

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA ELEKTROMECHANIKY A VÝKONOVÉ ELEKTRONIKY

BAKALÁŘSKÁ PRÁCE

Řídící jednotka mechatronického modelu výrobní linky

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Patrik KALAJ**
Osobní číslo: **E12B0221P**
Studijní program: **B2644 Aplikovaná elektrotechnika**
Studijní obor: **Aplikovaná elektrotechnika**
Název tématu: **Řídicí jednotka mechatronického modelu výrobní linky**
Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

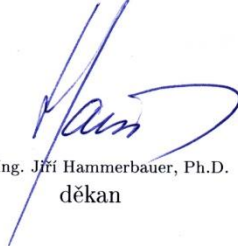
1. Navrhněte a realizujte hardware řídicí jednotky modelu výrobní linky.
2. Navrhněte vhodné řídicí algoritmy pohonů modelu.
3. Implementujte navržené algoritmy do řídicí jednotky.
4. Oživte a kalibrujte celý mechatronický model.

Rozsah grafických prací: **podle doporučení vedoucího**
Rozsah pracovní zprávy: **20 - 30 stran**
Forma zpracování bakalářské práce: **tištěná/elektronická**
Seznam odborné literatury:


Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: **Ing. Jan Štěpánek**
Regionální inovační centrum elektrotechniky

Datum zadání bakalářské práce: **15. října 2014**
Termín odevzdání bakalářské práce: **8. června 2015**


Doc. Ing. Jří Hammerbauer, Ph.D.
děkan




Prof. Ing. Václav Kús, CSc.
vedoucí katedry

V Plzni dne 15. října 2014

Abstrakt

Předmětem této bakalářské práce je návrh a realizace hardware řídicí jednotky modelu výrobní linky, návrh vhodných řídicích algoritmů pohonů modelu a oživení celého mechatronického modelu. Model se skládá z pásového dopravníku, zásobníku na barevné míčky s míchačkou a dávkovačem a šachty, která slouží k rozpoznávání barev. Cílem práce bylo přepracovat model linky tak, aby se dal snadno spustit a pracoval bez zásahu zvenčí. K ovládání modelu linky se použil mikrokontrolér ATmega32A, který je programován v jazyce C. Dále linka byla dovybavena novými senzory na detekci krabičky na pásu.

Klíčová slova

Řídicí jednotka, řídicí algoritmus, ATmega32, mikrokontrolér, vývojový kit EvB 4.3 v4, PWM, servomotor, LCD displej, senzor překážky.

Abstract

The subject of this bachelor thesis is the design and implementation of control unit hardware into the model of production line, the design of appropriate control algorithms of drives and revive the whole mechatronic model. The model consists of a belt conveyor, tray colored balls with a dispenser and mixer and shaft which is used for color recognition. The aim was to revise the model line so that it is easy to start and it works without outside interference. The microcontroller ATmega32 was used to control the model line that is programmed in C. In addition, the line was retrofitted with new sensors to detect the box on the belt.

Key words

Control unit, the control algorithm, ATmega32, microcontroller, evolutionary kit EVB 4.3 v4, PWM servo motor, LCD display, sensor barriers.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software použitý při řešení této bakalářské práce je legální.

.....

podpis

V Plzni dne 8.6.2015

Patrik Kalaj

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Janu Štěpánkovi za cenné rady, připomínky a vstřícný přístup při zpracování této práce.

Obsah

OBSAH	8
SEZNAM SYMBOLŮ A ZKRATEK	9
ÚVOD	10
1 MODEL AUTOMATIZOVANÉ TŘÍDÍCÍ LINKY	11
1.1 POPIS LINKY.....	12
1.1.1 Pásový dopravník.....	12
1.1.2 Zásobník míčků	12
1.1.3 Dávkovač míčků.....	12
1.1.4 Detektor na rozpoznání barev.....	13
1.2 ROBOTICKÁ RUKA	14
2 ŘÍDÍCÍ JEDNOTKA	15
2.1 VÝVOJOVÝ KIT EVB 4.3 v4	15
2.1.1 LCD displej.....	17
2.1.1 Ovládání LCD displeje.....	18
2.2 TLAČÍTKA.....	18
2.3 ATMEGA 32A.....	19
3 SENZORY PŘEKÁŽEK	20
3.1 VYHODNOCOVACÍ MODUL SENZORŮ.....	20
3.2 SENZOR	21
3.3 PRINCIP.....	22
4 VÝKONOVÝ MODUL	23
4.1 VYSOKOPROUDOVÝ H-MŮSTEK BTM7752G.....	24
5 SOFTWARE	26
5.1 PWM – RYCHLÝ REŽIM.....	26
5.2 ŘÍDÍCÍ ALGORITMY	27
5.2.1 Hlavní program.....	27
5.2.2 Obsluha senzorů.....	29
5.3 ROZPOZNÁNÍ BARVY.....	29
5.4 FUNKCE STOP.....	31
ZÁVĚR	32
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	1
PŘÍLOHY	3

Seznam symbolů a zkratek

A/D	analog/digitál
C	programovací jazyk
f_{clk}	frekvence hodinového taktu mikrokontroléru (16 Mhz)
f_{vys}	výstupní frekvence PWM
LCD	displej z tekutých krystalů
LED	elektroluminiscenční dioda
N	vybraný dělicí poměr (1,8,64,256 nebo 1024) f_{clk}
PWM	pulzně šířková modulace
RGB	červená - zelená - modrá
TTL	tranzistorově tranzistorová logika

Úvod

Mechatronický model výrobní linky byl zkonstruován v roce 2008 jako diplomová práce pana Ing. Petra Beneše. Model byl zhotoven pro virtuální laboratoř Katedry elektromechaniky a výkonové elektroniky, kde měl být řízen přes internet.

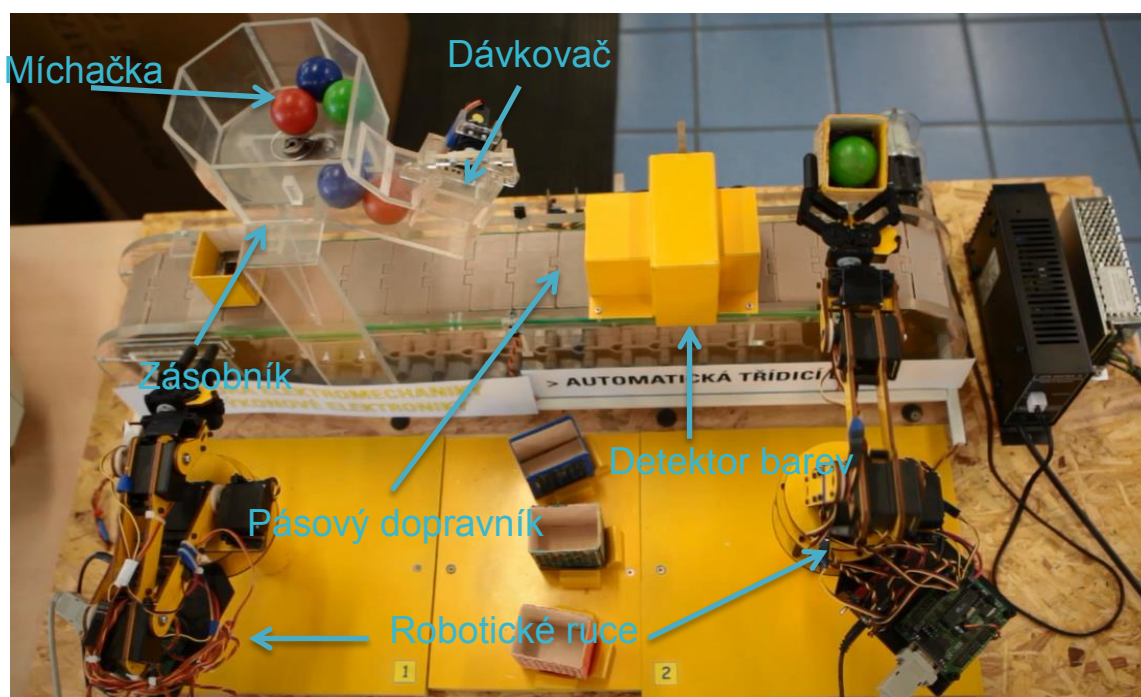
Po skončení projektu je model využíván při propagaci univerzity, fakulty a katedry při akcích jako jsou dny otevřených dveří či dny vědy a techniky. Ve výuce model nebyl využit z důvodu velmi komplikovaného spuštění linky (nutné použít stolní počítač s vhodným softwarem). Cílem bakalářské práce je upravit model linky tak, aby byl vhodný na ukázky v hodinách, což znamená snadný zásah do řídicího algoritmu a snadné spuštění linky.

Bakalářská práce se zabývá návrhem vhodné řídicí jednotky s mikrokontrolérem a sestavením řídicích algoritmů pro model automatizované třídící linky bez robotických ramen. Dále se zabývá implementováním navrženého algoritmu do řídicí jednotky a následným oživením a kalibrací celého modelu linky. Při návrhu řídicí jednotky a řídicích algoritmů může dojít k úpravě některých komponentů linky.

