



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

APLIKACE SYSTÉMU INTOUCH PRO OVLÁDÁNÍ PORTÁLOVÉHO JEŘÁBU

APPLICATION OF INTOUCH SYSTEM FOR CONTROL OF PORTAL CRANE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAELA DIBLÍKOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN PÁSEK, CSc.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Studentka: Michaela Diblíková

ID: 98076

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Aplikace systému InTouch pro ovládání portálového jeřábu

POKYNY PRO VYPRACOVÁNÍ:

Vytvoření SW pro ovládání a vizualizace (SCADA) laboratorního modelu portálového jeřábu pomocí systému InTouch. Předpokládá připojení modelu jeřábu na PLC a vytvoření funkčního programu v jazyku STEP7. Cílem je vytvoření objektů a oken pro vizualizaci a ovládání automatické funkce jeřábu.

DOPORUČENÁ LITERATURA:

Manuály firmy Siemens, manuály firmy Wonderware, projektové dokumentace automatizačních firem - např. APV.

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Jan Pásek, CSc.

prof. Ing. Pavel Jura, 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

Bakalářská práce se zabývá vytvořením SW pro ovládání a vizualizaci (SCADA) laboratorního modelu portálového jeřábu pomocí systému InTouch. Dále připojení modelu jeřábu na PLC a vytvoření funkčního programu v jazyku STEP7. Cílem je vytvoření objektů a oken pro vizualizaci a ovládání automatické funkce jeřábu. V práci je uveden a popsán program vytvořený v jazyku STEP 7 a program pro samotnou vizualizaci a ovládání modelu portálového jeřábu v systému InTouch.

ABSTRACT

Bachelor thesis deals with creating software for control and visualization (SCADA) laboratory model gantry crane system using InTouch. Furthermore, the connection to the PLC model crane and a functional language program in STEP7. The aim is to create objects and windows for visualization and control automatic functions crane. The work is shown and described a program created in STEP 7 language and program for the actual visualization and control model of the portal crane system in InTouch.

KLÍČOVÁ SLOVA

SCADA, HMI, operátorská stanice, funkční popisy, přehledové okno, procesní okno, alarmní okno, okno bilancí, pop-up okno

KEY WORDS

SCADA, HMI, Operator Station, functional descriptions, overview window, process window, an alarm window, window balances, pop-up window

BIBLIOGRAFICKÁ CITACE

DIBLÍKOVÁ, M. *Aplikace systému InTouch pro ovládání portálového jeřábu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 48 s. Vedoucí bakalářské práce Ing. Jan Pásek, CSc.

PROHLÁŠENÍ

„Prohlašuji, že svou bakalářskou práci na téma „Aplikace systému InTouch pro ovládání portálového jeřábu“ 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 § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **31. května 2010**

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské Ing. Janu Páskovi, CSc. a dále pak panu Ing. Petru Slezákovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **31. května 2010**

.....
podpis autora

OBSAH

ABSTRAKT	3
KLÍČOVÁ SLOVA	4
BIBLIOGRAFICKÁ CITACE	5
PROHLÁŠENÍ	6
PODĚKOVÁNÍ	7
OBSAH	8
1. ÚVOD	10
2. AUTOMAT SIMATIC S7-300	11
2.1 DEFINICE PLC PODLE IEC 1131-1	11
2.2 ZÁKLADNÍ VLASTNOSTI	11
2.3 POPIS MODULŮ	12
2.4 ROZMÍSTĚNÍ PRVKŮ NA CPU JEDNOTCE	13
2.5 POUŽITÁ HARDWAROVÁ KONFIGURACE	14
3. POPIS JEŘÁBU	17
3.1 VSTUPY A VÝSTUPY	17
4. SIMATIC STEP7	19
4.1 PLC PROGRAM	19
4.1.1 <i>Popis diagramu hlavních funkcí a jejich přechodů</i>	20
4.1.2 <i>Program pro obsluhu operátorem</i>	20
4.1.3 <i>Inicializace</i>	23
4.1.4 <i>Program přenášení plechovek</i>	27
5. SPOJENÍ SIMATIC A INTOUCH	30
5.1.1 <i>Vytvoření komunikace</i>	30
6. INTOUCH	31
6.1 ČLENĚNÍ SYSTÉMU	31
6.2 VIZUALIZACE	32

6.2.1	<i>Vytvoření nové proměnné</i>	32
6.2.2	<i>Vytváření animačních propojení</i>	34
6.2.3	<i>Vytváření skriptů QuickSkripts</i>	35
6.2.4	<i>Vytvořený ovládací panel</i>	36
6.2.5	<i>Okno jeřábu</i>	36
6.3	OVLÁDÁNÍ PROGRAMU	37
6.3.1	<i>Alarmy</i>	38
6.3.2	<i>Ruční ovládní jeřábu</i>	38
6.3.3	<i>Ovládní inicializace</i>	39
6.3.4	<i>Ovládní provozu jeřábu</i>	40
6.3.5	<i>Security</i>	41
7.	ZÁVĚR	43
8.	POUŽITÁ LITERATURA	44
9.	SEZNAM OBRÁZKŮ	45
10.	SEZNAM TABULEK	47
11.	SEZNAM PŘÍLOH	48
11.1	PŘILOŽENÉ DVD	48
11.1.1	<i>Funkční program v SIMATIC Manageru</i>	48
11.1.2	<i>Funkční program v systému InTouch</i>	48
11.1.3	<i>Dokumentace v elektronické podobě</i>	48

1. ÚVOD

Úkolem této práce bylo vytvoření SW pro ovládání a vizualizace (SCADA) laboratorního modelu portálového jeřábu pomocí systému InTouch. Dále pak připojení modelu jeřábu na PLC a vytvoření funkčního programu v jazyku STEP7. Hlavním cílem bylo vytvoření objektů a oken pro vizualizaci a ovládání automatické funkce jeřábu.

V první části práce jsem uvedla základní popis a vlastnosti použitého programovatelného automatu SIMATIC S7-300. Dále jsem se zabývala základním popisem modelu portálového jeřábu, kde je uvedeno umístění čidel a přehled vstupů a výstupů. V další kapitole je uveden popis a základní ideologie programu v jazyce STEP 7. V oddílu pět je popsáno vytvoření komunikace mezi systémy InTouch a SIMATIC. Poslední část práce se zabývá systémem InTouch a řešením vizualizace a ovládáním modelu jeřábu.

2. AUTOMAT SIMATIC S7-300

2.1 DEFINICE PLC PODLE IEC 1131-1

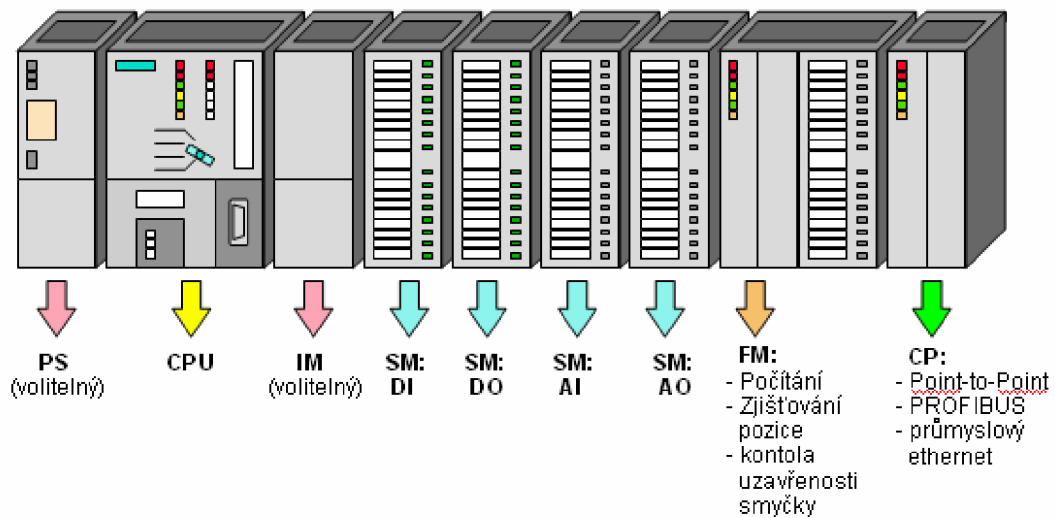
Programovatelný logický automat (PLC) je číslicový elektronický systém navržený pro použití v průmyslovém prostředí, který používá programovatelnou paměť pro uložení uživatelsky orientovaných instrukcí sloužících k implementaci specifických funkcí, jako jsou logické funkce, funkce pro vytváření sekvencí, funkce pro časování, funkce pro čítání a funkce pro aritmetické výpočty, a to za účelem řízení různých typů výrobních strojů a procesů pomocí číslicových a analogových vstupů a výstupů [3].

