

---

**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika  
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

**Rekonstrukce balení na výrobní lince  
„Miňonky“**

**Reconstruction of pacage on production line  
„Minonky“**

**Bakalářská práce**

Autor: **Bedřich Frieser**  
Vedoucí práce: Ing. Miloš Hernych  
Konzultant: Ing. Ladislav Pur

**V Liberci 7. 5. 2007**



# TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Katedra: Katedra řídicí techniky

Akademický rok: 2006/2007

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení: Bedřich Frieser

studijní program: B 2612 – Elektrotechnika a informatika

obor: 2612R011 – Elektronické informační a řídicí systémy

Vedoucí katedry Vám ve smyslu zákona o vysokých školách č.111/1998 Sb. určuje tuto bakalářskou práci:

Název tématu:

**Rekonstrukce balení na výrobní lince „Miňonky“**

Zásady pro vypracování:

1. Seznamte se se stávajícím stavem balící linky „Miňonky“ ve firmě Opavia – Lu, a.s. v Lovosicích.
2. Navrhněte modernizaci jejího stávajícího řídicího systému vč. softwarového vybavení s ohledem zejména na možnost budoucí změny výrobního sortimentu, rychlost rekonstrukce a uvedení modernizované linky do provozu.
3. Tento návrh zrealizujte a proved'te vyhodnocení.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah průvodní zprávy: cca 30-40 stran

Seznam odborné literatury:

[1] Firemní materiály firmy Siemens, AG.

[2] Firemní materiály firmy Omron corp.

[3] Firemní materiály firmy Lenze, s.r.o.

Vedoucí bakalářské práce: Ing. Miloš Hernych

Konzultant: Ing. Ladislav Pur

Zadání bakalářské práce: **18.10.2006**

Termín odevzdání bakalářské práce: **18. 5. 2007**

L.S.

.....  
Vedoucí katedry

.....  
Děkan

V Liberci dne 18. října 2006

## Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis



## **Poděkování**

Na tomto místě bych chtěl poděkovat zejména firmě ESP Lovosice, která mi umožnila seznámení se s aplikací průmyslové automatizace v praxi a realizaci celé Bakalářské práce a Ing. Ladislavu Purovi, za jeho pomoc při vývoji prototypu a řešení některých softwarových nejasností.

Dále bych rád poděkoval společnosti Opava-LU a.s. záводу DELI Lovosice za umožnění bezproblémového přístupu do výrobních prostor k odladění prototypu a následně i celého projektu.

## **Použitý software**

Pro vývoj řídicího software PLC Siemens Simatic S7 200 byl použit program STEP 7 MicroWIN verze 4.0 – Service pack 4.

Pro vývoj ovládacího software touch panelu Siemens TP 177micro byl použit program SIMATIC WinCC flexible 2005 Micro.

Pro kresbu ilustračních obrázků byl použit program

Microsoft® Malování, součást instalačního balíku operačního systému Microsoft® Windows XP Home Edition.

## **Kontakt**

E-mail : [beda.frieser@seznam.cz](mailto:beda.frieser@seznam.cz)





## **Anotace**

Cílem bakalářské práce bylo zrychlit a zefektivnit balicí část linky miňonky s ohledem na prostorovou náročnost, flexibilitu baleného sortimentu a efektivitu balení. Původní rozestavení linky zabíralo celou klimatizovanou halu a to především z toho důvodu, že na dva druhy baleného zboží se využívaly čtyři baličky Cavanna typu 05 a na jeden druh zboží čtyři baličky Cavanna typu 07. Navrženým a realizovaným řešením se podařilo docílit využití novějšího typu baliček, a to prvně zmiňovaných Cavanna 05, na celý vyráběný sortiment této linky. Koncepce daného problému byla vyřešena sestrojením čtyř dopravníků o specifických vlastnostech, jejichž orientací a rychlostí se docílí efektivní dopravy zboží do zubů řetězu baliček.

## **Abstract**

The aim of the bachelor's study is to speed up and increase efficiency package line Minonky with consideration on place demandingness, flexibility packaged assortment and effectivity. Original disposition of line take all space in air-conditioned hall because for packaging two kind of assortment was in use eight packaging machines. Four for one kind of assortment and other four for second kind of assortment. Suggest and implement solution was reached use only four packaging machines for all assortment. Conception of problem was solve with use four special conveyor belts, whose orientatin and speed is accomplished effective tarnsport product to packaging machine.



# Obsah

<b>1 Úvod</b>	13
<b>1.1 Původní řešení linky</b>	13
1.1.1 Rozestavení baliček	13
1.1.2 Způsob dopravy zboží do baliček	13
<b>1.2 Nové řešení linky</b>	14
1.2.1 Důvody nového řešení	14
1.2.2 Nové rozestavení baliček	14
1.2.3 Varianta Čokotatranky	15
1.2.4 Varianta Miňonky	15
<b>2 Hardware</b>	19
<b>2.1 Silová část</b>	19
2.1.1 Asynchronní motory	19
2.1.2 Frekvenční měniče	20
2.1.3 Servomotory	22
2.1.4 Servoměniče	23
2.1.5 Jištění	23
<b>2.2 Ovládací část</b>	24
2.2.1 PLC	24
2.2.2 Operátorské panely	26
2.2.3 Senzorika	27
<b>3 Software</b>	31
<b>3.1 PLC</b>	31
3.1.1 Základní parametry	31
3.1.2 Způsoby programování	34
3.1.3 Program	35
3.1.3.1 Některé použité rutiny	35
3.1.3.2 Program Čokotatranky	38
3.1.3.3 Program Miňonky	40
<b>3.2 TP</b>	41
3.2.1 Základní parametry	41
3.2.2 Způsob programování	41
3.2.3 Obrazovky	43
<b>4 Realizace</b>	47
4.1 Montáž zařízení	47
4.2 Odladění	47
4.3 Servisní zásahy	48
<b>5 Závěrečné zhodnocení</b>	49
5.1 Nesrovnalosti s teoretickými předpoklady	49
5.2 Práce v areálu závodu	49
5.3 Závěr	49



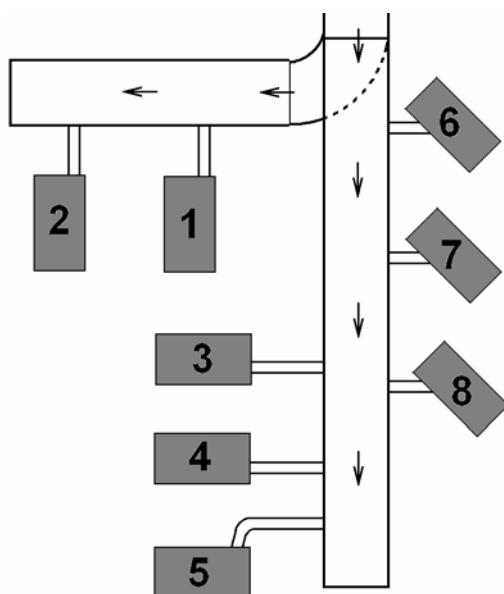
# Kapitola 1

## Úvod

Balící linky jsou dnes nedílnou součástí každého provozu v téměř všech odvětvích průmyslu. V potravinářství představují velice důležitou finální úpravu výrobku a to jak z hlediska kvality balení pro uchování trvanlivosti výrobku, tak především z hlediska estetiky.

### 1.1 Původní řešení linky

#### 1.1.1 Rozestavení baliček



Obrázek 1.1 : Rozestavení baličích strojů před úpravou

Na obrázku [1.1] vidíme názorně, jak prostorově náročné bylo uspořádání linky.

#### Původní varianta Miňonky

V první variantě se balily Miňonky po pěti, případně po šesti kusech a

využívalo se baliček 1,2,3 a 4. Při vstupu zboží do haly, takřka hned za chladícím tunelem, byl sklápějící se dopravník, který každou druhou řadu sklápěl na příčný dopravník a tím zajišťoval rovnoměrné zásobování baliček.

#### Původní varianta Čokotatranky

Ve druhé variantě se balily Čokotatranky jednotlivě. V tomto případě se využívalo baliček 5,6,7 a 8. Klapka byla v tomto případě pouze v horní poloze a využíval se tak jen hlavní dopravník.

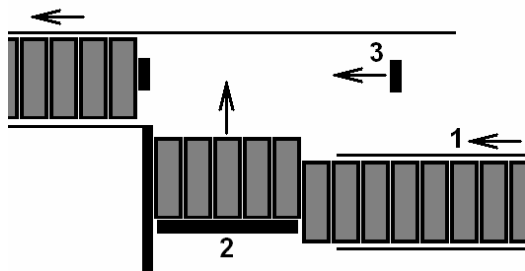
#### 1.1.2 Způsob dopravy zboží do baliček

##### Miňonky



Obrázek 1.2 : Řada Miňonky

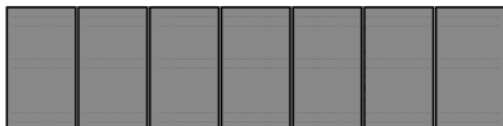
Řady zboží vyjžděly z chladícího tunelu jak je uvedeno na obrázku [1.2]. Operátorka pomocí upraveného hrábľa stáhla řadu na přivázející dopravníček baličky. Zde se vytvářela potřebná zásoba pro plynulý chod. Za tímto dopravníčkem byl klikový mechanismus znázorněný na obrázku [1.3], který po najetí oplatek z přivázejícího dopravníčku (1) na pevný doraz posouval ramenem (2) jejich nastavený počet do zubů řetězu baličky (3).



Obrázek 1.3 : Klikový mechanismus

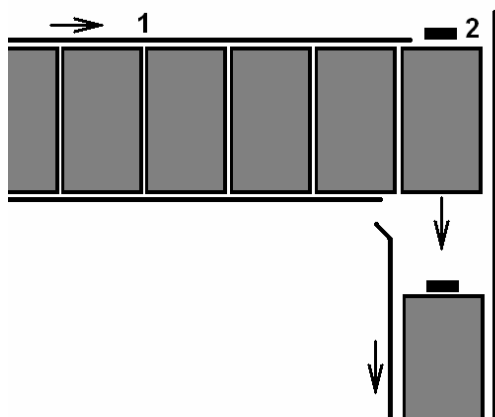
## Čokotatranky

U tohoto sortimentu vyjízděly řady z chladicího tunelu ve tvaru uvedeném na obrázku [1.4].



Obrázek 1.4 : Řada Čokotatranky

Operátorky, jak je uvedeno na obrázku [1.5], takto srovnanou řadu ručně přesunuly na přivázečí dopravníček baličky (1), který vedl přímo do řetězu baličky (2).



Obrázek 1.5 : Vstup zboží do baličky při variantě Čokotatranky

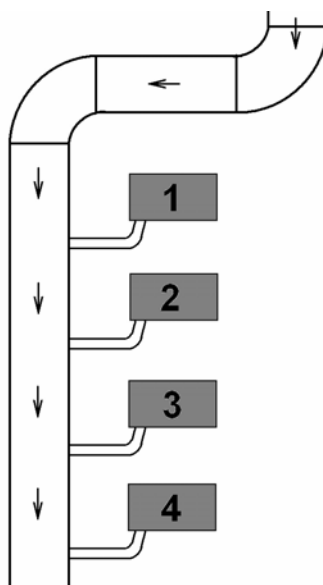
## 1.2 Nové řešení linky

### 1.2.1 Důvody nového řešení

Jak je patrné, na dva druhy sortimentu bylo celkem použito osmi balících strojů, což je vzhledem k nákladům na pořízení a prostorovou náročnost velice neefektivní. Dalším důvodem byla energetická náročnost provozu, jelikož se jedná o klimatizované pracoviště na kterém je nutné udržovat teplotu přibližně 18°C, která zaručuje tuhost čokolády na celomáčeném zboží. Je zde také nutno poznamenat, že různé druhy sortimentu jsou vyráběny v poměrně rychle za sebou se měnícím sledu a tak je třeba zajistit bezproblémové přejetí z jednoho druhu na druhý s co nejmenší časovou ztrátou.

### 1.2.2 Nové rozestavení baliček

Jako první bod řešení jsme zvolili nové prostorové uspořádání linky, jak je znázorněno na obrázku [1.6].

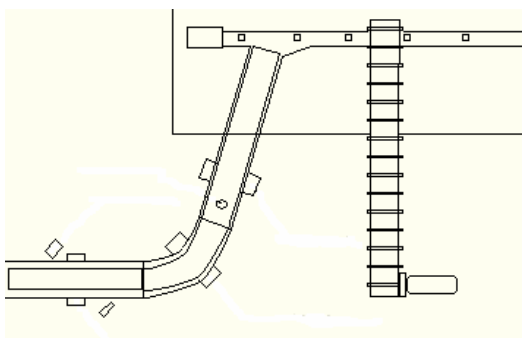


Obrázek 1.6 : Nové uspořádání baliček

Pokud toto porovnáme z původním rozestavením, je na první pohled patrná značná úspora prostoru, což samozřejmě vyplývá především z použití pouze čtyř baliček namísto osmi. V tomto okamžiku bylo nutné zajistit, aby tyto balily veškerý druh sortimentu. Konstrukce balících strojů je velice robustní a tak změna sortimentu na stroji jako takovém není obtížná. Problém se týká především řešení dopravníků přivážejících zboží do řetězu baličky.

### 1.2.3 Varianta Čokotatranky

Jak si můžeme všimnout na obrázku [1.4], zboží vstupuje do řetězu baličky přímo. Původní pozice baliček toto dovolovala, nicméně nové řešení počítá s otočením baličky o 90°. Po konzultaci s odborníky na strojní část úpravy linky bylo jako řešení tohoto problému zvoleno použití kruhového dopravníku, který nám tak zaručí požadovanou orientaci zboží. Jak je uvedeno na obrázku [1.7], celkové řešení se skládá z prvního přivážecího dopravníku, kruhového dopravníku a přivážecího dopravníku baličky.

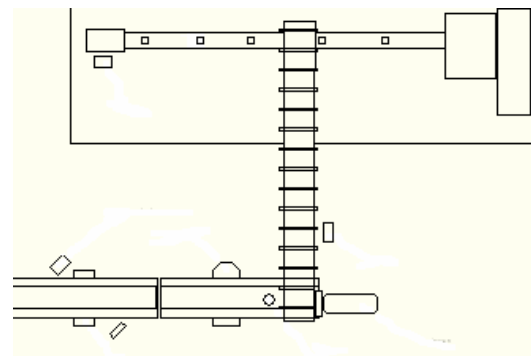


Obrázek 1.7 : Řešení přivážecích dopravníků – varianta Čokotatranky

### 1.2.4 Varianta Miňonky

Při porovnání sortimentů Čokotatranky a Miňonky je na první

pohled patrné, že jejich orientace se po výstupu z chladicího tunelu liší o 90°. Navíc ve druhém případě je třeba ještě odpočítávat přesný počet kusů do jednoho balení. V původním řešení bylo toto prováděno pomocí klikového mechanismu. Obrázek [1.3]. Ten byl však jak mechanicky tak i prostorově náročný a rychlost do které byl použitelný byla omezena. V principu však můžeme říci, že po dojezdu zboží na pevný doraz rameno mechanismu, nastavené do potřebné šířky odpovídající počtu oplatek, tyto vsunulo do řetězu baličky, zvedlo se nad již nově přijíždějící oplatky, přesunulo se před ně a opět provedlo stejný cyklus. Nejdůležitějším aspektem tohoto pohybu je tedy přesné odpočítání oplatek a bezkolizní přesun do řetězu baličky.

