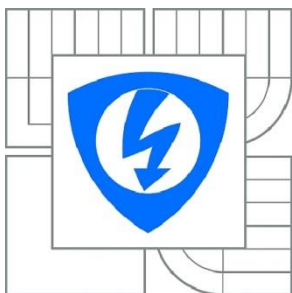




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

OVLÁDÁNÍ MODELŮ POMOCÍ PLC

CONTROL OF LAB MODELS USING PLC

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

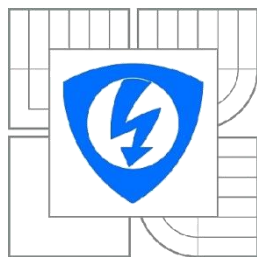
LUKÁŠ VOBĚJDA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. RADEK ŠTOHL, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Lukáš Vobejda

ID: 134440

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Ovládání modelů pomocí PLC

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s PLC firmy Rockwell Automation a vývojovým prostředím RS Logix 5000.
2. Navrhněte a realizujte připojení modelů Třídění klád, Dopravník beden k PLC.
3. Navrhněte a realizujte software ve 3 různých programovacích jazycích pro jednotlivé modely.
4. Porovnejte způsob programování pro dané modely v daných jazycích.
5. Navrhněte a realizujte vizualizaci modelů.
6. Ověřte funkčnost.

DOPORUČENÁ LITERATURA:

Logix5000 Controllers General Instructions (Reference Manual). Milwaukee: Rockwell Automation, Inc. 2008.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání: 11.2.2013

Termín odevzdání: 27.5.2013

Vedoucí práce: Ing. Radek Štohl, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce se zabývá ovládáním modelů pomocí PLC firmy Rockwell Automation. S využitím systému CompactLogix, programovacího prostředí RSLogix 5000, komunikačního prostředí RSlinx Classic a softwaru pro vizualizaci FactoryTalk View Studio. První část práce se zabývá popisem modelů, jejich připojením k PLC a prostředí. Ve druhé části jsou popsány a porovnány programovací jazyky (ladder diagram, diagram funkčních bloků a sekvenčně funkční grafy). Poslední část práce se zabývá vizualizací obou modelů ve FactoryTalk View Studiu.

Klíčová slova

RSLogix 5000, Ladder diagram, Diagram funkčních bloků, Sekvenčně funkční grafy, FactoryTalk View Studio

Abstract

This work deals with the control of lab models using PLC Rockwell Automation. As well as the use of CompactLogix, RSLogix 5000 programming environment, RSlinx Classic communication environment and software for process visualization such as FactoryTalk View Studio. The first part deals with the description of the models and their connection to a PLC environment. The second part describes and compares programming languages (ladder diagram, function block diagram and sequential function charts). The last part deals with the visualization of both models in FactoryTalk View Studio.

Keywords

RSLogix 5000, Ladder diagram, Function Block Diagram, Sequential Function Charts, FactoryTalk View Studio

Bibliografická citace:

VOBEJDA, L. *Ovládání modelů pomocí PLC*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 35s. Vedoucí bakalářské práce byl Ing. Radek Štohl, Ph.D..

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Ovládání modelů pomocí PLC jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **27. května 2013**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlovi, Ph.D., za účinnou metodickou, pedagogickou a odbornou pomoc a také Ing. Michalu Šírovi za velice zajímavé nápady a cenné rady, které přispěly ke zpracování mé bakalářské práce.

V Brně dne: **27. května 2013**

.....
podpis autora

Obsah

1	Úvod	9
2	Rockwell Automation.....	10
2.1	Systém CompactLogix	10
2.2	RSLinx Classic.....	11
2.3	RSLogix 5000	11
2.4	FactoryTalk View Studio	11
3	Modely	12
3.1	Model Třídění kmenů.....	12
3.1.1	Zadání úlohy.....	12
3.1.2	Popis připojení konektorů	13
3.1.3	Vývojový diagram.....	14
3.2	Model Počítání beden.....	15
3.2.1	Zadání úlohy.....	15
3.2.2	Popis připojení konektorů	16
3.2.3	Vývojový diagram.....	17
4	Programování	19
4.1	Základní koncepce programovacích jazyků	19
4.1.1	Ladder diagram	19
4.1.2	Function Block Diagram	20
4.1.3	Sequential Function Charts	21
4.1.4	Structured text	21
4.2	Aplikace programovacích jazyků na zadané úlohy	22
4.2.1	Programování – Třídění kmenů.....	22
4.2.2	Programování – Počítání beden.....	27
5	Návrh vizualizace	29
5.1	Návrh vizualizace pro model Třídění kmenů	29
5.2	Návrh vizualizace pro model Počítání beden	31
	Závěr	33

SEZNAM OBRÁZKŮ

Obrázek 1: CompactLogix [1]	10
Obrázek 2: Schéma modelu	12
Obrázek 3: Vstupy a výstupy konektorů.....	13
Obrázek 4: Vývojový diagram programu	14
Obrázek 5: Schéma modelu	16
Obrázek 6: Vstupy a výstupy konektorů.....	16
Obrázek 7: Vývojový diagram.....	17
Obrázek 8: Vstupní a výstupní instrukce [2]	19
Obrázek 9: Struktura programu [3].....	20
Obrázek 10: Sekvenčně funkční graf [4]	21
Obrázek 11: SFC: výchozí stav	22
Obrázek 12: LD: kmen na dopravníku	23
Obrázek 13: LD: zastavení dopravníku	23
Obrázek 14: FB: kmen na dopravníku	24
Obrázek 15: FB: zapínání a vypínání dopravníku	24
Obrázek 16: SFC: kmen na dopravníku.....	24
Obrázek 17: LD: nastavení proměnné <i>Dlouha</i>	25
Obrázek 18: FB: nastavení proměnné <i>Stredni</i>	25
Obrázek 19: SFC: vyhodnocení velikosti	25
Obrázek 20: Zařazení krátkého kmene do boxu 1	26
Obrázek 21: SFC: časovač.....	27
Obrázek 22: LD: čítač.....	27
Obrázek 23: FB: čítač	27
Obrázek 24: SFC: čítač	28
Obrázek 25: Vizualizace modelu	29
Obrázek 26: FTWS: animace pohybu dopravníku	30
Obrázek 27: FTWS: animace pohybu kmenů.....	30
Obrázek 28: FTWS: dlouhý kmen v boxu 3.....	31
Obrázek 29: Vizualizace modelu	31
Obrázek 30: FTWS: animace dopravníku	32

1 ÚVOD

Tato práce se zabývá ovládáním modelů pomocí PLC (programovatelného logického automatu) firmy Rockwell Automation. Popisuje systém CompactLogix, programovací prostředí RSLogix 5000, komunikační prostředí RSlinx Classic a software pro vizualizaci technologických procesů a aplikací FactoryTalk View Studio, firmy Rockwell Automation.

Modely Třídění kmenů a Počítání beden jsou připojeny k PLC firmy Siemens pomocí konektoru CANNON. První část práce se zabývá popisem modelů, laboratorních úloh k nim vytvořených, popisem připojení a vytvořením obdobného připojení k PLC firmy Rockwell Automation.

V následující části je popsáno prostředí RSLogix 5000 a programovací jazyky, které software nabízí (Ladder Diagram, Function Block Diagram - diagram funkčních bloků, Sequential Function Charts – sekvenčně funkční diagramy, Structured Text - strukturovaný text). Z toho tři programovací jazyky (Ladder Diagram, Function Block Diagram, Sequential Function Charts) jsou aplikovány na vypracování úloh k zadaným modelům. Smyslem této práce je porovnání všech třech programovacích jazyků na stejných úlohách, ukázka způsobu programování a odlišností mezi nimi.

Poslední část se zabývá vizualizací obou modelů v prostředí FactoryTalk View Studio. Software umožňuje grafické vytvoření obou modelů. Dále také umožňuje komunikaci s prostředím RSLogix 5000. Vytvořený program tak může přímo komunikovat s grafickým prostředím pomocí tagů. Jednotlivým grafickým částem je možné přidělit vlastnosti, které se mění podle hodnot přidělených tagů. Prostředí FactoryTalk View Studio umožňuje simulovat skutečné modely jak pohybem, tak signalizací. Informuje nás o správném chodu programu, případně o chybách, které se vyskytly.

2 ROCKWELL AUTOMATION

2.1 Systém CompactLogix

Systém CompactLogix[1] je navržený tak, aby zprostředkoval pokyny prostředí Logix. Systém může být složen ze samotného řídicího systému s jednou skupinou modulů vstupů /výstupů, a k tomu komunikace DeviceNet. Složitější systémy mohou obsahovat další síť nebo řízení pohybu a řízení bezpečnostních funkcí. Procesory CompactLogix tvoří součást systému Integrované Architektury, používají stejný programovací software, dále síťový protokol a způsob práce s informacemi jako všechny procesory Logix, čímž poskytují společné vývojové prostředí. CompactLogix nabízí tři různé typy procesorů:

- CompactLogix 5370 L3
- CompactLogix 5370 L2
- CompactLogix 5370 L1

Jelikož pracujeme s procesorem typu CompactLogix 1769-L32E, spadáme do oblasti procesorů CompactLogix 5370 L3. Procesory typu CompactLogix 5370 L3 poskytují škálovatelné, dostupné řízení. Jsou ideální pro aplikace od malých samostatných zařízení po krokově polohovací stoly s vysokým výkonem, procesní rámy, vkladače pro skupinová balení, zařízení pro skládání beden, obalové systémy apod. Procesory CompactLogix 5370 L3 rovněž poskytují integrované řešení pro řízení pohybu.



Obrázek 1: CompactLogix [1]

Tabulka 1: Technické parametry systému CompactLogix [1]

Vlastnosti	1769-L32E
Dostupná uživatelská paměť	750 KB
Karta CompactFlash	1784-CF128
Komunikační porty	1port EtherNet/IP 1port RS-232(DF1 nebo ASCII)
Kapacita rozšiřitelnosti dalšími moduly	16 modulů 1769
Výkonnost napájecího zdroje z hlediska vzdálenosti od něho	4 moduly

2.2 RSLinx Classic

RSLinx Classic [1] je prostředí, které se používá pro komunikaci mezi prostředím RSLogix 5000 a systémem CompactLogix. Využívá propojení typu EtherNet/IP a DeviceNet.

- Síťový protokol průmyslové sítě Ethernet Industrial (EtherNet/IP) je ověřený standard pro průmyslové sítě. Podporuje zasílání vstupních/výstupních zpráv v reálném čase a vzájemnou výměnu zpráv. Síť EtherNet/IP využívá dostupné komunikační čipy a fyzická media Ethernet.
- Síť DeviceNet je otevřená, nízkoúrovňová síť. Poskytuje spojení mezi jednoduchými průmyslovými zařízeními (jako jsou čidla a akční členy) a zařízeními vyšších úrovní (jako jsou procesory a počítače).

