České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídící techniky

Komunikační rozhraní pro demonstrační soustavu krokových motorů

Bakalářská práce

Autor práce: Michal Vokáč
Vedoucí bakalářské práce: Ing. Jan Havlík
© 2009

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Michal Vokáč**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný
Obor: Kybernetika a měření

Název tématu: **Kumunikační rozhraní pro demonstrační sestavu krokových motorů**

Pokyny pro vypracování:

1. Seznamte se s výukovým modulem pro demonstraci funkce krokových motorů.
2. Navrhněte komunikační rozhraní umožňující řízení výukového modulu pomocí osobního počítače.
3. Navržené rozhraní realizujte.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Jan Havlík, Ph.D.

Platnost zadání: do konce zimního semestru 2009/10

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 27. 2. 2009

Poděkování

Rád bych na tomto místě poděkoval panu Ing. Janu Havlíkovi za ochotu, odbornou pomoc a cenné rady, které mi poskytl během zpracování této bakalářské práce. Také bych rád poděkoval panu Ing. Aleši Kautskému, který mi pomohl vyřešit celou řadu problémů při programování řídicí aplikace.

Prohlášení

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

V Praze dne 7.6.2009 ..... Michal Pekař .....

podpis

Anotace

Tato práce se zabývá návrhem komunikačního rozhraní pro připojení demonstrační soustavy krokových motorů k PC. V první části práce je popsána použitá soustava s krokovými motory, její struktura a propojení jednotlivých částí. Dále popisuje několik možných variant návrhu komunikačního rozhraní. Poté je popsána zvolená varianta konstrukce komunikačního rozhraní s využitím integrovaného převodníku USB – UART, typ FT232BM, který je pomocí sériové linky propojen s řídicí deskou soustavy.

Další část práce pojednává o tvorbě řídicí aplikace s grafickým uživatelským rozhraním. Popisuje volbu programovacího jazyka, strukturu vytvořené aplikace napsané v jazyce C# a instalaci a provoz této aplikace.

Klíčová slova: krokový motor, USB rozhraní, FTDI převodník, C#

Annotation

This work deals with the design of communication interface to connect stepper motors demonstration system to the PC. In the first part of the work is described the system used with stepper motors, its structure and interconnection of individual parts. It also describes several possible options draft of communication interface. Following is described the chosen design alternative of the communication interface with the integrated converter USB - UART, FT232BM type, which is connected to the control board using serial line.

Another part of the work deals with the creation of control application with a graphical user interface. It describes the choice of programming language, the structure of the created application which is written in C# language and the installation and operation of this application.

Key words: stepper motor, USB interface, FTDI converter, C#

1. Obsah

1. Obsah....6
1.1 Seznam obrázků....8
2. Seznam použitých symbolů a zkratek....9
3. Úvod....10
4. Konstrukční část....11
4.1 Popis ovládané soustavy....11
4.1.1 Popis konstrukce s krokovými motory....11
4.1.2 Popis výkonových desek....11
4.1.3 Popis řídicí desky....12
4.1.4 Popis elektrického propojení....12
4.2 Návrh ovládacího rozhraní....15
4.2.1 Volba počítačového rozhraní....15
4.2.1.1 Paralelní port....15
4.2.1.2 Sériový port....16
4.2.1.3 USB....16
4.2.1.4 Bezdrátové připojení pomocí BlueTooth....16
4.2.1.5 Závěr....17
4.2.2 Konstrukce převodníku USB-UART....17
4.2.2.1 Základní vlastnosti obvodu FT232BM....17
4.2.2.2 Naprogramování EEPROM....19
4.2.2.3 Konstrukce desky....19
4.2.3 Konstrukce nové řídicí desky....20
4.2.4 Propojení desky převodníku USB-UART a řídicí desky....21
5. Programová část....22
5.1 Komunikace mezi aplikací a řídicí deskou....22
5.1.1 Zpráva RESET....22
5.1.2 Zpráva STOP....22
5.1.3 Zpráva ROTATE....23
5.1.3.1 Hlavička....23
5.1.3.2 Číslo motoru....23
5.1.3.3 Směr otáčení motorem....24
5.1.3.4 Způsob otáčení motorem....24
5.1.3.5 Rychlost otáčení motorem....24
5.1.3.6 Počet kroků/sektorů....24
5.1.3.7 Kontrolní součet....25
5.1.3.8 Struktura zprávy....25
5.1.4 Zpráva STATE....25
5.1.5 Zpráva OK....26
5.1.6 Zpráva ERROR....26
5.1.7 Zpráva OPTO....26
5.2 Program pro řídicí desku....27
5.2.1 Činnost programu řídicí desky....27

5.3 Řídicí aplikace pro PC....29
5.3.1 Volba ovladače a programovacího jazyka....29
5.3.1.1 VCP ovladače....29
5.3.1.2 D2XX ovladače....29
5.3.1.3 Závěr....30
5.3.2 Struktura aplikace....30
5.3.2.1 Třída FtCommnication....31
5.3.2.2 Panel ovládání soustavy....32
5.3.2.3 Panel vizualizace soustavy 1....34
5.3.2.4 Panel vizualizace soustavy 2....35
5.3.2.5 Menu....35
5.3.2.6 Informační panel....35
5.3.2.7 Celkový pohled na hlavní okno aplikace....36
5.4 Spuštění a provoz aplikace....37
5.4.1 Instalace .NET Framework....37
5.4.2 Instalace D2XX ovladače....37
5.4.3 Instalace aplikace....37
5.4.4 Ovládaní aplikace a soustavy....38
5.4.4.1 Spuštění aplikace....38
5.4.4.2 Vyhledání zařízení....38
5.4.4.3 Připojení zařízení....38
5.4.4.4 Nastavení konfigurace soustavy motorů....39
5.4.4.5 Nastavení parametrů otáčení....39
5.4.4.6 Ovládání motorů....39
5.4.4.7 Ukončení aplikace....39
6. Závěr....41
7. Použitá literatura a zdroje....42
8. Použitý software....43
9. Obsah přiloženého CD....44
10. Přílohy....45
10.1 Schéma zapojení desky převodníku USB-UART....45
10.2 Obrazec desky plošného spoje převodníku USB-UART....46
10.3 Schéma zapojení řídicí desky....47
10.4 Obrazec desky plošného spoje řídicí desky....48
10.5 Vývojový diagram funkce programu pro řídicí desku....49

1.1 Seznam obrázků

Obrázek č. 4.1 Vzájemné propojení řídicí desky a výkonových desek.....13
Obrázek č. 4.2 Vzájemné propojení převodníku USB-UART a řídicí desky.....21
Obrázek č. 5.1.1 Struktura zprávy ROTATE.....25
Obrázek č. 5.1.2 Struktura zprávy OPTO.....26
Obrázek č. 5.3.1 Komponenta DeviceManagement.....32
Obrázek č. 5.3.2 Komponenta MotorDependences.....33
Obrázek č. 5.3.3 Komponenta MotorControl.....33
Obrázek č. 5.3.4 Signalizace stavu soustavy, tlačítko STOP a RESET.....34
Obrázek č. 5.3.5 Komponenta MotorVisualization.....35
Obrázek č. 5.3.6 Hlavní okno aplikace.....36

2. Seznam použitých symbolů a zkratek

BT	BlueTooth
CD	Compact Disk
DPS	Deska plošných spojů
EEPROM	Electronically Erasable Programmable Read-Only Memory
FTDI	Future Technologi Devices International Ltd.
ISP	In System Programming
LE dioda	LED, Light-Emitting Diode
PC	Personal Computer
PDA	Personal Digital Assistant
PID	Product Identification
SPI	Serial Peripheral Interface
TTL	Transistor Transistor Logic
UART	Universal Asynchronous Reciver Transmitter
USB	Universal Serial Bus
VCP	Virtual COM Port
VID	Vendor Identification

3. Úvod

Tato práce navazuje na bakalářskou práci studenta Vladimíra Nekvasila, s tématem „Ovládání krokových motorů – didaktická pomůcka“ [1], která byla realizována v roce 2008 na Katedře řídicí techniky. Hlavním úkolem jeho práce bylo navrhnout a realizovat jednoduchou soustavu s krokovými motory, která by se dala použít jako didaktická pomůcka pro studenty nebo jako funkční model použitelný například při Dnech otevřených dveří na fakultě a měla by demonstrovat princip a způsob řízení krokových motorů. Pro vylepšení funkce soustavy a rozšíření oblasti aplikace soustavy bylo vhodné umožnit připojení realizované soustavy k počítači PC a řídit tuto soustavu prostřednictvím nějakého grafického uživatelského rozhraní.

Úkolem mé bakalářské práce tedy bylo podrobně se seznámit s vlastnostmi a funkcí realizované soustavy a navrhnout a realizovat vhodné rozhraní pro připojení již zkonstruované demonstrační soustavy krokových motorů k počítači PC a vytvořit aplikaci s grafickým rozhraním pro řízení této soustavy.

4. Konstrukční část

4.1 Popis ovládané soustavy

Popis ovládané soustavy lze rozdělit na popis jednotlivých částí, ze kterých soustava sestává:

- popis konstrukce s krokovými motory,
- popis výkonových desek,
- popis řídicí desky,
- popis elektrického propojení desek.

4.1.1 Popis konstrukce s krokovými motory

Realizovaná soustava sestává z jednoduché dřevěné konstrukce z překližkových desek, na které jsou pomocí kovových L-profilů namontovány dva unipolární krokové motory řady KH42. Ty jsou umístěny tak, že osa hřídele jednoho motoru směřuje kolmo směrem vzhůru a osa hřídele druhého motoru je kolmá na osu prvního motoru a směřuje směrem vpřed. Na hřídelích obou motorů jsou upevněny magnetické unašeče disků CD. Motory jsou pak umístěny v takové vzdálenosti od sebe, aby po přichycení CD disků na unašeče se tyto disky vzájemně protínaly do hloubky přibližně 3 cm. Pro tento účel jsou v discích vyříznuty čtyři drážky rozložené po 90° o šířce přibližně 1 – 1,5 cm. Tímto se hřídele obou motorů stávají vzájemně na sobě závislé a v jednom okamžiku je tak možné otáčet vždy jen jedním motorem, jehož disk prochází drážkou na disku na druhém motoru.

