

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra řídicí techniky

Diplomová práce

Řízení a vizualizace technologického procesu pomocí PLC

Vypracoval: Tomáš Henych

Prohlášení

Prohlašuji, že jsem svou diplomovou (bakalářskou) práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW, atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

.....

podpis

Abstrakt

Cílem této práce je obeznámit čtenáře s problematikou návrhu a řízení technologického modelu. Jedná se o model tří paralelních vzduchových výtahů. Řízení dvou z nich je realizováno digitálně a jednoho analogově. U digitálního řízení se využívá zpětné vazby pouze od čidel v patrech. Také řízení pohybu je pouze binárním výstupem (nahoru, dolů). Stav, kdy výtah stojí, je dosaženo rychlým přepínáním výstupu. U analogového řízení využíváme zpětné vazby od ultrazvukového čidla, čímž určíme pozici přesněji. Akční veličina je také analogová, což vede na možnost využití PID regulátoru. Model je řízen průmyslovým automatem Simatic 315-2DP od firmy Siemens. Vizualizace je vytvořena v programu InTouch od firmy Wonderware.

Abstract

The purpose of this work is acquaint readers with problems of design and control of technological model. It is model of three parallel air elevators. Two of them are digitally controlled and one is analog controlled. Feedback of digital control is only from optical sensor placed at floors. The control is realized by digital output (up, down). Position freeze is due to quickly switching of output. Feedback of analog control is from ultrasonic sensor. The advantage of this solution is better position. Output value is analog too. We can use a PID controller. Model is connected to programmable logical controller Simatic 315-2DP from Siemens. Visualization is made in InTouch from Wonderaware.

1. Úvod.....	1
1.1 Obecný úvod	1
1.2 Popis výtahu.....	1
2. Hardware.....	3
2.1 Filozofie návrhu	3
2.2 Zjednodušené schéma zapojení.....	3
2.3 Generátor trojúhelníkového průběhu	4
2.3.1 Návrh komparátoru s hysterezí	5
2.3.2 Návrh invertujícího integrátoru s OZ.....	7
2.4 Napěťový dělič s komparátory.....	8
2.5 Řídící logika.....	10
2.6 Výkonový stupeň	11
2.7 Plošný spoj	13
2.8 Napěťové přizpůsobení	13
2.9 Ovládací panel	15
2.10 Optické čidlo a ultrazvukový senzor.....	18
3. Identifikace systému	21
3.1 Fyzikální model	21
3.2 Identifikace ARX modelu	22
3.2.1 ARX model	22
3.2.2 Nejmenší čtverce.....	23
3.3 Identifikovaná soustava	24
3.4 Regulátor.....	26
4. Simatic	32
4.1 Teoretická část	32
4.1.1 Hardware.....	32
4.1.2 Programové vybavení	33
4.2 Praktická část	37
4.2.1 Automat	37
4.2.2 Simatic manager	38
4.2.3 Hardwarová konfigurace.....	39
4.2.4 Logika řízení výtahu	39
4.2.5 Struktura programu	41
5. Vizualizace.....	47
5.1 Simatic net	47
5.2 DDE Server.....	48
5.3 InTouch.....	49
5.3.1 WindowMaker	49
5.3.2 WindowViewer	51
6. Závěr	53
7. Použitá literatura	54
8. Seznam obrázků.....	55
9. Seznam tabulek	56
10. Přílohy.....	57

1. Úvod

1.1 Obecný úvod

V současné době se na naší planetě neustále zvyšuje počet lidí. Z tohoto důvodu je potřeba stavět výškové budovy. Protože není výhodné a občas ani možné, aby lidé chodili do vyšších pater pěšky, vznikla potřeba je tam nějak dopravit. Toto se řeší různými způsoby. Nejpoužívanější způsob je zcela určitě výtahem.

V této práci se čtenář seznámí s kompletním návrhem a řízením modelu pneumatických výtahů.

Jedná se o model tří výtahů, který bude sloužit pro výuku studentů v předmětech Řídicí systémy a Návrh automatizovaných zařízení.

V první kapitole detailně rozebereme návrh elektronických obvodů, v druhé identifikaci systému a návrh PID regulátoru, ve třetí řízení pomocí PLC a v poslední vizualizaci.

Model se řídí pomocí PLC Simatic 315-DP2 od firmy Siemens. K identifikaci systému a návrhu PID regulátoru byl použit program Matlab. Vizualizace je vytvořena v programu InTouch od firmy WonderWare.

1.2 Popis výtahu

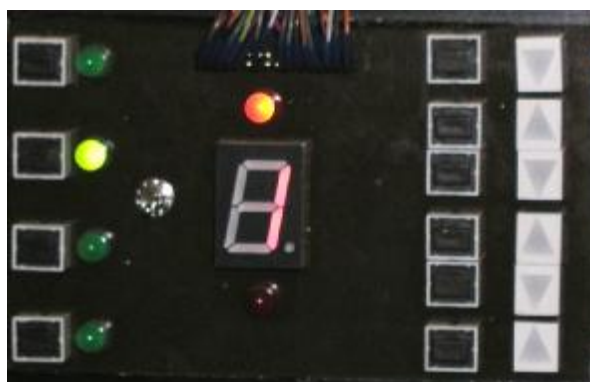
Model je sestaven ze tří plexisklových trubek zavěšených svisle (viz Obr. 1.2.1). Tím je vymezená oblast pohybu papírového válečku, který je umístěn uvnitř. Váleček představuje kabinu výtahu (viz Obr. 1.2.2). Pod každou trubkou je umístěn ventilátor podobný větráčku na procesoru počítače. Změnou otáček větráčku se mění intenzita foukání a tím se docílí pohybu kabinky nahoru a dolů. Výtah může být ovládán tlačítky u výtahu (viz Obr. 1.2.3) nebo z vizualizace. Logika řízení výtahů je navržena stejně jako u skutečných výtahů. Základní princip je zachování směru jízdy. Výtahy mohou kooperovat. To znamená, že přijede výtah, který je nejbližší.



Obr. 1.2.1 – Pohled na technologii



Obr. 1.2.2 – Detail válečku



Obr. 1.2.3 – Ovládací panel

2. Hardware

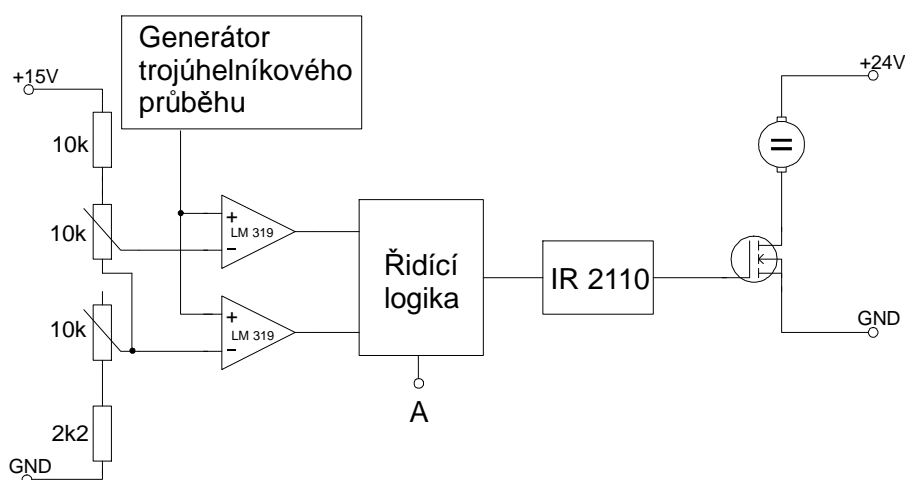
2.1 Filozofie návrhu

Dříve než se budeme věnovat samotnému návrhu elektroniky, je třeba zmínit se o způsobu realizace. Dva výtahy se řídí digitálně a jeden analogově. Digitální řízení znamená, že můžeme dát výtahu povel pouze pro pohyb nahoru nebo dolů. K tomu použijeme vstup A (viz Obr. 2.2.1). Zpětná vazba je realizována pouze čidly v jednotlivých patrech. Zastavení výtahu v patře docílíme rychlým přepínáním vstupu A. U analogové verze máme k dispozici akční zásah v plném rozsahu, také zpětná vazba je řešená analogově pomocí ultrazvukového čidla. Můžeme tedy zastavit výtah v jakékoliv pozici.

Jak již bylo zmíněno, změna směru pohybu výtahu je způsobena změnou intenzity foukání větráčku, to znamená změnou otáček. Otáčky ventilátoru jsou úměrné velikosti napětí přivedeného na vstup. Z tohoto důvodu se nabízí dvě řešení. První je vytvoření řízeného zdroje napětí. Druhá možnost je použít pulsně – šířkovou modulaci (PWM). Z důvodu jednodušší realizace použijeme druhou možnost.

2.2 Zjednodušené schéma zapojení

Nejprve uvedeme zjednodušené schéma zapojení (viz Obr. 2.2.1), které bude sloužit pro lepší orientaci v problému. Dále jednotlivé části rozebereme podrobněji.

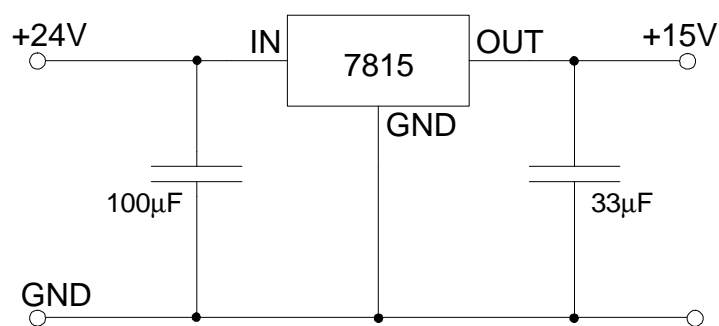


Obr. 2.2.1 – Zjednodušené schéma zapojení

Obvod lze rozdělit do čtyř částí. První část je generátor trojúhelníkového průběhu, druhá napěťový dělič s komparátory, třetí řídicí logika a čtvrtá výkonový stupeň.

Na obrázku není nakresleno napájení elektronických obvodů. Je to z důvodu přehlednosti.

Obvody jsou napájeny +15V. Model je opatřen napěťovým zdrojem 24V, 5A. Toto napětí změníme na 15V s využitím stabilizátoru napětí 7815 (viz Obr. 2.2.2).

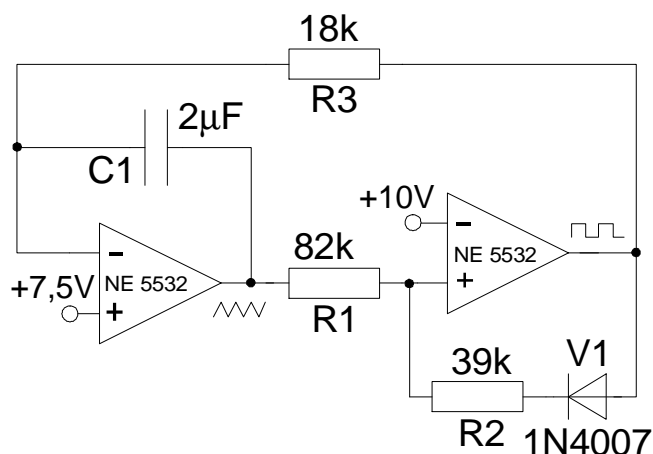


Obr. 2.2.2 – Napájení obvodů

Vstupní a výstupní stabilizační kondenzátory se při praktické realizaci umístí co nejbližší ke stabilizátoru. Pokud bychom kondenzátory nepoužili, nebo je dali příliš daleko, obvod by se rozkmital a výstupní napětí by bylo výrazně zašuměné.

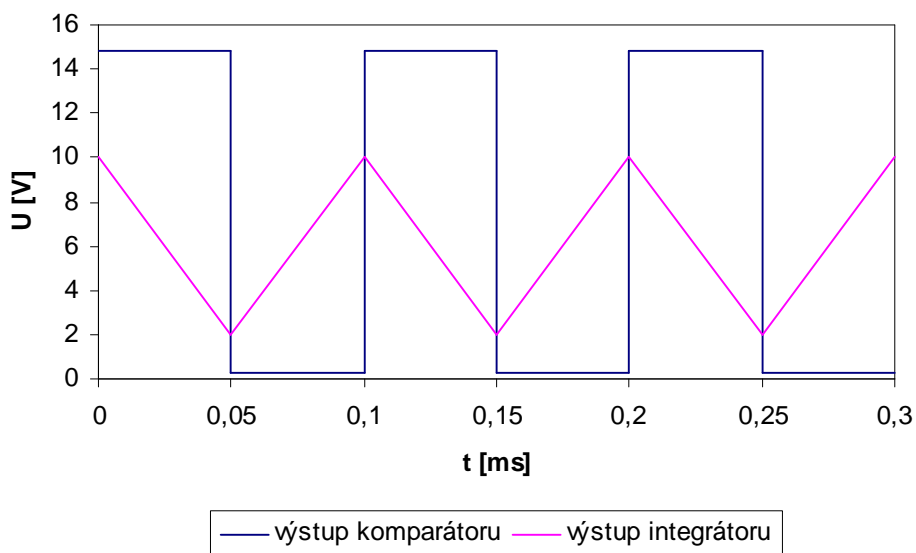
2.3 Generátor trojúhelníkového průběhu

Generátor trojúhelníkového průběhu se skládá z invertujícího integrátoru a komparátoru s hysterezí (viz Obr. 2.3.1).



Obr. 2.3.1 – Generátor trojúhelníkového průběhu

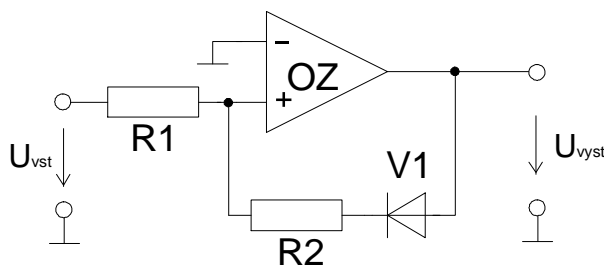
Parametry průběhu jsme zvolili následovně: frekvence 10 kHz, napětíové úrovně od 2V - 10V (viz Obr. 2.3.2).



Obr. 2.3.2 – Průběhy napětí na generátoru

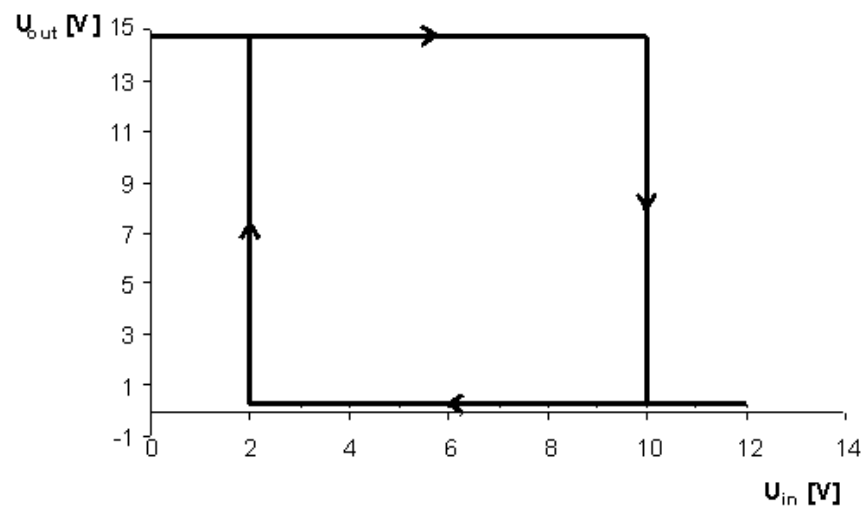
2.3.1 Návrh komparátoru s hysterezí

Komparátor porovnává napětí na vstupních svorkách operačního zesilovače. Pokud je rozdíl potenciálů mezi kladnou a zápornou svorkou kladný, na výstupu je kladné napájecí napětí (v našem případě 15V). Pokud je rozdíl záporný, je na výstupu záporné napájecí napětí (pro nás 0V). V praxi je občas potřeba, aby komparátor nepřeklápel výstup při malých změnách napětí, proto se zavádí hystereze. Toho se docílí uzavřením kladné zpětné vazby.



