ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDÍCÍ TECHNIKY

Dálkové řízení technologického procesu

DIPLOMOVÁ PRÁCE

Praha, 2004 Jiří Hodný

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém 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

Poděkování

Tímto děkuji všem, kteří přispěli radami i kritikou k napsání této diplomové práce. Především děkuji vedoucímu diplomové práce Doc. Ing. Bohuslavu Kirchmannovi, CSc. za vedení, připomínky a poznámky k práci. Dále děkuji Ing. Ondřeji Dolejšovi a Ing. Jaroslavovi Honců, CSc. za cenné rady. V neposlední řadě děkuji svým rodičům za to, že mě podporovali ve studiu i při tvorbě této práce.

Abstrakt

Cílem této práce je obeznámit čtenáře s realizací dálkového řízení modelu výtahu, který slouží jako učební pomůcka pro studenty katedry řídící techniky. Model se skládá ze tří svisle umístěných trubek, v nichž jsou pomocí proudu vzduchu nadnášeny válečky představující kabiny výtahu. Pro účely výuky je poloha jednoho z výtahů snímána i analogovým ultrazvukovým čidlem a polohy všech jsou pak snímány optickými čidly, ohraničující jednotlivá podlaží. Aktuální stav modelu je monitorován webkamerou a je tak umožněno distanční vzdělávání. Řízení je zajištěno programovatelným automatem firmy WAGO, připojeném na síť Internet. Mezi další úkoly, při realizaci této diplomové práce, patřily mimo jiné, tvorba plošných spojů řídící otáčení větráčků, identifikace modelu a realizace zpětnovazebního řízení, návrh demonstračního programu pro řízení všech tří výtahů a visualizace modelu výtahu.

Abstract

The purpose of this thesis is to present the realization of a distant control of an elevator model, which is used as an educational tool for the students of the Department of Control Engineering. The model consists of three vertically placed pipes where the cylinders, representing the cabs of an elevator, are levitated by the floating air. For the educational purposes is the actual position one of the elevators measured by an analog ultrasonic sensor and the locations of all of the cabs are scanned by optical sensors, denoting respective floors. The model is observed by a web-camera, which makes the distant control possible. The control of the model is provided by a programmable logic automat from the firm WAGO, connected to the Internet. Another tasks of the work presented here were to manufacture the printed circuits controlling the fan, to identify the elevator model, to design a feedback controller, to design a demonstrational program which controls all three elevators and to visualise the model.

Obsah

Úvod **11**

1 Dálkové řízení **12**

1.1 Internet 13

1.1.1 Protokol IP 14
1.1.2 Protokol TCP 15

2 Model výtahu **16**

2.1 Popis modelu 16

2.2 Ovládací panel 20

2.2.1 Převod 24 V PLC logiky na TTL logiku 23
2.2.2 Zapojení LED diod ovládacího panelu 23

2.3 Plošné spoje pro generování PWM 24

2.3.1 „Digitální“ plošný spoj 24
2.3.2 „Analogový“ plošný spoj 28

2.4 Konektory 30

3 Zpětnovazební řízení **31**

3.1 Pohybová rovnice válečku 31

3.2 Identifikace ARX modelu 32

3.2.1 ARX model 32
3.2.2 Metoda nejmenších čtverců 32

3.3 Návrh regulátoru 34

3.3.1 Regulátor P 34
3.3.2 Regulátor PD 35
3.3.3 Regulátor PI 36
3.3.4 Regulátor PID 36

4 PLC WAGO **38**

4.1 Použité moduly 39

4.2 Software 39

4.2.1 BootP Server a síťové nastavení 39

4.2.2 WAGO-I/O-PRO 32 40

5 Realizace dálkového řízení **45**

5.1 Algoritmus pro řízení modelu výtahu 45

5.1.1 Koncepce programu 46

5.1.2 Popis programu a jednotlivých funkčních bloků 46

5.2 Vizualizace 50

5.3 Webová kamera AXIS 2100 52

5.3.1 Připojení k síti Ethernet 53

6 Závěr **56**

Literatura **58**

Příloha A **59**

Příloha B **60**

Seznam obrázků

Obr. 2.1: Model výtahu ... 16
Obr. 2.2: Větráček a nasávací tunel ... 17
Obr. 2.3: Optické čidlo firmy SENS ... 17
Obr. 2.4: Ultrazvukový snímač polohy ... 19
Obr. 2.5: Ovládací panel ... 20
Obr. 2.6: Číslování pinů konektoru ovládacího panelu ... 20
Obr. 2.7: Zapojení vývodů IO MH 7447 ... 22
Obr. 2.8: Zapojení sedmisegmentového ... 22
Obr. 2.9: Přizpůsobení úrovně logiky ... 23
Obr. 2.10: Zapojení LED diod ... 23
Obr. 2.11: Schéma digitálního přizpůsobovacího členu ... 25
Obr. 2.12: Generátor pilového napětí ... 25
Obr. 2.13: PWM pro „digitální“ plošný spoj ... 27
Obr. 2.14: Schéma analogového přizpůsobovacího členu ... 29
Obr. 2.15: Vznik PWM „analogového“ plošného spoje ... 29
Obr. 2.16: Rozebíratelné konektory pro jeden výtah ... 30
Obr. 3.17: Síly působící na váleček ... 31
Obr. 3.18: Porovnání přechodových charakteristik simulovaného a reálného modelu .. 33
Obr. 3.19: Zpětnovazební regulace polohy válečku ... 34
Obr. 3.20: Přechodová charakteristika uzavřené smyčky s P regulátorem ... 35
Obr. 3.21: Přechodová charakteristika uzavřené smyčky s PD regulátorem ... 35
Obr. 3.22: Přechodová charakteristika uzavřené smyčky s PI regulátorem ... 36
Obr. 3.23: Přechodová charakteristika uzavřené smyčky s PID regulátorem ... 37
Obr. 4.24: WAGO I/O SYSTEM 750 ... 38
Obr. 4.25: Program BootP Server ... 40
Obr. 4.26: Nastavení IP adresy ... 41
Obr. 4.27: Výběr typu PLC ... 41
Obr. 4.28: Vytvoření nového projektu ... 42
Obr. 4.29: Programování v IL ... 42

Obr. 4.30: Blok programu v LD 43
Obr. 4.31: Blok programu v FBD 43
Obr. 4.32: Blok programu v SFC 44
Obr. 4.33: Programování v ST 44
Obr. 5.34: Vývojové prostředí WAGO-I/O-PRO 32 45
Obr. 5.35: Celkový koncept programu 46
Obr. 5.36: Vizualizace v prostředí WAGO-I/O-PRO 32 51
Obr. 5.37: Dialogové okno pro editaci vlastností struktury 52
Obr. 5.38: Webová kamera AXIS 2100 53
Obr. 5.39: Závislost frekvence snímků na osvětlení 54
Obr. 5.40: Pohled na model výtahu webkamerou AXIS 55

Seznam tabulek

Tab. 1.1: Pozice protokolu ve vrstvách v rámci ISO-OSI 14
Tab. 2.2: Zapojení optického čidla 18
Tab. 2.3: Zapojení ultrazvukového snímače 19
Tab. 2.4: Popis I/O prvků ovládacího panelu 21
Tab. 2.5: Pravdivostní tabulka IO MH 7447 22
Tab. 2.6: Označení vývodů konektorů a jejich adresa v PLC 30
Tab. 5.7: Nastavení PID regulátoru 48
Tab. 5.8: Priority pro globální rozhodování 49
Tab. 5.9: Datové objemy snímků 54

Úvod

Dálkové řízení je v dnešní době jedna z nejvyžadovanějších a nejpoužívanějších technologií nezávisle na tom, zda se jedná o technologickou nebo manažerskou úroveň. V našem případě je k vzdálenému řízení použit model výtahu.

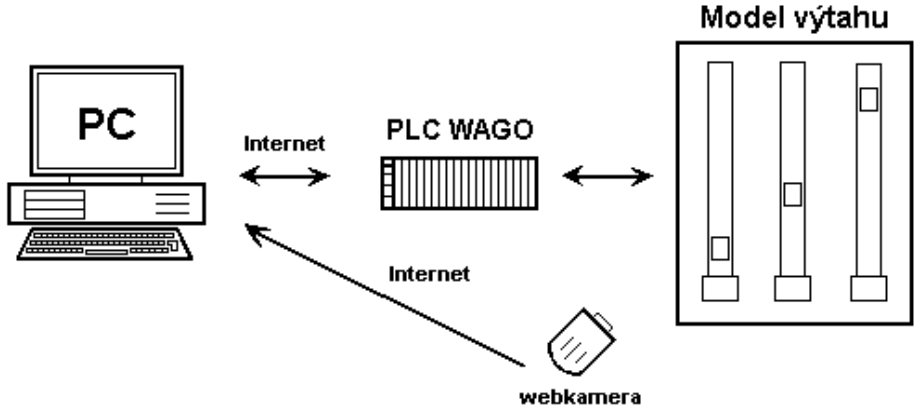
V reálném světě jsou požadavky na možnosti výtahu daleko větší než v dřívějších dobách a firmy nabízející montáž a servis výtahu, musí stále vyvíjet nové technologie. Mezi největší požadavky patří zejména monitoring a statistika používání výtahu. Vznikne-li závada na výtahu, předá výtah servisnímu centru, např. pomocí internetového spojení, hlášení o momentálním stavu a pravděpodobné příčině poruchy. Statistika slouží např. pro placení výtahu podle používání, což už dnes také není výjimkou. V budoucnu by mohla statistika vést k tzv. „předvídání“ výtahu, čímž by se značně snížila doba čekání na výtah a to hlavně ve výškových budovách.

Náš model výtahu je ozvláštněn pneumatickým nadnášením kabin a jako jeden z mála v místnosti K09 nabízí odzkoušení zpětnovazebního řízení a použití např. PID regulátoru v reálném provozu. S modelem se studenti seznámí v předmětech Řídící systémy a Návrh automatizovaných zařízení.

Samotné programování algoritmu pro řízení výtahu se uskutečňuje na PC pomocí aplikace WAGO-I/O-PRO 32, v které je možná i visualizace procesu.

Pomocí internetového spojení lze vytvořený program nahrát do PLC, které zajišťuje samotné ovládání vstupů a sledování výstupů modelu. Model je sledován webkamerou AXIS 2100. V praxi to tedy znamená, že se lze na model připojit z jakéhokoli místa poskytující internetové připojení, vzdáleně nahrát vytvořený program do řídícího PLC a sledovat on-line výsledek naprogramovaného algoritmu.

Následující obrázek zobrazuje obecnou strukturu dálkového připojení modelu.



1 Dálkové řízení

S rozvojem technologií, zvláště v oblasti IT, se zvýšily možnosti vzdáleného řízení. Existuje mnoho způsobů realizace. Je nutné si předem uvědomit, na jaké úrovni se bude vzdálené řízení používat a co se od něho očekává. Hlavním kritériem je úspěšně zvládnutá vzdálená komunikace. Základní požadavky na vzdálenou komunikaci v řízení jsou především tyto:

- **Časová odezva** – tj. čas, který uplyne od vzniku požadavku do vykonání skutečné akce v technologii. Délka této odezvy určuje, zda je možné řídit technologii v reálném čase.
- **Spolehlivost** – zaručení vykonání vzniklého požadavku a zobrazení správných hodnot z technologie. Tím je myšleno zamezení ztráty dat nebo jejich poškození při přenosu. To lze řešit použitím vhodných protokolů, které jsou založeny např. na potvrzování zpráv nebo takové, které zamezují kolizím na sběrnici.
- **Bezpečnost** – zamezení zneužití vzdálené komunikace nežádoucími subjekty. Obvykle se řeší šifrováním přenosu dat, přičemž zvláště důležité je šifrování přihlašovacích údajů do systému.
- **Dostupnost** – možnost využití již hotových veřejných sítí bez jakýchkoliv omezení. Splnění všech těchto požadavků zároveň není s dnešními dostupnými prostředky možné, neboť hlavním limitujícím faktorem je požadovaná časová odezva versus dostupnost připojení. V případe použití veřejných sítí, např. Internet, je dostupnost zajištěna téměř po celém světě, ale vesměs potřebné časové odezvy nelze vůbec zajistit.

V této diplomové práci se ke vzdálenému přístupu k řídícímu PLC a k monitoringu pomocí vizualizace a on-line webové kamery využívá sítě Internet, která je nejrozšířenější a nejdostupnější sítí na světě. Pokud bychom měli rozebrat předcházející požadavky na vzdálenou komunikaci, můžeme říci, že největší problém při použití komunikace přes Internet je časová odezva, která by v případě rychlejších procesů mohla být překážkou. U našeho modelu je ale rychlost takovéhoto spojení dostatečná, protože samotné řízení obstarává PLC, do kterého se řídící algoritmus vzdáleně pouze nahraje a všechny výpočty a operace obstarává samotné PLC. U vizualizace je

ovšem potřeba, aby se chování modelu dalo sledovat a řídit s co nejmenší časovou prodlevou. Ale i v tomto případě je rychlost komunikace pomocí Internetu dostačující. V následující kapitole je popsána základní charakteristika Internetu.

1.1 Internet

Síť internet využívá komunikačního standardu Ethernet. Ethernet je technologie, která byla vynalezena a uznána jako efektivní prostředek přenosu dat na poli kancelářské komunikace a informačních technologií, přičemž během krátké doby zaznamenal prudký vzestup v oblasti lokálních počítačových sítí na celém světě. Ethernet je vybaven vyšší úrovní komunikačního softwaru v souvislosti se standardem IEE 802.3, tj. TCP/IP (Transmission Control Protocol/Internet Protocol) pro umožnění komunikace mezi rozdílnými systémy. TCP/IP protokol nabízí vysoký stupeň spolehlivosti přenosu informace. Pro přenos dat v síti Ethernet se používá několik technologií s různými parametry, které se liší v typu přenosového média, přenosové rychlosti, délce segmentu a typu přenosu. Topologie sítě je závislá na použité přenosové technologii.

Ve standardu Ethernet mohou stanice přistupovat na sběrnici užitím metody CSMA/CD (Carrier Sense Multiple Access/Collision Detection).

- Carrier Sense: Vysílač odposlouchává provoz na sběrnici.
- Multiple Access: Na sběrnici může přistupovat několik vysílačů.
- Collision Detection: Jsou detekovány kolize.

Pracovní stanice sítě Ethernet může zahájit vysílání kdykoli, pokud zjistí, že žádná jiná pracovní stanice současně nevysílá. Pracovní stanice tak poslouchá „éter“ a pokud je prázdný, zahájí vysílání. Odtud také název Ethernet - „éterová síť“. V průběhu přenosu pracovní stanice neustále naslouchá a zjišťuje, jestli mezitím nezačala vysílat stanice jiná. Pokud dojde u vysílání ke kolizi (tedy pokud vysílají dvě stanice současně), počká každá pracovní stanice určitý náhodně zvolený časový interval a poté se o vysílání pokusí znovu. Při opakovaném vysílání může pochopitelně také dojít k další kolizi, pak nastupují různé speciální algoritmy. Jeden z takových algoritmu spočívá v čekání o exponenciálně narůstající interval.

Bohužel metoda CSMA/CD není pro průmyslové požadavky dostatečně spolehlivá, zvláště nelze vůbec zaručit časovou odezvu. Spolehlivost do jisté míry zajišťují až různé komunikační protokoly jako IP, TCP. Tabulka 1.1 zobrazuje pro přehlednost pozice protokolu ve vrstvách ISO-OSI.