4.1.2 Popis výkonových desek

Každý z motorů je šestižilovým plochým vodičem připojen k desce plošných spojů osazené mikrokontrolérem ATMEL ATmega8 – vždy jedna deska pro jeden motor.

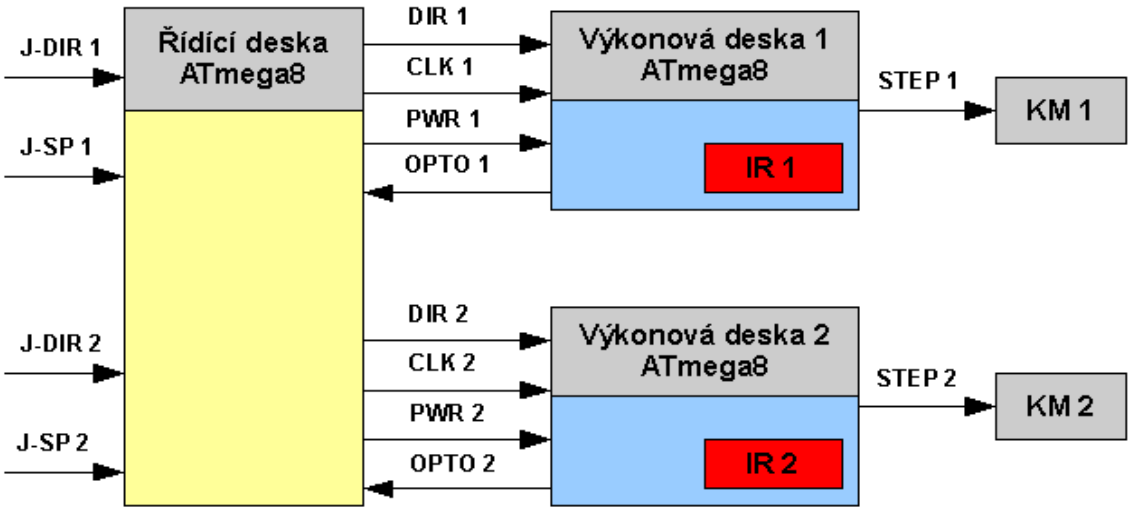
Mikrokontroléry na základě vstupních signálů z řídicí desky generují krokovací posloupnosti, které slouží k ovládání tranzistorových spínačů spínajících jednotlivé fáze krokových motorů a otáčejí motory. Součástí desek jsou také optické závory tvořené LE diodou vyzařující v infračerveném spektru a fototranzistorem citlivým na toto záření. Optické závory slouží k detekci výřezů na discích a jejich stav je v pravidelných intervalech sledován mikrokontrolérem který tuto informaci zpracovává a předává do řídicí desky k možnému rozhodování. Tyto desky použité ke krokování motorů a sledování stavu optických závor budu dále nazývat jako desky výkonové.

4.1.3 Popis řídicí desky

V původním řešení byla dále použita další deska plošných spojů osazená opět mikrokontrolérem ATmega8, ke které byly pomocí pětižilových plochých vodičů připojeny výkonové desky. Úkolem řídicí desky bylo po připojení napájení k soustavě zjistit stav optických závor na výkonových deskách a na základě jejich stavu generovat krokovací pulsy pro výkonové desky a otočit jednotlivými motory tak, aby disky s výřezy byly v poloze, kdy je možné otáčet kterýmkoli motorem. Poté mikrokontrolér začal provádět uživatelem nastavenou demonstrační sekvenci – cyklicky otáčet oběma motory v určitém směru a zvolenou rychlostí. Pro tento účel bylo na desce umístěno několik zkratovacích propojek – jumperů, pomocí kterých bylo možné nastavit směr otáčení a zvolit mezi rychlým nebo pomalým otáčením pro každý motor.

4.1.4 Popis elektrického propojení

Pro budoucí návrh ovládacího rozhraní pro soustavu a zachování úplné kompatibility s realizovaným zařízením bylo potřeba zjistit parametry vzájemného propojení jednotlivých desek a definovat úrovně přenášených signálů v jednotlivých stavech. Vzájemné propojení řídicí desky, výkonových desek a motorů je znázorněno na *Obrázku č. 4.1*.



Obrázek č. 4.1 Vzájemné propojení řídicí desky a výkonových desek

KM 1	krokový motor 1
KM 2	krokový motor 2
IR 1	infračervená optozávora 1
IR 2	infračervená optozávora 2

J-DIR 1	propojka - volba směru otáčení motoru 1
J-SP 1	propojka - volba rychlosti otáčení motoru 1
J-DIR 2	propojka - volba směru otáčení motoru 2
J-SP 2	propojka - volba rychlosti otáčení motoru 2

DIR 1	signál z řídicí desky do výkonové desky 1, určuje směr otáčení
CLK 1	signál z řídicí desky do výkonové desky 1, krokovací pulsy
PWR 1	napájení výkonové desky 1
OPTO 1	signál se stavem optozávory na výkonové desce 1 do řídicí desky

DIR 2	signál z řídicí desky do výkonové desky 2, určuje směr otáčení
CLK 2	signál z řídicí desky do výkonové desky 2, krokovací pulsy
PWR 2	napájení výkonové desky 2
OPTO 2	signál se stavem optozávory na výkonové desce 2 do řídicí desky

STEP 1 napájení a krokování motoru 1

STEP 2 napájení a krokování motoru 2

Měřením signálů a analýzou programů v mikrokontrolérech řídicí desky a výkonových desek jsem dospěl k těmto parametrům:

Napájení

Celá soustava je napájena ze stabilizovaného zdroje napětí +5V. Toto napětí je přivedeno na řídicí desku a prostřednictvím konektorů je dále distribuováno na výkonové desky (PWR). Maximální odběr spuštěné soustavy, který jsem naměřil je 450mA. Všechny logické signály na soustavě mají TTL úrovně.

DIR

Signál z řídicí desky do výkonové desky, určuje směr otáčení motorem.

Logické úrovně jsou:

logická 0 (0V) otáčení motorem vlevo

logická 1 (+5V) otáčení motorem vpravo

CLK

Signál z řídicí desky do výkonové desky, jedná se o obdélníkové pulsy o šířce 2ms. Po přijetí pulsu dojde k otočení motorem o jeden krok ve směru daném logickou úrovní na DIR.

Logické úrovně jsou:

logická 0 (0V) linka je v klidu, motor se neotáčí

logická 1 (+5V) 2ms puls, motor se otočí o jeden krok

OPTO

Signál z výkonové desky do řídicí desky se stavem optozávory. Mikrokontrolér na výkonové desce v pravidelných intervalech sleduje stav optozávory a poté tento stav předá do řídicí desky k možnému zpracování.

Logické úrovně jsou:

logická 0 (0V) optozávora je zastíněna, disk je v poloze kdy druhým motorem nelze otáčet

logická 1 (+5V) optozávora je volná, disk je v poloze kdy druhým motorem lze volně otáčet

4.2 Návrh ovládacího rozhraní

4.2.1 Volba počítačového rozhraní

Při návrhu ovládacího rozhraní bylo zapotřebí kvůli zachování zpětné kompatibility se současnou konstrukcí zařízení vzít v úvahu dříve zjištěné elektrické parametry soustavy. Dalšími významnými kritérii při volbě které počítačové rozhraní zvolit pro navrhované komunikační rozhraní bylo:

- dostupnost rozhraní na většině standardních i starších typech PC,
- možnost jednoduchého propojení a komunikace s mikrokontrolérem ATMEL,
- jednoduchost, snadná reprodukovatelnost zapojení,
- cenová dostupnost.

Podle předchozích kritérií přicházelo v úvahu použít pro ovládání některou z následujících možností:

- paralelní port,
- sériový port,
- USB,
- bezdrátové připojení pomocí BlueTooth.

4.2.1.1 Paralelní port

Využití paralelního portu počítače pro ovládání soustavy by bylo jednoduchým řešením zadaného požadavku, protože pro propojení mikrokontroléru s paralelním portem není nutné použití složitých podpůrných obvodů, převodníků apod. Proti použití paralelního portu při návrhu komunikačního rozhraní však hovoří to, že paralelní porty se ze stolních osobních počítačů pomalu vytrácejí. Také převážná většina přenosných počítačů v současné době již není tímto portem vybavena.

Protože by komunikační rozhraní připojené pomocí paralelního portu bylo odkázáno pouze na použití se stolními počítači a staršími laptopy, nepovažuji jeho použití za vhodné.

4.2.1.2 Sériový port

Návrh komunikačního rozhraní se sériovým portem by byl také velmi jednoduchý. Při použití převodníku úrovní z úrovní linky RS-232 použité v počítači, na TTL úrovně mikrokontroléru, by se dalo využít sériové rozhraní UART v mikrokontroléru. Pro konstrukci rozhraní by stačilo minimum součástek , rozhraní by bylo jednoduché a levné. Situace je však podobná jako u paralelního portu. Sériový port se na současných stolních i přenosných počítačích přestává vyskytovat.

4.2.1.3 USB

Rozhraní USB je asi nejrozšířenějším a nejvyužívanějším počítačovým rozhraním pro připojování periferních zařízení k PC. USB rozhraní je dostupné na všech typech stolních i přenosných počítačů, nových i relativně starých modelů. Z tohoto důvodu by bylo použití USB velice výhodné. Pro možné připojení mikrokontroléru k rozhraní USB jsou na trhu dostupné různé verze integrovaných převodníků USB – UART za relativně nízké ceny, cca. 150Kč/kus. Další značnou výhodou USB je vyvedené napájení použitelné pro napájení uživatelské aplikace. Napájecí napětí USB je +5V a maximální povolený proud je až 500mA.

