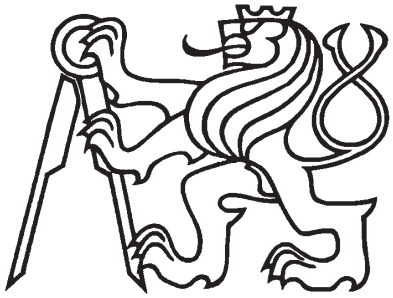
České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra řídicí techniky



Diplomová práce

Návrh a realizace řídicí jednotky servomotorů pro model vrtulníku

Květen 2006 Ota Herm

Prohlášení

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

V Praze dne 31. května 2006 _____

Poděkování

Rád bych poděkoval především vedoucímu diplomové práce Ing. Ondřeji Špinkovi za vytvoření výborných podmínek pro práci na tomto projektu, sestře Ing. Haně Hermové za pečlivou jazykovou korekturu textu a svým rodičům za podporu během studia. Dále bych chtěl poděkovat Ing. Michalu Sojkovi za podporu při práci s prostředím OMK i použitým mikrokontrolérem.

Abstrakt

Diplomová práce se zabývá návrhem a realizací jednotky řízení servomotorů. Je dílčí součástí komplexního projektu, zabývajícího se vývojem autonomně řízeného vrtulníku. Vyvinutá jednotka je řízena pomocí jednočipového mikrokontroléru Renesas H8S2638. Teoretická část se zabývá tímto mikrokontrolérem. Praktická část se zabývá implementací měření signálu modelářských servomotorů, komunikací po sběrnici CAN, návrhem použitého hardwaru, tedy popisuje vyvinuté řešení.

Abstract

This thesis deals with design and construction of a servo control unit. It is a part of a coplex project, whose goal is to design a distributed control system for an unmanned autonomous rotorcraft. The servo control unit is built around the Renesas H8S2638 microcontroller. The theoretical part of this thesis describes the servomotor control, CAN bus communication and HW/SW tools that were used. The other part doccuments our solution.

České vysoké učení technické v Praze - Fakulta elektrotechnická

Katedra řídicí techniky *Školní rok:* 2004/2005

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Ota Herm

Obor: Technická kybernetika

Název tématu: Návrh a realizace řídicí jednotky servomotorů pro model vrtulníku

Zásady pro vypracování:

1. Prostudujte existující specifikaci a rámcový návrh řídicí jednotky servomotorů. Seznamte se s mikrokontrolérem HITACHI H8S2638F a sběrnicí CAN.
2. Navrhněte a realizujte HW řídicí jednotky.
3. Vytvořte základní programové vybavení řídicí jednotky.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí diplomové práce: Ing. Ondřej Špinka

Termín zadání diplomové práce: zimní semestr 2004/2005

Termín odevzdání diplomové práce: leden 2006

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

v zast.
prof. Ing. Vladimír Kučera, DrSc.
děkan

V Praze dne 07.03.2005

Obsah

Seznam obrázků

Kapitola 1

Úvod

Tato diplomová práce se zabývá vývojem jednoho z klíčových elementů řídicího systému pro malý autonomní vrtulník, vznikající na Katedře řídicí techniky. Jeho účelem je navrhnout disribuovaný řídicí systém.

Obrázek 1.1: Vrtulník při přípravě na let

Složitost problému autonomního řízení vrtulníku spočívá především v nelinearitě jeho odezev, velmi rychlými reakcemi a nutnosti kvalitně kompenzovat vliv okolí. Zároveň je nutné klást zásadní důraz na bezpečnost a spolehlivost, protože každé selhání systému může vést ke zničení modelu a vzniku nezanedbatelných materiálních škod.

Projekt autonomního vrtulníku vzniká pod vedením Ing. Ondřeje Špinky. Kvůli své komplexnosti je systém rozdělen do několika bloků, vznikajících v rámci diplomových a bakalářských prací. Jednotka řízení servomotorů je tématem tohoto dokumentu. Hlavním řídicím počítačem se zabývá práce Jiřího Novotného. Software pro vizualizaci a sběr dat je téma bakalářské práce Miroslava Hájka. Dekompozice problému na více podúkolů, řešených v jednotlivých pracech, znamenala možnost vyřešit poměrně složitý problém. Navíc všem zúčastněným přinesla mnoho zkušeností z týmové práce, což je také neocenitelný výsledek.

V úvodní kapitole je popsána motivace projektu a organizace dokumentu. Kapitola 2 popisuje vlastní model vrtulníku, způsob řízení a koncepce řídicího systému. V kapitole 3 je popsán použitý mikrokontrolér, jsou však zmíněny převážně vlastnosti podstatné pro jeho využití v této práci. Kompletní popis lze samozřejmě najít v dokumentaci od výrobce. V kapitole 4 jsou zmíněny použité vývojové nástroje. Konstrukční vyřešení úlohy je popsáno v kapitole 2. Ta je rozdělena na dvě hlavní části, 2.1 se věnuje popisu softwarového vybavení jednotky řízení servomotorů a 2.2 se zabývá jejím hardwarovým provedením.

Kapitola 2

Popis soustavy

2.1 Model vrtulníku

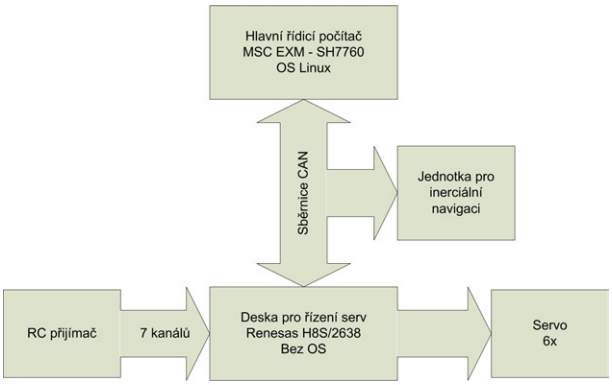
Obrázek 2.1: Model vrtulníku sst-eagle Freya

Jako model vrtulníku je využita stavebnice sst-eagle Freya od japonské firmy HIROBO. Jedná se o model poháněný spalovacím motorem s objemem 10 ccm, určeným pro speciální modelářské palivo na bázi methylalkoholu. Rozměry stroje jsou 1 375 x 200 x 453 mm. Rotor má průměr 1 600 mm. S kompletním osazením elektronikou model váží přibližně 7 kg.

Obrázek 2.2: Vysílač modelářské RC soupravy

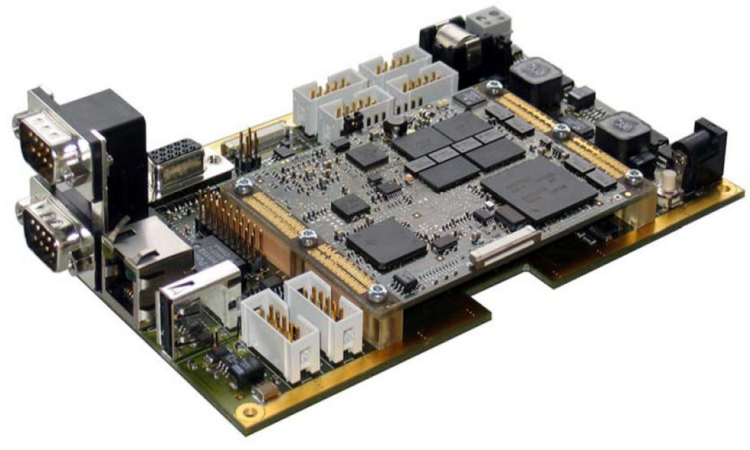
V původní podobě je model řízen radiem pomocí standardní RC soupravy (vysílač viz obrázek 2.2). Tato možnost zůstala v řídicím systému zachována. Je klíčová jednak pro vývoj řídicích algoritmů, aby bylo možné provést identifikaci dynamických parametrů modelu a vytvořit matematický model. Její úloha však zůstává zásadní i během dalších letů, protože je nutné stále kontrolovat činnost modelu a v případě jakýchkoliv problémů zasáhnout za účelem zabránění škodám. Tato metoda je dále v textu zmiňována jako *manuální řízení*. Jejím opakem je *automatické řízení*, které je cílem této i dalších souvisejících prací.

2.2 Řídicí systém



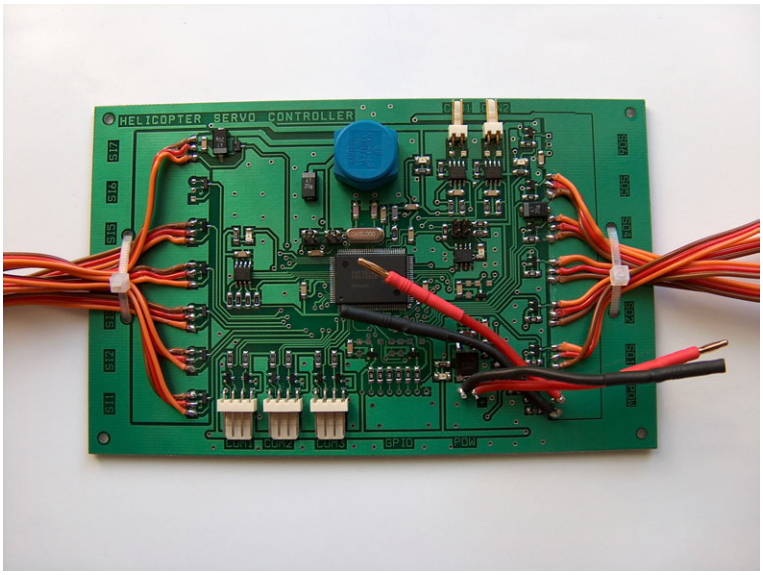
Obrázek 2.3: Blokové schéma řídicího systému

Vlastní řídicí systém je vzhledem ke své komplexnosti realizován jako distribuovaný systém rozdělený do tří bloků. Jejich struktura je naznačena na obrázku 2.3. Základními parametry, které bylo nutno splnit, jsou jednak požadavek na vysoký výkon řídicího systému, tak i požadavky na vysokou bezpečnost systému. Proto je systém dekomponován na tři hlavní části propojené sběrnicí CAN. Jedná se o řídicí počítač s mikroprocesorem SH7760 a operačním systémem Linux, jednotku řízení servomotorů a jednotku inerciální navigace.



Obrázek 2.4: Hlavní řídicí počítač MSC EXM32

Úkol hlavního řídicího počítače, zobrazeného na obrázku 2.4, spočívá především ve výpočtech řídicího algoritmu, komunikaci s připojenými zařízeními, shromažďování a analýze měřených dat i v bezdrátové komunikaci s pozemní stanicí. Protože všechny tyto úlohy dohromady vyžadují poměrně vysoký výpočetní výkon, byla zvolena tato platforma na bázi třicetidvoubitového mikroprocesoru Renesas SH7760.



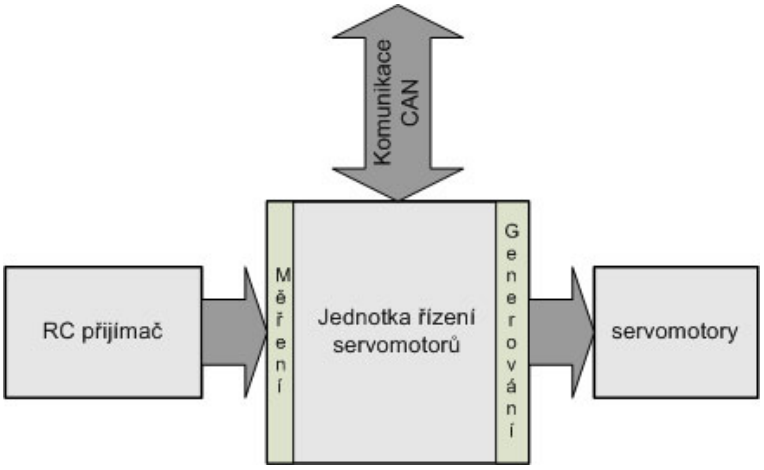
Obrázek 2.5: Jednotka řízení servomotorů

Vlastní připojení k akčním členům vrtulníku zajišťuje jednotka pro řízení servomotorů. Její návrh a realizace je obsahem této diplomové práce, podrobný popis je v následujících kapitolách.

Další nedílnou součástí řídicího systému je jednotka inerciální navigace. To je funkční blok připojený po sběrnici CAN, který zajišťuje měření zrychlení a úhlové rychlosti ve třech osách. Celkově se tedy jedná o inerciální navigační jednotku se šesti stupni volnosti. Tato jednotka využívá senzory typu MEMS, což předurčuje její vlastnosti. Pro zpětnovazební řízení modelu podle měřených veličin by měla poskytnout dostatečně kvalitní údaje. Pro určování polohy dvojí integrací zrychlení se však nehodí, protože chyby měření těchto zrychlení i rotací velmi rychle odhadovanou polohu znehodnotí. Tento fakt je ještě zhoršen silnými vibracemi modelu vrtulníku, které signál z akcelerometrů dále poškozují. V projektu je využita jednotka inerciální navigace typu BP3010, více informací o ní lze nalézt v [8].

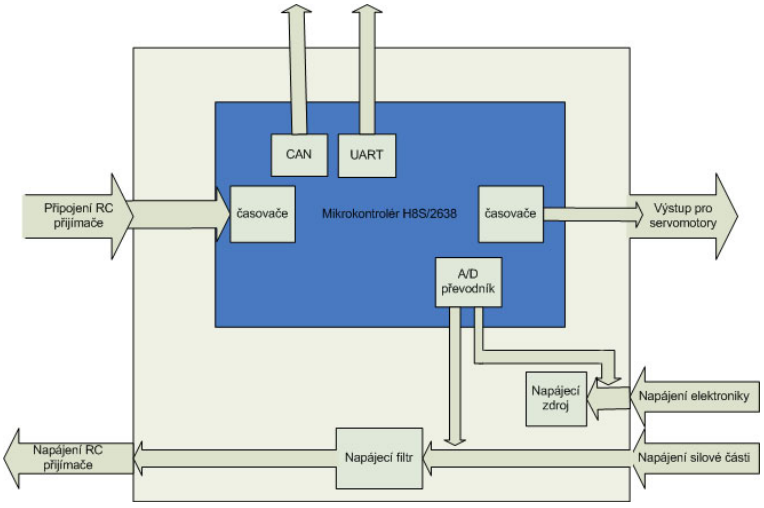
2.3 Úloha jednotky řízení servomotorů

Zásadní úlohou jednotky řízení servomotorů je bezpečně zajistit možnost ručního řízení vrtulníku, aby i v případě selhání hlavního řídicího počítače nedošlo k havárii a destrukci modelu. Proto je zde kladen velký důraz na spolehlivost a robustnost provedení. Dále též jednotka zajišťuje dodržení přesného časování řídicích signálů, protože hlavní řídicí počítač není vybaven vhodnými periferiemi k tomuto účelu, případně by jejich využití pod OS Linux bylo komplikované.



Obrázek 2.6: Nástin funkce jednotky řízení servomotorů

Jednotka je řízená jednočipovým mikrokontrolérem Renesas H8S/2638. Přijímá signály z modelářské RC soupravy, použité pro řízení v manuálním režimu, a generuje výstupní signály pro servomotory. Dále komunikuje s hlavním řídicím počítačem po sběrnici CAN, po níž se přenášejí požadované hodnoty polohy servomotorů a ostatní měřená data. Doplňkovou úlohou této jednotky je měření napětí akumulátorů, použitých k napájení modelu. Blokové schema jednotky je na obrázku 2.7.



Obrázek 2.7: Blokové schéma jednotky řízení servomotorů

2.4 Popis funkce jednotky řízení servomotorů

2.4.1 Synchronizace měření

Jednotka zajišťuje především měření dat z RC přijímače a generování výstupního signálu pro servomotory. Zde je nutné robustně zajistit časovou synchronizaci, a to jak mezi jednotlivými činnostmi této desky (měření a výstup), tak vzájemně s dalšími částmi systému (inerciální jednotka). Synchronizace je řešena pomocí zprávy na sběrnici CAN, která neobsahuje žádná data a má nejvyšší prioritu, nemůže tedy být za normálních podmínek nijak zpožděna. Pak nebude docházet k nepravidelným fázovým posunům mezi jednotlivými měřeními, které by kazily shodu s matematickým modelem soustavy. Od okamžiku přijetí této zprávy tedy jednotka řízení servomotorů odvozuje okamžik provedení důležitých činností.

2.4.2 Režim automat/ručně a maska serv

Model vrtulníku může být řízen buď plně automaticky hlavním řídicím počítačem, nebo je možné ho ovládat ručně pomocí RC soupravy. Dále lze automatické řízení redukovat jen na některé kanály a zbytek ovládat ručně. Účel tohoto režimu spočívá v usnadnění tvorby matematického modelu a návrhu regulátoru. Pak je možné do některých kanálů posílat speciálně generovaný identifikační signál a pomocí zbylých, ručně řízených, udržet model vrtulníku v letové poloze ve vzduchu.

