

**Univerzita Pardubice**

**Fakulta Elektrotechniky a Informatiky**

**Monitoring rozhraní I2C**

**Autor práce: Radek Šplíchal**

**Vedoucí práce: Ing. Martin Hájek**

**Bakalářská práce**

**2009**

**University of Pardubice**

**Faculty of Electrical Engineering and Informatics**

**I2C bus monitoring**

**Author: Radek Šplíchal**

**Supervisor: Ing. Martin Hájek**

**Bachelor work**

**2009**

## Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 14. 8. 2009

Radek Šplíchal

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Katedra elektrotechniky  
Akademický rok: 2008/2009

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Radek ŠPLÍCHAL**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Komunikační a mikroprocesorová technika**  
  
Název tématu: **Monitoring rozhraní I2C**

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

Navrhněte zařízení sloužící pro monitoring přístrojové sběrnice I2C. Zařízení se připojí paralelně k měřené sběrnici s tím, že nebude svou přítomností ovlivňovat její činnost. Zařízení bude spolupracovat s počítačem, na kterém bude probíhat vyhodnocení výsledků a ovládání přístroje. Bude-li zařízením detekován platný I2C rámec, pak jej ovládací software bude schopen zobrazit a dekodovat.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**Nagel Ch., Evjen B., Glynn J., Skinner M.,W.: C 2005 – Programujeme profesionálně, Computer Press, 2005**

**Petzold, Ch.: Programování Microsoft Windows Forms v jazyce C, Computer Press, 2006**

**Váňa, V: Mikrokontroléry ATMEL AVR - popis procesoru a instrukční soubor, BEN - technická literatura, Praha 2003**

**I2C-bus specification and user manual, dostupné online na <http://www.standardics.nxp.com/literature/i2c/?search=Specification>**

**Katalogové listy, aplikační poznámky a další informace dosažitelné u výrobců použitých součástí na stránkách www**

Vedoucí bakalářské práce:


**Ing. Martin Hájek**  
Katedra elektrotechniky

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**

  
doc. Ing. Simeon Karamazov, Dr.  
děkan

L.S.

  
Ing. Zdeněk Němec, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2009

## Poděkování:

Děkuji vedoucímu bakalářské práce panu Ing. Martinu Hájkovi za jeho všestrannou pomoc při výběru možností postupu při tvorbě této bakalářské práce.

Dále děkuji všem, jenž mi pomáhali při vyhledávání, ať už studijních zdrojů nebo materiálů použitých při konstrukci samotného zařízení.

V neposlední řadě děkuji přátelům, kteří mi v průběhu celého studia na vysoké škole pomáhali a podporovali mé úsilí.

## Souhrn:

Tato bakalářská práce se zaměřuje na sběrnici typu I2C. Práce se zabývá také teorií sběrnice a komunikací zařízení na sběrnici připojených. Cílem této bakalářské práce je návrh analyzátoru sběrnice I2C tak, aby byl schopen monitorovat sběrnici a platná data odeslat do počítače.

## Klíčová slova:

Sběrnice I2C, I2C monitoring, START podmínka, STOP podmínka

## **Abstract:**

This bachelor work focuses on the type of I2C bus. The work deals with the theory of the bus and communications-wise connected. The object of this bachelor work is the I2C bus analyzer to be able to monitor the bus and valid data to send to your computer.

## **Keywords:**

I2C bus, I2C monitoring, START condition, STOP condition



## Obsah:

Úvod .....	10
1 Sběrnice I2C .....	11
1.1 Fyzická vrstva .....	12
1.2 Linková vrstva .....	14
1.2.1 Arbitrace .....	14
1.2.2 Adresování .....	15
2 Analyzátor I2C .....	17
2.1 Návrh koncepce .....	17
2.1.1 Provedení s externím řadičem .....	19
2.1.2 Provedení s integrovaným řadičem .....	19
2.1.3 Provedení s připojením k PC bez MCU .....	20
2.1.4 Provedení bez řadiče I2C .....	21
2.2 Řadič sběrnice I2C – PCF8584 .....	22
2.3 Návrh konstrukce analyzátoru .....	25
3 Testovací pracoviště .....	27
3.1 Testovací obvod .....	29
3.2 Firmware .....	30
3.3 Výsledky testů .....	32
Závěr .....	34

# Úvod

Sběrnice I2C je dnes rozšířená v různých oblastech elektrotechniky, její využití je velmi široké, ať co se týče v oblasti průmyslového řízení, tak i v dnešní mikroelektronice, jako jsou mobilní telefony, osobní diáře, autorádia, atd.

Pro vývoj zařízení, které bude se svým okolím komunikovat pomocí sběrnice I2C, je důležité mít kontrolu toho, co se na sběrnici skutečně odehrává. Ke sledování sběrnice slouží různá zařízení, např. digitální osciloskopy nebo logické analyzátory, avšak cena těchto zařízení je velmi vysoká. Smyslem této bakalářské práce je navrhnout jednoduché a levné zařízení, které bude umět sběrnici sledovat a tyto data zaznamenávat do PC.

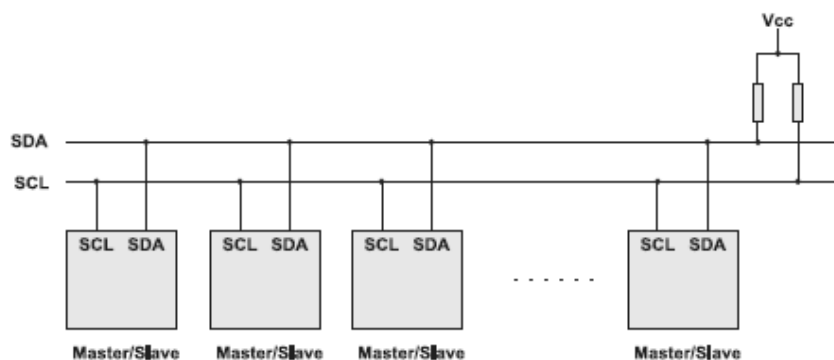
V první kapitole se práce zabývá teoretickým popisem sběrnice I2C, seznámením s jejím fungováním a komunikací po této sběrnici. Druhá kapitola se zabývá samotným analyzátorem, jeho možnostmi řešení konstrukce, výběrem vhodného řešení pro tento typ analyzátoru, z hlediska požadavků na funkce tohoto zařízení. Ve třetí kapitole je popsáno testovací pracoviště, testovacím obvodem a firmwarem v mikroprocesoru, řešení problémů s ovládáním jednotlivých obvodů a výsledky z testů zařízení. V závěru práce je shrnutí dosažených výsledků, vyhodnocení funkčnosti analyzátoru.

# 1 Sběrnice I2C

Základní myšlenkou této sběrnice, při vývoji firmou Philips v 80. letech, bylo zjednodušení počtu vodičů používaných pro komunikaci integrovaných obvodů, tím i jednodušší tvorba desek plošných spojů a zároveň zvýšení odolnosti zařízení proti různým druhům rušení z hlediska elektromagnetické kompatibility.

Sběrnice I2C (Inter-Integrated Circuit) [1] [2] [3] je dvou vodičové datové propojení mezi jedním nebo několika nadřízenými zařízeními (Masters) a podřízenými zařízeními (Slaves). Všechna zařízení jsou připojena na téže sběrnici a jsou cíleně vybírána svými adresami. Adresy i data se přenášejí týmiž vodiči. Sběrnice umožňuje velmi jednoduché propojení mezi několika integrovanými obvody a dodatečné rozšiřování. Mohou být připojeny všechny integrované obvody, které zvládají speciální protokol sběrnice. Mimo integrovaných obvodů RAM, EEPROM, obvodů pro rozšíření portů, A/D a D/A převodníků a obvodů hodinových signálů existuje ještě celá řada speciálních integrovaných obvodů, jako například budiče displejů nebo integrovaných obvodů pro televizní a audio techniku. Sběrnice I2C používá sériovou datovou linku SDA a linku hodinového signálu SCL. Data a adresy se přenášejí podobně jako v posuvných registrech společně hodinovými impulsy. Obě linky je možno používat jako obousměrné, avšak vždy pouze jedním směrem. Jsou vybaveny zdvihacím (pull-up) odporem a mohou být každým účastníkem sběrnice staženy na nízkou úroveň výstupem s otevřeným kolektorem.

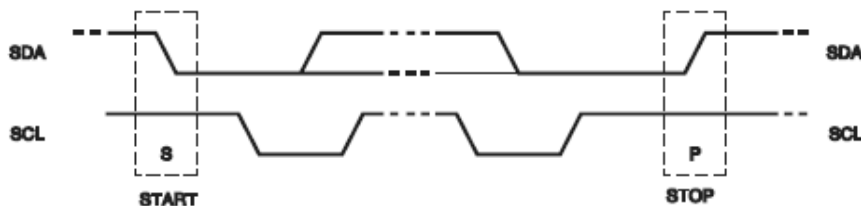
I2C sběrnice typu multimaster. Řeší proto i situace pro přístup na sběrnici pro více zařízení a má zabudovaný mechanismus adresování jednotlivých připojených stanic nebo integrovaných obvodů. Základní koncepce sběrnice I2C je na *Obr. 1.01*. Každá stanice má určenou svou vlastní adresu o délce 7 nebo 10 bitů, která slouží k jejímu výběru.



Obr. 1.01 Celková koncepce sběrnice I2C [3]

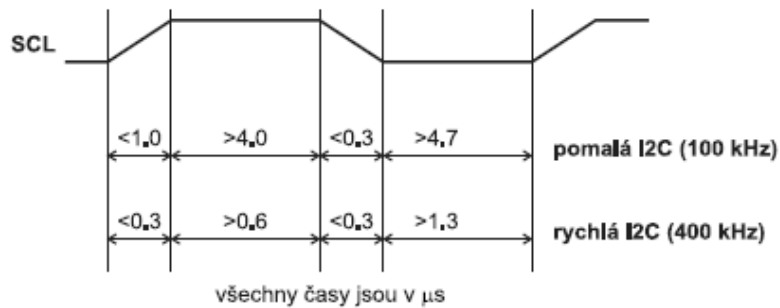
## 1.1 Fyzická vrstva

Jednotlivé stanice rozhraní I2C jsou propojeny jedním datovým vodičem (SDA) a jedním hodinovým vodičem (SCL). Z elektrického hlediska jsou oba vodiče typu otevřený kolektor. Jejich maximální délka je dána jejich nejvyšší přípustnou kapacitou 400 pF. V klidovém stavu (volná sběrnice) jsou oba vodiče v úrovni H. Při probíhající přenosu jsou na SDA vysílány jednotlivé datové bity přičemž platí pravidlo, že logická úroveň na SDA se smí měnit pouze je-li SCL v úrovni L. Toto pravidlo je porušeno ve dvou speciálních případech – vysílání podmínek START a STOP, které se používají k zahájení přenosu, k arbitraci a k ukončení přenosu (viz Obr. 1.02).