Transportní vrstva	TCP a UDP
Síťová vrstva	IP
Spojová vrstva	Ethernet
Fyzická vrstva	

Tab. 1.1: Pozice protokolu ve vrstvách v rámci ISO-OSI

1.1.1 Protokol IP

Internet se skládá z celé řady hostitelských počítačů a také zařízení, kterým se říká směrovače (routers). Propojením některých počítačů vznikají sítě, přičemž některé ze sítí tvoří ostrovy. Směrovače pak propojují tyto ostrovy sítí. Každý jednotlivý hostitelský počítač i směrovač musí mít jedinečnou adresu, podle které se na toto zařízení a zpět z něj dají posílat datové pakety. Je nutné vhodně směrovat zprávy sítí, aby došly do cíle v co možná nejkratší době. To má na starosti protokol IP (Internet Protocol), který popisuje formát pro přiřazování IP adres. Současně definuje mechanismus pro nespolehlivý přenos dat, kterému se říká datagram. Datagramy IP jsou v protokolech nižších úrovní zapouzdřeny jako součást dat. Těmito protokoly jsou například Ethernet, Token-ring atd. Protokol IP verze 4 (IPv4) definuje síťové adresy jako 32 bitové číslo. Adresy IP přiděluje úřad Internet Assigned Numbers Authority (IANA). Adresy IP se dělí do celé řady kategorií, kterým se říká třídy. Avšak pro samotnou komunikaci mezi aplikacemi na dvou různých počítačích samotná IP adresa nestačí. Každá aplikace musí mít možnost identifikovat svůj protějšek, se kterým chce komunikovat, protože na jednom počítači běží obvykle souběžně několik různých aplikací.

Tyto aplikace mohou zpravidla komunikovat s různými aplikacemi na různých vzdálených počítačích nebo i na jednom stejném počítači. Protokoly UDP a TCP identifikují odesílající a přijímající aplikaci pomocí portu, zdrojová a cílová IP adresa pak identifikuje počítače, na kterých tyto aplikace běží. Datový proud mezi dvěma aplikacemi potom identifikuje množina následujících čtyř hodnot:

- zdrojová IP adresa,
- číslo zdrojového portu,
- cílová IP adresa,
- číslo cílového portu.

Některým důležitým a známým aplikacím jsou čísla portu již pevně přidělena. Tato předem přiřazená čísla portu se používají pouze na serveru. Čísla portu na klientech se alokují dynamicky. Jediný port na serveru muže mít otevřených několik proudů, protože se na něj může připojit i několik klientů. Čísla portů získávají postupně na důležitosti, protože na jejich základě funguje filtrování firewallu.

1.1.2 Protokol TCP

Protokol IP poskytuje poměrně nespolehlivý přenos dat. Protokol TCP oproti tomu představuje téměř již spolehlivý, bezchybný, plně duplexní kanál mezi dvěma počítači. Pro přenos se využívá služeb protokolu IP, současně však definuje určité další mechanismy, které se starají o ztracené a zdvojené datagramy IP. Protokol TCP zabezpečuje také správné složení jednotlivých datagramů IP, které dorazí mimo správné pořadí. Pro zabezpečení těchto funkcí rozděluje protokol TCP vstupní proud dat do paketu, které musí být natolik malé, aby se vešly dovnitř datagramu IP. Těmto paketům se říká segmenty. Datagramy IP se očíslují a odešlou se. Jestliže z cíle nedorazí potvrzení o jejich přijetí, odesílají se znovu. Odesílatel vždy po určitou dobu čeká a pokud neobdrží potvrzení o přijetí, odešle datagram znovu. Avšak když na sběrnici přistupuje současně velké množství počítačů (stanic), dochází k tak velkému množství kolizí, že ani protokol TCP není schopen vždy zabezpečit přenos dat, natož zaručit určité časové odezvy. Podrobnější popis fungování internetu a jednotlivých komunikačních protokolů a lze nalézt např. v [10].

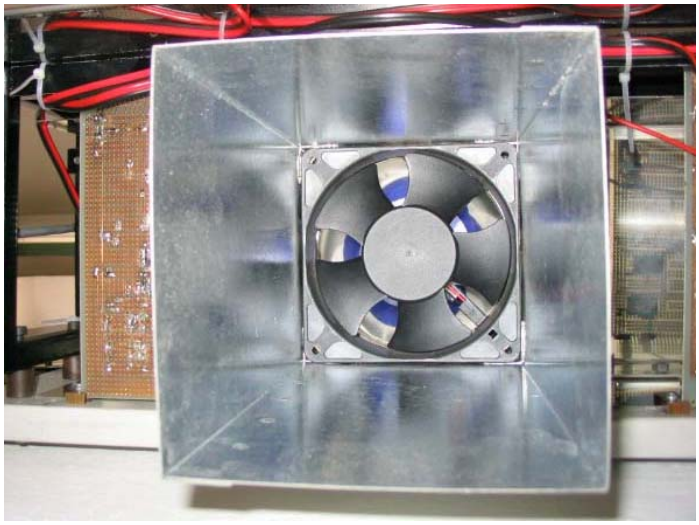
2 Model výtahu

2.1 Popis modelu

Model výtahu se skládá ze tří svislých průsvitných **trubek** o průměru 9 cm a délce 200 cm, v každé z nich je umístěn umělohmotný **váleček**, který představuje kabinu výtahu (obr. 2.1). Experimentálně bylo váleček nutné vyvážit tak, aby jeho pohyb byl dostatečně plynulý a v trubce vlivem nízké hmotnosti nekmital.

Obr. 2.1: Model výtahu

Na spodní straně každé trubky se nachází **větráček** (obr. 2.2), jenž tyto válečky pomocí proudu vzduchu nadnáší. Z důvodu co možná největší homogenity nasávaného vzduchu jsou pod větráčky umístěné tzv. **nasávací tunely** (obr. 2.2), které jsou pro optimální a stálý proud vzduchu nutné. Bez těchto tunelů by se větráčky navzájem ovlivňovaly a proudění vzduchu v trubicích by bylo daleko více nehomogenní, jak bylo při realizaci modelu zjištěno.



Obr. 2.2: Větráček a nasávací tunel

Na trubce jsou dále rozmístněna **optická čidla** firmy SENS typ WT170-P112 (obr. 2.3), která ohraničují jednotlivá podlaží. U „digitálních" výtahů obsahuje každé patro, kromě nejvyššího, po dvou těchto snímačích. U „analogového" výtahu ohraničuje každé patro pouze jedno takovéto čidlo.

Obr. 2.3: Optické čidlo firmy SENS

Tato čidla slouží jako optická brána a při přerušení světelného paprsku dávají na svém výstupu +24V. Citlivost tohoto snímače lze kolečkem na horní straně čidla nastavit v rozmezí 10-550 mm.

Snímač umožňuje celkem čtyři způsoby zapojení:

1. detekce předmětu je signalizována na výstupu vysokou logickou úrovní a výstup je realizován PNP tranzistorem s otevřeným kolektorem
2. detekce předmětu je signalizována na výstupu nízkou logickou úrovní a výstup je realizován PNP tranzistorem s otevřeným kolektorem
3. detekce předmětu je signalizována na výstupu vysokou logickou úrovní a výstup je realizován NPN tranzistorem s otevřeným kolektorem
4. detekce předmětu je signalizována na výstupu nízkou logickou úrovní a výstup je realizován NPN tranzistorem s otevřeným kolektorem

V našem případě jsou všechna tato optická čidla zapojena prvním způsobem a předmět je detekován vysokou logickou úrovní. Tohoto nastavení docílíme zapojením vodičů čidla podle tabulky 2.2

Hnědý	+ 24V
Bílý	+ 24V
Černý	Uvýst
Modrý	GND

Tab. 2.2: Zapojení optického čidla

Na vrcholu prostředního tunelu je pro účel výuky navíc umístěn **ultrazvukový snímač polohy** firmy MICROSONIC typ mic-31/IU/HV/M30 (obr. 2.4). Výstupem tohoto snímače je analogový signál 4–20mA a 2–10V. Tento snímač je pro správnou funkci nutné nakalibrovat. Je-li výtah na dně trubky, výstupem čidla je napětí 2.4 V a naopak je-li v nejvyšším patře výstupní napětí má hodnotu 9.2 V. Výstup tohoto čidla je přiveden do analogového vstupního modulu programovatelného automatu (dále jen PLC) 750-467 s adresou IW0. Zapojení vodičů tohoto čidla nalezneme v tabulce 2.3

Obr. 2.4: Ultrazvukový snímač polohy

Parametry použitého ultrazvukového snímače:

Napájecí napětí: 10-30V

Dosah: 30-250cm

Max. zatěžovací proud: 500mA

Ultrazvuková frekvence: 320kHz

Přesnost: ±1mm

Časová odezva výstupu: 40ms

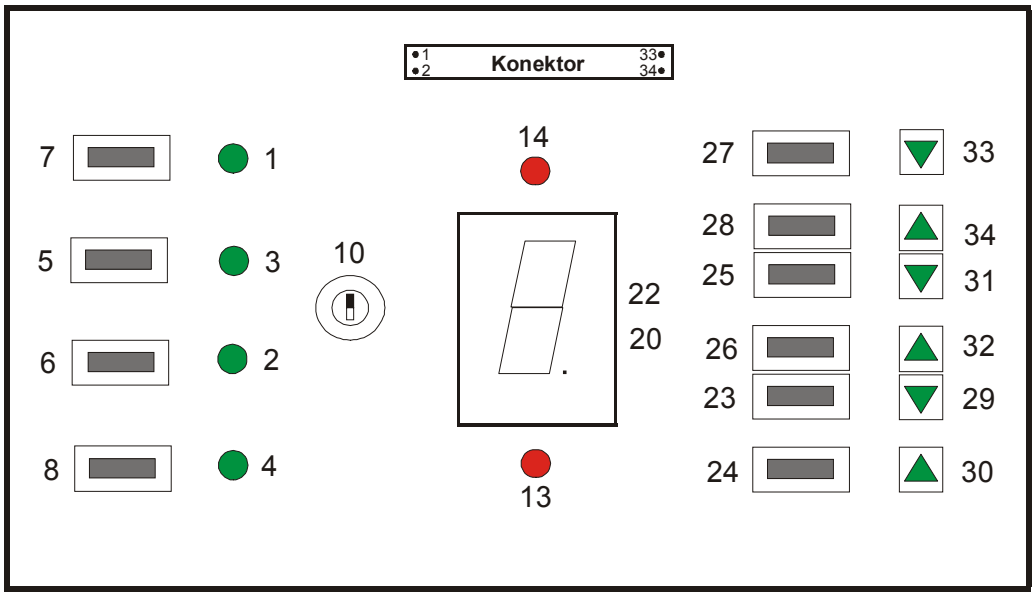
Ochrana proti zkratu a přepólování

Hnědý	+ 24V
Bílý	Uvýst
Šedý	nezapojen
Černý	nezapojen
Modrý	GND

Tab. 2.3: Zapojení ultrazvukového snímače

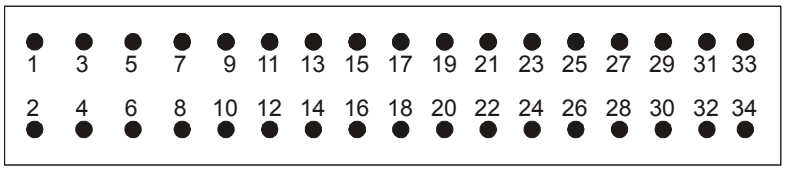
2.2 Ovládací panel

Ovládací panely se nachází na přední straně modelu, přičemž každá výtahová šachta má jeden ovládací panel (obr. 2.5). V levé polovině panelu jsou tlačítka představující tlačítka uvnitř kabiny výtahu, kdežto tlačítka vpravo představují umístění v jednotlivých patrech.



Obr. 2.5: Ovládací panel

Tlačítky 5–8, jenž představují umístění v kabině výtahu, volíme požadované patro. Stisk těchto tlačítek signalizují LED diody 1–4. Přepínač 10 slouží jako detekce obsazenosti výtahu. V reálném případě představuje tento přepínač obvykle tlakový senzor v podlaze kabiny výtahu. Diody 13 a 14 signalizují momentální pohyb výtahu a to směrem dolu nebo nahoru. Ovládací a signalizační prvky 23 až 34 představují umístění v příslušném patře a slouží pro přivolání výtahu s úmyslem jízdy nahoru nebo dolu. Vše je vyvedeno na 34-pinový konektor (obr. 2.6). V tabulce 2.4 lze nalézt popis všech vstupně/výstupních (dále jen I/O) prvků ovládacího panelu i s číselným označením pozice připojení na konektor.

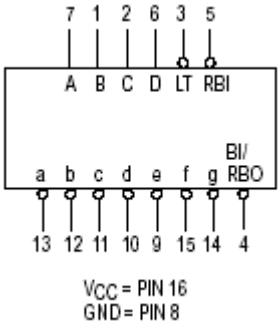


Obr. 2.6: Číslování pinů konektoru ovládacího panelu

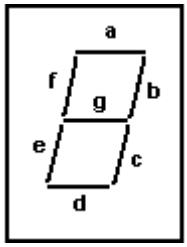
Číslo pinu na konektoru	Typ I/O prvku	Input/Output (z hlediska PLC)	Popis
1	LED zelená	O	signalizace stisku tlačítka pro jízdu do 3. patra
2	LED zelená	O	signalizace stisku tlačítka pro jízdu do 1. patra
3	LED zelená	O	signalizace stisku tlačítka pro jízdu do 2. patra
4	LED zelená	O	signalizace stisku tlačítka pro jízdu do 0. patra
5	Tlačítko	I	kabina výtahu – úmysl jízdy do 2. patra
6	Tlačítko	I	kabina výtahu – úmysl jízdy do 1. patra
7	Tlačítko	I	kabina výtahu – úmysl jízdy do 3. patra
8	Tlačítko	I	kabina výtahu – úmysl jízdy do 0. patra
9	nezapojen		
10	Přepínač	I	určení obsazenosti výtahu: log 1 – výtah obsazen log 0 – výtah prázdný
11	nezapojen		
12	nezapojen		
13	LED červená	O	signalizace pohybu kabiny směrem dolu
14	LED červená	O	signalizace pohybu kabiny směrem nahoru
15	nezapojen		
16	nezapojen		
17	nezapojen		
18	nezapojen		
19	nezapojen		
20	logická úroveň	O	řídící vstup B pro ovládání sedmisegmentového displeje – vyšší váha
21	nezapojen		
22	logická úroveň	O	řídící vstup A pro ovládání sedmisegmentového displeje – nižší váha
23	Tlačítko	I	tlačítko v 1. patře – úmysl jízdy nahoru
24	Tlačítko	I	tlačítko v 0. patře – úmysl jízdy nahoru
25	Tlačítko	I	tlačítko v 2. patře – úmysl jízdy nahoru
26	Tlačítko	I	tlačítko v 1. patře – úmysl jízdy dolu
27	Tlačítko	I	tlačítko v 3. patře – úmysl jízdy dolu
28	Tlačítko	I	tlačítko v 2. patře – úmysl jízdy dolu
29	LED zelená	O	signalizace stisku tlačítka v 1. patře pro jízdu nahoru
30	LED zelená	O	signalizace stisku tlačítka v 0. patře pro jízdu nahoru
31	LED zelená	O	signalizace stisku tlačítka v 2. patře pro jízdu nahoru
32	LED zelená	O	signalizace stisku tlačítka v 1. patře pro jízdu dolu
33	LED zelená	O	signalizace stisku tlačítka v 3. patře pro jízdu dolu
34	LED zelená	O	signalizace stisku tlačítka v 2. patře pro jízdu dolu

Tab. 2.4: Popis I/O prvků ovládacího panelu

Sedmisegmentový displej ovládacího panelu je řízen integrovaným obvodem (dále jen IO) MH 7447 (obr. 2.7), což je budič/dekodér BCD kódu [13] na sedmisegmentový displej (obr. 2.8). Pro tento obvod, který je jako jediný typu TTL (5V logika, [12]), jsme museli navrhnout přizpůsobení logického signálu (podrobněji v kapitole 2.2.1). IO MH 7447 je napájen napětím + 5V, jenž se získává pomocí IO 7805, což je stabilizátor napětí z +24V na +5V. Tento obvod je umístěn na přizpůsobovacím plošném spoji prostředního výtahu a to pro nedostatek místa na ovládacím panelu a kvůli snadnějšímu rozvodu na zbylé dva okrajové ovládací panely.