4.2.1.4 Bezdrátové připojení pomocí BlueTooth

Bezdrátová technologie BlueTooth se zdá být zajímavou možností, jak soustavu propojit s počítačem. Na trhu se vyskytuje široká nabídka komunikačních modulů různých výrobců podporujících technologii BT, které umožňují snadnou vestavbu do navrhovaných zařízení a jednoduché propojení s mikrokontroléry díky sériové lince UART. Výhodou použití tohoto řešení by bylo ovládání soustavy bez nutnosti použití kabeláže pro komunikaci. Také by bylo možné jednoduchou variantu řídicí aplikace vytvořit i pro malá mobilní zařízení jako mobilní telefony nebo PDA podporující technologii BT. Komunikačním rozhraním BT jsou však

vybaveny především přenosné počítače, pro většinu stolních a starších PC bylo nutné použít například USB BlueTooth modul. Řešení s použitím BT by bylo zajímavé a praktické, ovšem bylo by již poněkud složitější a především nákladnější. Cena BT komunikačního modulu se sériovou linkou UART se pohybuje řádově kolem 1000Kč a ceny USB BT modulů začínají na 200Kč.

4.2.1.5 Závěr

Především kvůli jeho největšímu rozšíření a také možnosti přímo využít napájecí napětí +5V a odebírat proud až 0.5A jsem jako nejvhodnější variantu pro konstrukci komunikačního rozhraní pro soustavu zvolil rozhraní USB.

4.2.2 Konstrukce převodníku USB – UART

Aby bylo možné připojit soustavu řízenou mikrokontrolérem ATMEL k počítači pomocí rozhraní USB bylo potřeba navrhnout desku s převodníkem USB rozhraní na sériovou linku podporovanou mikrokontrolérem. Pro tento účel jsem vybíral z produktů společnosti *Future Technology Devices International Ltd.* (FTDI) [2], která je jedním z předních dodavatelů integrovaných obvodů pro práci s USB rozhraním. Z nabídky obvodů pro různé aplikace jsem vybral integrovaný obvod FT232BM.

4.2.2.1 Základní vlastnosti obvodu FT232BM

- jednočipový převodník USB – UART v pouzdře LQPF-32
- plný handshake a plné rozhraní signálů modemu
- podpora 7/8 bitového přenosu, 1/2 stop bitů, parity – lichá, sudá, značená, mezerová, bez parity

- přenosová rychlost:
  - 300Bd až 3MBd – TTL
  - 300Bd až 1MBd – RS232
  - 300Bd až 3MBd – RS485,RS422
- přijímací buffer 384 bytů, vysílací buffer 128 bytů pro vysokou propustnost dat
- nastavitelný time-out přijímače
- automaticky řízený vysílací buffer pro rozhraní RS485
- integrovaný konvertor úrovní UART a řídicích signálů pro spojení s 5V a 3.3V logikou
- integrovaný regulátor 3.3V pro USB obvody
- integrovaný obvod Power-On Reset
- jednoduché napájení v rozsahu 4.35V – 5.25V
- kompatibilní s USB 1.1 a USB 2.0
- možnost uložení VID, PID, sériového čísla, popisu produktu do externí EEPROM
- EEPROM programovatelná přímo v aplikaci prostřednictvím USB
- VCP ovladače – po připojení obvod pracuje jako virtuální sériový port
- D2XX ovladače – přímé ovladače obvodu, umožňují využívat rozšířené funkce obvodu (Bit – Bang, přístup do EEPROM ..)
- podporované operační systémy:
  - Microsoft Windows 98 až Windows Vista, i 64bitové verze
  - Microsoft Windows CE 4.2 – 6.0
  - Linux 2.4.20 nebo vyšší
  - Mac OS 8 až X

Předchozí výčet základních vlastností obvodu FT232BM je stručný výtah z anglického originálu katalogového listu, který je k dispozici na přiloženém CD nebo na internetových stránkách výrobce. [3]

4.2.2.2 Naprogramování EEPROM

Při konstrukci jsem využil možnosti připojit k obvodu externí EEPROM, s níž lze pro ovládání obvodu používat přímé D2XX ovladače, a tak maximálně využít schopnosti obvodu. Pro použití D2XX ovladačů musí být ovšem v EEPROM uloženy parametry výrobce, tedy FTDI.

Naprogramování paměti lze provést například programem Ftd2xxst.exe, který poskytuje výrobce převodníku. Tento program je také součástí přiloženého CD. Údaje výrobce které je potřeba do paměti uložit jsou:

- Manufacturer = FTDI
- Manufatrurer ID = FT
- VID = 0403
- PID = 6001
- Desctription = FTDI HiSpeed Convertor

Pokud by v paměti byly uloženy jiné údaje, zařízení by po připojení k počítači nebylo identifikováno a vyžadovalo instalaci vlastních ovladačů, D2XX ovladače s takovým zařízením nebudou kompatibilní. Pokud ovšem paměť zůstane prázdná nebo se ji nepodaří přečíst, a nebo nebude vůbec připojena, zařízení je možné provozovat s VCP ovladači. Paměť ovšem také umožňuje uložení sériového čísla zařízení. Sériové číslo je tvořeno osmi libovolnými znaky a jeho změna nijak neovlivní kompatibilitu převodníku s D2XX ovladačem. Proto v řídicí aplikaci využiji identifikaci připojené soustavy právě podle sériového čísla.

4.2.2.3 Konstrukce desky

Vlastní deska s převodníkem je realizovaná podle jednoho ze základních výrobcem doporučených zapojení z katalogového listu. Pro připojení desky převodníku k PC je deska vybavena USB konektorem typu B. K připojení převodníku tak lze využít libovolný, snadno dostupný prodlužovací kabel USB A-B, protože tento typ kabelu je používaný například u většiny tiskáren, scannerů nebo podobných typů zařízení, které se k počítači připojují pomocí USB. Všechny vstupní/výstupní vývody převodníku jsou přivedeny na deseti pinový dvouřadý

konektor, který umožňuje připojení řídicí desky nebo i jiného zařízení. Schéma zapojení desky převodníku je v *Příloze č. 10.1*, obrazec desky plošného spoje v *Příloze č. 10.2*. Popis propojení převodníku s řídicí deskou je popsán v samostatné kapitole *4.2.4*.

4.2.3 Konstrukce nové řídicí desky

Vzhledem k tomu, že původní řídicí deska nebyla připravena pro připojení komunikačního rozhraní, tedy převodníku USB - UART – na desce nebyly vyvedeny piny sériové linky mikrokontroléru, případně další potřebné piny, musel jsem navrhnout řídicí desku novou. Pro návrh jsem si stanovil několik základních parametrů které jsem chtěl dodržet:

- použít osvědčený mikrokontrolér ATMET ATmega8,
- zachovat plnou kompatibilitu s realizovanými výkonovými deskami,
- umožnit připojení až 4 výkonových desek,
- možnost volby mezi externím napájením a napájením z USB převodníku,
- možnost přeprogramování mikrokontroléru bez nutnosti ho vyjímat z desky.

Mikrokontrolér ATmega8 jsem se rozhodl použít především po předchozích zkušenostech s jeho použitím a programováním a také proto, že byl použit i v předchozím řešení desky. Tento mikrokontrolér rozsahem vstupních/výstupních portů i výkonově plně dostačuje pro použití až se čtyřmi výkonovými deskami.

Pro provoz dvou výkonových desek s motory je napájení z USB dostačující, při provozu většího počtu motorů by však mohlo dojít k překročení maximálního dovoleného proudu odebíraného z USB portu počítače a tak k jeho možnému poškození. Proto bylo vhodné navrhnout desku s možností napájení z externího zdroje stabilizovaného napětí, například ze síťového adaptéru již dříve použitého pro napájení soustavy.

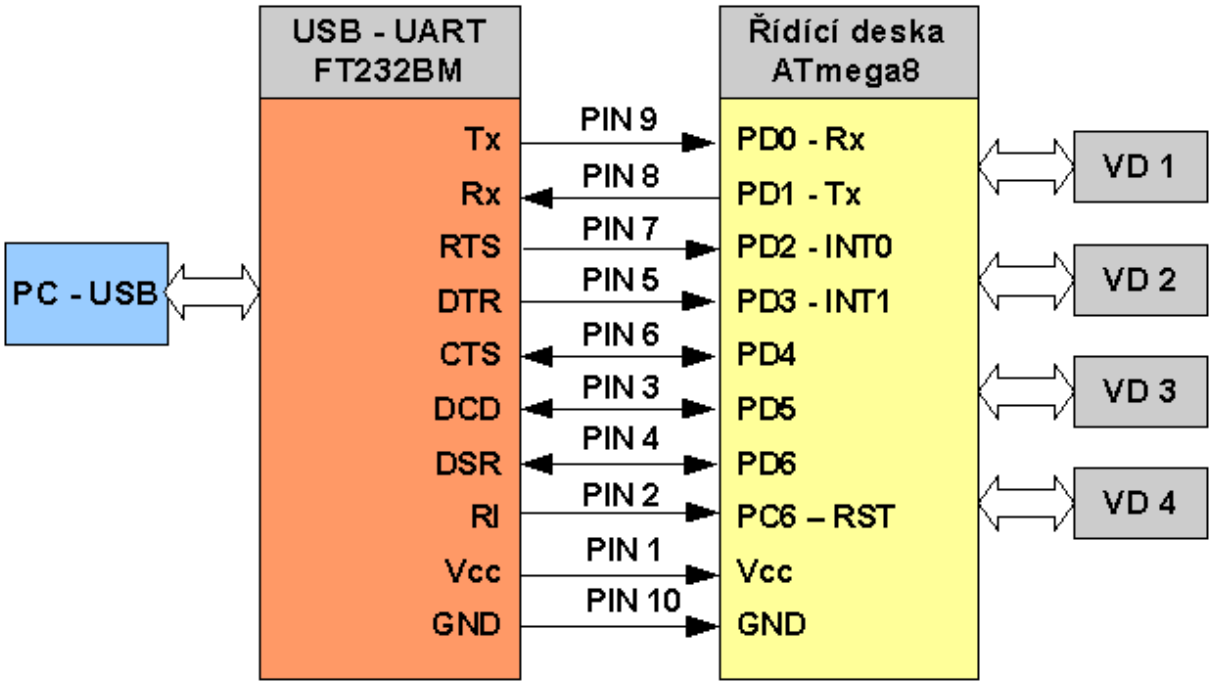
Původní řídicí deska nebyla vybavena konektorem, který by umožňoval přeprogramování mikrokontroléru bez jeho vyjmutí z desky. Pro ladění a testování programu bylo toto velice nepraktické. Protože zvolený mikrokontrolér podporuje možnost sériového naprogramování v aplikaci (ISP) prostřednictvím rozhraní SPI, je nová deska vybavena dvouřadým šestipinovým

konektorem pro připojení ISP programátoru. Schéma zapojení řídicí desky je v *Příloze č. 10.*3 a obrazec desky plošného spoje v *Příloze č. 10.4*.

