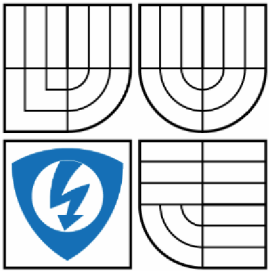


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A  
KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY  
FACULTY OF ELECTRICAL ENGINEERING AND  
COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# SIMULÁTORY LABORATORNÍCH MODELŮ KULIČKY A KMENY V PROSTŘEDÍ CONTROL WEB

SIMULATORS OF LABORATORY MODELS BALLS AND LOGS IN CONTROL WEB

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Petr Chalupa

**VEDOUČÍ PRÁCE**

SUPERVISOR

Ing. Miroslav Jirgl

BRNO 2016

## **Abstrakt**

Cílem této bakalářské práce je vytvoření programů schopných simulovat a vizualizovat dění na laboratorních modelech třídičky kmenů a dávkování kuliček, na kterých probíhá výuka programovatelných automatů.

V první části práce se zabývám programem Control Web. V další části popíši oba modely, jak jejich fyzické zapojení se schémata, tak i požadavky na funkce modelů. V následující části se zabývám popisem tvorby a fungování mnou vytvořených programů. V poslední části se zabývám propojením programů s PLC pomocí sběrnice MODBUS.

## **Klíčová slova**

Control Web, Modbus, TIA Portal, simulace, vizualizace

## **Abstract**

The aim of this bachelor's thesis are creation of two program, able to simulate and visualize actions on laboratory models sorter logs and dosing balls. These models are used to learning PLC programming.

In first part is deal with program Control Web. The next part describes both of models, their schema and functions. The following part describe creation and work of program, which I made. The last part of this work is deal with connection both programs with PLC by using MODBUS.

## **Keywords**

Control Web, Modbus, TIA Portal, simulation, visualization

## **Bibliografická citace:**

CHALUPA, P. *Simulátory laboratorních modelů kuličky a kmeny v prostředí Control Web*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 42s. Vedoucí bakalářské práce byl Ing. Miroslav Jirgl.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Simulátory laboratorních modelů kuličky a kmeny v prostředí Control Web 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: **23. května 2016**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu bakalářské práce Ing. Miroslavu Jirglovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **23. května 2016**

.....  
podpis autora

# OBSAH

Úvod.....	11
1 Control Web 5.....	12
1.1 Popis prostředí .....	12
1.1.1 Vzhled .....	13
1.1.2 Časování .....	13
1.1.3 Karty editorů .....	14
1.1.4 Paleta přístrojů .....	14
1.2 Založení nového projektu .....	15
2 Popis modelu Třidička kmenů .....	15
2.1 Popis funkce.....	16
2.1.1 Vývojový diagram.....	16
2.2 Tabulka vstupů a výstupů .....	18
2.3 Elektrické zapojení modelu .....	18
2.3.1 Propojení s PLC .....	19
2.3.2 Schéma zapojení desky .....	20
2.3.3 Ovládání lopatek .....	22
3 Popis modelu Dávkování kuliček.....	23
3.1 Popis funkce.....	23
3.1.1 Vývojový diagram.....	24
3.2 Tabulka vstupů a výstupů .....	26
3.3 Schéma zapojení .....	27
4 Popis projektu Třidička Kmenů .....	28
4.1 Popis programu .....	28
4.2 Popis přístrojů.....	29
4.3 Režim simulace.....	30
4.3.1 Pohybu kmenu po dopravníku .....	31
4.3.2 Přemístění kmenu do zásobníku.....	32
4.4 Režim vizualizace .....	33
5 Popis projektu Dávkování kuliček .....	34
5.1 Popis vizualizace.....	34
5.2 Popis přístrojů.....	34
5.3 Režim simulace.....	35
5.3.1 Plnění válců.....	35
5.3.2 Vypouštění kuliček.....	36
5.4 Režim vizualizace .....	38
5.4.1 Čtení z BCD číselníků.....	38
6 Propojení s PLC .....	39
6.1 Protokol MODBUS.....	39
6.2 Konfigurace v Control Webu.....	39
6.3 Propojení s TIA portálem .....	40

7	Závěr .....	41
	Použitá literatura .....	<b>Chyba! Záložka není definována.</b>

## Seznam obrázků

Obrázek 1: Control Web 5-Prostředí aplikace .....	12
Obrázek 2: Panel Vzhled.....	13
Obrázek 3: Panel Časování .....	13
Obrázek 4: Karty editorů.....	14
Obrázek 5: Vyvolání palety přístrojů .....	14
Obrázek 6: Paleta přístrojů.....	14
Obrázek 7: Nový projekt.....	15
Obrázek 8: Třídička kmenů-model .....	15
Obrázek 9: Třídička kmenů-diagram .....	16
Obrázek 10: Vývojový diagram modelu třídičky kmenů-Stop .....	16
Obrázek 11: Vývojový diagram modelu třídičky kmenů-Proces třídění .....	17
Obrázek 12: Univerzální řídicí deska.....	18
Obrázek 13: Konektor Canon: .....	19
Obrázek 14: Řídicí deska-hladina TOP.....	20
Obrázek 15: Řídicí deska-hladina BOTTOM .....	20
Obrázek 16: Schéma řídicí desky.....	21
Obrázek 17: Deska pro ovládání lopatek .....	22
Obrázek 18: Schéma zapojení desky.....	22
Obrázek 19: Dávkování kuliček-model .....	23
Obrázek 20: Dávkování kuliček-diagram .....	24
Obrázek 21: Vývojový diagram modelu dávkování kuliček-Proces vyprázdění .....	24
Obrázek 22: Vývojový diagram modelu dávkování kuliček-Proces dávkování .....	25
Obrázek 23: Dávkování kuliček-schéma zapojení .....	27
Obrázek 24: Třídička kmenů-vizualizace .....	28
Obrázek 25: Třídička kmenů-panel vzhledu .....	29
Obrázek 26: Panel ovládání simulace .....	31
Obrázek 27: Ukázka programu třídičky kmenů-pohyb po dopravníku .....	31
Obrázek 28: Ukázka programu třídičky kmenů-konec .....	32
Obrázek 29: Dávkování kuliček.....	34
Obrázek 30: Ovládání simulace kuliček .....	35
Obrázek 31: Dávkování kuliček-příkaz visibility .....	35
Obrázek 32: Ukázka programu pro plnění válců .....	36
Obrázek 33: Ukázka programu pro dávkování kuliček-horní západka.....	37
Obrázek 34: Ukázka programu pro dávkování kuliček-dolní západka .....	37
Obrázek 35: Dávkování kuliček-převod z BCD .....	38
Obrázek 36: Protokol MODBUS .....	39
Obrázek 37: Ovladač MODBUS.....	39
Obrázek 38: Komunikační kanály.....	40
Obrázek 39: Blok pro komunikaci v TIA Portal.....	40

## Seznam tabulek

Tabulka 1: Třídička kmenů-seznam vstupů .....	18
Tabulka 2: Třídička kmenů-seznam výstupů .....	18



Tabulka 3: Seznam vstupů připojených na konektor .....	19
Tabulka 4: Seznam výstupů připojených na konektor .....	19
Tabulka 5: Dávkování kuliček-seznam výstupů .....	26
Tabulka 6: Dávkování kuliček-seznam vstupů .....	26

# ÚVOD

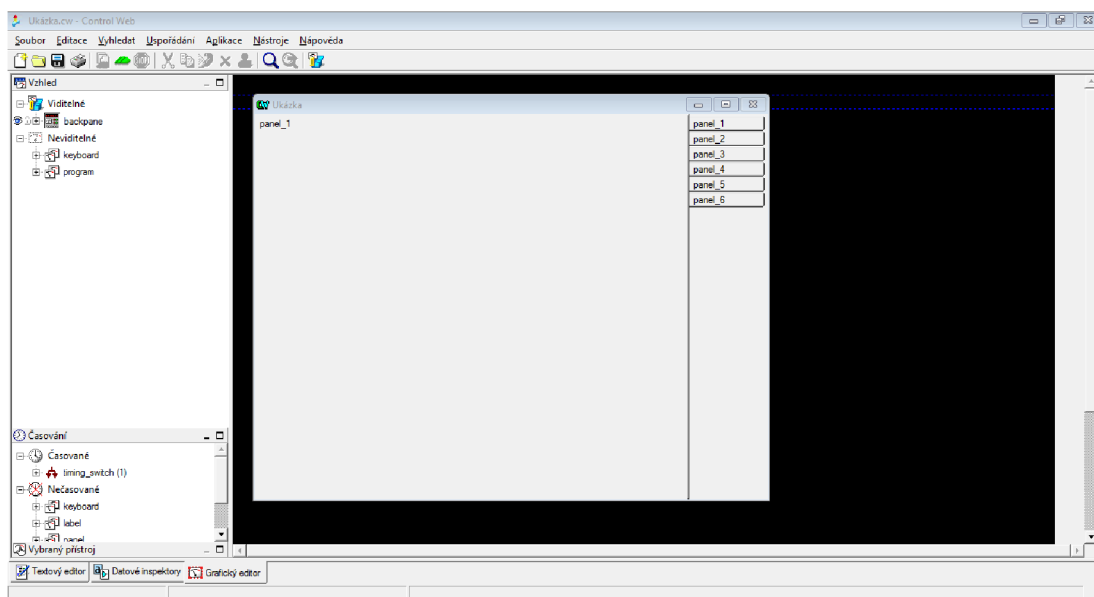
Laboratorní modely dávkování kuliček a třídičky kmenů, jsou využívány pro výuku programovatelných automatů. Oba modely obsahují modulární PLC od firmy Siemens Simatic S7-300, které je připojeno k počítači pomocí sběrnice MODBUS. Na počítači je nainstalován software TIA Portal v13 a Control Web 5. Softwarový nástroj Control Web 5 použiji pro tvorbu vizualizace a simulace obou modelů. Tento software obsahuje ovladač pro sběrnici MODBUS, který využiji pro zprovoznění komunikace mezi programy a PLC. Pro programování PLC využiji program TIA Portal v13, ve kterém vytvořím projekt, který bude schopný komunikovat se sběrnici MODBUS a bude obsahovat všechny důležité proměnné. Modely obsahují běžně užívané prvky v průmyslové automatizaci, jako jsou tlačítka, kontrolky, snímače, dopravníky, manipulátory. Modely mají reálnou předlohu z průmyslu.

# 1 CONTROL WEB 5

Control Web je sofistikovaný programový nástroj, který ve všech ohledech nahrazuje SCADA systémy a dokáže rozšířit schopnosti řídicího systému. Jak sám výrobce píše, je těžké definovat jeho možnosti použití, protože díky širokým možnostem tohoto softwaru, se jeho použití posouvá daleko za pouhý SCADA systém. Jednoduše nám dokáže posloužit tak, jak zrovna potřebujeme. Velkou výhodou představuje pro malé vestavěné systémy, které díky integraci Control Webu, získají více komunikačních kanálů bez nutnosti dokupovat hardware. Tímto krokem získá řídicí systém rozhraní pro Wi-Fi, Bluetooth, USB, tedy obvyklé vybavení dnešního PC. Control Web je také vybaven virtuálním HTTP serverem, díky kterému je možné realizovat bezobslužný běh. [1]

## 1.1 Popis prostředí

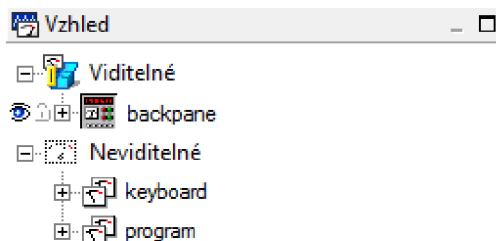
Tato verze programu byla uvedena na trh v roce 2010 a tomu odpovídá i grafické prostředí. V porovnání s dnešními moderními aplikacemi působí toto prostředí poměrně zastarale. Ovšem co tento program ztrácí na kráse, nalézá ve funkčnosti. Díky jeho jednoduchému prostředí, je pro začátečníka jednoduché, se v něm během chvíle zorientovat. Základní prostředí aplikace je na obrázku 1.



Obrázek 1: Control Web 5-Prostředí aplikace

### 1.1.1 Vzhled

V levé horní části programu se nachází karta *Vzhled*. Do této karty se vkládají přístroje, které chce uživatel použít v aplikaci. Karta je rozdělena na dvě sekce *Viditelné* a *Neviditelné*, viz obrázek 2



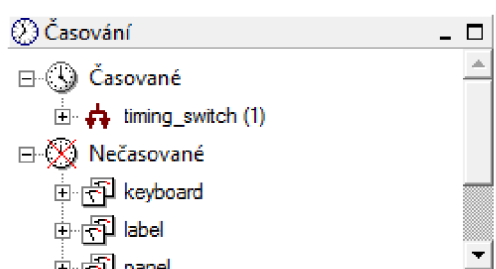
Obrázek 2: Panel *Vzhled*

Jak již název *Viditelné* napovídá, jedná se o přístroje, které budou zobrazeny na panelu a budou na něm plnit určitou funkci. Velkou výhodou programu je práce s vrstvami, kdy je možné „vrstvit“ objekty na sebe a docílit tak různých efektů. Příkladem může být vytvoření vlastního designu tlačítka, kde lze velice výhodně využít funkci vrstvení pro docílení efektu stisknutí.

