



Střední průmyslová škola a Vyšší odborná škola, Písek

Karla Čapka 402, Písek

Školní rok: 2011/2012

Kmenový obor: 2643M Elektronika

Studijní obor: 26–43-M/004 Slaboproudá elektrotechnika

Maturitní práce

Dopravník s krokovým motorem

Téma číslo: 36

Autor: Petr Lazor

Třída: C4.S

Vedoucí práce: Ing. Miroslav Paul

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

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Písku dne podpis:

Poděkování

Touto cestou bych rád poděkoval vedoucímu své dlouhodobé maturitní práce panu Ing. Miroslavu Paulovi a Milanu Lazorovi za pomoc s výrobou mechanických částí.

Anotace

Tato maturitní práce se zabývá řízením pásového dopravníku programovatelným logickým obvodem (PLC). Dopravník je poháněn unipolárním krokovým motorem, který ovládá mikroprocesor Atmega8 a je jednou z nejdůležitějších součástí celé práce, protože zároveň umožňuje komunikaci s PLC přes sériovou linku. Cílem je především ověření obecných znalostí z elektrotechniky, stejně tak znalostí mikroprocesorové techniky a programování v jazyce C a nakonec, použití vývojového prostředí Mosaic a jazyka ST pro programování PLC.

Annotation

This graduation thesis deals with the conveyor belt control programmable logic controller (PLC). The conveyor is powered by a unipolar stepper motor that controls the microprocessor Atmega8 and is one of the most important part of the whole work, because it allows to communicate with the PLC via a serial line. Aim is to verify the general knowledge of electrical engineering, as well as knowledge of microprocessor technology and programming in C, and finally, the use of the Mosaic development environment and language ST for programming PLC.

Klíčové slova / Key words

- Pásový dopravník / Belt conveyor
- Krokový motor / Stepper motor
- Mikroprocesor / Microprocessor
- Sériová komunikace / Serial communication
- Programovatelný logický automat / Programmable Logic Controller

Obsah

Úvod	5
1. Dopravníky obecně.....	7
1.1. Druhy dopravníků.....	7
1.2. Zvolený typ dopravníku	8
1.3. Konstrukční provedení	8
2. Krokový motor.....	9
2.1. Obecně	9
2.2. Druhy krokových motorů	9
2.3. Ovládání krokového motoru	10
2.4. Metody řízení krokových motorů.....	12
2.4.1. Krokování motoru.....	13
3. Elektronika.....	13
3.1. Napájecí část	13
3.2. Komunikační část.....	14
3.2.1. Obvod MAX232	14
3.2.2. Zapojení	14
3.3. Řídící část.....	15
3.3.1. Mikroprocesor AVR ATmega8-PU	15
3.3.2. Zapojení	17
3.4. Výkonová část.....	17
4. PLC.....	18
4.1. TECOMAT Foxtrot.....	18
4.2. Propojení s dopravníkem.....	19
5. Software	19
5.1. Mikroprocesor	19
5.1.1. Obsluha přerušení	20
5.1.2. Podprogramy led() a steps()	20
5.2. PLC.....	21
6. Závěr	22

Úvod

Cílem této dlouhodobé maturitní práce bylo vytvořit dopravník poháněný stejnosměrným nebo krokovým motorem, který dokáže pomocí sériové linky komunikovat s PLC TC650 od firmy Tecomat a sloužit tak, jako školní pomůcka pro výuku automatizace. Dopravník by měl být zakomponován do robotizovaného pracoviště v laboratoři automatizace.

Pro účely robotizovaného pracoviště je nejvhodnější pásový dopravník, který je konstrukčně jednoduchý a splňuje požadavky na potřebný úkon, a to přepravu kostek od podavače k vakuové ruce, která následně kostky odebírá.

Mechanická část by měla být nastavitelná a to zejména hřídele, které pohání pás. Bez tohoto prvku by mohl pás sjíždět k jedné straně a zaseknout se. Poté by docházelo k velké zátěži motoru, což by zapříčinilo závadu nebo úplné zničení motoru.

Pásky jsou běžně vybaveny stolcem umístěným pod pásem, protože pás sám o sobě není tak pevný a dochází k jeho prohybu, to by způsobilo značné potíže vakuové ruce, která by nedokázala kostku uchopit. Stolek může být tvořen za sebou jdoucími kolečky nebo hřídelkami. To je ale mechanicky náročnější a vhodné spíše tam, kde se přepravují předměty o větší hmotnosti, každá hřídelka by tak měla být vybavena ložisky pro zmenšení tření a tudíž zvýšení účinnosti dopravníku. Dalším způsobem je jednolitá deska, po které pás klouže. Dochází zde ale k většímu tření než u předcházejícího případu, pro tuto maturitní práci by ale tento způsob postačil, protože přepravovaný předmět má malou hmotnost.

Jako pohon je možné využít stejnosměrného motoru nebo krokového motoru. Při začátku realizace, byly k dispozici krokové motory spousty velikostí a druhů vymontované z tiskáren, skenerů apod. Rozhodnutí zvolit tento typ pohonu, ale zahrnuje více faktorů, které byli potřeba před zahájením práce řádně zvážit. Těmito důvody se budeme zabývat v kapitole 2. Krokové motory.

Pro ovládání jednotlivých cívek krokového motoru se vyrábějí specializované ovládací obvody, ale není to jediný způsob ovládání motoru. Pro tuto maturitní práci byl použit mikroprocesor AVR Atmega8 od firmy Atmel, s kterým mám dobrou zkušenost

z předešlých vlastních projektů. Program pro tento mikroprocesor je tvořen v programovacím jazyce C. Tento jazyk upřednostňuji před nižším programovacím jazykem jako je např. Assembler, především kvůli jeho přehlednosti a jednoduchosti. Nevýhodou může být jeho následná kompilace, která se poté může projevit v časování. U tohoto projektu jsou tyto časové nepřesnosti zanedbatelné.

Komunikace bude realizována podle zadání sériovou linkou, kterou disponuje použité PLC. Na výběr zde máme, dva komunikační protokoly a to RS 232 a RS 485. Výhodou RS 485 je například maximální délka vedení tohoto kabelu, která činí 1200 m, a to je oproti RS 232 s jeho maximální délkou vedení 20 m podstatný rozdíl. Pro potřeby řešené úlohy by stačilo RS 232 u kterého vidím jako podstatnou výhodu možnost připojení k PC, a tímto způsobem otestovat funkčnost komunikace dopravníku. Odesílání znaků ovládacího zařízení může být provedeno např. programem Terminal, který je určen pro odesílání zadaných znaků nebo dat.

Jako řídicí prvek dopravníku slouží PLC (Programovatelný logický automat) od české firmy Tecomat. Jako vývojové prostředí pro vytváření ovládacího programu je použit software Mosaic, který je také produktem firmy Tecomat. Program je napsán v jazyce ST, ten se řídí normou IEC 61 131-3. Pro zjednodušení ovládání dopravníku může být použit funkční blok, který usnadní práci případným uživatelům tohoto zařízení.

Konstrukčně je projekt úspěšně sestaven na samostatné desce pro větší kompaktnost a pro ovládání byl zvolen novější typ PLC FOXTROT, který má podstatně lepší dokumentaci a technickou podporu.

1. Dopravníky obecně

Manipulace a doprava jakéhokoli předmětu je nedílnou součástí všech odvětví průmyslu a technologických procesů. Použitím automatizovaných zařízení, tak výrazně zrychlujeme již zmíněnou manipulaci a dopravu, a také zvyšujeme bezpečnost, protože pro manipulaci není zapotřebí lidského faktoru.

Dopravník je zařízení pro nepřetržitý pohyb sypkého materiálu, kusového zboží nebo ucelených manipulačních jednotek. [1]

1.1. Druhy dopravníků

Pásový dopravník – tvořen především gumovým pásem podloženým válečky. Narazit na něj můžeme na pokladnách v obchodech, na letišti pro přesun zavazadel, na montážních linkách nebo pro přepravu sypkých materiálů.