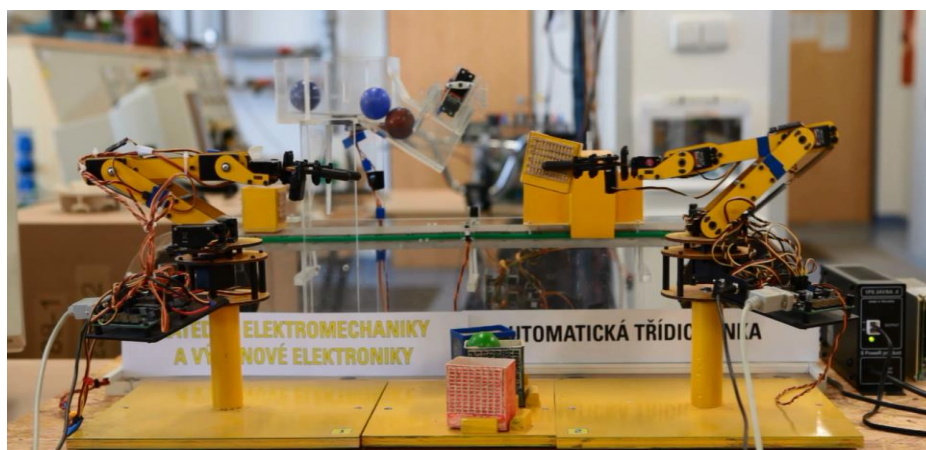
Model linky je tvořen z těchto částí: pásového dopravníku, zásobníku na barevné míčky s míchačkou, dávkovačem míčků, šachtou pro rozpoznávání barev a dvou robotických ramen. Krabičky jsou přemísťovány pásovým dopravníkem pod zásobník míčků, kde jsou tyto krabičky naplněny míčky pomocí dávkovače. Dále pás dopraví krabičku s míčkem do sekce rozpoznávání barev, kde je následně určena barva míčku. Na konci pásu je krabička s míčkem odebrána robotickou rukou nebo obsluhou linky.

1 Model automatizované třídící linky

Model slouží k ukázce postupu při automatizovaném třídění. Barevné míčky jsou promíchávané v zásobníku, aby došlo k náhodnému dávkování jednotlivých míčků. Pásový dopravník se využívá k přesunu krabiček, které jsou umístovány na pás pomocí první robotické ruky. Pod dávkovačem míčků je do krabičky umístěn barevný míček. Poté je krabička přemístěna za pomoci pásového dopravníku do části, kde dochází k rozpoznávání barvy míčku. Po rozpoznání barvy je krabička s míčkem dopravena ke druhé robotické ruce, která z krabičky vysype míček do příslušného barevného boxu. Prázdnou krabičku předá první robotické ruce a cyklus se opakuje. Model je zobrazen na obrázcích Obr. 1 a Obr. 2



Obr. 1 Model automatizované třídící linky s popisem jednotlivých částí (pohled shora)



Obr. 2 Model automatizované třídící linky (přední pohled)

1.1 Popis linky

1.1.1 Pásový dopravník

Pevné díly pásového dopravníku jsou vyrobeny z polymetylmetakrylátu (PMMA), který je běžně známý jako plexisklo nebo akrylátové sklo. Jako dopravní pás byl zvolen pás, který se v praxi využívá na přepravu lahví. Pás je uložen na dvou ocelových hřídelích, které umožňují posouvání. Nastavení napnutí pásu je řešeno pohyblivým uložením ložisek hřídele na straně, kde není dopravník připojen k pohonu. Nastavení se provádí pomocí čtyř šroubů uložených v bočnici. Jako pohon pásového dopravníku je využit stejnosměrný motor, který byl dovybaven převodovkou z důvodu dostatečného krouticího momentu. [4]

1.1.1.1 Ovládání pohonu dopravníkového pásu

Napájení a ovládání pohonu probíhá přes výkonový H-můstek BTM7752G, který je určený pro řízení stejnosměrných motorů. Můstek je řízen čítačem 1 v režimu rychlá PWM modulací a jeho výstupy jsou proti sobě invertované. H-můstek je blíže popsán v kapitole 4.1.

1.1.2 Zásobník míčků

Zásobník míčků shromažďuje míčky. Na něj plynule navazuje dávkovač míčku. Zásobník je konstruován rovněž z polymetylmetakrylátu. Součástí zásobníku je míchačka, která zajišťuje promíchání a tím i náhodné pořadí barevných míčků. Také zamezuje zaseknutí míčku v zásobníku. Míchání se provádí přes kotouč, který je na dně zásobníku a je poháněn stejnosměrným motorem s jednostupňovou převodovkou. Ovládání motoru, který zajišťuje míchání, probíhá přes tranzistor NPN. Ten je spínán PWM modulací z čítače 0. [4]

1.1.3 Dávkovač míčků

Dávkovač míčku je spojen se zásobníkem a slouží k umístění míčku do krabiček. Je také vyroben z polymetylmetakrylátu. Pod dávkovačem se nachází senzor, který detekuje krabičku pod ústím šachy. Samotné dávkování se provádí skrz šachtu se dvěma přepážkami, které se vysouvají proti sobě. Činnost přepážek zajišťuje modelářský servomotor HS-422. [4]

1.1.3.1 Ovládání dávkovače

Servomotor je řízen čítačem 2 režimu tzv. rychlé PWM s frekvencí 15,625 kHz a invertovaným výstupem (vysvětleno funkce rychlá PWM v kapitole 5.1). Do servomotoru jsou přivedeny obdélníkové pulzy s periodou 61 Hz. Nejdříve o délce 1,8633 ms – otevře se první přepážka a druhá se uzavře, míček se nachází v šachtě dávkovače. Nastavení registru čítače 2 pro délku pulzu 1,8633 ms je popsáno v tabulce Tab. 1. Poté pulz o délce 1,0923 ms – zavření první přepážky a otevření druhé, míček ze šachty spadne do připravené krabičky na pásu. Při nečinnosti je řídicí signál v logické 0.

Tab. 1 Nastavení čítače 2 tak, aby došlo k otevření šachy.

```
OCR2 = d7 = 0xe2; // nastavení registru OCR2 na hodnotu 226, pulz o délce
                  1,8633 ms
delay_ms(500); // čas po který jde do servomotoru obdélníkový pulz o délce
               1,8633 ms
OCR2 = d7 = 0xff; // nastavení logické 0 na vstup servomotoru/vystup OC2, hodnota
                  OCR2 je 255
```

1.1.4 Detektor na rozpoznání barev

Pro rozpoznání barev byla vytvořena snímací šachta. Tato šachta je vyrobena z tvrdého papíru, který zabraňuje přístupu okolního světla. Vnitřek šachty je nabarven na černo, aby nedocházelo k odrazu světla od vnitřku šachty. Šachta obsahuje RGB LED diodu a fotorezistor zapojený do napěťového děliče s 100 k Ω rezistorem. Šachta je vybavena na levém boku polymetylmakrylátovým proužkem, který má na spodní části instalovanou RGB LED diodu. Anody diody jsou připojeny na LED diodu v šachtě. Při osvětlení míčku v šachtě svítí stejná barva i v polymetylmakrylátovém proužku. Tento proužek slouží jako demonstrace, kterou barvou je osvětlen míček uvnitř šachty.

1.1.4.1 Postup při rozpoznávání barev

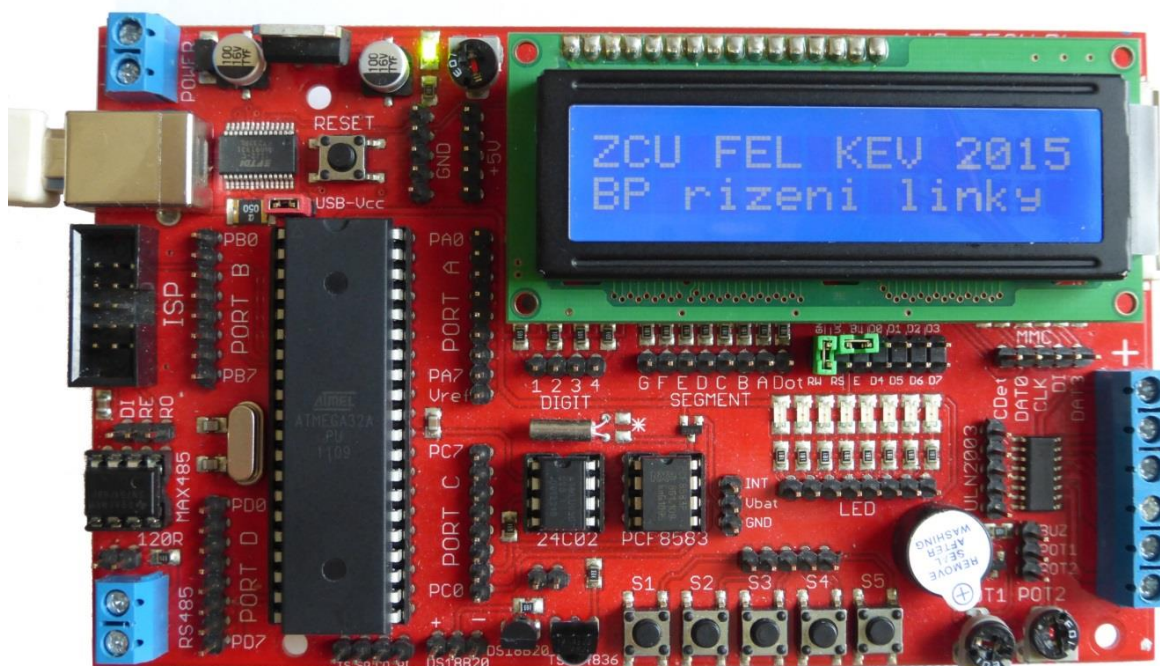
Míček se postupně osvětluje jednotlivými barvami RGB LED diody, nikdy nesvítí dvě barvy současně. Na fotorezistor dopadají odražené paprsky světla od povrchu míčku. Fotorezistor mění svůj odpor a tím i napětí na konstantním rezistoru v závislosti na množství paprsků konkrétní barvy. Napětí na konstantním rezistoru se snímá a je přivedeno do A/D převodníku mikrokontroléru. Podle hodnoty napětí algoritmus vyhodnotí, o jakou barvu se jedná.

