

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**Vývoj a implementace řídicího systému pro obráběcí stroj**

Radek Hauck

Diplomová práce

2017

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2015/2016

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Radek Hauck**  
Osobní číslo: **I14250**  
Studijní program: **N2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Vývoj a implementace řídicího systému pro obráběcí stroj**  
Zadávající katedra: **Katedra softwarových technologií**

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

Cílem diplomové práce je návrh, vývoj a implementace řídicího systému pro obráběcí stroj, který obrábí materiál formou broušení.

Obráběcí stroj je v počáteční fázi řízen jen relátkou a posuv brusných kotoučů a podávacího stolu je řešen hydraulickým olejem. Posuv Brusných kotoučů a podávacího stolu se zamění za krokové motory a jejich pohony za účelem zpřesnit pohyb s přesností 0,001mm a ulehčit životnímu prostředí tím, že se nebude používat hydraulický olej.

Etapy tvorby řídicí jednotky:

\* Stávající relátka se zamění za řídicí jednotku PLC za účelem náhrady relátek, které se časem opotřebují. Výhodou PLC je tak možnost naprogramování složitějších systémů za využití interních časovačů, čítačů a funkcí pro řízení krokových motorů.

\* V druhé fázi jde o naprogramování řídicí jednotky PLC spolu s pohony pro pohyby. Hodnoty pro obrábění, přepínání mezi druhy broušení a nastavování obráběcího stroje bude realizováno dotykovým panelem HMI. Odměřování polohy bude realizováno digitální indikací.

\* Ve třetí fázi jde o naprogramování HMI panelu který komunikuje s PLC přes sériovou linku. Programování panelu je rozděleno do dvou částí a to grafická neboli vizuální část a komunikační část pro komunikaci s PLC pomocí maker.

Výsledkem práce bude návrh NC obráběcího stroje.

Rozsah grafických prací: **Tabulky, plány, obrázky podle potřeby**

Rozsah pracovní zprávy: **60 stran**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

\* ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. 1. vyd. Praha: BEN - technická literatura, 1999, 223 s. ISBN 80-860-5658-9.

\* ŠMEJKAL, Ladislav. PLC a automatizace. 1. vyd. Praha: BEN - technická literatura, 2005, 207 s. ISBN 80-730-0087-3.

\* PLC Fatek návody. PLC Fatek. [online]. 5.10.2015 [cit. 2015-10-05]. Dostupné z: <http://fatek.seapraha.cz/hlavni-strana/dokumentace/navody/fbs/>

Vedoucí diplomové práce: **prof. Ing. Karel Šotek, CSc.**  
Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2015**

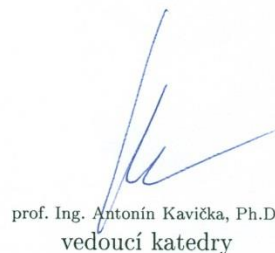
Termín odevzdání diplomové práce: **13. května 2016**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



prof. Ing. Antonín Kavička, Ph.D.  
vedoucí katedry

V Pardubicích dne 15. listopadu 2015

## **Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 07. 05. 2017

Radek Hauck

## **Poděkování**

Tímto bych chtěl poděkovat panu prof. Ing. CSc. Karlovi Šotkovi za pomoc při zpracování bakalářské práce za ochotu, vstřícný přístup a za cenné rady.

## **Anotace**

Cílem diplomové práce je návrh, vývoj a implementace řídicího systému pro obráběcí stroj, který obrábí materiál formou broušení.

## **Klíčová slova**

Obráběcí stroj, PLC, HMI, NC polohování, odměřování, servo motory, krokové motory

## **Title**

Development and implementation of a control system for the machine tool.

## **Annotation**

The goal of this thesis is the design, development and implementation of a control system for a working machine, which cuts the material in the form of grinding.

## **Keywords**

Machine tools, PLC, HMI, NC positioning, measuring system, servo motor, step motor

# Obsah

<b>Seznam zkratk</b> .....	<b>0</b>
<b>Seznam obrázků</b> .....	<b>1</b>
<b>Seznam tabulek</b> .....	<b>3</b>
<b>1 Úvod</b> .....	<b>4</b>
<b>2 Teoretická část</b> .....	<b>5</b>
2.1 Co je to obráběcí stroj.....	5
2.2 Rozdělení obráběcích strojů .....	5
2.2.1 Soustružení – hrotový soustruh .....	5
2.2.2 Frézování .....	6
2.2.3 Vrtání .....	7
2.2.4 Broušení.....	8
<b>3 Použité technologie</b> .....	<b>10</b>
3.1 Řídicí systém .....	10
3.1.1 PLC.....	10
3.1.2 Programování PLC .....	11
3.2 Ovládání .....	18
3.2.1 HMI .....	18
3.3 Realizace posuvu .....	18
3.3.1 Servomotor .....	18
3.4 Měření.....	20
3.4.1 Princip optických mikrometrů .....	20
3.4.2 Vlastnosti měřidel.....	21
<b>4 Broušení</b> .....	<b>23</b>
4.1 Bezhraté broušení .....	23
4.1.1 Definice broušení.....	23

4.1.2	Způsoby bezhrotého broušení.....	24
4.1.3	Funkce bezhrotého broušení.....	27
4.2	Rozdělení bezhrotých brusek podle posuvů .....	29
4.2.1	Popis brousícího stroje .....	30
<b>5</b>	<b>Praktická část.....</b>	<b>31</b>
5.1	Ovládání stroje.....	31
5.1.1	Multiplex pro tlačítka .....	38
5.2	Bezpečnost.....	39
5.3	PLC.....	40
5.3.1	Programování .....	41
5.3.2	Programování rychlých výstupů a pohybu servomotorů.....	42
5.3.3	Komunikace.....	44
5.3.4	Multiplex tlačítek.....	46
5.3.5	Vyhodnocení měřidla .....	48
5.3.6	Odměřování .....	49
5.3.7	Rychlost brusného kotouče.....	50
5.3.8	Realizace přepínače pomocí dvojkové soustavy .....	51
5.3.9	Automatický a ruční režim .....	53
5.4	HMI .....	54
5.4.1	Základní funkce .....	55
5.4.2	Makra.....	55
5.5	Servopohony.....	58
5.5.1	Základní nastavení parametrů.....	59
5.5.2	Nastavení servo měniče .....	60
5.6	Krokové motory.....	60
<b>6</b>	<b>Závěr .....</b>	<b>61</b>
<b>7</b>	<b>Literatura .....</b>	<b>62</b>



## **Seznam zkratek**

PLC	Programmable Logic Control
ČSN	Česká technická norma (dříve Československá státní norma)
HMI	Human machine interface
ISO	International Organization for Standardisation
P/D	Pulse/direction
HSC	High speed counter

## Seznam obrázků

Obrázek 1 - Soustruh a princip soustružení [VIII] .....	6
Obrázek 2 – Frézy [VIII] .....	7
Obrázek 3 - Princip pohybu frézy [VIII] .....	7
Obrázek 4 – Vrtačky [VIII] .....	8
Obrázek 5 - princip pohybu vrtačky [VIII].....	8
Obrázek 6 - Princip broušení na kulato [VIII] .....	9
Obrázek 7 - přehled způsobů broušení na kulato.....	10
Obrázek 8 - průřez servomotorem [4].....	19
Obrázek 9 - charakteristika servomotoru [4] .....	19
Obrázek 10 - Struktura Brusného kotouče.....	23
Obrázek 11 - princip broušení zápichem .....	24
Obrázek 12 - Zapichový cyklus .....	25
Obrázek 13 - funkce bezhrotého broušení .....	27
Obrázek 14 - vada obrobku 1 .....	28
Obrázek 15 - vada obrobku 2.....	28
Obrázek 16 - vada obrobku 3.....	28
Obrázek 17 - bruska typ A.....	29
Obrázek 18 - bruska typ B .....	29
Obrázek 19 - bruska typ C .....	30
Obrázek 20 - popis obráběcího stroje BBE1A.....	30
Obrázek 21 - ovládací panel .....	33
Obrázek 22- základní obrazovka HMI.....	34
Obrázek 23 - nastavení brusného kotouče .....	35
Obrázek 24 - pinola servis .....	36
Obrázek 25 - orovnávač brusného kotouče.....	36
Obrázek 26 - měřidlo nastavení .....	37
Obrázek 27 - přihlašovací obrazovka .....	38
Obrázek 28 – PLC.....	40
Obrázek 29 - tag tabulka.....	41
Obrázek 30 - řízení SrM1 .....	43
Obrázek 31 - propojení PLC .....	44
Obrázek 32 – metoda GET .....	45

Obrázek 33 – příklad měřidlo .....	49
Obrázek 34 - numerická klávesnice.....	55

## **Seznam tabulek**

Tabulka 1 - Topologie sítě.....	44
Tabulka 2 - nastavení měřidla.....	48
Tabulka 3 - tabulka kódování JOG.....	52
Tabulka 6 - přepínač JOG.....	53

# 1 Úvod

Téma diplomové práce je zaměřeno na programování řídicích jednotek neboli PLC. Tato práce je zaměřena na PLC značky Siemens a toto PLC bude řídit obráběcí stroj.

Cílem diplomové práce je návrh, vývoj a implementace řídicího systému pro obráběcí stroj, který obrábí materiál formou broušení.

Obráběcí stroj je v počáteční fázi řízen jen relátky a posuv brusných kotoučů a podávacího stolu je řešen hydraulickým olejem. Posuv Brusných kotoučů a podávacího stolu se zamění za krokové motory a jejich pohony za účelem zpřesnit pohyb s přesností 0,001mm a ulehčit životnímu prostředí tím, že se nebude používat hydraulický olej. Etapy tvorby řídicí jednotky:

\* Stávající relátka se zamění za řídicí jednotku PLC za účelem náhrady relátek, které se časem opotřebují. Výhodou PLC je tak možnost naprogramování složitějších systémů za využití interních časovačů, čítačů a funkcí pro řízení krokových motorů.

\* V druhé fázi jde o naprogramování řídicí jednotky PLC spolu s pohony pro pohyby. Hodnoty pro obrábění, přepínání mezi druhy broušení a nastavování obráběcího stroje bude realizováno dotykovým panelem HMI. Odměrování polohy bude realizováno digitální indikací.

\* Ve třetí fázi jde o naprogramování HMI panelu, který komunikuje s PLC přes sériovou linku. Programování panelu je rozděleno do dvou částí a to grafická neboli vizuální část a komunikační část pro komunikaci s PLC pomocí maker. Výsledkem práce bude návrh NC obráběcího stroje. Tabulky a obrázky bez odkazu na citaci, jsou čerpané z vlastního zdroje.

## 2 Teoretická část

### 2.1 Co je to obráběcí stroj

Je to specializovaný stroj pro obrábění materiálu. Nejběžnější jsou stroje na obrábění kovů a dřeva za nimi následují stroje na obrábění skla, umělé hmoty a dalších materiálů. Základem je tak zvané třískové obrábění. Proto mezi obráběcí stroj patří soustruhy, brusky, frézy atd. Nepatří do této skupiny lisy, válcovací stolice, ohýbačky, nůžky apod.

Původní rozdělení obráběcích strojů bylo podle hlavního řezného pohybu:

- Přímočarý
- Otáčivý

Z důvodu existence strojů, u kterých nelze určit hlavní pohyb, bylo prvotní rozdělení zanecháno a stroje se dne dělí na:

- Univerzální
- Speciální
- Jednoúčelové

Univerzální stroje jsou takové, na kterých se obrábí různými operacemi. Typickým stroje je fréza (vrtání, vyvrtávání frézováním, obtáčení, soustružení). U některých je směrodatný počet realizovatelných operací např. u konzolové frézky jde vždy jen frézování, nicméně volbou nástrojů a polohovacích přípravků může mít výsledný produkt dost složitý tvar.

Speciální stroje jsou ty, na kterých se obrábějí jedinou operací plochy stejného druhu na obrobkách různých druhů a velikostí. Příklad takového stroje je bruska na kulaté tyče.

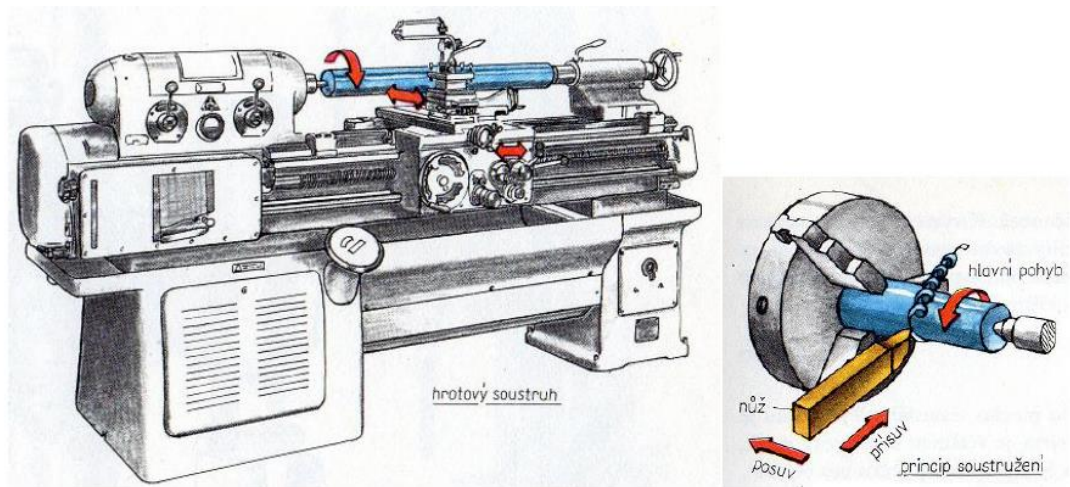
Jednoúčelové stroje jsou určeny pro stejné operace na stejném obrobku. Jsou určeny pro hromadnou výrobu nebo specializované technologie.

### 2.2 Rozdělení obráběcích strojů

#### 2.2.1 Soustružení – hrotový soustruh

Obrábění vnitřních nebo vnějších rotačních ploch, většinou jednobřítým obráběcím nástrojem, při kterém obrobek koná hlavní pohyb do řezu a nástroj pohyb vedlejší. Hlavním pohyb při soustružení je rotace kolem své osy.

Soustružením je možno obrábět vnitřní i vnější válcové plochy, zarovnání čel, zápichy, řezání závitů.



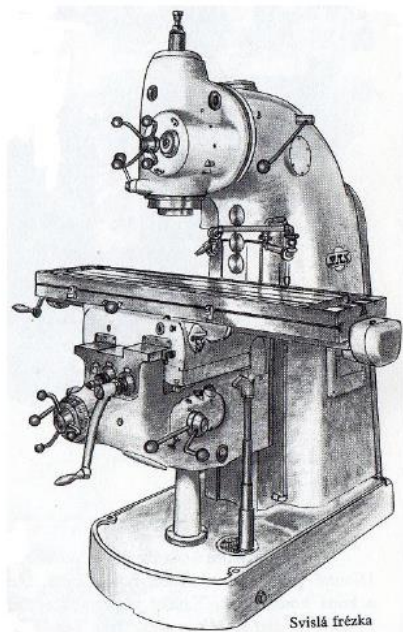
Obrázek 1 - Soustruh a princip soustružení [VIII]

### 2.2.2 Frézování

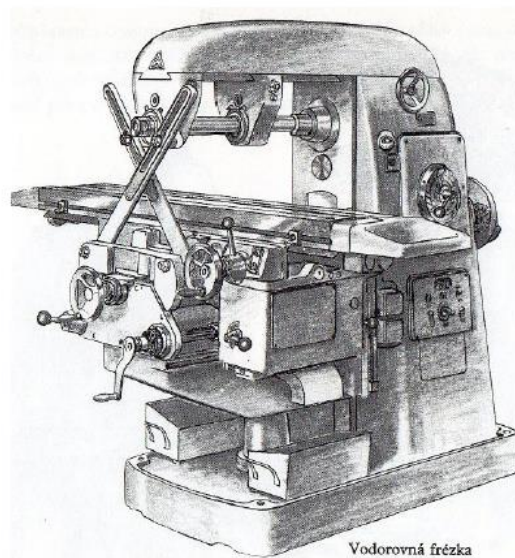
Obrábění rovinných nebo tvorových ploch vícebřítým nástrojem. Řezný pohyb je složen ze dvou pohybů. Posuvný pohyb je většinou přímočarý a hlavní pohyb je rotační. Jednotlivé břity jsou v záběru jen po určité době otáčky a tím pádem odebírají třísku proměnného průřezu.

Dva způsoby frézování:

- Frézování válcovou frézou
- Frézování čelní frézou

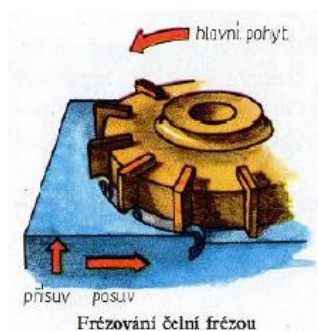


Svislá frézka

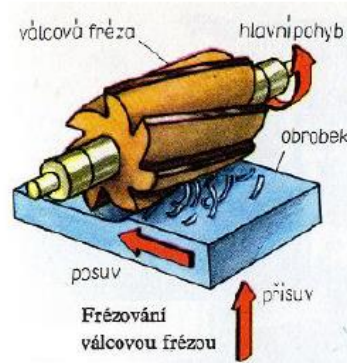


Vodorovná frézka

Obrázek 2 – Frézy [VIII]



Frézování čelní frézou



Frézování válcovou frézou

Obrázek 3 - Princip pohybu frézy [VIII]

### 2.2.3 Vrtání

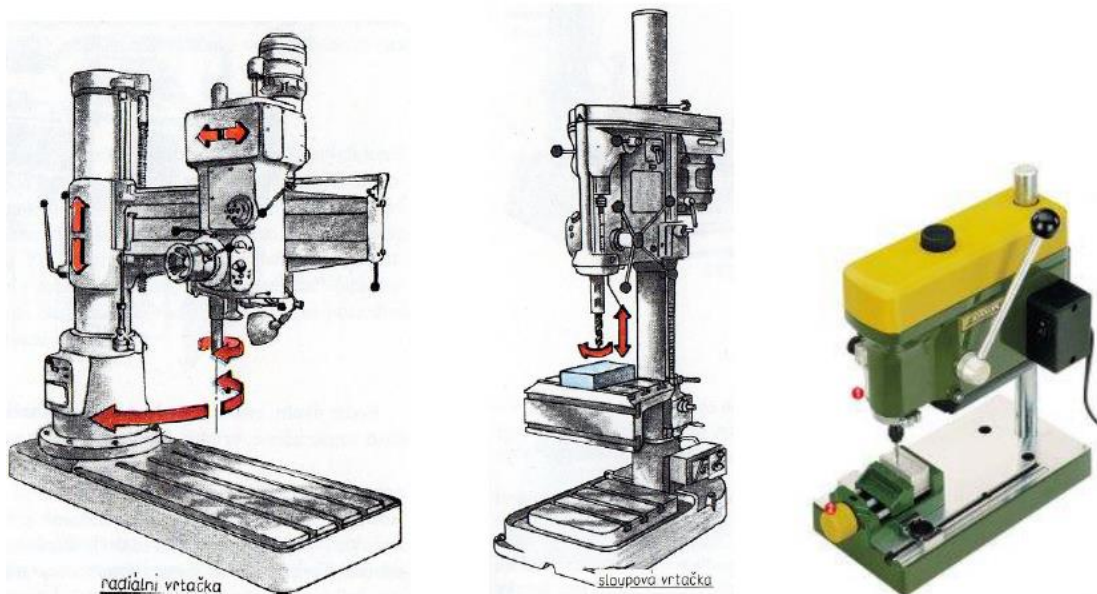
Obrábění vnitřních rotačních ploch neboli děr do plného materiálu. Zpravidla bývá vrtání dvoubřítým nástrojem. Hlavní pohyb je rotační a provádí ho nástroj. Vedlejší pohyb je taktéž rotační a provádí ho také nástroj.