2.2 ZÁKLADNÍ VLASTNOSTI

- Modulární mini PLC přístroj pro jednodušší a středně pokročilé aplikace
- Široká škála výkonově rozdílných CPU jednotek
- Velký výběr rozšiřujících periferních modulů
- Rozšiřitelnost až do konfigurace s 32 moduly
- Integrovaná systémová sběrnice uvnitř jednotek a modulů
- Kompatibilní s průmyslovými sítěmi
 - MPI (Multipoint interface)
 - PROFIBUS
 - Industrial Ethernet
- Nastavování parametrů pomocí programu „HWConfig“

2.3 POPIS MODULŮ

S7-300: Moduly



Obrázek 1. Moduly S7-300

SM (Signální moduly): Digitální vstupní moduly (DI): 24V DC

Digitální výstupní moduly(DO): 24V DC

Analogové vstupní moduly(AI)

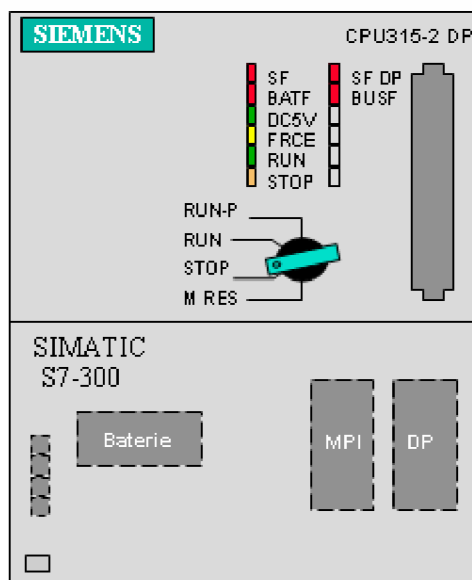
Analogové výstupní moduly(Ao)

IM (Propojovací modul)

FM (Funkční modul): Realizuje složité řídicí funkce

CP (Komunikační procesor): Pro připojení PLC do síťových datových systémů

2.4 ROZMÍSTĚNÍ PRVKŮ NA CPU JEDNOTCE



Obrázek 2. CPU jednotka S7-300

- **Přepínače režimů**

MRES :	funkce <u>M</u> odul <u>R</u> eset
STOP :	režim Stop; program neběží.
RUN :	program běží, přístup pouze-čtení z PG
RUN-P:	program běží, přístup čtení/zápis z PG.

- **Stavové LED diody**

SF :	skupinová chyba; vnitřní závada CPU
BATF :	závada v baterii (Battery fault)
DC5V :	interní ukazatel napětí 5 V DC
FRCE :	FORCE; indikuje, že je zpuštěn alespoň jeden vstup nebo výstup
RUN :	bliká, provádí-li CPU náběh, při cyklickém zpracování programu RUN svítí
STOP :	indikuje STOP stav; Pomalu bliká při

požadavku resetování paměti; Rychle bliká, když probíhá reset paměti; Pomalu bliká když je nezbytné resetovat paměť z důvodů vsunutí paměťové karty.

- **Paměťový modul:** CPU jednotka má zásuvnou pozici pro paměťový modul Flash-EPROM. Modul ukládá obsah programu pro případ výpadku proudu, takže není nutná baterie.
- **Rám pro baterii:** Pod krytem je přihrádka pro lithiovou baterii, která zálohuje paměť RAM.
- **MPI konektor:** Pro připojení PG/PC.
- **DP rozhraní:** Interface pro přímé spojení rozdělených I/O k CPU.

2.5 POUŽITÁ HARDWAROVÁ KONFIGURACE

Pro řízení je použit procesor CPU 315F-2 DP.

Pro čtení analogových hodnot je použita karta AI16x13Bit. Jedná se o kartu šestnácti analogových vstupů.

Ke čtení digitálních hodnot je využita karta DI24xDC 24V s 24 digitálními vstupy o napětí 24V.

Karta pro ovládání analogových výstupů AO4x12Bit, má 4 analogové výstupy.

K ovládání diskrétně řízených zařízení je použita karta DO10xDC24V/2A. Jedná se o kartu 10 digitálních výstupů o napětí 24V a maximálním proudovém zatížení 2A.

Na Obrázek 3. je zobrazena hardwarová konfigurace použitého PLC.

1	PS 307 5A	6ES7				
2	CPU 315F-2 DP	6ES7	V2.0	2		
X2	DP				204.7	
3						
4	AI6x13Bit	6ES7			8...23	8...11
5	AO4x12Bit	6ES7				272...
6	FDI24xDC24V	6ES7			24...3	24...2
7	FDO10xDC24V/2A	6ES7			40...4	40...4
8	CP 343-1 AdvancedIT	6GK7	V1.0	3	390...	390...
9						
10						
11						

Obrázek 3. Hardwarová konfigurace

Adresy vstupů a výstupů jsou v tabulka 1 a tabulka 2.

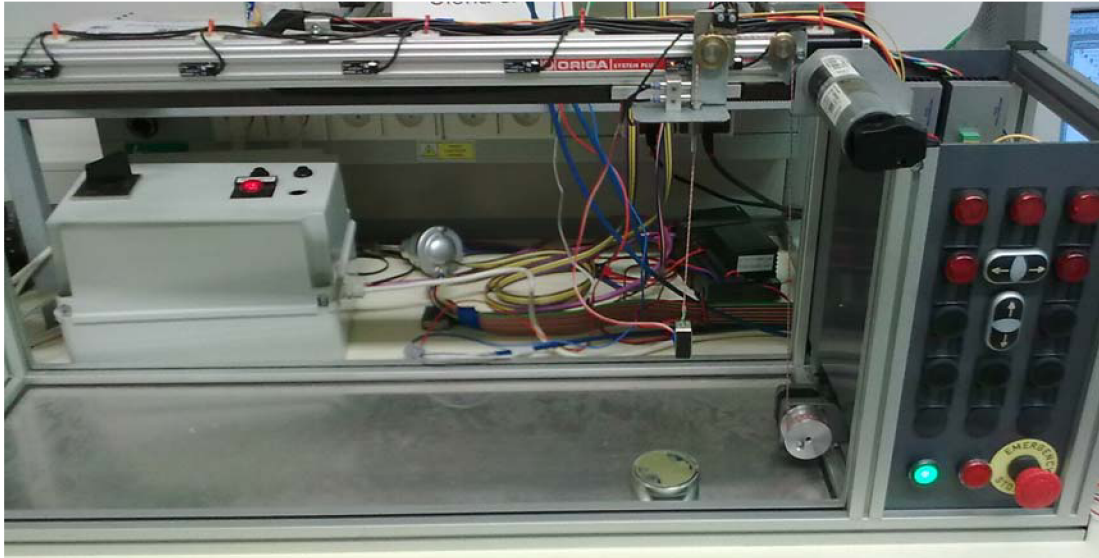
Vstupy	Adresy vstupů	Funkce
Digitální	I 24.7	Snímač 1 aktivní
	I 24.4	Snímač 2 aktivní
	I 25.0	Snímač 3 aktivní
	I 24.3	Snímač 4 aktivní
	I 25.1	Snímač 5 aktivní
	I 24.5	C1, jeřáb v horní poloze
	I 24.6	C1, odvinutí lana jeřábu
	I 24.1	Tlačítko dolů, nepoužito
	I 26.6	Tlačítko nahoru, nepoužito
	I 24.2	Tlačítko doprava, nepoužito
	I 24.0	Tlačítko doleva, nepoužito
	I 26.5	Tlačítko T4, využito jako alarmní tlačítko
Analogové		Potenciometr

tabulka 1: Vstupy PLC

Výstupy	Adresy výstupů	Funkce
Digitální	Q 40.0	DC motor, nastavení směru
	Q 40.1	Krokový motor, krok
	Q 40.2	Krokový motor, nastavení směru
	Q 40.3	Krokový motor, aktivace
	Q 40.4	Magnet, aktivace
Analogové		DC motor nastavení rychlosti posuvu

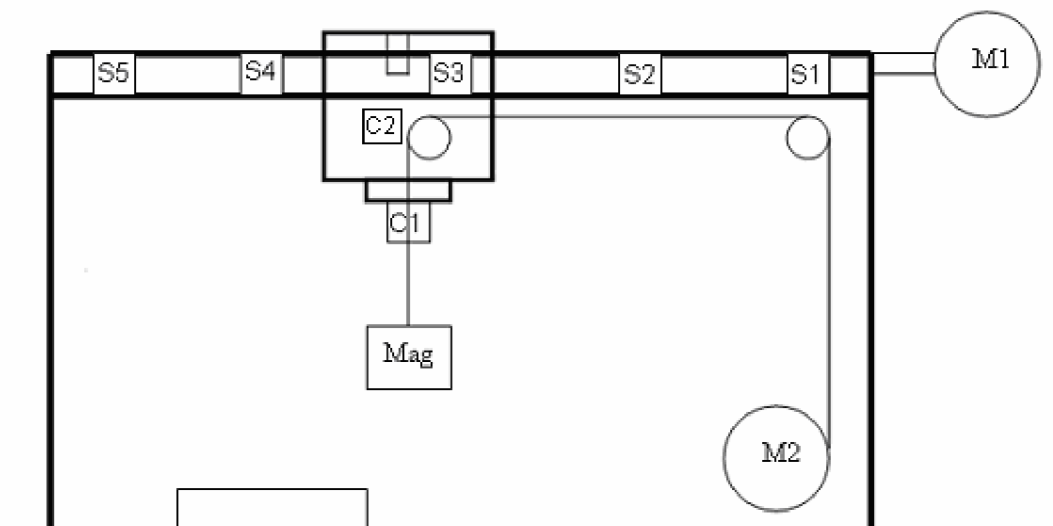
tabulka 2: Výstupy PLC

3. POPIS JEŘÁBU



Obrázek 4. Foto jeřábu

3.1 VSTUPY A VÝSTUPY



Obrázek 5. Jeřáb s označením prvků

Vodorovný pohyb jeřábu zajišťuje stejnosměrný motor M1, který je ovládán pomocí digitálního výstupů pro nastavení směru posunu a analogového výstupu (0-10V) pro nastavení rychlosti pohybu motoru.

Odměrování horizontální polohy jeřábu je zajištěno, buďto analogově a to pomocí potenciometru v rozsahu 0-10V, a nebo pomocí diskretních snímačů (S1-S5). Použijeme-li pro odměrování polohy snímače, musíme brát v úvahu, že reakce snímače po najetí jeřábu se liší podle toho zda najíždíme jeřábem zprava či zleva.