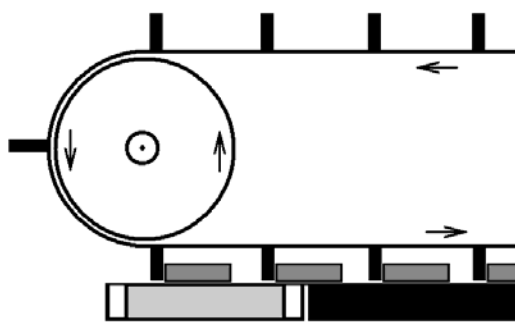


Obrázek 1.8 : Řešení přivážecích dopravníků – varianta Miňonky

Po konzultaci a přednesení několika návrhů jsme se shodli na řešení znázorněném na obrázku [1.8]. Jako první dopravníček jsme použili stávající z varianty Čokotatranky. Jako druhý, v tomto případě krokový dopravníček, jsme použili užší dopravník poháněný servomotorem a jako finální, třetí dopravník jsme použili lopatkový pás zavěšený nad deskou z chlazeného, duplikovaného plechu poháněný taktéž servomotorem.

## Popis cyklu

V první fázi nám zboží operátorka přesune z hlavníh pasu na přivázečí dopravníček, který jede stále. V momentě, kdy dorazí na konec dopravníčku, spustí se druhý dopravník stejnou rychlostí jako první. Jakmile dorazí zboží na konec tohoto dopravníku, balička se rozjede. Oplatky dojedou na doraz a lopatkový dopravník udělá jeden krok. Po jeho dokončení krokový dopravníček jedním krokem přesune zboží opět na doraz a cyklus se takto opakuje. Pohled na přechod krokového a lopatkového dopravníku nám nabízí obrázek [1.9].



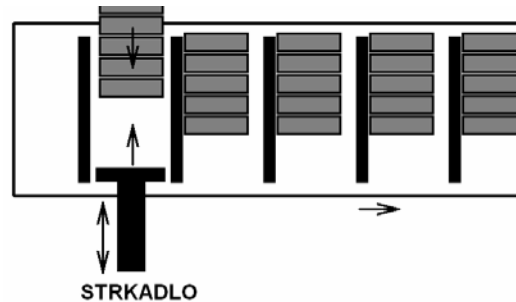
Obrázek 1.9 : Pohled na přechod krokového a lopatkového dopravníku

## Změna počtu oplatek

V takto vyřešeném přechodu mezi krokovým a lopatkovým dopravníkem nám však zůstával problém, jak docílit balení proměnného počtu oplatek. Pro řešení se nám nabízelo několik variant.

## Strkadlo

Jako první způsob jsme začali na prototypu odladovat takzvané strkadlo. V principu nám tento mechanismus znázorňuje obrázek [1.10].



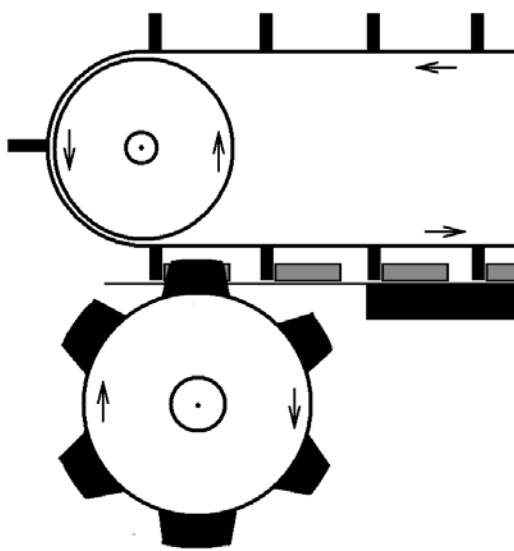
Obrázek 1.10 : Princip funkce strkadla (pohled ze spodu)

Po dokončení kroku lopatkového dopravníku došlo k vyjetí strkadla z výchozí pozice zároveň s krokem přivázejícího krokového dopravníku. Strkadlo bylo v pozici dříve než zboží na dopravníčku a sloužilo tak jako doraz. Po dokončení kroku dopravníčku a jeho zastavení strkadlo zajelo opět do výchozí pozice a v tomto momentu bylo možné udělat opět krok lopatkového dopravníku. Proměnný počet oplatek se tak dal ovlivnit vyjetím strkadla do patřičné pozice, tedy pro více oplatek stačilo menší vyjetí strkadla. Toto řešení z důvodu použití dalšího servopohonu pro strkadlo bylo ovšem finančně náročné a po testování ve zkušebním provozu nevyhovující. Časový sled událostí nebylo možno tak efektivně sladit, abychom dosáhli vyšších rychlostí a požadované efektivity. Navíc realizace strkadla byla i po strojní stránce poměrně náročná a nepodařilo se ji přizpůsobit našim požadavkům.

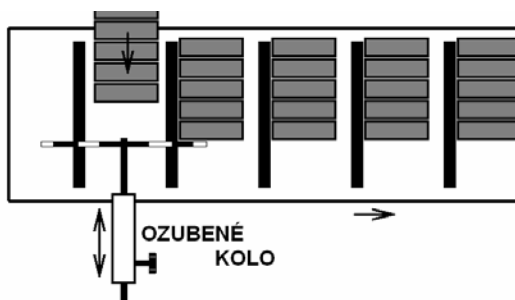
## Odvalovací ozubené kolo

Druhým způsobem dorazu, který jsme testovali, bylo odvalovací ozubené kolo sloužící jako doraz. Jeho funkci si opět znázorníme na obrázcích [1.11] a [1.12].





Obrázek 1.11 : Odvalovací ozubené kolo (pohled ze strany)



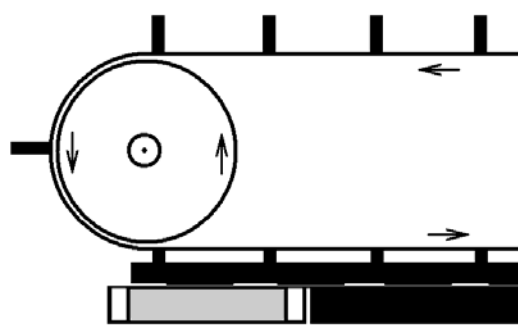
Obrázek 1.12 : Odvalovací ozubené kolo (pohled ze spodu)

Ozubené kolo bylo zhotoveno z nerezového plechu tloušťky 3 mm. Krokový dopravník tak v tomto případě končil ještě před lopatkovým dopravníkem a zboží posouval na plech, který byl součástí ozubeného kola. Vysunutím kola se nastavil počet oplatek. V takto provedené konstrukci se kolo odvalovalo pomocí lopatek lopatkového dopravníku bez vlastního pohonu. Tato varianta byla již provozu schopná, nicméně opět zde nastalo několik problémů. Lopatky dopravníku jsou zhotoveny z pružného PVC a tak docházelo kontaktem s ozubeným

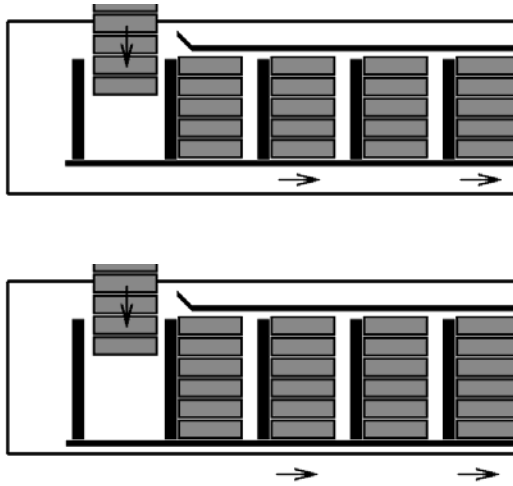
kolem k jejich poškozování. Je třeba si uvědomit, že rychlost kroku lopatkového dopravníku je přibližně 3 m/s a nerezové ozubené kolo má i při tak slabé konstrukci v tomto případě nezanedbatelnou setrvačnost. Díky této skutečnosti a díky elasticitě lopatky docházelo také na konci kroku lopatkového pásu k oscilaci kola. Kolo se tedy setrvačností odrazilo od lopatky předcházející, vrátilo se na dotek s lopatkou hnací, která ho opět odrazila a to se opakovalo několikrát v krátkém časovém úseku, než došlo k ustálení. Tím více však docházelo k poškození lopatek. Po zjištění těchto neodstranitelných překážek jsme byli tedy nuceni k realizaci varianty třetí.

### Pevné dorazy a vyměnitelné lopatkové pásy

Původně navrhovanou, avšak z důvodu časové náročnosti výměny pásů nepodporovanou, variantu jsme tedy byli nuceni použít. Nástin této konstrukce lépe pochopíme na obrázcích [1.13] a [1.14].



Obrázek 1.13 : Pevné dorazy (pohled ze strany)



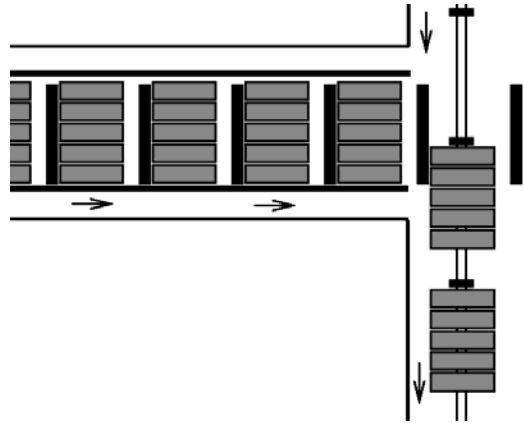
*Obrázek 1.14 : Pevné dorazy (pohled ze spodu)*

V tomto provedení je přesně definována délka lopatky dopravníku a pevný, ovšem přizpůsobitelný doraz. Oplatky tak krokový dopravník naveze přímo na doraz, kde si je lopatka, široká podle toho, kolik zboží se právě balí, převezme. Hlavním problémem tohoto řešení byla právě skutečnost, že pro různý počet oplatek je nutné použít pas s různě dlouhými lopatkami, jak je znázorněno na obrázku [1.14]. Po důkladném rozboru s projektanty strojních součástí a vedoucími pracovníky závodu byla nakonec schválena, neboť se po strojní stránce podařilo konstrukčně upravit vedení pasu tak, aby bylo možné jej během poměrně krátké doby efektivně vyměnit. Navíc bylo konstatováno, že proměnné množství oplatek se balí pouze pro propagační účely a tedy ne při běžném provozu.

### **Přechod oplatek do řetězu baličky**

Po projetí zboží lopatkovým dopravníkem je ještě potřeba zajistit jejich přechod do řetězu baličky. Jak znázorňuje obrázek [1.15], po dokončení kroku lopatkového dopravníku je zboží umístěno přesně ve dráze řetězu baličky. Následující krok

lopatkového dopravníku je možný však až po projetí zubu přes dráhu lopatkového dopravníku, aby nedošlo k poškození lopatky a zároveň byl zaručen prostor pro oplatky.



*Obrázek 1.15 : Přechod zboží z lopatkového dopravníku do řetězu baličky (pohled ze spodu)*

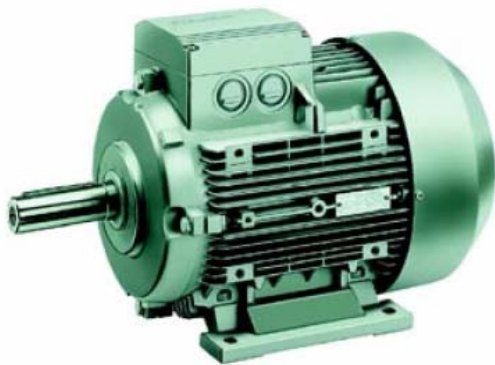
## Kapitola 2

# Hardware

Po návrhu strojních součástí pro přivážející dopravníky zboží do balíčků bylo nutné navrhnout pohony jednotlivých dopravníků a jejich řízení.

### 2.1 Silová část

#### 2.1.1 Asynchronní motory



Obrázek 2.1 : Asynchronní motor

#### Konstrukce

Každý trojfázový asynchronní motor je složen ze dvou hlavních částí.

**Stator** (pevná část) - je u většiny typů prakticky stejný. Je složen z nosné kostry motoru, svazku statorových plechů a statorového vinutí.

**Rotor** (pohyblivá část) - hřídel s nalisovanými rotorovými plechy. Bývá označována jako *kotva*. **Kotva na krátko** - v drážkách rotoru jsou nalisovány neizolované měděné, mosazné nebo hliníkové tyče, které jsou na obou koncích spojeny zkratovacími kroužky. Tyče spolu s kroužky mají podobu klece (*klecový rotor*).

**Kroužková kotva** - na hřídeli jsou

kromě svazku rotorových plechů i sběrné kroužky. V drážkách plechů je uloženo trojfázové vinutí rotoru z izolovaných vodičů, které je zapojeno většinou do hvězdy, zřídka do trojúhelníka. Na tři sběrací kroužky je připojeno vinutí rotoru, ke kterým lze připojit činné odpory sloužící k rozběhu motoru.

#### Princip činnosti

Základem činnosti asynchronního motoru je vytvoření točivého magnetického pole, které vznikne průchodem střídavého trojfázového proudu vinutím statoru. Toto magnetické pole indukuje v rotoru napětí a vzniklý proud vyvolává sílu otáčející rotorem. Otáčky točivého pole jsou dány kmitočtem napětí odebíraného ze sítě a počtem pólů trojfázového motoru.

$$n_s = \frac{f}{p} \quad (2.1)$$

Kde  $f$  je kmitočet proudu a  $p$  je počet pólových dvojic statoru. Rotor se nikdy nemůže otáčet stejnými otáčkami jako magnetické pole statoru. Pokud by se tak stalo, pak by se rotor a magnetické pole vůči sobě nepohybovaly a tím by se ani neindukovalo napětí a nevznikala by točivá síla. Míra rozdílu otáček pole a rotoru je nazývána *skluz*, udávána v procentech a definována jako:

$$s = \frac{n_1 - n}{n_1} \cdot 100 [\%] \quad (2.2)$$

kde  $n_1$  jsou otáčky magnetického pole statoru,  $n$  jsou otáčky rotoru.

## Regulace otáček

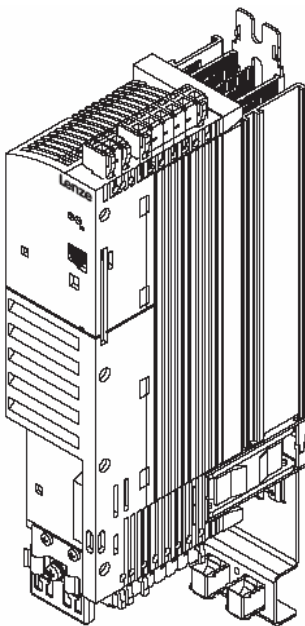
$$n = n_1 \cdot (1 - s) = \frac{f_1}{p} \cdot (1 - s) \quad (2.3)$$

Regulace změnou kmitočtu - používá se u motorů s kotvou na krátko. Připojením měniče kmitočtu můžeme řídit napětí a tím i vytvářené magnetické pole statoru. Při použití frekvenčního měniče rozlišujeme dva režimy. *Skalární řízení* - lze nastavovat velikost magnetického toku. *Vektorové řízení* - kromě velikosti magnetického toku lze nastavovat i jeho směr a tím můžeme docílit plynulou změnu otáček při jakémkoliv režimu práce a zatížení. Jde o nejdokonalejší způsob řízení otáček a lze jím docílit i otáček nadsynchronních.

## Použití

V našem případě jsme použili trojfázové asynchronní motory řízené frekvenčními měniči jako pohony pro první přivážecí dopravník pro obě varianty a pro kruhový dopravník ve variantě Čokotatranky.

### 2.1.2 Frekvenční měniče



Obrázek 2.2 : Frekvenční měnič

Pro regulaci otáček motorů a tím v našem případě rychlosti pasů jsme použili frekvenční měniče od firmy Lenze GmbH & Co KG typu 8200 Vector. Pro názornost je uveden na obrázku [2.2]. K jeho ovládání je třeba použít funkční panel, který nám umožňuje zadávat námi požadovanou frekvenci pomocí analogového signálu a samotné řízení měniče pomocí digitálních vstupů. Pokud bychom měnič používali v podmínkách, které to vyžadují, jsou zde ještě k dispozici analogový výstup a digitální výstup. Standartní funkční panel je na obrázku [2.3].



