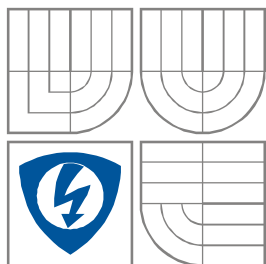


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

SYNCHRONNÍ ŘÍZENÍ SERVO MOTORU
PO SBĚRNICI CAN
SYNCHRONOUS SERVO MOTOR CONTROL WITH CAN BUS

DOPLOMOVÁ PRÁCE
MASTER'S THESIS

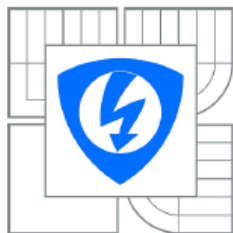
AUTOR PRÁCE
AUTHOR

Bc. JAN KOHNHEISER

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ALEŠ POVALAČ, Ph. D.

BRNO, 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Jan Kohnheiser
Ročník: 2

ID: 125483
Akademický rok: 2013/2014

NÁZEV TÉMATU:

Synchronní řízení servomotorů po sběrnici CAN

POKYNY PRO VYPRACOVÁNÍ:

Analyzujte požadavky pro synchronní řízení servomotorů po sběrnici CAN v průmyslu. Uvažujte o nasazení v aplikacích jako obráběcí stroj (třosá fréza), manipulátor s obsluhou (čistící stroje, vysokozdvížené vozíky) a automatický manipulátor s předem definovanou trajektorií (automatický zakladač). Zohledněte zpětnou vazbu z externích čidel polohy, manuální řízení, bezpečnostní funkce a provozní diagnostiku. Porovnejte vlastnosti průmyslových sběrnic pro komunikaci s nadřazeným řídicím systémem PC nebo PLC.

Pro vybranou aplikaci vytvořte programové vybavení řídicího systému (Indexeru) na bázi procesoru ARM Cortex-M3 pro synchronní řízení servomotorů. Pro odladění a vyzkoušení použijte modul LCD&Control board 2011 firmy PMControl a servomotory PMControl PMC-G6X nebo PMC-T4/H4. Zařízení připojte k nadřazenému PC s vhodným programovým vybavením.

DOPORUČENÁ LITERATURA:

[1] VOSS, W. A Comprehensible Guide to Controller Area Network. Greenfield, MA: Copperhill Media, 2005.

[2] Katalogové listy motorů [online]. PMControl s.r.o., 2012 - [cit. 4.6.2013]. Dostupné z [www: http://www.pmcontrol.cz/cz/pdf/katalog/ecm-datasheets-2012.pdf](http://www.pmcontrol.cz/cz/pdf/katalog/ecm-datasheets-2012.pdf)

Termín zadání: 10.2.2014

Termín odevzdání: 23.5.2014

Vedoucí práce: Ing. Aleš Povalač, Ph.D.

Konzultanti diplomové práce: Ing. Tomáš Jirásek

doc. Ing. Tomáš Kratochvíl, Ph.D.

Předseda oborové rady

Abstrakt

Tato diplomová práce se zabývá řízením motorů po sběrnici CAN. Teoretická část se věnuje v průmyslu užívanými sběrnicemi, motorům využívaným jako servopohony a základním aplikacím servomotorů. U sběrnic jsou zmíněny základní parametry, mechanismy komunikace a srovnání. Jednotlivé typy motorů jsou popsány z pohledu konstrukce a požadavků na řízení. Praktická část se zabývá využitím servomotorů v aplikaci obráběcího stroje.

Klíčová slova

sběrnice, CAN, servomotor, synchronní řízení

Abstract

This master's thesis is focused to engine control by CAN bus. Theoretical part describes industrial control buses, servo-traction engines and their application. There are mentioned principal and basic parameters of common control buses and their comparison. Each engine type is described in side of construction and control requirements. Practical part of of this paper works on servo-engines in metal cutting machine applications.

Key words

bus, CAN, servo motor, synchronous control

KOHNHEISER, J. Synchronní řízení servomotorů po sběrnici CAN. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 40 s. Vedoucí diplomové práce Ing. Aleš Povalač, Ph.D..

Prohlášení

Prohlašuji, že svou diplomová práci na téma Synchronní řízení servomotorů po sběrnici CAN jsem vypracoval samostatně pod vedením vedoucího diplomová práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomová práce dále prohlašuji, že v souvislosti s vytvořením této diplomová práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

Obsah

SEZNAM OBRÁZKŮ	VII
SEZNAM TABULEK	IX
ÚVOD	1
1 TEORETICKÁ ČÁST	2
1.1 SBĚRNICE	2
1.1.1 <i>Sériové porty</i>	4
1.1.1.1 RS - 232	4
1.1.1.2 RS - 422 a RS - 485	5
1.1.2 <i>Profibus</i>	7
1.1.3 <i>Sběrnice CAN (Controler Area Network)</i>	8
1.1.3.1 CANopen	12
1.1.3.2 DeviceNet	14
1.1.4 <i>Ethernet</i>	15
1.1.5 <i>GPIB</i>	16
1.1.6 <i>FIP</i>	16
1.1.7 <i>ModBus</i>	16
1.2 SERVOMOTORY	17
1.2.1 <i>Stejnoseměrný motor - kartáčový</i>	17
1.2.2 <i>Krokové motory</i>	19
1.2.3 <i>Elektronicky komutované motory - BLDC</i>	21
1.2.4 <i>Synchronní motor s permanentními magnety - PMSM</i>	23
1.3 OBLASTI MOŽNÉHO VYUŽITÍ SERVOMOTORŮ	24
1.3.3 <i>Pohony s jedním servomotorem</i>	24
1.3.4 <i>Pohony s více servomotory</i>	25
2 PRAKTICKÁ ČÁST	26
2.1 HARDWARE	27
2.1.1 <i>Ruční ovladač obráběcího stroje</i>	28
2.1.1.1 Nastavovací kolečko	29
2.1.2 <i>Propojovací skříňka</i>	30
2.1.3 <i>Servomotory</i>	31
2.2 SOFTWARE	32
2.2.1 <i>MCU</i>	32
2.2.2 <i>Struktura SW</i>	32
2.2.3 <i>Uživatelský interface (GUI)</i>	33
2.2.3.1 Ovladač v manuálním režimu	33
2.2.3.2 Ovladač v automatickém režimu	34
2.2.3.3 Bezpečnostní zastavení	36
2.2.4 <i>Výpočty pohybu</i>	37
2.2.4.1 Výpočty v manuálním režimu	37
2.2.4.2 Výpočet požadované polohy automatickém režimu	38
3 ZÁVĚR	40
POUŽITÁ LITERATURA	41

Seznam obrázků

OBR. 1.1	VRSTVOVÝ MODEL OSI [1]	2
OBR. 1.2	KOMUNIKACE SÉRIOVÝCH PORTŮ [11]	5
OBR. 1.3	A) 9PINOVÝ D-SUB KONEKTOR B) PŘEVODNÍK USB NA SÉRIOVÝ PORT [6]	5
OBR. 1.4	PŘÍKLAD KOMUNIKACE S PŘENOSEM INFORMACÍ DIFERENCIÁLNÍM NAPĚTÍM [12]	6
OBR. 1.5	ZAPOJENÍ RS - 422 [6].....	6
OBR. 1.6	ZAPOJENÍ RS - 485 [6].....	6
OBR. 1.7	KOMUNIKACE V PROFIBUS [9]	7
OBR. 1.8	KOMUNIKACE V CAN [13]	9
OBR. 1.9	USPOŘÁDÁNÍ SÍTĚ CAN [13].....	9
OBR. 1.10	FORMÁT STANDARTNÍ ZPRÁVY CAN [13]	10
OBR. 1.11	OBJEKTOVÝ MODEL ZAŘÍZENÍ V SÍTI DEVICENET[1]	14
OBR. 1.12	KONSTRUKCE DC MOTORU S BUZENÍM PERMANENTNÍM MAGNETEM [15].....	17
OBR. 1.13	ŘÍZENÍ OTÁČEK DC MOTORU POUZE V JEDNOM SMĚRU [15]	18
OBR. 1.14	ŘÍZENÍ DC MOTORU A) BIPOLÁRNÍM ZDROJEM B) MŮSTKOVĚ [15]	18
OBR. 1.15	ŘÍZENÍ POMOCÍ PWM [15].....	18
OBR. 1.16	RELUKTANČNÍ KROKOVÝ MOTOR[3] A) JEDNODUCHÝ B) VÍCE KROKŮ NA OTÁČKU	19
OBR. 1.17	KROKOVÝ MOTOR S PERMANENTNÍM MAGNETEM	19
OBR. 1.18	HYBRIDNÍ KROKOVÝ MOTOR [15]	20
OBR. 1.19	MOŽNÁ ZAPOJENÍ CÍVEK HYBRIDNÍCH KROKOVÝCH MOTORŮ [15].....	20
OBR. 1.20	PRINCIP MIKROKROKU KROKOVÝCH MOTORŮ[18]	21
OBR. 1.21	ŘEZ BLDC MOTOREM - RŮZNÝ POČET PÓLŮ STATORU [20]	21
OBR. 1.22	ROTOR BLDC MOTORU [17] A) S MAGNETY NA POVRCHU B) S MAGNETY UVNITŘ	22
OBR. 1.23	PODÉLNÝ ŘEZ BLDC MOTOREM I S HALLOVÝMI SONDAMI [17].....	22
OBR. 1.24	PŘESNÉ ZASTAVENÍ	24
OBR. 1.25	HMOTNOSTNÍ PLNĚNÍ	25
OBR. 2.1	POHYBY, KTERÝ ZAJIŠŤUJÍ SERVOMOTORY	26
OBR. 2.2	BLOKOVÉ SCHÉMA ZAPOJENÍ POHONU OBRÁBĚČÍHO STROJE	27
OBR. 2.3	RUČNÍ OVLADAČ - BLOKOVÉ SCHÉMA.....	28
OBR. 2.4	NASTAVOVACÍ KOLEČKO	29
OBR. 2.5	PRŮBĚH SIGNÁLŮ PRO DETEKCI QEI [23].....	29
OBR. 2.6	OVLADAČ	30
OBR. 2.7	PROPOJOVACÍ SKŘÍŇKA	30
OBR. 2.8	SW DEFINOVANÉ OCHRANY MOTORŮ PMCONTROL G6X [25]	31
OBR. 2.9	BLOKOVÉ SCHÉMA MIKROKONTROLÉRU [23]	32
OBR. 2.10	MANUÁLNÍ REŽIM	34
OBR. 2.11	AUTOMATICKÝ REŽIM	35

OBR. 2.12	OVĽADACÍ TLAČÍTKA PRO AUTOMATICKÝ REŽIM.....	36
OBR. 2.13	DISPLAY PO STISKU BEZPEČNOSTNÍHO TLAČÍTKA STOP	36
OBR. 2.14	DIAGRAM POHYBU V MANUÁLNÍM REŽIMU	37
OBR. 2.15	DIAGRAM POHYBU V AUTOMATICKÉM REŽIMU	38
OBR. 2.16	ÚHLOPŘÍČNÝ POHYB	39

Seznam tabulek

TAB. 1.1	PŘEHLED SBĚRNIC	3
TAB. 1.2	ROZSAH SBĚRNICE CAN PŘI RŮZNÝCH BITOVÝCH RYCHLOSTECH	8
TAB. 1.3	SLOVNÍK OBJEKTŮ CANOPEN[10].....	12
TAB. 1.4	ROZDĚLENÍ IDENTIFIKÁTORŮ CAN V CANOPEN [21].....	13
TAB. 1.5	SROVNÁNÍ BLDC A KARTÁČOVÝCH DC MOTORŮ [19]	23
TAB. 1.6	SROVNÁNÍ PMSM A BLDC MOTORŮ [3]	24

Úvod

Tato diplomové práce vznikla ve spolupráci s firmou PMControl s.r.o. a zabývá se synchronním řízením servomotorů po sběrnici CAN.

Text je dělen do dvou hlavních částí. V první kapitole jsou popsána řešení servopohonů. Jsou zde rozebrány některé v průmyslu běžné sběrnice, používané servomotory s důrazem na jejich řízení. V závěru první kapitoly věnované teoretickým podkladům poznatkům pro práci, jsou zmíněny principiálně některé možné aplikace využití servomotorů. Hlavní důraz je zde věnován, v druhé kapitole použitým prostředkům, EC motorům a sběrnici CAN.

Druhá kapitola je věnována popisu realizovaného zařízení. Cílem projektu nebyla tvorba aplikace. V zadání diplomové práce je zmíněno několik možných aplikací servomotorů. Po domluvě s konzultantem ve firmě PMControl s.r.o. byla vybrána pro diplomovou práci aplikace tříosé frézy, která je řešena jako praktická realizace synchronního řízení servomotorů.

1 Teoretická část

Tato kapitola obsahuje popis technických řešení servopohonů. Jsou zde zmíněné různé, v průmyslu užívané komunikační sběrnice, jejich stručný popis s důrazem na jejich využití v řídicích systémech. V kapitole Servomotory 1.2 jsou zmíněny různé motory používané k tomuto účelu. Ve třetí kapitole této Teoretické části práce jsou nastíněny možnosti využití těchto systémů.

1.1 Sběrnice

Sběrnici rozumíme přenosový kanál mezi dvěma nebo více body tvořený několika vodiči, který se využívá k vzájemné komunikaci a přenosu dat mezi jednotlivými zařízeními, čidly, či řídicím obvodem a jeho periferiemi.

Sběrnice je možné dělit z několika pohledů [7][9]:

- podle způsobu přenosu
 - sériová sběrnice
 - paralelní sběrnice
- podle umístění sběrnice
 - interní (propojení jednotlivých periferií s řídicí jednotkou v zařízení)
 - externí
- podle řízení toku dat
 - master - slave (centralizované)
 - multimaster (decentralizované)

Tato práce se zabývá pouze některými v průmyslu využívanými systémovými sběrnici. Každá sběrnice má své výhody i nevýhody, proto budou ve stručnosti tyto frekventovanější sběrnice popsány. V tabulce (Tab. 1.1) je uveden přehled zde popisovaných sběrnic spolu s některými parametry, tyto sběrnice jsou poté podrobněji popsány.

Sběrnice vycházejí ze sedmivrstvého modelu OSI definovaný organizací ISO, který obecně popisuje komunikační systémy.

Model OSI
Aplikační vrstva
Prezentační vrstva
Relační vrstva
Transportní vrstva
Síťová vrstva
Linková vrstva
Fyzický vrstva

Obr. 1.1 Vrstvový model OSI [1]

Každý komunikační systém nemusí nutně obsahovat všechny vrstvy tohoto modelu, ale běžně se charakter jednotlivých vrstev využívají k popisu komunikačních systémů. Celý model byl vytvořen pro komunikaci v rozlehlých sítích WAN, proto se při využití v lokálních sítích některé vrstvy redukuje a slučují.

- Fyzická vrstva
 - umožňuje přenos bitů po médiu bez ohledu na jejich význam
 - definuje přenosové médium, napěťové úrovně, rychlost přenosu, apod.
- Linková vrstva
 - zajišťuje přenos dat mezi sousedními stanicemi
 - vytváří rámce, které obsahují i údaje o adresování a zabezpečení dat proti chybám
- Síťová vrstva
 - zajišťuje adresování dat, cestu paketů sítí
- Transportní vrstva
 - rozděluje přenášená data do paketů, stará se o přenesení celé zprávy složené z více paketů
- Relační vrstva
 - řídí komunikaci mezi systémy na úrovni relací, organizuje a synchronizuje dialog mezi relačními vrstvami
- Prezentační vrstva
 - převádí formát dat do podoby, ve které se mohou přenášet, zajišťuje komprimaci, kryptografii, apod.
- Aplikační vrstva
 - na této úrovni využívají přenášené informace pro samotné aplikace

Tab. 1.1 Přehled sběrnic

sběrnice	max. rychlost	při délce vodiče	max. vzdálenost	při rychlosti	typ	poznámky
RS - 232	115,2 kb/s	15 m	900 m	2,4 kBd/s	master - slave	60. léta; s moderními čipy může být rychlost až 0,9 Mb/s
RS - 422	10 Mb/s	12 m	1200 m	100 kb/s	master - slave	
RS - 485	12 Mb/s	100 m	1200 m	100 kb/s	master - slave	
CAN	1 Mb/s	30 m	5 Km	10 kb/s	multimaster	koncem 80.let (Bosh)
Ethernet	podle použitého média fyzické vrstvy				multimaster	70. léta (Xerox)
GPIB	7 MB/s		4 / 20 m *	1 MB/s	master - slave	1965 (Hewlett-Packard); paralelní přenos dat

* 4 m - délka 1 segmentu sítě, 20 m maximální vzdálenost krajních zařízení

1.1.1 Sériové porty

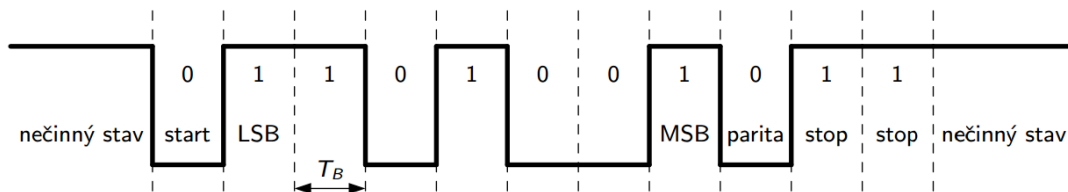
1.1.1.1 RS - 232

Sériová linka RS - 232 se původně ve velké míře používala pro komunikaci počítačů s periferiemi, jako například tiskárny, joysticky, klávesnice, myši apod. Postupem času se k těmto účelům začalo používat rozhraní USB (Universal Serial Bus). Tím se význam RS - 232 ve spotřební elektronice výrazně snížil.