Obr 2.3.3 – Komparátor s hysterezí

Nejprve je třeba zvolit při jakém napětí na vstupu má komparátor změnit výstup. V tomto případě jsou to vlastně parametry trojúhelníkového průběhu. Převodní charakteristika komparátoru je na obrázku Obr. 2.3.4



Obr. 2.3.4 – Převodní char. komparátoru

Výpočet odporů

Na zápornou vstupní svorku OZ (viz Obr. 2.3.1) je nastálo přivedeno napětí 10V, což je komparační napětí. Pokud je na kladné svorce OZ napětí menší než 10V a výstup je 15V, dioda V1 je zavřená a obvod se chová jako kdyby tam zpětná vazba nebyla. Když zvýšíme vstupní napětí nad 10V obvod překlopí výstup na 0V. Dioda je polarizovaná v propustném směru a proto se otevře. Zpětná vazba je zapojená. Při snižování vstupního napětí OZ překlopí výstup, když má na vstupu menší napětí než 10V. Na vstupu je napěťový dělič. Z toho vypočteme hodnoty odporů.

$$U_+ = (U_{vyst} - U_{V1}) \times \frac{R2}{R1 + R2} \quad (2.1)$$

$$10 = (15 - 2 - 0.7) \times \frac{R2}{R1 + R2}$$

$$10R1 + 10R2 = 14.3R2$$

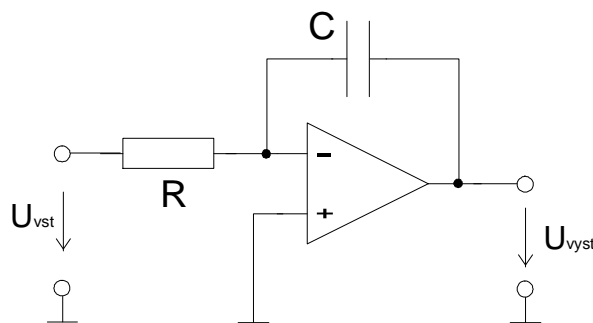
$$10R1 = 4.3R2$$

Odpor $R1 = 89k\Omega$ zvolíme a $R2 = 39k\Omega$ dopočítáme.

Odpor $R1$ i $R2$ jsou součástí řady R12 a proto je již nemusíme korigovat.

2.3.2 Návrh invertujícího integrátoru s OZ

Invertující integrátor s OZ má na invertující vstupní svorce OZ zapojený odpor a zápornou zpětnou vazbu uzavřenou kondenzátorem. Toto zapojení má výhodu, že kondenzátorem teče stále stejný proud, proto na něm lineárně vzrůstá napětí. To znamená, že je lineární průběh napětí i na výstupu, což je přesně to čeho jsme chtěli docílit.



Obr 2.3.5 – Invertující integrátor

Návrh hodnot součástek

Přenos integrátoru je:

$$P(s) = \frac{U_{vyst}(s)}{U_{vst}(s)} = -\frac{1}{sCR} \quad (2.2)$$

v časové oblasti lze napsat

$$u_{vyst}(t) = -\frac{1}{CR} \int u_{vst}(t) dt + U_{CO} \quad (2.3)$$

po integraci

$$u_{vyst}(t) = -\frac{1}{CR} u_{vst}(t) \cdot t + U_{CO} \quad (2.4)$$

$$u_{vyst} = 2, u_{vst} = 15, t = 50ms, U_{CO} = 10$$

po dosazení a úpravě

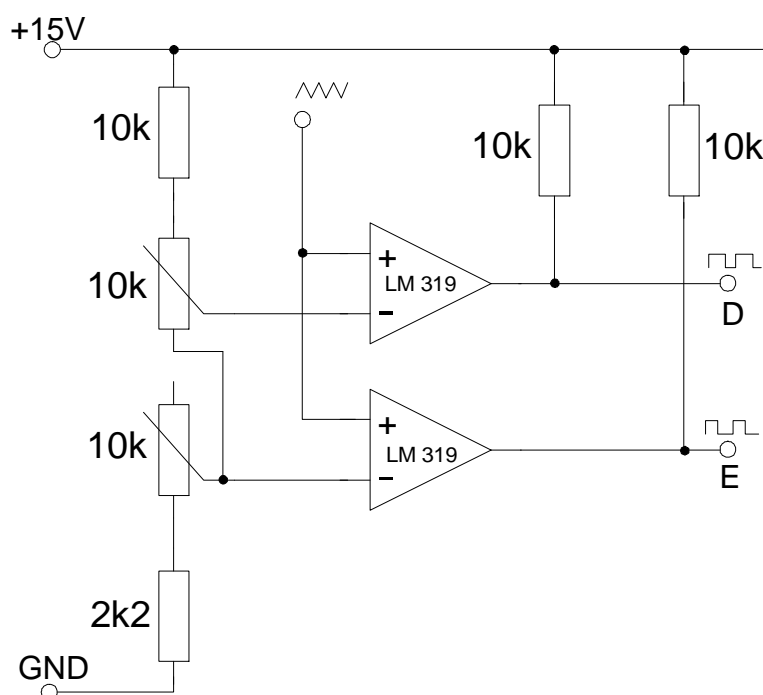
$$CR = 36.5 \times 10^{-9}, \text{ zvolíme } R = 18k\Omega, \text{ dopočteme } C = 2nF.$$

Integrátor integruje do té doby dokud nedosáhne saturačního napětí. V našem případě však ani této hodnoty nedosáhne, protože před tím komparátor změní výstupní hodnotu a integrátor začne integrovat na opačnou stranu.

2.4 Napěťový dělič s komparátory

Na napěťovém děliči nastavíme experimentálně pomocí potenciometrů napěťové úrovně odpovídající střídě signálů potřebné k pohybu výtahu nahoru a dolů.

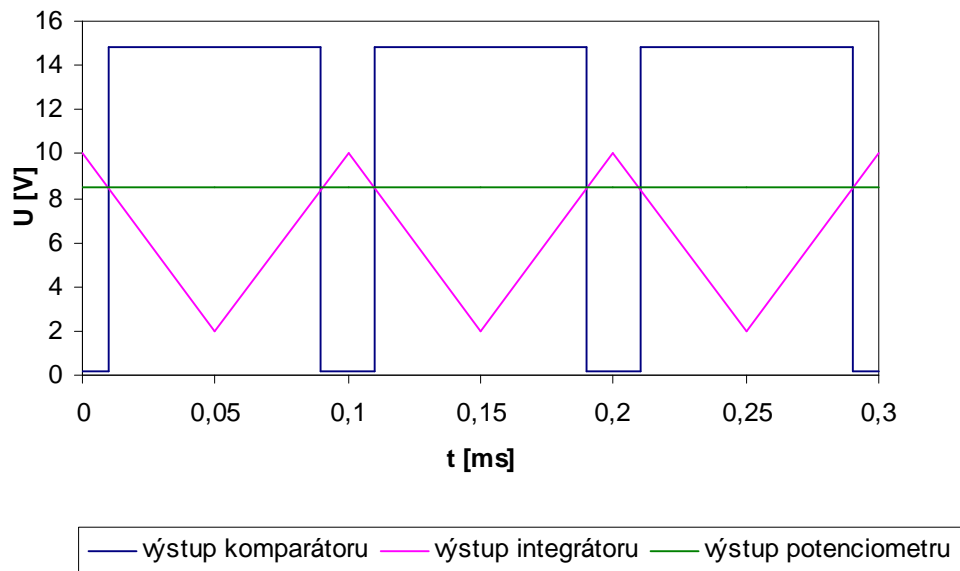
Jako komparátory použijeme obvodů LM 319, které mají výstup s otevřeným kolektorem. Z tohoto důvodu musíme dát na výstup odpor připojený proti napájecímu napětí (viz Obr. 2.4.1). Odpor nesmí být moc malý, aby jím netekl příliš velký proud. Zatěžovali bychom zbytečně stabilizátor. Nesmí být ale ani příliš velký. Odpor s výstupní cestou na plošném spoji tvoří RC článek a průběh napětí není obdélníkový. Náběžná hrana je výrazně zaoblená, čemuž se chceme vyhnout. V tomto případě bylo vyhovující použít odpor 10 k Ω .



Obr. 2.4.1 – Napěťový dělič s komparátory

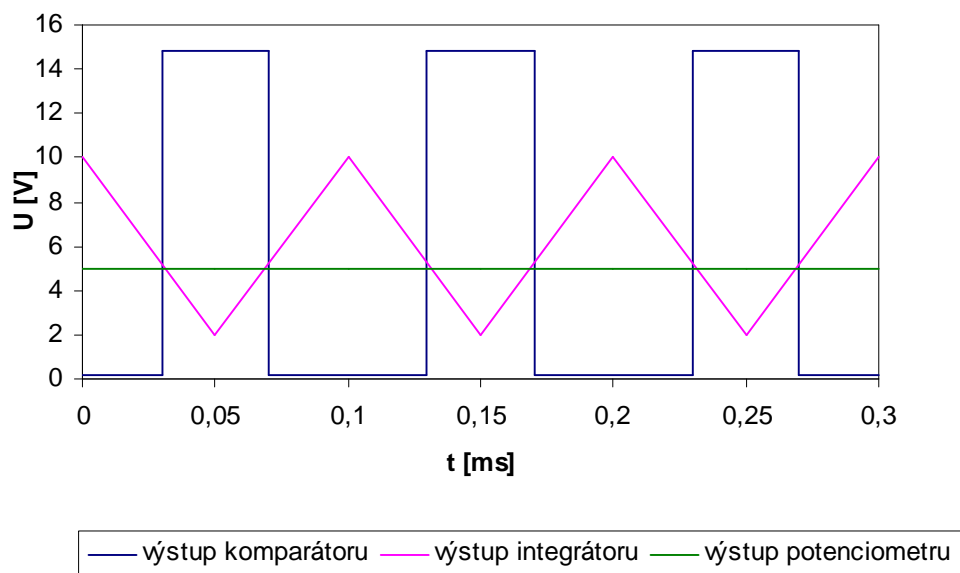
Průběhy napětí na komparátorech

Horní komparátor



Obr 2.4.2 – Průběhy napětí na horním komparátoru

Dolní komparátor



Obr 2.4.3 – Průběhy napětí na dolním komparátoru

2.5 Řídící logika

Jak již bylo výše zmíněno, změna směru jízdy výtahu se ovládá vstupem A (viz Obr. 2.2.1). Pokud je na vstupu logická nula výtah jede dolů, pokud jednička nahoru. Přepínáme pouze výstupy (D,E) komparátorů. Směru nahoru odpovídá signál horního komparátoru D, směru dolů signál druhého komparátoru E.

A	D	E	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Tab. 2.5.1 – Pravdivostní tabulka

Pravdivostní tabulku přepíšeme do Karnaughovy mapy.

		A	
		0	0
E	D	1	0
		1	1
		0	1
		0	1

Tab. 2.5.2 – Karnaughova mapa

V mapě jsme pokryli jedničky a můžeme vyčíst výraz

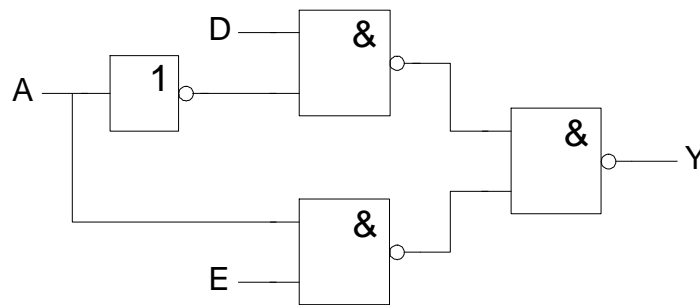
$$Y = \overline{AD} + AE \quad (2.5)$$

pomocí DeMorganových zákonů převedeme na konjunktivní normální formu

$$Y = \overline{\overline{\overline{AD}} \cdot \overline{\overline{AE}}} \quad (2.6)$$

Při tvorbě elektronického obvodu nyní můžeme použít pouze hradla typu NAND, protože tvoří úplný logický systém. Veškeré logické výrazy lze převést do tohoto tvaru. Negace se vytvoří spojením vstupů hradla. Přesto použijeme i hradla typu NOT (invertor). Elektronický obvod je přehlednější.

Výsledné elektronické zapojení:



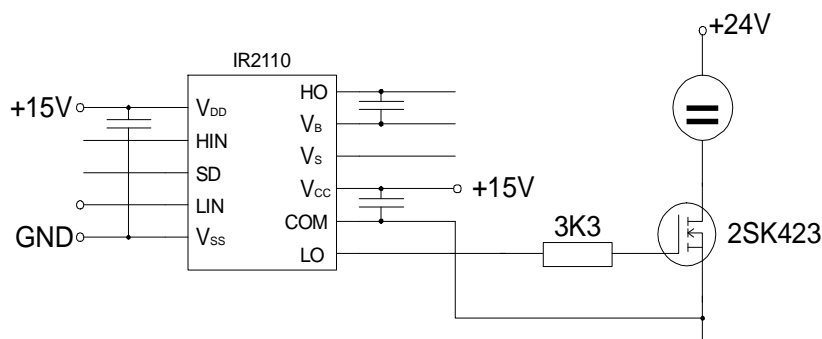
Obr 2.5.1 – Schéma logického obvodu

Jelikož jsou všechny obvody napájeny 15 V, při praktické realizaci bylo použito obvodů CMOS. Výhodou je nízký příkon, rozhodovací úroveň v polovině napájecího napětí, vyšší rychlosti změn a hlavně nebylo třeba řešit nový zdroj napájení, což by se u hradel TTL muselo.

2.6 Výkonový stupeň

Protože potřebujeme spínat motor ventilátoru, musíme vyřešit jak. Možností se nabízí více. Určitě nelze použít třeba relé, protože frekvence spínání je příliš vysoká. V našem případě je použito tranzistoru typu JFET. Tranzistor nelze připojit přímo na výstup logického hradla, protože hradlo nemá potřebný výkon. Použijeme tedy výkonový budič pro MOSFET tranzistor IR2110 od firmy Internal Rectifier.

Zapojení výkonového stupně



Obr. 2.6.1 – Schéma zapojení IR2110

IR2110:

IR2110 je vysoko napěťový, rychlý budič pro tranzistory MOSFET s kanálem typu N a IGBT. Logické vstupy jsou kompatibilní s CMOS a LSTTL výstupy. Tímto obvodem lze spínat tranzistory připojené až na napětí 600V. Obvod může spínat dva tranzistory, má dva nezávislé výstupní kanály „high“ a „low“. Tranzistor připojený na „low“ výstup se připojí proti zemi. Tranzistor připojený na „high“ výstup lze připojit i na jiné napětí, toho můžeme využít pokud nemáme k dispozici (není vyvedena) zápornou svorku spínaného zařízení (v našem případě motor).