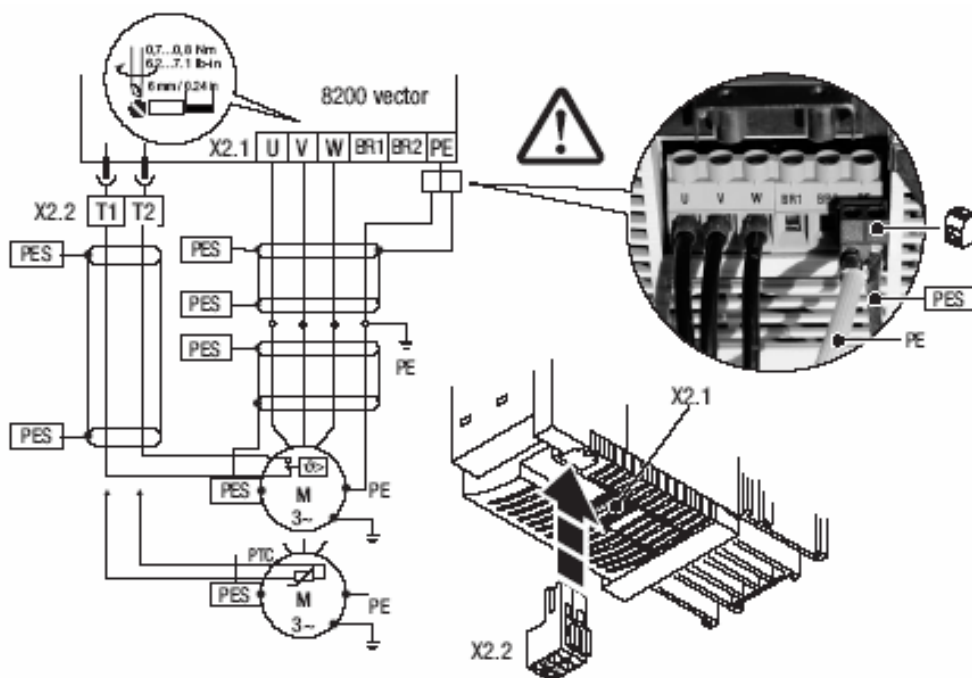
Obrázek 2.3 : Funkční panel frekvenčního měniče Lenze 8200 Vector

Pro řízení měničů využíváme v našem případě dvou digitálních signálů a jednoho analogového. Prvním digitálním signálem spouštíme a zastavujeme motor, druhým pak přepínáme sadu parametrů v měniči. Analogovým signálem řídíme přímo výstupní frekvenci měniče a zároveň tak tedy i otáčky motoru.

Parametry měniče jsou pro úplnost uvedeny v tabulce [2.1].

spínací frekvence	1**, 2, 4, 8, 16 kHz						
stupeň krytí	IP20						
EMK(stupeň odrušení A/B podle EN55011)	integrován (případně s kompaktními filtry pro montáž pod měnič)						
řídící a regulační vlastnosti	volitelně vektorová regulace, řízení podle charakteristiky U/f (lineární, kvadratické), regulace točivého momentu						
teplota okolí	-10 °C...+55 °C (40 °C...55 °C s redukcí výkonu)						
standardní vlastnosti	PTC vstup, PID regulátor, integrovaný brzdový tranzistor, automatické zjišťování a nastavení parametrů motoru, perioda snímání stavu digitálních vstupů 1 ms, programovatelný reléový výstup, rozběhové rampy ve tvaru S, invertování úrovní signálů, blokování rezonančních frekvencí, pevné otáčky, čtyři parametrové sady přepínatelné online, ochrana heslem, bipolární zadávání žádané hodnoty, ...						
pohonné vlastnosti 0,25-11 kW	1,8 × M <sub>men</sub> (60 s), rozsah nastavení momentu 1:10 pro 3...50 Hz, rozsah nastavení otáček 1:50 s M <sub>men</sub> (50 Hz), kruhovitost běhu ± 0,1 Hz						
pohonné vlastnosti 15-90 kW	vysoký rozběhový moment (2,1 × M <sub>men</sub> (3 s) po odblokování regulátoru), 1,8 × M <sub>men</sub> (60 s), rozsah nastavení momentu 1:10 pro 3...50 Hz, rozsah nastavení otáček 1:50 s M <sub>men</sub> (50 Hz), kruhovitost běhu +/- 0,1 Hz						
moduly	funkční moduly I/O	analogové vstupy	analogové výstupy	digitální vstupy	digitální výstupy	frekvenční výstupy	
		standardní I/O PT	1	1	4*	1	-
		aplikační I/O PT	2	2	6*	2	1
	funkční moduly pro datovou komunikaci	* obsahuje volitelně 1 frekvenční vstup					
		PROFIBUS-DP PT, INTERBUS PT, CAN PT (Systembus), CAN-I/O (Systembus PT), CAN-I/O RS PT (Systembus), AS-Interface PT, LECOM-B PT (RS485),					
komunikační moduly	PROFIBUS-DP, INTERBUS, INTERBUS-Loop, DeviceNet/CANopen, CAN, LON, LECOM-A/B (RS232/485), LECOM-LI (světlovody)						
ovládací jednotka	Keypad XT						
certifikace	CE, UL, cUL						

Tabulka 2.1 : Parametry frekvenčního měniče Lenze 8200 Vector



Obrázek 2.4 : Zapojení silové části frekvenčního měniče Lenze 8200 Vector

## 2.1.3 Servomotory



Obrázek 2.5 : Servomotory Omron Smartstep

**Servomotor** zkráceně **Servo** je motor pro pohony (většinou elektrické, ale existují také hydraulická, pneumatická či dokonce parní serva), u kterých lze na rozdíl od běžného motoru nastavit přesnou polohu natočení osy. Ovládají se jím například posuvy u CNC strojů. Elektrické servomotory jsou řízeny prakticky výhradně tranzistorovými měniči s pulzní šířkovou modulací. Poloha hřídele servomotoru je zjišťována elektricky pomocí fotoelektrického snímače (encoder) nebo pomocí rozkladače (selsynu). Pro levné aplikace lze použít optické snímání pomocí kódového kotoučku či proužku. Signál snímače polohy je přiveden pomocí zpětné vazby na regulátor, který porovnává skutečnou polohu motoru s žádanou polohou. Na základě rozdílu žádané a skutečné polohy regulátor (často velmi složitý) řídí měnič a tak nastavuje motor na žádanou polohu.

**Střídavé servomotory** (AC) jsou dnes nepoužívanější typ servomotorů. Jsou to synchronní motory (bez kartáčů) s permanentními magnety na rotoru a třífázovým vinutím ve statoru. Dnešní konstrukce motorů používají permanentní magnety na bázi vzácných zemin (nejčastěji typ neodym - železo - bór). Motory lze několikanásobně momentově přetížít a proto jsou vhodné pro dynamicky náročné úlohy.

## Specifikace modulů SmartStep

**Výkon 30–750 W**

**Napájení 100 V<sub>st</sub> a 220 V<sub>st</sub>**

Jednofázové a třífázové pohony 200 V<sub>st</sub>

**Dva typy motorů:**

Válcové (pro obecné aplikace). Ploché (v aplikacích, kde je prioritou zástavbový prostor)

**3 000 ot./min. (jmenovitá hodnota), ve špičkách 4 500 ot./min.**

**300% kroucí moment ve špičkách**

**Vstup pulsů**

Snadné řízení libovolným pulsním výstupem.

**Automatické ladění za provozu**

Pohon plynule přizpůsobuje zesílení podle změn zatížení.

**Snadné používání:**

Při základním nastavení nejsou třeba žádné úpravy.

Nastavení pomocí přepínačů DIP:

Zesílení

Vypnutí a zapnutí automatického ladění za provozu

Nastavení rozlišení:

500/1 000/5 000/10 000 pulsů/ot.

Dynamická brzda

Jediný kabel od ovladače k motoru.

Široké spektrum řídicích kabelů pro připojení ke všem typům řídicích systémů Omron.

**Snadné sledování**

Softwarový nástroj (Wmon) pro sledování a programování servopohonu.

**Snadný výběr**

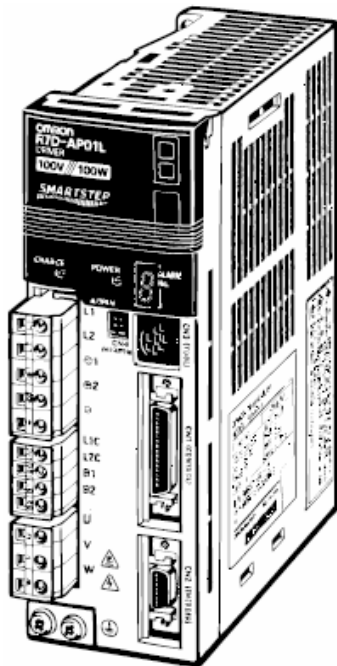
Nástroj pro určení správné velikosti servomotoru pro danou aplikaci.

**Volitelná parametrizační jednotka**

Funkce kopírování.

**Splňuje mezinárodní normy včetně norem CE, UL a cUL**

## 2.1.4 Servoměniče



Obrázek 2.6 : Servoměnič Omron SmartStep

Servoměniče jsou speciální druhy frekvenčních měničů určené pro řízení servomotorů. Pracují na principu zpětné vazby od motoru, která přesně definuje pozici motoru. Jako snímače polohy se pro tyto případy nejčastěji používají selsyny, resolvery, nebo inkrementální čidla (encoder). V našem případě řídíme pohyb motoru takzvaným režimem krok a směr. Princip tohoto řízení je použití dvou digitálních vstupů. Prvním řídíme směr otáčení servomotoru. Druhý využíváme jako pulsní vstup, kdy pro jedno otočení servomotoru je definován určitý přesný počet impulsů, například 1000 pulsů/ot. V tomto případě odpovídá jeden impuls otočení hřídele o 0,00628 rad. Pokud bychom tedy poslali na pulsní vstup měniče například 10000 impulsů, hřídel se nám celkem desetkrát otočí. Rychlost otáčení je pak dána frekvencí pulsů.

## 2.1.5 Jištění

**Jistič** je elektrický přístroj, který při nadproudu (přetížení nebo zkratu) automaticky rozpojí elektrický obvod a tím jej chrání před poškozením. Na rozdíl od pojistky, která musí být vyměněna, lze jistič znovu zapnout a obnovit tak dodávku proudu do elektrického obvodu.



Obrázek 2.7 : Příklad konstrukce jističe

1 - ovládací páčka

2 - aretační mechanismus

3 - kontakty

4 - přívodní šroubová svorka

5 - bimetalový člen pro vybavení přetížením

6 - regulační prvek nastavení citlivosti

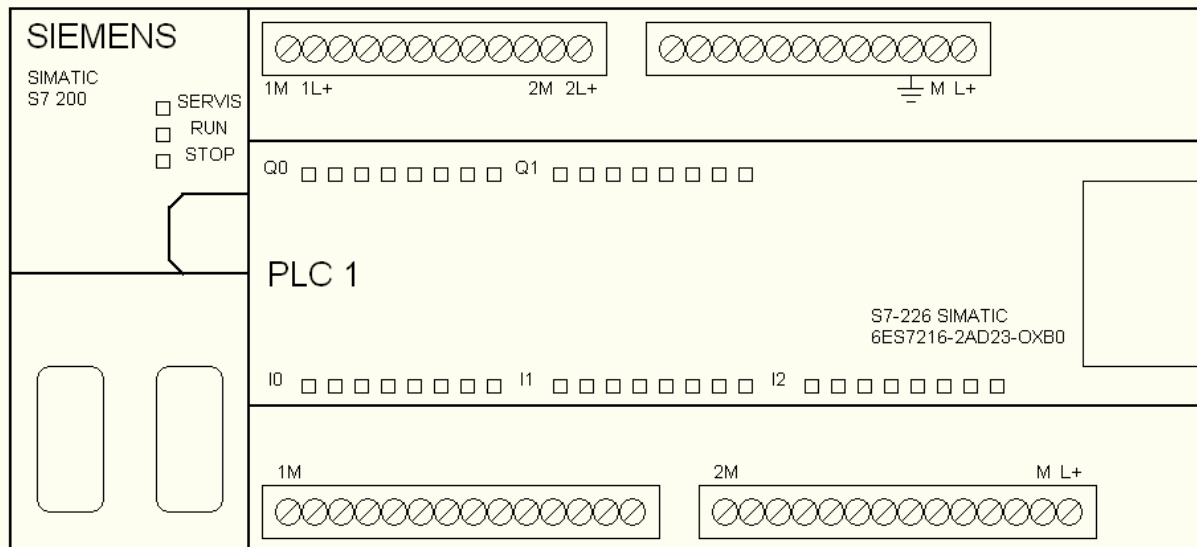
7 - elektromagnetická spoušť pro vybavení zkratem

8 - zhášecí komora

Pro jištění jednotlivých zařízení jsme použili celkem šesti jističů. Pro jištění motorů a servomotorů jističe o jmenovité hodnotě proudu 16 A. Pro interní zásuvku rozvaděče jistič o jmenovité hodnotě proudu 6 A a pro zdroj stejnosměrného napětí jistič o jmenovité hodnotě proudu 10 A. Pro jištění stejnosměrného obvodu pak jistič o jmenovité hodnotě proudu 4 A.

## 2.2 Ovládací část

### 2.2.1 PLC



Obrázek 2.8 : PLC Simatic S7 226

#### PLC obecně

Původně byly určeny k programovému řešení jednoduchých logických obvodů, dnes je jejich použití mnohem širší. Umožňují provádět kromě základních logických funkcí i matematické operace, přesuny bloků dat, zpracovávat spojité signály, signály ze speciálních zařízení (např. CCD kamera, impulsní snímače polohy, selsyny, atd.). Často jsou součástí většího řídicího celku, tzv. distribuovaného řídicího systému, jehož jednotlivé součásti jsou propojeny soustavou sítí. Použití PLC je velmi široké, od jednoduchých zařízení realizujících logické funkce např. při řízení kotelen a klimatizačních jednotek, přes aplikace ve sklářském a potravinářském průmyslu, až po PLC zabudované jako subsystém v CNC systémech pro řízení obráběcích strojů.

#### Hlavní požadavky

**Spolehlivost** - požadavek na „bezporuchový“ chod - výpadek řídicího systému mnohdy není přípustný a udává se parametrem MTBF (střední doba mezi poruchami) nebo MTBR (střední doba mezi opravami) - řádově desítky až stovky tisíc hodin.

#### Prizpůsobení HW potřebám uživatele

- možnost postavit PLC podle potřeb řízené technologie

- široký sortiment CPU, I/O jednotek, komunikačních jednotek, napájecích zdrojů, atd.

#### Odolnost proti ztíženým pracovním podmínkám a rušení

- výběr elektronických součástek a konektorů (svorkovnic)

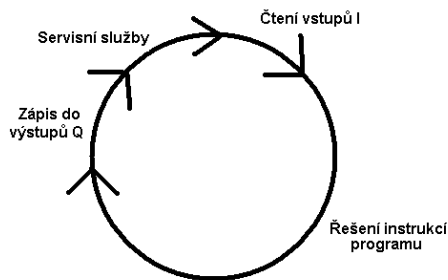
- krytí, elektromagnetické stínění

#### Cyklický pracovní režim

Programovatelný automat pracuje v uzavřeném cyklu. U většiny řídicích systémů probíhají postupně tyto činnosti:



- načtení signálů ze vstupních jednotek do vstupních registrů I automatu
- zpracování instrukcí programu
- zápis obsahu výstupních registrů Q do výstupních jednotek
- servisní služby automatu (příprava CPU k řešení dalšího cyklu)



Obrázek 2.9 : Schéma cyklického zpracování programu v PLC

#### Důsledky použití cyklického režimu:

- ochrana před „hazardy“
- stabilita programu
- nižší rychlost

#### PLC Simatic S7 226

SIMATIC S7-200 je řada malých programovatelných automatů (mikro - PLC), určených k řízení v různých automatizačních aplikacích. Zařízení S7-200 monitoruje vstupy a řídí výstupy pomocí uživatelského programu, který může obsahovat Booleovu logiku, počítání, časování, složité matematické operace a komunikaci s jinými inteligentními zařízeními. Kompaktní design, flexibilní konfigurace a výkonný instrukční soubor jsou důvody, proč je zařízení S7-200 optimálním řešením pro řízení široké škály aplikací.