Maska serv je číslo, určené k rozlišení, které servomotory se mají v automatickém režimu ovládat z počítače a které ručně. Jedná se o číslo reprezentované jako byte, kde každý bit odpovídá jednomu servu. Při hodnotě 1 je servomotor v automatickém režimu řízen z počítače, při hodnotě 0 je řízen vždy z RC soupravy nezávisle na volbě režimu automat/ručně.

Režim automat/ručně je přepínán pomocí RC soupravy signálem pro servo připojené na vstup desky řízení serv číslo 7 (označen SI7). V případě přijetí pulzu užšího než 1,3 ms se nastaví automatický režim, v případě pulzu širšího než 1,7 ms se nastaví manuální režim. V oblasti mezi 1,3 až 1,7 ms se nachází hysterezní pásmo a je ponechán původní stav. Tento signál dobře odpovídá výchylce páky na RC soupravě, kdy např. vychýlení vpravo vyvolá přepnutí do manuálního režimu, vlevo přepne do automatického a ve střední poloze nedochází ke změně. Po startu je vždy zvolen manuální režim. V případě, že tento signál vůbec není připojen, nebo je v hysterezním pásmu, tak je možné přepnutí mezi automatickým a manuálním režimem provést také ze servisního sériového rozhraní. Signál z RC soupravy má však vždy přednost. Pro masku platí to samé, také ji lze nastavit přes servisní rozhraní, ale nastavení z hlavního řídicího počítače má přednost.

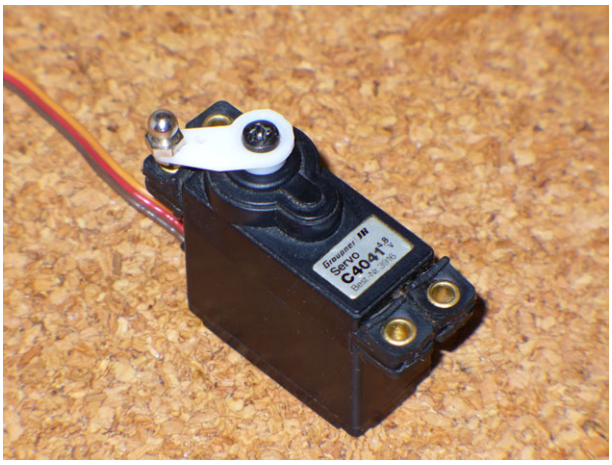
2.4.3 Nastavení/čtení hodnot serv

Požadované hodnoty serv se v automatickém režimu zasílají pomocí CAN sběrnice z hlavního řídicího počítače. Aby se zamezilo nepravidelným zpožděním, je jejich zápis synchronizován zprávou s ID 20 (kapitola 2.4.1). V případě nevyužití sběrnice CAN je možné hodnoty servomotorů zadávat z sériového rozhraní (kapitola 2.4.7).

2.4.4 Měření palubních napětí

Jako doplňkovou činnost jednotka měří napětí akumulátorů. Ty jsou rozdělené do dvou skupin, pro napájení akčních členů a pro elektroniku. Napětí na obou akumulátorech jsou měřena s rozlišením na 0,1V. Jedná se však pouze o orientační měření, aby se snížilo riziko havárie kvůli vybití akumulátorů. Interval měření a odesílání je 1 s. Pro možnost budoucího rozšíření jsou připraveny ještě 2 další měřící kanály. Na ně lze připojit další senzory, například tlakoměr nebo elektronický kompas. Na vstupech jsou připraveny pájecí plošky pro odporový dělič a RC filtr, takže lze dobře přizpůsobit rozsah měření i dynamiku vyhodnocování signálu.

2.4.5 Řízení modelářských servomotorů

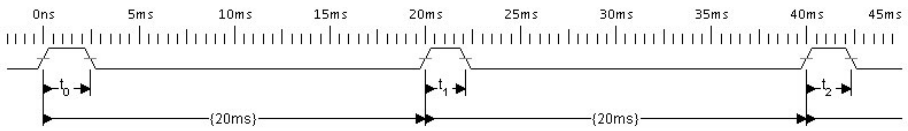


Obrázek 2.8: Modelářské servo

Pro řízení všech modelářských akčních členů (servomotorů a motorů s regulátory otáček) se obecně používá pulzní logický signál. Doba trvání pulzu v logické úrovni 1 stanovuje hodnotu akční veličiny. To může znamenat úhel natočení serva, výstupní napětí z regulátoru nebo přímo hodnotu požadovaných otáček motoru. Vzhledem k tomu, že se tento signál používá ve stejné podobě v podstatě pro všechny RC soupravy i akční členy, umožňuje výbornou kompatibilitu mezi výrobky různých firem.

Vstupní napěťové úrovně jsou kompatibilní s TTL, můžeme je tedy přímo spojit s vstupy nebo výstupy mikrokontroléru provozovaného na 5V.

Perioda řídicího signálu je 20 ms. Doba trvání signálu se může pohybovat cca v rozmezí 0,8 ms až 2 ms. Střed intervalu obvykle znamená nulovou výchylku serva, případně zastavení motoru. U analogových serv je nutné úhly natočení ramene odpovídající příslušným časům určit experimentálně, protože obvykle není dostupná dostatečná dokumentace, kde by tyto informace bylo možné zjistit. U digitálních serv by však konkrétní délka pulzu měla odpovídat příslušnému úhlu.



Obrázek 2.9: Průběh řídícího signálu serv

Vzhledem k nutnosti rychlé odezvy serv je však jejich elektronika hodně citlivá na jakékoliv nepřesnosti v generovaném signálu. Pokud je v signálu nezanedbatelný jitter, tak servo zaškubává, má výrazně vyšší spotřebu a může být hlučné. Pokud se tak provozovalo dlouhodobě, mohlo by dojít k jeho rychlému opotřebení. Pokud je to tedy možné, tak je vhodné signál generovat pomocí hardware, např. časovače TPU v režimu PWM. Potom generování tohoto signálu není závislé na ostatních činnostech programu a výstupní signál bude z časového hlediska velice stabilní. V případě, že je generování signálu vyřešeno z programu na základě přerušení od časovače, tak je nutno věnovat velkou pozornost prioritě přerušení, aby k náhodným zpožděním v generovaném signálu docházelo co nejméně. Obecně to na některých mikrokontrolérech může znamenat problém. V této práci je pro generování signálu použita jednotka TPU, tyto vlivy jsou tedy vyřešeny robustně a problémy s jitterem zde nehrozí.

2.4.6 Komunikace na sběrnici CAN

Sběrnice CAN[1] je standard pro sériovou komunikaci mezi několika zařízeními vyvinutý v osmdesátých letech firmou Robert Bosch GmbH. Původně byl uvažován především pro komunikaci v motorových vozidlech, postupem času se však uplatnil i v mnoha jiných oblastech. Jeho zásadní výhodou je determinismus, rychlost a odolnost proti rušení. Do značné míry vychází ze sběrnice RS-485, která je podobná fyzické vrstvě CANu, implementuje však navíc další vrstvy.

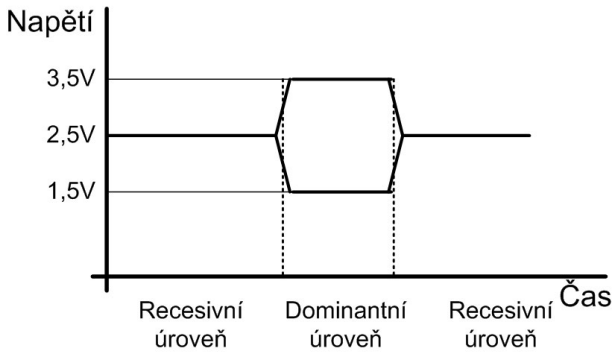
Tato sběrnice má však velmi dobře vyřešený systém sdílení přenosového media. Využívá se CSMA/BA[2] přístup. Každá stanice má v síti přidělenou unikátní prioritu. Pokud dojde k současnému vysílání ze dvou stanic, tak stanice s vyšší prioritou „přebije“ stanici s nižší (číselně nižší ID znamená zprávu s vyšší prioritou). Dojde k odvysílání důležitější zprávy a stanice vysílající zprávu s nižší prioritou musí počkat, než se uvolní sběrnice. Zásadní výhoda

[1] Controller Area Network

[2] Carrier Sense Multiple Access/Bitwise Arbitration

tedy spočívá v tom, že v případě kolize nedojde k poškození zprávy s vyšší prioritou a musí se opakovat pouze ta s nižší. Nedochází tedy k zbytečným prostojům na sběrnici, opakováním přenosu a efektivní přenosová rychlost se nesnižuje s rostoucím objemem přenášených dat.

Tento princip priorit je realizován pomocí použitých napěťových úrovní. Existují dvě, recesivní a dominantní. Sběrnice se pak chová jako wired AND. Pokud kterákoliv ze stanic vyšle dominantní úroveň, tak převáží nad všemi ostatními, které vyslaly recesivní. Zároveň každá stanice sleduje stav na sběrnici, a pokud se liší od vlastní vyslané úrovně (t.j. vysílá recesivní a na sběrnici se objeví dominantní), tak je povinna přestat vysílat až do doby uvolnění sběrnice.



Obrázek 2.10: Recesivní a dominantní úrovně na sběrnici CAN

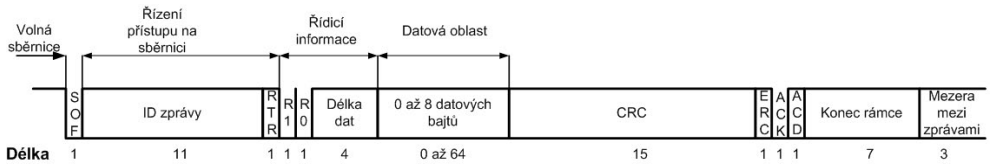
Ve verzi CAN2.0A má identifikátor zprávy délku 11 bitů, ve verzi CAN2.0B je rozšířen na 29 bitů.

Zprávy na sběrnici CAN jsou zabalené do čtyř typů rámců. To jsou:

- Data frame: Používá se pro přenos dat mezi stanicemi.
- Remote frame: Zajišťuje požadavek na odeslání dat.
- Error frame: Rámec vyslaný stanicí, která detekovala chybu přenosu.
- Overload frame: Rámec pro vložení zpoždění mezi datovým a remote rámcem.

Data frame

Data frame se používá pro vlastní komunikaci po sběrnici CAN. Délka jeho datové oblasti se může lišit od 0 do 8 bytů. Jeho struktura je naznačena na obrázku 2.11 a v následujícím výpisu.



Obrázek 2.11: Datová zpráva dle standardu CAN2.0A

- SOF – Start Of Frame – značka signalizující začátek paketu
- ID – identifikátor určující prioritu zprávy a nepřímo jejího odesílatele
- RTR – Remote Transmission Request – bit určující, zda se jedná o Data/Remote frame. V datové zprávě je tento bit dominantní
- R0, R1 – rezerva
- délka – délka datové oblasti
- datová oblast – 0 až 8 bytů dat
- CRC – zabezpečení rámce
- ACK, ACD – potvrzení
- mezera mezi zprávami – musí být dlouhá minimálně 3 recesivní bity

Remote frame

Tento rámec je v podstatě identický jako datový rámec, pouze RTR bit je v recesivní úrovni a v poli obsahujícím délku je počet požadovaných bytů. Jeho vlastní datová oblast je prázdná. Využívá se především jako požadavek na jinou stanici o zaslání dat.

Error frame

Pokud kterákoliv ze stanic na CAN sběrnici detekuje chybu přenosu, tak je její povinností odvysílat příslušný chybový rámec. V závislosti na stavu stanice dojde k odvysílání 6 dominantních bitů nebo 6 recesivních bitů.

Overload frame

Tento rámec se používá v případě, že některá stanice nestihne zpracovat data. Jeho struktura je obdobná chybovému rámci, může však být odeslán na konci zprávy.

2.4.7 Sériová komunikace

Pro základní komunikaci s uživatelem a pro zavádění programu do mikrokontroléru je využit asynchronní sériový kanál. Jedná se o sériové porty vycházející z RS232, na desce však nejsou osazeny budiče a komunikuje se na napěťových úrovních TTL. Pro spojení desky s počítačem, nebo jiným zařízením, požadujícím splnění napěťových úrovní RS232C lze využít například kabely s vestavěnými převodníky MAX232. V současné době probíhá komunikace s parametry 57600 bit/s, 8 datových bitů, žádná parita, 1 stopbit. V případě potřeby může být toto v programu snadno upraveno změnou inicializačního příkazu pro sériový port.

Porty jsou vyvedeny na konektory typu PSH02-04, kde je kromě samotných datových vodičů zapojeno ještě napájení +5V, aby šlo snadno připojit případné převodníky na napěťové úrovně RS232C. Vodiče pro řízení toku se v této konstrukci nevyužívají. Jednotka má v aktuální verzi vyvedeny tři sériové komunikační porty. Ty jsou označené COM1 až COM3. Port COM1 se používá pro servisní komunikaci (viz. kapitola A), port COM2 pro zavádění programu a port COM3 zatím není využit.

Vodič	Funkce
1	+5V
2	RXD
3	TXD
4	GND

Tabulka 2.1: Zapojení konektorů sériového portu

Kapitola 3

Mikrokontrolér Renesas H8S2638

3.1 Základní popis

Tento mikrokontrolér patří do třídy H8S/2600 firmy Renesas Technology. Architektura procesoru je 32-bitová, procesor je schopen adresovat 16 MB linárního adresového prostoru. Předložený popis vychází z [7]

Jádro procesoru obsahuje šestnáct 16-bitových registů, použitelných i jako osmibitové, případně po dvou zřetězitelné jako osm 32-bitových registrů. Maximální hodinová frekvence je 20 MHz, vzhledem k vybavení PLL násobičkou je možné využít i krystal s menší frekvencí a hodiny vynásobit již přímo v mikrokontroléru. Pro usnadnění ladění programu je jádro vybaveno řadičem breakpointů. Tato jednotka umožňuje nastavit dvě adresy v programu, při jejichž dosažení dojde k vygenerování speciálního přerušení. Další zajímavou vlastností je Data Transfer Controller (DTC).

Periferie mikrokontroléru:

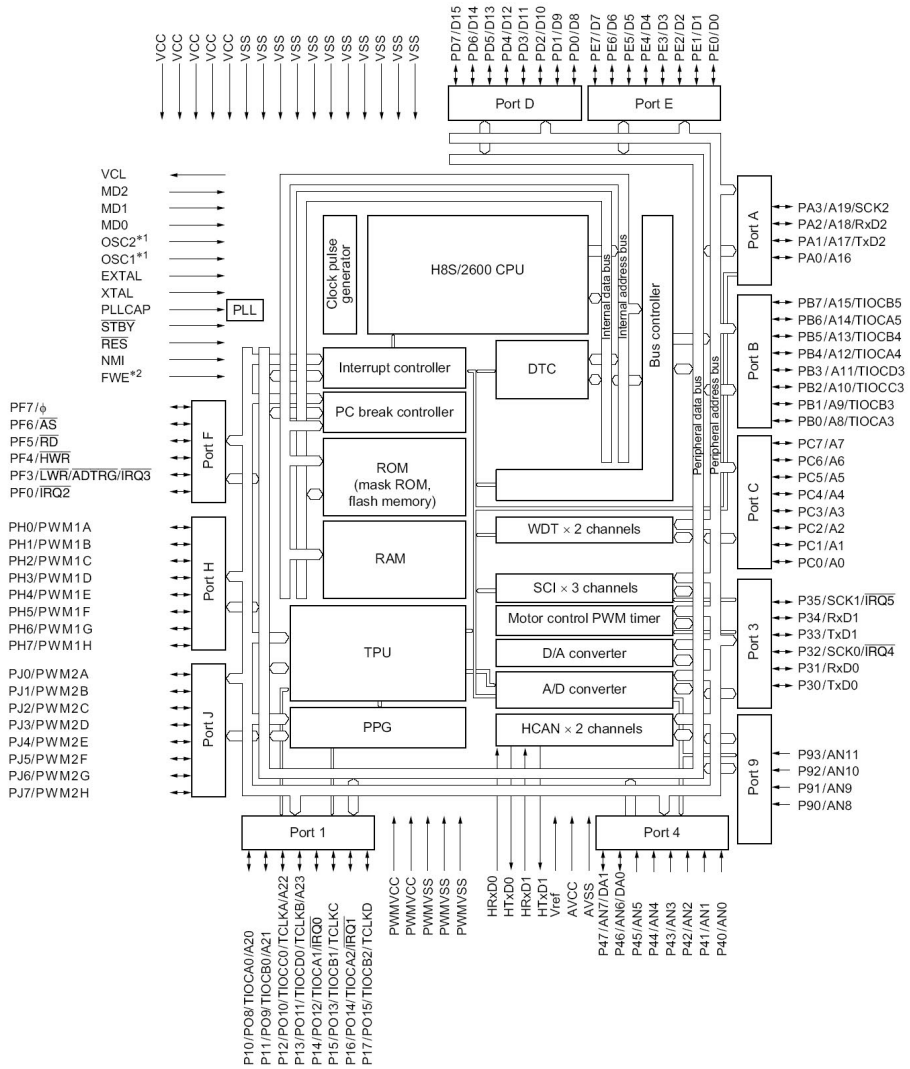
- DTC – Data Transfer Controller : jednotka pro přenos dat mezi periferiemi a pamětí (podrobněji posána dále).
- paměti ROM (FLASH) a RAM
- jednotka časovačů - TPU[1]
- programovatelný generátor pulzů (ve spojení s TPU jednotkou)
- watchdog časovač
- Motor Control PWM
- 3x sériové komunikační rozhraní

[1] Time Pulse Unit

- AD převodník
- DA převodník
- 2x řadič CAN sběrnice
- volitelně v některých provedeních řadič I2C sběrnice

Mikrokontrolér je napájen z napětí 5V. Vzhledem ke spotřebě se nehodí na aplikace dlouhodobě napájené z baterií. V našem případě se však jedná o krátké doby provozu z akumulátorů a ve srování se spotřebou akčních členů vrtulníku není odběr tohoto mikrokontroléru příliš podstatný. Jeho použití tedy nic nebrání. Maximální hodinová frekvence 20 MHz již dnes také nepatří mezi nejvyšší. Vzhledem k možnosti využívat technologii DTC však lze část jednoduchých úloh, které jsou zpracovávány velice často a zabírají tak podstatné množství strojového času (například přesun dat přijatých ze sériového portu do paměti) přenést na jádro mikrokontroléru. Tím je možné dosáhnout vysokého výkonu i při poměrně nízké taktovací frekvenci.