Popis výstupů obvodu s doporučenými hodnotami :

Symbol	Popis	Min	Max
V _{DD}	napájení logické části	-0,3	V _S + 25
HIN	logický vstup pro high kanál	V _S - 0,3	V _{DD} + 0,3
SD	logický vstup pro shutdown	V _S - 0,3	V _{DD} + 0,3
LIN	logický vstup pro low kanál	V _S - 0,3	V _{DD} + 0,3
V _{SS}	zem logické části	V _{CC} - 25	V _{CC} + 0,3
V _B	napájení high kanálu, kladnější napětí	-0,3	525
HO	výstup high kanálu	V _S - 0,3	V _S + 0,3
V _S	napájení high kanálu, zápornější napětí	V _B - 25	V _B + 0,3
V _{CC}	napájení low kanálu, kladnější napětí	-0,3	25
LO	výstup low kanálu	-0,3	V _{CC} + 0,3
COM	napájení low kanálu, zápornější napětí	-0,3	V _{CC}

Tab. 2.6.1 – Popis výstupů IR2110

2.7 Plošný spoj

Potřebujeme, aby elektronické součástky držely pohromadě a abychom se v nich mohli lehce orientovat, proto použijeme plošný spoj. Je mnoho variant plošných spojů. My jsme si vybrali tu nejjednodušší. Použili jsme univerzální plošný spoj, který můžeme osadit libovolnými součástkami. Nevýhoda tohoto řešení je v náročnosti propojení součástek. Musí se provést propojovacími drátky. Další nevýhoda je, že takto vyrobený spoj není tak estetický. Samozřejmě zde jsou i výhody, jinak bychom se k tomuto řešení neuchýlili. Asi největší výhoda je, že pokud vznikne změna v návrhu zařízení, přepájíme pouze změny. Další výhodou je, že nemusíme navrhovat plošný spoj klasickou metodou. Pokud bychom chtěli udělat změnu v zapojení, museli bychom navrhnout vše znovu. Toto sebou přináší i větší finanční náklady.

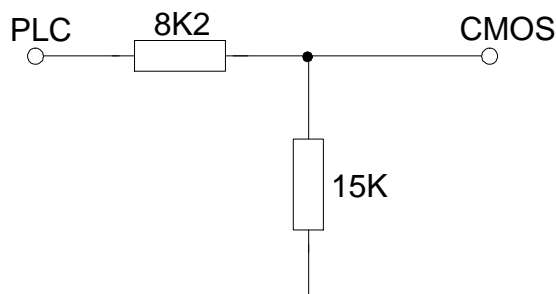
2.8 Napěťové přizpůsobení

Další problém, se kterým jsme se museli vyrovnat, bylo napěťové přizpůsobení technologií. Používáme CMOS obvody, které napájíme 15V, TTL obvody 5V a PLC pracuje s 24V.

Připojení PLC a obvody

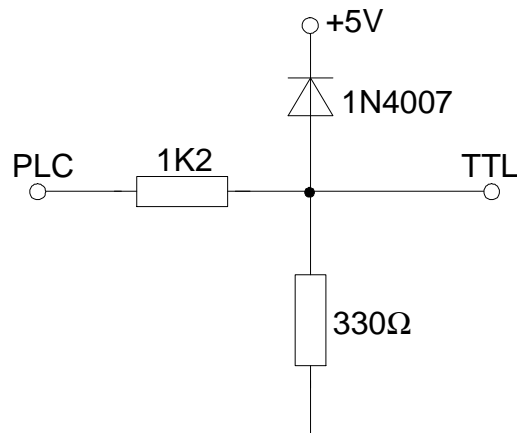
Základní problém máme, pokud je na výstupu PLC logická nula, výstup je pouze odpojen a není připojen oproti zemní sorce. Výrobce zaručuje napětí okolo 2V.

Výstup PLC pro CMOS lze vyřešit odporovým děličem (viz Obr. 2.8.1). Komparační úroveň napětí je okolo poloviny napájecího napětí.



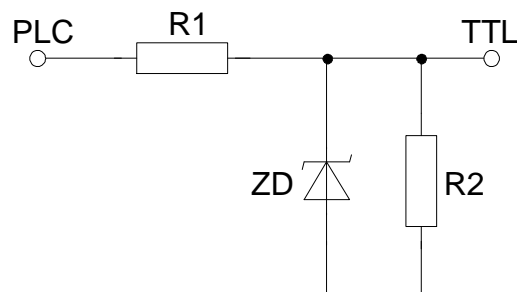
Obr. 2.8.1 – Napěťové přizpůsobení pro CMOS

Pro TTL obvody to není tak snadné, protože pokud je na výstupu PLC logická nula, TTL obvod má na vstupu neustále logickou jedničku. Toto jsme vyřešili s použitím odporů a diody (viz Obr. 2.8.2).



Obr. 2.8.2 - Napěťové přizpůsobení pro TTL

Toto zapojení má nevýhodu, že má na výstupu napětí 5,6V, což je trochu víc než bychom potřebovali, je ovšem plně funkční. Jiné možné pravděpodobně lepší řešení je s použitím Zenerovy diody (viz Obr. 2.8.3).



Obr. 2.8.3 - Napěťové přizpůsobení pro TTL

Bylo by třeba dopočítat odpory, abychom diodu nespálili. Vybrali bychom diodu se zenerovým napětím 5V.

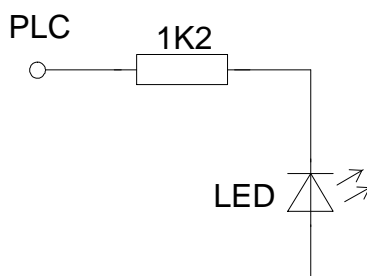
2.9 Ovládací panel

Ovládací panel je určen pro obsluhu výtahu a pro informace pro cestující v jakém patře se výtah nachází a kde zastaví. Jsou na něm použita tlačítka, LED diody různých tvarů, přepínač, patice konektoru a 7-segmentový displej.

Tlačítka jsou spínacího typu jedním vstupem připojená ke zdroji napětí 24V a druhým na PIN patice konektoru. Zde nebylo potřeba přidávat žádné jiné součástky.

Přepínač má svorky tři, dvě vstupní a jednu výstupní. Na jednu vstupní svorku je připojeno napětí 24V a na druhou 0V. Výstupní svorka je připojená k patici.

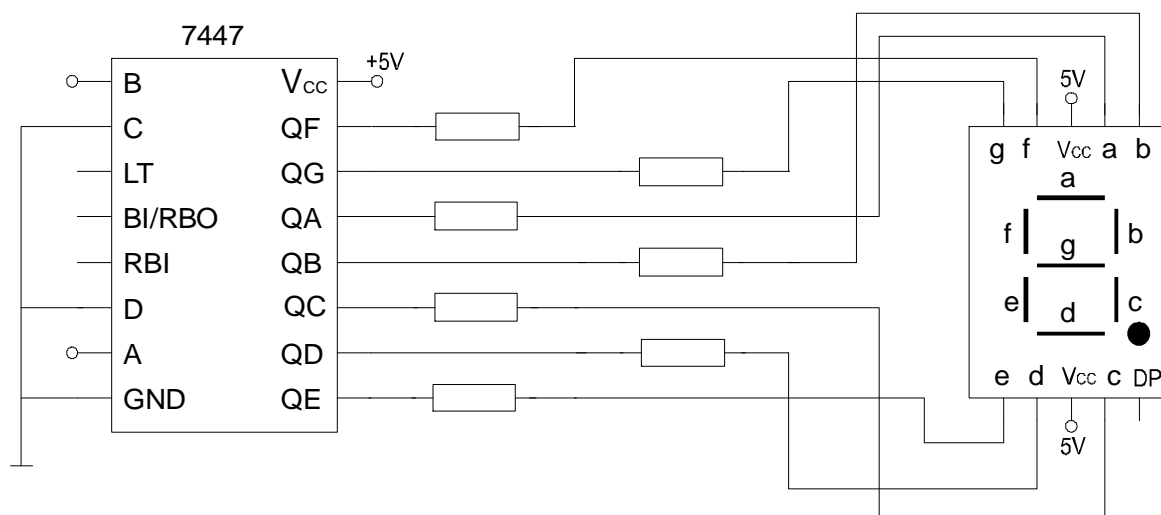
LED diody máme dvou typů, první běžné diody kruhové tvaru a druhé směrové šipky. Oboje svítí světle zeleně. Dioda je proudová součástka a proto musí být připojena přes odpor (viz Obr. 2.9.1), který omezí proud. Hodnota odporu je pro oba typy stejná. Podle velikosti proudu se mění intenzita svícení.



Obr. 2.9.1 – Zapojení LED diody

7-segmentový displej má deset vstupů, 7 segmentů, desetinou tečku a dva vstupy napájení. Abychom neobsadili sedm výstupů PLC použijeme dekodér pro 7-segmentové displeje 7447 nebo 74147. Toto je jediná TTL součástka. Použili jsme 7-segmentový displej se společnou anodou. To znamená, že segment svítí pokud je na jeho vstupu logická nula. Protože dekodér nemá implementované ochranné odpory, musíme je do zapojení doplnit. Platí zde stejné zákonitosti jako pro běžné LED diody.

Zapojení 7-segmentového displeje a dekodéru:



Obr. 2.9.2 – Zapojení 7-seg. a dekodéru

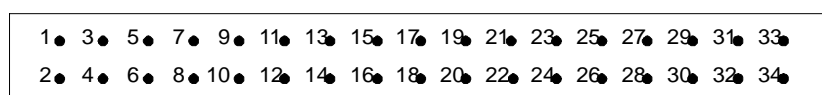
Použité odpory mají velikost 270 Ω . Schéma je nakresleno ze strany součástek.

MH 7447

Na vstupy A,B,C,D se přivede číslo v binárním kódu, které chceme zobrazit. V našem případě to je pouze 0 až 3, proto jsou vstupy C a D uzemněny. Výstupy QA až QG mají inverzní logiku, segment svítí pokud je na nich nula. Více pozornosti věnujeme zbývajícím vstupům, které jsou také aktivní v nule, proto nemusí být zapojeny.

- LT - svítí všechny segmenty, výstupy jsou v logické nule
- BI/RBO - vstup - zhasne všechny segmenty
- výstup - při sériovém použití, slouží jako vstup pro další dekodér
- RBI - zhasne všechny segmenty a aktivuje RBO do nuly

Patice, která slouží k připojení má 34 pinů. Je zde proto, abychom mohli obvod kdykoli odpojit. Nyní patici popíšeme, ale ze strany spojů.



Obr. 2.9.3 – Konektor pro připojení ovládacího panelu

Nyní popíšeme piny, ale pouze ty které jsou zapojené.

číslo pinu	popis
1	3 patro, dioda, venku, dolů
2	2 patro, dioda, venku, nahoru
3	2 patro, dioda, venku, dolů
4	1 patro, dioda, venku, nahoru
5	1 patro, dioda, venku, dolů
6	0 patro, dioda, venku, nahoru
7	3 patro, tlačítko, venku, dolů
8	2 patro, tlačítko, venku, nahoru
9	2 patro, tlačítko, venku, dolů
10	1 patro, tlačítko, venku, nahoru
11	1 patro, tlačítko, venku, dolů
12	0 patro, tlačítko, venku, nahoru
14	vstup A, 7-segment
16	vstup B, 7-segment
21	dioda směr dolů
22	dioda směr nahoru
26	přepínač zatížení výtahu
27	3 patro, tlačítko, uvnitř
28	0 patro, tlačítko, uvnitř
29	2 patro, tlačítko, uvnitř
30	1 patro, tlačítko, uvnitř
31	2 patro, dioda, uvnitř
32	0 patro, dioda, uvnitř
33	3 patro, dioda, uvnitř
34	1 patro, dioda, uvnitř

Tab. 2.9.1 – Popis pinů konektoru na ovládacím panelu

„Venku“ jsou tlačítka a diody, které by se běžně objevily vně výtahu. „Uvnitř“ jsou ve výtahu.

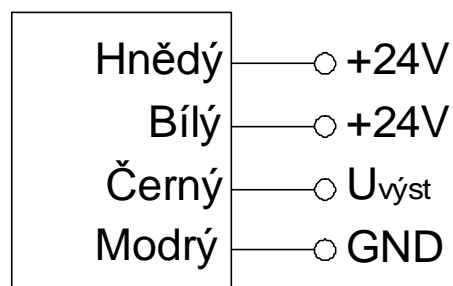
2.10 Optické čidlo a ultrazvukový senzor

Polohu výtahu v patře detekujeme optickým čidlem (viz Obr. 2.10.1) zapojeným podle (viz Obr. 2.10.2).



Obr. 2.10.1 – Optické čidlo

Na čidle můžeme nastavit práh, jak daleko je překážka od čidla. To se provede žlutým kolečkem, které je na horní stěně čidla. Čidlo je připojené čtyřmi vodiči, dva jsou vstupní a dva výstupní. Zapojením zvolíme režim výstupu. Při zastíněném čidle je na výstupu buď napájecí napětí 10V až 30V, nebo nula. My jsme vybrali kladnou logiku. Pokud bychom chtěli opačnou logiku, přivedli bychom na bílý vodič 0V.

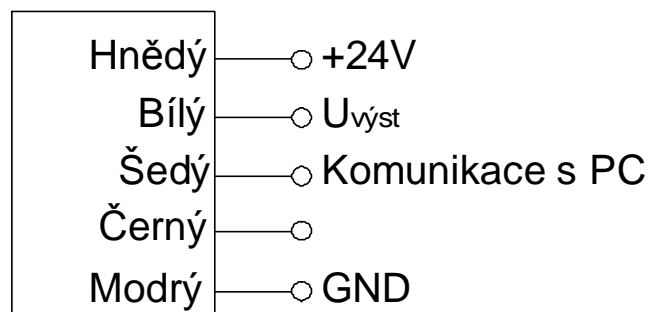


Obr. 2.10.2 – Zapojení optického čidla

Polohu analogově řízeného výtahu ještě snímáme ultrazvukovým senzorem Mikrosonic (viz Obr. 2.10.3) zapojeným podle (viz Obr. 2.10.4).



Obr. 2.10.3 – Ultrazvukový senzor



Obr. 2.10.4 – Zapojení ultrazvukového senzoru

Ultrazvuková čidla pracují na principu měření času odezvy na vyslaný sled impulsů. Z rozdílu času vyslaného a přijatého signálu odraženého od předmětu vypočítá čidlo vzdálenost detekovaného předmětu. Podle zvoleného výstupu je hodnota převedena na analogový nebo digitální signál. Veškeré nastavování funkcí čidla je uskutečněno pomocí dvou tlačítek na stěně čidla. Dvě tříbarevné diody informují o všech možných funkčních stavech. Analogový signál je možné zvolit rostoucí, nebo klesající při přibližování detekovaného předmětu.

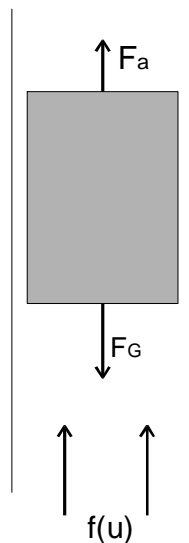
Technické parametry ultrazvukového senzoru:

Napájecí napětí čidla	10-30V.
Analogový výstup	0-10 V / 4-20 mA
Dosah	30 - 250 cm
Max. zatěžovací proud	500 mA
Ultrazvuková frekvence	320 kHz
Přesnost	-+ 1 mm
Časová odezva výstupu	40 ms
Ochrana proti zkratu a přepólování	

3. Identifikace systému

Jak již bylo zmíněno, prostřední výtah je řízen analogově. Máme k dispozici analogovou zpětnou vazbu i akční zásah. Proto můžeme použít PID regulátor, jenže ten je potřeba nejprve navrhnout. K tomu potřebujeme nejprve znát parametry regulované soustavy. Nejprve tedy zidentifikujeme soustavu. To provedeme nasnímáním přechodových charakteristik a následnou identifikací ARX modelu. Napíšeme fyzikální rovnice a zjistíme jaký je řád soustavy.

3.1 Fyzikální model



Obr. 3.1.1 – Fyzikální model

Sestavíme pohybovou rovnici

$$F_a + F(u) - F_G = 0, \quad (3.1)$$

kde F_a je pohybová síla, F_G je tíhová síla a $f(u)$ je síla proudění vzduchu. Rovnici převedeme.

$$m \cdot a + F(u) - m \cdot g = 0, \quad (3.2)$$

kde m - hmotnost válečku, a - zrychlení, g - tíhové zrychlení.

$$\ddot{x} + \frac{1}{m} F(u) - g = 0, \quad (3.3)$$

síla $F(u)$ je funkcí vstupu. Na vstup přivádíme napětí, tím roztočíme ventilátor a tím měníme intenzitu foukání. Tím se také mění síla, která vynáší váleček. Jelikož se napětí mění asi jenom v rozsahu 5% můžeme vztah linearizovat a napsat, že je síla úměrná vstupnímu napětí. $F(u) = c \cdot u$, kde c je konstanta, která v sobě zachycuje geometrii válečku a jiné.

$$\ddot{x} + k \cdot u - g = 0 \quad (3.4)$$

V rovnici zanedbáme konstantu g . Nemá vliv na dynamiku systému. Po navržení regulátoru, budeme muset posunout akční zásah na hodnotu, které odpovídá stání válečku na jednom místě.

