VYSOKÁ ŠKOLA BÁNSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA FAKULTA STROJNÍ
KATEDRA AUTOMATIZAČNÍ TECHNIKY A ŘÍZENÍ

ROZPOZNÁVÁNÍ OBRAZU V ŘÍDICÍCH SYSTÉMECH

Autor práce: Lukáš Smolka
Vedoucí práce: doc. Ing. Jaromír Škuta, Ph.D.

Ostrava 2017

Obsah

1 ÚVOD 1

2 Složení třídicí linky 2

3 Použité součástky 3

3.1. Mikroprocesor PIC 3

3.2. Modul pro ovládání stejnosměrných motorů L293D 3

3.3. Ovládání krokových motorů POLOLU A4988 4

3.4. Deska plošného spoje pro ovládání linky 4

4 Programy pro chod linky 5

4.1. Program v mikroprocesoru 5

4.2. Program v ControlWebu 8

4.3. Strojové vidění VisionLab 9

5 Závěr 10

6 Literatura 11

1 ÚVOD

V první části práce bude rozebrána samotná třídicí linka. Jak funguje, a k čemu vůbec slouží.

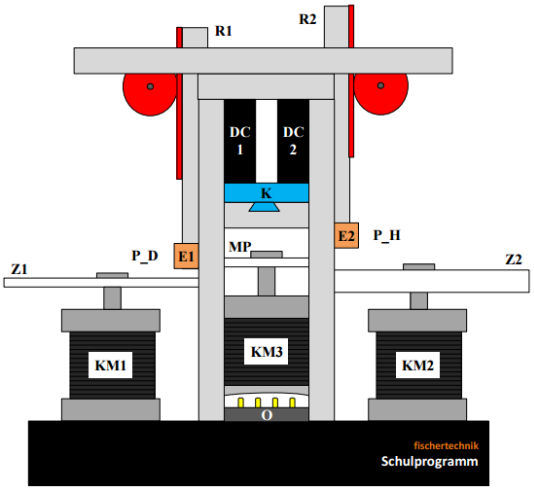
K tomu budou stručně popsány součástky a moduly, které tuto linku ovládají. Hlavní součástkou je mikroprocesor řady PIC, ten je „mozkem“ celého systému. Periférie, jako je krokový motor nebo stejnosměrný motor, jsou ovládány běžně dostupnými moduly. V této kapitole bude ukázán návrh DPS pro ovládání třídicí linky. Na této DPS budou osazeny všechny moduly, které byly výše popsány.

V poslední části práce budou popsány programy pro chod linky. První z nich je program, který je nahrán v mikroprocesoru a druhý, který běží na PC a ovládá celý proces „shora“. Je to program vytvořený v software ControlWeb 7. Je zde udělaný manuální i automatický režim. V této aplikaci je udělána i jednoduchá vizualizace, která ukazuje to, co dělá linka v reálném prostředí. Neméně důležitou součásti je strojové vidění. To je realizováno pomocí software VisionLab, to je rozšíření samotného programu ControlWeb. V této aplikaci je strojové vidění zprostředkováno obyčejnou webkamerou.

2 Složení třídicí linky

Třídicí linka, která je v této práci popisována může být spíše chápána jako inspekční linka. Jelikož jsou zde rozlišovány objekty, pomocí rozpoznávání obrazu a následně je s nimi nakládáno, tak jak bylo předem naprogramováno. (1)

Složení této třídicí linky je velice jednoduché. Jsou zde použité pouze základní součástky. Tím je myšleno, tři krokové motory, dva stejnosměrné motory, jako efektory jsou zde použity dva elektromagnety. Abychom měli informaci o poloze ramen efektorů, je zde zpětná vazba v podobě koncových snímačů. (1)



Obr. 2. 2 - Třídicí linka [Petrtýl, 2013]

Výše není uveden popis kamery, ta je zastoupena písmenem K, dále P_D a P_H je dolní, resp. horní poloha ramene. Z1 je zásobník dílů k inspekci a Z2 je zásobník roztříděných dílů. MP je manipulační plošina. (1)

Tato třídicí linka bude třídit podložky pod šrouby. V zásobníku roztřízených dílů je osm pořadačů, to znamená, že zde bude k třízení osm různých druhů podložek.