Obr. 1.02 Podmínky START a STOP [3]

Maximální přípustná frekvence signálu SCL je podle verze I2C 100 kHz nebo 400 kHz. Pro obě frekvence je specifikována minimální povolená doba setrvání SCL v úrovni L i H (viz *Obr. 1.03*). Při arbitraci i při přenosu dat si jednotlivé stanice synchronizují generátory hodin tak, že trvání úrovně H na SCL je odměřováno 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 (viz *Obr. 1.04*). 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.



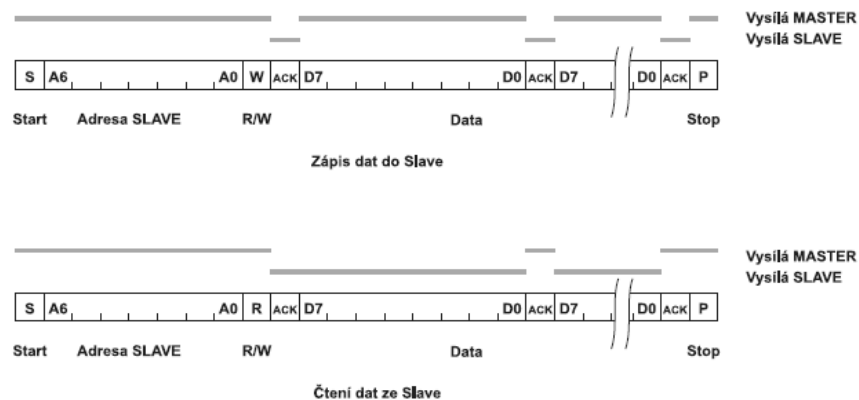
*Obr. 1.03 Časování signálu SCL [3]*



*Obr. 1.04 Synchronizace hodin SCL [3]*

## 1.2 Linková vrstva

Formát rámce se sedmibitovou adresou při přenosu na I2C je na *Obr. 1.05*. Každému přenosu předchází vyslání podmínky START. Potom je vysílána sedmibitová adresa příjemce a jeden bit R/W, který indikuje požadovanou operaci (čtení/zápis). Další bit ACK je vysílán s úrovní H a je určen k potvrzení přijímací stanicí. Dále jsou přenášena data ve směru určeném předchozím bitem R/W. Každý byte je následován jedním bitem ACK. Po ukončení přenosu je vysílána podmínka STOP.

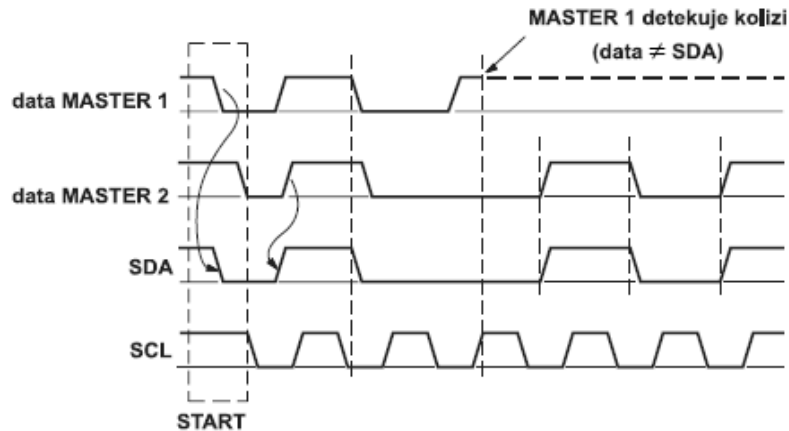


*Obr. 1.05 Formát rámců na sběrnici I2C [3]*

### 1.2.1 Arbitrace

Arbitrace je způsob řešení situací, kdy je na sběrnici více zařízení, která chtějí ve stejnou dobu komunikovat. Pro arbitraci se na I2C používá metoda s detekcí kolize. Každá ze stanic může zahájit vysílání, je-li předtím sběrnice v klidovém stavu. Během vysílání musí neustále porovnávat vysílané bity se skutečným stavem SDA. Je-li zjištěn rozdíl mezi očekávaným a skutečným stavem linky SDA, je to indikace kolize mezi několika stanicemi. Vzhledem k charakteru sběrnice (otevřené kolektory) může k této situaci dojít, pokud

určitá stanice vysílá úroveň H, zatímco jiná stanice vysílá úroveň L. Stanice, která na lince zjistí úroveň L zatímco sama vysílá H musí vysílání okamžitě ukončit (viz Obr. 1.06).



Obr. 1.06 Arbitrace na I2C. Master1 ztrácí právo na sběrnici [3]

## 1.2.2 Adresování

Každá stanice připojená na I2C má přidělenou sedmibitovou adresu. Po zachycení podmínky START porovnávají všechny obvody svou adresu s adresou, která je vysílána na sběrnici. Zjistí-li některý z obvodů shodu, je vysílání určeno právě jemu a musí přijetí adresy potvrdit bitem ACK. Potom přijímá resp. vysílá další data.

V praxi je většinou několik adresních bitů jednotlivých obvodů určeno už při výrobě (viz Tab.1.01). Zbývající bity se volí pomocí příslušných vývodů daného obvodu, které se podle potřeby připojí na úroveň L nebo H. Několik adres je na I2C vyhrazeno pro speciální účely. Např. adresa 0000000 je určena pro vysílání „broadcast“, adresy 0000011, 00001XX a 11111XX jsou rezervovány pro další účely. Adresa 11110aa indikuje 10bitové adresování: aa zde označuje dva nejvyšší bity adresy stanice, zbývajících 8 bitů je vysíláno v následujícím bytu.

Tab. 1.01 Tabulka adres udaných výrobcem [1]

General

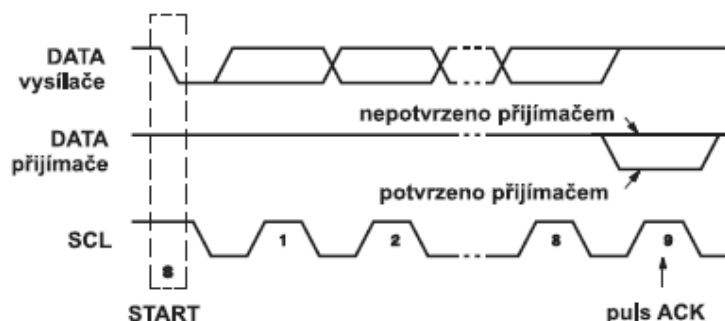
Assigned I<sup>2</sup>C-bus addresses

TYPE NUMBER	DESCRIPTION	I <sup>2</sup> C SLAVE ADDRESSES <sup>(1)</sup>						
		A6	A5	A4	A3	A2	A1	A0
TEA6320/1/2/3	Sound fader control circuit	1	0	0	0	0	0	0
TEA6330	Tone/volume controller	1	0	0	0	0	0	0
TEA6360	5-band equalizer	1	0	0	0	1	1	A
TEA6821/2	Car radio AM	1	1	0	0	0	1	0
TEA6824T	Car radio IF IC	1	1	0	0	0	1	0
TSA5511/2/4	1.3 GHz PLL frequency synthesizer for TV	1	1	0	0	0	A	A
TSA5522/3M	1.4 GHz PLL frequency synthesizer for TV	1	1	0	0	0	A	A
TSA6057	Radio tuning PLL frequency synthesizer	1	1	0	0	0	1	A
TSA6060	Radio tuning PLL frequency synthesizer	1	1	0	0	0	1	A
UMA1000T	Data processor for mobile telephones	1	1	0	1	1	A	A
UMA1014	Frequency synthesizer for mobile telephones	1	1	0	0	0	1	A

**Note**

1. X = Don't care, A = Programmable address bit, P = Page selection bit.

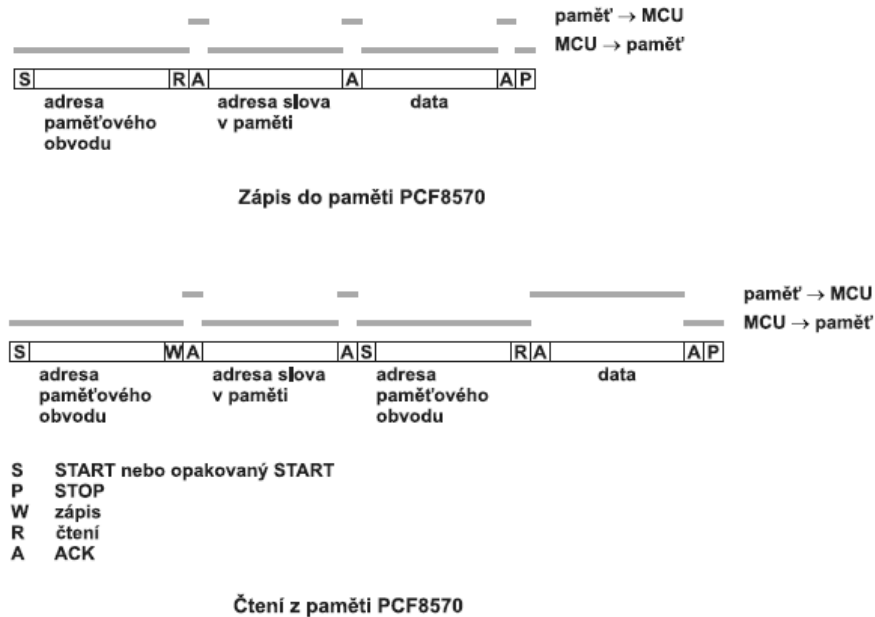
Každý vysílaný byte (včetně adresy) je následován vysláním jednoho bitu ACK. Vysílající stanice jej vysílá v úrovni H. Přijímající stanice potvrzuje přijetí tím, že v době vysílání ACK připojí SDA na úroveň L (viz Obr. 1.07). Pokud vysílající stanice nedostane potvrzení příjmu, ukončí vysílání podmínkou STOP.



Obr. 1.07 Potvrzování příjmu bitem ACK [3]

Na Obr. 1.08 jsou uvedeny příklady komunikace na I<sup>2</sup>C při čtení a zápisu do paměti Philips PCF8570C. Při čtení s náhodným přístupem se musí do paměti nejprve zapsat adresa dat, která potom mohou být zapsána nebo přečtena. Před vlastním čtením dat z paměti opakuje master (tj. mikrokontrolér) podmínku START a vysílá rámec s povelům READ.