Rovnici (3.4) napíšeme s neznámými konstantami a přepíšeme do Laplaceových obrazů (3.5). Vidíme že je soustava je druhého řádu.

$$(a_2 s^2 + a_1 s + a_0) \cdot Y = b_0 U \quad (3.5)$$

3.2 Identifikace ARX modelu

Zde uvedeme ARX model a jeho identifikaci pomocí metody nejmenších čtverců.

3.2.1 ARX model

ARX model je definovaný rovnicí (3.6).

$$y(k) = -a_1 \cdot y(k-1) - a_2 \cdot y(k-2) + b_2 \cdot u(k-2) + e, \quad (3.6)$$

kde e je chyba měření, náhodná veličina s normálním rozložením

$$e \approx N(0, \sigma^2). \quad (3.7)$$

Sestavíme matici dat

$$\begin{bmatrix} y(k) \\ y(k+1) \\ \mathbf{M} \\ y(k+n) \end{bmatrix} = \begin{bmatrix} -y(k-1) & -y(k-2) & u(k-2) \\ -y(k) & -y(k-1) & u(k-1) \\ \mathbf{M} & & \mathbf{M} \\ -y(k+n-1) & -y(k+n-2) & u(k+n-2) \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ b_2 \end{bmatrix} + e \quad (3.8)$$

$$Y = Z \cdot \Theta + e \quad (3.9)$$

3.2.2 Nejmenší čtverce

Cílem našeho snažení je nalézt matici parametrů Θ , to můžeme řešit několika způsoby. My jsme si vybrali metodu nejmenších čtverců. Tato metoda minimalizuje střední kvadratickou odchylku. Vycházíme z předpokladu, že je chybou zatížena pouze výstupní veličina Y . Matici dat Z známe přesně. Definice úlohy nejmenších čtverců v tomto smyslu je nakonec následující

$$\min_{\Theta \in R^n} \|Z\Theta - Y\|_2 \quad (3.10)$$

Řešení metody přímou minimalizací normy

Norma, která se má minimalizovat (kritériu 3.10) lze napsat takto

$$\|e\|_2^2 = \|Z\Theta - Y\|_2^2 = (Z\Theta - Y)^T (Z\Theta - Y), \quad (3.11)$$

po roznásobení dostaneme výraz

$$\|e\|_2^2 = \Theta^T Z^T Z\Theta - \Theta^T Z^T Y - Y^T Z\Theta + Y^T Y. \quad (3.12)$$

Hledáme minimum tohoto výrazu. Využijeme-li vztahy pro maticové derivace

$$\frac{\partial}{\partial x} (x^T Ay) = Ay \quad \frac{\partial}{\partial y} (x^T Ay) = A^T x, \quad (3.13)$$

musí pro minimum platit

$$Z^T Z\Theta - 2Z^T \Theta Y = 0. \quad (3.14)$$

Výsledné řešení je dáno vztahem

$$\Theta = (Z^T Z)^{-1} Z^T Y. \quad (3.15)$$

3.3 Identifikovaná soustava

Po vynásobení matic a převedení do spojitého času je přenos

$$P(s) = \frac{0.456}{2s^2 + s} \quad (3.16)$$

Z přenosu je vidět, že je soustava astatická (má nulový člen a_0) a tudíž má integrační charakter.

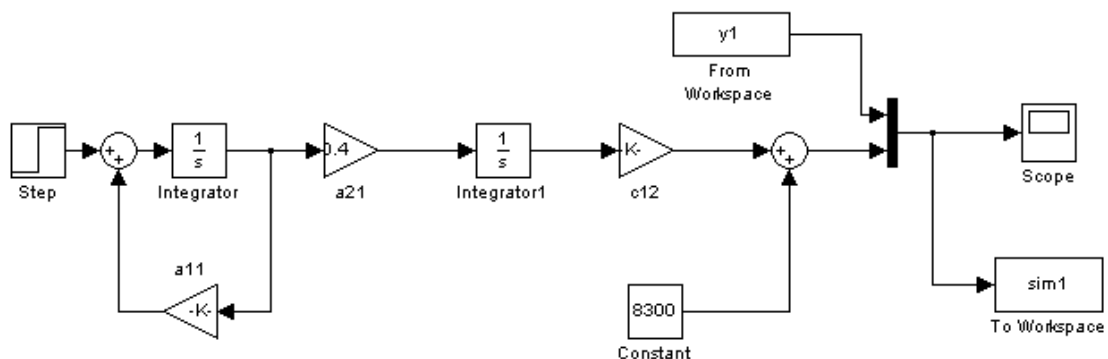
Stavové rovnice

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (3.17)$$

Stavové matice mají ve spojitém případě tento tvar

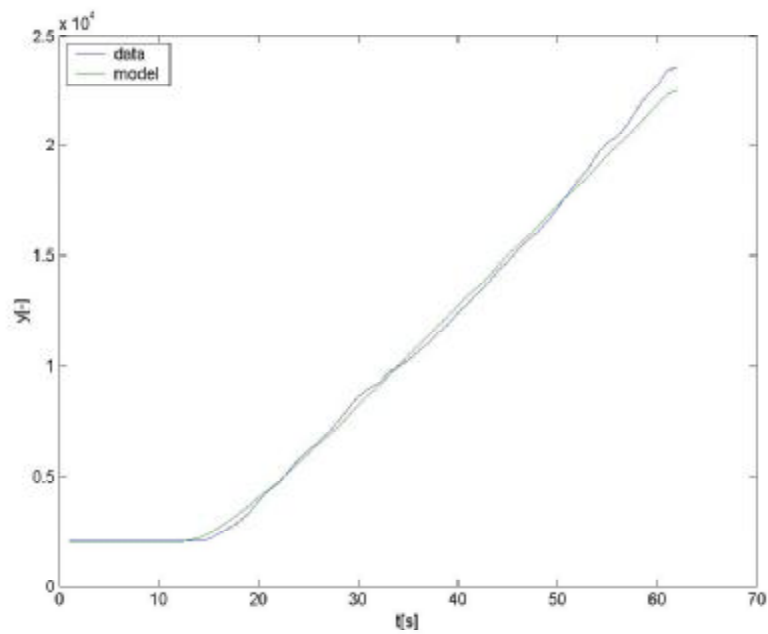
$$A = \begin{bmatrix} -0.5 & 0 \\ 0.5 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = [0 \quad 0.456], \quad D = 0. \quad (3.18)$$

Ze stavových matic jednoduše sestavíme simulační schéma (viz Obr. 3.3.1).

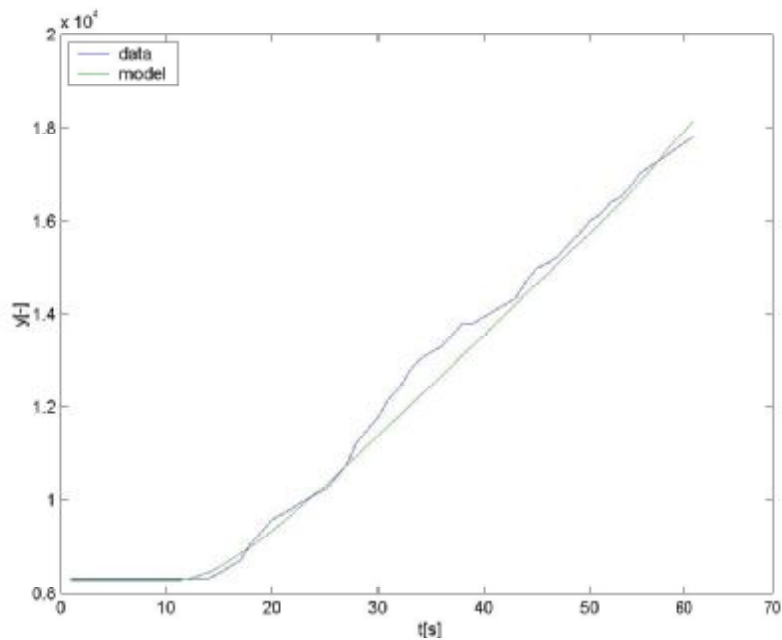


Obr. 3.3.1 – Simulační schéma

Nastavíme vypočítané konstanty a odsimulujeme přechodové charakteristiky nejprve pro přechodovou charakteristiku, kterou jsme použili pro identifikaci parametrů (viz Obr. 3.3.2) a poté ověříme správnost konstant pro jinou hodnotu vstupního signálu (viz Obr. 3.3.3).

Přechodové charakteristiky

Obr. 3.3.2 – Přechodová charakteristika

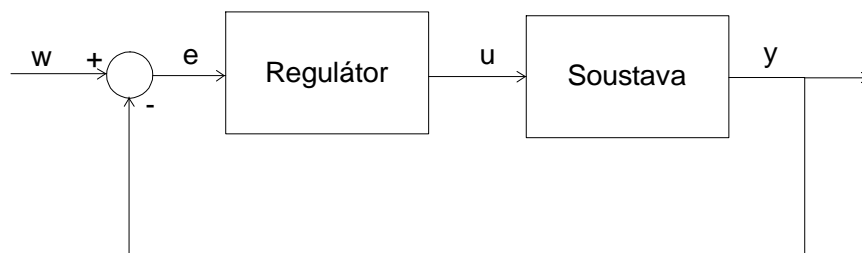


Obr. 3.3.3 – Přechodová charakteristika

Z předcházejících charakteristik vidíme, že se identifikace parametrů povedla docela úspěšně.

3.4 Regulátor

Jelikož chceme výtah ovládat, tím myslíme přivolávat a zastavovat v patrech, uzavřeme zpětnou vazbu pomocí ultrazvukového čidla a zařadíme do obvodu regulátor (viz Obr. 3.4.1).



Obr. 3.4.1 – Schéma uzavřené smyčky

kde w – řídicí veličina (žádaná hodnota)
 e – regulační odchylka
 u – akční veličina regulátoru (vstup do soustavy)
 y – regulovaná veličina (skutečná hodnota)

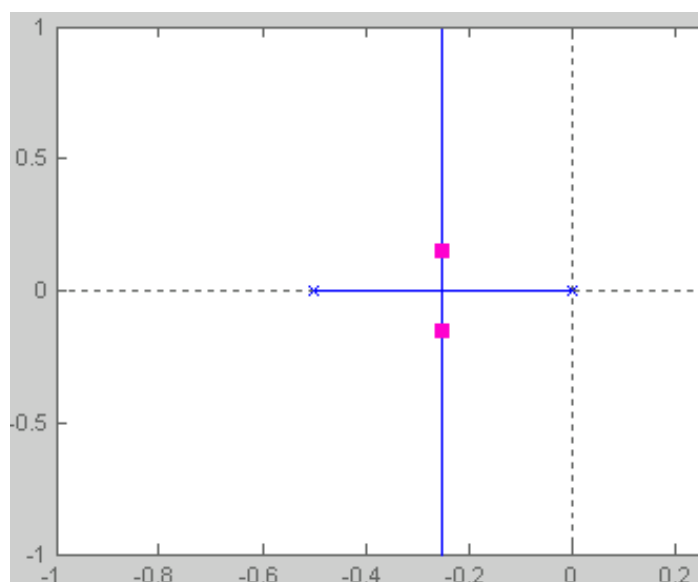
Už nám jenom zbývá navrhnout regulátor. Toto provedeme pomocí geometrického místa kořenů charakteristického polynomu.

P regulátor

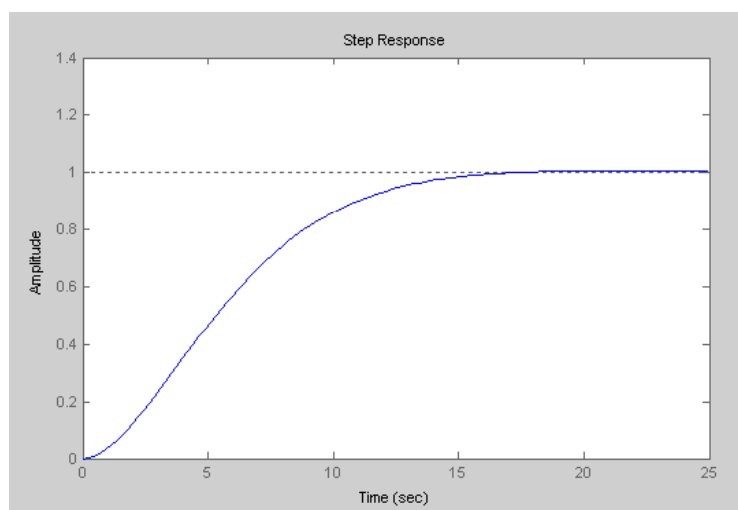
P regulátor má pouze proporcionální složku. Je to vlastně pouze zesilovač odchylky. Přenos regulátoru je:

$$R = r_0.$$

$$R = 0.375$$



Obr. 3.4.2 – Pozice pólů uzavřené smyčky



Obr. 3.4.3 – Přejchodová charakteristika uzavřené smyčky

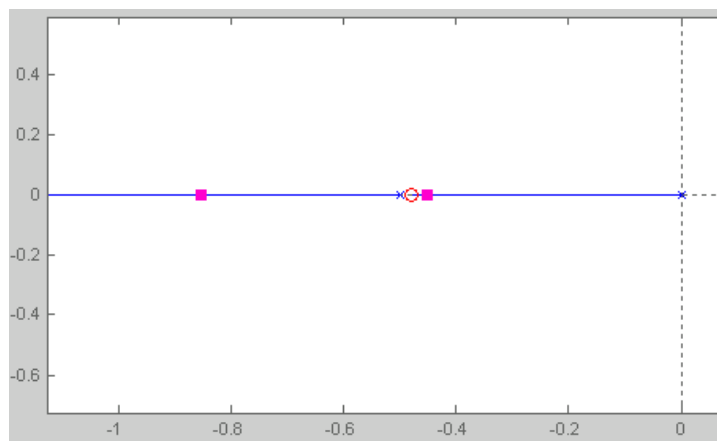
PD regulátor

PD regulátor má proporcionální i derivační složku.