Sekce *Neviditelné* slouží ke vložení objektů, které běží na pozadí aplikace a nejsou vidět. Nejčastěji se do této sekce vkládá přístroj *Program*. Prostřednictvím přístroje *Program* je možné naprogramovat, vše od jednoduchých aplikací, až po komplexní funkce.

### 1.1.2 Časování

Hned pod panelem vzhledu se nachází panel *Časování*, viz obrázek 3.

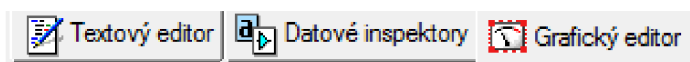


Obrázek 3: Panel *Časování*

Tento panel zobrazuje informace o obnovování přístrojů. Obnovování přístrojů probíhá cyklicky a je na uživateli, jakou periodu obnovování nastaví nebo jestli se objekt má vůbec obnovovat. Panel *Časování* shromažďuje všechny tyto informace. Nastavení časování bývá častou příčinou špatně fungující aplikace, kdy uživatel buďto zapomněl nebo špatně zvolil velikost periody obnovování. Díky funkci časování, lze dosáhnout různých efektů v aplikaci. Od blikající kontrolky, až po téměř spojitě pohybující se objekt.

### 1.1.3 Karty editorů

Pod panelem *Časování* se nachází tři záložky, viz obrázek 4.



Obrázek 4: Karty editorů

Pod každou z nich se nachází různý editor. Po spuštění programu se nacházíme v *Grafickém editoru*. V tomto editoru můžeme pracovat na panelu a rovnou vidět jeho vzhled.

Záložka *Datové inspektory* slouží k editaci proměnných.

Poslední záložka *Textový editor*, slouží také ke tvorbě panelů. Je zde možné jednoduše a rychle provést mnoho úkonů, ovšem tato práce vyžaduje zkušenějšího uživatele.

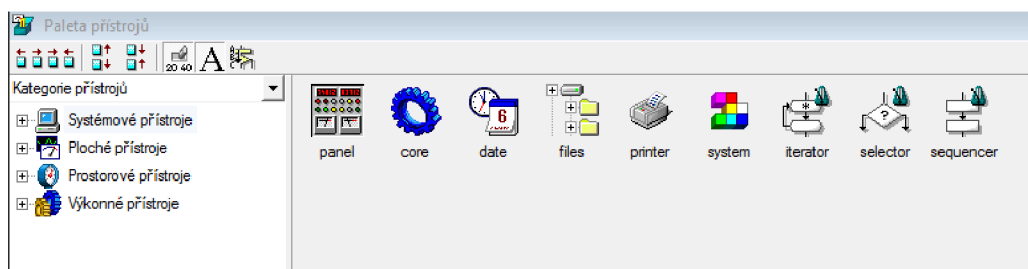
### 1.1.4 Paleta přístrojů

Paleta přístrojů je seznam všech přístrojů, které Control Web poskytuje. Tuto nabídku spustíme stiskem tlačítka v panelu nástrojů. Toto tlačítko je žlutě zvýrazněno na následujícím obrázku.



Obrázek 5: Vyvolání palety přístrojů

Jak již bylo zmíněno, *Paleta přístrojů* obsahuje seznam všech dostupných přístrojů. Paleta je rozdělena na čtyři kategorie: *Systémové přístroje*, *Ploché přístroje*, *Prostorové přístroje*, *Výkonové přístroje*, viz obrázek 6. Každá z těchto kategorií obsahuje specifické přístroje. V následujícím textu bude stručně nastíněn obsah každé z kategorií a jejich důležitost pro běžného uživatele.



Obrázek 6: Paleta přístrojů

Nejpoužívanější přístroje se nachází v kategorii *Ploché přístroje*. Zde je možné nalézt přístroje jako například tlačítka, zobrazovače, bloky schopné vykreslit graf, přepínače, popisky, základní geometrické tvary a další obyčejné přístroje.

Další zajímavou kategorií jsou *Prostorové přístroje*. V této kategorii se nachází 3D objekty, od efektů kouře, až po prostorový text.

V kategorii *Výkonové přístroje* se nachází neviditelné přístroje, pracující na pozadí aplikace. Patří mezi ně například: přístroj na zpracování povelů z klávesnice, přístroj, do

kterého můžeme zapsat program, přístroj obsluhující reproduktor nebo i virtuální HTTP server a další.

Poslední kategorií jsou Systémové přístroje. Nejpoužívanějším přístrojem z této kategorie je pro uživatele přístroj *Panel*, který je základem každé aplikace. Jako další jsou zde přístroje pro časování aplikace, ovládání tiskárny a další.

## 1.2 Založení nového projektu

Založení nového projektu provedeme kliknutím na tlačítko zvýrazněné na následujícím obrázku v panelu nástrojů.

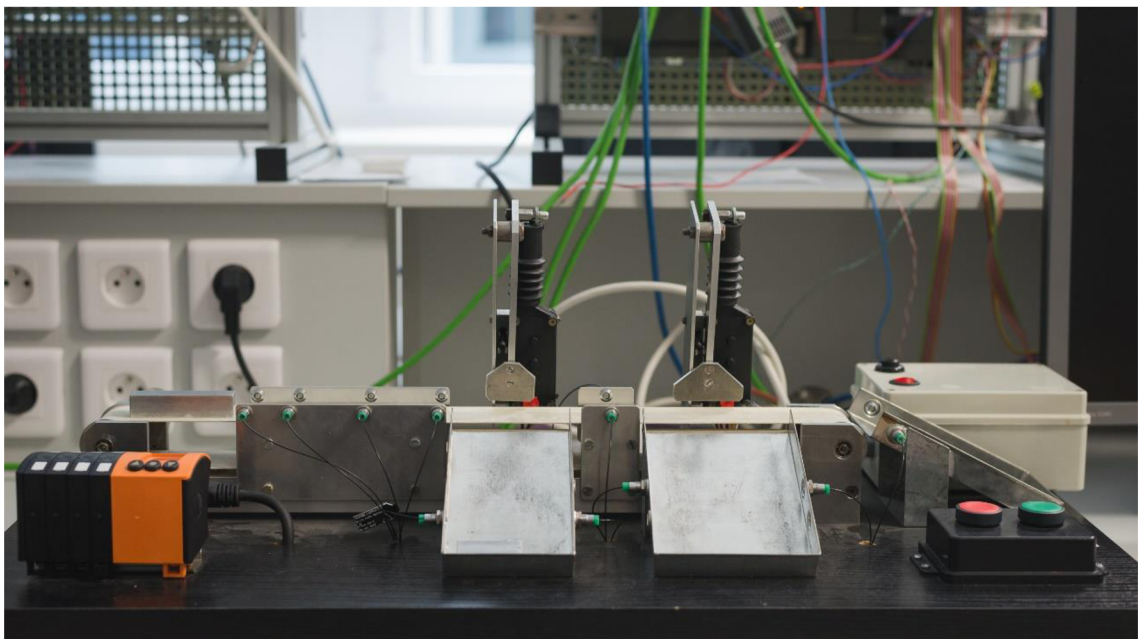


Obrázek 7: Nový projekt

Dojde k vyvolání nabídky, kde je třeba vybrat položku *Vytvořit novou aplikaci*. Poté se spustí *Průvodce novou aplikací*, kde je na následujících devíti obrazovkách třeba zvolit požadované parametry. Volíme parametry rozměrů, prvotní rozložení panelu, způsob časování nebo i dovolené zpoždění čtení dat z ovladače.

## 2 POPIS MODELU TŘÍDIČKA KMENŮ

Tento laboratorní model, viz obrázek 8, ukazuje linku na třídění kmenů. Třídění kmenů probíhá podle velikosti. Tato průmyslová linka je používána v dřevařské technologii. Studenti si na tomto modelu mohou vyzkoušet práci s dopravníkem, manipulátory, snímači a časovači. Tato úloha je vhodná na logické řízení, tudíž se dá použít jazyk STL a LAD.

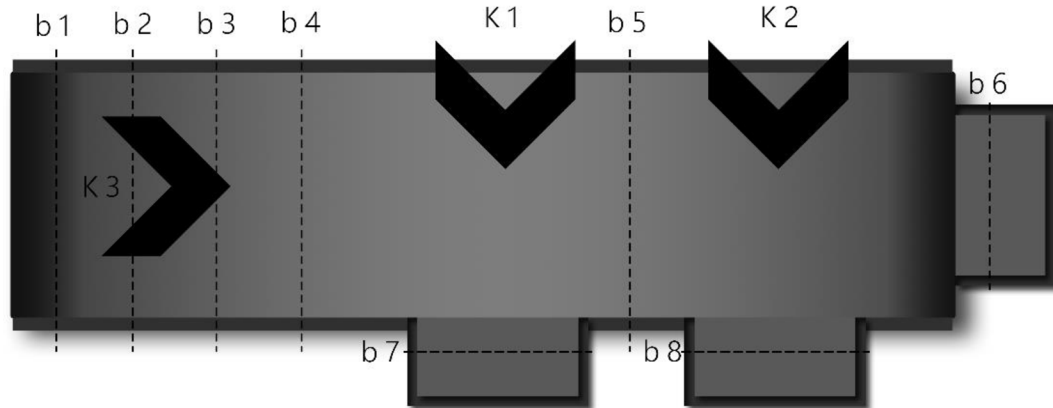


Obrázek 8: Třídíčka kmenů-model

## 2.1 Popis funkce

Kmeny přijíždějí po dopravníku a podle jejich velikosti jsou umístěny do správných zásobníků. Pomocí soustavy čidel  $b1$ ,  $b2$ ,  $b3$  a  $b4$  model rozpozná velikost kmenu. Malé kmeny patří do zásobníku 1, kam jsou z dopravníku přemístěny pomocí manipulátoru  $K1$ . Střední kmeny se umísťují do zásobníku 2, kam se z dopravníku přemístí pomocí manipulátoru  $K2$ . Velké kmeny se umísťují do zásobníku 3, kam se nechají dojet po dopravníku. Je důležité znát počet kmenů v zásobnících, pro pozdější použití.

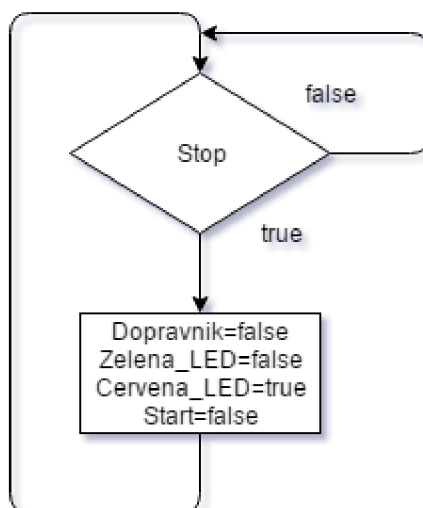
Model se spouští po stisku tlačítka *Start* a zastavuje v jakémkoli okamžiku po stisku tlačítka *Stop*. Model obsahuje i zelenou a červenou kontrolku, které ukazují stav modelu.