2.3 RSLogix 5000

RSLogix 5000 Enterprise Series [1] je software navržený pro práci s platformami řídicích systémů. Jedná se o softwarový balíček odpovídající IEC 61131-3, který nám pro vývoj aplikačních programů nabízí editory kontaktních schémat, strukturovaného textu, diagramů funkčních bloků a sekvenčního programování. Je tedy možné vytvářet vlastní instrukce zabalením určité logické části v kterémkoli programovacím jazyce do podoby tzv. Add-On instrukcí.

2.4 FactoryTalk View Studio

FactoryTalk View Studio [1] je konfigurační software pro vývoj a testování vizualizačních aplikací. Je to společný vývojový editor pro aplikace FactoryTalk View Machine Edition i Site Edition.

FactoryTalk View ME a FactoryTalk View SE jsou softwarové produkty pro vizualizaci technologických procesů, vyvinuté s jednotným vzhledem, prostředím a navigací. Umožňují přímou komunikaci s prostředím RSLogix 5000. Pro tvorbu vizualizace byl použit software FactoryTalk View Machine Edition. Jedná se o vizualizační software na úrovni jednotlivých strojů. Slouží k vytváření aplikací jak pro otevřená, tak i kompaktní operátorská rozhraní, pro monitorování a řízení jednotlivých strojů nebo malých procesů.

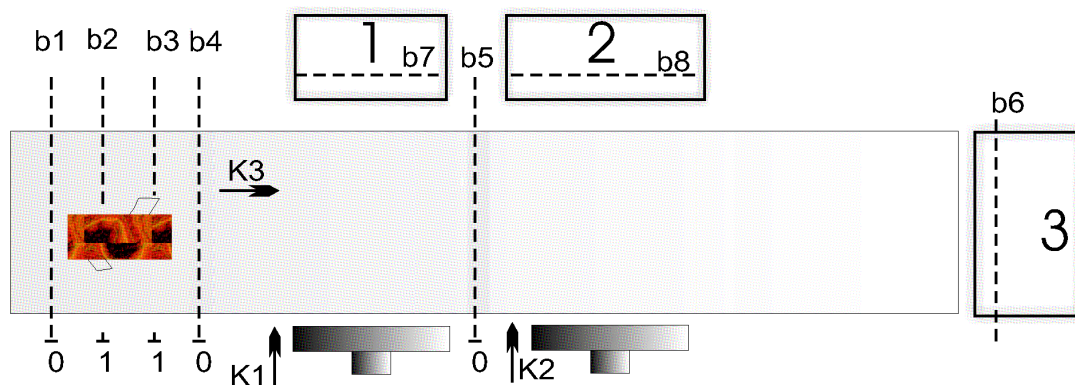
3 MODELÝ

Připojení modelů k PLC bylo vytvořeno prostřednictvím konektoru CANNON 25 a vícežilového kabelu, zapojeného přímo na digitální V/V karty PLC. Popis připojení +24V, digitálních vstupů/výstupů - (DI/DO) a zemnicího vodiče – GND pro jednotlivé konektory je popsán pro jednotlivé úlohy. Připojení konektorů bylo provedeno na základě připojení modelů k PLC firmy Siemens. Modely mají tedy kompatibilní připojení pro PLC firmy Siemens a Rockwell Automation.

3.1 Model Třídění kmenů

3.1.1 Zadání úlohy

Kmeny tří různých velikostí přijíždějí po dopravníku. Velikost kmenů se rozlišuje soustavou čtyř čidel **b1**, **b2**, **b3** a **b4**. Krátké kmeny se zatlačí manipulátorem **K1** do boxu 1. Manipulátor **K1** se ovládá 2s dlouhým impulsem tak, aby kmen zachytil ve správné době. Po spadnutí kmenu do boxu 1 (identifikováno **b7**) je možné očekávat další kmen. Střední kmeny se zatlačí manipulátorem **K2** do boxu 2. Manipulátor se ovládá 2s dlouhým impulsem, se zpožděním od čidla **b5** tak, aby kmen zachytil ve správné době. Po spadnutí kmenu do boxu 2 (identifikováno **b8**) je možné očekávat další kmen. Dlouhé kmeny se nechají dojet na konec pásu, po odjetí z čidla **b6** jsou v boxu a lze očekávat další kmen.

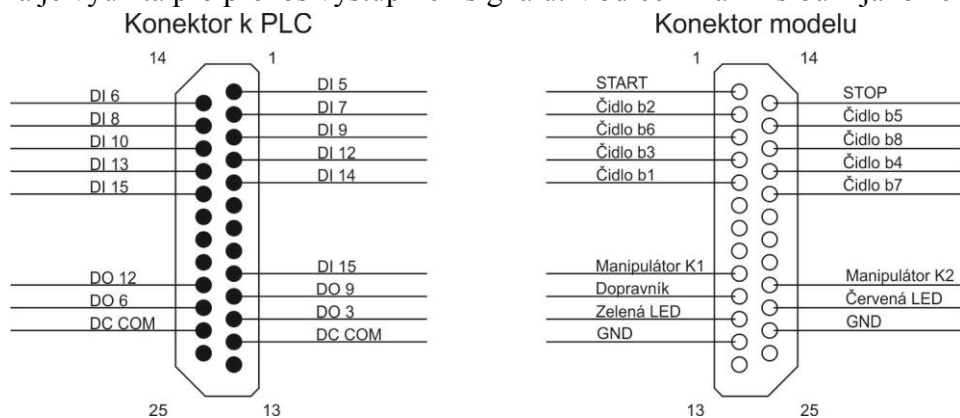


Obrázek 2: Schéma modelu

Pokyny: Systém se uvede do chodu stisknutím tlačítka START, zastavit lze v libovolném okamžiku tlačítkem STOP. Na identifikačních světlech zobrazuje možnost položení dalšího kmene (červená – kmen je na páse, zelená – je možné položit další kmen).

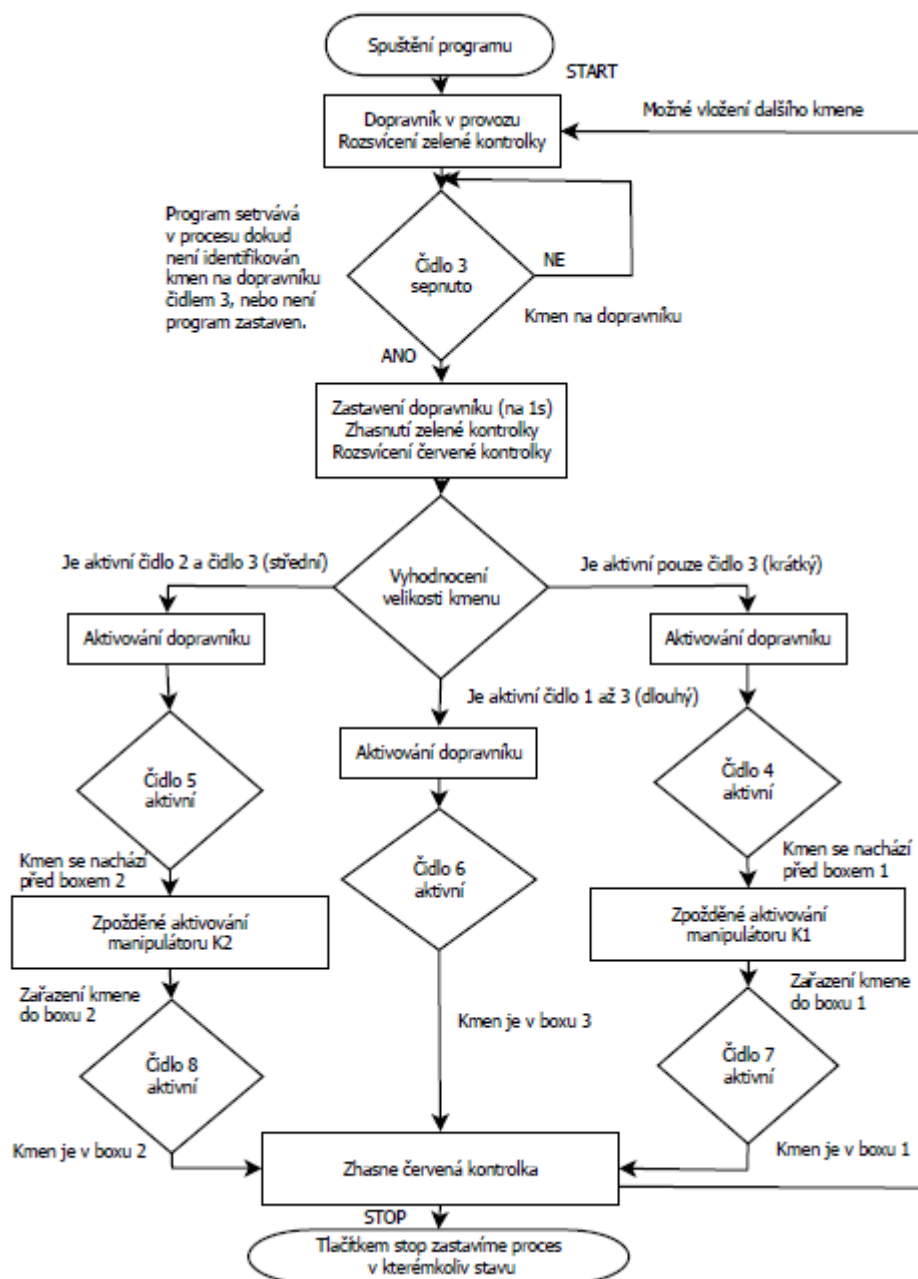
3.1.2 Popis připojení konektorů

Na Obrázek 6 je zobrazeno připojení modelu k PLC skrze konektor CANNON 25. První polovina konektoru (od pinu 1) slouží k přenosu vstupních signálů. Druhá polovina je využita pro přenos výstupních signálů. Vodiče 12 a 24 slouží jako zemní.