4.2.4 Propojení desky převodníku USB-UART a řídicí desky

Z praktických důvodů při návrhu a odladění funkce převodníku USB – UART a řídicí desky jsou tyto realizovány každá na samostatné DPS a jsou propojeny pomocí krátkého plochého desetižilového vodiče s dvouřadými konektory.

Blokové schéma vzájemného propojení s popisem čísel pinů konektorů je patrné z *Obrázku č. 4.2*.



Obrázek č. 4.2 Vzájemné propojení převodníku USB – UART a řídicí desky

Blok PC-USB představuje připojený počítač prostřednictvím USB kabelu.
Bloky VD 1 až VD 4 představují eventuálně připojené výkonové desky.

5. Programová část

5.1 Komunikace mezi aplikací a řídicí deskou

Před vlastním psaním programu pro mikrokontrolér řídicí desky a řídicí aplikace pro PC bylo nejprve potřeba navrhnout vhodný způsob komunikace mezi aplikací a řídicí deskou. Vycházel jsem z toho, že bude potřeba přenášet tyto zprávy:

- zpráva RESET – požadavek z PC na provedení resetu mikrokontroléru,
- zpráva STOP – požadavek z PC na zastavení činnosti soustavy,
- zpráva ROTATE – požadavek z PC na otáčení motorem s parametry otáčení motoru,
- zpráva STATE – požadavek z PC na odeslání stavu optozávor,
- zpráva OK – potvrzení přijetí zprávy mikrokontrolérem,
- zpráva ERROR – chyba přijetí zprávy mikrokontrolérem,
- zpráva OPTO – stav optozávor.

5.1.1 Zpráva RESET

Zpráva RESET slouží k přerušení činnosti soustavy a provedení resetu mikrokontroléru v případě chybné funkce soustavy, např. zacyklení programu, chyba komunikace apod. Zpráva není přenášena sériovou linkou, ale je realizovaná náběžnou hranou na vývodu RTS FTDI převodníku, který je propojen s vývodem PD2 mikrokontroléru a vyvolá jeho přerušení.

5.1.2 Zpráva STOP

Zpráva STOP slouží k zastavení činnosti soustavy z bezpečnostních, technických nebo jiných důvodů. Zpráva není přenášena sériovou linkou, ale je realizovaná náběžnou hranou na vývodu DTR FTDI převodníku, který je propojen s vývodem PD3 mikrokontroléru a vyvolá jeho přerušení.

5.1.3 Zpráva ROTATE

Zpráva je přenášena sériovou linkou z PC do mikrokontroléru. Slouží pro přenesení požadavku na otáčení motorem a předání parametrů podle kterých bude otáčení probíhat. Při správném přijetí zprávy následuje odpověď zprávou OK, při chybném přijetí zprávou ERROR. Po dokončení otáčení motorem následuje ještě zpráva OPTO. Ve zprávě ROTATE jsou přenášeny tyto parametry:

- hlavička,
- číslo motoru,
- směr otáčení motorem,
- způsob otáčení motorem,
- rychlost otáčení motorem,
- počet kroků / sektorů pro otočení,
- kontrolní součet všech parametrů.

5.1.3.1 Hlavička

Hlavička slouží pro počáteční identifikaci zprávy. Ve zprávě ROTATE je hlavička reprezentována znakem 'H'.

5.1.3.2 Číslo motoru

Parametr určuje, jakým motorem má mikrokontrolér otáčet. Může nabývat následujících hodnot:

0 – otáčení motorem 1
1 – otáčení motorem 2
2 – otáčení motorem 3
3 – otáčení motorem 4

5.1.3.3 Směr otáčení motorem

Parametr určuje, kterým směrem se motor bude otáčet. Hodnoty jsou:

0 - vlevo
1 - vpravo

5.1.3.4 Způsob otáčení motorem

Podle parametru se volí, zda se motorem bude otáčet po jednotlivých krocích nebo po celých sektorech. Odpovídající hodnoty jsou:

0 – po krocích
1 – po sektorech

5.1.3.5 Rychlost otáčení motorem

Parametr obsahuje celé číslo, které v milisekundách určuje dobu mezi dvěma kroky motoru. Dovolené hodnoty jsou:

10 - minimum
1000 – maximum

5.1.3.6 Počet kroků/sektorů

Parametr obsahuje celé číslo, které udává počet kroků nebo počet sektorů, v závislosti na parametru určujícím způsob otáčení, o které se bude motorem otáčet. Dovolené hodnoty jsou:

0 – minimum
2000 – maximum

5.1.3.7 Kontrolní součet

Parametr obsahuje součet všech předchozích parametrů pro kontrolu správného přijetí zprávy na straně mikrokontroléru.

5.1.3.8 Struktura zprávy

Po zhodnocení rozsahu všech parametrů, které je potřeba ve zprávě přenést, jsem dospěl k závěru, že zpráva bude tvořena sedmi byty. Struktura zprávy a uložení jednotlivých parametrů ve zprávě je patrné z *Obrázku č. 5.1.*

1. byte	2. byte			3. byte	4. byte	5. byte	6. byte	7. byte
0bHHHHHHHH	0bMMDZxxxx			0bSSSSSSSS	0bSSSSSSSS	0bSSSSSSSS	0bSSSSSSSS	0bCCCCCCCC
H	M	D	Z	Speed L	Speed H	Steps L	Steps H	CRC

Obrázek č. 5.1.1 Struktura zprávy ROTATE

H – hlavička
M – číslo motoru
D – směr otáčení motorem
Z – způsob otáčení motorem
Speed L – dolní byte rychlosti otáčení
Speed H – horní byte rychlosti otáčení
Steps L – dolní byte počtu kroků/sektorů
Steps H – horní byte počtu kroků/sektorů
CRC – dolní byte součtu všech parametrů

5.1.4 Zpráva STATE

Zpráva STATE je odeslána z PC po sériové lince jako požadavek na zjištění stavu optozávor. Zpráva sestává z jediného bytu který obsahuje znak 'O'. Odpovědí na tuto zprávu je zpráva OPTO.

5.1.5 Zpráva OK

Zpráva OK je odesílána mikrokontrolérem jako potvrzení korektního přijetí jiné zprávy. Skládá se ze dvou bytů, které obsahují znaky „OK“.

5.1.6 Zpráva ERROR

Zpráva ERROR je odesílána mikrokontrolérem po chybném zpracování jiné zprávy. Skládá se ze dvou bytů obsahujících znaky „ER“.

5.1.7 Zpráva OPTO

Zpráva OPTO je reakcí na zprávu STATE. Mikrokontrolér zjistí stav všech optozávor a tento stav odešle v jednom bytu do PC. Struktura bytu je patrná z *Obrázku č. X*.

1. byte			
0bxxxx|O4|O3|O2|O1|			
Opto 4	Opto 3	Opto 2	Opto 1

Obrázek č. 5.1.2 Struktura bytu zprávy OPTO

Opto 1 – stav optozávory 1
Opto 2 – stav optozávory 2
Opto 3 – stav optozávory 3
Opto 4 – stav optozávory 4

Logické hodnoty jsou stejné jako u signálu OPTO, který propojuje řídicí desku s výkonovou deskou, tedy:

0 – optozávora je zastíněná
1 – optozávora je volná

5.2 Program pro řídicí desku

Program pro mikrokontrolér řídicí desky je napsán v jazyce C v programovacím prostředí CodeVisionAVR C Compiler Evalution. Hlavním úkolem programu v mikrokontroléru řídicí desky je přijímat zprávy odeslané z aplikace v počítači, dekódovat je a na jejich základě provádět požadované úkony a odeslat zpět odpovídající zprávu o stavu provedení operací.

5.2.1 Činnost programu řídicí desky

1. V klidovém stavu program čeká v hlavní smyčce na přerušení od sériové linky a příjem řídicích příkazů, na přerušení pro zastavení činnosti soustavy nebo na přerušení pro provedení resetu mikrokontroléru.