Šnekový dopravník – se skládá ze šnekového šroubu, který se otáčí v trubce. Pomocí něj lze přepravovat sypké materiály jako je obilí, granule apod.

Řetězový dopravník – na řetězu jsou připevněny úchyty, na které se připevní předmět, který chceme přepravit. Využívá se především v lakovnách.

Hřeblový dopravník – v korytu se pohybuje řetěz, na kterém jsou připevněny hrabla. Ty shrabují materiál. Používá se např. v dolech pro hrubší materiály jako je uhlí apod.

Destičkový dopravník – na pohybujícím se pásu jsou připevněny destičky, na kterých se pohybuje přepravovaný předmět mezi nepohyblivým hrazením. Tento typ se používá např. pro dopravu lahví.

Korečkový dopravník – je tvořen kapsami (korečky), které jsou přichyceny na řetězu nebo pásu. Přeprava mokrých sypkých materiálů.

Válečkový dopravník – tvořen za sebou jdoucími válečky, které tvoří ucelenou dráhu. Nemusí být poháněn. Používá se pro přepravu krabic ve skladech, balící linky atd.

Kladičková trať – podobná válečkovému dopravníku, jen je tvořen kolečky – kladkami, které mají mezi sebou mezery. Na tomto dopravníku se přesunují ploché předměty (plechy, dřevěné desky, sklo).

1.2. Zvolený typ dopravníku

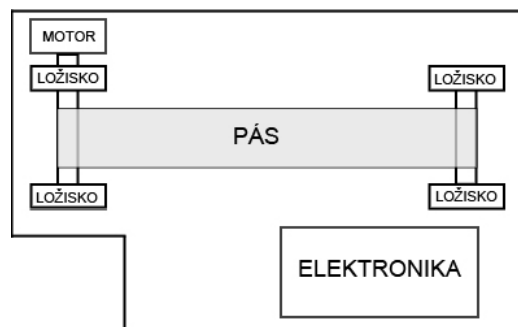
Z výše uvedených typů dopravníků byl zvolen pásový. Jeho vlastnosti se hodí pro tuto maturitní práci, a také je to jeden z nejpoužívanějších dopravníků. Ve své podstatě je velice univerzální a dá se na něm přepravovat téměř vše. Jeho využití můžeme vidět v praxi opravdu často. Jak bylo výše uvedeno, tak ho můžeme vidět např. v obchodech, na letištích, v povrchových dolech pro přepravu sypkých materiálů atd.

Pás dopravníku je vytvořen ze starého širokého škrtidla, který je spojen kvalitní, lepící páskou. Gumové škrtidlo zajišťuje, aby pás neprokluzoval na hliníkových hřídelích, které pohánějí a napínají pás.

Bližší informace o konstrukčním provedení je uvedeno v další kapitole.

1.3 Konstrukční provedení

Dopravník je umístěn na samostatné dřevěné desce o rozměrech 25 x 40 cm. Tyto malé rozměry zajišťují kompaktnost dopravníku, jako učební pomůcky.



Obr. 1 Konstrukční rozvržení dopravníku

Mezi ložiskovými domečky jsou umístěné hřídelky, na kterých je napnut gumový pás. Ložiska zajišťují malé tření a lepší běh dopravníku. Byla použita speciální, natáčecí ložiska, tím můžeme měnit úhel obou hřídelí a doladit běh pásu.

Motor je přes pevnou spojku připojen k jedné z hřídelí, tudíž lze motor také natáčet. Musí být zajištěn k desce dopravníku, aby se nezačal otáčet společně s pohyblivými částmi. Ve spoji motoru s přídržnou součástí je gumová vložka pro tlumení vibrací.

Elektronika je překryta plexisklovou deskou pro ochranu při přepravě. V desce jsou vyvrtané otvory pro odvětrávání výkonových rezistorů, které mají vysokou teplotu. Pro

větší odvod tepla od elektroniky by se dal použít menší ventilátor, ale to by bylo zapotřebí při delším provozu.

2. Krokový motor

2.1. Obecně

Princip **krokového motoru** (synchronního točivého stroje) spočívá v příchozích impulzech z elektronického ovladače (řídící elektronika, která je nutná pro běh motoru) na řídící vinutí, které udávají úhel natočení rotoru. Proto je pro tento pohon charakteristické, že pohyb hřídele motoru je **nespojité**. Je dán úhlovými skoky = **kroky**. Tyto kroky jsou dány impulzy. Jeden impulz jeden krok. Rychlost je možná řídit frekvencí impulzů.

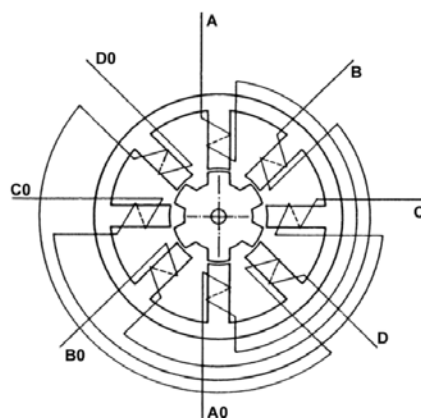
Výhodou je, že k mechanickému kontaktu a otěru dochází pouze v ložiscích. Nemusí se, proto tyto motory skoro udržovat, jsou mechanicky velice odolné a mají tudíž i dlouhou životnost.

Nevýhodou může být tzv. ztráta kroku, ke které dochází při překročení mezního zatížení. Stejně tak vysoká frekvence příchozích impulzů může vést ke ztrátě kroku, protože rotor “nestihá” zareagovat na změnu magnetického pole vytvořeného řídícím vynutím z toho vyplývá, že při vyšších rychlostech ztrácí motor kroutící moment. Poslední nevýhodou je mechanické zakmitávání při nižších rychlostech. Tyto negativa lze vyloučit správným výběrem typu motoru s příslušnými parametry.

[3] [4]

2.2. Druhy krokových motorů

Pasivní krokové motory – krokový motor s proměnnou reluktancí, na rotoru je nalisován svazek plechů (rotor nemá vinutí), stator tvoří určitý počet pólů s navinutými cívkami. Na Obr.2 můžeme vidět zjednodušený řez.



Obr. 2 Magnetický obvod čtyřfázového krokového motoru s pasivním rotorem [5]

Aktivní krokové motory – rotor je tvořen permanentním magnetem, který má po obvodu střídavě jižní a severní pól.

Hybridní krokové motory - speciální druh aktivních krokových motorů, rotor je tvořen axiálně uloženým permanentním magnetem, na jehož konci jsou umístěny feromagnetické nástavce, je tedy kombinací pasivního a aktivního krokového motoru.

Lineární krokové motory – tento stroj vykonává nespojitý, přímý pohyb. Stator je rozvinut do roviny (přímky).

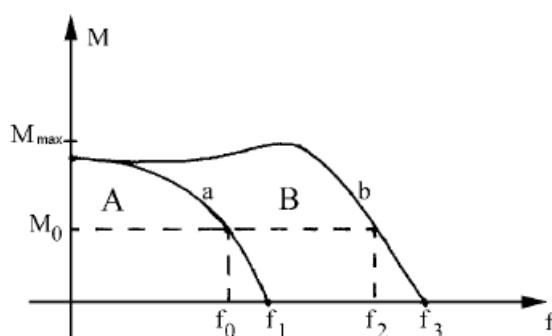
[3] [4] [5]

2.3. Ovládání krokového motoru

Elektronický ovladač krokového motoru řídí pohyb, rychlost a směr krokového motoru. Tento ovladač se skládá z rozdělovače řídicích signálů, kterému se také říká **elektronický komutátor**, ke kterému jsou přidány **výkonové spínací zesilovače** (pro každou fázi krokového motoru jeden). Dále **obvod řídicí logiku**, do kterého je přiváděn signál z **generátoru řídicího signálu** (např. astabilní klopný obvod). Druhý signál, který přivádíme, nese údaj o směru a určuje, v jakém pořadí bude spínat jednotlivé fáze.