Obrázek 3: Vstupy a výstupy konektorů

3.1.3 Vývojový diagram



Obrázek 4: Vývojový diagram programu

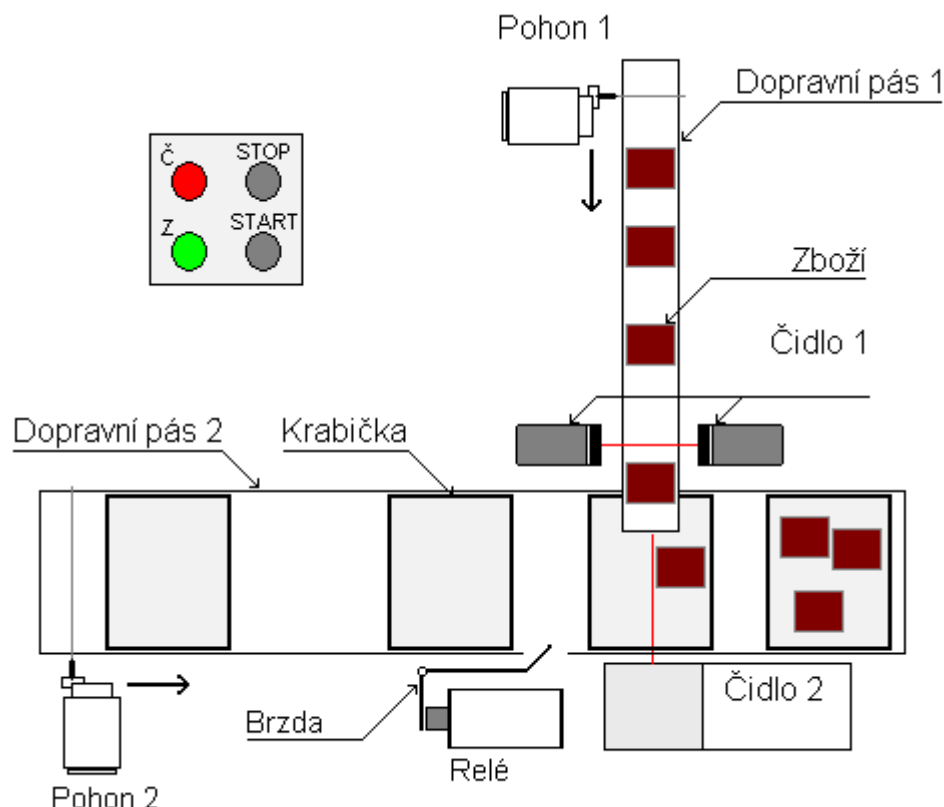
Na Obrázek 4 je zobrazen vývojový diagram programu pro model Třídění kmenů. Po spuštění programu setrvává v klidu ve výchozí pozici a čeká na signál od obsluhy. Stisk tlačítka START aktivuje dopravník a rozsvítí zelenou LED diodu. Program očekává vložení kmene. Kmen na dopravníku je identifikován čidlem 3. Zelená LED je deaktivována a červená LED je v log 1. Dopravník je zastaven na 1 sekundu. Probíhá vyhodnocení velikosti. Je-li aktivní pouze čidlo 3, jedná se o krátký kmen. Je-li sepnuto čidlo 2 a čidlo 3, jedná se o kmen střední. Čidlo 1 až 3 ve stavu log 1 charakterizuje dlouhý typ kmene. S vyhodnocením je zároveň dopravník uveden do provozu. Dlouhý

typ kmene pokračuje do boxu 3 na konci dopravníku, kde je indikován čidlem 6. To rozsvítí zelenou kontrolku, červená je v log 0 a program očekává vložení dalšího kmene. Jedná-li se o střední velikost kmene, kmen pokračuje po dopravníku do doby, kdy je identifikován čidlem 5. Časovačem je zpožděně aktivován manipulátor K2, který zařadí kmen do boxu 2. Čidlo 8 identifikuje kmen v boxu 2. Signalizace LED diod značí, že je možné vložit na dopravník další kmen. Kmen krátké velikosti pokračuje po dopravníku do té doby, než je identifikován čidlem 4. Kmen je zpožděným aktivováním manipulátoru K1 zařazen do boxu 1. Čidlo 6 signalizuje kmen v boxu. Program se navrácí do výchozí pozice, kdy očekává další vložení kmene. Program lze zastavit tlačítkem STOP v kterémkoli stavu. Stiskem tlačítka START program pokračuje ze stavu, ve kterém byl ukončen.

3.2 Model Počítání beden

3.2.1 Zadání úlohy

Napište program pro programovatelný automat, pro přemístění určitého počtu kusů zboží do krabic. Zboží v nepravidelných intervalech přijíždí po dopravníku č. 1 a je identifikováno čidlem č. 1. Po dopravníku č. 2 přijíždí prázdné krabice, které se mají naplnit třemi kusy zboží. Přítomnost krabice na pozici před pásem je identifikována čidlem č. 2, přičemž následující krabice je zastavena brzdou. Po naplnění třemi kusy zboží tato odjede a brzdou uvolněná prázdná krabice ji nahradí, je naplněná třemi kusy zboží a tak dále. Dopravník č. 1 se musí vypnout v případě, že po dobu 5s nedorazí žádné zboží. Systém musí reagovat na tlačítko START, jenž uvede zařízení opět do provozu. V souladu se stavem zařízení rozsvěcuje příslušnou diodu (zelená pro chod a červená pro zastavení). Systém vybavte nouzovým STOP, které v libovolném okamžiku zastaví oba dopravníky a uvolní brzdu. Po stisknutí STOP lze obnovit chod pouze stiskem tlačítka START.

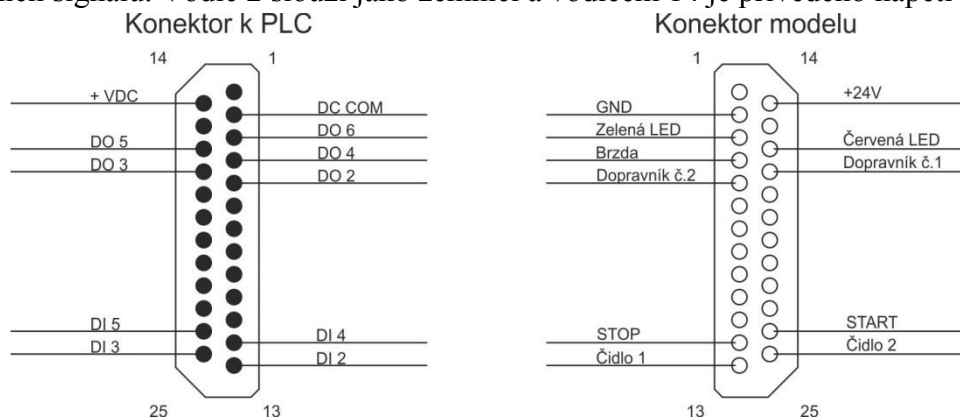


Obrázek 5: Schéma modelu

Pokyny: Čidlo 2 neindikuje pozici krabice v poloze před dopravníkem 1 – po sepnutí čidla 2 je nezbytné krabici posunout ještě o kus dál. Pro zajištění odstupu krabic před pásem 1 použijte brzdu, která podrží následující krabici. Pro pojmenování proměnných využijte symbolická jména.

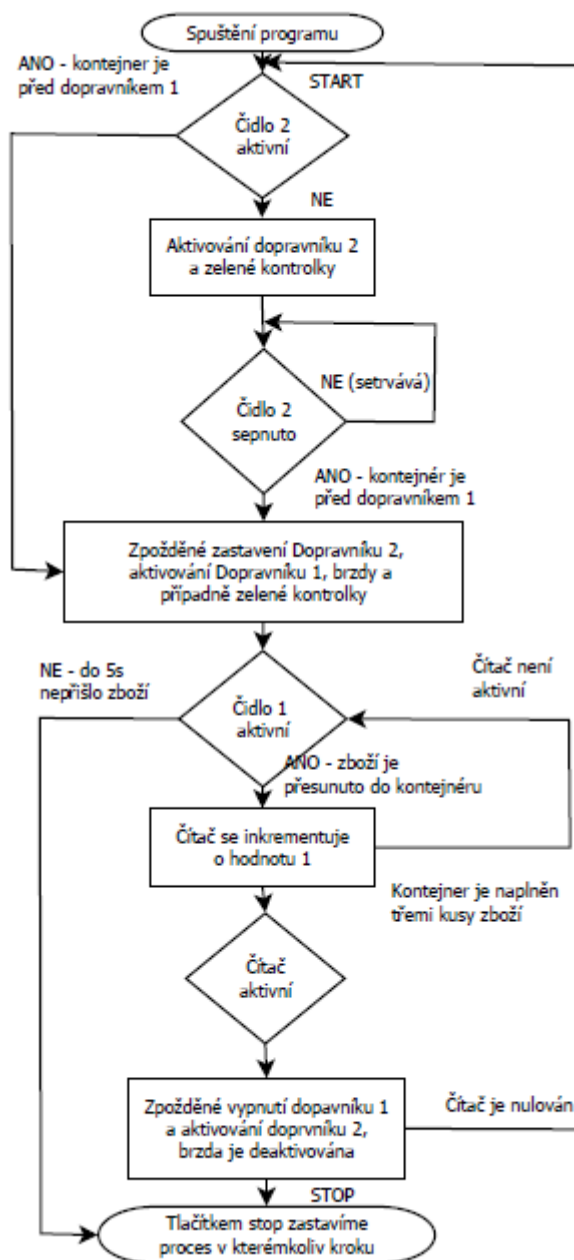
3.2.2 Popis připojení konektorů

Na Obrázek 6 je zobrazeno připojení modelu k PLC. První polovina konektoru (od pinu 1) využita naopak pro přenos výstupních signálů a druhá polovina slouží pro přenos vstupních signálů. Vodič 2 slouží jako zemní a vodičem 14 je přivedeno napětí +24V.



Obrázek 6: Vstupy a výstupy konektorů

3.2.3 Vývojový diagram.



Obrázek 7: Vývojový diagram

Na Obrázek 7 je zobrazen vývojový diagram programu pro model Počítání beden. Program po spuštění setrvává ve výchozí pozici a čeká na signál od obsluhy. Stiskem tlačítka START je aktivován dopravník 2 (nenachází-li se před dopravníkem 1 kontejner) a zelená kontrolka. V tomto stavu setrvává, dokud není aktivováno čidlo 2, které zpožděně vypne dopravník 2 a zároveň aktivuje brzdu a dopravník 1. Po dopravníku 1 přijíždí zboží, které je identifikováno čidlem 1. Čidlo 1 inkrementuje čítač o hodnotu 1. Je-li kontejner naplněn třemi kusy zboží, zpožděně se vypne dopravník 1 a sepne dopravník 2. Brzda je deaktivována. Program je ve výchozí pozici, kdy

očekává další kontejner. Program lze tlačítkem STOP zastavit v kterémkoli stavu. Z tohoto stavu opět pokračuje po stisku tlačítka START. Program je dále zastaven časovačem, pokud po dopravníku 1 nepřijde do 5 sekund žádné zboží. Je-li pak program opět spuštěn, je již kontejner identifikován čidlem před dopravníkem 1 a pokračuje ze stavu, kdy je aktivován dopravník 1

4 PROGRAMOVÁNÍ

Prostředí RSLogix5000 nabízí programovací jazyky:

- Ladder Diagram - LD
- Function Block Diagram - FB (diagram funkčních bloků)
- Sequential Function Charts – SFC (sekvenčně funkční grafy)
- Structured Text - ST(strukturovaný text)

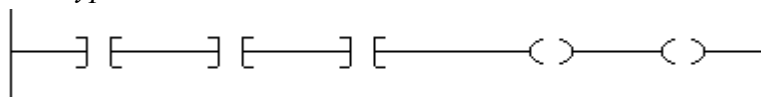
Tato práce se zabývá porovnáním pouze tří programovacích jazyků firmy Rockwell Automation, a to Ladder Diagramu, Funkčních bloků a sekvenčně funkčních grafů. Sekvenčně funkční grafy jsou však z velké části tvořeny strukturovaným textem, proto je v této práci také zahrnut.

