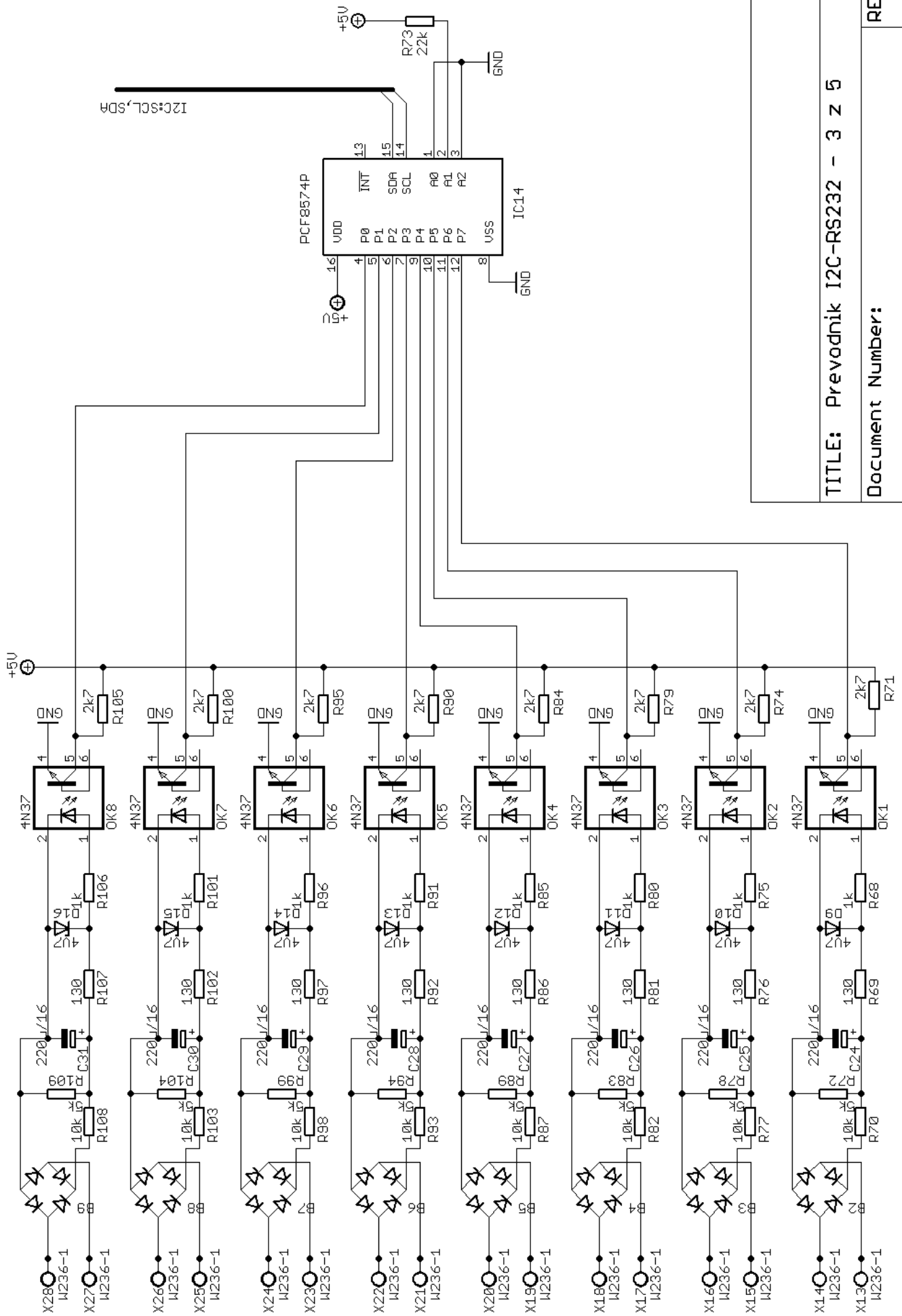


TITLE: Prevodnik I2C-RS232 - 2 z 5

Document Number: REV:

Date: 25.5.2009 20:14:28

Sheet: 1/1



I2C:SCL,SDA

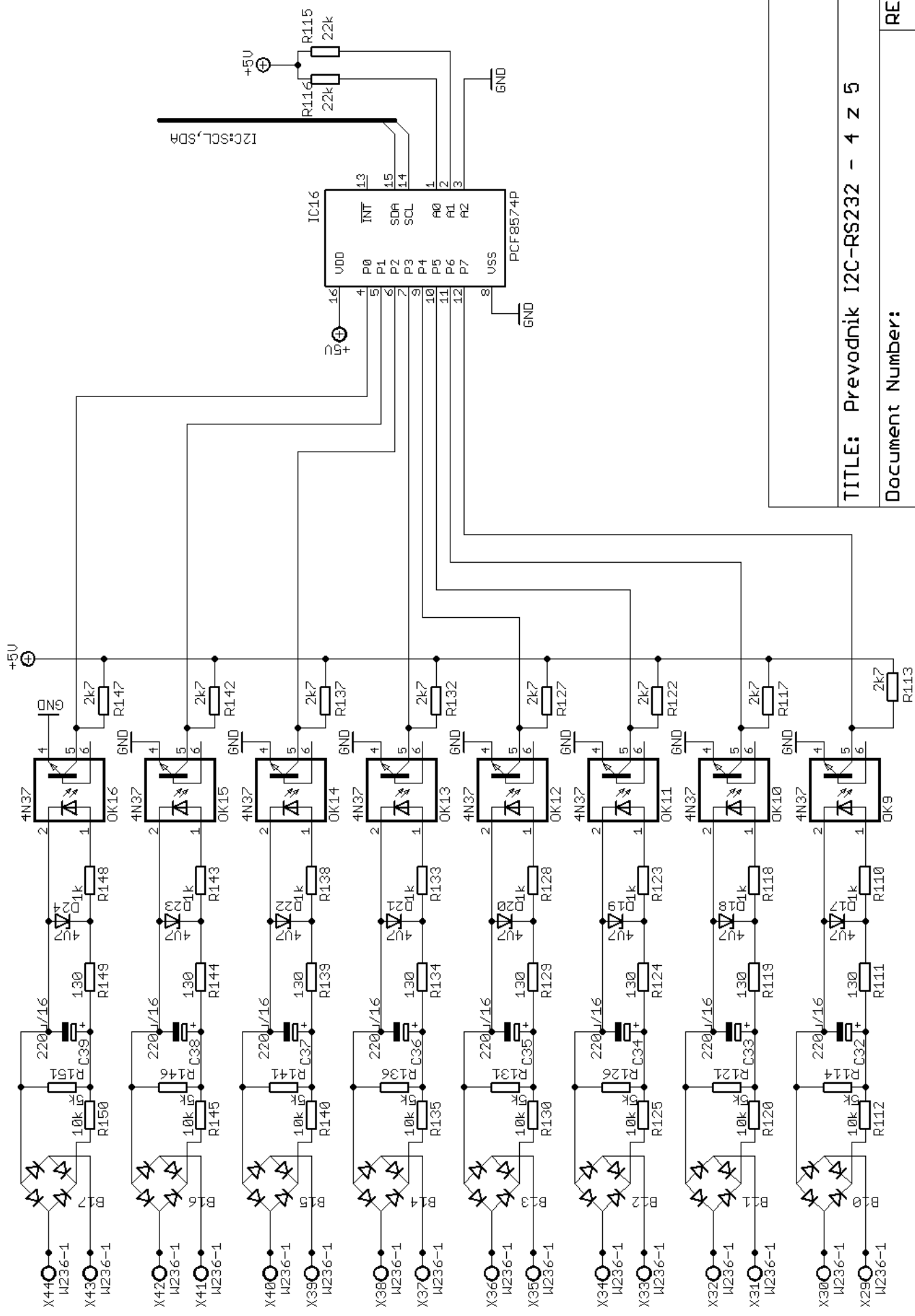
TITLE: Prevodnik I2C-RS232 - 3 z 5

Document Number:

REV:

Date: 25.5.2009 16:11:26

Sheet: 1/1



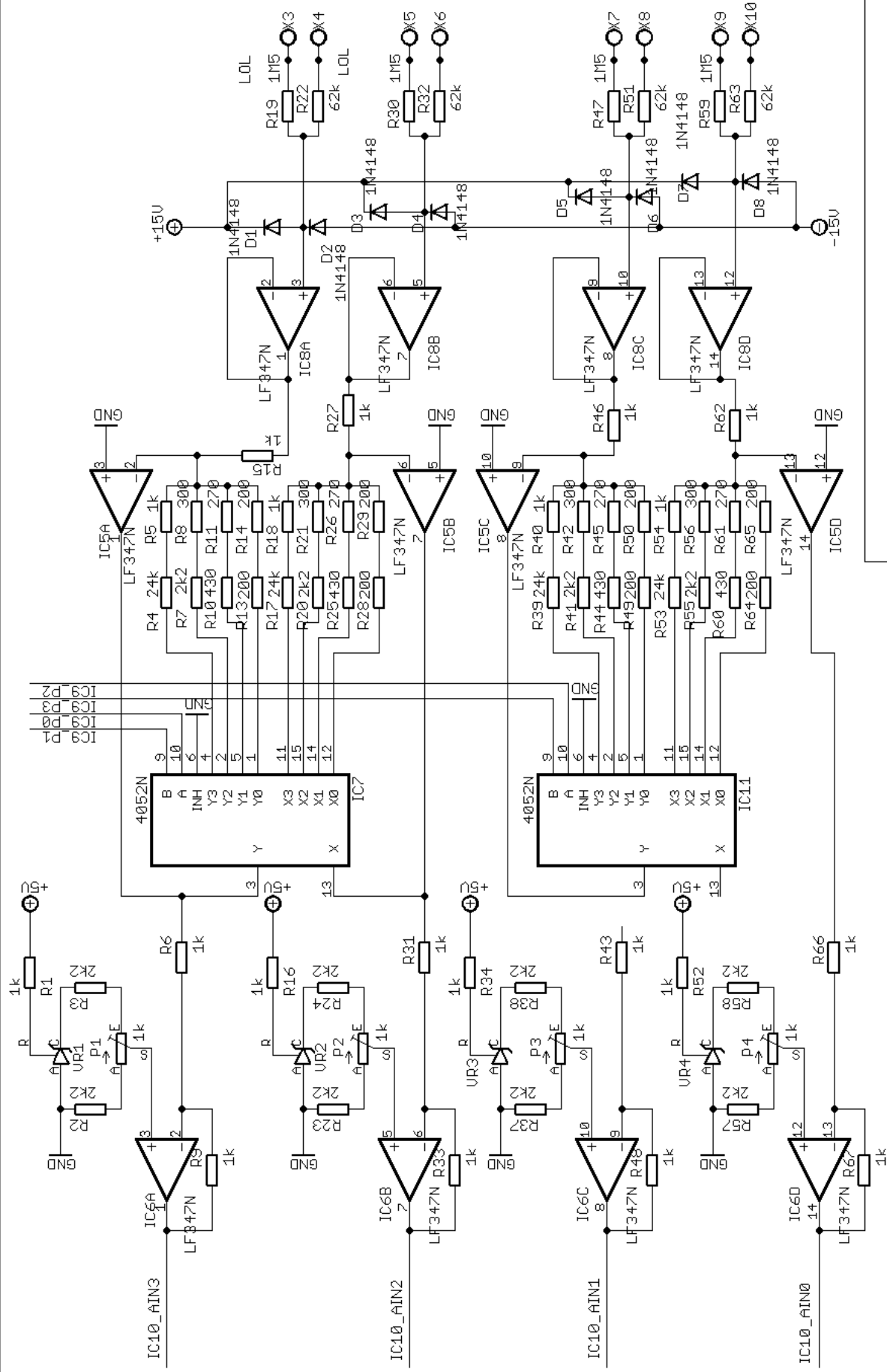
TITLE: Prevodnik I2C-RS232 - 4 z 5

Document Number:

REV:

Date: 25.5.2009 16:11:43

Sheet: 1/1



TITLE: Prevodnik I2C-RS232 - 5 z 5

Document Number:

REV:

Date: 25.5.2009 20:15:51

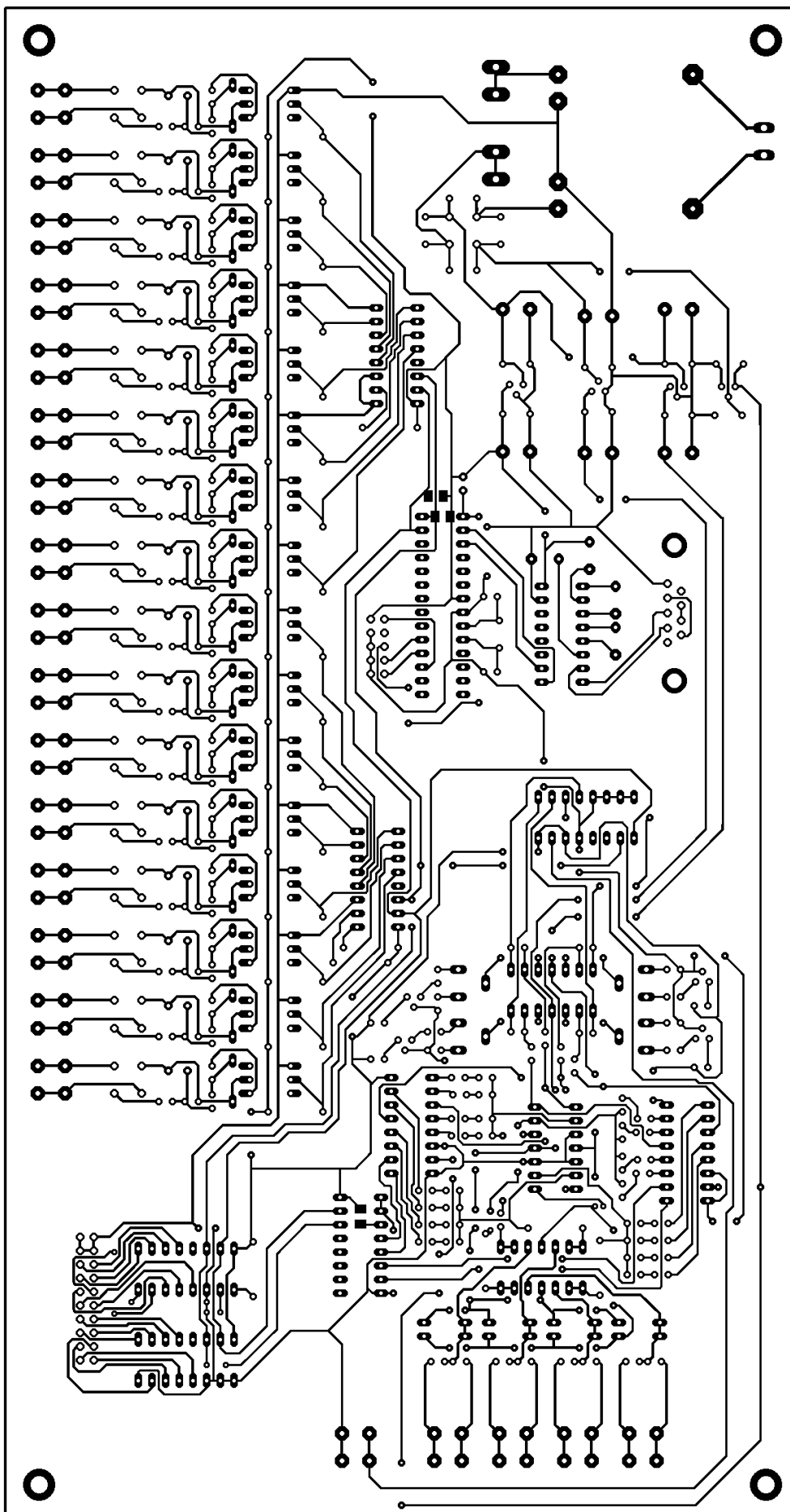
Sheet: 1/1

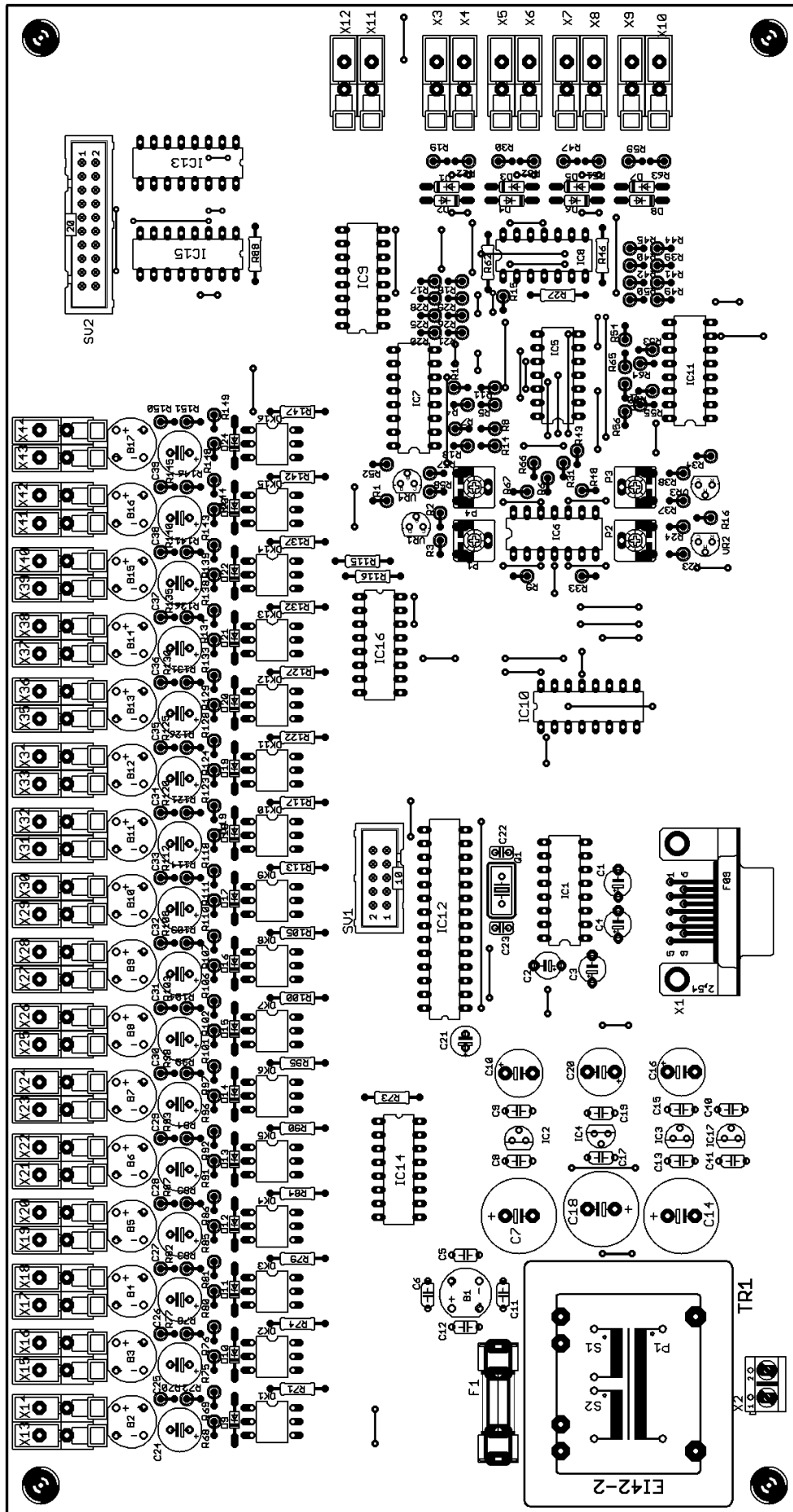
## B Deska plošných spojů

Strana spojů .....	str. 45
Strana součástek .....	str. 46

Na straně spojů lze vidět plošky pro umístění 3 SMD rezistorů. Jsou jimi pull-up rezistory R35, R36 přímo pod mikrokontrolérem a dále R12 pod expandérem IC9.







## C Seznam součástek

130Ω	R69, R76, R81, R86, R92, R97, R102, R107, R111, R119, R124, R129, R134, R139, R144, R149
200Ω	R13, R14, R28, R29, R49, R50, R64, R65
270Ω	R11, R26, R45, R61
300Ω	R8, R21, R42, R56
430Ω	R10, R25, R44, R60