Parametr	CPU 226
Rozměry (mm)	190 x 80 x 62
Paměť pro program: s editací v režimu RUN bez editace v režimu RUN	16384 bytů 24576 bytů
Paměť pro data	10240 bytů
Zálohování dat	standardně 100 hodin
Integrované I/O digitální analogové	24 vst./16 výst. -
Rozšiřovací moduly	7 modulů <sup>1</sup>
Vysokorychlostní čítače Jednofázové Dvofázové	6 při 30 kHz 4 při 20 kHz
Pulzní výstupy (DC)	2 při 20 kHz
Analogové potenciometry	2
Hodiny reálného času	Integrované
Komunikační porty	2 RS-485
Matematika s pohyblivou řádovou čárkou	Ano
Velikost registru obrazu digitálních I/O	256 128in/128out
Rychlost booleovských instrukcí	0.22 us/instr.

Tabulka 2.2 : Parametry PLC Simatic S7 226

V naší aplikaci jsme se rozhodli pro tento typ PLC z důvodů širokých zkušeností právě se zmiňovaným systémem a díky pro nás ideální kombinaci počtu digitálních vstupů a výstupů. K CPU, jak se nazývá hlavní jednotka systému, jsme pro ovládání frekvenčních měničů ještě přidali výstupní analogový modul obsahující dva analogové výstupy o napěťovém rozsahu 0-10 V DC. Každý z těchto výstupů je však také možné použít jako proudovou smyčku 4-20 mA.

## Rozšiřující moduly

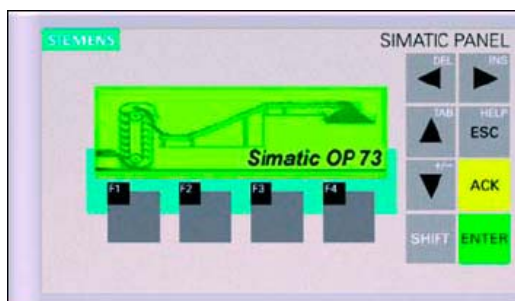
Pro lepší řešení požadavků naší aplikace obsahuje řada S7-200 širokou škálu rozšiřovacích modulů. Těmito rozšiřovacími moduly můžete do S7-200 přidat další funkce.

Rozšiřovací moduly	Typy			
Digitální moduly	vstup	8 x DC vst	8 x AC vst	16 x DC vst
	výstup	4 x DC	4 x relé	
		8 x DC výst	8 x AC výst	8 x relé
	kombinace	4 x DC vst / 4 x DC výst	8 x DC vst / 8 x DC výst	16 x DC vst / 16 x DC výst
		4 x DC vst / 4 x relé	8 x DC vst / 8 x relé	16 x DC vst / 16 x relé
Analogové moduly	vstup	4 x analogový vstup	4 x termočlánek vstup	2 x RTD vstup
	výstup	2 x analogový výstup		
	kombinace	4 x analogový vstup / 1 x analogový výstup		
Inteligentní moduly		Polohovací Ethernetový	Modemový Internetový	PROFIBUS-DP
	Jiné moduly	AS interface		

Tabulka 2.3 : Rozšiřující moduly pro PLC Siemens Simatic S7 200

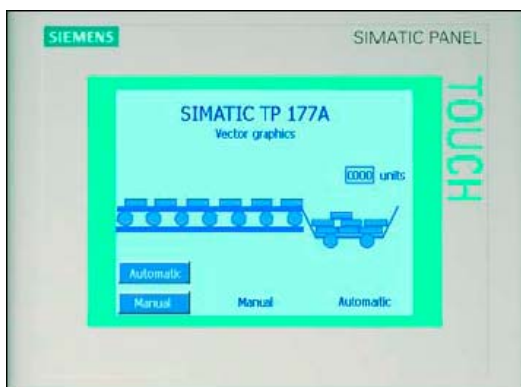
### 2.2.2 Operátorské panely

Námi používané mikropanely jsou zobrazovací panely, které jsou optimalizovány pro komunikaci s PLC Simatic S7-200 firmy Siemens. Tato PLC jsou určena k řízení jednodušších automatizačních aplikací a umožňují řešení s nejnižšími finančními nároky. Na koncepci programovatelných automatů Simatic S7-200 navazují svojí funkčností a především cenou i operátorské panely. Plně grafický operátorský panel OP 73 micro, obrázek [2.10], představuje ve své kategorii jedno z nejvýhodnějších řešení. Je vybaven monochromatickým displejem o velikosti 3" s rozlišením 160 × 48 bodů. Umožňuje jednoduché zobrazení stavu sledovaných objektů pomocí bitových map a sloupcových grafů.



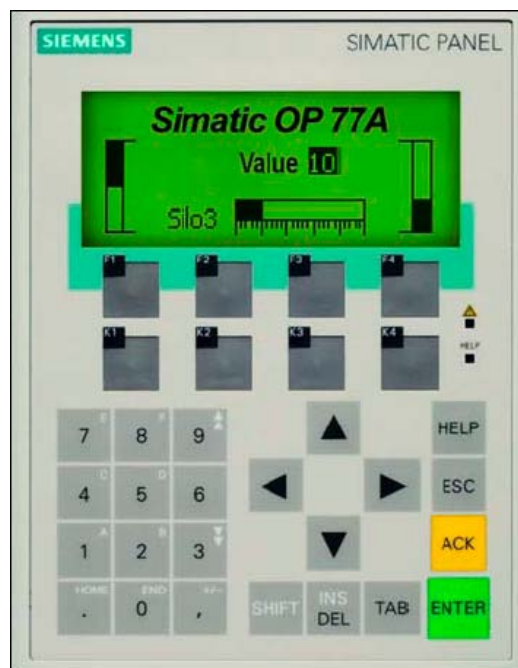
Obrázek 2.10 : Operátorský panel Siemens OP 73 micro

Grafický operátorský panel Simatic TP 177 micro, obrázek [2.11], který jsme použili v naší aplikaci, představuje nejvýkonnější model v řadě mikropanelů. Je vybaven dotykovým displejem (odstíny modré) o velikosti 5,7" a má rozlišení 320 × 240 bodů. Navíc umožňuje používat vektorovou grafiku. Panel je nástupcem produktu Simatic TP 170 micro, a přestože představuje výkonnější variantu, je nabízen za nižší cenu. Panely řady 70



Obrázek 2.11 : Operátorský panel Siemens TP 177 micro

představují úsporné řešení obslužných a sledovacích úloh pro jednodušší automatizační aplikace. Na plně grafickém displeji lze znázorňovat probíhající procesy prostřednictvím bitových map nebo sloupcových grafů. Pro dobrou čitelnost i z větší vzdálenosti nabízí panel různé velikosti písma. Ke znázornění významu funkčních kláves lze použít jak texty, tak i názorné grafické objekty, což snižuje projekční náročnost u vícejazyčných projektů. Aplikace pro tuto řadu panelů lze tvořit pomocí softwaru WinCC flexible Compact. Panel Simatic OP 77B je nejvýkonnější variantou panelů řady 70. Tento nástupce velmi rozšířeného panelu OP 7 je osazen novým plně grafickým displejem o velikosti 4,5" s množstvím moderních funkcí a rozhraní, např. USB, MMC (MultiMedia Card), Profibus-DP, MPI (Multiprotocol Interchange). Díky rozhraní USB je panel připraven na tisk protokolů. Pro snadnou změnu výrobních parametrů lze využít velmi propracovaný recepturní systém. Prostřednictvím karty MMC je možné provést zálohování a zpětné obnovení aktuálního projektu nebo ukládání recepturních dat. Levnější variantou OP 77B je panel Simatic OP 77A, obrázek [2.12], který nabízí za podstatně nižší cenu většinu vlastností předchozí verze.



Obrázek 2.12 : Operátorský panel Siemens OP 77A

Panel je určen pro aplikace, které nepotřebují nasazení receptur, tisku a rychlejší (12 Mbit/s) variantu komunikace Profibus-DP. Dotykový panel Simatic TP 177A s displejem STN o velikosti 5,7" se vyznačuje velmi výhodným poměrem mezi výkonem a cenou. Nahrazuje panel Simatic TP 170A, jenž doplňuje o možnosti využití jednoduché vektorové grafiky (čára, kružnice, obdélník a elipsa). Panel nabízí uživateli praktické funkce pro všechny požadované úlohy. Systém hlášení umožňuje volně definovat jednotlivá hlášení, zvolit pro ně typ zobrazení a nutnost potvrzení. Aplikace pro tento panel lze tvořit pomocí softwaru WinCC flexible Compact.

### 2.2.3 Sensorika

Poslední částí, co se hardwarového vybavení týká, je samozřejmě nepostradatelná sensorická struktura, bez které bychom nebyli schopni

rozpoznat jednotlivé stavy v celém soustrojí. V původním řešení jsme používali celkem tři typů senzorů. Optických reflexních, optických difuzních a indukčních. Optický snímač slouží k indikaci přerušní světelného paprsku. Fotoelektrické senzory se skládají z vysílače impulzního infračerveného světla a přijímače. Indukční snímač slouží k vyhodnocování přítomnosti kovového materiálu. Indukční snímač lze použít jako bezdotykový koncový spínač na strojích, automatických zařízeních apod., čítač kovových předmětů, monitorování pozice části stroje, měření rotační rychlosti axiálního výstředníku apod.

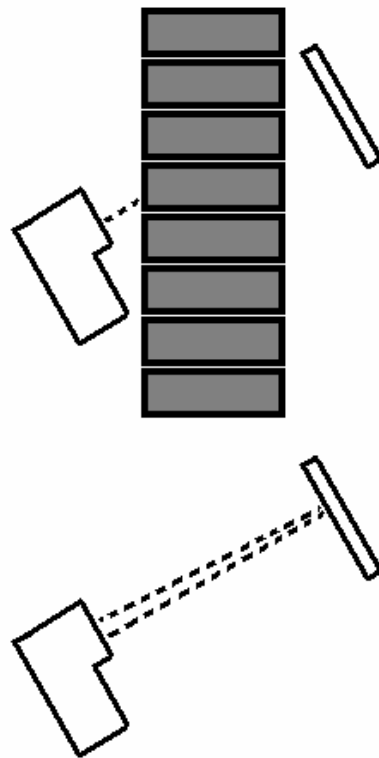
### Vysílané světlo

Optoelektronické snímače obvykle používají následující vysílací komponenty. Červené světlo LED je viditelné světlo, výhodné pro zaměření a nastavení snímače. Infračervené světlo LED (IR) je neviditelné světlo s vyšší energií. Červené světlo Laser je viditelné světlo, které je díky svým fyzikálním vlastnostem ideální pro snímání malých dílů a dlouhé dosahy.

### Optické reflexní senzory

Reflexní optické závory mají vysílač a přijímač ve společném pouzdře. Odrazka umístěná na opačné straně odrazí paprsek zpět do přijímače. Snímaný objekt přeruší odražený světelný paprsek a způsobí změnu stavu výstupního signálu. U lesklých povrchů je doporučováno, aby světlo odražené od objektu bylo filtrováno polarizačním filtrem před přijímačem a tím se předešlo vzniku falešných signálů. Dosah může být až 12 metrů. Tento typ senzorů jsme použili pro detekci zboží na vstupu, pro detekci pozice lopatky u

lopatkového pasu a pro detekci pozice zubu řetězu baličky.

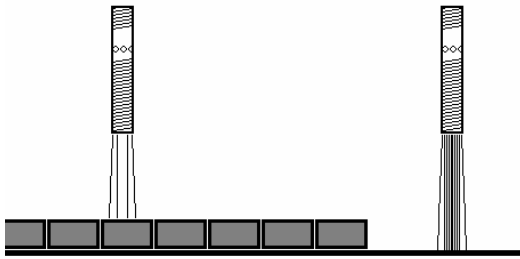


Obrázek 2.13 : Princip snímání optického reflexního senzoru

### Optické difuzní senzory

Difuzní snímače mají přijímač i vysílač ve společném pouzdře. Nasměrování na objekt není kritické. Snímaný objekt (např. standardní objekt s odrazivostí 90%) odrazí část světla svým povrchem zpět do přijímače. Jakmile vstoupí standardní objekt do účinného paprsku, změní se stav výstupu snímače. Snímací dosah závisí na velikosti, tvaru, barvě a povrchových vlastnostech odrážejícího snímaného objektu. Při použití šedé karty Kodak s 90% odrazivostí (podobná bílému papíru) může být snímací vzdálenost až 2 m. Tento typ senzorů jsme použili pro detekci přítomnosti zboží před vstupem lopatkového pasu a pro detekci mezer

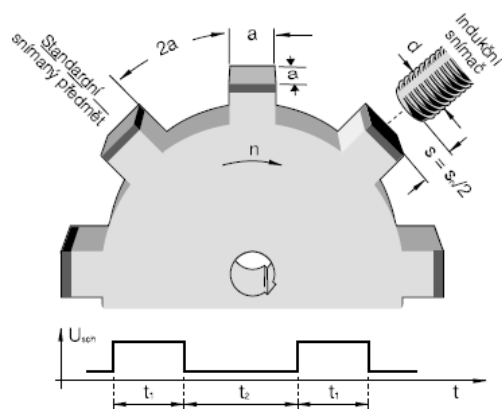
mezi zbožím na přivážecím dopravníčku baličky.



Obrázek 2.14 : Princip snímání optického difuzního senzoru

### Indukční senzory

Princip indukčního snímače je založen na vzájemném působení mezi kovovými vodiči a střídavým elektromagnetickým polem. V kovovém snímaném tlumícím materiálu jsou indukovány vířivé proudy, které odebírají energii z pole a snižují velikost oscilační amplitudy. Tato změna je indukčním snímačem vyhodnocena. V našem případě jsme pomocí tohoto snímače detekovali pozice lopatky u lopatkového pasu a pozici zubu řetězu baličky. Během provozu jsme však z důvodu přesnosti a efektivity přešli kompletně na senzory optické reflexní.



Obrázek 2.15 : Princip snímání indukčního senzoru



# Kapitola 3

## Software

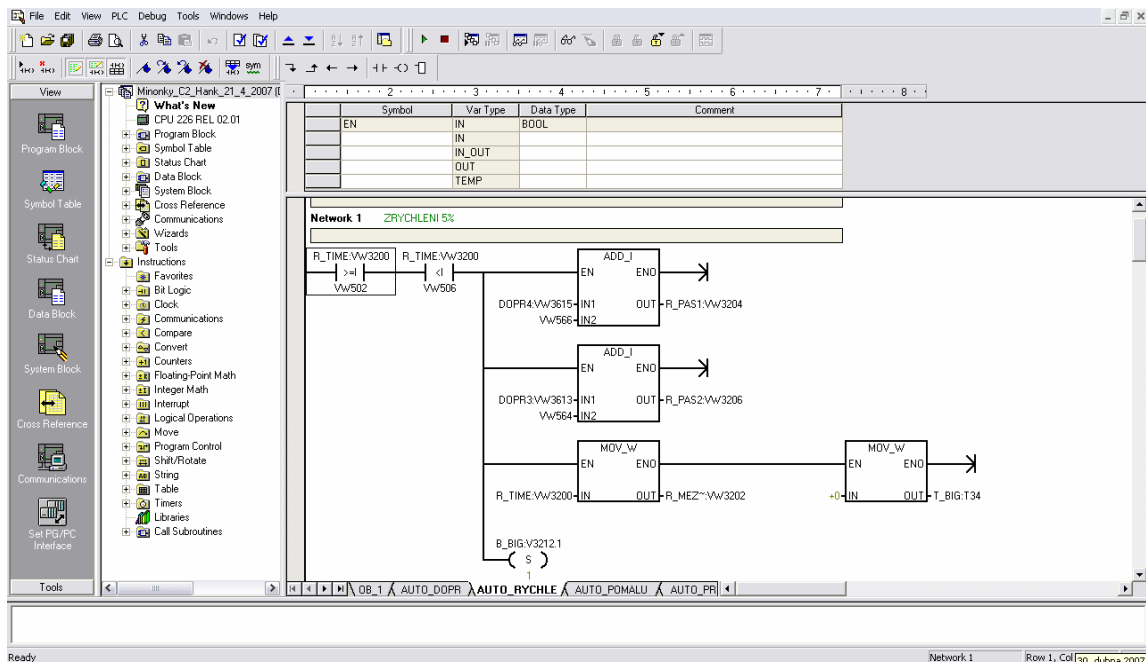
Pro již kompletní hardwarové řešení bylo třeba navrhnout řídicí a ovládací software, bez kterého takto sestavené zařízení není schopné provozu.