1.1. Požadavek na reset mikrokontroléru vyvolá přerušení INT 0, v obsluze tohoto přerušení se zastaví činnost soustavy a pomocí nastavení a spuštění WatchDog timeru se provede reset mikrokontroléru (zatím není implementováno). Přerušení INT 0 je vyvoláno náběžnou hranou na vývodu PD2. Náběžná hrana je generována vývodem RTS převodníku USB – UART. Propojení viz. *4.2.3 Propojení desky převodníku USB-UART a řídicí desky.*

1.2. Požadavek na zastavení činnosti soustavy vyvolá přerušení mikrokontroléru INT 1, v obsluze tohoto přerušení se provedou nezbytné úkony vedoucí k co nejrychlejšímu zastavení činnosti soustavy. Přerušení INT 1 je vyvoláno náběžnou hranou na vývodu PD3. Náběžná hrana je generována vývodem DTR převodníku USB – UART. Propojení viz. *4.2.3 Propojení desky převodníku USB-UART a řídicí desky.*

1.3. Příchod dat po sériové lince vyvolá přerušení a v jeho obsluze se provede:

- příjem hlavičky dat
- pokud je hlavička „O“, odešle zpět stav optozávor a vrátí se do hlavní smyčky
- pokud je hlavička „H“, pokračuje v příjmu dat
- pokud je hlavička špatná, ukončí příjem, odešle zpět chybovou zprávu „ER“ a vrátí se do hlavní smyčky

- provedení kontrolního součtu hlavičky a přijatých dat, pokud součet odpovídá přijatému kontrolnímu součtu v posledním přijatém bytu, odešle zpět potvrzení o korektním přijetí dat „OK“, jinak odešle chybovou zprávu „ER“
- pokud jsou přijatá data v pořádku, nastaví příznak přijetí dat povolující v hlavní smyčce zpracovat přijatá data a provést požadované operace a vrátí se do hlavní smyčky

2. Byla přijata data sériovou linkou a povoleno jejich zpracování:

2.1. dekódují se přijatá data a jednotlivé parametry otáčení se uloží do struktury

2.2. podle uložených parametrů se rozhoduje, kterým motorem se bude otáčet

2.3. v metodě otáčení vybraným motorem se nastaví otáčení po jednotlivých krocích nebo po celých sektorech na discích

2.4. po dokončení otáčení motorem odešle zpět stav optozávor

Detailní popis činnosti programu je znázorněn ve vývojovém diagramu v *Příloze č. 10.5*.

5.3 Řídicí aplikace pro PC

5.3.1 Volba ovladače a programovacího jazyka

Před volbou programovacího jazyka, ve kterém bude řídicí aplikace napsána přicházely v úvahu dvě možnosti:

- použití VCP ovladačů,
- použití D2XX ovladačů.

5.3.1.1 VCP ovladače

Při nainstalování VCP ovladačů se převodník po připojení k počítači chová jako virtuální sériový port. V tomto případě na použitém programovacím jazyce vůbec nezáleží. Je možné použít libovolný jazyk, ve kterém lze naprogramovat komunikaci po sériové lince.

5.3.1.2 D2XX ovladače

V případě, že je požadována možnost plně ovládat funkce převodníku a využít všechny jeho možnosti, je potřeba použít přímé D2XX ovladače. S těmito ovladači výrobce převodníku poskytuje podporu ve formě různých knihoven, jejichž prostřednictvím je možné k obvodu přistupovat z těchto programovacích jazyků:

- Java
- C++
- C# .NET
- Visual Basic
- Delphi
- Python

- PERL
- LabView
- LabWindows/CVI

5.3.1.3 Závěr

Protože jsem při práci s převodníkem požadoval možnost jeho plného ovládání, použil jsem přímé ovladače. Po předchozích dobrých zkušenostech s programovacím prostředím Microsoft Visual Studio a jazykem C# během studia jsem se rozhodl pro tvorbu řídicí aplikace právě v tomto prostředí. Určitým omezením z hlediska použití aplikace je to, že aplikace napsaná v jazyce C# je použitelná pouze na platformě Microsoft Windows, a to od verze 98 a vyšší.

5.3.2 Struktura aplikace

Programování aplikace jsem rozdělil na několik menších samostatných komponent, které jsou umístěny na hlavním formuláři (okně) aplikace, pomocí kterého jsou mezi sebou propojeny. Každá komponenta je navržena pro plnění určitého úkolu. Aplikace sestává z následujících částí:

- třída FtCommunication
- panel ovládání soustavy
  - karta Nastavení
    - komponenta DeviceManagement
    - komponenta MotorDependences
  - karta Ovládání
    - 4x komponenta MotorContol
  - karta Přidat zařízení (zatím není využita)
  - karta Log
  - signalizace stavu soustavy
  - tlačítko STOP

    - tlačítko RESET
- panel vizualizace soustavy 1
    - 4x komponenta MotorVizualization
        - obrázek disku na hřídeli motoru
        - počítadlo pozice disku
        - signalizace stavu optozávory
- panel vizualizace soustavy 2 (zatím není využit)
- menu
- informační panel

5.3.2.1 Třída FtCommunication

Třída FtCommunication slouží k propojení celé aplikace s FTDI převodníkem a řídicí deskou. Třída využívá externí dll knihovnu FTD2XX_NET.DLL poskytovanou výrobcem obvodu, která je speciálně určena pro programovací jazyk C#. Prostřednictvím metod této knihovny může třída plně využívat všechny funkce převodníku. Ve třídě FtCommunication jsou implementovány tyto hlavní funkce poskytované ostatním komponentám aplikace:

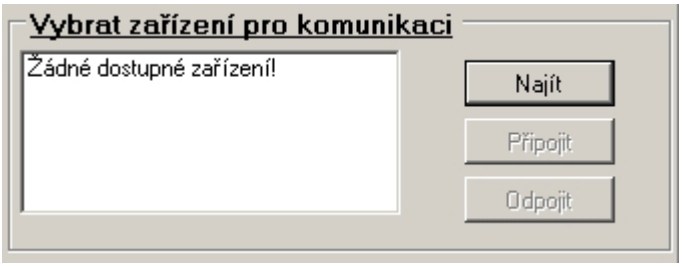
- prohledání všech USB portů počítače a vytvoření seznamu připojených FTDI zařízení a jejich parametrů (PID, VID, sériové číslo, popis zařízení)
- prohledáni všech USB portů a kontrola připojení zařízení s konkrétním sériovým číslem
- otevření spojení se zařízením podle sériového čísla
- uzavření spojení se zařízením podle sériového čísla
- odeslání zprávy ROTATE do řídící desky – požadavek na otáčení motorem podle zadaných parametrů, poté čeká na odpověď zprávou OK nebo ERROR. Po dokončení otáčení přijímá zprávu OPTO
- odeslání zprávy STATE – požadavek na zjištění stavu optozávor, poté přijímá zprávu OPTO
- odeslání zprávy STOP
- odeslání zprávy RESET

5.3.2.2 Panel ovládání soustavy

Na panelu ovládání soustavy jsou umístěny všechny potřebné prvky pro nastavování parametrů a ovládání soustavy. Ovládací prvky jsou rozděleny do skupin na samostatných kartách.

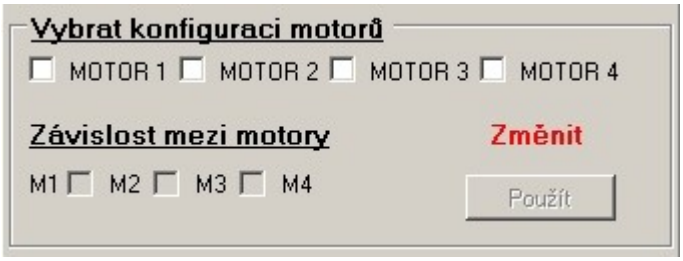
Karta Nastavení

Na kartě Nastavení je umístěna komponenta DeviceManagement, která po stisku tlačítka *Najít* vyhledá všechna dostupná FTDI zařízení připojená k počítači a vypíše jejich sériová čísla do seznamu. V případě, že je nějaké zařízení dostupné, po jeho označení je možné s tímto zařízením stiskem tlačítka *Připojit* otevřít spojení. Pokud je naopak nějaké zařízení již připojeno, umožní jeho odpojení tlačítkem *Odpojit*. Obrázek komponenty je na *Obrázku č. 5.3.1.*



Obrázek č. 5.3.1 Komponenta DeviceManagement

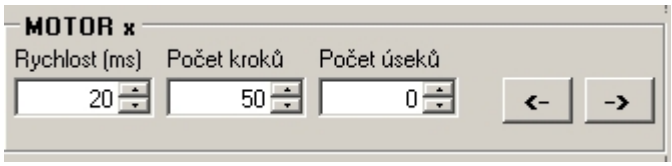
Dále je na kartě umístěna komponenta MotorDependances, na které je možné, pokud je připojeno FTDI zařízení, nastavit konfiguraci soustavy motorů – tedy vybrat které výkonové desky s motory jsou připojeny k řídicí desce. Komponenta je také připravena na možnost nastavit závislosti (vzájemné provázání) mezi disky jednotlivých motorů. Podle nastavené závislosti mezi motory a podle aktuálního stavu optozávor všech motorů by se mělo rozhodovat, kterým motorem lze otáčet a který je v pozici kdy jím nelze otočit. Na základě těchto rozhodnutí pak znemožnit uživateli otočit blokovaným motorem. Tato funkčnost však není v současné verzi programu naprogramována. Po stisku tlačítka *Použít* se aktivují ovládací prvky vybraných motorů na kartě *Ovládání.* Obrázek komponenty je na *Obrázku č. 5.3.2.*



Obrázek č. 5.3.2 Komponenta MotorDependences

Karta Ovládání

Na kartě Ovládání jsou umístěny 4 identické komponenty MotorControl, které slouží pro nezávislé ovládání jednotlivých motorů. Umožňují nastavit požadovanou rychlost otáčení (v ms mezi jednotlivými kroky) a počet kroků nebo počet sektorů o kolik se bude motorem otáčet. Po stisku tlačítek ← (vlevo) nebo → (vpravo) komponenta odesílá zprávu ROTATE s nastavenými parametry otáčení do řídicí desky. Podoba komponenty je na *Obrázku č. 5.3.3.*



Obrázek č. 5.3.3 Komponenta MotorControl

Karta Přidat zařízení

Karta Přidat zařízení není zatím využita. Na této kartě jsem chtěl uživateli aplikace umožnit měnit hodnotu sériového čísla v EEPROM vybraného zařízení, podle kterého je soustava při ovládání identifikována a připojena. Pro změnu údajů v EEPROM je tedy potřeba použít některý speciální k tomu určený program, například Ftd2xxst.exe od výrobce převodníku. Tento program je k dispozici na přiloženém CD.