Obr. 1.08 Zápis a čtení paměti Philips PCF8570 [3]

## 2 Analyzátor I2C

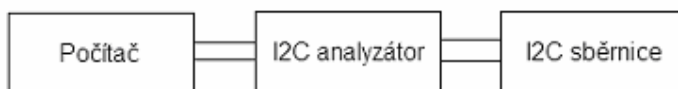
Analyzátor sběrnice I2C by mělo být jednoduché zařízení, které se napojuje paralelně k měřené sběrnici. Cíl analyzátoru sběrnice I2C je umožnit uživateli monitorovat dění na sběrnici, zároveň převést data ze sběrnice do počítače, uživatel poté může s daty v počítači podle potřeb dále pracovat. Analyzátor by neměl dění sběrnici svým působením nijak ovlivňovat, měl by to být pasivní prvek.

### 2.1 Návrh koncepce

Analyzátor sběrnice I2C navrhovaný v této bakalářské práci je zařízení, které je připojeno mezi uživatelský počítač a sběrnici I2C. Zařízení lze k počítači připojit několika způsoby. Jedním z možných způsobů je sériový port, ale ten je již zastaralý, většina dnešních počítačů již ani tento port nemá. Otázkou sériového portu by byla také jeho přenosová rychlost, aby stačil data

odesílat do počítače. Paralelní port je další možností, jak zařízení připojit k počítači. Paralelní port je rychlejší než sériový port, avšak z pohledu dnešních počítačů jeho využití klesá a moderní počítače jím již také nedisponují. Jako nejvíce perspektivní připojení se jeví připojení přes sběrnici USB, je to univerzální sériová sběrnice. Napájecí napětí sběrnice USB je 5V. Zařízení připojená ke sběrnici USB mohou být napájena přímo ze sběrnice, pokud jejich odběr nepřekročí 100mA, příp. se může softwarově aktivovat vysokoodběrový režim, kdy sběrnice poskytne až 500mA (max. jedno zařízení na celé USB sběrnici). Pokud mají USB zařízení vlastní zdroj (např. počítač má vlastní systém pro distribuci napájení nezávislý na USB), je řízen USB sběrnici (zapínání, vypínání, atd.). Každý hub USB umožňuje omezený přenos výkonu pro napájení USB zařízení, přičemž zařízení může být současně napájeno z vlastního zdroje.

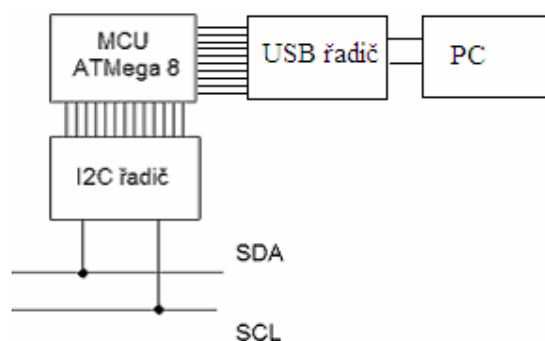
Dalším problémem je připojení ke sběrnici I2C, to je také možné řešit několika způsoby, např. připojení mikrokontroléru s externím řadičem, tedy pokud není řadič integrován na mikrokontroléru, což může být další možnost. Jde také provést propojení externího řadiče a vhodného USB řadiče bez mikrokontroléru. Připojení sběrnice I2C je možné připojit také na vhodné vývody mikrokontroléru a v něm simulovat řadič sběrnice I2C. Blokové schéma připojení analyzátoru je na obrázku *Obr 2.01*.



*Obr 2.01 Blokové schéma připojení I2C analyzátoru*

## 2.1.1 Provedení s externím řadičem

Tento systém využívá klasického mikroprocesoru, který nemá k dispozici řadič zpracovávající data přijatá ze sběrnice I2C. Tento systém tedy řeší problém s komunikací pomocí externího řadiče, který je integrován na samostatném čipu. Princip je zobrazen na obrázku *Obr 2.02*.



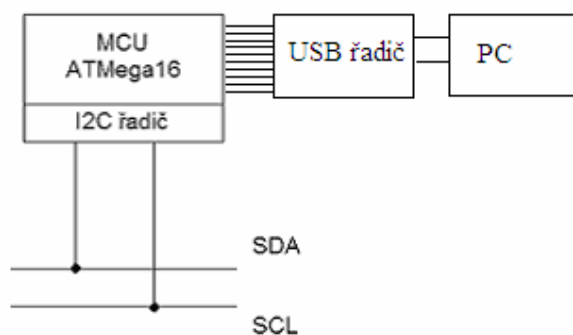
*Obr 2.02 Princip připojení k I2C*

Základní části ze kterých se tento systém skládá jsou mikroprocesor a řadič. Mikroprocesor, např. ATMEGA8, který řídí ovládání řadiče a zpracovává přijatá data. Řadič – přijímá a ukládá zprávy do bufferů, ve kterých si je může mikroprocesor vyzvednout nebo do nich data zapsat. Řadič je obousměrný konvertor, převádí data přijatá ze sběrnice na posloupnost jedniček a nul a data od řadiče na úrovně napětí pro sběrnici. Vhodným řadičem sběrnice I2C může být např. PCF8584 [4], který má v sobě i funkci sledovače sběrnice I2C.

## 2.1.2 Provedení s integrovaným řadičem

Tento systém využívá při zpracovávání dat mikroprocesor, který má na svém čipu integrovaný řadič I2C. Jedná se tedy o systém jednodušší pro konstrukci, než tomu bylo u systému předchozího. Avšak integrované řadiče nepodporují funkci sledovače sběrnice, jsou schopné komunikovat pouze

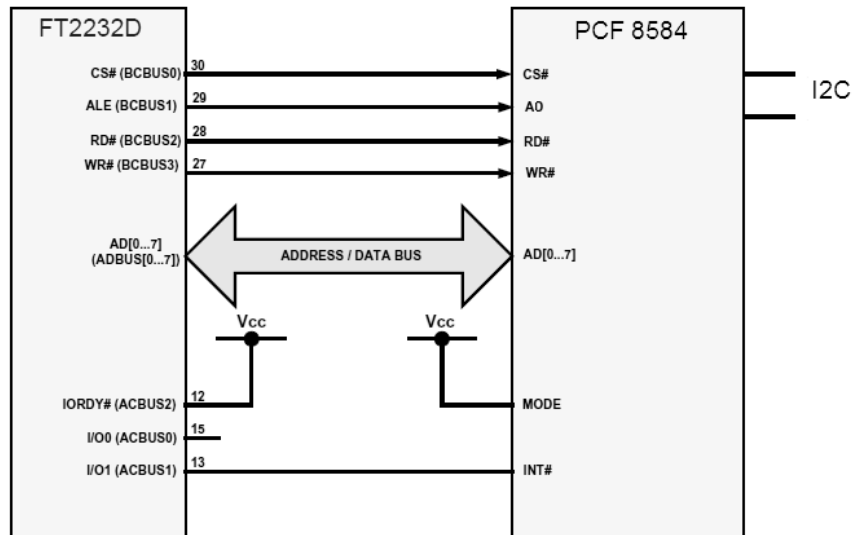
s pevně nastavenou adresou I2C, což pro jeho použití v analyzátoru není vhodné. Většina dnešních mikrokontrolérů má řadič sběrnice I2C, avšak z důvodu drahé licence používání I2C se jmenuje rozhraní jinak, např. výrobce mikrokontrolérů společnost ATMEL [5] rozhodl používat princip sběrnice I2C, ale pod svým označením TWI (Two Wire Interface) jako dvouvodičový interface, čímž obešel nutnost licencování svých řadičů. Princip tohoto řešení je zobrazen na obrázku *Obr. 2.03*.



*Obr 2.03 Princip možnosti připojení I2C*

### 2.1.3 Provedení s připojením k PC bez MCU

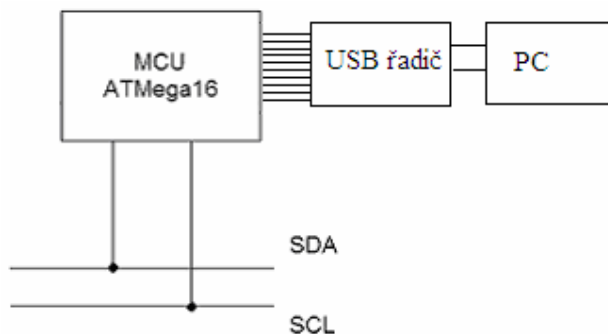
Tento systém data pouze přijímá nebo odesílá. Dokáže v bufferech udržet tolik zpráv, kolik mu umožní použitý řadič. V tomto systému se data nezpracovávají, jsou do PC odesílány přímo z řadiče pomocí speciálního USB řadiče – například FT2232 [6]. Zapojení s FT2232 by tedy bylo nejjednodušší technické řešení I2C analyzátoru z hlediska součástek. Celá komunikace po sběrnici I2C by se musela řešit složitě v počítači. Dalším úskalím by byla nutnost ovládat externí řadič I2C přes řadič USB přímo z počítače. Vynechání mikrokontroléru znamená řešit mnoho složitých situací, které mikrokontrolér řeší jednoduchým způsobem sám. Toto provedení bez mikrokontroléru je znázorněno na obrázku *Obr. 2.04*.



Obr 2.04 Připojení I2C sběrnice k PC bez mikroprocesoru[6]

## 2.1.4 Provedení bez řadiče I2C

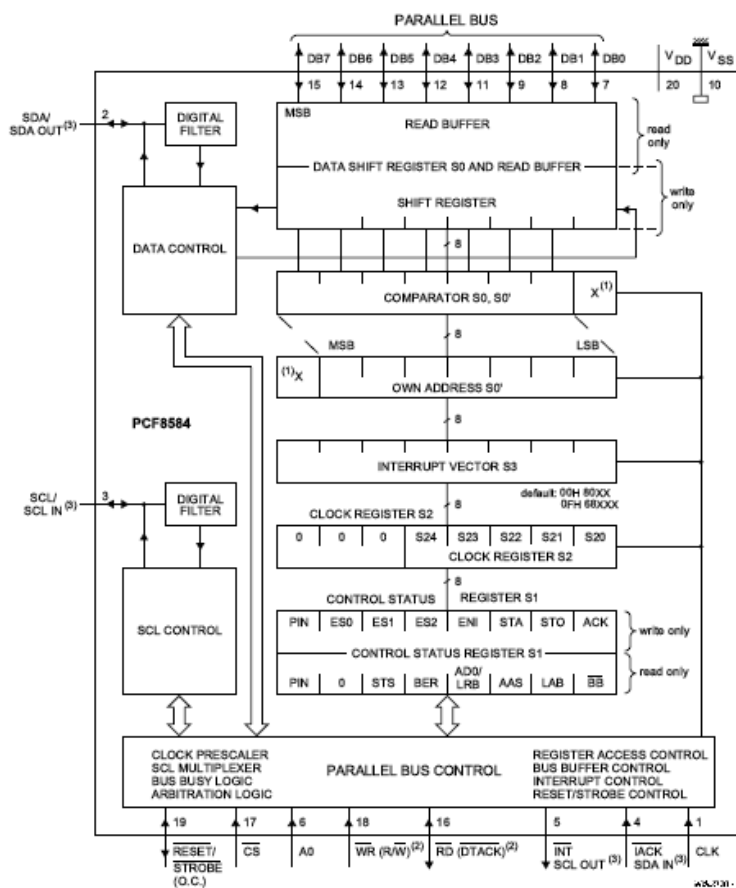
V tomto provedení je vynechán řadič I2C, ten je simulován softwarově v mikrokontroléru. Toto provedení má velké nároky na mikrokontrolér, na jeho rychlost zpracovávání instrukcí, tak aby stačil sledovat změny na sběrnici I2C a zároveň tyto data zpracovávat. Při programování by se musel nejdříve odladit program simulace řadiče sběrnice I2C a potom teprve řešit propojení s počítačem. Blokově je provedení zobrazeno na obrázku Obr. 2.05.



