

Bakalářská práce
Profibus DP slave s SPC4

Martin Váňa

23. srpna 2006

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

podpis

Anotace

Tato práce je zaměřena na krátké seznámení se sběrnicí Profibus DP a dále pak na návrh HW zařízení Profibus DP slave s využitím obvodu ASIC SPC4-2 od firmy Siemens. Tento obvod je využíván jako periferie k mikroprocesorovému modulu MCF5235BCC, který je založen na procesoru Freescale ColdFire M5235. Poslední část se zabývá vývojem softwaru pro obsluhu obvodu SPC4-2 pomocí procesoru.

Na práci se bude nadále pokračovat. Postupnou implementací vyšších verzí Profibus DP V1, V2. Dále se předpokládá využití k polohování pohonů prostřednictvím sítě Profibus.

Annotation

This thesis is aimed at the brief introduction to industrial field bus Profibus DP. Further goal is hardware design of Profibus DP slave machine based on ASIC circuit Siemens SPC4-2. Microprocessor evolution kit MCF5235BCC based on Freescale ColdFire M5235 makes use of SPC4-2 as a peripheral device. Final part deals with development of software for controlling SPC4-2 circuit.

I am going to continue on this thesis in the following years as a sequent implementation of higher version of Profibus DP V1, V2. Furthermore this work supposes for utilization to motion control application via Profibus field bus.

Poděkování

Na tomto místě bych rád poděkoval své rodině za podporu při studiu. Dále bych chtěl poděkovat vedoucímu mé práce ing. Pavlu Burgetovi za cenné rady a připomínky při kompletním řešení práce a umožnění seznámit se s touto zajímavou oblastí řídicích systémů. Nakonec děkuji ing. Pavlu Píšovi za rady při vývoji části programu a ing. Lukáši Ručkayovi za rady při návrhu desky plošných spojů.

Obsah

1	Úvod	8
2	Profibus DP	9
2.1	Profibus	9
2.2	FDL vrstva	9
2.2.1	Typy rámců	9
2.2.2	SAP pro cyklickou datovou výměnu	11
2.2.3	Typy služeb	11
2.2.4	Stavový automat DP-V0 slave	13
3	Obvod SPC4-2	15
3.1	SAP List	16
3.1.1	Struktura SAP Listu	16
3.1.2	Control byte	16
3.1.3	Request SA	18
3.1.4	Request SSAP	18
3.1.5	Access byte	19
3.1.6	Reply-Update-Ptr/SDN-/DDB-Tln-Tab-Ptr	19
3.2	Indication fronta	19
3.3	Reply-on-Indication blok	20
4	Návrh schéma zapojení a desky s SPC4-2	22
4.1	Popis návrhu	22
4.1.1	Výběr podpůrných obvodů	23

4.1.2	Návrh desky plošného spoje	24
4.2	Výsledné schéma zapojení	24
4.3	Výsledný návrh DPS	24
4.4	Časování a propojení signálů s M5235BCC	25
5	Návrh programu pro obsluhu SPC4-2	27
5.1	Vývojové prostředí	27
5.2	Základní práce s procesorem	28
5.2.1	Ovládání výstupního pinu	28
5.2.2	Obsluha přerušení	29
5.2.3	Nastavení chip select a obsluha sběrnice	30
5.2.4	Připojení k přípravku s SPC4-2	33
6	Závěr	37
7	Příloha	39

Seznam obrázků

2.1	Struktura SD1	10
2.2	Struktura SD2	10
2.3	Struktura SD3	10
2.4	Struktura SD4	11
2.5	Struktura SC	11
2.6	Stavový automat Profibus DP slave	14
3.1	Control byte	16
3.2	Struktura indication fronty	20
3.3	Struktura reply-on-indication bloku	21
4.1	Asynchronní časování sběrnice pro procesor Motorola	25
5.1	Náčrt průběhů z osciloskopu	33
7.1	Navržené schéma zapojení	40
7.2	Rozmístění součástek pro osazování	41
7.3	Spoje vrstvy Top	42
7.4	Spoje vrstvy Bottom	43
7.5	Navržená úprava zapojení	44

Seznam tabulek

2.1	Zkratky v rámcích SDx	12
2.2	SAP používané pro cyklickou datovou výměnu	12
3.1	Struktura SAP listu	17
4.1	Asynchronní časování sběrnice pro procesor Motorola	26
4.2	Neshodně pojmenované signály	26
5.1	DC specifikace pinů	35

Kapitola 1

Úvod

V úvodu práce se krátce seznámíme s protokolem Profibus DP, který je jednou z nejvíce rozšířených verzí standardu Profibus a průmyslových sběrnic vůbec. Seznámení je zaměřeno zejména z pohledu vrstvy FDL (Field Data-link Layer). Zádání práce je navrhnout zařízení s obvodem Siemens SPC4-2. SPC4-2 je tzv. ASIC obvod (Aplication Specific Integrated Circuit), který obsahuje komunikační standard Profibus DP. Tento obvod však není schopen samostatně pracovat. Potřebuje ke své činnosti řídící mikroprocesor, takže vlastně tvoří řadič sběrnice. Takto lze tedy vytvořit vlastní inteligentní zařízení dle vlastních požadavků a nároků, které bude schopno pracovat v průmyslové síti i s jinými komerčně vyráběnými zařízeními.

Pro náš navržený hardware byl vybrán procesor architektury Motorola Freescale M5235. Jeho bohatá výbava je pro další rozvoj zařízení velice vhodná.

V poslední kapitole se věnujeme vývoji programu pro obsluhu SPC4-2 a řešení vzniklých problémů.

Kapitola 2

Profibus DP

2.1 Profibus

Kompletní popis protokolu Profibus je nad rámec této kapitoly a není cílem této práce. Proto v této kapitole uvedu pouze stručný úvod do verze DP a další popis bude uveden z pohledu FDL (spojové) vrstvy.

Profibus DP (Distributed Periphery) vznikl jako rozšíření předchozího protokolu FMS viz. [1]. Jeho jednoduchost způsobila rozšíření do aplikací automatizované výroby. Dále byl rozšířen o dvě verze DP-V1 a DP-V2.

2.2 FDL vrstva

Datová výměna v Profibus DP je prováděna zasíláním telegramů (rámců) ze zařízení typu master do zařízení typu slave a naopak. Vrstva 2 ISO/OSI modelu zahrnuje obecný popis rámců, zabezpečovací mechanismus a typy služeb.

2.2.1 Typy rámců

FDL vrstva rozlišuje čtyři typy rámců. Druh rámce se rozpozná podle prvního znaku SD = start delimiter.

- **SD1** = Request_FDL_Status: Tento rámc je zaslán vždy aktivní stanicí po vypršení GAP time. Podrobnější vysvětlení a popis je uveden v knize [2]. Struktura rámce je uvedena na obrázku 2.1.

SD	DA	SA	FC	FCS	ED
0x10	xx	Xx	X	x	0x16

Obrázek 2.1: Struktura SD1

- **SD2** = Telegram: Rámc pro přenos s proměnnou délkou dat. Obsahuje políčka LE a LER, které obsahují délku dat včetně DA, SA, FC, DU. Používá se ve službě SRD viz. 2.2.3. Obrázek 2.2.

SD	LE	LER	SD	DA	SA	FC	DU	FCS	ED
0x68	X	X	0x68	xx	xx	x	x ...	X	0x16

Obrázek 2.2: Struktura SD2

- **SD3** = Telegram: Rámc s pevnou délkou dat. DU je vždy dlouhé 8 bytů. Obrázek 2.3.

SD	DA	SA	FC	DU	FCS	ED
0xA2	xx	Xx	x	x ...	x	0x16

Obrázek 2.3: Struktura SD3

- **SD4** = Token telegram: Rámc zasílaný mezi dvěma aktivními stanicemi, určený k předání přístupu na sběrnici. Obrázek 2.4.
- **SC** = Short acknowledgement: Používá se pouze pro potvrzení příjmu. Hodnota je vždy 0xE5. Obrázek 2.5.

Použité zkratky v popisu struktur jednotlivých rámců jsou vysvětleny v tabulce 2.1.

Pro adresu je použit jeden znak, ale využívá se pouze 7 bitů, takže lze adresovat 127 stanic - 0 až 126. Adresa 127 má speciální význam, je-li tento

SD	DA	SA
0xDC	xx	Xx

Obrázek 2.4: Struktura SD4

SC
0xE5

Obrázek 2.5: Struktura SC

bit nastaven, v adrese DA nebo SA, začíná datové pole adresním rozšířením - tzv. přístupovým bodem SAP (Service Acces Point). Tyto body nabízejí další adresní možnosti, mohou identifikovat určitou funkci. Podrobnější popis lze nalézt v [2, 1].

