ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická

Bakalářská práce

Realizace řadiče sběrnice CAN

Praha 2006 Lukáš Kováčik

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne 30.6.2006 …….

podpis

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce Ing. Pavlu Píšovi za konkrétní rady a připomínky, které mi pomáhaly při její tvorbě. Dále děkuji firmě MEFI, s.r.o. za zapůjčení hardware.

České vysoké učení technické v Praze, fakulta elektrotechnická

Katedra řídicí techniky Školní rok:2005/2006

Zadání bakalářské práce

Student: Lukáš Kováčik

Obor: Kybernetika a měření

Název tématu: Realizace řadiče sběrnice CAN v jazyce VHDL pro programovatelné obvody

Zásady pro vypracování:

1. Seznamte se s návrhem logiky pro obvody FPGA v jazyce VHDL a prostudujte normy popisující fyzickou a linkovou vrstvu komunikace CAN 2.0B.
2. Proveďte návrh a syntézu řadiče. Řadič doplňte dalšími bloky pro realizaci jednoduchých distribuovaných digitálních vstupů a výstupů.
3. Dále se seznamte s navrženou rozšiřující deskou s FPGA obvodem pro průmyslové PC na bázi standardu PC-104 a navrhněte logiku pro mapování Vámi navrženého řadiče do paměťového prostoru PC.
4. K návrhu je možné využít znalostí a výsledků z většího množství podobných projektů a diplomových prací řešených na katedře řídicí techniky.

Seznam odborné literatury:

J. O. Hamblen and M. D. Furman, Rapid Prototyping of Digital Systems, Atlanta, Georia, 2001

Vedoucí bakalářské práce: Ing. Pavel Píša

Datum zadání bakalářské práce: zimní semestr 2005/06

Termín odevzdání bakalářské práce: 30. 6. 2006

Prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta elektrotechnická -4-

v. z.

Prof. Ing. Zbyněk Škvor, CSc.
děkan

V Praze, dne 1. 2. 2006

Anotace

Cílem této bakalářské práce je návrh řadiče CAN, který používá programovatelné hradlové pole a obzvlášť software zacházející s tímto řadičem. Hardwarové provedení je založeno na existujícím projektu a byly na něm udělány jen malé změny. Na začátku mojí práce jsou uvedeny základní informace o datových sběrnicích CAN. Dále je základní popis řadiče CAN a jeho registrů. Práce pokračuje popisem vytvořeného programu, který tvoří hlavní část mojí práce. Na konci práce je testování hardwaru a softwaru dohromady.

Annotation

The aim of this bachelor thesis is design of CAN controller using field programmable gate array and particularly software handling this controller. The hardware design is based on existing project and only small changes was made on it. At the beginning of my thesis there is some common information about CAN-BUS. Next there is basic information about CAN controller implementation, especially register description. The thesis continues by description of the program written as a main part of my work. At the end of thesis is tested hardware and software, together.

Obsah

1 Rozbor zadání ... 1
2 Sběrnice CAN-BUS ... 1
2.1 Základní prvky sběrnice CAN-BUS ... 1
2.2 Stručný popis řízení komunikace na sběrnici CAN-BUS ... 2
2.3 Základní typy zpráv CAN ... 3
2.3.1 Datová zpráva ... 3
2.3.2 Žádost o data (Remote Frame) ... 4
2.3.3 Chybová zpráva (Error Frame) ... 4
2.3.4 Zpráva o přetížení (Overload Frame) ... 5
3 Řadič CAN a řízení přes sériovou linku ... 5
3.1 Řadič CAN ... 5
3.2 Umístění registrů řadiče CAN ... 6
3.3 Popis registrů řadiče CAN ... 7
3.3.1 Adresa 0 - Registr režimu (MODE REGISTER) ... 7
3.3.2 Adresa 1 - Registr příkazů (COMMAND REGISTER) ... 7
3.3.3 Adresa 2 - Stavový registr (STATUS REGISTER) ... 8
3.3.4 Adresa 6 - Registr časování 0 (Bus Timing Register 0) ... 9
3.3.5 Adresa 7 - Registr časování 1 (Bus Timing Register 1) ... 9
3.3.6 Adresa 16 až 28 - Vysílací zásobník ... 9
3.4 Ovládání řadiče CAN po sériové lince ... 11
4 Realizace CAN řadiče ... 11
4.1 Vnitřní schéma zapojení modulů řadiče s RS232 ... 12
4.2 FIFO modulu řadiče CAN ... 14
5 Programové prostředky pro ovládání a diagnostiku ... 15
5.1 Dělení vrstev programu ... 15
5.2 Popis rozhraní jednotlivých vrstev ... 15
5.2.1 Vrstva pro práci s pakety CAN ... 15
5.2.2 Vrstva pro práci s registry ... 16
5.2.3 Fyzická vrstva RS232 ... 16
5.3 Popis jednotlivých vrstev programu ... 17
5.3.1 Práce s pakety CAN ... 17
5.3.2 Práce s registry řadiče CAN ... 17
5.3.3 Fyzická komunikace po sériové lince ... 17
6 Závěr ... 17
7 Literatura ... 18
8 Obsah přiloženého CD ... 19

1 Rozbor zadání

Tato práce se zabývá využitím řadiče sběrnice CAN implementovaného programovatelným obvodem (FPGA). Základem práce bylo seznámení se s existující implementací, její oživení, úprava a vytvoření software pro otestování funkčnosti a využití řadiče pro zasílání a příjem zpráv.

Vytvořená aplikace má být schopna sledovat komunikaci na průmyslové sběrnici CAN-BUS (zobrazovat příchozí pakety, odesílat pakety, zapisovat konkrétní hodnoty do registru řadiče a načíst aktuální hodnoty registrů).

2 Sběrnice CAN-BUS

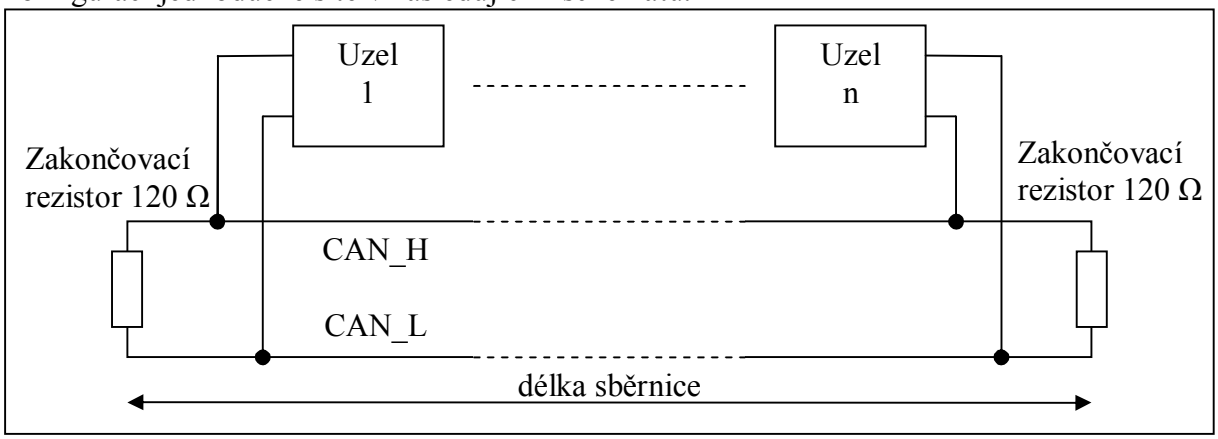
V únoru 1986, představila firma Robert Bosch GmbH na kongresu Společnosti automobilových inženýrů SAE (Society of Automotive Engineers) v Detroitu novou sběrnici CAN. V současné době se ze sběrnice CAN stala jedna z nejrozšířenějších sběrnic, která se nepoužívá zdaleka jen v automobilovém průmyslu, pro nějž byla původně určena.