Obr 2.05 Připojení I2C sběrnice bez řadiče I2C

## 2.2 Řadič sběrnice I2C - PCF8584

Obvod PCF8584 funguje jako převaděč mezi paralelní sběrnicí mikroprocesoru a sběrnicí I2C. Obvod je kompatibilní s mikroprocesory řady 8051 společnost INTEL, řadou mikroprocesorů 6800 nebo 68000 společnosti MOTOROLA a v řadě mikroprocesorů Z80 společnosti ZILOG. Obvod umí komunikovat po sběrnicí I2C jako podřízené zařízení (SLAVE), ale i jako nadřízené zařízení (MASTER). Obvod je napájen 5V a je postavený technologií CMOS. Blokové schéma obvodu je na obrázku *Obr. 2.06* [4].



*Obr. 2.06* Blokové schéma obvodu PCF8584 [4]

Na obrázku *Obr. 2.06* vlevo je připojení ke sběrnicí I2C, nahoře je naznačeno vyvedení paralelní sběrnice pro komunikace s mikroprocesorem, dole jsou ovládací signály. Obvod obsahuje několik osmi bitových registrů. Obvod má pouze jednu datovou bránu, proto k nastavování ostatních registrů

slouží přístup do registru S1 a podle jeho nastavení obvod zapisuje nebo vyčítá data z ostatních registrů. Registr S0 je datový registr, zde obvod vystavuje data přijatá ze sběrnice I2C. Registr S0' je registr vnitřní adresy, pokud je tato adresa v hodnotě 00h, pak se obvod chová jako pasivní sledovač sběrnice I2C. Registr S3 slouží pro nastavení přerušení pro módy MASTER nebo SLAVE, pro mód sledovače je tento registr neaktivní. V registru S2 se nastavuje hodnota externího oscilátoru a rychlost sběrnice I2C. Registr S1 je hlavní kontrolní a nastavovací registr je rozdělen do dvou částí, z jedné je možné data pouze číst, to je STATUS registr S1 a do druhé lze pouze data zapisovat, to je CONTROL registr S1. každý bit tohoto registru má svůj specifický význam. Registr S1 je na obrázku *Obr. 2.07*.

CONTROL/STATUS	BITS								MODE
Control <sup>(1)</sup>	PIN	ESO	ES1	ES2	ENI	STA	STO	ACK	write only
Status <sup>(2)</sup>	PIN	0 <sup>(3)</sup>	STS	BER	AD0/LRB	AAS	LAB	BB	read only

*Obr 2.07 Registr S1 obvodu PCF8584 [4]*

PIN bit je jediný bit, který lze zároveň číst i zapisovat, pokud je tento bit v hodnotě 1, pak ostatní bity stavu jsou resetovány do nuly. Jedná se v podstatě o možnost softwarového resetu. Bit ESO (Enable Serial Output) zapíná nebo vypíná interface sběrnice I2C. Bity ES1 a ES2 slouží pro výběr přístupu k ostatním registrům, jak je zobrazeno v tabulce *Tab. 2.01*.

*Tab. 2.01 Tabulka nastavení bitů ES1 a ES2 [4]*

INTERNAL REGISTER ADDRESSING 2-WIRE MODE				
A0	ES1	ES2	ACK	FUNCTION
<b>ESO = 0; serial interface off (see note 1)</b>				
1	0	X	1 <sup>(2)</sup>	R/W S1: control
0	0	0	1 <sup>(2)</sup>	R/W S0': (own address)
0	0	1	1 <sup>(2)</sup>	R/W S3: (interrupt vector)
0	1	0	1 <sup>(2)</sup>	R/W S2: (clock register)
<b>ESO = 1; serial interface on</b>				
1	0	X	1	W S1: control
1	0	X	1	R S1: status
0	0	0	1	R/W S0: (data)
0	0	1	1	R/W S3: (interrupt vector)
X	0	X	0	R S3: (interrupt vector ACK cycle)

Bit ENI zapíná přerušení. Bity STA a STO slouží pro vyslání START nebo STOP podmínky. Bit ACK je tu pro nastavení potvrzování bitem ACK. Toto byla část CONTROL registru S1, dále bude popsána STATUS část

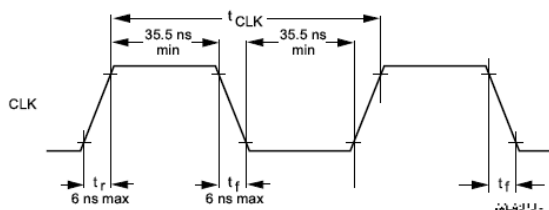
registru S1. Bit STS je nastavován externě, reaguje na STOP podmínku detekovanou na sběrnici I2C. Bit BER slouží k detekci chyby na sběrnici I2C. Bit LRB/AD0 zobrazuje buď hodnotu posledního přijatého bitu nebo pokud je zařízení jako podřízené a je detekována jeho adresa na sběrnici I2C. Bit AAS je nastaven do hodnoty 1, pokud je zařízení nastaveno jako podřízené. Bit LAB značí ztrátu arbitrace. A poslední bit BB značí dění na sběrnici, pokud je tento bit v hodnotě 0, pak se na sběrnici právě vysílá.

Pokud je obvod nastaven ve sledovacím módu, pak:

- je vždy vybrán pro příjem dat ze sběrnice
- vždy je v módu podřízeného zařízení
- nikdy negeneruje bit ACK jako potvrzení přijatých dat
- obvod nikdy negeneruje požadavek na přerušeni
- bit BB je nastaven do hodnoty 0 po detekci START podmínky a po detekci STOP podmínky se vrací do hodnoty 1
- dění na sběrnici je monitorováno bitem PIN, který je v hodnotě 0, je-li příchozí byte přijat, s každým dalším příchozím bitem je v hodnotě 1
- bit AAS je nastaven do hodnoty 1 s každou detekovanou START podmínkou a nastaven do hodnoty 0 po každém devátém pulzu sběrnice I2C.

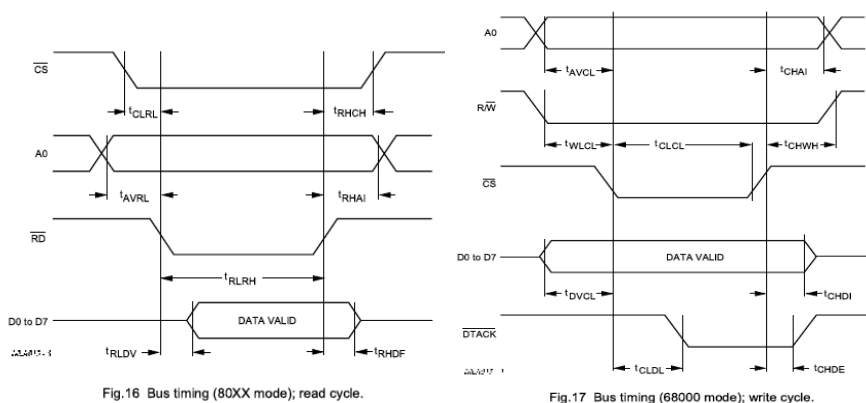
Obvod PCF8584 nemá integrovaný oscilátor, musí být proto do něho přiveden signál z externího oscilátoru, na obrázku *Obr. 2.08* jsou požadavky na hodnoty signálu oscilátoru.





Obr 2.08 Požadavky na oscilátor [4]

Pro správné přijetí nebo odeslání dat obvodem PCF8584 musí být také dodrženo časování čtení a příjmu, což je znázorněno na obrázku Obr. 2.09.



Obr 2.09 Požadavky na časování [4]

## 2.3 Návrh konstrukce analyzátoru

Konstrukce analyzátoru se musela podřídit několika faktorům. Provedení bez řadiče sběrnice I2C by vyžadovalo velmi rychlý procesor, což by znamenalo vysoké náklady na stavbu zařízení. Provedení bez mikroprocesoru by vyžadovalo veškerou komunikaci řešit v počítači, což je časově náročné na programování a odladění programu, také by výsledný program mohl být trochu nepřehledný. Provedení s integrovaným řadičem není vhodné, protože integrované řadiče neumí přijímat vysílání po sběrnici ze všech adres, i kdybychom znali adresu zařízení, které zprávu vysílá, řadič mikroprocesoru by data potvrdil vysláním bitu ACK, čímž by mohl ovlivnit chování sběrnice, pokud by zařízení pro které byla zpráva určena

neodpovídalo, řadič mikroprocesoru by tuto komunikaci uzavíral posláním ACK za toto zařízení, čímž by zprovoznil nefunkční vysílání na sběrnici.

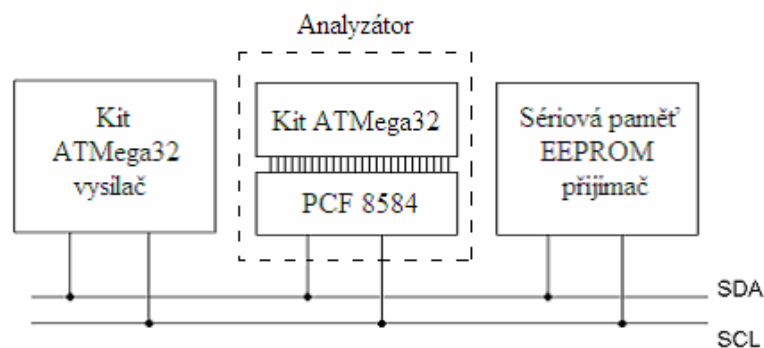
Základem analyzátoru byl zvolen mikroprocesor ATmega32. Pro komunikaci mezi mikroprocesorem ATmega32 a sběrnici I2C byl vybrán řadič PCF8584. USB lze k mikroprocesoru připojit mnoha způsoby, jedním z možných připojení je připojit konektor pomocí sériového rozhraní převodníku (např.: pomocí FT232RL), ten pracuje jako převodník USB - UART a ovladače simulují sériový port, to má výhodu v tom, že se může při tvorbě obslužného softwaru lze využít třídy komunikace SERIAL z .NET. Hlavní nevýhodou řadiče FT232 je oproti FT245, že je pomalejší. Pro komunikaci mikroprocesoru ATmega32 a sběrnice USB, byl zvolený řadič FT245 [9], ten má osmi bitovou paralelní sběrnici pro komunikaci s mikroprocesorem, z toho plyne vysoká datová propustnost, avšak složitější program pro ovládání tohoto řadiče.