Své místo má nadále v průmyslu, kde se jako sériové asynchronní rozhraní RS - 232 používá, na příklad pro připojení měřících přístrojů, nastavování a servisní zásahy v PLC měničích servomotorů, atd. Standard definuje pouze fyzickou vrstvu, logická úroveň 1 je v tomto standardu indikována záporným napětím menším, než -3 V a logická úroveň 0 pak kladným napětím větším, než 3 V. Jedná se o sběrnici typu master – slave, tedy komunikaci nadřazeného prvku (např. počítače) s podřízeným.

Použití je velmi snadné, při konfiguraci přenosu se zabýváme pouze několika parametry[11].

- rychlost přenosu
 - je udávána v Baudech (Bd) – počet změn úrovně signálu za sekundu (u RS - 232 odpovídá přenosové rychlosti bit/s)
 - maximální rychlost je závislá na délce vedení při vedení délky 15 m je možná rychlost až 19,2 kBd, pokud ale bude vedení na kratší délka vedení je možné se dostat s moderními integrovanými obvody i na 960 kBd (115,2 kBd), v druhém extrému při délce vedení 900 m je maximální rychlost 2,4 kBd
 - nejnižší rychlost přenosu dle normy je 110 Bd
- paritní bit
 - je-li použito zabezpečení paritou, pak je možné použít sudou paritu, nebo lichou paritu
 - parita nemusí být vypočítávána, pak je tento bit možné nastavit pevně na hodnotu jedna, či nula - to se dá využít jako další 9. bit datového přenosu používaný například k rozlišení přenášených dat mezi adresou a daty
- 7bitový / 8bitový datový formát
 - dnes se běžně využívá pouze 8bitový formát dat, ale standardizován je i formát s pouze sedmi datovými bity
 - pořadí bitů je postupně od LSB (nejméně významný bit) po MSB (nejvýznamnější bit)
- počet STOP BITŮ
 - běžně se používá jeden, ale dle standardu je možné použít i stop bit o délce 1,5, nebo 2 bity, což se hodí pro pomalejší zařízení
 - po dobu trvání stop bitu se zpracovává přijatý znak

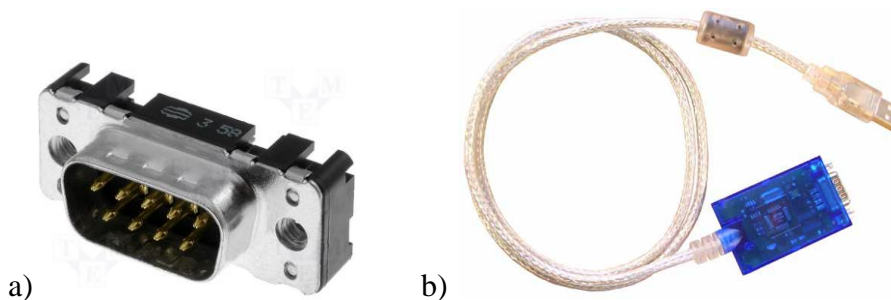


Obr. 1.2 Komunikace sériových portů [11]

Zde na Obr. 1.2:

- LSB - Least significant bit - nejméně významný bit
- MSB - Most significant bit - nejvýznamnější bit
- T_B - čas na jeden bit (určuje rychlost přenosu)

Pro připojení k počítači se využívá devítipinový D-sub konektor. Protože ale přestává být běžný u spotřební elektroniky (např. u notebooků se běžně nevyskytuje), používá se převodník USB/RS-232.

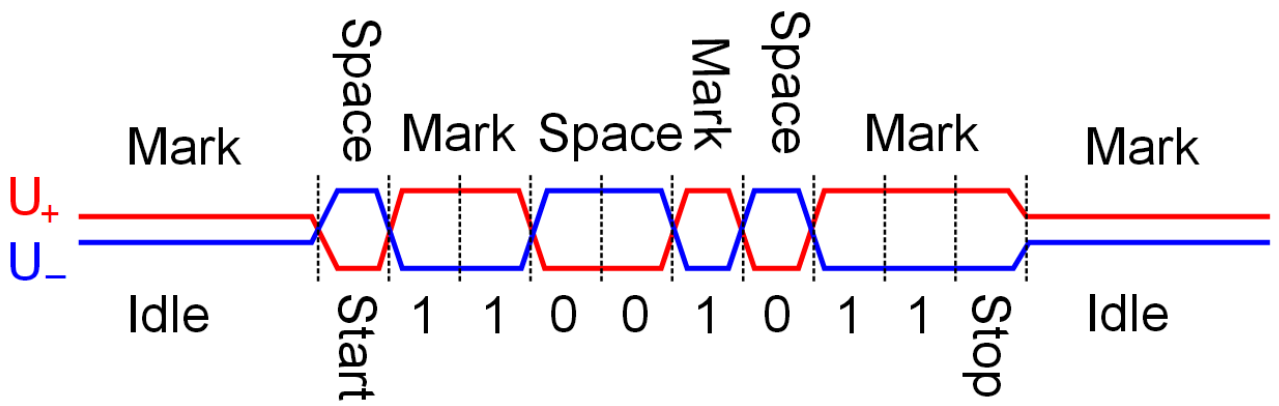


Obr. 1.3 a) 9pinový D-sub konektor b) převodník USB na sériový port [6]

Pro přenos na větší vzdálenosti a rychlosti je vhodnější než RS - 232 použít sběrnice RS - 422 a RS - 485. RS - 232 se proto většinou používá pro nastavování a servis zařízení, naproti tomu sběrnice RS - 422 a RS - 485 se používají jako provozní sběrnice.

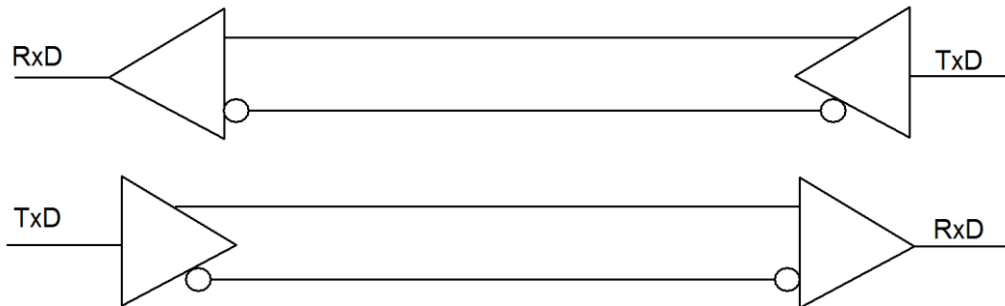
1.1.1.2 RS - 422 a RS - 485

Od sériové linky RS - 232 liší především tím, že logické úrovně jsou přenášeny pomocí diferenciálního napětí mezi dvěma vodiči, ne napětím na jednom vodiči proti zemi. Pro přenos se u těchto sběrnic užívá kroucený pár vodičů. Vodiče jsou označovány jako A (někdy také "-") a B (označované jako "+"). Pokud je rozdíl napětí $A - B < - 200 \text{ mV}$, je přenášena logická úroveň 1, která se také označuje jako *Mark*. Když je přenášena logická úroveň 0, označovaná také jako *Space*, je diferenciál napětí $A - B > + 200 \text{ mV}$. Díky tomu nemá na komunikaci takový vliv rušení, protože oba vodiče podléhají rušení stejnými vlivy. Odolnosti vůči rušení také napomáhá používaná kroucená dvoulinka. Nemění se proto jejich napětí vůči sobě navzájem. Při přenosu dat na větší vzdálenosti musí být vedení na obou koncích ukončeno zakončovacím odporem o velikosti 110Ω . Pořadí datových bitů je stejné jako při komunikaci RS - 232, první se přenáší nejméně významný bit (LSB), na konec nejvýznamnější bit (MSB).



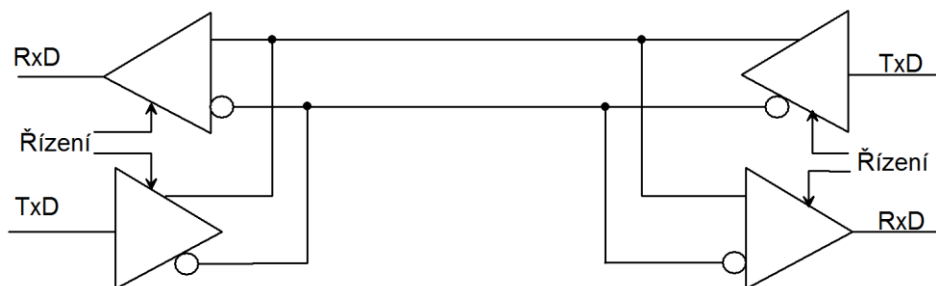
Obr. 1.4 Příklad komunikace s přenosem informací diferenciálním napětím [12]

- RS - 422
 - Pro komunikaci používá dva páry vodičů - jeden pro směr Tx, druhý pro Rx, zachovává možnost duplexního provozu.



Obr. 1.5 Zapojení RS - 422 [6]

- RS - 485
 - Komunikace probíhá pro oba směry po stejném páru vodičů, je tedy nutné zajistit střídání změny směru komunikace.



Obr. 1.6 Zapojení RS - 485 [6]

1.1.2 Profibus

V devadesátých letech byla, spoluprací několika německých firem s podporou ministerstva pro výzkum a technologie, vyvinuta sběrnice Process Fieldbus označovaná jako Profibus. Zařízení využívající Profibus jsou testována a posléze certifikována organizací uživatelů Profibus (PNO).

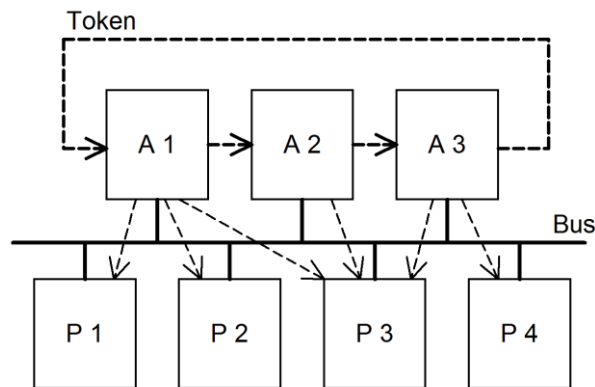
Profibus nabízí výběr ze tří kompatibilních variant. Jedná se o *PROFIBUS - FMS (Field Message Specification)*, univerzální řešení komunikačních úloh, *PROFIBUS - DP (Decentralized Periphery)*, variantu optimalizovanou pro rychlý přenos dat, vhodnou tedy například pro komunikaci mezi jednotlivými automatizačními systémy, a variantu *PROFIBUS - PA (Process Automation)*, vyvinutou pro řídicí a měřicí úlohy ve výbušném prostředí.

Fyzická vrstva sběrnice Profibus má tři varianty provedení:

- přenos po metalickém kabelu podle standardu RS - 485
- přenos po optickém vlákně
- přenos v nebezpečném prostředí
 - v nebezpečném prostředí se komunikace provádí po vedení napájení [7]

Linková vrstva - vrstva přístupu k síti, je u sběrnice Profibus rozdělena do dvou pohledů podle typu zařízení. Zařízení ve sběrnici Profibus se dělí na aktivní a pasivní.

Komunikuje-li aktivní zařízení s pasivním, probíhá komunikace na principu master-slave. Aktivní zařízení na sběrnici Profibus, ale nemůže začít komunikovat kdykoli, nýbrž pouze, je-li k tomu pověřeno. Pověření (token) se neustále dokola předává mezi aktivními zařízeními, aktivní zařízení tedy může komunikovat, pouze vlastní li token. Pasivní stanice komunikují jen ve chvíli jsou-li k tomu vyzvány některou z aktivních stanic. Aktivní zařízení může komunikovat s jakýmkoli pasivním, které je v síti připojeno.



Obr. 1.7 Komunikace v Profibus [9]

Z důvodů optimalizace pro rychlý přenos dat i pro přenášená data o velkém objemu definuje Profibus 4 komunikační služby:

- SDN - Send Data with No Acknowledge
- SDA - Send Data with Acknowledge
- SRD - Send and Request Data
- CSRD - Cyclic Send and Request Data

1.1.3 Sběrnice CAN (Controler Area Network)

Firma Bosch koncem 80.let vyvinula sběrnici CAN, která byla původně určena pro automobilový průmysl k propojení nejrůznějších senzorů. Záměrem byla především úspora materiálu na mnoha spojích napříč celým automobilem. Díky nízké ceně, vysoké spolehlivosti, velké přenosové rychlosti, snadnému nasazení a dalším výhodám si tato sběrnice našla cestu i do mnoha jiných aplikací. Přední výrobci integrovaných obvodů, pro tyto přednosti, začali podporu protokolu CAN implementovat do svých obvodů. Sběrnice CAN je definována normou ISO 11898, podle které je definovaná fyzická vrstva a specifikace CAN 2.0A a později vytvořená specifikace CAN 2.0B, která zavádí standardní a rozšířený formát zprávy, jenž se liší v délce identifikátoru zprávy.

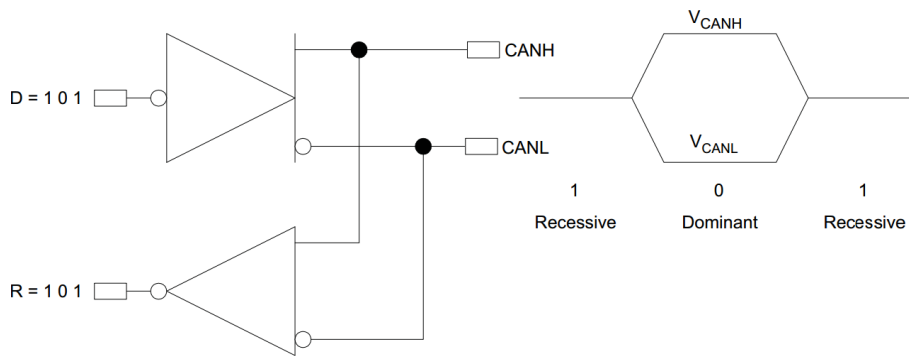
Tab. 1.2 Rozsah sběrnice CAN při různých bitových rychlostech

Bitová rychlost	Maximální délka sběrnice
1 Mbit/s	30 m
800 kbit/s	50 m
500 kbit/s	100 m
250 kbit/s	250 m
125 kbit/s	500 m
62,5 kbit/s	1 km
20 kbit/s	2,5 km
10 kbit/s	5 km

CAN je sériový komunikační protokol, který je díky svému návrhu odolný proti chybám přenosu. Každý uzel připojený na sběrnici může být master a řídit tedy chování ostatních uzlů, mluvíme tedy o multi-master sběrnici, která nemá žádný nadřazený uzel. Pokud je na sběrnici klid, každý připojený uzel může začít vysílat.