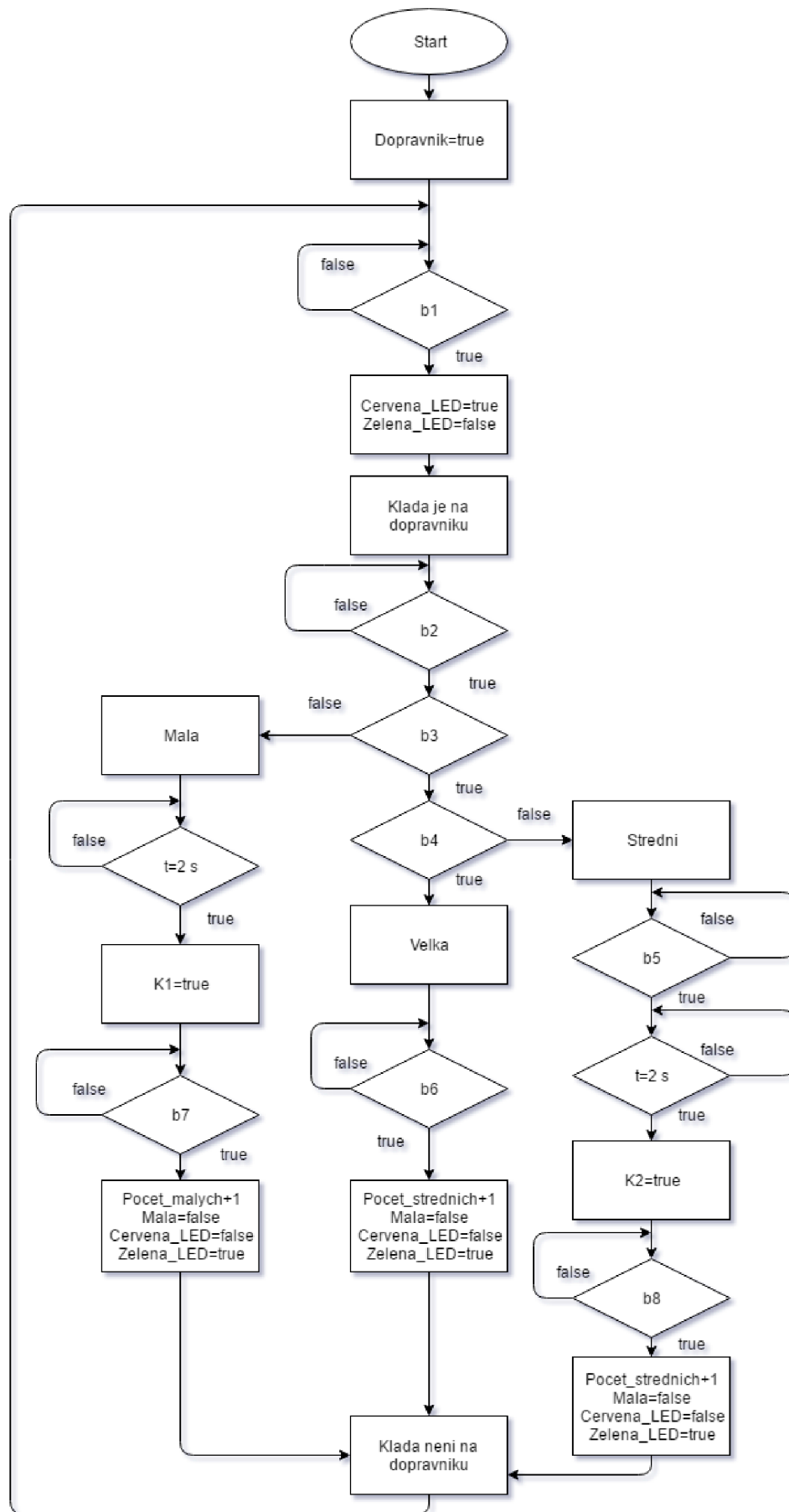
Obrázek 9: Třídíčka kmenů-diagram

### 2.1.1 Vývojový diagram

Vývojové diagramy funkce modelu jsou na následujících obrázcích. Na prvním vývojovém diagramu je proces zastavení, který má nejvyšší prioritu.



Obrázek 10: Vývojový diagram modelu třídíčky kmenů-Stop



Obrázek 11: Vývojový diagram modelu třídičky kmenů-Proces třídění



## 2.2 Tabulka vstupů a výstupů

Tabulku vstupů a výstupů jsem převzal z dokumentu [2].

*Tabulka 1: Třídíčka kmenů-seznam vstupů*

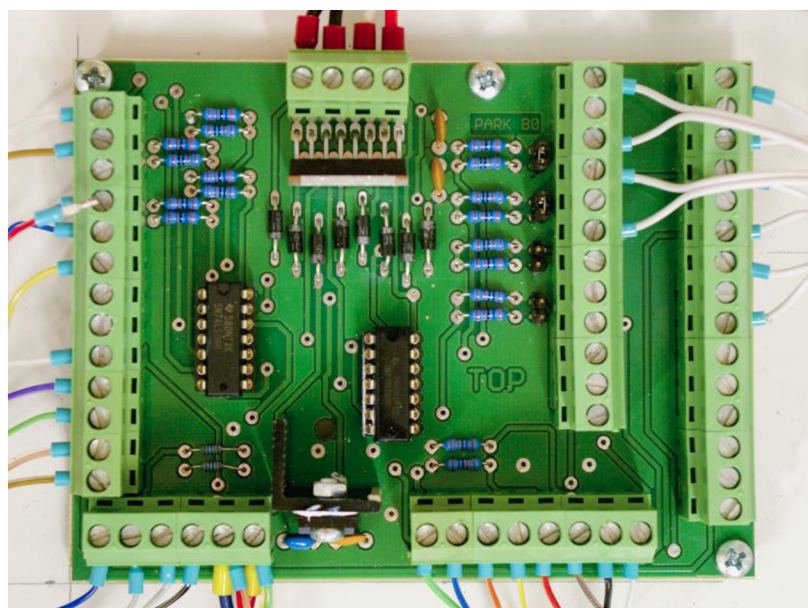
Adresa	Popis
I 24.0	Zelené tlačítko START
I 24.1	Červené tlačítko STOP
I 24.2	Snímač b2
I 24.3	Snímač b5
I 24.4	Snímač b6
I 24.5	Snímač b8
I 24.6	Snímač b3
I 24.7	Snímač b4
I 25.0	Snímač b1
I 25.1	Snímač b7

*Tabulka 2: Třídíčka kmenů-seznam výstupů*

Q 40.0	Zelená LED
Q 40.1	Červená LED
Q 40.2	Dopravník
Q 40.3	Lopatka 2
Q 40.4	Lopatka 1

## 2.3 Elektrické zapojení modelu

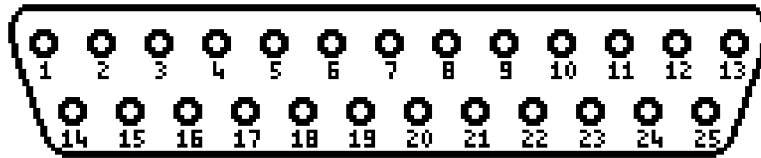
Model obsahuje univerzální řídicí desku, viz obrázek 10, která byla navržena pro řízení laboratorních modelů v laboratoři programovatelných automatů.



*Obrázek 12: Univerzální řídicí deska*

### 2.3.1 Propojení s PLC

Propojení modelu s PLC je realizováno pomocí 25-pinového Canon konektoru, na jehož piny jsou připojeny jednotlivé vstupy a výstupy modelu. Seznam jednotlivých vstupů a výstupů s číslem pinu je v tabulce 3 a 4.



Obrázek 13: Konektor Canon:

Tabulka 3: Seznam vstupů připojených na konektor

Adresa	Popis	Pin
I 24.0	Start	1
I 24.1	Stop	14
I 24.2	b2	2
I 24.3	b5	15
I 24.4	b6	3
I 24.5	b8	16
I 24.6	b3	4
I 24.7	b4	17
I 25.0	b1	5
I 25.1	b7	18

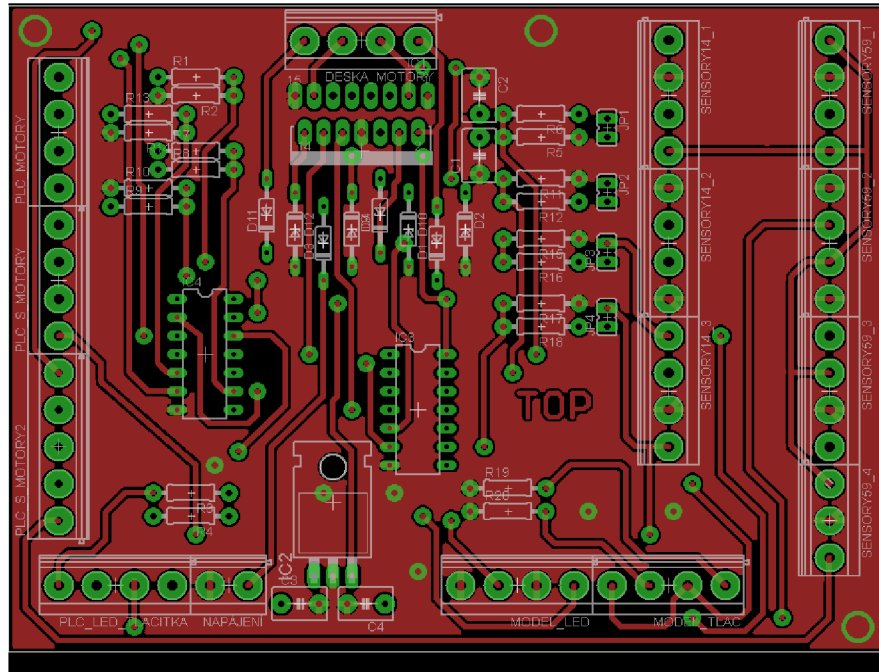
Tabulka 4: Seznam výstupů připojených na konektor

Q 40.0	Zelená LED	11
Q 40.1	Červená LED	23
Q 40.2	Dopravník	10
Q 40.3	Lopatka 2	22
Q 40.4	Lopatka 1	9

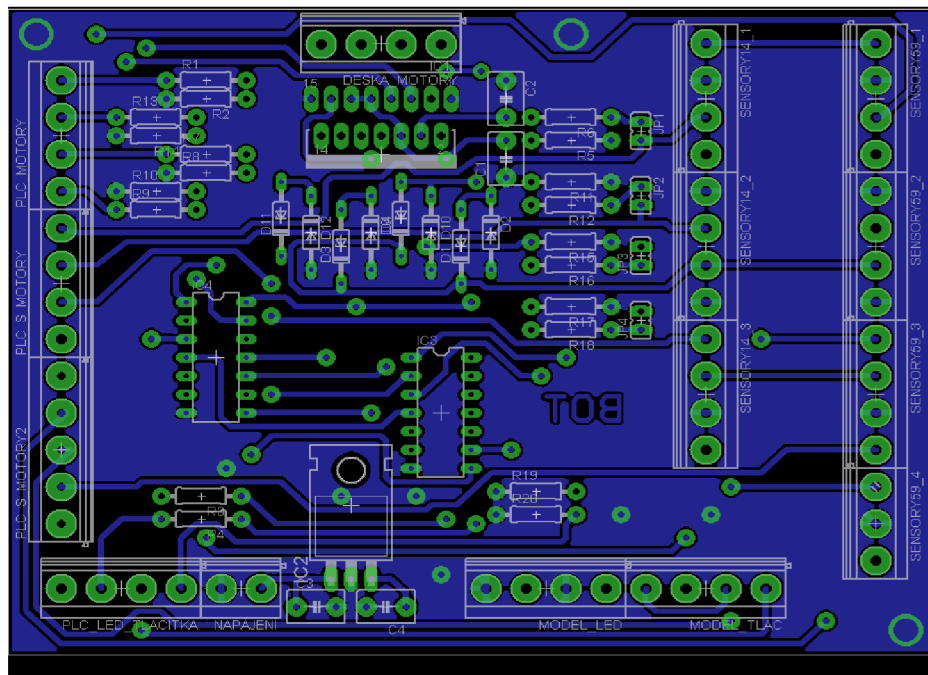
Zem je připojená na piny 12 a 24.