Karta Log

Karta Log slouží pro výpis kontrolních zpráv, sledování průběhu komunikace a stavu aplikace.

Signalizace stavu soustavy

Signalizace stavu soustavy je prvek zobrazující aktuální stav soustavy z hlediska činnosti. Možné stavy jsou *Připraven* nebo *Pracuji*. Prvek signalizace stavu soustavy je na *Obrázku č. 5.3.4*.

Tlačítko STOP

Tlačítko *STOP* odesílá zprávu STOP do řídicí desky a zastavuje tak činnost soustavy. Styl tlačítka je na *Obrázku č. 5.3.4*.

Tlačítko RESET

Tlačítko *RESET* odesílá zprávu RESET do řídicí desky a provádí tak zastavení činnosti soustavy a reset mikrokontroléru řídicí desky. Styl tlačítka RESET je na *Obrázku č. 5.3.4*.

Obrázek č. 5.3.4 Signalizace stavu soustavy, tlačítko STOP a RESET

5.3.2.3 Panel vizualizace soustavy 1

Panel vizualizace soustavy 1 slouží pro grafické zobrazení činnosti soustavy. Pro tento účel jsou na panelu umístěny 4 totožné komponenty MotorVisualization. Komponenta se skládá z obrázku, který představuje disk s výřezy umístěný na hřídeli motoru, z počítadla absolutní pozice disku na hřídeli motoru v rozsahu 0 - 199 a signalizace stavu optozávory *Volná* nebo *Zastíněná*. Na *Obrázku č. 5.3.5* je pohled na komponentu.

Obrázek č. 5.3.5 Komponenta MotorVisualization

5.3.2.4 Panel vizualizace soustavy 2

Panel vizualizace soustavy 2 není zatím využitý. Možností by bylo umístit na tento panel například vizualizaci průběhů napětí na fázích motorů pro lepší demonstraci funkce krokového motoru.

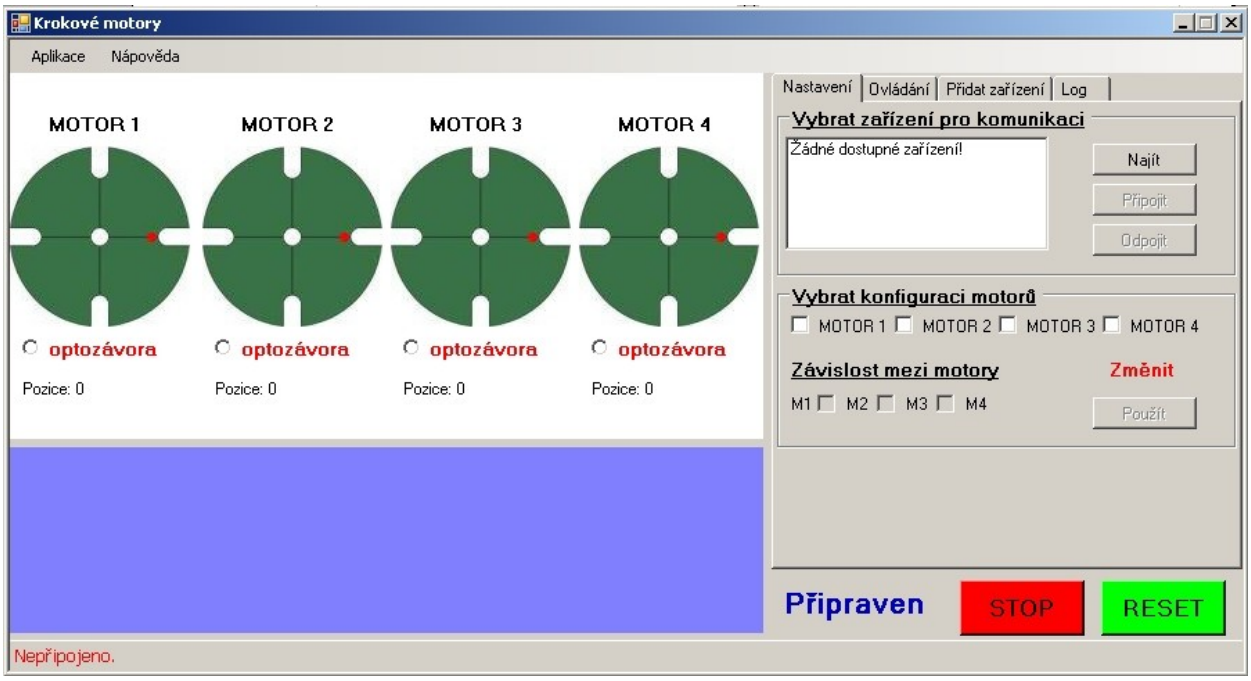
5.3.2.5 Menu

Menu aplikace sestává z položek *Aplikace* a *Nápověda.* V menu Aplikace je pouze jediná položka, a to *Konec* pro ukončení aplikace. V menu *Nápověda* jsou dvě položky: *Nápověda* a *O aplikaci.* Obě tyto položky nejsou zatím funkční.

5.3.2.6 Informační panel

V informačním panelu ve spodní části aplikace se vypisuje informace o stavu připojení aplikace k soustavě. Pokud aplikace není připojena k žádné soustavě, v panelu je vypsán nápis *Nepřipojeno*, v opačném případě je v panelu vypsán nápis *Připojeno k: 'sériové číslo zařízení'*.

5.3.2.7 Celkový pohled na hlavní okno aplikace



Obrázek č. 5.3.6 Hlavní okno aplikace

5.4 Spuštění a provoz aplikace

5.4.1 Instalace .NET Framework

Pro spuštění aplikace napsané v jazyce C# .NET na cílovém počítači je potřeba nainstalované prostředí .NET Framework, které umožňuje provoz .NET aplikací. Toto prostředí je zdarma k dispozici ke stažení na webových stránkách společnosti Microsoft. [4]

5.4.2 Instalace D2XX ovladače

Pro provoz FTDI převodníku a jeho komunikaci s řídicí aplikací je potřeba nainstalovat přímé D2XX ovladače. Poslední dostupná verze ovladačů je již kombinovanou verzí VCP a D2XX ovladačů a umožňuje na jednom PC provozovat více zařízení v režimu virtuálního sériového portu společně s přímo ovládanými zařízeními. Ovladače jsou k dispozici volně ke stažení na internetových stránkách výrobce [5] nebo na přiloženém CD.

5.4.3 Instalace aplikace

Vlastní aplikaci není třeba na cílový počítač instalovat. Pouze je potřeba umístit spustitelný soubor aplikace KrokoveMotory.exe do stejného adresáře s vygenerovanými knihovnami jednotlivých součástí aplikace. Adresář se spustitelným souborem aplikace a potřebnými knihovnami je uložen na přiloženém CD.

5.4.4 Ovládání aplikace a soustavy

Ovládání aplikace probíhá v několika základních krocích, jejich sled může být například následující:

- spuštění aplikace
- karta Nastavení
  - vyhledání zařízení
  - připojení zařízení
  - nastavení konfigurace soustavy motorů
- karta Ovládání
  - nastavení parametrů otáčení
  - ovládání motorů
- ukončení aplikace

5.4.4.1 Spuštění aplikace

Po spuštěni aplikace jsou všechny prvky vizualizace soustavy v počátečním stavu a je otevřena karta *Nastavení*.

5.4.4.2 Vyhledání zařízení

Pokud je k počítači připojena soustava, po stisku tlačítka *Najít* se v seznamu objeví její sériové číslo a aktivuje se možnost *Připojit* soustavu.

5.4.4.3 Připojení zařízení

Po označení položky v seznamu a stisku tlačítka *Připojit* se aplikace pokusí otevřít spojení se zařízením s vybraným sériovým číslem. Pokud je spojení úspěšně otevřeno a soustava s vybraným sériovým číslem je spojena s aplikací, v informačním panelu je vypsána informace

o aktivním připojení s vybraným sériovým číslem. Současně se deaktivuje možnost vybírat zařízení v seznamu a aktivuje se možnost nastavit konfiguraci soustavy motorů.

5.4.4.4 Nastavení konfigurace soustavy motorů

V nastavení konfigurace se označí ty motory, které jsou fyzicky připojeny k řídící desce. S tím, jak jsou označovány jednotlivé motory, tak se aktivuje i možnost nastavit závislost mezi těmito motory, pokud jsou sousední. Nastavení konfigurace soustavy je potřeba věnovat zvýšenou pozornost. Pokud je v aplikaci nastavená odlišná konfigurace soustavy než je skutečný stav, může dojít k poškození soustavy! Po stisku tlačítka *Použít* konfiguraci je uživatel dotázán zda chce opravdu použít nastavenou konfiguraci. Po kladném potvrzení dotazu se aktivují ovládací prvky vybraných motorů na kartě *Ovládání*.

5.4.4.5 Nastavení parametrů otáčení

Na kartě *Ovládání* jsou podle nastavené konfigurace aktivovány ovládací prvky vybraných motorů. Na ovladači motoru lze nastavit rychlost otáčení a počet kroků nebo počet sektorů o které se má motor otočit. Pokud je nastavený nenulový počet kroků, počet sektorů se automaticky vynuluje a opačně.

5.4.4.6 Ovládání motorů

Stiskem tlačítka otáčení vpravo nebo vlevo se spustí otáčení motoru a dokud otáčení neskončí, aplikace na další požadavky na otáčení motory nereaguje. V případě potřeby je otáčení motorem možné zastavit stiskem tlačítka STOP.