Označení CAN-BUS vzniklo spojením termínu CAN, který je zkratkou anglického výrazu Controller Area Network a ve volném překladu znamená řídicí síť, a BUS z výrazu Bussystem, pro který má čeština výraz sběrnice, v tomto případě datová.

Výhodou CAN-BUSu je snížení počtu spojení a vodičů při dodržení stejné intenzity komunikace.

2.1 Základní prvky sběrnice CAN-BUS

Jednotlivé komunikační uzly musí mít mezi sebou fyzické spojení. CAN sice nepředepisuje ani úrovně ani fyzikální média, ale další normy přesně specifikují napěťové úrovně a proudy pro CAN High-Speed a Low-Speed včetně zakončení, konektorů atd. Je důležité věnovat pozornost impedančnímu přizpůsobení a ochraně na vzářené i vyzářené signály. V podstatě je počet uzlů pro komunikaci neomezený, prakticky je limitován jen možnostmi jednotlivých uzlů a výkonovým zatížením sítě. Z principu protokolu není nutno upravovat nadřazenost a prioritu jednotlivých uzlů. Jak poznáme později, jsou tyto informace kódovány v přenášené zprávě. Výhodou je možnost modifikace vystavěné sítě přidáváním nebo ubíráním počtu členů sítě. Využití principu je například v odstavení zdroje chybných zpráv nebo naopak přidání uzlu pro diagnostiku systému. V principu je možno představit konfiguraci jednoduché sítě v následujícím schématu.



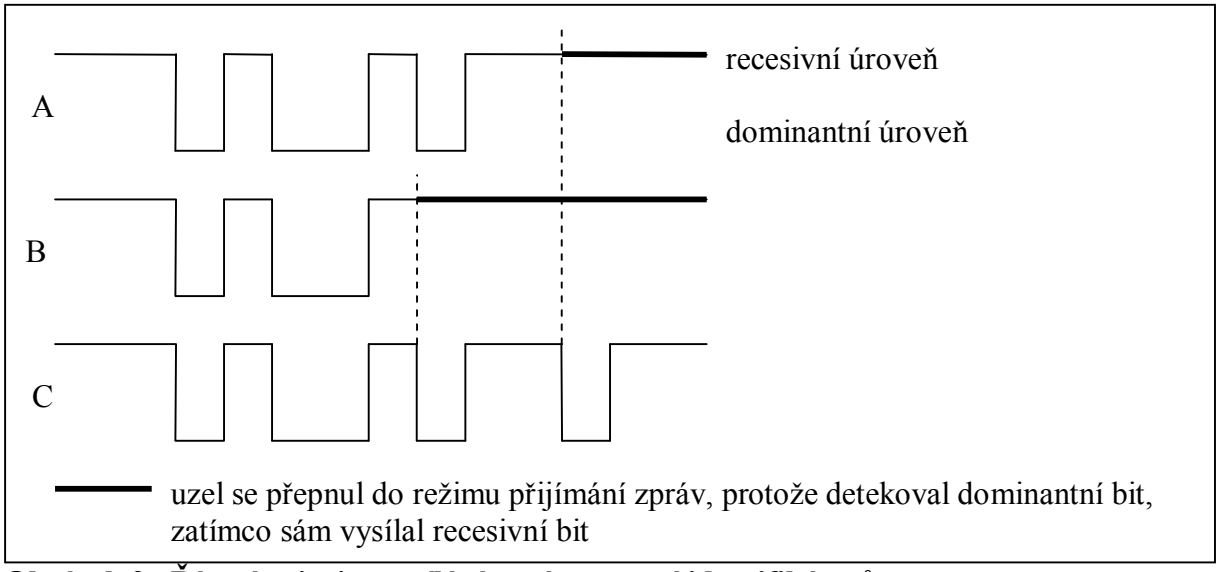
Obrázek 1 – Struktura sběrnice CAN podle ISO 11898

2.2 Stručný popis řízení komunikace na sběrnici CAN-BUS

Od samého počátku byla sběrnice CAN navržena jako síť s liniovou topologií pracující v režimu multimaster. To znamená, že žádný z účastnických uzlů není pevně určen jako nadřízená stanice – master.

Stanice, která chce ostatním podat zprávu, začne vysílat. V tom okamžiku se stává nadřízenou jednotkou – masterem – a ostatní uzly musí počkat, až je přenos dokončen a linka uvolněna. Není určeno, který uzel má vyslanou zprávu přijmout. Zpráva má identifikátor, z nějž ostatní uzly poznají, co zpráva obsahuje, a podle toho ji přijmou nebo ignorují. O přijetí zprávy tedy nerozhoduje adresa odesílatele ani příjemce, ale její obsah.

Je ovšem nutné vytvořit mechanismus, který zajistí, aby byly důležité zprávy doručeny včas a aby se o vysílání nepokoušely současně dva uzly. CAN to řeší takto: stane-li se, že se dva uzly pokoušejí vysílat současně, dostane přednost ten s vyšší prioritou. Priorita je zakódována v identifikátoru. V praxi probíhá celý postup tak, že uzly, které chtějí vysílat, zjistí, zda je volná linka, a v případě, že je, začnou vysílat svůj identifikátor. Přitom kontrolují po jednotlivých bitech shodu vysílané a přijímané zprávy.



Obrázek 2– Řízení priority vysílání zpráv pomocí identifikátorů

Dokud jsou identifikátory shodné, nic se neděje. Teprve v okamžiku, kdy jeden účastnický uzel zjistí, že se v identifikátoru druhého uzlu objevil dominantní bit, zatímco on má ve svém identifikátoru ve stejném okamžiku bit recesivní, usoudí, že jeho zpráva má nižší prioritu, stáhne se a uvolní sběrnici. Druhý účastník dokončí vysílání a první pokus opakuje, jakmile se linka uvolní. Takové metodě se říká nedestruktivní řízení komunikace a její důležitou předností je to, že se vysílání nezdržuje žádným opakováním úvodní sekvence zprávy.

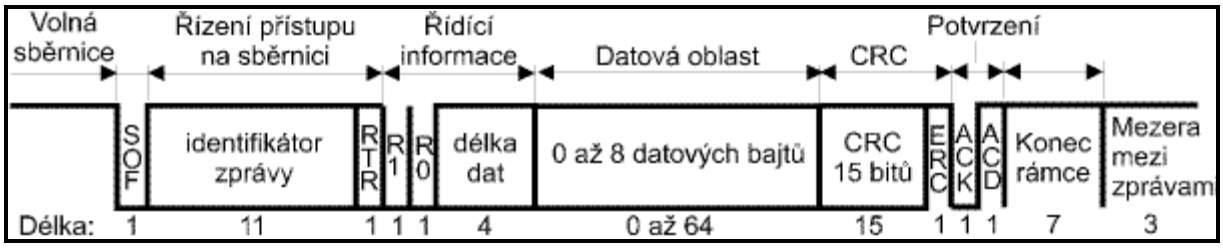
Podmínkou je, že žádné dva identifikátory nesmí být stejné. Postup, který to zabezpečuje, je součástí specifikace CAN.

2.3 Základní typy zpráv CAN

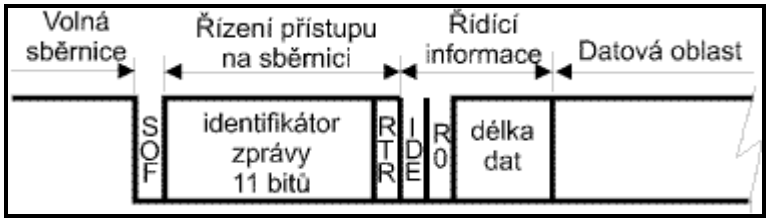
Specifikace protokolu CAN definuje čtyři typy zpráv.