### 2.3.2 Schéma zapojení desky

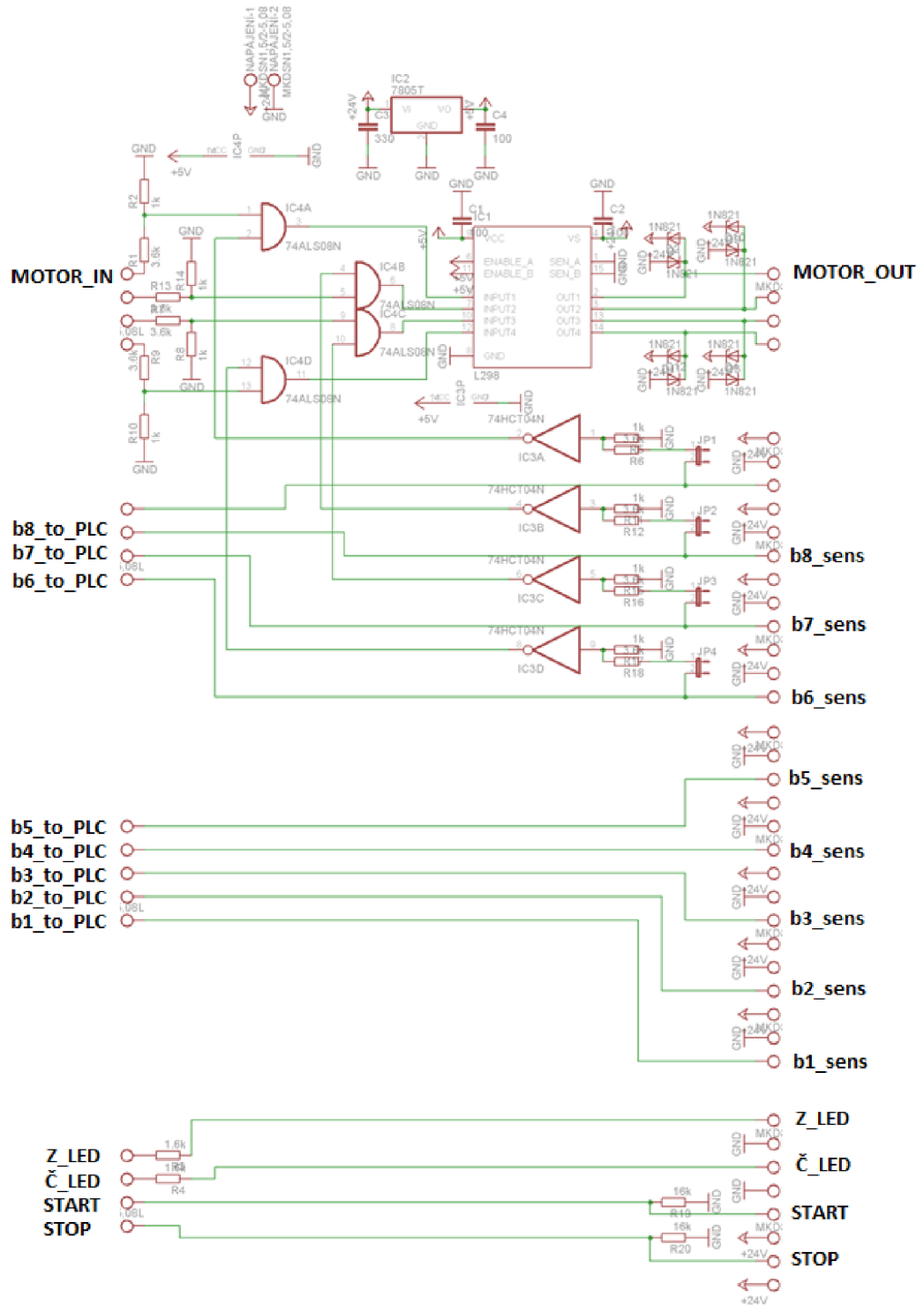
Rozmístění součástek a vodivých vrstev na řídicí desce a schéma zapojení jsou na následujících obrázcích.



Obrázek 14: Řídicí deska-hladina TOP



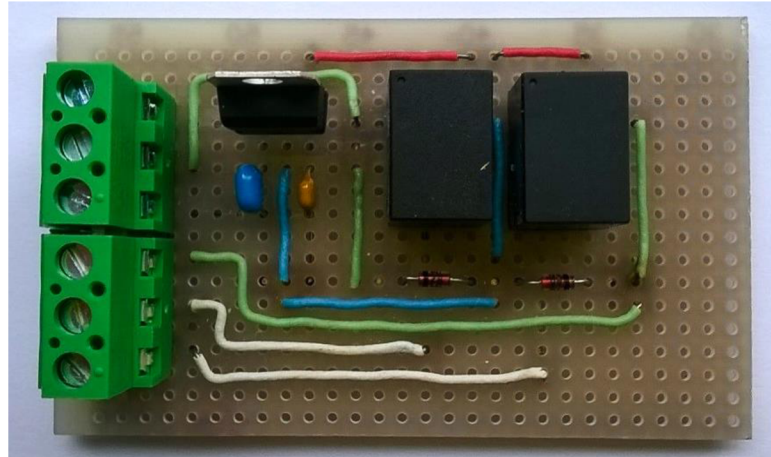
Obrázek 15: Řídicí deska-hladina BOTTOM



Obrázek 16: Schéma řídicí desky

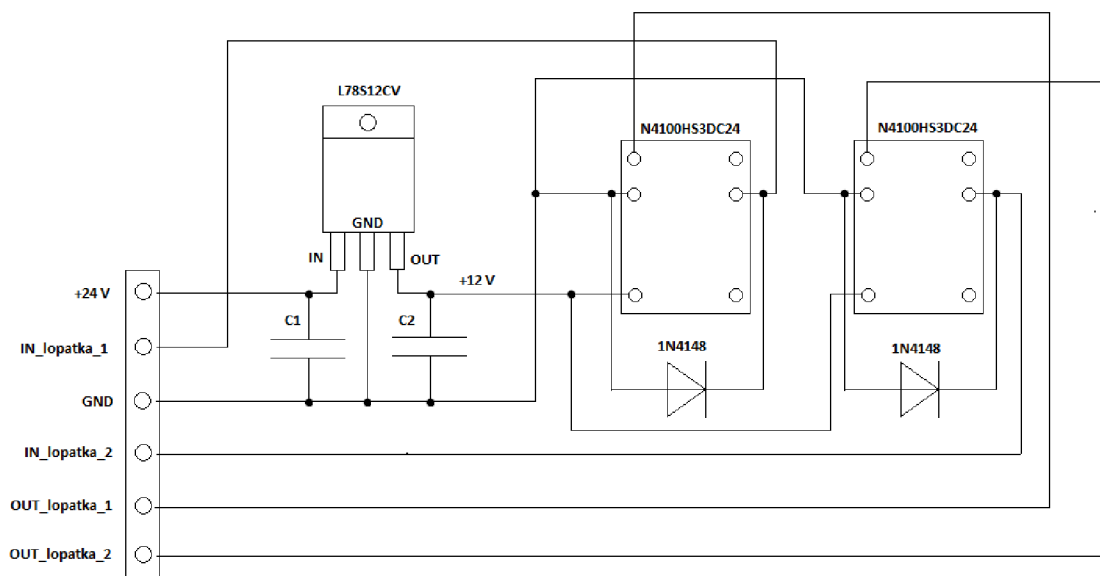
### 2.3.3 Ovládání lopatek

Lopatky pro výhoz kmenů obsahují 24 V solenoidy. Při připojení solenoidů na 24 V ně síla výhozu příliš velká, proto bylo nutné použít menší napětí. Proto byla zhotovena deska, viz obrázek 15, která stabilizuje napětí na 12 V DC. Toto napětí je dostatečné pro výhoz kmenu.



Obrázek 17: Deska pro ovládání lopatek

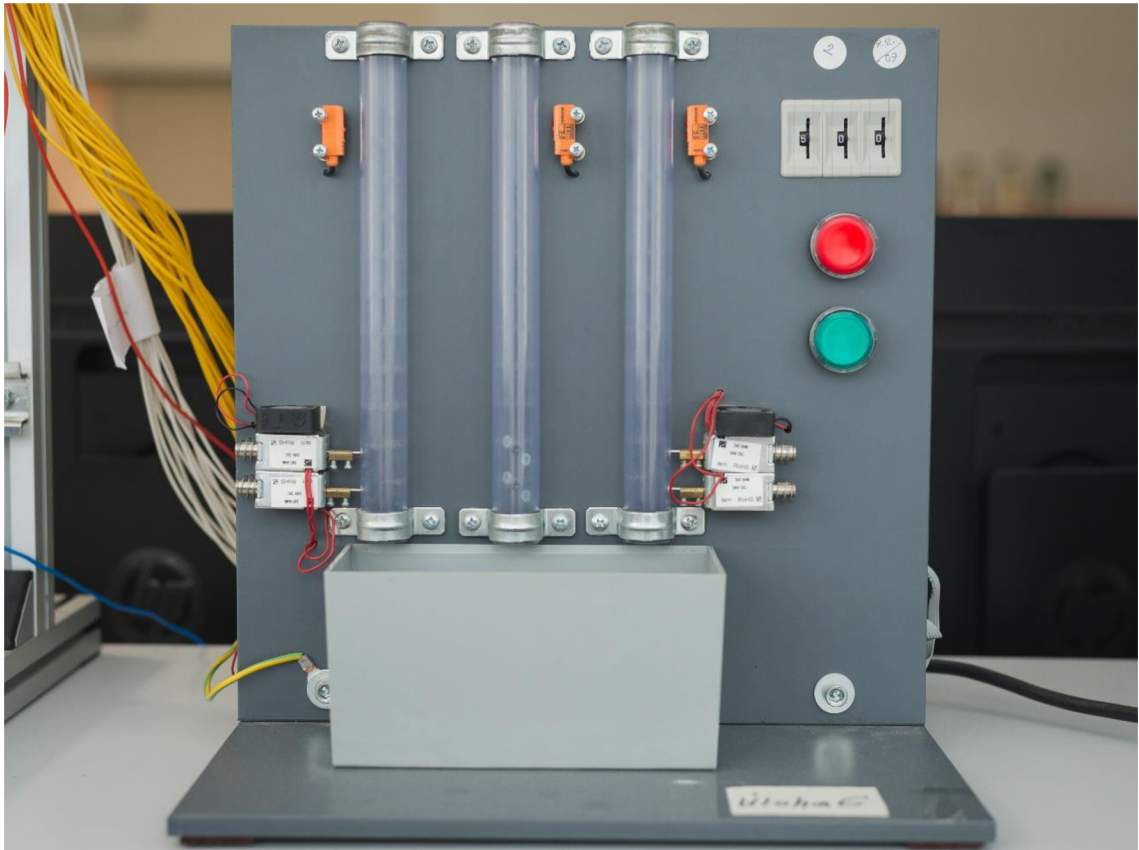
Schéma zapojení desky je na následujícím obrázku.



Obrázek 18: Schéma zapojení desky

## 3 POPIS MODELU DÁVKOVÁNÍ KULIČEK

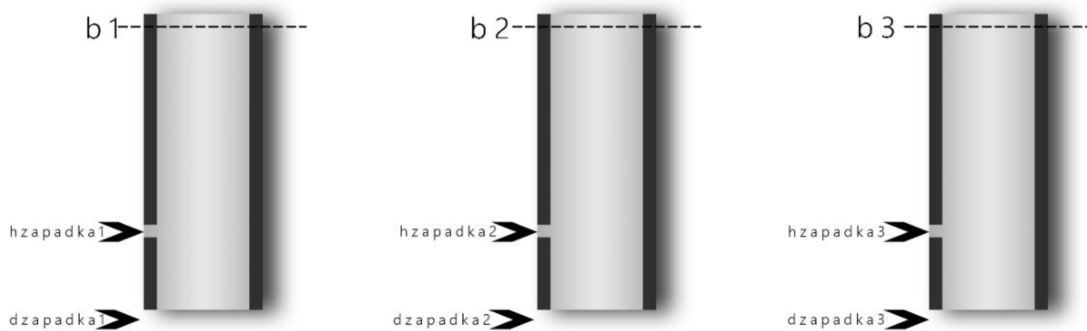
Tento laboratorní model představuje ukázkou dávkování kuliček, viz obrázek 13. Jako předmět pro dávkování jsou zde použity různobarevné kuličky, které se vkládají do válců pro dávkování. Studenti si zde vyzkouší práci s otočným nastavovačem a ovládání západek válců.



Obrázek 19: Dávkování kuliček-model

### 3.1 Popis funkce

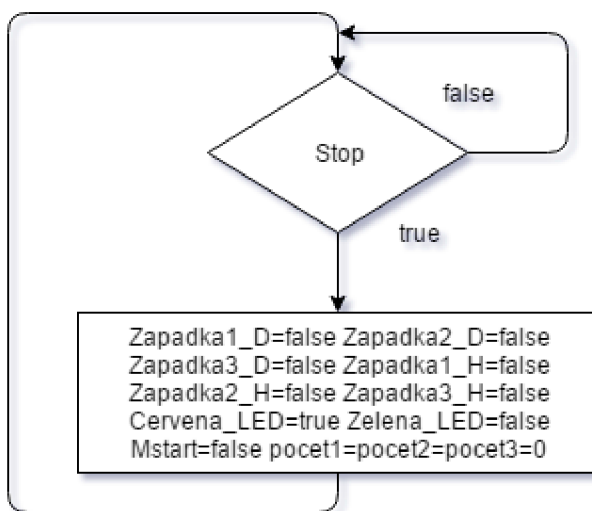
Po stisku tlačítka *Start* se vystrčí spodní západky a je tak možné naplnit válce kuličkami. Díky sensorům na vrchu každého válce, je možné určit počet kuliček v dávkovači. Po naplnění válců, je možné spustit dávkování. Počet kuliček, které mají být uvolněny, se nastavuje na otočném voliči. Pro spuštění dávkování je nutné mít pod válci nádobu na kuličky a znovu stisknout tlačítko *Start*. Dávkování probíhá tak, že se vystrčí horní západka, tak se oddělí spodní kulička od řady a zastrčí se spodní západka. Kulička vypadne a západky se vrátí na svoje místa. Tlačítkem *Stop* je možné kdykoliv vyprázdnit obsah válců. Rozmístění sensorů a lopatek je na obrázku 14.