1.2 Robotická ruka

Robotická ruka zajišťuje na začátku pásu umístění prázdné krabičky na dopravníkový pás. Na konci linky odejme druhá ruka krabičku z pásu a následně obsah krabičky vysype do připraveného barevného boxu. Prázdnou krabičku předá první robotické ruce. Pro tento účel jsou použité dvě robotické ruce LYNX 6 od firmy Lynxmotion. Jedná se o šesti-osové robotické ruce primárně určené na studijní účely. Pohon robota je zajištěn modelářskými servomotory HITEK. [4]

2 Řídící jednotka

Pro realizaci řídicí jednotky byl zvolen vývojový kit EvB 4.3 v4 (Obr. 3). K tomuto rozhodnutí došlo po konzultaci s vedoucím bakalářské práce. Kit umožňuje změnu zapojení řídicí jednotky a využití dalších periférií na desce např.: LCD displej, tlačítka, atd. Řídící jednotka navržená pro konkrétní řešení by variabilitu značně omezila. Další výhodou kitu je, že na desce se nachází programátor pro nahrání algoritmu z počítače do mikrokontroleru přes USB port a tím může dojít ke snadné změně algoritmu bez nutnosti odpojit řídicí jednotku od linky.



Obr. 3 Vývojový kit EvB 4.3 v4

2.1 Vývojový kit EvB 4.3 v4

Vývojový kit EvB 4.3 v4 je vyroben polskou společností AND-TECH. Kit je osazen mikrokontrolér ATmega32 v provedení DIL40 od společnosti Atmel. Mikrokontrolér je možné snadno z desky vyndat a případně vyměnit za jiný. Konstrukce celého kitu umožňuje flexibilní zapojení mikrokontroléru a periférie. Vstupy a výstupy jednotlivých periférií a mikrokontroléru jsou vyvedeny na piny, které se propojují pomocí drátových propojek.

Vývojový kit obsahuje tyto periferie:

- 8 LED diod
- LCD displej 2 x 16 znaků – HD44780
- napájecí piny 5x +5V a 5x GND
- 5 tlačítek
- 3x tranzistorové výstupy 500mA
- 2x tranzistorové výstupy 1A
- 2x analogové potenciometry
- buzzer
- 4x sedmissegmentový LED zobrazovač
- hodiny reálného času PCF8583
- paměť EEPROM AT24C02
- infračervený přijímač TSOP4836
- teplotní čidlo DS18B20,
- převodník sběrnic RS485
- patice pro karty MMC/SD
- USB převodník – FT232RL
- Konektor ISP

Programování mikrokontroléru může probíhat přímo na kitu, jelikož kit obsahuje integrovaný programátor. Komunikaci s počítačem umožňuje vestavěné USB rozhraní realizované převodníkem USB / UART FT232RL (virtuální sériový port – COM). [5]

Napájení kitu je řešeno přes USB port (spojeny piny USB-Vcc) nebo externím napájecím zdrojem o minimální hodnotě 9 V (rozpojeny piny USB-Vcc). [5]

2.1.1 LCD displej



Obr. 4 LCD displej 2x16 HD44870

Nejvíce využívanou periférií na kitu je LCD displej (Obr. 4). Vypisuje se na něj činnost linky a výsledek z rozpoznávání barev. Jedná se o dvouřádkový displej se 16 znaky na 1 řádek. Displej je ovládán řadičem HD44780 od firmy Hitachi. V řadiči se nachází znaková sada se základními abecedně-číselnými znaky, která je doplněná o řadu speciálních znaků či písmen. Nemusí se řídit rozsvícení jednotlivých pixelů, ale stačí napsat znak a řadič automaticky rozsvítí dané pixely v matici 5×8.

Celkem má displej 16 pinů: 8 pinů (DB0 – DB7), které slouží jako datová sběrnice, pin RS, který určuje, zda se jedná o data nebo instrukce pro řadič, pin R/W, který rozlišuje mezi zápisem dat do displeje a čtením dat z displeje a pin E, který řídí vzorkování řídicích i datových pinů. Dále piny VCC a GND, Pin VO, který slouží pro nastavení kontrastu displeje (hodnotou napětí mezi VCC a GND) a piny LED+ a LED- pro podsvícení displeje. Popis pinů je v tabulce Tab. 2.

Pokud je pin RS v logické 1, tak jsou přenášeny znaky k zobrazení, nebo k uložení do paměti. Ve stavu kdy RS je v logické 0 jsou přenášeny instrukce jako např. smazání displeje, apod. Pin R/W řídí směr komunikace. Pro přenášení dat do displeje se na tento pin nastaví logická 0. Ve stavu logické 1 se data přenáší z displeje do mikrokontroléru. Posledním řídicím pinem je pin E vzorkuje komunikaci sestupnou hranou signálu. Pro zajištění korektní komunikace musí být při vzorkování na ostatních pinech ustálená hodnota. K předávání dat mezi mikrokontrolérem a LCD lze použít dva režimy. Jednotlivé režimy se liší v použití počtu datových vodičů. V prvním režimu se využívají všechny datové piny. V druhém režimu

používáme pouze čtyři datové piny D4 – D7 a každý byte je rozdělen na dvě čtveřice bitů a ty jsou posílány za sebou. [10]

Tab. 2 Popis pinů LCD displeje s řadičem HD 44870

Pin	Název	Funkce
1	GND	Zem
2	Vcc	Napájení +5V
3	VO	Nastavení kontrastu
4	RS	Volba mezi instrukcí/data
5	R/W	Volba mezi zápis/čtení
6	E	Hodinový vstup
7 – 14	DB0 - DB7	Data
15	LED+	Anoda podsvícení
16	LED-	Katoda podsvícení

2.1.1 Ovládání LCD displeje

Pro ovládání displeje je využita knihovna `alcd.h` určená přímo pro tento typ. Pro přenos dat je využit čtyřbitový režim. Jsou použity poslední čtyři datové piny (DB4 - DB7). Knihovna obsahuje řadu příkazů pro ovládání displeje. Jako je například: příkaz pro vymazání displeje, nastavení pozice výpisu na displej, kde první číslo je sloupec a druhé řádek nebo příkaz pro vypsání textu v uvozovkách a mnoho dalších. Příklad použití těchto příkazů je uveden v tabulce Tab. 3.

Tab. 3 Příklad výpisu textu na displej

<code>lcd_clear();</code>	<code>//smazání displeje</code>
<code>lcd_gotoxy(0,0);</code>	<code>// nastavení prvního znaku na výpis displej</code>
<code>lcd_putsf("ZCU FEL KEV 2015BP rizeni linky");</code>	<code>// vypsání textu na displej</code>

2.2 Tlačítka

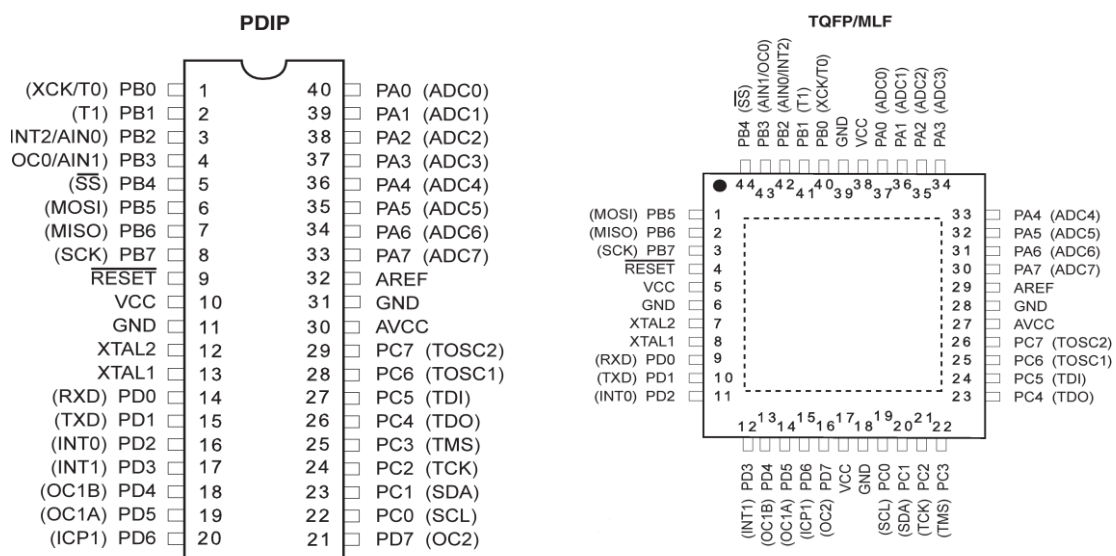
Pro komunikaci s okolním prostředím jsou využity tlačítka na vývojovém kitu. Tlačítka jsou zapojeny se společnou zemí, pro přesnější detekci musí být zapnut pull-up rezistor na vstupu mikrokontroléru. Tlačítka nejsou nijak hardwarově ošetřeny. Ošetření zákmitů je provedeno softwarově, opětovným snímáním konkrétních pinů. Princip ošetření je obdobný jako u výstupu ze senzoru. Tento princip je vysvětlen v kapitole 5.2.2.