Obr. 2.7: Zapojení vývodů IO MH 7447



Obr. 2.8: Zapojení sedmisegmentového displeje

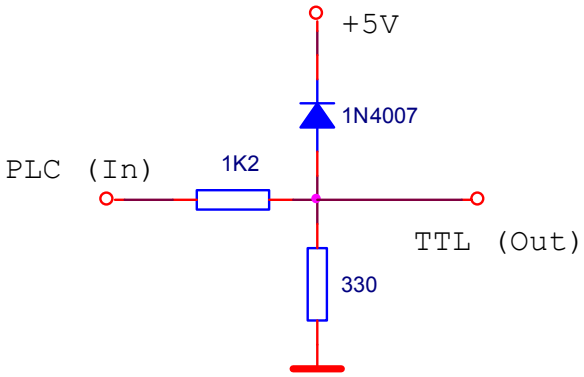
Z pravdivostní tabulky dekodéru MH 7447 (tab. 2.5) je vidět, že volbou číslování pater od 0 do 3 potřebujeme na ovládání pouze 2 bity PLC a to vstup A a B přičemž vstupy C a D jsou uzeměny. Námi použitý sedmisegmentový displej je se společnou anodou, což znamená, že se jednotlivé segmenty rozsvítí při logické 0. Do propojovací cesty jednotlivých vývodů z IO MH 7447 k odpovídajícím vstupům sedmisegmentového displeje, je nutné zařadit ochranné rezistory, jejichž odpor činí 270 Ω.

D	C	B	A	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$	Zobrazované číslo
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
. .	. .	. .	. .	. .	. .	. .	. .	. .	. .	. .	. .

Tab. 2.5: Pravdivostní tabulka IO MH 7447

2.2.1 Převod 24 V PLC logiky na TTL logiku

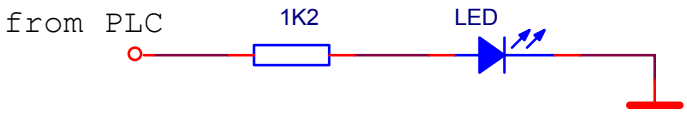
Abychom mohli signálem z PLC, jenž má 24V logiku, ovládat IO MH 7447, jenž je 5V logiky, musíme signál napěťově přizpůsobit. Na obr. 2.9 je vidět schéma námi použitého přizpůsobení.



Obr. 2.9: Přizpůsobení úrovně logiky

2.2.2 Zapojení LED diod ovládacího panelu

Schéma zapojení všech LED diod ovládacího panelu lze vidět na obr. 2.10. Hodnotou sériově zapojeného ochranného rezistoru regulujeme proud LED diodou a určujeme tak její svítivost. V našem případě činí hodnota odporu 1,2kΩ.



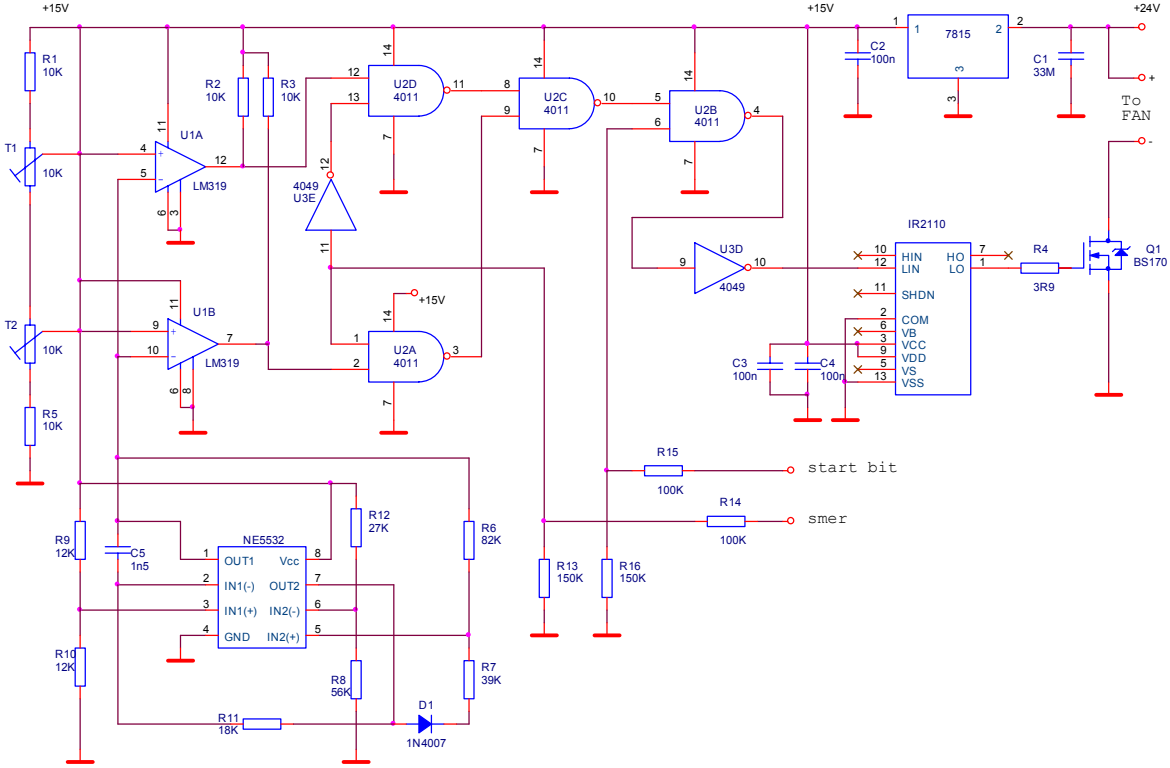
Obr. 2.10: Zapojení LED diod

2.3 Plošné spoje pro generování PWM

Směr pohybu každého výtahu je řízen rychlostí otáček větráčku. Pro řízení otáček větráčku je použita pulsní šířková modulace (dále jen PWM) [1]. Abychom nemuseli PWM generovat softwarově, navrhli jsme k tomuto generování zvláštní plošné spoje. Tyto plošné spoje jsou totožné pro oba krajní výtahy, kde se využívá pouze 2 nastavení PWM a to pro pohyb válečku nahoru nebo dolu. Pohyb prostředního výtahu je ovládán analogovým členem PLC a přizpůsobovací obvod je tak o něco jednodušší.

2.3.1 „Digitální“ plošný spoj

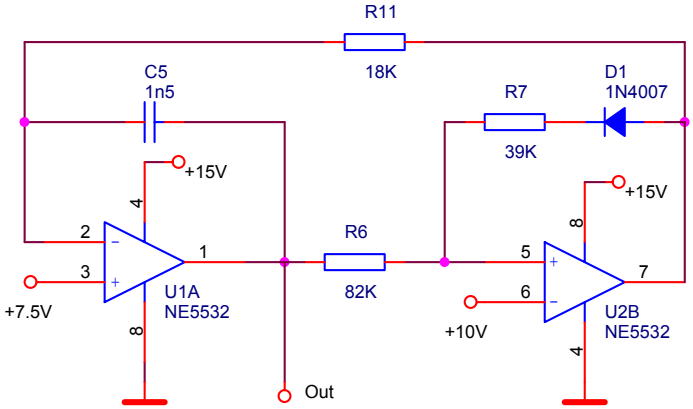
Tzv. „digitální“ plošný spoj je umístěn za čelním panelem modelu a slouží ke generování PWM o frekvenci f=10kH a k samotnému spínání výkonového tranzistoru FET [11]. Ke generování PWM je třeba generátor pilového napětí a dostatečně rychlý komparátor, v našem případě IO LM319 [14]. Ze schématu (obr. 2.11) je vidět, že pilové napětí o frekvenci 10kHz a napětí „špička-špička“ 8V s offsetem +2V je přivedeno na vstup IO LM319, kde se komparuje s pevným stejnosměrným napětím. Výstupem z LM319 je tedy obdélníkové napětí o frekvenci 10kHz a střídě dané poměrem odporů trimrů T1 a T2. Na výstupy obou LM319 je paralelně přivedeno napětí +15V, čímž se zvedne úroveň výstupního napětí na hodnotu vhodnou pro další zpracování. Jelikož se váleček musí pohybovat nahoru i dolu, je potřeba mít dvě PWM o různých střídách, které jsou přepínány podle požadovaného směru pohybu. Tento směr je určen jedním bitem z PLC přičemž log. 1 znamená směr nahoru a log. 0 směr dolu. Dalším bitem z PLC (start bit) se zapíná a vypíná chod větráčku. IO IR2110 [14] je určen ke konečnému přizpůsobení logického signálu, který přes tranzistor MOSFET BS170 [14] spíná napětí +24V na větráčku. K napájení celého obvodu se využívá stejnosměrné napětí +24V, které je pomocí IO 7815 stabilizováno na +15V. Toto napětí je dále použito pro napájení všech IO a přes trimry T1 a T2 také přivedeno na vstupy IO LM319. V dalších odstavcích probereme jednotlivé části obvodu důkladněji.



Obr. 2.11: Schéma digitálního přizpůsobovacího členu

Generátor pilového napětí

Generátor pilového napětí je založen na IO NE5532 [14] a pro svou funkci využívá invertující integrátor a komparátor napětí s hysterezí (obr. 2.12).



Obr. 2.12: Generátor pilového napětí

Pro výpočet hodnot rezistorů a kondenzátoru lze použít následující vztahy.

Invertující integrátor

$$u_{OUT}(t) = -\frac{1}{CR}\int u_{IN-}(t)dt + U_C = -\frac{1}{CR}u_{IN-}(t)\cdot t + U_C,$$

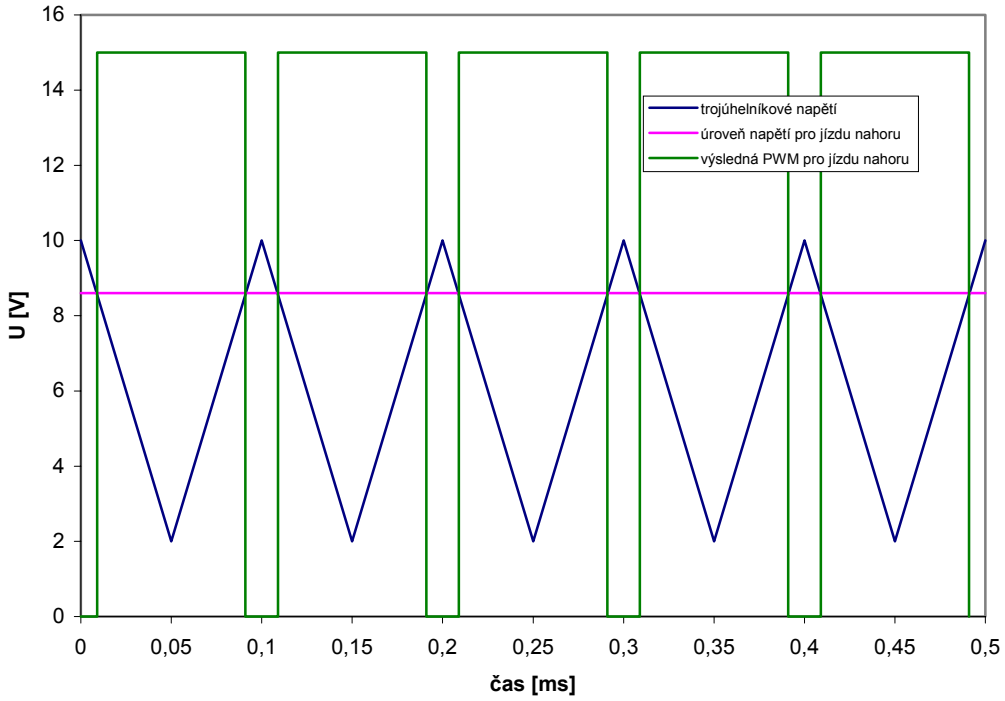
kde $u_{OUT}(t)$ je výstupní napětí integrátoru (offset integrátoru, =2V),
$u_{IN-}(t)$ je napětí na invertujícím vstupu integrátoru (=15V),
t je čas (požadovaná frekvence = 10kHz $\Rightarrow$ t = 50µs).
Po dosazení dostáváme $C{\cdot}R$=36,5×10^{-9}.
Volbou hodnoty rezistoru *R11*=18kΩ dostáváme hodnotu kondenzátoru *C5*=1,5nF.

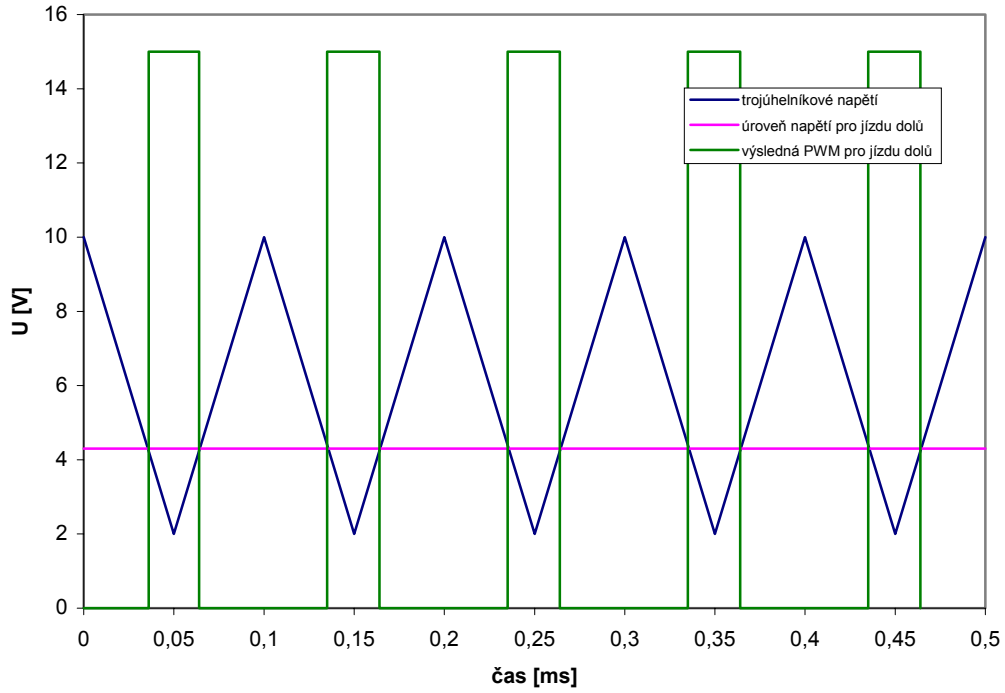
Komparátor s hysterezí

$$U_{IN+} = \left(U_{OUT} - U_{D1} - U_i\right)\times\frac{R6}{R6+R7}$$

kde U_{IN+} je napětí na neinvertovaném vstupu komparátoru s hysterezí (=10V),
U_{OUT} je napětí na výstupu komparátoru s hysterezí (=15V),
U_{D1} je napětí na diodě (=0,7V),
U_i je výstupní napětí integrátoru.
Po dosazení vyjde rovnost: 10*R7* = 4,3*R6*.
Volbou hodnoty rezistoru *R6*=82kΩ z řady R12 dostaneme i odpor *R7*=39kΩ.