Schéma analyzátoru je v příloze A1. V příloze A2 je rozpis použitých součástek. Celé zařízení je koncipováno tak, že je napájeno ze sběrnice USB přes cívku L1 o velikosti 10mH/10A a dva kondenzátory C13 vyhlazovací elektrolytický 100uF/50V a C14 blokovací keramický o velikosti 100nF. Jako signalizace napájení je zde LED dioda D1 s předřadným odporem R3 o velikosti 2k2 ohmu. Na bráně PA mikroprocesoru ATmega32 je napojena datová sběrnice řadiče FT245BM, sem na tuto bránu vystavuje mikroprocesor data k odeslání na sběrnici USB nebo odtud data přijímá. Na bráně PB jsou ovládací signály řadiče FT245BM. Na napájení řadiče FT245BM jsou přidány blokovací keramický kondenzátor C4 o velikosti 100nF a elektrolytický kondenzátor C3 o velikosti 100uF/50V, ty jsou zde z důvodu vyhlazení napájecího napětí. Na bráně PC jsou připojeny řídicí signály řadiče sběrnice I2C obvodu PCF8584, zároveň je tu připojen programovací konektor JTAG pro možnost ladění programu v mikroprocesoru pro případné problémy s ožívováním zapojení. Na poslední bráně PD je datová sběrnice řadiče sběrnice I2C obvodu PCF8584, tudy putují data ze sběrnice I2C do mikroprocesoru nebo touto cestou mikroprocesor nastavuje řídicí registry řadiče PCF8584. K napájení obvodu PCF8584 jsou rovněž připojeny

kondenzátory C10 elektrolytický 100uF/50V a C12 keramický 100nF z důvodu vyhlazení napájecího napětí. Obvod PCF8584 je připojen ke sběrnici I2C přes odpory R1 a R2 o velikosti 100 ohmů, takto je připojen v aplikační poznámce výrobce [7]. Napájení mikroprocesoru je vyhlazeno elektrolytickým kondenzátorem C8 o velikosti 2M2 F a blokováno keramickým kondenzátorem C9 o velikosti 100nF. Mikroprocesor je řízen krystalem o hodinové frekvenci 12MHz, to je nejvyšší hodnota frekvence, při které je schopen pracovat obvod PCF8584, obvod PCF8584 nemá vlastní oscilátor, musí být proto napojen na oscilátor externí. Mezi přívod hodin do obvodu PCF8584 a vývod mikroprocesoru XTAL2 jsou vložena dvě hradla NAND obvodu 74HCT00 z důvodu potřeby natvarování signálu pro řadič PCF8584 a zároveň jako impedanční oddělení vstupu mikroprocesoru XTAL2 a vstupu CLK řadiče PCF8584. Pro správnou funkci mikroprocesoru je také nutný resetovací obvod složený z diody D5 D1N4148, odporu R8 o velikosti 4k7 ohmu a keramického kondenzátoru C11 o velikosti 100nF.

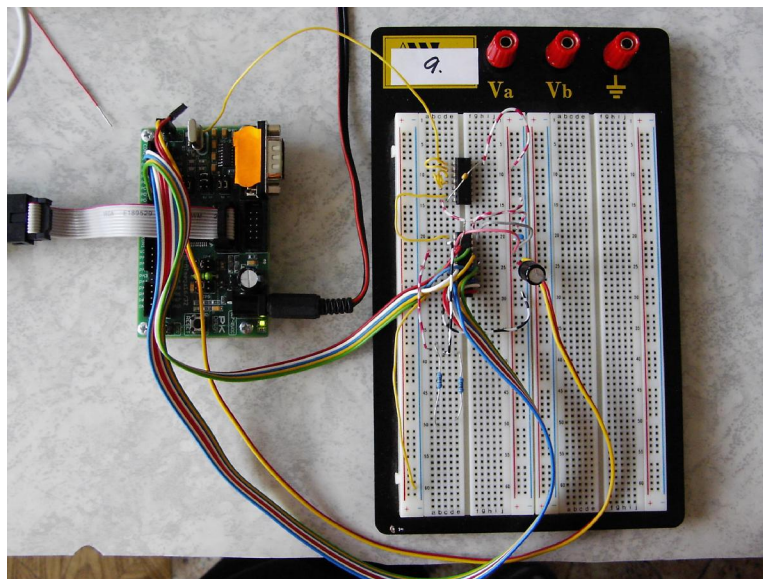
### 3 Testovací pracoviště

Navrženou konstrukci jsem se rozhodl otestovat na testovacím pracovišti. Blokové schéma zapojení testovacího pracoviště je na obrázku *Obr. 3.01*.



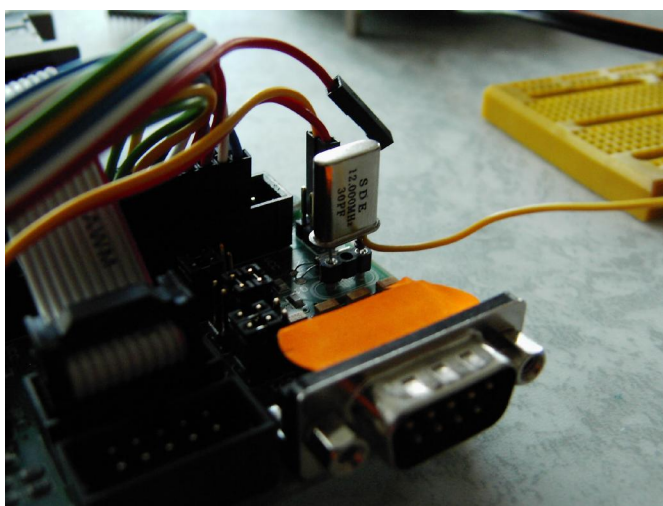
*Obr. 3.01 Blokové schéma testovacího pracoviště*

I2C analyzátor byl realizován pomocí vývojového kitu s procesorem ATMega32 a řadičem PCF8584 zapojeném v nepájivém kontaktním poli, podle schématu navrženém v předchozí kapitole . Fotografie realizace je na obrázku *Obr. 3.02*. Schéma zapojení je v příloze C.



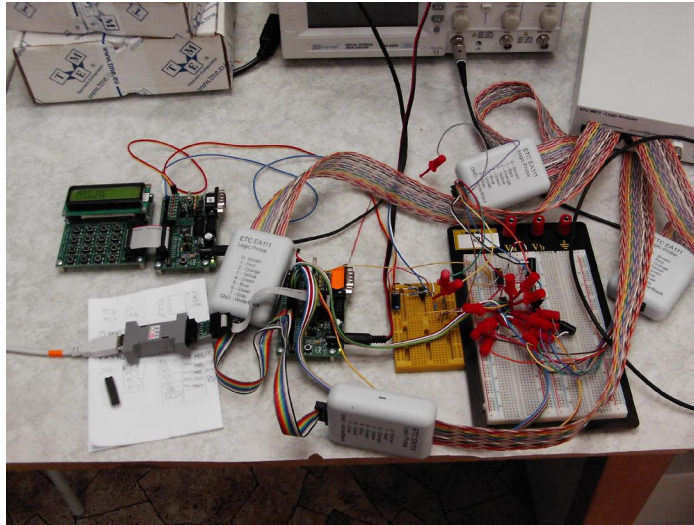
*Obr. 3.02 Fotografie realizace analyzátoru*

Pro zapojení obvodu PCF8584 musel být vyveden od procesoru signál hodin. Toto bylo realizováno tak, že se vyvedl signál od krystalu z kitu mikroprocesoru, jak je zobrazeno na obrázku *Obr. 3.03*.



*Obr. 3.03 Fotografie připojení na krystal*

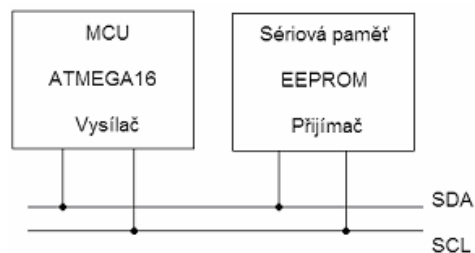
Celé testovací pracoviště je vyobrazeno na obrázku *Obr. 3.04* a vzniklo připojením analyzátoru k testovacímu obvodu ke sběrnici I2C. Místo řadiče USB byl k branám mikroprocesoru připojen logický analyzátor, aby bylo možné ověřit data a průběhy signálů z obvodu PCF8584. Dále byly na analyzátor připojeny signály ze sběrnice I2C a datová brána obvodu PCF8584.



*Obr. 3.04 Fotografie testovacího pracoviště*

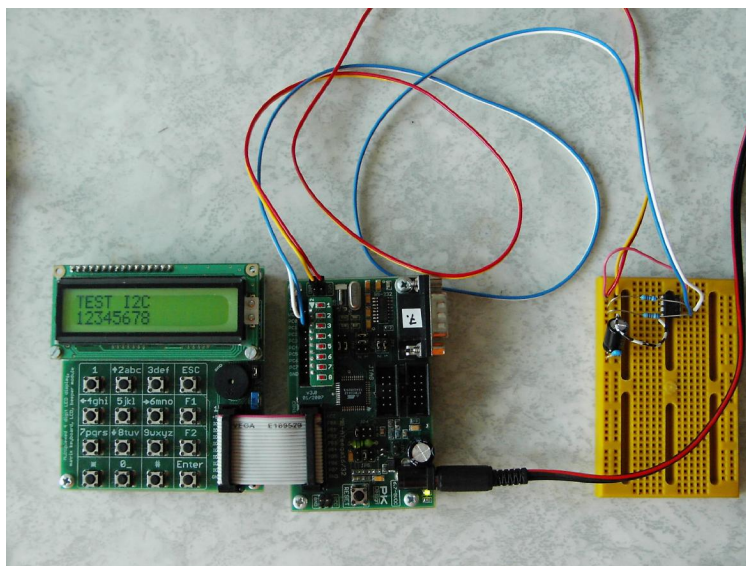
### 3.1 Testovací obvod

Zapojení testovacího obvodu bylo realizováno takto: mikroprocesor s integrovaným řadičem I2C jako vysílač rámců I2C a sériovou paměť EEPROM, které slouží I2C jako datová sběrnice, jako přijímač rámců. Blokové schéma zapojení testovacího obvodu je na obrázku *Obr. 3.05*.