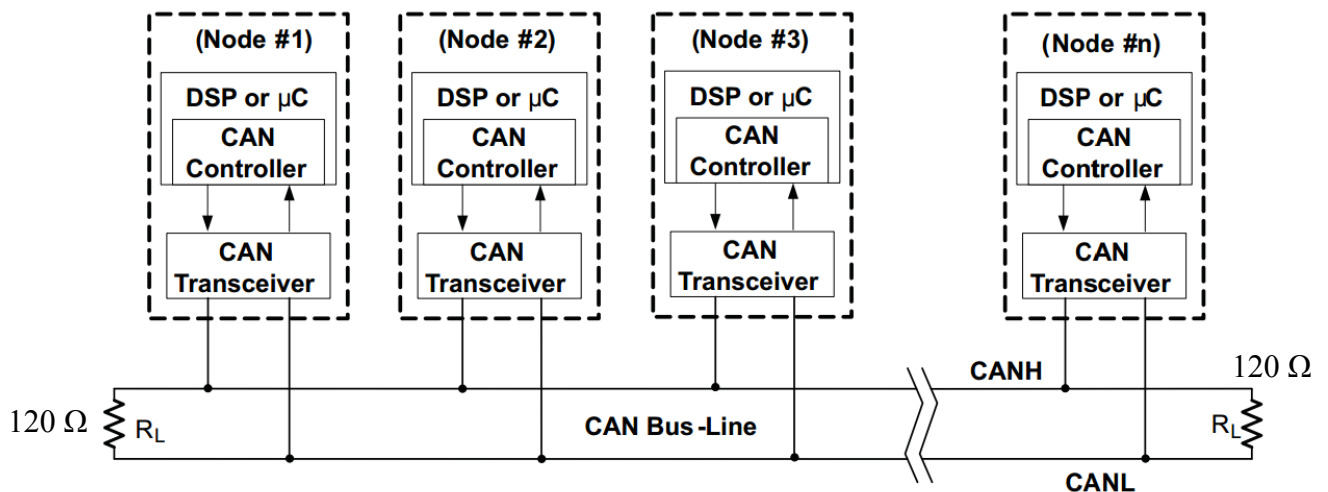
Předností sběrnice CAN je návrh její fyzické vrstvy. Základním požadavkem na ni je, aby realizovala funkci logického součtu. Standard definuje dva stavy sběrnice *dominant* a *recessive*, které jsou interpretovány jako logické úrovně 0 a 1.

Fyzickou vrstvu tvoří dva vodiče označované CAN L a CAN H a úrovně *dominant* a *recessive* jsou definovány rozdílným napětím mezi těmito vodiči. Konkrétně *recessive* jako rozdílové napětí $U_{\text{diff}} \leq 0,5 \text{ V}$ a *dominant* $U_{\text{diff}} \geq 0,9 \text{ V}$. Zde je zásadní rozdíl mezi sběrnici CAN a RS-485, kde jsou obě logické úrovně definovány jako rozdíl napětí (v jednom případě kladný, v druhém záporný). Sběrnice CAN chápe pasivní stav, kdy je stejné napětí na obou vodičích jako úroveň *recessive*, která představuje logickou 1.[22]



Obr. 1.8 Komunikace v CAN [13]

Aby nedocházelo k odrazům, je vedení na obou koncích přizpůsobeno zakončovacím odporem 120Ω . Teoreticky je možné připojit libovolný počet segmentů do sběrnice, při větším počtu uzlů by však došlo k velkému zatížení sběrnice, proto se maximální počet uzlů omezuje na 64 na segment.[4]



Obr. 1.9 Uspořádání sítě CAN [13]

Linková vrstva protokolu se dělí na dvě podvrstvy [4]:

- MAC (Medium Access Control)
 - kóduje data
 - vkládá doplňkové bity
 - řídí přístup uzlů k médiu – rozlišení priorit zpráv
 - detekce chyb, jejich hlášení
 - potvrzování správně přijatých zpráv
- LLC (Logical Link Control)
 - řízení datového spoje
 - filtrování přijatých zpráv
 - hlášení o přetížení

S O F	11-bit Identifier	R T R	I D E	r0	DLC	0...8 Bytes Data	CRC	ACK	E O F	I F S
----------------------	------------------------------	----------------------	----------------------	-----------	------------	-------------------------	------------	------------	----------------------	----------------------

Obr. 1.10 Formát standardní zprávy CAN [13]

Formát standardní zprávy vysílané po sběrnici CAN[13]:

- SOF (start of frame) - 1 bit
 - začátek zprávy
- identifikátor zprávy - 11 bitů
 - určuje prioritu zprávy
- RTR (remote request) - 1 bit
 - rozlišuje zda se jedná o datovou zprávu, nebo žádost o data
- IDE (identifier extended) - 1 bit
 - rozlišuje mezi standardním (*dominant*) a rozšířeným (*recessive*) rámcem
 - rozšířený formát zprávy obsahuje 29 bitový identifikátor, který je rozdělen do dvou částí, první (11 bitů) stejná jako u standardního formátu a druhá část (18 bitů) následuje za identifikátorem IDE
- r0 (rezervovaný bit) - 1 bit
- DLC (data length code) - 4 bity
 - označuje, jak dlouhý je blok dat
- přenášená data - 0-8 Byte
- CRC (cyclic redundancy check) - 16 bitů
 - výsledek zabezpečení cyklickým kódem (15 bitů + 1 bit oddělovač)
- ACK (acknowledges) - 2 bity
 - potvrzení přijetí zprávy (1 bit + 1 bit oddělovač)
- EOF (end of frame) - 7 bitů
 - ukončení zprávy (7 bitů ve stavu *recessive*)
- IFS (interframe space) - 7 bitů
 - mezera mezi zprávami

Tím, že je CAN multimaster sběrnice, může vysílat jakýkoli uzel ve chvíli kdy je připraven. Zpráva neobsahuje žádnou informaci o tom, komu je určena, každou tak přijímají všechny uzly, všechny ověří správnost přijetí. Identifikátor zprávy, kterým každá zpráva začíná, uvádí její prioritu a v případě kdy více zařízení začne vysílat současně, je doručena zpráva s vyšší prioritou.

Každý vysílající uzel hlídá, zda úroveň na sběrnici odpovídá jím vysílané úrovni. Pokud je detekovaná kolize (vysílaná úroveň *recessive*, detekovaná *dominant* – opačně to není

možné detekovat), okamžitě přestává vysílat. Protože je dominantním stavem na sběrnici logická úroveň 0, má vyšší prioritu identifikátor zprávy s nižší hodnotou. Nejvyšší prioritu mají obvykle chybové zprávy, které se tak mohou vysílat kdykoli. Jakmile některé zařízení vysílá ať už jakoukoli zprávu, ostatní podle této synchronizují své vnitřní hodiny.

Zabezpečení komunikace je u sběrnice CAN zajištěno hned několika mechanismy, které se vzájemně doplňují:

- monitoring
 - vysílač sám hlídá, zda je na sběrnici jím vysílaná úroveň signálu
- CRC kód
 - Cyklický kód tvoří poslední pole vysílané zprávy a může tedy být generována ze všech do té doby odvysílaných bitů zprávy. Chybu CRC může detekovat kterýkoli uzel.
- vkládání bitu (bit stuffing)
 - Pokud se ve zprávě vyskytuje pět po sobě jdoucích bitů jedné úrovně, vkládá se do zprávy navíc jeden bit opačné úrovně. Opatření napomáhá správnému časování přijímajících uzlů.
- bitová synchronizace (Bit timing)
 - Při přenosu dat se všechny zařízení při každém bitu synchronizují s vysílacím uzlem
- kontrola zprávy (message frame check)
 - Zprávy se porovnávají s formátem zpráv uvedených ve specifikaci, pokud se na některém z bitů objeví nepovolená hodnota je vygenerováno hlášení chyby rámce.
- potvrzení přijaté zprávy (acknowledge)
 - Každé zařízení, které je připojeno ke sběrnici musí potvrdit správné přijetí zprávy. To platí i pro zařízení, která mají zapnuté filtrování a zprávu tedy nezpracovává.

Sběrnice CAN má ale jeden nedostatek. Posílá-li jedno zařízení zprávu pro několik dalších zařízení, všichni příjemci potvrzují správnost přijetí zprávy najednou změnou jednoho bitu v potvrzovací zprávě. Z toho ale není možné detekovat, zda zprávu skutečně v pořádku přijala všechna zařízení, která měla. Kdyby tuto zprávu přijalo v pořádku jen jediné zařízení, vysílač dostane stejnou odpověď, jako kdyby proběhl v správně přenos všem zařízením.

1.1.3.1 CANopen

Aplikační vrstva CANopen je postavena na sběrnici CAN. Podporuje přímý přístup k parametrům zařízení a přenos časově kritických procesních dat. Každý komunikační objekt má přiřazen jeden, nebo více identifikátorů, které běžně definují jejich prioritu na sběrnici. Vhodné přiřazení identifikátorů je při návrhu komunikačního systému CANopen jeden ze zásadních kroků, ale pro usnadnění návrhu jsou v protokolu definovány výchozí hodnoty identifikátorů pro všechny povinné objekty. Všechny komunikační objekty jsou zařazeny v tzv. slovníku objektů, který je uložen v zařízení, které je součástí sítě. Tento slovník funguje jako rozhraní mezi samotným zařízením a aplikačním programem. [10]

Rozlišují se dva základní mechanismy přenášení zpráv v CANopen. Procesní data určená pro časově kritickou výměnu, nazývaná Process Data Objects (PDO) a služební zprávy, jejichž doručení není časově kritické, nazývané Service Data Objects (SDO).

Zpráva na sběrnici CAN může obsahovat maximálně 8 Bytů dat. V případě, že není možné data přenést v jedné zprávě, musí být zpráva rozsegmentována do několika zpráv. Maximální délka dat není omezena. SDO se používá především pro nastavení zařízení a pro přenos delších zpráv.

Procesní data se přenášejí buď cyklicky při změně, nebo na vyžádání jako broadcastové zprávy. Rozmístění aplikačních objektů (položek z slovníku objektu) do přenášeného objektu PDO je určen pomocí tzv. mapování PDO - tato informace je uložena v adresáři objektu a podle požadavků aplikace může být změněna. [21]

Organizace CiA (CAN in Automation), která sdružuje výrobce a uživatele a stará se o standardizaci tohoto protokolu vydala ke sběrnici CANopen normu CiA 306, kde definuje tzv. *Electronic Data Sheet* (EDS). Zde je definován formát zpráv. Bez dodržení EDS není možné úspěšně absolvovat testy shody CiA CANopen. [14]

Tab. 1.3 Slovník objektů CANopen[10]

CANopen Object Dictionary	
Index	Object
0000	<i>nepoužívá se</i>
0001 - 001F	Static Data Types
0020 - 003F	Complex Data Types
0040 - 005F	Manufacturer Specific Complex Data Types
0060 - 007F	Device Profile Specific Static Data Types
0080 - 009F	Device Profile Specific Complex Data Types
00A0 - 0FFF	<i>rezervováno</i>
1000 - 1FFF	Communication Profile Area
2000 - 5FFF	Manufacturer Specific Profile Area
6000 - 9FFF	Standardised Device Profile Area
A000 - FFFF	<i>rezervováno</i>

Schéma přiřazení identifikátorů má funkční část, určující prioritu objektu, a část označenou jako Node-ID, která umožňuje rozlišovat mezi více zařízeními plnícími stejnou funkci. Nastavení Node-ID je obvykle provedeno HW přepínačem. Rozsah Node-ID je 1 až 127.

CANopen využívá standardní formát zpráv s 11-bitovým identifikátorem. Identifikátory jsou rozděleny do několika skupin podle následující tabulky:

Tab. 1.4 Rozdělení identifikátorů CAN v CANopen [21]

CAN Identifikátor	Hex	Kód	Účel
0	0	0000b	NMT Services
1-127	001-07F	-	free
128	80	0001b	Sync Message
129-255	081-0FF	0001b	Emergency Messages
256	100	0010b	Time-Stamp Message
257-384	101-180	-	free
385-511	181-1FF	0011b	Transmit Process Data Objects 1 (TPDO1)
512	200	-	free
513-639	201-27F	0100b	Receive Process data Objects 1 (RPDO1)
640	280	-	free
641-767	281-2FF	0101b	Transmit Process Data Objects 2 (TPDO2)
768	300	-	free
769 - 895	301-37F	0110b	Receive Process data Objects 2 (RPDO2)
896	380	-	free
897-1023	381-3FF	0111b	Transmit Process Data Objects 3 (TPDO3)
1024	400	-	free
1025-1151	401-47F	1000b	Receive Process data Objects 3 (RPDO3)
1152	480	-	free
1153-1279	481-4FF	1001b	Transmit Process Data Objects 4 (TPDO4)
1280	500	-	free
1281-1407	501-57F	1010b	Receive Process data Objects 4 (RPDO4)
1408	580	-	free
1409-1535	581-5FF	1011b	Service Data Objects
1536	600	-	free
1537-1663	601-67F	1100b	Service Data Objects
1664-1792	680-700	-	free
1793-1919	701-77F	1110b	Node Guarding (Boot-Up) Messages
1920-2015	780-7DF	-	free
2016-2031	7E0-7EF	-	Reserved for NMT, LMT and DBT Services

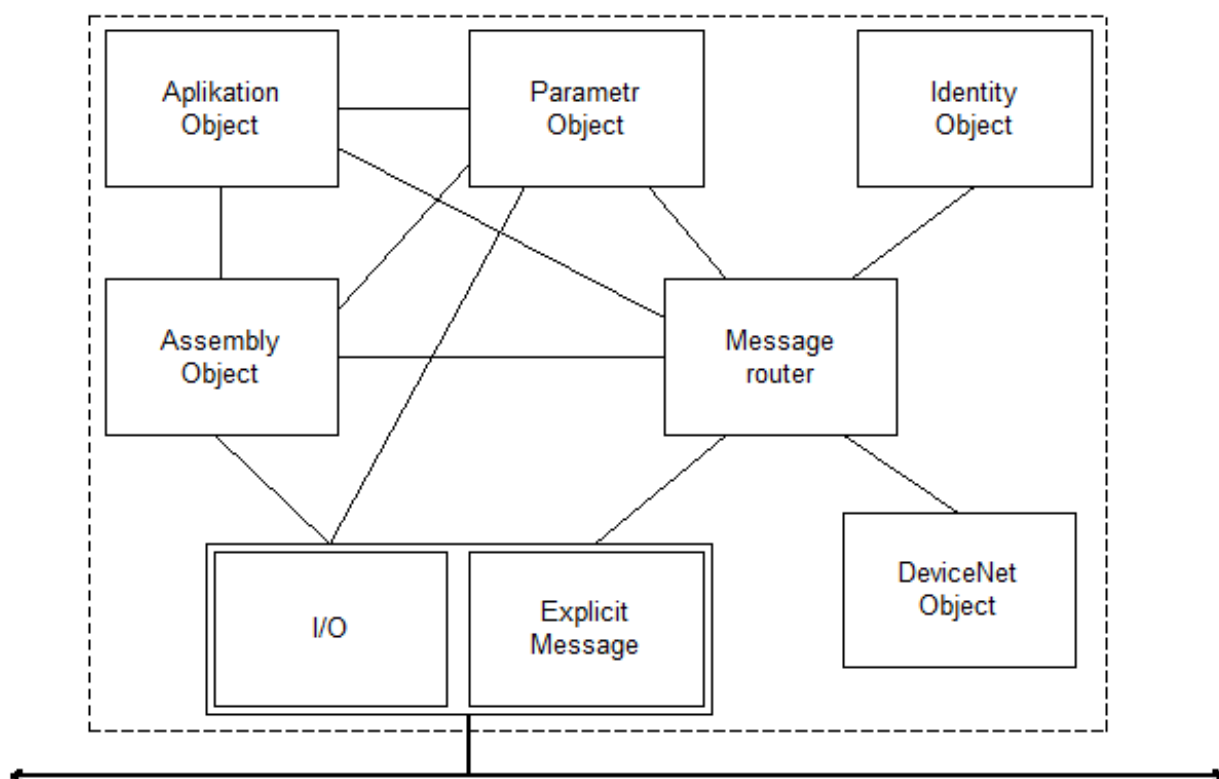
1.1.3.2 DeviceNet

Sběrnice DeviceNet je další z možností řešení aplikační vrstvy komunikačního systému postaveného na fyzické a linkové vrstvě sběrnice CAN.