3 Použité součástky

V této kapitole budou stručně popsány použité součástky. Nejdůležitější je PIC16F877A, pro ovládání krokových motorů slouží POLOLU A4988 a pro ovládání stejnosměrných motorů je modul L293D. Pro spínání a vypínání elektromagnetů slouží relé.

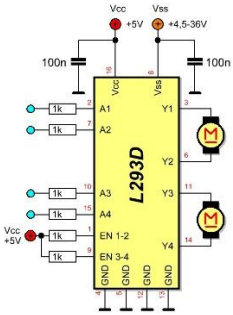
3.1. Mikroprocesor PIC

Mikroprocesor PIC16F877A obsahuje 40 pinů. Piny využité v této aplikaci jsou nastaveny jako digitální.

Obr. 3. 1 - Mikroprocesor PIC16F877

3.2. Modul pro ovládání stejnosměrných motorů L293D

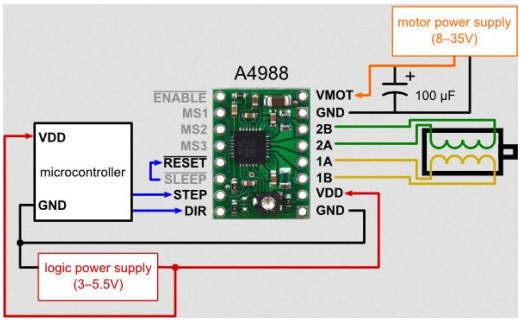
Tento modul dokáže ovládat dva stejnosměrné motory. Jde vlastně o dva plné H-můstky. Každý z těchto můstků má své vlastní ovládání. Modul je ovládán pomocí logických úrovní TTL a to z praktických důvodů. Je zde totiž možnost připojení na další logický obvod, anebo jako v tomto případě na mikroprocesor. (2)



Obr. 3. 1 – Modul L293D – pinout

3.3. Ovládání krokových motorů POLOLU A4988

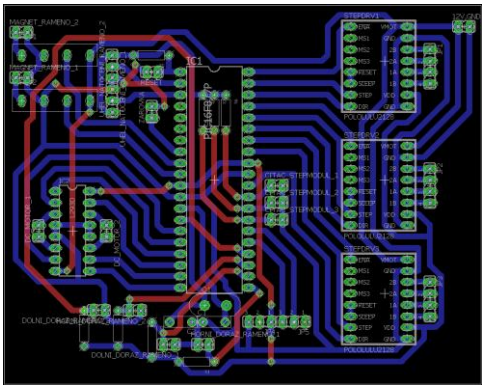
Pin ENABLE slouží na protáčení motoru, čili přivedeme-li zde logickou 1, není motor nijak brzděn a volně se protáčí. Pin RESET mluví sám za sebe, slouží k restartování modulu, po přivedení logické 0. Každý impuls na pin STEP, odpovídá kroku motoru. Pin DIR slouží k nastavení směru otáčení. (3)



Obr. 3. 3 – A4988 minimální pinout

3.4. Deska plošného spoje pro ovládání linky

Návrh probíhal v programu EAGLE. Snažil jsem se, aby všechny motory a vše co třídící linka obsahuje, bylo ovládáno pomocí jedné DPS.



Obr. 3.4 – Návrh DPS v programu EAGLE

Pomocí tohoto návrhu jsme DPS vyleptal a začal osazovat potřebnými součástkami. Deska je napájena pomocí modulu pro převod z USB na RS232. Dále je zde napájení 12V, to slouží k napájení modulů pro motory a napájení elektromagnetickým cívek.