2.3.1 Datová zpráva

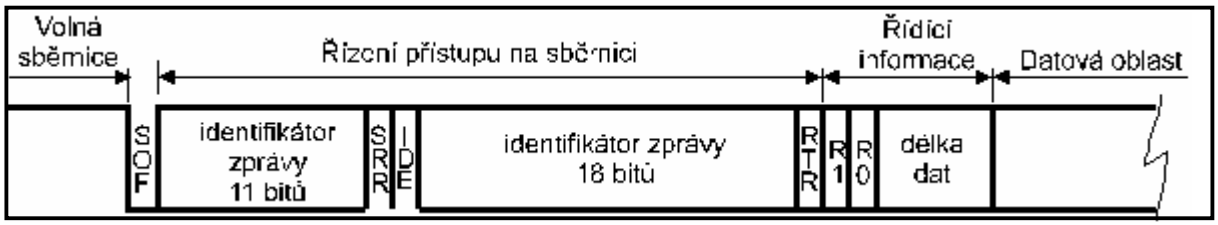
Protokol CAN používá dva typy datových zpráv. První typ je definován specifikací 2.0A a je v literatuře označován jako standardní formát zprávy (Standard Frame), zatímco specifikace 2.0B definuje navíc tzv. rozšířený formát zprávy (Extended Frame). Jediný podstatný rozdíl mezi oběma formáty je v délce identifikátoru zprávy, která je 11 bitů pro standardní formát a 29 bitů pro rozšířený formát. Oba dva typy zpráv mohou být používány na jedné sběrnici, pokud je použitým řadičem podporován protokol 2.0B.



Obrázek 3 - Datová zpráva podle specifikace CAN 2.0A



Obrázek 4 - Začátek datové zprávy (standardní formát) podle specifikace 2.0B



Obrázek 5 - Začátek datové zprávy (rozšířený formát) podle specifikace 2.0B

<u>Význam jednotlivých částí datové zprávy</u>
SOF (Start Of Frame) - začátek zprávy, 1 dominantní bit

Řízení přístupu na sběrnici (Arbitration Field) - určení priority zprávy
Identifikátor zprávy - 11 bitů, udává význam přenášené zprávy
RTR (Remote Request) - 1 bit, příznak udává, zda se jedná o datovou zprávu nebo o žádost o vyslání dat. V datové zprávě musí být tento bit dominant, v žádosti o data recessive.

Řídící informace (Control Field)
R0, **R1** - rezervované bity
Délka dat - 4 bity, počet přenášených datových bajtů. Povolené hodnoty jsou 0 až 8.
Datová oblast (Data Field) - datové bajty zprávy. Maximálně 8 bajtů je vysláno od MSB.

CRC (CRC Field) - 16 bitů, zabezpečovací CRC kód
- **CRC** kód - 15 bitů
- **ERC** (CRC oddělovač) - 1 bit recessive

Potvrzení (ACK Field) - 2 bity
- **ACK** (bit potvrzení) - 1 bit
- **ACD** (oddělovač potvrzení) - 1 bit recessive

Konec zprávy (End Of Frame) - 7 bitů recessive

Mezera mezi zprávami (Interframe Space) - 3 bity recessive, odděluje dvě zprávy

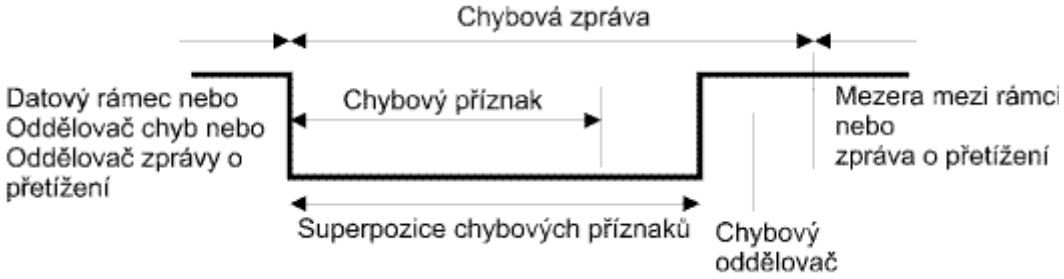
Navíc pro rozšířený rámec (Extended Frame)
SRR (Substitute Remote Request), který má v rozšířeném formátu vždy hodnotu recessive.
IDE (Identifier Extended) má vždy recessive hodnotu.
RTR je přesunut za konec druhé části identifikátoru.

2.3.2 Žádost o data (Remote Frame)

Formát žádosti o data je podobný jako formát datové zprávy. Pouze je zde RTR bit (pole řízení přístupu na sběrnici) nastaven do úrovně recessive a chybí datová oblast. Pokud nějaký uzel žádá o zaslání dat, nastaví takový identifikátor zprávy, jako má datová zpráva, jejíž zaslání požaduje. Tím je zajištěno, že pokud ve stejném okamžiku jeden uzel žádá o zaslání dat a jiný data se stejným identifikátorem vysílá, přednost v přístupu na sběrnici získá uzel vysílající datovou zprávu, neboť úroveň RTR bitu datové zprávy je dominant a tudíž má tato zpráva vyšší prioritu.

2.3.3 Chybová zpráva (Error Frame)

Chybová zpráva slouží k signalizaci chyb na sběrnici CAN. Jakmile libovolný uzel na sběrnici detekuje v přenášené zprávě chybu (chyba bitu, chyba CRC, chyba vkládání bitů, chyba rámce), vygeneruje ihned na sběrnici chybový rámec. Podle toho, v jakém stavu pro hlášení chyb se uzel, který zjistil chybu, právě nachází, generuje na sběrnici buď aktivní (šest bitů dominant) nebo pasivní (šest bitů recessive) příznak chyby. Při generování aktivního příznaku chyby je přenášená zpráva poškozena (vzhledem k porušení pravidla na vkládání bitů), a tedy i ostatní uzly začnou vysílat chybové zprávy. Hlášení chyb je pak indikováno superpozicí všech chybových příznaků, které vysílají jednotlivé uzly. Délka tohoto úseku může být minimálně 6 a maximálně 12 bitů.

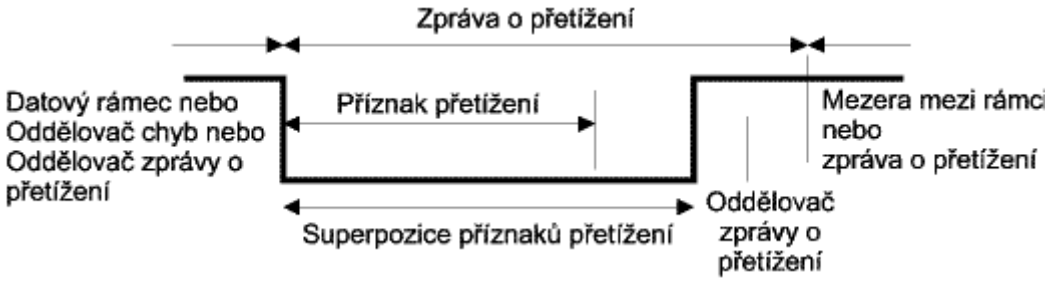


Obrázek 6 - Chybová zpráva

Po vyslání chybového příznaku vysílá každá stanice na sběrnici bity recessive. Zároveň detekuje stav sběrnice a jakmile najde první bit na sběrnici ve stavu recessive, vysílá se dalších sedm bitů recessive, které plní funkci oddělovače chyb (ukončení chybové zprávy).

2.3.4 Zpráva o přetížení (Overload Frame)

Zpráva o přetížení slouží k oddálení vyslání další datové zprávy nebo žádosti o data. Zpravidla tento způsob využívají zařízení, která nejsou schopna kvůli svému vytížení přijímat a zpracovávat další zprávy. Struktura zprávy je podobná zprávě o chybě, ale její vysílání může být zahájeno po konci zprávy (End of Frame), oddělovače chyb nebo předcházejícího oddělovače zpráv přetížení.