V našem případě odpovídají hodnoty rezistorů a kondenzátorů generátoru pilového napětí frekvenci 10kHz a rozsahu napětí 2 až 10 V. Vznik PWM u digitálního přizpůsobovacího plošného spoje pro jízdu nahoru a dolu je znázorněn na obr. 2.13. Bližší popis zapojení a funkce tohoto generátoru lze nalézt v [1].





Obr. 2.13: PWM pro „digitální“ plošný spoj

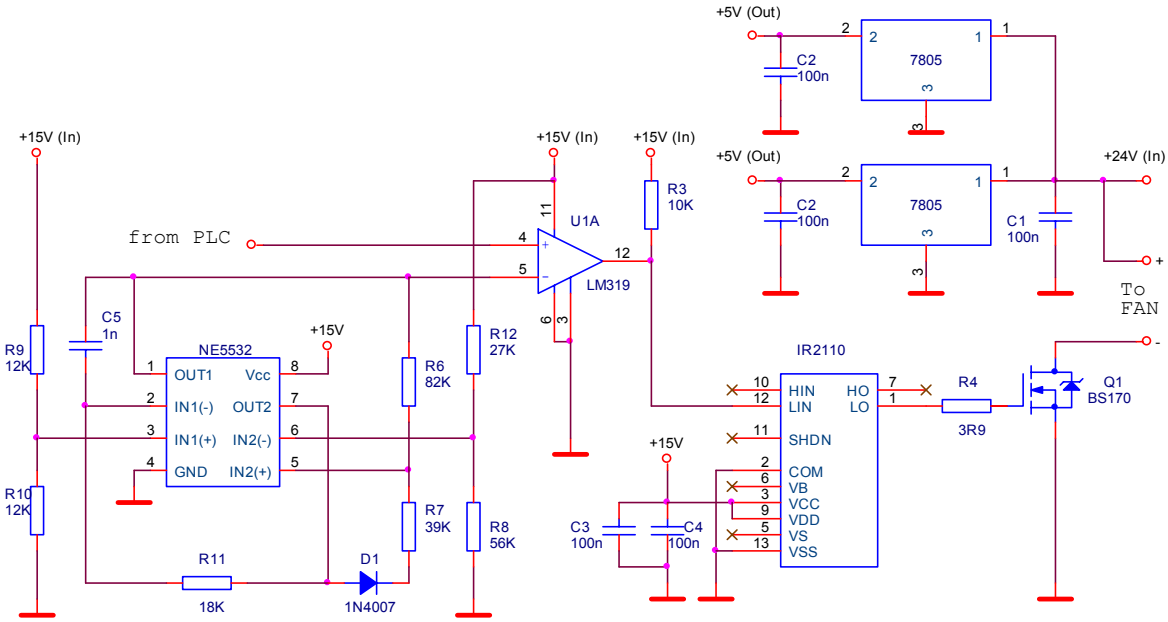
Logika obvodu

Směr jízdy výtahu řídíme jediným bitem PLC. Tento bit tedy přivádí na větráček PWM o střídě představující jízdu nahoru nebo jízdu dolu. Nejdříve musíme přizpůsobit logický signál PLC (24V logika) na signál CMOS (15V logika) [12]. Na to stačí zařadit jednoduchý dělič napětí na vstup „digitálního“ plošného spoje (obr. 2.11). Samotné přepínání je potom velmi jednoduché. Přizpůsobený logický signál rozdělíme na dva, přičemž jeden negujeme pomocí IO 4049 (CMOS, NOR). K samotnému „propuštění“ signálu pak dochází na IO 4011 (CMOS, NAND), kde využíváme skutečnosti, že signál na jednom vstupu (PWM) neprojde, pokud je na druhém vstupu (ovládací bit z PLC) logická 1 a naopak.

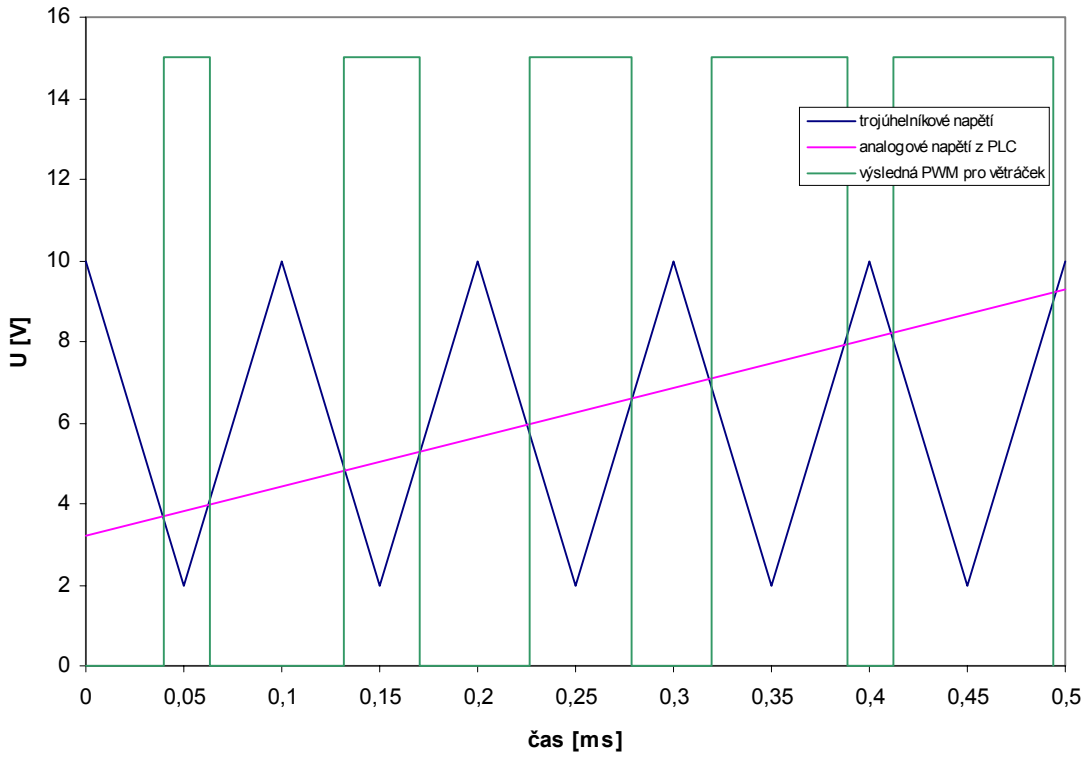
Abychom mohli chod větráčku ovládat dálkově, byla na výstup směru přepínání zařazena logika pro úplné vypnutí chodu větráčku. Signál pro směr jízdy nahoru nebo dolu je přiveden na vstup IO 4011, kde dochází k logickému součinu s tzv. start bitem. Nachází-li se tento bit v log. 1, signál směru jízdy „projde“ a větráček je v chodu.

2.3.2 „Analogový“ plošný spoj

Tzv. „analogový“ plošný spoj (obr. 2.14) vychází z koncepce „digitálního“ plošného spoje, ale díky analogovému řízení nevyžaduje logiku pro přepínání směrů a střída výstupní PWM se mění analogově. Napětí z analogového členu PLC v rozsahu 0-10V je přivedeno rovnou na neinvertující vstup IO LM319 kde se komparuje s pilovým napětím stejným jako u „digitálních“ plošných spojů. Výstupem z tohoto IO je opět obdélníkové napětí o frekvenci 10kHz a střídě dané velikostí vstupního napětí. Velikost výstupního obdélníkového napětí zvedneme paralelně připojeným ss. napětím +15V na velikost vhodnou pro vstup do IO IR2110, což je budič pro tranzistor MOSFAT BS170. Tento tranzistor opět spíná +24V na větráčku. Obvody 7805 jsou stabilizátory napětí z +24V na +5V a jak již bylo zmíněno slouží k napajení IO MH 7447 na ovládacích panelech. Princip vzniku „analogového“ PWM lze vidět na obrázku 2.15.



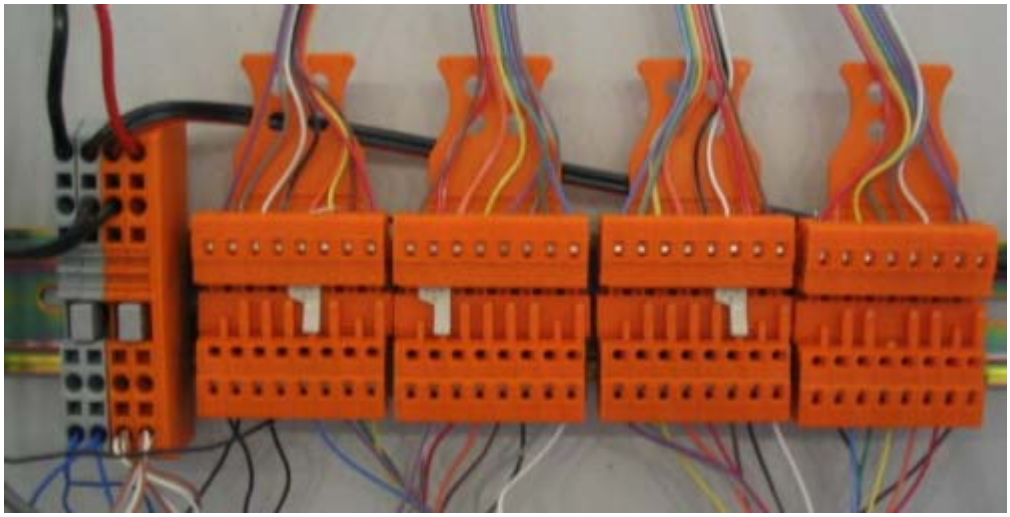
Obr. 2.14: Schéma analogového přizpůsobovacího členu



Obr. 2.15: Vznik PWM „analogového“ plošného spoje

2.4 Konektory

Veškeré I/O ovládacího panelu a plošných spojů a vývody optických čidel jsou připojeny k I/O modulům PLC přes rozebíratelné konektory firmy WAGO. Na každý výtah připadají celkem 4 osmi-kolíkové konektory umístěné na montážní liště (obr. 2.16). Zapojení jednotlivých pinů konektorů pro všechny 3 výtahy lze nalézt v tabulce 2.6. Čísla v posledním řádku tabulky označují I/O prvky z ovládacího panelu a vzájemně si odpovídají (viz kapitola 2.2).



Obr. 2.16: Rozebíratelné konektory pro jeden výtah

Označení vstupů do konektoru	Konektor	Adresa v PLC		
		levý výtah	střední výtah	pravý výtah
spodní čidlo	•	IX2.15	IX3.15	IX4.15
spodní čidlo	•	IX2.14	IX3.14	IX4.14
spodní čidlo	•	IX2.13	IX3.13	IX4.13
spodní čidlo	•	IX2.12	IX3.12	IX4.12
5	•	IX2.11	IX3.11	IX4.11
7	•	IX2.10	IX3.10	IX4.10
6	•	IX2.9	IX3.9	IX4.9
8	•	IX2.8	IX3.8	IX4.8
start bit větrá-	•	QX4.15		QX6.15
bit větráku	•	QX4.14	QW0	QX6.14
1	•	QX4.13	QX5.13	QX6.13
3	•	QX4.12	QX5.12	QX6.12
2	•	QX4.11	QX5.11	QX6.11
4	•	QX4.10	QX5.10	QX6.10
14	•	QX4.9	QX5.9	QX6.9
13	•	QX4.8	QX5.8	QX6.8
20	•	QX4.7	QX5.7	QX6.7
22	•	QX4.6	QX5.6	QX6.6
23	•	QX4.5	QX5.5	QX6.5
33	•	QX4.4	QX5.4	QX6.4
34	•	QX4.3	QX5.3	QX6.3
31	•	QX4.2	QX5.2	QX6.2
32	•	QX4.1	QX5.1	QX6.1
29	•	QX4.0	QX5.0	QX6.0
nezapojen	•	IX2.7	IX3.7	IX4.7
10	•	IX2.6	IX3.6	IX4.6
27	•	IX2.5	IX3.5	IX4.5
28	•	IX2.4	IX3.4	IX4.4
25	•	IX2.3	IX3.3	IX4.3
26	•	IX2.2	IX3.2	IX4.2
23	•	IX2.1	IX3.1	IX4.1
24	•	IX2.0	IX3.0	IX4.0

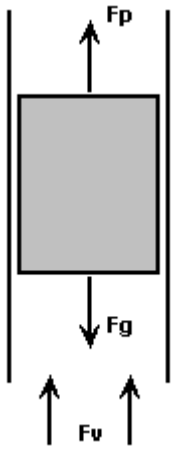
Tab. 2.6: Označení vývodů konektorů a jejich adresa v PLC

3 Zpětnovazební řízení

Pomocí ultrazvukového snímače polohy a analogových modulů PLC, lze polohu válečku regulovat spojitě. K regulaci potřebujeme znát matematický model, s jehož pomocí můžeme poté navrhnout vhodné zpětnovazební řízení. Problematiku dynamických systémů a jejich řízení lze nalézt v [2].

3.1 Pohybová rovnice válečku

Z obrázku 3.17 je vidět, že na výtah působí tři síly: síla gravitační F_G, síla pohybová F_P a síla proudícího vzduchu F_V.



Obr. 3.17: Síly působící na váleček

Stojí-li váleček na místě, můžeme pro rovnost sil psát vztah:

$$F_P - F_G + F_V = 0 \Rightarrow m \cdot a - m \cdot g + F_V = 0 \qquad (1)$$

kde m je hmotnost válečku, a je zrychlení a g je gravitační zrychlení.

Síla F_V je přímo úměrná vstupnímu napětí větráčku a pro náš případ ji můžeme psát jako $F_V = k \cdot u$, kde u je vstupní napětí a k je konstanta, zahrnující v sobě ztráty přenosu mezi proudícím vzduchem a pohybem válečku.

Upravenou pohybovou rovnici můžeme tedy psát ve tvaru:

$$m \cdot \ddot{x} - m \cdot g + k \cdot u = 0 \qquad (2)$$

Vytkneme-li konstantní hmotnost m a zanedbáme-li konstantu g, která nemá vliv na dynamiku systému (při návrhu regulátoru, ale tuto konstantu nelze zanedbat a musíme vhodně posunout akční zásah), dostáváme rovnici:

$$\ddot{x} + c \cdot u = 0 \tag{3}$$

kde $c = \dfrac{k}{m}$

Rovnici (3) můžeme přepsat do tzv. Laplaceových obrazů pro soustavu druhého řádu s neznámými koeficienty a_2, a_1, a_0 a b_0,:

$$(a_2 s^2 + a_1 s + a_0) \cdot Y = b_0 U \tag{4}$$

3.2 Identifikace ARX modelu

ARX model znamená autoregresní model s externím vstupem a slouží pro odhadování parametrů modelu. V následujících odstavcích jsou uvedeny pouze základní vztahy čerpané z [3] a [4].

3.2.1 ARX model

ARX model je popsán lineární diferenční rovnicí:

$$y(k) + a_1 \cdot y(k-1) + a_2 \cdot y(k-2) = b_2 \cdot u(k-2) + e(k) \tag{5}$$

kde *e(k)* je chyba měření, náhodná veličina s normálním rozložením $e \approx N(0, \sigma^2)$.