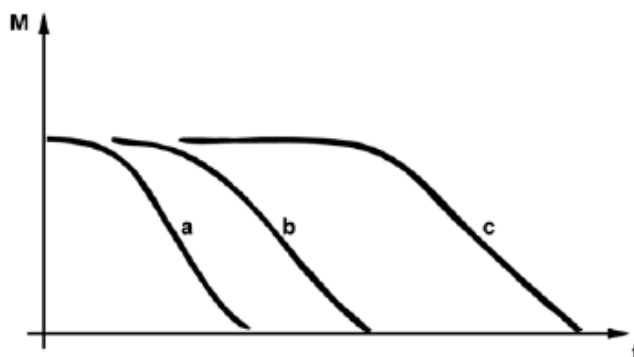
Na Obr. 3 můžeme vidět momentovou charakteristiku krokového motoru. Je rozdělena na část A, na které můžeme pozorovat, že při momentu M_0 lze skokově zvýšit frekvenci impulzů na f_0 bez ztráty kroku. Frekvence f_1 můžeme dosáhnout při nulovém zatížení motoru. V části B pozorujeme dosažení frekvence f_2 plynulým zvyšováním, nikoliv skokovým nárůstem. Frekvence f_3 je opět možná dosáhnout při nulovém zatížení. Pokud

by v části B nebylo využito plynulého zvyšování frekvence signálu, došlo by ke ztrátě kroku.



Obr. 3 Momentová charakteristika krokového motoru [5]

Ve značné míře závisí na kvalitě budících výkonových obvodů. Vinutí krokového motoru představuje indukčnost L s činným odporem R , který je v sérii. Při zapnutí a vypnutí jednotlivých fází krokového motoru probíhá přechodový děj, jehož doba je určena časovou konstantou τ , kterou vypočteme ze vztahu $\tau=L/R$. Tento děj je zanedbatelný u nízkých kmitočtů, ale s rostoucí frekvencí začíná být časová konstanta čím dál více důležitější.



Obr. 4 Průběh momentové charakteristiky v závislosti na typu buzení [5]

V praxi můžeme použít tři druhy buzení:

Buzení z napětového zdroje – tento druh buzení je znázorněný na obr. 3a, je to nejjednodušší způsob buzení, nevýhody můžeme vidět na průběhu, oproti dalším typům buzení ztrácí moment již při nízké frekvenci.

Buzení z napětového zdroje s vyšším napětím – v tomto případě je do obvodu přidán sériově odpor R_s . Po přidání tohoto odporu se sníží časová konstanta a frekvenční pásmo se rozšiřuje. To vidíme na obr. 3b. Do vztahu pro výpočet časové konstanty musí být tento

odpor zařazen následujícím způsobem $\tau = \frac{L}{R+R_s}$. Nevýhodou tohoto zapojení je vyšší energetická náročnost.

Tento typ buzení je použit pro krokový motor tohoto dopravníku. Zlepší se vlastnosti běhu motoru, získá větší sílu a konstrukční řešení zapojení je i po přidání rezistoru do série, velice jednoduché.

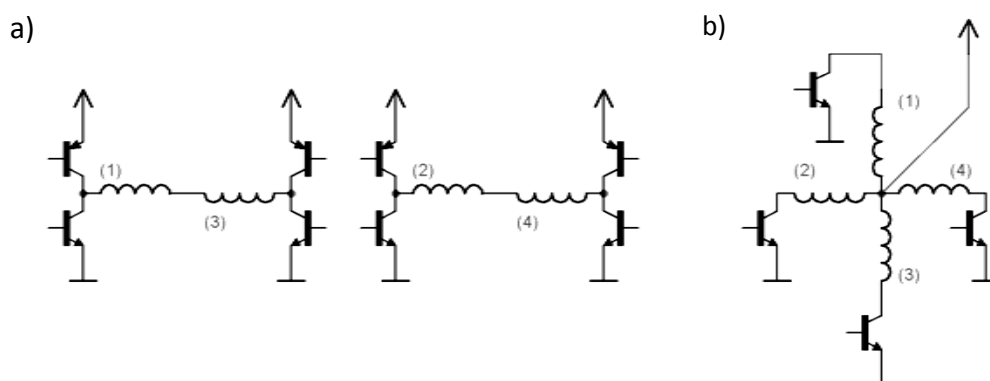
Buzení z pulzního proudového zdroje – využívá se zdroje s konstantním proudem, odpadá tak nutnost použití odporu R_s , tím pádem i energetické nároky nejsou tak velké jako v předchozím případě. U tohoto typu buzení dosahujeme nejlepších výsledku (nejnižší časové konstanty), to můžeme pozorovat na obr. 3c.

[5]

2.4. Metody řízení krokových motorů

Unipolární řízení – Výhodami tohoto zapojení je malá spotřeba a jednoduché zapojení ovládací elektroniky. Stačí pouze jeden spínací tranzistor na jednu větev. Nevýhodou je malý kroutící moment. Zapojení můžeme vidět na obr. 4 b).

Bipolární řízení – Výhodami u bipolárního řízení je velký kroutící moment, protože jsou spínány najednou dvě protilehlé cívky, tím pádem přichází nevýhoda v podobě větší spotřeby, a také je složité na zapojení, kde je potřeba dvou H můstků, to můžeme vidět na obr. 4 a).



Obr. 4 - a) Schéma zapojení bipolárního řízení, b) schéma zapojení unipolárního řízení

Jednofázové řízení – Znamená, že magnetické pole generuje pouze jedna cívka (v případě bipolárního zapojení dvě cívky). Zde je opět nevýhodou malý kroutící moment.

Dvoufázové řízení – Jsou současně sepnuty dvě sousední cívky, tím zajistíme větší kroutící moment, ale dosáhneme dvounásobné spotřeby.

2.4.1. Krokování motoru

Rozlišujeme dva způsoby krokování. S plným krokem a polovičním krokem. V prvním případě je jemnost zajištěna počtem zubů na rotoru. V případě druhém je zajištěna dvojnásobná jemnost a přesnost oproti minulému případu. Technicky jsou spouštěny střídavě dvě a jedna fáze.

[6]

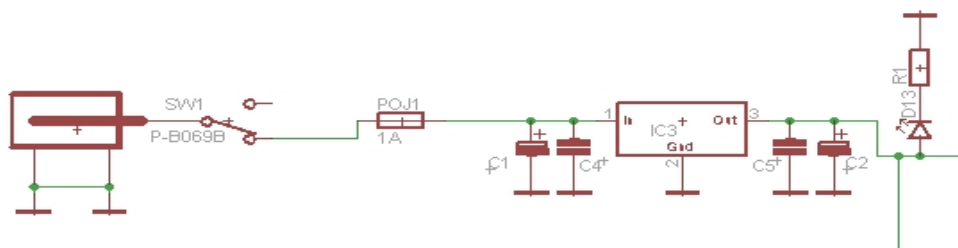
3. Elektronika

3.1. Napájecí část

Napájecí část je tvořena klasickým **stabilizátorem napětí** s označením 7805, který udržuje napětí na 5V. Tato součástka je schopna stabilizovat až z 31V při 1A. Při těchto maximálních hodnotách musí být opatřena chladičem, aby byl zajištěn dobrý odvod tepla.

Dále je opatřen dvěma keramickými a dvěma elektrolytickými kondenzátory, které můžeme vidět na Obr. 5. Tyto kondenzátory jsou umístěny po dvou (elektrolytický, keramický) před a za stabilizátorem. V obvodu mají funkci ochrany proti napěťovým výkyvům.

Vstup do stabilizátoru napětí je chráněn 1A pojistkou a napájecí obvod je opatřen LED diodou pro indikaci zapnutého stavu.