2.3 ATmega 32A

ATmega 32A je 8 bitový mikrokontrolér s nízkým příkonem. Je založen na architektuře typu Harvard a RISC. Harvardská architektura má fyzicky oddělenou paměť programu a dat. Architektura RISC (Reduced Instruction Set Computing) označuje procesory s redukovanou instrukční sadou. Mikrokontrolér na kitu je napevno taktován externím krystalem 16MHz, ale může se využít interní RC oscilátor 8Mhz. ATmega nabízí 32 kB Flash paměti pro program, 1024 B EEPROM paměti pro ukládání dat a 2kB SRAM pro registry. Celkem obsahuje 131 výkonných instrukcí a 32 obecných registrů, které jsou přímo připojeny na ALU (aritmeticko-logickou jednotku). Je dosaženo výpočetní rychlosti 1MIPS (Milion Instruction Per Second milion instrukcí za sekundu) na 1 MHz hodinového kmitočtu. Dále ATmega obsahuje čtyři 8bitové vstupní/výstupní porty, 32 pracovních registrů, dva 8bitové čítače/časovače, jeden 16bitový čítač/časovač, 8 kanálový 10bitový A/D převodník, watchdog, čtyři PWM kanály, analogový komparátor, jednotky USART, SPI a TWI, JTAG ladění a zabudovaný RC oscilátor. [6]

Napájení mikrokontroléru se pohybuje od 2,7-5 V. Vstupní proud nesmí překročit 20 mA na jeden pin při napájení 5 V, při 3 V je to 10 mA na pin. Odebíraný proud z portu by neměl překročit hodnotu 400 mA.[6]

ATmega32A se vyrábí ve dvou typech pouzder PDIP (DIP 40) a TQFP/MLF (TQFP44). Pouzdra jsou znázorněna na obrázku Obr. 5. Rozdíl mezi těmito provedeními je, že TQFP/MLF obsahuje na každé straně vývod VCC a GND, tedy má o 4 vývody navíc.[6]



Obr. 5 Mikrokontrolér ATmega 32A, vlevo pouzdro PDIP, vpravo pouzdro TQFP/MLF [4]

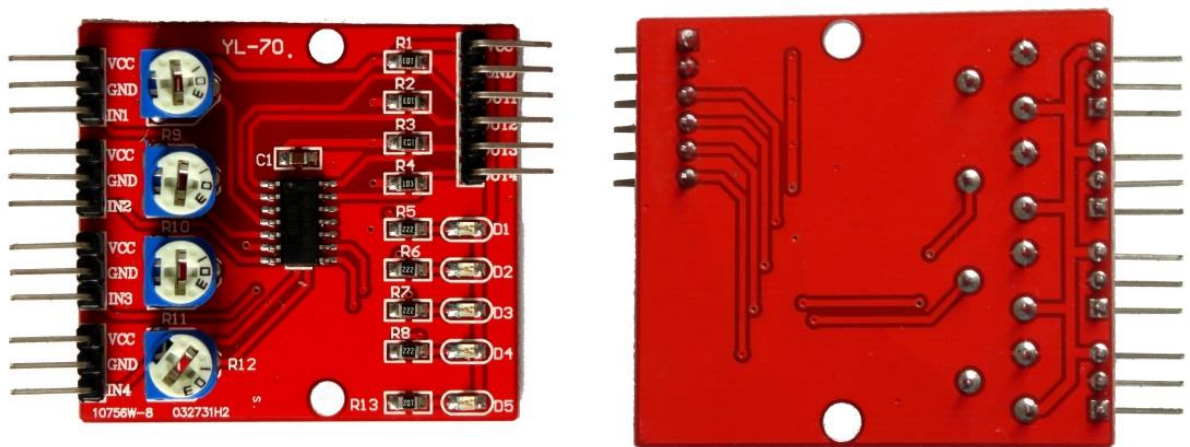
3 Senzory překážek

Na linku byly namotávány čtyři nové senzory namísto původních tří. Senzory jsou na konci linky, v šachtě na rozpoznávání barev, pod šachtou na dávkování míčků a nově i na začátku linky. Senzor na začátek linky byl instalován z důvodu, aby byla kontrola, že krabíčka byla opravdu položena na pás a nemuselo se spoléhat na informaci od robotické ruky. Rovněž lze linku nyní provozovat bez použití robotických ramen. Krabíčky lze umisťovat a odebírat z pásu ručně.

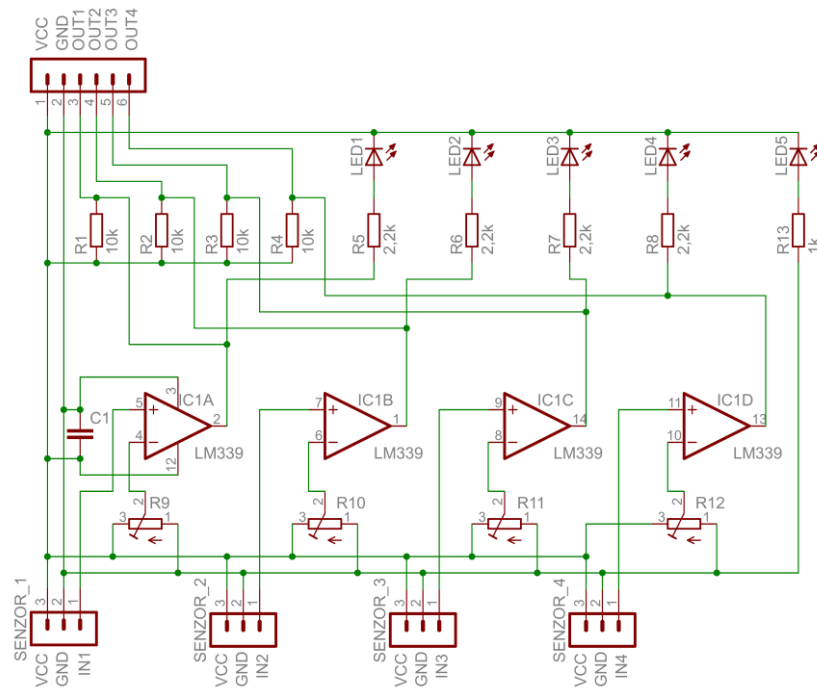
Původní senzory pracovaly na principu světelné závory. Když byl paprsek přerušen, byla detekována krabíčka. Nynější senzory pracují na principu odrazu, je-li paprsek odražen do snímače, je detekována krabíčka. Obsluha senzorů je vysvětlena v kapitole 5.2.2 Obsluha senzorů.

3.1 Vyhodnocovací modul senzorů

Vyhodnocovací modul senzorů je vyobrazen na obrázku Obr. 6 Základem modulu je integrování obvodu LM339, který obsahuje čtyři komparátory. Jeden komparátor na jeden senzor. Pro nastavení vzdálenosti, na kterou má senzor reagovat, je určený rezistorový trimer. Schéma zapojení je znázorněno na obrázku Obr. 7. Sensor dokáže reagovat na vzdálenost od 1 mm do 60 mm. Dále jsou na modulu indikační diody. Dioda D5 indikuje správně připojené napájení modulu. Diody D1, D2, D3 a D4 se rozsvítí, když senzor zaznamená překážku v nastavené vzdálenosti. Výstup z modulu je v TTL logice. Napájení se pohybuje od 3,3 V do 5 V a proud by měl být alespoň 1 A.



Obr. 6 Vyhodnocovací modul senzorů na detekci překážek



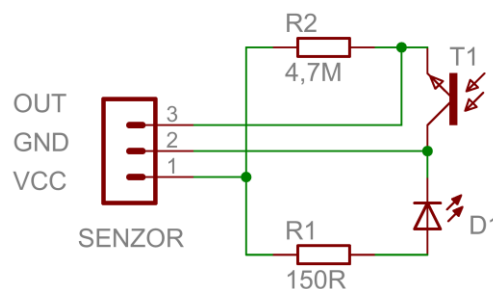
Obr. 7 Schéma zapojení modulu senzorů

3.2 Senzor

Senzor, znázorněn na obrázku Obr. 8, se skládá z infračervené LED diody (940nm) a fototranzistoru. Dioda má zapojen v sérii rezistor R1 o hodnotě 150 Ω . Emitor fototranzistoru je spojen s rezistorem R2 o hodnotě 4,7 M Ω , z tohoto uzlu je vyveden výstup senzoru. Rezistor je pak připojen na napájení, kolektor je připojen na „zem“. Schéma zapojení je znázorněno na obrázku Obr. 9.



Obr. 8 Senzor



Obr. 9 Schéma zapojení senzoru

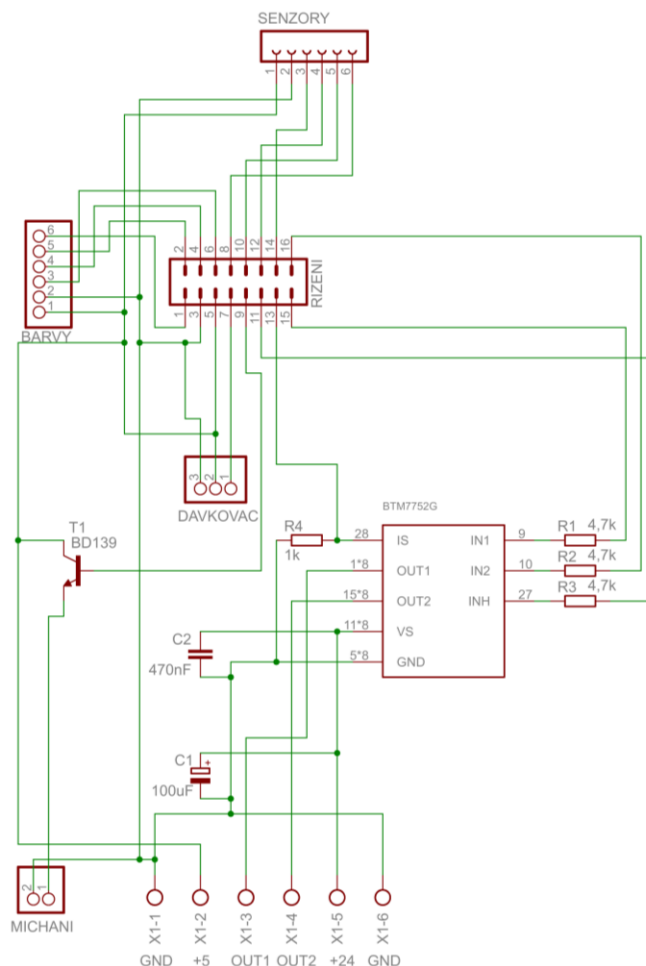
3.3 Princip detekování krabičky

LED dioda vysílá paprsek infračerveného záření. Když se tento paprsek odrazí od krabičky a má dostatečnou intenzitu, fototranzistor se otevře. Čím více dopadne záření do báze fototranzistoru, tím se více otevírá, klesá na něm napětí a protéká proud. Na invertující vstup komparátoru je přivedeno napětí, které je nastaveno pomocí trimru. Na neinvertující vstup je přiveden výstup ze senzoru.