3.2 Blokové schema



Obrázek 3.1: Blokové schema mikrokontroléru

3.3 Porty

Připojení vnějších součástí je možné pomocí 10 vstupně výstupních portů a dalších dvou výhradně vstupních. Je však nutné dát pozor na přiřazení funkcí jednotlivým portům, aby nedošlo ke kolizi s použitou periferií. Všechny vstupně výstupní porty mají svůj registr směru (DDR – Data Direction Re-

gister), kterým je možné nastavit, zda se příslušný pin má chovat jako vstup nebo výstup. Dále obsahují datový registr (DR – Data Register), který obsahuje požadovaný výstup. Skutečnou hodnotu z pinu můžeme přečíst v registru PORT. Porty vybavené funkcí „pull-up“ mají navíc registr PCR (Pull-up Control Register), porty s funkcí „open-drain“ mají navíc registr ODR (Open-Drain control Register). Některé porty mají na vstupu Schmittův klopný obvod. Podrobněji popisuje konfiguraci portů tabulka 3.1.

Port	Vstup	Výstup	Šířka [bit]	Schmitt.k.o.	Pull-up	Open-drain
1	ano	ano	8	ano	ne	ne
3	ano	ano	6	ano	ne	ano
4	ano	ne	8	ne	ne	ne
9	ano	ne	4	ne	ne	ne
A	ano	ano	4	ne	ano	ano
B	ano	ano	8	ne	ano	ano
C	ano	ano	8	ne	ano	ano
D	ano	ano	8	ne	ano	ne
E	ano	ano	8	ne	ano	ne
F	ano	ano	6	ano	ne	ne
H	ano	ano	8	ne	ne	ne
J	ano	ano	8	ne	ne	ne

Tabulka 3.1: Konfigurace portů

3.4 Paměťový prostor

Mikrokontrolér H8S2638 obsahuje 256kB flash ROM a 16 kB statické RAM paměti. Je tedy možné ho efektivně používat pro menší projekty v jednočipové konfiguraci. V případě potřeby většího množství paměti je k dispozici datová sběrnice pro její připojení.

Celkově může mikrokontrolér přímo adresovat 16 MB paměti. Tento rozsah je rozdělen do 8 oblastí po 2 MB, které lze jednotlivě a nezávisle konfigurovat. Je možné nastavit způsob přístupu (8 nebo 16 bitů) a vkládání čekacích cyklů pro pomalejší paměti.

3.5 Přerušovací systém

Systém přerušení zajišťuje rychlou a efektivní reakci programu na vnější události, případně na činnost periferií. Umožňuje nezávislé nastavení priorit jed-

notlivých přerušení v osmi skupinách. Přerušení přiřazené v rámci jedné skupiny mají priority rozdělené napevno, odpovídající velikosti vektoru přerušení, viz. tabulky 3.2, 3.3 a 3.4. Čím menší je hodnota vektoru přerušení, tím vyšší prioritu má v rámci jedné skupiny. Systém obsahuje 7 zdrojů externích přerušení a 42 interních. Za všech okolností nejvyšší prioritu má přerušení NMI, nevztahuje se na něj ani bit zákazu všech přerušení. Všechny další je možné zakázat nebo povolit buď dohromady pomocí bitu I v registru CCR, nebo jednotlivě pomocí příslušné masky. Každé přerušení má přiřazen vlastní vektor a adresu, není tedy nutné v programu nějak problematicky zdroj přerušení identifikovat.

Při tvorbě programu je však nutné dávat velký pozor na správné nastavení povolení jednotlivých přerušení. V případě, že bude vyvoláno špatně obsloužené přerušení (tj. nebude vymazán jeho příznak), tak bude mikrokontrolér dokola volat jeho obsluhu a zacyklí se.

V prostředí OMK se přerušení identifikuje podle hodnoty vektoru z tabulek 3.2, 3.3 a 3.4. Masky pro povolení/zakázání jednotlivých přerušení najdeme v dokumentaci příslušných periferií.

Zdroj	Jednotka	Vektor	Adresa	IPR
NMI	pin	7	0x001C	—
IRQ0	pin	16	0x0040	IPRA6 – IPRA4
IRQ1	pin	17	0x0044	IPRA2 – IPRA0
IRQ2	pin	18	0x0048	IPRB6 – IPRB4
IRQ3	pin	19	0x004C	IPRB6 – IPRB4
IRQ4	pin	20	0x0050	IPRB2 – IPRB0
IRQ5	pin	21	0x0054	IPRB2 – IPRB0
vyhrazeno	—	22 – 23	0x0058 – 0x005C	—
SWDTEND	DTC	24	0x0060	IPRC2 – IPRC0
WOVI0	Watchdog 0	25	0x0064	IPRD6 – IPRD4
vyhrazeno	—	26	0x0068	—
PC break	PC break	27	0x006C	IPRE6 – IPRE4
ADI	A/D	28	0x0070	IPRE2 – IPRE0
WOVI1	Watchdog 1	29	0x0074	IPRE2 – IPRE0
vyhrazeno	—	30 – 31	0x0078 – 0X007C	—

Tabulka 3.2: Zdroje přerušení, 1. část

Zdroj	Jednotka	Vektor	Adresa	IPR
TGI0A	TPU 0	32	0x0080	IPRF6 – IPRF4
TGI0B	TPU 0	33	0x0084	IPRF6 – IPRF4
TGI0C	TPU 0	34	0x0088	IPRF6 – IPRF4
TGI0D	TPU 0	35	0x008C	IPRF6 – IPRF4
TGI0V	TPU 0	36	0x0090	IPRF6 – IPRF4
vyhrazeno	—	37 – 39	0x0094 – 0x009C	—
TGI1A	TPU 1	40	0x00A0	IPRF2 – IPRF0
TGI1B	TPU 1	41	0x00A4	IPRF2 – IPRF0
TGI1V	TPU 1	42	0x00A8	IPRF2 – IPRF0
TGI1U	TPU 1	43	0x00AC	IPRF2 – IPRF0
TGI2A	TPU 2	44	0x00B0	IPRG6 – IPRG4
TGI2B	TPU 2	45	0x00B4	IPRG6 – IPRG4
TGI2V	TPU 2	46	0x00B8	IPRG6 – IPRG4
TGI2U	TPU 2	47	0x00BC	IPRG6 – IPRG4
TGI3A	TPU 3	48	0x00C0	IPRG2 – IPRG0
TGI3B	TPU 3	49	0x00C4	IPRG2 – IPRG0
TGI3C	TPU 3	50	0x00C8	IPRG2 – IPRG0
TGI3D	TPU 3	51	0x00CC	IPRG2 – IPRG0
TGI3V	TPU 3	52	0x00D0	IPRG2 – IPRG0
vyhrazeno	—	53 – 55	0x00D4 – 0x00DC	—
TGI4A	TPU 4	56	0x00E0	IPRH6 – IPRH4
TGI4B	TPU 4	57	0x00E4	IPRH6 – IPRH4
TGI4V	TPU 4	58	0x00E8	IPRH6 – IPRH4
TGI4U	TPU 4	59	0x00EC	IPRH6 – IPRH4
TGI5A	TPU 5	60	0x00F0	IPRH2 – IPRH0
TGI5B	TPU 5	61	0x00F4	IPRH2 – IPRH0
TGI5V	TPU 5	62	0x00F8	IPRH2 – IPRH0
TGI5U	TPU 5	63	0x00FC	IPRH2 – IPRH0
vyhrazeno	—	64 – 79	0x0100 – 0x013C	—
ERRI0	SCI 0	80	0x0140	IPRJ2 – IPRJ0
RXI0	SCI 0	81	0x0144	IPRJ2 – IPRJ0
TXI0	SCI 0	82	0x0148	IPRJ2 – IPRJ0
TEI0	SCI 0	83	0x014C	IPRJ2 – IPRJ0

Tabulka 3.3: Zdroje přerušení, 2. část

Zdroj	Jednotka	Vektor	Adresa	IPR
ERRI1	SCI 1	84	0x0150	IPRK6 – IPRK4
RXI1	SCI 1	85	0x0154	IPRK6 – IPRK4
TXI1	SCI 1	86	0x0158	IPRK6 – IPRK4
TEI1	SCI 1	87	0x015C	IPRK6 – IPRK4
ERRI2	SCI 2	88	0x0160	IPRK2 – IPRK0
RXI2	SCI 2	89	0x0164	IPRK2 – IPRK0
TXI2	SCI 2	90	0x0168	IPRK2 – IPRK0
TEI2	SCI 2	91	0x016C	IPRK2 – IPRK0
vyhrazeno	—	92 – 99	0x0170 – 0x018C	—
I2CI0	I2C 0	100	0x0190	IPRL2 – IPRL0
DDCSW1	I2C 0	101	0x0194	IPRL2 – IPRL0
I2CI1	I2C 1	102	0x0198	IPRL2 – IPRL0
vyhrazeno	—	103	0x019C	IPRL2 – IPRL0
PWM1	PWM 1	104	0x01A0	IPRM6 – IPRM4
PWM2	PWM 2	105	0x01A4	IPRM6 – IPRM4
HCAN0	HCAN0	106	0x01A8	IPRM6 – IPRM4
HCAN0	HCAN0	107	0x01AC	IPRM6 – IPRM4
HCAN1	HCAN1	108	0x01B0	IPRM2 – IPRM0
HCAN1	HCAN1	109	0x01B4	IPRM2 – IPRM0
vyhrazeno	—	110 – 111	0x01B8 – 0x01BC	—

Tabulka 3.4: Zdroje přerušení, 3. část

3.6 Technologie DTC

Jednotka DTC[2] zajišťuje přenosy dat mezi pamětmi bez nutnosti přímého zásahu uživatelského programu. Je možné nakonfigurovat ji tak, aby jako reakci na událost (přerušení, případně aktivaci programem) přenesla skupinu dat z cílové do zdrojové oblasti. Přenosy je možné podle potřeby řetězit. Počet přenosových kanálů je omezen pouze blokem paměti SRAM 1kB, do něhož jsou mapovány registry konfigurace DTC. Cílová i zdrojová adresa je z plného rozsahu 16 MB, přenos je tedy možný mezi všemi regulérně připojenými paměťmi. Pro cílovou i zdrojovou adresu lze nastavit režim inkrementování, dekrementování, nebo ponechání adresy po provedení akce.

Události schopné startovat přenos DTC:

- vnější přerušení IRQ
- jednotka časovačů TPU

[2]Data Transfer Controller

- sériové porty SCI
- převodník A/D
- PWM jednotka pro řízení motorů
- řadič CAN pro mailbox 0
- software

Dále je možné nastavit jednotku tak, aby příznak přerušení, který vyvolal přenos, buď zamaskovala, nebo ponechala pro vyvolání obsluhy přerušení v programu. To může být navíc ovlivněno počtem přenosů, tzn. teprve po provedení nastaveného počtu přenosů se zavolá přerušení. Tato vlastnost by mohla mít využití například ve spojení s AD převodníkem, jako načtení určitého počtu vzorků a následnému vyvolání podprogramu zpracovávajícího naměřená data.

Jednotka DTC může při pravidelně opakovaných úkonech velmi zefektivnit činnost mikrokontroléru. Například ve spojení se SCAN režimem AD převodníku (viz. kapitola 3.7.3) může zajistit zpracování až 4 analogových signálů při výrazně vyšší rychlosti, než by bylo možné realizovat při obsluze pouze programem.

3.7 Vybrané periferie

3.7.1 Jednotka časovačů TPU

Jednotka Timer Pulse Unit obsahuje 6 kanálů, všechny jsou vybavené šestnáctibitovými registry. Každému kanálu odpovídá jeden čítač připojený na zdroj hodin a několik dalších registrů. Přídavných registrů je celkem 16 (4 pro kanály 0 a 3, po dvou pro kanály 1, 2, 4 a 5). Tyto registry mohou být nezávisle použity pro funkce input capture nebo output compare. Každý kanál má k dispozici 8 zdrojů hodin, ty mohou být vybrány nezávisle. Dostupnost zdrojů hodin pro jednotlivé kanály je naznačena v tabulce 3.5, kde ϕ je hodinová frekvence mikrokontroléru a TCLKA až TCLKD jsou příslušné vstupy mikrokontroléru.

V případě potřeby vyššího rozlišení je možné jednotky zřetězit, spojit kanály 1. s 2. a 4. s 5. pro zvýšení jejich rozlišení a využití jako 32-bitové jednotky čítačů.

Jednotlivé kanály jednotky TPU je možné nastavit do následujících funkcí:

- Změna výstupu při rovnosti registrů (compare match) - základ funkce PWM.

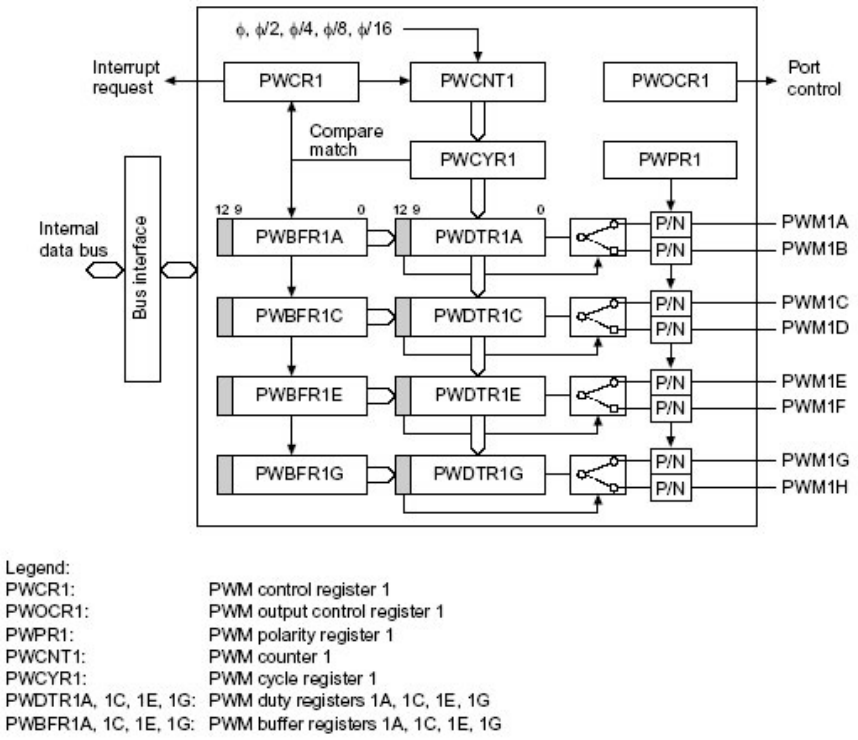
- Záchytný registr - zaznamenání hodnoty časovače do registru při hraně náběžné, sestupné nebo při obou hranách vstupního signálu.
- Vynulování čítače při změně vstupu nebo rovnosti registru (synchronizace s vnějšími událostmi nebo generování periody PWM).
- Synchronní režim - propojení více kanálů TPU dohromady.
- Čítač nahoru/dolů - pro kanály 1, 2, 4 a 5.
- Start AD převodníku - událost compare match nebo input capture může být použita pro spouštění konverze (tato funkce je však omezena na první registr každého kanálu).