4.1 Základní koncepce programovacích jazyků

4.1.1 Ladder diagram

Název Ladder lze volně přeložit jako žebřík, což vystihuje strukturu tohoto programu. Ladder diagram [2] je rozdělen do řádků podobně jako jednotlivé příčky žebříku. Každý řádek se pak skládá z instrukcí dvojího typu, viz Obrázek 8 [2], cituji:

- **„Input instruction** (Vstupní instrukce) *Instrukce kontroluje, porovnává nebo testuje specifické podmínky stroje nebo procesu.*
- **Output instruction** (Výstupní instrukce) *Instrukce vykonává určitou činnost, například zapne prvek, vypne prvek, zkopíruje data nebo vypočte hodnotu.“*



Obrázek 8: Vstupní a výstupní instrukce [2]

Ladder diagram nefunguje krokově. Všechny příčky programu se vyhodnocují naráz. O tom, která příčka se vykoná jako první, nerozhoduje tedy pořadí, ale vstupní instrukce.

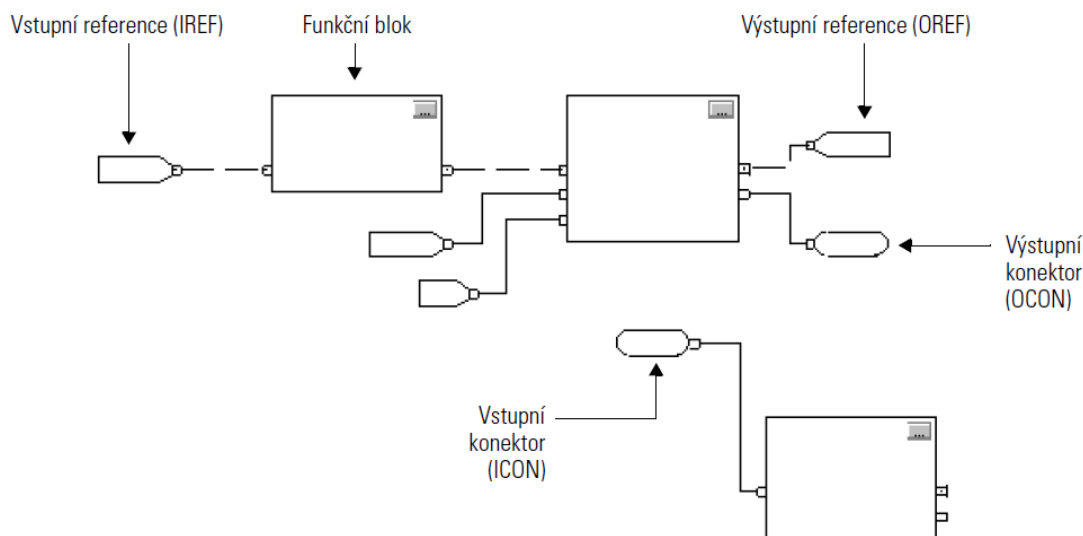
Ladder diagram dále umožňuje duplicitu vstupních i výstupních instrukcí. Pro správný chod programu je tedy nezbytné ošetřit, aby nedocházelo ke spínání a zároveň vypínání stejné výstupní proměnné, za stejných podmínek.

4.1.2 Function Block Diagram

Funkční bloky [3] mají odlišnou strukturu proti Ladder diagramu. Obsahují:

- Vstupní reference (IREF)
- Výstupní reference (OREF)
- Funkční bloky
- Vstupní a výstupní konektor (ICON, OCON)

Vstupní reference je prvek pro vložení vstupních zařízení nebo tagu. Výstupní reference slouží zase pro nastavení výstupních zařízení či tagu. Funkční bloky slouží k vykonávání funkcí. Struktura zapojení je zobrazena na Obrázek 9. Funkční bloky nemusí být tvořeny jako jedno celistvé schéma, dokonce se doporučuje pro přehlednost schéma rozdělit na menší části podle vykonávaných operací. Tato schémata lze mezi sebou provázat vstupně výstupními konektory (ICON, OCON), jenž slouží k přenosu dat. Diagram funkčních bloků je strukturou zčásti podobný LD diagramu a zčásti SFC diagramu. Pořadí provádění (tok dat) se definuje pomocí spojování prvků kabely, čímž je vytvořeno schéma podobné přičce v LD diagramu, které se vykonává zleva doprava. Umístění bloku nemá na pořadí, v jakém bude blok proveden, vliv. Ovšem při složitějším schématu, zejména při použití zpětné vazby, je třeba určit, který blok se provede jako první. V tomto případě se strukturou podobá SFC diagramu.



Obrázek 9: Struktura programu [3]

Diagram funkčních bloků obsahuje širokou škálu funkcí a operací. Mezi ně patří například logické funkce (AND, OR, NOT, XOR), klopné obvody (RS, JK, D), matematické operace, čítače a časovače.

4.1.3 Sequential Function Charts

Sekvenčně funkční grafy [4] jsou obdobou vývojového digramu. Jsou složeny z tzv. stepů, neboli kroků, ve kterých se sekvenčně vykonávají operace. Ty se zadávají v části Actions. V této části je třeba také nastavit, o jaký typ bitu se jedná. Např:

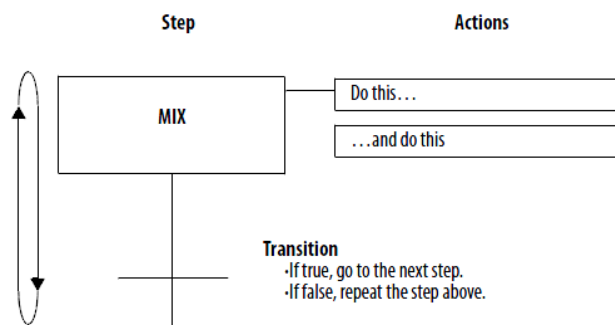
- N - Non-Stored, bit je nastaven pouze, pokud je operace aktivní
- S - Stored, bit je trvale sepnut do chvíle, dokud není jinde deaktivován
- P - Pulse, v kroku se provede pouze jednou
- R – Reset

Je možné nastavit i časování:

- D – Time Delayed, zpožděné sepnutí aktivní pouze v aktivním kroku
- DS - Delayed and Stored, zpožděné sepnutí, bit zůstane nastaven, dokud není jinde resetován

SFC obsahuje pouze úzkou část operací. Ty je však možno zadat do okénka Actions formou strukturovaného textu.

Poslední základní částí grafu je Transition, neboli přechod. Zde se píší podmínky, za kterých se posuneme do dalšího kroku. Pokud není podmínka splněna, program dále cyklicky opakuje operace kroku, ve kterém se nachází. Tyto tři základní prvky jsou zobrazeny na Obrázek 10.



Obrázek 10: Sekvenčně funkční graf [4]

4.1.4 Structured text

Strukturovaný text [5] je textový programovací jazyk, který používá k definici prováděných úkolů příkazy. Nerozlišuje velikost písma a doporučuje se na koncích řádků používat tabulátory či entery pro přehlednost. Tato znaménka nijak neovlivňují příkazy.

Strukturovaný text obsahuje přiřazovací operátor `:=`, který slouží k přidělení hodnot tagům. Je vždy ukončen středníkem. Dále využívá tzv. konstrukty, neboli podmíněné příkazy, používané k aktivaci kódu (IF THEN, CASE, FOR DO, WHILE DO, REPEAT UNTIL, EXIT), logické proměnné (AND, OR, NOT), relační a aritmetické operátory, čítače a časovače. Jelikož se však jedná o holý, textový programovací jazyk, je nutné

vždy nadefinovat v textu i vlastnosti, názvy a typy časovačů, čítačů a dalších funkcí, které budeme používat.

4.2 Aplikace programovacích jazyků na zadané úlohy

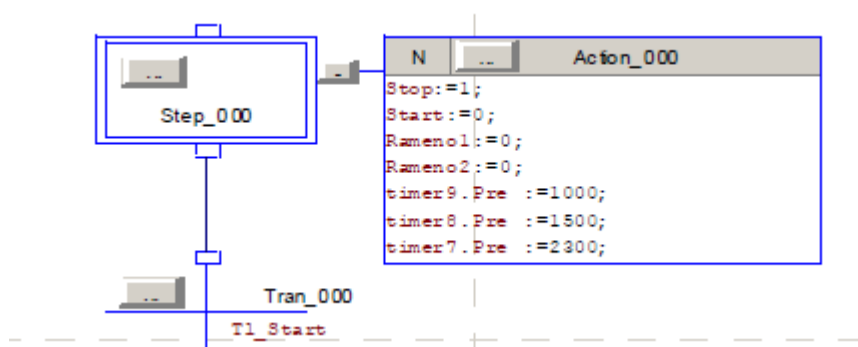
Pro zadané úlohy je kompletní software ve všech třech programovacích jazycích uveden a popsán v přílohách. V následujících částech je ukázka toho, jak by byly vyřešeny nejdůležitější body úlohy v různých programovacích jazycích a jejich rozdíly.

4.2.1 Programování – Třídění kmenů

Model Třídění kmenů obsahuje optická čidla a vyhodnocovací modul. Optická čidla pracují vždy v páru jako optické závory (vysílací a přijímací čidlo). Je tedy nezbytné si uvědomit, že pokud je optická závora ve stavu sepnutém (log 1), není mezi čidla vložena překážka. Vložíme-li překážku je optická závora ve stavu rozepnutém (log 0) Model Třídění kmenů obsahuje kromě optických čidel pouze dvě vstupní proměnné, tlačítko START a tlačítko STOP, která spouští a vypínají program. Do stisku tlačítka START obsluhou zůstává program ve výchozí pozici.

Výchozí stav

LD diagram ani FB diagram neobsahují nic takového jako výchozí stav. Průběh programu určují splněné vstupní podmínky. Oproti tomu SFC diagram má pevnou strukturu. Krokově postupuje strukturou programu podle splněných podmínek. Proto je nutné jeden z kroků nastavit jako výchozí (v nastavení kroku) a v něm nastavit počáteční podmínky. Je zde také vhodné inicializovat čítače či časovače pro přehlednost kódu, viz Obrázek 11.