Obrázek 7 – Zpráva o přetížení

Zpráva o přetížení je složena z příznaku přetížení (šest bitů dominant) a případné superpozice všech příznaků přetížení, pokud jsou generovány více uzly současně. Za příznaky přetížení následuje dalších sedm bitů recessive, které tvoří oddělovač zprávy o přetížení.

3 Řadič CAN a řízení přes sériovou linku

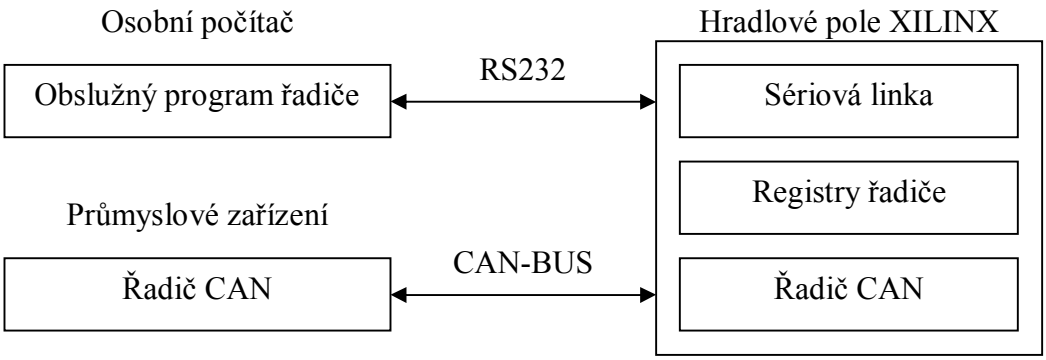
3.1 Řadič CAN

CAN controllery (řadiče) sběrnice CAN je možno rozdělit do dvou základních skupin podle principu jejich funkce a to nezávisle na tom, zda se jedná o samostatné obvody, jádra pro hradlová pole nebo periferie různých procesorů. První skupina obvodů obvykle obsahuje několik bufferů, nebo jeden FIFO (SJA1000) buffer, kdy buffer(y) může(mohou) obsahovat několik zpráv se stejným identifikátorem. Druhá skupina obsahuje obvykle větší množství bufferů, z nichž každý je určen pro jednu konkrétní zprávu nebo skupinu zpráv v případě implementace lokálních masek. Samozřejmě hranice mezi těmito skupinami není nijak ostrá. U mnoha obvodů se dá diskutovat o tom, do které skupiny by bylo možno je zařadit.

První skupina controllerů je vhodná zejména pro různé karty pro PC, kdy potřebujeme přijímat obvykle všechny zprávy bez ohledu na identifikátor. Druhá skupina je pak určena pro cílová zařízení jako jsou různé IO karty, řídící jednotky pohonů apod.

Řadič CAN zprostředkovává kompletní řízení sběrnice CAN-BUS:
- Datová zpráva (posílání datový paketů)
- Žádost o data (Remote Frame)
- Chybová zpráva (Error Frame)
- Zpráva o přetížení (Overload Frame)

Řadič sběrnice CAN v této práci je ovládán pomocí sériové linky RS232.



Obrázek 8 – Celkové schéma zapojení s řadičem CAN-BUS

Po sériové lince lze u řadiče číst a nastavovat jednotlivé registry řadiče.

3.2 Umístění registrů řadiče CAN

Adresa reg.		Operační režim		Servisní režim	
Dec	**Hex**	**Čtení**	**Zápis**	**Čtení**	**Zápis**
0	0	Režim	Režim	Režim	Režim
1	1	00h	Příkaz	00h	Příkaz
2	2	Stav	-	Stav	-
3	3	00h - rezerva	-	00h - rezerva	-
4	4	00h - rezerva	-	00h - rezerva	-
5	5	00h - rezerva	-	00h - rezerva	-
6	6	Časovač 0	-	Časovač 0	Časovač 0
7	7	Časovač 1	-	Časovač 1	Časovač 1
8	8	00h - rezerva	-	00h - rezerva	-
9	9	00h - rezerva	-	00h - rezerva	-
10	A	00h - rezerva	-	00h - rezerva	-
11	B	00h - rezerva	-	00h - rezerva	-
12	C	00h - rezerva	-	00h - rezerva	-
13	D	00h - rezerva	-	00h - rezerva	-
14	E	00h - rezerva	-	00h - rezerva	-
15	F	00h - rezerva	-	00h - rezerva	-
16	10	RX frame info.	TX frame info.	00h - rezerva	-
17	11	RX Ident. 1	TX Ident. 1	00h - rezerva	-
18	12	RX Ident. 2	TX Ident. 2	00h - rezerva	-
19	13	RX Ident. 3	TX Ident. 3	00h - rezerva	-

20	14	RX Ident. 4	TX Ident. 4	00h - rezerva	-
21	15	RX data 1	TX data 1	00h - rezerva	-
22	16	RX data 2	TX data 2	00h - rezerva	-
23	17	RX data 3	TX data 3	00h - rezerva	-
24	18	RX data 4	TX data 4	00h - rezerva	-
25	19	RX data 5	TX data 5	00h - rezerva	-
26	1A	RX data 6	TX data 6	00h - rezerva	-
27	1B	RX data 7	TX data 7	00h - rezerva	-
28	1C	RX data 8	TX data 8	00h - rezerva	-
29	1D	00h - rezerva	-	00h - rezerva	-
30	1E	00h - rezerva	-	00h - rezerva	-
31	1F	00h - rezerva	-	00h - rezerva	-

3.3 Popis registrů řadiče CAN

3.3.1 Adresa 0 - Registr režimu (MODE REGISTER)

Slouží ke změně chování řadiče, jednotlivé bity mohou být nastaveny nebo vynulovány prostřednictvím CPU, pro kterou se registr jeví jako paměť umožňující čtení i zápis. Rezervované bity jsou čteny jako logická nula.

Interpretace bitů registru režimu (MOD), CAN adresa 0

Bit	Symbol	Jméno	Hodnota	Význam
MOD.7 –MOD.1	-	Rezerva	-	-
MOD.0	RM	Servisní režim	1	Servisní režim
			0	Operační režim

Po hardwarovém resetu je hodnota bitu RM vždy 1 (Servisní režim), po nastavení komunikačních parametrů uvede řadič do operačního režimu nadřízený mikroprocesor změnou hodnoty bitu RM do nuly. Při příchodu požadavku na změnu režimu na operační čeká CAN řadič na výskyt jedenácti po sobě jdoucích recesivních bitů na CAN sběrnici. Poté přejde do operačního režimu a je připraven přijímat a vysílat zprávy.

3.3.2 Adresa 1 - Registr příkazů (COMMAND REGISTER)

Registr příkazů slouží k zadávání povelů řadiči CAN ze strany CPU, je umožněn pouze zápis, při požadavku na čtení je vrácena hodnota logická 0.

Interpretace bitů registru příkazů (CMR), CAN adresa 1

Bit	Symbol	Jméno	Hodnota	Význam
CMR.7 –CMR.4	-	Rezerva	-	-
CMR.3	CDO	Vynuluj příznak přetečení	1	Příznak přetečení vynulován
			0	(žádná akce)
CMR.2	RRB	Uvolni přijímací	1	Přijímací registr reprezentován

		zásobník		oblastí v přijímací FIFO paměti je uvolněn
			0	(žádná akce)
CMR.1	AT	Stornuj vysílání	1	Pokud ještě nezačalo, vysílání aktuální zprávy je zrušeno
			0	(žádná akce)
CMR.0	TR	Požadavek na vysílání	1	Požadavek na vysílání
			0	(žádná akce)

3.3.3 Adresa 2 - Stavový registr (STATUS REGISTER)

Položky stavového registru představují hodnoty jednotlivých stavů CAN řadiče, hodnoty registru lze pouze číst, případný zápis nebude proveden.