Rovnici (5) ARX modelu lze přepsat do přehlednějšího tvaru:

$$y = Z\theta + e \tag{6}$$

kde y je vektor měření výstupu $[y(1), y(2),..., y(k)]^T$
Z je matice dat $[z^T(1), z^T(2),...,z^T(k)]^T$
θ je vektor hledaných parametrů $[a_1, a_2, b_2]^T$
e je vektor chyb měření

3.2.2 Metoda nejmenších čtverců

Chceme-li nalézt vektor parametrů θ, můžeme použít metodu nejmenších čtverců, která minimalizuje střední kvadratickou odchylku. Předpokladem je, že chybou je zatížena pouze výstupní veličina y a matice dat Z je jednoznačně známa. Vycházíme z definice:

$$\min_{\theta \in R^n} \| Z\theta - y \|_2 \tag{7}$$

Tuto minimalizaci lze napsat jako:

$$\|e\|_2^2 = \|Z\theta - y\|_2^2 = (Z\theta - y)^T (Z\theta - y) = \theta^T Z^T Z\theta - \theta^T Z^T y - y^T Z\theta + y^T y \qquad (8)$$

Hledáme-li minimum tohoto výrazu, využijeme vztahů pro maticové derivace:

$$\frac{\partial}{\partial x}(x^T Ay) = Ay \qquad , \qquad \frac{\partial}{\partial y}(x^T Ay) = A^T x \qquad (9)$$

Pro hledané minimum tedy platí:

$$Z^T Z\theta - 2Z^T \theta y = 0 \qquad (10)$$

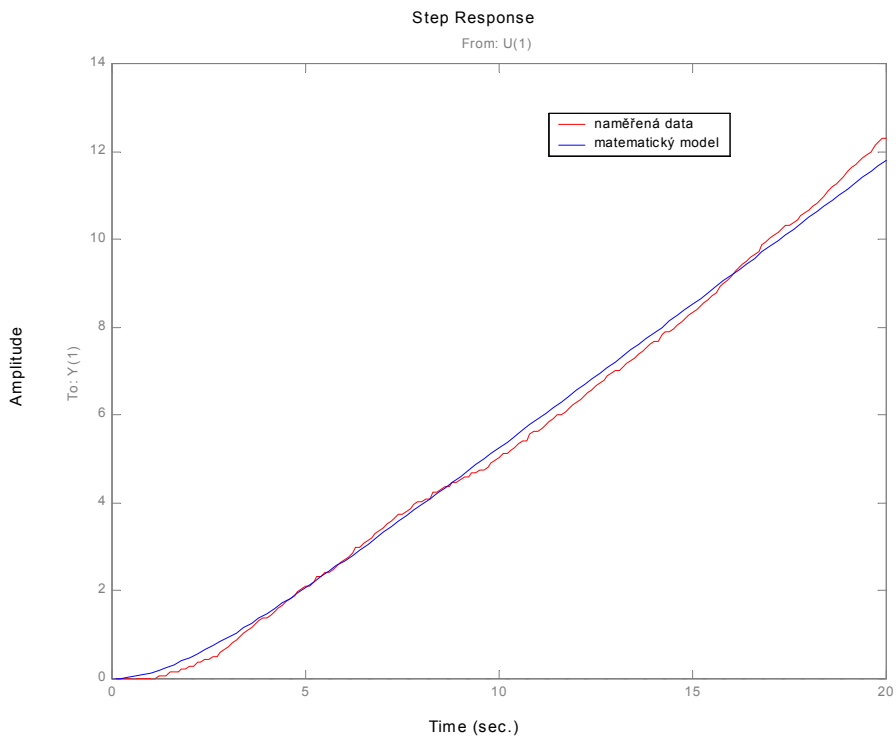
Hledané parametry tak získáme ze vztahu:

$$\theta = (Z^T Z)^{-1} Z^T y \qquad (11)$$

Dosazením naměřených dat do vztahu (11) dostáváme koeficienty: b_2=0.456, a_2=2, a_1=1
Podle rovnice (4) můžeme tedy psát přenos:

$$P(s) = \frac{0.456}{2s^2 + s} \qquad (12)$$

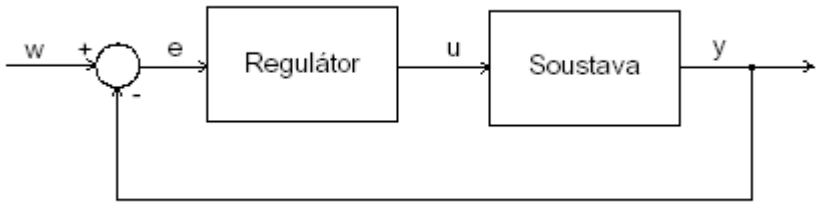
Pomocí programu Matlab a simulačního schéma soustavy lze ověřit správnost provedené identifikace. Přechodové charakteristiky simulovaného a reálného modelu můžeme pozorovat na obrázku 3.18. Soustava je astatická a má tedy integrační charakter.



Obr. 3.18: Porovnání přechodových charakteristik simulovaného a reálného modelu

3.3 Návrh regulátoru

Polohu válečku můžeme regulovat klasickým zpětnovazebním regulačním schématem znázorněném na obrázku 3.19.



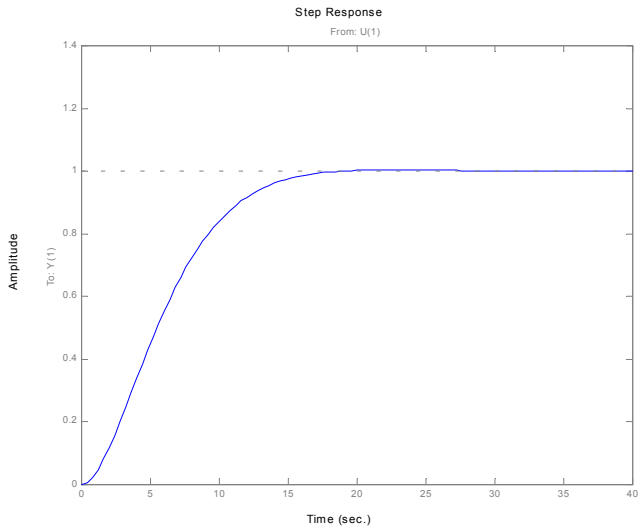
Obr. 3.19: Zpětnovazební regulace polohy válečku

kde w je řídící veličina
e je regulační odchylka
u je akční veličina
y je regulovaná veličina

Opět pomocí programu Matlab můžeme např. metodou geometrického místa kořenů charakteristického polynomu (dále jen GMK) najít nejvhodnější regulátor. Pro ilustraci navrhneme touto metodou všechny následující typy regulátoru. Problematikou hledání vhodného regulátoru se podrobněji zabývá např. [2]

3.3.1 Regulátor P

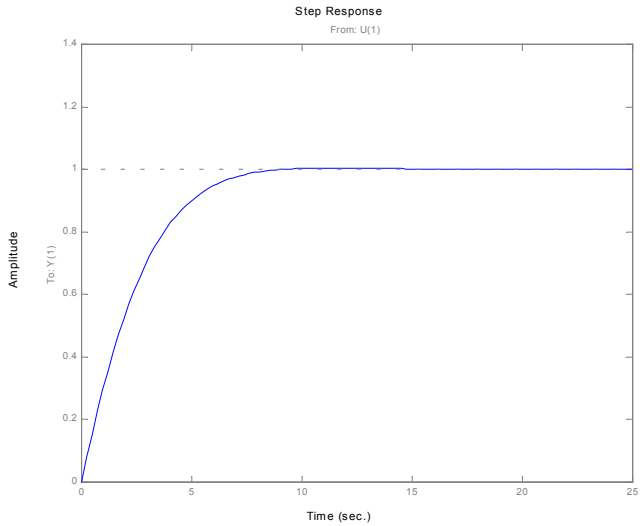
Přenos P regulátoru je $R = r_0$, z čehož je vidět, že tento regulátor pouze zesiluje. Pomocí GMK jsme určili R = 0,36. Přechodová charakteristika zpětnovazebního systému je vidět na obrázku 3.20. Zesílení jsme nastavili tak, aby nedošlo k překmitu.



Obr. 3.20: Přechodová charakteristika uzavřené smyčky s P regulátorem

3.3.2 Regulátor PD

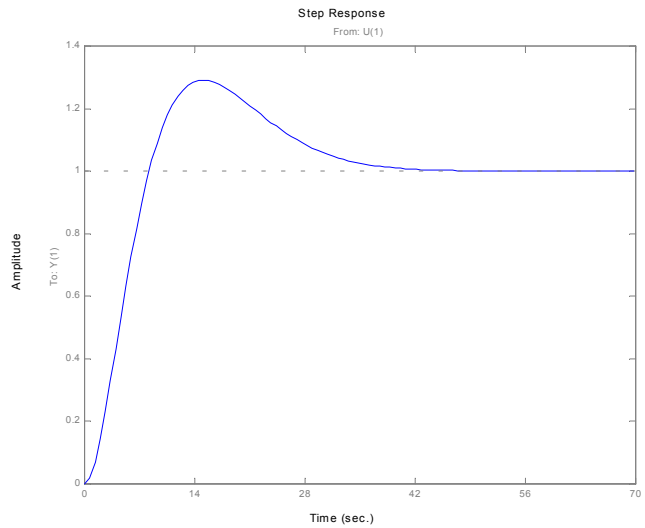
Přenos PD regulátoru je $R = r_0 + r_1 s$, kde r_0 je proporcionální složka a r_1 je složka derivační. Pomocí GMK jsme došli k nastavení: R = 1,5s + 0,975.



Obr. 3.21: Přechodová charakteristika uzavřené smyčky s PD regulátorem

3.3.3 Regulátor PI

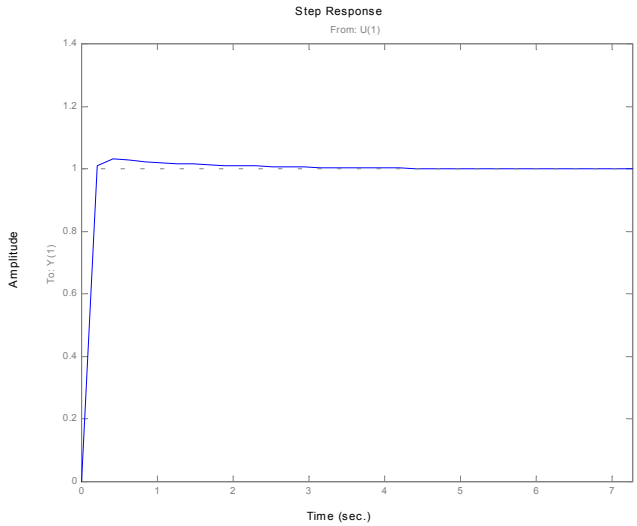
Přenos regulátoru je pro PI regulátor $R = r_0 + \frac{r_{-1}}{s}$, kde r_0 je proporcionální složka a r_{-1} je složka integrační. Pro nastavení $R = 0.43 + \frac{0.03}{s}$ je přechodová charakteristika následující.



Obr. 3.22: Přechodová charakteristika uzavřené smyčky s PI regulátorem

3.3.4 Regulátor PID

PID regulátor je nejpoužívanější regulátor v řídící technice. Jeho přenos je $R = r_0 + \frac{r_{-1}}{s} + r_1 s$. Pomocí GMK opět určíme nejvhodnější nastavení a to tak, aby doba ustálení a překmit byly co nejmenší. Výsledný regulátor je $R = 92.75 + \frac{26.5}{s} + 77.9s$ a přechodovou charakteristiku uzavřené smyčky s tímto regulátorem lze vidět na obr. 3.23.

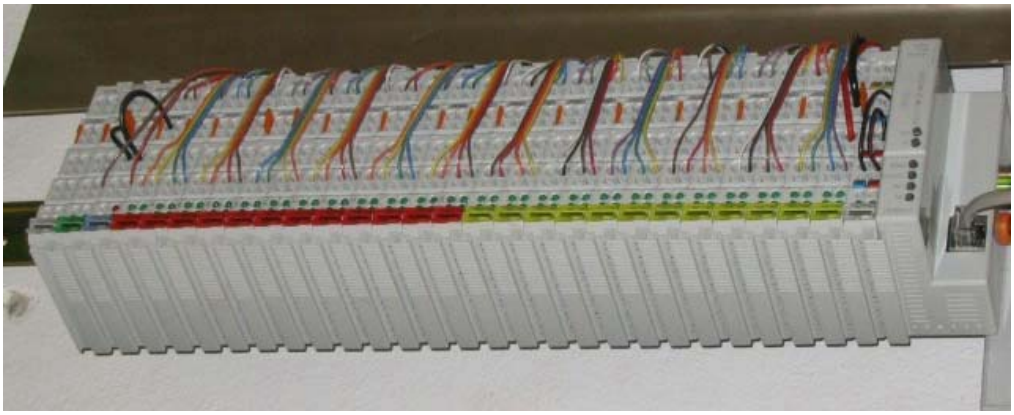


Obr. 3.23: Přechodová charakteristika uzavřené smyčky s PID regulátorem

Z přechodových charakteristik uzavřené smyčky je vidět, že nejrychlejší regulátor je PID. Jelikož má tento regulátor jen malý překmit, použijeme ho pro regulaci polohy prostředního výtahu. V reálném výtahu by samozřejmě překmit nemohl nastat a za cenu nižší rychlosti, bychom museli použít např. PD regulátor.

4 PLC WAGO

Za posledních několik málo let jsou technologie řízeny pomocí decentralizované automatizace. Mnoho těchto systémů se převážně používá jako propojení mezi decentralizovanými moduly a řídícím systémem. S WAGO I/O SYSTEM 750 (obr. 4.24), nejmenším a na protokolu sběrnice nezávislým modulárním systémem vstupů a výstupů, mohou být data posílána a přijímána sama o sobě decentralizovaně. Jednotlivé uzly systému obsahují 1 až 8 kanálů vstupů nebo výstupů. Binární I/O moduly jsou k dostání ve verzích s úrovní signálu od 5V do 250V, rozdílným výkonem a rychlou signalizací poruchy. Analogové I/O moduly podporují všechny unifikované signály s rozsahy 0(4)-20mA, 0-10V, +-10V. Jsou dostupné i pro odporová čidla teploty a různé druhy termočlánků. Speciální moduly jsou dostupné s rozhraním pro převodníky SSI, pro inkrementální čidla, seriové linky TTY, RS232 a RS485. Strana napájení sensoru a akčních členů je v průmyslové sběrnici realizována pomocí speciálních modulů s nebo bez pojistkové ochrany. Vysoká spolehlivost WAGO I/O SYSTEM 750 je postavena na základech rychlé, vibracím odolné a bezúdržbové spojovací technologie CAGE CLAMP. Podrobnější popis tohoto modulárního systému lze nalézt v [6] a [8].



Obr. 4.24: WAGO I/O SYSTEM 750

V následující kapitole budou stručně popsány jednotlivé členy WAGO I/O SYSTEM 750 použité na modelu výtahu.

4.1 Použité moduly

750-842 V anglické literatuře je označován jako Programmable Fieldbus Controller (PFC), což v překladu znamená Programovatelný sběrnicový automat. Tento člen v sobě zahrnuje TCP/IP protokol, podle něhož je možné s okolím komunikovat přes síťový kabel s koncovkou RJ-45. Rychlost komunikace je 10Mbit/s. Dále se tento člen stará o „sběr“ dat z podřízených I/O členů.

750-402 4-kanálový digitální vstupní člen určený pro příjem binárního signálu z ovládacího panelu a optických čidel. Je schopen přijímat log. signály o úrovni 15-30V.

750-504 4-kanálový digitální výstupní člen určený pro ovládání signalizačních prvků ovládacího panelu a pro určení směru pohybu výtahu. Výstupní úroveň log. signálu je +24V.