Rozdělení vrtaček:

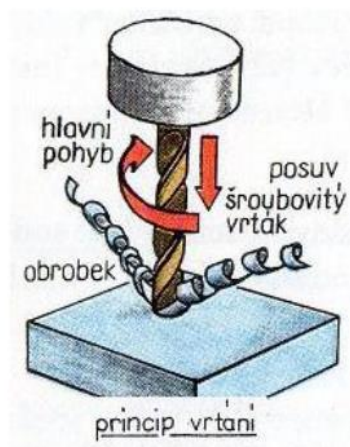
- Stolní
- Sloupová
- Radiální
- Vodorovná



- Aku
- Ruční



Obrázek 4 – Vrtačky [VIII]



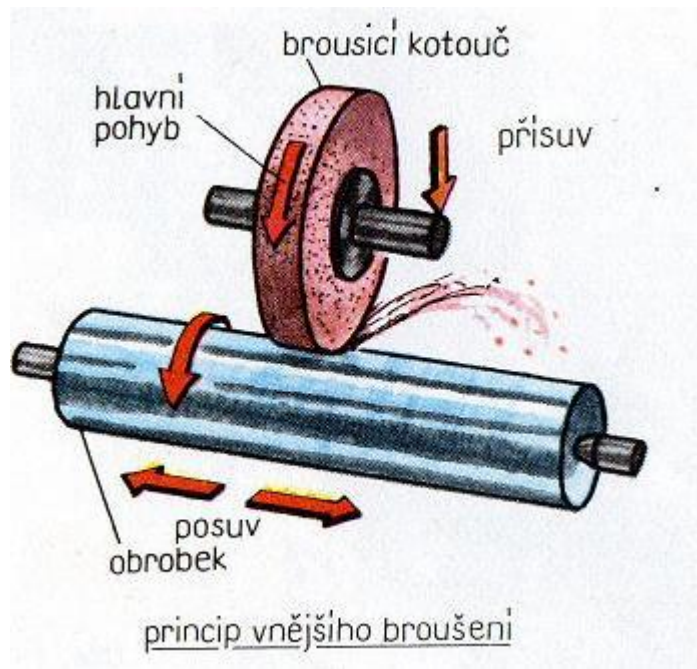
Obrázek 5 - princíp pohybu vrtačky [VIII]

### 2.2.4 Broušení

Broušení je technologie obrábění, při kterém se materiál obrábí tvrdým brusivem. Při strojním broušením se dosahuje vysoké přesnosti a velmi kvalitního a hladkého povrchu bez kazů, jako jsou například spirálové mapy na obrobeném kusu.

Broušení na strojích podstatně urychlí a usnadní finálový obrobek. Princip je takový, že se otáčivý brusný kotouč přibližuje k obrobku na přesnou míru.

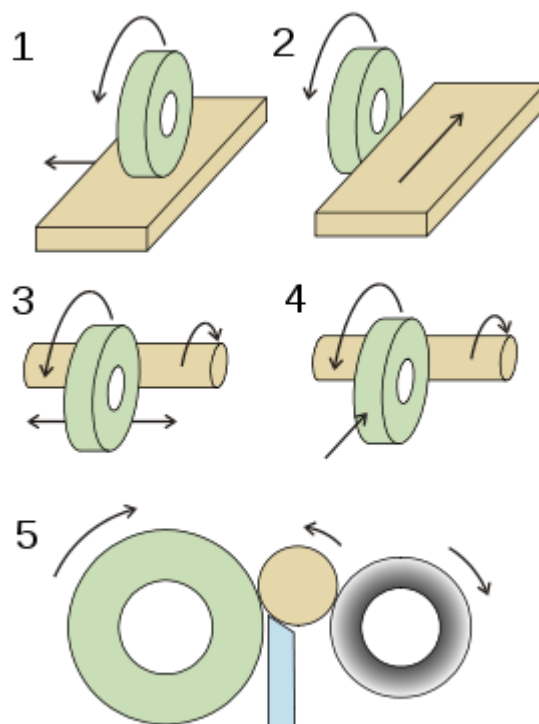
V obecném i přesném strojírenství se používají brusky na plocho, na kulato nebo brusky pro ostření nástrojů. Pro naši práci budeme vyvíjet řídicí systém pro brusku, která brousí bezhrotě na kulato.



Obrázek 6 - Princíp broušení na kulato [VIII]

Typy brusných operací:

1. Rovinné broušení obvodem kotouče
2. Boční broušení čelem kotouče
3. Průběžné broušení na kulato
4. Zápichové broušení na kulato
5. Bezhraté broušení na kulato



Obrázek 7 - přehled způsobů broušení na kulato

## 3 Použité technologie

### 3.1 Řídicí systém

#### 3.1.1 PLC

Zkratka PLC znamená Programmable Logic Controller je malý průmyslový počítač používaný pro automatizaci procesů v reálném čase – řízení strojů nebo výrobních linek. Původně byly PLC určeny jako náhrada za pevnou reléovou logiku. Tomu odpovídal i programovací jazyk prvních PLC automatů. První programovací jazyky disponovali několika příkazy, kterými byli spínací a rozpínací kontakty, paralelní a sériové řazení, jednoduché čítače a časovače. V současné době má téměř každý programovatelný automat k dispozici několik programovacích jazyků. Kromě jazyka kontaktních schémat mají automaty jazyk logických schémat, mnemokódů, instrukčních listů a jazyk sekvenčního programování.

### 3.1.2 Programování PLC

#### **Základní jednotka:**

Základní jednotka má v sobě mikroprocesor a dodává automatu inteligenci. Realizuje systémové služby, souboru instrukci, komunikace s dalšími rozšiřovacími moduly a nadřazenými systémy. Paměťový prostor je obvykle rozdělen do dvou částí. První část slouží k uložení uživatelského programu, datových bloku a tabulek. Druhá část je zápisník. Jsou zde uloženy registry, čítače, časovače, obrazy vstupů a výstupů, časové a systémové proměnné. Obsah operační části se dynamicky mění.

Pro komunikaci s okolím je PLC vybaven vstupními periferiemi (vstupy) na které jsou přivedeny signály v binární podobě (např. snímání polohy koncovým snímačem) nebo v podobě spojitých analogových signálů (například teplota, tlak, hladina ...) . PLC má také výstupní periferie (výstupy), které slouží pro řízení. Jsou na ně připojeny akční prvky řízeného procesu, opět v podobě binárního řídicího signálu (např. stykač motoru, cívka ventilu, atp.) nebo spojitého výstupního řídicího signálu analogové veličiny (např., pro řízení rychlosti, polohy regulačního ventilu, atp.).

Centrální jednotka na základě stavu vstupů ovládá výstupy tak, aby bylo dosaženo minimální odchylky od žádaného nebo zadaného stavu celého zařízení. PLC reaguje na změnu stavu vstupních signálů podle toho, jak programátor napíše řídicí program. Řídicí program je programový algoritmus řešení zadané úlohy (zkráceně program), který uloží do paměti PLC. Operační systém pak zajistí, aby byl program opakovaně (cyklicky) vykonáván.

Mimo základních periferií je PLC vybaveno rozhraním (interface) pro komunikaci s programátorem, stejným nebo dalším rozhraním pro komunikaci s obsluhou, nadřazeným systémem nebo například dotykovým panelem HMI. Toto rozhraní je buď řešeno sériovou linkou nebo komunikace po sběrnici nazývané CAN nebo po ethernetovém portu. Ethernetový port nám může dále sloužit pro připojení PLC do sítě, kdy může komunikovat s dalšími PLC, periferiemi, obecně systémy v hierarchii sítě.

#### **Základní cyklus PLC**

Je to základní znak průmyslového řídicího systému, PLC opakovaně neboli cyklicky provádí svoji činnost.

Na začátku každého cyklu je vykonání systémových operací, které uživatel nemá možnost ovlivnit. Jedná se o vnitřní kontrolu, komunikaci s programátorem, manipulaci s programovými bloky a další.

Další krok je synchronní načtení stavu dostupných vstupních periférií a uložení tohoto stavu do paměti, která je označována jako **obraz procesních vstupů**. Pokud je pak v programu odkaz na nějaký vstupní, je jeho hodnota převzata právě z tohoto obrazu vstupů.

Obraz se vytváří proto, aby během jednoho cyklu programu byla zajištěna jednoznačnost stavu vstupního signálu. Mohlo by se například stát, že kdyby se program během cyklu pokaždé dotazoval na právě aktuální stav vstupu, tak na začátku programu byla hodnota konkrétního vstupu v logické jedničce, ve středu programu logické nule a na konci programu opět v logické jedničce. Z toho plyne, že by mohlo dojít ke zpracování několika logických podmínek během jednoho cyklu s různými stavy vstupního signálu.

Po načtení stavu vstupních signálů je zahájeno **zpracování programu**, uloženého v paměťové oblasti. Program je zpracováván shora dolů. Pro chod programu se využívá pomocná paměť pro ukládání dočasných výsledků, paměť pro lokální proměnné, platné pouze po určitou dobu a nezbytný zásobník hloubky vnoření pro uložení volání podprogramů, návratových adres, obnovu registrů aj.

Během programu se dle algoritmu generují výstupní signály. Stav signálů ukládají do paměti zvané **obraz procesních výstupů**. Důvod je stejný jako u vstupních signálů, a to zajištění jednoznačnosti stavu. Po dokončení programu, je požadovaný stav výstupů synchronně zapsán na fyzické výstupy periférií.

Některé větší PLC musejí mít samostatnou řídicí logiku periférií, tj. Periphery Bus Controller, který může pracovat samostatně, ale přesto je synchronizován se systémem CPU, to znamená, že když je zpracování programu rychlejší než načtení aktuálního stavu všech periférií, vyčká systém na dokončení aktualizace stavu. Množství i vzdálenost periférií ovlivňuje rychlost předávání signálů i synchronizace mezi nimi.

### **Základní doba**

Základní doba je doba, za kterou PLC zpracuje všechny nutné operace pro řízení procesu.

Je to časový interval, za který PLC provede všechny potřebné operace ke zpracování programu a vrátí se do stejného bodu činnosti. Tento bod musí být definován a je to čas kdy je dokončena inicializace CPU.

Do základní doby cyklu se počítají činnosti:

1. Aktualizace interních proměnných systému
2. Aktualizace stavu časovačů
3. Aktualizace stavu vstupů
4. Uložení obrazu vstupu
5. Zpracování programu
6. Uložení obrazu výstupů
7. Aktualizace stavu výstupů

Tato doba pak udává nejkratší dobu mezi změnou stavu signálu na vstupu a reakci na výstup. V praxi záleží na tom, jak je napsán program a jaké další události budou působit na PLC.

### **Vzorkování signálu**

Vzorkováním se zajišťuje stav signálů v procesu. Cyklus a i vzorkování je vzhledem k řízenému procesu vždy asynchronní, musí být četnost vzorkování signálu dostačující, aby mohl být spolehlivě zjištěn jeho stav.

- Optimální vzorkování

Optimální stav vzorkování znamená četnost změn signálu je taková, kdy PLC spolehlivě zjistí každý ze stavů **minimálně 2x**, tedy z každého stavu signálu pořídí minimálně dva vzorky. Doba trvání každého stavu signálu je tedy delší než cyklus PLC a tím i perioda vzorkování. Optimální vzorkování je to proto, že stav signálu je spolehlivě zjištěn a přitom je využita maximální možná frekvence signálu.

- Chybné vzorkování

Chybné vzorkování je, když v jednom případě dojde ke ztrátě informace o skutečném stavu signálu. Tento stav pak vede k chybné funkci celého. Postupem času se zvyšuje výkon CPU i zrychluje taktovací frekvence, tedy tím se za normálního stavu zvyšuje četnost vzorkování.

- Trvalé výpadky signálu

K trvalým výpadkům signálu oproti reálnému stavu dochází v případě, kdy doba trvání stavu signálu je výrazně menší, než jeden cyklus, tedy perioda vzorkování stavu signálu. Oproti chybnému vzorkování, kdy je možno programovým algoritmem odhadnout stav signálu, v tomto případě je převážná část stavu signálu systémem zcela ignorována.

- Občasné výpadky signálu

Perioda vzorkování, stejně jako cyklus není konstantní a je ovlivněna vytížením, které je zpravidla proměnné.

### **Reakční doba**

Reakční doba je čas, který uplyne od okamžiku změny sledovaného signálu až po okamžik odpovídající reakce systému na výstupu.

#### 1. Optimální reakční doba

Je závislá na vytížení CPU, dobou cyklu a vzorkování.

Příklad pracujeme s jednoduchým přenosem signálu ze vstupu na výstup s ideálním vzorkováním. Signál ideálně změní stav na logickou jedničku těsně před okamžikem vzorkování, vznikne minimální prodleva zachycení a je předán do programu ke zpracování. Stav změny je zpracován a přenesen na odpovídající výstup. Stejně tak probíhá přechod do stavu logické nuly. Z toho vyplývá, že reakční doba je:

$$T_{\text{REACT}} = T_{\text{CYCLE}} + T_{\text{PRE}}$$

a doba trvání signálu na výstupu:

$$T_{\text{OUT}} \approx T_{\text{IN}}$$

není zatížena časovou chybou oproti signálu na vstupu, která by byla větší než doba cyklu.

$$\Delta T < T_{\text{CYCLE}}.$$

Zatímco vstupní signál může a vzniká, asynchronně k cyklu PLC, výstup je synchronizován s cyklem PLC.

#### 2. Odchylka od reakční doby

Odchylkám v reakční době se nedá vyhnout. V určitých případech se tomuto stavu nedá předcházet a nastává. Je však nutno na tyto odchylky brát zřetel. Vzniká například u snímačů polohy pro detekci polohy využívající kódování dvojkové soustavy nebo BCD kód.

V optimální reakční době jsme signál optimálně vzorkovali, tj. 2 vzorky v průběhu trvání stavu logické jedničky a dojde ke změně periody vzorkování nebo se změna stavu signálu logické jednotky opozdí za okamžikem vzorkování, je tato změna zaznamenána až při dalším vzorkování. Ke ztrátě impulsu signálu nedojde, ale nastane zkreslení trvání stavu signálu na výstupu oproti době trvání signálu na vstupu.

### 3. Opoždění detekce ukončení signálu

Opačná situace od předchozího případu. Ukončení stavu signálu logické jedničky se o málo opozdí za okamžikem vzorkování nebo dojde k posunu okamžiku vzorkování. Konec signálu je zpracován až po okamžiku vzorkování. Ke ztrátě impulsu signálu sice nedojde, ale dojde k prodloužení doby trvání stavu signálu.

## **Zpracování přerušení**

Přerušení je stav, kdy vznikne situace, při které je požadováno aktuální provedení okamžité nebo žádné reakce na tento stav. Přerušení se dělí do dvou skupin a to na uživatelské a systémové.

Uživatelské přerušení vzniká na základě programátora. Programátor definuje, za jakých podmínek může nastat.

Systémová přerušení jsou generována na základě vzniku systémové události, například chyby, požadavku na diagnostiku atd. Programátor nedefinuje, kdy a jak vzniknou ale má možnost tato přerušení povolit nebo zakázat a zvolit vhodnou reakci na dané přerušení. Například při přerušení kdy nastane chyba vypsát číslo chyby a co znamená na display pro jednodušší odstranění chyby.

Pokud není definována reakce na vznik chybového stavu, přejde PLC do stavu STOP, tedy do bezpečného stavu, kdy vzniklá chyba nemůže ovlivnit řízený proces v tom smyslu, že by se zpracovávaly například neplatné hodnoty měření nebo by nebylo možno vlivem závady periferie ovládat část zařízení.

### Typy uživatelského přerušení:

#### 1. Hardwarové / procesní přerušení

Toto přerušení reaguje na události procesu, které přicházejí od hardwaru / periférii PLC. Například na změna digitálního signálu, překročení limitu analogové hodnoty, která bývá dle



standardu 0 – 10V, přerušení od hardwarového čítače to znamená naplnění maximální hodnoty registru. Při tomto přerušení musí být periferie schopné generovat požadavek přerušení při konkrétní události.

## 2. Cyklické přerušení

Cyklické přerušení, jak z názvu vyplývá, je prováděno v pravidelném cyklu, která je mu přidělena. Přerušení s nastavenou kratší periodou mají vyšší prioritu. To je z důvodu toho že by mohlo docházet k neobsloužení přerušení s kratší periodou.

## 3. Kalendářní přerušení

Toto přerušení generuje volání konkrétní rutiny dle nastaveného data. Počátkem je první nastavení data a času. Poté se dále pakuje dle nastavení v řádu minut, hodin, dnů, měsíců, roků.

## 4. Synchronní cyklické přerušení

Synchronní přerušení se provádí synchronně v konstantní periodě, která cykluje. Používá se například pro komunikaci po sběrnici CAN nebo pro obsluhu periférií připojených na síti PROFIBUS nebo PROFINET IO.

## 5. Časově zpožděné přerušení

Toto přerušení nám dává možnost provést odložení zpracování části programu od okamžiku přijetí požadavku o předem nastavenou dobu.

Typy systémových přerušení:

### 1. Přerušení DPV0, DPV1,DPV2

DPV přerušení jsou generována podřazenými částmi systému. DP znamená decentralními perifériemi směrem k nadřazenému CPU, který je označen jako master a požaduje, aby se v CPU provedla rutina obsluhy přerušení. Charakter přerušení je různý podle verzí. Nabízí nám verze V0, V1, V2.