Obr. 5 -Schéma zapojení napájecí části

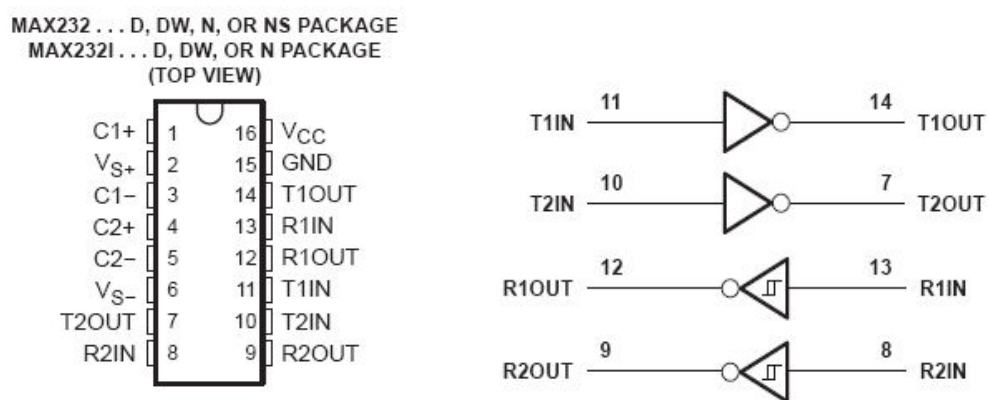
- Vstupní kondenzátor C1 – 220uF (35V), C4 – 330nF
- Výstupní kondenzátor C5 – 100nF, C2 – 220uF (35V)

- D13 - 3,3 V; 20 mA
- $R1 - R = \frac{5V - 3,3V}{0,02A} = 85 \Omega$ (91 Ω)

3.2. Komunikační část

3.2.1. Obvod MAX232

Integrovaný obvod MAX232 obsahuje dva převodníky a to z RS232 (sériové linky), tedy +12V a – 12V na úroveň TTL 0V a +5V. Jeho výhodou je, že mu stačí jedno napájení +5V. Funguje na principu nábojové pumpy, proto nám odpadá nutnost napájet obvod 12V.

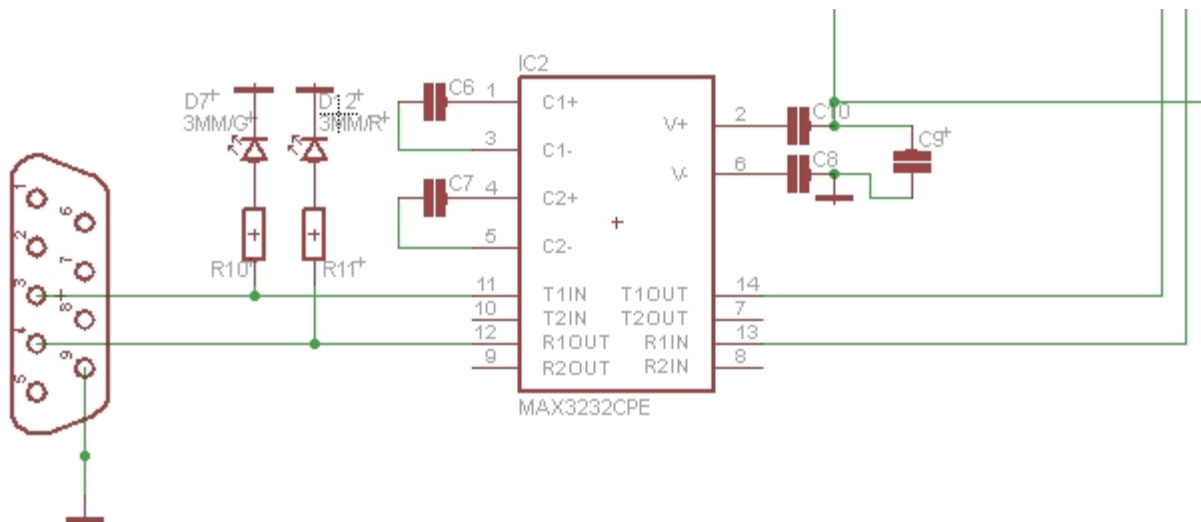


Obr. 6 - Popis vývodů obvodu MAX232 [8]

[8]

3.2.2. Zapojení

Pro vstup informací je použit konektor CAN9, kde je využito dvou vodičů (TxD, RxD) pro příjem a vysílání zprávy. Při řešení dané práce je zapotřebí pouze příjem, tudíž by stačil pouze jeden vodič a samozřejmě signálová zem, bez které by komunikace nedokázala pracovat. Na signálech jsou paralelně umístěny dvě nízko ampérové LED diody (2mA), které signalizují běh komunikace. Signál pro příjem je přiveden na vstup integrovaného obvodu MAX232, kde se zpracuje na logické úrovni TTL, které dále vedou do mikroprocesoru.



Obr. 7 – Zapojení MAX232

Aby obvod správně fungoval, jsou podle katalogového zapojení umístěny keramické kondenzátory o hodnotě 100nF, které zajišťují běh nábojové pumpy MAX232.

- D7, D12 - 1,9 V; 2 mA
- R10, R11 - $R = \frac{12V - 1,9V}{0,002A} = 5050 \Omega$ (5100 Ω)

3.3. Řídící část

Pro celou řídicí logiku byl zvolen mikroprocesor, který je vhodný pro daný případ. Zajišťuje nám spojení PLC s celým dopravníkem a zároveň zajišťuje běh krokového motoru, který by sám osobě nepracoval. Tato volba snižuje náročnost elektrického zapojení a zvyšuje flexibilitu a velkou univerzálnost celého zařízení. Řízení směru, rychlosti atd. je provedeno programově a mikroprocesor tak funguje i jako generátor impulzů potřebných pro běh krokového motoru.

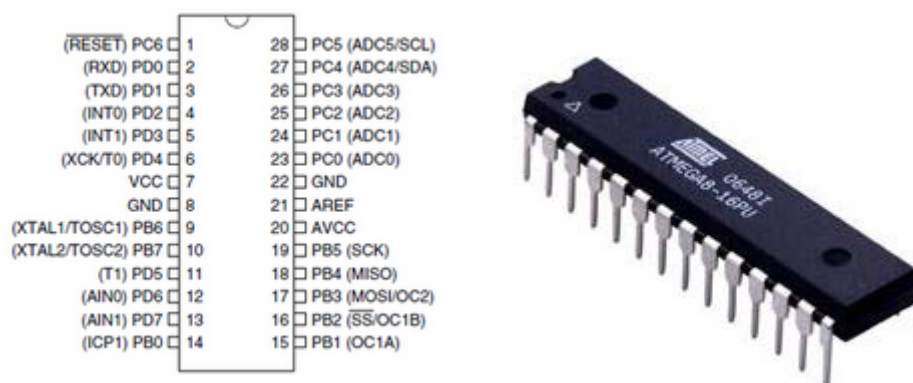
3.3.1. Mikroprocesor AVR ATmega8-PU

Mikroprocesor nebo také mikrokontrolér ATmega8 je 8bitový čip. Ve srovnání s procesory používanými ve stolních počítačích, které jsou 32bitové, je pomalý. Ale pro běh aplikací jako je třeba řízení dopravníku a jiným podobným, je naprosto postačující. V základu tento čip pracuje na frekvenci 1MHz, která se dá ovšem přetaktovat na 16MHz pro běh náročnějších aplikací. Pro tyto frekvence je ale nutné dodat externí oscilátor (krystal).

Napájení je +5V, odběr proudu je malý, při frekvenci 1MHz jsou to 3mA. Odběr proudu je závislý právě na frekvenci. Pouzdro mikroprocesoru je 28 vývodové PDIP, které je oproti klasickému zúžené.