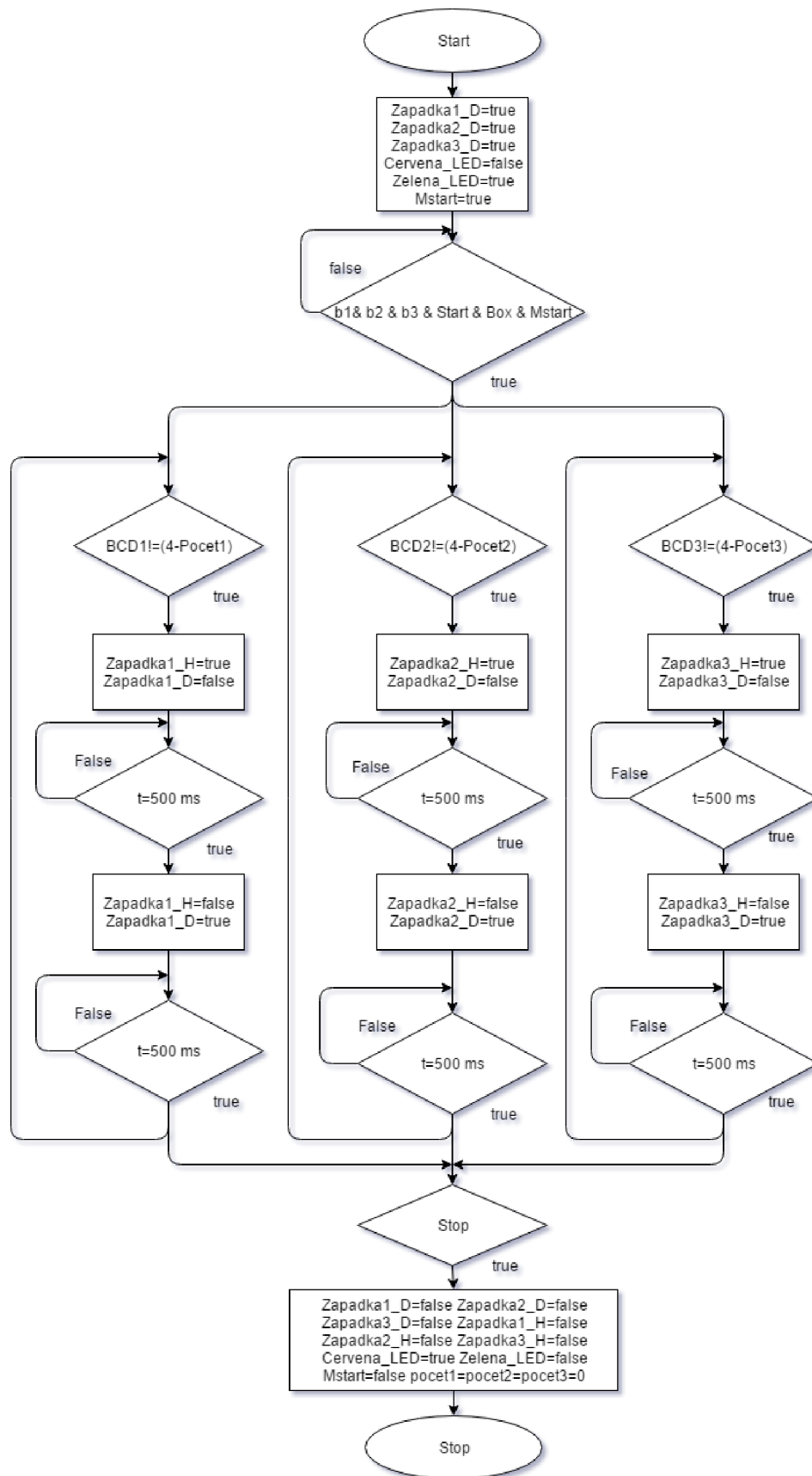
Obrázek 20: Dávkování kuliček-diagram

### 3.1.1 Vývojový diagram

Vývojové diagramy funkce modelu jsou na následujících obrázcích. Na prvním vývojovém diagramu je proces vyprázdnění, který má nejvyšší prioritu.



Obrázek 21: Vývojový diagram modelu dávkování kuliček-Proces vyprázdění



Obrázek 22: Vývojový diagram modelu dávkování kuliček-Proces dávkování



## 3.2 Tabulka vstupů a výstupů

Tabulku vstupů a výstupů jsem převzal z dokumentu [2].

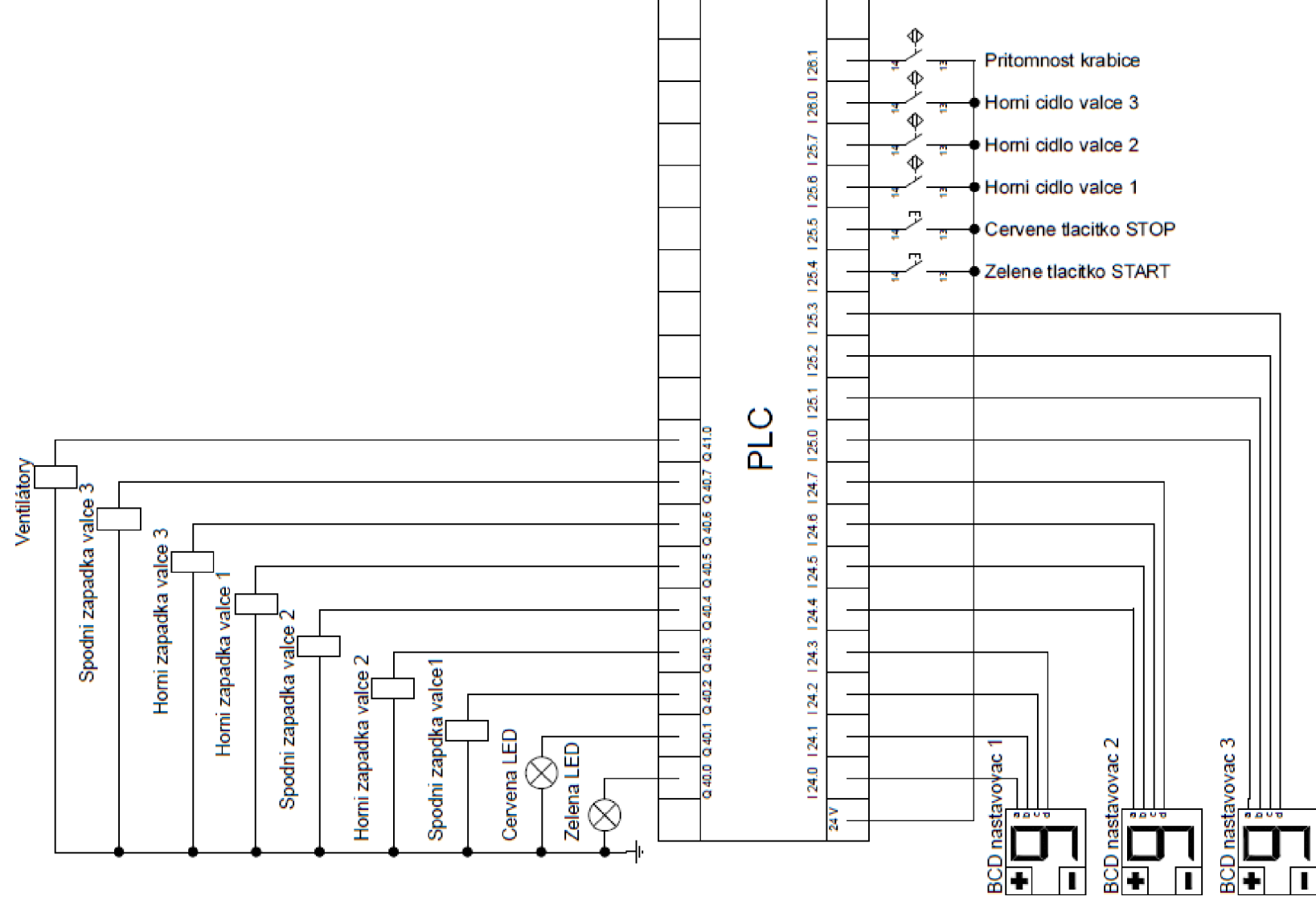
*Tabulka 5: Dávkování kuliček-seznam výstupů*

Q 40.0	Zelená LED
Q 40.1	Červená LED
Q 40.2	Spodní západka válce 1
Q 40.3	Horní západka válce 2
Q 40.4	Spodní západka válce 2
Q 40.5	Horní západka válce 1
Q 40.6	Horní západka válce 3
Q 40.7	Spodní západka válce 3

*Tabulka 6: Dávkování kuliček-seznam vstupů*

Adresa	Popis
I 24.0	BCD číslo otočného nastavovače válce 1
I 24.1	BCD číslo otočného nastavovače válce 1
I 24.2	BCD číslo otočného nastavovače válce 1
I 24.3	BCD číslo otočného nastavovače válce 1
I 24.4	BCD číslo otočného nastavovače válce 2
I 24.5	BCD číslo otočného nastavovače válce 2
I 24.6	BCD číslo otočného nastavovače válce 2
I 24.7	BCD číslo otočného nastavovače válce 2
I 25.0	BCD číslo otočného nastavovače válce 3
I 25.1	BCD číslo otočného nastavovače válce 3
I 25.2	BCD číslo otočného nastavovače válce 3
I 25.3	BCD číslo otočného nastavovače válce 3
I 25.4	Zelené tlačítko Start
I 25.5	Červené tlačítko Stop
I 25.6	Horní čidlo válce 1
I 25.7	Horní čidlo válce 2
I 26.0	Horní čidlo válce 3
I 26.1	Přítomnost krabice

### 3.3 Schéma zapojení



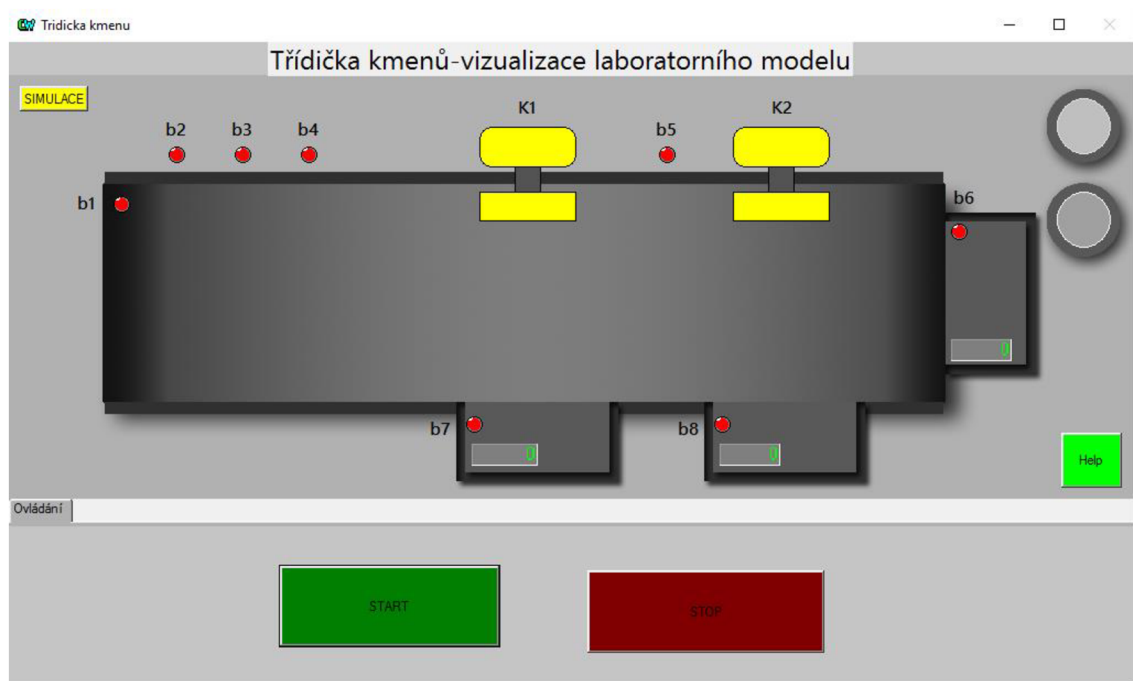
Obrázek 23: Dávkování kuliček-schéma zapojení

## 4 POPIS PROJEKTU TŘÍDIČKA KMENŮ

Tento projekt se zaměřuje na simulaci a vizualizaci modelu třídičky kmenů v laboratoři programovatelných automatů. Aplikace vznikla v prostředí Control Web 5 a některé grafické části v programu Adobe Photoshop. Aplikace vznikala postupně vytvořením proměnných, následně byl vytvořen grafický návrh a poté jsem programoval funkce přístrojů. Jména proměnných jsou převzata přímo z laboratorního návodu k této úloze a simulace je přizpůsobená tak, aby zvládla simulovat všechny zadané úkoly.