5.4.4.7 Ukončení aplikace

Ukončení aplikace je možné provést v libovolném okamžiku přes menu tlačítkem *Konec* nebo standardním způsobem křížkem v pravém horním rohu aplikace. Pokud je při zavírání k aplikaci připojena soustava, uživatel je dotázán zda chce opravdu aplikaci ukončit. Po kladném

potvrzení zprávy dojde k zastavení činnosti soustavy, odpojení připojeného zařízení a ukončení aplikace.

6. Závěr

Úkolem mé práce bylo navrhnout a realizovat vhodné komunikační rozhraní pro připojení již zhotovené demonstrační soustavy s krokovými motory k počítači.

Před samotným návrhem komunikačního rozhraní jsem si sestavil podrobný popis celé soustavy. Pro návrh vhodného rozhraní jsem potřeboval především podrobně prozkoumat a popsat strukturu soustavy, elektrické a logické úrovně propojení jednotlivých součástí a fyzikální možnosti a omezení soustavy. Podle sestaveného popisu soustavy jsem poté navrhl a krátce popsal několik možných variant konstrukce komunikačního rozhraní. Po zhodnocení možností využití jednotlivých variant jsem nakonec jako nejvhodnější pro tento účel zvolil variantu konstrukce komunikačního rozhraní na bázi technologie USB s integrovaným převodníkem USB-UART.

Protože původní řídicí deska nebyla připravena na možnost připojení nějakého komunikačního rozhraní, musel jsem navrhnout a vyrobit řídicí desku novou. Při návrhu této desky jsem musel dodržet kompatibilitu s původní deskou a také výkonovými deskami. Zároveň jsem však desku navrhl s určitými vylepšeními oproti předchozí verzi. Především jsem rozšířil počet konektorů pro připojení výkonových desek a také umístil konektor pro připojení programátoru. Mikrokontrolér tak není nutné před programováním vyjímat z desky, což bylo velice nepraktické.

Řídicí aplikaci pro ovládání soustavy jsem se po zhodnocení různých kritérií rozhodl napsat v jazyce C# v programovacím prostředí Microsoft Visual Studio 2008. Hlavním důvodem pro tuto volbu bylo především uživatelsky velice přívětivé a zároveň výkonné programovací prostředí a také to, že jsem se s tímto jazykem a prostředím dobře seznámil při studiu předmětu Programovací jazyky pro řízení na Katedře řídicí techniky v předchozím semestru.

Při programování a ladění funkce řídicí aplikace jsem bohužel narážel na řadu problémů, které byly způsobeny nespolehlivou konstrukcí soustavy, především výkonových desek a na nich umístěných optozávor. Řešením těchto problémů jsem bohužel strávil dost času, který bych potřeboval pro dokončení a zajištění spolehlivé činnosti některých funkcí aplikace.

7. Použitá literatura a zdroje

[1] NEKVASIL, V., Ovládání krokových motorů – didaktická pomůcka, Bakalářská práce, ČVUT v Praze, FEL, 2008.

[2] Webové stránky výrobce integrovaných obvodů [online], *Future Technology Devices International Ltd.*,2009 [cit. 24. 5. 2009].
Dostupný z WWW: <http://ftdichip.com>.

[3] Katalogový list integrovaného obvodu FT232BM [online], *Future Technology Devices International Ltd.*,2009 [cit. 1. 6. 2009].
Dostupný z WWW: <http://ftdichip.com/Documents/DataSheets/DS_FT232BM.pdf>.

[4] Prostředí .NET Framework, Mirosoft.
Dostupný z WWW: <http://www.microsoft.com/NET/>.

[5] Přímé ovladače pro integrovaný obvod FT232BM [online], *Future Technology Devices International Ltd*.
Dostupný z WWW: <http://ftdichip.com/Drivers/D2XX.htm>.

[6] MATOUŠEK, D., USB prakticky s obvody FTDI - 1. díl, Nakladatelství BEN – technická literatura, Praha 2003, ISBN 80-7300-103-9.

[7] NAGEL, Ch., EVJEN, B., GLYNN, J., WATSON, K., SKINNER, M., JONES, A., C# 2005 Programujeme profesionálně, Nakladatelství Computer Press, a.s. 2006, ISBN 80-251-1181-4.

[8] VÁŇA, V., Mikrokontroléry ATMEL AVR - programování v jazyce C, Nakladatelství BEN – technická literatura, Praha 2003, ISBN 80-7300-102-0.

8. Použitý software

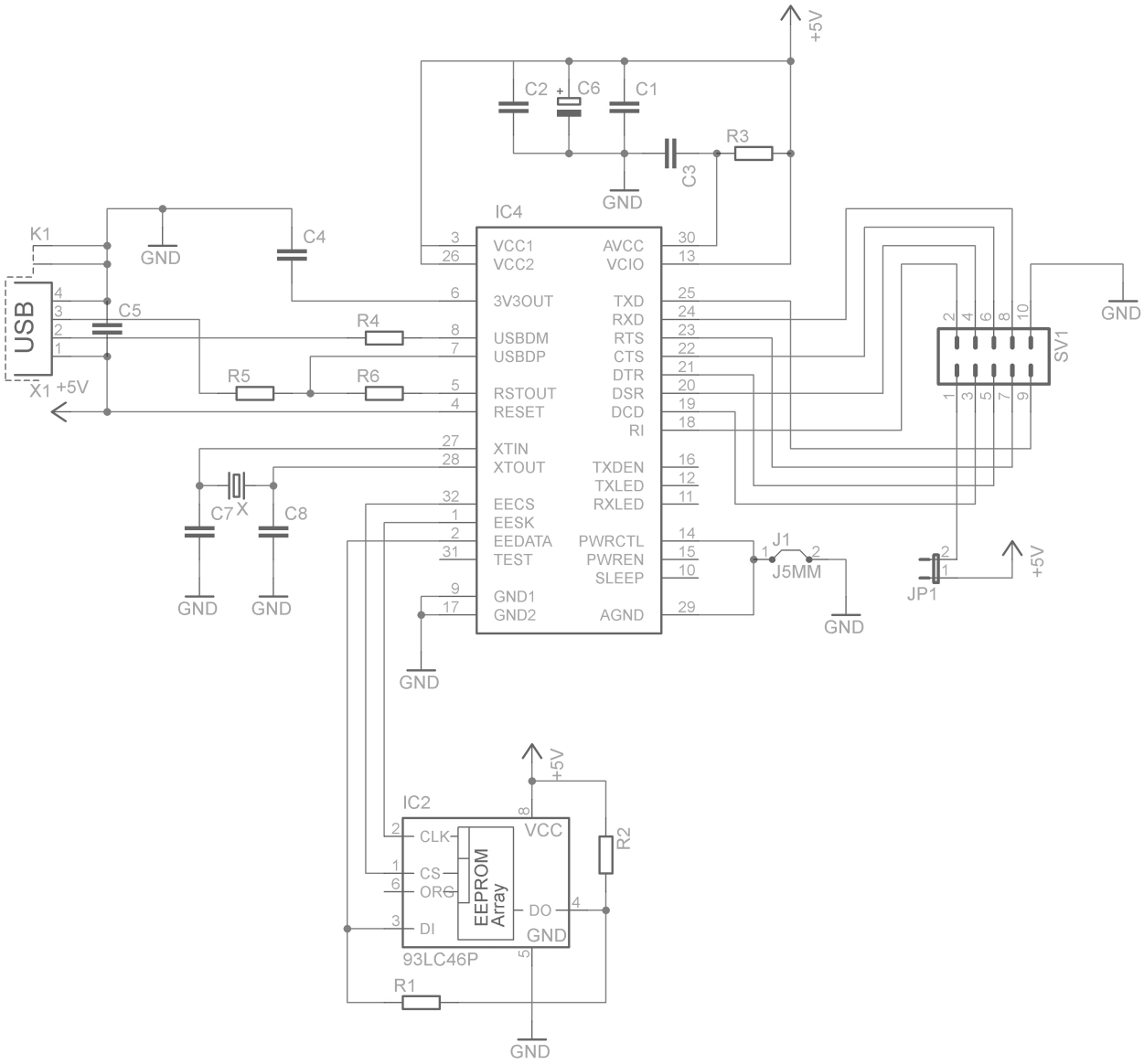
- CodeVisionAVR C Compiler Evalution
- Ftd2xxst.exe
- Microsoft Visual Studio 2008
- Eagle Layout Editor 5.0.0
- ASIX UP

9. Obsah přiloženého CD

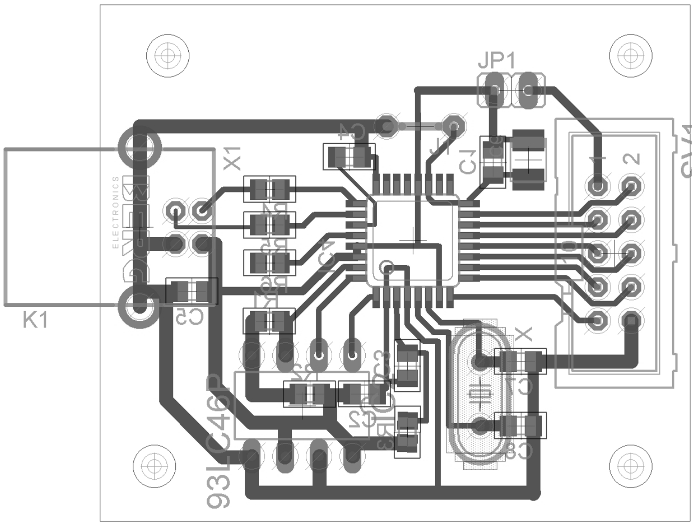
- elektronická verze této bakalářské práce ve formátu .PDF
- katalogový list integrovaného obvodu FT232BM ve formátu .PDF
- schéma zapojení desky převodníku USB – UART ve formátu Eagle .sch
- obrazec plošného spoje desky převodníku USB – UART ve formátu Eagle .brd
- schéma zapojení řídicí desky ve formátu Eagle .sch
- obrazec plošného spoje řídicí desky ve formátu Eagle .brd
- zdrojový kód pro mikrokontrolér řídicí desky v jazyce C
- program Ftd2xxst.exe pro programováni sériové EEPROM
- kombinované VCP a D2XX ovladače pro FTDI převodník
- knihovna FTD2XX_NET.DLL
- adresář s kompletním projektem řídicí aplikace pro Microsoft Visual Studio 2008 a zdrojové kódy v jazyce C#
- adresář se spustitelným souborem řídicí aplikace KrokoveMotory.exe spolu s potřebnými knihovnami