### 3.1 PLC

#### 3.1.1 Základní parametry

Programový editor pro PLC Siemens Simatic S7 200, se tvoří pomocí grafického vývojového prostředí MicroWIN STEP7, které je znázorněno na obrázku [3.1].

Jak je již u podobných editorů zvykem, dělí se na jednotlivá okna pro snazší orientaci a efektivitu práce. V horní části obrazovky máme panel pro rychlé vykonávání příkazů, jako jsou například založení nového projektu, kopírování částí programu, kompilace vytvořeného programu, “upload”, což je nahrání aktuálního software z pracujícího PLC do počítače, “download”, což je nahrání vytvořeného software do PLC, nastavení režimu PLC jako například režim RUN a režim STOP, spuštění takzvaného monitoru, který nám umožňuje sledovat a především odlaďovat software a spuštění monitoru paměti, kde se můžeme přesvědčit o skutečných hodnotách uložených v jednotlivých bytech paměti.



Obrázek 3.1 : Hlavní obrazovka editoru programu pro PLC Simatic S7 200 MicroWIN STEP7

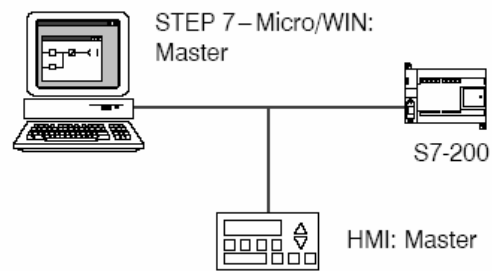
První sloupec na levé straně obrazovky reprezentuje nabídku jednotlivých částí komplexního nastavení PLC. Nejpodstatnějšími jsou samotný program editor, editor symbolických názvů, data blok, kde můžeme nastavovat hodnoty definovaných bytů, system blok, který nám umožňuje přizpůsobovat systémové nastavení PLC pro naši aplikaci a nastavení komunikace. Druhý sloupec z levé strany obrazovky nám pak nabízí již kompletní seznam nastavení PLC, instrukční sadu, nástroje pro řízení servopohonů, PID regulátor a jiné průvodce nastavením, které jsou pro každý model specifické.

## Komunikace

Jedním z klíčových prvků dnešních PLC je právě komunikace. Často se pro řízení výrobních linek používá několik programovatelných automatů, které spolu komunikují pomocí takzvané průmyslové sběrnice. Ještě dnes máme možnost se setkat v poměrně velké míře s protokolem RS232, který v mnoha případech postačuje k jednodušší, a především co se objemu dat týče, časově nenáročné komunikaci. Pro stabilnější komunikaci se potom používají protokoly typu RS485, což jsou Profibus, nebo MPI.

### Protokol PPI

PPI je protokol typu master-slave: zařízení master posílají požadavky zařízením slave, které jim odpovídají. Viz obrázek [3.2]. Zařízení slave negenerují zprávy, ale čekají, dokud master nepošle požadavek nebo je nevyzve k odpovědi. Zařízení master komunikují se zařízením slave prostřednictvím sdíleného spojení, které je řízeno protokolem PPI. Tento protokol neomezuje počet masterů, které mohou komunikovat s kterýmkoliv zařízením slave. Není



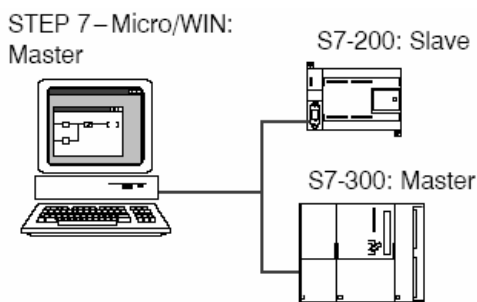
Obrázek 3.2 : Schéma zapojení pomocí komunikačního rozhraní PPI

však možné instalovat více než 32 masterů. CPU S7-200 se mohou v režimu RUN chovat jako master, pokud v uživatelském programu povolíte režim PPI master. Po povolení režimu PPI master můžete použít instrukce "čtení ze sítě" pro čtení nebo "čtení do sítě" pro zápis do jiných jednotek S7-200. I když pracuje automat jako PPI master, stále reaguje jako zařízení typu slave na požadavky jiných masterů. PPI Advanced umožňuje navázat vzájemné logické spojení mezi zařízeními. V případě PPI Advanced má každé zařízení omezený počet připojení. Všechny CPU S7-200 podporují protokol PPI i PPI Advanced, ale PPI Advanced je jediný protokol PPI podporovaný modulem EM 277.

### Protokol MPI

Protokol MPI umožňuje komunikaci master – master i master – slave. Viz obrázek [3.3]. Pro komunikaci s CPU S7-200 naváže STEP7 Micro/WIN spojení master – slave. Protokol MPI nepodporuje CPU S7-200 ve funkci master. Síťová zařízení spolu komunikují prostřednictvím oddělených spojení (řízených protokolem MPI) mezi libovolnými dvěma zařízeními. Komunikace mezi zařízeními je omezena na počet spojení podporovaných CPU S7-200 nebo moduly EM 277.

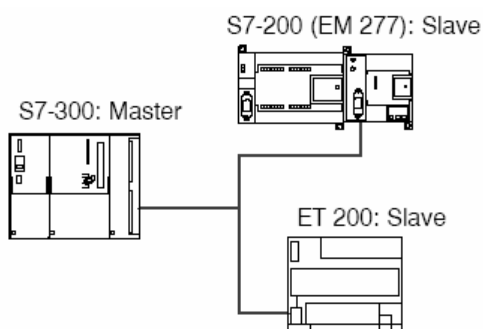




Obrázek 3.3 : Schéma zapojení pomocí komunikačního rozhraní MPI

### Protokol PROFIBUS

Protokol PROFIBUS je určen pro vysokorychlostní komunikaci s distribuovanými vstupními/výstupními zařízeními (vzdálené vstupy/výstupy). Mnoho různých výrobců nabízí řadu zařízení PROFIBUS. Tato zařízení se pohybují od jednoduchých vstupních nebo výstupních modulů až po regulátory motorů a programovatelné automaty. Síť PROFIBUS běžně obsahují jedno zařízení typu master a několik vstupních/výstupních zařízení typu slave. Viz obrázek [3.4]. Zařízení typu master je nakonfigurováno tak, že zná, jaké typy vstupních/výstupních zařízení typu slave jsou k němu připojeny a jaké mají adresy. Master inicializuje síť a ověřuje, zda zařízení typu slave, zapojená do sítě, odpovídají konfiguraci. Master nepřetržitě zapisuje výstupní data do zařízení typu slave a čte z nich data vstupní.



Obrázek 3.4 : Schéma zapojení pomocí komunikačního rozhraní Profibus

### Přístup k datům

#### Registr obrazu vstupů: I

Automat S7-200 na začátku každého programového cyklu načte aktuální hodnoty vstupních bodů a zapíše je do registru obrazu vstupů. Přístupovat k registru obrazu vstupů můžete v bitech, bytech, word nebo double word:

Bit:  $I[\text{adresa bytu}].[adresa bitu]$

I0.1

Byte, word, double word:

$I[\text{velikost}][adresa počátečního bytu]$

IB4, IW4, ID4

#### Registr obrazu výstupů: Q

Na konci každého programového cyklu zapisuje S7-200 hodnoty uložené v registru obrazu výstupů na výstupy. K registru obrazu výstupů můžete přistupovat v bitech, bytech, word nebo double word:

Bit:  $Q[\text{adresa bytu}].[adresa bitu]$

Q1.1

Byte, word, double word:

$Q[\text{velikost}][adresa počátečního bytu]$

QB5, QW5, QD5

#### Oblast proměnné paměti: V

Paměť V můžete používat pro uložení mezivýsledků operací prováděných řídicí logikou programu. Můžete ji také používat pro ukládání ostatních dat souvisejících s procesem nebo úlohou. Do paměťové oblasti V můžete přistupovat v bitech, bytech, word nebo double word:

Bit:  $V[\text{adresa bytu}].[adresa bitu]$

V10.2

Byte, word, double word:

$V[\text{velikost}][adresa počátečního bytu]$

VB100, VW100, VD100

#### Oblast bitové paměti: M

Oblast bitové paměti (paměť M) se využívá na řídicí kontakty pro uložení mezistavu procesu nebo mezistavu jiných řídicích informací. Do bitové paměťové oblasti můžete vstupovat v

bitech, bytech, word nebo double word:  
Bit: M[adresa bytu].[adresa bitu]  
M26.7

Byte, word, double word:  
M[velikost][adresa počátečního bytu]  
MB20, MW20, MD20

### **Speciální paměť: SM**

SM bity slouží pro výměnu informací mezi CPU a programem. Tyto bity se používají pro nastavení a řízení některých speciálních funkcí CPU S7-200, jako jsou: bit prvního programového cyklu; bit, který se mění s pevnou frekvencí; nebo bit stavu matematických nebo provozních instrukcí. K datům v této paměti můžete přistupovat po bitech, bytech, word nebo double word:

Bit: SM[adresa bytu].[adresa bitu]  
SM0.1

Byte, word, double word:  
SM[velikost][adresa počátečního bytu]  
SMB86, SMW100, SMD 68

### **Analogové vstupy: AI**

Automat S7-200 převádí analogové hodnoty (jako je teplota nebo napětí) na digitální hodnoty o délce word (16 bitů). Tyto hodnoty se dají získat pomocí identifikátoru oblasti (AI), velikosti dat (W) a adresy počátečního bytu. Protože analogové vstupy mají velikost word a vždy začínají na adrese bytu se sudým číslem (jako je 0, 2 nebo 4), přistupuje se k nim vždy se sudou adresou bytu (např. AIW0, AIW2 nebo AIW4). Analogové vstupní hodnoty se dají pouze číst.

Formát: AIW[adresa počátečního bytu]  
AIW4

### **Analogové výstupy: AQ**

Automat S7-200 převádí 16bitovou digitální hodnotu velikosti word na proud nebo na napětí úměrné digitální hodnotě. Tyto hodnoty se dají zapsat pomocí identifikátoru oblasti (AQ), velikosti dat (W) a adresy počátečního bytu. Protože analogové výstupy mají

velikost word a vždy začínají na adrese bytu se sudým číslem (jako je 0, 2 nebo 4), zapisují se vždy se sudou adresou bytu (například AIW0, AIW2 nebo AIW4). Hodnoty analogových výstupů se dají pouze zapisovat.

Formát:

AQW[adresa počátečního bytu]

AQW4

## 3.1.2 Způsoby programování

Jednou ze základních vlastností programovatelných automatů je jejich snadné programování. Technik z příslušného oboru (strojírenství, potravinářství, tepelná technika atd.) by měl zvládnout základy programování PLC daného typu během několika týdnů. Tato vlastnost společně s vysokou spolehlivostí a nízkou cenou (ve srovnání s řídicími počítači klasického typu) předurčila obrovské rozšíření průmyslových automatů prakticky do všech průmyslových oblastí. V oblasti PLC není bohužel ujednocen univerzální programovací jazyk jako je tomu u personálních počítačů, ale každý výrobce má svůj způsob programování. Tyto způsoby programování jsou si však natolik podobné, že přechod na další typ automatu je velmi snadný. Programovatelné automaty se programují v podstatě následujícími způsoby:

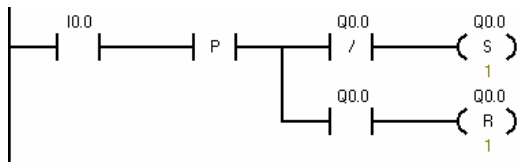
- jazyk vycházející z logických instrukcí STL

- jazyk vycházející ze symbolů liniového schématu - tzv. Ladder diagram LAD

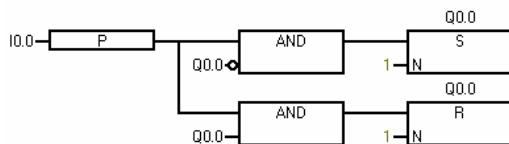
- jazyk vycházející ze symbolů blokového schématu (hradla AND, OR, klopné obvody R-S atd.) FBD

- jazyk sekvenčních blokových schémat GRAPHCET, SFC.

Pro demonstraci si předvedeme jednoduchý program pro spuštění motoru výstupem Q0.0, a to start i stop pomocí tlačítka na vstupu I0.0. Jako první reprezentaci jsem zvolil programování pomocí liniového schéma, tedy LAD. Jak si můžeme všimnout na obrázku [3.5], detekujeme vzestupnou hranu na vstupu I0.0. Pokud není výstup Q0.0 log. 1, pomocí instrukce SET ho nastavíme do hodnoty log. 1. Pokud výstup Q0.0 nastaven je, pomocí instrukce RESET ho nastavíme do log. 0.



Obrázek 3.5 : Reprezentace programu pomocí jazyka LAD



Obrázek 3.6 : Reprezentace programu pomocí jazyka FBD

```

Network 1
-----
LD      I0.0
EU
LPS
AN      Q0.0
S        Q0.0, 1
LPP
A        Q0.0
R        Q0.0, 1

```

Obrázek 3.7 : Reprezentace programu pomocí jazyka STL

Je zde také třeba uvést skutečnost, že program vytvořený pomocí jazyků FBD, nebo LAD, je kdykoli možno převést do textového instrukčního tvaru, tedy do jazyka STL. Pokud ovšem tvoříme software přímo pomocí jazyka STL, není pravidlem jeho převoditelnost do jazyků LAD či FBD.

### 3.1.3 Program

Pro tvorbu programu jsem zvolil jazyk typu LAD. Tento princip je mi blízký pro jeho názornost a velice dobré, transparentní vlastnosti z hlediska následného odladění programu.

#### 3.1.3.1 Některé použité rutiny

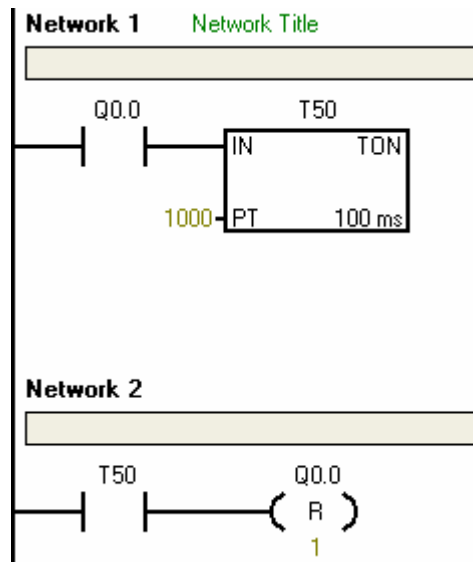
##### Časovač

V automatu S7-200 máme k dispozici celkem 256 časovačů (T0 – T255), které mají definovaný časový interval inkrementace (1ms, 10ms a 100ms). Pro jejich osluhu máme k dispozici tři typy instrukcí a to:

- TON – časovač zpožděného sepnutí
- TONR – časovač zpožděného sepnutí s pamětí
- TOF – časovač zpožděného vypnutí

Pro příklad použití si uvedeme jednoduchou aplikaci navázanou na příklad programu znázorněný na obrázku [3.5]. Rutiny zobrazené na obrázku [3.8], nám budou reprezentovat funkci vypnutí motoru po uplynutí časového intervalu. Pokud tedy zapneme stiskem tlačítka připojeným na vstup I0.0 motor připojený na výstup Q0.0, spustí se nám zároveň časovač T50, který je ovládaný taktéž registrem obrazu výstupu Q0.0. Za předpokladu, že do stanovené doby motor opětovným stiskem tlačítka na I0.0 nevypneme, po uplynutí, v našem případě 10s, se nastaví bit časovače do log. 1 a tím se

povolí provedení instrukce RESET Q0.0. Jelikož nám přímo Q0.0 také ovládá časovač i jeho hodnota se tedy vynuluje a bit časovače se opět nastaví do log. 0.