750-467 Analogový vstupní člen určený pro příjem signálu z ultrazvukového čidla. Je schopen přijímat signál o napětí 0-10V, který je pak převeden na digitální signál s rozlišením 12 bitů.

750-559 Analogový výstupní člen ovládající rychlost otáčení větráčku prostředního výtahu. Je schopen dávat napětí 0-10V (popř. ±10V).

750-600 Zakončovací člen I/O pole. Zabezpečuje správný chod dat v I/O poli.

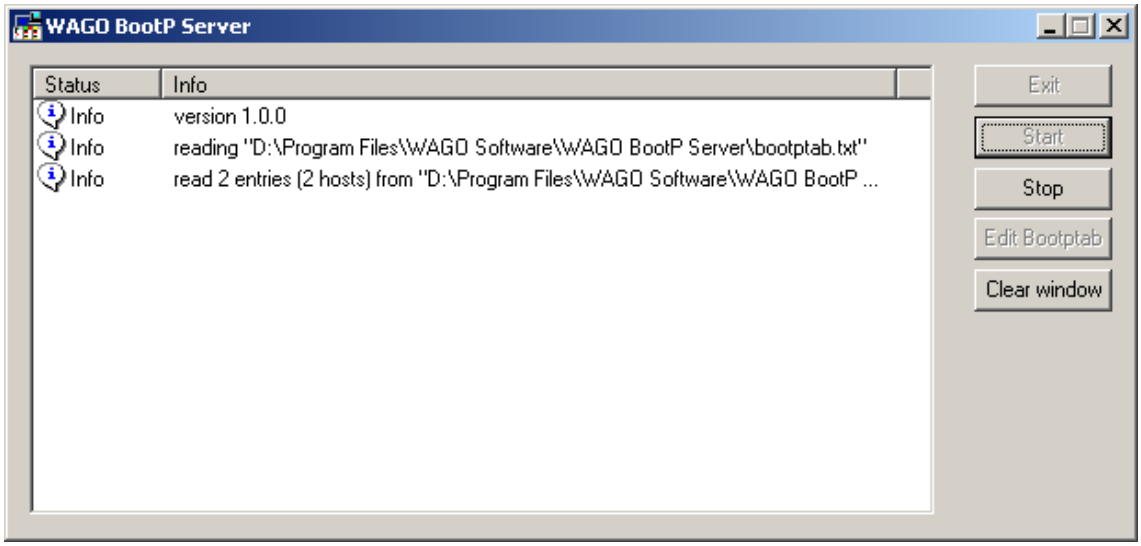
787-903 Stejnosměrný napájecí zdroj pro PLC, DC 24V, 5A.

4.2 Software

Aby se model choval jako skutečný výtah, je třeba do řídícího PLC nahrát vhodný program. Tato kapitola seznamuje čtenáře s prostředky pro prvotní detekci PLC v síti, aplikací pro samotný návrh ovládacího algoritmu a seznamuje s možností následné vizualizace.

4.2.1 BootP Server a síťové nastavení

K prvotní detekci PLC v síti je nutné použít program BootP Server, zde provést nastavení sítě v souboru bootptab.txt a poté spustit server. Na obrázku 4.25 můžeme vidět obrazovku aplikace BootP Server po úspěšném nadetekovaní PLC v síti.



Obr. 4.25: Program BootP Server

Nastavení PLC WAGO v síti

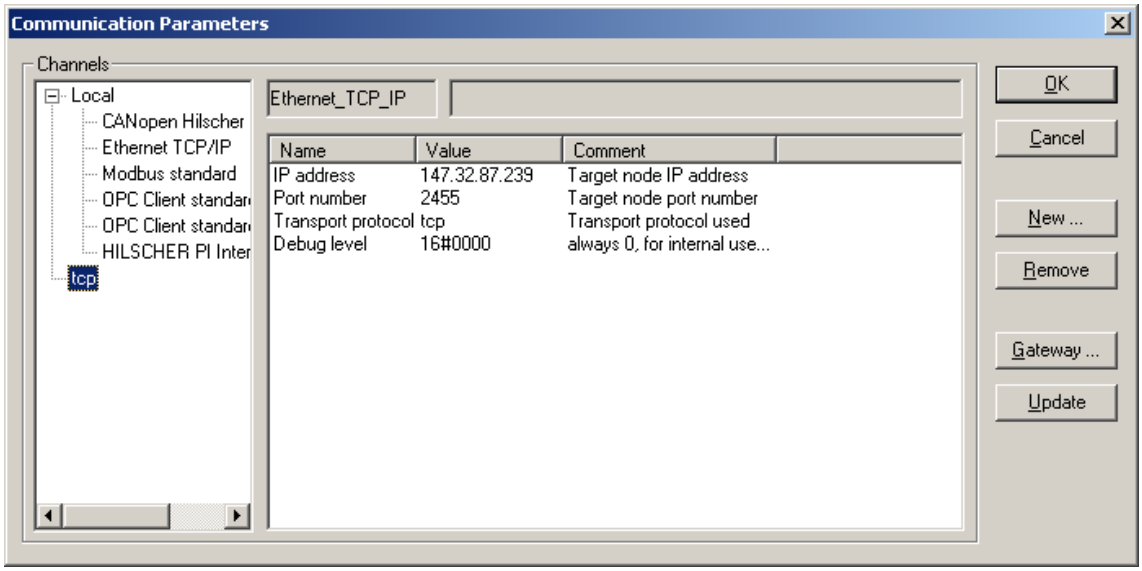
PLC WAGO je připojeno k PC přes síťový kabel s koncovkou RJ-45.

Nastavení PLC v síti je následující:

IP adresa:	147.32.87.239
def. GW:	147.32.87.129
maska sítě:	255.255.255.128
name server (DNS):	147.32.80.9

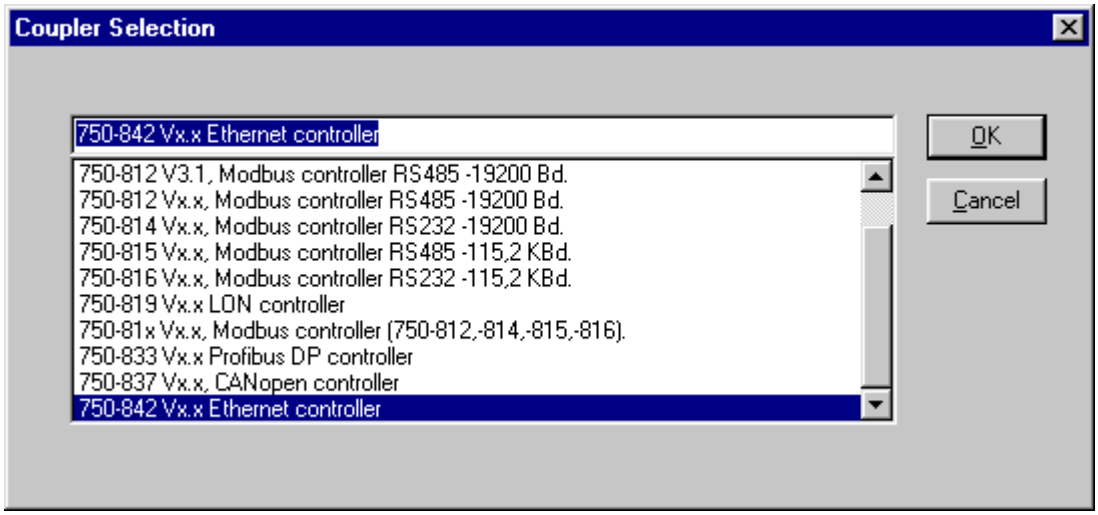
4.2.2 WAGO-I/O-PRO 32

Jak již bylo zmíněno, k programování PLC slouží aplikace zvaná WAGO-I/O-PRO 32. Tato aplikace běží na PC a proto je nejdříve nutné, nastavit síťovou komunikaci s řídícím PLC. Pro zdařilou komunikaci s reálným PLC je nutné příkazem menu "**Online**" » "**Communication Parameters**" » "**New**" vytvořit nový komunikační kanál s protokolem TCP/IP a zadat IP adresu přidělenou PLC, jak ukazuje obrázek 4.26.



Obr. 4.26: Nastavení IP adresy

V záložce "**Resources**" (vpravo dole) je pak nutné poklepat na "**PLC Configuration**" a vybrat zařízení dle obrázku 4.27.

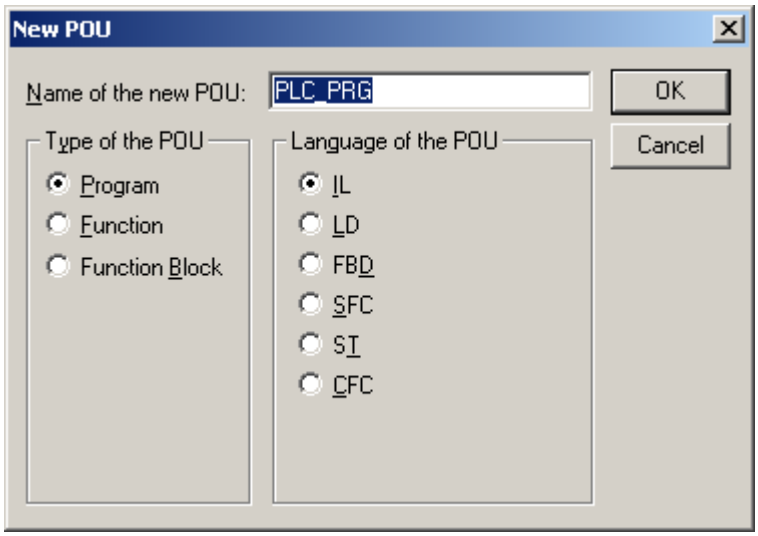


Obr. 4.27: Výběr typu PLC

Program se nyní může vzdáleně nahrávat pomocí protokolu TCP/IP po síti Internet do PLC WAGO Ethernet TCP/IP Programmable Fieldbus Controller 750-842 umístěném v laboratoři K09.

Založení nového projektu

Založení samotného projektu pak provedeme následovně: V menu **"File"** zvolíme položku **"New"**. Objeví se nám obrazovka nabízející jméno, typ a jazyk POU (Program Organization Unit). Název předvoleného POU je PLC_PRG a není vhodné ho měnit, protože PLC defaultně spouští POU právě s tímto jménem.



Obr. 4.28: Vytvoření nového projektu

Programovací jazyky WAGO-I/O-PRO 32

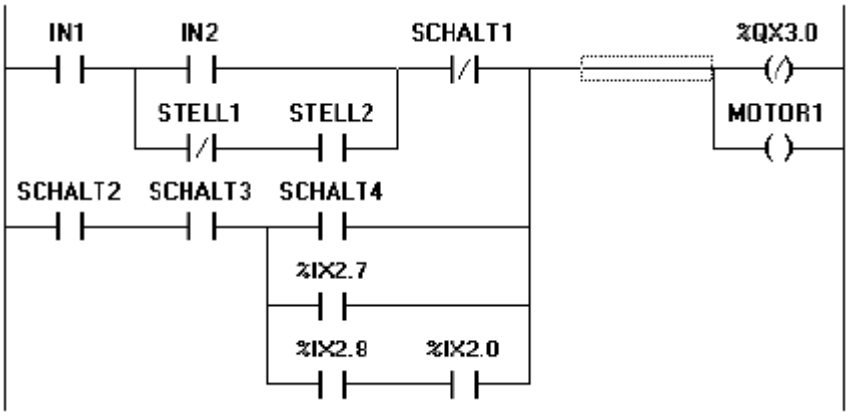
V následujícím přehledu rozebereme jednotlivé programovací jazyky POU:

IL (Instruction List) Instrukční list se skládá ze série instrukcí, přičemž každá instrukce začíná na novém řádku a obsahuje operátor a v závislosti na typu příkazu jeden nebo více operandů. Tento jazyk je nejvíce podobný programování v Assembleru (obr. 4.29).

```
LD   17
ST   lint    (* comment *)
GE   5
JMPC       next
EQ   istruct.sdword
STN test
next:
```

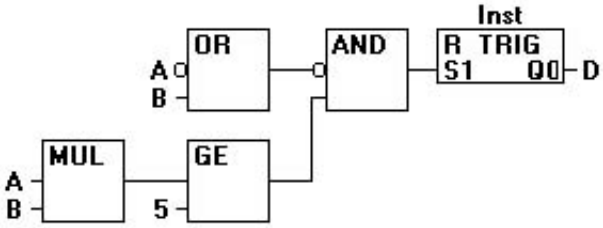
Obr. 4.29: Programování v IL

LD (Ladder Diagram) Jedná se o graficky orientované programování v tzv. žebříčkovém schéma a každý blok představuje strukturu kontaktů v elektrickém obvodu. Program se tak skládá ze série těchto bloků. Velkou výhodou tohoto programování je jednoduchost a názornost. Ve větších projektech naopak dochází k velké nepřehlednosti, vlivem velké potřeby místa na obrazovce pro jeden blok programu (obr. 4.30).



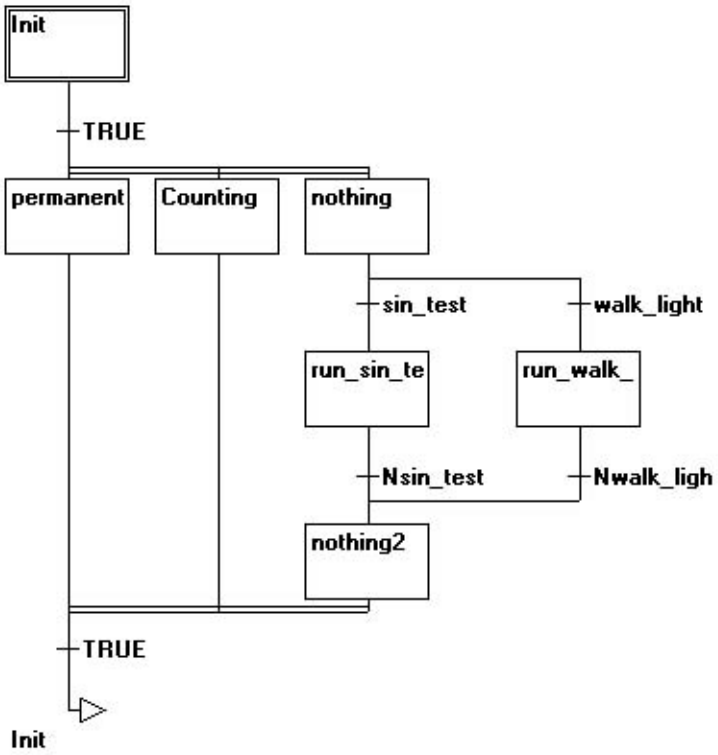
Obr. 4.30: Blok programu v LD

FBD (Function Block Diagram) je graficky orientovaný programovací jazyk. Pracuje s listem tzv. sítí, přičemž každá síť obsahuje strukturu představující logickou nebo aritmetickou operaci, volání funkčního bloku, skok nebo návratovou instrukci. Příklad tohoto programování je vidět na obrázku 4.31.



Obr. 4.31: Blok programu v FBD

SFC (Sequential Function Chart) je graficky orientovaný programovací jazyk, jenž umožňuje popsat chronologické pořadí jednotlivých akcí uvnitř programu (obr. 4.32).