### 2. Asynchronní chybové přerušení

Asynchronní chybové přerušení reaguje na vznik chybové události, například chyba.

### 3. Diagnostické přerušení

Diagnostické přerušení je generováno částí systému PLC. Může se jednat nejen o hardwarovou závadu ale i o chybu zpracování instrukcí v programu. Na základě tohoto přerušení systém vyžaduje diagnostiku.

Počet zpracovávaných přerušení je různý a závisí na velikosti zásobníku přerušení. Do zásobníku přerušení se ukládají příchozí požadavky na přerušení. Levné a jednoduché CPU poskytují možnost zpracování 2 - 4 přerušení, střední CPU 8 přerušení a výkonné CPU mohou zpracovat až 16 přerušení.

Ze zásobníku jsou přerušení vybírána dle dané priority. Každé přerušení má přiřazenu určitou prioritu a na jejím základě se přerušení zpracuje. Pokud se stane, že v jednom okamžiku dojde k výskytu více požadavků na přerušení tak jako první se zpracuje přerušení s nejvyšší prioritou. Paradoxně je obvykle označována jako 0. Toto označení znamená, že tento proces je neodkladný. Přerušení s nižší prioritou tedy označena 1 a výš jsou zpracována až po dokončení a zpracování přerušení s vyšší prioritou. Pokud již probíhá zpracování přerušení, může být přerušeno požadavkem zpracování přerušení s vyšší prioritou. Tyto přerušení se odloží a jsou obsloužena nebo dokončena po dokončení zpracování přerušení s vyšší prioritou.

Za zpracování přerušení je odpovědný systém CPU. Ten zpracovává požadavky na přerušení podle dané priority a pořadí pokud by nastalo, že bude více přerušení se stejnou hodnotou priority a dle toho volá zpracování obslužných rutin.

Rutina obsluhy libovolného přerušení by měla být co možná nejkratší a nejjednodušší. Žádné zpětné skoky, žádné cykly, žádné nadbytečné volání podprogramů z přerušení. V případě, že je rutina obslužení moc dlouhá, může dojít k tomu, že by nastala stejná událost, která požadovala vyvolání již prováděné rutiny obsluhy přerušení. Rekurzivní neboli zpětné volání rutin není povoleno. Pokud tak nastane tak to končí chybou, kdy je PLC převedeno do bezpečného stavu STOP. Do stejného stavu se PLC dostane, pokud je přetížen systém častým voláním přerušení, kdy systém tuto zátěž již nestačí zpracovat nebo dojde k nadměrnému prodloužení doby cyklu.

V praxi mohou však nastat případy, kdy se zpracovává úsek programu, který může být v přerušení použit anebo toto přerušení může naopak změnit hodnoty proměnných použitých při výpočtu. V takovém případě by mohl být výsledek různý. V tomto případě se používá zákaz přerušení kdy je možno reakci na přerušení odložit nebo upravit používání proměnných tak, aby k této situaci nedošlo.

Pro realizaci bylo použito PLC značky Siemens. Programovací prostředí Tia-Portál je velmi intuitivní pro programování, ale nevýhodou je velká hardwarová náročnost na počítač. Toto prostředí umožňuje programovat ve schémátovém jazyku, strukturovaným textem a instrukčním listem.

## **3.2 Ovládání**

Základní ovládání je řešeno tlačítky. Z důvodu velkého množství tlačítek bylo nutné provést komunikaci s PLC přes speciální kódování zvané multiplex o kterém se dozvíme v praktické části práce.

Nastavování brousícího stroje je pomocí dotykového panelu HMI.

### **3.2.1 HMI**

Zkratka HMI znamená (Human Machine Interface) rozhraní pro ovládání stroje. Operátorské panely umožňují a usnadňují přístup obsluhy k datům a ovládání technologie řízené programovatelným automatem PLC.

## **3.3 Realizace posuvu**

Posuv brusných kotoučů je realizován kuličkovým šroubem, který je přes tuhou spojku spojen se servomotorem.

### **3.3.1 Servomotor**

Servomotor zkráceně servo je rotační motor u kterého je přesně známá poloha natočení na rozdíl od obyčejného motoru. Poloha otočení hřídele je snímána snímačem, který se nazývá encoder. Ten komunikuje s elektropohonem, který natáčí osu motoru podle zadaných parametrů.

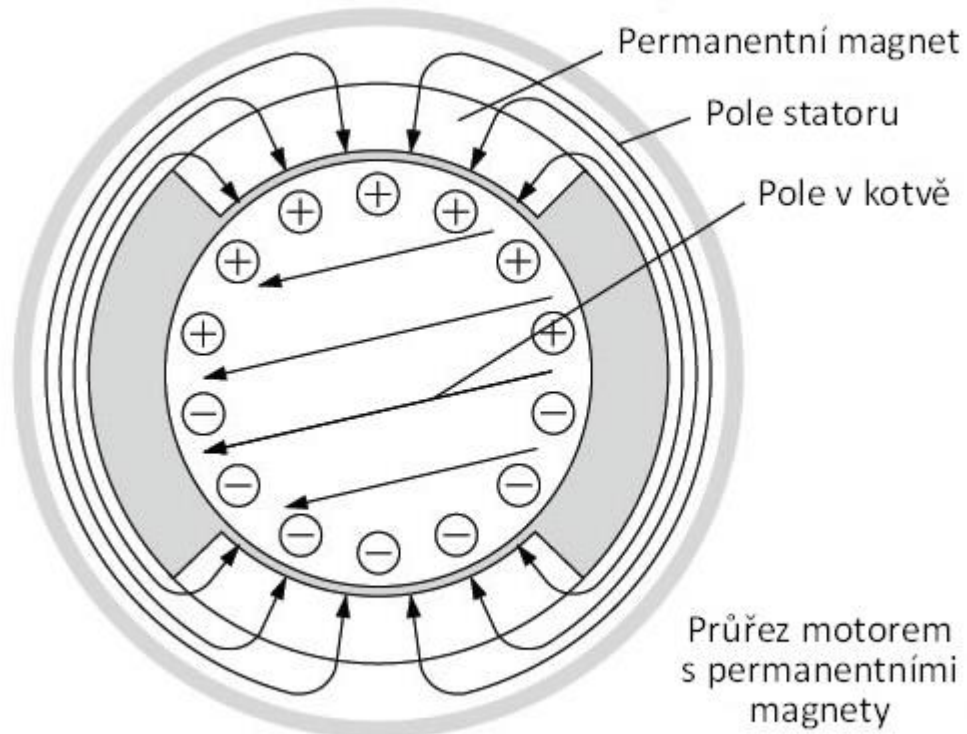
#### **Princip servomotoru**

Pole permanentně magnetizovaného motoru (PM) se tvoří mezi dvěma nebo více trvalými magnety. Nejčastější magnety jsou typy feritových magnetů. PM motor nemá vinutí pro tvorbu pole. Z toho vyplývá, že pro připojení motoru, stačí dva vodiče. Jeho osa se neustále pohybuje o ze strany na stranu a udržuje polohu. Tento pohyb je zanedbatelný.

Každý servomotor musí být připojen k pohonu, který zajišťuje nastavení motoru a komunikaci.

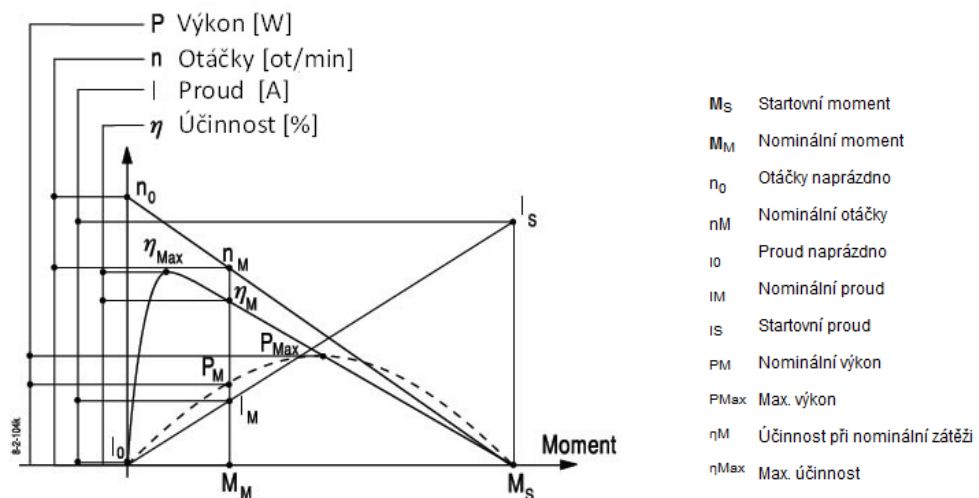
#### **Charakteristika**

Nejvýznamnější vlastností PM motoru je, že při konstantním napětí, jsou otáčky a proud úměrné momentu.



Obrázek 8 - průřez servomotorem [4]

U tohoto motoru je snadné měřit otáčky a točivý moment za provozu. Je důležité, aby byl moment motoru jednou desetinou momentu síly, proti které bude pracovat.



Obrázek 9 - charakteristika servomotoru [4]

Důležité údaje:

- rezistance  $R_A$
- indukčnost  $L$
- setrvačnost  $J_M$

Z diagramu je vidět, že rozběhový moment je obvykle 6-8 krát vyšší než nominální točivý moment. Jmenovitý moment je omezen tepelnými vlastnostmi a může být při občasných zatíženích použit motor s nižším výkonem, než je požadováno. V tomto případě je třeba vzít v úvahu životnost motoru, maximální otáčky motoru v závislosti na zatížení motoru, teplota, atd.

### 3.4 Měření

Měření výsledných kusů probíhá na výstupu z obráběcího stroje. Pro měření se používá optického paprsku. Je to bezkontaktní přesná metoda měření.

#### 3.4.1 Princip optických mikrometrů

Pojem „optický mikrometr“ je používán pro různá měřicí zařízení, která se svou konstrukcí a funkcí velmi liší. Tento optický mikrometr je tzv. thru-beam. Výrobek je osvětlen kolimovaným svazkem paprsků. Svazek paprsků zčásti zastíněný objektem dopadá na přijímač. Jsou používány dva způsoby vyhodnocení.

- První se používá pro jednoduché, ale velmi rychlé měření. Kolimovaný svazek paprsků je soustředěn pomocí čočky na fotodiodu. Podle množství světla dopadajícího na fotodiodu je určena velikost popřípadě hrana.
- Druhým, přesnějším způsobem je použití řádkových snímačů CCD-CMOS. Řádkovými snímači lze z paprsků dopadajících na přijímač získat mnohem více informací o měřených objektech. Oproti prvnímu způsobu je vyhodnocení těchto informací komplikovanější.

Pozice hrany objektu není na profilu jasů jednotlivých obrazových bodů (pixelů) snímače čitelná jednoznačně: hrana stínu objektu na snímači mění svůj jas plynule, především vlivem ohybu světla na hraně měřeného objektu (*obr. 2, obr. 3*). Strmost hrany závisí na vzdálenosti objektu od snímače. Z informace o pořadí jednotlivých pixelů a jejich jasů je možné určit pozice

hran, popř. rozměry objektů před snímačem. Mezi metody hledání hran nejpoužívanější pro řádkový i plošný obraz patří:

- komparační metody (intenzita jednotlivých bodů je porovnávána s nastavenou rozhodovací úrovní),
- metody využívající vlastnosti první a druhé derivace funkce jasu v závislosti na poloze (první derivace nabývá v místě hrany extrému, druhá derivace prochází v bodě hrany nulou).

Při menších požadavcích na přesnost určení rozměru či pozice měřeného objektu se lze spokojit s rozlišením daným rozměrem jednotlivých pixelů snímače (běžně např. 14  $\mu\text{m}$ ). Potom stačí např. každou analogovou hodnotu odpovídající jasu jednotlivých pixelů porovnat pomocí komparátoru (realizovaného např. operačním zesilovačem) s nastavenou analogovou rozhodovací úrovní (např. napět'ovým děličem). Výstup z komparátoru představuje pro každý pixel logickou informaci „objekt je“ – „objekt není“. Druhou možností je digitalizovat signál ze snímače prostřednictvím AD převodníku, a využít tak prostředky i propracovanou metodiku zpracování digitálního signálu, jež jsou v současné době k dispozici. Digitalizované hodnoty jasů jednotlivých pixelů je možné mnohem efektivněji použít k určení pozice hrany s rozlišením významně lepším, než je rozměr jednoho pixelu snímače – tzv. subpixelové rozlišení. Využitím informace z většího množství pixelů sousedících svým jasem s komparační hladinou lze získat i jisté zvýšení imunity naměřené hodnoty proti chybám způsobeným např. šumem v analogovém signálu snímače, nečistotami na optických plochách, šumem osvětlení atd. Se zvyšující se propracovaností použité metody zpracování informace z pixelů na hraně poroste i výpočetní náročnost algoritmu a velikost potřebného hardwaru. [3]

### 3.4.2 Vlastnosti měřidel

- rozlišovací schopnost
- opakovatelnost
- linearita v pracovním rozsahu
- pracovní rozsah měření
- vzdálenost mezi vysílačem a přijímačem
- druh světelného zdroje, popř. jeho výkon
- výstupní rozhraní

Je nutné podotknout, že zmíněné vlastnosti týkající se přesnosti měření mohou být klasifikovány různými metodami, při dodržení různých omezení kladených na prováděná měření. Linearita měření v pracovním rozsahu může být běžně udávána ve zkalibrované oblasti – to je často střední vzdálenost mezi vysílačem a přijímačem, popř. může být vyjádřena v konkrétní obdélníkové výšce. Opakovatelnost měření bývá spojena s nastavenou hodnotou počtu průměrování změřené hodnoty, daným rozměrem testovacího objektu atd.

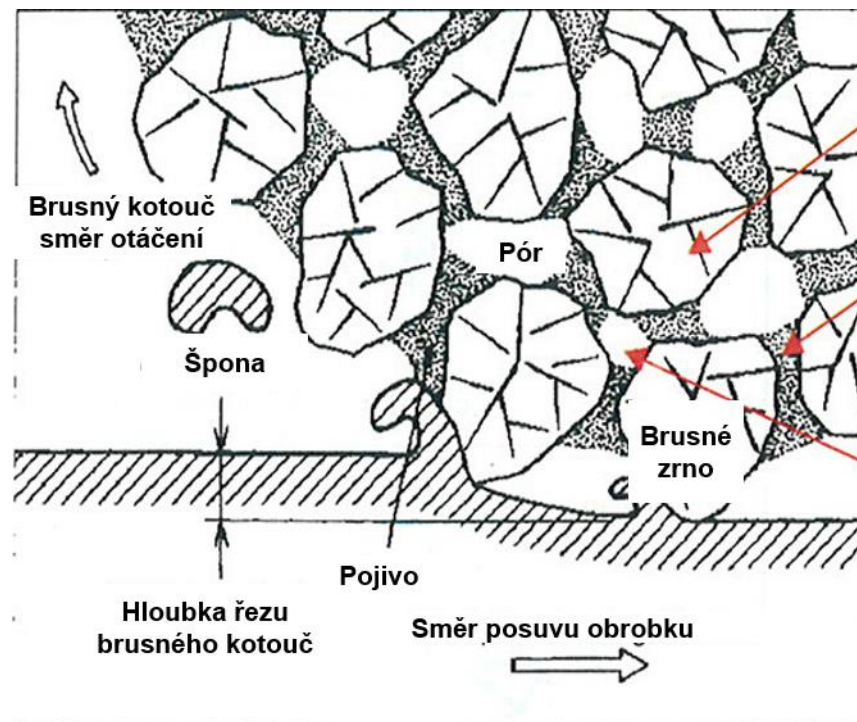
Výhodou měření objektů kolimovaným svazkem paprsků je bezesporu absence fyzického kontaktu s měřeným objektem. Je třeba brát v úvahu i drobné nevýhody vyplývající z podstaty této metody. Výchozím bodem pro dosažení uspokojivých výsledků je kvalita světla dopadajícího na řádkový snímač. Ta může být ovlivněna několika nežádoucími faktory, v podstatě nesouvisejícími se samotným mikrometrem. Jedním z nich je nutnost udržet čistotu vnějších optických ploch vysílače a přijímače. Proti nechtěnému zanechání otisků prstů při neopatrné manipulaci mohou být krycí sklíčka chráněna např. mechanickým odsazením od vnějšího profilu krabičky, což naproti tomu komplikuje jejich čištění. Usazování prachových částic a jiných nečistot na vnějších optických plochách je možné omezit vhodným umístěním mikrometru uvnitř měřicího systému. Tam, kde nelze odstranit riziko znečištění jinak, je třeba před vnější optické plochy instalovat ofukovací systém. Na přesnosti měření se může také negativně projevit nevhodné umístění měřicí hlavy, např. proti oknu, do kterého svítí slunce. Měřicí hlava je opatřena optickými filtry, které propouštějí jen záření o vlnových délkách blízkých záření vysílače, avšak záleží na intenzitě a vlnovém spektru okolních zdrojů, jaký bude poměr užitečného a rušivého záření propuštěných do přijímače. Problém okolních zdrojů záření může být také vyřešen optickou soustavou uvnitř přijímače, která propustí pouze kolimované paprsky z vysílače. Ochrana celého systému proti těmto nežádoucím vlivům může být zvýšena také využitím náročných výpočetních algoritmů, ale za cenu zvýšení komplikovanosti a ceny celého zařízení. [3]

## 4 Broušení

### 4.1 Bezhraté broušení

#### 4.1.1 Definice broušení

Broušení je metoda obrábění, přičemž brusný kotouč se otáčí vysokou rychlostí až čtyřicet metrů za sekundu a je používán k broušení obrobku. Brusný kotouč tvoří extrémně pevná a jemná brusná zrna.



Obrázek 10 - Struktura Brusného kotouče

- Brousne zrno – slouží k řezání materiálu z obrobku
- Pojivo – spojuje brousne zrna k sobě
- Pór – slouží jako úniková zóna pro špony a další funkcí je chlazení brousneho kotouče

Podávací kotouč slouží k brzdění obrobku. Ve většině případů má rychlost přibližně jedem metr za sekundu.

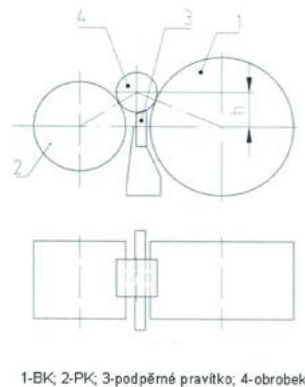


## 4.1.2 Způsoby bezhrotého broušení

### Broušení zápichem

Toto broušení se používá pro broušení geometricky náročných a přesných součástí, na které je vyžadován vysoký nárok na přesnost. Takto se brousí například klikové hřídele do automobilů, válcové čepy s osazením, součásti převodovek, kuličkové šrouby, atd.