2.2.2 SAP pro cyklickou datovou výměnu

Pro cyklickou datovou výměnu se využívá tzv. SAP - Service Access Point. Zjednodušeně si lze představit, že každý SAP je místem v paměti, které je vstupním nebo výstupním bodem nějakého cyklického procesu. Například vstupy do zařízení Profibus DP slave nastavují hodnoty v příslušném SAP a na druhé straně jsou tato data čtena nadřazeným zařízením.

Pro cyklickou datovou výměnu pro verzi Profibus DP-V0 jsou definovány SAP uvedené v tabulce 2.2.

2.2.3 Typy služeb

Uživatel má k dispozici služby poskytované vrstvou FDL, kterou o provedení žádá prostřednictvím komunikačních primitiv REQ - request = žádost. FDL vrstva odpovídá primitivou CON - confirm = potvrzení pozitivní/negativní. Naopak vrstva FDL může volat vrstvu nadřazenou primitivou IND - indication = událost. Pro odpověď je používána primitiva RES - response = odpověď'.

Pro vrstvu FDL jsou definovány dva typy stanic:

SD:	Start Delimiter
LE:	Length: délka dat včetně DA, SA, FC, DU
LER:	Repetition of the lenght: opakování délky, protože není zahrnuta v HD=4 ¹
DA:	Destination Address: cílová adresa
SA:	Source Address: adresa vysílající stanice
FC:	Function Code: obsahuje řídící znak
DU:	Data Unit: uživatelská data, rozsah 1..244 byte
FCS:	Frame Checking Sequence: obsahuje kontrolní součet CRC
ED:	End Delimiter: obsažen vždy na konci mimo SD4. Hodnota vždy 0x16

Tabulka 2.1: Zkratky v rámcích SDx

Default SAP:	Data exchange (Input_Output_Data)
SAP 0x36:	Master-master SAP (M-M communication)
SAP 0x37:	Change station address (Set_Slave_Add)
SAP 0x38:	Read inputs (Rd_Inp)
SAP 0x39:	Read outputs (Rd_Outp)
SAP 0x3A:	Control commands to the DP slave (Global_Control)
SAP 0x3B:	Read configuration data (Get_Cfg)
SAP 0x3C:	Read diagnostic information (Slave_Diagnosis)
SAP 0x3D:	Send parametrization data (Set_Prm)
SAP 0x3E:	Check configuration data (Set_Cfg)

Tabulka 2.2: SAP používané pro cyklickou datovou výměnu

- aktivní - může sama vysílat zprávy, jakmile získá přístup ke sběrnici,
- pasivní - může pouze odpovídat na žádosti aktivní stanice.

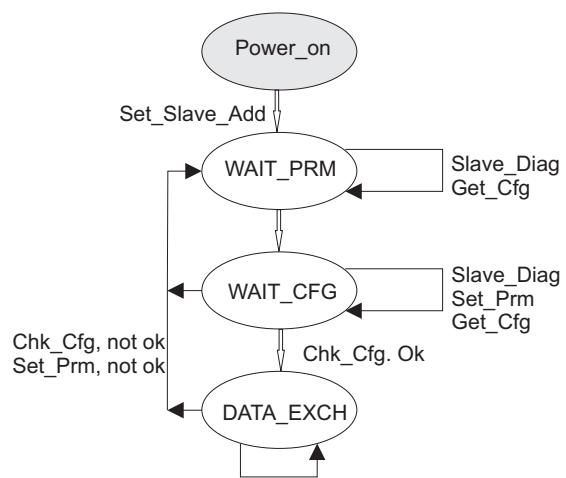
Nyní popíšeme definované služby vrstvy FDL.

- **SDA** Send Data with Acknowledge = pošli data a čekej na potvrzení. Data jsou poslána stanici master nebo slave a jako odpověď je posláno potvrzení. Je používáno pouze aktivními stanicemi.
- **SRD** Send and Request Data with acknowledge = pošli data a čekej zprávu s odpovědí. Tato služba je určena pro přímou výměnu dat mezi dvěma stanicemi. Bud' mezi dvěma aktivními stanicemi nebo mezi aktivní a pasivní stanicí.
- **SDN** Send Data with No Acknowledge = pošli data a neočekávej potvrzení. Používá se pro odeslání zprávy od jedné stanice k více stanicím či skupinám. Uživatel dostane potvrzení o odeslání zprávy, nikoliv však o jejím doručení.

2.2.4 Stavový automat DP-V0 slave

Pro snadné pochopení, jak Profibus DP funguje, je znázorněn na obrázku 2.6 jednoduchý stavový automat DP-V0 slave. Ten popisuje, jak se stanice bude chovat v možných situacích.

Ve stavu DATA_EXCH přijímá stanice tyto telegramy: Data Exchange-ok, Rd_Inp, Rd_Outp, příkazy (Sync, Freeze...), Slave_Diag, Chk_Cfg ok, Prm-ok, Get_Cfg. Podrobný popis těchto příkazů lze nalézt v knize [2].



Obrázek 2.6: Stavový automat Profibus DP slave

Kapitola 3

Obvod SPC4-2

Pro jednoduchou a rychlou datovou výměnu mezi zařízeními komunikujícími pomocí protokolu Profibus poskytuje firma Siemens různé ASIC obvody. Obvod SPC4-2 (Siemens Profibus Controller) pracuje na první vrstvě OSI/ISO modelu, a proto vyžaduje ke své činnosti procesor, který implementuje vrstvy nadřazené(vrstvy 2..7). V obvodu je integrována část vrstvy 2, která obsluhuje protokol na sběrnici. Další funkce jsou požadovány po přidaném procesoru. Na obvodu jsou ještě dále integrovány následující služby: Data_Exchange, Read_Input, Read_Output a Global_Control.

SPC4-2 podporuje tyto komunikační profily: Profibus FMS, Profibus PA, Profibus DP a Foundation Fieldbus (FF). Profily jsou definovány v normě IEC 61784-1. Pro naši práci bude využíváno profilu Profibus DP.

Obvod poskytuje komunikační interface pro oba typy architektur Intel a Motorola. Datový formát a synchronní nebo asynchronní komunikace se volí pomocí dvou konfiguračních pinů.

Komunikace mezi procesorem a SPC4-2 je prováděna prostřednictvím FLC firmware (Field Bus Link Control) přes dual-port RAM. Z pohledu uživatele zabírá adresový prostor SPC4-2 1KByte.

3.1 SAP List

3.1.1 Struktura SAP Listu

Service Access Point je tvořen třemi následujícími částmi:

- 5-ti SM-SAP (System Management SAP), každý po 5 bytů dlouhý
- 1 DEFAULT-SAP 16 bytů
- 64-mi SAP, každý 5 bytů dlouhý

Služby pro přenos dat jsou na vrstvě FLC poskytovány prostřednictvím SAP. Každý services access point včetně default SAP má speciální vstup, který umožňuje vrstvě FLC příjímat informace od zdroje. Pokud SPC4-2 přijme rámc pro neexistující SAP, odpoví ”no service activated” (SD1 response). Jednotlivé registry jsou přiřazeny ke každému SAP v SAP listu, jakmile je do nich zapsáno.

3.1.2 Control byte

Bit Position								Meaning
7	6	5	4	3	2	1	0	
SAP locked	SDN/ DDB filter	RS/RA or UE	RR	IN USE	Buffer available			Control byte

Obrázek 3.1: Control byte

Jak je vidět z tabulky 3.1, tento byte je obsažen v každém typu SAP. První tři bity jsou použity jako čítač. FLC inkrementuje čítač, jakmile je zdroj přístupný. SPC4-2 dekrementuje čítač, když byl signalizován přijatý blok. Pokud je z čítače čtena nula, přeruší se příjem, nastaví se ”No Resource” (RR) a odpoví se no resource (SD1 response).