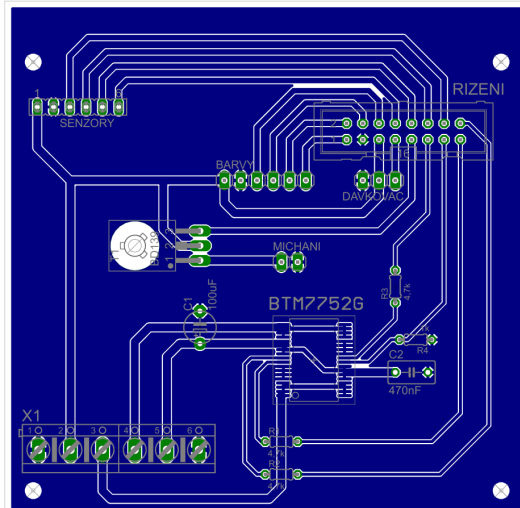
Pokud na fototranzistor nedopadá záření nebo ho není dostatek, je komparátor ve vysoké úrovni (logická 1). Napětí ze senzoru je vyšší než napětí nastavené pomocí trimru. Dopadá-li na fototranzistor dostatek záření z diody, je výstup komparátoru v nízké úrovni (logická 0). Napětí na trimru je vyšší než ze senzoru.

4 Výkonový modul

Modul je navržen jako základní deska. Na desce se nachází výkonový H-můstek BTM7752G, pro ovládání pohonu pásového dopravníku a NPN tranzit pro ovládání míchání. Modul je dále vybaven konektorem, do kterého se zasouvá modul senzoru. Dále jsou zde piny pro připojení řídicí jednotky, detektoru na rozpoznávání barev, piny pro připojení servomotoru, piny pro připojení stejnosměrného motoru, který ovládá míchání v zásobníku a dále svorkovnice pro připojení stejnosměrného motoru k můstku, který je určený pro pohon dopravníkového pásu. Na modulu je svorkovnice, která slouží pro připojení dvou stejnosměrných napájecích zdrojů. Napájecí napětí z prvního zdroje je 24 V, ten slouží pro napájení H-můstku. Napájecí napětí druhého zdroje je 5 V a napájí řídicí jednotku a ostatní periferie. Na obrázku Obr. 10 je znázorněno schéma zapojení výkonového modulu a na obrázku Obr. 11 je plošný spoj výkonového modulu.



Obr. 10 Schéma zapojení výkonového modulu



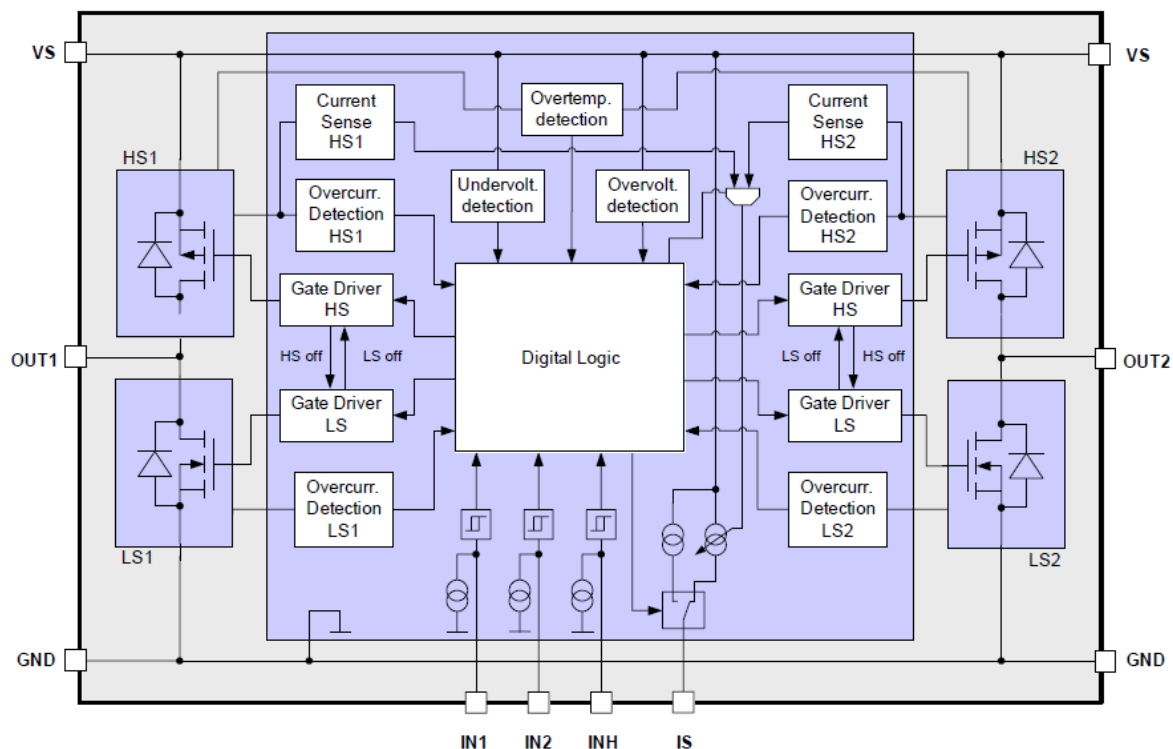
Obr. 11 Plošný spoj výkonového modulu

4.1 Výkonový H-můstek BTM7752G

BTM7752G je plně integrovaný výkonový H-můstek pro řízení stejnosměrných motorů PWM regulací. Obvod je vybaven ochrannými diodami, ovládacími a ochrannými obvody. Blokové schéma je znázorněno na obrázku Obr. 12. Řídicí signál je přiveden na vstupy IN1 a IN2. Dalším vstupem je INH. Ten slouží k nastavení režimu blokování (INH = 0) nebo pohotovostnímu (INH = 1). V blokovacím režimu dochází k sepnutí spodních MOSFET tranzistorů, oba výstupy H-můstku jsou připojeny na „zem“. Vnitřní uspořádání zabraňuje sepnutí dvou tranzistorů nad sebou, tedy zkratu. Výstup IS slouží k informování o směru proudu a případné chybě. V tabulce Tab. 4 jsou uvedeny základní parametry H-můstku a v tabulce Tab. 5 je uveden popis jednotlivých pinů.[8]

Tab. 4 Základní parametry H-můstku BTM7752G[8]

Maximální napájecí napětí	28 V
Minimální napájecí napětí	5 V
Maximální proud	12 A
Maximální frekvence PWM	25 kHz
Minimální pracovní teplota	150°C
Maximální pracovní teplota	-40°C
Klidový proud	5 µA



Obr. 12 Blokové schéma H-můstku BTM7752G [8]

Tab. 5 Popis pinů H- můstku BTM7752G[8]

Pin	Symbol	Funkce
1..4, 33..36	OUT1	Výstup první poloviny můstku
5..8, 23..26	GND	Zem
9	IN1	Vstup první poloviny můstku
10	IN2	Vstup druhé poloviny můstku
11..14, 29..32	VS	Napájení
15..22	OUT2	Výstup druhé poloviny můstku
27	INH	Volba mezi blokovacím a aktivním režimem
28	IS	Směr proudu a hlášení chyb

5 Software

Mikrokontrolér je programován v jazyce C. Pro programování byl zvolen program Atmel Studio 6.1, který je volně ke stažení na stránkách výrobce mikrokontroléru. Do Atmel Studia byl dále doinstalován plugin CodeVisionAVR, který byl rovněž volně na stránkách výrobce. Tento plugin umožňuje rychle a přehledně nastavit příslušný mikrokontrolér. Pro nahrání algoritmu do mikrokontroléru se využívá program AVRDUDE.

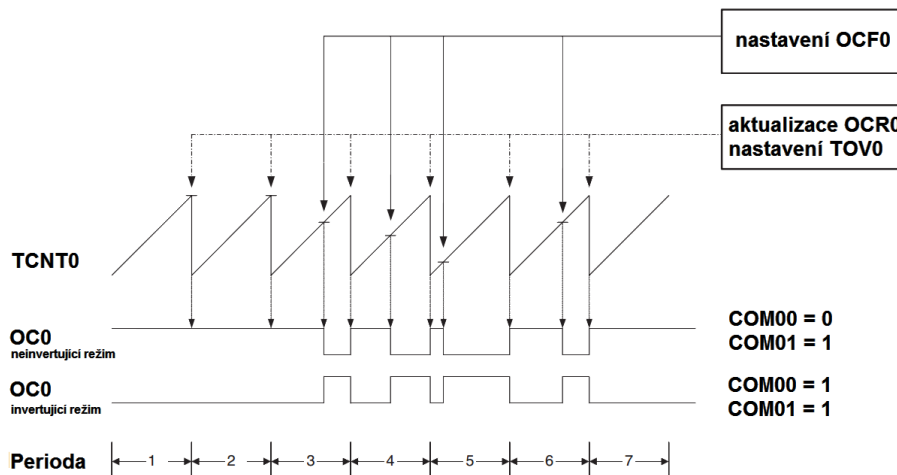
5.1 PWM – rychlý režim

Tento režim čítače se využívá k řízení servomotoru a výkonového můstku, který následně řídí pohon dopravníkového pasu. Rychlý PWM režim se liší od ostatních PWM režimů svojí jednofázovou realizací. Čítač čítá od nuly do maxima. V případě čítače 0 je maximum 255. Po přetečení se vrací opět na nulu a celé čítání se opakuje. Jedná se o režim s dvojnásobnou pracovní frekvencí oproti fázově korigovanému PWM režimu. Princip je vysvětlen pro čítač 0, jedná se o 8 bitový čítač. Pro výstupní kmitočet platí tato rovnice (1).

$$f_{vys} = f_{clk} / (N \times 256) \quad (1)$$

Registr TCNT0 čítá nahoru, až dosáhne maxima a poté se jeho hodnota opět vrátí na nulu. Při shodě s registrem OCR0 (TCNT0 = OCR0) se nastaví příznakový bit OCF0 a tím se stav výstupu OC0 (PB3) nastaví na hodnotu podle nastavení bitů COM00 a COM01 v registru TCCR0. Při dosažení maxima registru TCNT0 dojde k nastavení příznaku přetečení TOV0.