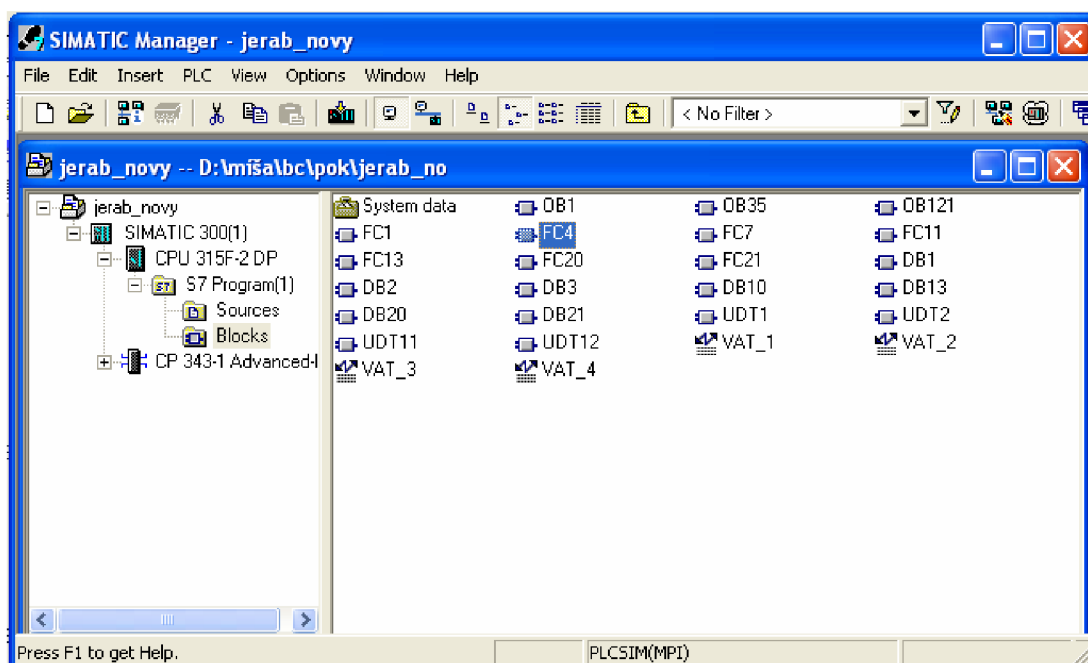
Odvíjení a navíjení lanka magnetu je zajištěno krokovým motorem M2. Krokový motor je řízen digitálními impulsy a diskretními výstupy pro nastavení směru navíjení a aktivaci motoru.

Nejvyšší poloha magnetu je indikována pomocí senzoru C1, umístěným na vozíku jeřábu. Senzor C2 je odvíjením a navíjením lana spínán a rozpínán, lze jím tedy zjišťovat aktuální vertikální polohu magnetu.

Pro přenos nákladu je použit elektromagnet, řízen digitálním výstupem pro aktivaci a deaktivaci magnetu.

4. SIMATIC STEP7

STEP 7 je standardní software pro konfiguraci a programování SIMATIC programovatelných logických automatů. Je součástí výrobního odvětví SIMATIC software. Zahrnuje výkonné nástroje a funkce pro podporu množství úloh spjatých s automatizačními projekty [2]. Ústředním nástrojem STEP 7 je SIMATIC Manager. Pomocí SIMATIC Manageru obstaráváme veškerá data použitá v projektu.

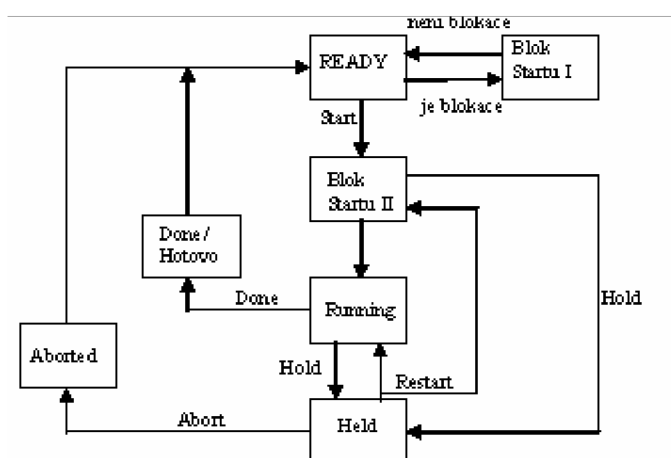


Obrázek 6. SIMATIC Manager

4.1 PLC PROGRAM

Program lze rozdělit do tří bloků z hlediska řízení jeřábu. První blok slouží pro obsluhu operátorem, druhý blok je použit pro inicializaci jeřábu a posledním blokem se řídí samostatný program pro přeložení plechovky.

V programu je brán zřetel na standardní řízení technologických procesů, tj. je dodržen diagram hlavních funkcí a jejich přechodů viz. Obrázek7.



Obrázek 7. Stavový diagram hlavních funkcí

4.1.1 Popis diagramu hlavních funkcí a jejich přechodů

Jestliže nejsou přítomny žádné blokace je program ve stavu READY, pokud systém SCADA vyšle požadavek na start, je znova provedena kontrola blokování a teprve po té se přepne do stavu Running (toto blokování je v této práci prováděno pomocí podmínky v systému InTouch).

Nyní se provádí hlavní program, pokud není vyslán požadavek Hold, pak přejde do stavu Done (tedy program je hotov) a přepne do stavu READY. Přechod Hold pozastaví program, nastaví Held a vyčkává na další povel, s příchodem povelu Abort je program přerušen, s příkazem Restart se vrátíme do posledního kroku.

4.1.2 Program pro obsluhu operátorem

Pro obsluhu byl nejprve vytvořen datový blok DB (Data Blocks) Obrázek 8 a uživatelsky definované datové typy UDT (User-Defined Data Types) Obrázek 9, které jsou stejné pro motor M1 (DC motor) i pro motor M2 (krokový motor). Toto je důležité pro komunikaci mezi systémy SIMATIC a InTouch.

Program se nachází ve funkčním bloku FC4 a je vepsán do dvou Networků , první network obsahuje instrukce pro ovládání DC motoru M1 Obrázek 11 a druhý pro řízení krokového motoru M2 Obrázek 12.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	M1	"motor"		
+4.0	M1_rychlost	REAL	0.000000e+000	
+8.0	M1_enable	BOOL	FALSE	
+8.1	M1_SMER	BOOL	FALSE	
+10.0	M2	"motor"		
+14.0	M2_magnet	BOOL	FALSE	
+14.1	M2_SMER	BOOL	FALSE	
+14.2	M2_enable	BOOL	FALSE	
=16.0		END_STRUCT		

Obrázek 8. Datový blok DB

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	stav	STRUCT		
+0.0	automat	BOOL	FALSE	
+0.1	aktivni	BOOL	FALSE	
+0.2	alarm	BOOL	FALSE	
=2.0		END_STRUCT		
+2.0	Cmd	STRUCT		
+0.0	plc	BOOL	FALSE	
+0.1	operator	BOOL	FALSE	
+0.2	start	BOOL	FALSE	
+0.3	stop	BOOL	FALSE	
=2.0		END_STRUCT		
=4.0		END_STRUCT		

Obrázek 9. UDT

Address	Name	Type	Initial value	Actual value	Comment
0.0	M1.stav.automat	BOOL	FALSE	FALSE	
0.1	M1.stav.aktivni	BOOL	FALSE	FALSE	
0.2	M1.stav.alarm	BOOL	FALSE	FALSE	
2.0	M1.Cmd.plc	BOOL	FALSE	FALSE	
2.1	M1.Cmd.operator	BOOL	FALSE	FALSE	
2.2	M1.Cmd.start	BOOL	FALSE	FALSE	
2.3	M1.Cmd.stop	BOOL	FALSE	FALSE	
4.0	M1_rychlost	REAL	0.000000e+000	0.000000e+000	
8.0	M1_enable	BOOL	FALSE	FALSE	
8.1	M1_SMER	BOOL	FALSE	FALSE	
10.0	M2.stav.automat	BOOL	FALSE	FALSE	
10.1	M2.stav.aktivni	BOOL	FALSE	FALSE	
10.2	M2.stav.alarm	BOOL	FALSE	FALSE	
12.0	M2.Cmd.plc	BOOL	FALSE	FALSE	
12.1	M2.Cmd.operator	BOOL	FALSE	FALSE	
12.2	M2.Cmd.start	BOOL	FALSE	FALSE	
12.3	M2.Cmd.stop	BOOL	FALSE	FALSE	
14.0	M2_magnet	BOOL	FALSE	FALSE	
14.1	M2_SMER	BOOL	FALSE	FALSE	
14.2	M2_enable	BOOL	FALSE	FALSE	

Obrázek 10. Adresy prvků

Při ovládání motoru M1 se musí aktivovat i motor M2, který odvíjí či navíjí lano podle směru pohybu motoru M1.

Příkazem cmd.start aktivujeme motor a nastavíme stav.aktivni. Pokud splníme podmínky pro pohyb motoru doprava, resetují se (nastaví se log. 0). proměnné M1_SMER a M2_SMER, které určují směr pohybu motorů a setují se (nastaví se log. 1) podmínky M1_enable a M2_enable pro aktivaci motorů. Splněním podmínek pro pohyb vlevo měníme pouze proměnné M1_SMER a M2_SMER, takže je setujeme. Podmínky pro aktivaci motorů jsou, jako při pohybu vpravo v logické 1 (setovány). Po příchodu cmd.stop deaktivujeme (resetujeme) stav.aktivni a oba motory Obrázek 11.