Obrázek 3.8 : Příklad použití instrukce časovače

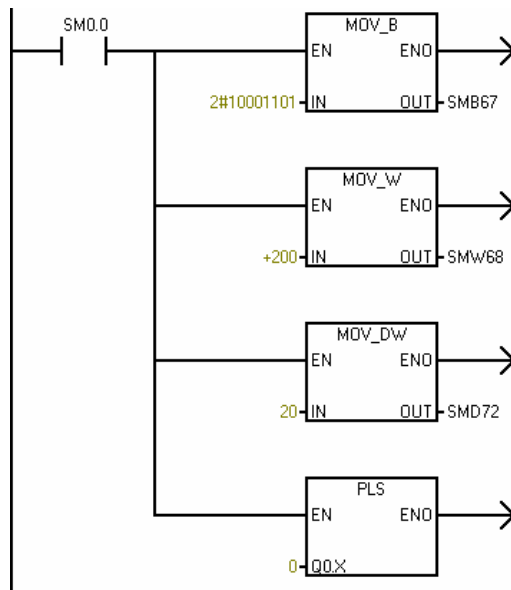
## Pulsní výstup

V automatu S7-200 máme k dispozici dva pulsní výstupy. Každý z nich lze využít buď jako generátor pulsů, nebo jako generátor PWM. První je na výstupu Q0.0, druhý je na výstupu Q0.1. V našem případě používáme oba jako pulsní výstupy pro řízení pohybu servomotorů. Pro názornost si uvedeme rutinu nastavení prvního výstupu. Jako první je třeba nastavit řídicí registry generátoru, které jsou pro Q0.0 na adresách :

SMB67 - význam jednotlivých bitů je následující :

- 0. Aktualizace hodnoty doby cyklu
- 1. Aktualizace šířky pulsu
- 2. Aktualizace hodnoty počtu pulsů
- 3. Časová základna 0 = 1us na puls  
1 = 1ms na puls
- 4. Asynchronní aktualizace
- 5. Práce s jedním segmentem
- 6. Režim PTO
- 7. Povolovací bit

SMW68 - Hodnota doby cyklu jednoho pulsu x časová základna  
SMD72 - Hodnota počtu pulsů



Obrázek 3.9 : Příklad nastavení pulsního výstupu

Jako poslední použijeme instrukci PLS, která provede námi nakonfigurované nastavení ve speciálních registrech SM.

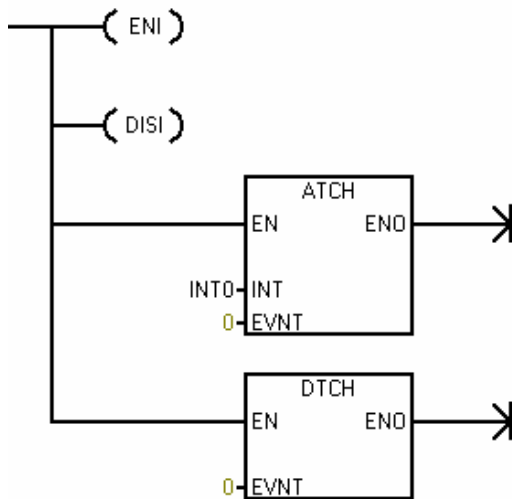
## Přerušení

Další velice důležitou funkcí je používání tzv. přerušení (interrupt). Druhů přerušení je velké množství a tak zmíníme jen ty, které jsou využity v naší aplikaci.

- 1) Přerušení od náběžné hrany vstupu I0.0
- 2) Časové přerušení 1
- 3) Časové přerušení 2
- 4) Přerušení od dokončení operace pulsního výstupu 0
- 5) Přerušení od dokončení operace pulsního výstupu 1

V principu je funkce přerušení následující. V okamžiku, kdy daná událost vyvolává přerušení nastane, přestane se vykovávat uživatelský

program, přeruší se a zavolá se podprogram obsluhy daného přerušení, který se vykoná okamžitě. Po jeho dokončení se opět začne vykonávat uživatelský program od místa, kde byl přerušen. Nejčastěji se tento způsob používá v případech potřeby rychlé reakce na danou událost. Nastavovací instrukce pro přerušení jsou uvedeny na obrázku [3.10]



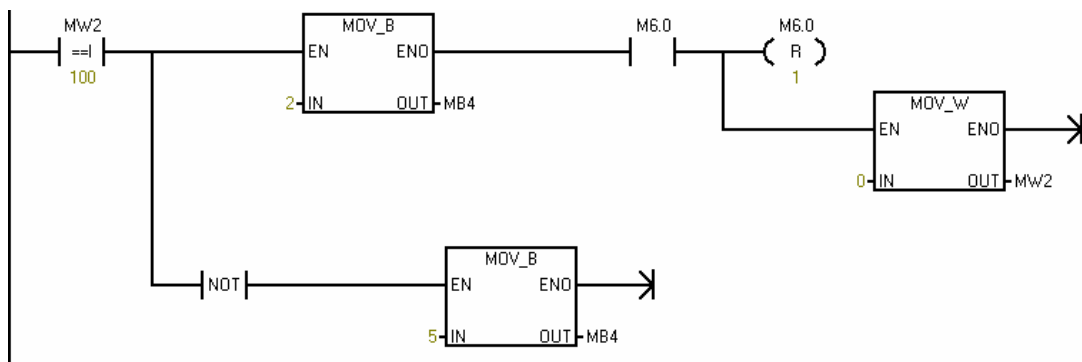
Obrázek 3.10 : Nastavovací instrukce pro obsluhu přerušení

Jako první použijeme instrukci ATCH, která nám definuje, pro jaký typ přerušení se má použít jaký podprogram obsluhy přerušení. Tato instrukce slouží jako lokální povolení

daného typu přerušení. Pro zapnutí přerušovacího systému je však ještě nutné použít globální povolení přerušení a to instrukcí ENI. Stejným způsobem pak můžeme pomocí instrukce DTCH určitý druh přerušení zakázat, nebo pomocí instrukce DISI zakázat globálně vykonávání všech typů přerušení.

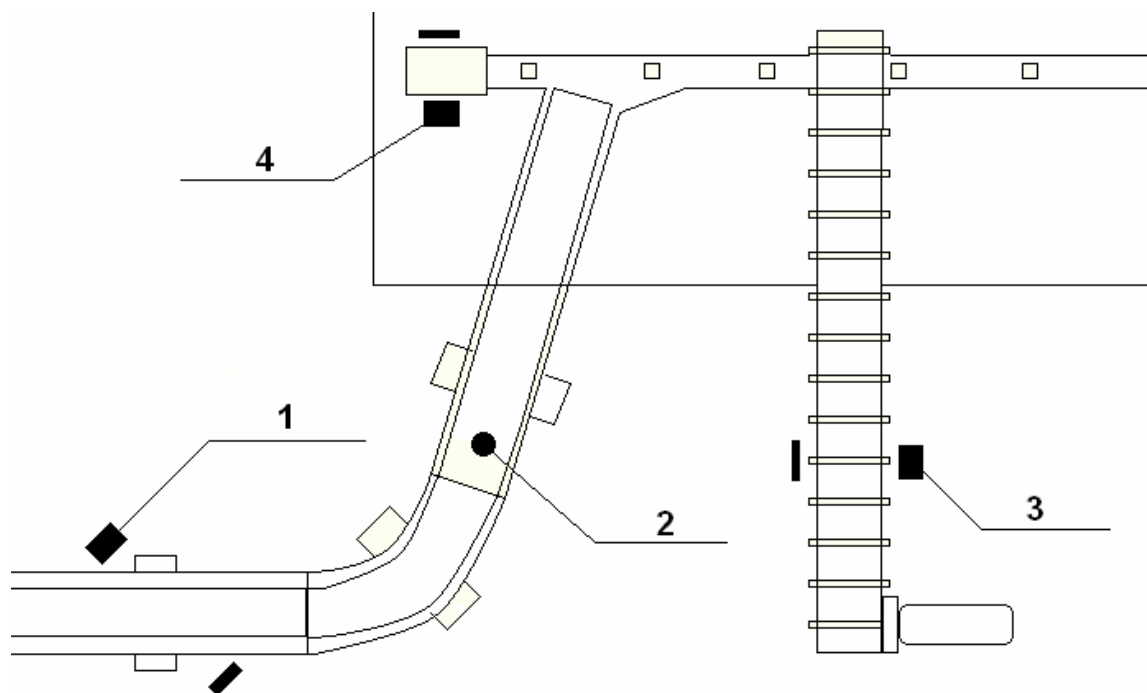
## Heslo

Pro zabezpečení některých nastavení je třeba použít ochranu vstupu do obrazovky pomocí hesla. Tento způsob jsem vyřešil pomocí rutiny zobrazené na obrázku [3.11]. Na operátorském panelu je kolonka pro zadání hesla (do paměti MW2) a potvrzovací tlačítko, které nám vykonává funkci – volání obrazovky podle čísla uloženého v bytu MB4. Průběh je následující. Pokud se zadané heslo neshoduje s přednastaveným, volá se po stisku tlačítka znovu obrazovka zadávání hesla. Pokud se ovšem heslo shoduje, zavolá se obrazovka krytá heslem a při jejím načtení se nastaví bit M6.0. V prvním scanu, kdy je tento bit nastaven, sám sebe resetuje a přesouvá na místo zadaného hesla hodnotu 0, čím zajistí při dalším zavolání této rutiny nové korektní zadávání hesla.



Obrázek 3.11 : Rutina pro obsluhu HESLA

### 3.1.3.2 Program Čokotatranky



Obrázek 3.12 : Situace při variantě Čokotatranky

Pro pochopení jednotlivých posloupaností programu je na obrázku [3.12] znázorněno technologické řešení této varianty. Nejdříve si projdeme samotnou koncepci řešení a poté jednotlivé dílčí části programu.

Po zapnutí elektrického napájení je celý stroj v klidovém stavu. Na ani jeden komponent není tedy přiveden signál ENABLE. Operátorský panel nám bezprostředně signalizuje poruchový stav, pokud se stroj v nějakém nachází (Central stop, špatné pozice servopohonů, vypadlé napájení nějaké z komponent, atd.). Pokud je stroj bez detekovaných poruch, operátorský panel hlásí, že zařízení je připraveno ke spuštění a čeká na stisk startovacího tlačítka, které je umístěno u stanoviště operátora. Zde již nastává první dělení programu. Pokud máme zvolen režim manuálního řízení, po stisku tlačítka START se okamžitě rozjedou všechny

přivázeční dopravníčky a zároveň je spuštěna i balička. Systém v tomto režimu nereaguje na stavy jednotlivých senzorů, až na senzor pozice lopatky (3), který nám signalizuje, že lopatka není v kolizním stavu s řetězem baličky a pokud je, stroj hlásí poruchu. Vše je tedy závislé čistě na operátorovi. Pokud máme ovšem zvolen režim automatika, řízení přivázečních dopravníček je zcela závislé na stavech senzorů. V první řadě musí být samozřejmě opět lopatka u lopatkového dopravníku ve správné poloze. Po stisku tlačítka START je celý stroj v klidu a čeká až operátor přisune zboží na senzor (1) prvního přivázečního dopravníčku. V tuto chvíli se rozjede jak první přivázeční dopravníček, tak kruhový dopravníček. V momentě, kdy zboží najede na senzor (2) přivázečního dopravníčku baličky, je spuštěna také balička a dokud senzor (1) detekuje přítomnost zboží, je celé soustrojí v chodu.

Pro řízení rychlosti prvního přivážecího dopravníčku a kruhového dopravníčku jsou použity dva analogové výstupy s rozsahem 0-10 V DC, adresované AQW0 a AQW2. Pro jejich nastavení musíme ovšem znát rychlost baličky. Pro její zjištění jsme použili přerušení od náběžné hrany senzoru (4) palce řetězu baličky. Při spuštění chodu baličky je zároveň spuštěn časovač T64 (1ms), který měří periodu mezi jednotlivými palci. Při vstupu do přerušení se nejprve uloží jeho hodnota do registru VW3008 a časovač se vyresetuje. Podle vzorce (3.1) se pak vypočítá rychlost baličky v baleníh za minutu.

$$v_{bal} = \frac{60000}{T} \quad (3.1)$$

Z této hodnoty jsou následně pomocí koeficientů zadaných z operátorského panelu vypočítány rychlosti jednotlivých dopravníků. Rychlost přivážecího dopravníčku baličky nejsme schopni ovlivnit, protože si jej řídí přímo balička tak, aby jeho rychlost odpovídala dodávání zboží do řetězu.

## Regulátor

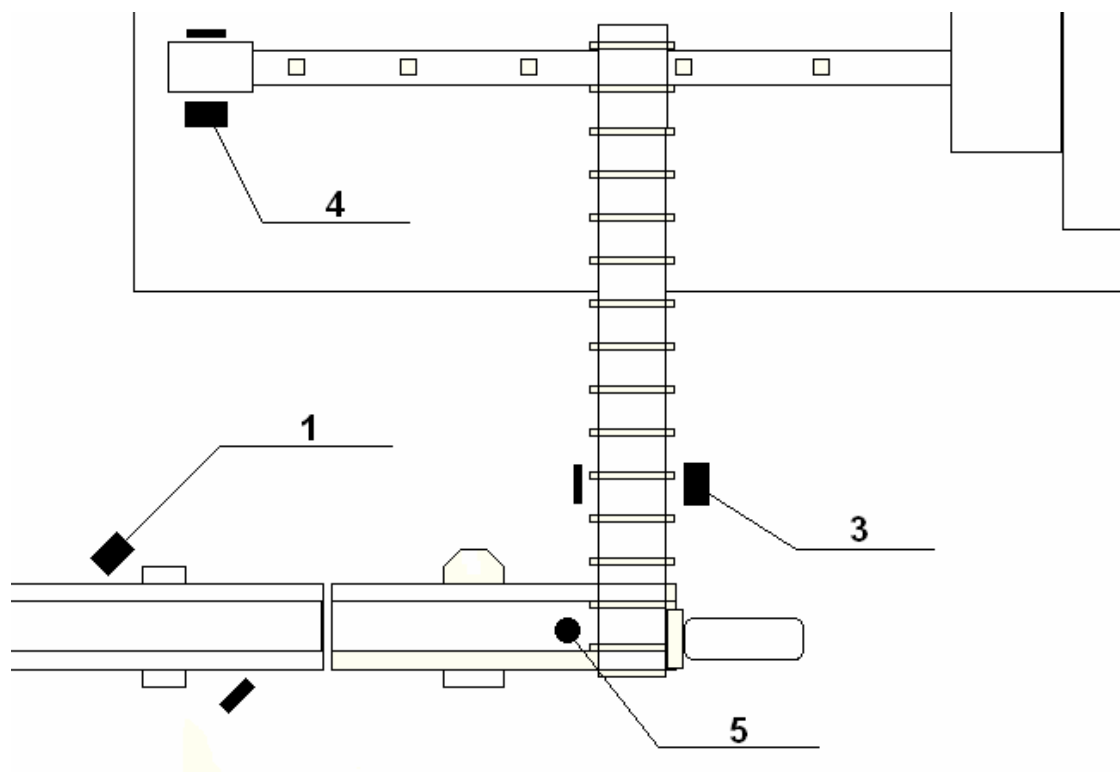
V původním návrhu byl použit algoritmus jednotné rychlosti dopravníků. V závislosti na rychlosti baličky a koeficientech se nastavila pevně rychlost dopravníků. Toto řešení bylo však do značné míry efektivní v závislosti na kvalitě obsluhy, která přesunovala zboží z hlavního pasu na první přivážecí dopravníček. Pokud obsluha nechávala mezi zbožím mezery, v dané koncepci nebyla možnost jak tyto mezery “dohnat” a docházelo ke kolizím na vstupu zboží do řetězu baličky. V opačném případě, pokud obsluha vkládala zboží bez mezer, to se začalo na kruhovém dopravníčku “hrnout” a ke kolizím docházelo opět.