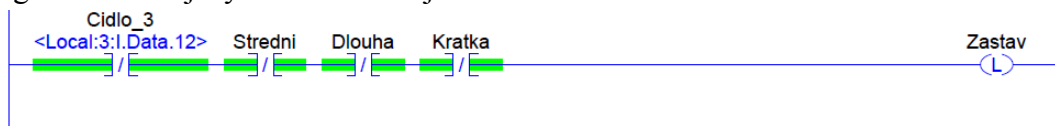


Obrázek 11: SFC: výchozí stav

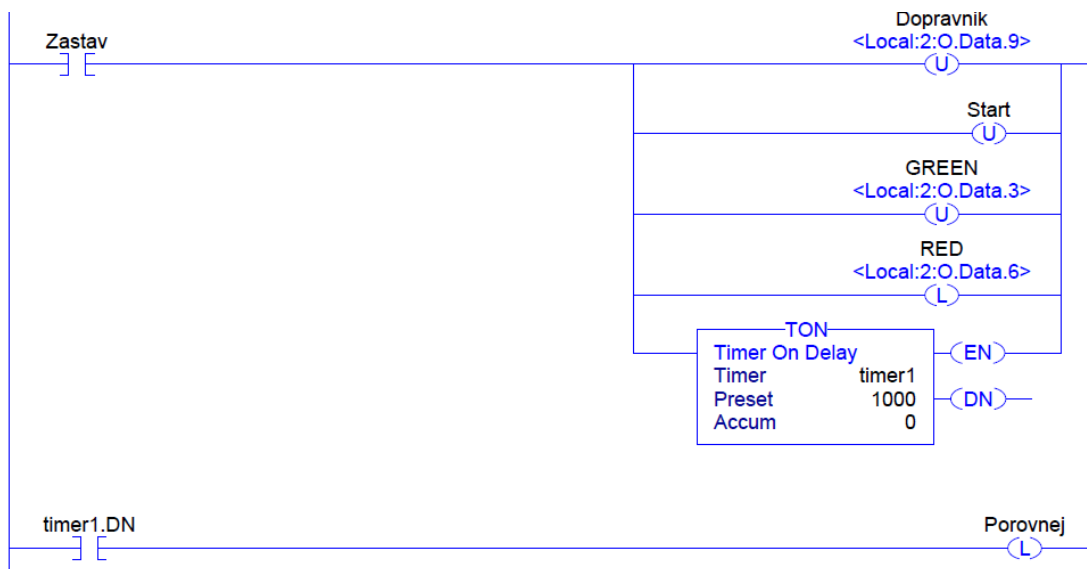
Kmen na dopravníku

Kmen na dopravníku je ve všech programovacích jazycích vždy identifikován čidlem 3. Poté je časovačem zastaven dopravník na dobu jedné sekundy. Po dočasování časovač nastaví pomocnou proměnnou, která spustí proces vyhodnocování.

Na Obrázek 12 je znázorněno řešení v LD diagramu. Čidlo 3 nastaví pomocnou proměnnou *Zastav*, která následně zastaví dopravník a spustí časovač *timer1*. Ten po dočasování sepne pomocnou proměnnou *Porovnej*. Jak již bylo řečeno, LD diagram je systém žebříku, ve kterém se všechny příčky vykonávají naráz. Proto musí být každá příčka ošetřena tak, aby docházelo k nastavení výstupních instrukcí pouze tehdy, kdy to požadujeme. V této situaci je zamezeno vyhodnocení velikosti kmene v případě, že je již některá z proměnných definující velikost nastavena. Na Obrázek 13 je použit časovač TON – timer on-delay (časovač zpožděné sepnutí), který je ve všech programovacích jazycích. Časovač je nutné inicializovat.

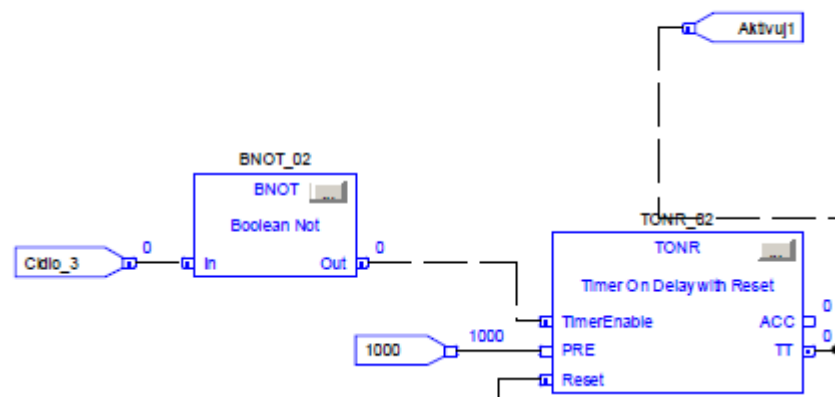


Obrázek 12: LD: kmen na dopravníku

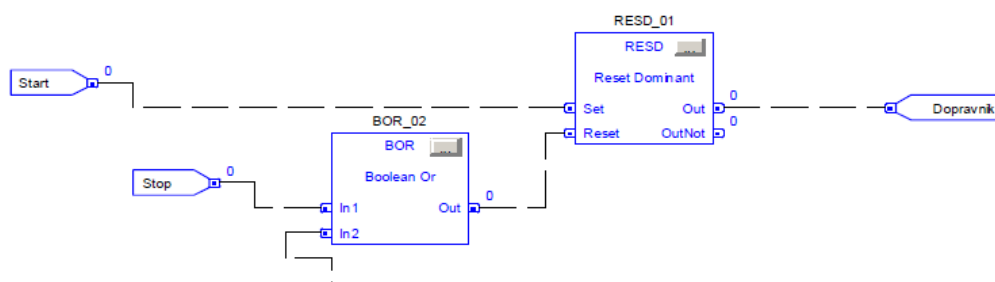


Obrázek 13: LD: zastavení dopravníku

Na Obrázek 14 je ukázka vypnutí dopravníku pomocí časovače v diagramu FB, kde čidlo 3 aktivuje časovač, který v době časování sepne proměnnou *Aktivuj1* (ta se dále využívá k vyhodnocení velikosti kmene) a provede zastavení dopravníku. Po dočasování časovač dále na čidlo nereaguje, protože časovače TON oproti časovači v LD diagramu vyžadují vždy resetování. Trvalé sepnutí nebo trvalé vypnutí výstupního bitu, jako je tomu v LD diagramu (L – latch, U - unlatch), se u diagramu FB provádí pomocí klopných obvodů. Při setování výstupní instrukce, tedy zároveň nastavujeme, za jakých podmínek bude resetována, viz Obrázek 15.

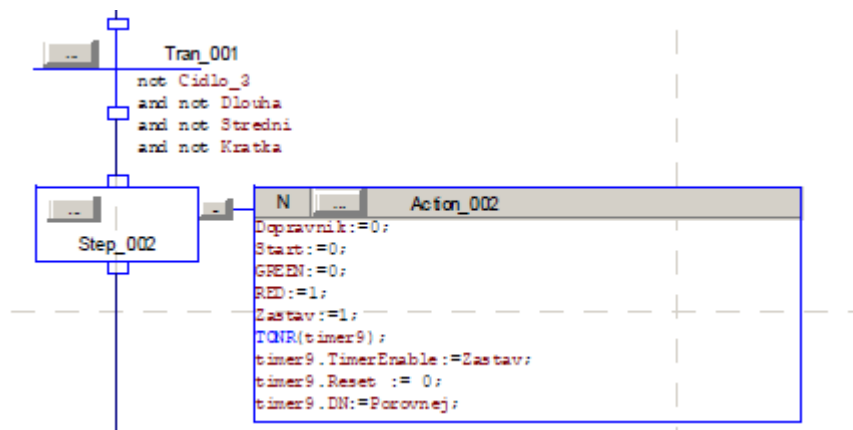


Obrázek 14: FB: kmen na dopravníku



Obrázek 15: FB: zapínání a vypínání dopravníku

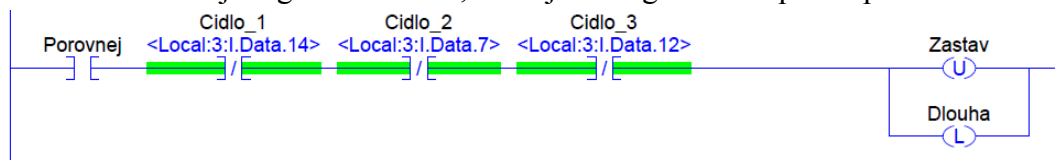
V SFC diagramu používáme nastavení proměnných pomocí N, to znamená, že proměnné jsou takto nastaveny pouze v tomto kroku, v dalším kroku je můžeme opět měnit, aniž bychom je museli nejprve resetovat. Operace v části Action se cyklicky opakují, dokud není splněna výstupní podmínka. Formou strukturovaného textu jsou zde přiřazeny proměnným stavy a vytvořen časovač. Po zastavení dopravníku časovač časuje na inicializovanou hodnotu. Po dočasování nastaví proměnnou *Porovnej* do log 1 a program pokračuje k dalšímu kroku. Ve ST je časovač TON značen jako TONR, opět vyžaduje resetování.



Obrázek 16: SFC: kmen na dopravníku

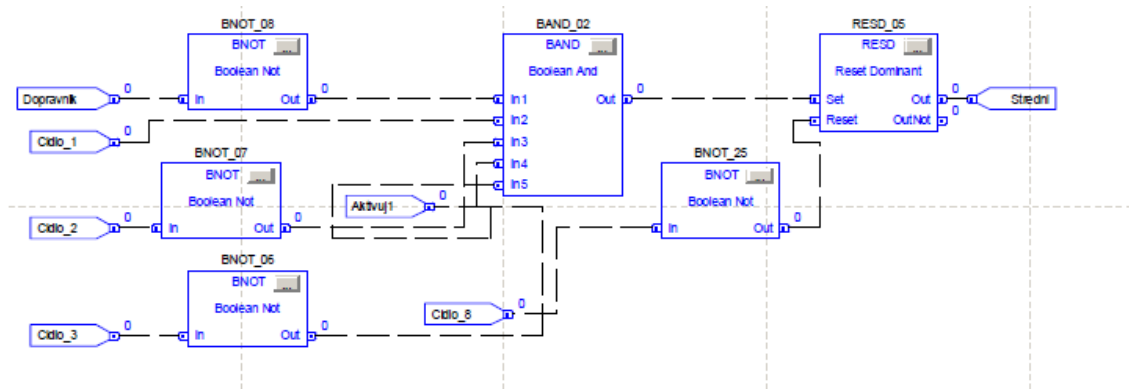
Vyhodnocení velikosti kmene

Obrázek 17 zobrazuje nastavení pomocné proměnné *Dlouha* v LD diagramu, v případě, že jsou sepnuta čidla 1 až 3. Zároveň však musí být nastavena pomocná proměnná *Porovnej*, která signalizuje, že je požadováno vyhodnocení velikosti. Sériové zapojení kontaktů tak simuluje log funkci AND, která je v diagramu FB přímo předdefinována.



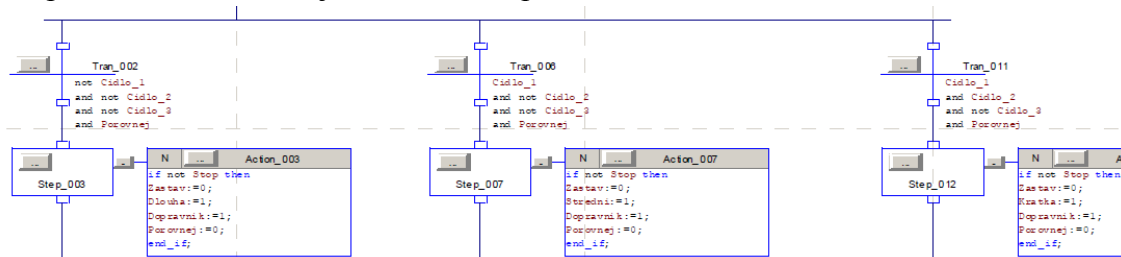
Obrázek 17: LD: nastavení proměnné *Dlouha*

Na Obrázek 18 je znázorněno řešení vyhodnocení velikosti v FB diagramu. Je zde využito funkčního bloku AND, který umožňuje zadat libovolný počet vstupních podmínek, které musí být splněny. Rozpínací kontakt zde nahrazuje blok logické negace. V tomto případě je namísto proměnné *Porovnej* použita proměnná *Aktivuj1*, která plní stejnou funkci.