Interpretace bitů stavového registru (SR), CAN adresa 2

Bit	Symbol	Jméno	Hodnota	Význam
SR.7 –SR.6	-	Rezerva	-	-
SR.5	TS	Stav vysílání	1	Zpráva je vysílána
			0	Momentálně není vysílána žádná zpráva
SR.4	RS	Stav přijímání	1	Zpráva je přijímána řadičem
			0	Momentálně není přijímána žádná zpráva
SR.3	TCS	Vysílání dokončeno	1	Poslední požadovaná zpráva byla odvysílána
			0	Poslední požadovaná zpráva se stále vysílá
SR.2	TBS	Stav vysílacího zásobníku	1	CPU může zapisovat do tohoto zásobníku
			0	CPU nesmí zapisovat do zásobníku, poslední požadovaná zpráva se stále vysílá
SR.1	DOS	Příznak přetečení	1	Přijatá zpráva byla ztracena, protože přijímací FIFO paměť je zaplněna
			0	Od posledního vynulování tohoto bitu nedošlo k přetečení zásobníku
SR.0	RBS	Stav přijímacího registru	1	Jedna nebo více zpráv jsou dostupné v přijímací FIFO
			0	Žádná zpráva není k dispozici

3.3.4 Adresa 6 - Registr časování 0 (Bus Timing Register 0)

Obsah tohoto registru definuje hodnoty předděliče a synchronizačního skoku. Registr je přístupný pouze v servisním režimu, v operačním režimu je možné ho pouze číst.

Interpretace bitů registru časování 0 (BTR 0), CAN adresa 6

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SJW.1	SJW.0	BRP.5	BRP.4	BRP.3	BRP.2	BRP.1	BRP.0

3.3.5 Adresa 7 - Registr časování 1 (Bus Timing Register 1)

Obsah tohoto registru pomocí proměnných časový segment 1 (Time Segment 1) a časový segment 2 (Time Segment 2) definuje délku periody jednoho bitu a dobu, kdy se vzorkuje vstupní signál. Registr je přístupný pouze v servisním režimu, v operačním režimu je možné ho pouze číst.

Interpretace bitů registru časování 1 (BTR 1), CAN adresa 7

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
-	TSEG2.2	TSEG2.1	TSEG2.0	TSEG1.3	TSEG1.2	TSEG1.1	TSEG1.0

3.3.6 Adresa 16 až 28 - Vysílací zásobník

Vysílací zásobník má délku 13 byte a je umístěn v adresové části CAN řadiče mezi adresami 16 a 28. Zásobník je rozdělen do tří částí. První nese informace o vysílané zprávě jako formát zprávy a počet datových byte, ve druhé jsou umístěny identifikátory a ve třetí data pro vysílanou zprávu.

Členění vysílacího zásobníku

Adresa	Funkce registru	Adresa	Funkce registru
16	Informace o vysílané zprávě	23	TX data byte 3
17	TX identifikátor 1	24	TX data byte 4
18	TX identifikátor 2	25	TX data byte 5
19	TX identifikátor 3	26	TX data byte 6
20	TX identifikátor 4	27	TX data byte 7
21	TX data byte 1	28	TX data byte 8
22	TX data byte 2		

Informace o vysílané zprávě (TX frame information), CAN adresa 16

Bit	Symbol	Jméno	Hodnota	Význam
7	FF	Formát rámce	1	Rozšířený rámec
			0	Standardní rámec
6	RTR	Požadavek na vzdálené vysílání	1	Vyslán dotazový rámec s nulovým počten dat. bytů
			0	-

5	X	Nepoužito	-	-
4	X	Nepoužito	-	-
3 - 0	DLC.3	Počet datových bytů odesílané zprávy	1	Zadání délky počtu bytů zprávy pomocí čtyřbitového čísla
			0	

TX identifikátor 1 - 4, CAN adresa 17 - 20

TX identifikátor 1			TX identifikátor 2		
Bit	**Symbol**	**Popis**	**Bit**	**Symbol**	**Popis**
7	ID.28	Normální i rozšířený rámec	7	ID.20	Normální i rozšířený rámec
6	ID.27	Normální i rozšířený rámec	6	ID.19	Normální i rozšířený rámec
5	ID.26	Normální i rozšířený rámec	5	ID.18	Normální i rozšířený rámec
4	ID.25	Normální i rozšířený rámec	4	ID.17	Rozšířený rámec
3	ID.24	Normální i rozšířený rámec	3	ID.16	Rozšířený rámec
2	ID.23	Normální i rozšířený rámec	2	ID.15	Rozšířený rámec
1	ID.22	Normální i rozšířený rámec	1	ID.14	Rozšířený rámec
0	ID.21	Normální i rozšířený rámec	0	ID.13	Rozšířený rámec
TX identifikátor 3			**TX identifikátor 4**		
Bit	**Symbol**	**Popis**	**Bit**	**Symbol**	**Popis**
7	ID.12	Rozšířený rámec	7	ID.4	Rozšířený rámec
6	ID.11	Rozšířený rámec	6	ID.3	Rozšířený rámec
5	ID.10	Rozšířený rámec	5	ID.2	Rozšířený rámec
4	ID.9	Rozšířený rámec	4	ID.1	Rozšířený rámec
3	ID.8	Rozšířený rámec	3	ID.0	Rozšířený rámec
2	ID.7	Rozšířený rámec	2	X	-
1	ID.6	Rozšířený rámec	1	X	-
0	ID.5	Rozšířený rámec	0	X	-

3.4 Ovládání řadiče CAN po sériové lince

Komunikace po sériové lince probíhá pomocí tří typů zpráv:

1. Zápis bytu do registru řadiče(délka zprávy 4 byty)

Začátek zprávy 0xAAh	Adresa registru 1byte	Data 1 byte	Součet všech předchozích bytů

2. Žádost o přečtení bytu z registru řadiče(délka zprávy 2 byty)

Začátek zprávy 0xA5h	Adresa registru 1byte

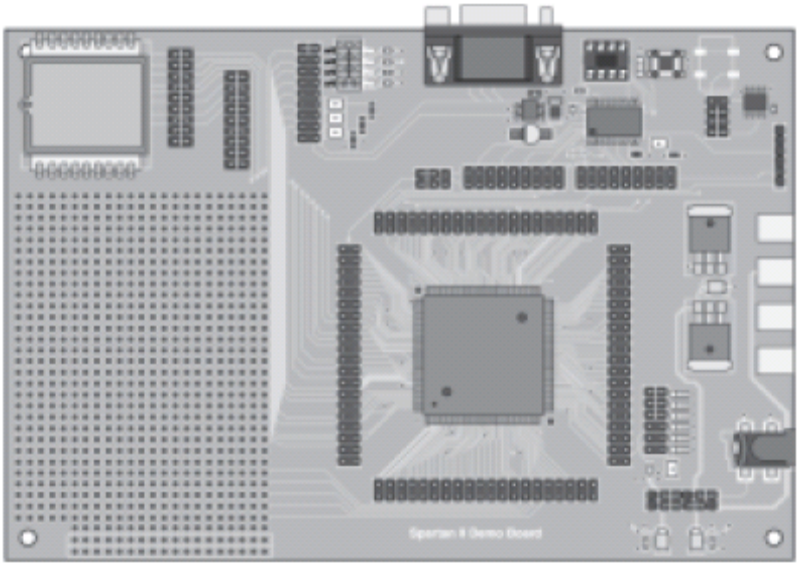
3. Odpověď řadiče na žádost o přečtení bytu z registru(délka zprávy 3 byty)