Mezi základní znaky patří, že osy obou brousících kotoučů jsou rovnoběžné s osou podpěrného pravítka a zároveň jeden z kotoučů má pohyb v radiálním směru. Obráběný kus nemá osový pohyb. Podle profilu kotoučů a podpěrného pravítka dokážeme brousit daný tvar. Obrobek ho opisuje a tím jsme limitováni velikostí obrobku. Velikost obráběného kusu záleží na velikosti a orovnaní brusných kotoučů.



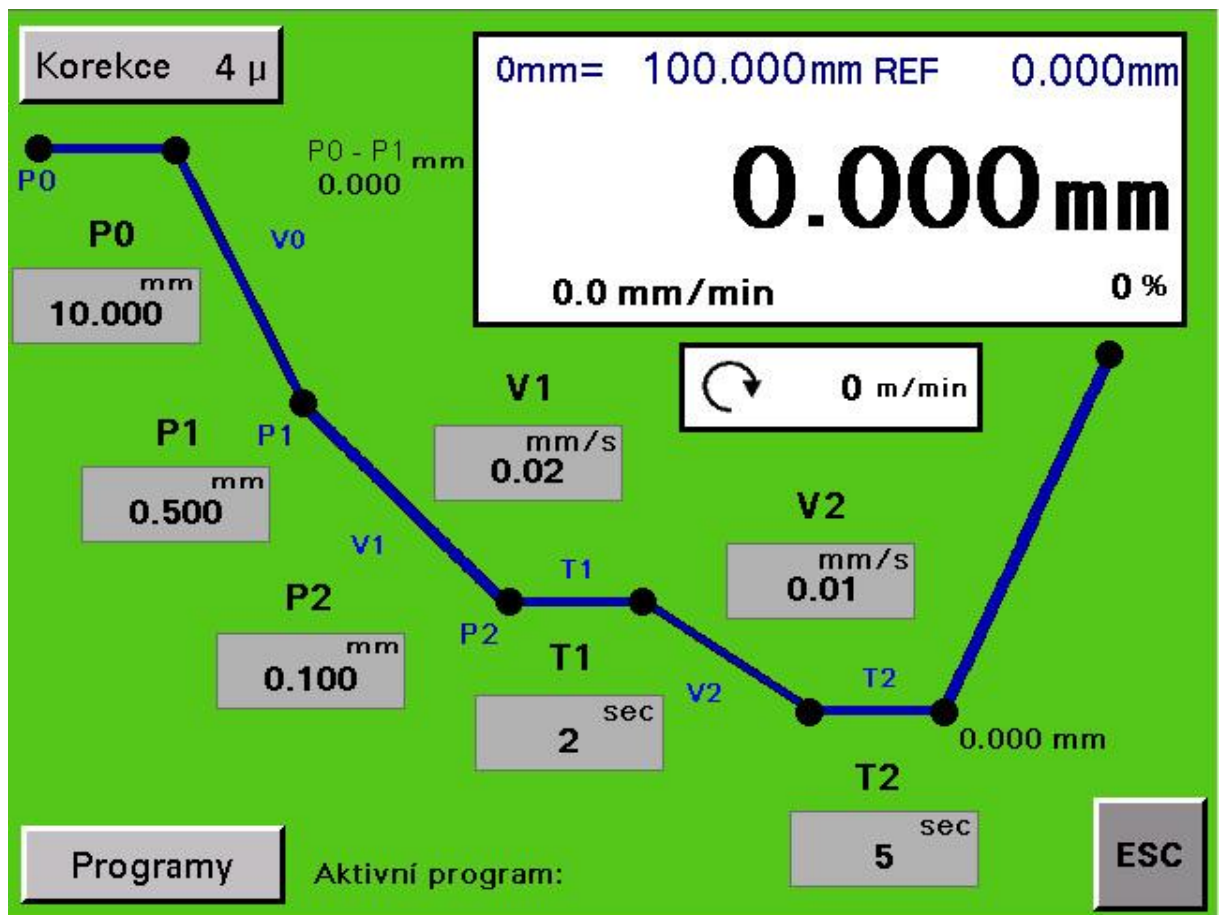
### Obrázek 11 - princip broušení zápichem

Princip zápichového broušení spočívá v přijetí brusného kotouče k obrobku. Nejdůležitější je rychlost příjezdu. Průměrná rychlost broušení je 0,01 mm za sekundu. Záleží na tvrdosti obráběného materiálu, směsi brusného kotouče a velikosti chlazení.

Cyklus zápichového broušení má šest částí:

1. Rychlý příjezd na celkový úběr
2. Hrubý úběr materiálu
3. Vyjiskření po hrubém broušení
4. Jemný úběr materiálu
5. Vyjiskření po jemném broušení
6. Rychlý odjezd na startovací pozici

Graf zápchového broušení pro představu cyklu



Obrázek 12 - Zapichový cyklus

### Nastavení programu:

- P0 – počáteční pozice cyklu -> odjezd z této pozice na hrubovací část je konstantní a nelze měnit parametrem
- P1 – Počáteční pozice broušení. Je to celkový úběr obrobku. (nesmí být nulová a musí být vždy větší nebo stejná než P2)
- P2 – Hodnota do které je posuv vřeteníku prováděn rychlostí V1 neboli hrubovací rychlost a taktéž hodnota od které začíná jemná část broušení S rychlostí V2.(může být nulová ale nesmí být větší než P1) -> Pokud je P2 nulové je jemná část broušení vypnutá a celý cyklus se pohybuje rychlostí V1.
- V1 – rychlost hrubého broušení.
- V2 – Rychlost jemného broušení.
- T1 – Čas vyjiskření po hrubém broušení. Část, při které brusný vřeteník stojí po danou dobu na stejném místě.
- T2 - Čas vyjiskření po jemném broušení. Část, při které brusný vřeteník stojí po danou dobu na stejném místě.
- Korekce – Hodnota korekce slouží pro eliminaci opotřebení brusného kotouče. Stiskem tlačítka **korekce** na ovládacím panelu a stiskem tlačítka se směrem brusného vřeteníku se vykoná korekce.

- Nulová hodnota – je to nejdůležitější parametr programu. Je to nula do které brusný vřeteník přijíždí. Nastaví se vynulováním osy. Od této hodnoty je řízen celý cyklus broušení.

Všechny tyto hodnoty se dají ukládat do programů, které se dají pojmenovat a přepínat dle daného výrobního postupu.

Při broušení zápichem tedy automatického cyklu je nutné mít přepnutý režim posuvu do Automatu. V tomto stavu je zakázán plynulý posuv brusného vřeteníku. Plynulý posuv je povolen jen v manuálním režimu nebo v pauze automatu.

Stiskem tlačítka **Stop cyklu** se cyklus zastaví a posuv zůstane ve stavu, ve kterém bylo stisknuto tlačítko stop.

Stiskem tlačítka **Posuv brusného vřeteníku směrem vzad** se aktivuje nouzové odjetí a brusný vřeteník odjede rychloposuvem do startovací pozice.

Příklad nastavení zápichového cyklu:

P0 – 10.000 mm, P1 - 0.500 mm, P2 – 0.100 mm

V1 – 0.02 m/min, V2 – 0.01 m/min

T1 – 2 sekund, T2 – 5 sekund

Korekce – 0.004 mm

Hodnota referenční nuly je -100.000 mm -> hodnota nuly je od referenčního bodu vzdálena 100.000 mm tzn. 10 cm směrem vpřed k pravítku.

Začátek broušení je v 10.000 mm. Po prvním zmáčknutí **Start cyklu** brusný vřeteník přijede na začátek hrubého broušení konstantní rychlostí. Začátek hrubého broušení je 0.500 mm a bude s rychlostí 0.02 m/min posouvat brusný vřeteník až na hodnotu 0.100 mm. Z toho vyplývá, že hrubý úbytek obrobku je 0.400 mm.

$$\text{hrubý úbytek obrobku} = P1 - P2$$

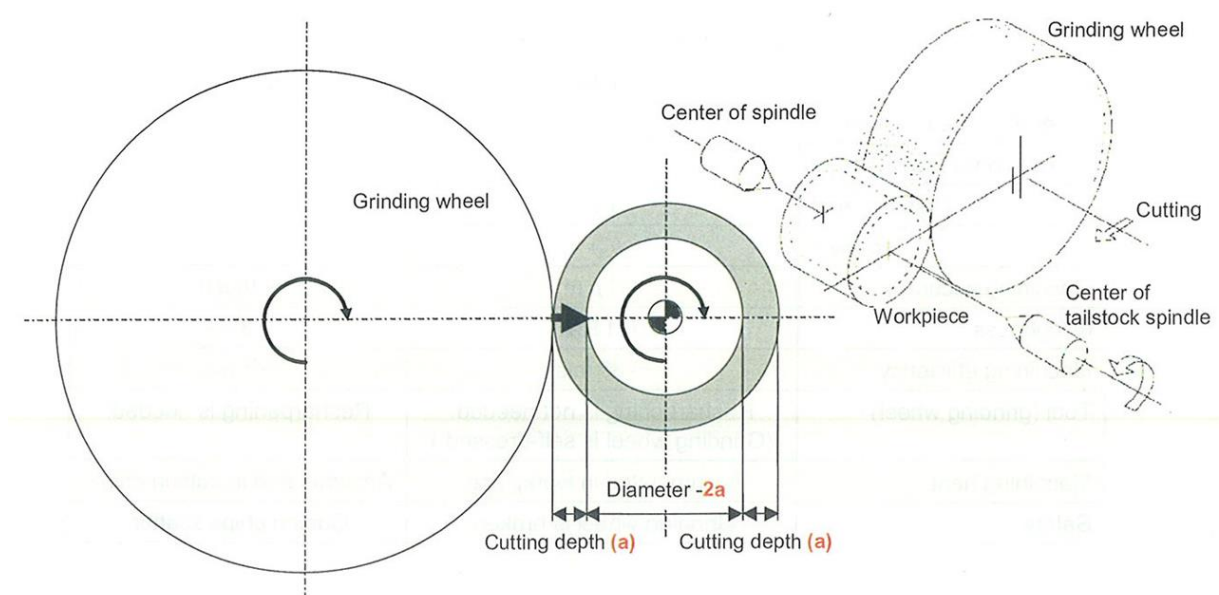
Po hrubém broušení je nastaveno 2 sekund vyjiskření. Po vyjiskření je část jemného broušení. To znamená, že zbytek úběru je 0.100 mm a bude přijíždět do 0.000 mm/s rychlostí 0.01 m/min. Po dobroušení je rychlý odjezd za počáteční polohu našem případě 10.000 mm. To znamená že se brusný vřeteník po dokončení automatu zastaví na pozici -10.000 mm bráno od referenčního bodu a 10.000 m od pravítka.

## Průběžné broušení

Tyto stroje jsou určeny pro broušení přesných válcových výrobků. Nelze měnit broušený průměr během obráběcího procesu. Takto se brousí válcové čepy, hydraulické písty, středové trubky tlumičů automobilů.

Při průběžném broušení jsou osy obou kotoučů nehybné. Nerovnoběžností podávacího kotouče a brusného kotouče se dosahuje osového posuvu obrobku v pracovním pásmu. Brusný kotouč je ve tvaru válce, zatím co podávací kotouč má tvar hyperboloidu. Tím se dosáhne dotyk celé plochy obrobku. U toho druhu broušení může být délka obrobku od několika milimetrů až po několik metrů. Do brusné části jsou obrobky naváděny po naváděcích příložkách a stejným způsobem jsou i odváděny.

### 4.1.3 Funkce bezhrotého broušení



Obrázek 13 - funkce bezhrotého broušení

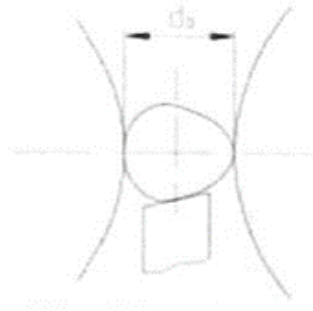
#### Výhody:

- Vysoký výkon broušení
- Vysoká rozměrová a geometrická přesnost obroušených částí
- Odpadá upínání součástí v pracovním prostoru
- Broušení součástí s různým profilem o délce až 550mm na jednu operaci v zápichovém broušení

- Broušení velmi dlouhých částí řádově několik metrů při průběžném broušení

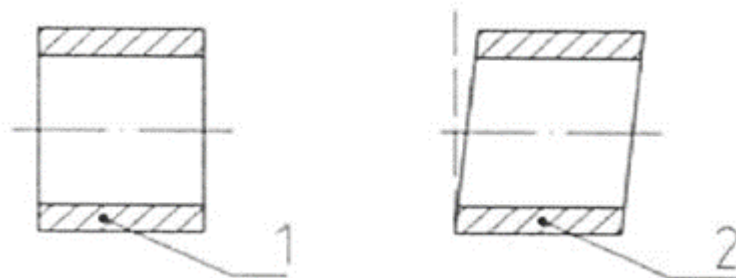
**Nevýhody:**

- Náchylnost na kopírování špatného geometrického tvaru polotovaru (zejména se jedná o nekruhový tvar s nízkým počtem hran)



**Obrázek 14 - vada obrobku 1**

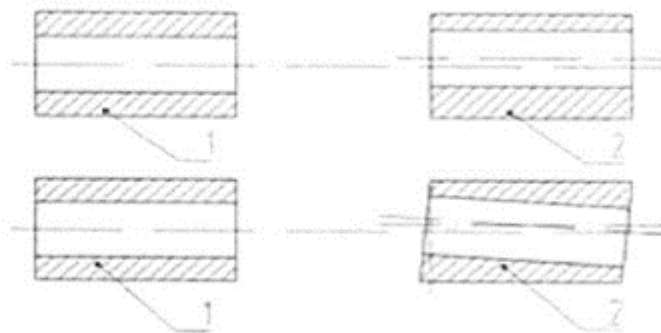
- Může dojít ke zhoršení kolmosti čela k broušenému povrchu



**Obrázek 15 - vada obrobku 2**

1 – před broušením / 2 – po broušení

- Nelze zaručit souosost vnější a vnitřní válcové plochy

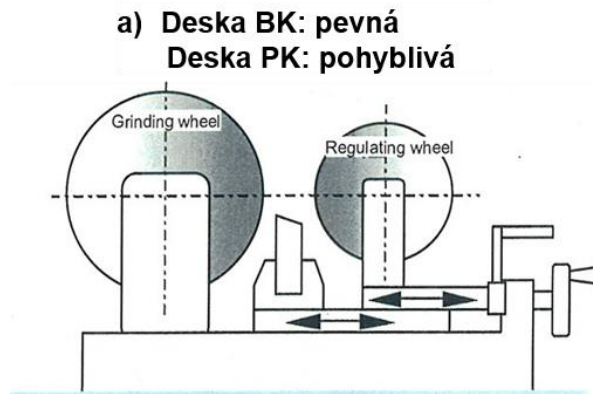


**Obrázek 16 - vada obrobku 3**

- Pro daný tvar součástí je zapotřebí odpovídající podpěrné pravítko (platí především pro zápichový způsob)

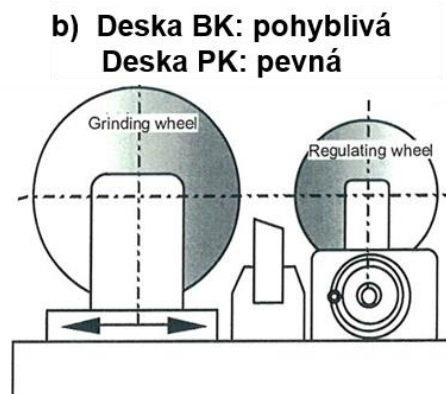
## 4.2 Rozdělení bezhrotých brusek podle posuvů

- a) Deska brousícího kotouče je pevná a deska podávacího kotouče je pohyblivá



Obrázek 17 - bruska typ A

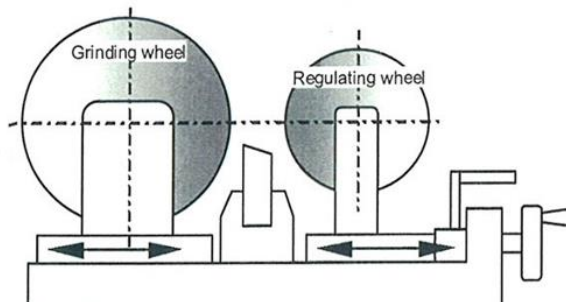
- b) Deska brousícího kotouče je pohyblivá a deska podávacího kotouče je pevná



Obrázek 18 - bruska typ B

- c) Support pravítka je pevný, deska brousícího kotouče je pohyblivá a deska podávacího kotouče je pohyblivá

**c) Saport pravítka: pevné  
BK a PK: pohyblivá**

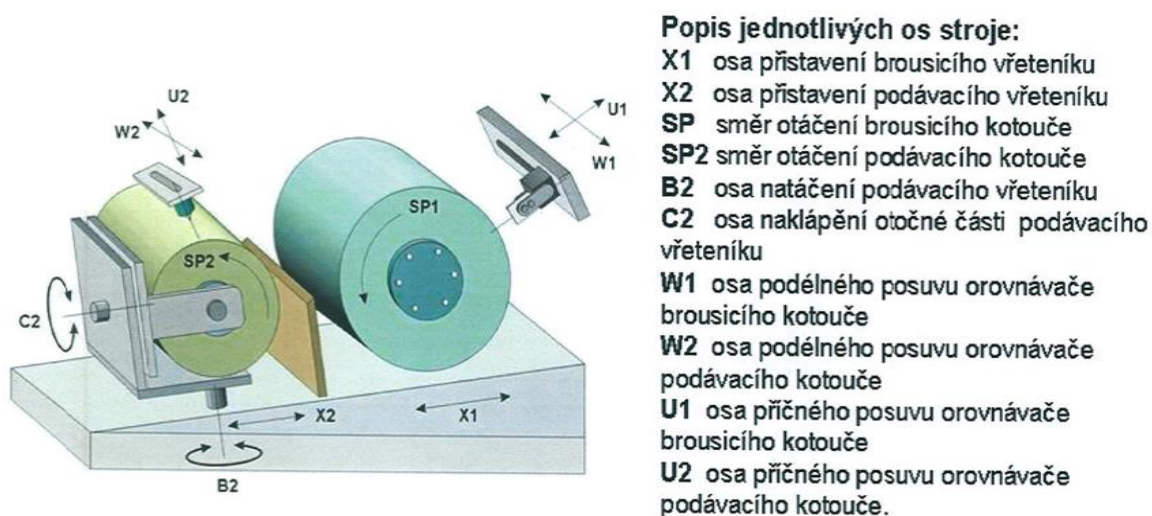


**Obrázek 19 - bruska typ C**

Pro naši práci máme brusku typu C, Kdy oba brusné vřeteníky se posouvají k obrobku. Hlavní brusný kotouč nám udává velikost průměru obrobku a druhý podávací kotouč nám zajišťuje pohyb obrobku z brusného pole stroje. Podávací kotouč má velmi malé otáčky a je nakloněn pod úhlem aby vytlačoval obrobek. Brusný kotouč je v rovině s podpěrným pravítkem a ubírá materiál. Otáčky kotoučů jsou různé. Podávací kotouč se oproti Brusnému otáčí deset krát menší obvodovou rychlostí, to je z důvodu rovnoměrného úběru materiálu po celé jeho ploše.