10. Přílohy

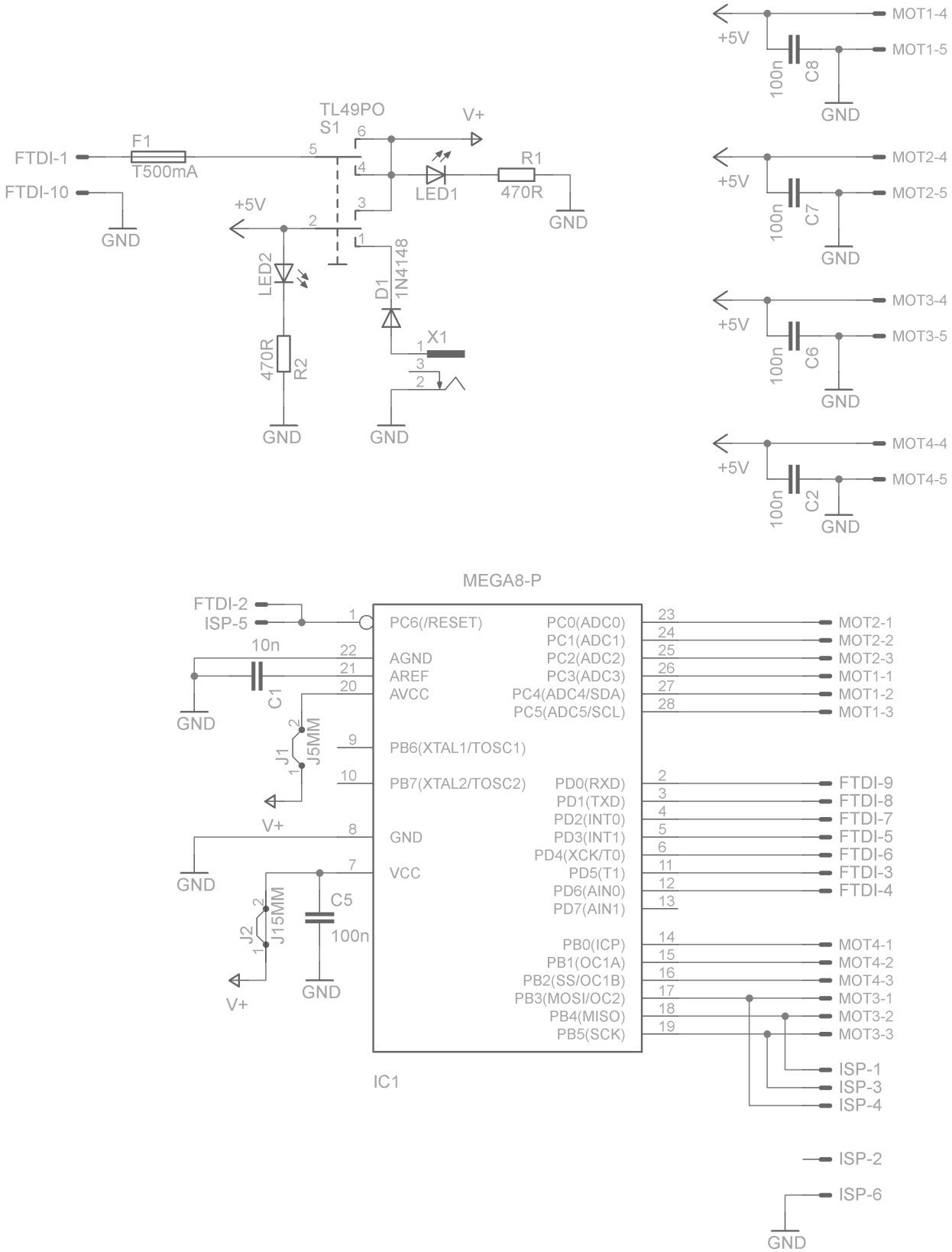
10.1 Schéma zapojení desky převodníku USB – UART



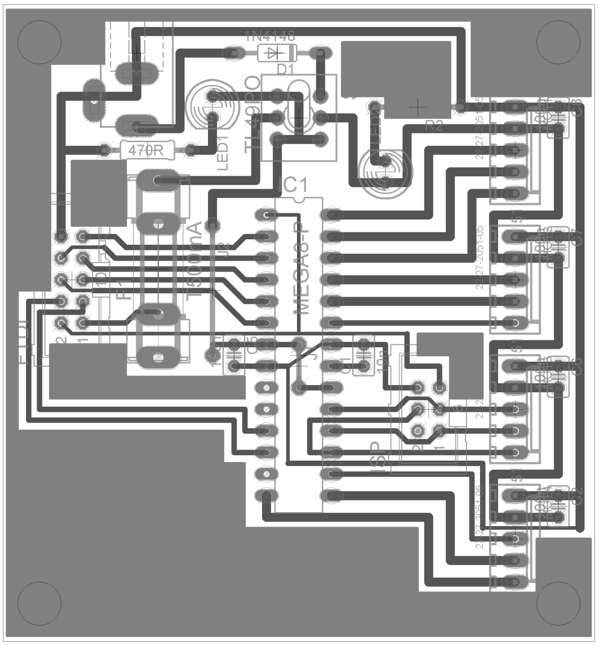
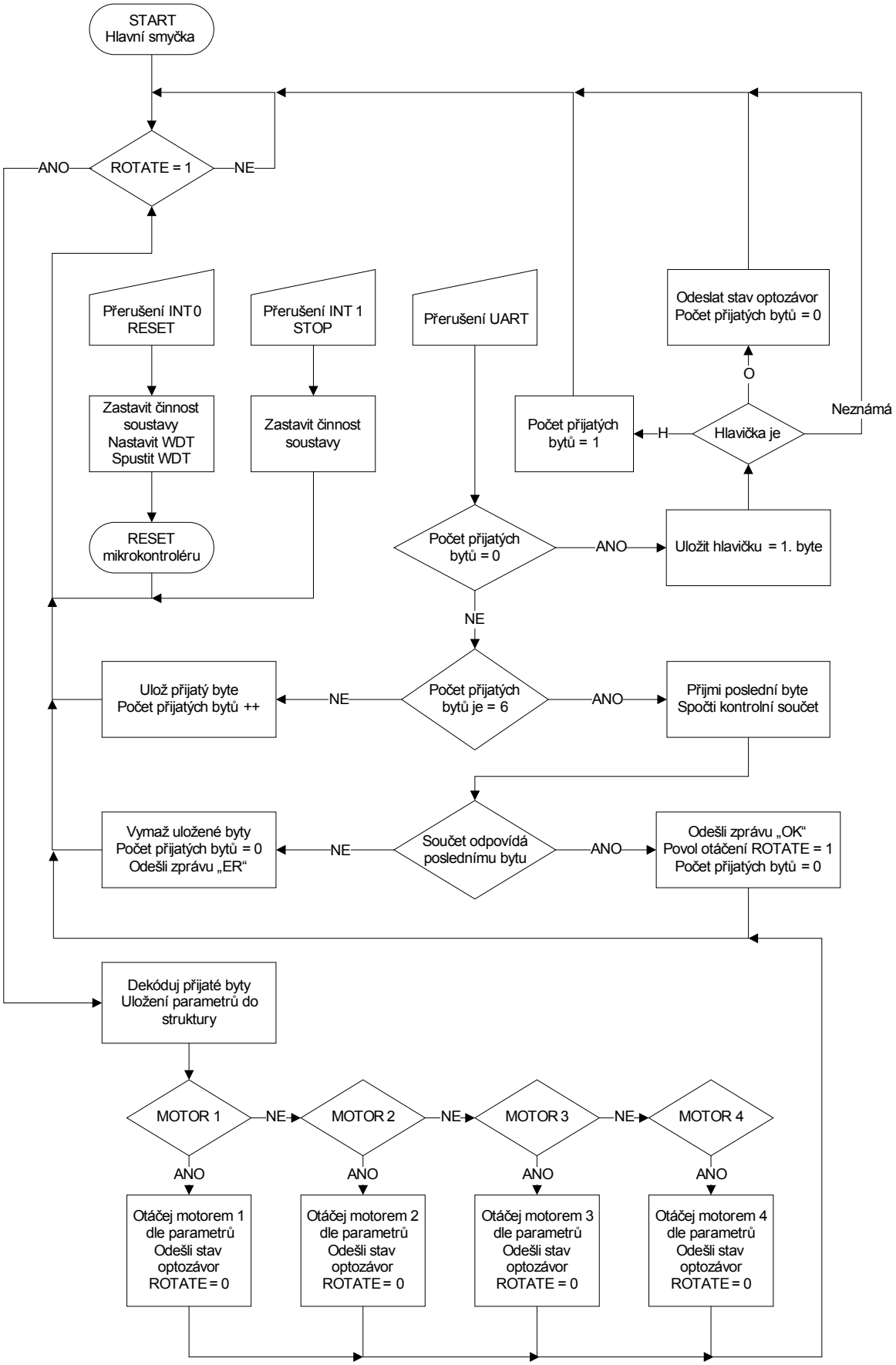
10.2 Obrazec desky plošného spoje desky převodníku USB - UART



10.3 Schéma zapojení řídicí desky



10.4 Obrazec desky plošného spoje řídicí desky



10.5 Vývojový diagram funkce programu pro řídicí desku