Zdroj hodin	Kanál 0	Kanál 1	Kanál 2	Kanál 3	Kanál 4	Kanál 5
ϕ	ano	ano	ano	ano	ano	ano
$\phi/4$	ano	ano	ano	ano	ano	ano
$\phi/16$	ano	ano	ano	ano	ano	ano
$\phi/64$	ano	ano	ano	ano	ano	ano
$\phi/256$	ne	ano	ne	ano	ne	ano
$\phi/1024$	ne	ne	ano	ano	ano	ne
$\phi/4096$	ne	ne	ne	ano	ne	ne
TCLKA	ano	ano	ano	ano	ano	ano
TCLKB	ano	ano	ano	ne	ne	ne
TCLKC	ano	ne	ano	ne	ano	ano
TCLKD	ano	ne	ne	ne	ne	ano

Tabulka 3.5: Zdroje hodin pro TPU kanály

3.7.2 Jednotka Motor Control PWM

Tato jednotka je blok speciálních časovačů, určených primárně pro řízení motorů, nic ale nebrání jejich obecnému využití. Jednotka má dva nezávislé kanály, z nichž každý je vybaven osmi výstupy. Blokové schema prvního kanálu je na obrázku 3.2, druhého kanálu na 3.3. U obou můžeme nezávisle nastavit hodinovou frekvenci i periodu generování PWM.

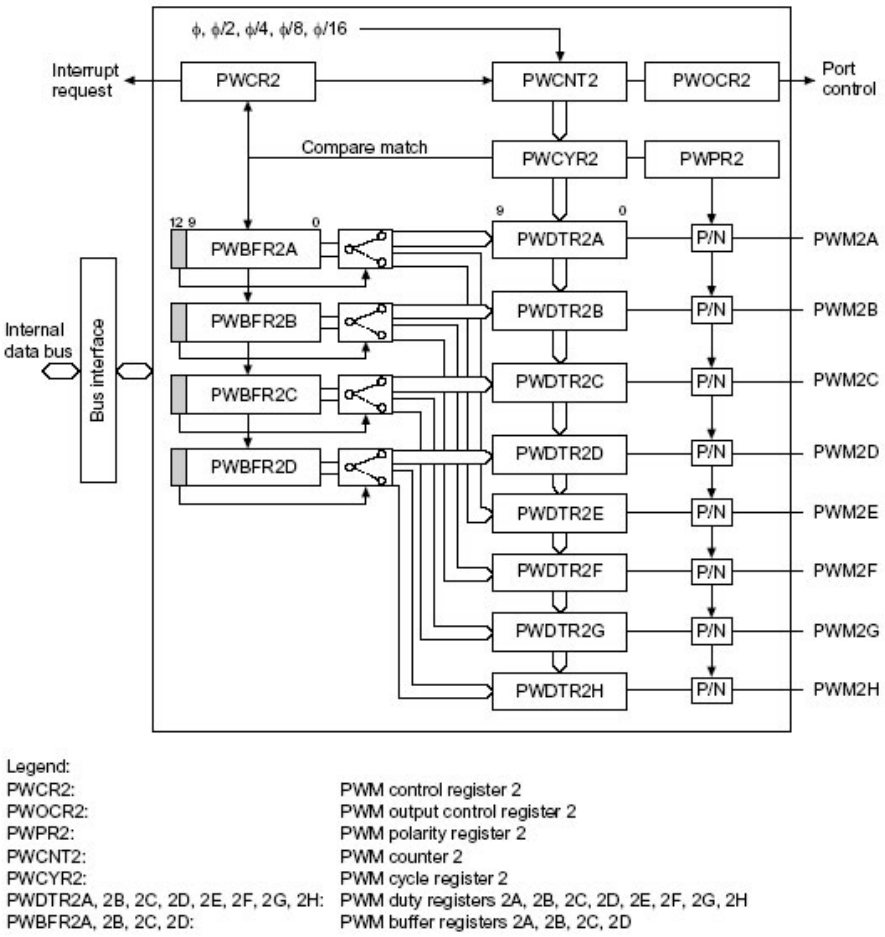


Legend:

PWCR1:	PWM control register 1
PWOCR1:	PWM output control register 1
PWPR1:	PWM polarity register 1
PWCNT1:	PWM counter 1
PWCYR1:	PWM cycle register 1
PWDTR1A, 1C, 1E, 1G:	PWM duty registers 1A, 1C, 1E, 1G
PWBFR1A, 1C, 1E, 1G:	PWM buffer registers 1A, 1C, 1E, 1G

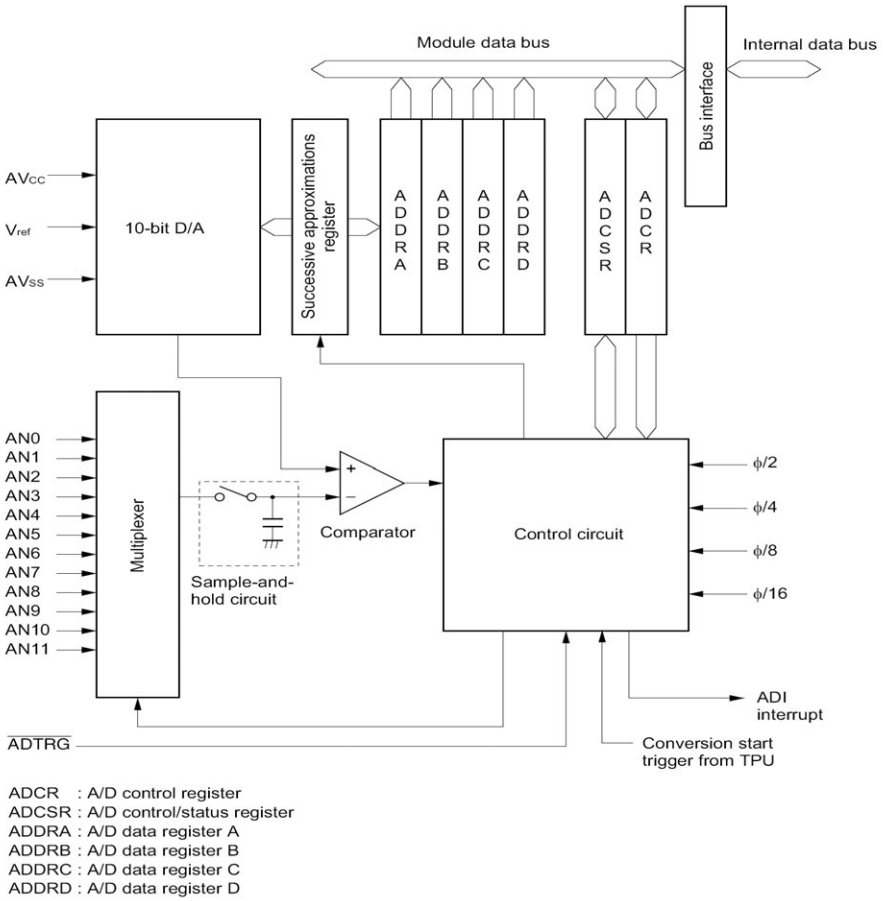
Obrázek 3.2: Blokové schema PWM jednotky pro řízení motorů 1

Ačkoliv je však každý z kanálů vybaven osmi výstupy, vždy jsou však spojeny po párech. PWM1A sdílí společný registr s PWM1B, PWM1C sdílí registr s PWM1D atd.. Mírně odlišná situace je u druhého kanálu, kde se sdílí pouze časovač. Stejně však není možné použít všech osm výstupů najednou. Reálně je tedy možné najednou generovat pouze čtyři PWM výstupy. V případě nutnosti ale lze snadno výstupy přepínat.



Obrázek 3.3: Blokové schema PWM jednotky pro řízení motorů 2

Jednotka může využívat hodinový signál ϕ, $\phi/2$, $\phi/4$, $\phi/8$, nebo $\phi/16$, kde ϕ je taktovací signál mikrokontroléru. Čítače a registry této funkce jsou však pouze desetibitové. Je tedy nutné vzít na vědomí, že minimální obnovovací frekvence PWM časovače je poměrně vysoká, například při $\phi = 20MHz$ vyjde tato frekvence přibližně 1,22 kHz. V naprosté většině případů to rozhodně není na závadu, využít ale tuto jednotku například pro generování signálu modelářských servomotorů není možné. Proto se také v této úloze používá na signál servomotorů jednotka TPU, která je podstatně univerzálnější.



Obrázek 3.4: Blokové schema AD převodníku

3.7.3 Převodník A/D

Jednotka A/D převodníku v mikrokontroléru H8S2638 je řešena jako desetibitový převodník s postupnou aproximací. Pomocí multiplexeru umožňuje měření maximálně 12 kanálů. Tyto kanály lze rozdělit do tří skupin po čtyřech. Pak je možné, v rámci jedné skupiny, měřit průběžně 4 kanály, aniž by bylo nutné průběžně programem měnit nastavení multiplexeru. Tento režim může být velmi výhodný například pro měření efektivní hodnoty několika signálů, případně implementaci různých filtrů. Po dokončení převodu je vygenerováno přerušení ADI, aby bylo možné na nově měřená data reagovat v programu efektivně. Při běhu systému na hodinové frekvenci 20 MHz dosáhne minimální doba převodu hodnoty 13.3 μs. V méně náročných případech je možné konverzi zpomalit pomocí nastavení předděličky hodinového signálu. Použitý

AD převodník vyžaduje připojení externího napětí na pin Vref jako referenci. Vzhledem ke vstupnímu rozsahu je však možné pro orientační měření použít přímo napájecí napětí Vcc 5V. V případě požadavku na přesné měření samozřejmě použijeme kvalitní napěťovou referenci. Pro náročnější aplikace je také nutné oddělit napájení digitální a analogové části včetně zemí. Blokové schema AD převodníku je na obrázku 3.4.

CH3	CH2	CH1	CH0	Režim SINGLE	Režim SCAN
0	0	0	0	AN0	AN0
0	0	0	1	AN1	AN0, AN1
0	0	1	0	AN2	AN0, AN1, AN2
0	0	1	1	AN3	AN0, AN1, AN2, AN3
0	1	0	0	AN4	AN4
0	1	0	1	AN5	AN4, AN5
0	1	1	0	AN6	AN4, AN5, AN6
0	1	1	1	AN7	AN4, AN5, AN6, AN7
1	0	0	0	AN8	AN8
1	0	0	1	AN9	AN8, AN9
1	0	1	0	AN10	AN8, AN9, AN10
1	0	1	1	AN11	AN8, AN9, AN10, AN11
1	1	0	0	—	—
1	1	0	1	—	—
1	1	1	0	—	—
1	1	1	1	—	—

Tabulka 3.6: Možnosti volby vstupních kanálů AD převodníku

V tabulce 3.6 jsou zaznamenány možné režimy vstupních skupin AD převodníku. V SINGLE režimu se tedy vždy načítá hodnota pouze z jednoho vstupu, ve SCAN režimu můžeme nastavit měření až ze 4 kanálů najednou. Není však možné volit kanály naprosto libovolně, proto je nutné postupovat při návrhu zapojení obezřetně.

Měření AD převodníku může být spouštěno buď zásahem programu nastavením bitu ADST, případně je možné využít automatické spouštění od jednotky TPU nebo externím pinem $\overline{ADTRG}$.

Použití AD převodníku v SINGLE mode

Tento režim je vhodný především pro jednotlivá, nepříliš často opakovaná měření. Jeho výhoda spočívá v možnosti nastavit multiplexer na libovolný

vstup AN0 až AN11. Po dokončení převodu je nastaven příznak ADI, na který může mikrokontroler zareagovat vyvoláním přerušení nebo přenosu DTC.

Použití AD převodníku v SCAN mode

Při potřebě pravidelně měřit jeden nebo několik (až 4) analogové signály se vhodně uplatní SCAN režim. Signály je nutné připojit do jedné skupiny podle tabulky 3.6, tak, aby nebylo nutné provádět během měření změnu konfiguračních bitů CH0 až CH3. Po odstartování měření pak začne AD převodník cyklicky měřit napětí na AD převodníku. Po dokončení každého měření je nastaven příznak ADI.

3.7.4 Řadič sběrnice CAN

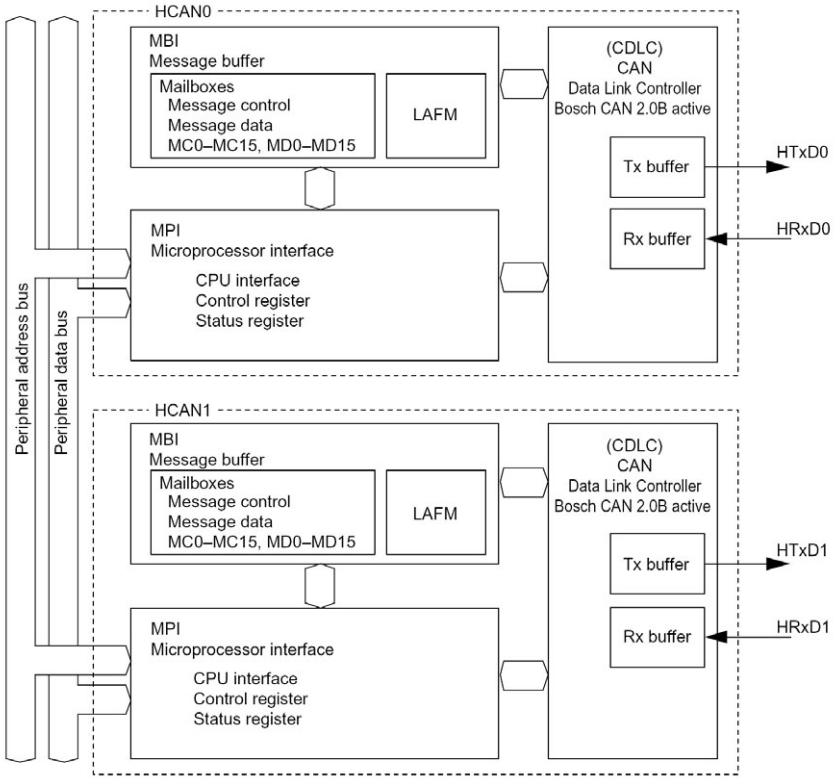
Mikrokontrolér H8S2638 obsahuje dva nezávislé řadiče sběrnice CAN. V rámci dokumentace firmy Renesas jsou tyto řadiče označovány názvem HCAN[3]. Řadiče jsou kompatibilní s CAN 2.0B normou. Maximální dosažitelná rychlost je 1 Mbps. Každý řadič má 16 bufferů (mailboxů), z niž 15 je určených pro příjem nebo vysílání a 1 je pouze pro příjem). Struktura řadiče je na obrázku 3.5. Popis sběrnice CAN je možné nalézt v [2] a [3].

Mailboxy 1 až 15 vyžadují přesné nastavení ID zprávy pro příjem nebo vysílání. Mailbox 0 je možné použít pro příjem většího rozsahu zpráv. To je realizováno pomocí masky LAFM. Při přijmu zprávy se provede binární AND na přijatém ID a masce. Pokud se výsledek této operace shoduje s ID nastaveným v mailboxu, tak je zpráva přijata.

Po přijetí zprávy je vyvolán příslušný příznak, který může vyvolat přerušení. Pro mailbox 0 je také možné využít přenos pomocí DTC (kapitola 3.6), u ostatních mailboxů to však možné není. Řadič CAN zároveň informuje pomocí přerušení na tyto události:

- Přerušení při chybě přenosu
- Reset
- Přijetí zprávy (mailboxy 1 až 15)
- Přijetí zprávy (mailbox 0) – může startovat DTC
- Dokončení odesílání zprávy

[3] Hitachi Controller Area Network



Obrázek 3.5: Blokové schema řadiče CAN

Kapitola 4

Použité vývojové prostředky

4.1 Návrh zapojení a plošného spoje

Pro návrh elektrického zapojení byl použit programový balík OrCAD 9.2 firmy Cadence Design Systems. Při jeho obsluze bylo použito dokumentace uvedené v [5].

4.2 Tvorba programového vybavení

Pro mikrokontrolér Renesas H8S2638 průběžně vzniká na Katedře řídicí techniky vývojový systém OMK, který byl také pro vývoj programového vybavení v této práci použit. Aktuální verze systému je na přiloženém CD. Vzhledem k vývoji prostředí lze do budoucna očekávat drobné problémy při kompilaci dále neudržovaných projektů pomocí aktuální verze prostředí. Kompilace pomocí verze zálohované na přiloženém CD však bude těchto vlivů ušetřena. Vývojový systém OMK je dokumentován v [6].