Základní vlastnosti sběrnice [9]:

- podpora až 64 účastníků sítě
- připojení/odpojení bez přerušení funkce sítě
- přenosová rychlost 125/250/500 kBd
- společný rozvod datových signálů a napájení pro účastníky
- fragmentace dlouhých bloků dat
- detekce duplicitních adres zařízení
- zaměnitelnost zařízení od různých výrobců (standardizovány typy konektorů, diagnostika i profily různých typů zařízení)

Sběrnice DeviceNet funguje na principu objektového modelu přenosu. Každé zařízení, které je připojeno ke sběrnici je rozděleno na dílčí části - *objekty*. Celé zařízení je pak popsáno jako množina těchto *objektů*. Dále je definovaná tzv. *třída*, ta spojuje *objekty* stejných, nebo podobných vlastností. Pro každou *třídu* je definovaná množina atributů, či množina služeb, které popisují chování jednotlivých *objektů*. Posledním výrazem zaváděným v této souvislosti je *instance*. *Instance* popisuje konkrétní realizaci objektu určité třídy. Při adresaci všechny tyto čtyři složky musíme zohlednit. Proto se jako adresa přenáší adresa samotného zařízení, identifikátor třídy, číslo instance a číslo atributu, nebo služby.[1]



Obr. 1.11 Objektový model zařízení v síti DeviceNet[1]

Typické zařízení obsahuje objekty[1]:

- Identity object
 - povinný objekt
 - obsahuje identifikaci výrobce, typ zařízení, číslo série, atd.
- Message router object
 - povinný objekt
 - zajišťuje předávání zpráv mezi objekty
- DeviceNet object
 - povinný objekt
 - obsahuje adresu zařízení, údaj o přenosové rychlosti, atd.
- Assembly object
 - nepovinný objekt - nemusí být realizován
 - jedna, nebo více instancí, která má za úkol spojovat různé atributy různých objektů, které spojuje do jednoho atributu, přenášeného najednou po komunikační lince
- Connection object
 - povinný objekt
 - nejméně dvě instance - reprezentuje vždy jeden koncový bod virtuálního komunikačního kanálu, mezi dvěma účastníky sítě
- Parametr object
 - nepovinný objekt - nemusí být realizován
 - používá se v zařízeních, kde se dají konfigurovat parametry, pro každý parametr existuje jedna instance, skrz kterou se tento parametr edituje
- Application object
 - nepovinný objekt - nemusí být realizován
 - zajišťuje konkrétní aplikační činnost zařízení

Mezi zásadní výhody protokolu DeviceNet patří možnost napájení zařízení po sběrnici. K napájení je možné využít stejnosměrné napětí 24 V, proud po celé sběrnici je omezen na 8 A, pro jednotlivá zařízení je nejvyšší možný proud 3 A. Další velkou výhodou je také plná otevřenost protokolu. [9]

1.1.4 Ethernet

Ethernet je dnes asi celosvětově nejpoužívanější standard lokálních sítí. Význam standardu Ethernet se dnes v průmyslu zvyšuje, výrazné využití ale nachází i v domácnostech a to díky protokolům IP a IPv6 a síti Internet, která může sdružovat lokální síť Ethernetu do celosvětové sítě. Standard Ethernet vychází ze specifikace IEEE 802.3 pro fyzickou vrstvu a doplňuje ho o linkovou vrstvu.

Fyzická vrstva umožňuje využívat podle původní specifikace tlustý padesátiohmový koaxiální kabel, vývojem se ale do standardu zařadil i tenký koaxiální kabel, kroucená

dvojlinka a optický kabel. Až kroucená dvojlinka a optický kabel umožňují přechod na rychlosti přenosu 100 Mb/s a více, což začalo být perspektivní pro užití sběrnice pro průmyslovou komunikaci. [9]

Rozsah a specifikace Ethernetu jsou výrazně nad rámec této práce.

1.1.5 GPIB

V roce 1965 navrhla firma Hewlett-Packard sběrnici HP-IB (Hewlett-Packard Interface Bus) jako firemní propojovací soustavu pro své měřicí přístroje. Systém s touto sběrnici se stal pro svou rychlost velmi populárním a v roce 1975 byl přijat za standart IEEE 488. V současnosti je jedním ze systémů, které se často používají pro automatizaci a především měřicí techniku. Na rozdíl od ostatních zde zmiňovaných sběrnic, se GPIB nejedná o seriovou sběrnici, používá 24 paralelních vodičů, které jsou sdruženy ve třech dílčích sběrnících: sběrnici dat (8 vodičů), řízením přenosu (3 vodiče), řízení stykového systému (5 vodičů), zbylých 8 vodičů je určeno k uzemnění pro stínění. Propojení zařízení může být lineární, či hvězdicové, délka jednoho kabelu je maximálně 4 m a celý systém nemůže přesáhnout vzdálenost 20 m. Nejvýše je možné připojit 15 zařízení. Rychlost přenosu GPIB sběrnice je až 1 MB/s, je ale vždy závislá na nejpomalejším segmentu sítě. Při kratším vedení se ale můžeme dostat i na vyšší rychlosti (7 MB/s). [7]

1.1.6 FIP

Skupina francouzských, německých a italských firem vyvinula standard používaný především ve Francii, který nese název FIP (Factory Instrumentation Protocol). Pro komunikaci používá kroucenou dvoulinku a umožňuje rychlost přenosu dat 31,25 kbit/s až 2,5 Mbit/s. Standard používá tzv. zdrojově orientovanou polling metodu. Řízení sběrnice zajišťuje tzv. arbitr. Ten cyklicky pověřuje všechny stanice, které by chtěli data přijímat, či vysílat. Arbitr má v uložený seznam proměnných, jenž jsou v připojených stanicích používány a každá proměnná má svoji vlastní 16 bitovou adresu (existuje tedy 65 536 identifikátorů). Ve chvíli, kdy arbitr pošle adresu některé proměnné po sběrnici a některá stanice ji rozpozná, zpět odešle obsah dané proměnné. Tento obsah i předešlou adresu vidí i přijímač, a tak zná proměnnou i její hodnotu. Arbitr postupně žádá všechny používané proměnné neustále dokola, tím vzniká distribuovaná databáze, která je také pravidelně obnovována. Potřebuje-li některé zařízení hodnotu některé proměnné stačí vyčkat až bude arbitrem vyžádána a poté si její hodnotu může zařízení uložit.

1.1.7 ModBus

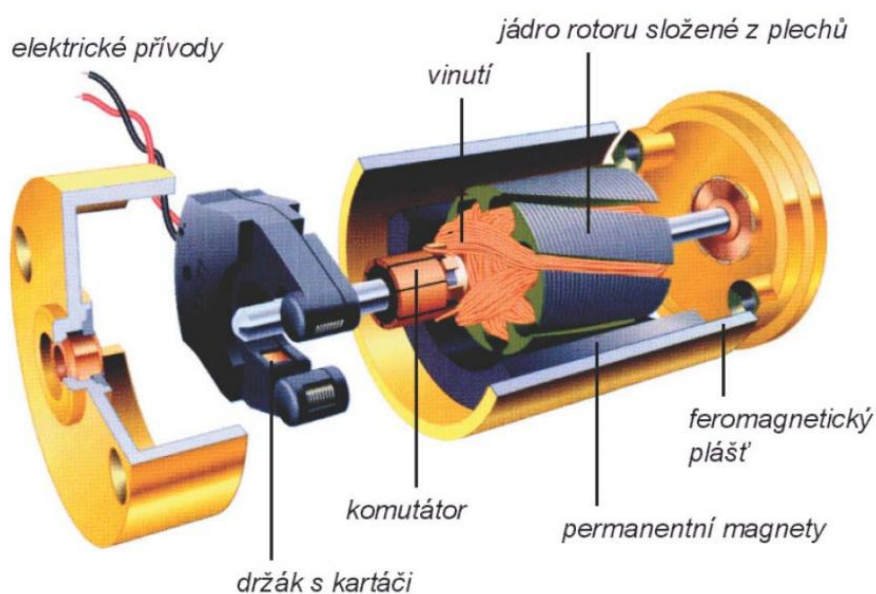
Původně firemní Fieldbus, vyvinutý firmou Modicon. Pro své kvality se rozšířil a z interního firemního fieldbusu se rozvinul v běžně používaný. ModBus definuje pouze zprávy, je tedy nezávislý na použití fyzické vrstvy, kterou běžně zastává některý ze standardů sériové linky (RS232/422/485). Nejnovější specifikace umožňují i využití protokolu TCP/IP. ModBus je zaměřený pouze na přenos dat, nijak nedefinuje formát dat, nebo význam. Je využíván pro svoji jednoduchost a otevřenost, ale značnou nevýhodou je, že od svého vzniku na konci sedmdesátých let neprošel žádnou významnou modernizací.

1.2 Servomotory

Elektrický servopohon je regulační pohon, který se skládá z jednoho, nebo více elektrických motorů a řídicích a regulačních obvodů. Zatím co běžný elektropohon ve většině případů pracuje bez zpětné vazby, servopohon vždy pracuje v uzavřené regulační smyčce se zpětnou vazbou, kontrolující směr otáčení, rychlost otáčení, aktuální polohu, nebo všechny tyto parametry. Hlavním požadavkem na servopohon je jeho možnost regulace, rozumíme tedy především rychlost, přesnost a spolehlivost regulace. [3]

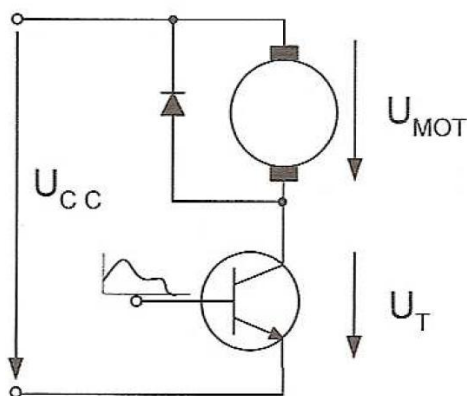
1.2.1 Stejnosměrný motor - kartáčový

Stejnosměrný motor je jeden z nejstarších elektrických strojů. Pro servopohony se používají zejména stejnosměrné motory s buzením permanentními magnety. Stator takového motoru je tvořen permanentními magnety. Rotor s vinutím, je tvořen cívkami, které jsou uloženy v drážkách a vyvedeny na komutátor, který spolu s uhlíkovými kartáči tvoří tzv. sběrací ústrojí. [3]



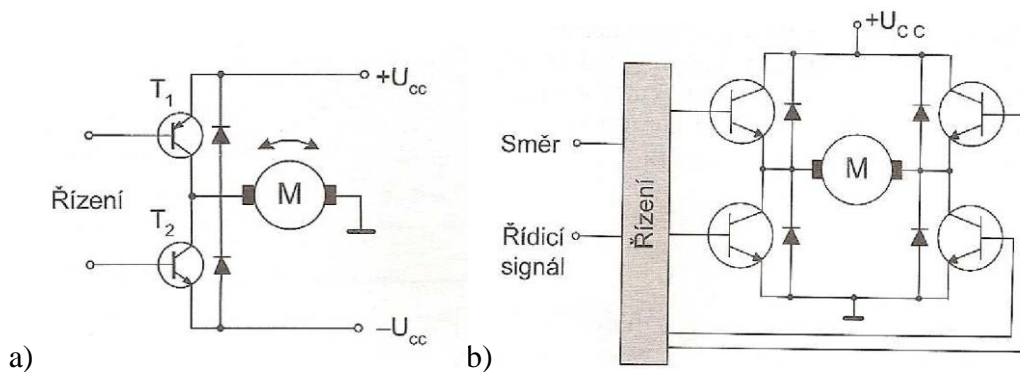
Obr. 1.12 Konstrukce DC motoru s buzením permanentním magnetem [15]

Stejnosměrný motor vhodný pro využití pro servopohon, protože se jeho otáčky dají snadno plynule měnit. Přivedením jmenovitého napětí na napájecí svorky se nezatížený motor roztočí na otáčky naprázdno. Změnou polaritě napětí změním směr otáčení rotoru, změnou napětí pak plynule měníme otáčky motoru. Řízení takového motoru je tedy principiálně velmi snadné.[15]



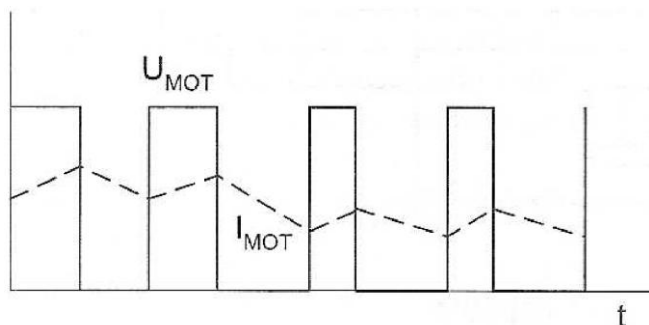
Obr. 1.13 Řízení otáček DC motoru pouze v jednom směru [15]

Dioda v tomto zapojení plní pouze funkci ochrany tranzistoru před záporným napětím generovaným vinutím motoru. K řízení nejen otáček, ale i směru otáčení je však toto principiální zapojení nedostatečné. V takovém případě je potřeba přistoupit k řízení pomocí bipolárního zdroje, nebo v můstkovém zapojení, jak je naznačeno na Obr. 1.14.



Obr. 1.14 Řízení DC motoru a) bipolárním zdrojem b) můstkově [15]

K řízení se také používá pulzně šířková modulace (PWM). Řídicí obvod spíná napětí přiváděné k motoru.



Obr. 1.15 Řízení pomocí PWM [15]

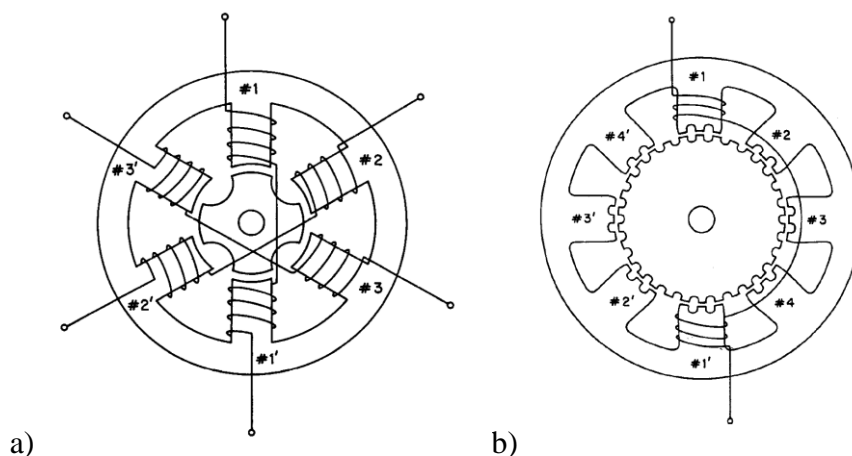
Mezi nevýhody tohoto motoru se dá jednoznačně zařadit komutátor. Je zdrojem rušení a mechanický kontakt v podobě kartáčů podléhá opotřebení a vyžaduje údržbu. Další zde zmíněné motory jsou již řešeny tak, aby komutátor nemusely využívat a této značné nevýhodě se tak vyhnuly.