4 Programy pro chod linky

V této kapitole budou popsány programy, které zajišťují chod linky. V první podkapitole bude ukázat program, který je nahrán v mikroprocesoru. A v druhé podkapitole program psaný v ControlWebu, ten běží na PC a dohlíží na chod linky „shora“.

4.1. Program v mikroprocesoru

Prvním krokem, který program udělá, je to, že nastaví výstupní porty jako výstupní nebo vstupní. Dále program uvede linku do základní pozice.

Prvním krokem je nastavení ramen do základních poloh, na každém rameni jsou dva koncové spínače, ty říkají, zda jsou ramena v horní nebo dolní poloze. Ty jsou neustále napájeny přes pull-up rezistor a napojeny na mikroprocesor. To znamená, že při sepnutí vynulují pin mikroprocesoru. Pro základní polohu je vyžadováno, aby bylo rameno v poloze horní.

```
//Rameno 1//
    do {
        portd.f0 = 1;
        portc.f3 = 0;
    } while(portd.f3 != 0);

portd.f0 = 0;
portc.f3 = 0;
spinace += 0x02;
delay_ms(1000);

//Rameno 2//
    do {
        portc.f1 = 1;
        porta.f5 = 0;
    } while(portc.f5 != 0);

portc.f1 = 0;
porta.f5 = 0;
spinace += 0x08;
delay_ms(1000);
```

Obr. 4. 1 - Kód pro nastavení ramen do základních pozic

Poté se nastavují krokové motory. Pod každým talířem je taktéž koncový spínač. Fungují stejně jako u ramen.

```
//Krokovy motor 1 do nulove pozice
    do {
        portb.f6 = 1;
        delay_ms(10);
        portb.f6 = 0;
        delay_ms(10);
    } while(portd.f5 != 0);
```

Obr. 4. 2 - Kód pro nastavení krokového motoru do základní pozice

Mikroprocesor přijímá data od PC. Musel být vytvořen určitý protokol, aby mikroprocesor mohl data zpracovat a vykonat požadovanou akci. Kód pro příjem dat lze vidět na dalším obrázku.

```
void framesave() {
unsigned char buffer[10];
unsigned short pom = 0;

        if(Uart1_Data_Ready()) {
                interm = Uart1_Read();
                if(interm == 38) {
                        Uart1_Read_Text(buffer, "$", 10);
                        MCU_adress = buffer[0];
                        adress = buffer[1];
                        data1_1 = buffer[2];
                        data1_2 = buffer[3];
                        data1_3 = buffer[4];
                        data2_1 = buffer[5];
                        data2_2 = buffer[6];
                        data2_3 = buffer[7];
                        data1 = data1_1*100 + data1_2*10 + data1_3;
                        data2 = data2_1*100 + data2_2*10 + data2_3;
                        checksumin = buffer[8];
                        interm = 0;
                        checksum = MCU_adress + adress + data1_1 + data1_2 + data1_3 + data2_1 + data2_2 + data2_3;
                }
        }
}
```

Obr. 4. 3 - Kód pro příjem dat

Data se začnou ukládat v případě, že první znak je ‚&', poté ukládá do bufferu až do doby než přijde ‚$'. Data z bufferu jsou využívány programem pro různé periférie. V proměnné MCU_adress je adresa mikroprocesoru, v proměnné adress je adresa periférie (krokový motor, stejnosměrný motor, elektromagnety a světlo). Další proměnné data1_1 až data1_3 obsahují pouze znaky 1 až 9, ty se potom sloučí ve výpočtu data1. To stejné platí pro data2. Toto opatření je z důvodu, aby se znaky v datech nerovnaly koncovému znaku.

Data se „vytáhnou" pomocí bitových součinů např.: *ENABLE = mask_enable & data1;*

mask_enable: 10000000

&

data1: 11101100

výsledek: 10000000

Odesílání dat probíhá podobně, ovšem data jsou odlišné. Zpět se vždy posílá aktuální poloha ramen a činnost magnetů a světla.