Kapitola 5

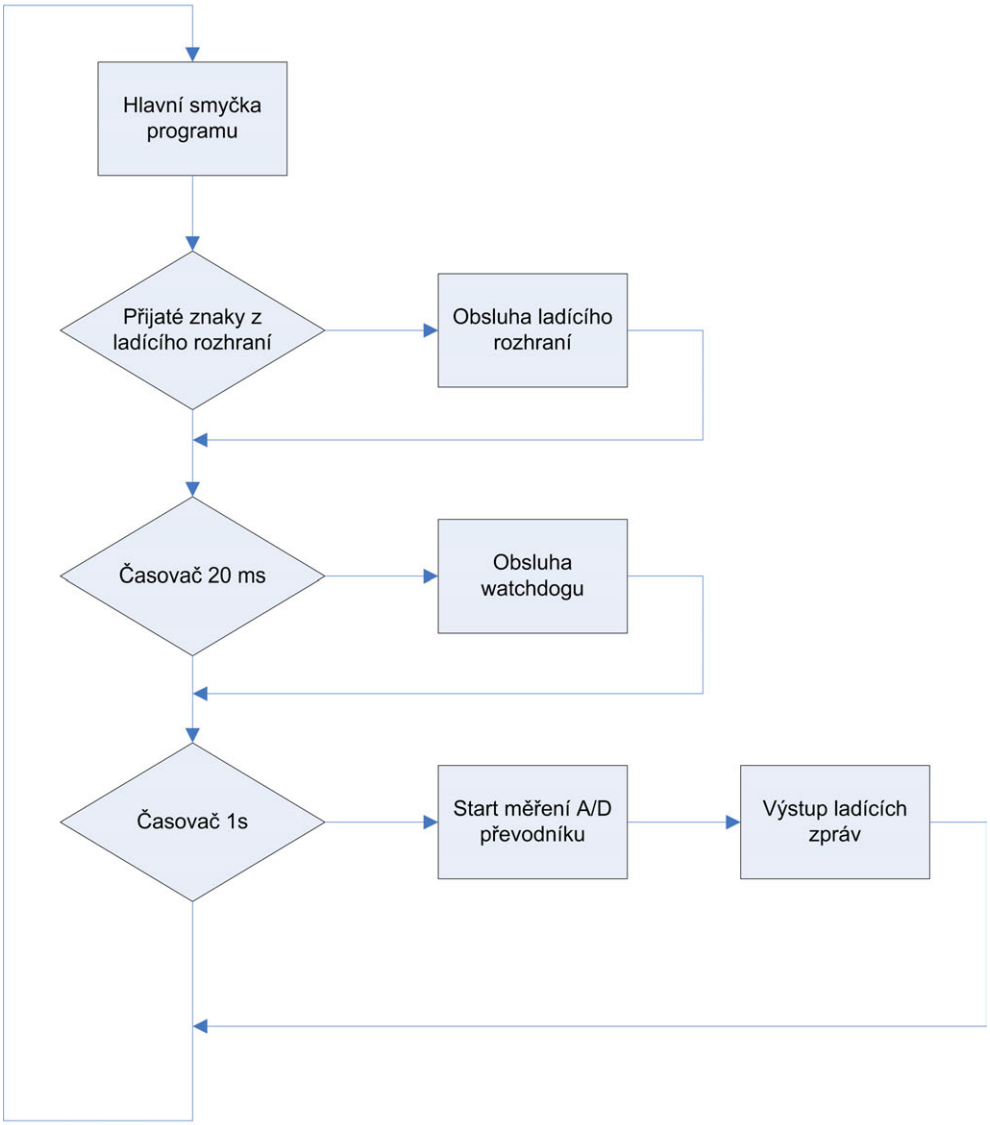
Realizace úlohy

5.1 Software

5.1.1 Struktura programu

Programové vybavení je vytvořeno v jazyce C[1] pro kompilátor gcc a prostředí OMK. Zdrojové soubory jsou k dispozici na přiloženém CD. V systému pro správu verzí DARCS využívaném v rámci skupiny RTIME na Katedře řídicí techniky je možné najít pod názvem „servodrv“. Projekt obsahuje tyto soubory:

- *servodrv.c*, *servodrv.h* – hlavní program projektu, obsluha sériové linky
- *servoout.c*, *servoout.h* – příjem signálů z RC soupravy, generování signálů pro serva
- *mycan.c*, *mycan.h* – ovladač sběrnice CAN



Obrázek 5.1: Nástin činnosti hlavní smyčky programu

5.1.2 Implementace komunikace CAN

Implementace komunikace CAN je v programovém souboru *mycan.c*. Je rozdělena na dvě základní činnosti, inicializaci CAN řadiče a reakci na události vzniklé při běhu programu. Ty jsou reprezentovány většinou přerušením, které odpovídá příchozí zprávě, chybě na sběrnici, resetu řadiče nebo dokončení odesílání zprávy. Události ve směru z mikrokontroléru jsou reprezen-

továny voláním příslušné funkce v tomto programovém souboru, například `can_send_servo_auto()`. Jejich úkolem je zapouzdřit přípravu a odeslání zprávy, tedy volitelně připravit data do mailboxu a nastavit příslušný bit požadavku na odeslání zprávy (jednotlivé bity registru TXPRL).

V současném stav projektu se využívá pouze komunikace po prvním kanálu sběrnice CAN. Druhý kanál je zatím volný a je možné ho využít způsobem podobným, jako se pracuje s již využitým řadičem.

Inicializace řadiče, komunikační rychlosti i dalších parametrů se provádí ve funkci `init_CAN0`. Na jejím konci se také připraví konfigurace mailboxů, tak, aby při dalším běhu programu stačilo pouze ukládat data do příslušných datových oblastí a požádat jejich odeslání.

5.1.3 Protokol CAN komunikace s hlavním řídicím počítačem

Pro komunikaci na sběrnici CAN se používá fyzická vrstva typu CAN H. Řadič CAN je nastaven na rychlost 100 kBit/s. Z větší části je komunikace řízena z hlavního počítače, pouze některé zprávy se odesílají samostatně.

Přijímané zprávy

1. Synchronizace – ID 20
   Zajišťuje synchronizaci mezi jednotlivými subsystémy. Princip je popsán v kapitole 2.4.1. Obvykle je vysílána v intervalech 50 ms, z hlediska desky řízení serv však tento interval není nijak kritický. Po přijetí této zprávy jsou odeslána data o poloze serv (ID 35) a zároveň jsou zapsána na výstup do serv poslední přijatá data.

   Zpráva neobsahuje žádná data, pouze svoje ID.

Délka	ID	D0	D1	D2	D3	D4	D5	D6	D7
0	20	—							

Tabulka 5.1: Synchronizační zpráva – ID 20

   Zpráva se přijímá do mailboxu 2.

2. Nastavení požadovaných hodnot serv – ID 25, 26
   Zprávy jsou určeny pro nastavení požadovaných hodnot a masky serv z hlavního řídicího počítače, více viz. kapitola 2.4.2 a 2.4.3. V okamžiku přijetí zprávy se však data pouze zapíší do bufferu. K jejich přepsání

na výstup do serv dojde až při přijetí nejbližší synchronizační zprávy, aby bylo zajištěno stále stejné časování systému. Hodnota pro serva je uvedena jako délka pulzu aktivní úrovně signálu měřená v násobcích $0,8\mu s$.

Délka	ID	D0	D1	D2	D3	D4	D5	D6	D7
8	25	maska	rezerva	servo 1		servo 2		servo 3	
8	26	servo 4		servo 5		servo 6		rezerva	

Tabulka 5.2: Nastavení hodnot serv – ID 25,26

První část zprávy (s ID 25) se přijímá do mailboxu 1, druhá část (ID 26) jde do mailboxu 5.

Odesílané zprávy

1. Skutečná poloha serv – ID 35,36
   Obsahuje informace o aktuální nastavené poloze serv nezávisle na volbě režimu řízení automaticky/manuálně. Hodnota pro serva je uvedena jako délka pulzu aktivní úrovně signálu měřená v násobcích $0,8\mu s$. Dále obsahuje údaj o nastavené masce. Tato zpráva je odeslána v nejkratší možné době po přijetí synchronizační zprávy s ID 20.

Délka	ID	D0	D1	D2	D3	D4	D5	D6	D7
8	35	maska	rezerva	servo 1		servo 2		servo 3	
8	36	servo 4		servo 5		servo 6		servo 7	

Tabulka 5.3: Čtení hodnot serv – ID 35,36

   Aktuální poloha serv je odesílána přes mailboxy 3 (zpráva s ID 35) a 4 (zpráva s ID 36).

2. Přepnutí do automatického režimu – ID 10
   Po přepnutí do automatického režimu jednotka řízení servomotorů informuje hlavní řídicí počítač zprávou s ID 10 bez datového těla o této skutečnosti.

   Pro odesílání této zprávy je využit mailbox 6.

Délka	ID	D0	D1	D2	D3	D4	D5	D6	D7
0	10	—							

Tabulka 5.4: Přepnutí do automatického režimu – ID 10

3. Přepnutí do manuálního režimu – ID 11
   Po přepnutí do manuálního režimu jednotka řízení servomotorů informuje hlavní řídicí počítač zprávou s ID 11 bez datového těla o této skutečnosti.

Délka	ID	D0	D1	D2	D3	D4	D5	D6	D7
0	11	—							

Tabulka 5.5: Přepnutí do manuálního režimu – ID 11

   Tato zpráva využívá pro své odeslání mailbox 7.

4. Napětí – ID 50
   Měření palubních napětí je podrobněji popsáno v kapitole 2.4.4. Data jsou po sběrnici CAN odesílána paketem s ID 50 a obsahem popsaným v tabulce 2.6. Hodnoty napětí 1 i 2 jsou uvedené v desetinách voltu.

Délka	ID	D0	D1	D2	D3	D4	D5	D6	D7
2	50	napětí 1	napětí 2	—					

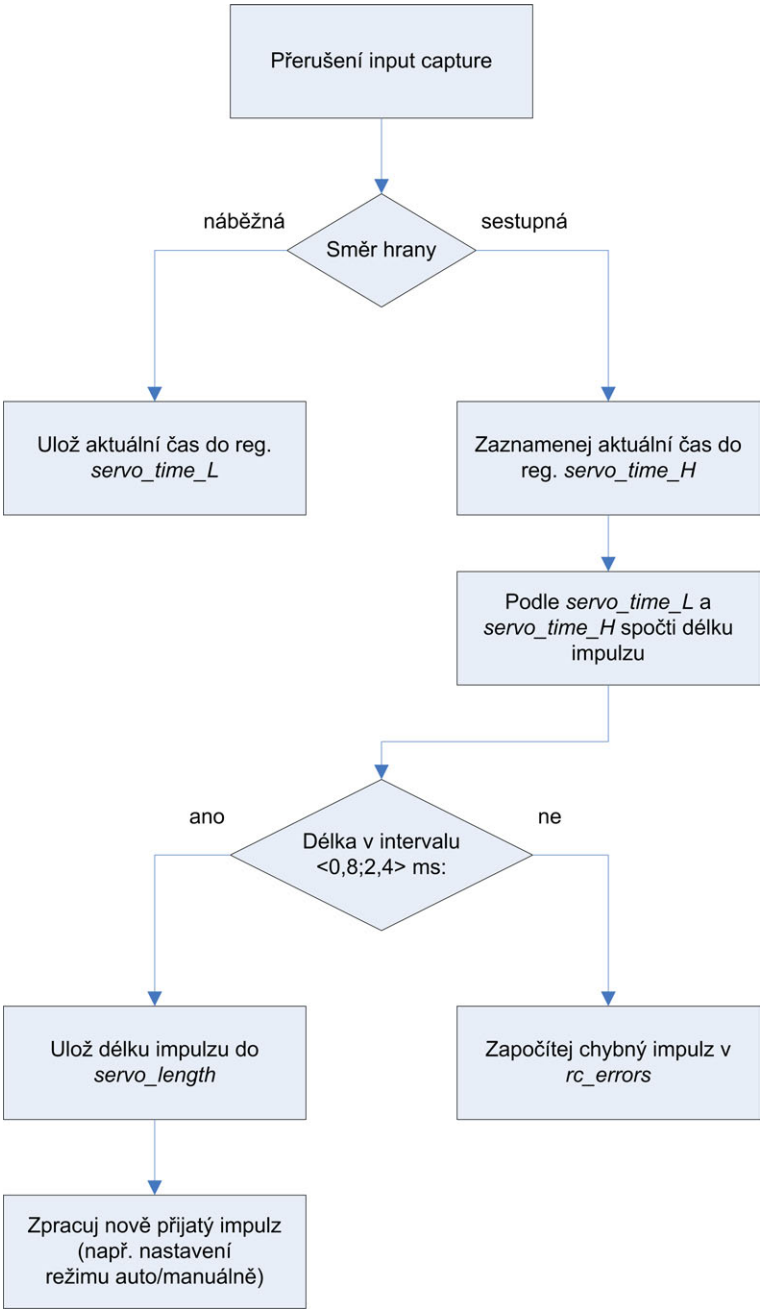
Tabulka 5.6: Čtení palubních napětí – ID 50

   Informace o napětích je odesílána mailboxem 8.

5.1.4 Použití TPU jednotek pro měření signálu z RC přijímače

Pro měření signálu modelářských serv používáme TPU jednotku v záchytném režimu (input capture). Teoreticky by nebylo nutné používat přímo TPU jednotku a stačil by volně běžící časovač a pro každý vstup jedno přerušení INTn, nastavené na sledování obou hran. Zde by však hrozilo riziko vzniku nepřesností, protože odezva na přerušení nemusí být vždy přesně stejná a měření by na ní záleželo. Tento problém je však zcela bezpečně řešen použitím

funkce input capture, protože v tomto režimu je hodnota z časovače sejmuta pomocí hardware. Následně je vyvoláno přerušení, které tuto hodnotu přečte a vyhodnotí. Zde již však přerušení v řádu jednotek taktů není nijak na závadu. Postup měření je naznačen ve vývojovém diagramu 2.2, činnost v něm naznačená se provádí nezávisle pro všech 7 vstupních kanálů.



Obrázek 5.2: Vývojový diagram měření signálu z RC soupravy

Signál z RC soupravy měříme pomocí TPU kanálů 2, 3 a 4. Vstup hodin těchto jednotek je nastaven na $\phi/16$. Jednotky jsou nastavené pro input capture náběžné i sestupné hrany z pinů TIOCA2, TIOCB2, TIOCA3, TI-

OCB3, TIOCC3, TIOCD3 a TIOCA4. Toto nastavení se provádí ve funkci `init_servo_outputs`. V okamžiku změny vstupní úrovně dojde k zapsání aktuální hodnoty časovače do registru TGR2A až TGR4A. Po vyvolání příslušného přerušení TGI2A, TGI2B, TGI3A, TGI3B, TGI3C, TGI3D resp. TGI4A je smazán příznak přerušení a hodnota časovače se zpracuje funkcí `servo_capture`. Funkce se využívá ve tvaru:

```
int servo_capture(int num,int input_mask,int timer_value);
```

Parametr `num` udává číslo kanálu, pomocí `input_mask` se funkci předá hodnota vstupu pro rozhodnutí o směru hrany. V `timer_value` je hodnota časovače přečtená pomocí TPU jednotky v okamžiku hrany. Návratová hodnota je 1 v případě, že byl právě dokončen platný impulz a může být zpracován, jinak je vrácena 0. Činnost této funkce spočívá ve vyhodnocení směru hrany a délky pulzu. V případě, že se jedná o přerušení vyvolané náběžnou hranou, aktuální čas se pouze uloží do pole `servo_time_L`. Pro sestupnou hranu se spočítá delka impulzu (vzhledem k předchozí náběžné hraně) a zkontroluje se jeho rozsah. Pokud ten odpovídá, tak se výsledná délka pulzu uloží do pole `servo_length` a funkce vrátí 1 jako příznak úspěšně načteného pulzu. Samotné zpracování přerušení je pak v tomto tvaru:

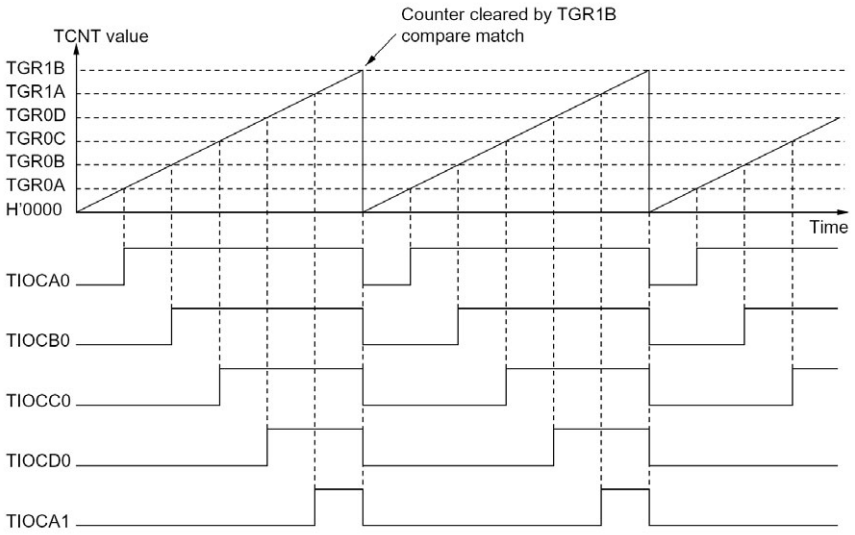
```
void servo_in_capture_2A(void) {
 *TPU_TSR2 &= ~TSR2_TGFAm;
 servo_capture(0,*DIO_PORT1 & PORT1_P16m,*TPU_TGR2A);
}
```

Pro ostatní kanály je zpracování podobné, pouze pro vstup serva číslo 7 se v případě načtení nového pulzu vyhodnocuje jeho velikost za účelem přepínání režimů AUTO/MAN.