Address	Name	register	Meaning
18H	SM1	Control Byte	Bit information
19H		Request-SA	Request source address
1AH		reserved	
1BH		reserved	
1CH		Reply-Update-Ptr/ SDN-DDB-TIn-Tab-Ptr	Pointer to the reply buffer
1DH - 21H	SM2	analogous to SM1	analogous to SM1
22H - 26H	SM3	analogous to SM1	analogous to SM1
27H - 2BH	SM4	analogous to SM1	analogous to SM1
2CH - 30H	SM5	analogous to SM1	analogous to SM1
31H	DEFAULT SAP	Control byte	Bit information
32H		Request-SA	Request source address
33H		Request SSAP	Request source service access point
34H		Access-Byte	Access protection
35H		Reply-Update-Ptr/ SDN-DDB-TIn-Tab-Ptr	Pointer to the reply buffer
36H		Reply-Update-Ptr D	
37H		Reply-Update-Ptr N	
38H		Reply-Update-Ptr U	
39H		Response-Buffer-Length	
3AH		Response Status	
3BH		Indication-Buffer-Ptr D	
3CH		Indication-Buffer-Ptr N	
3DH		Indication-Buffer-Ptr U	
3EH		Indication-Buffer-Length	
3FH		Active-Group-Ident	
40H		Control-Command	
41H	SAP[0]	Control byte	Bit information
42H		Request-SA	Request source address
43H		Request SSAP	
44H		Access-Byte	
45H		Reply-Update-Ptr/ SDN-DDB-TIn-Tab-Ptr	Pointer to the reply buffer
46H - 4AH	SAP[1]	analogous to SAP [0]	analogous to SAP [0]
4BH - 17BH	SAP[2]- SAP[62]-	analogous to SAP [0]	analogous to SAP [0]
17CH	SAP[63]	Control byte	Bit information
17DH		Request-SA	Request source address
17EH		Request SSAP	
17FH		Access-Byte	
180H		Reply-Update-Ptr/ SDN-DDB-TIn-Tab-Ptr	Pointer to the reply buffer

Tabulka 3.1: Struktura SAP listu

IN USE SPC4-2 nastaví tento bit, když je celý rámec požadavku v indication frontě. Nuluje bit, jakmile je indikace provedena (platný/neplatný). Pokud chce FLC postoupit na další blok, musí počkat, až bude bit vynulován.

RR = No Resource SPC4-2 nastaví tento bit, pokud obsah čítače buffer available je nula.

RS/RA or UE No service activated/Service access point blocked or user error. SPC4-2 nastaví tento bit, pokud platnost *request SA* je negativní. Tento bit je také nastaven pokud je *request SA*=7Fh, SAP je neaktivní.

SDN/DDB filter - tento bit povoluje SDN/DDB filtr. Je-li nula, Reply-Update-Ptr/SDN-/DDB-Tln-Tab-Ptr ukazuje do Reply-On-Indication bloku, a proto se zašle odpověď. Jestliže je ukazatel 00h, pak není v bufferu odpověď a zašle se SRD požadavek s krátkým potvrzením SC. Je-li bit nastaven, Reply-Update-Ptr/SDN-/DDB-Tln-Tab-Ptr ukazuje do tabulky uzlu a SPC4-2 je ”odebíratel” pro tento SAP.

SAP locked SAP nyní nepřijímá data. Pokud SPC4-2 přijme data pro tento SAP, nastaví se příznak user error (UE) a odpoví user error (SD1 response)

3.1.3 Request SA

Přijatá SA (Source address) je porovnána s tímto vstupem. Pokud nekoresponduje, SPC4-2 nastaví příznak no service activated (RS) a odpoví no service activated [RS] - v našem případě v Profibus modu (SD1 response). Pro Default SAP jsou přístupné adresy 00-7Eh, pro ostatní SAP 80-FEh (nastaven extension bit). 7Fh vede zablokování SAP (No Service Activated). Pokud je adresa FFh = All, nepotvrzuje se příjem.

3.1.4 Request SSAP

SSAP = Source Service Access Point. Vyhodnocení probíhá shodně jako v případě request SA. Pokud request SA je mezi 00-7Eh, request SSAP je FFh, vybere se default SAP. Pokud extension bit v request SA je nastaven

a request SSAP je FFh, tak SSAP je neplatný.

3.1.5 Access byte

Access byte řídí ochranu přístupu k odpovídajícím SAP. Pokud je vstup 0h, znamená to "No Access Protection". Jestliže SPC4-2 přijme rámec, kde ne-koresponduje access byte, odpovídá "No Service Activated". Bit RS je nastaven. Ze směru FLC jsou všechny přístupy filtrovány. Odpověď [RS] (no service activated) je zaslána žadateli. Výjimka je v případě, že DDB odpovědi (subscriber) není odpovězeno negativně. Podrobné popisy úrovní zabezpečení přístupu jsou popsány tabulkou 5-3 na straně 36 v [3].

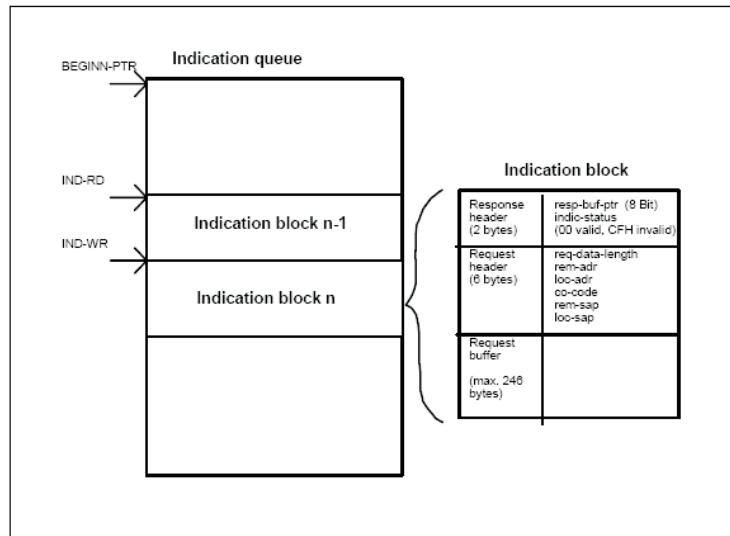
3.1.6 Reply-Update-Ptr/SDN-/DDB-Tln-Tab-Ptr

Tento ukazatel ukazuje do indication reply bufferu nebo do SDN/DDB-Tln listu (podrobný popis je v [3]). Rámce SDN, kromě SM-TIME, mohou být filtrovány pomocí tabulky (request SA, request SSAP). Filtr je aktivní, pokud je nastaven bit SDN/DDB Filter přijímajícím SAP.

3.2 Indication fronta

Pokud SPC4-2 přijme rámec, zapíše se hlavička rámce do indication fronty a poté se zkонтroluje volná délka ve frontě (toto je možné, protože jeden segment musí vždy zůstat volný). Pokud je alespoň jeden segment (8 nebo 16 byte) volný, pokračuje se v příjmu, dat dokud je volná paměť. Když je rámec žádosti přijat, SPC4-2 ověří znaky v hlavičce s hodnotami, které jsou nastaveny v SAP listu.

Indication fronta je řízena jako ring buffer s ukazatelem pro čtení (IND-RD) a ukazatelem pro zápis (IND-WR), jak je znázorněno na obrázku 3.2. SPC4-2 odpovídá za ukazatel zápisu a FLC odpovídá za ukazatel čtení.



Obrázek 3.2: Struktura indication fronty

3.3 Reply-on-Indication blok

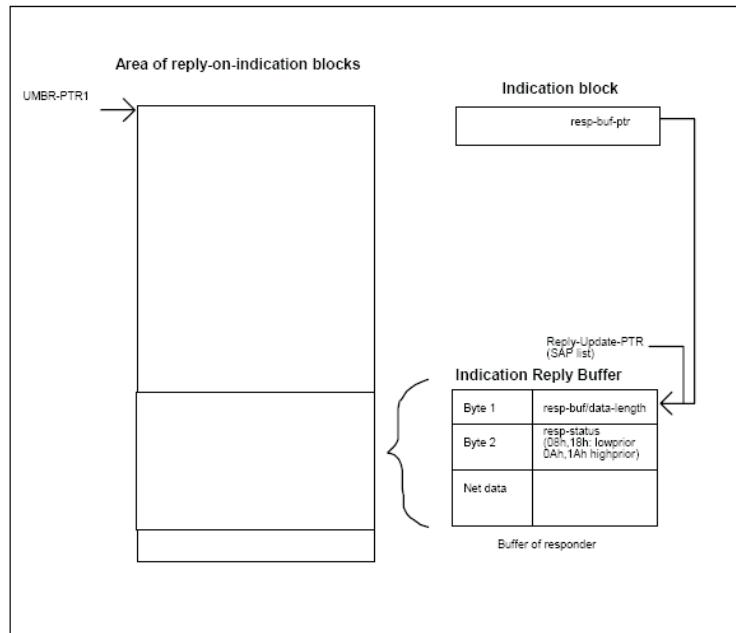
FLC musí poskytovat data s odpovědí v reply-on-indication bloku. Pokud je odpověď vyžadována, SPC4-2 způsobí update ukazetele z odpovídajícího SAP listu a předá data do reply bufferu. Jakmile se událost kompletně provede, SPC4-2 označí úkol vložením valid indication do hlavičky odpovědi, nastaví ukazatel zápisu na další volný segment a vygeneruje přerušení IND. Úkol je kompletní pokud:

- SM, SDN nebo DDB rámec byl přijat bez chyb a potvrzen.
- SDA nebo SRD rámec byl přijat bez chyb a potvrzen, odpověď byla odeslána a další požadavek rámce k jinému uzlovému bodu nebo lokální adrese byl v pořádku přijat.