1.2.2 Krokové motory

Krokový motor představuje nejjednodušší možnost převodu digitálního signálu na polohu. Jedná se o jediný typ motoru, který umožňuje nastavit polohu motoru bez zpětné vazby. Je však nutné zajistit, aby každý krok byl motorem vykonán.[3]

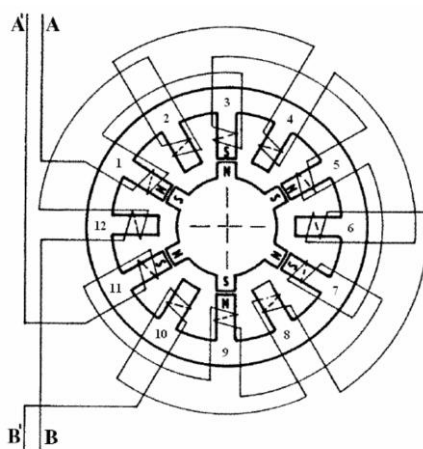
Stator krokového motoru je tvořen několika cívkami (obvykle čtyřmi). Jejich kontakty jsou vyvedeny. Přiváděním stejnosměrného napětí na tyto cívký řídíme pohyb motoru. Podle konstrukčního uspořádání potom dělíme krokové motory [3],[15]:

- krokové motory reluktanční
 - motory s vyjádřenými póly na statoru i rotoru, využívají výrazně rozdílného magnetického odporu (reluktance) v příčné a podélné ose
 - podmínkou je rozdílný počet pólů na statoru a rotoru



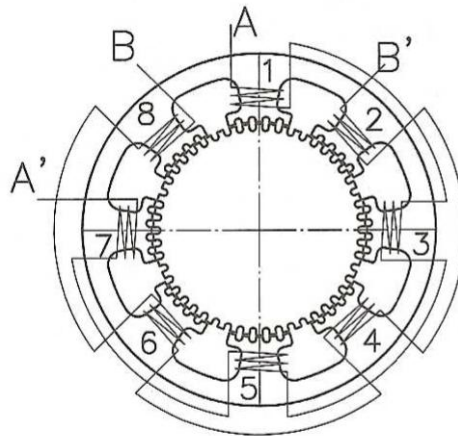
Obr. 1.16 Reluktanční krokový motor[3] a) jednoduchý b) více kroků na otáčku

- krokové motory s permanentním magnetem (s aktivním rotorem)
 - využívají výrazný reluktanční moment i ve stavu, kdy statorové vinutí není napájeno, udržuje tak klidovou polohu rotoru
 - mají různý počet pólů na statoru i rotoru
 - póly rotoru tvoří permanentní magnet



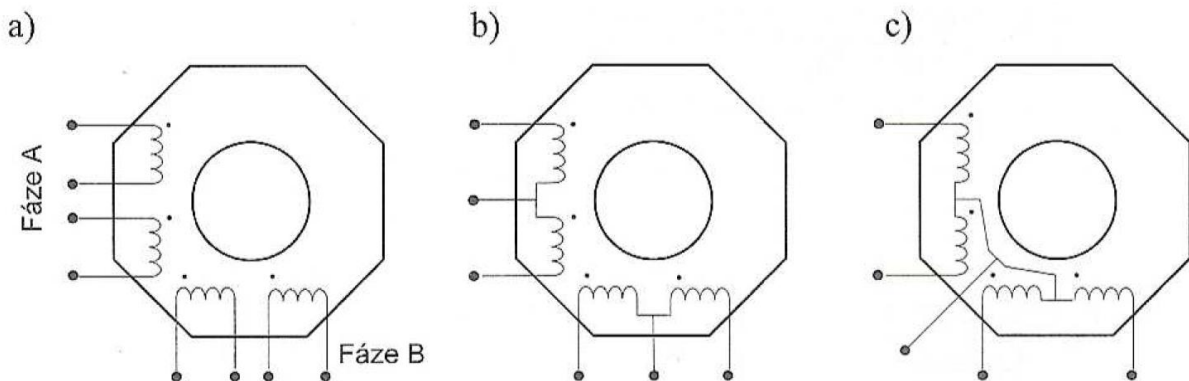
Obr. 1.17 Krokový motor s permanentním magnetem

- hybridní krokové motory
 - kombinují konstrukční principy obou předcházejících typů
 - mají rozdílný počet pólů statoru a rotoru a tím využívají rozdílný magnetický odpor v příčné a podélné ose
 - jako rotor se používá permanentní magnet, pohyb je tedy aretován i při odpojení napájení
 - má malý moment



Obr. 1.18 Hybridní krokový motor [15]

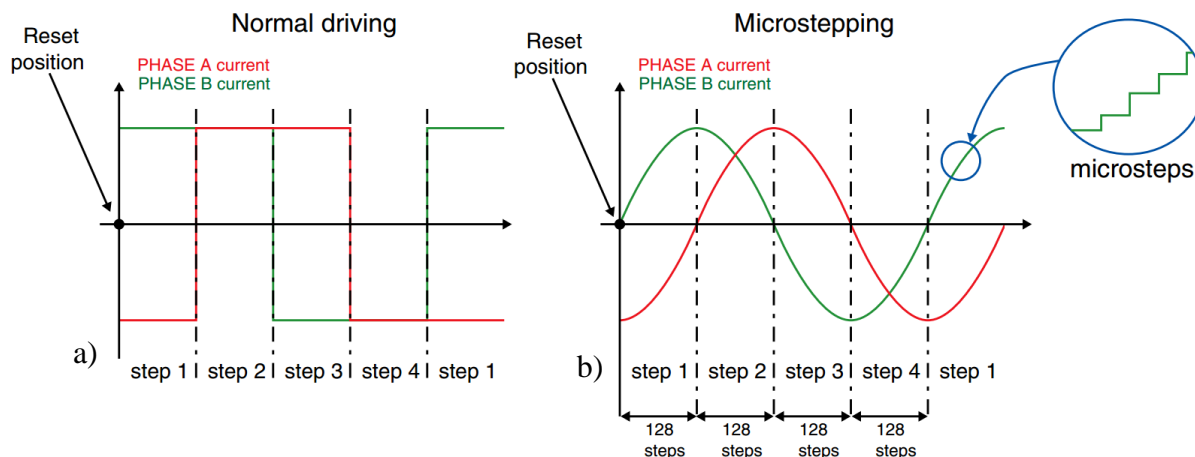
Hybridní motory jsou dnes patrně nejpoužívanějším typem krokového motoru. Můžou se lišit vnitřním zapojením cívek statoru.



Obr. 1.19 Možná zapojení cívek hybridních krokových motorů [15]

Řízení je možné unipolárně nebo bipolárně. Při unipolárním prochází proud v cívkách vinutí pouze jedním směrem a motor se ovládá spínáním cívek k jednomu pólu. Při bipolárním řízení se mění směr proudu v cívkách viz Obr. 1.20 a).

Běžně se dnes setkáváme s motory, které mají 200 kroků na otáčku, což odpovídá otočení o $1,8^\circ$ na jeden krok. Pokud by takové rozlišení nestačilo, je možné pracovat jemněji s buzením cívek statoru a díky tomu přecházet s jemnějším krokem mezi jednotlivými velkými kroky, tzv. mikrokrokování viz Obr. 1.20 b).

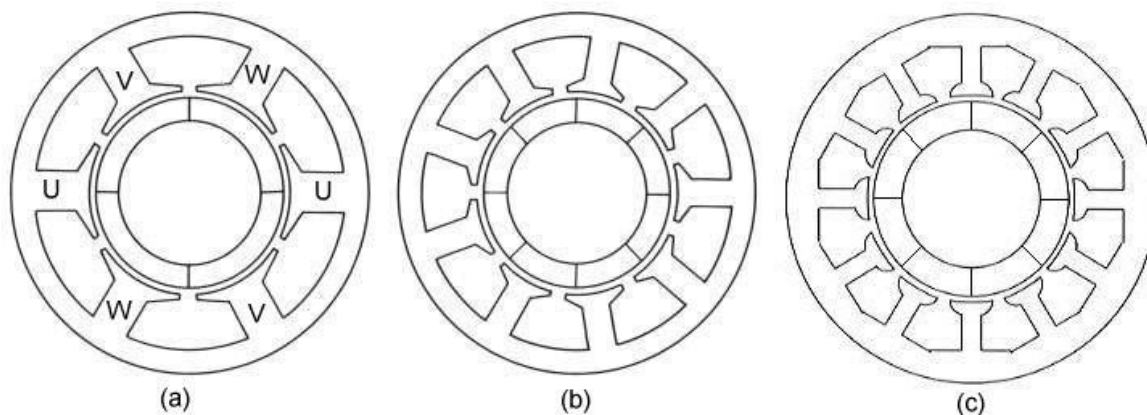


Obr. 1.20 Princip mikrokroku krokových motorů [18]

1.2.3 Elektronicky komutované motory - BLDC

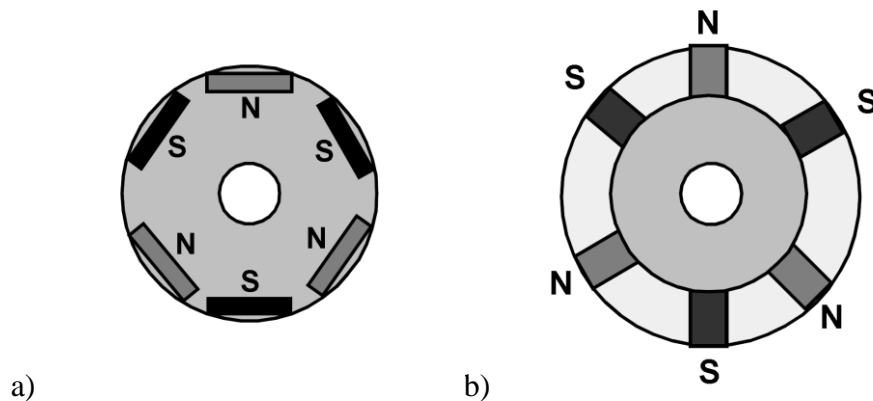
Běžně používaná anglická zkratka BLDC značí Brushless DC motor, tedy bezkartáčový stejnosměrný motor. Jedná se o synchronní motor, který je napájen stejnosměrným proudem a elektronika se stará o komutaci proudu. K pohybu nepotřebuje sinusový průběh proudu, elektronický komutátor proud do jednotlivých cívek většinou jen přepíná. [19]

Stator BLDC motoru se skládá z ocelových plechů s vinutím umístěným v drážkách, které jsou podél vnitřního obvodu. Většina BLDC motorů má tři statorové cívky spojené do hvězdy. Každá z těchto cívek je konstruována s mnoha závity vzájemně propojeny tak, že tvoří vinutí. [17]



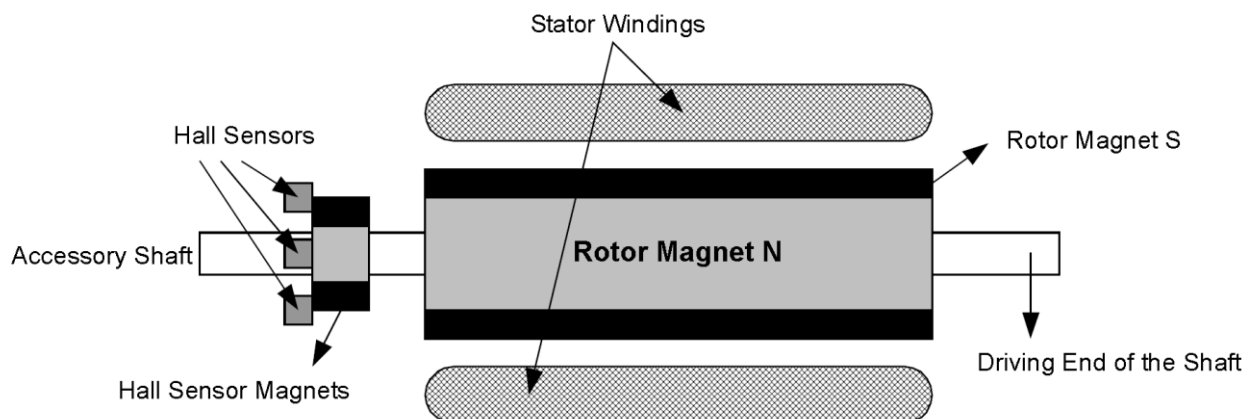
Obr. 1.21 Řez BLDC motorem - různý počet pólů statoru [20]

Rotor je konstrukčně uspořádán buď s magnety na povrchu, nebo s magnety vestavěnými uvnitř rotoru. Rozdílné uložení magnetů má vliv na koncentraci magnetického toku ve vzduchové mezeře. Rotor může mít dva až osm pólových párů.[17]



Obr. 1.22 Rotor BLDC motoru [17] a) s magnety na povrchu b) s magnety uvnitř

Pro správnou komutaci statorového stejnosměrného proudu BLDC motoru, je nezbytně nutné znát polohu rotoru. Součástí motoru je tedy snímač polohy rotoru, který je konstruován pomocí trojice Hallových sond, které jsou umístěny ve statoru. Případně tyto sondy mohou být umístěny samostatně. V tom případě musí být na hnané hřídeli rotoru permanentní magnet, který svým zmagnetováním a uložením odpovídá rotoru. Hallové sondy pak detekují polohu podle tohoto magnetu. Kdykoli kolem každé ze sond projde severní nebo jižní pól magnetu, indikují vysokou, nebo nízkou úroveň. Signály ze všech sond se vyhodnocují a na jejich základě se řídí elektronická komutace - spínání proudů trojicí cívek.



Obr. 1.23 Podélný řez BLDC motorem i s Hallovými sondami [17]

Tab. 1.5 Srovnání BLDC a kartáčových DC motorů [19]

funkce	BLDC motor	Kartáčový DC motor
Komutace	Elektronická komutace	Kartáčová komutace
Údržba	Méně nutná	Vyžadovaná periodicky
Životnost	Dlouhá	Kratší
Rychlost / točivý moment charakteristika	Umožňuje provoz na všech rychlostech	Při vyšších rychlostech jsou problémy s komutací - rychlá změna polaritu proudu ve vinutích rotoru
Setrvačnost rotoru	Nízký, protože má permanentní magnet na rotoru. To snižuje dynamické zpoždění.	Vyšší setrvačnost rotoru, která limituje dynamickou charakteristiku
Rychlostní rozsah	Vyšší	Nižší - mechanické omezení kartáči
Generování rušení	Nízké	Oblouky na kartáčích generují rušení
Náklady	Dražší	Levnější
Řízení	Složitě a nákladné	Jednoduché, levné
Efektivita	Vysoká	Střední

1.2.4 Synchronní motor s permanentními magnety - PMSM

Zkratka PMSM označuje Permanent Magnet Synchronous Motor, tedy synchronní motor s permanentním magnetem. Synchronní označuje, že se rotor otáčí se stejnými se stejnou rychlostí jako se otáčí magnetické pole generované statorem.

Konstrukce PMSM je podobná jako konstrukce motoru BLDC. Stator tvoří póly, na kterých je navinuto třífázové statorové vinutí. Rotor je osazen permanentními magnety. Počet pólů rotoru může být různý. Uložení magnetů je opět dvojí – na povrchu rotoru, nebo uvnitř rotoru. Je-li používán rotor s magnety na povrchu, pak zde můžeme vidět jeden z rozdílů v konstrukci mezi BLDC a PMSM. BLDC motory pro využívání obdélníkového buzení potřebují pólové krytí (poměr obvodu pólových nástavců k celkovému obvodu vzduchové mezery) blízký jedničce. PMSM oproti tomu vyžaduje pólové krytí přibližně dvě třetiny, protože je buzena sinusovým průběhem.

Podobně jako u BLDC je také ke správné funkci motoru potřeba snímat polohu motoru a podle toho generovat budící proud statorem. Rozdíl je především v tom, že u BLDC motoru se pouze přepínalo mezi třemi vinutími motoru tak, že dvě byly napájeny, u PMSM se současně napájí všechny tři fáze sinusovým průběhem proudu.

Tab. 1.6 Srovnání PMSM a BLDC motorů [3]

PMSM	BLDC
napájení 3 fázovým sin. proudem	napájení obdelníkovým proudem, vždy jen 2 fáze
3 regulátory proudu (vektorové řízení)	1 regulátor proudu
spojité měření polohy rotoru	stačí diskretní měření polohy rotoru
bez momentových pulzací	při malých otáčkách momentové pulzace
lepší rovnoměrnost otáčení	horší rovnoměrnost otáčení
dražší	levnější

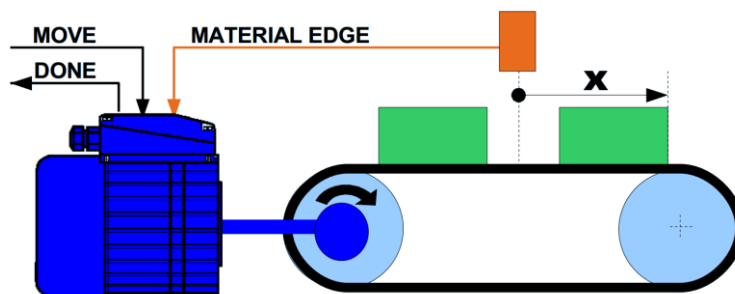
1.3 Oblasti možného využití servomotorů

Servomotory se využívají v celé řadě aplikací motorů a to často i na místech, kde využití jejich zpětné vazby nemusí výt na první pohled zřejmé.

1.3.3 Pohony s jedním servomotorem

Jak již bylo zmíněno, základem servopohonu je řízení založené na zpětné vazbě motoru, která je zapojená do řídicího obvodu motoru. Může se tedy jednat o vazbu přímou kontrolu aktuální polohy motoru, rychlosti otáčení a podobně, nebo může být zdrojem zpětné vazby čidlo, které se samotným chodem motoru nemusí mít nic společného.