Při dalším vyhodnocení signálů v programu tedy stačí sledovat hodnotu v poli `servo_length`, kde je možné nalézt poslední platné délky pro všechny kanály.

5.1.5 Použití TPU jednotek pro generování signálu serv

Signál pro řízení serv generujeme TPU jednotkou v režimu output compare. Je nutné zároveň také nastavit režim PWM 2 pro příslušné kanály. Použité bloky TPU jednotek jsou zřetězené, jejich časování je však propojené i s bloky zajišťujícími měření signálu z RC přijímače. Inicializaci těchto časovačů provádí funkce `init_servo_outputs`. Funkce TPU jednotky v tomto režimu je naznačena na obrázku 2.3.



Obrázek 5.3: Ukázka funkce TPU při zřetězení PWM

Perioda signálu pro serva je generována předvolbou v registru TGR4B. V programu je definována jako konstanta `servo_period`. Pro frekvenci krystalu 20 MHz a použitou konfiguraci časovačů odpovídá předepsanému času 20 ms hodnota 25000. Následkem toho je, že registr TCNT všech zřetězených TPU kanálů čítá od 0 do 25000, kde jeden krok odpovídá $0,8\mu s$. Zároveň při dosažení hodnoty 25000 dojde k nastavení všech výstupů do 1 jako inicializace cyklu. Požadované délky signálů se ukládají do registrů TGR0A, TGR0B, TGR0C, TGR0D, TGR1A a TGR1B. Hardware mikrokontroléru je pak porovnává s aktuální hodnotou čítače TCNT. Pokud dojde ke shodě, tak nastaví příslušný výstup do 0. Pro zamezení rizika vzniku nepravidelných zákmitů je přidán „double-buffering“. Přímo do registrů TPU jednotky se zapisuje pouze při konci periody generování signálu, takže v té době nemůže dojít k závažné chybě způsobené přepsáním hodnoty. Provádí se to v funkci `tpu50Hz`, která je volána právě přerušením od konce periody.

5.1.6 Měření napětí AD převodníkem

Pro měření napětí napájecích akumulátorů se využívá konfigurace AD převodníku ve SCAN režimu. Vzhledem k tomu, že napětí je snímáno pouze jednou za sekundu, je měření odstartováno z časovače funkcí `start_AD_conversion`. Následně AD převodník změří 4 kanály AN0 až AN4, uloží je do příslušných registrů a nastaví příznak přerušení ADI. Z něj se vyvolá přerušení `int_AD_convert`, které přečte naměřené hodnoty a zastaví další převod, aby

neprobíhal cyklicky. Následně se do hodnoty `ad0_val` resp. `ad1_val` uloží přepočtená hodnota napětí na hodnotu v $0,1V$. Rezervní třetí a čtvrtý kanál se nijak nepřepočítává, protože v současné verzi není využit. Převod probíhá podle vzorce:
`ad0_val = (ad0_temp * ad0_mul) >> ad0_shiftr;`,
kde `ad0_mul` a `ad0_shiftr` jsou hodnoty nastavené podle použitého odporového děliče. Při použití rezistorů s nižší přesností je vhodné tyto hodnoty doladit experimetálně. Teoreticky spočtené hodnoty jsou v tabulce 2.7. Byly určeny ze vztahů pro odporový dělič a následně přepočteny tak, aby bylo možné dělení nahradit binárním posunem (tj. dělit celou mocninou 2).

Konstanta	Hodnota
`ad0_mul`	37
`ad0_shiftr`	9
`ad1_mul`	37
`ad1_shiftr`	8

Tabulka 5.7: Konstanty pro přepočet napětí

5.1.7 Komunikace po RS232

Komunikace po lince RS232 je použita především pro účely ladění a zjednodušení práce s modelem. V zadání práce nebyla požadována, ale vzhledem k její praktičnosti byla také implementována.

Způsob práce s ladícím rozhraním je uveden v kapitole A. Realizován je pomocí jednoduchého stavového automatu. Ten po přijetí příslušného znaku pro jednodušší činnosti pouze změní hodnotu bitu, která určuje, zda se má vysílat příslušná ladící hláška, případně vypíše odpovídající text. Pro složitější činnosti, jako je nastavení hodnot pro servomotory, nebo zadání masky přijme předpokládaný počet znaků, a pak je zpracuje funkcí `sscanf`.

V prostředí OMK se sériové rozhraní inicializuje příkazem `sci_rs232_setmode`, zároveň je nutné před použitím funkcí typu `printf` určit, na který sériový port mají tyto funkce posílat výstup. Ten je možné nastavit v proměnné sci_rs232_chan_default. Pro příjem znaků je použita funkce `sci_rs232_recch`, protože čte znaky ze sériového portu neblokujícím způsobem. V případě pokusu o čtení znaku, když žádný není k dispozici vrátí tato funkce -1.

5.2 Hardware

5.2.1 Popis zapojení

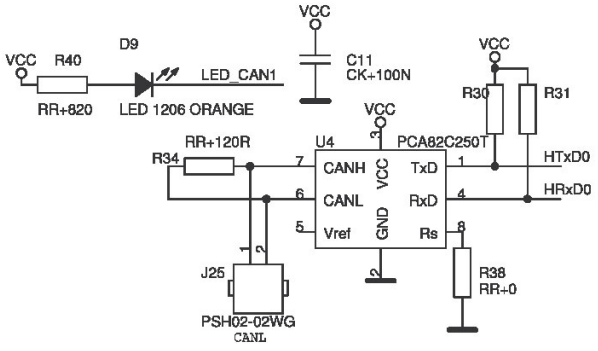
Schéma zapojení jednotky řízení servomotorů je v příloze A. Zapojení jednotky vychází z doporučení výrobce mikrokontroléru, specifikovaného v katalogovém listu [7]. Jako zdroj hodin využíváme krystal 10 MHz, který je však v mikrokontroléru násoben PLL jednotkou na hodinovou frekvenci 20 MHz.

Signál reset však na rozdíl od doporučeného zapojení generujeme samostatným obvodem MAX1232. Jeho dokumentaci je možné nalézt v [9]. Tento obvod v sobě implementuje následující funkce:

- generování signálu reset (možné v pozitivní i negativní úrovni)
- kontrola napájecího napětí Vcc +5V
- funkce watchdog
- tlačítko reset (v tomto případě není využité)

Podrobnější informace o nastavení tohoto obvodu, týkající se zabezpečení aplikace, jsou v kapitole 2.2.3.

Pro připojení sběrnice CAN je využit budič PCA82C250T. Podrobně je popsaný v katalogovém listu [10]. Rezistor R34 je určen pro zakončení sběrnice, můstek R38 nastavuje režim činnosti PCA82C250T, v připadě této konstrukce má být osazen. Zapojení CAN budiče je na obrázku 2.4.



Obrázek 5.4: Zapojení CAN budiče

Pro možnost dalšího rozšíření je na konektor J2, na plošném spoji označený jako GPIO, vyvedeno 6 datových vodičů, napájecí napětí +5V a zem.

Dále jsou dva vodiče připojené přes odporový dělič a RC filtr na A/D převodník. V případě potřeby je možné tento konektor využít jak pro připojení digitálních, tak analogových obvodů.

Jednotka je také připravena pro využití sériové paměti EEPROM, vhodné například pro uložení konfiguračních údajů. Využívá se paměť typu 93C46, viz [11]. V současné verzi systému však není zapotřebí, proto zatím nebyla osazena a v programovém vybavení zatím nemá podporu.

5.2.2 Konfigurační propojky v jednotce

Pro možnost nastavení režimů práce je deska vybavena několika konfiguračními propojkami. Jejich polohu najdeme v osazovacím výkresu na obrázku A.4 uvedeném v příloze A. Jejich funkce je následující:

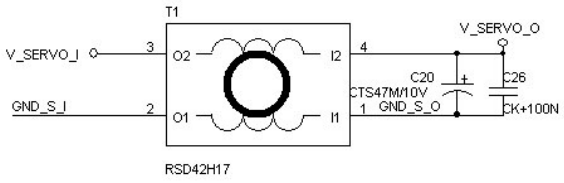
- J18 – výstup signálu reset. Při normálním provozu má být zapojená, slouží k zamezení resetu procesoru v době, když není možné zajistit obnovování watchdogu. Tato situace nastává například při zavádění programu.
- J20 – signál pro obnovování watchdogu. Obvykle by měla být propojená, jejím rozpojením můžeme vyvolat reset mikrokontroléru.
- J21 – vodič FWE. Normálně je propojená a zamezuje zápisu do paměti flash. Při rozpojení se po restartu mikrokontroléru spustí program pro zavádění software.
- J22 – volba režimu MD2. Má být rozpojená, pouze v případě nutnosti nahrát nový zavaděč programu (do nového mikrokontroléru) se spojí.

5.2.3 Prvky pro zvýšení odolnosti proti rušení

Z hlediska bezpečnosti celé soustavy je jednotka řízení servomotorů naprosto kritickou záležitostí. Při jejím selhání by s velkou pravděpodobností došlo ke zničení velké části zařízení. Proto bylo nutné na spolehlivou funkci desky dát maximální důraz. To také znamená snažit se minimalizovat vlivy rušení na mikrokontrolér i ostatní prvky.

Závažným problémem během vývoje bylo rušení působící na RC přijímač. Při vývoji jsme využívali starší typ s analogovým přenosem, který měl s rušením zásadní problémy. Projevovalo se například velmi silným zaškubáváním servomotorů při přiblížení k počítači, při zapnutí elektrických spotřebičů na stole, nebo rušením způsobeným od spínání servomotorů. Bylo jednoznačně prokázáno, že se nejednalo o vliv desky řízení serv, ale špatnou

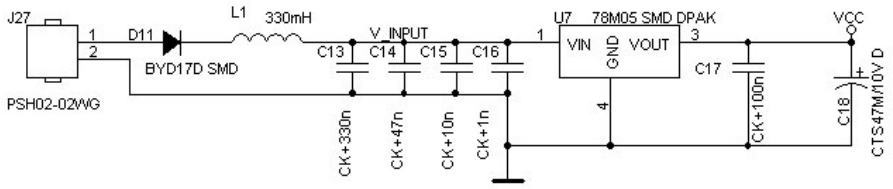
vlastnost přijímače. Tento vliv však bylo nutné minimalizovat. To bylo provedeno vložením oddělovací cívky (typ RSD42H17, ve schématu označeno T1) do přívodu napájení RC přijímače, aby se snížil alespoň přenos rušení napájecími vodiči. Zapojení je naznačeno na obrázku 2.5, kde je napájení RC přijímače připojeno na stranu `V_SERVO_I` a serva s akumulátory jsou na straně `V_SERVO_O`. Pro samotný model vrtulníku byl však použit modernější RC přijímač s digitálním přenosem, který danými problémy viditelně netrpí.



Obrázek 5.5: Oddělení napájení RC soupravy od servomotorů

Dále byl při návrhu desky dáván velký důraz na správnou distribuci napájení. V převážné většině plochy spodní vrstvy desky je zem v podobě rozlité mědi, ve volných prostorech se rozlitá měď nachází i nahoře. Zároveň se napájení filtruje velkým množstvím blokovacích kondenzátorů 100nF. Pro zvýšení robustnosti a odolnosti byl použit lineární stabilizátor typu 7805. Vzhledem k odběru desky a charakteru napájení zde spínaný zdroj není nutný, pouze by zbytečně zvyšoval složitost a možnou poruchovost systému. Napájecí zdroj však také obsahuje velké množství filtračních kondenzátorů různých hodnot, aby se maximálně zamezilo riziko zavlečení rušení do mikrokontroléru.

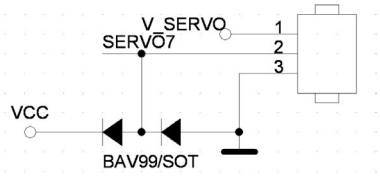
Dalším významným prvkem pro zvýšení odolnosti proti rušení je indukčnost L11. Ve spojení s kondenzátory C13 až C16 výrazně omezí případné špičky v napájecím napětí. Jako ochranu před přepólováním jednotky jsme použili diodu D11.



Obrázek 5.6: Zapojení stabilizátoru napětí

Pro snížení rizika zavlečení napětí většího než je napájecí, nebo naopak záporného, byly na významné vstupy a výstupy mikrokontroléru doplněny ochranné diody podle obrázku 2.7.

Dalším důležitým prvkem pro zvýšení spolehlivosti je použití obvodu watchdog s funkcí monitoringu napájecího napětí. V této konstrukci se využívá obvod MAX1232. Pomocí konfiguračních pinů je na něm možné nastavit toleranční rozsah pro napájecí napětí i nejdelší povolený čas pro refresh watchdogu. V použité konstrukci je nastaveno minimální napájecí napětí na 4,5 V a maximální čas obnovy watchdogu na 150 ms. Samozřejmě je však nutné především dbát na to, aby k resetům díky watchdogu nedocházelo a jeho aktivaci považovat za poslední záchranu a významné varování.



Obrázek 5.7: Zapojení ochranných diod

5.2.4 Návrh plošného spoje

Deska plošných spojů byla navržena ve velikosti 100 x 160 mm. Samotné zapojení by bylo možné realizovat na podstatně menší desce. V modelu vrtulníku je však tato deska připevněná na desce hlavního řídicího počítače, která má stejné rozměry. Zmenšením desky řízení serv by se tedy montáž pouze zkomplikovala, nemělo proto smysl se snažit její rozměry minimalizovat.

Osazení plošného spoje je provedeno technologií SMD. Pouze v případě konektorů byla z důvodu vyšší mechanické odolnosti záměrně použita klasická technologie. Z dalších součástek je v klasické montáži použit krystal a oddělovací indukčnost napájení RC přijímače. Rezistory a kondenzátory byly preferovány ve velikosti 1206.

Plošný spoj byl vyroben jako dvoustranný s prokovy bez servisního potisku. Vzhledem k rozměrům pouzdra mikrokontroléru deska spadá do páté třídy přesnosti. Takto jemné vodiče se však využijí pouze na přívodech k mikrokontroléru. Jinde jsou využity větší šířky spojů, aby se snížilo riziko špatných propojení z důvodů vlasových trhlinek, podleptání při výrobě, případně poškození při dodatečných úpravách. Při návrhu jsme postupovali podle doporučení uvedených v [4].

Kapitola 6

Závěr

Cílem této diplomové práce bylo vyvinout řídicí jednotku servomotorů pro model vrtulníku, vznikající na Katedře řídicí techniky. Úloha spočívala v prostudování stávajících požadavků a specifikací, seznámení se s mikrokontrolérem Renesas (dříve Hitachi) H8S2638 a vývoji hardware i programového vybavení pro tuto jednotku. Všechny úkoly se podařilo splnit a byly ověřeny provozem jednotky řízení serv ve spolupráci s navazujícím systémem.

Jednalo se o týmový projekt realizovaný v týmu několika studentů. Všichni zúčastnění získali další cenné zkušenosti s prací v týmu, vzájemném předávání si poznatků i nutnosti časového sladění průběhu projektu. Vzhledem k zaměření této práce však docházelo k širší spolupráci v rámci katedry, jak mezi lidmi zaměřenými na operační systém Linux, tak s dalšími diplomanty pracujícími na mikrokontroléru H8S2638.

Obrázek 6.1: Fotografie ze zkušebního letu

Testy systému probíhaly v několika etapách. Nejprve byl vždy testován každý element systému samostatně. V této fázi byly nalezeny a odstraněny všechny závažné problémy. Další fází testů byly zkoušky celé soustavy. Zde se již objevily jen menší závady, které by již nevedly k nebezpečným situacím při provozu vrtulníku, ale pouze by zablokovaly některou z jeho funkcí. Poslední fází testů byly letové zkoušky, které kompletně ověřily provozuschpnost systému. Zároveň již byla v jejich průběhu zaznamenávána data, která je možné využít pro následnou identifikaci systému a návrh řídicích algoritmů. Dokumentace v podobě krátkého filmu ze zkušebního letu je k dispozici na přiloženém CD.

Literatura

[1] Kernighan, B., Ritchie, D. *Programovací jazyk C* Bratislava, Praha : Alfa, 1988. 249 s.