Když SPC4-2 přijímá SRD nebo DDB rámec s nulovou délkou dat a pokud odpověď má také nulovou délku dat, SPC4-2 nevpustí tento rámec do indication fronty a nesignalizuje jej (empty pooling).

FLC může řídit, jak často budou data odesílána z indication reply bufferu nastavením odpovídajích bitů v responder status (byte 2). Podrobý popis je

na straně 44 v [3]. Na obrázku 3.3 je zobrazena struktura bloku reply-on-indication.



Obrázek 3.3: Struktura reply-on-indication bloku

Response buffer obsažený v bloku Reply-On-Indication obsahuje délku dat, status a samotná data s odpovědí.

Kapitola 4

Návrh schéma zapojení a desky s SPC4-2

4.1 Popis návrhu

Druhým úkolem práce po seznámení se s protokolem je navrhnut desku s obvodem SPC4-2. Tento ASIC obvod vyžaduje ke své činnosti řídící procesor [3]. Ze tří procesorů Siemens XC167, Philips LPC2104-ARM a Motorola-Freescale ColdFire5235 byl původně vybrán procesor Philips, avšak po krátkém nastudování jsme došli k závěru, že nepřítomnost vyvedení adresové a datové sběrnice z pouzdra je příliš omezující pro předpokládanou požadovanou rychlost komunikace mezi procesorem a obvodem SPC4-2. Proto jsme ve finálním výběru použili procesor ColdFire M5235 od firmy Freescale. Tento procesor navíc obsahuje eTPU (Enhanced Time Processing Unit) jednotku, která bude velmi přínosná v později uvažovaném polohování pohonů. Pro usnadnění realizace byl zakoupen přímo M5235BCC EVB - Evalution Module for Freescale MCF5235 MCU. Tato deska již obsahuje paměti, ethernet rozhraní, BDM rozhraní, sériovou linku - například pro připojení k terminálu.

K této kostře byly postupně doplnovány další podpůrné obvody ke kompletní realizaci zařízení DP slave.

4.1.1 Výběr podpůrných obvodů

Připojení obvodu SPC4-2 do sítě Profibus přes rozhraní RS485 musí být galvanicky odděleno. Na výběr se naskytly dvě možnosti:

1. Použití klasických optočlenů a budičů sběrnice podle doporučeného zapojení v manuálu k obvodu SPC4-2 [3].
2. Využít obvod ADM2486 - rychlý budič sběrnice RS485 a galvanické oddělení v jednom pouzdře.

Nakonec byl pro návrh i přes obtížnou dostupnost použit obvod ADM2486. K tomuto obvodu je zapotřebí pouze resetovací obvod. Podle doporučeného zapojení [4] jsem použil obvod ADM809z. Ten byl v ČR taktéž nedostupný¹, ale lze ho nahradit obvodem MAX809 s velmi podobnými parametry.

Kvůli galvanickému oddělení je dále použit standardní DC-DC měnič. Pro generování hodinových pulsů pro SPC4-2 je využit oscilátor 48MHz v pouzdře DIL8.

Další problém, který by vznikl u použití jak procesoru Philips ARM, tak u Freescale Coldfire, je rozdílné napájecí napětí. Všechny procesory využívají pro napájení 3,3V. Obvod SPC4-2 pro maximální výkon potřebuje napájení 5V. Pokud by bylo napájení 3,3V, klesl by jeho výkon na polovinu. Proto je v zapojení použito dvou napájecích napětí. Jedno ze stabilizovaného zdroje na navržené desce 5V pro napájení budiče RS485 a SPC4-2. Druhé 3,3V je odebíráno z mikroprocesorového kitu.

Rozdílnost napájecího napětí u navzájem komunikujících obvodů nás přivede k problému, jak vyřešit úrovně na datové a adresové sběrnici. Pro adresovou sběrnici ponecháváme přímé propojení, protože komunikace je pouze jednosměrná směrem ze 3,3V na 5V, což by rozhodovací úroveň vstupních obvodů nemělo ovlivnit. Mezi datové sběrnice byl vložen obvod 74LCX245 [5], což je nízkonapájetový obousměrný transceiver s 5V tolerancí pro vstup a výstup. Směrem z SPC4-2 je úroveň snížena na 3,3V. V opačném směru

¹nakonec se obě součástky podařilo sehnat jako free samples z www.analog.com

je ponechána, takže dochází ke stejnemu případu jako na adresové sběrnici. Směr je přepínán na základě signálu read/write.

V kapitole 5.2.4 jsou podrobně popsány problémy, které vznikly propojením sběrnice na dvou logických úrovních, a návrh řešení.

4.1.2 Návrh desky plošného spoje

Pro návrh desky plošných spojů jsem použil software EAGLE v4.11 od firmy CadSoft. Pro tento program jsem se rozhodl, protože jsem ho již dříve používal pro osobní projekty a nemusel jsem se tak učit obsluhovat zcela novou aplikaci.

Všechny použité součástky - kromě pasivních - nejsou k dispozici ve standardních knihovnách, ani v knihovnách dostupných na internetu. Výrobce tento návrhový program nepodporuje, tudíž jsem si vytvořil vlastní knihovnu a všechny součástky nakreslil. Místo BCC kitu s procesorem jsem použil pouze tří dvouřadých konektorů, které jsem na DPS rozmištěl podle dokumentace k desce [6].

Plošný spoj je dvouvrstvý, při návrhu jsem využíval rad ze skript pro návrh plošných spojů [7]. Výsledná deska je navržena podle 4.třídy přesnosti.

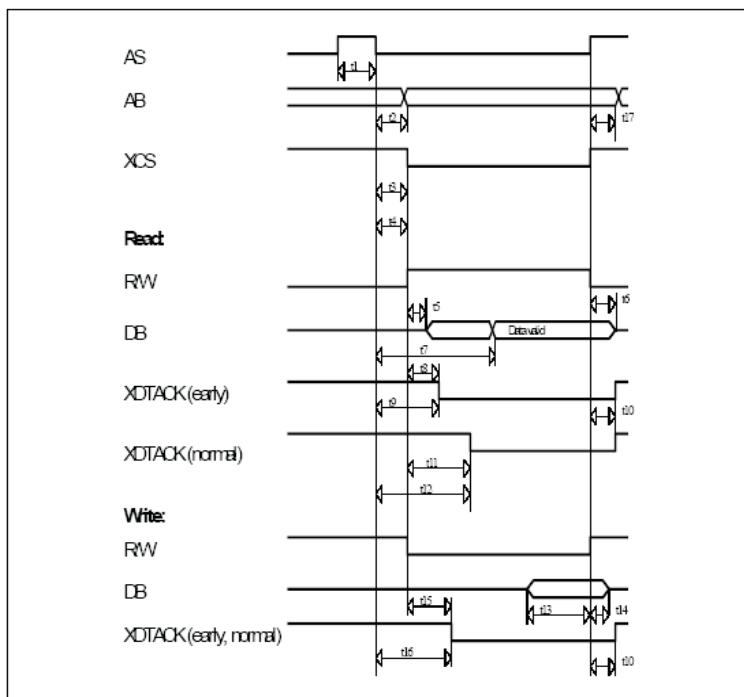
4.2 Výsledné schéma zapojení

Výsledné schéma zapojení je znázorněno na obrázku 7.1.

4.3 Výsledný návrh DPS

Navržená deska plošných spojů je znázorněna třemi obrázky:

- rozmištění součástek - obrázek 7.2,
- vrstva top - obrázek 7.3,
- vrstva bottom - obrázek 7.4.



Obrázek 4.1: Asynchronní časování sběrnice pro procesor Motorola

4.4 Časování a propojení signálů s M5235BCC

Pro návrh propojení jsem využil doporučeného zapojení v kapitole Processor Interface [3] pro Motorolu 68HC16. Popis průběhu signálů na sběrnici je popsán na obrázku 4.1 a v tabulce 4.1. Některé signály však na desce BCC nejsou obsaženy nebo se jmenují jinak. Propojení neshodně pojmenovaných signálů je popsáno v tabulce 4.2.