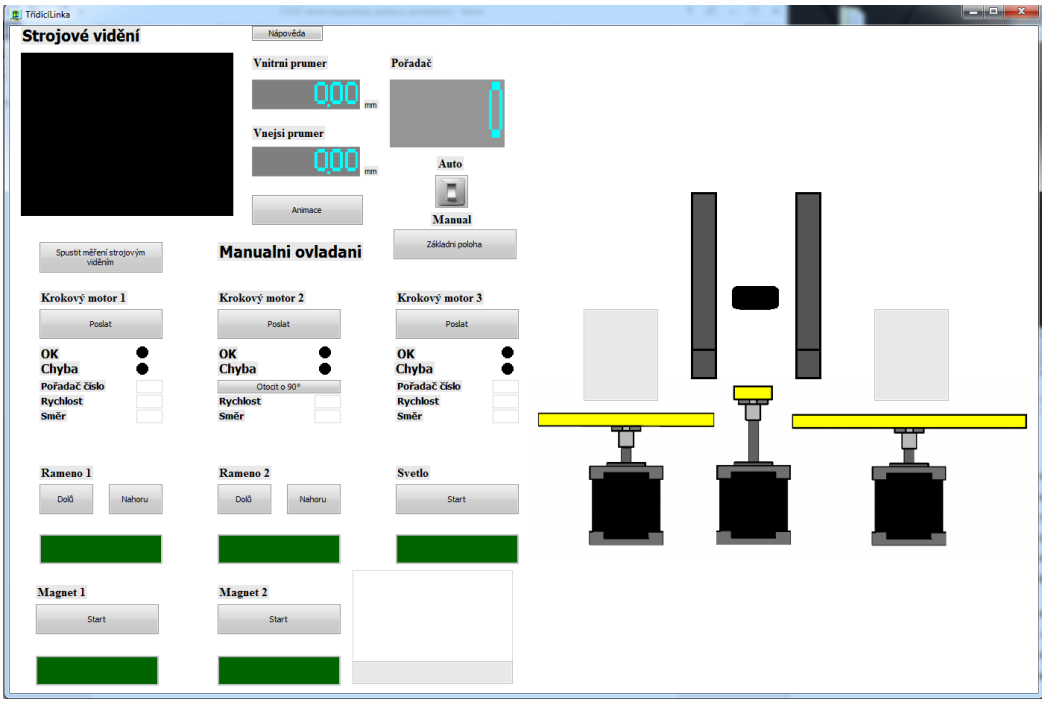
Na následující, obrázku lze vidět, jak program pro krokový motor využívá příchozí data. Data2 je počet kroků o kolik se má krokový motor otočit.

```
void MCUA() {
    switch(adress) {
        case 'a':   adressOUT = 'a';
                    STEP = mask_stepstep & data1;
                    DIR = mask_stepdir & data1;
                    ENABLE = mask_enable & data1;

                    if(ENABLE == 128)
                              portb.f7 = 1;
                    else portb.f7 = 0;

                    if(DIR == 64)
                           portb.f5 = 1;
                    else portb.f5 = 0;

                    //Krokovy motor 1 do nulove pozice
                    do {
                       portb.f6 = 1;
                       delay_ms(10);
                       portb.f6 = 0;
                        delay_ms(10);
                       } while(portd.f5 != 0);


                    for (i=0; i<data2; i++) {
                        portb.f6 = 1;
                        Delay_ms(10);
                        portb.f6 = 0;
                        Delay_ms(10);
                    }
                    delay_ms(1000);
                    ackframe(adressOUT,spinace,mag12svetlo);

                    adressOUT = 0;
                    break;
```

Obr. 4. 4 - Kód pro ovládání krokového motoru

4.2. Program v ControlWebu

Aplikace naprogramována v tomto programu řídí celou třídicí linku. A to tak, že posílá framy do mikroprocesoru, ten již dokáže frame vyhodnotit. Aplikace je musí umět i vyhodnotit, protože mikroprocesor odpovídá ve stejném formátu.