Začátek zprávy 0xAAh	Adresa registru 1byte	Data 1 byte

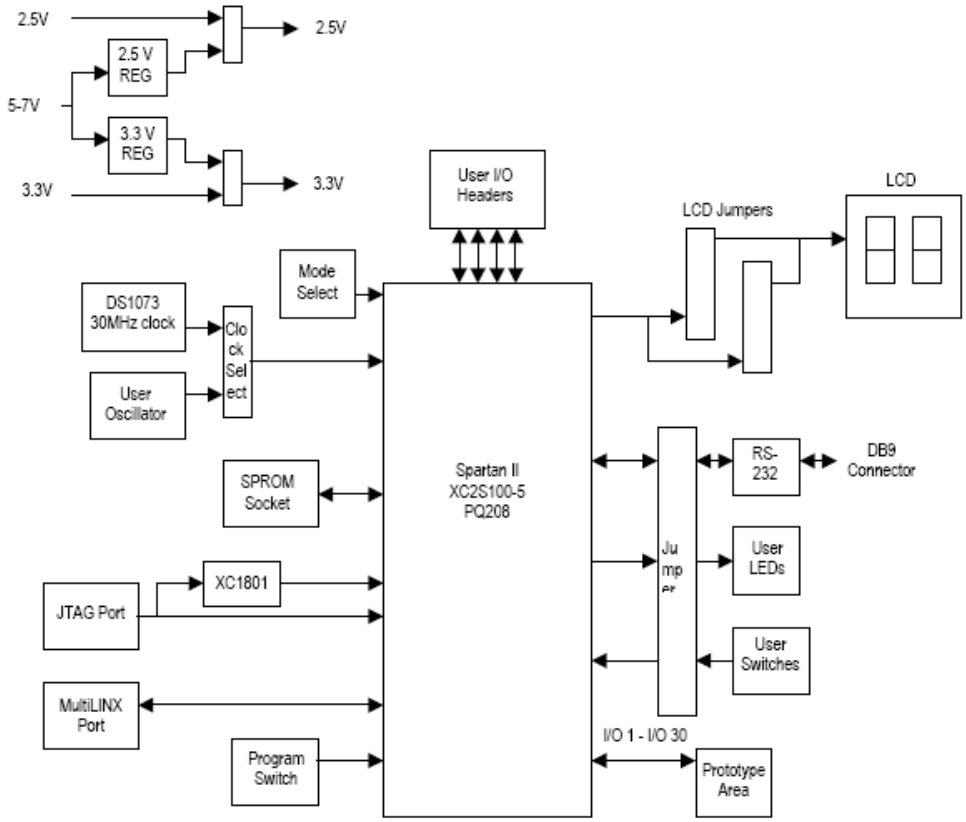
4 Realizace CAN řadiče

Řadič CAN, převzatý z předcházející diplomové práce[1], je realizovaný na ladicí desce Spartan II Demo Board obsahující hradlové pole XC2S100 od firmy XILINX. K programování hradlového pole XC2S100 od firmy XILINX se používá jazyk VHDL.

Pro tuto ladicí desku byl vytvořen modul galvanického oddělení zabezpečující bezpečné připojení do sítě CAN BUS.

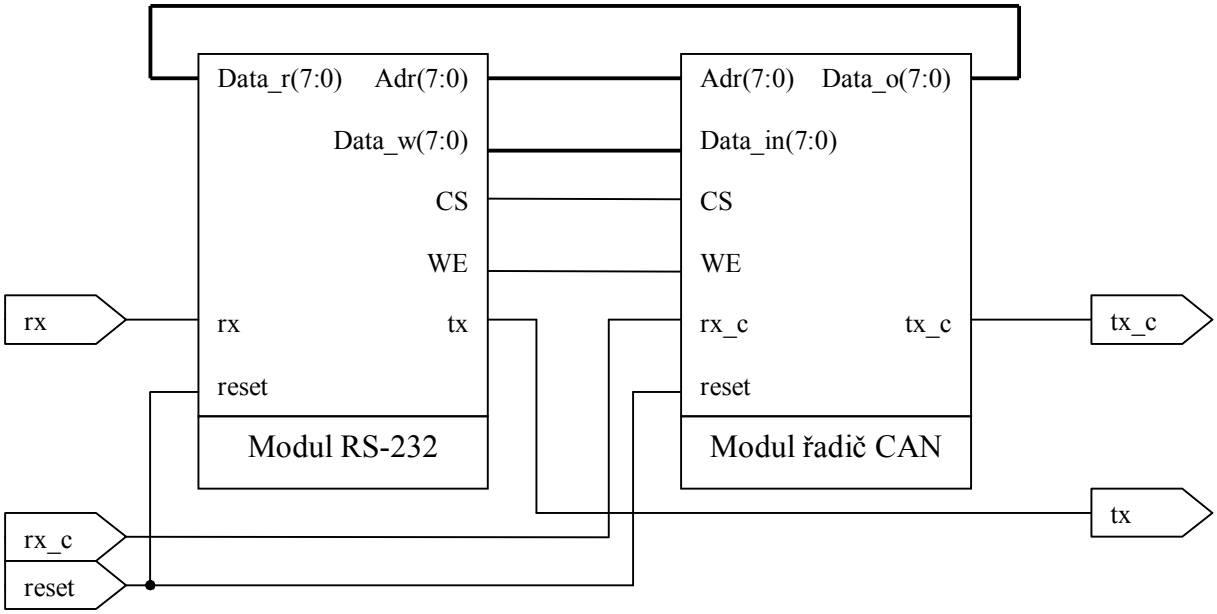


Obrázek 9 – Fyzická realizace



Obrázek 10 – Blokové schéma [2] ladicí desky Spartan II Demo Board

4.1 Vnitřní schéma zapojení modulů řadiče s RS232



Obrázek 11 – Zjednodušená vnitřní struktura řadiče s RS232 implementovaná v hradlovém poli

Popis jednotlivých vstupů a výstupů do modulu RS-232

Vstupy	Popis	Výstupy	Popis
rx	přijímaná data ze sériové linky	tx	odesílaná data na sér. linku
reset	obnovení počátečního stavu	Adr(7:0)	adresa pro zápis do reg. řadiče
Data_r(7:0)	načtená data z reg. řadiče	Data_w(7:0)	data pro zápis do reg. řadiče
		WE	0..povolení čtení dat z registru 1..povolení zápisu dat do reg.
		CS	1..provedení požadované oper.