Je-li nastaven registr OCR0 na nulu (0x00), tak je na výstupu OC0 velmi krátký impuls a to při přetečení registru TCNT0. Pro zjednodušení lze uvažovat, že výstup OC0 je v logické 0 při neinvertovaném výstupu (COM00 = 0 COM01 = 1) a při inferovaném výstupu (COM00 = 1 COM01 = 1) je logická 1. Při neinvertovaném výstupu OC0, a pokud je registr OCR0 nastaven na maximální hodnotu (0xFF), je výstup OC0 logická 1. Při invertovaném výstupu a nastavení OCR0 na maximální hodnotu je na výstupu OC0 logická nula. Průběh funkce režimu je znázorněn na obrázku Obr. 13. [1]



Obr. 13 Průběh rychlého PWM režimu, částečně převzato z [11]

5.2 Řídicí algoritmy

Řídicí algoritmus obsahuje hlavní program, ten řídí chod celé linky. Dále obsahuje podprogram s funkcí stop linky, podprogram na nastavení míčku v zásobníku, podprogram na ovládání A/D převodu a podprogram na rozpoznání barev.

5.2.1 Hlavní program

Hlavní program zajišťuje chod celé linky. Je rozdělen pomocí cyklů while na bloky, které nemohou být spuštěny bez proběhnutí předchozího bloku. Tento proces je zajištěn podmínkou cyklu, která je splněna jen pokud proběhl předchozí blok. Piny, které snímají tlačítka nebo senzory, reagují na logickou 0. Činnost linky a případné hlášení pro uživatele se vypisuje na LCD displej. Příkazy na zobrazení textu jsou součástí každého bloku. Hlavní program proběhne tolikrát, kolik je míčku v zásobníku. Tento počet nastavuje obsluha.

Tab. 6 Příklad návaznosti jednotlivých bloků

```

while(start == 1 & pom == 1)           // cyklus while proběhne jen, pokud je pom = 1 a
{                                       start = 1
    lcd_gotoxy(0,0);                    // nastavení prvního znaku na displej
    sprintf(text0,"michani             //zapsání textu do pomocného textového řetězce
    michani ");
    lcd_puts(text0);                     // vypsání obsahu textového řetězce na displej
    OCR0 = b3 = 0x00;                   // registr OCR0 je nastaven na hodnotu 0, výstup
                                        OC0 = 1, probíhá míchání
    delay_ms(5000);                     // doba po kterou je prováděno míchání
    OCR0 = b3 = 0xFF;                   // registr OCR0 je nastaven na hodnotu 255,
                                        výstup OC0 = 0, zastaveno míchání

    pom = 2;
}                                       // po provedení předchozích příkazů se pomocná
                                        proměna pom nastaví na 2
while(pom == 2)                         // cyklus while proběhne jen, pokud je pom = 2
{
    ...

```

V prvním bloku je obsluha vyzvána ke stisknutí tlačítka „start“, toto tlačítko je na portu a pin7, po jeho stisku se nastaví proměna „start“ do logické 1. Tato proměna je po celou dobu v 1, je to z důvodu, aby pro každý cyklus se nemuselo stisknout tlačítko „start“ dále se spustí podprogram na nastavení počtu míčku v zásobníku a H-můstek přepne do aktivního režimu (INH = 1). Podprogram na nastavení počtu míčku v zásobníku je blíže popsán kapitole 5.3.

Pro spuštění následujícího bloku musí být $pom = 1$ a zároveň $start = 1$ jinak nedojde ke spuštění. V tomto bloku dochází k míchání míčku, registr čítače 0 OCR0 je nastaven na potřebnou hodnotu. Po 5 sekundách se míchání zastaví a pomocná proměna je $pom = 2$.

V druhém bloku dochází k obsluze senzoru 1, který detekuje krabíčku na pásu. Pro detekci krabíčky se využívá příkaz if, způsob obsluhy senzorů je vysvětlen v kapitole 5.2.2. Po provedení příkazu je $pom = 3$.

V další bloku dochází ke spuštění linky. Do registrů OCR1AL a OCR1BL je nastavena hodnota příslušná hodnota. Když je krabíčka přemístěna k senzoru 2 je pás vypnut (OCR1AL= 0xFF, OCR1BL=0x00). Pomocná proměna je $pom = 4$.

V následujícím bloku dochází k ovládní servomotoru (dávkování míčku). Pro ovládní servomotoru je využitý čítač 2. Do servomotoru je poslán obdélníkový impuls o délce 1,8633 ms s frekvencí 61 Hz po dobu 500 ms. Poté je délka impulzu zkrácena na 1,097 ms frekvence a doba zůstává stejná. Po umístění míčku do krabíčky je opět spuštěna linka a $pom = 5$.

V pátém bloku je pás zastaven (OCR1AL=0xFF, OCR1BL=0x00) v šachtě na rozpoznávání barev. Krabíčku detekuje senzor 3 za pomoci příkazu if. Následuje podprogram na rozpoznání barev, ten je blíže popsán v kapitole 5.4. Po rozpoznání barvy je pomocná proměna $pom = 6$.

Následuje blok, který opět spustí pás, $pom = 7$. V dalším bloku dochází k detekci krabíčky na konci pásu. Je-li krabíčka na konci pásu, pás se vypne. Další cyklus linky začne až po odebrání krabíčky z pásu. Pomocná proměna pom se nastaví na výchozí hodnotu, a to 1.

5.2.2 Obsluha senzorů

Výstupní signál z vyhodnocovacího modulu senzoru je přiveden na vstupy mikrokontroléru, konkrétně port B piny 1, 2, 3 a 4. Na těchto pinech je aktivován pull-up rezistor. Jedná se o interní rezistor, který je připojen mezi napájecí napětí a příslušný vstupní pin. Rezistor vytahuje vstupní vývod směrem k logické 1.

Při zaznamenání krabičky je výstup z modulu logická 0. Algoritmus na to zareaguje prostřednictvím příkazu `if`. Na ošetření případných zákmitů je využito opakovaného snímání vstupu a to pomocí dvou příkazů `if` za sebou s časovým zpožděním mezi nimi. Jak jsou softwarově ošetřeny zákmity, je ukázáno v tabulce Tab. 7. Změnu časového zpoždění lze rovněž využít jako přesné nastavení pozice krabičky na pásu.

Tab. 7 Softwarové ošetření zákmitů

```

if(PINB.1 == 0)           // první senzor - testování pinu 1 portu B - je-li krabička
                          // na pásu
{
    delay_ms(100);       // časové zpoždění 100ms pro ošetření zákmitu
    if (PINB.1 == 0)     // opětovné testování pinu 1 portu B zda je krabička na pásu
    {
        pom = 3;        // nastavení proměnné pom na hodnotu 3
    }
}

```

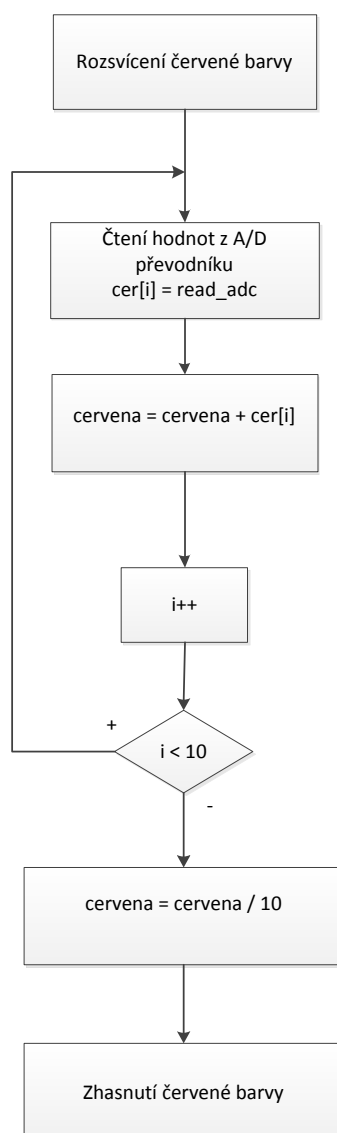
5.3 Nastavení počtu míčků v zásobníku

V tomto podprogramu dochází k nastavení počtu opakování hlavního programu. Automaticky je nastaven počet míčku v zásobníku na šest, došlo by tedy k šesti opakováním. Obsluha může změnit počet opakování pomocí tlačítek. Nastavení se provádí pomocí dvou tlačítek. Tlačítko na portu B pinu 1 zvyšuje počet opakování o jedno a tlačítko na pinu 2 snižuje o jeden. Po potvrzení volby je nutné stisknout třetí tlačítko, to je na pinu 0 portu B. Informace o počtu nastavení je vypisována na LCD displej. Poté je podprogram ukončen a pokračuje hlavní program.