#### 4.2.1 Popis brousícího stroje

Pro práci jsme použili brousící stroj BBE1A po celkové generální opravě mechanických částí. Naším úkolem je doplnit obráběcí stroj o kompletní elektroinstalaci a naprogramovat kompletní řízení celého obráběcího stroje.



**Obrázek 20 - popis obráběcího stroje BBE1A**

Tato bruska byla vyrobena v roce 1966 v Holicích ve firmě TOS. Posuvy os X1, X2, W1 a W2 byli řešeny hydraulickým olejem. Za postupného přivírání a otevírání ventilů pro hydraulický olej se regulovala rychlost, vzdálenost a přesnost posuvu. Rychloposuvy vřeteníkových supportů byli realizovány jedním elektromotorem, který se přes magnetickou spojku připínal a otáčel převodovkou pro pohyb supportů. Celé bruska byla respektive řízena jen hydraulickými ventily. Osy U1, U2, B2, C2 byli ovládány ručně natáčením ručních převodovek. Hlavní částí byli čerpadla na olej a motory pro pohony brusných vřeteníků. Celková bezpečnost byla na velmi malé úrovni. Přesnost toho stroje se pohybovala kolem 0,007mm. To znamená, že obrobek měl toleranci 0,007mm plus stanovená míra a 0,007mm do mínusu od stanovené míry. Sedm tisíc milimetrů neleželo ručními měřáky naměřit. V tu dobu to byla špička mezi brousícím stroji a jsou jich desítky po celém světě.

Rekonstrukcí toho stroje dosáhneme přesnosti 0,002mm. Pomůžeme životnímu prostředí, protože všechny hydraulické posuvy budou nahrazeny elektrickými. Zastaralé a nepřesné převodky jsou nahrazeny kuličkovým šrouby se stoupáním 10mm. Stoupne celková bezpečnost práce a vznikne nový brousící stroj BBE1A\_NC. Název je doplněn o NC, což znamená elektronické řízení os se zpětnou vazbou a přesnou informací o poloze.

#### **Řízení je rozděleno do částí:**

- Posuvy brusných kotoučů
- Posuvu orovnávacích diamantů pro orovnění kotoučů
- Bezpečnost práce
- Nastavování systému
- Ovládání brousícího stroje
- Programování a spojení frekvenčních měničů s řídicí jednotkou PLC
- Komunikace se servo pohony a aktuální odměřování pozice supportů

## **5 Praktická část**

### **5.1 Ovládání stroje**

Základní ovládání je tlačítka a HMI ovládacím panelem. Vedlejší ovládání stroje koncovými spínači, magnetickými čidly, plováky a bezpečnostními zámky.



První částí základního ovládání je pomocí mechanických tlačítek. Ovládací panel je rozdělen opticky do čtyř částí. V první části je obsluha brusných kotoučů a to start kotouče a stop kotouče, zapínání světla nad kotoučem, přepínání posuvu z ručního na automatický a přístup do servisního režimu. V druhé části jsou posuvy brusných vřeteníků. K těmto posuvům patří kruhový přepínač JOG, který nám určuje způsob pohybu. Máme plynulý pohyb nebo skokový pohyb s rozdělením od 0,001mm do 0,1mm. Vedle přepínače JOG je umístěna procentuální rychlost posuvu od nula do sta procent. Tlačítko korekce slouží pro dorovnání brusného vřetene o předem zadanou dráhu v automatické režimu. V třetí části je zapínání a vypínání dopravních pásů. Nastavování jejich rychlostí pro vstup do brusného pole a výstup z brusného pole. Tyto rychlosti jsou zobrazeny na dotykovém panelu. Mimo dopravních pásů je zde zapínání chlazení brusných kotoučů a odsávání brusných zplodin. Ve čtvrté části je emeregenci tlačítko neboli centrální stop. Je to bezpečnostní tlačítko pro zastavení celého stroje. Tlačítko je připojené do bezpečnostního relé, které při přerušení bezpečnostního obvodu vypne celkový výkon stroje a zůstane zapnuté jen ovládací napětí. Start stroje nám slouží k prvním nastartování stroje a přivedení do výkonu.

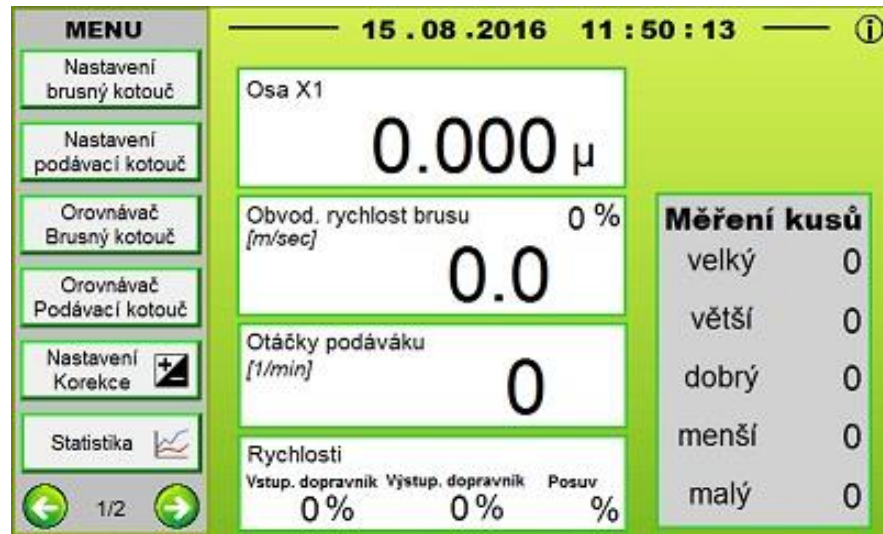


**Obrázek 21 - ovládací panel**

V druhé části základního ovládání je dotykový panel. Důležitým základním cílem bylo vymyslet a naprogramovat co nejjednodušší systém pro ovládání. Z toho jsem při tvorbě vzhledu dotykového panelu vycházel a snažil jsem se ho vytvořit přehledný a jednoduchý na ovládání. Panel je rozdělen do několika obrazovek. Na levé straně je přehledné menu pro přepínání do dalších částí nastavení obráběcího stroje. V rozích obrazovek jsou tak zvané servisní skryté obrazovky. Do těchto obrazovek se dostane jen pověřený pracovník, který má přístupové jméno a heslo. Tlačítka pro přepnutí jsou průhledná. Toto opatření je to z důvodu bezpečnosti. Zásahem do těchto parametrů od neproškoleného pracovníka by mohlo dojít k destrukci obráběcího stroje.

## Rozdělení obrazovek:

- Základní obrazovka



Obrázek 22- základní obrazovka HMI

Na této obrazovce je zobrazena aktuální poloha osy X1, rychlost brusných kotoučů a rychlost dopravníků. Slouží nám pouze jako orientační zobrazení aktuálních stavů brousícího stroje. Měření kusů slouží k statistickému počítání výroby celé linky. Dodatečně pro obsluhu jsou zobrazeny hodiny a datum.

Menu po levé straně nám nabízí další obrazovky pro nastavení stroje. Toto menu je rozděleno do dvou listů s přepínáním ve spodní části menu. Nejdůležitější nastavení je v první části menu z důvodu rychlého přepínání. V druhé části jsou spíše jen obrazovky pro nastavení základních funkcí stroje.

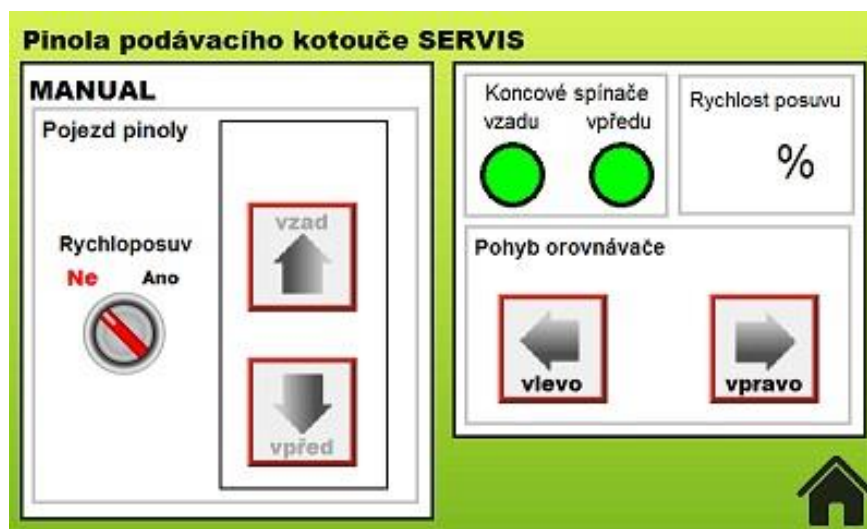
- **Nastavení brusného kotouče**



**Obrázek 23 - nastavení brusného kotouče**

Na této obrazovce je nastavení a diagnostika pro brusný kotouč. Diagnostika mazání vřetene nám zobrazuje, jaká část uložení ložisek je bez dostatečného mazání. V této situaci nelze spustit brusný kotouč. Mohlo by dojít k zadření uložení a zničení domečků pro uložení kotouče. Pokud nastane tato situace je rozblikán maják, červený rámeček kolem panelu a zobrazí se na hlavní obrazovce chyba mazání. Toto nám indikuje nečekanou chybu, kterou je nutno ihned vyřešit a vypne danou věc, která by se mohla poškodit. Obrazovka pro podávací kotouč je velice podobná. Na rozdíl od obrazovky brusného kotouče je obohacena od zobrazení odměřování podávacího supportu tedy osy X2. Umístění odměřování je v jiné obrazovce z důvodu že se s podávacím kotoučem posouvá jen při nastavování průměrů.

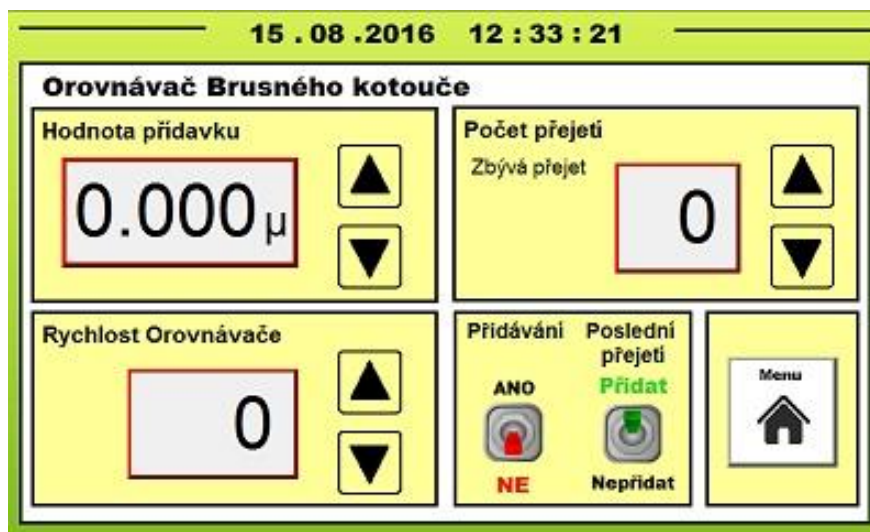
- Pinola podávacího kotouče SERVIS



Obrázek 24 - pinola servis

Tato obrazovka je pro servis pinoly. Pinola slouží k posouvání orovnávacího diamantu blíž ke kotouči, na základě dotknutí diamantu kotouče se může spustit orovnáání. Respektive nám tato obrazovka slouží pro hrubé přijetí pinoly ke kotouči na tak zvané škrtnutí. Obrazovka pro brusný kotouč je stejná. Stiskem tlačítek manuálně posouváme osou U1 a U2.

- Orovnávač brusného kotouče



Obrázek 25 - orovnávač brusného kotouče

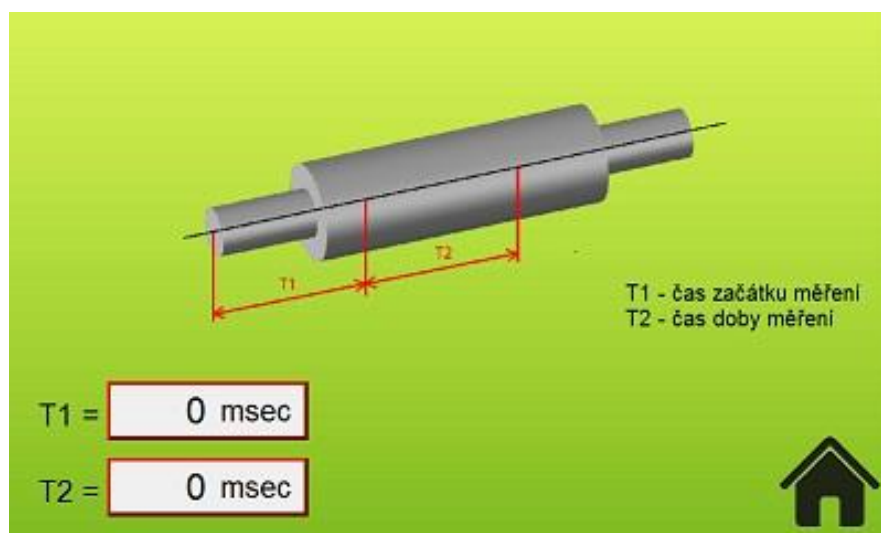
Zde se nastavuje základní parametry pro orovnávač neboli automatický posuv osy U1 a U2. Hodnota přídavku znamená, o kolik posune pinola diamant blíž k brusnému kotouči, pokud je v levé nebo v pravé úvratí. Počet přejetí udává, kolikrát se orovnávací suport otočí v cyklickém

čtyřúhelníku. Rychlost orovnávače udává hodnotu pro frekvenční měnič. Číslo je v PLC převedeno na analogový signál 0 – 10V a je tím řízen frekvenční měnič pro posuv osy W1 a W2. Na této obrazovce můžeme zastavit přidávání v úvratích nebo zamezit jen poslednímu přidání pro vyhlazení kotouče. Pro příklad nejpoužívanější hodnota orovnění je 0.010mm s počtem orovnění 10 a rychlostí 75%. Obrazovka pro podávací kotouč je obdobná.

Zadávání hodnot je dvěma způsoby:

1. Šipkami nahoru a dolů se skokem jedna
2. Stisknutím číselného pole
  - Po stisku zobrazíme numerickou klávesnici
  - Zadá se nová hodnota do pole
  - Potvrzení tlačítkem ENTER

- **Nastavení měřidla**



**Obrázek 26 - měřidlo nastavení**

Z důvodu různých délek trubek a dalších obráběcích polotovarů bylo nutné vytvořit obrazovku pro nastavení měřidla. Každý brousící stroj v celé lince má na výstupu digitální měřidlo průměrů, které komunikuje s PLC a vyhodnocuje velikost kusů. Na této obrazovce se nastavuje, za jakou dobu od první indikace předmětu u měřidla se začne měřit a komunikovat s PLC. Druhá doba je délka jak dlouho bude měřidlo měřit. Na této době také velmi záleží a u různých sérií výrobku je tato doba rozdílná. Do této obrazovky se dostanou jen proškolení zaměstnanci s přístupem přes přihlášení.

- **Status ostatních brousíčních strojů**

Celá linka se skládá ze tří strojů, kteří mezi sebou komunikují. Každá bruska má v panelu obrazovku status ostatních strojů kde se zobrazují chyby. Je to z důvodu ulehčení obsluhy celé linky.

- **Přihlášení uživatele do nadstandardního přístupu**



Obrázek 27 - přihlašovací obrazovka

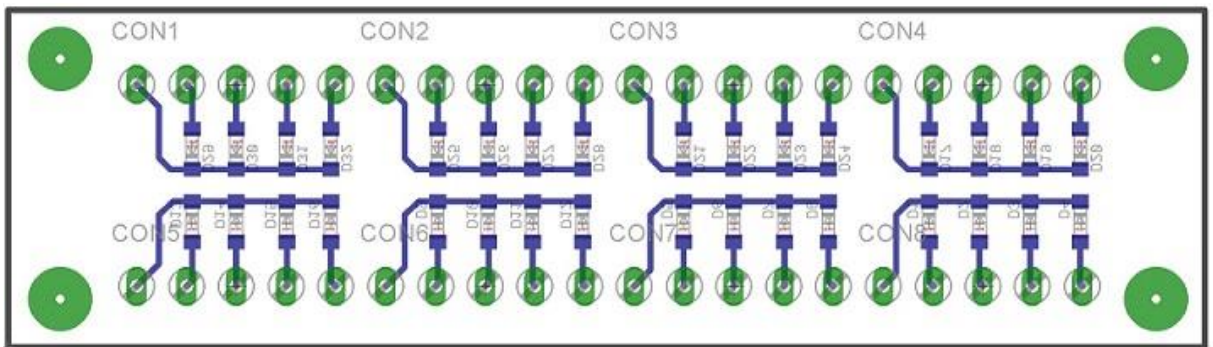
Pro získání nadstandardního přístupu je nutné se přihlásit. V obrazovce INFO, která se skrývá v pravém horním rohu na základní obrazovce, je ikona pro přihlášení. Po stisku na přihlášení se rozbalí přihlašovací obrazovka. Po zadání příslušného jména a hesla se otevrou nadstandardní možnosti nastavení brousíčního stroje a HMI panelu.

### 5.1.1 Multiplex pro tlačítka

Pro ušetření počtu drátů a počtu vstupů na PLC mě napadlo vymyslet plovoucí kód, kdy se bude posílat několik výstupových signálů a na základě rozdílného výstupu kódovat vstupní tlačítka. V prvotním návrhu bylo potřeba kolem šedesáti pěti počtu vstupů na PLC a tím pádem i minimálně sto žilový kabel v délce 25 metrů na jeden obráběcí stroj. Oproti tomu za použití a naprogramování plovoucího kódu by bylo potřeba jen čtyři výstupy a deset vstupů. To znamená, že na základní ovládání celého stroje bude potřeba jen 24 žilový kabel. Cena toho kabelu pětkrát klesne a i počet přídatných modulů se sníží. Pro kódování vstupů bylo potřeba navrhnout plošný spoj osazený diodami pro zabránění zpětného chodu signálu.