[2] Etschenberger, K. *Controller Area Network* Weingarten : IXXAT Press, 2001.

[3] Kocourek, P., Novák, J. *Přenos informace* Praha : ČVUT, 2004. 164 s. ISBN 80-01-02892-5.

[4] Záhlava, V. *Metodika návrhu plošných spojů* Praha : ČVUT, 2000. 81 s. ISBN 80-01-02193-9.

[5] Záhlava, V. *OrCad pro Windows : praktický průvodce návrháře* Praha : Grada, 1999. 121 s. ISBN 80-7169-876-8.

[6] *Vývojový systém OMK* [online]. Poslední revize 2006-03-26 [cit. 25. 5. 2006]. <http://rtime.felk.cvut.cz/hw/>

[7] *Katalogový list mikrokontroléru Renesas H8S/2638* [online]. <http://www.renesas.com/>

[8] *Katalogový list jednotky inerciální navigace Bec Nav BP3010* [online]. <http://www.bec-nav.de/>

[9] *Katalogový list obvodu MAX1232* [online]. <http://www.maxim-ic.com/>

[10] *Katalogový list obvodu PCA82C250* [online]. <http:///www.semiconductors.philips.com/>

[11] *Katalogový list obvodu 93C46* [online]. <http:///www.atmel.com/>

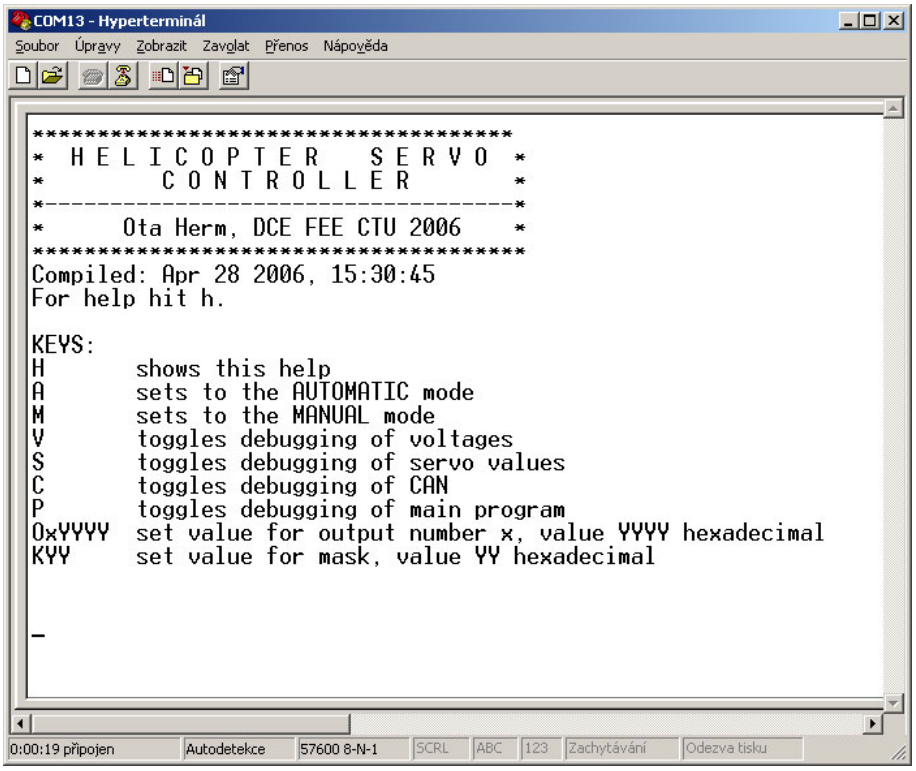
Dodatek A

Servisní komunikace po sériovém portu

Tato komunikace probíhá na portu COM1 (označení na desce plošných spojů), což odpovídá pinům TXD0 a RXD0 na mikrokontroléru. Jako uživatelské rozhraní lze využít libovolný terminálový program. Komunikační parametry jsou 57600 bitů za sekundu, 8 datových bitů a 1 stopbit. Jednotka po přijetí znaku H vypíše nápovědu pro ovládání programu. V případě stisknutí neznámého znaku napoví, že nápovědu vytiskne po stisknutí H. Komunikace je tedy poměrně intuitivní. Stiskem příslušných kláves je možné zapínat/vypínat jednotlivé ladící hlášky. K dispozici jsou ladící hlášky pro CAN, PWM vstupy a výstupy, výstupní hodnoty AD převodníku a stav programu. Dále je možné desku přepínat mezi automatickým a manuálním režimem, nastavit hodnoty pro serva a zadat hodnotu masky. Zároveň je v hlášce s nápovědou uvedeno datum kompilace projektu, aby šlo snadno ověřit, že mikrokontrolér obsahuje aktuální verzi programu. Ladící hlášky se obnovují průběžně v intervalech cca 1s.

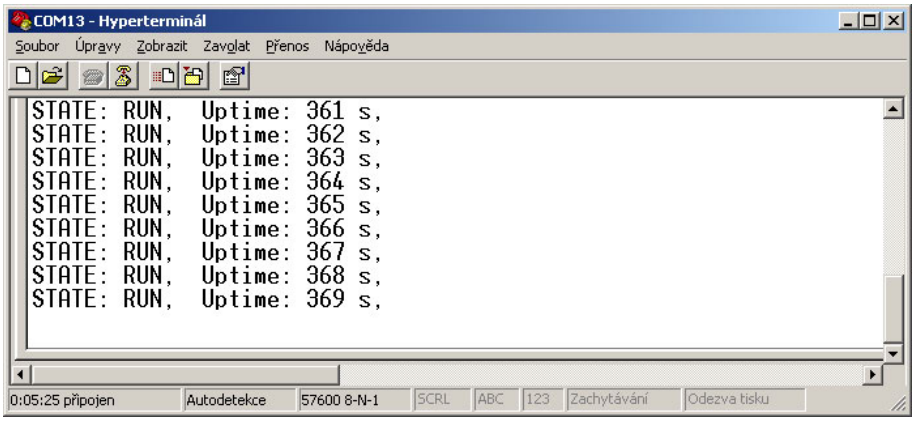
Ukázka použití ladícího rozhraní

1. Odesláním znaku „H“ zobrazíme nápovědu



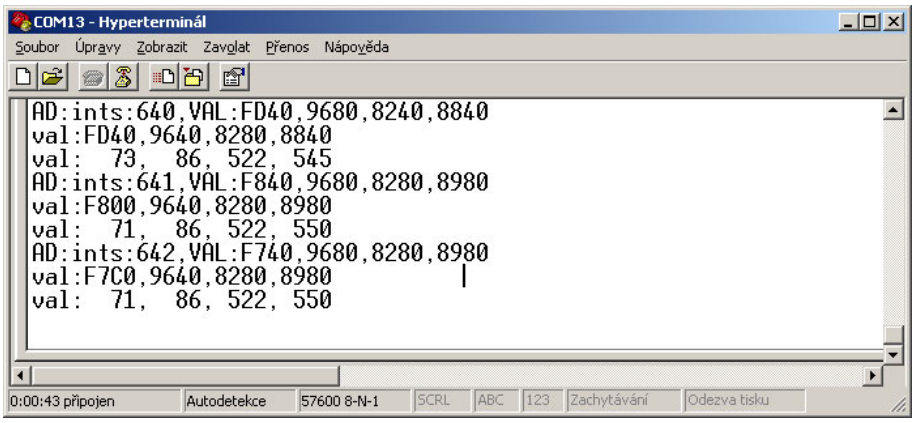
Obrázek A.1: Ukázka nápovědy v ladícím rozhraní

2. Zadáním znaku „P“ zapínáme, případně vypínáme pravidelné zobrazení hlavních informací o programu. To je především režim jednotky (AUTO/MAN) a počet sekund od resetu desky.



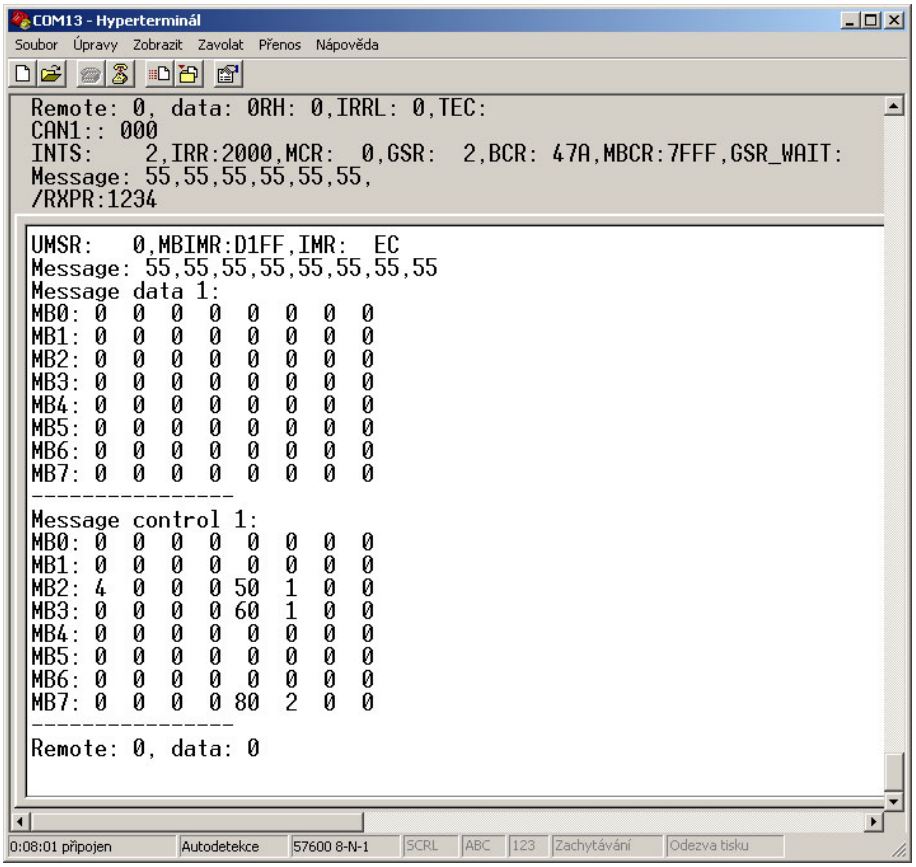
Obrázek A.2: Servisní zobrazení – základní systémové informace

3. Pomocí znaku „V“ se lze přesvědčit o měření napětí pomocí AD převodníku.



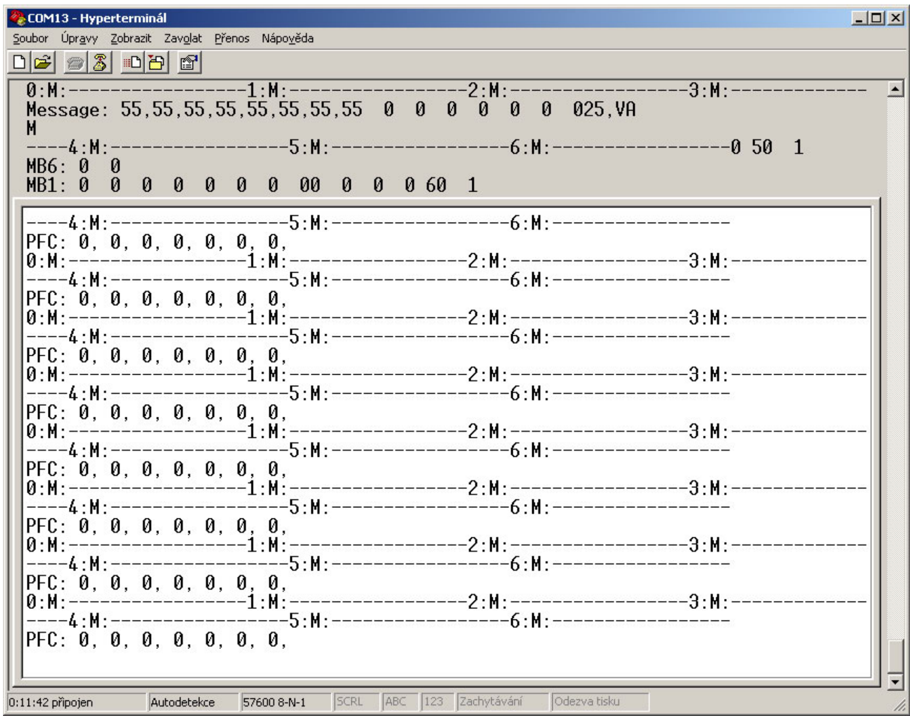
Obrázek A.3: Servisní zobrazení – údaje AD převodníku

4. Po stisknutí „C“ je možné sledovat hodnoty některých podstatných registrů řadiče CAN.



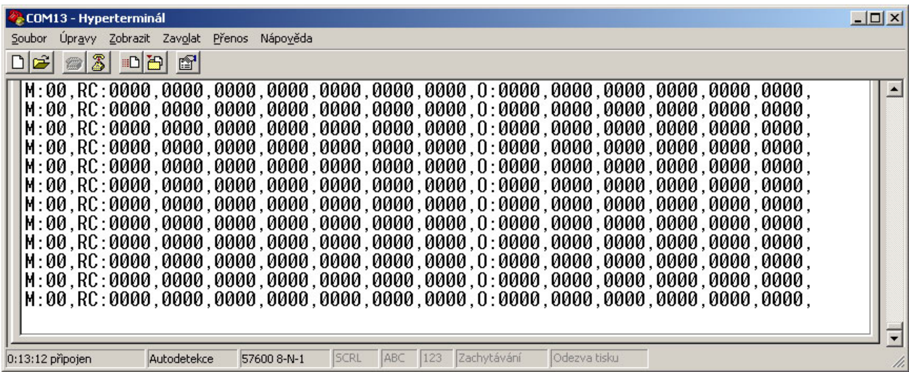
Obrázek A.4: Servisní zobrazení – údaje řadiče CAN

5. Znakem „S“ lze zapnout, případně vypnout zobrazení hodnot pro serva



Obrázek A.5: Servisní zobrazení – údaje pro serva

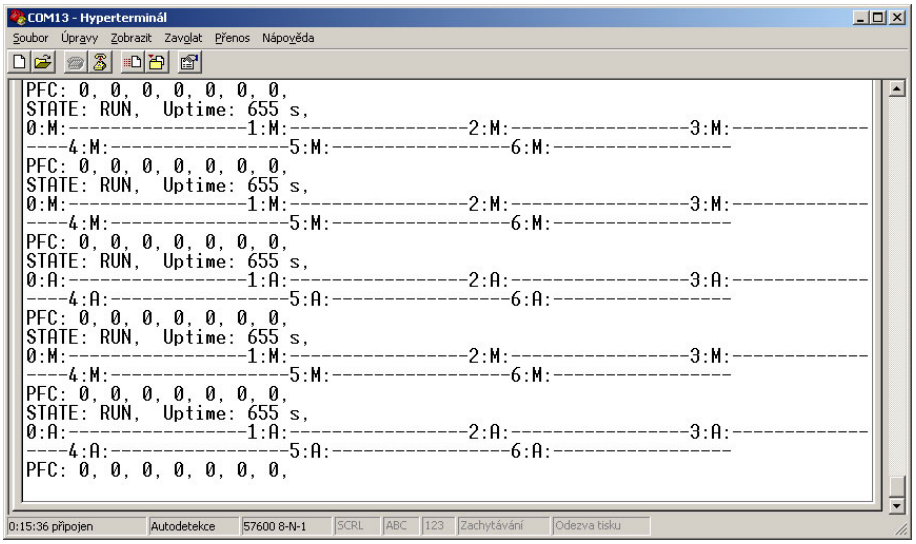
6. Zkrácenou stavovou informaci získáme pomocí „?“.



Obrázek A.6: Servisní zobrazení – zkrácená stavová informace

7. Stiskem „A“ je možné provést přepnutí do automatického režimu, pouze však v případě, že příkaz z RC soupravy nepožaduje manuální režim.

Stiskem „M“ se jednotka přepne do manuálního režimu, platí pro ní stejná podmínka jako pro „A“.



Obrázek A.7: Servisní zobrazení – přepínání automaticky/ručně

8. Maska se přes servisní rozhraní dá zadat odesláním znaku „K“ a následujících dvou znaků v šestnáctkové soustavě reprezentujících hodnotu masky.

9. Hodota příslušného serva může být zadána odesláním znaku „O“ následovaného číslem serva (číslováno od 0) a pak hodnotou k zapsání na servo čtyřciferně šestnáctkově. Tento příkaz bude proveden okamžitě, není závislý na synchronizační zprávě na sběrnici CAN.