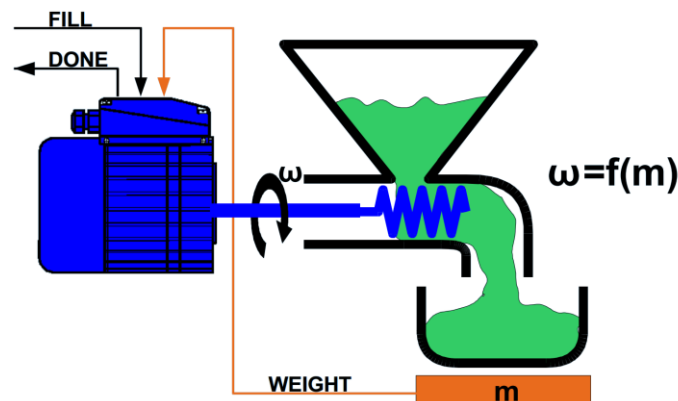
V aplikaci naznačené na Obr. 1.24 se servopohon využívá pro pohon a zastavení posuvníku v přesně definovaném místě. Pokud by vzdálenost výrobků byla konstantní, nemuselo by být nutné využívat zde naznačené externí čidlo a kontrolovat přesné zastavení pomocí snímání polohy motoru. Při použití BLDC, nebo PMSM motoru je takové snímání nutně součástí každého motoru.



Obr. 1.24 Přesné zastavení

Zdrojem zpětné vazby nemusí být nutně údaj o poloze hnané hřídele motoru. V aplikaci servopohonu naznačené na Obr. 1.25 je jako funkce, která řídí ve zpětné vazbě chování motoru zvolena hmotnost nádoby, která se plní materiálem. Dostatečná hmotnost tak

zastavuje motor ve chvíli, kdy je nádoba plná. Další senzory by mohli například podle hustoty, nebo jiných vlastností materiálu, řídit otáčky motoru.



Obr. 1.25 Hmotnostní plnění

Mezi další aplikace servomotorů se dnes začínají řadit i čerpadla, či ventilátory. Na první pohled se zde může servopohon zdát zbytečný, ale pro jeho využití mluví snadná regulace pomocí řídicí jednotky (např. v moderních budovách). Takto se také například ve stanech polních nemocnic pomocí řídicí techniky, senzorů a regulovatelné vzduchotechniky neustále udržuje mírný přetlak. Vzduch je nasáván přes filtry a při otevření dveří potom vzduch proudí směrem ven ze stanu, nedochází tak ke kontaminaci vnitřního prostředí.

Řízení motorů nemusí být závislé na čidlech, samozřejmě jsou také ovládací prvky, které do řídicího procesu mohou zapojit člověka. Je potom na konkrétní aplikaci, jakou rozhodovací a řídicí úlohu má obsluha zařízení. Jsou-li přístupné otáčející se části stroje, nezbytnou součástí aplikace musí být bezpečnostní prvky (např. STOP tlačítka, optické závory, apod.). Tyto bezpečnostní prvky jsou také zapojeny v řídicím systému.

1.3.4 Pohony s více servomotory

Úkolem této práce je v praktické části navrhnout aplikaci, která pracuje s více servomotory. Jedná se o aplikace, které například zajišťují pohyb ve více osách, jako jsou různé manipulátory, zakladače, případně obráběcí stroje, jako frézy a podobně. Podle konkrétní aplikace je nutné řešit synchronní řízení motorů na potřebné úrovni. Pokud si vystačíme pohybem kolmým k osám, stačí řídit motory sekvenčně, jednu osu po druhé. Když má ale systém vykonávat složitější pohyby, jako například opisovat kružnice, je již zapotřebí řídit motory synchronně. Jako aplikaci s více servopohony můžeme vidět také například automatizované výrobní linky.

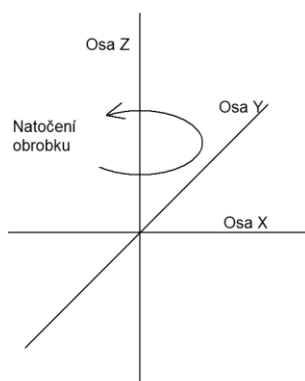
2 Praktická část

Praktickou částí této diplomové práce je využití servomotorů synchronně řízených po sběrnici CAN. Jak už bylo v předchozí části zmíněno, servomotory je možné používat v celé řadě nejrůznějších aplikací. V zadání diplomové práce nebyla konkrétní aplikace přesně stanovena. Po domluvě s konzultantem ve firmě PMControl s.r.o., se kterou na této práci spolupracuji, byl jako aplikace synchronně řízených servomotorů zvolen obráběcí stroj - tříosá frézka.

Frézka

Pohon upravované frézky byly původně osazena asynchronními motory. Motory byly využity pouze k rychloposuvům. Při obrábění se stroj ovládal ručně.

Asynchronní motory byly nahrazeny synchronními servomotory, které je možné mnohem lépe řídit a využít je k pohonu při obrábění.

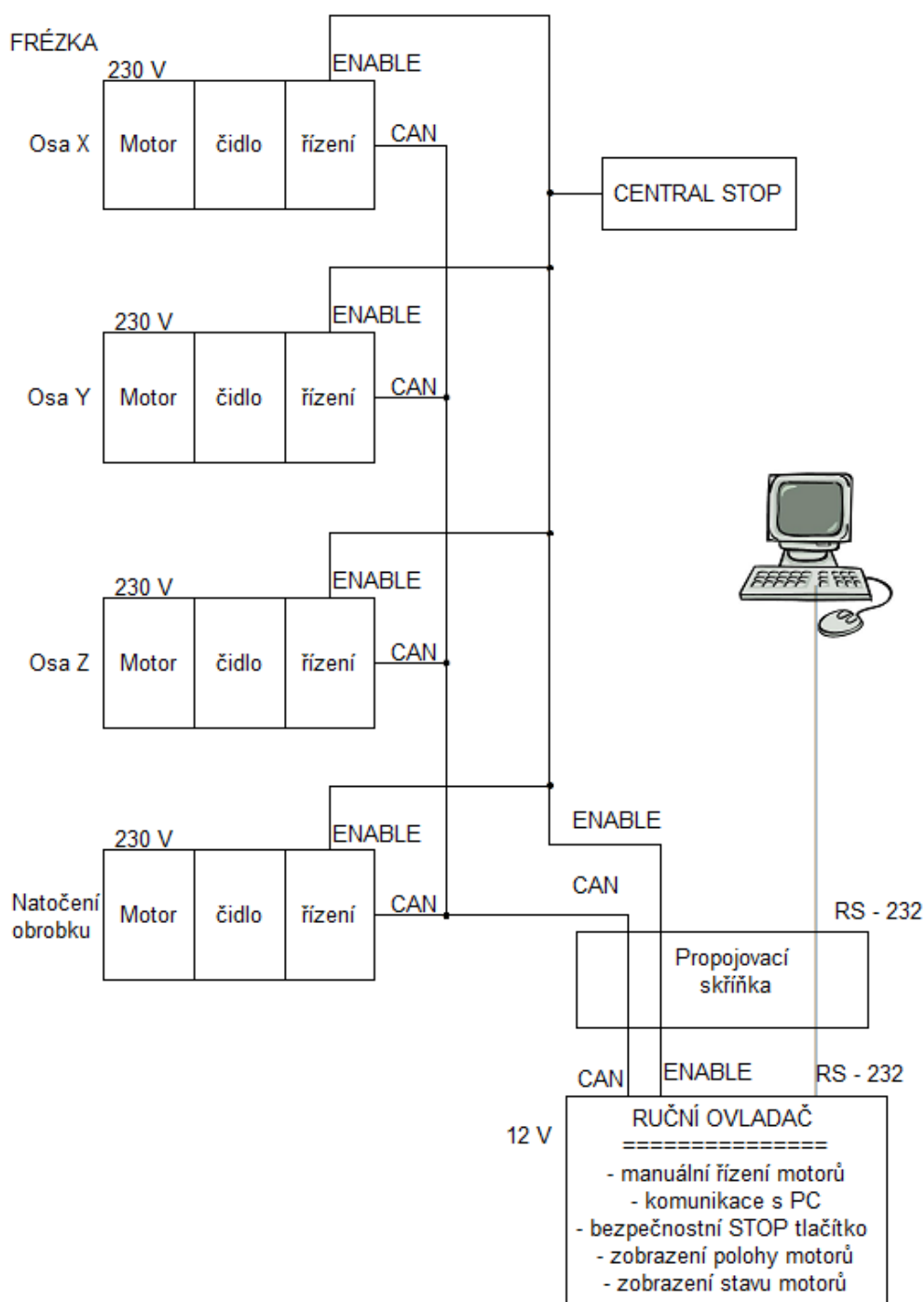


Obr. 2.1 Pohyby, který zajišťují servomotory

Pohony zajišťují pohyb ve čtyřech osách (viz Obr. 2.1). Tři servomotory v aplikaci frézy zajišťují pohyb v osách x, y, z a čtvrtý motor umožňuje natáčení obráběného materiálu kolem osy z. V automatickém režimu je pohyb řízen programem, pro manuální ovládání bylo nutné doplnit ruční ovladač viz Obr. 2.6.

Propojení s počítačem, ve kterém se dá předem definovat pohyb dělá z původně ručně ovládané frézky, stroj, který pro jednodušší aplikace může nahradit mnohem dražší CNC obráběcí stroje. Výrazně se zvýší produktivita práce. Předdefinované pohyby v osách umožňují rychlejší a přesnější výrobu. Náklady na hodinu práce jsou vzhledem k pořizovací ceně nižší než u CNC. Tento přístroj je tak optimální především pro prototypovou výrobu a malé až střední série (stovky kusů).

2.1 Hardware



Obr. 2.2 Blokové schéma zapojení pohonu obráběcího stroje

Blokové schéma zařízení frézky je na obr 2.2.

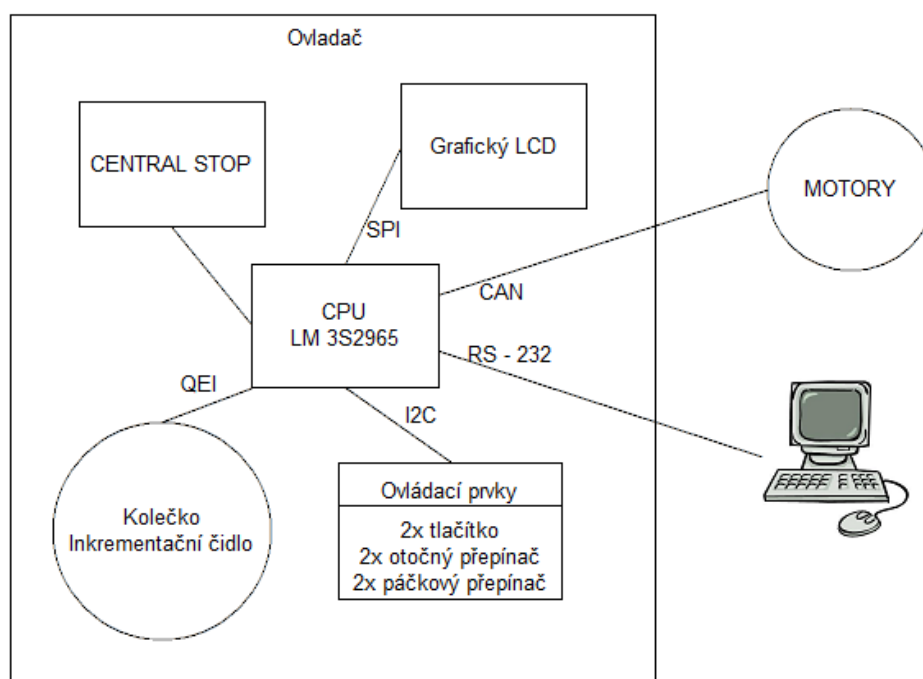
Pro pohon byly použity servomotory s integrovanou elektronikou firmy PMControl G66UF317. Řídící a výkonová elektronika a čidlo polohy je integrovaná přímo do motoru. Všechny servomotory jsou řízeny po sběrnici CAN. Každý motor má vstup ENABLE, jehož deaktivací dojde k automatickému zastavení motoru. Tato funkce je využita pro bezpečnostní funkci – tlačítko STOP, koncové spínače, atd.

Z ručního ovladače (Obr. 2.6) jsou ovládány všechny funkce stroje. Je možné řídit jednotlivě každý motor zvlášť nebo přepnout na režim, kdy je pohyb řízen po předem v počítači definované trajektorii. Na displeji ovladače se zobrazuje stav a aktuální polohu motorů.

Jednotlivé požadované polohy jsou definovány na PC. Program ručního ovladače řeší přepočítání trajektorie do jednotlivých os. Ovladač komunikuje s PC po sběrnici RS – 232.

Propojení vodičů je mechanicky realizováno v propojovací skříňce viz 2.1.2.

2.1.1 Ruční ovladač obráběcího stroje



Obr. 2.3 Ruční ovladač - Blokové schéma

Centrem celého systému je ruční ovladač. Jako základ byl použit ovladač HC32 firmy PMControl, který byl upraven pro účely této aplikace. Z původní aplikace, která řešila ovládání vysokozdvizného vozíku, byla využita deska s řídicím mikrokontrolérem Stellaris LM3S2965 od firmy Texas Instruments s připojeným grafickým LCD. Centrálním prvkem ovládání aplikace je na nastavovací kolečko (Obr. 2.4), připojené na Quadrature encoder vstup procesoru Stellaris.

Ovladač byl doplněn deskou plošného spoje, na které jsou mechanicky upevněny všechny ovládací prvky – přepínače, tlačítka, nastavovací kolečko. Tato deska je osazena mikrokontrolérem Texas Instruments MSP430G2453, který zajišťuje čtení tlačítek a přepínačů. Díky využití sběrnice I2C ke komunikaci s procesorem Stellaris došlo k výraznému snížení počtu propojovacích vodičů.

Vývoj HW a SW ovladače je výsledkem této diplomové práce.

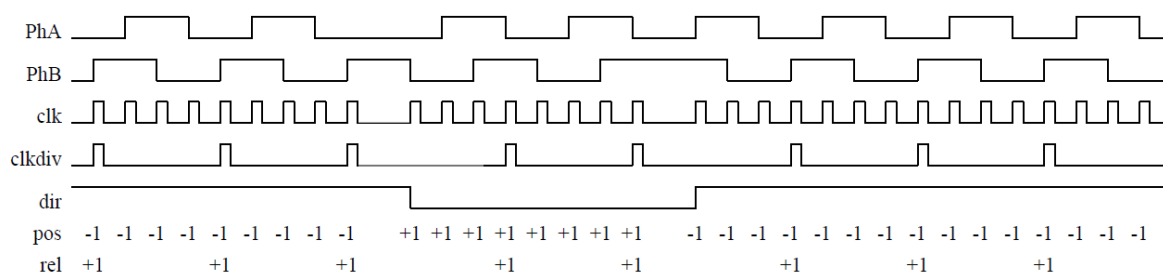
2.1.1.1 Nastavovací kolečko



Obr. 2.4 Nastavovací kolečko

Nastavovací kolečko se využívá v manuálním režimu k řízení pohybu v jednotlivých osách (osa a rychlost pohybu se volí přepínači). V principu se jedná o inkrementální čidlo polohy (Quadrature encoder).

Nastavovací kolečko je mechanicky aretováno na 100 kroků na jedno otočení. Vlastní čidlo má však rozlišení 400 kroků na otáčku. Pro čtení polohy ovladače je tedy toto nutné zohlednit. Výhodou je plynulejší pohyb motorů mezi jednotlivými kroky. Současně je omezena pravděpodobnost chybně detekovaného přechodu na další krok. Průběhy pro oba směry pohybu jsou vyobrazeny na Obr. 2.5.



Obr. 2.5 Průběh signálů pro detekci QEI [23]



Obr. 2.6 Ovladač

2.1.2 Propojovací skříňka

Pro spojení ovladače, motorů, koncových spínačů a mezi ovladačem s počítačem, je vyrobena propojovací skříňka. Ovladač se připojuje pomocí konektoru CAN 15, kterým jsou k němu připojena ostatní zařízení, včetně napájecího napětí 12V. Pro připojení počítače po sběrnici RS - 232 a motorů po sběrnici CAN, jsou standardně konektory CAN 9. Pro RS - 232 zásuvka, pro CAN vidlice, aby nemohlo dojít k záměně. Připojení koncových spínačů frézky je řešeno dvoupólovým konektorem. To zajišťuje odolnost proti připojení nežádoucích zařízení.



Obr. 2.7 Propojovací skříňka

Koncové spínače na osách a tlačítka pro nouzové zastavení jsou zapojeny jako rozpínací tlačítka v sérii, která jsou připojena k napětí 12 V, řídicí procesor hlídá, zda je napětí na požadované úrovni. V případě stisku STOP tlačítka, či aktivaci koncového spínače dojde k rozpojení obvodu, napětí poklesne. Na to reaguje procesor, zastavením všech motorů včetně vypnutí výkonových částí motorů po sběrnici CAN. Z bezpečnostních důvodů je tato funkce duplicitně řešena přivedením signálu ze stop tlačítka současně na vstup ENABLE každého motoru. Každý motor samostatně vypne výkonový stupeň nezávisle na ovladači.