## Schéma plošného spoje pro plovoucí kód:



## Princip plovoucího kódu:

1. Vyslání jednoho výstupu na 10ms jako napájení první osmice tlačítek
2. Získání informace na vstupech z prvních osmy tlačítek
3. Vyhodnocení stisknutých tlačítek
4. Kontrola zda bylo opravdu tlačítko stisknuté
5. Posílání signálu z dalšího výstupu, na který jsou napojeny další tlačítka a to se opakuje na všech čtyřech výstupech.
6. Návrat na krok 1

Čas periody jednoho cyklu plovoucího kódu neboli multiplexu:

$$4ms \times 4 \text{ výstupy} = 40ms$$

Celý cyklus se opakuje neustále dokola s periodou 40ms. Čas 10ms je poměrově na stisk tlačítka a vyhodnocení ideální. Pokud bychom čas zvedli, byla by velká časová konstanta pro držení stisku tlačítka a tím by se prodlužovala celá odezva. Naopak kdybychom čas zmenšili mohlo by dojít k zanedbání stisku a špatnému vyhodnocování.

Startovací tlačítka jsou na spínací kontakt, oproti tomu stop tlačítka jsem na rozpínací kontakt. Je to z důvodu bezpečnosti. Pokud by došlo k porušení vodičů nebo selhání programu v PLC nešlo by nastartovat příslušné části obráběcího stroje.

## 5.2 Bezpečnost

Hlavní částí je bezpečnost obráběcího stroje. Na každém stroji v celé lince jsou dva centrální stopové vypínače. Slouží k náhlému vypnutí z důvodu poruchy nebo možného úrazu. Po

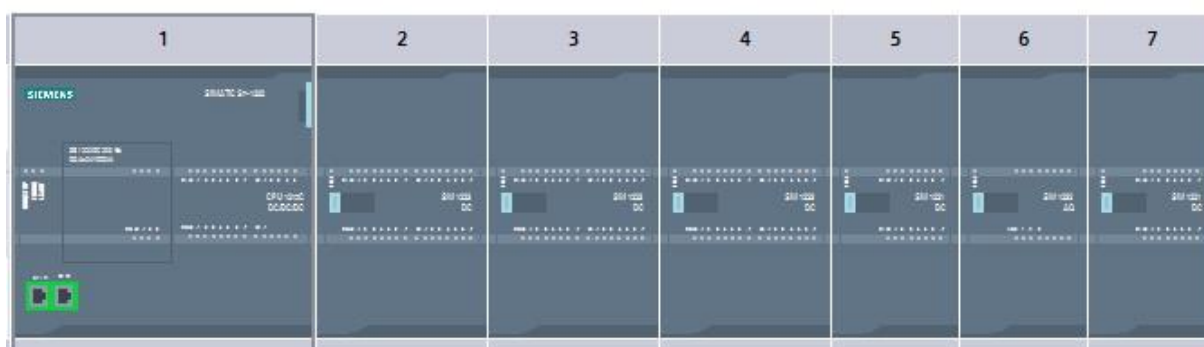


zamáčknutí tohoto vypínače se ihned vypne výkonová část obráběcího stroje. To znamená, že ihned začnou brzdít brusné kotouče, vypnou se všechny čerpadla, motory a i výkon do pohonů.

Bezpečnostní elektrické zámky na kryty kotoučů slouží k zabránění neproškoleným zaměstnancům přibližovat jakoukoliv část svého těla nebo předmětů do brusného pásma. Do brusného pásma má přístup jen proškolený zaměstnanec. Přístup toho zaměstnance je unikátní má svůj klíč na přepnutí do servisního stavu. Pokud přepne stroj do servisního přístupu a aktivuje ho stiskem tlačítka, může se pohybovat v brusném pásmu. Při prvním startu stroje musejí být kryty uzavřeny. Až po startu stroje lze přepnutím do servisního režimu a stiskem aktivačního tlačítka odemknout bezpečnostní zámek a odblokovat kryty.

### 5.3 PLC

Jako hlavní řídicí jednotku neboli PLC obráběcího stroje jsem vybral výrobce značky Siemens. Typ jednotky je S7-1200, ke který se připojují postupně další moduly pro rozšíření vstupů a výstupů.



Obrázek 28 – PLC

Hlavní řídicí jednotka má napájení 24V. Pracovní paměť je 125kb. Počet digitálních vstupů je 14, tyto vstupy se dají zapojit jak na SINK tak na SOURCE. SINK znamená, že opačný proti pól vstupu je nula voltů. V případě zapojení typu SOURCE je protipól vstupu kladný to znamená 24V. Výstupů je 10 ve formě spínacích tranzistorů. Na základní jednotce jsou na přídatné desce umístěny analogové vstupy a výstupy s rozsahem 0-10V. Toto PLC disponuje šesti high-speed čítačů pro připojení například odměřovacích pravítek, které nám udávají pozici v signálu A/B nebo pro odměřování pulzů pro řízení krokových a servo motorů. Mimo 6 čítačů jsou zde čtyři rychlé výstupy neboli pulsní výstupy s frekvencí spínání 100000Hz. Tyto výstupy slouží pro řízení motorů. V našem případě na řízení osy X1 a osy X2 (viz Obrázek 19). Jako programovací port je zde použit profinetový port RJ45. Tyto porty jsou tu dva pro

programování, komunikace s HMI operátorským panelem a pro komunikaci mezi ostatními jednotkami.

K této hlavní jednotce lze dále připojit tři komunikační moduly pro sériovou komunikaci dále pak osm signálních modulů pro vstupní a výstupní signály.

Druhý, třetí a čtvrtý modul připojení k hlavní jednotce jsou digitální vstupy a výstupy. Každý modul má šestnáct vstupů a také šestnáct výstupů. Pátý a sedmý modul disponuje jen šestnácti digitálními vstupy. Šestý modul jsou přídavné analogové výstupy pro řízení frekvenčních měničů.

### 5.3.1 Programování

Programování je rozděleno na mnoho částí. Každá část je důležitá pro správný chod. Nejdůležitější při programování PLC je si uvědomit že program jde po sobě cyklicky za sebou. Následnost příkazů je velmi důležitá při špatném navazování může dojít ke ztracení informace a špatnému provedení algoritmu.

- **Tag table**

Default tag table							
	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	PulsySrM1	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	pulzy pro čítač HSC0 z SrM1
2	Start Stroje	Bool	%I0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	start stroje
3	Manual/Automat	Bool	%M10.7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	pomocna blokace Automat/Manual
4	StartPodavak	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start podávacího kotouče
5	StopPodavak	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stop podávacího kotouče
6	Brus Stop	Bool	%I0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stop brusného kotouče
7	Brus Start	Bool	%I0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start brusného kotouče
8	Chlazení	Bool	%I1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start chladicí kapaliny
9	Odsavení	Bool	%I1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start odsávání
10	BrusRychPosVpred	Bool	%I1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rychloposuv brusného kotouče vpřed
11	BrusRychPosS	Bool	%I1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rychloposuv Brusného kotouče potvr...
12	BrusRychPosVzad	Bool	%I1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rychloposuv brusného kotouče vzad
13	Korekce	Bool	%I8.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Přistavení mikroposuv vpřed
14	Volny_1	Bool	%I8.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Přistavení mikroposuv vzad
15	BrusOrovStop	Bool	%I8.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stop orovnávače brusného kotouče
16	BrusOrovStart	Bool	%I8.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start orovnávače brusného kotouče
17	PodOrovStop	Bool	%I8.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Stop orovnávače podávacího kotouče
18	PodOrovStart	Bool	%I8.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Start orovnávače podávacího kotouče
19	PodRychPosVpred	Bool	%I9.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rychloposuv podávacího kotouče vpř...
20	PodRychPosPodS	Bool	%I9.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rychloposuv podávacího kotouče potv...
21	PodRychPosVzad	Bool	%I9.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Rychloposuv podávacího kotouče vzad
22	PodKotoucOK	Bool	%I9.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Podávací kotouč OK
23	BrusKotoucOK	Bool	%I9.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Brusný kotouč OK
24	PodChod	Bool	%I9.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Podávací kotouč chod
25	BrusChod	Bool	%I9.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Brusný kotouč chod

Obrázek 29 - tag tabulka

První a nejdůležitější částí je tabulka tagů, jsou v ní uloženy všechny proměnné. Jakoukoliv proměnnou si můžeme pojmenovat a musíme jí přiřadit paměť. Pozor přiřazování paměti registrů není automatické, je to čistě programátorská věc a při nepozornosti se mohou třiceti

dvou bitová čísla překrývat a tím i ovlivňovat nadcházející registry. Jediná daná paměť je pro vstupy a výstupy.

Po vytvoření tabulky tagů se dále při programování odkazuje na prefixové jméno proměnné, nemusíme si pamatovat statické místo v paměti.

Pro příklad tag Manual/Automat je přiřazena paměti %M10.7. to je část paměti v registrech a dále si tuto paměť pamatovat nemusíme. Při programování se odkazujeme na jméno proměnné Manul/Automat a přistupovat k ní přes tečkovou notaci.

- **Datové bloky**

Program je rozdělen do datových bloků. Hlavní datový blok je Main. Je spouštěn se základní periodou PLC a postupuje směrem shora dolů a neustále se opakuje.

- **Programové bloky**

Jsou to bloky, ve kterých jsou části programu, které jsou volané z hlavního Main bloku. Jsou to podprogramy neboli funkce co se často opakují. Po vykonání programového bloku se přesuneme zpět do hlavního bloku.

- **Technologické objekty**

Technologické objekty slouží například pro řízení krokových motorů nebo připojených indikací.

V našem případě jako technologický objekt používáme PTO.

PTO je pulsní řízení výstupu. Zadáváme frekvenci a počet pulsů na výstup a ten dále posíláme do servo měniče pro polohování.

### **5.3.2 Programování rychlých výstupů a pohybu servomotorů**

Nedá se říct, že by řízení krokových motorů bylo to hlavní, ale patří mezi nejdůležitější části programu. Na posuvu brusných vřeteníků je založena přesnost výsledných kusů.

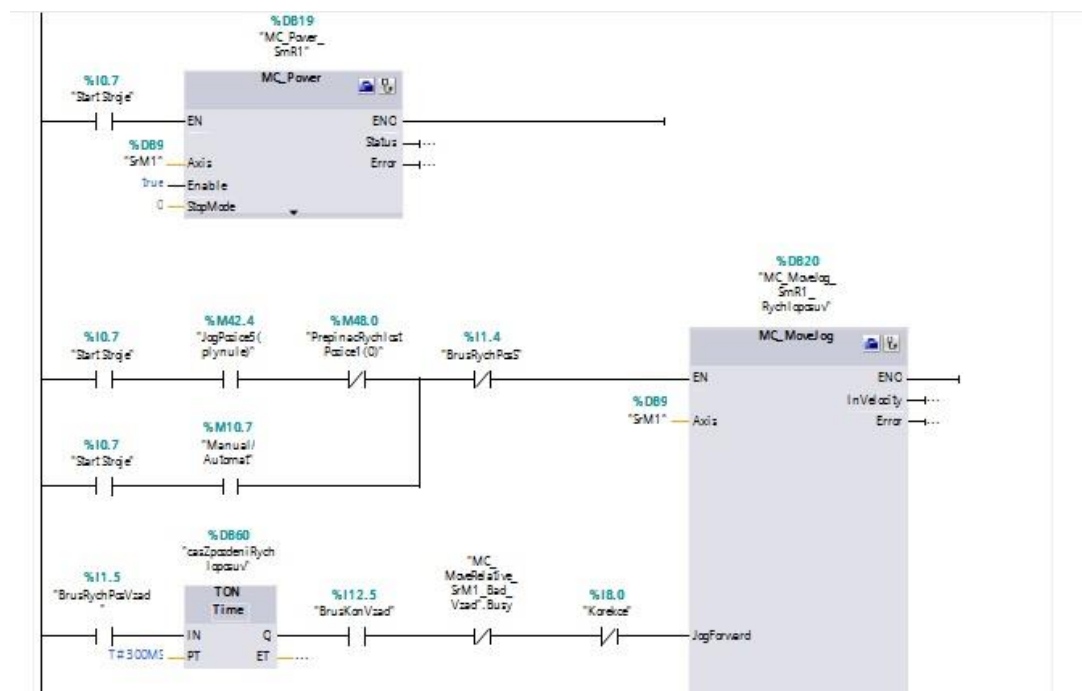
Základním krokem je nastavení pulsních výstupů a nakonfigurování technologických objektů. V části technology objekt nastavíme, jaký výstup bude sloužit pro pulsní řízení a jaký bude pro změnu směru. Po konfiguraci portů musíme nastavit maximální frekvence na výstupu nástupné a sestupné hrany pro akceleraci a deakceleraci motoru. Pokud to nenastavíme, tak dochází

k nepřesnému počtu pulsů na výstupu. Mimo náběžných a sestupných hran jde v části technologických objektů pod záložkou PTO (pulse train output) neboli pulsní řízený výstup lze nastavit softwarový limit switch. Je to softwarový koncový spínač založený na počtu pulsů vyslaných v daném směru. Toto nastavení nám v našem případě nevyhovuje z důvodu, že potřebujeme nulovat pozici osy a vlivem úběru brusných kotoučů nedokážeme ve směru k podávacímu pravítku určit havarijní koncový spínač. Koncový spínač ve směru od podávacího pravítka je řešen mechanickým spínačem, který nám v případě rozepnutí zastaví výstup pulsů do servo měniče.

Prvním částí, která je potřeba naprogramovat pro pohyb motoru je programové zapnutí pulsních výstupů. Toto zapnutí se provede funkcí MC\_POWER. Touto funkcí vypínáme pulsy, pokud dojde na najetí na koncový spínač.

Pohyb brusných vřeteníku je buď plynulý, nebo skokový. Pokud je plynulý používáme funkci MC\_MOVERELATIVE kde zadáváme rychlost výstupních pulsů a směr otáčení. Pokud používáme skokový pohyb, použijeme funkci MC\_MOVEABSOLUTE kde zadáváme počet pulsů, neboli jakou dráhu chceme ujet, rychlost neboli frekvenci pulsů na výstupu a náběžné a sestupné rychlosti pro start a brzdění.

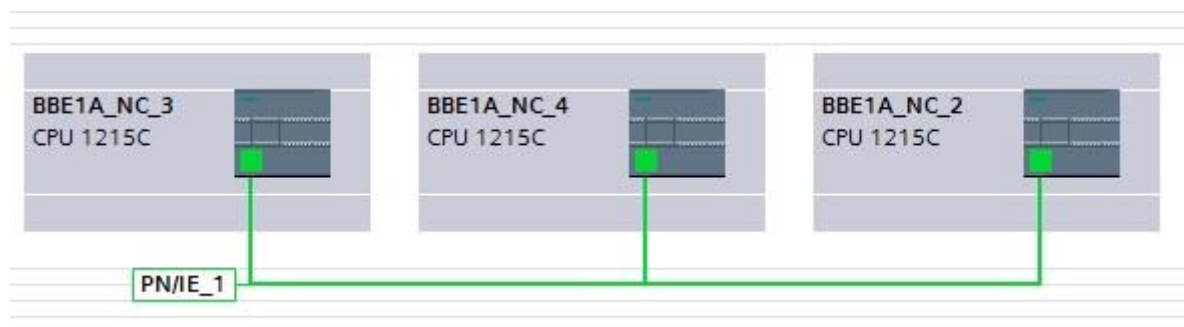
### Příklad programu pro plynulý pohyb vpřed



Obrázek 30 - řízení SrM1

### 5.3.3 Komunikace

V požadavku na obráběcí linku byla komunikace brusek mezi sebou. První bruska musí vědět status té třetí i druhé a naopak. Z tohoto důvodu jsou všechny stroje mezi sebou propojeny přes síť.



Obrázek 31 - propojení PLC

Propojení je důležité nejenom z důvodu zasílání informací, chyb a statistik, ale především proto, že dopravníkové pásy jsou napájeny a připojeny k prostřední brusce. Pro tuto linku je nutné ovládat dopravníkové pásy ze všech ostatních strojů. Tím pádem nám propojení slouží i pro řízení ostatních obráběcích strojů.

#### Topologie sítě:

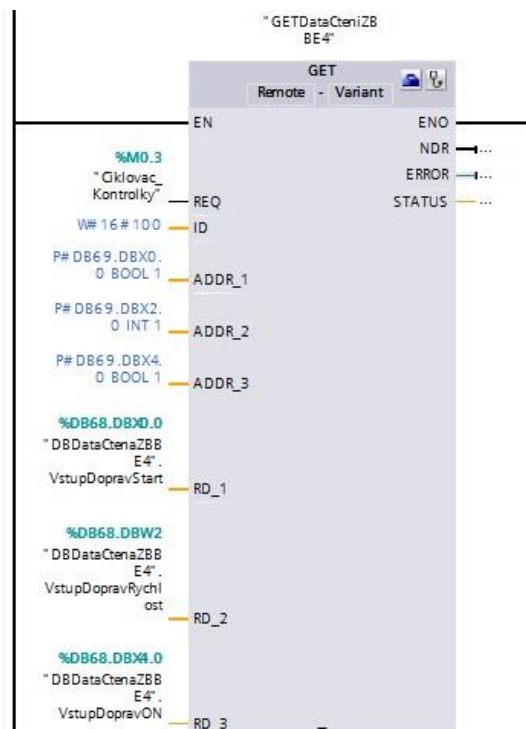
Obráběcí stroj	IP Adresa
BBE1A_NC_2_PLC	192.168.1.50
BBE1A_NC_2_HMI	192.168.1.51
BBE1A_NC_3_PLC	192.168.1.52
BBE1A_NC_3_HMI	192.168.1.53
BBE1A_NC_4_PLC	192.168.1.54
BBE1A_NC_4_HMI	192.168.1.55

Tabulka 1 - Topologie sítě

Pro komunikaci je nutné vytvořit v PLC spojení pro daný typ řídicí jednotky. Toto spojení se nazývá S7\_CONNECTION. V tomto spojení se přiřadí IP adresy příslušných řídicích jednotek. V našem případě máme dvě spojení. První spojení je mezi BBE3 a BBE4 a druhé spojení je mezi BBE4 a BBE2. Automaticky se propojí i BBE3 s BBE2.

## Přenos dat mezi PLC

Pokud používáme spojení S7, tak nám pro přenos dat neboli celých datových bloku slouží metoda GET a PUT. Při používání těchto metod se po síti přesouvají celé datové bloky. Z tohoto důvodu je nutné mít v obou PLC stejný datový blok z důvodu adresace. Princip přenosu je takový, že v časových intervalech přesouváme po síti danou část paměti toho prvního PLC. Pokud použijeme metodu GET, bereme část paměti z druhého konce a v případě použití metody PUT zasíláme danou část do druhého PLC.