5.4 Rozpoznání barvy

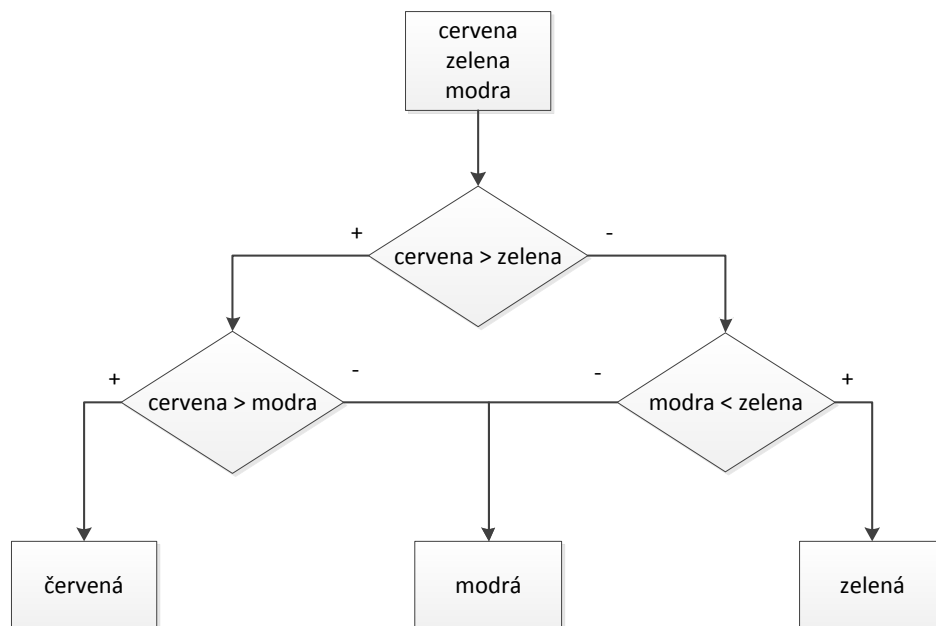
Algoritmus na rozpoznání barev je mimo hlavní program. Na obrázku Obr. 14 je znázorněn postup při snímání barvy, konkrétně při osvětlení červenou LED diodou. K obsluze senzoru barvy je využíván jeden kanál A/D převodníku (PA0) a tři výstupy pro spínání RGB LED (PA1, PA2 a PA3). Nejdříve dochází k osvětlení červenou barvou a je snímáno deset

hodnot napětí na konstantním rezistoru. Mezi jednotlivými měřeními je časový rozdíl. Z naměřených hodnot se udělá průměr, ten slouží jako ukazatel chování míčku při vystavení osvětlení červenou barvou v šachtě. Stejný postup se provádí u zelené a modré barvy.



Obr. 14 Algoritmus pro snímání napětí při osvětlení červenou LED diodou

Po ozáření míčku jednotlivými barvami RGB LED diody dochází k porovnání hodnot od jednotlivých barev. Čím je hodnota vyšší, tím je větší podíl odraženého záření od míčku. Nejvyšší hodnota určí barvu míčku. Výsledná barva je vypsána na displej a do pomocné proměnné „barva“ se uloží číslo barvy, která je detekována (červená = 1, zelená = 2, modrá = 3) Algoritmus rozpoznání barvy míčku je znázorněn ve vývojovém diagramu Obr. 10.



Obr. 15 Algoritmus pro rozpoznání barvy míčku

5.5 Funkce stop

Linku lze zastavit v běhu hlavního algoritmu. Funkce stop je řešena pomocí externího přerušení. K přerušení dojde, pokud je na pin 2 portu D (INT 0) přivedena logická 0. Přeruší se hlavní program a provádí se podprogram. V podprogramu se veškerá činnost linky pozastaví.

Pohon pásu je zastaven, výstup z čítače OC1A a OC1B jsou logické 0, registry CR1AL je nastaven na hodnotu 0xFF a OCR1BL je nastaveny na 0x00. H-můstek je přepnut do režimu blokování (INH = 0). Pokud dochází k míchání míčků, je míchání přerušeno. Výstup OC0 je v logické 0, registr OCR0 je nastaven na hodnotu 0xFF. Je-li v činnosti servomotoru, jeho činnost je přerušena a zůstane v aktuální poloze, výstup OC2 je v logické nule, registr OCR2 je nastaven na 0xFF. Displej informuje, že bylo stisknuto tlačítko „stop“.

Pro ukončení přerušení je třeba stisknout tlačítko „start“. Po jeho stisku se vrátí mikrokontrolér na místo, kde bylo vyvoláno přerušení. Aby činnost linky pokračovala stejně jako před přerušením, je při nastavování registrů OCR0, OCR2, OCR1AL a OCR1BL ukládáno do pomocných proměnných (d4, d5, b3, d7). Po stisknutí tlačítka „start“, se do registru nastaví původní hodnota, která je uložena v příslušné pomocné proměnné a aktivuje v H-můstku aktivní režim (IHN = 1).

Závěr

Cílem této bakalářské práce bylo navrhnout a realizovat řídicí jednotku a příslušné řídicí algoritmy modelu automatizované třídící linky.

Při návrhu řídicí jednotky a řídicích algoritmů muselo dojít k úpravě některých částí modelu. Jednou z těchto částí byl detektor na rozpoznávání barev. Původní detektor obsahoval různý počet LED diod, konkrétně se jednalo o dvě červené, dvě zelené a jednu modrou LED diodu. Každá dioda osvětlovala míček z jiného úhlu, mohlo docházet k chybnému snímání odrazu záření od míčku. Z tohoto důvodu byl tento detektor vyměněn za detektor, který obsahuje jednu RGB led diodu. Vnitřek šachty je nastříkán černou barvou, kvůli pohlcení případných nežádoucích odrazů.

Dále bylo nutné vyměnit výkonový H-můstek L6203 za jiný. Pro tuto aplikaci byl zvolen výkonový H-můstek BTM7752G. K této výměně došlo z důvodu proudového přetěžování můstku L6203. Můstek bylo nutné chladit ventilátorem i po vypnutí celého modelu. Doba chlazení po vypnutí byla 10-15 minut.

Na lince byly také vyměněny původní senzory, které byly použity na detekci krabičky na pásu. Z důvodu kontroly přítomnosti krabičky na začátku linky bylo potřeba linku osadit čtvrtým senzorem. Jelikož původní senzory vykazovaly chybovost a nebyla možnost přidat čtvrtý senzor, byly vyměněny za čtveřici nových senzorů s vlastním vyhodnocovacím modulem.

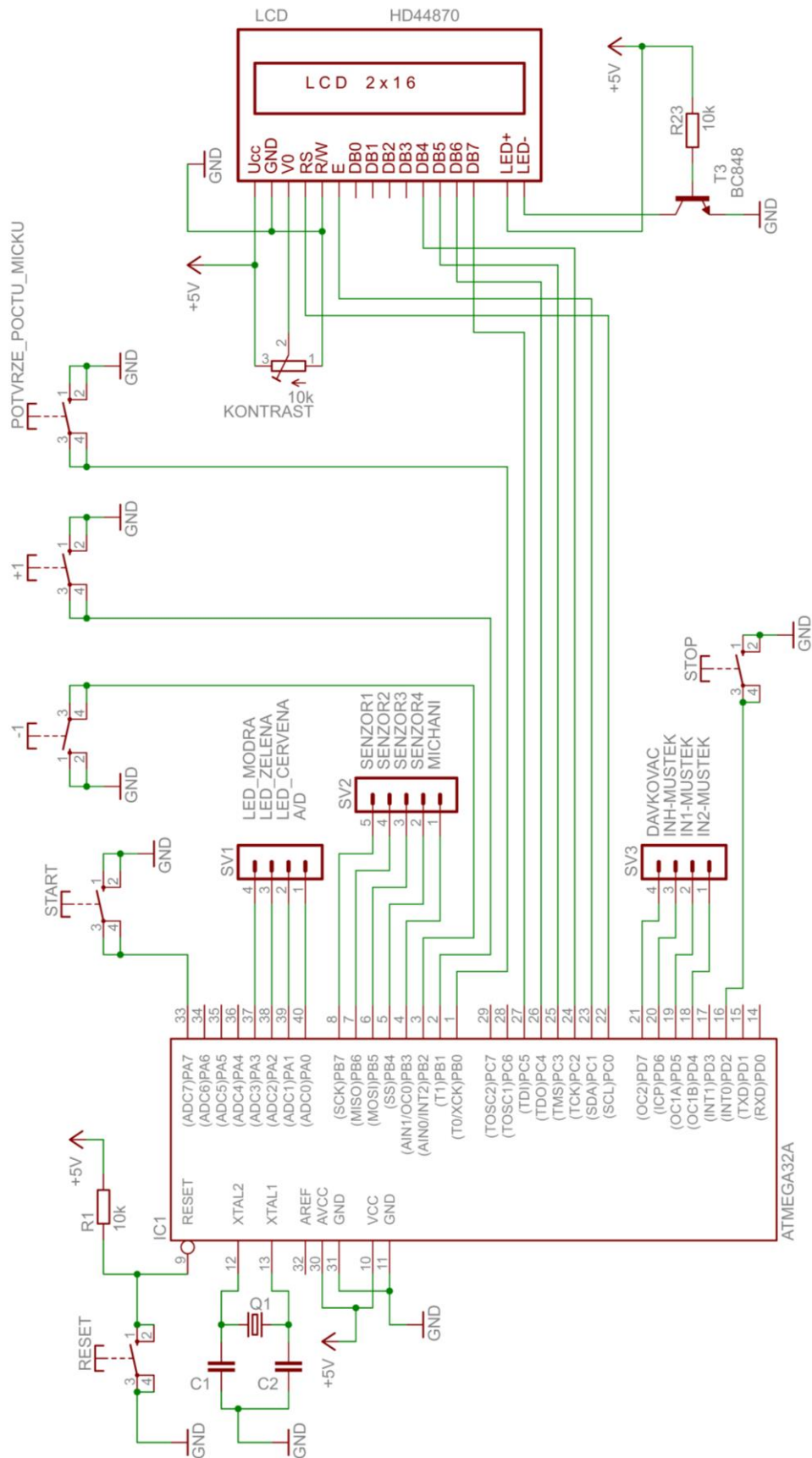
Celý řídicí program je rozdělen do menších bloků, aby se dal program snadno pozměnit. Při vývoji řídicích algoritmů vždy docházelo k testování jednotlivých částí algoritmu na modelu, případně na perifériích, které měly obdobnou funkci. Testování bylo možné především díky variabilitě vývojového kitu, který byl posléze použit jako řídicí jednotka a byl dovybaven LCD displejem.