		Min.	Max.	Unit
t_1	Inactive time AS	10		ns
t_2	AS↓ until AB valid		$2 T_{SPC\ 4-2} - 20$ (30)	ns
t_3	AS↓ until XCS↓		$2 T_{SPC\ 4-2} - 10$ (15)	ns
t_4	AS↓ until R/W		$T_{SPC\ 4-2} - 10$ (15)	ns
t_5	XCS↓, R/W↑ until DB low resistance		18 (27)	ns
t_6	XCS↑, R/W↓ until DB high resistance		18 (27)	ns
t_7	Access time of AS↓ until DB valid		$3 T_{SPC\ 4-2} + 52$ (77)	ns
t_8	XCS↓, R/W↑ until XDTACK↓ (early)	0	$T_{SPC\ 4-2} + 18$ (27)	ns
t_9	AS↓ until XDTACK↓ (early)	$2 T_{SPC\ 4-2}$	$3 T_{SPC\ 4-2} + 18$ (27)	ns
t_{10}	AS↑ until XDTACK↑ (early, normal)	0	16 (24)	ns
t_{11}	XCS↓, RWX↑ until XDTACK↓ (normal)	$T_{SPC\ 4-2}$	$2 T_{SPC\ 4-2}$	ns
t_{12}	AS↓ until XDTACK↓ (normal)	$3 T_{SPC\ 4-2} + 4$ (6)	$4 T_{SPC\ 4-2} + 18$ (27)	ns
t_{13}	Setup time DB before AS↑	5 (8)		ns
t_{14}	Hold time DB after AS↑	12 (18)		ns
t_{15}	XCS↓ at XDTACK↓ (early, normal)		18 (27)	ns
t_{16}	AS↓ at XDTACK↓ (early, normal)	$T_{SPC\ 4-2}$	$2 T_{SPC\ 4-2} + 18$ (27)	ns
t_{17}	Hold time AB after AS↑		10 (15)	ns

Tabulka 4.1: Asynchronní časování sběrnice pro procesor Motorola

68HC16	M5235BCC
AS XDTACK(normal)	\overline{TIP} \overline{TA}

Tabulka 4.2: Neshodně pojmenované signály

Kapitola 5

Návrh programu pro obsluhu SPC4-2

V této kapitole se budu věnovat z části návrhu programu, ale především popisu a řešení vzniklých problémů při snaze oživit desku a komunikaci procesoru s SPC4.

5.1 Vývojové prostředí

Pro návrh programu pro M5235 jsem použil vývojové prostředí CodeWarrior od firmy Metrowerks. Tato verze je komerční, a tudíž velice drahá, proto jsem měl pouze 30-ti denní free verzi. Tento software má pro začátečníky výhodu grafického prostředí, které usnadňuje obsluhu aplikace. V novém projektu jsou již importovány důležité soubory (stdio.h, vectors.s, ...). Předdefinované a importované nástroje a soubory jsou pro začátečníka výhodou, avšak neznáme přesně do hloubky všechny detaily, a proto mohou vzniknout problémy, které se poté hůře řeší a odstraňují jejich chyby.

Pro další vývoj softwaru jsem se proto rozhodl seznámit se a využít free opensource nástrojů pod operační systém linux.

5.2 Základní práce s procesorem

V grafickém prostředí lze nový projekt zkomplikovat a prostřednictvím BDM debugovat. První program nám vypíše na konzoli známé klišé z učebnic programování ”Hello World!”.

Úlohu obsluhy SPC4-2 jsem si rozdělil na několik podúkolů, které jsem postupně implementoval a každý odladil zvlášť.

5.2.1 Ovládání výstupního pinu

Jako první krůček jsem zvolil ovládání pinu na GPIO portu. Musíme tedy v příslušných registrech nastavit, že chceme, aby pin patřil GPIO, protože může mít i jinou funkci od periferních zařízení. V dalším registru nastavíme směr input/output. Nakonec do registru, který obsahuje obraz portu, zapíšeme požadovanou výstupní hodnotu. Toto lze v debug režimu pozorovat a pokud chceme, tak i měnit pomocí grafického nástroje. Změny, které byly způsobeny v předcházejícím hodinovém cyklu, jsou zvýrazněny červenou barvou.

```
/* Pin assignments for ports DATAH and DATAL

.
.
.

Pin D[1]      : GPIO input
Pin D[0]      : GPIO output
*/
MCF_PDDR_DATAH = 0;
MCF_PDDR_DATAL = MCF_PDDR_DATAL_DDDATAL0;

.
.
.

MCF_PODR_DATAL = 0x01; // set pin D0
.
.
.

MCF_PODR_DATAL = 0x00; // reset pin D0
.
.
```

5.2.2 Obsluha přerušení

Tento procesor má více než sto zdrojů přerušení. K rozpoznání a obsluze slouží složitý mechanismus, který mimo jiné využívá tabulky přerušení, která je obsažena v souboru vectors.s. Zde jsem našel požadované řádky pro obsluhu stisknutí tlačítka ABORT na KITu desky, které je připojeno na $\overline{IRQ7}$ a dále pro dva piny $\overline{IRQ1}$ a $\overline{IRQ2}$. V této tabulce jsem přepsal odkaz na obecný handler a nadefinoval si pro každé přerušení vlastní. Každý vypsal, které přerušení bylo zdrojem a kolikrát bylo stisknuto. Přerušení mají různé priority a různé úrovně, některé se dají programově měnit, jiné jsou dány pevně typem přerušení. V mé případě je priorita dána pevně uprostřed všech priorit. Úroveň je pro každý pin jiná, pro $\overline{IRQ7} = 7$, pro $\overline{IRQ1} = 1$. Prioritou a úrovní se lze jednoduše maskovat zdroje přerušení. Ve status registru je uložena hodnota nejmenší priority a úrovně přerušení, která může v daný moment vyvolat obsluhu. Proto jsem musel tuto hodnotu snížit, aby se provedla obsluha $\overline{IRQ2}$ a $\overline{IRQ1}$. Při ladění obsluhy přerušení se vyskytl problém, že se nechtěla spustit žádná obsluha přerušení a vůbec nedošlo ke skoku do tabulky přerušení. Problém byl dán právě předdefinovaným souborem init, který zajišťoval překopírování tabulky do paměti při spuštění od řádku 0 až 255. Pokud jsem změnil kopírování od řádku 64 dále obsluha již fungovala, přerušení 0 až 63 jsou interní přerušení - například dělení nulou.

```
/* vectors.s */
.

vector40: .long _irq_handler
vector41: .long _irq_handler_irq1
vector42: .long _irq_handler_irq2
vector47: .long _irq_handler_irq7
.

/* int_handlers.c */
.

__interrupt__
void irq_handler_irq1(void)
```

```

{
    akt_irq1();
}

__interrupt__
void irq_handler_irq7(void)
{
    stisk_abort();
}

/*
 * main.c */
.

void my_init(void)
{
    MCF_EPORT_EPPAR = 0x8028; //falling edge triggered irq 1,2,7
    MCF_EPORT_EPIER = 0x86;   //edge port interrupt enable pin 1,2,7
    MCF_INTC0_IMRL = 0xFFFFFFF78; //enable irq 1,2,7
    MCF_INTC1_IMRL = 0xFFFFFFFF;
.

void akt_irq1(void)
{
    printf("Aktivovano irq1. Pocet aktivaci:
           %i\n", ++pocet_stisknuti_t1);
    MCF_EPORT_EPFR = MCF_EPORT_EPFR_EPFI;
}
.
```

5.2.3 Nastavení chip select a obsluha sběrnice

Dalším krokem bylo nastavení komunikační sběrnice mezi SPC4-2 a M5235. Zde se naskytlo mnoho problémů, které se nakonec podařilo vyřešit. Každý chip select (\overline{CS}) má pro své nastavení tři registry.

CSCR - Chip Select Control Register, který umožňuje nastavit šířku datové sběrnice 32,16,8 bitů a speciální režimy přenosu. Například pokud chceme využít signálu \overline{TA} - Transfer Acknowledge pro potvrzení přijetí dat. Procesor je 32 bitový, z KITu má vyvedených horních 8 bitů, proto jsem nastavil 8 bitů datovou sběrnici a externí \overline{TA} prozatím zakázal. Čas pro generování vnitřního \overline{TA} jsem nastavil na 14 cyklů. Ostatní nastavení jsem ponechal ve výchozím stavu.

Další regist CSAR - Chip Select Address Register určuje bázovou adresu, od které bude \overline{CS} aktivní.

Poslední CSMR - Chip Select Mask Register nastavujeme velikost bloku, pro který je \overline{CS} aktivní a zároveň můžeme určit blok, který bude pouze pro čtení. Nastavil jsem bázovou adresu do volného prostoru a blok zvolil nejmenší možný = 64kB.

Ostatní řídící signály \overline{OE} , \overline{TS} , \overline{TIP} , \overline{TSIZ} , R/\overline{W} jsou generovány automaticky a nic dalšího není třeba nastavovat.