*Obr. 3.05 Blokové schéma testovacího obvodu*

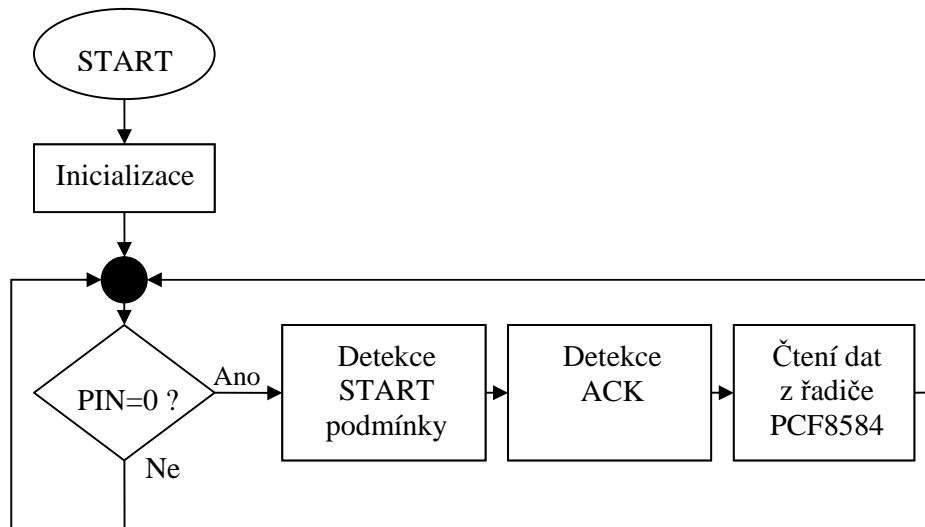
Zapojení testovacího obvodu bylo realizováno částečně pomocí vývojových kitů společnosti PK-Design a částečně na nepájivém kontaktním poli. V mikroprocesoru ATmega32 je nahrán program pro komunikaci se sériovou pamětí EEPROM, a to tak, že procesor do paměti po I2C nahraje nejdříve adresu místa v paměťovém prostoru, kam chce zapisovat a pak jednu stránku paměti, tzn. nahraje 8 po sobě jdoucích bytů a po té je z paměti vyčte nazpět a vypíše přijatá data na LCD display, tím je potvrzeno, že spolu obvody komunikují. Realizace tohoto testovacího obvodu a důkaz funkce je na fotografii na *Obr. 3.06*. Program testovacího obvodu je v příloze B1, pro program byly použité knihovny pro práci s LCD displejem a komunikaci po I2C [8]. Schéma zapojení je umístěno v příloze B2.



*Obr. 3.06 Fotografie testovacího obvodu*

## 3.2 Firmware

Aby byl analyzátor schopen provozu, je nutné do mikroprocesoru naprogramovat firmware, který bude zprostředkovávat komunikaci mezi sběrnici a mikroprocesorem [10]. Tento firmware musí být schopen měnit nastavení registrů řadiče PCF8584. Vývojový diagram je na obrázku *Obr. 3.07*.



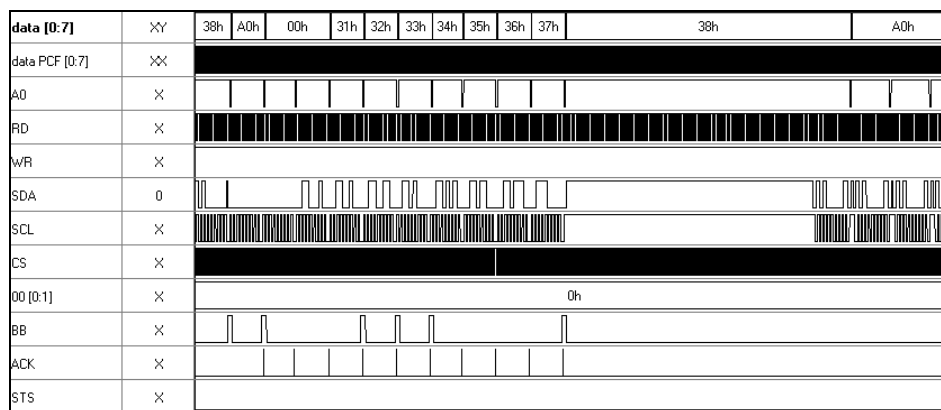
Obr. 3.07 Vývojový diagram programu testovacího pracoviště

Při programování byly vytvořeny funkce pro zápis do registru  $WriteSx(uchar A0value, uchar value)$ , kde  $A0value$  je hodnota úrovně na signálu A0 a jako druhý parametr vstupující do funkce je hodnota, kterou chceme do registru zapsat. Funkce  $ReadSx(uchar A0value)$  pro čtení z registru, její návratová hodnota je hodnota zapsaná v registru.

Po startu programu proběhne inicializace pro provoz, musí se zde nastavit registry řadiče PCF8584, do registru  $S0'$  nahrát hodnotu 00h, což přepne řadič do módu sledovače sběrnice I2C, pak se musí nastavit registr S2, tak aby obvod sledoval sběrnici I2C na frekvenci sběrnice 100kHz a obvod musí vědět, s jakou frekvencí pracuje externí oscilátor, tudíž 12MHz. Po té již běží program v nekonečné smyčce, kde sleduje hodnotu bitu PIN ve STATUS části registru S1, pokud je tento bit nulový, byla přijata data do registru S0 ze sběrnice I2C. Zároveň jsou zde naprogramovány čtení příznakového bitu BB, zda-li byla detekována podmínka START a jestli byl přijatý byte potvrzený bitem ACK. Výsledný program firmwaru je v příloze C.

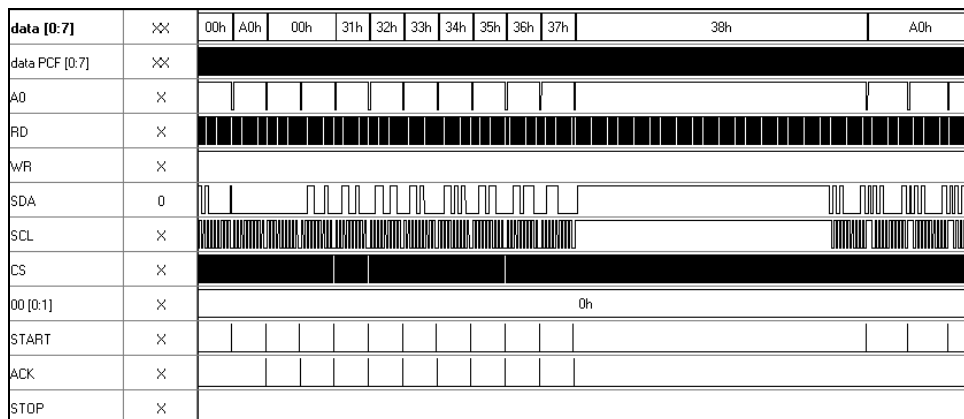
### 3.3 Výsledky testů

Při programování dle informací uvedených v datasheetu pro PCF8584, se obvod nechoval, jak bylo popsáno v jeho dokumentaci. Program se po prvním spuštění choval tak, jak je vyobrazeno na obrázku *Obr. 3.08*. Signály data PCF[0:7], A0, RD, WR a CS jsou na pinech pouzdra obvodu PCF8584, signály SDA a SCL značí komunikaci po sběrnici I2C a bity BB, ACK a STS jsou vnitřní příznakové bity STATUS registru obvodu PCF8584.



*Obr. 3.08 První spuštění – průběhy signálů*

Řešení programu tak, aby plnil zadaný účel nebylo jednoduché, muselo se hledat, na kterém příznaku jsou žádané průběhy. Chování programu, který byl takto sestaven je vyobrazeno na obrázku *Obr. 3.09*. Signály START, ACK a STOP jsou na vývodech brány mikroprocesoru.

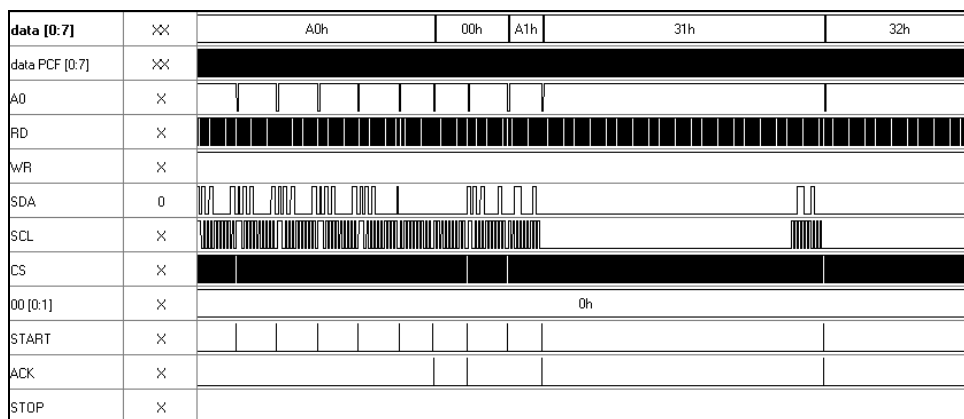


*Obr. 3.09 Průběh komunikace*

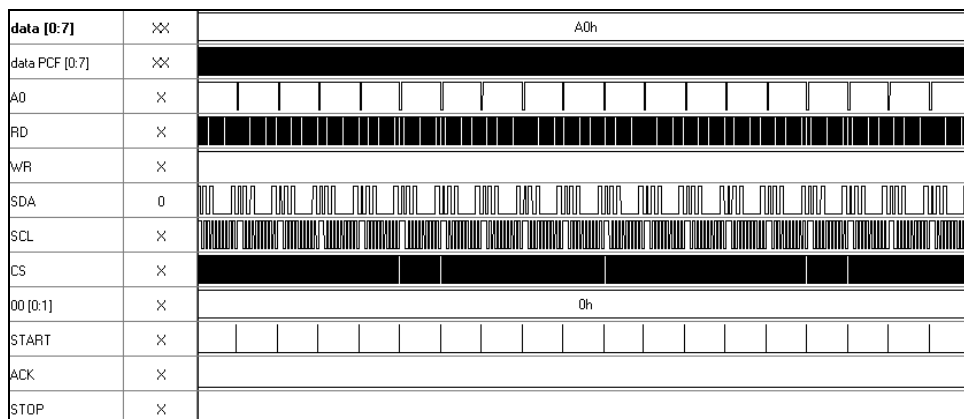


Bit ACK, příznakový bit – přijatý byte, není zobrazen na hodnotě bitu LRB ve STATUS registru obvodu PCF8584, ale je zobrazen na bitu AAS, který má indikovat příjem dat jako podřízené zařízení. Avšak nikdy nedošlo k detekci konce zprávy, tzn. STOP podmínka nebyla nikdy detekována, tudíž nelze rozeznat jednotlivé zprávy od sebe.