Jako řešení tohoto problému jsme navrhli použití regulátoru rychlosti dopravníků, který by toto eliminoval. Vstupní informace pro regulátor je dodávána senzorem (2) umístěným nad začátkem přivážecího dopravníčku baličky. Tento senzor měří délku mezer mezi zbožím, respektive čas trvání mezery mezi zbožím. Funkce samotná spočívá ve vyhodnocování délky mezer a akčního zásahu zpomalení nebo zrychlení prvního přivážecího dopravníčku a kruhového dopravníčku. Jako reference slouží ovšem původní hodnota rychlosti dopravníků odvozená od rychlosti baličky. Nastavení regulátoru nám umožňují tři obrazovky z operátorského panelu. Na první obrazovce nastavujeme základní údaje a to :

- 1) Ideální délku mezery mezi zbožím
- 2) Za jak dlouho, pokud není detekována mezera, se má přejít do zpomalování
- 3) Dobu trvání zrychlení dopravníků po detekci velké mezery

Na druhé obrazovce nastavujeme režim zrychlování dopravníků. Hodnoty je třeba nastavovat jako vzestupnou kaskádu, jinak by funkce regulátoru nebyla optimální. V principu zde volíme o kolik procent mají dopravníčky zrychlit po mezeře větší než zvolený časový údaj. Na třetí obrazovce poté nastavujeme zpomalování dopravníků. V principu opět, pokud jede zboží v ideálních mezerách nastavenou dobu, o kolik procent se dopravníčky zpomalí. Při provozu si tak vlastně regulátor sám vytváří mezery, které jakoby opět dohání. Tím je zajištěn jak kontinuální tok bez mezer, tak zároveň tok odlehčený, který odpovídá požadavkům na efektivní přechod zboží do řetězu baličky. Při správném nastavení řídicích konstant regulátoru jsme se již nesetkali s problémy uvedenými na začátku tohoto oddílu a stroj tak pracuje spolehlivě.

### 3.1.3.2 Program Miňonky



Obrázek 3.13 : Situace při variantě Miňonky

Pro pochopení jednotlivých poslopností programu je opět na obrázku [3.13] znázorněno technologické řešení této varianty. Po zapnutí stroje jsou opět všechny jeho komponenty v klidovém stavu bez signálu ENABLE.

První a bezpodmínečně nutnou podmínkou správného chodu je nastavení pozice senzoru zubu řetězu baličky (4). Ten je třeba nastavit tak, aby při náběžné hraně detekující přítomnost zubu, nebyl žádný ze zubů v trajektorii pohybu lopatkového pasu, respektive aby poslední zub právě trajektorii lopatkového dopravníku opustil, jak je znázorněno i na obrázku [3.13]. Opět je třeba zkontrolovat nastavení senzoru pozice lopatky (3), který nám zaručuje správný přechod

zboží z lopatkového pasu do zubů řetězu baličky. Pokud není detekován poruchový stav, stroj opět čeká na stisk tlačítka START od operátora. Znovu se zde nabízí možnost ručního a automatického režimu, nicméně v tomto případě není ruční režim doporučen. Pokud se přesto rozhodneme pro jeho použití, po stisku tlačítka START se okamžitě rozjede první přivázeční dopravníček a současně s ním se zapne také balička. V okamžiku, kdy zub řetězu baličky najede na senzor (4) a tím detekuje bezkolizní pozici zubu a lopatkového dopravníku, se provede krok lopatkového pasu. Po jeho dokončení a zaručení tak správné pozice pro zasunutí zboží mezi lopatky pomocí servodopravníčku, udělá krok servodopravníček. V momentě, kdy je dokončen i jeho krok, čeká opět systém



na náběžnou hranu senzoru (4). Pokud je zvolen automatický režim, je po stisku tlačítka START opět okamžitě spuštěn první přivázeční dopravníček, balička ovšem spuštěna není. V momentě, kdy zboží, které operátor nasune z hlavního pasu na přivázeční dopravníček, dojde na senzor přítomnosti zboží (1), servodopravníček se rozjede plynulým pohybem rychlostí odpovídající prvnímu přivázečnímu dopravníčku. Při detekci přítomnosti zboží senzorem před vstupem lopatkového pasu (5) je zachován plynulý pohyb servodopravníčku, ovšem zapne se časovač nastavený tak, aby při splnění jeho časové podmínky bylo zboží na dorazu lopatkového pasu. V tomto momentě se servodopravníček zastaví, zapne se balička a čeká se na přerušení od náběžné hrany senzoru zubu řetězu baličky (4). V obsluze přerušení se opět spustí krok lopatkového pasu, po jehož dokončení následuje spuštění kroku servodopravníčku. Tlak na zboží zaručující jeho bezkolizní přesun nám zaručuje stále jedoucí první přivázeční dopravníček. Tato posloupnost je opět dodržována až do doby, kdy senzor přítomnosti zboží (1) detekuje mezeru delší, než je v jeho časovém filtru, nebo senzor přítomnosti zboží před vstupem lopatkového pasu (5) detekuje mezeru ve zboží delší, než odpovídá jeho časovému filtru. V těchto případech se balička zastaví a systém se chová způsobem, jak je výše uvedeno.

## 3.2 TP

PLC může bez problémů fungovat i bez připojení operátorského panelu. Operátorský panel však jako takový, je bez připojení k PLC nevyužitelný. V dnešních aplikacích, kde se kladou nároky především na flexibilitu a z toho plynoucí možnost upravovat nastavení parametrů řídicího programu, je tak

ovšem podstatnou částí právě operátorský panel, který nám toto umožňuje.

### 3.2.1 Základní parametry

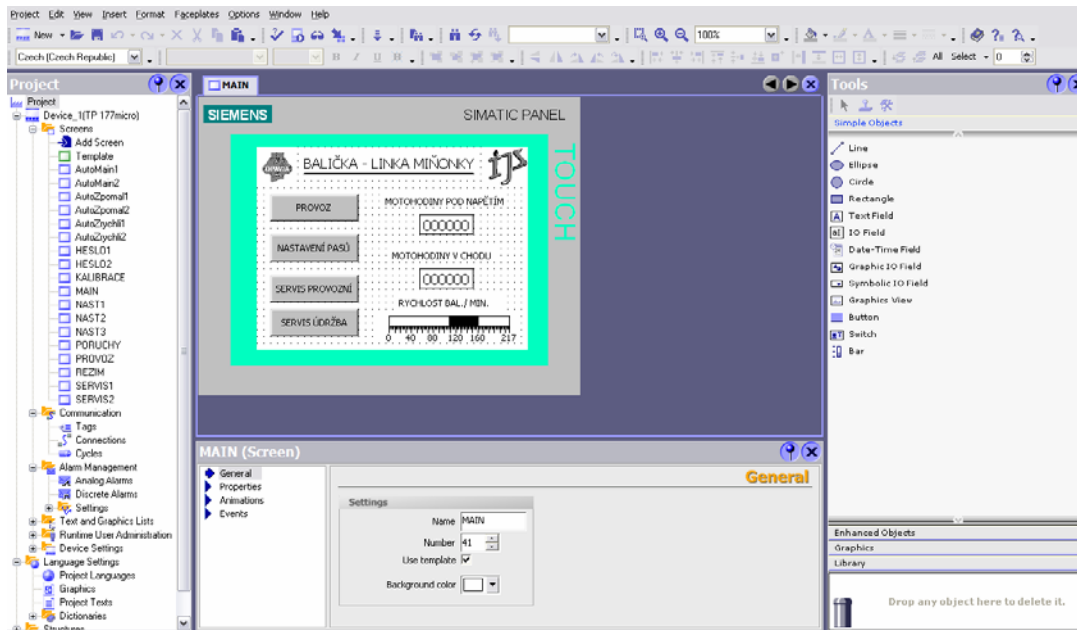
Po založení nového projektu je třeba nastavit pro jaký typ operátorského panelu budeme program vytvářet. V našem případě jsme použili operátorský panel TP 177 micro, pro který je určena verze MicroWinCC flexible. Jako první věc je dobré nastavit parametry komunikace s PLC. Ty jsou k dispozici v záložce Communication na levé straně editačního okna, které je znázorněno na obrázku [3.14].

### 3.2.2 Způsob programování

Programování operátorských panelů značky Siemens se provádí pomocí vývojového software WinCC flexible. Jedná se o interaktivní grafické prostředí umožňující vkládání jednotlivých komponent přímo na obrazovku TP. Jedná se o komponenty typu:

- 1) Přímka
- 2) Elipsa
- 3) Kružnice
- 4) Obdélník
- 5) Textové pole
- 6) Vstupní/Výstupní pole
- 7) Pole data a času
- 8) Grafické Vstupní/Výstupní pole
- 9) Symbolické Vstupní/Výstupní pole
- 10) Zobrazení grafiky
- 11) Tlačítko
- 12) Přepínač
- 13) Stavový diagram

Každá komponenta má své specifické vlastnosti a jsou na ní definované události, kterým je možné přiřadit příkazy. Pro příklad si uvedeme tlačítko.



Obrázek 3.14 : Hlavní obrazovka editoru WinCC flexible

Jako události jsou na něm definované :

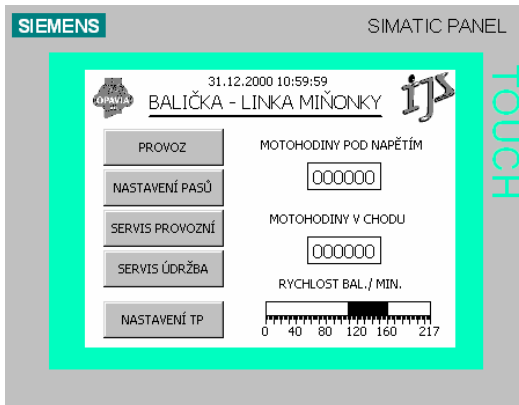
- 1) Klik
- 2) Stisk
- 3) Uvolnění
- 4) Aktivování
- 5) Deaktivování
- 6) Změna stavu

K libovolné z těchto událostí můžeme tedy přiřadit libovolný příkaz nebo posloupnost příkazů. Kompletní seznam příkazů je uveden v tabulce [3.1]. Pro zadávání číselných údajů potom využijeme vstupní/výstupní pole. Jako vlastnosti tohoto komponentu pak zvolíme formát vstupních dat a definujeme takzvaný TAG, což je paměťové místo v paměti PLC, do kterého se námi zadávané číslo edituje. Tagy jsou přesně definované datové typy využitelné v PLC jako například bool, byte, word, doubleword, integer, string, nebo real. U některých objektů potom lze definovat jejich viditelnost, což můžeme využít například u chybových hlášení.

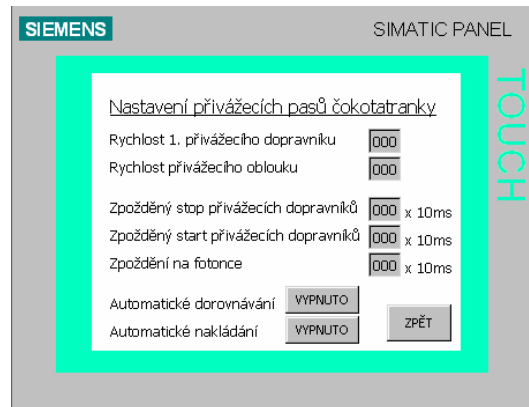
- ... ActivateCleanScreen
- ... ActivatePreviousScreen
- ... ActivateScreen
- ... ActivateScreenByNumber
- ... AdjustContrast
- ... AlarmViewAcknowledgeAlarm
- ... AlarmViewEditAlarm
- ... AlarmViewShowOperatorNotes
- ... CalibrateTouchScreen
- ... ChangeConnection
- ... ClearAlarmBuffer
- ... ClearAlarmBufferProtocolLegacy
- ... DecreaseValue
- ... GetGroupNumber
- ... GetPassword
- ... GetUserName
- ... IncreaseValue
- ... InverseLinearScaling
- ... InvertBit
- ... LinearScaling
- ... Logoff
- ... Logon
- ... ResetBit
- ... ScreenObjectCursorDown
- ... ScreenObjectCursorUp
- ... ScreenObjectPageDown
- ... ScreenObjectPageUp
- ... SetBit
- ... SetConnectionMode
- ... SetDeviceMode
- ... SetLanguage
- ... SetValue
- ... ShowAlarmWindow
- ... ShowLogonDialog
- ... ShowOperatorNotes
- ... StopRuntime
- ... UpdateTag

Tabulka 3.1 : Příkazy WinCC flexible

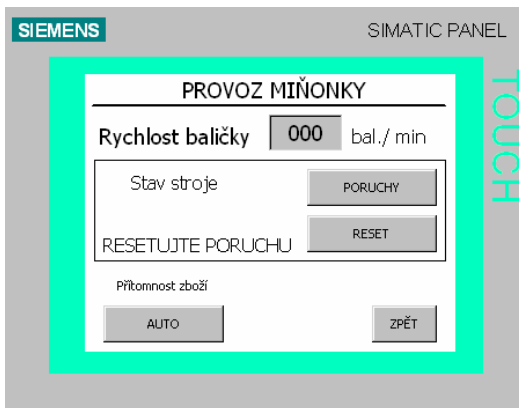
### 3.2.2 Obrazovky



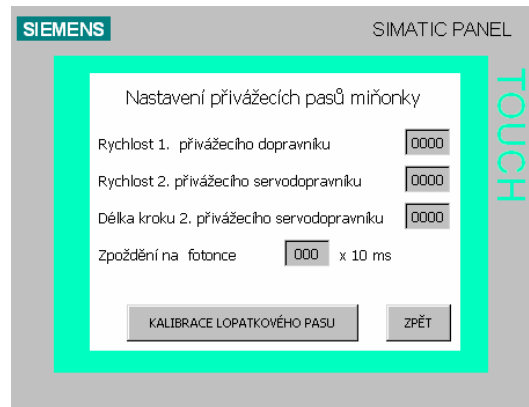
Obrázek 3.15 : Hlavní obrazovka touch panelu po zapnutí



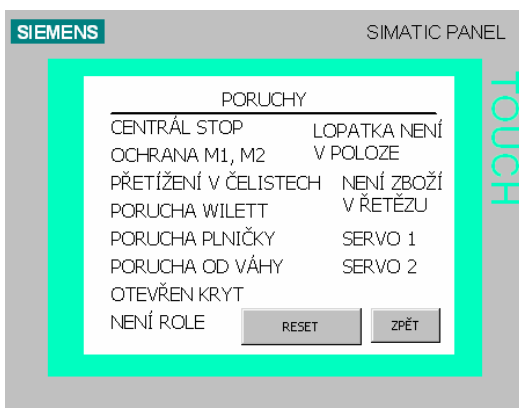
Obrázek 3.18 : Obrazovka nastavení přivážecích dopravníků pro variantu Čokotatranky



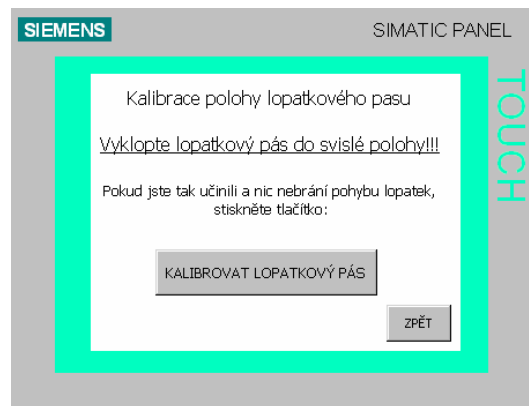
Obrázek 3.16 : Obrazovka provozních nastavení