Procesorový KIT jsem zatím k mému připravku nepřipojoval. Do nekonečné smyčky jsem umístil instrukci čtení a zápisu z adresy příslušející SPC4-2, abych mohl proměřit všechny signály, které jsou využity pro přenos. V řídících signálech sběrnice \overline{OE} , \overline{CS} , \overline{TS} se však objevovaly úrovňové špičky, jak je znázorněno na obrázku 5.1. Nejprve jsem si myslел, že se jedná o rušení, protože se však nevyskytovaly u signálu R/\overline{W} , který byl naprosto bez rušení, došel jsem k závěru, že toto nemůže být způsobeno rušením na sběrnici. Začal jsem proto zkoušet nejrůznější nastavení registru CSCR. Jedním bitem pro cyklické zpracování dat ze sběrnice jsem odstranil dvě ze tří špiček. Avšak nastavoval jsem funkci, kterou jsem jistě nechtěl použít. Prostřední špičku se nepodařilo ničím odstranit.

Vyřešení předcházejícího problému bylo více než jednoduché a ukazuje, že procesor je velice dobře vybaven pro komunikaci po sběrnici. Protože se jedná o přístup do vnější paměti, zapisují a čtou se data pomocí pointerů. Na úplném začátku ”pokusů” s chip selectem jsem si vytvořil ukazatel `#define SPC_CB(*(volatile unsigned int *) (SPC_BASE_ADDR+0x018))`.

Tento procesor je však 32 bitový a datový typ int je v něm také 32 bitový, proto, když měl procesor nastavenou šířku sběrnice na 8 bitů, sám data rozdělil do čtyř bytů, které na sběrnici vysílal bezprostředně po sobě. V signálech \overline{OE} , \overline{CS} , \overline{TS} byly krátké špičky, které na osciloskopu dosahovaly přibližně 80% logické úrovně 1. Změnil jsem tedy ukazatel na typ char
`#define SPC_CB (*(volatile char *) (SPC_BASE_ADDR+0x018)).`

```
#include <stdio.h>
#include "m523xevb.h"
#include "mcf5xxx.h"

#define SPC_BASE_ADDR 0xa0000000
#define SPC_CB      (*(volatile char *) (SPC_BASE_ADDR+0x018))
#define SPC_RSA     (*(volatile char *) (SPC_BASE_ADDR+0x019))

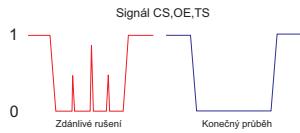
.

void pulse_cs1(void) // rucni ovladani CS1
{
    MCF_GPIO_PDDR_CS |= 1<<1;
    MCF_GPIO_PAR_CS &= ~MCF_GPIO_PAR_CS_PAR_CS1;
    MCF_GPIO_PODR_CS &= ~(1<<1);
    printf("Pulse CS1\r\n");
    MCF_GPIO_PODR_CS |= 1<<1;
}

.

MCF_CS_CSAR1 = SPC_BASE_ADDR>>16;
MCF_CS_CSMR1 = 0x00000001;
/* 15-14    13-10    9      8      7-6    5      4      3      2-0 */
/* SRWS      IWS      -      AA      PS      BEM    BSTR   BSTW   SWWS*/
MCF_CS_CSCR1=
=(0<<14)|(14<<10)|(1<<8)|(1<<6)|(0<<5)|(0<<4)|(0<<3)|0;

.
```



Obrázek 5.1: Náčrt průběhů z osciloskopu

```

int main()
{
    char m1 = 0, m2 = 0;
    my_init();
    for(;;)
    {
        SPC_M1 = 0xAA;
        prodleva();
        m1 = SPC_M1;
        prodleva();
    }
}

```

Tímto byl problém zdánlivého rušení na sběrnici vyřešen a všechny průběhy signálu na osciloskopu odpovídaly požadavkům v dokumentaci k SPC4-2.

5.2.4 Připojení k přípravku s SPC4-2

Poté, co průběh všech signálů, které byly použity pro komunikaci s SPC4-2 při měření osciloskopem, odpovídal dokumentaci k SPC4-2, připojil jsem KIT s procesorem k mému přípravku s SPC4-2.

Prvním cílem bylo zapsat data a následně je přečíst z paměti v SPC4-2. Toto se však vůbec nedařilo. Debugger hlásil chyby typu bus error, address error apod. Rozhodl jsem se tedy zjistit následujícím postupem, co problém způsobuje. Procesorový modul je k mé desce připojen přes dutinkový a kolíkový konektor. Vložil jsem tedy mezi tuto dvojci ještě jednu řadu

konektorů. Nikoliv však celou, ale pouze po částech. Připojil jsem adresovou a datovou sběrnici a postupně další řídící signály. Většinou však komunikace právě až po připojení posledního řídícího signálu zkolaovala a nezáleželo, který signál to byl. Poté jsem odpojil datovou sběrnici a všechny ostatní signály připojil. Komunikace se neukončila chybou. Z tohoto mně vyplynulo, že chyba není způsobena jedním signálem, jak jsem se domníval na začátku, ale všemi signály najednou. Z chyby address error, kterou debugger hlásil po připojení datové sběrnice hned na první instrukci ještě v inicializaci nebo poté, když jsem k pozastavenému programu připojil další signály a následně pokračoval, jsem dospěl k závěru, že SPC4-2 přistupuje na sběrnici v době, kdy nemá. Po hodinách bádání, cím by mohl být problém způsoben, mě napadlo, že SPC4-2 možná není ani v činnosti a řídící signály jeho činnost neovlivňují.

Možnost, že dvě úrovně napájení mohou způsobovat potíže, se následně potvrdila. Na úplném začátku práce při výběru procesoru jsme se dlouho nemohli rozhodnout pro konkrétní typ. První procesor od firmy Siemens měl napájení 5V, další měly 3,3V. Úvaha, že vstupním obvodům při napájení 5V nebude napětí 3,3V činit problém, zvláště, když obvod může být také napájen 3,3V, byla mylná. Když jsem znova prozkoumal statické charakteristiky obvodu, čekalo mě nemilé překvapení, které odpovídalo na otázku nefunkčnosti současného přípravku.

Při pohledu do tabulky 5.1 je zřejmé, že vstupní obvod je vybaven Schmittovým klopným obvodem a pro napájení 5V se do úrovně log. 1 překládí při napětí 3,7V, což použitým napájením 3,3V nedosáhneme. Z tohoto vyplývá, že tento přípravek je nepoužitelný. Navrhl jsem tedy následující řešení pro oživení komunikace. Změním napájení SPC4-2 na 3,3V, které je stejně jako napájecí napětí z adaptéra vyvedeno na piny KITu. Snížení napájecího napětí by mělo ovlivnit pouze výkon SPC4-2, který nyní není zapotřebí. Při přepájení napájecích drátků jsem však omylem udělal hloupou chybu, když jsem odpájal stabilizátor na 5V a na plošku ”výstupu” připájal drátek s 3,3V z modulu. Při pájení jsem ale otočil desku plošného spoje podle špatné osy a došlo tak

Parameters	NAME	MIN.	Type	MAX.	Unit
Threshold voltage 0 level Schmitt trigger at 3.3 V	V+	0		1.8	V
Threshold voltage 1 level Schmitt trigger at 3.3 V	V-	0.7		VDD	V
Threshold voltage 0 level Schmitt trigger at 5.0 V	V+	0		3.7	V
Threshold voltage 1 level Schmitt trigger at 5.0 V	V-	1.5		VDD	V
Input leakage current	II			1	uA
Output leakage current	IOZ			10	uA
Output current 0 level 4 mA cell	IOL	4		(1)	mA
Output current 1 level 4 mA cell	IOH	-4		(2)	mA
Output current 0 level 10 mA cell	IOL	10		(1)	mA
Output current 1 level 10 mA cell	IOH	-10		(2)	mA
Short-circuit current	IOS			TBD(3)	mA
Input capacitance	CIN		5		pF
Output capacitance	COUT		5		pF
I/O capacity	CI/O		5		pF

Tabulka 5.1: DC specifikace pinů

k záměně vstupní a výstupní plošky stabilizátoru. Po zapnutí se tedy spojilo napájení přibližně 9V z adaptéru a 3,3V. Tuto skutečnost jsem zjistil okamžitě, když svítila LED error na modulu a obvod hned vypnul. Procesor má podle všeho přepěťovou ochranu, protože po této události jsem ho přepnul do DB režimu a na terminál po sériové lince přicházela data jako dříve. Ale modul BDM používaný pro nahrávání programu a debugger tuto ochranu nemá, a tak byl zničen. Nový modul je však stále pouze objednaný a na práci nyní nelze pokračovat. Je škoda, že ke zničení došlo již tak blízko k cíli vyřešení úlohy.