Specifikace ATmega8:

- **8kB In-System-Programmed FLASH memory** – v této paměti je uložen samotný program, paměť se po odpojení napájení nevynuluje
- **1kB RAM** – slouží jako pomocná paměť při běhu procesoru, po odpojení napájení se paměť vymaže
- **512B EEPROM** – je volná uživatelská paměť pro ukládání libovolných uživatelem zvolených dat, tato paměť zůstává po odpojení napájení nevmazána
- 2x 8bit + 1x16bit **čítač/časovač**
- 3x **PWM** – pro generování modulovaného signálu
- 4x 10bitový **A/D**, 2x 8bitový A/D – Analogo-digitální převodníky pro měření napětí
- **USART** – umožňuje komunikaci mikroprocesoru přes sériovou linku
- vnitřní **oscilátor** 1 - 8 MHz
- **watchdog timer** – pro kontrolu správného běhu programu (proti zacyklení)



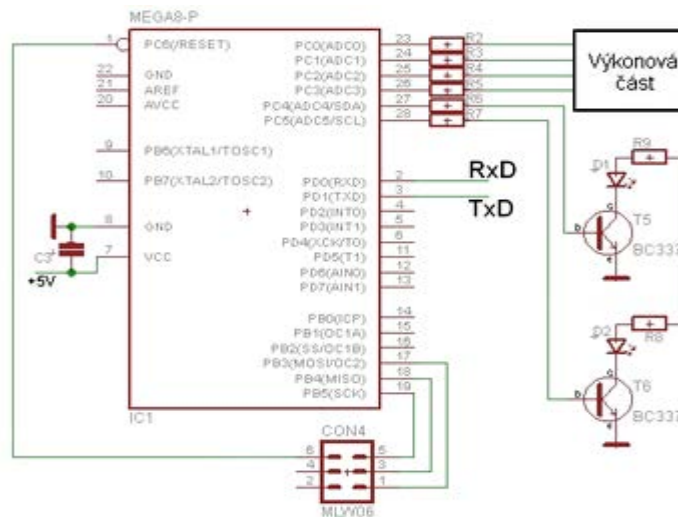
Obr. 8 – Popis vývodů, reálné pouzdro čipu [9]

Mikroprocesor obsahuje 3 porty, dva osmibitové PORTB, PORTD a jeden šestibitový PORTC. Pro ovládání krokového motoru jsou využity 4 piny portu C, zbylé dva piny jsou použity pro ovládání LED diod, které je možno ovládat také pomocí PLC.

[7]

3.3.2. Zapojení

Mezi piny mikroprocesoru pro napájení a GND je umístěn 100nF keramický kondenzátor pro filtrování případného rušení v podobě napěťových výkyvů. Na jednotlivé piny portu C jsou připojeny bázové rezistory pro spínání tranzistorů výkonové části a pro ovládání dvou LED. Na schématu je vidět i konektor pro programování mikroprocesoru, který nakonec v konečné verzi nebyl použit.



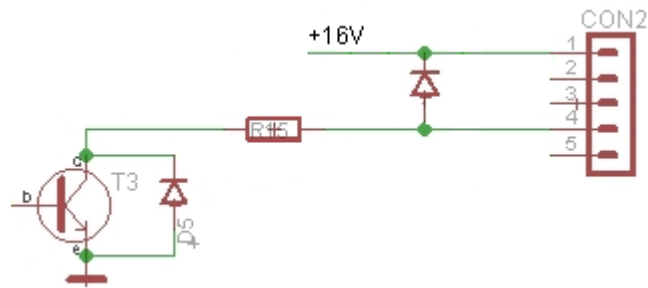
Obr. 9 – Zapojení mikroprocesoru

- T5, T6 (spínání Led D1, D2) - BC337 NPN
- R6, R7 (bázové R) – 10 k Ω
- D1, D2 - 2,2 V; 30 mA
- R8, R9 - $R = \frac{5V - 2,2V}{0,03A} = 93,3 \Omega$ (100 Ω)

3.4 Výkonová část

Do krokového motoru je přivedeno napětí +16V a jednotlivé fáze jsou spínány přes výkonové tranzistory TIP21 na zem. Před kolektorem je zapojen do série výkonový rezistor (obr. 9), to vysvětlujeme v kapitole 2.3, konkrétně **buzení z napěťového zdroje s vyšším napětím**.

Rezistor má jmenovitou hodnotu 33 Ω .



Obr. 9 – Zapojení jedné výkonové větve

Při přerušení napájení jednotlivých fází je induktivní zátěž zdrojem rušení, které vzniká vlivem doběhu motoru. Při doběhu se v cívkách podle známého zákona o elektromagnetické indukci, vytváří napětí, kterého se zbavíme použitím ochranných diod. To vidíme na Obr. 9. K cívice motoru a k tranzistoru byla tedy přidána paralelně usměrňovací dioda typu 1N4007.

4. PLC

Jedná se o průmyslový počítač s analogovými a digitálními vstupy/výstupy, který slouží pro automatizaci výrobního procesu. Program nahraný do PLC se vyznačuje tím, že běží cyklicky a proces se může vykonávat nekonečně dlouho.

K PLC je možné přidat různé doplňkové moduly v podobě dalších vstupních a výstupních bran, ovládacích panelů atd. Je možné také připojení různých snímačů a frekvenčních měničů.

4.1 TECOMAT Foxtrot

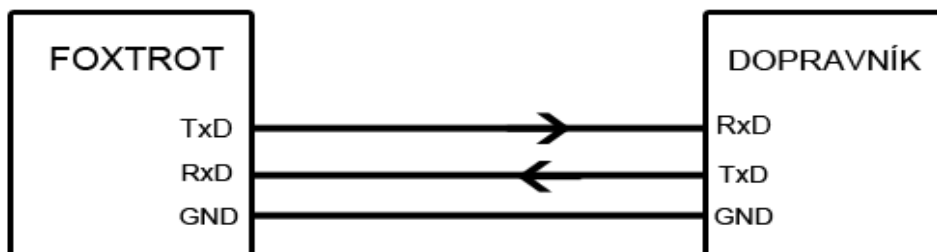
Konkrétním typem Foxtrotu, který používáme je **CP-1005**. Je to jeden ze základních modulárních programovatelných automatů řady Foxtrot.

Je vybaven šesti vstupy, každý z nich může být použit jako analogový pro měření napětí nebo proudu. Dál je vybaven dvěma analogovými 10V a šesti 24V binárními výstupy.

Pro komunikaci zde můžeme použít rozhraní ethernet a dva sériové kanály (RS-232, druhý je možné zvolit pomocí sub-modulů).

4.2 Propojení s dopravníkem

Jak už jsme se zmínili, při této komunikaci stačí pouze tři vodiče. V našem případě by stačily pouze dva, protože přijímáme znaky od PLC.



Obr. 10 – Znárodnění komunikace

5. Software

5.1. Mikroprocesor

Program pro mikroprocesor ATmega8-PU je psán v programovacím jazyce C a ve vývojovém prostředí AVR Studio 5.0 určené přímo pro programování procesorů firmy Atmel. V tomto software je možná také simulace vytvořeného programu. První částí je deklarování knihoven, proměnných a funkcí. K čemu slouží jednotlivé knihovny, je zřejmé z komentářů.

```
#include <stdio.h>           // funkce pro standardní výstup
#include <avr/io.h>          // definice periferii procesoru
#include <avr/interrupt.h>   // povolení a zakázání přerušení
#include <stdlib.h>
```

Přesunuli jsme se do samotného hlavního programu, kde je jako první nastavení všech proměnných důležitých pro chod celé aplikace. Jako první je nastavení sériové linky označované u mikroprocesoru jako USART, pro nastavení máme 3 hlavní registry. V registru **UCSRA** nijak nastavení neměníme (jednotlivé bity jsou popsány v datasheetu), v registru **UCSRB** jsme nastavily bit **RXCIE** pro povolení přerušení od přijímaného znaku a v bitu **RXEN** je nastaveno povolení příjmu. V posledním registru **UCSRB** se nastavují vlastnosti přenosu. U této aplikace bylo nastaveno **bez parity, asynchronní, dva stop bity**. Jako poslední nastavení **USART** je rychlost, kterou můžeme nastavit v registrech **UBRR1** a

UBRRH. Byla zvolena rychlost 2400 baudů (2400 bit/s), při které je menší náchylnost na rušení.

Dále nastavujeme 8 bitový čítač pro řízení motoru. Do registru **TCNT0** je načtena předvolba čítače, která se násobí rychlostí. V registru **TCCRO** je nastavena předdělička, pomocí které můžeme zastavit běh čítače nebo upravujeme rychlost čítání. Posledním nastavením čítače je registr **TIMSK**, kde bitem **TOIE0** nastavíme povolení přerušení.

Ještě nastavíme číslo kroku, od kterého začne program běžet a směr pokud by nebyl nastaven pomocí **PLC**. Funkce **sei()** zaručí globální povolení přerušení. Následuje nekončená smyčka, ve které program běží než je přerušen buď sériovou linkou nebo čítačem.