Obrázek 32 – metoda GET

Znázorněná metoda GET nám posílá část paměti svého PLC. Rychlost přenosu je s frekvencí 25HZ pro správný přenos. ID spojení uloženo ve vnitřní paměti a pro tento příklad je id spojení 100. Parametr ADDR\_1 je parametr z jaké části paměti a jak velkou část paměti chceme poslat do druhého PLC. Parametr RD\_1 je místo uložení té první části přenášené paměti.

Bohužel v tomto případě spojení nelze posílat proměnné mezi sebou jednoduše je nutné posílat celou část paměti a v druhém PLC si tu část paměti rozdělit podle velikost posílaných dat.

Celkový přenos je omezen maximální velikostí přenášeného datového bloku.

### 5.3.4 Multiplex tlačítek

V předchozí části jsem psal o tak zvaném multiplexu pro kódování tlačítek. Zde je ukázka části programu pro řízení výstupů a dekodování tlačítek.

#### 1. Řízení výstupů

-časové spínání jednotlivých výstupu po zadaném čase

#### 2. Dekódování tlačítek

- Dle stisknutého tlačítka, které odpovídá vstupu na základě spuštěného výstupu se namapuje tlačítko do přiřazené proměnné.

```
//instance casovacu
timCasovacVystup1(IN:=xCasovacVystup1Start , PT:= timCasVystupu, Q=>
xCasovacVystup1Done, ET=> );
timCasovacVystup2(IN:=xCasovacVystup2Start , PT:= timCasVystupu, Q=>
xCasovacVystup2Done, ET=> );
timCasovacVystup3(IN:=xCasovacVystup3Start , PT:= timCasVystupu, Q=>
xCasovacVystup3Done, ET=> );
timCasovacZpozdeniVystup1(IN:= xCasovacZpozdeniVystup1Start, PT:=
timCasZpozdeni, Q=> xCasovacZpozdeniVystup1Done, ET=> );
timCasovacZpozdeniVystup2(IN:= xCasovacZpozdeniVystup2Start, PT:=
timCasZpozdeni, Q=> xCasovacZpozdeniVystup2Done, ET=> );
timCasovacZpozdeniVystup3(IN:= xCasovacZpozdeniVystup3Start, PT:=
timCasZpozdeni, Q=> xCasovacZpozdeniVystup3Done, ET=> );

IF xStartStroje = TRUE OR xWindowDiagnostikaTlacitek THEN

CASE wKrok OF

0:
wKrok := 5;

5: // krok = 5 zapnutí výstupu A
xCasovacVystup1Start := TRUE;
xCasovacZpozdeniVystup1Start := TRUE;
// kontrola vstupu 1 ze slučovače
IF xCasovacZpozdeniVystup1Done = TRUE THEN
IF xVstupSlucovac0 = TRUE THEN
xVstupBrusStop := TRUE;
ELSIF xVstupSlucovac0 = FALSE THEN
xVstupBrusStop := FALSE;
```

Sepnutí prvního časovače  
a výstupu pro napájení

Dekódování tlačítka dle  
daného vstupu

```

END_IF
// kontrola vstupu 2 ze slučovače
    IF xVstupSlucovac1 = TRUE THEN
        xVstupBrusStart := TRUE;
    ELSIF xVstupSlucovac1 = FALSE THEN
        xVstupBrusStart := FALSE;
END_IF
// kontrola vstupu 3 ze slučovače
    IF xVstupSlucovac2 = TRUE THEN
        xVstupSvetlo := TRUE;
    ELSIF xVstupSlucovac2 = FALSE THEN
        xVstupSvetlo := FALSE;
END_IF
// kontrola vstupu 4 ze slučovače
    IF xVstupSlucovac3 = TRUE THEN
        xVstupPosuvRezim := TRUE;
    ELSIF xVstupSlucovac3 = FALSE THEN
        xVstupPosuvRezim := FALSE;
END_IF
// kontrola vstupu 5 ze slučovače
    IF xVstupSlucovac4 = TRUE THEN
        xVstupPodStop := TRUE;
    ELSIF xVstupSlucovac4 = FALSE THEN
        xVstupPodStop := FALSE;
END_IF
// kontrola vstupu 6 ze slučovače
    IF xVstupSlucovac5 = TRUE THEN
        xVstupPodStart := TRUE;
    ELSIF xVstupSlucovac5 = FALSE THEN
        xVstupPodStart := FALSE;
END_IF
END_IF
// konec casu vystupu A
IF xCasovacVystup1Done = TRUE THEN
    xCasovacVystup1Start := FALSE;
    xCasovacZpozdeniVystup1Start := FALSE;
    wKrok := 10;
END_IF

10: // krok = 10 zapnuti vystupu B

```



```

xCasovacVystup2Start := TRUE;
xCasovacZpozdeniVystup2Start := TRUE;
// kontrola vstupu 1 ze slučovače
IF xCasovacZpozdeniVystup2Done = TRUE THEN
IF xVstupSlucovac0 = TRUE THEN
    xVstupBrusVpred := TRUE;
    ELSIF xVstupSlucovac0 = FALSE THEN
        xVstupBrusVpred := FALSE;

```

Kód pro dekódování tlačítek dále pokračuje.

### 5.3.5 Vyhodnocení měřidla

Měřidlo na výstupu každého stroje není jen z důvodu statistiky a kontroly výstupního výrobku, ale v případě že je linka v automatickém režimu, tak na základě nastavené korekce si stroj automaticky přisouvá brusné kotouče podle potřeby.

Pro měření jsme vybrali měřidlo značky Keyence. V řídicí jednotce se nastaví rozpětí pro malý kus, menší kus, dobrý kus, větší kus, velký kus. Do řídicí jednotky měřidla lze ukládat více programů na různé průměry.

Příklad nastavení měřidla:

Velikost kusu	rozměr
LO – malý kus	18.980mm
LL – menší kus	18.990mm
GO – dobrý kus	19.000mm
HL – větší kus	19.010mm
HH – velký kus	19.020mm

Tabulka 2 - nastavení měřidla

Měřidlo posílá signály o měření do PLC. V řídicí jednotce přijmeme signál, pokud je naplněn čas měření, tak zachytáváme dané impulsy od měřidla. V případě že během měření nastane situace, že měřidlo zašle signál o velikosti výrobku mimo nastavený rozsah, je automaticky označen jako špatný kus. Tento špatný kus dojde po dopravníkovém pásu na čidlo. Toto čidlo sepne vzduchový ventil a formou nožového dopravníku vyhodí špatný kus z linky do odběrného místa. Pokud je měření v tolerancích mezi LO – LL nebo HL- HH pokračuje kus dále v lince.

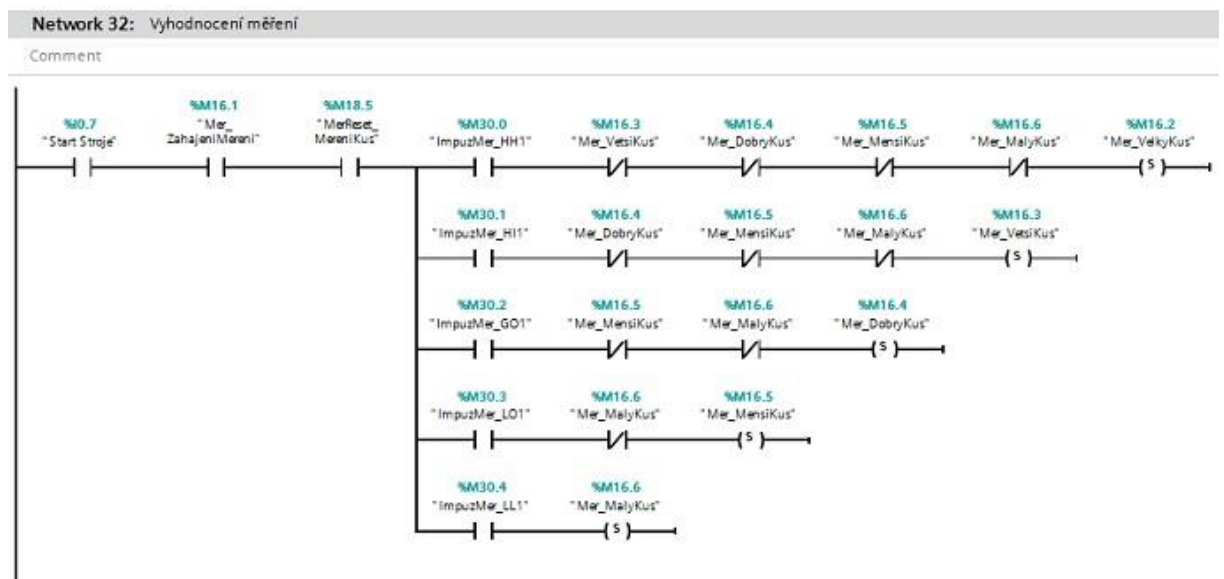
V nastavení brousícího stroje na obrazovka nastavení korekce. Korekce znamená dorovnání na požadovanou hodnotu. V této obrazovce obsluha nastaví velikost automatické korekce, ruční

korekce a počet kusů pro korekci. Pro zapnutí automatické korekce musí být obráběcí stroj ve stavu automatu.

V případě nastavení automatické korekce 0,005mm a počtu kusů pro korekci 2, tak po vyhodnocení 2 po sobě jdoucích větších nebo menších kusů se automaticky posune brusný vřeteník o 0,005mm ve směru nedobrych kusů. Pokud měřidlo vyhodnotí dvakrát velký nebo malý kus, automatická korekce je navýšena dvakrát. To znamená, že se brusný vřeteník posune o 0,010mm ve směru špatných kusů.

Pokud obsluha vidí, že automatický režim nestíhá dorovnávat korekci, může přidat ručně. Stiskem tlačítka korekce na ovládacím panelu a směr brusného vřeteník stroj udělá nastavenou korekci dle daného směru. Při stisku tlačítka korekce se na panelu místo obvodové rychlosti zobrazí nastavená hodnota ruční korekce. Měnit hodnoty lze pouze v příslušné obrazovce pro nastavení korekce. Toto nastavení lze měnit i v automatickém režimu, není nutno přepínat se do manuálního. Je to z důvodu že se toto nastavení často mění v plném a nepřetržitém provozu.

### Vyhodnocení měření:



Obrázek 33 – příklad měřidlo

### 5.3.6 Odměrování

Odměrování pozice brusných kotoučů je velmi důležité. Obsluha se podle toho řídí a při náhlém odjetí brusných kotoučů se musejí vrátit přesně na danou pozici zpět.

Odměrování je realizováno High-speed counters neboli ultra rychlými čítači. Pulsní výstup je přiveden nejen do řídicího servomotoru, ale i zpět na vstup PLC do rychlého čítače.

Pro správnou funkci čítačů se musejí nastavit v technologický objektech. Po přiřazení vnitřní paměti, názvu a vstupů čítače, můžeme dále v programu s čítačem pracovat.

Odměrování musí být naprosto přesné. Tím nám vzniká obrovský problém. Každý brousící stroj, kde je převodovka nebo kuličkový šroub, má tak zvaný „mrtvý bod“ neboli bad. Ten je na každém brousícím stroji jiný, ale v každé části posuvu stejný. Posuv vřeteníků je realizován kuličkovým šroubem. Při změně směru nastane jev, kdy dopnutí matky v kuličkovém šroubu je volné. To znamená, že na displeji ujedeme při změně směru jeden milimetr, ale v realitě vřeteník stojí. Bohužel u toho typu PLC nejde maskovat vstupní čítač. Pro dokonalou přesnost jsem tento jev musel obejít softwarově a to tak, že při změně směru nastavíme pomocnou proměnou do logické jedničky. Pokud je tato proměnná v jedničce je start pohybu vřeteníku časově zpomalen. V době zpomalení si musíme zapamatovat hodnotu čítače, ujet vřeteníkem daný „mrtvý bod“ a opět nahrát do čítače tu prvotní hodnotu čítače před ujetím „mrtvého bodu“. Celá tato operace se musí uskutečnit v době 200ms. Při těchto operacích záleží na postupném uspořádání příkazů za sebou bez zbytečných přeskokování v programu. Kdybychom přeskakovali do jiných částí, zpomalovali bychom provedení sekvence a mohlo by nastat špatné vykonání bad a to by vedlo k vyrobení několika zmetků než by si to automat vyrovnal pomocí měřidla a nastavené korekce.

Bohužel v tomto odvětví jsou zmetkové výroby hodně prodělečné a firma si nemůže dovolit vyrábět špatné kusy.

### **5.3.7 Rychlost brusného kotouče**

Rychlost brusného kotouče se nastavuje z dotykového panelu. Procentuálně od 30% do 100%. To znamená, že maximální obvodová rychlost brusného kotouče na sto procent je 45 m/s. vlivem broušení se opotřebovává brusný kotouč. Tím se zmenšuje celková ujetá dráha za sekundu a i celková rychlost kotouče. Abych eliminoval zmenšení kotouče, je na orovnávači brusného vřete přidělané analogové inkrementální pravítko připojené na analogový vstup do PLC. V případě nového kotouče je pinola orovnávače zajeta co nejdále a tím pádem i analogové pravítko je na nulové hodnotě. Postupným broušením a orovnááním kotoučů nám analogovém pravítko propouští určité napětí v rozsahu 0 – 10V do analogového vstupu PLC. Podle velikosti vstupu procentuálně zvyšujeme analogový výstup pro frekvenční měnič brusného kotouče.

Tímto způsobem zůstává obvodová rychlost nepozměněna ani při úbytku brousícího kotouče. Na velikosti obvodové rychlosti při broušení velmi záleží. Rychlost nám udává přesnost broušení a povrch broušeného materiálu.

#### Výpočet konstantní rychlosti kotouče:

1. První je kontrola, jestli hodnota analogového pravítka dále jen HAP je v rozmezí.

$$2765 < HAP < 22118$$

Pokud není v rozmezí je automaticky  $HAP = 0$

2. Odečteme od HAP minimální hodnotu 2765 a uložíme ji do pomocné pom

$$pom = HAP - 2765$$

3. Pomocnou pom vydělíme maximálním rozsahem pro analogový výstup a uložíme do druhé pomocné pom2

$$pom2 = \frac{pom}{37737}$$

4. V posledním kroku pokud je pom2 menší jak nula uložíme do ní automaticky 0

$$pom2 > 0$$

5. Pom2 nám řídí analogový výstup pro frekvenčním měnič brusného kotouče.

#### 5.3.8 Realizace přepínače pomocí dvojkové soustavy

Pro kruhový přepínač JOG, který slouží pro přepínání hodnoty skoku, nebo plynulého posuvu jsem použil kódování dvojkové soustavy. Pro sedm pozic byly použity jen tři dráty a tři vstupy.

## Kódovací tabulka

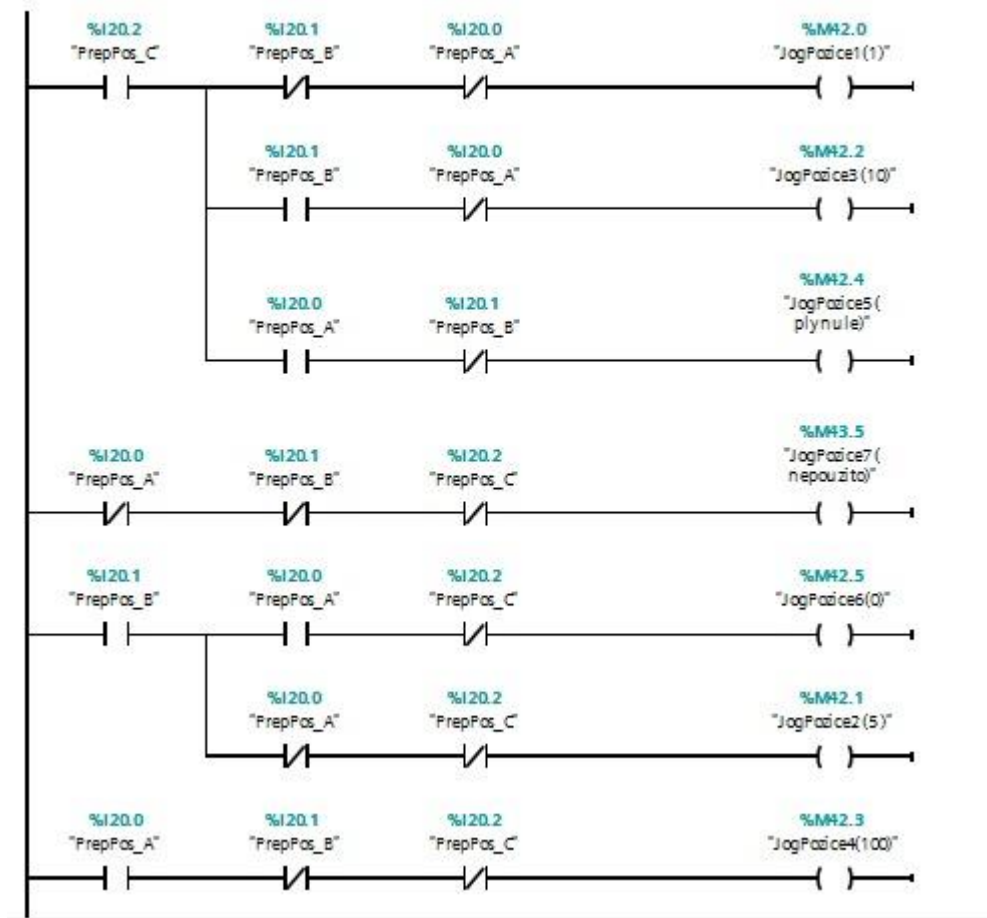
A	B	C	POZICE
0	0	1	Skok 1 (pozice 1)
0	1	1	Skok 10 (pozice 3)
1	0	1	Plynule (pozice 5)
0	0	0	Nepoužito (pozice 7)
1	1	0	Skok 0 (pozice 6)
0	1	0	Skok 5 (pozice 2)
1	0	0	Skok 100 (pozice 4)

**Tabulka 3 - tabulka kódování JOG**

Tímto přepínačem bylo určeno, jaká funkce pro ovládání přísuvu brusných kotoučů se bude používat. Pokud byl na přepínači plynulý posuv, používala se funkce MC\_MOVEJOG s předem danou rychlostí. Pokud byla hodnota skoková, používala se funkce MC\_MOVERELATIVE s předem danou vzdáleností. Podle natočení JOG přepínače, bylo rozkódováno, o jakou vzdálenost se jedná. Na základě hodnoty přepínače byla do funkce zapsána hodnota skoku přepočtená na počet výstupních pulsů.