Obr. 4.32: Blok programu v SFC

ST (Structured Text) se skládá ze série instrukcí jako ve vyšších programovacích jazycích (např. IF..THEN..ELSE). Tento jazyk se nejvíce podobá programovacímu jazyku Pascal. Příklad je uveden na obrázku 4.33.

```
IF value < 7 THEN
    WHILE value < 8 DO
        value:=value+1;
    END_WHILE;
END_IF;
```

Obr. 4.33: Programování v ST

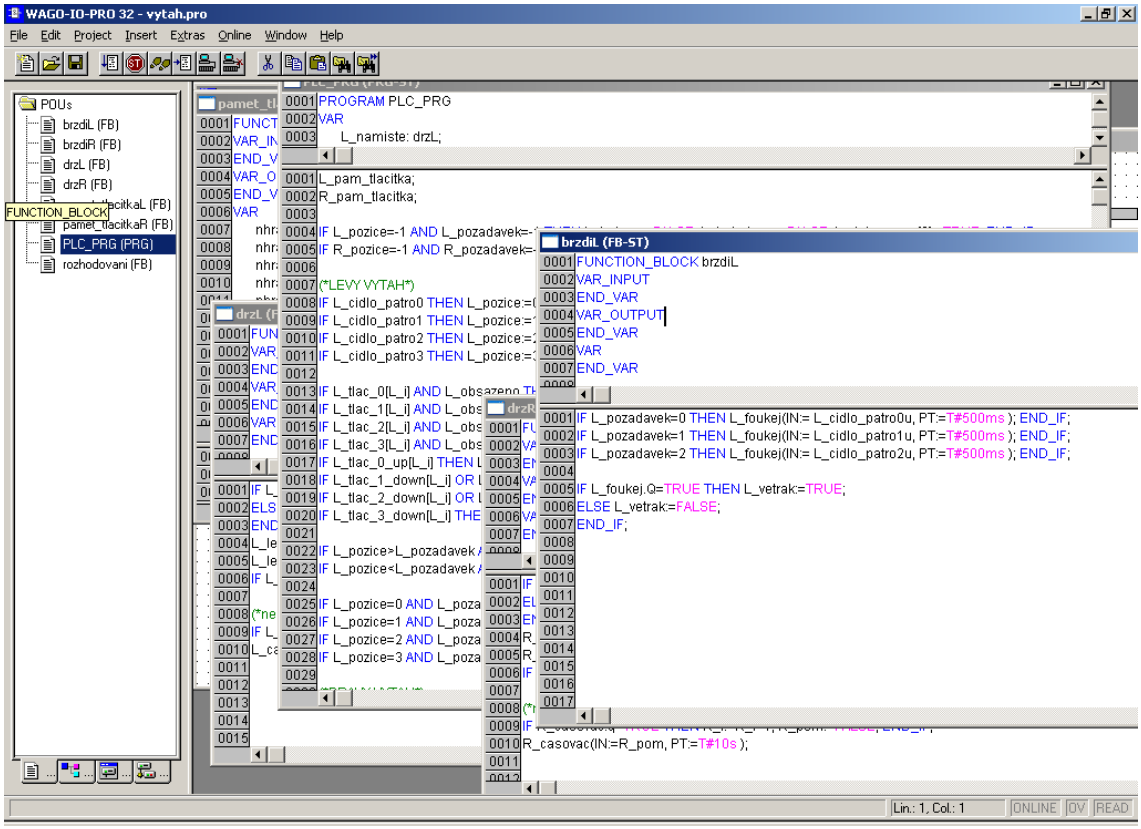
Po zvolení vhodného programovacího jazyka je tedy možno začít s programováním řídícího algoritmu.

5 Realizace dálkového řízení

V této kapitole bude popsán algoritmus pro řízení modelu výtahu, vizualizace celého modelu a popis použité webové kamery pro on-line sledování technologického procesu.

5.1 Algoritmus pro řízení modelu výtahu

Po založení projektu, popsaném v kapitole 4.2.2, již můžeme vyvíjet vlastní algoritmus řízení. Jako programovací jazyk jsem zvolil jazyk ST a to kvůli jeho přehlednosti a dobré strukturovatelnosti. Celý funkční řídící algoritmus lze nalézt na přiloženém CD a výpis části kódu programu v Příloze B. Na obrázku 5.34 je vidět vývojové prostředí aplikace WAGO-I/O-PRO 32.



Obr. 5.34: Vývojové prostředí WAGO-I/O-PRO 32

5.1.1 Koncepce programu

Algoritmus pro řízení výtahu vychází z koncepce naznačené na obrázku 5.35. Po stisku jakéhokoliv ovládacího prvku se do paměti zapíše, v jakém pořadí byl ovládací prvek stisknut. Je to realizováno tak, že každý prvek má svou paměť, která se po stisku jakéhokoliv tlačítka globálně inkrementuje. Je-li v paměti více stisků několika prvků, potřebujeme požadavky na pohyb výtahu logicky setřídit, aby výtah jezdil ve správném sledu. Logikou rozhodování se tedy myslí proces, který zajistí logické přenastavení paměťových míst a to nejdříve globálně, v rámci všech tří výtahů a poté lokálně, pro jednotlivé výtahy. Následuje samotný pohyb kabiny výtahu a potvrzení o dokončení pohybu.

Obr. 5.35: Celkový koncept programu

5.1.2 Popis programu a jednotlivých funkčních bloků

Program se skládá z hlavní části PLC_PRG a z několika funkčních bloků, jejichž funkce si nyní rozebereme podrobněji.

PLC_PRG: Hlavní program má za úkol provést již logicky setříděný pohyb „digitálních“ výtahů a „analogového“ výtahu. K tomu využívá několika funkčních bloků popsaných dále. Rozhodování, zda „digitální“ výtah pojede dolu nebo nahoru, se děje porovnáním aktuální pozice s požadovanou. Po dojetí výtahu na cílové patro se vygeneruje potvrzení o dojetí, spustí se časovač, který zajistí stání v patře po určitou dobu a následně se obslouží další požadavek v paměti.

***brzdiL, brzdiR*:** U „digitálních“ výtahu může nastat problém s „přejížděním“ přes patro. Je to způsobeno tím, že výtah při sestupování z vyšších pater má velkou setrvačnou energii a i při sepnutí optického čidla a následném foukání větráku pro směr nahoru, se nestačí výtah udržet na čidle a přejede. To lze řešit přidáním dalších čidel těsně nad patro a následném brzděním výtahu. Tyto dva funkční bloky mají tedy za úkol foukat „nahoru“ po dobu 500ms v případě, že výtah jede z vyššího patra, sepnou se optická „brzdící“ čidla a výtah bude v daném patře zastavovat. Brždění při jízdě směrem nahoru není třeba, protože nám pomáhá gravitace. Funkční blok *brzdiL* je určen pro levý výtah a obdobně funkční blok *brzdiR* je určen pro pravý výtah.

***drzL, drzR*:** Tyto funkční bloky mají za úkol udržet výtah v patře a to u obou „digitálních“ výtahů. Je-li čidlo v patře sepnuté, větráček fouká „nahoru“, v opačném případě fouká „dolu“. Ve výsledku tedy kabina výtahu kmitá těsně nad čidlem, ale kvůli povaze digitálního řízení nelze „stání“ výtahu realizovat jiným způsobem. Dalším úkolem tohoto bloku je zpracování dalšího požadavku po splnění aktuálního. Dojede-li tedy výtah na určenou pozici, počká zde 10 vteřin (čas pro výstup a nástup, otevření a zavření dveří, apod.), a načte z paměti další požadavek na výtah. Je-li další paměťové místo prázdné (není žádný další požadavek), zůstává výtah na aktuální pozici.

***pamet_tlacitkaL, pamet_tlacitkaR, pamet_tlacitkaM*:** Stisknutá tlačítka je třeba ukládat do paměti, aby mohl výtah jezdit podle předvolby. Musíme tedy testovat náběžnou hranu stisku tlačítka, zapsat do paměti a poté paměť inkrementovat. Po stisku tlačítka se rozsvítí i příslušná signalizační LED dioda.

***pohybM*:** U „analogového“ výtahu se pro pohyb používá PID regulátor. Hodnota z ultrazvukového čidla je vstupní aktuální hodnotou do regulátoru. Požadovaná hodnota je předem známá pozice patra (viz funkční blok *pozice*), která se mění podle toho, jaké patro je pro výtah cílové. Výstupem PID regulátoru je tedy odchylka požadované hodnoty od aktuální ovlivněná nastavením časových konstant regulátoru. Výstupní hodnotou z analogového modulu PLC pro otáčky větráku je tedy předem zjištěná hodnota pro „stání“ výtahu, ke které se přičítá výstup z regulátoru. Stání se v tomto případě rozumí taková hodnota napětí, při kterém váleček v trubici nevykazuje žádný pohyb a vznáší se namístě. Tabulka 5.7 ukazuje nastavení softwarového PID regulátoru v řídícím algoritmu.

Položka regulátoru	Hodnota nastavení	Vysvětlivky
ACTUAL	analog_in	aktuální hodnota z ultrazvukového čidla
DESIRED	patro	požadovaná hodnota ultrazvukového čidla patro = patro0 or patro1 or patro2 or patro3, v závislosti na požadovaném koncovém patře
KP	0.55	Proporcionální složka regulátoru
TR	76800 ms	Integrační časová konstanta regulátoru
TD	400	Derivační časová konstanta regulátoru
RESET	false	Resetování regulátoru

Tab. 5.7: Nastavení PID regulátoru

***pozice*:** Tento funkční blok určuje pozici jednotlivých pater. Je to hodnota ultrazvukového čidla, při sepnutí optické závory v patře.

Hodnoty pro jednotlivá patra jsou následující:

```
patro0:= 4500;
patro1:= 9800;
patro2:= 15300;
patro3:= 19900;
```

***rozhodovani*:** Rozhodování se dělí na dva dílčí úkoly. Prvním je globální rozhodování v rámci všech tří výtahů a druhé je lokální logické srovnání sledu požadavků v rámci jednoho výtahu.

Globální rozhodování: Toto rozhodování se uplatňuje pouze při přivolávání výtahů, kdy do příslušného patra přijede výtah jenž je nejblíže. Výběr výtahu provádíme porovnáním aktuálních pozicí a aktuálních směrů jízdy všech tří výtahů s požadovaným poschodím, kam se má výtah přivolat. Priority pro výběr výtahu uvádí tabulka 5.8. Je-li požadavek přivolání výtahu např. v prvním patře s úmyslem jízdy dolu, zjišťuje se, který výtah splňuje požadavek, uvedený v 3. a 4. sloupci tabulky. Postupuje se od nejvyšší priority k nejnižší. Pokud tedy v našem případě nebude žádný výtah již v prvním patře a stát, zjistí se je-li některý z výtahu v patře druhém a nejede zrovna směrem dolů. Pokud ani tento požadavek žádný z výtahu nesplňuje, pokračuje se v porovnávání s dalším řádkem tabulky, dokud jeden z výtahů toto kritérium nebude splňovat. Poté se do požadavků pro vybraný výtah

zařadí požadavek pro zastavení v prvním patře s úmyslem jízdy dolů. Pokud není ani jeden z výtahů v „tabulkové“ pozici, přivolá se výtah, u něhož byla tlačítko pro přivolání stisknuto.

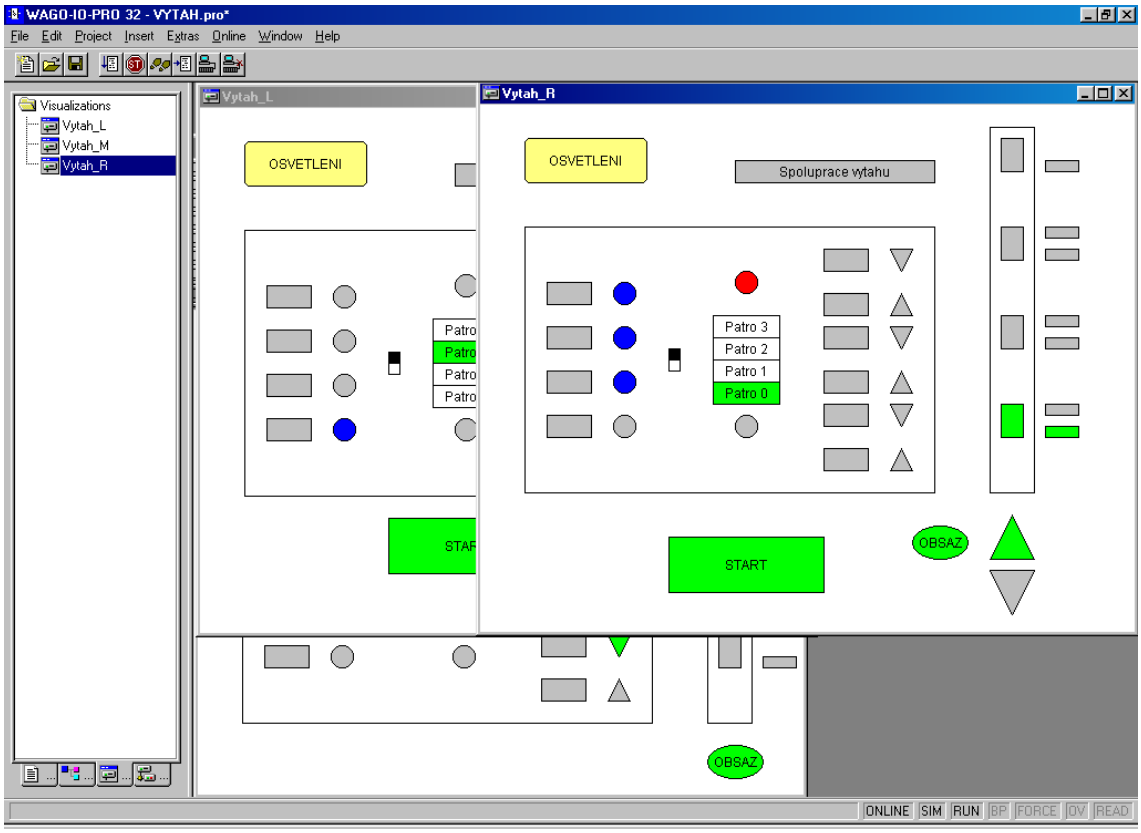
Stisknuté tlačítko pro přivolání		Priority pro výběr výtahu	
poschodí	úmysl jízdy	aktuální pozice výtahu (poschodí)	směr jízdy výtahu
0	nahoru	0	stojí
		1	dolů
		1	stojí
		2	dolů
		2	stojí
		3	dolů
		3	stojí
1	dolů	1	stojí
		2	dolů
		2	stojí
		3	dolů
		3	stojí
1	nahoru	1	stojí
		0	nahoru
		0	stojí
2	dolů	2	stojí
		3	dolů
		3	stojí
2	nahoru	2	stojí
		1	nahoru
		1	stojí
		0	nahoru
		0	stojí
3	dolů	3	stojí
		2	nahoru
		2	stojí
		1	nahoru
		1	stojí
		0	nahoru
		0	stojí

Tab. 5.8: Priority pro globální rozhodování

Lokální rozhodování: Tímto rozhodováním se myslí logické přerovnání v paměti nastřádaných požadavků. Je-li např. aktuální pozice výtahu v nultém patře a my v rychlém sledu zmáčkneme např. tlačítka uvnitř výtahu a to v pořadí 3, 2, 1 výtah si „přerovná“ tyto požadavky v paměti a výsledným pohybem bude jízda do třetího patra se zastávkami v prvním a druhém patře. Pro srovnání požadavků byla použita metoda „Bubble sort“ [13], která prochází pole požadavků, v našem případě pole s čísly od 0 do 3, a řadí je podle velikosti. Rozhodování, zda se bude řadit vzestupně nebo sestupně je dáno aktuální pozicí výtahu. U přivolávání je tento problém opačný a uplatňuje se zde preference okrajových pater. Pokud tedy výtah stojí v přízemí a někdo přivolá výtah např. v druhém patře s úmyslem jízdy dolů a současně ve třetím patře, výtah pojede nejdříve do patra třetího. V případě, že v druhém patře bylo zmáčknuto tlačítko pro přivolání s úmyslem jízdy nahoru, výtah samozřejmě nejdříve zastaví v tomto patře. Problematika rozhodování v demonstračním algoritmu je procesem, který vyžaduje hlubší rozbor.