Obr. 4. 5 – Aplikace v ControlWebu

Výstupní frame se postupně skládá. U krokových motorů je to z čísla pořadače, rychlosti a směru otáčení. Výsledkem je nějaké číslo, to je následně rozděleno a převedeno na znaky. Poté se složí se vstupním znakem a adresami a odešle se po sériové lince do mikroprocesoru.

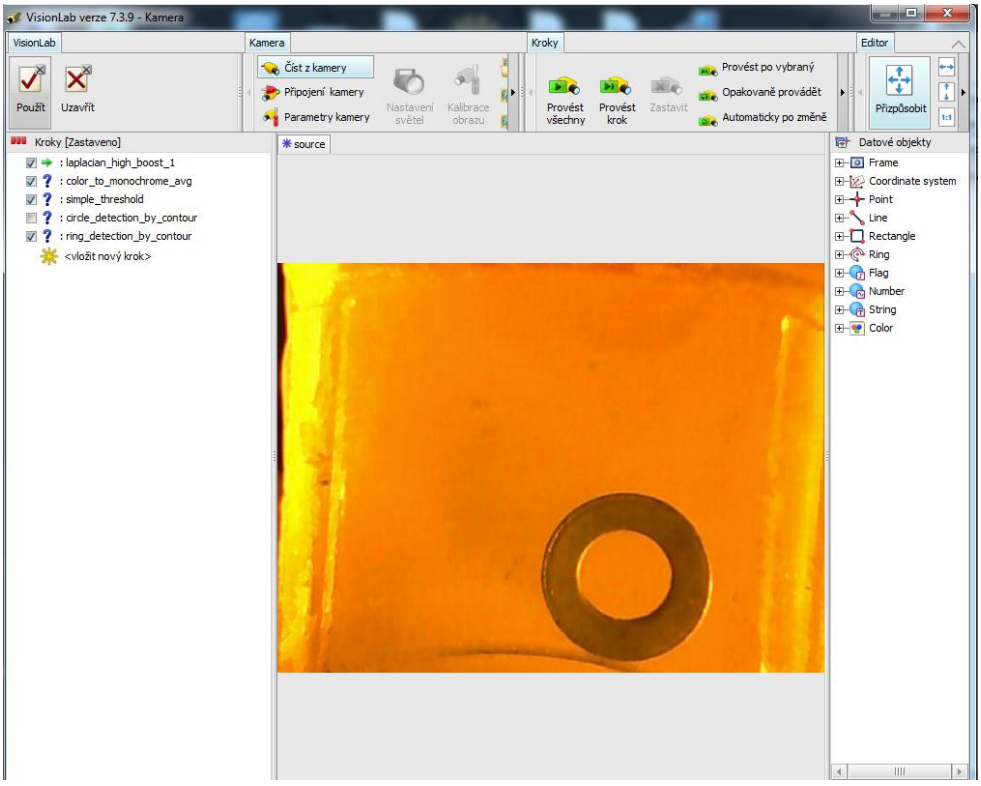
Na předchozím obrázku je ukázán manuální režim. Všechny periférie se musí ovládat zvlášť. Na animaci můžeme vidět, co třídicí linka dělá, spíše v jaké poloze je. Pokud je příkaz vykonán, rozsvítí se indikátor OK v případě krokových motorů, v případě ostatních periférií se na zeleném displeji napíše pozice, či činnost dané periférie.

Automatický režim se zapíná pouze tlačítkem Start.