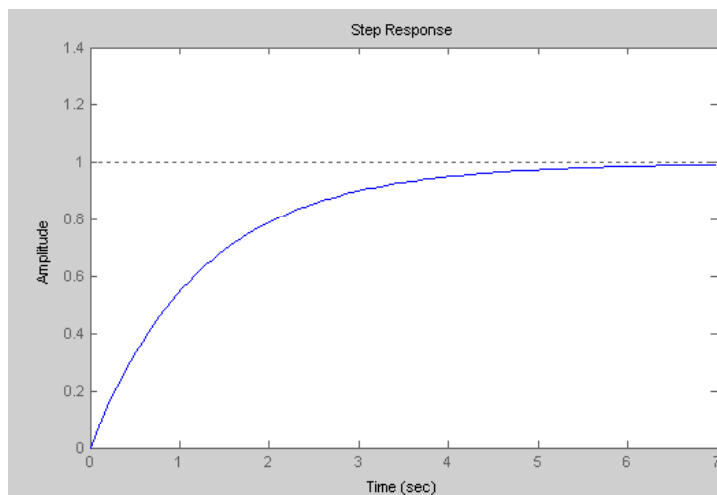
Přenos regulátoru je:

$$R(s) = r_0 + r_1 s = r_0 \left(\frac{r_1}{r_0} s + 1 \right) = r_0 \left(\frac{s}{w_d} + 1 \right), \quad w_d = \frac{r_0}{r_1}. \quad (3.19)$$

$$R(s) = 1.7(2s + 1)$$



Obr. 3.4.4 – Pozice pólů uzavřené smyčky



Obr. 3.4.5 – Přejchodová charakteristika uzavřené smyčky

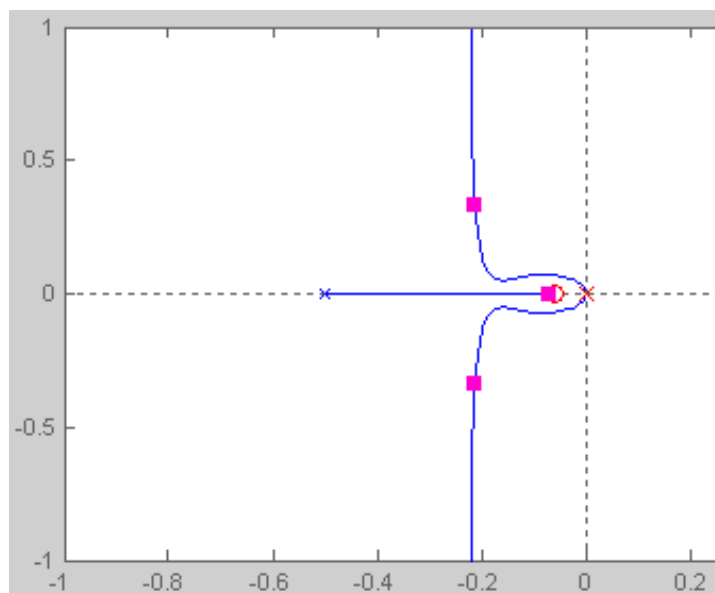
PI regulátor

PI regulátor má proporcionální a integrační složku.

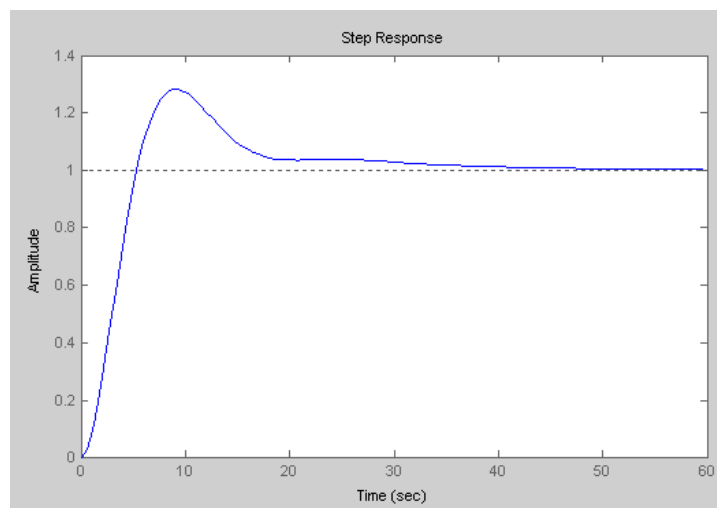
Přenos regulátoru je:

$$R(s) = r_0 + \frac{r_{-1}}{s} = r_{-1} \frac{\left(\frac{r_0}{r_{-1}}s + 1\right)}{s} = r_{-1} \frac{\left(\frac{s}{w_i} + 1\right)}{s}, w_i = \frac{r_{-1}}{r_0}. \quad (3.20)$$

$$R(s) = 0.05 \frac{(16.9s + 1)}{s}$$



Obr. 3.4.6 – Pozice pólů uzavřené smyčky



Obr. 3.4.7 – Přejchodová charakteristika uzavřené smyčky

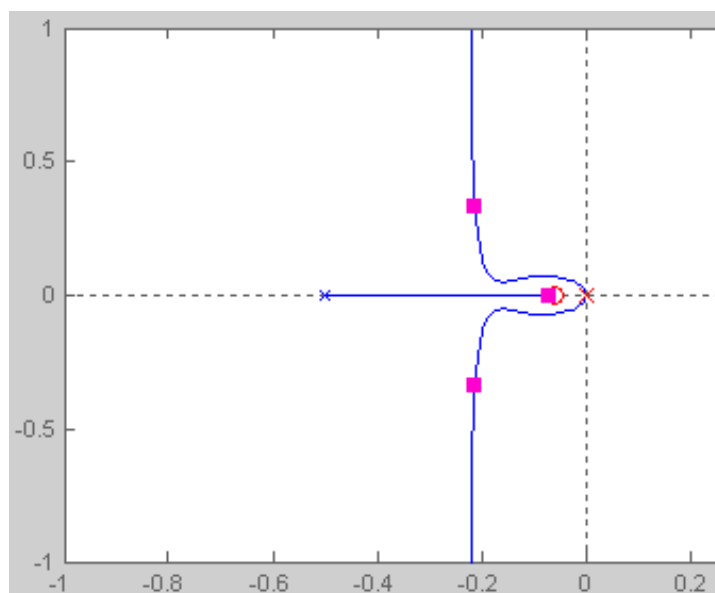
PID regulátor

PID regulátor má všechny tři složky, proporcionální, derivační i integrační složku.

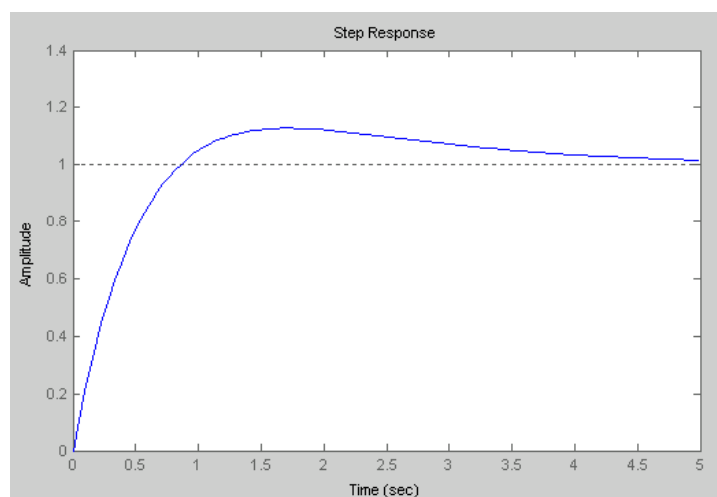
$$R(s) = r_0 + \frac{r_{-1}}{s} + r_1 s = \dots = \frac{r_{-1}}{s} \left(\frac{s}{w_i} + 1 \right) \left(\frac{s}{w_d} + 1 \right), \quad (3.21)$$

$$\frac{1}{w_i w_d} = \frac{r_1}{r_{-1}}, \quad \frac{1}{w_i} + \frac{1}{w_d} = \frac{r_0}{r_{-1}}, \quad w_i = \frac{w_d}{10}.$$

$$R(s) = \frac{2.8}{s} (1.54s + 1)(2.41s + 1)$$



Obr. 3.4.8 – Pozice pólů uzavřené smyčky



Obr. 3.4.9 – Přechodová charakteristika uzavřené smyčky

Z uvedených přechodových charakteristik je zřejmé, že použijeme PD regulátor pro jeho rychlost a také proto, že je bez překmitu, což je důležité pro pohodlí cestujících. Výsledný přenos regulátoru je tedy

$$R(s) = 1.7(2s + 1). \quad (3.22)$$

4. Simatic

4.1 Teoretická část

4.1.1 Hardware

Firma Siemens dodává na trh několik typů PLC s různými výkony a samozřejmě i jinou cenou. Dříve se vyráběly automaty řady S5, které jsou v dnešní době nahrazeny modernějšími automaty řady S7. Ty se dále dělí podle výkonu na řady 200, 300 a 400. Nejvýkonnější je řada 400, nejméně výkonná je řada 200, tyto automaty se používají jenom v jednoduchých aplikacích. Dále popisované skutečnosti se týkají pouze řady 300 a 400.

Automaty se vyrábějí modulární technologií. Automat lze sestavit z různých modulů. To znamená, že si zákazník může vybrat přesně to co potřebuje.

Nabízené moduly:

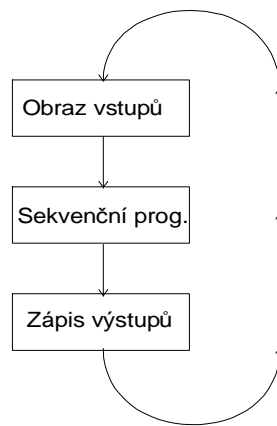
- různé typy procesorů
- vstupní moduly - digitální
- analogové
- výstupní moduly - digitální
- analogové
- komunikační procesory
- funkční moduly, například čítače

Vstupní a výstupní karty se vyrábí s různými počty kanálů od 8 do 32 pro digitální karty a od 2 do 8 kanálů pro analogové karty.

Automat má na přední části přepínací klíček, kterým se volí v jakém módu se má PLC nacházet. Možnosti jsou RUN, RUN-P, Stop a Reset. Pokud je PLC v RUN, automat cyklicky vykonává program. Pokud je PLC v RUN-P, chování automatu je stejné jako u RUN, můžeme ale nahrávat program do PLC. Pokud je v programu nějaká chyba, automat se přepne do Stop módu. Pokud je klíček v pozici Stop, automat nic neprovádí, lze nahrávat program. Pozice Reset umožňuje vymazat paměť PLC .

Scan cyklus PLC

Automaty PLC pracují na principu sekvenčního zpracování instrukcí. Automat neustále zpracovává takzvaný scan cyklus (viz *Obr. 4.1.2.1*).



Obr. 4.1.2.1 – Zjednodušený scan cyklus

V každém cyklu se nejprve načtou fyzické vstupy a uloží se do tabulky vstupů. Provádí se proto, aby v programu nevznikaly logické hazardy. Potom proběhne sekvence příkazů, pracuje se s obrazy vstupů, výstupy se zapisují do tabulky výstupů. Na konci cyklu se hodnoty z tabulky výstupů přepíší na fyzické výstupy. Dále se budeme zabývat pouze sekvenční částí. Na Obr. 4.1.2.1 je pouze zjednodušená verze scan cyklu. Ve skutečnosti ještě přibývá diagnostika a komunikace.

4.1.2 Programové vybavení

Základní program, ve kterém se realizuje celý návrh řízení, je *Simatic manager*. Po spuštění tohoto programu nás uvítá vizard na založení nového projektu. V projektu se musí vybrat typ procesoru, který máme k dispozici. Dále se v hardwarové konfiguraci vyberou veškeré moduly.

Struktura programu

OB - organizační blok

OB1 - základní blok , volá se v každém cyklu, z něho se volají další bloky.

OB100 - volá se pouze po restartu PLC

OBxx - dají se použít i další OB, např. opravy chyb, volání po určitém čase, atd.

FC - funkce

Funkce se používá jako v jiných programovacích jazycích. Lze volat s parametry. Parametry jsou vstupní, výstupní a vstupně/výstupní. Funkce se většinou napíše pro kód, který se opakuje ve více částech programu.

FB - funkční blok

Funkční blok se používá stejně jako funkce. Jeho výhodou je v možnosti použití statických proměnných. Ty jsou uloženy v instančním DB a nemusí se FB předávat jako parametr. FB se může volat s různými datovými bloky.

DB - datový blok

Datové bloky slouží k ukládání proměnných. Jsou dvojího typu. Instanční, ty se přidělují k FB. Sdílené, slouží jako úložiště s různou možností použití.

Zdrojové kódy se píše v programu *Step 7*. Toto prostředí se spustí otevřením příslušných bloků (viz výše). Vývojové prostředí nám nabízí tři základní možnosti psaní programu. Uvedeme je za sebou podle toho jak se využívají a uvedeme ukázkou pro stejný příklad.

STL

Zřejmě nejpoužívanější prostředí je STL (statement list). Toto prostředí se hodně podobá assembleru na PC. Na první pohled sice vypadá nepřehledně, ale zkušený programátor se v něm rychle zorientuje. Dají se v něm vytvořit nejpokročilejší techniky.

Network 1: Ukazka programu v STL

Logicky soucin v jazyce STL

```
A      M      0.0
A      M      0.1
=      M      0.2
```

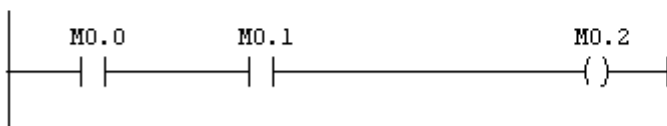
Obr. 4.1.2.1 – Ukázka programu v STL

LAD

Další využívané prostředí je LAD (ladder = žebřík). Toto je zaměřené na psaní žebříčkových diagramů, někdy také nazývané jako reléová schémata. Používají se zde kontakty relé. Na první pohled je velice přehledný. Složitě programové konstrukce se zde již nedají tak lehce vytvořit a program se stává nepřehledný.

Network 1: Ukazka programu v LAD

Logicky soucin v LAD



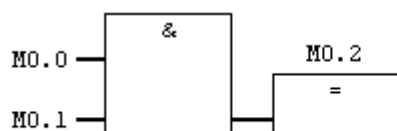
Obr. 4.1.2.2 – Ukázka programu v LAD

FBD

Další možností je použít schémata funkčních bloků (neplést si s FB jak jsme si ho uvedli výše).

Network 1: Ukázka programu v FBD

Logický součin v FBD



Obr. 4.1.2.3 – Ukázka programu v FBD

4.2 Praktická část

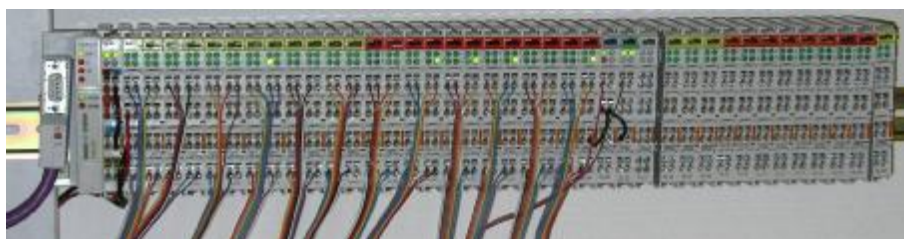
4.2.1 Automat

Model se řídí průmyslovým automatem Simatic 315-2DP od firmy Siemens (dále jenom PLC) (viz Obr. 4.2.1.1).



Obr. 4.2.1.1 – Simatic 315 2DP

Automat má vlastní vstupy a výstupy. Model je však připojen na vstupy a výstupy firmy Wago (viz Obr. 4.2.1.2), které jsou připojeny přes profibus k PLC. To je uděláno proto, že model je od automatu poměrně daleko a při použití normálních vodičů by se zvyšovala pravděpodobnost chyb v přenosu signálu.



Obr. 4.2.1.2 – Periferní I/O wago

Profibus je průmyslová sběrnice, mající vlastní komunikační protokol, založená na standardu RS485. Veškerá komunikace se tedy mezi PLC a periférií uskutečňuje po dvou vodičích. Jinak bychom museli mít od PLC k technologii téměř 100 vodičů. Při psaní programu se periferní vstupy chovají jako obyčejné vstupy a proto to pro programátora nepřináší žádné nevýhody. Nevýhoda tohoto řešení je, že pokud se nám přeruší jeden ze dvou vodičů, ztratíme informaci o všech vstupech a výstupech.