### 4.1 Popis programu

Při tvorbě programu jsem se zaměřil na podobnost s fyzickým modelem. Rozmístění senzorů a jednotlivých prvků přímo koresponduje s reálným umístěním. Pouze navíc jsem přidal zobrazovače, které ukazují počet jednotlivých kusů. Barvy jednotlivých prvků byly voleny tak, aby byly co nejvíce kontrastní. Cílem bylo, aby výsledná vizualizace byla co nejjednodušší a přehledná. Celý program se spouští v systémovém okně, které dovoluje uživateli je přemísťovat a minimalizovat. Výsledný program, viz Obrázek 16.



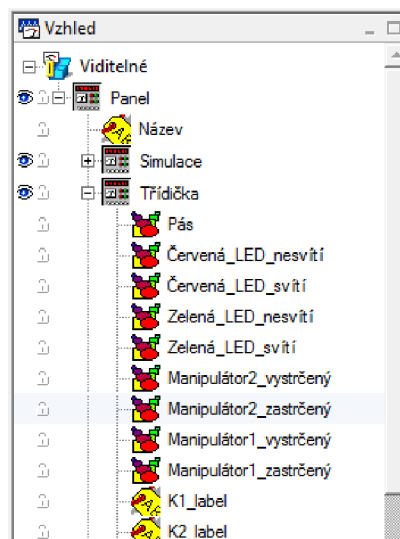
Obrázek 24: Třídíčka kmenů-vizualizace

Pod názvem aplikace se nachází pozadí vizualizace. Za pozadí se dá považovat i dopravník, který stejně jako pozadí vznikl v grafickém programu. Díky možnosti vkládat do programu obrázky, bylo možné více ovlivnit vizuální vzhled aplikace. Manipulátory jsou zde vyhotoveny jako žluté obdélníky, které se při aktivaci vysunou. Kolem pásu jsou umístěny indikátory stavu senzoru. Pokud senzor detekuje kmen, indikátor změní barvu z červené na zelenou. Zapnutý dopravník je zde zobrazen šedými šipkami ve směru pohybu pásu. Kmen je zde znázorněn hnědým obdélníkem. V každém zásobníku je umístěn displej, který zobrazuje počet kmenů v tomto zásobníku. Vpravo

nahore jsou umístěny dvě signálky, které ukazují stav modelu. Dole pod signálkami se nachází tlačítko *Help*, které po stisku zobrazí u jednotlivých prvků popisky. Vlevo nahore se nachází tlačítko, které po stisku uvede program do režimu simulace.

## 4.2 Popis přístrojů

Protože je aplikace rozsáhlejší a je zde požadavek na budoucí úpravy, bylo důležité, aby každý prvek měl přesné jméno podle toho, jako funkci splňuje. Aplikace je rozdělena do více panelů, do kterých se následně vkládají jednotlivé přístroje příslušící tomuto panelu, viz obrázek 25.



Obrázek 25: Třídíčka kmenů-panel vzhledu

Takto bylo možné realizovat přepínání panelů *Simulace* a *Ovládání*. Perioda obnovování jednotlivých prvků, je nastavena na 0,1 s. Toto zajišťuje okamžitou změnu stavu přístroje.

Důležitým prvkem v programu je přístroj jménem *Indicator*. Díky tomuto přístroji, lze jednoduše podle stavu proměnné, měnit vzhled prvku. Ve výchozím stavu, je reprezentován žárovkou. Vzhled přístroje lze měnit, proto jsem vložil již připravenou ikonu od výrobce, která lépe charakterizuje snímač. Přístroj může nabývat dvou stavů, sepnuto a rozepnuto. Stav sepnuto se projevuje zeleným zabarvením přístroje, stav rozepnuto naopak červeným zabarvením. Dále bylo potřeba nastavit, jakou proměnnou tento indikátor vyhodnocuje. Z pravidla to byla ta, která odpovídá danému umístění na fyzickém modelu.

Dalším často použitým prvkem je přístroj *Label*. Jak již název napovídá, tento přístroj umožňuje popis. Tento přístroj umožňuje vkládat řetězce a měnit formátování. Skrze celou aplikaci používám stejný font Segoe UI. Tento přístroj jsem využil pro popis jednotlivých snímačů a také jsem do něj vložil název aplikace.

Funkci tlačítka zastupuje přístroj *Switch*. Tento přístroj dokáže měnit stav booleovské proměnné. Dále umožňuje více režimů stisku. Lze nastavit režim *set\_true/false*, který po stisku nastaví proměnnou na hodnotu true nebo false. Dále umožňuje

*set\_true/false\_on\_press*, který jsem využil u tlačítka *Start*. Tento režim drží proměnou v *true* nebo *false* po dobu stisku. Jde tedy o interpretaci klasického tlačítka. Posledním režimem je *set\_flip\_flop*, který po stisku uvede proměnnou do stavu *true* a tlačítko drží v této pozici. Až po dalším stisku se proměnná uvede do stavu *false* a tlačítko se uvolní. Tento režim jsem využil u tlačítka *Stop*. Jde o interpretaci bezpečnostního tlačítka. Dále lze měnit typ přístroje, na výběr máme grafické tlačítko, textové tlačítko a zaškrtačací pole. Pro mě bylo nejvhodnější použít textové tlačítko, které již nepotřebuje další popis a jako indikaci stavu lze využít změnu barvy. Tlačítku *Start* jsem nastavil tmavý odstín zelené pro stav *false* a světlí odstín zelené pro stav *true*. Podobně jsem nastavil tlačítko *Stop*.

Dalším důležitým prvkem v aplikaci je *Rounded box*. Jedná se o obyčejný geometrický prvek, který jsem využil při tvorbě kmenů. Tento přístroj má stejné možnosti nastavení, jako ostatní geometrické prvky v aplikaci. Rozměry přístroje lze jednoduše měnit chycením a tažením jednoho z krajů přístroje. Stejně jednoduše lze měnit jeho polohu. Dále je možnost přesně nastavit tyto dva atributy, po rozbalení položky *společné* se v kolonce *position* nachází číselný kód ve tvaru *a, b, x, y*. První dvě čísla označují velikost v rastru aplikace a další dvě souřadnice znamenají umístění přístroje v tomto rastru. Dále je zde možnost nastavit podmínku viditelnosti v kolonce *visibility*. Pokud bude podmínka *true*, bude přístroj zobrazen, v opačném případě nebude vidět. Posledním užitým nastavením je barva v kolonce *colors*. Zde se nastavuje RGB hodnota, ale program nabízí již před chystané kombinace z palety. Zde jsem vybral tmavě hnědou. Přístroj jsem dále naprogramoval tak, aby se v režimu simulace, dokázal plynule pohybovat po pásu. Tento program popíši v kapitole 4.3.1.

Posledním prvkem v aplikaci je přístroj *Draw*. Tento přístroj umožňuje nakreslit pár základních objektů a v jisté formě pracovat s vrstvami. Také umožňuje nastavení tloušťky čáry a volbu barvy výplně a obrysu. Tento přístroj jsem použil při tvorbě manipulátorů a signálků. Efekt vysunutí manipulátoru a změny barvy u signálky jsem vytvořil pomocí atributu *visibility*, kdy mám dva přístroje „na sobě“ a zobrazují pouze jeden z nich. Přístroj také dovoluje nastavit atribut *transparent*, který nahradí bílé pozadí transparentním a lze se tak zbavit rámu obrázku.

### 4.3 Režim simulace

V režimu simulace dochází k autonomnímu řízení, kdy chod simulace ovládá algoritmus implementovaný v programu. Uživatel ovládá simulaci pomocí tlačítek na dolním panelu. K dispozici jsou tlačítka *Start*, *Stop* a tlačítka pro volbu velikosti kmenu. V tomto režimu program neposlouchá příkazy od PLC, ale chová se podle toho, jak byl naprogramován. Tento režim slouží k vyzkoušení všech dějů, které na modelu mohou nastat.

Tento režim se zapíná pomocí tlačítka *SIMULACE* v pravém horním rohu. Po jeho stisku se změní dolní ovládací panel na panel pro simulaci, viz Obrázek 26.



Obrázek 26: Panel ovládání simulace

### 4.3.1 Pohybu kmenu po dopravníku

Plynulý pohyb kmenů po dopravníku je realizován pomocí *for* cyklu a iterováním souřadnice *x* v příkazu *MoveTo(x, y)*. Ve stejném programu realizují i spínání senzorů v závislosti na poloze, kdy pomocí příkazu *if*, vyhodnocují pozici kmenu. Kód jsem vložil do procedury *OnActive()*, která je aktivní po celou dobu běhu programu. Na Obrázku 19, se nachází kód pro chod malé kmenů, který následně popíši.

```

timer = 0.1;
owner = Třidička;
position = 90, 160, 60, 70;
visibility = mala and not stredni;
colors
  interior = 140, 98, 57;
  border = 102, 75, 20;
end_colors;

procedure OnActivate();
var
  i      : integer;
  Period : real {init_value = 0.03};
begin
  if(mala=true)&(stredni=false)&(velka=false)&(start=true)then
  for i = 0+pom2 to 355 do
  if(start=false)then
  i=i-1;
  end;
  if(start=true)then
  MoveTo(90+i, 160);
  end;
  if(i=0)then
  b1=true;
  end;
  if((i=30))then
  b2=true;
  b1=false;
  end;
  if((i=60))then
  b2=false;
  end;
  if((i=70))then
  b3=true;
  end;
  if((i=100))then
  b3=false;
  end;
  if((i=120))then
  b4=true;
  end;
  if((i=180))then
  b4=false;
  end;
  while(stopka=true)do
  start=false;
  end;
  pause Period;
  send self;
end;
end;

```

Obrázek 27: Ukázka programu třídičky kmenů-pohyb po dopravníku

Na prvních řádcích programu se nachází definice objektu. Je zde jeho obnovovací frekvence, kterému panelu patří, iniciální pozice, podmínka zobrazení a barva. Následuje hlavička procedury *OnActive()* a pod ní definice proměnných. Proměnná *i* je typu integer a proměnná *Period* je typu real a její hodnota je 0,03. Proměnná *i* je zde kvůli iteraci v cyklu *for*, kdy se její hodnota přičítá k souřadnici *x* v příkazu *MoveTo(90+i, 160)*. Tento příkaz funguje tak, že objekt přemístí na pozici, kterou mu udává hodnota v závorkách. Proměnná *Period* obsahuje čas, po který je cyklus *for* pozastaven. To provádí příkaz *pause Period*, který se provede v každém kroku cyklu a jde tedy o nastavení dočasné periody objektu. Tento příkaz zde byl nutný z důvodu vyladění rychlosti na podobnou, jako je dopravník fyzického modelu. Dále je zde příkaz *while(stopka=true)do*, jedná se o cyklus, který po dobu aktivního tlačítka *Stop* pozastaví pohyb kmenu a cyklu. V tomto cyklu se nachází příkaz *start=false*, který znemožňuje aktivaci tlačítka *Start* a tak i spuštění pásu, dokud je *Stop* aktivní. Dále program obsahuje sérii příkazů *If*, které vyhodnocují pozici kmenu a podle ní nastavují senzory na *true* nebo *false*.

### 4.3.2 Přemístění kmenu do zásobníku

V přechodí kapitole jsme si popsali, jak funguje pohyb kmenu po dopravníku, nyní si popíšeme, jak se kmen přemístí z dopravníku do zásobníku.

Stejně jako přechodí program, i tento se nachází v proceduře *OnActive()*. Na prvním řádku kontrolují, jestli se kmen nachází na pozici pro přemístění do zásobníku. Následuje ho *for* cyklus, který stejně jako v přechodím případě iteruje proměnou *i*, kterou si zinicilizuje na hodnotu nula. Pokud je splněna podmínka a kmen je v požadované pozici, příkazem *MoveTo(445, 220+i)* se přesune kmen o 60 v rastru ve směru osy *y* a aktivuje manipulátor K1. Toto způsobí efekt výhozu kmenu. Dále kmen pokračuje směrem do zásobníku. Zde se již pohybuje po klouzačce a není tak možné kmen zastavit stiskem tlačítka *Stop*. Po dosáhnutí senzoru *b7* dojde k jeho aktivaci. Toto hlídá podmínka *if(i=40)then* a po přemístění na pozici 445, 295 dojde k jeho deaktivaci. Manipulátor se deaktivuje na pozici 445, 290. Dále se zde nachází příkaz *pause*, který funguje jako v předchozím programu. Po dokončení cyklu dojde k iteraci počtu malých kmenů v zásobníku o jedna a proměnná *mala* se nastaví do *false*. Tím dojde k zneviditelnění kmenu a návratu na výchozí pozici.

```

if(i>=314)then
for i = 0 to 75 do
  MoveTo(445, 220+i);
  if((i=0))then
    K1=true;
  end;
  if((i=70))then
    K1=false;
  end;
  if((i=40))then
    b7=true;
  end;
  if((i=75))then
    b7=false;
  end;
  pause Period;
  send self;
end;
if(i>70) then
  pocet_malych=pocet_malych+1;
  mala=false;
end;
end;
end procedure;

```

Obrázek 28: Ukázka programu třídičky kmenů-konec

## 4.4 Režim vizualizace

V tomto režimu program reaguje na proměnné přicházející z PLC. Je zároveň výchozím režimem po spuštění programu. Zde se program chová tak, jak jej uživatel naprogramoval v PLC. Jedná se tedy o vizualizaci dějů, které se odehrávají na fyzickém modelu. Uživatel má k dispozici pouze tlačítka *Start* a *Stop*, které mohou nahradit fyzické tlačítka na modelu.