Network: 1		Ovládání DC motoru	
//start			
A	"motor_db".M1.Cmd.start		DB1.DBX2.2
S	"motor_db".M1.stav.aktivni		DB1.DBX0.1
//smer do prava			
A	"motor_db".M1.stav.aktivni		DB1.DBX0.1
AN	"param".smer		DB10.DBX10.0
R	"motor_db".M1_SMER	//doprava	DB1.DBX8.1
R	"motor_db".M2_SMER		DB1.DBX14.1
S	"motor_db".M1_enable		DB1.DBX8.0
S	"motor_db".M2_enable		DB1.DBX14.2
//smer do leva			
A	"motor_db".M1.stav.aktivni		DB1.DBX0.1
A	"param".smer		DB10.DBX10.0
S	"motor_db".M2_enable		DB1.DBX14.2
S	"motor_db".M1_SMER	//doleva	DB1.DBX8.1
S	"motor_db".M2_SMER		DB1.DBX14.1
S	"motor_db".M1_enable		DB1.DBX8.0
//stop			
A	"motor_db".M1.Cmd.stop		DB1.DBX2.3
R	"motor_db".M1.stav.aktivni		DB1.DBX0.1
R	"motor_db".M2_enable		DB1.DBX14.2
R	"motor_db".M1_enable		DB1.DBX8.0
// osetreni			
O{			
A	"senzory".SENZOR1		DB3.DBX0.0
A	"motor_db".M1_enable		DB1.DBX8.0
AN	"param".smer		DB10.DBX10.0
}			
O{			
A	"senzory".SENZOR5		DB3.DBX0.4
A	"motor_db".M1_enable		DB1.DBX8.0
A	"param".smer		DB10.DBX10.0
}			
R	"motor_db".M1_enable		DB1.DBX8.0
R	"motor_db".M2_enable		DB1.DBX14.2

Obrázek 11. Program ovládání DC motoru

Řízení motoru M2 je jednodušší tím, že nemusíme brát v úvahu oba motory a oba směry, a zabýváme se tedy pouze motorem M2 Obrázek 12.

```

Network: 2      ovládání krokového motoru

//start
A  "motor_db".M2.Cmd.start      DB1.DBX12.2
S  "motor_db".M2.stav.aktivni   DB1.DBX10.1

//pohyb vzhuru
A  "motor_db".M2.stav.aktivni   DB1.DBX10.1
AN "motor_db".M2_SMER          DB1.DBX14.1
S  "motor_db".M2_enable        DB1.DBX14.2

//pohyb dolu
A  "motor_db".M2.stav.aktivni   DB1.DBX10.1
A  "motor_db".M2_SMER          DB1.DBX14.1
S  "motor_db".M2_enable        DB1.DBX14.2

//stop
A  "motor_db".M2.Cmd.stop       DB1.DBX12.3
R  "motor_db".M2.stav.aktivni   DB1.DBX10.1
R  "motor_db".M2_enable        DB1.DBX14.2

//konecova poloha
A  "C1"                          I24.5
R  "motor_db".M2_enable        DB1.DBX14.2

```

Obrázek 12. Ovládání krokového motoru

4.1.3 Inicializace

Z kroku 0 přejdeme do kroku 1 po příchodu podmínky cmd.start Obrázek 14.

V kroku 1 vyjedeme magnetem vzhůru, to ovládají podmínky pro nastavení směru krokového motoru (resetujeme MOTOR2_SMER) a pro sepnutí impulsů pro ovládání krokového motoru (setujeme MOTOR2_enable). Aktivací senzoru C1 (indikace horní polohy magnetu) přejde program do kroku 2 Obrázek 15.

Zde posuneme jeřáb do pravé krajní polohy podmínkami reset MOTOR1_SMER a set MOTOR1_enable na senzor S1 Obrázek16. Bude-li aktivní senzor S1 (senzory.senzor1) program přejde do stavu done a je ukončen. Inicializaci provádíme vždy před spuštěním programu přeskládávání plechovek, aby bylo zcela zřejmé v jaké pozici se jeřáb nachází. Program inicializace je v bloku FC20 a její proměnné jsou v blocích DB20 a UDT11 a UDT12 .

Address	Name	Type	Initial value	Actual value	Comment
0.0	CMD.Start	BOOL	FALSE	FALSE	
0.1	CMD.Hold	BOOL	FALSE	FALSE	
0.2	CMD.Restart	BOOL	FALSE	FALSE	
0.3	CMD.Abort	BOOL	FALSE	FALSE	
0.4	CMD.Reset	BOOL	FALSE	FALSE	Pouziti spolu s nadrazenym programem
0.5	CMD.Acknowledge	BOOL	FALSE	FALSE	Potvrzovani od operatora
2.0	STA.Ready	BOOL	FALSE	FALSE	
2.1	STA.Interlocked	BOOL	FALSE	FALSE	
2.2	STA.Running	BOOL	FALSE	FALSE	
2.3	STA.Held	BOOL	FALSE	FALSE	
2.4	STA.Aborted	BOOL	FALSE	FALSE	
2.5	STA.Done	BOOL	FALSE	FALSE	
4.0	INTERLOCKS.Interlock1	BOOL	FALSE	FALSE	Neni zadan zdroj
4.1	INTERLOCKS.Interlock2	BOOL	FALSE	FALSE	Neni zadan cil
6.0	ALARM.Alarm1	BOOL	FALSE	FALSE	Nouzove tlacitko
6.1	ALARM.Alarm2	BOOL	FALSE	FALSE	
8.0	ALARM_SET	WORD	W#16#0	W#16#0	Alarmy pro SCADA
10.0	ALARM_RST	WORD	W#16#0	W#16#0	Pomocne pro potvrzeni SCADA alarmu
12.0	STEP	INT	0	0	Cislo kroku
14.0	STEP_OLD	INT	0	0	Cislo kroku pro restart
16.0	STEP_TIME	INT	0	0	Doba trvani kroku
18.0	spare2	INT	0	0	
20.0	START_POS	INT	0	0	Startovaci pozice
22.0	DEST_POS	INT	0	0	Cilova pozice
24.0	MOTOR1_SMER	BOOL	FALSE	FALSE	smer doprava 0 doleva 1
24.1	MOTOR2_SMER	BOOL	FALSE	FALSE	smer nahoru(do prava) 0 dolu(doleva)
24.2	MOTOR2_enable	BOOL	FALSE	FALSE	sepnuti impulsu krok. motoru
24.3	MOTOR1_enable	BOOL	FALSE	FALSE	nastaveni rychlosti

Obrázek 13. Proměnné inicializace

```

Network: 4   Ridici oddil

L   "PRG_DB_INIT".STEP // Load jump destination number  DB20.DBW12   -- Cislo kr
    into ACCU 1-L-L.   oku
JU  LSTX              // Jump destination if ACCU 1-L-L > 3.
JU  S0                // Idle
JU  S1                // Magnet nahoru
JU  S2                // Jerab doprava
JU  END              // 3 - rezerva
JU  END              // 4 - rezerva
JU  END              // 5 - rezerva
JU  END              // 6 - rezerva
JU  END              // 7 - rezerva
JU  END              // 8 - rezerva
JU  END              // 9 - rezerva
JU  HELD             // 10 - Held
JU  RSTR             // 11 - Restart

LSTX: JU  END

Network: 5   Ready, interlocked

S0:  NOP  0          //Permitted instruction
R   "PRG_DB_INIT".MOTOR2_enable  DB20.DBX24.2  -- sep
    nuti impulsu krok. motoru
R   "PRG_DB_INIT".MOTOR1_enable  DB20.DBX24.3  -- nas
    taveni rychlosti

// R   "PRG_DB_INIT".MOTOR1_SMER
// Start
A   "PRG_DB_INIT".CMD.Start  DB20.DBX0.0
AN  #Interlock
JCN A0

```

Obrázek 14. Podmínka pro přechod do kroku 1

Network: 6	Krok 1
------------	--------

```

Vyjetí magnetu nahoru

S1:  NOP  0                                //Permitted instruction

// Held
A    "PRG_DB_INIT".CMD.Hold                DB20.DBX0.1
O    #Alarm_
JCN  A1
L    10
T    "PRG_DB_INIT".STEP                    DB20.DBW12    -- Cis
                                           lo kroku
L    1
T    "PRG_DB_INIT".STEP_OLD                DB20.DBW14    -- Cis
                                           lo kroku pro restart

A1:  NOP  0

AN   "C1"                                  I24.5
R    "PRG_DB_INIT".MOTOR2_SMER            DB20.DBX24.1  -- sme
                                           r nahoru(do prava) 0 dolu
                                           (doleva) 1
S    "PRG_DB_INIT".MOTOR2_enable          DB20.DBX24.2  -- sep
                                           nuti impulsu krok. motoru

// Prechod krok 2
A    "C1"                                  I24.5
JCN  A1a
L    2
T    "PRG_DB_INIT".STEP                    DB20.DBW12    -- Cis
                                           lo kroku

A1a: NOP  0
JU   END

```

Obrázek 15. Krok 1

Network: 7	Krok 2
------------	--------

Posu jerabu doprava	
---------------------	--

```

S2:  NOP  0

      AN  "senzory".SENZOR1                DB3.DBX0.0
      R  "PRG_DB_INIT".MOTOR1_SMER        DB20.DBX24.0    -- sme
                                           r doprava 0 doleva 1
      S  "PRG_DB_INIT".MOTOR1_enable      DB20.DBX24.3    -- nas
                                           taveni rychlosti

      AN  "C1"                              I24.5
      -  "PRG_DB_INIT".MOTOR2_enable      DB20.DBX24.2    -- sep
                                           nuti impulsu krok. motoru

// Held
      A  "PRG_DB_INIT".CMD.Hold            DB20.DBX0.1
      O  #Alarm
      JCN A2
      L  10
      T  "PRG_DB_INIT".STEP                DB20.DBW12      -- Cis
                                           lo kroku
      L  2
      T  "PRG_DB_INIT".STEP_OLD            DB20.DBW14      -- Cis
                                           lo kroku pro restart

A2:  NOP  0

// Prechod krok 3
      A  "senzory".SENZOR1                DB3.DBX0.0
      JCN A2a

      R  "PRG_DB_INIT".MOTOR1_enable      DB20.DBX24.3    -- nas
                                           taveni rychlosti
      R  "PRG_DB_INIT".MOTOR2_enable      DB20.DBX24.2    -- sep
                                           nuti impulsu krok. motoru
      S  "PRG_DB_INIT".STA.Done           DB20.DBX2.5
      L  0
      T  "PRG_DB_INIT".STEP                DB20.DBW12      -- Cis
                                           lo kroku

A2a: NOP  0

      JU  END