4.2.2 Simatic manager

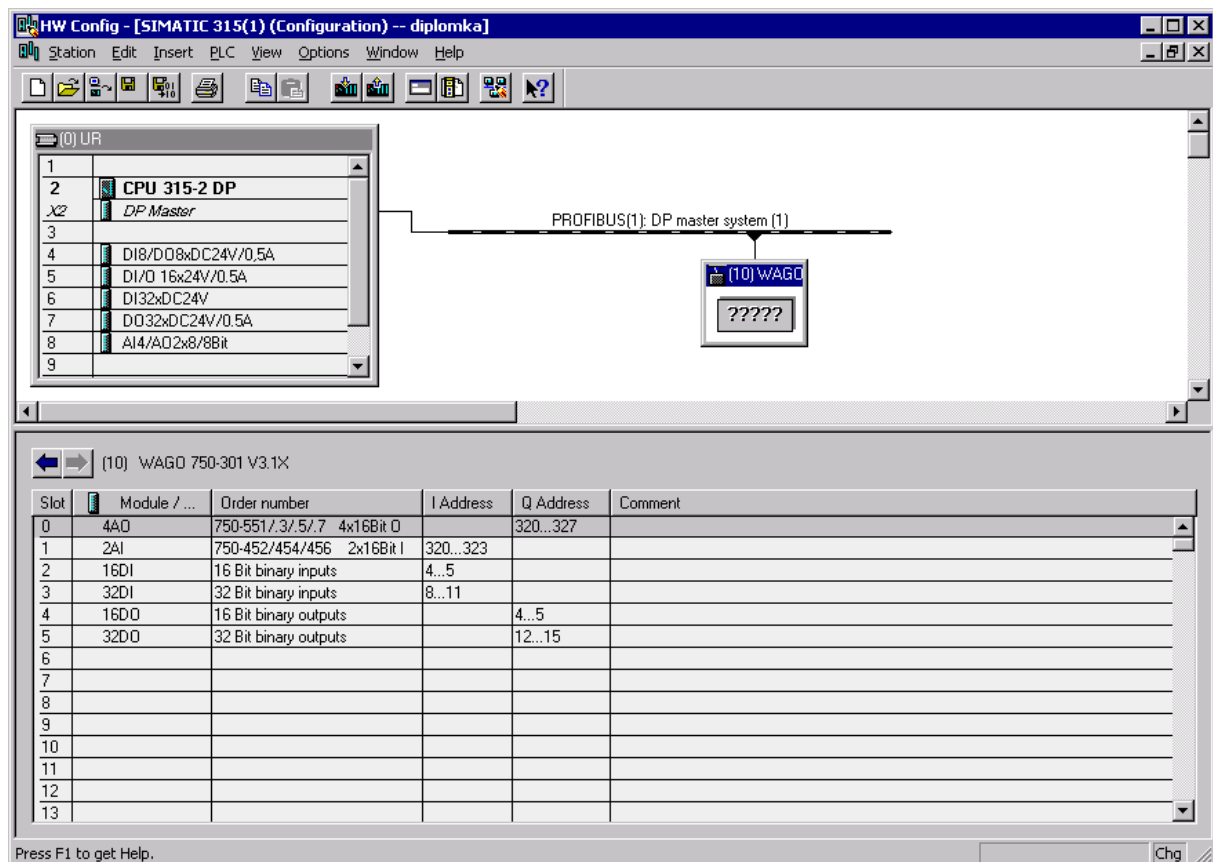
Základní aplikace vývojové prostředí pro Simatic programovatelné automaty je *Simatic manager*. Na následujícím obrázku je ukázka programu a hlavně výpis použitých bloků (viz obr. 4.2.2.1).

Object name	Symbolic name	Created in language	Type	Size in file vol. in...	Author	Name (Head)
System data	---	---	SDB	--	--	--
OB1		STL	Organization Block	210		
FB1	Rizeni_pater_digit	STL	Function Block	1686	Herej	
FB2	Rizeni_pater_analog	STL	Function Block	1682	Herej	
FB3	Srimani	STL	Function Block	256		
FB41	CONT_C	SCL	Function Block	1160	SIMATIC	CONT_C
FC1	ObrazVstupu	STL	Function	344	Herej	
FC2	ObrazVystupu	STL	Function	284	Herej	
FC3	RizeniPaterDigit	STL	Function	362	Herej	
FC4	RizeniPaterAnalog	STL	Function	440	Herej	
FC5	RizeniTai	STL	Function	1092	Herej	
FC6	RizeniTalLogika	STL	Function	2458	Herej	
FC7	RizeniDvou	STL	Function	1092	Herej	
FC8	RizeniDvouLogika	STL	Function	2458	Herej	
DB1	DBleft	DB	Data Block	114	Herej	
DB2	DBmiddle	DB	Data Block	124	Herej	
DB3	DBright	DB	Data Block	114	Herej	
DB4	DBrimani	DB	Data Block	480		
DB5	DBpid	DB	Data Block	162	SIMATIC	
VAT_1	VAT_1		Variable Table	--		
VAT_6	VAT_6		Variable Table	--		
VAT_L	VAT_L		Variable Table	--		
VAT_M	VAT_M		Variable Table	--		
VAT_pid	VAT_pid		Variable Table	--		
VAT_R	VAT_R		Variable Table	--		
SFB4	TON	STL	System function block	--	SIMATIC	TON

Obr. 4.2.2.1 – Náhled na aplikaci Simatic manager

4.2.3 Hardwarová konfigurace

Aby PLC mohlo spolupracovat se vstupy a výstupy, musí se nakonfigurovat podle následující tabulky (viz obr. 4.2.3.1) .



Obr. 4.2.3.1 – Ukázka hardwarové konfigurace

V tomto obrázku je ukázáno jaké se použijí moduly na PLC, ale hlavně jaké se použijí na Wagu.

4.2.4 Logika řízení výtahu

Řízení výtahů je vymyšleno tak, že přijede nejbližší výtah. Logika vychází samozřejmě z řízení každého výtahu zvlášť. Z vizualizace lze zvolit, které výtahy se zapojí ke společnému řízení. Pokud by vznikla potřeba výtah například opravit, vyřadíme ho ze společného řízení. Nefunkčnost jednoho výtahu by nenarušila řízení ostatních. Přivolávání výtahu je řešeno

prioritou zachování směru jízdy. Pokud výtah dojede do prvního patra a tam nastoupí člověk, který chce jet dolů a zároveň je požadavek dojet do druhého patra, výtah jede do patra druhého. Toto zvýhodňuje lidi, kteří výtah chtějí použít v krajních patrech. Jinak by pro ně výtah nemusel dojet.

Pokud je zapojených do řízení víc výtahů, u tlačítek, které jsou vně výtahu se rozhodujeme podle následující tabulky (*viz tab. 4.2.4.1*).

požadavek		výběr výtahu	
patro	směr	patro	směr
0	1	0	0
		1	2
		1	0
		2	2
		2	0
		3	2
		3	0
1	2	1	0
		2	2
		2	0
		3	2
		3	0
1	1	1	0
		0	1
		0	0
2	2	2	0
		3	2
		3	0
2	1	2	0
		1	1
		1	0
		0	1
		0	0
3	2	3	0
		2	1
		2	0
		1	1
		1	0
		0	1
		0	0

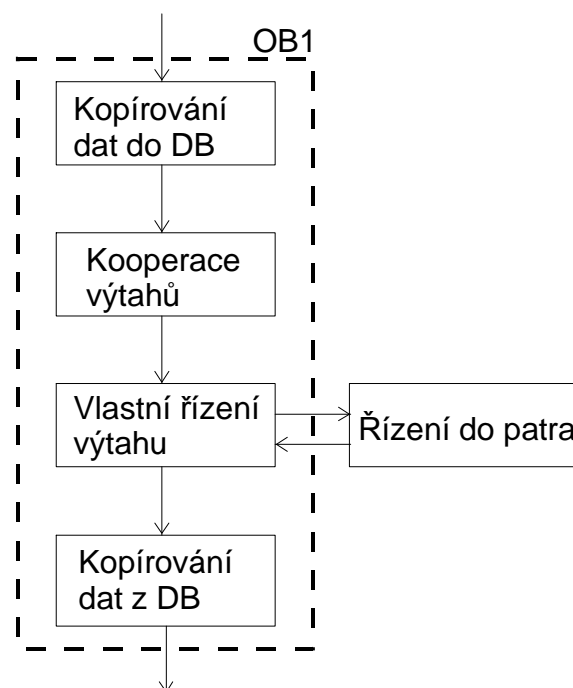
směr 0-stoji, 1-nahoru, 2-dolů

Tab. 4.2.4.1 – Priorita rozhodování při společném řízení

Ve sloupci požadavek je patro a směr tlačítka, které bylo stisknuté „vně“ výtahu. Toto je logický součet od všech výtahů zařazených ve společném řízení. Ve sloupci výběr výtahu, se rozhodujeme od horního řádku. Pokud u některého výtahu odpovídá směr a patro aktuálnímu, zařadí se požadavek na tento výtah. To znamená, že tlačítko bude „stlačené“ pouze u tohoto výtahu. U ostatních bude situace, jako kdyby tlačítko stisknuté nebylo. Pokud neodpovídá žádnému výtahu ani jedna kombinace z uvedené tabulky, požadavek zapíšeme na všechny výtahy. Například pokud je ve druhém patře zmáčknuté tlačítko dolů, nemůžeme se dívat na výtah, který je v prvním patře a jede nahoru, protože nevíme jestli nepojede až do patra třetího.

4.2.5 Struktura programu

Hlavní blok, který se cyklicky vykonává, je OB1. Z něho se volají různé funkce a funkční bloky pro zjednodušení a přehlednost programu (viz obr. 4.2.5.1).



Obr. 4.2.5.1 – Struktura programu

Dále popíšeme jednotlivé bloky z předchozího obrázku.

Kopírování dat do DB

Základní myšlenka programu je, že každý výtah má vlastní datový blok. V tomto DB se ukládají všechny potřebné hodnoty k výtahu, jako jsou obrazy zmáčknutých tlačítek, aktuální patro, kde se výtah nachází a mnoho dalších. Tento datový blok je instanční blok k použitému bloku funkčnímu. Nejprve je tedy zapotřebí naplnit DB aktuálními daty. Všechny vstupy pro jednotlivé výtahy, se tedy postupně nakopírují dovnitř. U tlačítek pro ovládání chodu výtahu nelze udělat pouze kopii vstupů, protože při tomto postupu bychom měli hodnotu v DB pouze po dobu sepnutého tlačítka. Nejprve se tedy načte hodnota, která je v DB a udělá se logický součet v hodnotami na fyzických vstupech. Tuto hodnota se ještě logicky sečte s hodnotou paměťových bitů tlačítek, které se ovládají z vizualizace.

Kopírování dat z DB

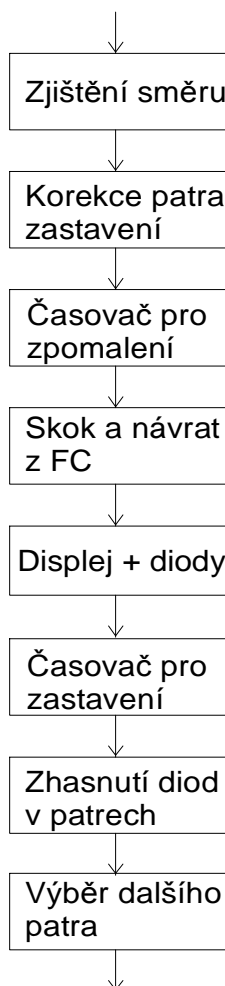
Tato funkce je obdobná funkci předchozí. Postupně se pro jednotlivé výtahy vykopírují hodnoty z DB na fyzické výstupy. Jedná se především o LED diody a ovládací signály.

Kooperace výtahů

Podle toho, které výtahy spolu kooperují (navolí se z vizualizace) se zavolá funkce. Toto se nerozlišuje v jedné funkci, ale je pro to napsáno funkcí více. Volné paměti máme dost a tato varianta není tolik časově náročná a samozřejmě je přehlednější. Nejprve se provede logický součet paměťových míst, kde jsou uložena „stisknutá“ tlačítka vně výtahu. Potom se pro jednotlivá tlačítka rozhoduje podle tab. 4.2.4.1, na kterém výtahu zůstane tlačítko „stisknuté“, neboli, který výtah požadavek obslouží. Někdo například stiskne tlačítko dolů ve třetím patře na levém výtahu. Nejlépe však vyhovuje přijetí výtahu pravého. Rozsvítí se tedy dioda na pravém výtahu a cestující ihned ví, který výtah přijede.

Vlastní řízení výtahu

Vlastní řízení výtahu je napsáno ve dvou funkčních blocích, jeden pro digitální a druhý pro analogové řízení. Datové pozadí je však podobné.



Obr. 4.4.5.2 – Vlastní řízení FB1,2

Trochu podrobněji popíšeme předchozí obrázek. Zjištěním směru, se myslí pouze porovnání aktuálního a patra zastavení. Porovnáním získáme směr jízdy výtahu. Logika funkce je napsaná tak, že by zde tato část být nemusela, pomůže nám ale vyloučit případné problémy při řízení.

Korekce patra zastavení je kontrola patra do kterého se výtah pohybuje, jestli není patro pro zastavení. Například pokud je výtah v prvním patře a byl přivoláno do patra třetího, musíme ještě zkontrolovat, jestli má zastavit v patře druhém. Cestující mohou výtah zastavit, dokud neprojde patrem.

Při jízdě hlavně dolů musíme výtah zpomalit, aby nepřejel patro, kde má zastavit. K tomu nám slouží časovač pro zastavení. Časovač se spouští poté, co kabinka opustí patro.

Výstup časovače je on-delay, takže po dočítání nastavené doby se na výstupu objeví logická jednička. Tím poznáme, že se má zastavit.

Funkci, která zajistí dořízení kabinky do patra zastavení podrobně popíšeme v další podkapitole.

V tomto funkčním bloku taky zajistíme správné číslo na sedmi-segmentovém displeji a správně svítící diody ukazující směr jízdy, ty ukazují směr i když výtah stojí v patře. Tam je to předpokládaný směr jízdy.

Pokud výtah dojede do patra, kde má zastavit spustí se časovač, který odměří dobu pro nastoupení a vystoupení cestujících. Musíme toto simulovat časovačem, protože nemáme k dispozici kontakt zavřených dveří. Ve skutečném výtahu, ale časovač taky je, který odměří dobu mezi nastupujícím člověkem a zavření dveří. Časovač je také on-delay. Po dočítání časovače se jeho výstup nastaví do logické jedničky. To je podmínka pro přechod do další části programu. Pokud není splněna, funkční blok zde končí svoji činnost.

Poté co výtah opustí patro, zhasneme diody, které signalizovali zmáčknutí tlačítka. Zhasneme tedy diodu ve výtahu a diodu vně výtahu, jejíž orientace souhlasila se směrem jízdy.

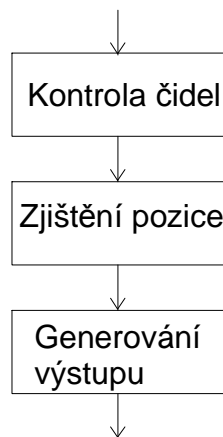
Nakonec ještě vybereme do jakého patra se bude pokračovat. Pokud je ještě zmáčknuté tlačítko v patře, které souhlasí se směrem jízdy, pokračujeme do tohoto patra. Pokud, ale takový požadavek nebyl, nerozhodne se v tomto cyklu nic. Ukončí se provádění funkce. V dalším cyklu se již můžu podívat i na patra v opačném směru. Toto je zde uděláno proto, že by nezhasla dioda šipky vně výtahu, která ukazuje na druhou stranu než byl aktuální směr. Po vybrání nového patra se výtah opět rozjede, v opačném případě zůstane stát v požadovaném patře.

U analogového řízení pouze odpadá blok pro zpomalení výtahu, není potřeba, protože toto zajistí PID regulátor.

Řízení do patra

Funkce pro řízení do patra jsou opět dvě. Jedna je pro analogové řízení a druhá pro diskrétní. Tyto funkce jsou poměrně odlišné, proto je probereme zvlášť.

Digitální řízení

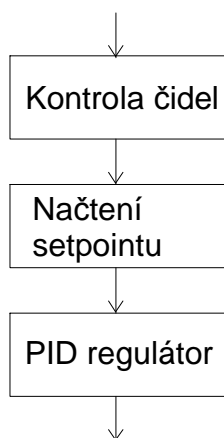


Obr. 4.4.5.3 – Digitální řízení FC3

Nejprve se vyhodnotí, jestli je zastíněno nějaké čidlo. To provedeme v cyklu. Pokud je čidlo v některém patře vyhodnocené, zapíšeme toto patro jako aktuální.