4.3. Strojové vidění VisionLab

Strojové vidění zajišťuje program VisionLab, je k tomu využita obyčejná webkamera. Aby mohla webkamera kvalitně sejmout obraz, musí být objekt dostatečně osvětlen. To zajištuje SUPER FLUX LED. Obraz se sejme a následně jsou provedeny určité kroky pro zpracování obrazu. Prvním krokem je *laplacian_high_boost*, tato operace slouží k tomu, aby byl obraz zaostřen, a byly zvýrazněny hranice přechodu mezi barvami. Druhým krokem je *color_to_monochrome_avg*, tento krok převede barevný obraz do monochromatického. Třetím krokem je *simple_threshold*, tento krok převádí barevné pixely na jednu ze dvou hraničních hodnot podle určitého prahu. Jednoduše převede obraz na černobílý. Posledním krokem je *ring_detection_by_contour*, tento krok již vyhodnotí rozměry prstence a uloží do proměnných.

Proměnné jsou následně převedeny do CotrolWebu a vyhodnoceny na displejích. Rozměry se porovnávají podle naprogramovaných kritérií a je vyhodnocen pořadač, do kterého má být podložka umístěna.



Obr. 4. 6 – VisionLab

5 Závěr

Tato práce popisuje třídicí linku na podložky, její použití a hardware i software pro ovládání. Linka se skládá z krokových motorů, na kterých jsou umístěny zásobníky a pracovní plošina. Dále ze stejnosměrných motorů, ty ovládají ramena, elektromagnety, ty slouží jako efektory a z koncových spínačů pro zpětnou vazbu. Dále je zde webkamera pro strojové vidění a LED dioda pro osvětlení obrazu.

Hardware se skládá z DPS, na které je osazen mikroprocesor, modul Pololu A4988 pro krokové motory, modul L293D pro stejnosměrné motory, relé pro elektromagnety a pull-up rezistory pro koncové spínače. DPS je umístěna v krabičce, která je pomocí CANON konektorů napojena k třídicí lince.

Program nahraný v mikroprocesoru obsluhuje všechny periférie třídicí linky. Program reaguje na frame přijatý z vyšší vrstvy, tj. z PC. Na PC je spuštěný program v ControlWebu, ten do mikroprocesoru posílá příkazy a zajišťuje strojové vidění. Strojové vidění poskytuje program VisionLab, ten je přídavnou součástí ControlWebu. V něm jsou naprogramovány určité kroky pro zpracování obrazu, to proto, aby bylo možné odečíst rozměry podložky.

Aplikace umožňuje jak manuální režim, tak automatický. U obou režimů je možné si zapnout vizualizaci linky. U manuálního režimu je třeba každou periférii ovládat zvlášť, zatímco automatický režim dokáže třídit sám a pořád.

Tuto práci bych rád dále rozšiřoval, a to tak, že pro přístup k manuálnímu režimu by mělo být zadáno přihlašovací jméno a heslo. Dále, aby bylo strojové vidění aktivní pouze určitý čas, a pokud by za tento čas rozměry nerozpoznalo, podložka by byla umístěna do zvláštního pořadače. A po roztřízení celého zásobníku (20 podložek) by linka zkusila nerozpoznané podložky znovu roztřídit.

6 Literatura

[1] PETRTÝL, O. *Využití systému strojového vidění pro polohování objektu:* diplomová práce. Ostrava: VŠB – Technická univerzita Ostrava, Univerzitní studijní programy, Katedra automatizační techniky a řízení (Fakulta strojní), 2013, 55 s. Vedoucí práce: ŠKUTA, J.

[2] Robotem sem, robotem tam II: H-můstek s tranzistory. *Robodoupě: Web nejen o robotice* [online]. 2011 [cit. 2017-04-17]. Dostupné z: http://robodoupe.cz/2011/robotem-sem-robotem-tam-ii-%E2%80%93-elektronika-take-neni-k-zahozeni/

[3] *A4988* [online]. 1. Massachusetts, 2014, s. 20 [cit. 2017-04-17]. Dostupné z: https://www.pololu.com/file/download/A4988.pdf?file_id=0J450