```

Obrázek 16. Krok 2

```

Network: 9      Held
-----

HBLD: NOP      0

      SET
      R      "PRG_DB_INIT".MOTOR2_enable DB20.DBX24.2      -- sepnuti impulzu krok. motoru
      R      "PRG_DB_INIT".MOTOR1_enable DB20.DBX24.3      -- nastaveni rychlosti
// Restart
      A      "PRG_DB_INIT".CMD.Restart   DB20.DBX0.2
      JCN    HLD1
      L      11
      T      "PRG_DB_INIT".STEP          DB20.DBW12        -- Cislo kroku
HLD1: NOP      0

// Abort
      A      "PRG_DB_INIT".CMD.Abort     DB20.DBX0.3
      JCN    HLD2
      L      0
      T      "PRG_DB_INIT".STEP          DB20.DBW12        -- Cislo kroku
HLD2: NOP      0

      JU     END

Network: 9      Restart
-----

RSTR: NOP      0

      L      "PRG_DB_INIT".STEP_OLD     DB20.DBW14        -- Cislo kroku pro restart
      T      "PRG_DB_INIT".STEP        DB20.DBW12        -- Cislo kroku

      JU     END

Network: 10     End
-----

END: NOP      0

// Reset prikazu
      L      0
      T      DB20.DBW      0

```

Obrázek 17. Ošetření stavů Held, Restart a Aborted

4.1.4 Program přenášení plechovek

Program provozu se skládá z deseti kroku je vepsán, stejně jako program inicializace, do jump listu (listu skoků). Poslední dva kroky ošetřují příkazy Held, Restar a Abort.

V kroku 0 (Obrázek 18) čekáme na příkaz start, pokud přijde a nejsou přítomny blokace pokračuje program kroky pro horizontální pohyb na zdroj (krok1 Obrázek 19), vertikální pohyb na předmět (krok2), sepnutí magnetu (krok3), pohybem vzhůru (krok4) dále pak vodorovným pohybem na cíl (krok5), spuštěním lana (krok6), deaktivaci magnetu (krok7) a vyjetím lana do horní pozice (krok8).

```

Network: 4      Řídící oddíl

L      "PRG_DB_PROVOZ".STEP // Load jump destination number DB21.DBW12      -- Cislo k
                               into ACCU 1-L-L.      roku
JL     LSTX                  // Jump destination if ACCU 1-L-L > 3.
JU     S0                    // Idle
JU     S1                    // vodor. pohyb nad zdroj
JU     S2                    // svisly pohyb na predmet
JU     S3                    // aktivace magnetu
JU     S4                    // pohyb vzhuru
JU     S5                    // pohyb na cil
JU     S6                    // svisly pohyb na predmet
JU     S7                    // deaktivace magnetu
JU     S8                    // pohyb vzhuru
JU     END                   // 9 - rezerva
JU     HELD                  // 10 - Held
JU     RSTR                  // 11 - Restart

LSTX: JU     END

Network: 5      Ready, interlocked

S0:   NOP      0              //Permitted instruction
      R      "PRG_DB_PROVOZ".MAGNET      DB21.DBX24.4      -- ak
                                       tivace/deaktivace magnet
      R      "PRG_DB_PROVOZ".MOTOR2_enable      DB21.DBX24.2
      R      "PRG_DB_PROVOZ".MOTOR1_SMER      DB21.DBX24.0
      L      0
      T      "param".rychlost      DB10.DBW0

// Start
A      "PRG_DB_PROVOZ".CMD.Start      DB21.DBX0.0
AN     #Interlock
JCN    A0
L      1
T      "PRG_DB_PROVOZ".STEP      DB21.DBW12      -- C1
                                       slo kroku

A0:   NOP      0
      JU     END

```

Obrázek 18. Krok 0

```

Network: 6      Krok 1
-----
vodor. pohyb nad zdroj
-----
S1:  NOP  0                                //Permitted instruction

// Held
A    "FRG_DB_PROVOZ".CMD.Hold              DB21.DEX0.1
O    #Alarm
JCN  A1
L    10
T    "FRG_DB_PROVOZ".STEP                  DB21.DEW12    --
                                           slo kroku
L    1
T    "FRG_DB_PROVOZ".STEP_OLD             DB21.DEW14    --
                                           slo kroku pro restart
A1:  NOP  0

// Act pozice = pozadovana
A(
L    "GLOBAL_DATA".ACT_POS_HOR            DB2.DEW0
L    "FRG_DB_PROVOZ".START_POS           DB21.DEW20    --
                                           artovaci pozice
--I
)
JCN  y1

R    "FRG_DB_PROVOZ".MOTOR1_enable        DB21.DEX24.3
L    2
T    "FRG_DB_PROVOZ".STEP                DB21.DEW12    --
                                           slo kroku
y1:  NOP  0

// Act pozice < pozadovana
L    "GLOBAL_DATA".ACT_POS_HOR            DB2.DEW0
L    "FRG_DB_PROVOZ".START_POS           DB21.DEW20    --
                                           artovaci pozice
<I
)
JCN  k1
SET
S    "FRG_DB_PROVOZ".MOTOR2_enable        DB21.DEX24.2
S    "FRG_DB_PROVOZ".MOTOR2_SMER         DB21.DEX24.1    --
S    "FRG_DB_PROVOZ".MOTOR1_SMER        DB21.DEX24.0
S    "FRG_DB_PROVOZ".MOTOR1_enable        DB21.DEX24.3
k1:  NOP  0

// Act pozice > pozadovana
A(
L    "FRG_DB_PROVOZ".START_POS            DB21.DEW20    --
                                           artovaci pozice
L    "GLOBAL_DATA".ACT_POS_HOR            DB2.DEW0
<I
)
JCN  k2

SET
S    "FRG_DB_PROVOZ".MOTOR2_enable        DB21.DEX24.2
R    "FRG_DB_PROVOZ".MOTOR2_SMER         DB21.DEX24.1    --
R    "FRG_DB_PROVOZ".MOTOR1_SMER        DB21.DEX24.0
S    "FRG_DB_PROVOZ".MOTOR1_enable        DB21.DEX24.3
k2:  NOP  0

JU   END

```

Obrázek 19. Krok 1

5. SPOJENÍ SIMATIC A INTOUCH

Pro komunikaci mezi systémy, musíme všechny proměnné, které budeme chtít zobrazit nebo zadat v systému InTouch, v SIMATIC Manager zapsat do datového bloku.

5.1.1 Vytvoření komunikace

Program v SIMATIC Manageru otevřeme a modifikujeme *Hardware Configuration*. Připojujeme PLC na síť Profičet pomocí komunikačního procesoru CP.

1. Dvojklik na komunikační procesor, ten se otevře a hlásí *not networked*. Klikneme na *Properties* a vytvoříme novou síť Ethernet. V ní zadáme IP komunikačního procesoru CP. Pak OK. Provedeme kompilaci a *download*. CP musí být ve stavu RUN.
2. Spustíme System Management Console. Zde je DAServer Manager, označíme ArchestrA.DASSIDirect.1 a aktivujeme server.
3. Spustíme InTouch.
4. V development verzi se musí vytvořit Access Name. Postup je následující: **Speciál**→ **Access Name**→ **Add** →**zadat název**.

6. INTOUCH

InTouch software je výkonný systém pro vytváření aplikací pro sběr dat, vizualizaci, sledování a řízení technologických procesů. Umožňuje vytvořit grafické zobrazení výrobních technologií na PC, jejich ovládání a dynamické animace, které v reálném čase zobrazují aktuální stavy systémů.[1]

InTouch patří mezi systémy SCADA/HMI (Supervisory Control and Data Acquisition / Human-Machine Interface) systémy pro vizualizaci a supervizní řízení výrobních technologií a procesů.

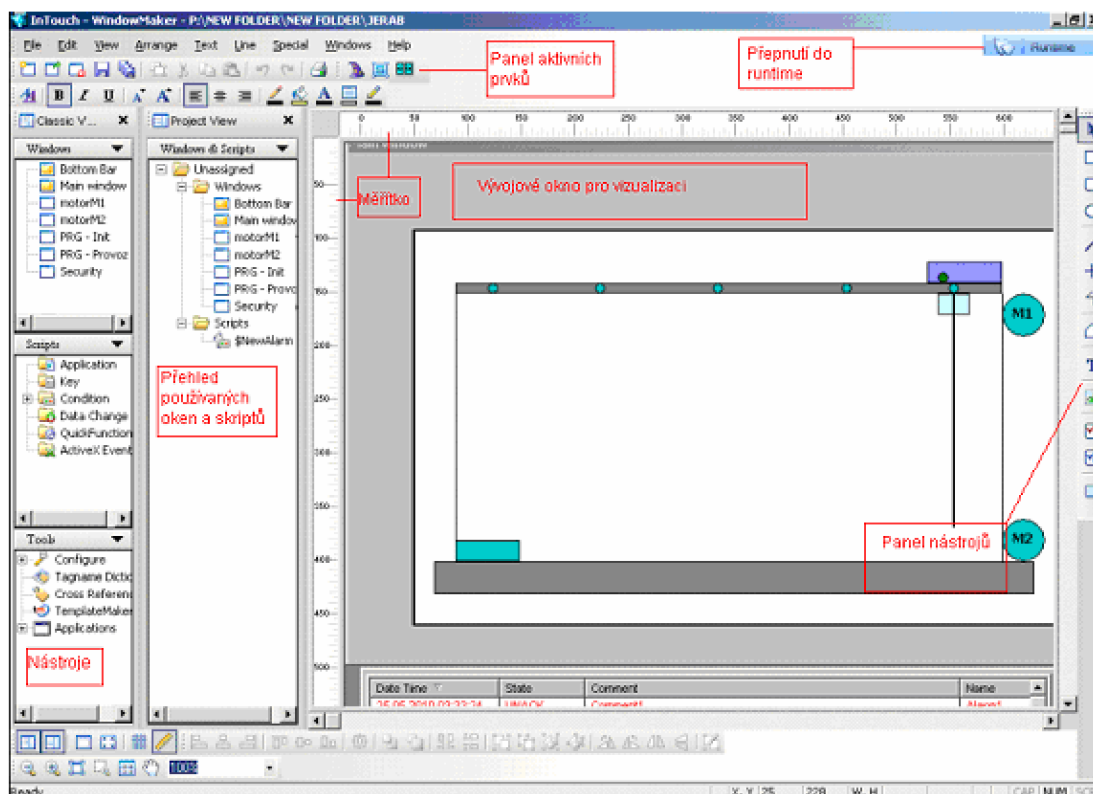
SCADA – Supervisory Control And Data Acquisition, Systém shromažďující v reálném čase data z čidel v provozu a posílající je na centrální počítač pro další zpracování a řízení. SCADA systém obsahuje vstupně-výstupní hardware, regulátory, HMI, síť, komunikace, databáze a software.[4]