5.1.1 Obsluha přerušení

Jako první si vysvětlíme obsluhu **přerušení od čítače**. Přerušení se volá funkcí **ISR()**, ve které zvolíme od koho přerušení přijde. V obsluze je podmínka, která rozhodne o jaký směr se jedná a pomocí toho se buď proměnná **step** dekrementuje nebo inkrementuje. Při překročení osmého nebo prvního kroku (podle směru) začne dekrementace nebo inkrementace znovu. Poté se vykoná podprogram **steps()**, který bude vysvětlena níže a nastaví se znovu předvolba čítače, který byl po přerušení vynulován.

Druhá obsluha přerušení se řídí příjmem znaku od **sériové linky**. V této obsluze je řada za sebou jdoucích podmínek, které podle přijatého znaku nastaví příslušné proměnné pro rychlost, směr, start/stop a rozsvícení a zhasnutí dvou LED diod.

Při zastavení dopravníku (neboli stavu stop) je nastaven port C, kterým jsou řízeny spínací tranzistory, nastaven do nuly. Tím se rozeprnou sepnuté větve a nedochází ke zbytečnému zahřívání.








5.1.2 Podprogramy **led()** a **steps()**

Podprogram led() obsahuje dvě podmínky **if**, které rozhodnou jestli má být bit pro spínání LED diod nastaven do log. 1. Bez tohoto podprogramu by docházelo k přepisování logických stavů na portu C jednotlivými kroky, a tím by LED diody ihned zhasnuly.

Podprogram **steps()** obsahuje větvení switch, které na základě číselné hodnoty v proměnné step rozhodne jaký krok se má vykonat, potom už je jen na port C zapsán logický stav jednotlivých pinů.

5.2 PLC

Pro realizaci programu, který bude řídit dopravník, bylo zapotřebí vývojového prostředí Mosaic. Po založení nového projektu byla nutnost přidat knihovnu ComLib, ve které se nachází funkční blok pro odesílání zvolených znaků. Tento blok se nazývá **fbSendTo**. Má tři vstupní, jednu vstupně výstupní a 3 výstupní proměnné. Nutností je předem nastavit komunikační kanál CH1 na režim UNI.

VAR_INPUT			
	<i>rq</i>	BOOL	Řídící proměnná. Je-li TRUE, funkční blok vyšle data do komunikačního kanálu
	<i>chanCode</i>	UINT	Kód komunikačního kanálu <i>ETH1_uni0</i> ethernet kanál <i>ETH1</i> , spojení uni0 <i>ETH1_uni7</i> ethernet kanál <i>ETH1</i> , spojení uni7 <i>ETH2_uni0</i> ethernet kanál <i>ETH2</i> , spojení uni0 <i>ETH2_uni7</i> ethernet kanál <i>ETH2</i> , spojení uni7 <i>CH1_uni</i> sériový kanál CH1, režim uni <i>CH10_uni</i> sériový kanál CH10, režim uni
	<i>lenTx</i>	UINT	Délka vysílané zprávy (počet bytů)
VAR_IN_OUT			
	<i>data</i>	TUniBuf	Proměnná, ve které je připravena vysílaná zpráva
VAR_OUTPUT			
	<i>mesSent</i>	BOOL	Příznak odeslané zprávy Je-li TRUE vysílání zprávy bylo zahájeno
	<i>error</i>	USINT	Chybový kód. Je-li 0 (<i>COM_OK</i>), vše proběhlo bez chyby V případě chyby vrací kódy <i>COM_ERR1</i> , ..., <i>COM_ERR64</i>
	<i>lenData</i>	UINT	Délka vyslané zprávy (počet bytů)

Obr. 11 – Vstupní/výstupní proměnné [11]

Vycházíme z ukázkového programu, který je uveden v nápovědě pro funkční blok, který si vysvětlíme níže.

```
PROGRAM ExampleSendTo
VAR
SendToCH1 : fbSendTo;
txBuf : STRING[100];
END_VAR

txBuf := 'a';
SendToCH1( rq := TRUE, chanCode := CH1_uni,
lenTx := len(txBuf), data := void(txBuf) );
```

Nadeklaruje si funkční blok a proměnou txBuf jako pole, kam budeme zapisovat odesílaný znak. Dále už jen do proměnné txBuf přiřadíme zvolený znak a odešleme pomocí funkčního bloku, kde je rq nastaven do hodnoty true (povolení odesílání), chanCode je přiřazen název kanálu, v lenTx je pomocí funkce len () udán počet odesílaných znaků a do poslední vstupní proměnné je přiřazen odesílaný znak.

1. Seznam řídicích znaků

Znak	Význam
a	Start
b	Stop
0 - 1	Směr
2 - 7	Rychlost
c	LED červená zapnuta
d	LED zelená zapnuta
e	LED červená vypnutá
f	LED zelená vypnuta

[11]

6. Závěr

Při zpracování daného problému nastal problém hlavně v komunikaci. Potíž spočívala v nastavení komunikace na PLC TC650. Byla vyzkoušena opravdu velká množství řešení, probíhala konzultace s výrobcem, poslední možností bylo vyzkoušet funkční blok pro odesílání dat, který nefungoval na TC650, a tak byl po konzultaci s vedoucím práce zvoleno PLC typu Foxtrot ke kterému je daleko lepší podpora. Po zapojení fungovala komunikace na první zapojení. Těmto potížím, ale předcházely problémy s připojením dopravníku k PC pro vyzkoušení komunikace. Problém spočíval v chybě na plošném spoji, kde byl prohozen signál RxD a TxD.

Dokončení práce hodně zpomalovala nutnost spolupráce s PLC, které vlastní škola a značně tak stěžovala podmínky dokončení.

Maturitní práce byla časově náročná především, co se týče odladování chyb programu v mikroprocesoru, aby celá aplikace fungovala bezchybně. Výsledkem je jednoduchý a funkčně správný program.

Momentálně je dopravník umístěn na samostatné desce pro lepší manipulaci, není problémem ale přesunutí na robotizované pracoviště.

Práce mě obohatila o znalosti, co se týče elektrotechniky obecně, především při řešení zapojení a rozšířila moje vědomosti o mikroprocesorech a jejich programování. Kromě již zmíněných problémů s komunikací, fungovala elektronika na první pokus, protože byla předtím odladěna na nepájivém poli, což mnohonásobně ulehčilo celkovou práci a případné problémy s předěláváním desky plošných spojů.

Celou práci bych shrnul jako úspěšnou. Nyní je připravena pro účely výuky.

Použitá literatura

[1] Dopravník. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 24. 7. 2011 [cit. 2012-03-24]. Dostupné z:

<http://cs.wikipedia.org/wiki/Dopravn%C3%ADk>

[2] DOPRAVNÍKY. *Strojírenství pro střední školy* [online]. [cit. 2012-03-24]. Dostupné z:

<http://strojirenstvi-ucivo.blogspot.com/2011/04/292-dalsi-dopravniky.html>

[3] CHLEBNÝ, Jan. *Automatizace a automatizační technika*. 4. [kniha], aktualiz. vyd. Brno: Computer Press, 2009, 296 s. ISBN 978-80-251-2523-6 (BROŽ.).

[4] Krokový motor. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 22. 8. 2011 [cit. 2012-03-24]. Dostupné z:

http://cs.wikipedia.org/wiki/Krokov%C3%BD_motor

[5] ING. RYDLO, Pavel. *KROKOVÉ MOTORY A JEJICH ŘÍZENÍ: Studijní texty* [online]. Liberec, 2000. Dostupné z: <http://www.mti.tul.cz/files/ats/krok2.pdf>

[6] ŘEZÁČ, Kamil. Krokové motory. *Robotika.cz* [online]. 10.28. 2002 [cit. 2012-03-24].