Seznam literatury a informačních zdrojů

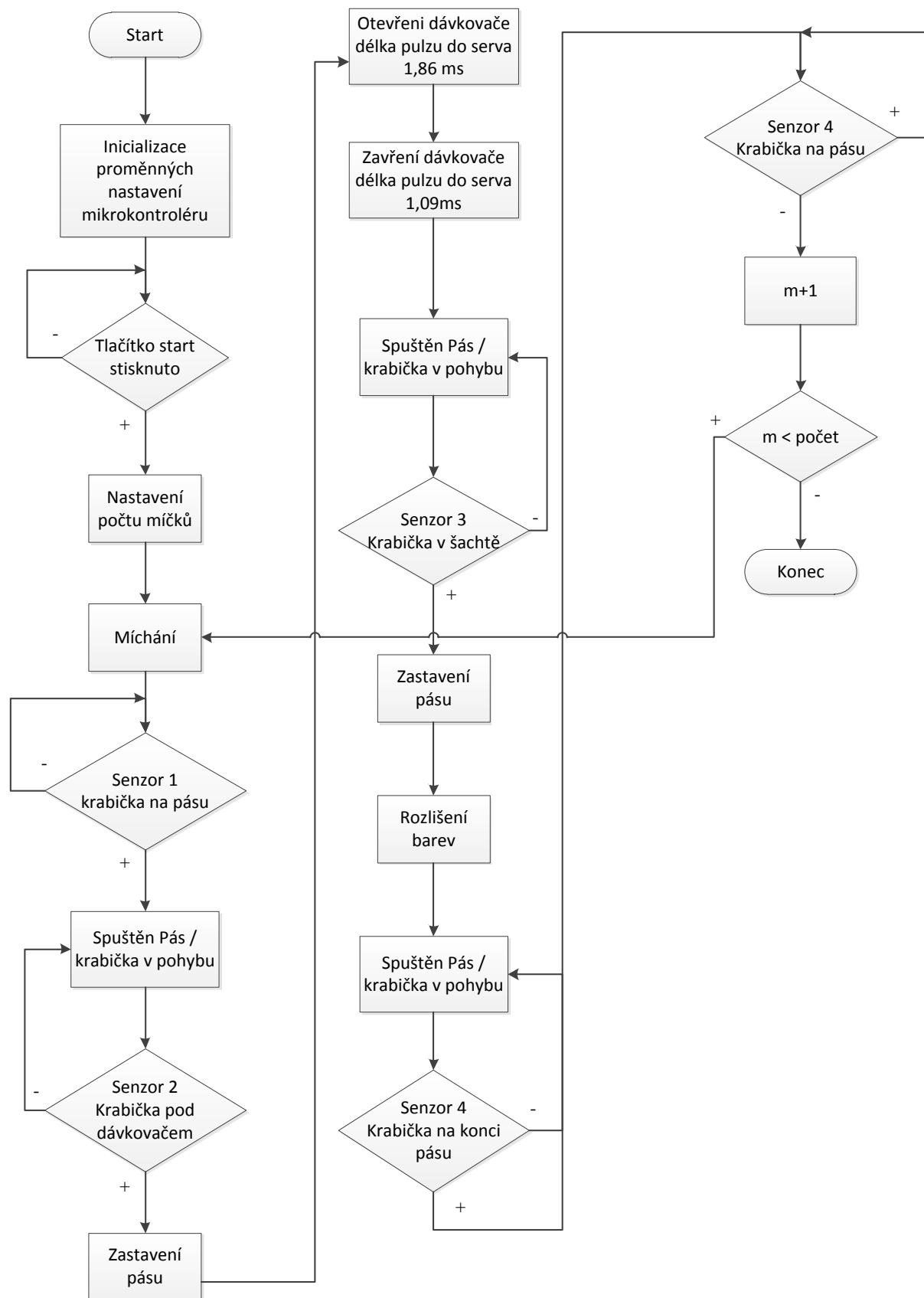
- [1] MATOUŠEK, David. *Práce s mikrokontroléry ATMEL AVR ATmega16*. 1. vyd. Praha: BEN - technická literatura, 2006, 319 s. ISBN 80-730-0174-8.
- [2] VÁŇA, Vladimír. *ARM pro začátečníky*. 1. vyd. Praha: BEN - technická literatura, 2009, 195 s. ISBN 978-80-7300-246-6.
- [3] VÁŇA, Vladimír. *Mikrokontroléry Atmel AVR*. 1. vyd. Praha: BEN - technická literatura, 2003, 215 s. ISBN 80-7300-102-0.
- [4] BENEŠ, Petr. *Mechatronické modely výrobních procesů*. [Diplomová práce]. Plzeň: ZČU, 2008
- [5] MASÁK, Tomáš. *Řídicí jednotka systému detekce barvy s autokalibrací*. [Bakalářská práce]. Plzeň: ZČU, 2013
- [5] Návod k použití: Vývojový kit EvB 4.3 v4 [online]. [cit. 2015-03-25].
Dostupné z: <http://www.vo.gme.cz/dokumentace/752/752-445/czn.752-445.1.pdf>
- [6] *Data sheet ATmega 32A* [online]. [cit. 2015-05-12].
Dostupné z: http://www.atmel.com/Images/Atmel-8155-8-bit-Microcontroller-AVR-ATmega32A_Datasheet.pdf
- [7] *Data sheet LM 339* [online]. [2015-05-31].
Dostupné z: <http://www.farnell.com/datasheets/1639828.pdf>
- [8] *Data sheet BTM7752G* [online]. [cit. 2015-31-5].
Dostupné z: <http://www.farnell.com/datasheets/1836127.pdf>
- [9] *CodeVisionAVR User Manual CodeVisionAVR* [online]. [2015-04-25] .
Dostupné z: http://www.hpinfotech.ro/cvavr_documentation.html

- [10] *Stránky Základy ovládání mikrokontroléru ATmega* [online]. [cit. 2015-05-13].
Dostupné z: <http://www.tajned.cz/2014/11/zaklady-ovladani-mikrokontroleru-atmega-14-dil-dvouradkovy-lcd-displej/>
- [11] *Stránky Základy ovládání mikrokontroléru ATmega* [online]. [cit. 2015-05-16].
Dostupné z: <http://www.tajned.cz/2014/10/zaklady-ovladani-mikrokontroleru-atmega-11-dil-citac-casovac-1-cast-zakladni-popis/>
- [12] *Stránky Amatérské elektrotechniky* [online]. [2015-04-20]
Dostupné z: <http://vlastikd.webz.cz/bastl/serva.htm>

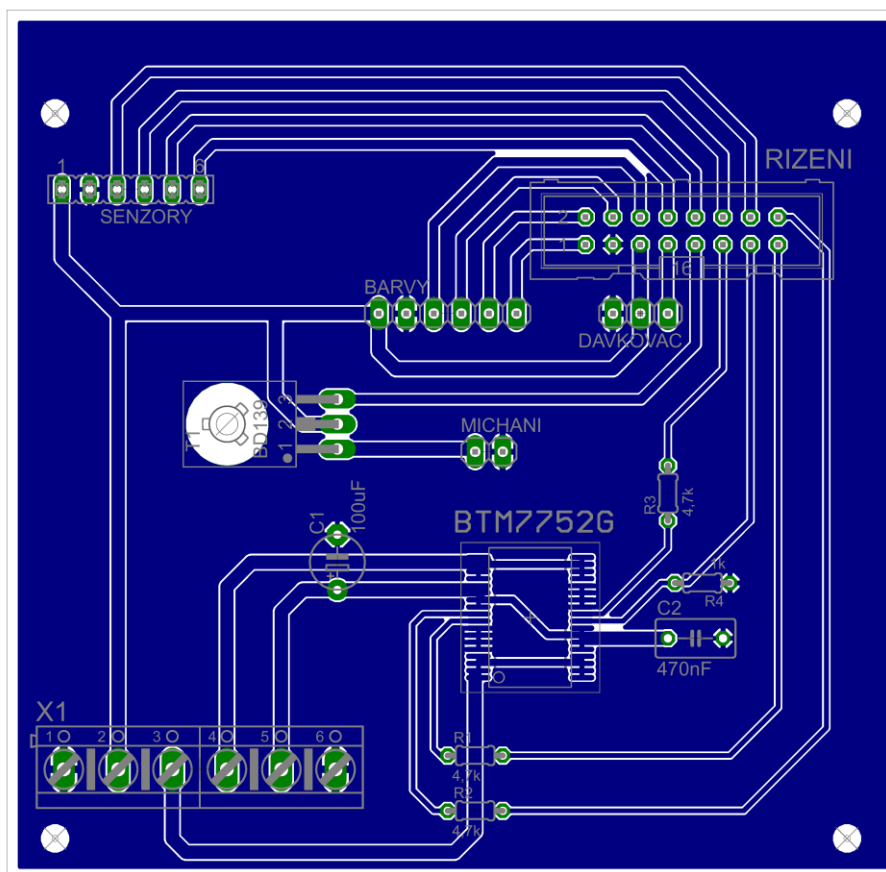
Přílohy



Obrázek A - Schéma zapojení řídicí jednotky na vývojovém kitu



Obrázek B Vývojový diagram řídicího algoritmu



Obrázek C Plošný spoj výkonového modulu

Obsah přiloženého CD

- Bakalářská práce ve formátu *.pdf a *.docx
- Program v jazyce C pro řízení třídící linky
- Veškeré obrázky a schémata použité v práci
- Návrh plošného spoje výkonového modulu