HMI – Human Machine Interface, dříve MMI (Man Machine Interface) Software (typicky s grafickým uživatelským prostředím – GUI), zobrazující operátorovi informace o stavu procesu a umožňující zadávat operátorské povely (příkazy). [4]

6.1 ČLENĚNÍ SYSTÉMU

InTouch je složen ze tří hlavních programů, Intouch Application Manager, WindowMaker a WindowViewer.

- Intouch Application Manager: Organizace projektů vyvíjených v systému InTouch. Umožňuje vytvářet nové aplikace, nastavovat jejich vlastnosti. Spouští programy Window Maker a WindowViewer.
- WindowMaker: Vývojové prostředí.
- WindowViewer: Běh zkonfigurované aplikace v reálném čase.



Obrázek 20. Vývojové okno WindowMaker

6.2 VIZUALIZACE

Vizualizační program slouží pro zobrazování aktuálního stavu a ovládání technologie. Pro vlastní správu technologie (ovládání jeřábu) není chod počítače potřebný, ale jeho trvalý běh je nezbytný pro ovládání, vizualizaci i archivaci.

6.2.1 Vytvoření nové proměnné

Spustíme *Tagname Dictionary* (Obrázek 21), který nalezneme buď v bloku nástrojů *Tools*, a nebo přes *Special/Tagname Dictionary*.

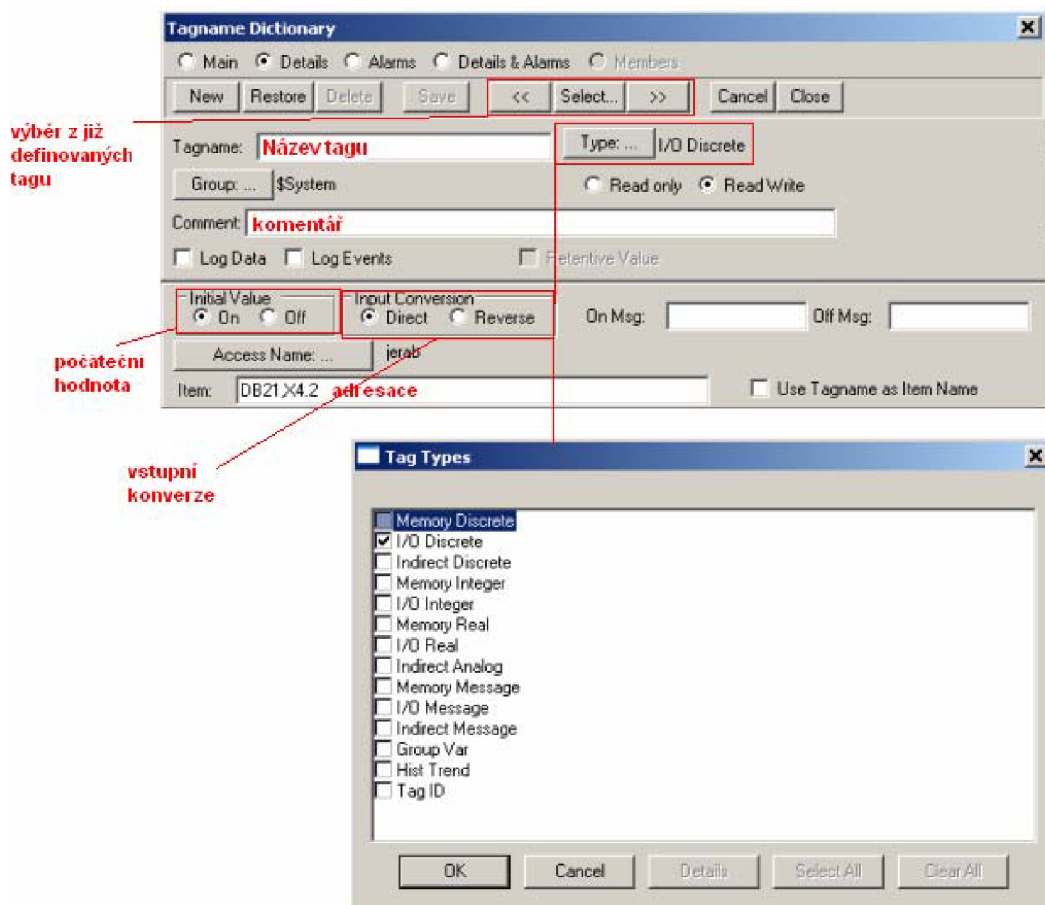
Pro vytvoření nového tagu zvolíme *New*. Nyní můžeme zadat název a typ tagu. Můžeme vybrat z několika základních typů, diskrétní, celočíselné, reálné, textové a skupinové proměnné. Podle zvoleného typu se odvíjí další nastavení proměnné, na Obrázku 21 byl zvolen typ *I/O Discrete*.

Dále můžeme napsat komentář, zvolit počáteční hodnotu, vstupní konverzi a nebo přidělit proměnné schopnost čtení **Read Only** či čtení a zápisu **Read Write** při běhu programu.

Nejdůležitějším pro funkci proměnné a komunikaci mezi SIMATIC Managerem a InTouchem je zadat správnou adresu, pod kterou je žádaná proměnná definována v programu. Adresu můžeme zjistit přímo v programovém bloku, pokud máme zapnuté Symbol Information. Adresa tagu se skládá z názvu datového bloku (DB), ve kterém je napsaná a dále pak z typu proměnné (X pro diskretní hodnoty BOOL, W pro WORD atd.) a její adresy. Např. **DB1.BDX0.0**.

Takto vytvořenou proměnnou může přiřadit k vizualizaci určitého objektu.

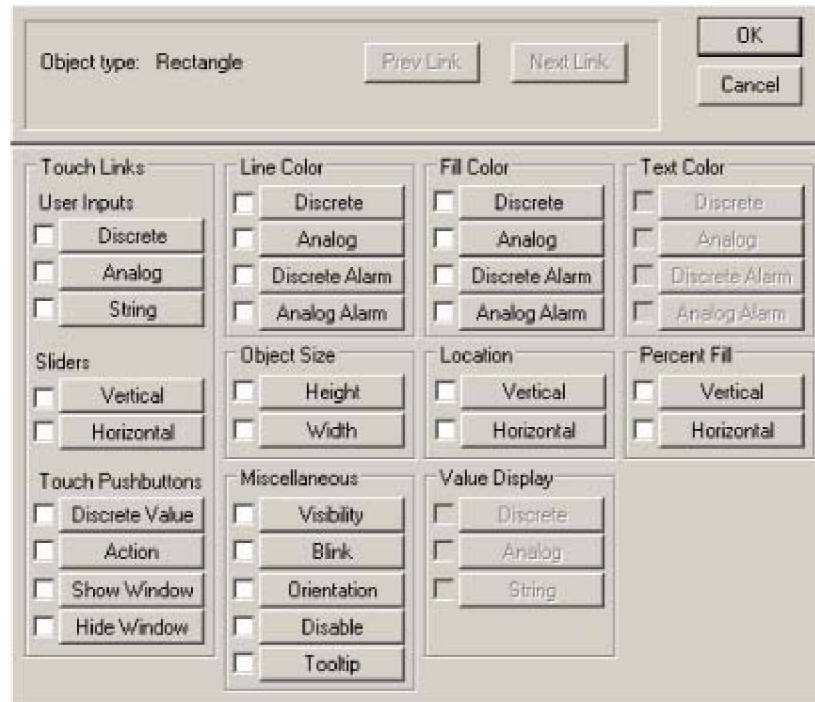
Pro každý typ specifikované proměnné, existuje specifické dialogové okno podrobností, které se používá pro definici podrobností typu proměnné.



Obrázek 21. Nejdůležitější prvky v Tagname Dictionary (databáze proměnných)

6.2.2 Vytváření animačních propojení

Pro vizualizaci musíme k nakreslenému či vloženému objektu přiřadit proměnnou. To uděláme, dvojklikem na daný objekt, tím se zobrazí dialogové okno.