Aby nedocházelo k ovlivňování jednotlivých režimů, obsahují přístroje podmínku díky, které bude přístroj vyhodnocovat pouze stav buď z PLC nebo z programu. Například snímač b2 obsahuje podmínku (*sim and b2*) or *M\_b2*. Pomocná proměnná *sim*, je nastavována tlačítkem *Simulace* a je v logickém součinu s proměnnou *b2*, která je nastavována programem pro simulaci. Toto je v logickém součtu s proměnnou *M\_b2*, která je odečítána z PLC. Konfigurace této proměnné je popsána v kapitole 6.2.

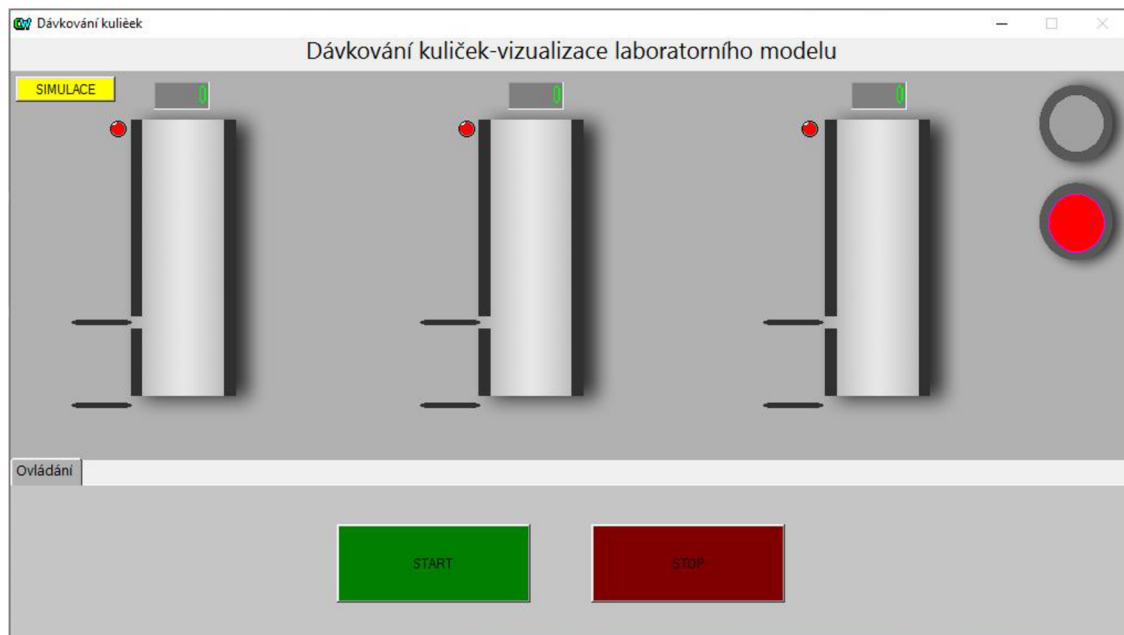


## 5 POPIS PROJEKTU DÁVKOVÁNÍ KULIČEK

Tento program se zaměřuje na simulaci a ovládání modelu dávkování kuliček v laboratoři programovatelných automatů. Aplikace vznikala postupně vytvořením proměnných, následně byl vytvořen grafický návrh a poté jsem programoval funkce přístrojů. Jména proměnných jsou převzata přímo z laboratorního návodu k této úloze a simulace je přizpůsobená tak, aby zvládla simulovat všechny zadané úkoly.

### 5.1 Popis vizualizace

Při tvorbě toho programu jsem se zaměřil na jednoduchost a přehlednost. Důležité zde také bylo také zanechat podobnost s fyzickým modelem. Ukázka programu je na obrázku 29.



Obrázek 29: Dávkování kuliček

Vizualizace se spouští v systémovém okně. Stejně jako v předchozí aplikaci, má uživatel možnost pouze přesouvat a minimalizovat toto okno. Pod názvem aplikace se nachází pozadí s dávkovacími válci. Toto pozadí vzniklo v grafickém programu s ohledem na vzhled vizualizace modelu Třídíčky kmenů.

### 5.2 Popis přístrojů

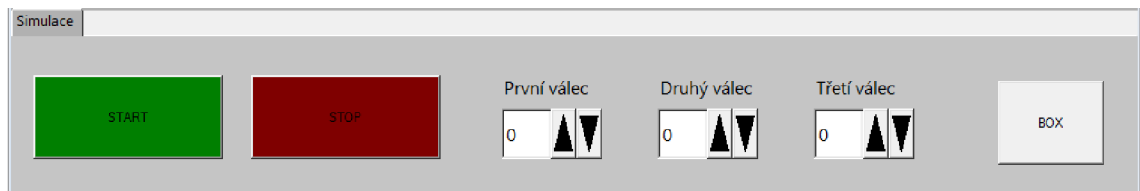
Prvky použité v tomto programu jsou stejné jako ve třídíčce kmenů a změny se týkají pouze atributů. Na následujících řádcích, budou popsány přístroje, které se dočkaly největších změn.

Kuličky ve válcích jsem vytvořil pomocí přístroje *Draw*. Nejdříve bylo potřeba vytvořit dostatek kuliček do každého válce a zvolit správný průměr. Dále zvolit vhodnou barvu, kterou jsou nakonec odstíny šedi, tak aby se každá série kuliček lišila od jiné série.

Pomocí atributu *visibility* se mi podařilo vytvořit efekt přidávání a ubírání kuliček. Podrobněji je tento efekt popsán v kapitole 5.3.1.

## 5.3 Režim simulace

Tento režim, stejně jako v případě třídičky kmenů, je řízen autonomně, pomocí algoritmu implementovaného do programu. Uživatel má v tomto režimu k dispozici sadu tlačítek pro *Start*, *Stop*, *Box* a číslíkové nastavovače na počet dávkování kuliček. V tomto režimu není program řízen PLC, ale chová se podle požadavků v zadání pro tuto laboratorní úlohu. Tento režim slouží k vyzkoušení všech dějů, které mohou na modelu nastat. Do tohoto režimu se uživatel dostane po stisku tlačítka *SIMULACE* v pravém horním rohu. Po jeho stisku se změní dolní ovládací panel za panel pro simulaci, viz Obrázek 30.



Obrázek 30: Ovládání simulace kuliček

### 5.3.1 Plnění válců

Plnění válců pracuje pro oba režimy na stejném principu. V obou případech vyhodnocuje přístroj výraz v kolonce *visibility*, viz obrázek 23. Na tomto obrázku je kus kódu, který vyhodnocuje podmínku zobrazení kuličky na vrcholu prvního válce.

```
visibility = pocet1 >= 4;
```

Obrázek 31: Dávkování kuliček-příkaz *visibility*

Program pro simulaci dávkování se nachází v proceduře *OnActive()*. Díky této proceduře je možné spustit dávkování kdykoliv v běhu programu, pokud to podmínky dovolí. Na následujícím obrázku, se nachází kus kódu, který slouží pro plnění prvního válce. Cyklus *for*, iteruje proměnnou *pocet1*, což je počet kuliček v prvním válci. Každá kulička obsahuje podmínku zobrazení podle hodnoty této proměnné. Tato podmínka se nachází na obrázku 23. Dále bylo potřeba přidat zpoždění mezi plněním jednotlivými kuličkami. To jsem realizoval pomocí Funkce *pause* a proměnné *Period*, která je nastavena na 1 s. Aby senzor na vrcholu válce reagoval na přítomnost kuličky, vložil jsem do programu příkazy *b1=true* a následně, po zpoždění *b1=false*. Tyto příkazy postupně spínají a vypínají indikátor stavu senzoru na vrcholu prvního válce. Celý kus programu je podmíněn stiskem tlačítka *Start*, *Box* a podmínkou, že na otočném nastavovači je nula. Dále je tělo cyklu v podmínce *if(not (pocet1=4))then*, která hlídá, aby proměnná *pocet1* nebyla větší než 4.

```

if(Pom1=true)&(davk1=0)&(Box=true)then
for i=0 to 4 do
if(not(pocet1=4))then
pocet1=pocet1+1;
b1=true;
pause Period;
send self;
b1=false;
end;
end;
end;

```

Obrázek 32: Ukázka programu pro plnění válců

## 5.3.2 Vypouštění kuliček

Vypouštění kuliček probíhá ve třech krocích, kdy je střídavě měněna pozice horní a dolní západky ve válci a vně válce. Také se snižuje počet kuliček v příslušném válci do doby, než dosáhne dávkování požadovaného počtu vypuštěných kuliček. Oba kroky i s programy se nachází v následujících dvou podkapitolách.

### 5.3.2.1 Horní západka

Na obrázku 33 se nachází kód pro ovládání horní západky prvního válce při dávkování. Kód je v proceduře *OnActive()*, díky kterému bude aktivní po celou dobu běhu programu. Na začátku si definuji proměnnou *i*, kterou využiji pro iterování for cyklu. Pod začátkem procedury, označeném klíčovým slovem *begin*, se nachází podmínka startu pohybu západky. Tato podmínka hlídá stisk tlačítka *Start*, aktivaci proměnné *Pom2*, což je pomocná proměnná, která je *true* ve chvíli dávkování a při stisku tlačítka *Start*. Dále je v podmínce výraz *not (davk1=0)*, který hlídá to, aby v době spuštění dávkování, nebylo na otočném voliči nastaveno jiné číslo než nula. Výrazem *not (pocet1=0)*, program hlídá to, aby ve válci byla alespoň jedna kulička. Následuje přiřazení proměnné *pom1* hodnotu výrazu *davk1*, která obsahuje požadovaný počet kuliček pro dávkování. Následuje for cyklus, který začíná od nuly a končí na hodnotě *pom1-1*, která v tuto chvíli obsahuje požadavek na dávkování zmenšený o jedna. Následuje příkaz *if(pocet1=0)then*, který zapříčiní ukončení dávkování v případě, že ve válci již nejsou kuličky. Poté následuje příkaz *MoveTo(110,225)*, který přesune západku do válce. Následuje pauza, která je požadována v zadání. Poté se západka přesune zpět do polohy mimo válce příkazem *MoveTo(55,225)* a následuje další pauza. Po ukončení dávkování se vynuluje proměnná *davk1*, která obsahovala požadovaný počet kuliček pro dávkování a pomocná proměnná *Pom2* se nastaví na *false*. Poté je procedura ukončena.

```

procedure OnActivate();
var
  i      :integer;
begin
  if(START=true)&(Pom2=true)&not(davk1=0)&not(pocet1=0)then
  pom1=davk1;
  for i=0 to (pom1-1) do
  if(pocet1=0)then
  davk1=0;
  end;
  MoveTo(110,225);
  pause 1;
  send self;
  MoveTo(55,225);
  pause 1;
  send self;
  end;
  davk1=0;
  Pom2=false;
  end;
end_procedure;

```

Obrázek 33: Ukázka programu pro dávkování kuliček-horní západka

### 5.3.2.2 Dolní západka

Na obrázku 34 je kus kódu pro ovládání spodní západky prvního válce. Tento program je koncipovaný podobně jako ovládání horní západky, proto se v popisu zaměřím na nové řádky.