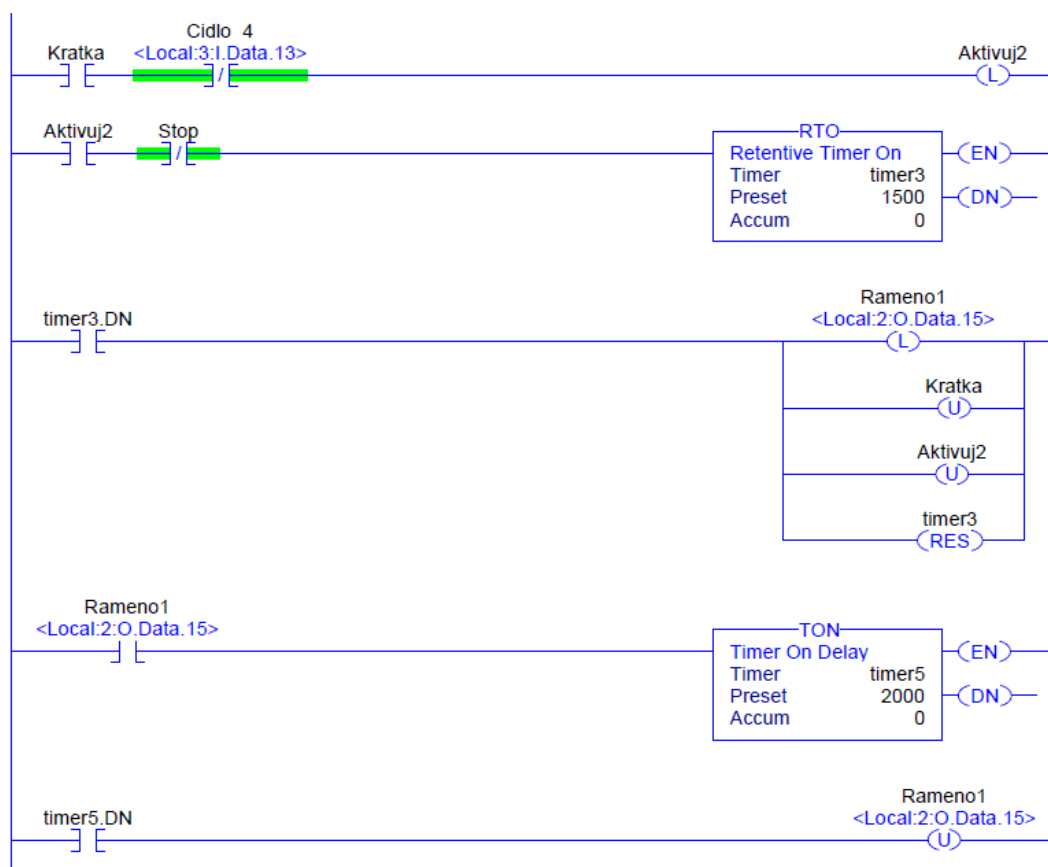
Obrázek 18: FB: nastavení proměnné *Stredni*

Obrázek 19 znázorňuje vyhodnocení velikosti kmene pomocí SFC diagramu. To je vyřešeno pomocí větvení. Každá větev obsahuje různé podmínky pro jednu velikost. Ty jsou kombinovány opět pomocí log funkcí formou strukturovaného textu, který má pro každou log funkci definován příkaz (and, or, not). V části Action pak nastavuje příslušná pomocná proměnná. Průběh programu pak probíhá takto: je-li sepnuto čidlo 1 až 3 pokračuje program krokem 003 na Obrázek 19 vlevo, kde nastaví pomocnou proměnnou *Dlouha*. Pokud je sepnuto pouze čidlo 2 a 3 pokračuje program krokem 007 uprostřed, kde nastaví proměnnou *Stredni*. Poslední cestou pokračuje v případě, že je sepnuto pouze čidlo 3, kde dojde k nastavení proměnné *Kratka*.



Obrázek 19: SFC: vyhodnocení velikosti

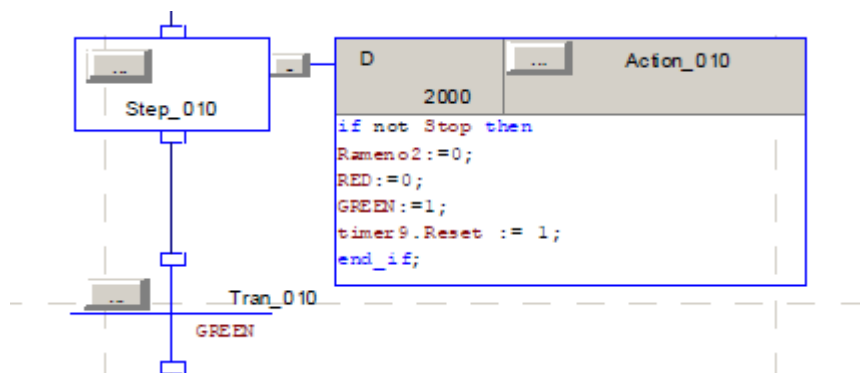
Zařazení kmene do boxu



Obrázek 20: Zařazení krátkého kmene do boxu1

Zařazení kmene do boxu v LD diagramu je zobrazeno na Obrázek 20. Po sepnutí čidla 4 se přes pomocnou proměnnou *Aktivuj2* spustí časovač, který zároveň slouží k vyměření vzdálenosti kmene od manipulátoru. Po jeho dočasování se kmen nachází před boxem a je aktivováno *Rameno1*, které zařadí kmen do boxu. Je zde použit časovač RTOR, který uchovává paměť. V případě, že zastavíme dopravník v době, kdy je časovač aktivní, zastaví se i časování. Po spuštění opět pokračuje časování od hodnoty před zastavením dopravníku, stejně jako kmen pokračuje po dopravníku od pozice, ve které setrval. Časovač RTOR je použit i v diagramu FB. V SFC diagramu je zapsán formou strukturovaného textu. Časovač je ve všech případech nutné resetovat.

Na závěr programu je manipulátor zpožděně deaktivován pomocí časovače po dvou sekundách. U SFC diagramu je vypnutí ramene vyřešeno zpožděným nastavením bitu D, který zde zastupuje časovač TON, viz Obrázek 21. Lze tedy využít časovače jak ve formě strukturovaného textu, tak ve formě SFC diagramu.



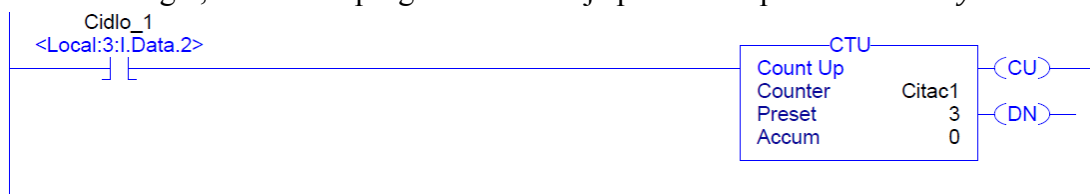
Obrázek 21: SFC: časovač

4.2.2 Programování – Počítání beden

U modelu Počítání beden jsou použita čidla, která jsou ve stavu bez překážky v log 0, v případě, kdy identifikují překážku jsou v log 1. U modelu Počítání beden se využívá kombinace obdobných příkazů jako u modelu Třídění kmenů. Rozdíl nastává v bodě, kdy je kontejner zastaven před dopravníkem 1, a po kterém se přemísťují do kontejneru kusy zboží. Ty detekuje čidlo 1

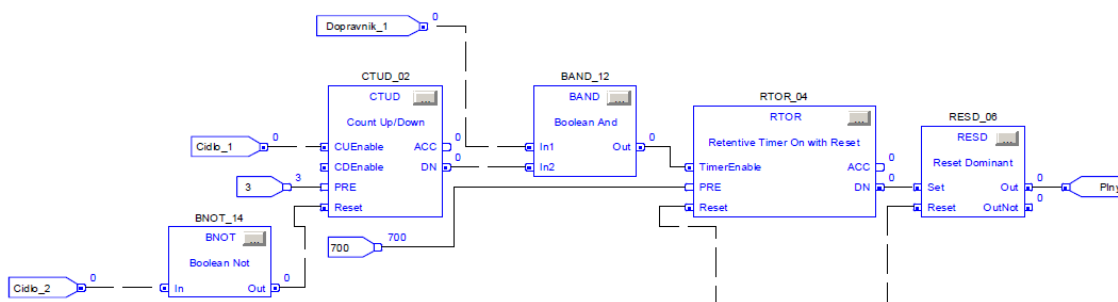
Počítání kusů zboží.

Na Obrázek 22 je zobrazeno čítání kusů v LD diagramu. CTU je čítač s inkrementem. Na náběžnou hranu signálu inkrementuje hodnotu o 1. Po dosažení požadované hodnoty přejde čítač do log 1, kde dále v programu nastavuje pomocnou proměnnou *Plny*.



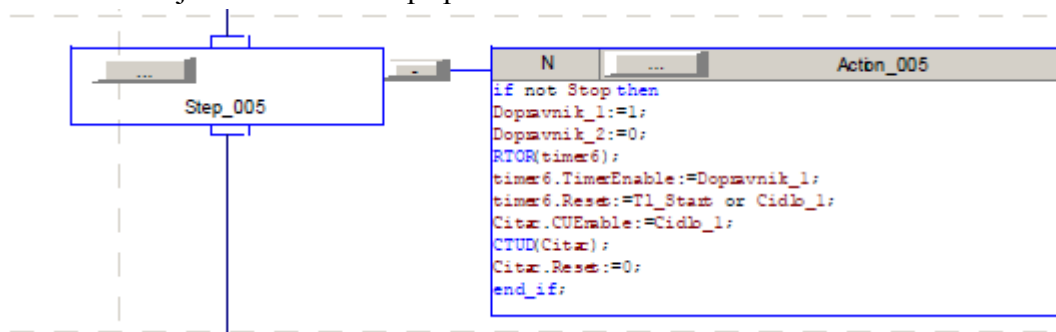
Obrázek 22: LD: čítač

Obrázek 23 zobrazuje čítání v diagramu FB. Diagram FB má také předdefinován blok pro čítač, který se zde nazývá CTUD. Jedná se o stejný čítač s inkrementem jako u LD digramu.



Obrázek 23: FB: čítač

SFC diagram nemá svoji vlastní předdefinovanou funkci pro čítání, proto je opět nutné čítač vytvořit v části Action, a to pomocí strukturovaného textu, jak je tomu na Obrázek 24. Čítač je nutné ve všech případech resetovat.



Obrázek 24: SFC: čítač

Stop programu

U modelu Počítání beden i Třídění kmenů je řešeno stopnutí programu stejným způsobem. Program je navržen tak, aby umožňoval zastavit proces v kterémkoli kroku. Po stisku tlačítka START pak pokračuje od kroku, ve kterém byl zastaven. Například u modelu Počítání beden, pokud nepřijede po dopravníku 1 po dobu pěti sekund žádné zboží, systém se automaticky vypne. Po spuštění program opět pokračuje od kroku, kdy je kontejner před dopravníkem a očekává vložení zboží.