Mezitím jsem navrženou desku lehce upravil pro další řešení, které spočívá v prozatimním snížení napájecího napětí na 3,3V, aby se zde nevyskytoval problém dvou napájecích napětí. První úpravou je snížení kmitočtu oscilátoru na maximálně 20MHz. Další je odstranění obvodu s tolerančním vstupem na datové sběrnici. Jediné napájení, které musí zůstat na úrovni 5V, je budič sběrnice Profibus. DC-DC měnič z 3,3V na 5V se mně prozatím nepodařilo sehnat, avšak toto napětí je od ostatního galvanicky odděleno, proto můžeme prozatím použít externí zdroj.

Kapitola 6

Závěr

V této kapitole krátce shrnu výsledky mé dosavadní práce a další zamýšlené části úloh k rozšíření.

Úplně prvním úkolem práce bylo seznámit se s protokolem Profibus DP, k čemuž jsem čerpal z literatury [2, 1]. Dalším krokem bylo nastudovat dokumentaci k obvodu SPC4-2. Poté jsme s vedoucím práce vybrali procesor. Po částečném seznámení se s procesorem jsem navrhl schéma zapojení a postupně jsme jej doladovali do finální podoby. Pak jsem navrhl dvoustrannou desku plošných spojů, kterou jsem nechal vyhotovit ve firmě Pragoboard. Vyrobenou desku jsem osadil součástkami a snažil se oživit komunikaci mezi SPC4-2 a procesorem M5235, což se mně nakonec nepodařilo, jak je popsáno dříve. Proto ani nedošlo k naprogramování jednoduchého stavového automatu verze V0.

Po získání nového BDM modulu budu pokračovat v oživení komunikace, naprogramování stavového automatu verze V0, následně dalších verzí V1 a rozšíření o bezpečnou komunikaci. Dále bych chtěl přejít na vývojové prostředí, které není omezené 30-ti denní licencí. Finální podoba tohoto projektu, v podobě diplomové práce, je zařízení Profibus DP slave pro polohování motorů.

Literatura

- [1] *Normative parts of Profibus FMS,DP and PA according to EN50170 volume 2.* PROFIBUS Nutzerorganisation e.V., 1998.
- [2] Manfred Popp. *The New Rapid Way to PROFIBUS DP.* PROFIBUS Nutzerorganisation e.V., Karlsruhe, 2003.
- [3] www.siemens.de. Profibus controller spc4-2 manual. 2003.
- [4] www.analog.com. Adm2486.pdf datasheet. 2006.
- [5] www.fairchildsemi.com. 74lcx245.pdf datasheet. 2005.
- [6] www.axman.com. Cmm-5235 user manual v1.2.
- [7] V. Záhlava. *Metodika návrhu plošných spojů.* ČVUT, Praha, 2002.
- [8] T. Oetiker, H. Partl, I. Hyna, E. Schlegl, M. Kočer, and P. Sýkora. *Ne příliš stručný úvod do systému LATEX.* Verze překladu CZ-0.9-Beta, 1998.
- [9] Pavel Herout. *Učebnice jazyka C.* Kopp, České Budějovice, 2001.
- [10] www.freescale.com. Mcf5235rm.pdf reference manual. 2004.
- [11] www.analog.com. Adm803-9-10.pdf datasheet. 2006.

Kapitola 7

Příloha

V této kapitole jsou návrhy plošných spojů a schémat.