Pod klíčovým slovem *begin* se nachází příkaz *if*, který vyhodnocuje podmínku (*START=true*) & (*davk1=0*), která po splnění přesune západku do polohy pod válce, kde má zabránovat vypadnutí kuličky při plnění. Pohyb západky zde vychází z pozice, kde je zasunutá ve válci. Proto je první příkaz *MoveTo(55,300)*, který ji přesune vně válec. Po dokončení dávkování je nastavena proměnná *Pom2* na hodnotu *false*, což zamezí spuštění dávkování. Nastavením proměnné *START* na hodnotu *false* zamezím stavu, kdy by uživatel držel po dobu dávkování tlačítko *Start* a došlo by tak k okamžitému spuštění dávkování.

```

procedure OnActivate();
var
  i      :integer;
  Period :real {init_value=1};
begin
  if(START=true)&(davk1=0)then
  MoveTo(110,300)
  end;
  if(pocet1>0)then
  MoveTo(110,300);
  end;
  if(START=true)&(Pom2=true)&not(davk1=0)&not(pocet1=0)then
  pom1=davk1;
  for i=1 to davk1 do
  if(pocet1=0)then
  davk1=0;
  end;
  MoveTo(55,300);
  pocet1=pocet1-1;
  pause 1;
  send self;
  MoveTo(110,300);
  pause 1;
  send self;
  Pom2=false;
  end;
  davk1=0;
  START=false;
  end;
end_procedure;

```

Obrázek 34: Ukázka programu pro dávkování kuliček-dolní západka

## 5.4 Režim vizualizace

V tomto režimu program reaguje na proměnné přicházející z PLC. Tento režim je aktivní po spuštění programu. Zde se program chová podle toho, jak jej uživatel naprogramoval v PLC. Jde tedy o vizualizaci dějů na fyzickém modelu. Uživatel má zde k dispozici pouze tlačítka *Start* a *Stop*, které mohou nahradit fyzická tlačítka na modelu. Dále jsou zde zobrazovače, které ukazují aktuální hodnotu otočných voličů.

Aby nedocházelo k ovlivňování jednotlivých režimů, obsahují přístroje podmínku, díky které bude přístroj vyhodnocovat pouze stav buď z PLC nebo z programu. Tato podmínka se skládá z proměnné *Sim*, která je aktivní pouze v době simulace a příslušného výrazu programu.

### 5.4.1 Čtení z BCD číselníků

Model obsahuje BCD číselník, na kterém uživatel nastaví požadovanou hodnotu pro dávkování. Tento číselník je k PLC připojen pomocí 4 vstupů pro jedno číslo. Samo elektrické připojení nereprezentuje BCD hodnotu. Před použitím je jej potřeba převrátit, tedy na vstupu *I 24.3* se objeví logická jednička v případě, že bude na nastavovači jedna. Další vstupu z číselníku do PLC lez najít v tabulce 6.

Následující obrázek ukazuje program, který převádí hodnotu z prvního číselníku do desítkové soustavy. Program je vložen v proceduře *OnActive()*, přístroje *meter*, označený jako *Pocet1*. V první části definuju proměnné typu integer, do kterých budu ukládat celá čísla v závislosti na kombinaci *M\_V11* až *M\_V14*. První podmínkou je, že režim simulace je vypnutý. Následuje série podmínek, podle nichž se pomocné proměnné *cislo1* až *číslo 4* nastaví na příslušnou hodnotu. Poté následuje sečtení těchto proměnných a přiřazení do proměnné *davk1*, která nyní obsahuje číslo z číselníku. Tato proměnná se následně zobrazí v přístroji *Pocet1*.

```
procedure OnActivate();
var
cislo1      :integer;
cislo2      :integer;
cislo3      :integer;
cislo4      :integer;
begin
if(Simulace=false)then
if(M_V11=true)then
cislo4=8;
else
cislo4=0;
end;
if(M_V12=true)then
cislo3=4;
else
cislo3=0;
end;
if(M_V13=true)then
cislo2=2;
else
cislo2=0;
end;
if(M_V14=true)then
cislo1=1;
else
cislo1=0;
end;
davk1=cislo1+cislo2+cislo3+cislo4;
end;
end_procedure;
```

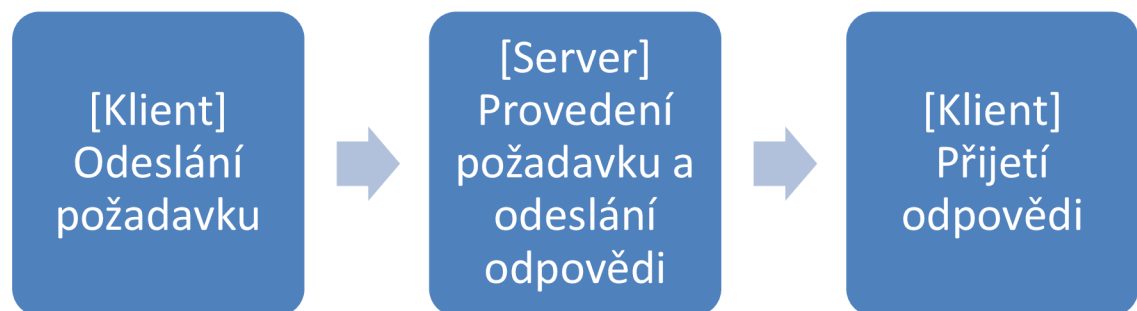
Obrázek 35: Dávkování kuliček-převod z BCD

## 6 PROPOJENÍ S PLC

K propojení vytvořených programů s PLC jsem využil komunikačního protokolu MODBUS.

### 6.1 Protokol MODBUS

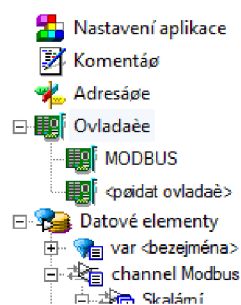
Protokol MODBUS je komunikační protokol, který v 1979 vytvořila firma MODICON. Pracuje na úrovni aplikační vrstvy ISO/OSI modelu. Umožňuje komunikaci typu klient-server mezi různými zařízeními. Může komunikovat po sériové lince, ethernetu nebo po optické síti s využitím protokolu TCP/IP. Komunikace probíhá metodou požadavek-odpověď, viz obrázek 24. Požadavek i odpověď jsou opatřeny kódem funkce. [6]



Obrázek 36: Protokol MODBUS

### 6.2 Konfigurace v Control Webu

Prostředí Control Web má v sobě integrovaný ovladač pro komunikaci protokolem MODBUS. [7] Nejprve bylo potřeba vytvořit mapovací soubor a tím nadefinovat typ proměnné na určitém rozsahu kanálů. V mém parametrickém souboru jsem definoval pro kanály 100-199 typ real a obousměrný zápis, poté na kanálech 200-299 booleovskou proměnnou také s obousměrným zápisem. Dále bylo potřeba vytvořit parametrický soubor a určit tak adresu PLC v síti a jaké kanály budu využívat. Poté bylo potřeba vložit tyto soubory do programu a vybrat komunikační protokol, viz obrázek 25.



Obrázek 37: Ovladač MODBUS

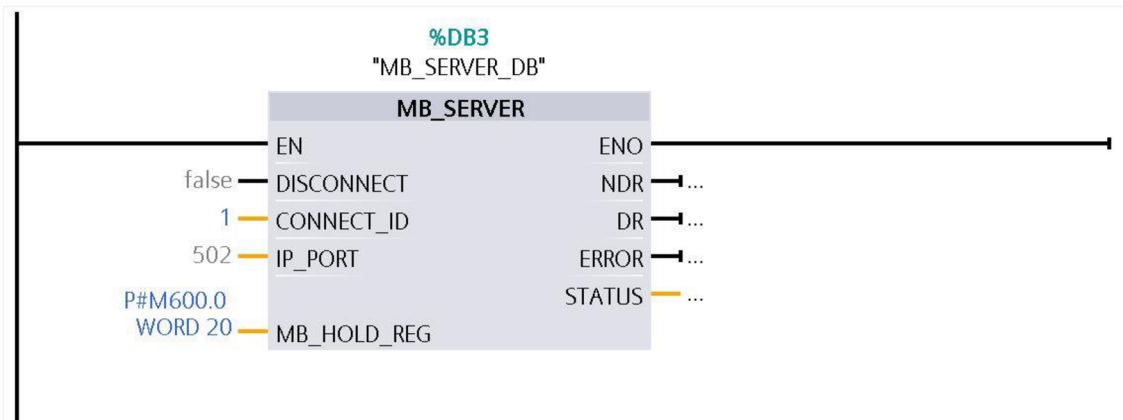
Nyní mohu vkládat proměnné a definovat, na kterém kanále se nacházejí. Při vytváření proměnných vybírám driver, který jsem dříve vytvořil. Od tohoto se přímo odvíjí i typ proměnné, který jsem definoval v mapovacím souboru. Na obrázku 26, se nachází tabulka proměnných i s kanály.

name	type	init_value	driver	driver_index	direction
M_Start	boolean	false	MODBUS	201	bidirectional
M_Stop	boolean	false	MODBUS	202	bidirectional
M_b2	boolean	false	MODBUS	203	bidirectional
M_b5	boolean	false	MODBUS	204	bidirectional
M_b6	boolean	false	MODBUS	205	bidirectional
M_b8	boolean	false	MODBUS	206	bidirectional
M_b3	boolean	false	MODBUS	207	bidirectional
M_b4	boolean	false	MODBUS	208	bidirectional
M_b1	boolean	false	MODBUS	209	bidirectional
M_b7	boolean	false	MODBUS	210	bidirectional
M_Zelena_LED	boolean	false	MODBUS	211	bidirectional
M_Cervena_LED	boolean	false	MODBUS	212	bidirectional
M_Dopravnik	boolean	false	MODBUS	213	bidirectional
M_K1	boolean	false	MODBUS	214	bidirectional
M_K2	boolean	false	MODBUS	215	bidirectional
M_Mala	boolean	false	MODBUS	216	bidirectional
M_Stredni	boolean	false	MODBUS	217	bidirectional
M_Velka	boolean	false	MODBUS	218	bidirectional
M_Pocet_malych	real	0	MODBUS	102	bidirectional
M_Pocet_stred...	real	0	MODBUS	105	bidirectional
M_Pocet_velk...	real	0	MODBUS	108	bidirectional
<paidat>	<paidat>	<paidat>	<paidat>	<paidat>	<paidat>

Obrázek 38: Komunikační kanály

### 6.3 Propojení s TIA portálem

Na cvičení se oba modely programují v prostředí TIA Portal V13, bylo tedy potřeba zajistit komunikaci i s tímto prostředím. TIA Portal obsahuje konfigurační blok MB\_SERVER, který tuto komunikaci zajišťuje. Tento blok jsem vložil do hlavního programu OB1 a následně provedl jeho konfiguraci, viz obrázek 27. Proměnné se ukládají do merkrů od adresy M600.0 a je na ně připravených celkem 20 wordů. Tyto merkry jsem pojmenoval a uložil do tabulky tagů. Takto připravené projekty jsem přiložil do přílohy.



Obrázek 39: Blok pro komunikaci v TIA Portal

## **7 ZÁVĚR**

Cílem této práce bylo vytvořit dvě aplikace pro simulaci a vizualizaci dějů na modelech třídičky kmenů a dávkovače kuliček. Tyto programy následně propojit s PLC a připravit je tak na budoucí užití při výuce programovatelných automatů. Paměťová náročnost obou aplikací je optimalizována s ohledem na zadání úkolů pro práci na obou modelech. Zadání se mi podařilo splnit a modely jsou tak připraveny na použití.



## 8 POUŽITÁ LITERATURA

- [1] **Moravské přístroje a.s.** Co je Control Web? *Moravské přístroje a.s.* [Online] 4. 10 2010. <http://www.mii.cz/art?id=380&cat=146&lang=405>.
- [2] **Ing. Radek Štohl, Ph.D., Ing. Jan Pásek, CSc.** *Programovatelné automaty-laboratorní cvičení*. Brno, Jihomoravský, Česká republika : autor neznámý.
- [3] **Moravské přístroje.** Control Web 5. [Online]  
<http://www.mii.cz/download/doc/cw5cz.pdf>.
- [4] —. *Nápověda systému Control Web 5*. místo neznámé : Moravské přístroje a.s.
- [5] **Ing. Radek Štohl, Ph.D., a další.** *Laboratorní cvičení BPPA*.
- [6]. **Ing. Ronešová, Andrea.** Přehled protokolu Modbus. [Online] 2005.  
<http://home.zcu.cz/~ronesova/bastl/files/modbus.pdf>.
- [7] **Moravské přístroje a.s.** Ovladač pro komunikaci protokolem MODBUS přes sériové rozhraní nebo pomocí TCP/IP. *Moravské přístroje a.s.* [Online] [Citace: 10. 5 2016.]  
<http://www.mii.cz/art?id=734&lang=405>.

## 9 SEZNAM ZKRATEK

<b>Zkratka</b>	<b>Význam</b>
AND	Logický součin
BCD	Binary Coded Decimal
GND	Zemní vodič
HTTP	Hypertext Transfer Protocol
LAD	LADDER
LED	Svítivá dioda
PLC	Programovatelný logický automat
RGB	Červená-zelená-modrá
SCADA	Správa řízení a sběr dat
STL	Strukturovaný text
TCP/IP	Transmission Control Protocol/Internet Protocol

## 10 SEZNAM PŘÍLOH

- Příloha 1. Kmeny.zip
- Příloha 2. Kuličky.zip