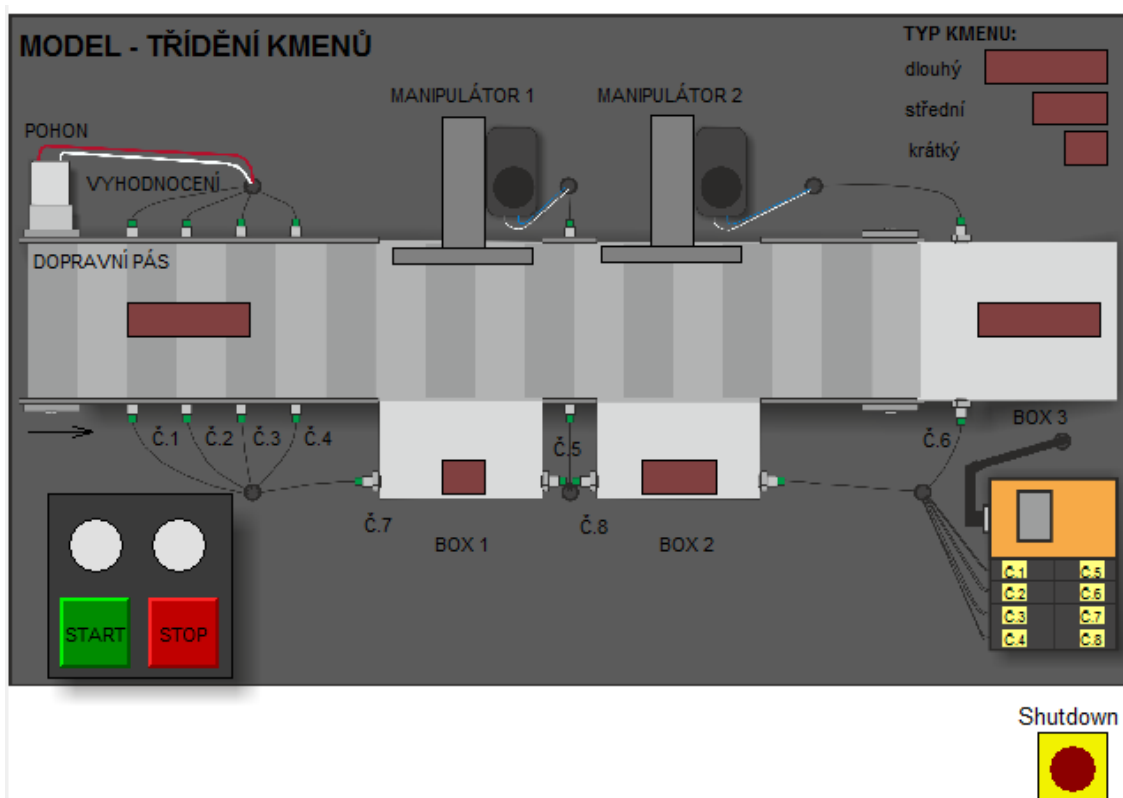
U LD diagramu se všechny příčky vykonávají souběžně, proto jednoduše přiřadíme tlačítkům STOP a START proměnné, které chceme měnit. Pouze je nutné ošetřit, aby nedošlo ke konfliktu programu na jiných příčkách, kde jsou použity stejné proměnné. Stejně tak je tomu i u diagramu FB.

U SFC diagramu program postupuje jako u vývojového diagramu. Proto by bylo nutné větvením v každém kroku vrátit proces do výchozího stavu nebo pomocí funkce STOP vyskočit z cyklu. V obou případech by bylo nutné použít větvení. Takový program by byl zbytečně složitý a nepřehledný. Naštěstí programovací prostředí RSLogix 5000 umožňuje komunikaci mezi více rutinami a dokonce i jedná-li se o různé programovací jazyky. Pro SFC diagram je tedy vytvořen Task, ve kterém je nadefinováno, aby běžel souběžně s programem SFC. Pomocí rutiny v LD diagramu jsou pak nastaveny proměnné pro tlačítka START a STOP.

5 NÁVRH VIZUALIZACE

Vizualizace byla vytvořena pomocí prostředí FactoryTalk View Studio a grafického prostředí CorelDraw. Animace pohybu je vyřešena pomocí LD diagramu v rutině FTWS. Vizualizace obou modelů jsou doplněny o tlačítko Shutdown, které slouží k ukončení vizualizace z online režimu.

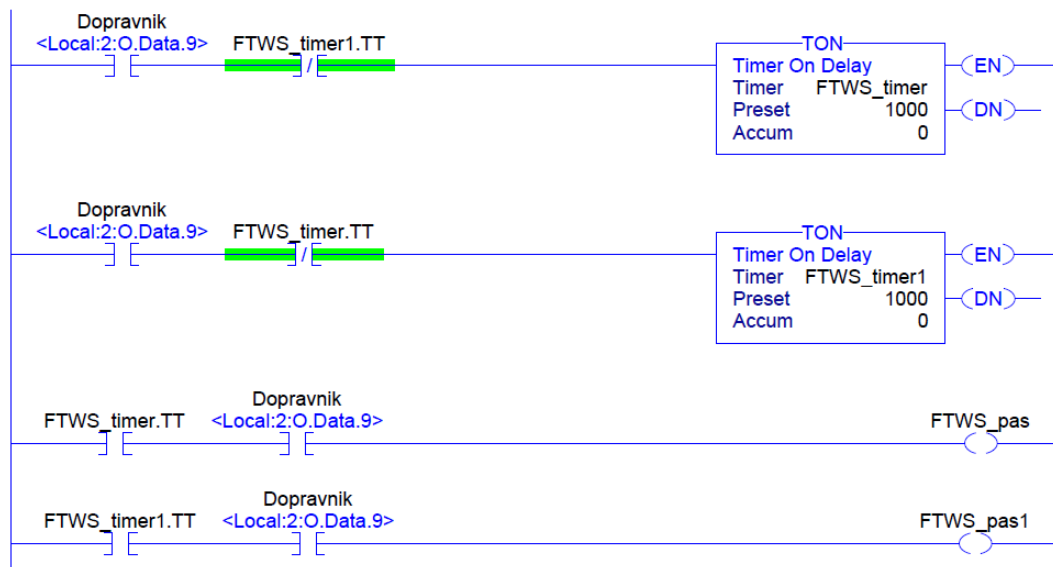
5.1 Návrh vizualizace pro model Třídění kmenů



Obrázek 25: Vizualizace modelu

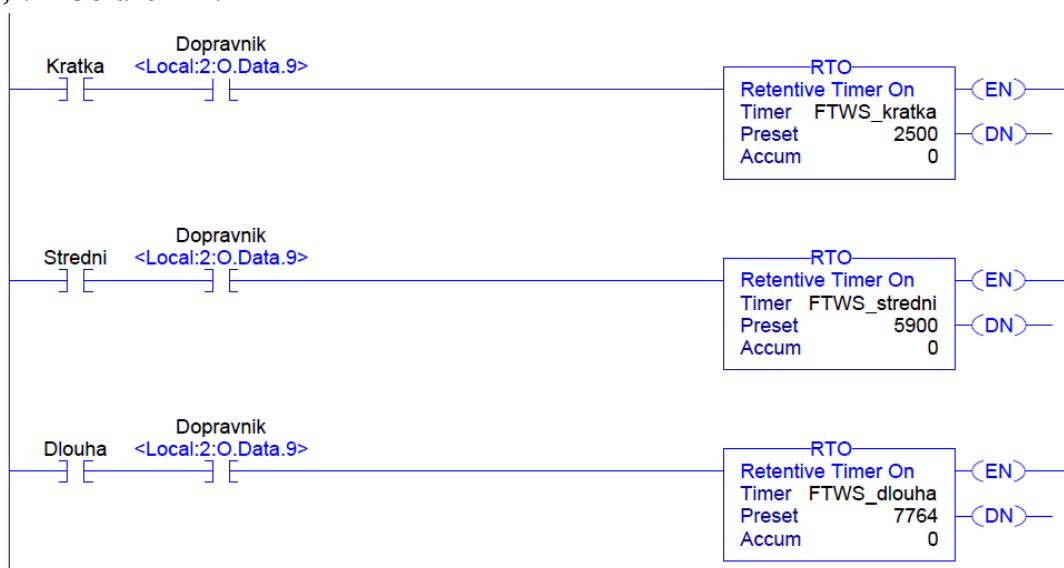
Grafický model Třídění kmenů je zobrazen na Obrázek 25. Je tvořen tlačítky START a STOP, která spínají pomocné proměnné v programu vytvořeném pro tuto úlohu v LD diagramu. Tato tlačítka tedy slouží pro sepnutí a vypnutí programu. Dále mění barvy LED diod znázorněných ve vizualizaci nad tlačítky, dle zadání, podle stavu, ve kterém se nachází.

V RSLogix5000 je vytvořena rutina v LD diagramu, která slouží pro animaci pohybu, viz Obrázek 26. Tím, že se uvede dopravník do provozu se zároveň střídavě spouští časovače *FTWS_timer1* a *FTWS_timer2*. Ty také střídavě spouští proměnné *FTWS_pas* a *FTWS_pas1*. Tyto proměnné střídavě zviditelňují části dopravního pásu ve vizualizaci a animují tak jeho pohyb.



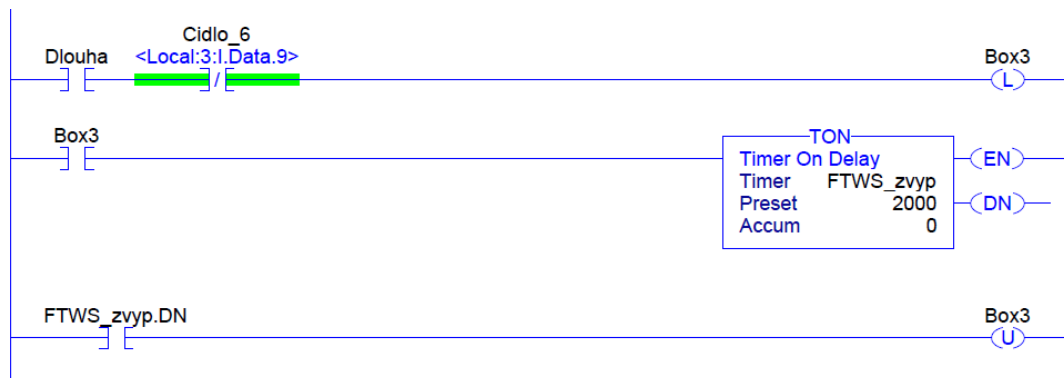
Obrázek 26: FTWS: animace pohybu dopravníku

Krátce poté, co čídl 3 zareaguje na kmen dopravníku, program vyhodnotí stav, o jaký typ kmenu se jedná (dlouhý, střední, krátký). Podle příslušné proměnné se zobrazí ve vizualizaci kmen příslušné velikosti. Sepne se časovač, který animuje pohyb kmenu, viz Obrázek 27.