1kΩ	R1, R5, R6, R9, R15, R16, R18, R27, R31, R33, R34, R40, R43, E46, R48, R52, R54, R62, R66, R67, R68, R75, R80, R85, R91, R96, R101, R106, R110, R118, R123, R128, R133, R138, R143, R148
1,5kΩ SMD	R35, R36
2,2kΩ	R2, R3, R7, R20, R23, R24, R37, R38, R41, R55, R57, R58
2,7kΩ	R71, R74, R79, R84, R90, R95, R100, R105, R113, R117, R122, R127, R132, R137, R142, R147
5kΩ	R72, R78, R83, R89, R94, R99, R104, R109, R114, R121, R126, R131, R136, R141, R146, R151
10kΩ	R70, R77, R82, R87, R93, R98, R103, R108, R112, R120, R125, R130, R135, R140, R145, R150
22kΩ	R73, R88, R115, R116
22kΩ SMD	R12
24kΩ	R4, R17, R39, R53
62k	R22, R32, R51, R63
1,5MΩ	R19, R30, R47, R59
trimr 1kΩ	P1, P2, P3, P4
27pF	C22, C23
100nF	C5, C6, C9, C11, C12, C41
1uF/10V	C21
4,7uF/16V	C2, C3
10uF/16V	C1, C4
220uF/16V	C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39
330nF	C8, C13, C15, C17, C19, C40
470uF/16V	C10, C16, C20
2200uF/25V	C7, C14, C18
78L05	IC2
79L05	IC17
78L15	IC3
79L15	IC4
4052N	IC7, IC11
LF347N	IC5, IC6, IC8
MAX232	IC1
MEGA8-P	IC12
PCF8574P	IC9, IC13, IC14, IC15, IC16
PCF8591P	IC10
4N37	OK1 až OK16
B250C1500	B1 až B17
TL431	VR1, VR2, VR3, VR4
1N4148	D1, D2, D3, D4, D5, D6, D7, D8
4V7	D9 až D24
4MHz	Q1
AK500/2	X2
CAN 9 Z 90	X1
MLW10G	SV1
MLW20G	SV2
TRHEI422-2X15	TR1
F100mA	F1
WAGO256-401	X3 až X44