2.1.3 Servomotory

Pro tuto aplikaci jsou použity motory G66UF317 od firmy PMControl.

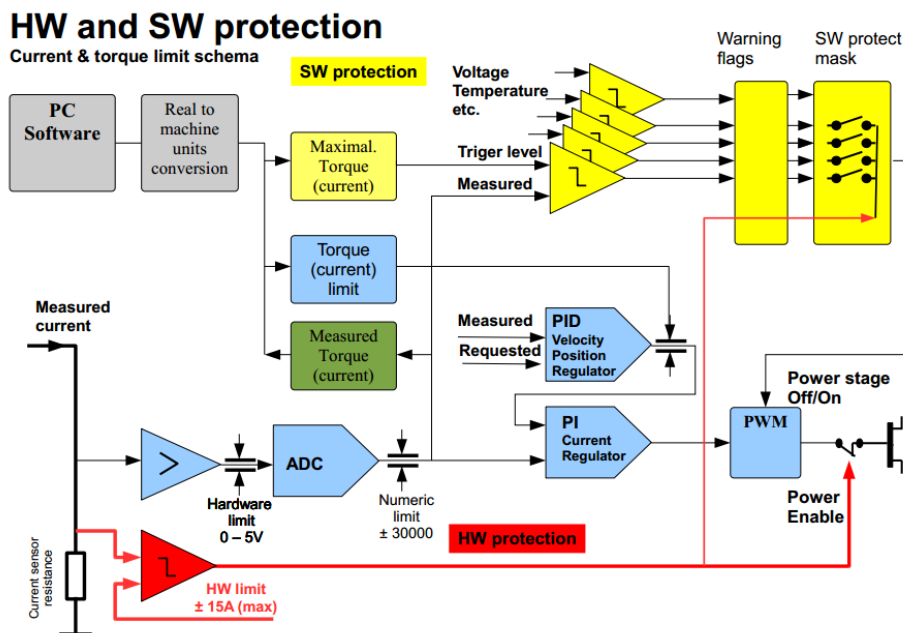
Polohové čidlo servomotoru má rozlišení 3072 inkrementů na otáčku. Rozlišení je tedy přibližně $\pm 0,06^\circ$ bez převodovky. Skutečná přesnost nastavení absolutní polohy je pod 1° . Opakovatelnost nastavení polohy je do $0,25^\circ$. Výsledná přesnost pohonu se přímo úměrně zvyšuje použitím převodovky. Na otáčku výstupního hřídele je zapotřebí více otáček motoru. S planetovou převodovkou je přesnost pro uvedenou aplikaci dostatečná. Při stoupání pohybového šroubu 5 mm na otáčku a převodovkou 1:5 vychází absolutní přesnost v osách přibližně 0,003 mm.

V této aplikaci pracují motory v polohovém režimu, kde se kromě cílové polohy také nastavuje rychlost otáčení, zrychlení a zpomalení motoru, případně omezení momentu.

Komunikace s motory probíhá po sběrnici CAN, přičemž je dodržen standard CANOpen a jeho rozdělení identifikátorů (Tab. 1.4). Každý motor v systému má svoji specifickou část identifikátoru (adresy), která tvoří spodní 4 bity každého identifikátoru. Identifikátor 0 je vyhrazen pro servisní účely, zbytek identifikátorů umožňuje do systému připojit až 15 motorů.

Každý motor má ve své řídicí jednotce implementovanou řadu ochranných funkcí (např. Přepnutí, podpětí, přetížení, překročení teploty, atd.) (Obr. 2.8). Programově lze nastavit reakci na chybový stav – pouze upozornění (warning), nebo přímo vypnutí výkonové části motoru (error).

Tzv. HW chyba motoru je reakcí na maximální proud motoru nezávisle na SW dojde k vypnutí výkonové části motoru.

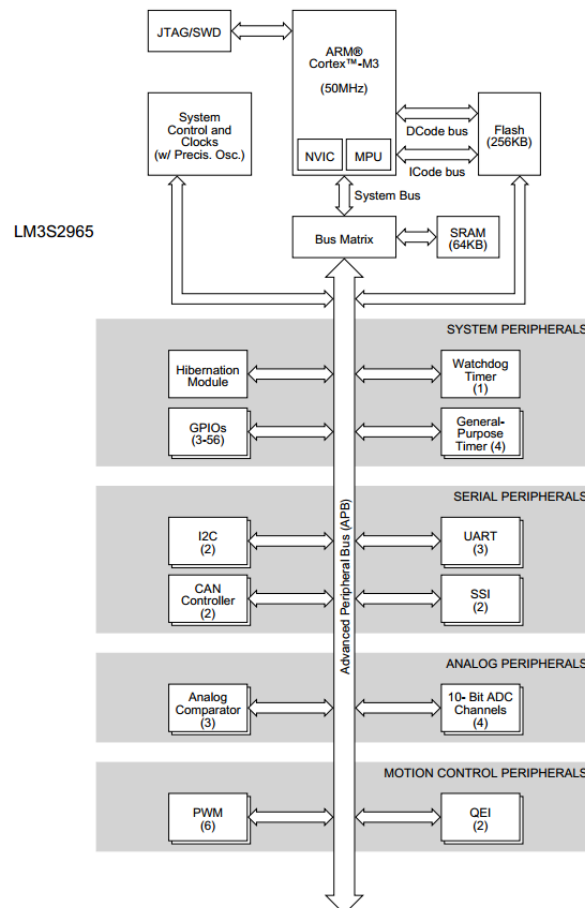


Obr. 2.8 SW definované ochrany motorů PMControl G6x [25]

2.2 Software

2.2.1 MCU

Mikrokontrolér Stellaris LM3S2965 od firmy Texas Instruments má jádro architektury ARM Cortex-M3, 256 KiB Flash a 64KiB RAM. Obsahuje všechny standardní periferie (vstupně výstupní porty, čítače/časovače, A/D převodníky, atd.), a pro tuto aplikaci důležitou periferii QEI enkodér.



Obr. 2.9 Blokové schéma mikrokontroléru [23]

Při práci na SW jsem využil knihovny StellarisWare, DRVLIB (knihovna pro periferie), GRLIB (grafické prvky displeje), IQMATH (knihovna pro matematické funkce) od Texas Instruments. Dále také knihovnu PMControl - pro LCD - SPI, pro černobílý display (4 úrovně šedé), která vychází z GRLIB.

Navržená aplikace má přibližně 51KiB, takže paměťové možnosti procesoru umožňují i rozsáhlý další vývoj.

2.2.2 Struktura SW

V hlavní smyčce programu se kontroluje, jestli nedošlo k pohybu nastavovacího kolečka, aktualizuje se displej podle přijatých dat o stavech a polohách motoru.

Pokud dojde ke stisku tlačítka, nebo přepnutí kteréhokoli přepínače, dojde k přerušení od I2C, přes které sekundární mikroprocesor pošle aktuální stavy ovládacích prvků. Podle těchto změn se potom v přerušení upraví nastavení. Stav, ve kterém se ovladač nachází je

uložen v 32 bitové globální proměnné nazvané *akt_stav*, kde každý nastavitelný parametr má svůj příznakový bit.

Timer vyvolává přerušení každých 25 ms, v samotném přerušení se pouze nastaví příznak, že tento časový interval uběhl. V hlavní smyčce programu se pak na základě tohoto příznaku vyžádá aktuální poloha a stav jednoho z motorů (stavy všech 4 motorů se tedy přečtou každých 100 ms). Také se po tomto přerušení timerem v manuálním režimu nastavují nové cílové polohy motorů.

Sběrnice CAN vyvolává přerušení, buď jako potvrzení odeslané zprávy, nebo po přijetí zprávy. Při přijetí zprávy s aktuální polohou a stavem motoru (oba údaje jsou v jedné zprávě) se ještě v přerušení načtené údaje uloží do příslušných proměnných struktury, která je definovaná pro každý motor. V hlavní smyčce se podle nově přidaných dat aktualizuje displej.

Po sběrnici RS - 232 se do ovladače přenáší řádek po řádku data podle kterých se vykonává pohyb v automatickém režimu. Každý řádek obsahuje informace o koncových polohách všech motorů v tomto řádku, rychlost pohybu v ose Z, rychlost otáčení v pro motor A a rychlost posunu společnou pro osy X a Y. Dále vlastní identifikátor a identifikátor řádku, kterým se při automatickém obrábění bude pokračovat. Ovladač si vyžádá z počítače řádek jeho identifikátorem. Po přijetí, se tato data rozdělí a uloží do příslušných proměnných ve struktuře NextRow.

Procesor vyhodnocuje vstup bezpečnostní smyčky (stop tlačítka), při rozpojení vyšle procesor po sběrnici CAN povely k vypnutí výkonové části motorů.

2.2.3 Uživatelský interface (GUI)

Uživatelské rozhraní je navrženo tak, aby umožňovalo neustálou a přehlednou kontrolu všech čtyř servomotorů. Pracovat se systémem servomotorů s ovladačem se dá ve dvou módech: manuálním a automatickém. Funkce těchto režimů je popsána dále (2.2, 2.2.3.2), pro přechod mezi režimy je na ovladači horní dvoustavový přepínač.

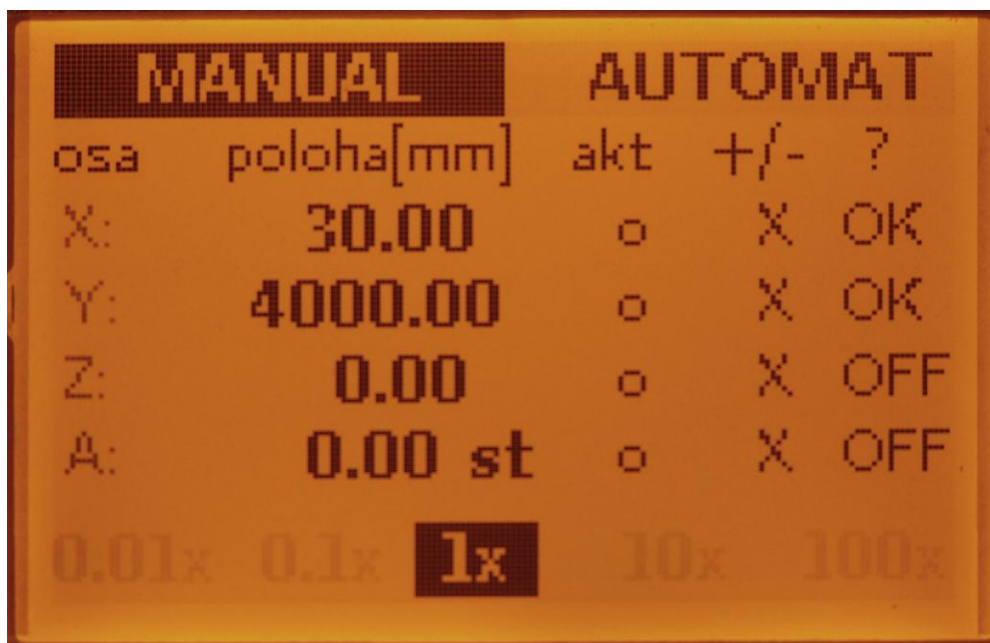
2.2.3.1 Ovladač v manuálním režimu

Manuální režim slouží k ovládání jednotlivých motorů přímo nastavovacím kolečkem, bez předdefinované trajektorie. Uživatel zvolí velikost kroku (otočný přepínačem vpravo), po kterých se bude pohybovat, a osu ve které se bude pohybovat pracovat (otočný přepínačem vlevo), nastavení se může měnit. Nastavovacím kolečkem ovládá pohyb v nastavené ose.

Na displeji má uživatel neustále k dispozici základní informace o motorech včetně aktuální polohy. První sloupec označuje osu, kterou daný motor ovládá, aktivní osa je zvýrazněna tučně. Ve druhém sloupci je aktuální poloha motorů. Dále následuje informace, který z motorů je právě aktivní (**x** – aktivní motor, **o** - neaktivní), zda je motor nyní v pohybu, pokud ano, tak na kterou stranu se točí (+ otáčí se v kladném směru, **X** motor stojí, - otáčí se v záporném směru), v posledním sloupci je zobrazen stav motoru.

Stavy motorů mohou být 4:

- **OK** - bezproblémový stav motoru
- **!** - motor hlásí varování
- **ERR** - motor hlásí chybu
- **OFF** - motor není připojen



Obr. 2.10 Manuální režim

Nabízí se několik variant, jak uživateli umožnit řízení. Při vývoji jsem pracoval především s variantou, kdy se využívá minimální počet ovládacích prvků. Pro celou aplikaci jsem vystačil s již zmiňovaným otočným ovladačem a jedním tlačítkem. Při stisknutí tlačítka uživatel otočným ovladačem mění aktivní řádek: volí mezi motory, případně může volit změnu kroku nebo režimu, ve kterém ovladač pracuje. Když je aktivní řádek spodní, volí velikost kroku, na výběr je z předdefinovaných možností: 0,01x, 0,1x, 1x, 10x a 100x. Pokud uživatel zvolil jako aktivní řádek první, může přecházet mezi Automatickým a Manuálním režimem. Je-li vybrán některý z motorů, uživatel otočným ovladačem po krocích řídí pohyb ve vybrané ose. ***Tento přístup byl ale nakonec shledán jako nevhodný. Když by při frézování došlo omylem ke stisku tlačítka, mohlo by dojít k nechtěné změně aktivního motoru a tím také ke zničení obrobku, zranění, atd.***

Ovládání manuálního režimu je po všech úpravách řešeno větším počtem ovládacích prvků. Jsou zde dva otočné přepínače, levým se vybírá pohon osy, kterým se bude otáčet. Pravým otočným přepínačem pak volíme velikost kroku. Velkým otočným ovladačem se používá jen pro samotný pohyb motorů. Pro přepínání mezi manuálním a automatickým režimem je zde také samostatný přepínač. Z důvodu nárůstu počtu ovládacích prvků přestalo být vhodné připojovat všechny přímo k mikrokontroléru Stellaris LM3S2965. Na dlouhých vodičích by v elektromagneticky silně rušeném prostředí dílny docházelo k problémům s rušením čtení stavů ovládacích prvků, navíc se výrazně zvýšil počet potřebných pinů potřebných pro vstupy. Proto byla pro čtení ovládacích prvků navržena DPS s mikrokontrolérem MSP430 a k přenosu informací o stavech se používá dvou vodičová sběrnice I2C.

2.2.3.2 Ovladač v automatickém režimu

Při práci v automatickém režimu ovladač řídí servomotory z tabulky předem definované v PC. Každý řádek tabulky má svůj identifikátor a obsahuje informace o koncových polohách motorů, rychlostech a zrychleních pohybu motorů, a také identifikátor následujícího řádku, který se má vykonávat. Tím je možné pohyby motorů cyklit.

	MANUAL	AUTOMAT
osa	poloha[mm]	dalsi pol.[mm]
X:	30.07	4000.00
Y:	4000.00	3.00
Z:	0.00	0.00
A:	0.00 st	0.00
krok:	2	nasl.: 3
		OK

Obr. 2.11 Automatický režim

V dolní části displeje je zobrazen identifikátor aktuálního kroku (který se právě vykonává, nebo po stisku tlačítka vykonávat bude), identifikátor následujícího kroku (na který odkazuje aktuální krok) a stav motorů. Stav motorů je zde zobrazen společný pro všechny motory, stav "OK" je pouze když jsou všechny motory ve stavu "OK", když alespoň jeden motor hlásí varování, již se zobrazuje stav "!". Stačí, aby jeden motor hlásil chybu a zobrazí se stav "ERR". Chybový stav má před varováním přednost. Na levé straně displeje jsou stejně jako v manuálním režimu aktuální polohy motorů. Pravá strana se ale oproti manuálnímu režimu změnila, jsou zde vypsány koncové polohy motorů po následujícím řádku.

Různé nástroje se v různých obráběných materiálech nemohou pohybovat libovolnou rychlostí, aby nedošlo ke zničení ať už nástroje či obrobku. Pro manuální režim se rychlost nastavuje konstantní pro všechny osy, při práci v automatickém režimu se rychlosti posunu mohou měnit s každým krokem. Rychlosti posunu jsou v přenášených datech z počítače tři:

- Rychlost frézování - společná rychlost pro osy XY, z počátečních a koncových poloh v osách X a Y se rychlost rozdělí do obou os
- Rychlost posunu v ose Z
- Rychlost otáčení obrobku - rychlost pro motor otáčející s obrobkem, označovaný v aplikaci jako motor A

Držením tlačítka ROW spolu s otáčením kolečka je možné přecházet mezi řádky tabulky souřadnic a vybírat požadovaný řádek. Podle požadovaného způsobu obrábění je pak možnost spustit obrábění tlačítkem HOP, podle polohy SPODNÍHO PŘEPÍNAČE na sebe kroky navazují automaticky nebo se na další přejde opětovným stiskem tlačítka HOP.