Například pro ujetí jedné desetiny milimetru bylo potřeba na výstup pustit tisíc pulsů. To je z důvodu že na ujetí jednoho milimetru potřebujeme poslat na výstup deset tisíc pulsů.

Comment



Tabulka 4 - prepínač JOG

### 5.3.9 Automatický a ruční režim

Obráběcí stroj pracuje ve dvou režimech. První režim je manuální a druhý automatický.

#### 1. Manuální režim

Slouží pro mechanické nastavení brusky. Jsou povoleny všechny tlačítka pro pohyby brusných vřeteníků, orovnávačů, pinol, dopravníků a čerpadel. Nemá žádné omezení. Je to z důvodu aby obsluha mohla stroj připravit pro sériovou výrobu. V manuální režimu nefunguje indikace od měřícího přístroje. Tento režim je jen pro nastavení a přípravu. Pouze v manuálním režimu lze orovnávat brusný kotouč.

#### 2. Automatický režim

V tomto režimu jsou zakázány manuální pohyby brusných kotoučů. Jediný povolený pohyb v automatické režimu je havarijný. To znamená rychloposuv

brusného supportu vzad při nečekané události. O ostatní pohyb vpřed a vzad se stará PLC na základě změření výrobku od měřidla, které v automatickém režimu komunikuje s řídicí jednotkou. V té se vyhodnotí počet a velikost špatných kusů a automaticky se support přisune.

## 5.4 HMI

Pro tento obráběcí stroj jsem zvolil dotykový panel firmy Weintek a typ 8050ie. Je to širokoúhlý deseti palcový dotykový panel s možností komunikace po profientu. Programovací prostředí je přímo od té firmy zdarma a název je EasyBuilderPro.

Zprvu bylo důležité rozvrhnout umístění numerických displejů, tlačítek a bitových lamp. Vytvořit menu pro přepínání obrazovek a udělat co nejjednodušší ovládání. Po vytvoření základního grafické vzhledu bylo nutné nastavit komunikaci mezi PLC.

Komunikace je založena na protokolu IPV4. PLC jako master má svojí IP adresu 192.168.1.50. aby mohl HMI panel komunikovat s příslušným PLC musí mít interní IP adresu na stejné podsíti jako má dané PLC. Pro tento případ jsem zvolil IP adresu 192.168.1.51. Spojení musíme vytvořit v a nastavit v HMI panelu správnou IP adresu PLC.

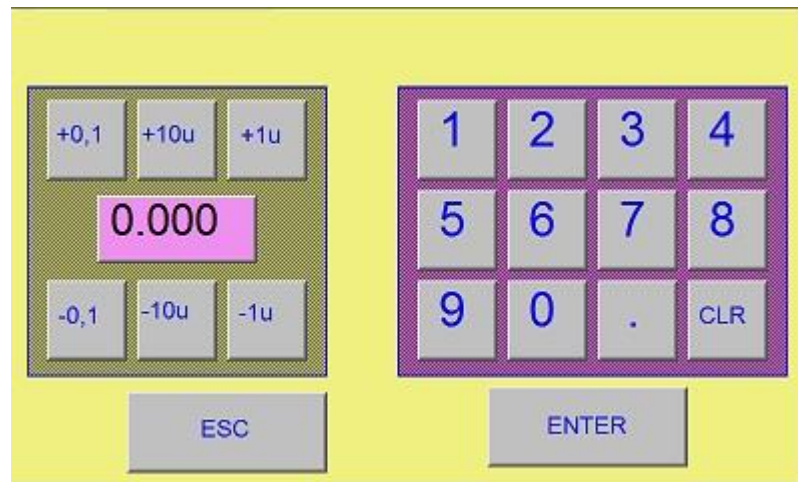
V případě, že máme komunikaci správně nastavenou, je nutné nahrát proměnné z PLC neboli tag tabulku do dotykového panelu. Po nahrání a konverze proměnných musejí být obě dvě zařízení spolu spojena pro následnou verifikaci dat. Pokud tabulka tagů nesouhlasí s tabulkou v druhém zařízení tak dotykový panel nepracuje v pořádku.

Pro zobrazování bitů z PLC se používají bitové lampy, které mají stav logické nuly nebo logické jedničky. Pro dotyková pole používám nastavovací bity. Ty mají různé funkce. Například základní funkce zapnout, vypnout, při stisku je v logické jedničce nebo jen přepínání po stisku.

Pro zobrazování numerických hodnot se používají numerické displeje. Ty mohou být pouze pro zobrazování nebo mají funkci pro zapisování hodnot. Tuto funkci nevyužívám. Pro zadávání hodnot jsem navrhnul a naprogramoval vlastní numerickou klávesnici z důvodu jednoduššího zadávání a hlavně z důvodu dvojitého potvrzení nové hodnoty, aby nedocházelo k mylnému zadání špatného čísla. Vlastní numerická klávesnice pracuje pomocí maker a je nezávislá na vnitřní numerické klávesnici. Vnitřní numerická klávesnice se používá pouze pro nastavení panelu, jako jsou hodiny, IP adresa a přihlašování.

**Postup ukládání nové hodnoty:**

1. Spuštění makra pro uložení do jakého numerického pole chceme vkládat novou hodnotu.
2. Rozvinutí numerické klávesnice



Obrázek 34 - numerická klávesnice

Na této klávesnici můžeme zadávat hodnotu po jednotkách, desítkách, nebo v případě velkého rozdílu zadat celé číslo nové. Většinou se hodnota mění jen o malé číslo v řádu setin a tisícín. To byl další důvod k navrhnutí jiné numerické klávesnice.

3. Po zadání nového čísla se dotykem na ENTER spustí následující makro, které uloží danou hodnotu do paměti v PLC a zobrazí nám ji v numerickém displeji.

#### 5.4.1 Základní funkce

Mimo grafického návrhu zobrazení používáme pro komunikaci zobrazovací numerické displeje, bitové lampy pro zobrazení jednotlivých bitů z PLC, nastavovací set bity pro nastavení bytů do PLC a makra pro přenos dat mezi PLC a HMI.

#### 5.4.2 Makra

Pro přenos hodnot do PLC, vnitřní nastavení PLC, a přenos interních informací z panelu se používají makra. Jsou to vnitřní programy, které reagují na předem dané podmínky z PLC.

Makra pracují i v opačném směru. Vytahují si informace z PLC a zobrazují je do HMI, nebo slouží pro vnitřní program.



Mají spoustu možností, jako jsou například cyklické spouštění, spouštění jen při startu, reakce na vstup do PLC nebo reakce na dotek panelu. Je to programování vnitřního systému dotykového panelu.

Dotykový panel je brán jako slave oproti PLC ale i přesto se musí programovat vnitřní správný chod, abych dosáhli správného fungování.

### **Příklad makra pro zobrazování rychlosti posuvu v procentech:**

```
macro_command main()

int bitNula = 0

int bitDeset = 0

int bitDvacet = 0

int bitTricet = 0

int bitCtyricet = 0

int bitPadesat = 0

int bitSedesat = 0

int bitSedmdesat = 0

int bitOsmdesat = 0

int bitDevadesat = 0

int bitSto = 0

int bitKrok = 0

int bitNulaWordLamp = 0

int bitDesetWordLamp = 1

int bitDvacetWordLamp = 2

int bitTricetWordLamp = 3

int bitCtyricetWordLamp = 4

int bitPadesatWordLamp = 5
```

```

int bitSedesatWordLamp = 6

int bitSedmdesatWordLamp = 7

int bitOsmdesatWordLamp = 8

int bitDevadesatWordLamp = 9

int bitStoWordLamp = 10

int bitKrokWordLamp = 1

GetData(bitDeset, "PLC BBE4", "MArea-PrepinacRychlostPozice2(10)", 1)

GetData(bitDvacet, "PLC BBE4", "MArea-PrepinacRychlostPozice3(20)", 1)

```

```

GetData(bitTricet, "PLC BBE4", "MArea-PrepinacRychlostPozice4(30)", 1)

GetData(bitCtyricet, "PLC BBE4", "MArea-PrepinacRychlostPozice5(40)", 1)

GetData(bitPadesat, "PLC BBE4", "MArea-PrepinacRychlostPozice6(50)", 1)

GetData(bitSedesat, "PLC BBE4", "MArea-PrepinacRychlostPozice7(60)", 1)

GetData(bitSedmdesat, "PLC BBE4", "MArea-PrepinacRychlostPozice8(70)", 1)

GetData(bitOsmdesat, "PLC BBE4", "MArea-PrepinacRychlostPozice9(80)", 1)

GetData(bitDevadesat, "PLC BBE4", "MArea-PrepinacRychlostPozice10(90)", 1)

GetData(bitSto, "PLC BBE4", "MArea-PrepinacRychlostPozice11(100)", 1)

GetData(bitNula, "PLC BBE4", "MArea-PrepinacRychlostPozice1(0)", 1)

GetData(bitKrok, "PLC BBE4", "MArea-PrepinacRychlostPozice12(0)", 1)

if bitNula == 1 then

SetData(bitNulaWordLamp, "Local HMI", LW, 100, 1)end if

if bitDeset == 1 then

SetData(bitDesetWordLamp, "Local HMI", LW, 100, 1)end if

if bitDvacet == 1 then

SetData(bitDvacetWordLamp, "Local HMI", LW, 100, 1)end if

if bitTricet == 1 then

SetData(bitTricetWordLamp, "Local HMI", LW, 100, 1)end if

```

```

if bitCtyricet == 1 then

SetData(bitCtyricetWordLamp, "Local HMI", LW, 100, 1)end if

if bitPadesat == 1 then

SetData(bitPadesatWordLamp, "Local HMI", LW, 100, 1)

SetData(bitDeset, "PLC BBE4", "MArea-Mer_DobryKus", 1)end if

if bitSedesat == 1 then

SetData(bitSedesatWordLamp, "Local HMI", LW, 100, 1)end if

if bitSedmdesat == 1 then

```

```

SetData(bitSedmdesatWordLamp, "Local HMI", LW, 100, 1) end if

if bitOsmdesat == 1 then

SetData(bitOsmdesatWordLamp, "Local HMI", LW, 100, 1) end if

if bitDevadesat == 1 then

SetData(bitDevadesatWordLamp, "Local HMI", LW, 100, 1) end if

if bitSto == 1 then

SetData(bitStoWordLamp, "Local HMI", LW, 100, 1) end if

if bitKrok == 1 then

SetData(bitKrokWordLamp, "Local HMI", LW, 100, 1)end if

end macro_command

```

Toto makro přenáší vyhodnocení kruhového přepínače z PLC na obrazovku panelu. Funkce GETDATA získává data z programového bloku PLC BBE4. Uloží si paměť například tagu MArea-PrepinacRychlostPozice2(10) do vnitřní paměti která se jmenuje int bitDeset a je typu integer. Tuto hodnotu pokud je příslušný byt nastavený, tak funkcí SETDATA nastaví do vnitřního bytu hodnotu deset, kterou numerickým displejem zobrazíme.

## 5.5 Servopohony

Značku servopohonu jsme použili Schneider typ Lexium 32C. Tento typ je určen pro pulsní řízení. Základem pro správnou funkci servopohonu je správné nastavení proporcionálních, rychlostních, a pozicových zesílení. Dále se nastavují frekvenční filtry a proudy.

Při výběru velikosti servomotoru, záleží na momentu síly, jakou on sám dokáže vyvinout. Dále na momentu síly, kterou bude pohybovat. Moment motoru může být maximálně patnáctkrát menší než moment, se kterým bude pohybovat. Toto vše je z důvodu přetížení a přesnosti. Pokud bychom tento poměr nezachovali nedařilo by se nám správně nastavit proudové a poziční smyčky.

V našem případě je moment motoru desetkrát menší než moment zátěže. V další řadě záleží na maximální rychlosti a velikosti celého motoru. V našem případě jsme nebyli omezeni velikostí ani otáčkami. Pro náš případ jsem zvolil motor s menšími maximálními otáčkami a s větším momentem. Větší moment z důvodu zabránění kmitání od brusných kotoučů.

### 5.5.1 Základní nastavení parametrů

Základní nastavení parametrů spočívá v přepnutí řízení ze vstupu A/B na vstup P/D, to znamená pulse/direction neboli pulsy a směr. Další a nutný parametr v nastavení je parametr přepnutí serva do režimu RUN. V našem případě nastaven parametr AUTO. Hned po napájení se přepne do režimu RUN.

Mezi důležité parametry patří gear factor, neboli převodovka. Záleží na poměru pulsů na jednu otáčku, a jaká je ujetá dráha na jednu otáčku motoru. V našem případě máme na kuličkový šroub se stoupáním 10mm. Z toho plyne, že na jednu otáčku motoru ujedeme jeden centimetr. Výpočet převodového poměru:

$$\text{převodový poměr} = \frac{\text{jeden přírustek na motoru}}{\text{skutečně ujetý přírustek pohyblivé části}}$$

Základní počet pulsů na otáčku 16356. Z převodového poměru neboli převodovky a základního dělení pulsů na otáčku zjistím správný gear-ratio neboli přesný počet pulsů pro ujetí 0,001mm.

Mezi další parametry pro základní spuštění patří nastavení digitálních vstupů a výstupů. Toto nastavení neleze provést jen z operátorského panelu. Musí se připojit počítač se softwarem SoMove. Po připojení se dají nastavit příslušné vstupy a výstupy dle naší potřeby.

PC se servo měničem komunikuje pomocí speciálního převodníku, který nám převádí sériovou linku RS485 na USB.

## 5.5.2 Nastavení servo měniče

Pro základní nastavení musí být stroj v klidném stavu a musí mít motor prostor pro udělení pohybu o otáčku vpřed a vzad. Pokud máme připravený stroj, připojíme se k měniči a po zadání hesla vstoupíme do exklusivního přístupu. V obyčejném přístupu nelze provádět comfortní autotuning nebo automatické ladění. Po přihlášení spustíme comfort tuning a sledujeme výstup na připojeném osciloskopu. Po dokončení tuningu se nám servo hrubě naladí. V mnoha případech je slyšet rezonance vlivem harmonické frekvence, nebo po doteku na motor je cítit chvění. Tyto špatné vlivy se musí doladit pomocí osciloskopu.

Dle grafu na osciloskopu zjistíme harmonickou frekvenci kmitání serva. Podle hodnoty z grafu napíšeme tu samou hodnotu do frekvenčního filtru. Dojde ke zklidnění servomotoru. Pohybem a zapnutým osciloskopem bychom měli vidět, jak moc se nám servomotor odchyluje od zadané dráhy. Správně nastavené servo vypadá tak, že žádaná hodnota pozice je stejná jako ujetá dráha pozice a otáčkový parametr stojí na stejném místě. Čím lépe se nám podaří vyladit servo, tím lepší budeme mít přesnost.

## 5.6 Krokové motory

Krokové motory máme taky od firmy Schneider a použitý typ je SD68S.

Krokové motory používáme pro automatické přidání diamantu při orovnávání kotoučů. V každé úvratí orovnávacího mechanismu posune krokový motor diamant ve směru k brusnému kotouči. Tato akce se opakuje v tolika cyklech v jakých je údržba nastavila v dotykové panelu s nastaveným přírůstkem

U krokových motorů není žádné nastavování jako u servo motorů. Jediné nastavení spočívá v přiřazení počtu kroků na jednu otáčku. Dají se nastavit základní dělení pulsů na otáčku pomocí switch přepínače:

Možné kroky: 200, 400, 500, 1000, 2000, 4000, 5000, 10000.

V našem případě použijeme dělení 10000 pro získání přesnějšího přírůstku. Na jednu tisícinu milimetru potřebujeme deset pulsů.

Na rozdíl od servo motorů nemají krokové motory žádnou zpětnou vazbu o vykonání pohybu.

## 6 Závěr

Při návrhu řídicího systému jsem poznal jak složité je vymyslet celkový software ve prospěch obou stran. Kdy zákazník chce po programátorovi zbytečné věci nebo dokonce věci které nejdou naprogramovat. Postupem času se při vývoji udělal kompromis, aby to bylo schůdné pro obě strany.

Po dobu tvorby diplomové práce jsem nasbíral spoustu nových zkušeností ať už z manuálu nebo z internetových fór, tak ale i od zkušenějších programátorů. Při programování PLC značky Siemens mi nejvíce pomohl jejich vlastní help systém neboli nápověda.

Největším úskalím tohoto projektu byla společná komunikace mezi více hlavními jednotkami po vnitřní profinetové síti. Nejdůležitější bylo pochopit, jak probíhá přenos části paměti a jak ji v druhém PLC zpět rozparsovat na potřebné byty.

Celkově mě tato práce posunula dál ve vědomostech a naučila dostatečně používat nápovědu k programům vyhledávat v manuálech debugovat PLC v run režimu a přicházet na další a další zatím neobjevené funkce, které nám ve výsledku ulehčí práci a šetří náš čas.

## 7 Literatura

- [I] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. 1. vyd. Praha: BEN - technická literatura, 1999, 223 s. ISBN 80-860-5658-9.
- [II] ŠMEJKAL, Ladislav. PLC a automatizace. 1. vyd. Praha: BEN - technická literatura, 2005, 207 s. ISBN 80-730-0087-3.
- [III] PLC. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2013 [cit. 2016-08-16]. Dostupné z: <https://cs.wikipedia.org/wiki/PLC>
- [IV] ING. ZELINKA, Ondřej. Bezkontaktní měření rozměrů – optické mikrometry. *Automa: časopis pro automatizační techniku* [online]. 2009, **2009**(04), 1 [cit. 2016-08-16]. Dostupné z: [http://automa.cz/index.php?id\\_document=38860](http://automa.cz/index.php?id_document=38860)
- [V] AUTOR NEUVEDEN. *pohonnatechnika* [online]. [cit. 2016-08-16]. Dostupný na WWW: <http://www.pohonnatechnika.cz/skola/motory/servo-motor>
- [VI] PLC Fatek návody. PLC Fatek. [online]. 5.10.2015 [cit. 2016-10-05]. Dostupné z: <http://fatek.seapraha.cz/hlavni-strana/dokumentace/navody/fbs/>
- [VII] PLC AUTOMATIZACE. *PLC AUTOMATIZACE*. [online]. 16.8.2016 [cit. 2016-08-16]. Dostupné z: <http://www.plc-automatizace.cz/>
- [VIII] 134. Strojní obrábění. *Ou Horní Slavkov* [online]. Horní Slavkov: WEBHOUSE, s.r.o., 2010 [cit. 2016-08-16]. Dostupné z: [http://www.ouhornislavkov.cz/VismoOnline\\_ActionScripts/File.ashx?id\\_org=400056&id\\_dokumenty=1675](http://www.ouhornislavkov.cz/VismoOnline_ActionScripts/File.ashx?id_org=400056&id_dokumenty=1675)