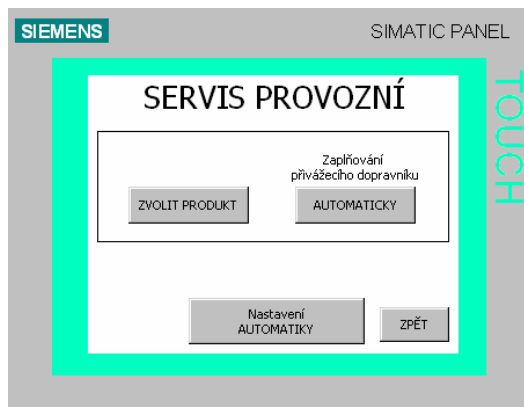
Obrázek 3.19 : Obrazovka nastavení přivážecích dopravníků pro variantu Miňonky



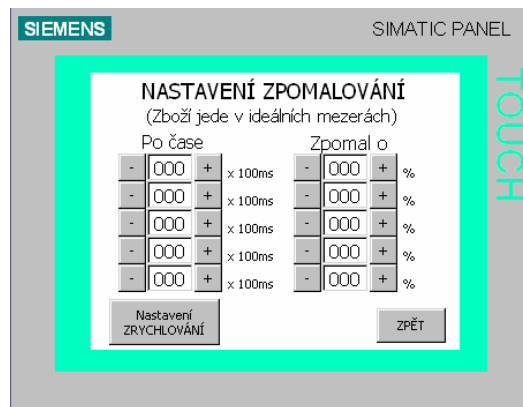
Obrázek 3.17 : Obrazovka chybových hlášení



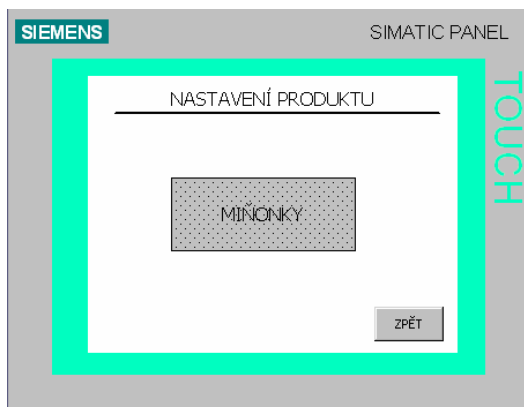
Obrázek 3.20 : Obrazovka kalibrace lopatkového pasu



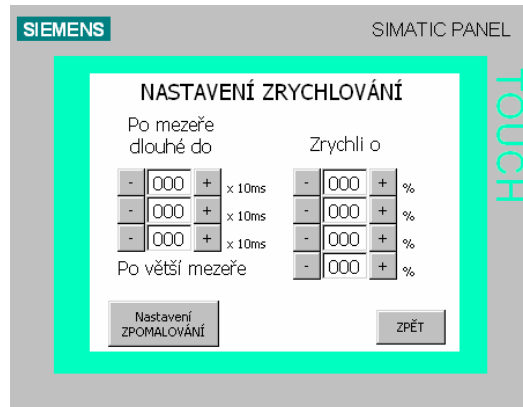
Obrázek 3.21 : Obrazovka provozního servisu



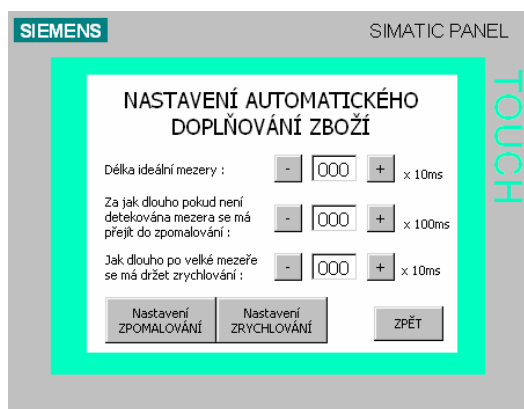
Obrázek 3.24 : Obrazovka nastavení parametrů pro zpomalování regulátoru rychlosti dopravníků



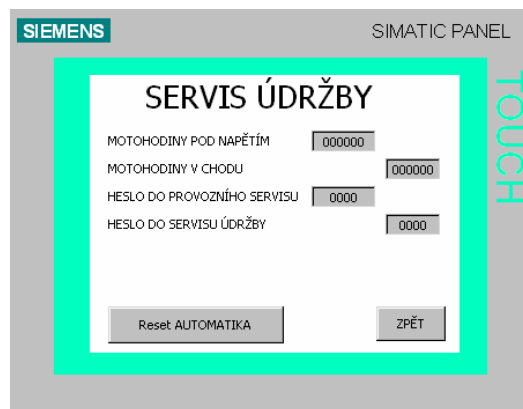
Obrázek 3.22 : Obrazovka nastavení právě baleného zboží



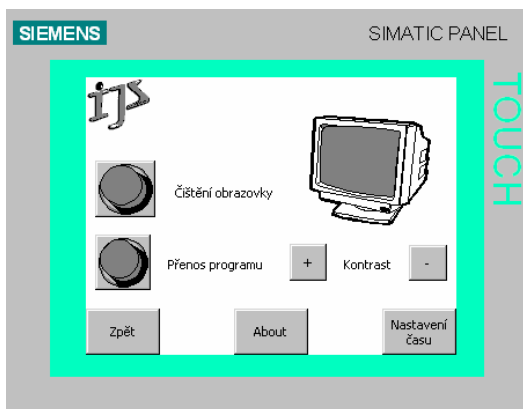
Obrázek 3.25 : Obrazovka nastavení parametrů pro zrychlování regulátoru rychlosti dopravníků



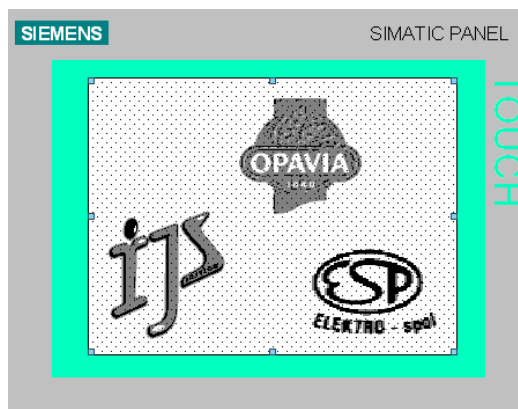
Obrázek 3.23 : Obrazovka nastavení hlavních parametrů regulátoru rychlosti dopravníků



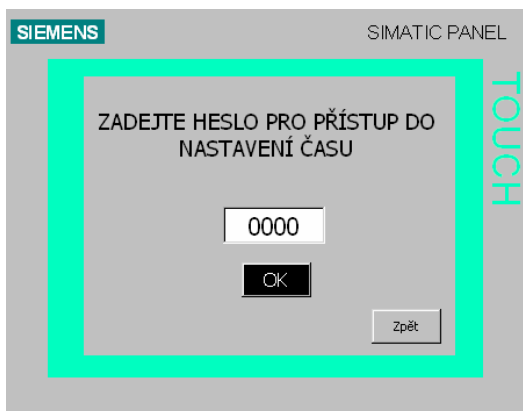
Obrázek 3.26 : Obrazovka nastavení servisu údržby



Obrázek 3.27 : Obrazovka pro nastavení Touch Panelu



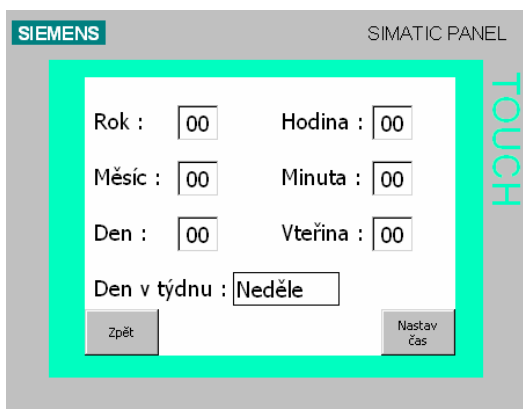
Obrázek 3.30 : Obrazovka About



Obrázek 3.28 : Obrazovka pro zadání hesla při vstupu do nastavení času



Obrázek 3.31 : Obrazovka pro zadání hesla při přístupu do chráněných nastavení



Obrázek 3.29 : Obrazovka pro zadání aktuálního času



## Kapitola 4

# Realizace

Celková koncepce tohoto projektu se skládala ze dvou částí. V první části jsme vyvíjeli prototyp, který jsme testovali přímo na jedné z balíček v původním rozestavení. Vzhledem k tomu, že jsme nepracovali pouze na tomto projektu, ale i na projektech jiných, vývoj prototypu trval přibližně dva měsíce. Ve druhé části jsme již odladěný prototyp připravili k jeho zadání do výroby čtyř kompaktních kusů, které jsme následně při plánované odstávce linky namontovali. Doba vyhrazená pro úpravu linky byla stanovena od 18.12.2006 do 7.1.2007.

### 4.1 Montáž zařízení

Návrh konečného zařízení vycházel z odladěného, funkčního prototypu, nicméně z důvodu provádění velkého množství úprav a zřehlednění zapojení řídicího systému, bylo navrženo zapojení nové, vycházející z poslední verze, nicméně transparentnější.

V první fázi jsme tedy vyrobili čtyři kusy řídicích rozvaděčů. Z důvodu zásobenosti materiálem a přijatelnějším pracovním prostředím se toto realizovalo v dílnách firmy.

Ve druhé fázi, po odstavení linky z běžného provozu, jsme museli odpojit a bezpečně zajistit elektrické přívoody k balíčkářům, dopravníkům a ostatním strojům tak, aby je bylo možno libovolně přesouvat. V této fázi k tomu určená firma zajistila přesun jednotlivých strojů na nově určená místa a zbývající stroje pak byly transportovány do skladu závodu.

Ve třetí fázi bylo nutné zanalyzovat stávající elektrické rozvody a rozhodnout o jejich použití v novém řešení linky. Zde je třeba uvést, že ze stávajících rozvodů nebylo použito téměř nic a byly tak nataženy nové kabelové trasy.

Ve čtvrté fázi bylo třeba upravit stávající silový rozvaděč a řídicí rozvaděč hlavních dopravníků linky. Tato část byla z hlediska časové náročnosti nejobtížnější, jelikož bylo nutné provést poměrně razantní změny jak v rozvodech silové části, tak v části řídicí.

V páté fázi se provedla kompletáž nových přívazecích dopravníků balíček a připojení celého zařízení na nové řídicí rozvaděče.

V šesté fázi pak bylo celé zařízení testováno, aby se předešlo, v co možná největší míře, komplikacím při začátku provozu nové koncepce linky.

### 4.2 Odladění

Při prvním spuštění linky do běžného provozu se podle předpokladů vyskytlo několik komplikací. Především bylo nutné optimalizovat řídicí konstanty z operátorského panelu, seřadit pozice jednotlivých senzorů a upravit jejich citlivost. Po tomto zásahu byla funkce řídicího systému stabilizována, nicméně se začaly projevovat nedostatky na strojní části zařízení. Především bylo nutné stabilizovat "cestování" pasů, upravit nastavení vodících lišt a přítlačných lišt.

### 4.3 Servisní zásahy

Především v prvních čtrnácti dnech provozu bylo nutné stabilizovat systém z důvodu vymezení vůlí na nových strojích a provést dílčí úpravy, především na strojní části, vedoucí k zefektivnění. Sami operátoři na lince pak přišli s návrhem několika úprav, které, pokud byly konstruktivní, jsme se pokusili zrealizovat. Příkladem je regulátor rychlosti dopravníčků. V původním řešení byl nastaven na jednu sadu parametrů, kterou nebylo možno editovat. Na návrh operátorů jsem poté doprogramoval obrazovky pro nastavení nezávislých konstat regulátorů pro různé druhy zboží (receptury).

V současné době je zařízení stabilizované a funguje spolehlivě. Je zde třeba ovšem opět zdůraznit, že úpravy ve smyslu seřizování sensoriky a řídicích konstant jsou nutností, kterou je však schopna nezávisle provádět obsluha linky.



## Kapitola 5

# Závěrečné zhodnocení

### 5.1 Nesrovnalosti s teoretickými předpoklady

Při řešení teoretické úlohy a při jejím následném převedení do praxe jsme si mohli ověřit, že je zapotřebí skutečně propracovaný vývoj. S první variantou řešení, jakou bylo použití strkadla, jsme strávili téměř měsíc, než jsme dospěli k rozhodnutí, že právě tato varianta, tak jak je koncepčně navržena, není vhodným řešením. Zároveň jsme však předešli použití dalšího speciálního přídatného modulu pro řízení servopohonu, čímž jsme zajistili značnou úsporu finančních prostředků. Využití tohoto modulu bylo nutné právě kvůli řízení třetího serva, neboť CPU Simatic S7 200 je navrženo tak, aby bylo schopno řídit pouze dva servopohony. Dalším zjištěním bylo, že rozdílný druh čokolády nám působí odlišně na chování difuzních optických senzorů. Z tohoto důvodu jsme se snažili jejich použití minimalizovat a nastavení jejich snímání zefektivnit. Poslední z předpokládaných skutečností bylo použití indukčních senzorů. Původní palce baliček byly totiž zhotoveny z kovu a tak se nám toto snímání zdálo jako nejefektivnější. Došlo však na výměnu kovových palců za palce z PVC a tak bylo použití indukčních senzorů nemožné. Z tohoto důvodu jsme se poté rozhodli vyměnit indukční senzory za senzory optické reflexní, které fungují spolehlivě.

### 5.2 Práce v areálu závodu

Velice cennou zkušeností byla samotná práce přímo v areálu závodu společnosti Opavia-LU a.s., DELI Lovosice. Před vstupem do výrobních prostor je pracovník proškolen o zásadách pohybu a chování se v závodě. Jelikož se jedná o potravinářský závod, je zde také nutné čisté pracovní ustrojení a pokrývka hlavy zabraňující případnému zachycení vlasů do rotačních strojů. Není zde také povoleno konzumovat potraviny ani pít ve výrobních prostorách. K tomu jsou zde určeny speciální odpočinkové místnosti. Za každých okolností se pak pracovník musí chovat ohleduplně a bezpečně tak, aby minimalizoval případné riziko vzniku pracovního úrazu.

### 5.3 Závěr

Realizace celého projektu, která, i co se týká vývoje, trvala přes tři měsíce, proběhla v konečném stádiu úspěšně a nyní je plně využita k výrobním účelům. Pokud si tedy dnes zakoupíme některý ze zde vyráběných druhů sortimentu, jako jsou například Miňonky nebo Čokotatranky, můžeme konstatovat, že toto zboží bylo zabaleno pomocí námi navrženého systému přivázacích dopravníků do baliček.



# Literatura

- [1] Ivan Švarc : ZÁKLADY AUTOMATIZACE  
Skriptum VUT Brno, říjen 2002
  
- [2] Programovatelný automat Siemens Simatic S7-200 Systémový manuál
  
- [3] Frekvenční měnič Lenze 8200 Vector Systémový manuál
  
- [4] Servopohony Omron SmartStep Systémový manuál
  
- [5] Plachý Ladislav : Nové operátorské panely společnosti Siemens  
Časopis Automatizace, ročník 48, číslo 1, strana 33
  
- [6] <http://www1.siemens.cz/ad/current/prezentace/as/index.php?mi=1>
  
- [7] [http://omron-industrial.com/cz\\_cs/home/](http://omron-industrial.com/cz_cs/home/)
  
- [8] <http://www.lenze.cz/>
  
- [9] [http://cs.wikipedia.org/wiki/Asynchronn%C3%AD\\_motor](http://cs.wikipedia.org/wiki/Asynchronn%C3%AD_motor)
  
- [10] [http://edumat.hyperlink.cz/ucebnice/index\\_u.htm](http://edumat.hyperlink.cz/ucebnice/index_u.htm)
  
- [11] [http://www.balluff.cz/bos\\_principy-definice.asp](http://www.balluff.cz/bos_principy-definice.asp)