Dostupné z: <http://robotika.cz/articles/steppers/cs>

[7] Programujeme jednočipy. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 12. 1. 2012 [cit. 2012-03-24]. Dostupné z:

http://cs.wikibooks.org/wiki/Programujeme_jedno%C4%8Dipy

[8] TEXAS INSTRUMENTS. *MAX232* [online]. 2002. Dostupné z:

<http://www.datasheetcatalog.org/datasheet/texasinstruments/max232.pdf>

[9] ATMEL CORPORATION. ALL. *ATmega8 datasheet* [online]. 2011. Dostupné z:

<http://www.atmel.com/Images/doc2486.pdf>

[10] TECO A.S. *TECOMAT FOXTROT PROGRAMOVATELNÉ AUTOMATY: CP-1005* [online]. leden 2009. Dostupné z:

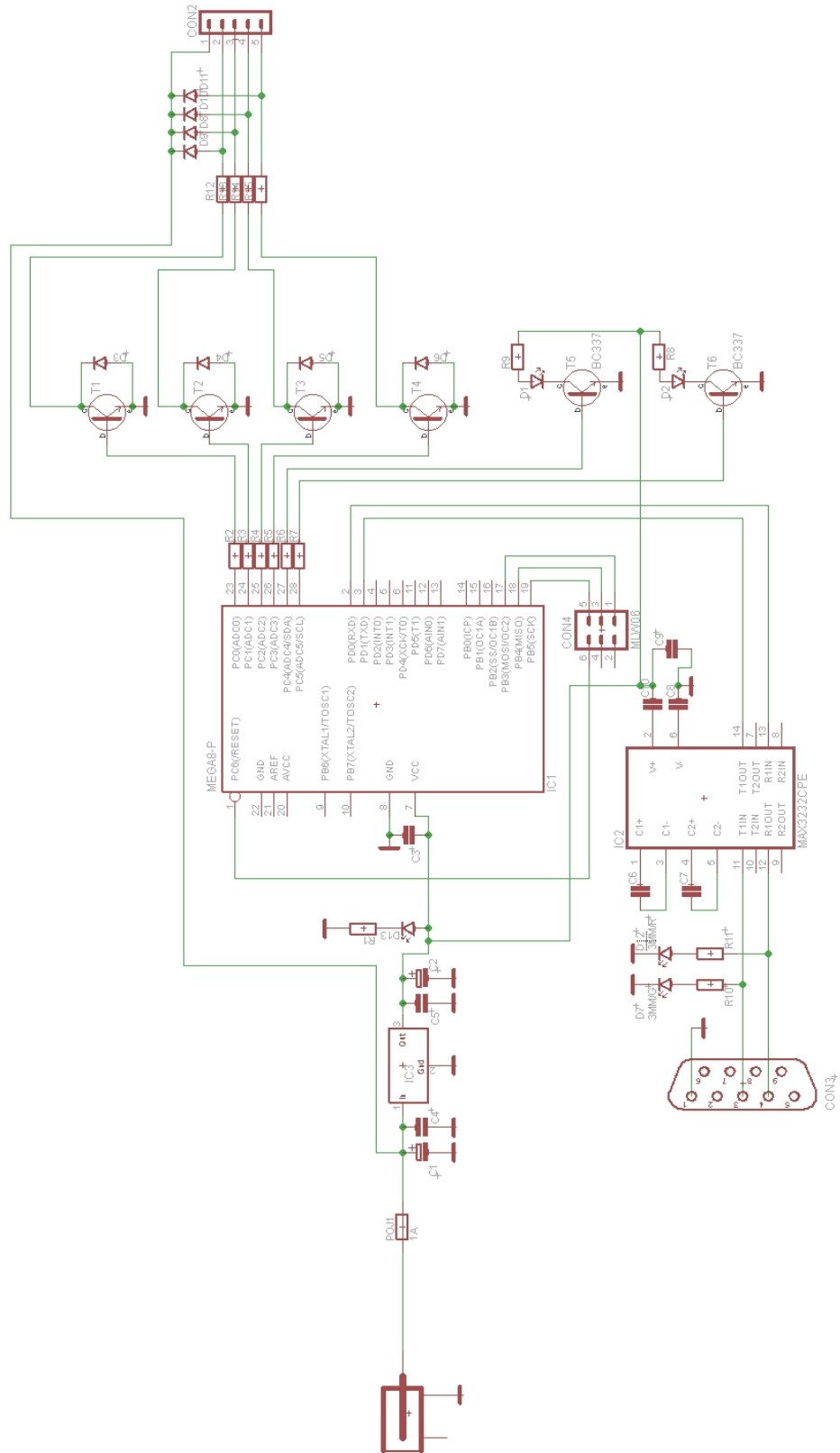
http://www.tecomat.com/wpimages/other/DOCS/cze/TXV11005_00_Foxtrot_CP-1005_cz_en.pdf

[11] TECO A.S. *Mosaic nápověda: 6.2 Funkční blok fbSendTo*.

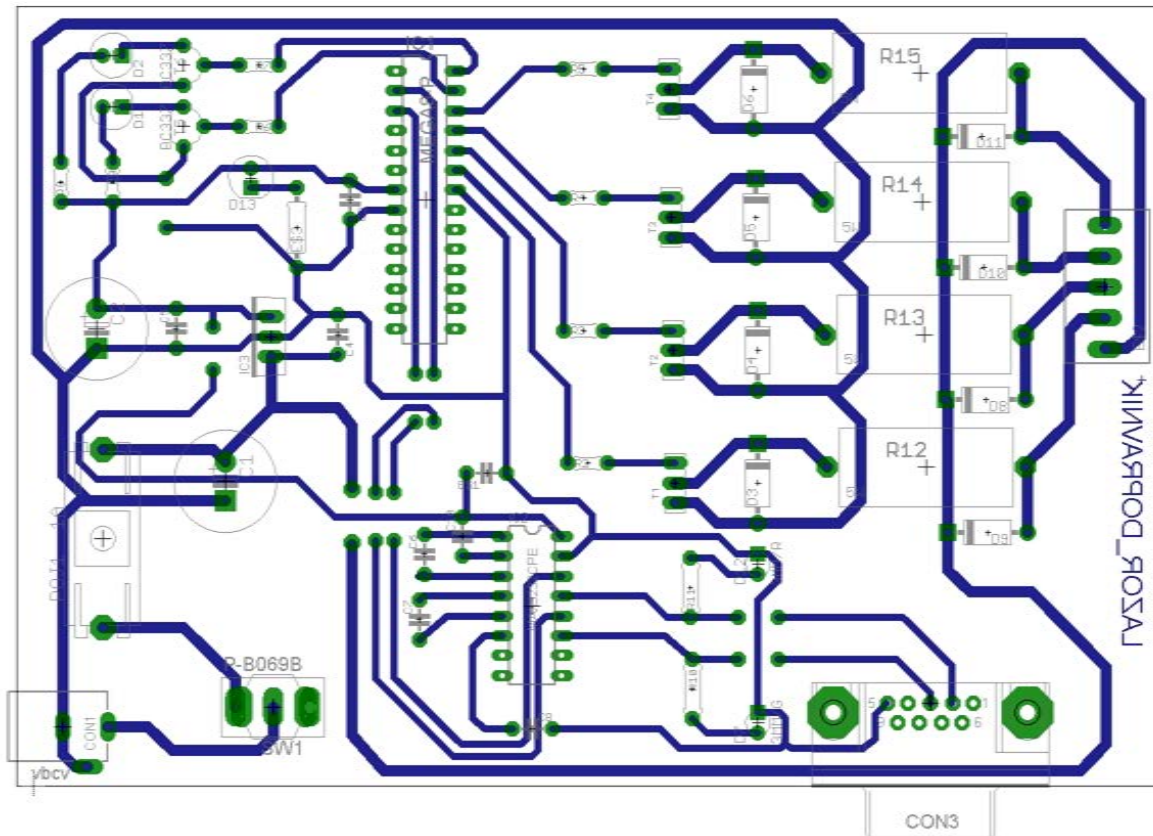
Seznam příloh

- Schéma, tištěný spoj, rozložení součástek, výpis programu mikroprocesoru, seznam součástek, DVD (program pro mikroprocesor, program PLC, schéma v Eagle)