Obrázek 22. Dialogové okno

Zde můžeme vybrat z možností:

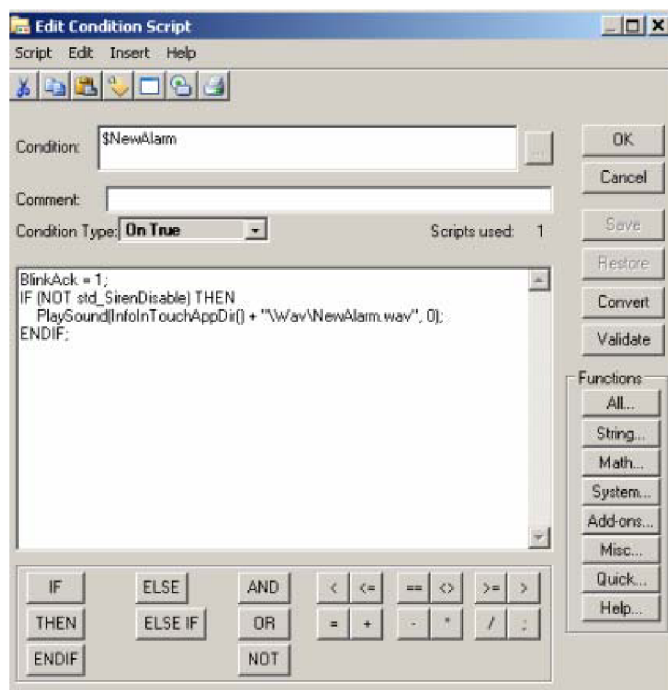
- **Touch Links (Dotyková propojení):** použití pro objekty nebo symboly, které chceme, aby byly „citlivé na stisknutí“. Dotyková propojení jsou rozdělena do tří skupin User Inputs (vstupy uživatele), Sliders (posuvníky) a Touch Pushbuttons (dotyková tlačítka).
- **Line Color (Barva čáry):** pro animování atributů nějakého objektu
- **Fill Color (Barva výplně):** propojením barvy, existují čtyři typy barvy
- **Text Color (Barva textu):** výplně a textu
- **Object size (Velikost objektu):** pro změnu výšky a nebo šířky objektu podle

- Location (Umístění): analogové hodnoty
automatické přesunutí objektu vodorovně, svisle nebo v obou směrech dle dané analogové hodnoty
- Percent Fill (Procentuální výplň): změna stupně vyplnění útvaru
- Miscellaneous (Různé): zde můžeme nastavit viditelnost, blikání, orientaci a vypnutí funkce objektu
- Value Display (Zobrazení hodnoty): umožňuje schopnost použít textový objekt pro zobrazení hodnoty

6.2.3 Vytváření skriptů QuickSkripts

Skriptování programu InTouch je funkce z nejširším použitím. Skripty umožňují provádět příkazy a logické operace na základě splnění určitých kritérií.






Použitím skriptů lze vytvořit četné množství uživatelských funkcí, které může systém využívat.



Obrázek 23. Příklad skriptu

6.2.4 Vytvořený ovládací panel

Date Time	State	Comment	Name
25.05.2010 03:36:46	UNACK	Nouzove tlačitko	PRG_PROVOZ_Alarm1
25.05.2010 03:36:46	UNACK	Nouzove tlačitko	PRG_INIT_Alarm1

Alarm
Ack

INIT

PROVOZ

Obrázek 24. Ovládací panel

Alarmy - v okně se zobrazují aktivní alarmy.

Alarm Ack - tlačítko pro potvrzení poruchy, po rozpoznání poruchy začne tlačítko červeně blikat

INIT - po stisku vyskočí pop-up okno PRG-INIT pro ovládání inicializace

PROVOZ - stiskem tlačítka provoz vyskočí pop-up okno PRG-PROVOZ



Ovládání zvuku – vypne nebo zapne zvuk



Kalkulačka – spustí kalkulačku



NotePad – spustí poznámkový blok



Přihlášení uživatele – spustí okno pro přihlášení



Odhlášení uživatele – odhlásí uživatele

Administrator **Uživatel** – jméno uživatele

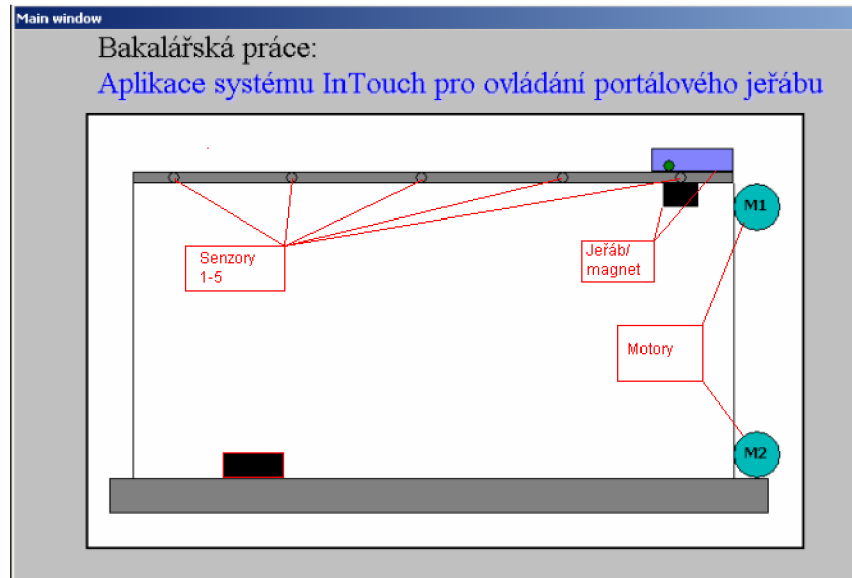


Security – spustí okno security pro konfiguraci uživatele a změnu hesla

6.2.5 Okno jeřábu

Jeřáb se horizontálně pohybuje po panelu s senzory 1 až 5 a je ovládán tagem **poloha**. Při vertikálním pohybu je znázorněno odvíjení a navíjení lana řízený proměnnou **up**. Magnet po aktivaci změní barvu z černé na zelenou a je řízen proměnnou **magnet**.

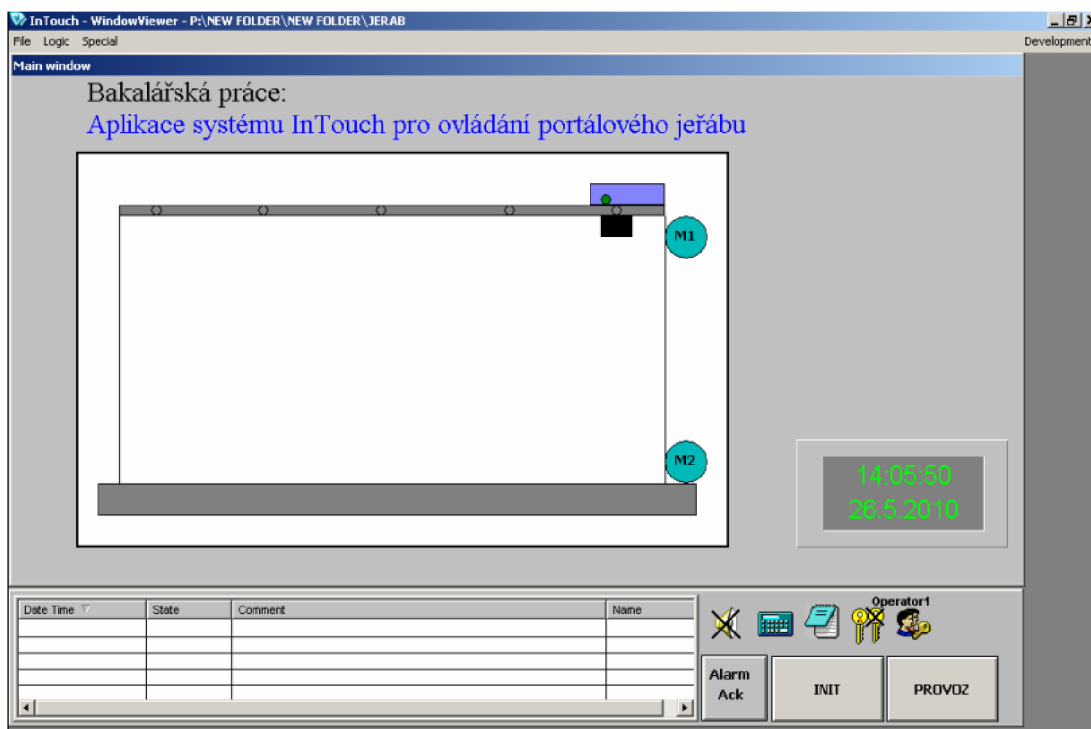
M1 a M2 jsou objekty zastupující funkci motorů.



Obrázek 25. Okno vizualizace jeřábu

6.3 OVLÁDÁNÍ PROGRAMU

Spuštěním aplikace v reálném čase, se objeví na monitoru okno k ovládání jeřábu Obrázek 26.



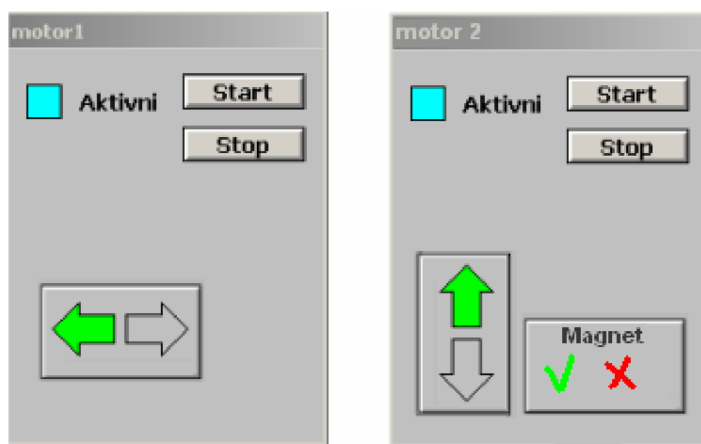
Obrázek 26. Okno ovládání jeřábu

6.3.1 Alarmy

V dolním okně se zobrazují výstražná hlášení, alarmy. Spuštění alarmu může dojít stiskem nouzového tlačítka přímo na ovládacím panelu jeřábu, nebo jsou-li přítomny nějaké blokace (např. není-li zadána zdrojová či cílová poloha). Pro deaktivaci alarmu musí být alarm nejdříve potvrzen tlačítkem Alarm Ack.