Toto rozhraní je uvažované pouze jako servisní, pro otestování a seznámení se s funkcí desky. V předložené podobě není vhodné pro trvalý provoz, a to především proto, že datové zprávy nejsou vhodně zabezpečeny proti chybám (zabezpečení by však v tomto případě znesnadňovalo ruční použití pomocí terminálového programu). Pro trvalý provoz by tedy zprávy musely být ošetřeny minimálně kontrolním součtem, lépe však pomocí CRC[1]. Dále by pravděpodobně měly být zahrnuty nějaké synchronizační znaky (počáteční, koncový), aby se v případě ztráty nebo poškození zprávy snadno komunikace obnovila a nepokazil se tak celý přenos. Je však pravděpodobné, že tyto

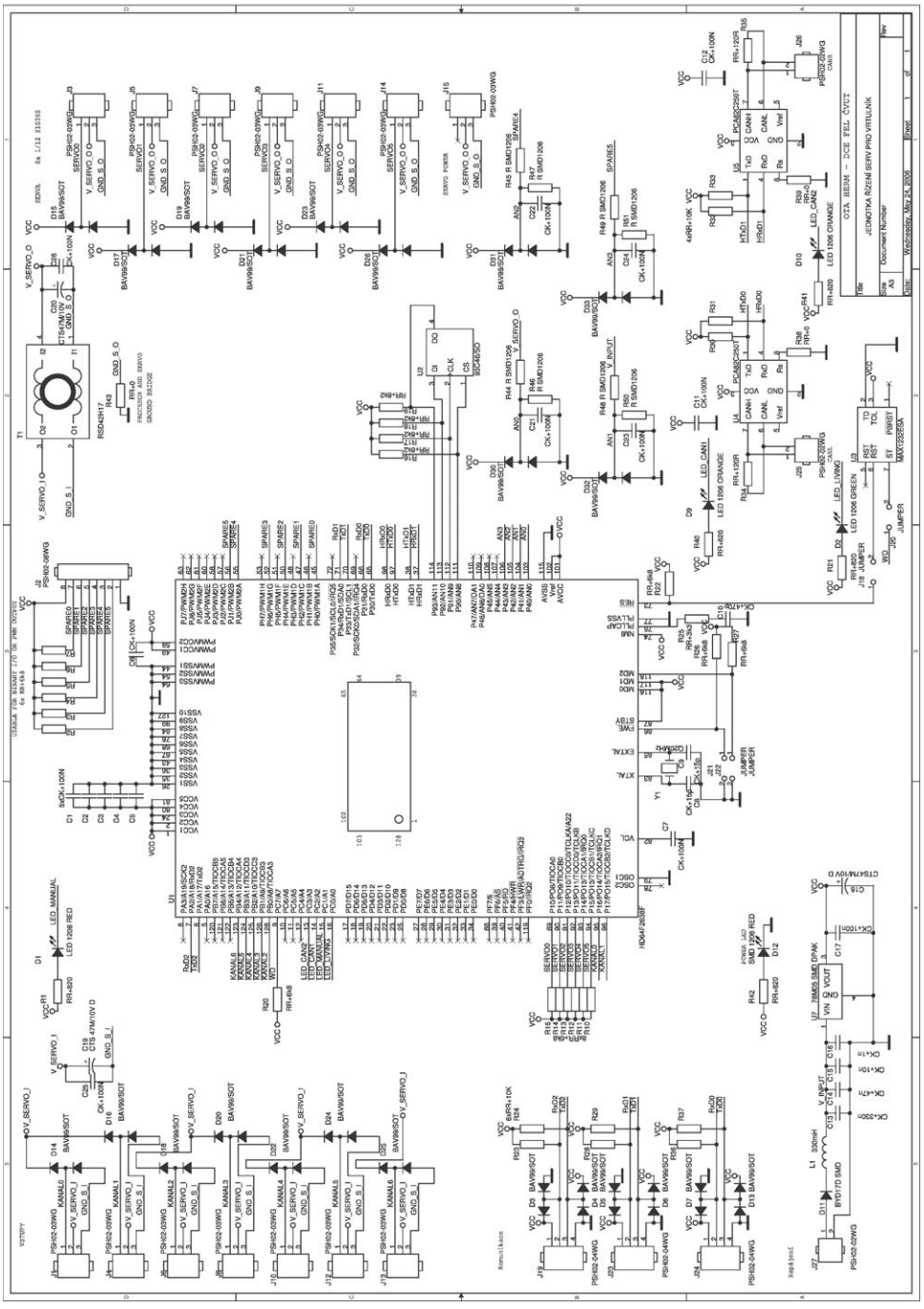
[1] Cyclic Redundancy Check

úpravy budou v budoucnu provedeny, aby bylo možné dále desku používat i v jiných systémech, kde CAN sběrnice není k dispozici a jednotka řízení servomotorů bude k řídicímu počítači připojena přímo sériovou linkou.

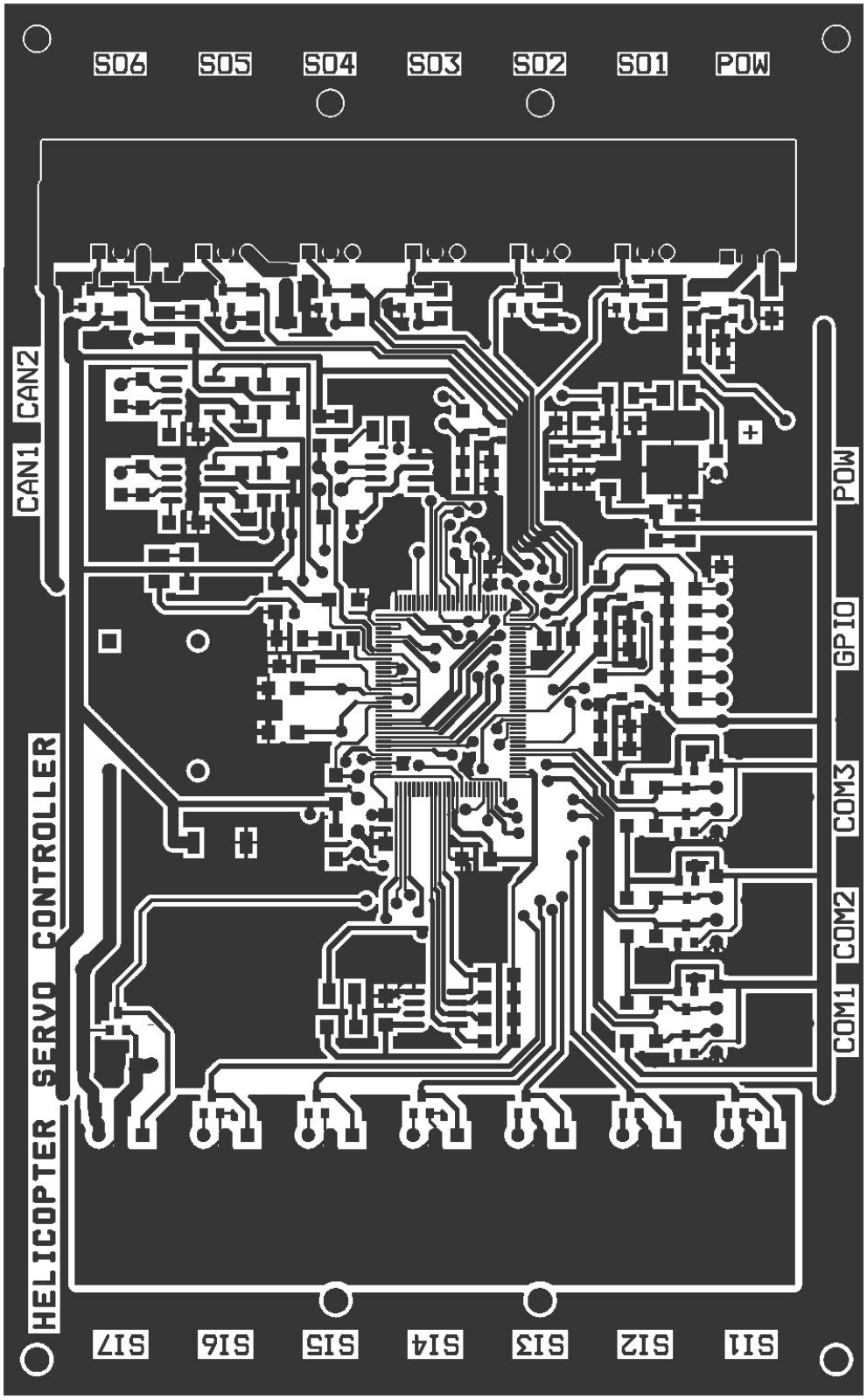
Dodatek B

Schéma zapojení a podklady plošného spoje

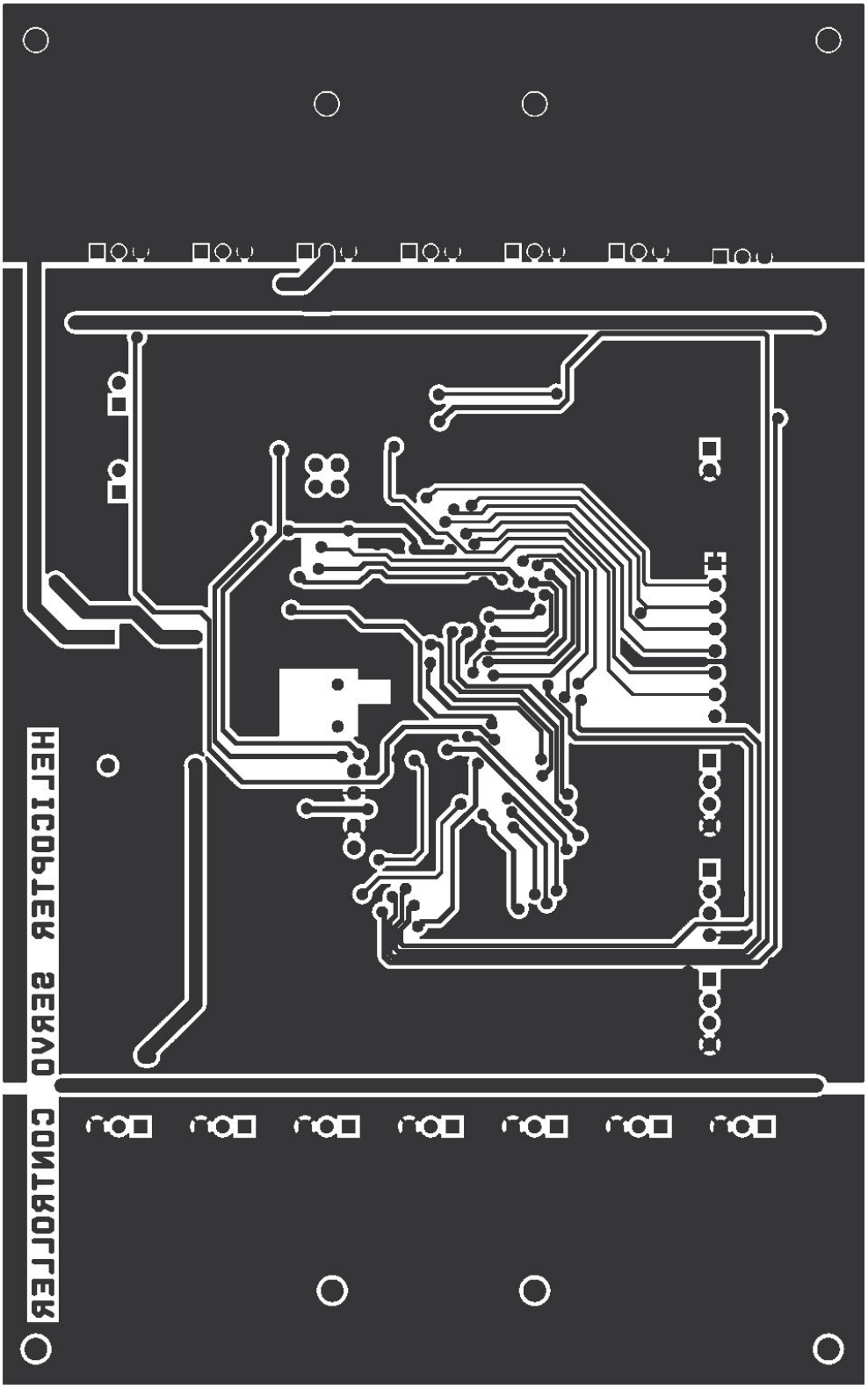
Tato příloha obsahuje schéma zapojení a náhled předloh pro tvorbu desky plošných spojů. Ty se také nacházejí v digitální podobě, vhodné pro výrobní použití, na přiloženém CD.



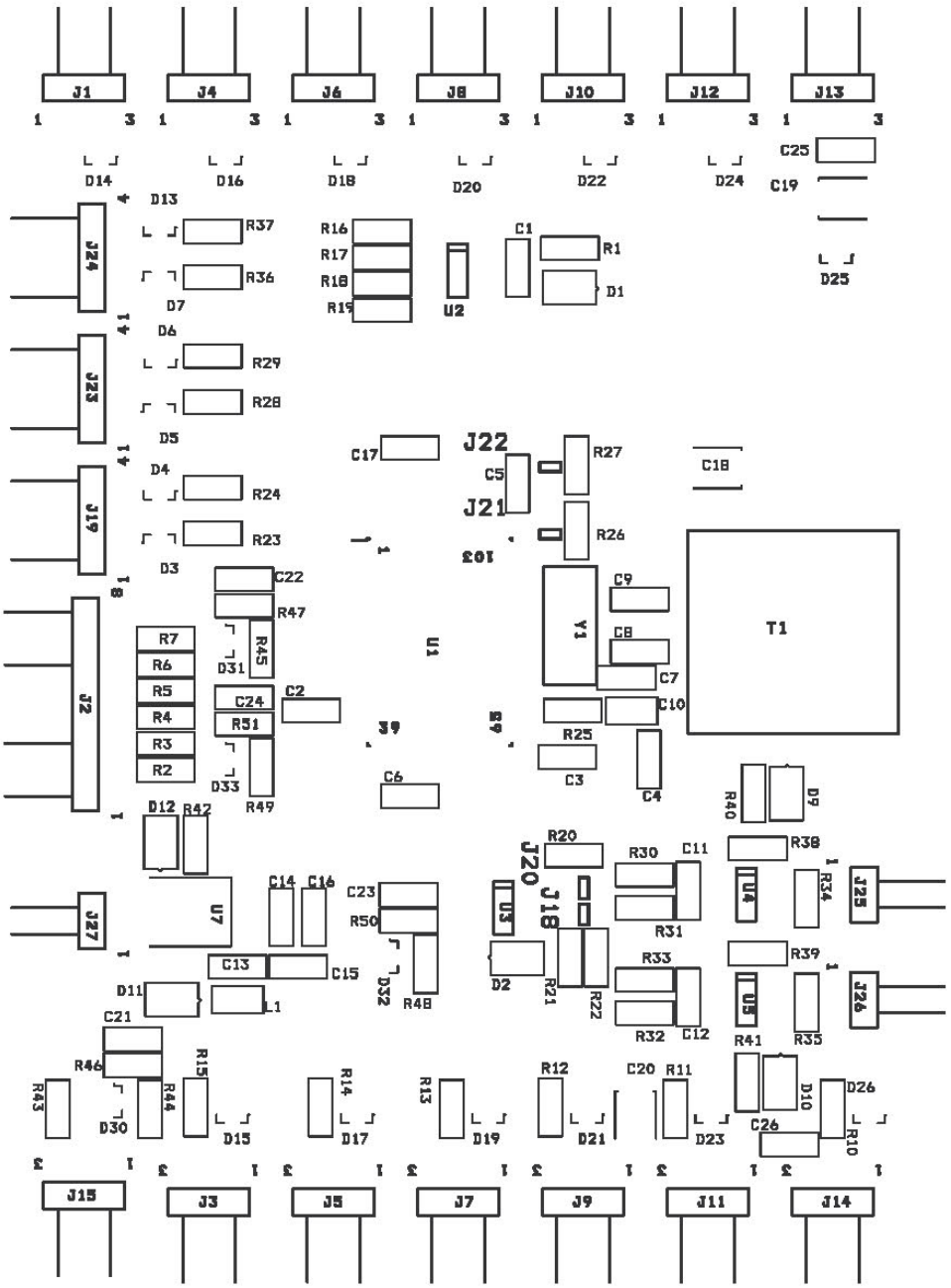
Obrázek B.1: Schéma zapojení jednotky řízení servomotorů



Obrázek B.2: Předloha pro stranu součástek plošného spoje



Obrázek B.3: Předloha pro stranu spojů plošného spoje



Obrázek B.4: Osazovací výkres plošného spoje

Dodatek C

Obsah přiloženého CD

Datový nosič přiložený k této diplomové práci obsahuje tyto soubory:

- *dp_herm_2006.pdf* – text diplomové práce
- *servodrv.tar.gz* – archív s programem pro jednotku řízení servomotorů a prostředím OMK
- *schema.pdf* – schéma zapojení jednotky (vhodné pro tisk)
- adresář *datasheety* – katalogové listy důležitých součástek
- adresář *video* – adresář s videoprezentací projektů