Chování obvodu dle přepracovaného programu je jinak správné, jak je ukázáno na obrázku *Obr. 3.10*, kde je zobrazeno zpětné vyčítání dat z EEPROM a na obrázku *Obr. 3.11*, kde je průběh s odpojenou pamětí EEPROM.



*Obr. 3.10 Průběh vyčítání dat z EEPROM*



*Obr. 3.11 Průběh komunikace bez EEPROM*

## Závěr

Účelem práce bylo navrhnout zařízení pro sledování sběrnice I2C jako pasivního prvku neovlivňujícího dění na sběrnici, s tím, že platná data na sběrnici bude schopno detekovat a zobrazit na počítači.

Vybíralo se z několika vhodných koncepcí možných provedení analyzátoru: provedení s externím řadičem, provedení s integrovaným řadičem, provedení bez mikrokontroléru a provedení bez řadiče I2C. Z těchto možností provedení bylo vybráno provedení s externím řadičem a toto provedení bylo vypracováno do schématu zapojení. Aby bylo možné ověřit správnost zapojení, byl sestaven funkční prototyp analyzátoru a sestaven program pro komunikaci mikroprocesoru s řadičem sběrnice I2C. Funkčnost prototypu byla vyzkoušena na testovacím pracovišti.

Jak se ukázalo při testech na testovacím pracovišti, navržené zařízení svou činností sběrnici neovlivňuje, je schopno přijímat data ze sběrnice, dále je zařízení schopno detekovat podmínku START a potvrzovací bity ACK. Zařízení ale nebylo schopné detekovat STOP podmínku – konec zprávy na sběrnici I2C, z toho plyne, že by nebylo možné rozeznat od sebe jednotlivé zprávy na sběrnici I2C. Data by bylo možné na počítači sice zobrazit, jednalo by se však pouze o shluk nerozlišitelných dat. Věřím, že pokud by na tuto práci bylo více času, bylo by možné problém detekce podmínky STOP řešit přímo s výrobcem firmou NXP. Po té by již zařízení bylo zcela funkční a stačilo by pouze doprogramovat komunikaci s obvodem FT245BM pro odesílání dat do počítače.

## Seznam použité literatury

- [1] *NXP Semiconductors* [online]. 2006-2009 [cit. 2009-07-10]. Dostupný z WWW: <<http://www.nxp.com/#/homepage>>.
- [2] I2C-bus specification and user manual. *UM10204* [online]. 2007, no. 3 [cit. 2009-07-10], s. 1-50. Dostupný z WWW: <<http://www.standardics.nxp.com/support/documents/i2c/pdf/i2c.bus.specification.pdf>>.
- [3] DUDÁČEK, K. Sériová rozhraní SPI, Microwire, I2C a CAN. *Sériová rozhraní SPI, Microwire, I2C a CAN*. 2002, 1, s. 5-10. Dostupný z WWW: <[http://home.zcu.cz/~dudacek/NMS/Seriova\\_rozhrani.pdf](http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf)>.
- [4] PCF8584 : I2C-bus controller. *Product specification* [online]. 1997, vol. 1, no. 1 [cit. 2009-07-10], s. 1-40. Dostupný z WWW: <[http://www.nxp.com/acrobat\\_download/datasheets/PCF8584\\_4.pdf](http://www.nxp.com/acrobat_download/datasheets/PCF8584_4.pdf)>.
- [5] *ATMEL Corporation* [online]. 2009 [cit. 2009-07-10]. Dostupný z WWW: <<http://www.atmel.com/>>.
- [6] *FT2232C* [online]. 2009 [cit. 2009-07-20]. Dostupný z WWW: <<http://www.ftdichip.com/Products/FT2232C.htm>>.
- [7] AN425 : Interfacing the PCF8584 I2C-bus controller. *Application note* [online]. 1994, vol. 1, no. 1 [cit. 2009-07-20], s. 1-20. Dostupný z WWW: <<http://www.standardics.nxp.com/support/documents/i2c/pdf/an425.pdf>>.
- [8] KARAS, Ondřej. *Květákov.net - články - avr* [online]. 2007 [cit. 2009-07-22]. Dostupný z WWW: <<http://www.kvetakov.net/clanky/avr/>>.
- [9] FT245BM Designers Guide. *Version 2.0* [online]. 2003, vol. 1, no. 2 [cit. 2009-07-20], s. 1-14. Dostupný z WWW: <[http://www.ftdichip.com/Documents/AppNotes/DG245\\_20.pdf](http://www.ftdichip.com/Documents/AppNotes/DG245_20.pdf)>.
- [10] Váňa, Vladimír. *Atmel AVR programování v jazyce C : Popis a práce ve vývojovém prostředí CodeVisionAVR C*. 1. vyd. Ostrava : BEN - technická literatura, 2003. 216 s. ISBN 80-7300-102-0.

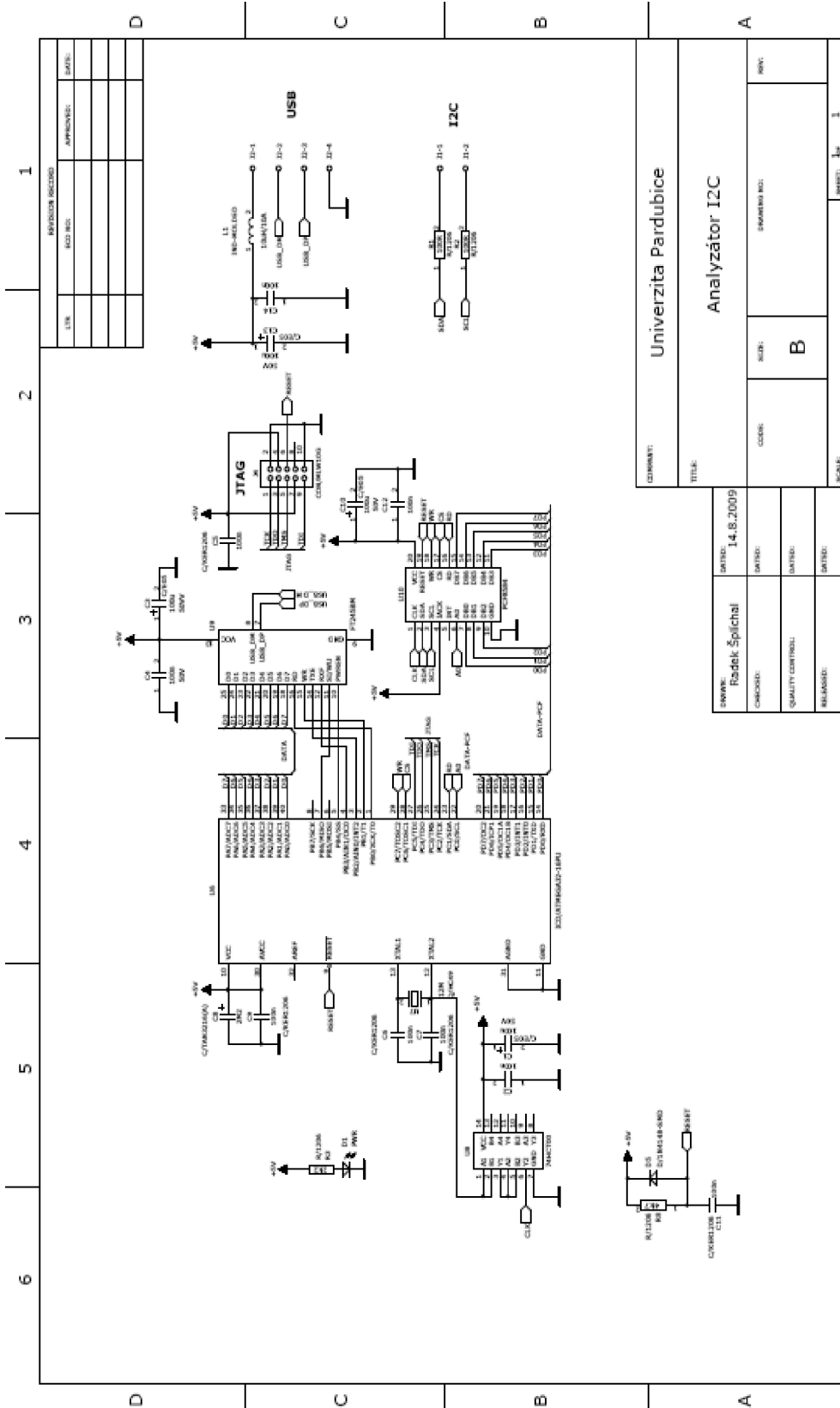
## Seznam obrázků

<i>Obr. 1.01 Celková koncepce sběrnice I2C [3]</i>	12
<i>Obr. 1.02 Podmínky START a STOP [3]</i>	12
<i>Obr. 1.03 Časování signálu SCL [3]</i>	13
<i>Obr. 1.04 Synchronizace hodin SCL [3]</i>	13
<i>Obr. 1.05 Formát rámců na sběrnici I2C [3]</i>	14
<i>Obr. 1.06 Arbitrace na I2C. Master1 ztrácí právo na sběrnici [3]</i>	15
<i>Obr. 1.07 Potvrzování příjmu bitem ACK [3]</i>	16
<i>Obr. 1.08 Zápis a čtení paměti Philips PCF8570 [3]</i>	17
<i>Obr. 2.01 Blokové schéma připojení I2C analyzátoru</i>	18
<i>Obr. 2.02 Princip připojení k I2C</i>	19
<i>Obr. 2.03 Princip možnosti připojení I2C</i>	20
<i>Obr. 2.04 Připojení I2C sběrnice k PC bez mikroprocesoru[6]</i>	21
<i>Obr. 2.05 Připojení I2C sběrnice bez řadiče I2C</i>	21
<i>Obr. 2.06 Blokové schéma obvodu PCF8584 [4]</i>	22
<i>Obr. 2.07 Registr S1 obvodu PCF8584 [4]</i>	23
<i>Obr. 2.08 Požadavky na oscilátor [4]</i>	25
<i>Obr. 2.09 Požadavky na časování [4]</i>	25
<i>Obr. 3.01 Blokové schéma testovacího pracoviště</i>	27
<i>Obr. 3.02 Fotografie realizace analyzátoru</i>	28
<i>Obr. 3.03 Fotografie připojení na krystal</i>	28
<i>Obr. 3.04 Fotografie testovacího pracoviště</i>	29
<i>Obr. 3.05 Blokové schéma testovacího obvodu</i>	29
<i>Obr. 3.06 Fotografie testovacího obvodu</i>	30
<i>Obr. 3.07 Vývojový diagram programu testovacího pracoviště</i>	31
<i>Obr. 3.08 První spuštění – průběhy signálů</i>	32
<i>Obr. 3.09 Průběh komunikace</i>	32
<i>Obr. 3.10 Průběh vyčítání dat z EEPROM</i>	33
<i>Obr. 3.11 Průběh komunikace bez EEPROM</i>	33