6.3.2 Ruční ovládání jeřábu

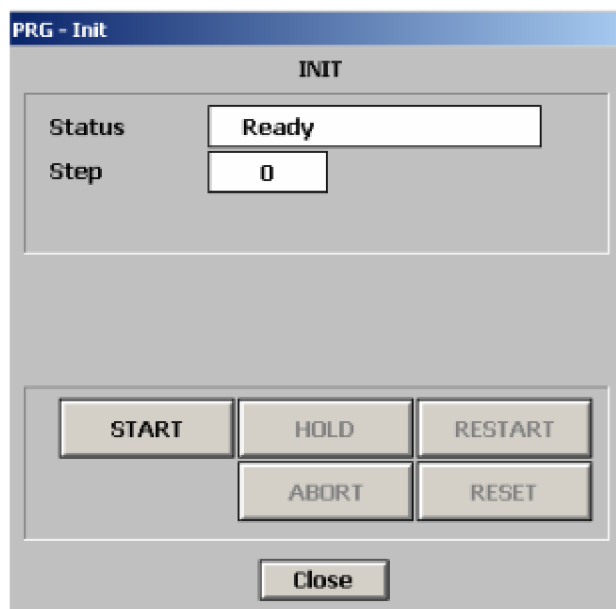
Chceme-li jeřáb ovládat ručně, stiskem myši vybereme motor který chceme ovládat M1 nebo M2 Obrázek 27. Motor M1 slouží k horizontálnímu posuvu jeřábu a motor M2 slouží k vertikálnímu pohybu, v okně řízení M2 je také panel pro aktivaci a deaktivaci magnetu. V obou oknech jsou tlačítka pro aktivaci a deaktivaci motorů, to je zde také vizualizováno, aktivitu indikuje modrá barva u nápisu aktivní. Směr pohybu motorů řídíme klikáním na šipky, zelená šipka signalizuje aktuální směr.



Obrázek 27. Ovládání motorů M1 a M2

6.3.3 Ovládání inicializace

Pro spuštění inicializace jeřábu, kliknem na tlačítko INIT, na monitoru se objeví okno pro řízení inicializace Obrázek 28.

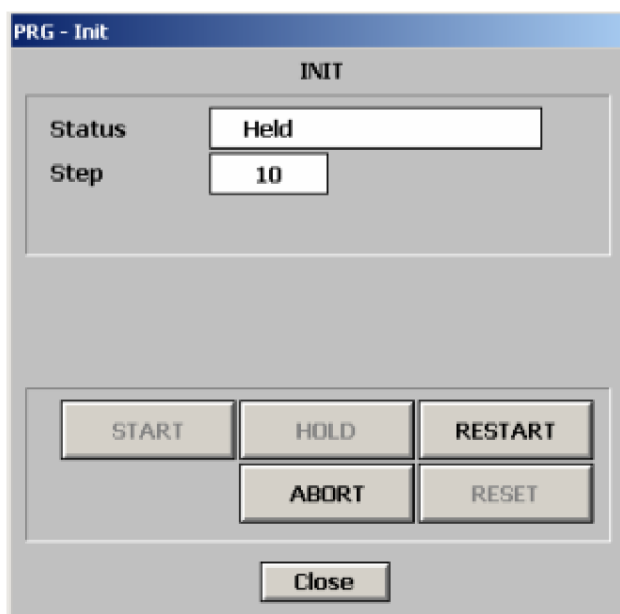


Obrázek 28. Okno ovládání inicializace

Status určuje v jakém stavu se inicializace nachází, může zde nastat sedm stavů, viz. Obrázek 7.

Step ukazuje v jakém kroku se právě inicializace nachází. Tady existují tři kroky. Krok 0 program je ve stavu Ready a čeká na příkaz start. Pokud jsou přítomny nějaké blokace (Interlocky) pak je tlačítko start disable (neaktivní). Do kroku 1 se přepneme stiskem Start, program bude ve stavu Running. Nyní se navijí lano a magnet směrem vzhůru. Jestliže přijde požadavek Hold pak se step nastaví do kroku 10 a status do Held Obrázek 29 , zde program vyčkává na další příkaz. Abort přeruší program, nastaví status Aborted, a po stisknutí Reset se vrátíme na začátek. Požadavkem Restart se navrátíme zpět do kroku 1. Krok 2 nastane, jsou-li splněny podmínky pro přechod do dalšího kroku. V tomto kroku se jeřáb přesune do polohy 1 (vpravo). Je-li v této poloze je program ukončen, status se nastaví na Done.

Pro opětovné spuštění programu musíme nejprve stisknou reset.



Obrázek 29. Inicializace ve stavu Held

6.3.4 Ovládání provozu jeřábu

Stiskem na tlačítko PROVOZ se otevře podobné okno jako pro inicializaci. V tomto oknu jsou navíc kolonky pro zadávání zdrojové a cílové pozice plechovky. Podle počtu senzorů a tedy poloh jeřábu můžeme zadat pozici 1-5. Počet kroků v tomto programu je 8.



Obrázek 30. Okno ovládání provozu

6.3.5 Security

Pro ovládání operátorem nemusí být uživatel přihlášen, pro spuštění inicializace a pro spuštění provozu, se musí uživatel přihlásit a to buď pod administrátorem a nebo pod operátorem Obrázek 31.

Standardně je v systému InTouch nastaven administrátor pod jménem Administrator a heslem Wonderware.

Dalším nastaveným uživatelem v programu je Operator1 s heslem 1111. Všeobecně by měl mít administrátor větší přístupová práva než běžný uživatel. To lze nastavit při konfiguraci.



Obrázek 31. Přihlášení uživatele



Obrázek 32. Okno konfigurace uživatelů

7. ZÁVĚR

Hlavním úkolem této bakalářské práce bylo zhotovení softwaru pro ovládání a vizualizaci modelu portálového jeřábu pomocí systému InTouch. Dalším úkolem bylo vytvoření funkčního programu v jazyce STEP 7.

Při tvorbě bakalářské práce jsem se naučila ovládání a práci se systémem InTouch. Dále jsem se naučila propojit systém InTouch se systémem SIMATIC Manager.

Při popisu programu vytvořeném v jazyce STEP 7 jsem popsala jen některé části programu, protože je program velice obsáhlý.

Celý program vytvořený jak v SIMATIC Manageru tak v systému InTouch a text bakalářské práce ve formátu pdf je uložen na CD, které je zde vloženo jako příloha.

8. POUŽITÁ LITERATURA

- [1] PANTEK (CS) S.R.O. , Pantek [online]. 2007-2009 [cit. 2010-05-30].
Wonderware software. Dostupné z WWW:
< <http://www.pantek.cz/produkty/wonderware-software/> >.
- [2] KOSEK, Rostislav. Řídicí systémy Simatic pro moderní automatizaci.
Automa : Časopis pro automatizační techniku [online]. 2005, 02,
[cit. 2010-05-30]. Dostupný z WWW:
< http://www.odbornecasopisy.cz/index.php?id_document=30312 >.
- [3] MARTINÁSKOVÁ, Marie . *PLC -programovatelné automaty* [online].
6.12.2001 [cit. 2010-05-30]. Dostupné z WWW:
< http://iat.fs.cvut.cz/109/files/psar/prednasky_2007/plc_vznik_hw_sw.ppt >.
- [4] BALDA, Pavel. *Vendulka.czu* [online]. 20. 04. 2007 [cit. 2010-05-30].
Informační a řídicí systémy I. Dostupné z WWW:
< http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-08_SCADA_HMI.pdf >.

9. SEZNAM OBRÁZKŮ

Obrázek 1. Moduly S7-300	12
Obrázek 2. CPU jednotka S7-300	13
Obrázek 3. Hardwarová konfigurace	15
Obrázek 4. Foto jeřábu	17
Obrázek 5. Jeřáb s označením prvků	17
Obrázek 6. SIMATIC Manager	19
Obrázek 7. Stavový diagram hlavních funkcí	20
Obrázek 8. Datový blok DB	21
Obrázek 9. UDT	21
Obrázek 10. Adresy prvků	21
Obrázek 11. Program ovládání DC motoru	22
Obrázek 12. Ovládání krokového motoru	23
Obrázek 13. Proměnné inicializace	24
Obrázek 14. Podmínka pro přechod do kroku 1	24
Obrázek 15. Krok 1	25
Obrázek 16. Krok 2	26
Obrázek 17. Ošetření stavů Held, Restart a Aborted	27
Obrázek 18. Krok 0	28
Obrázek 19. Krok 1	29
Obrázek 20. Vývojové okno WindowMaker	32
Obrázek 21. Nejdůležitější prvky v Tagneme Dictionary (databáze proměnných) ...	33
Obrázek 22. Dialogové okno	34
Obrázek 23. Příklad skriptu	35
Obrázek 24. Ovládací panel	36
Obrázek 25. Okno vizualizace jeřábu	37
Obrázek 26. Okno ovládání jeřábu	38
Obrázek 27. Ovládání motorů M1 a M2	39
Obrázek 28. Okno ovládání inicializace	39
Obrázek 29. Inicializace ve stavu Held	40

Obrázek 30. Okno ovládání provozu	41
Obrázek 31. Přihlášení uživatele.....	41
Obrázek 32. Okno konfigurace uživatelů.....	42

10. SEZNAM TABULEK

tabulka 1: Vstupy PLC	15
tabulka 2: Výstupy PLC	16

11. SEZNAM PŘÍLOH

11.1 PŘILOŽENÉ DVD

11.1.1 Funkční program v SIMATIC Manageru

11.1.2 Funkční program v systému InTouch

11.1.3 Dokumentace v elektronické podobě