Popis jednotlivých vstupů a výstupů do modulu řadič CAN

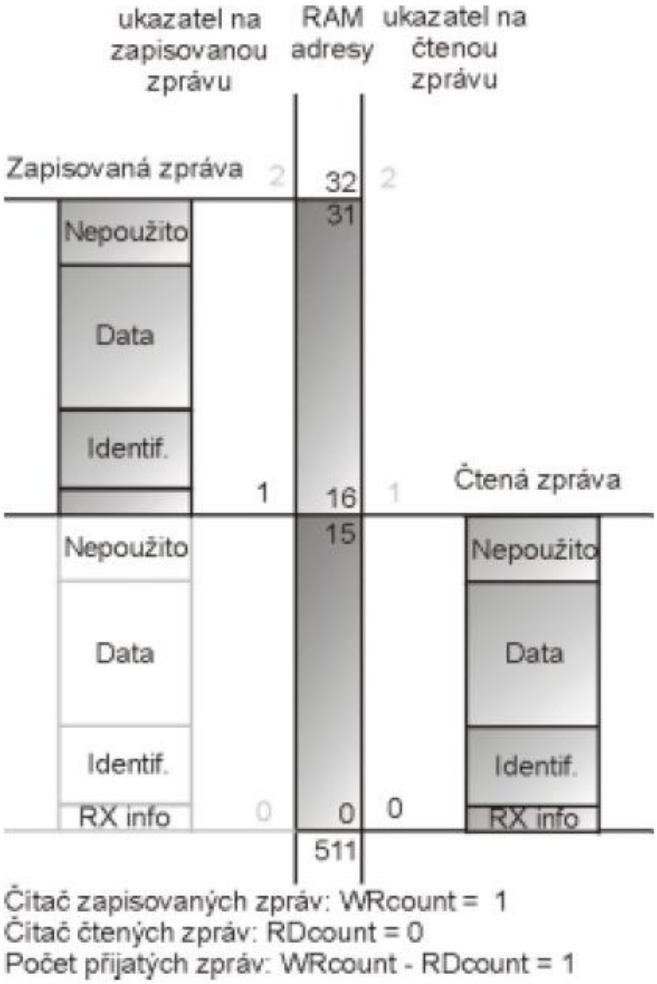
Vstupy	Popis	Výstupy	Popis
rx_c	přijímaná data z CAN-BUS	tx_c	odesílaná data na CAN-BUS
reset	obnovení počátečního stavu	Data_o(7:0)	přečtení dat z reg. řadiče
Adr(7:0)	adresa pro zápis do reg. řadiče		
Data_in(7:0)	data pro zápis do reg. řadiče		
WE	0..povolení čtení dat z registru 1..povolení zápisu dat do reg.		
CS	1..provedení požadované oper.		

Aktuální nastavení komunikačních parametrů
rychlost sběrnice CAN: 500kbit/s
parametry sériové linky: 19200Bd , parita: sudá, datových bitů: 8, stopbit:1

Řadič implementovaný v hradlovém poli pomocí jazyka VHDL je rozdělen na dva základní moduly:

- Modul RS-232 – zpracovává příchozí zprávy ze sériové linky. Pokud přijde žádost zápisu bytu do registru řadiče, tak se do modulu řadiče CAN odešle adresa (1 byte), data (1 byte), řídicí signál WE se nastaví na logickou 1 (povolení zápisu) a po nastavení řídicího signálu CS na logickou 1 se při náběžné hraně provede požadovaný zápis. Pokud přijde po sériové lince žádost o přečtení bytu z registru řadiče, tak se do modulu řadiče CAN odešle pouze požadovaná adresa registru, řídicí signál WE zůstane v logické 0 (čtení registru) a po nastavení řídicího signálu CS na logickou 1 se při náběžné hraně provede požadovaný zápis.
- Modul řadič CAN – zpracovává požadavky z modulu RS-232 na zápis nebo čtení dat z registru. Dále přijímá zprávy ze sběrnice CAN, jejíchž obsah si je možné po zpracování přečíst z příslušných registrů řadiče. Pomocí řadiče lze rovněž odesílat zprávy na sběrnici CAN. Pokud zprávy nedokážeme dostatečně rychle číst, tak se začnou nepřečtené (nebo neodeslané) uchovávat v FIFO zásobníku v modulu řadiče CAN.

4.2 FIFO modulu řadiče CAN



Obrázek 12 – Struktura paměti FIFO v modulu řadiče CAN

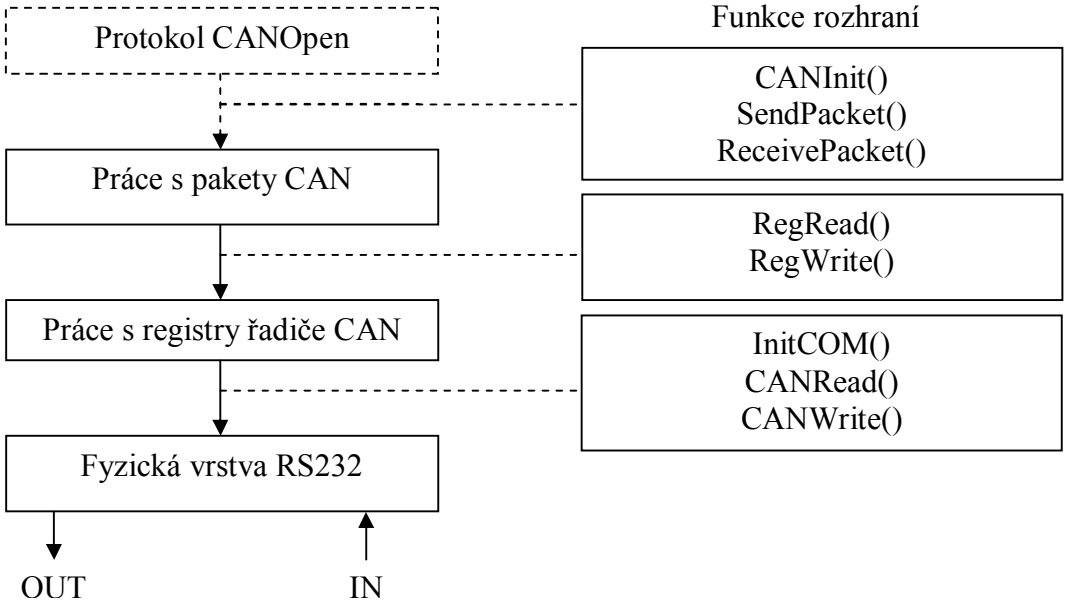
Tento modul využívá vnitřní dvoubranovou paměť RAM o velikosti 4kb, která je implementována na čipu hradlového pole. Každá brána paměti má nezávisle volitelný počet datových a tomu odpovídajících adresových signálů. Paměť v tomto případě umožňuje uchovávat 512 bytů dat. Počet bytů každé přijaté zprávy je proměnlivý. Každá zpráva se skládá z hlavičky a datové části. Hlavička obsahuje informace o umístění v FIFO, o typu zprávy a příznak, jestli se jedná o zapisovaný nebo čtený záznam. V paměti je možno najednou uchovat maximálně 32 zpráv. Ukazatel na začátek a konec fronty a počet zpráv ve frontě je udržován v čítačích.

5 Programové prostředky pro ovládání a diagnostiku

K naprogramování této aplikace byl použit programovací jazyk C++ z vývojového prostředí Microsoft Visual Studio 6.0.

5.1 Dělení vrstev programu

Aplikace TestComWin slouží pro přímé ovládání řadiče CAN z Windows a pro diagnostiku komunikace na síti CAN BUS.



Obrázek 13 – Struktura obslužného programu řadiče

5.2 Popis rozhraní jednotlivých vrstev

5.2.1 Vrstva pro práci s pakety CAN

CANInit() – inicializace řadiče CAN
- přechod řadiče do servisního režimu
- nastavení parametrů řadiče pro komunikaci na sběrnici CAN
- přechod do operačního režimu

SendPacket() – odeslání paketu na sběrnici CAN
- načtení adresy a dat z textových okének
- pomocí funkce CANWrite postupně odesílá jednotlivé byty, ze kterých je paket složený, na příslušné adresy v řadiči
- na konci odešle řadiči příkaz na odeslání celého paketu

ReceivePacket() – příjem příchozího paketu ze sběrnice CAN
- pomocí funkce CANRead postupně načítá jednotlivé byty, ze kterých je paket složený, z příslušných adres v řadiči a zobrazuje je do příslušných textových okének
- po přijmutí celé zprávy je poslán řadiči příkaz pro příjem nového paketu

5.2.2 Vrstva pro práci s registry

RegRead() – načtení registru řadiče CAN
- z textového okénka se načte adresa registru, ze kterého chceme přečíst aktuální hodnotu
- pomocí funkce CANRead se načte požadovaná hodnota registru a zobrazí se