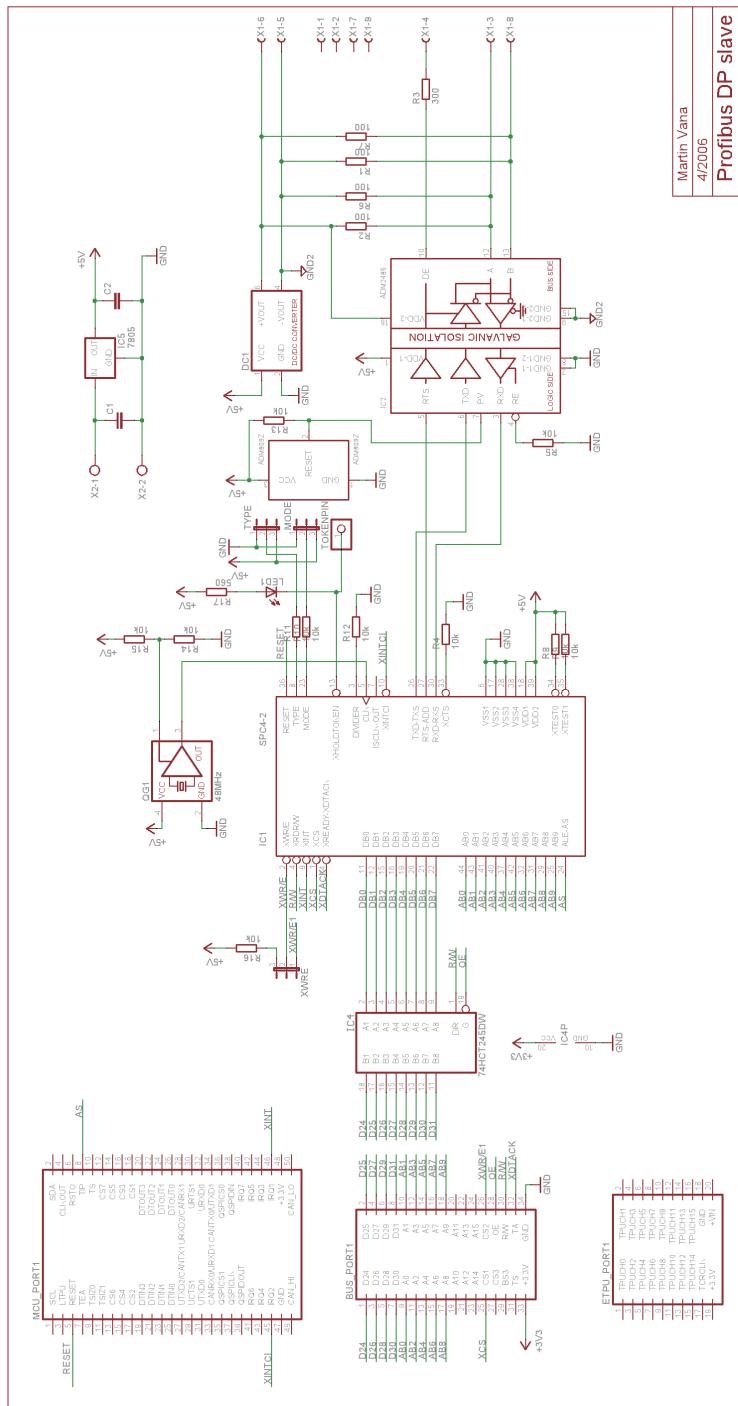
Obrázek 7.1 - původní schéma

Obrázek 7.2 - rozmístění součástek

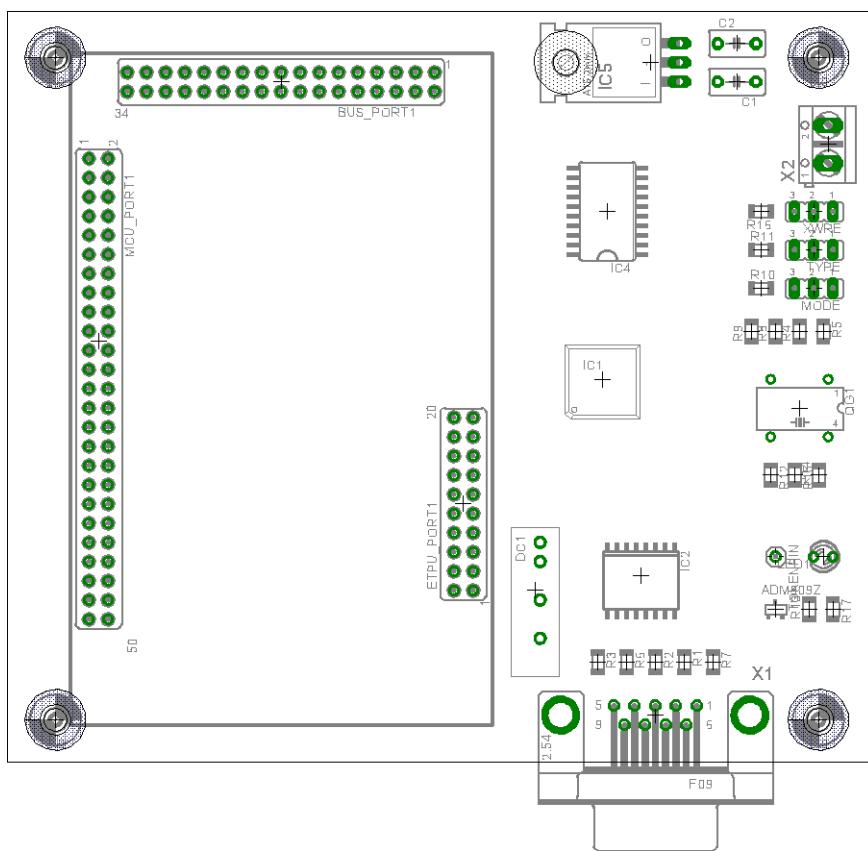
Obrázek 7.3 - spoje vrstvy top

Obrázek 7.4 - spoje vrstvy bottom

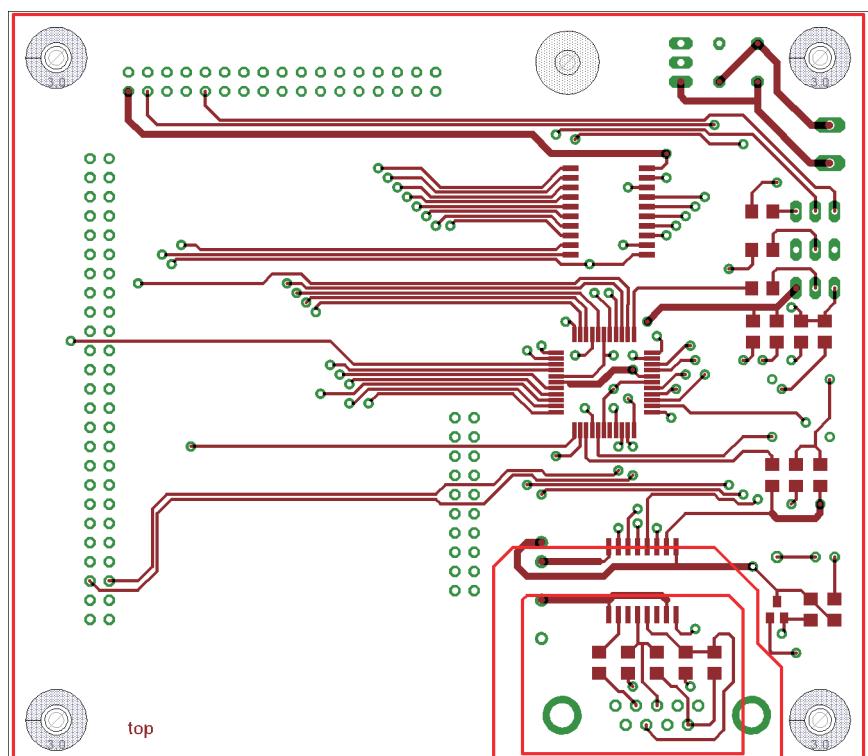
Obrázek 7.5 - schéma s řešením problému dvou logických úrovní



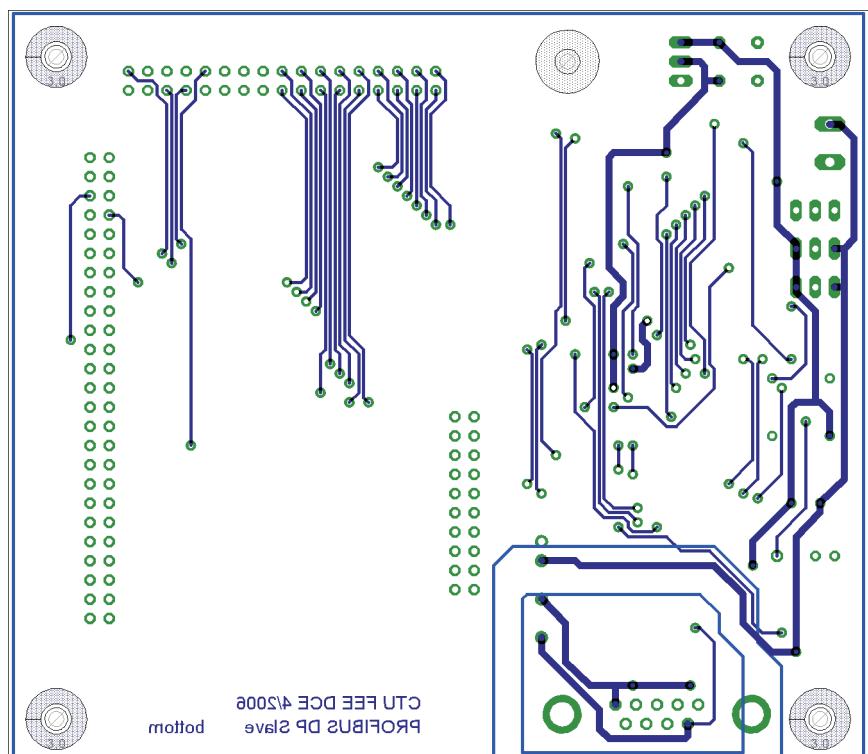
Obrázek 7.1: Navržené schéma zapojení



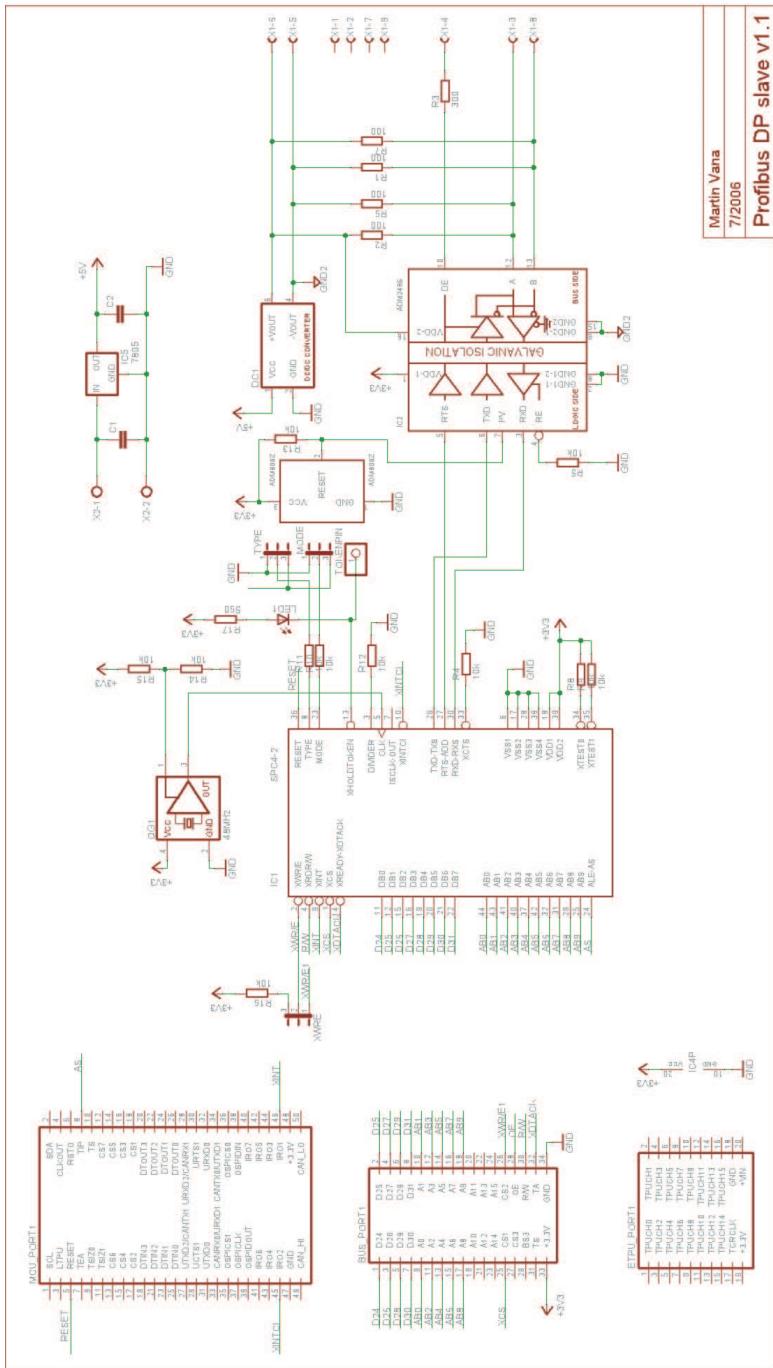
Obrázek 7.2: Rozmístění součástek pro osazování



Obrázek 7.3: Spoje vrstvy Top



Obrázek 7.4: Spoje vrstvy Bottom



Obrázek 7.5: Navržená úprava zapojení