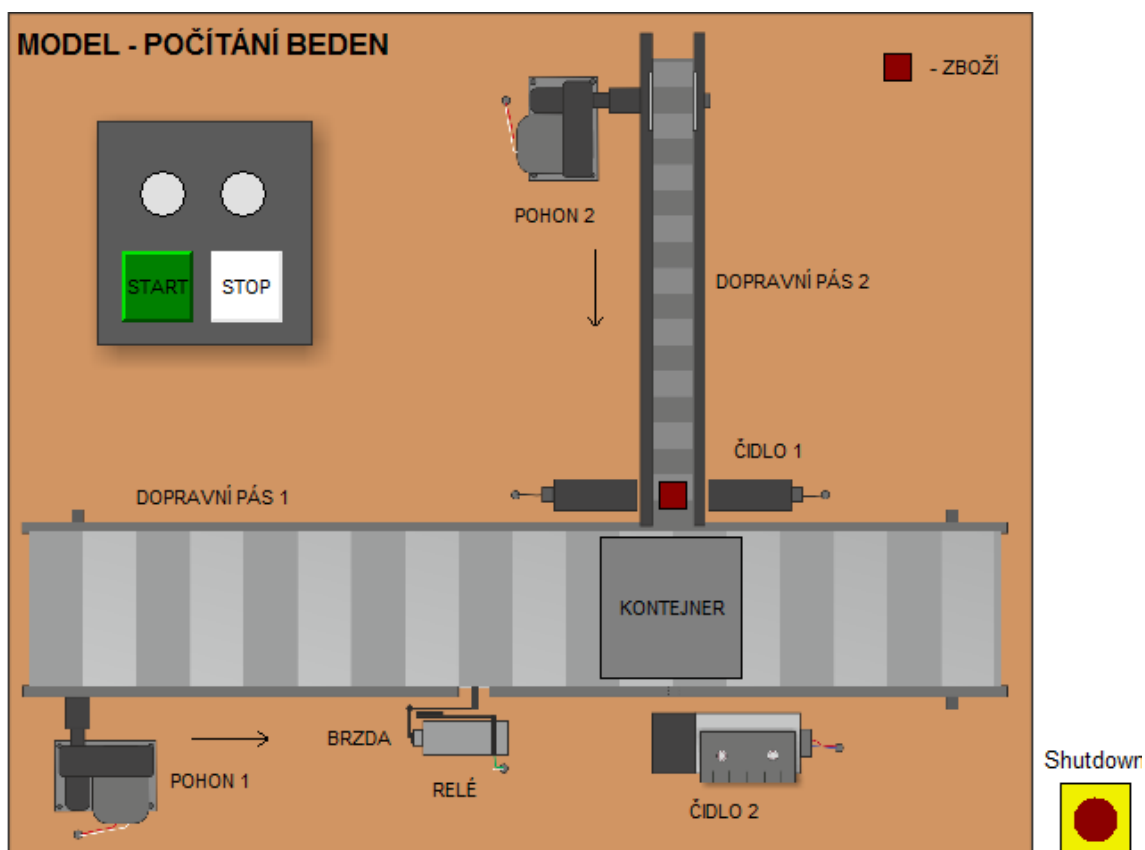
Obrázek 27: FTWS: animace pohybu kmenů

Kmen dlouhý putuje až do boxu 3, kde je ještě po dobu 2 sekund zobrazen než časovač *FTWS_zvyp* dočasuje, viz Obrázek 28. Kmen střední a krátký se zastaví před příslušným manipulátorem. V tuto dobu se sepne proměnná *Rameno1* či *Rameno2*, která ve vizualizaci skryje manipulátor v původní pozici a zobrazí manipulátor v pozici na páse. Animuje tak pohyb. Zároveň je také zviditelněn kmen v boxu, a to vše po dobu 2 sekund, kdy je rameno aktivní. Na Obrázek 25 je také znázorněn vyhodnocovací modul čidel tak, jak je tomu u skutečného modelu. Vyhodnocovací model nás tak informuje o stavu všech čidel. Toho lze využít pro detekci poruchy.



Obrázek 28: FTWS: dlouhý kmen v boxu 3

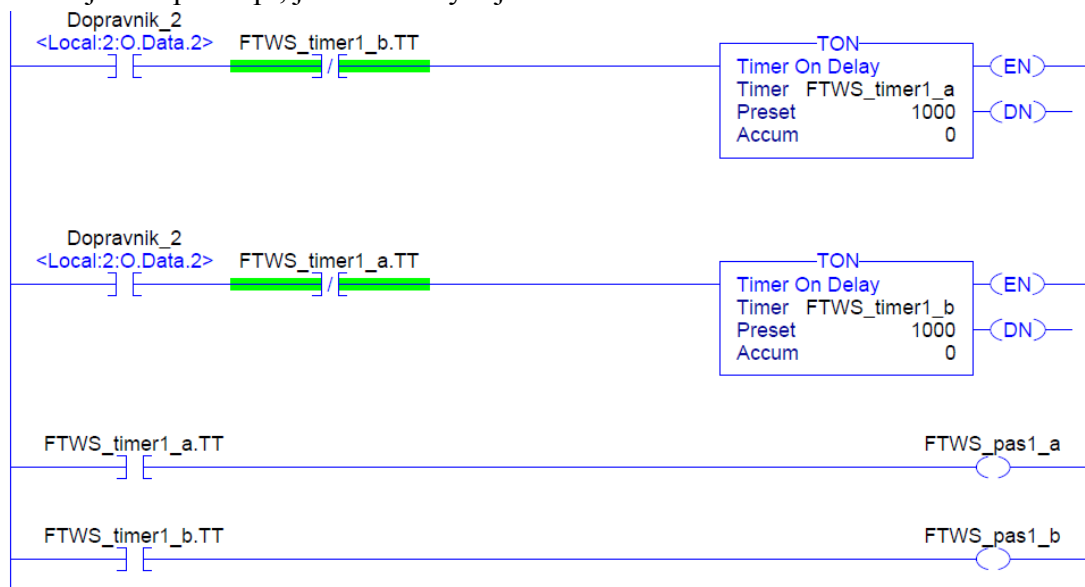
5.2 Návrh vizualizace pro model Počítání beden



Obrázek 29: Vizualizace modelu

Grafický model Počítání beden je zobrazen na Obrázek 29. Pro tento model je návrh poněkud jednodušší, jelikož model obsahuje pouze dvě čidla. Ta nás pouze informují o tom, že je kontejner před dopravníkem 1 nebo, že zboží padá do kontejneru. Tlačítka start a stop mění barvy LED diod nad tlačítka podle stavu, ve kterém se nachází. Je-li v programu aktivována brzda, změní se pozice brzdy ve vizualizaci. Pro animaci

pohybu obou dopravníků jsou opět použity dva časovače a dvě pomocné proměnné, které se vzájemně překlápí, jako tomu bylo již u modelu Třídění kmenů viz Obrázek 30



Obrázek 30: FTWS: animace dopravníku

ZÁVĚR

V první části této práce je popsán systém CompactLogix, programovací prostředí RSLogix 5000, komunikační prostředí RSlinx Classic a software pro vizualizaci FactoryTalk View Studio.

Druhá část práce se zaměřuje na modely Třídění kmenů a Počítání beden. Je zde uveden popis a úlohy pro jednotlivé modely. Následně je uvedeno, jak je řešeno připojení obou modelů k PLC pomocí konektoru.

Třetí část se zabývá programovacími jazyky Ladder diagramem, diagramem funkčních bloků a sekvenčně funkčními grafy. Jsou nastíněny základní struktury a programovací prvky. Na ukázkách programů jsou předvedeny základní rozdíly a nedostatky. Úplné programy pro oba modely jsou uvedeny v příloze.

Ve čtvrté části je zobrazena a popsána vizualizace obou modelů vytvořená v prostředí FactoryTalk View Studio, dále také programy použité pro animaci modelů vytvořené v Ladder diagramu. Kompletní programy jsou opět uvedeny v příloze.

Všechny programy byly prozkoušeny na laboratorních úlohách. Programovací jazyky se lišily především strukturou.

Strukturovací jazyk, který byl součástí sekvenčně funkčního grafu, se jeví jako nejsložitější. Svojí strukturou připomíná čistě zdrojový kód a vyžaduje zapsání všech vlastností tvořené funkce. V praxi se také nejčastěji využívá pouze k manipulaci s řetězcí ve formátu ASCII, ke zpracování protokolů nebo matematických operací. Sekvenčně funkční graf je proto programem nejméně vhodným. Z použití v úlohách vyplývá, že program obsahuje minimum funkcí a musí být z velké části tvořen právě strukturovaným textem, či dokonce doplněn jiným programovacím jazykem. Sekvenčním vykonáváním jednotlivých kroků a strukturou programu se stává použitelným pouze pro úzké pásmo zařízení. Nejčastěji bývá využíván pro opakované sekvence operací, dávkovací procesy, stavové operace strojních zařízení a řízení pohybu s využitím strukturovaného textu.

Diagram funkčních bloků oproti ostatním programovacím jazykům obsahuje širokou škálu funkcí, což při aplikaci na úlohu velice usnadnilo programování. Odbornost a rozsáhlost tohoto programovacího jazyka tak umožňuje řízení komplikovanějších procesů. Specializuje se na souvislé procesy a řízení pohonů, řízení ve smyčce a cyklické výpočty. V těchto úlohách však tento programovací jazyk nenachází své využití.

Ladder diagram se pro tyto úlohy svou jednoduchostí a přehledností jeví jako nejvhodnější. Obsahuje jednoduché schéma a programovací prvky. V praxi napodobuje reléové schéma, proto se využívá zejména právě pro operace, u kterých existuje možnost, že je bude muset servisní personál nebo personál údržby interpretovat, aby mohl u strojního zařízení nebo procesu vyhledávat a odstraňovat závady. Při řízení složitějších procesů by bylo ovšem vhodnější použít programovací jazyk diagram funkčních bloků.

Literatura

- [1] Publikace Rockwell Automation 1769-SG001N-CS-P [online]. ©2012 [cit. 2013-05-22]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/sg/1769-sg001_-cs-p.pdf.
- [2] Copyright ©2009 Rockwell Automation [online]. ©2009 [cit. 2013-05-22]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/pp/ftalk-pp013_-cs-p.pdf.
- [3] Publikace 1756-PM008B-CS-P [online]. ©2006 [cit. 2013-05-22]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm008_-cs-p.pdf.
- [4] Publikace 1756-PM009B-CS-P [online]. ©2008 [cit. 2013-05-22]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm009_-cs-p.pdf.
- [5] Publication 1756-PM006E-EN-P [online]. ©2012 [cit. 2013-05-22]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm006_-en-p.pdf.
- [6] Publication 1756-PM006E-EN-P [online]. ©2012 [cit. 2013-05-22]. Dostupné z: http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm006_-en-p.pdf.

SEZNAM PŘÍLOH

Příloha 1. CD:

- Bakalářská práce v elektronické podobě
- Programy z RSLogix 5000
- Reporty z RSLogix 5000
- Programy z FactoryTalk View Studio
- Reporty z FactoryTalk View Studio