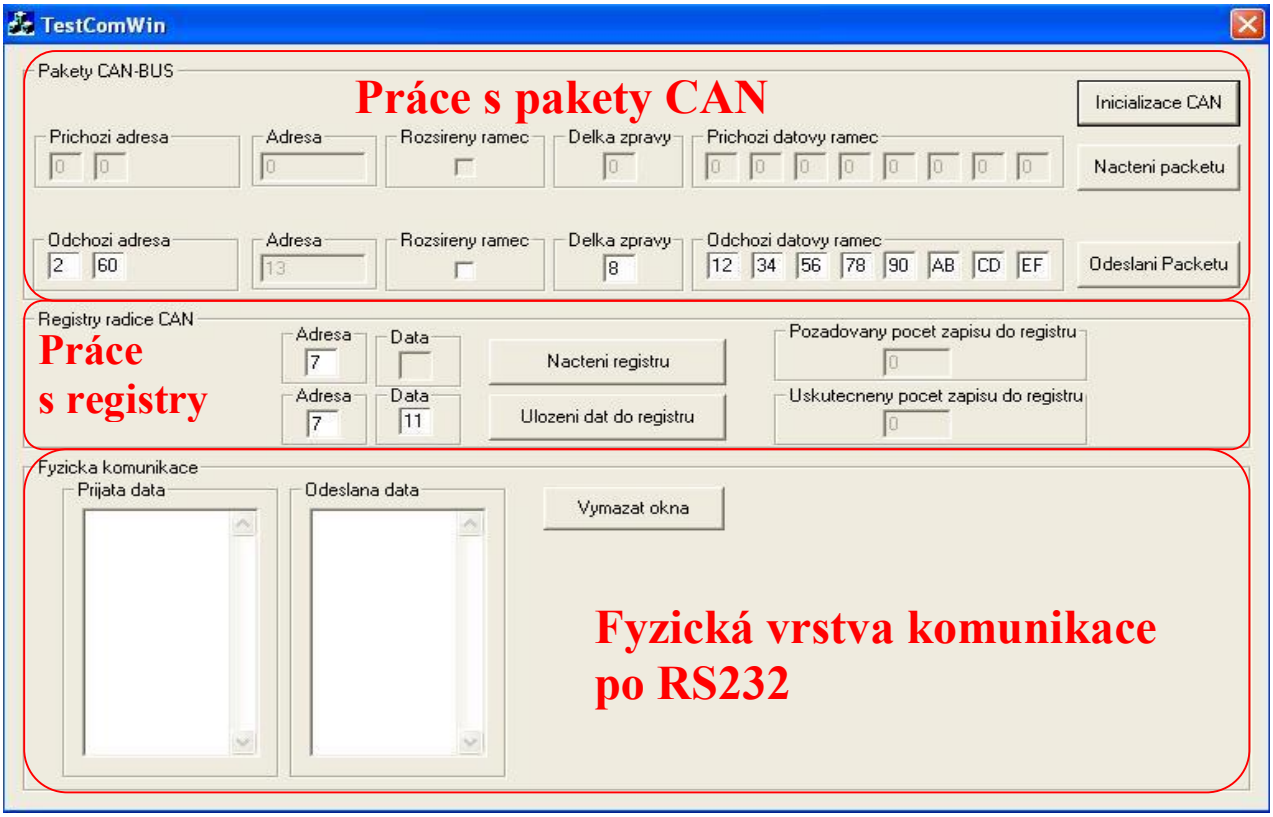
RegWrite() – zápis do registru řadiče CAN
- z textových okének se načte adresa a data určená pro zápis do registru
- pomocí funkce CANWrite se data zapíší do zvoleného registru

5.2.3 Fyzická vrstva RS232

InitCOM() – inicializace sériové linky
- před zahájením komunikace provede nastavení parametrů sériové linky

CANRead(Byte x) – čtení dat ze sériového kanálu
x ... požadovaná adresa registru, ze které chceme načíst data
- před adresu se přidá začátek zprávy (0xA5h) a odešle

CANWrite(Byte x1, Byte x2) – zápis dat po sériovém kanálu
x1 ... požadovaná adresa registru, do které chceme uložit data
x2 ... požadovaná data, které chceme uložit do registru řadiče
- před adresu se přidá začátek zprávy (0xAAh) a za zprávu se přidá součet všech předchozích bytů
- poté je celá zpráva odeslána



Obrázek 14 – Základní okno aplikace s vyznačením jednotlivých vrstev

5.3 Popis jednotlivých vrstev programu

Program je rozdělen do tří hlavních vrstev:

- Práce s pakety CAN
- Práce s registry řadiče CAN
- Fyzická komunikace po sériové lince

5.3.1 Práce s pakety CAN

Tato vrstva je rozdělena na zobrazování příchozího paketu po lince CAN a na zadávání odchozího paketu. Adresa se zadává v čtyř bytovém tvaru. Každý byte zastupuje jeden registr v řadiči. Tato adresa je také přepočtena a zobrazena jako jedno více bytové číslo. Zadává se také počet datových bytů k odeslání. Podle zadání délky zprávy se zobrazí příslušný počet okének datové části paketu.

5.3.2 Práce s registry řadiče CAN

Tato vrstva slouží pro přímé čtení a zápis registrů řadiče CAN. Pomocí ní lze zapisovat do příslušného registru řadiče po zadání jeho adresy a dat, která se mají zapsat. Pro zobrazení obsahu jednotlivých registrů řadiče stačí napsat požadovanou adresu registru.

5.3.3 Fyzická komunikace po sériové lince

Vrstva fyzické komunikace zabezpečuje komunikaci této aplikace se službami jádra operačního systému Windows pro obsluhu sériové linky.

6 Závěr

Pomocí vytvořené testovací aplikace TestComWin lze nyní snadno ovládat řadič CAN a tím také monitorovat komunikaci na sběrnici CAN-BUS. K naprogramování této aplikace byl použit programovací jazyk C++ z vývojového prostředí Microsoft Visual Studio 6.0. Toto vývojové prostředí bylo zvoleno kvůli snadné práci s knihovnou pro vývoj aplikací MFC a rozsáhlé dokumentaci MSDN Subscription Library.

Koncepce programu je modulární, proto lze například zaměnit fyzickou vrstvu komunikace po sériové lince za komunikaci po USB.

Účelem této práce bylo ovládání CAN řadiče a proto neobsahuje implementaci protokolu CanOpen. Byla odzkoušena komunikace řadiče na sběrnici CAN-BUS s druhým zařízením. Jednalo se o PCI řadič PCAN-PCI od firmy Peak.

7 Literatura

[1] Ditrich, M.: diplomová práce - Řešení komunikace mezi systémem NX5030 firmy INTRONIX a sériově vyráběnými automaty PLC.

[2] Spartan-II Demo Board, User's Guide [online]
Poslední úprava: 3/2000
http://old.iot.dk/jjm/adsen/XiLinx/Spartan2_kit/Spartan-II%20User's%20Guide.pdf

[3] Kocourek, P., Novák, J.: Přenos informace
Poslední úprava: 3/2004

[4] Závidčák M.: CAN - popis struktury [online]
http://www.hw.cz/Rozhrani/ART1111-CAN---popis-struktury.html

[5] Hlava J.: CAN – stručný návod [online]
http://www.fm.vslib.cz/~krtsub/fm/par/can.pdf

[6] Spurný F.: Úvod do problematiky sběrnice CAN [online]
http://fieldbus.feld.cvut.cz/can/

[7] Románek D.: Komunikační protokol CANopen
Sdělovací technika 6/2006 (strana 56-57)

[8] MSDN Subscription Library [online]
http://msdn.microsoft.com/library/default.asp

8 Obsah přiloženého CD

Obsah kořenového adresáře přiloženého CD

Dokumentace	Elektronická verze bakalářské práce
TestComWin	Spustitelná verze aplikace pro obsluhu řadiče CAN
TestComWinSRC	Kompletní projekt k aplikaci včetně zdrojových kódů