Obr. 2.12 Ovládací tlačítka pro automatický režim

2.2.3.3 Bezpečnostní zastavení

Pokud se rozpojí bezpečnostní smyčka, procesor zastaví další vykonávání programu, všem motorům dá povel pro vypnutí výkonové části a tím k zastavení pohybu a uvolnění hřídel.

Stav je indikován nápisem na displeji.



Obr. 2.13 Display po stisku bezpečnostního tlačítka STOP

2.2.4 Výpočty pohybu

V programu se pracuje se třemi parametry pro polohu každého motoru:

- aktuální poloha - aktuální poloha snímače polohy motoru, pro zobrazení na displej je přepočtena na milimetry nebo stupně
- požadovaná poloha - cílová poloha, kam má motor dojet
- rozdíl polohy - rozdíl mezi aktuální a nastavenou požadovanou polohou, využívá se v manuálním režimu a při pohybu po ose se zde z jednotlivých kroků v milimetrech opět při zohlednění všech převodů přepočítá

Při inicializaci se čte aktuální poloha motorů, na tuto hodnotu se nastavuje požadovaná poloha, aby se motory netočily. Tím je definována výchozí (aktuální) pozice motorů.

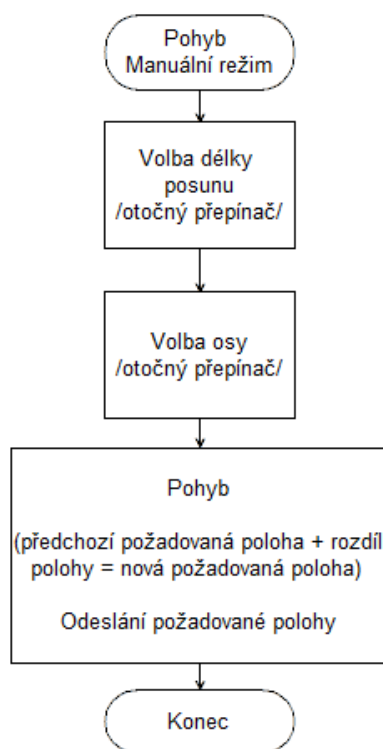
Čtení polohy a ovládání motorů probíhá ve smyčce každých 100 ms. Po tomto čase se vždy přečtou všechny polohy a stavy motorů, je-li vyžadován pohyb, zašle se nová požadovaná poloha.

2.2.4.1 Výpočty v manuálním režimu

V manuálním režimu se pohyb řídí velkým nastavovacím kolečkem. Pohyb se odvíjí po krocích od poslední požadované hodnoty. Tato poloha se ale vlivem dynamické chyby (zrychlení, změna zatížení, PID regulace) nemusí přesně shodovat s aktuální polohou.

Pro pohyb ale není možné brát jako vztažnou aktuální polohu. Chyba by se v průběhu řízení integrovala.

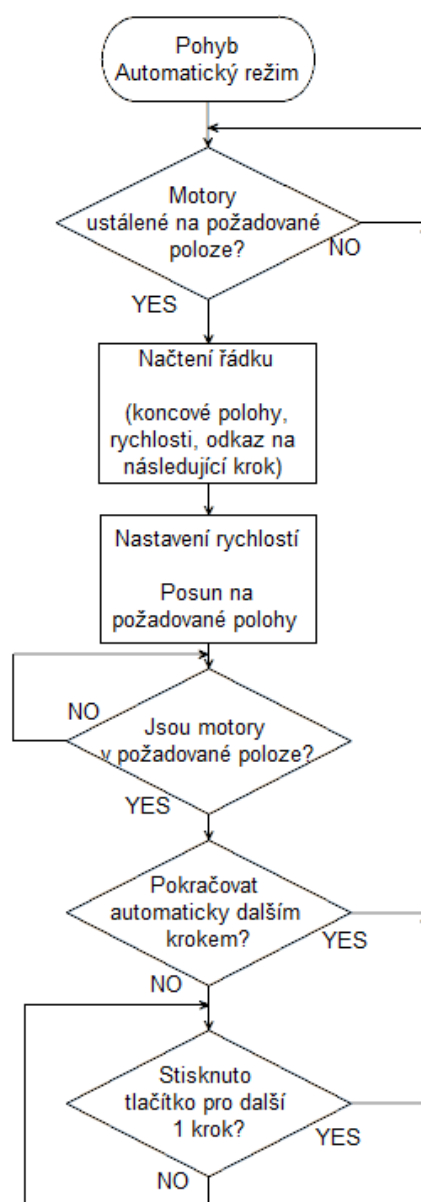
K dalším chybám by docházelo, považováním aktuální pozice za vztažnou během pohybu o více kroků. Krok by se počítal od aktuální polohy, ne až od konečné polohy po předcházejícím kroku.



Obr. 2.14 Diagram pohybu v manuálním režimu

2.2.4.2 Výpočet požadované polohy automatickém režimu

V automatickém režimu jsou polohy zadané v tabulce koncových souřadnic, která se řádek po řádku přenáší z PC. Pro každý řádek se přepočítává pohyb po osách zvlášť.



Obr. 2.15 Diagram pohybu v automatickém režimu

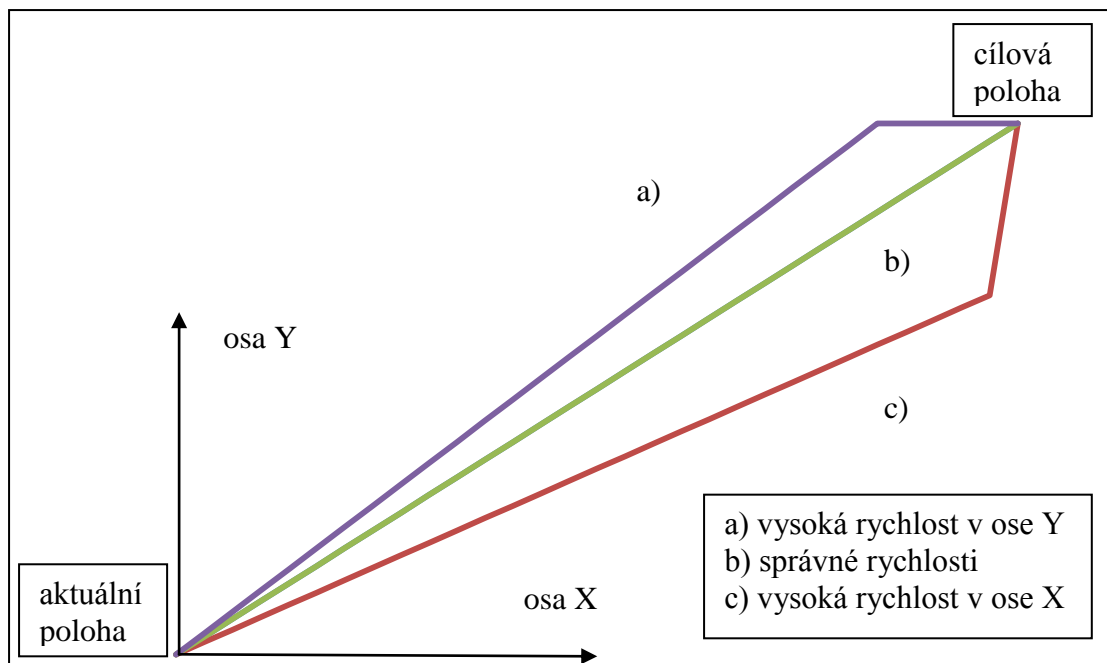
Pro určení, zda jsou motory v ustálené požadované cílové poloze se sledují dva parametry – poloha a rychlost. Poloha se vyhodnocuje porovnáním požadované a aktuální pozice. Při porovnání je nutno povolit možnou malou odchylku, protože se polohy vlivem dynamické chyby nemusí přesně shodovat.

K vyhodnocení nulové rychlosti motoru je využit bit `Motor_Run` ve stavovém slově motoru. Ten indikuje, že rychlost motoru klesla k nule s tolerancí danou firmware motoru.

Samotné hlídání pouze jednoho parametru nestačí.

Při pohybu v rovině X a Y je požadovaný pohyb nutno rozdělit mezi motory X a Y tak, aby výsledný pohyb odpovídal technologickým požadavkům obrábění. Prozatím je řešen pouze lineární pohyb, ale při dalším vývoji jistě dojde i na pohyb po obecné křivce. To bude však vyžadovat real-time výpočet požadované polohy a zadání souřadnic do motorů v taktu maximálně 5ms.

Z aktuálních a cílových poloh motorů se vypočítá pomocí Pythagorovy věty celková dráha pohybu. Podle poměru mezi drahou v dané ose ku celkové dráze pohybu se rozdělí společná rychlost pohybu, zrychlení a zpomalení. Tím se zajistí, že se oba motory budou pohybovat stejně dlouho a výsledný vektorový součet jejich pohybů bude lineární. Případy, jak výsledný pohyb vypadá, jsou na Obr. 2.16. Průběh b) znázorňuje správně provedený pohyb, průběhy a) a c) znázorňují nesprávné rozdělení parametrů pohybu.



Obr. 2.16 Úhlopříčný pohyb

Výrobce motorů definuje přepočty mezi interními proměnnými pro polohu, rychlost a zrychlení a jejich fyzikálním rozměrem. [24]

Pro nastavení polohy se používá přepočet:

$$\text{hodnota pro nastavení polohy} = \text{požadovaná poloha} [^\circ] * \frac{3072}{360} * \text{převod}$$

Pro nastavení rychlosti otáčení:

$$\text{nastavená reprezentace rychlosti} = \text{požadovaná rychlost} \left[\frac{\text{ot}}{\text{min}} \right] * \frac{16}{3} * \text{převod}$$

Pro zrychlení a zpomalení je stejný přepočet. Motor ale umožňuje rozdílné hodnoty těchto dvou parametrů.

$$\text{nastavená reprezentace zrychlení} = \text{požadované zrychlení} \left[\frac{\text{ot}}{\text{min}} / \text{ms} \right] * \frac{2048}{30} * \text{převod}$$

3 Závěr

V této diplomové práci jsou popsány prostředky které se v průmyslu používají k řešení aplikací, které využívají servopohonů.

V první řadě zde jsou zmíněny nejběžnější průmyslové sběrnice, jejich stručný popis vlastností i principu funkce, srovnání. Vznikl tak přehled v průmyslu nejpoužívanějších sběrnic a to nejen těch, které jsou v praktické části používány.

Dále zde jsou popsány motory, které se běžně používají jako serva. Jedná se především o elektronicky komutované motory a synchronní motory s permanentním magnetem, které se používají nejběžněji. Popsány jsou ale i další typy motorů, které je možné jako servopohony využít.

Praktická aplikace, která vznikla při diplomové práci řeší implementaci servomotorů do obráběcího stroje – frézky.

Byl vyvinut ruční ovladač, který synchronně řídí čtyři motory po sběrnici CAN. Navržený ovladač obsahuje dva procesory firmy Texas Instruments, které mezi sebou komunikují po sběrnici I2C. V hlavním procesoru běží řídicí program ovladače, druhý kontroluje stav ovládacích prvků a zajišťuje ošetření tlačítek a přepínačů proti chybné detekci. Vývoj ovládání a funkcí ručního ovladače vznikl ve spolupráci s koncovým zákazníkem.

Ovladač může pracovat v manuálním, nebo automatickém režimu. Při manuálním řízení se nastavuje požadovaná poloha jednotlivých motorů přímo ovladačem. V automatickém režimu se koncové polohy přenáší do ovladače po sběrnici RS-232 z PC. Ovladač přepočítá z požadovaných poloh pohyby motorů.

Použitá literatura

- [1] ZEZULKA, František. *Automatizační prostředky*. 2. vyd. Brno: VUT, 2000, 110 s. ISBN 80-214-1739-0.
- [2] VLACH, Jaroslav. *Řízení a vizualizace technologických procesů*. 1. vyd. Praha: BEN - technická literatura, 1999, 159 s. ISBN 80-860-5666-X.
- [3] SKALICKÝ, Jiří. *Elektrické servopohony*. 2. vyd. Brno: Vysoké učení technické, 2001, 86 s. ISBN 80-214-1978-4.
- [4] POLÁK, Karel. Sběrnice CAN. *Elektrorevue* [online]. [cit. 2013-10-22]. Dostupné z: <http://www.elektrorevue.cz/clanky/03021/index.html>
- [5] ROBENEK, Jan. RS-232 aneb jak šel čas – 1. část. [online]. [cit. 2013-11-15]. Dostupné z: <http://www.hw.cz/teorie-a-praxe/rs-232-aneb-jak-sel-cas-1-cast.html>
- [6] OLMR, Vít. Sériová linka RS-232. [online]. 2005 [cit. 2013-11-16]. Dostupné z: <http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>
- [7] ZAT CONTROL SYSTEMS. *PROFIBUS: Popis PROFIBUS Proces field bus*. Příbram, 1997.
- [8] GESCHIEDTOVÁ, Eva; REZ, Jiří; STENBAUER, Miloslav. *Měření v elektrotechnice*. Brno : VUTIUM, 2002. 182 s. ISBN 80-214-1990-3.
- [9] ZEZULKA, František. *Prostředky průmyslové automatizace*. 1. vyd. Brno: VUTIUM, 2004, 176 s. ISBN 80-214-2610-1.
- [10] ROMÁNEK, David. Co je CANopen a jak na něj. *Hw.cz* [online]. 2006 [cit. 2013-11-30]. Dostupné z: www.hw.cz/produkty/co-je-canopen-a-jak-na-nej.html
- [11] FRÝZA, *Mikroprocesorová technika a embedded systémy: Přednášky(ver.2011-12-16)*.
- [12] RS-485. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-05]. Dostupné z: <http://cs.wikipedia.org/wiki/RS-485>
- [13] Introduction to the Controller Area Network (CAN). CORRIGAN, Steve. [online]. 2008 [cit. 2013-12-07]. Dostupné z: <http://www.ti.com/lit/an/sloa101a/sloa101a.pdf>
- [14] CANopen. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation [cit. 2013-12-09]. Dostupné z: <http://en.wikipedia.org/wiki/CANopen>
- [15] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Praha: BEN - technická literatura, 2005, 247 s. ISBN 80-730-0141-1.
- [16] SKALICKÝ, Jiří. *Elektrické regulované pohony*. Brno, 2007. VUT skriptum.
- [17] YEDAMALE, Padmaraja. Brushless DC (BLDC) Motor Fundamentals. [online]. 2003 [cit. 2013-12-10]. Dostupné z: <http://ww1.microchip.com/downloads/en/appnotes/00885a.pdf>
- [18] L6470: Datasheet – production data. In: [online]. 2012 [cit. 2013-12-12]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/CD00255075.pdf>
- [19] FEDRA. *Text k 5.přednášce předmětu MMIA: Motorky pro řízení mikroprocesorem*. FEKT VUT Brno, 2013.

- [20] KIKUCHI, KENJO, FUKUDA. Remote Laboratory for a Brushless DC Motor. In: [online]. 2001 [cit. 2013-12-12]. Dostupné z: <http://www.ewh.ieee.org/soc/es/May2001/02/Begin.htm>
- [21] CANopen a a SERVOSTAR 400/600: Komunikujeme s měniči SERVOSTAR 400/600. In: [online]. [cit. 2013-12-18]. Dostupné z: <http://www.rs.canlab.cz/?q=cs/node/44>
- [22] WATTERSON, Conal. Controller Area Network (CAN) Implementation Guide. In: [online]. [cit. 2013-12-18]. Dostupné z: http://www.analog.com/static/imported-files/application_notes/AN-1123.pdf
- [23] TEXAS INSTRUMENTS. Stellaris LM3S2965 Microcontroller: DATA SHEET. 2011.
- [24] PMControl s.r.o. EC Motor: CAN – komunikační profil. [online]. [cit. 2014-05-19]. Dostupné z: http://www.pmcontrol.cz/cz/pdf/manual/can_profile.pdf
- [25] JIRÁSEK, Tomáš. PMCONTROL S.R.O. *HW and SW protection*.