Dále se zjistí jestli je aktuální patro rovno, větší nebo menší než patro, kde se má zastavit. Podle toho se rozhoduje v další části programu. Pokud je aktuální patro menší než patro, kde se má zastavit, znamená to, že máme jet nahoru. Na výstup tedy zapíšeme jedničku. Pokud jsou si patra rovny, jsme v patře, kde se zastaví. Hodnota výstupu se řídí z podmínky zastínění čidla. Pokud je na čidle logická jednička, výtah jede nahoru, ale s jinou střídou. To znamená, že se do impulsů pro pohyb nahoru se vkládají impulsy pro pohyb dolů. Je jich tam ale výrazně méně, proto výtah jede pořád nahoru. Pokud není čidlo zastíněno výtah jede dolů. Tímto se výtah udrží těsně nad čidlem a můžeme prohlásit, že výtah stojí. Když je aktuální patro větší než patro zastavení, výtah jede dolů. Na výstupu je logická nula, ale jenom do doby, dokud nedostaneme povel k brždění. Při brždění prokládáme na výstupu logickou nulu logickou jedničkou.

Analogové řízení



Obr. 4.4.5.4 – Analogové řízení FC4

Kontrola čidel je zde stejná jako u digitálního řízení. Vyhodnotí se čidlo a podle zastíněného zapíšeme aktuální patro. Načtení setpointu, neboli hodnoty na kterou se bude regulovat, provedeme podle patra zastavení. Tato hodnota se zadává v procentech rozsahu ultrazvukového čidla, což je pro nás zpětná vazba. Na výstupu PID regulátoru dostaneme přímo hodnotu, kterou zapíšeme na výstupní analogovou kartu. Parametrizace PID regulátoru není nejjednodušší, proto ho popíšeme.

<pre> CALL "CONT_C", DB5 COM_RST :=#COMP_RST MAN_ON :=#MAN_ON PVPER_ON:=TRUE P_SEL :=#GAIN_ON I_SEL :=#INT_ON INT_HOLD:=FALSE I_ITL_ON:=FALSE D_SEL :=#DER_ON CYCLE :=#CYCLE SP_INT :=#setpoint PV_IN := PV_PER :=#POZICE MAN :=#MAN GAIN :=#GAIN TI :=#TI TD :=#TD TM_LAG := DEADB_W :=#DEAD_BAND LMN_HLM :=#OUT_LIM_HI LMN_LLM :=#OUT_LIM_LOW PV_FAC := PV_OFF := LMN_FAC := LMN_OFF :=#OUT_OFFSET I_ITLVAL:= DISV := LMN := LMN_PER :=#ZASAH QLMN_HLM:= QLMN_LLM:= LMN_P := LMN_I := LMN_D := PV := ER := </pre>	<p><u>Comp_rst</u> – kompletní restart, blok má inicializační rutinu, která se pustí při nastavení bitu na log 1</p> <p><u>Man_on</u> - pokud je bit nastaven, rozpojí se smyčka a na výstupu je hodnota man</p> <p><u>Pvper_on</u> – při používání vstupu z I/O musíme nastavit na true</p> <p><u>P_sel, I_sel, D_sel</u> – zapojení do výpočtu příslušné složky P, I, D</p> <p><u>Int_hold</u> – zmražení výstupu integrační složky</p> <p><u>I_int_on</u> – inicializace integrační složky</p> <p><u>Cycle</u> – sample time, po touto dobu je výstup neměnný</p> <p><u>Sp_int</u> – setpoint, žádaná hodnota</p> <p><u>Pv_in</u> – inicializační hodnota skutečné hodnoty</p> <p><u>Pv_per</u> – skutečná hodnota, vstup z I/O</p> <p><u>Man</u> – manuální hodnota, na výstupu při Man_on</p> <p><u>Gain, TI, TD</u> – konstanty PID regulátoru</p> <p><u>Tm_lag</u> – časové zpoždění derivační složky</p> <p><u>Deadb_w</u> – šířka pásma necitlivosti, pro výpočet PID</p> <p><u>Lmn_hlm, Lmn_llm</u> – horní a dolní limit pro výstup</p> <p><u>Pv_fac, Lmn_fac</u> – multiplikační konstanta pro vstup</p> <p><u>Pv_off, Lmn_off</u> – hodnota, která se připočte k vypočítané hodnotě</p> <p><u>I_intval</u> – inicializační hodnota integrační složky</p> <p><u>Disv</u> – poruchová veličina, přičítá se k vypočítané hodnotě</p> <p><u>Lmn</u> – akční zásah</p> <p><u>Lmn_per</u> – akční zásah přepočtený na I/O</p> <p><u>Qlmn_hlm, Qlmn_llm</u> – signalizace překročení mezí</p> <p><u>Lmn_p, Lmn_i, Lmn_d</u> – vypočítané hodnoty pro jednotlivé složky</p> <p><u>Pv</u> – výstup, skutečná hodnota která se jeví jako vstup</p> <p><u>Er</u> – skutečná odchylka, hodnota která vstupuje do výpočtu</p>
---	---

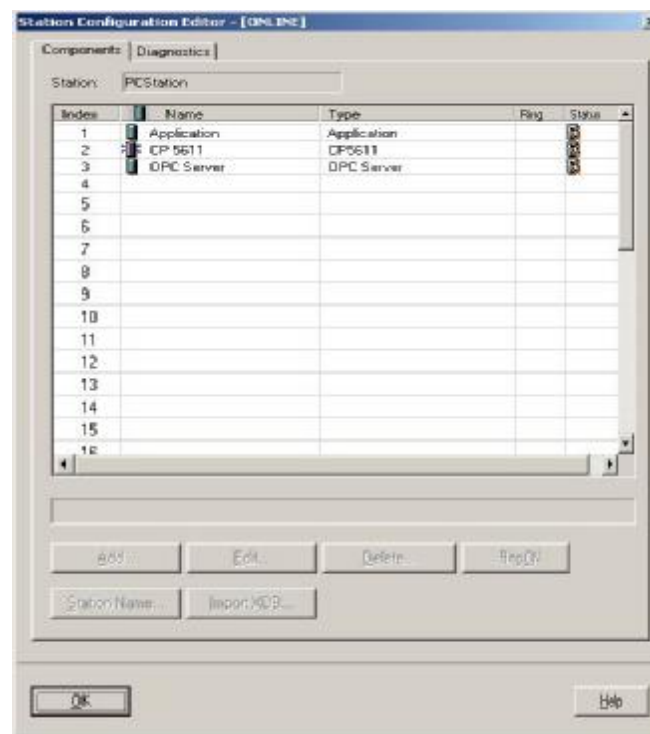
Tab. 4.4.5.1 – Popis parametrů PID regulátoru

5. Vizualizace

Vizualizace slouží ke sledování a ovládání technologie nejčastěji na počítači. Velice často se také používají operátorské panely.

5.1 Simatic net

Přesto že se může zdát, že tato kapitola patří do kapitoly Simatic, uvádíme ji zde. Pokud není správně nastaven Simatic net, síť užívaná automatem a programy firmy Siemens, nepodaří se spojit DDE server s daty. V počítači, na kterém chceme pozorovat technologii, máme v našem případě PCI profibus kartu CP-5611. Touto kartou je spojen počítač s PLC. Je nutné mít tuto kartu správně nastavenou v programu Communication console. Dále je nutné vytvořit projekt v S7 ve kterém nastavíme síť. Správné nastavení ověříme v Station Configuration Editoru (viz. Obr. 5.1.1).

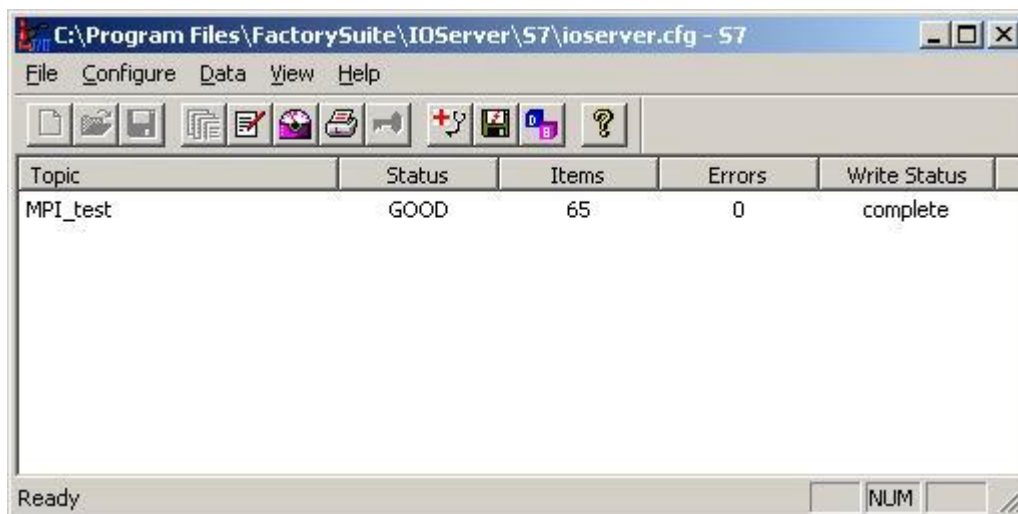


Obr. 5.1.1 – Station configuration editor

První položka Application zastává jakoukoliv aplikaci, kterou lze přistupovat na Simatic net. Druhá položka CP-5611 odpovídá komunikační PCI kartě, kterou je počítač připojen k PLC. Třetí položka OPC-Server zde pro potřeby vizualizace nemusí být.

5.2 DDE Server

DDE Server (Dynamic Data Exchange) je komunikační nástroj mezi aplikací vizualizace a daty. DDE Server dodává firma od které používáme vizualizační program v našem případě WonderWare. Je nutné tedy spustit a nastavit DDE Server. Pokud je DDE server nastaven a je zpuštěna vizualizace, potom v aplikaci DDE Serveru vidíme stav přenášených dat (*viz. Obr. 5.2.1*).



Obr. 5.2.1 – Aplikace DDE Serveru

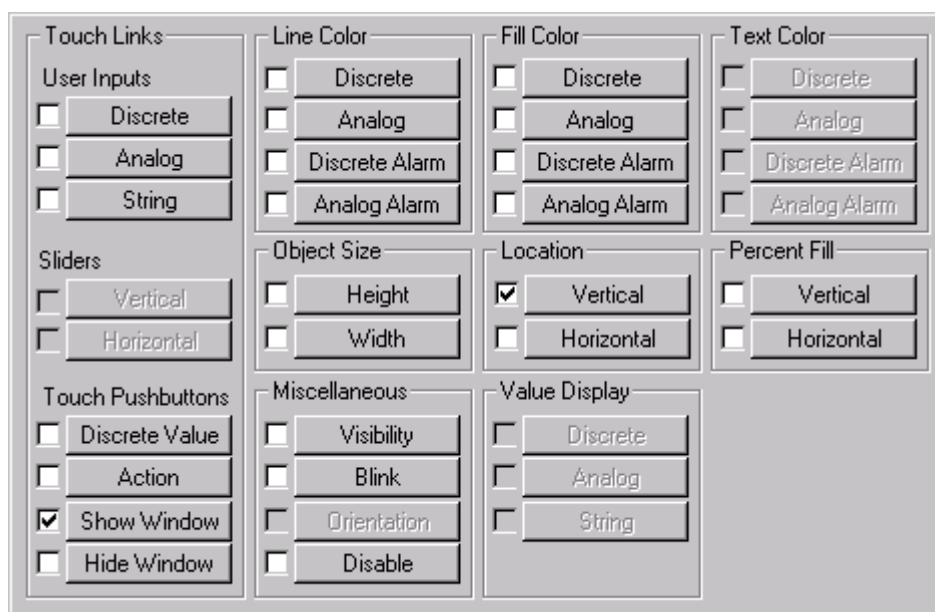
Na obrázku vidíme název spojení, stav přenosu, počet přenášených hodnot, kolik jich je chybových a stav zápisu proměnných. Pokud je status bad a musí se hledat chyba. Nejčastěji bývá chyba v nastavení spojení nebo v nastavení simatic net.

5.3 InTouch

Program, ve kterém se vizualizace vyvíjí a ve kterém je i následně spuštěna, se nazývá InTouch od firmy WonderWare. Aplikace se dělí do dvou částí, vývojová - WindowMaker a runtime – WindowViewer.

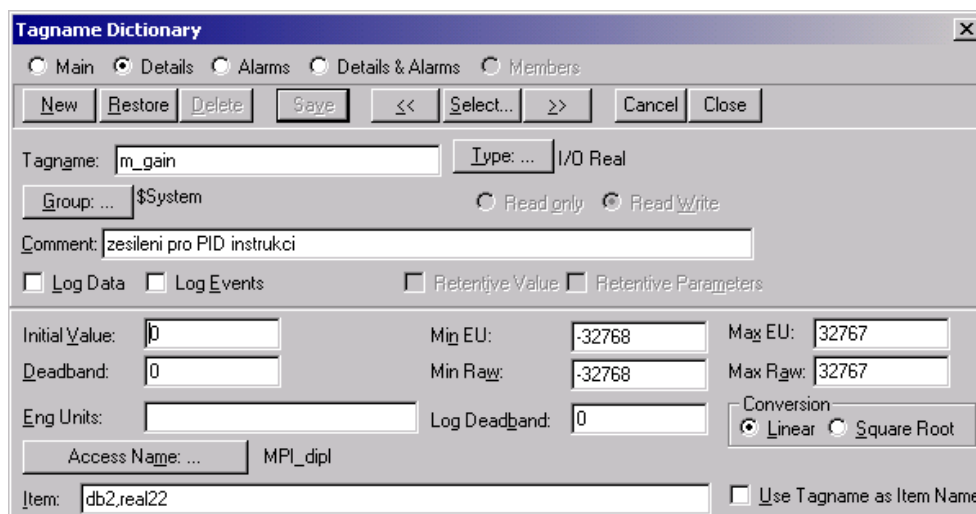
5.3.1 WindowMaker

Jak již název napovídá, jedná se o aplikaci, ve které se navrhují obrazovky. Je zde široká možnost kreslení statických objektů, tvorby uživatelských objektů. Můžeme použít již předdefinované objekty z knihoven takzvané vizardy. Každému objektu, který umístíme na obrazovku, můžeme následující vlastnosti (viz Obr. 5.3.1.1).



Obr. 5.3.1.1 – Vlastnosti objektu

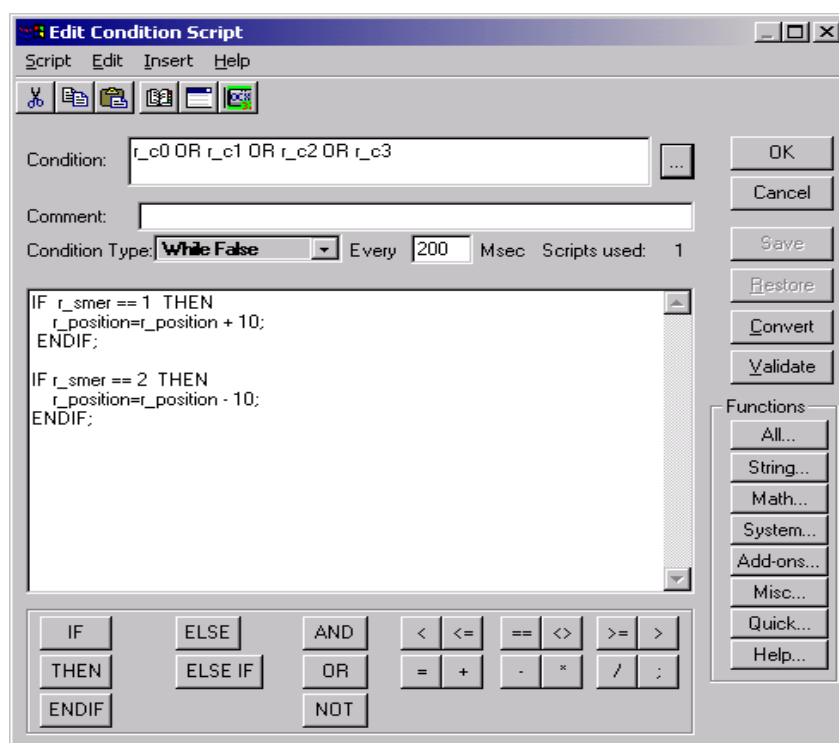
Některé vlastnosti můžeme kombinovat, např. pozici a barvu. Některé kombinovat nelze, např. pozici a velikost. Předchozí vlastnosti je třeba nějakým způsobem ovlivňovat. K tomu slouží takzvané Tag (viz Obr. 5.3.1.2), neboli proměnné. Máme k dispozici několik typů proměnných. Jsou to diskrétní a analogové. Tyto můžou být paměťové v programu, nebo vstupně/výstupní přes DDE.