## Seznam tabulek

<i>Tab. 1.01 Tabulka adres udaných výrobcem [1]</i>	16
<i>Tab. 2.01 Tabulka nastavení bitů ES1 a ES2 [4]</i>	23

# Příloha A1



## Příloha A2

Počet	Komponenta	Hodnota/Typ
1	Integrovaný obvod	ATMEGA16
1	Integrovaný obvod	FTDI245RL
1	Integrovaný obvod	74HCT00
1	Integrovaný obvod	PCF8584P
1	Konektor	USB
1	Konektor	I2C
1	Konektor	JTAG
1	Dioda	1N4148
1	Led dioda	L-13GD
1	Cívka	10mH/10A
1	Krystal	12MHz
8	Kondenzátor keramický	100nF
4	Kondenzátor elektrolyt.	100uF/50V
1	Kondenzátor elektrolyt.	2M2
2	Rezistor	100 $\Omega$
1	Rezistor	4k7 $\Omega$
1	Rezistor	2k2 $\Omega$

## Příloha B1

```
#define F_CPU 14745600UL
#include <avr/io.h>
#include <string.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "lcd_lib.h"
#include "twi_lib.h"

#define Dev24C32 0xA0

unsigned char pole[8]={'1','2','3','4','5','6','7','8'};

int main(void)
{
    //inicializace
    unsigned char i=0;
    LCD_Init();
    i2c_init();
    LCD_Position(0,0);
    LCD_WriteCString("TEST I2C");

    //odeslani dat do eeprom
    i2c_start_wait(Dev24C32+I2C_WRITE);
    i2c_write(0x00);
    i2c_write(0x00);

    for (i=0; i<8; i++)
        i2c_write(pole[i]);

    i2c_stop();

    //zpetne vyceni dat z eeprom
    LCD_Position(1,0);

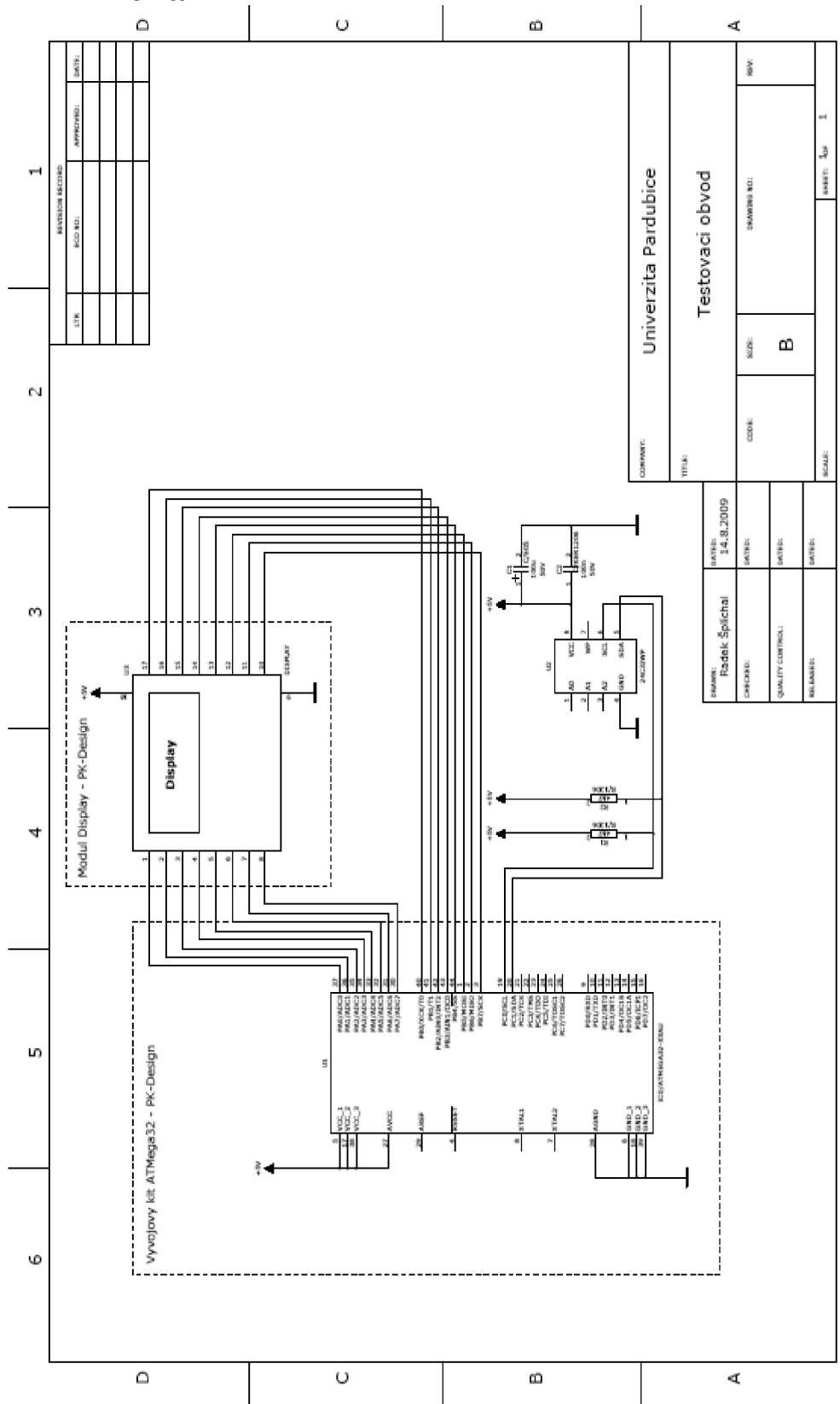
    i2c_start_wait(Dev24C32+I2C_WRITE);
    i2c_write(0x00);
    i2c_write(0x00);
    i2c_rep_start(Dev24C32+I2C_READ);

    for (i=0; i<7; i++)
        LCD_WriteData(i2c_readAck());
    LCD_WriteData(i2c_readNak());

    i2c_stop();

    while(1);
}
```

# Příloha B2



1	2	3	4	5	6	BROUČEK REČENÉ	
						SK. NO.:	SK. NO.:
						SK. NO.:	SK. NO.:
						SK. NO.:	SK. NO.:
						SK. NO.:	SK. NO.:
						SK. NO.:	SK. NO.:

COMPANY: **Univerzita Pardubice**

TITLE: **Testovací obvod**

CODE:	SCALE: <b>B</b>	REVISION: <b>1</b>
CHECKED:	DATE:	REVISION:
QUALITY CONTROL:	DATE:	REVISION:
RELEASED:	DATE:	REVISION:

DESIGNED:	DATE:
CHECKED:	DATE:
QUALITY CONTROL:	DATE:
RELEASED:	DATE:



## Příloha C

```
#include "stddef.h"
#include <util/delay.h>
#include <avr/interrupt.h>

// makra
//brana D je DB
#define WR PC7
#define RD PC1
#define CS PC6
#define A0 PC0
#define BB PB0
#define AAS PB2
#define STS PB5

#define mCS_1      PORTC |= (1 << CS)
#define mCS_0      PORTC &= ~(1 << CS)
#define mA0_1      PORTC |= (1 << A0)
#define mA0_0      PORTC &= ~(1 << A0)
#define mWR_1      PORTC |= (1 << WR)
#define mWR_0      PORTC &= ~(1 << WR)
#define mRD_1      PORTC |= (1 << RD)
#define mRD_0      PORTC &= ~(1 << RD)
#define mBB_1      PORTB |= (1 << BB)
#define mBB_0      PORTB &= ~(1 << BB)
#define mACK_1     PORTB |= (1 << PB1)
#define mACK_0     PORTB &= ~(1 << PB1)
#define mSTS_1     PORTB |= (1 << PB2)
#define mSTS_0     PORTB &= ~(1 << PB2)

#define mSetBusRd  DDRD = 0x00
#define mSetBusWr  DDRD = 0xFF

void WriteSx(uchar A0value, uchar value);
uchar ReadSx(uchar A0value);
uchar PocetPrijatychBytu = 0, vycet;

// hlavni program
int main(void)
{
    // init
    DDRD = 0xFF;
    DDRA = 0xFF;
    DDRB = 0xFF;
    DDRC = (1 << WR) | (1 << RD) | (1 << A0) | (1 << CS);
    mSetBusRd;
    mCS_1; mWR_1; mRD_1; mA0_1;
```

```

_delay_us(3);
WriteSx(1, 0x80);
WriteSx(0, 0x00); // adresa = 0x00;
WriteSx(1, 0xA0); // dalsi zapis bude do S2
WriteSx(0, 0x1C); // SCL = 90kHz pri 12MHz
WriteSx(1, 0xC1);
for (;;)
{
    vycet = ReadSx(1); // cteni S1

    //pokud je PIN v 0, prichodi byte je prijat
    if ((vycet & (1 << 7)) == 0)
    {
        // detekce START podminky
        if ((vycet & (1 << BB)) == 0)
        {
            mBB_1;
            asm volatile("nop");
            mBB_0;
        }
        //detekce ACK
        if ((vycet & (1 << AAS)) == 0)
        {
            mACK_1;
            asm volatile("nop");
            mACK_0;
        }
        //vystaveni prijatych dat
        PORTA = ReadSx(0); // cteni dat z S0
        PocetPrijatychBytu++;
    }

    //ODESLANI DAT PO USB
}
return 0;
}

```

```

void WriteSx(uchar A0value, uchar value)
{
    mSetBusWr;
    asm volatile("nop");
    PORTD = value;
    if (A0value)
        mA0_1;
    else
        mA0_0;
    asm volatile("nop");
    mCS_0;
    asm volatile("nop");
}

```

```

    mWR_0;
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    mWR_1;
    asm volatile("nop");
    mCS_1;
    asm volatile("nop");
    mSetBusRd;
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
}

```

```

uchar ReadSx(uchar A0value)
{
    uchar temp;
    if (A0value)
        mA0_1;
    else
        mA0_0;
    asm volatile("nop");
    mCS_0;
    asm volatile("nop");
    mRD_0;
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    temp = PIND;
    asm volatile("nop");
    mRD_1;
    asm volatile("nop");
    mCS_1;
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
    asm volatile("nop");
}

```

```
asm volatile("nop");  
asm volatile("nop");  
asm volatile("nop");  
asm volatile("nop");  
return temp;  
}
```