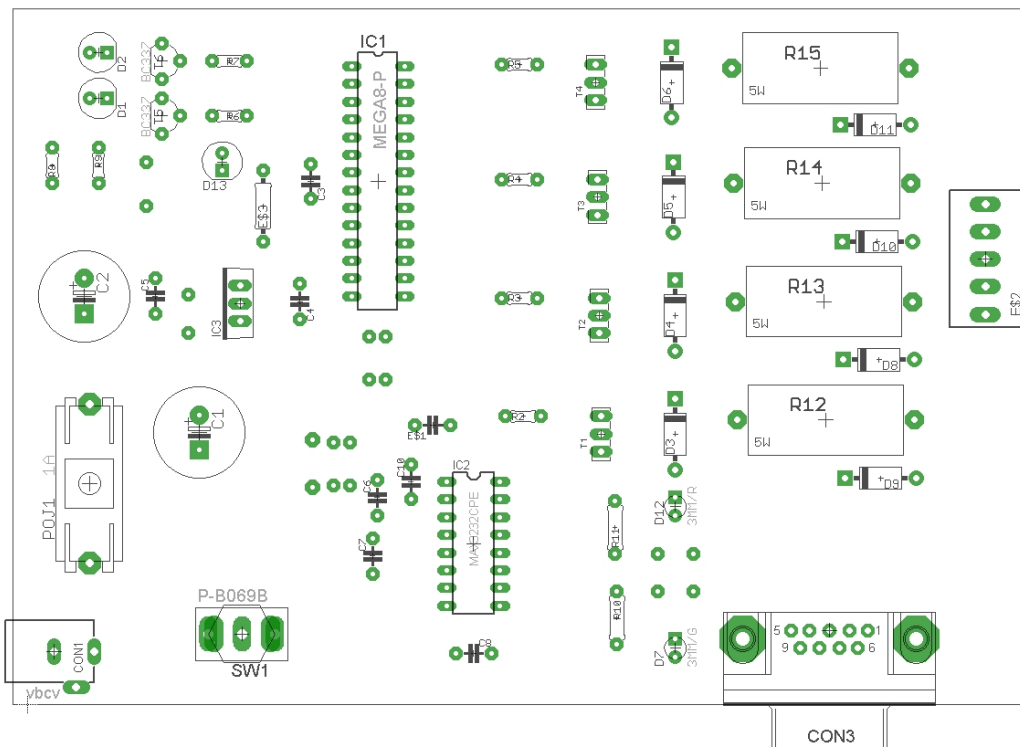
Schéma



Tištěný spoj



Rozmístění součástek



Program pro mikroprocesor

```
/*
 * AVRGCC1.c
 *
 * Created: 14.12.2011 17:46:15
 * Author: Petr Lazor
 */

#include <stdio.h>           // funkce pro standardní výstup
#include <avr/io.h>          // definice periferii procesoru
#include <avr/interrupt.h>   // povolení a zakázání přerušení
#include <stdlib.h>

char step;
volatile char led_red;
volatile char led_green;
volatile char smer;
volatile char speed;
volatile char r_byte;

void steps (void);
void led (void);

int main(void)
{
    DDRC=0x3F;              // nastavení výstupních pinů
    UCSRA=0x00;             // nastavení USART (sériové linky)
    UCSRB=0x90;
    UCSRC = 0x8E;
    UBRR1=0x19;

    TCNT0= 0x20 * speed;    // nastavení předvolby čítače
    TCCR0 = 0x00;           // nastavení předděličky čítače
    TIMSK = (1<<TOIE0);    // povolení přerušení pro čítač 0
    step=1;                 // krok, který se má provést
    smer=0;                 // směr otáčení
    sei();                  // globální povolení přerušení
    while(1) {}            // nekonečná smyčka
}

ISR(TIMER0_OVF_vect) {     // obsluha přerušení čítače 0
    if (smer==0) {         // kontroluje jakým směrem poběží
        if (step==8) step=1;
        else
            ++step; }      // inkrementace kroku
    else {
        if (step==1) step=8;
        else
            --step; }     // dekrementace kroku
    steps();
    TCNT0= (0x20 * speed); // znovu nastaví předvolbu čítače
}

ISR(USART_RXC_vect){      // přerušení od sériové linky
    r_byte = UDR;
    if(r_byte == 0x30) smer = 0;           // smer
    if(r_byte == 0x31) smer = 1;
    if(r_byte == 0x65) led_red = 0;        // led
    if(r_byte == 0x64) led_green = 1;
    if(r_byte == 0x63) led_red = 1;
}
```

```

if(r_byte == 0x66) led_green = 0;
if(r_byte > 0x31 & r_byte < 0x40) speed = r_byte; // rychlost
if(r_byte == 0x61) TCCR0 = 0x03; // start
if(r_byte == 0x62) { // stop
    TCCR0 = 0x00;
    PORTC = 0x00;
} }

void led(void) { // funkce pro led
    if (led_green == 1) {
        PORTC |= 1<<PINC5;
    } else {
        PORTC |= 0<<PINC5;
    }
    if (led_red == 1) {
        PORTC |= 1<<PINC4;
    } else {
        PORTC |= 0<<PINC4;
    }
}

void steps(void) { // kroky
    switch(step){
        case 1: { PORTC= 0x08;
            led();
            break; }
        case 2: { PORTC= 0x0C;
            led();
            break; }
        case 3: { PORTC= 0x04;
            led();
            break; }
        case 4: { PORTC= 0x06;
            led();
            break; }
        case 5: { PORTC= 0x02;
            led();
            break; }
        case 6: { PORTC= 0x03;
            led();
            break; }
        case 7: { PORTC= 0x01;
            led();
            break; }
        case 8: { PORTC= 0x09;
            led();
            break; }
    }
}

```

Seznam součástek

Označení	Název	č. GM	Cena	Počet
CON1	Napájecí souosý konektor WEALTHMETAL DS-241B	806-303	4,50 Kč	1
CON2	Konektor se zámkem PSH04-05PG	800-199	2,60 Kč	1
CON3	Konektor CAN 9 V 90	801-037	9,60 Kč	1
CON4	MLW06G	800-083	9 Kč	1
POJ1	FSF01	633-017	3,50 Kč	3
IC1	ATMEGA8A-PU	432-027	60 Kč	1
IC2	MAX232IN	329-036	10 Kč	1
IC3	7805	330-001	7 Kč	1
C1,C2	E220M/35V	123-140	4 Kč	2
C2	E47M/16V	123-478	0,16 Kč	1
C3-C5	CK 100N/50V	120-106	2 Kč	1
C6-C10	CK 1N0/100V2	120-069	2,50 Kč	5
R1	R W0185 91R	110-234	2 Kč	1
R2-R5	MRR 5K1	119-217	2 Kč	4
R6,R7	MRR 10K	119-124	2 Kč	2
R8,R9	MRR 100R	119-057	2 Kč	2
R10,R11	MRR 5K1	119-217	2 Kč	2
R12-R15	RR W5-33R	114-030	5 Kč	4
D1	LED 5MM RED 80MCD	511-467	1,50 Kč	1
D2	LED 5MM GREEN	511-060	1,60 Kč	1
D3-D6,D8-D11	1N4007	220-002	1,10 Kč	8
D7	LED 3MM 2MA/Y	511-202	1,50 Kč	1
D12	LED 3MM 2MA/G	511-200	1,50 Kč	1
D13	LED 5MM BLUE 400/60° DIFF	511-749	4,80 Kč	1
T1-T4	TIP121	211-071	5,90 Kč	4
T5,T6	BC549C	210-032	1 Kč	2
PATICE_IC1	SOKL 28U	823-026	3,20 Kč	1
PATICE_IC2	SOKL 16	823-002	1,80 Kč	1
PROTI_CON2	Konektor se zámkem PFF04-05P	800-207	1,10 Kč	1
PROTI_CON4	KON-071.00	821-257	3,30 Kč	1
VYPÍNAČ	SS-12F51-G6	631-382	2,50 Kč	1
	Konektor se zámkem PFF02-01FG	800-162	0,60 Kč	8
POJ-DRZAK	SI-HA#112000	000-065	5,70 Kč	1