Obr. 5.3.1.2 – Definice proměnných

Tento tag je vstupně/výstupní typu real. Můžeme vidět možnosti nastavení a jak se odkážeme do datového bloku.

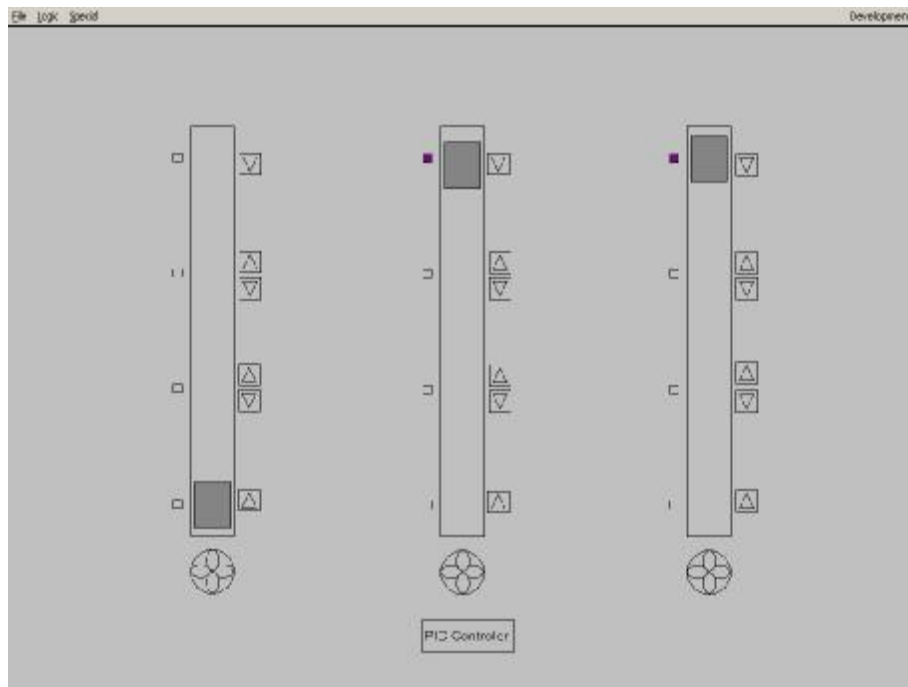
Velké možnosti nám poskytne použití skriptů (viz Obr. 5.3.1.3). Skripty, které můžeme použít jsou: aplikační, podmínkový, klávesový, změna dat, rychlá funkce a aktivX události. Ve skriptech se píše kódy jednoduchým programovacím jazykem za pomoci předpřipravených funkcí.



Obr. 5.3.1.3 – Ukázka skriptu

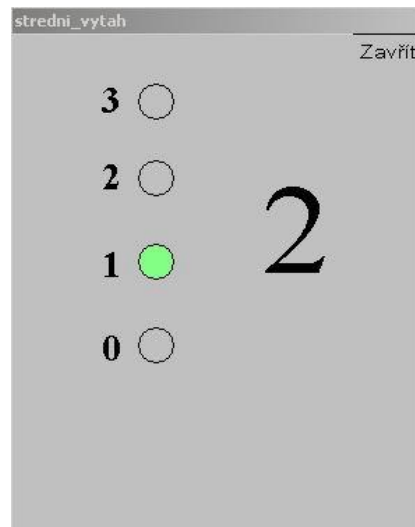
5.3.2 WindowViewer

Po nakreslení obrazovek spustíme aplikaci WindowViewer, kde můžeme obrazovky ovládat. V našem případě máme pět obrazovek. Jednu hlavní (viz Obr. 5.3.2.1) a z ní se spouští obrazovky další.



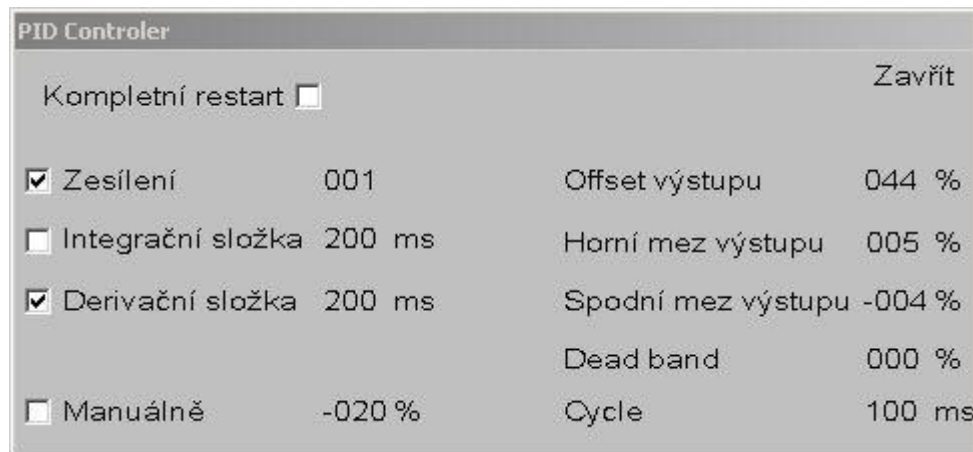
Obr. 5.3.2.1 – Hlavní obrazovka

Na hlavní obrazovce vidíme schéma technologie. Dynamizujeme zde polohu výtahu, podle skutečnosti. Můžeme odtud ovládat tlačítka vně výtahů, šipky nahoru a dolů napravo od výtahu. Kliknutím myši na jednotlivé kabinky výtahů vyvoláme obrazovky, které ukazují vnitřky výtahu (viz Obr. 5.3.2.2).



Obr. 5.3.2.2 – Obrazovka kabinka výtahu

Tato obrazovka nám ukazuje, v jakém jsme patře. Můžeme odtud také ovládat výtah pomocí tlačítek v patrech. Z hlavní obrazovky také můžeme vyvolat servisní obrazovku pro nastavení konstant pro prostřední analogový výtah (viz Obr. 5.3.2.3).



Obr. 5.3.2.3 – Obrazovka PID controler

Tato obrazovka nám slouží především pro jednodušší měnění konstant PID regulátoru. V běžném provozu by byla pro obsluhu nedostupná. Chráněná heslem, aby chyba obsluhy nemohla přivodit neodpovídající chování výtahu.

6. Závěr

Úkolem této diplomové práce bylo vytvořit model tří pneumatických výtahů. To zahrnovalo vytvoření modelu a elektronických obvodů, které lze připojit k programovatelnému automatu. Dále bylo třeba napsat program pro PLC Simatic 300 splňující obecné zásady řízení výtahů v halách. Jako poslední byla vytvořena vizualizace.

Po zprovoznění modelu se vyskytlo několik problémů, které jsme museli vyřešit. Po zapnutí větráčku se váleček roztočil na místě, místo toho aby se pohyboval směrem vzhůru. To jsme vyřešili přiděláním polarizačního kříže nad větráček, který proudící vzduch usměrňuje. Plný výkon větráčku je při napětí +24V. Použitelný rozsah napětí pro řízení je v rozmezí 19V až 20V. Pokud je napětí pod touto úrovní, váleček padá příliš rychle, na druhou stranu příliš rychle stoupá. Tuto nevýhodu jsme zkoušeli odstranit změnou váhy válečku. To však nepřineslo uspokojivé výsledky. Problém vyřešilo použití pulsně-šířkové modulace. Rozdíly ve střídě pro pohyb nahoru a dolů byly o něco větší. Další problém se vyskytnul při digitálním řízení. Po zapnutí je intenzita foukání nižší, než po delší době, kdy je ventilátoru v chodu. To je nejspíš způsobeno zahřáním ložisek. Tento problém se nepodařilo odstranit. Řešení je nechat model v chodu nějakou dobu. Potom se již hodnota nemění. Další problém, který jsme řešili byl ovlivnění výtahů navzájem. Pokud jsme na jednom ventilátoru přidali intenzitu foukání ovlivnilo to i intenzitu u vedlejšího výtahu. To bylo způsobeno změnou proudu vzduchu pod ventilátory. To jsme vyřešili přidáním krytů na krajní výtahy. Po odstranění zmíněných problémů bylo napsání řízení a vytvoření vizualizace poměrně jednoduché.

Na vytvořeném modelu se bude pokračovat další diplomovou prací. Model se bude také řídit pomocí PLC od firmy Wago, které lze programovat přes internet. K modelu bude připojena Web kamera, což bude umožňovat distanční práci s modelem. Model se bude využívat v předmětech Řídící systémy a Návrh automatizovaných zařízení.

7. Použitá literatura

- [1] Horáček, P. *Systémy a modely*. Vydavatelství ČVUT, Praha 2000.
- [2] John, J. *Systémy a řízení*. Vydavatelství ČVUT, Praha 1999.
- [3] Bayer, J., Hanzálek, Z., Šusta, R. *Logické systémy pro řízení*. Vydavatelství ČVUT, Praha 2000.
- [4] Vysoký, O. *Elektronické systémy II*. Vydavatelství ČVUT, Praha 1999.
- [5] Roubal, J. a Pekař, J. *Odhadování a filtrace* [online]. Poslední revize 2003-12-18 [cit. 2003-12-18], <http://dce.felk.cvut.cz/of> .
- [6] Sidat. *Školící materiály PROG 1,2 Simatic*. Praha 2001.
- [7] Siemens AG. *Step 7* [online help].

8. Seznam obrázků

Obr. 1.2.1 – Pohled na technologii	2
Obr. 1.2.2 – Detail válečku	2
Obr. 1.2.3 – Ovládací panel.....	2
Obr. 2.2.1 – Zjednodušené schéma zapojení.....	3
Obr. 2.2.2 – Napájení obvodů	4
Obr. 2.3.2 – Průběhy napětí na generátoru.....	5
Obr. 2.3.3 – Komparátor s hysterezí	5
Obr. 2.3.4 – Převodní char. komparátoru	6
Obr. 2.3.5 – Invertující integrátor	7
Obr. 2.4.1 – Napěťový dělič s komparátory.....	8
Obr. 2.4.2 – Průběhy napětí na horním komparátoru.....	9
Obr. 2.4.3 – Průběhy napětí na dolním komparátoru	9
Obr. 2.5.1 – Schéma logického obvodu	11
Obr. 2.6.1 – Schéma zapojení IR2110	12
Obr. 2.8.1 – Napěťové přizpůsobení pro CMOS	13
Obr. 2.8.2 - Napěťové přizpůsobení pro TTL	14
Obr. 2.8.3 - Napěťové přizpůsobení pro TTL	14
Obr. 2.9.1 – Zapojení LED diody.....	15
Obr. 2.9.2 – Zapojení 7-seg. a dekodéru	16
Obr. 2.9.3 – Konektor pro připojení ovládacího panelu.....	16
Obr. 2.10.1 – Optické čidlo	18
Obr. 2.10.2 – Zapojení optického čidla	18
Obr. 2.10.3 – Ultrazvukový senzor	19
Obr. 2.10.4 – Zapojení ultrazvukového senzoru	19
Obr. 3.3.1 – Simulační schéma	24
Obr. 3.3.2 – Přechodová charakteristika	25
Obr. 3.3.3 – Přechodová charakteristika	25
Obr. 3.4.1 – Schéma uzavřené smyčky	26
Obr. 3.4.2 – Pozice pólů uzavřené smyčky	27
Obr. 3.4.3 – Přechodová charakteristika uzavřené smyčky	27
Obr. 3.4.4 – Pozice pólů uzavřené smyčky	28
Obr. 3.4.5 – Přechodová charakteristika uzavřené smyčky	28
Obr. 3.4.6 – Pozice pólů uzavřené smyčky	29
Obr. 3.4.7 – Přechodová charakteristika uzavřené smyčky	29
Obr. 3.4.8 – Pozice pólů uzavřené smyčky	30
Obr. 3.4.9 – Přechodová charakteristika uzavřené smyčky	30
Obr. 4.1.2.1 – Zjednodušený scan cyklus	33
Obr. 4.1.2.1 – Ukázka programu v ŠTL.....	35
Obr. 4.1.2.2 – Ukázka programu v LAD.....	35
Obr. 4.1.2.3 – Ukázka programu v FBD	36
Obr. 4.2.1.1 – Simatic 315 2DP	37
Obr. 4.2.1.2 – Periferní I/O wago.....	37
Obr. 4.2.2.1 – Náhled na aplikaci Simatic manager.....	38
Obr. 4.2.3.1 – Ukázka hardwarové konfigurace.....	39
Obr. 4.2.5.1 – Struktura programu	41
Obr. 4.4.5.2 – Vlastní řízení FB1,2	43
Obr. 4.4.5.3 – Digitální řízení FC3	45
Obr. 4.4.5.4 – Analogové řízení FC4	46

9. Seznam tabulek

Tab. 2.5.1 – Pravdivostní tabulka.....	10
Tab. 2.5.2 – Karnaughova mapa	10
Tab. 2.6.1 – Popis výstupů IR2110	12
Tab. 2.9.1 – Popis pinů konektoru na ovládacím panelu	17
Tab. 4.2.4.1 – Priorita rozhodování při společném řízení	40
Tab. 4.4.5.1 – Popis parametrů PID regulátoru.....	46

10. Přílohy

1) CD obsahující elektronickou verzi diplomové práce, projekt v programu Simatic manager a vizualizaci.

2) Ukázka programu – výpis ze step 7

Uvedeme pouze ukázkou programu, protože výpis celého programu je na 80 stran.

OB1

```
ORGANIZATION_BLOCK OB 1
TITLE = "Main Program Sweep (Cycle)"
VERSION : 0.1

BEGIN
NETWORK
TITLE =

    CALL "ObrazVstupu" ;// obraz vstupu

    A    "LeftCoop";
    A    "MidleCoop"; // rozhodnuti o kooperaci
    A    "RightCoop";
    JCN  DVA1;

    CALL "RizerniTri" ;// kooperace vseh tri vytahu
    JU   DALE;

DVA1: A    "LeftCoop";
      A    "MidleCoop";
      JCN  DVA2;

    CALL "RizeniDvou" (//kooperace levy a stredni
      KOMBINACE      := 1);

    JU   DALE;

DVA2: A    "MidleCoop";
      A    "RightCoop";
      JCN  DVA3;

    CALL "RizeniDvou" (//kooperace stredni a pravy
      KOMBINACE      := 2);

    JU   DALE;

DVA3: A    "LeftCoop";
      A    "RightCoop";
      JCN  DALE;

    CALL "RizeniDvou" (// kooperace levy a pravy
      KOMBINACE      := 3);

DALE: NOP    0;
      CALL "Rizeni_pater_digit" , "DBleft" ;//rizeni jednotlivych vytahu
      CALL "Rizeni_pater_analog" , "DBmiddle" ;
      CALL "Rizeni_pater_digit" , "DBright" ;

      CALL "ObrazVystupu" ;// obraz vystupu
//      CALL FB    3 , DB4

END_ORGANIZATION_BLOCK
```