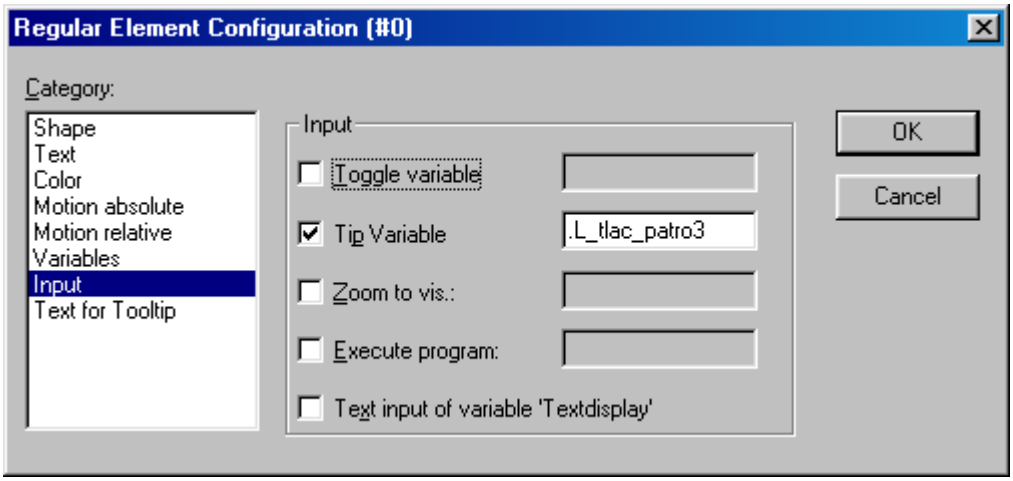
5.2 Vizualizace

Vizualizace modelu se provádí ve stejné aplikaci jako programování, tedy WAGO-I/O-PRO 32. V záložce **"Vizualizations"** lze založit objekt pomocí pravého kliknutí na tuto záložku a zvolením položky **"Add object…"**. Následně si s pomocí jednoduchého grafického editoru můžeme vytvořit libovolný element, např. jednoduchá tlačítka, LED prvky, čidla, atd. (obr. 5.36).



Obr. 5.36: Vizualizace v prostředí WAGO-I/O-PRO 32

Provázání s proměnnými modelu se děje dvojitým poklepáním na již vytvořený element (např. obdelník, trojúhelník,...), kdy se nám otevře dialogové okno pro editaci vlastností tohoto elementu (obr. 5.37). Zde se především nastavuje textový popis elementu, barva struktury (vnitřní a alarmová), umístění na pracovní ploše, provázání s nadefinovanou proměnnou a změna vizuálního atributu struktury při změně této proměnné (neviditelnost, změna barvy a zobrazení textu). Abychom mohli pomocí vizualizační obrazovky ovládat i reálný model, musíme námi nadefinovaná tlačítka „propojit“ s proměnnými programu. To lze jednoduše udělat nastavením položky v kategorii **"Input"** v atributech elementu a provázáním s nadefinovanou proměnnou. Ve výsledku tedy můžeme simulovat sepnutí tlačítka pouhým kliknutím myši na strukturu představující toto tlačítko.



Obr. 5.37: Dialogové okno pro editaci vlastností struktury

Vizualizace pro model výtahu je vidět na obr. 5.36. Kvůli nedostatku místa na jedné obrazovce bylo nutné vizualizovat každý výtah zvlášť a mít tak tři okna pro celý model. V každém okně se nachází vizualizace ovládacího panelu, jenž je podobný ovládacímu panelu popsanému v kapitole 2.2 a dále znázornění výtahové šachty modulu s rozmístěním optických čidel, znázornění aktuální pozice kabiny výtahu, stavu větráčku a obsazenosti kabiny.

Pokud tedy není námi vytvořená aplikace v simulačním módu, můžeme pozorovat, ale i ovládat vstupy a výstupy modelu přes vizualizační obrazovku.

5.3 Webová kamera AXIS 2100

Jak již bylo zmíněno, k distančnímu vzdělávání je zapotřebí on-line sledování výsledků navrženého algoritmu. Model výtahu proto sledujeme síťovou kamerou AXIS 2100 (obr. 5.38). Kamera má vestavěný webserver, a proto již nepotřebuje žádný nadřazený počítač pro zpracování a distribuci snímků v síti. Přístup na kameru je možný z libovolného webového prohlížeče, kdy se po zadání IP adresy kamery objeví výrobcem vytvořené webové stránky. Tyto stránky lze modifikovat podle vlastní potřeby.

Obr. 5.38: Webová kamera AXIS 2100

5.3.1 Připojení k síti Ethernet

Kameru lze připojit k síti buď pomocí modemu nebo pomocí síťového kabelu s koncovkou RJ-45. V našem případě použijeme připojení pomocí kabelu. IP adresu lze nastavit dvěma způsoby:

a) Nastavení IP adresy příkazem *arp*

Nastavení příkazem *arp* se provádí z příkazového řádku:

arp -s <IP adresa kamery> <MAC adresa kamery> <IP adresa přistupujícího počítače>

tedy např.: arp -s 147.32.87.221 00-40-8c-10-00-86 147.32.87.176

b) Nastavení kamery pomocí aplikace AXIS IP Installer

AXIS IP installer je aplikace dodávaná s kamerou, běžící pod systémy Windows 95/98/NT/2000/XP. Lze ji také získat na webových stránkách výrobce [9]. Po instalaci a spuštění aplikace se na obrazovce objeví dialogové okno, v němž by se po zapnutí kamery mělo objevit její sériové číslo, které je shodné s MAC adresou. Poté stačí nastavit požadovanou IP adresu a zkontrolovat správnost nastavení pohledem na webové stránky kamery.

Pro správné načtení webových stránek kamery je ovšem nejdříve zapotřebí, zakázat v prohlížeči používání proxy serveru pro IP adresu kamery. Na samotné stránky pak přistupujeme zadáním URL http://147.32.87.218 popř. http://147.32.87.218/admin, pokud chceme editovat síťová nastavení kamery. Po nastavení adresy routeru, je možné zrušit zákaz používání proxy serveru pro IP adresu kamery.

Parametry kamery

Mezi nejdůležitější parametry patří frekvence posílání snímků. Ta je závislá na následujících veličinách.

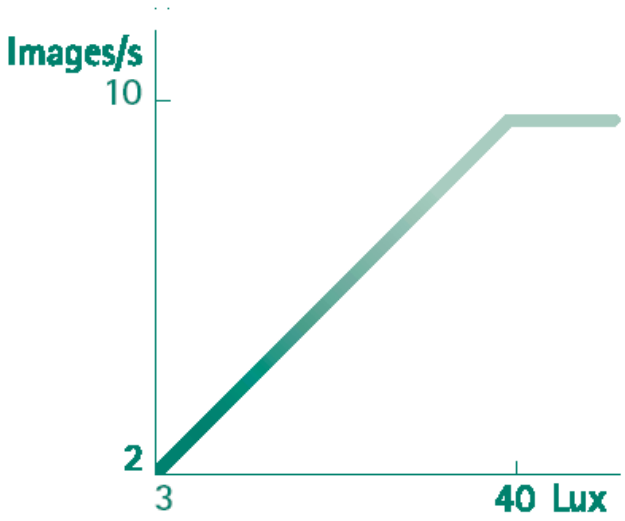
- rychlosti počítače, z kterého na kameru přistupuji,
- rozlišení posílaných snímků,
- rychlosti síťového připojení,
- světelných podmínkách v místě snímání snímků.

V tabulce 5.9, lze vidět datový objem posílaných snímků v závislosti na rozlišení a kompresi.

Resolution	Compression level			
	Lowest	Low	Medium	High
640x480	250 Kb	20 Kb	13 Kb	8 Kb
320x240	70 Kb	8 Kb	5 Kb	3 Kb

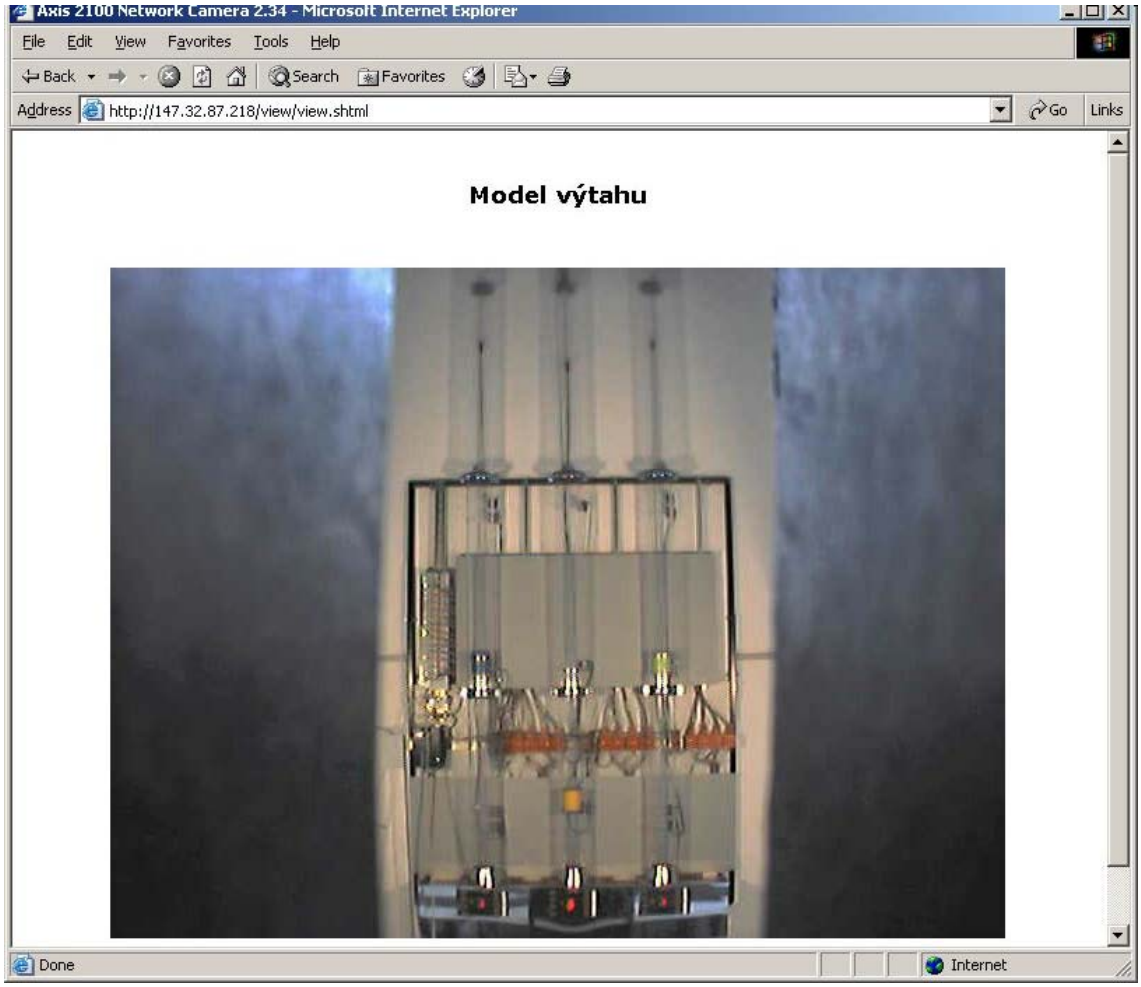
Tab. 5.9: Datové objemy snímků

Jak již bylo zmíněno, frekvence posílaných snímků závisí také na osvětlenosti snímaného objektu. Obrázek 5.39 nám ukazuje, že tato závislost je lineární a při dobrém osvětlení je kamera schopna posílat až 10 snímků za sekundu.



Obr. 5.39: Závislost frekvence snímků na osvětlení

Osvětlení modelu je zajištěno dvěma osvětlovacími prvky firmy RENDL, typ VANA 150W. Ty jsou spínány přes relé RELPOL RM 94P-24 ~250V, 8A. Bit pro spínání světel QX5.15 je ovládán z vizualizační obrazovky (obr. 5.36). Pohled na model výtahu webkamerou AXIS je vidět na obr. 5.40



Obr. 5.40: Pohled na model výtahu webkamerou AXIS

Úplné síťové nastavení kamery je následující:

IP adresa:	147.32.87.218
def. GW:	147.32.87.129
maska sítě:	255.255.255.128

6 Závěr

Vlastní diplomovou práci lze rozdělit na hardwarovou a softwarovou část. V hardwarové části byl hlavním úkolem návrh a tvorba plošných spojů pro řízení otáček větráčku, propojení čidel, tlačítek a signalizačních prvků s PLC, realizace sepnutí světel a chodu větráčků přes PLC a realizace dálkového řízení propojením PLC se sítí Internet a připojením webkamery. Softwarová část zahrnovala především konfiguraci PLC a webkamery v síti, návrh demonstračního algoritmu pro řízení výtahů a vizualizace modelu. Vizualizace umožňuje simulovat stisk ovládacích prvků a sepnutí čidel, aniž bychom museli u modelu fyzicky být nebo naopak signalizovat chování I/O prvků fyzického modelu.

Po úspěšném zvládnutí vytyčených cílů je tedy nyní možné připojit se na PLC z jakéhokoliv místa poskytující internetové připojení, pomocí aplikace WAGO-I/O-PRO 32 nahrát vzdáleně vlastní algoritmus do PLC WAGO a online sledovat chování výtahů, umožněném webkamerou AXIS.

Při realizaci této diplomové práce vzniklo několik problémů, jenž bylo nutné vyřešit. Některé z nich byly složitější a měla by jim být věnována větší pozornost..

Pro ovládání otáček větráčku slouží tzv. přizpůsobovací plošné spoje, které jsou realizovány na univerzálních pájecích destičkách a to kvůli relativně časté potřebě změny tohoto obvodu v průběhu realizace. Pokud bychom chtěli tyto plošné spoje miniaturizovat, museli bychom navrhnout speciální plošné spoje podle schémat uvedených v kapitole 2.3

Konektory použité na ovládacích panelech jsou předimenzovány a namísto použitého 32-pinového konektoru by stačil konektor 26-pinový. Stávající konektor byl použit z důvodu špatného přístupu pro pájení.

Jako částečně nevhodné se ukázalo ultrazvukové čidlo, jenž má malý rozsah a váleček prostředního výtahu přestane snímat, je-li níže než v nultém patře. Tento problém lze ovšem řešit softwarově. Dalším vhodným řešením by bylo umístění zarážek v nultém patře nebo výměna ultrazvukového čidla za jiný typ.

Dalším problémem se ukázalo nevhodné umístění modelu výtahu blízko oken. Při snímání webkamerou proti oknu dochází k světelné nevyrovnanosti, kdy se kamera přizpůsobí ostrému dennímu světlu a snímaný model je tak velice tmavý a to i přes přídavné umělé osvětlení. To jsem vyřešil papírovými krytkami na webkameře, které omezují

výhled kamery do stran a zastiňují tak okna. Lepší řešením by bylo, umístnit model proti oknům, což ale v laboratoři K09 není možné.

Algoritmus pro řízení výtahu v tomto případě slouží pouze jako demonstrační a logika pro rozhodování popsaná v kapitole 5.1 by zasloužila hlubší rozbor.

Model se bude využívat pro výuku studentů v předmětech Řídící systémy a Návrh automatizovaných zařízení a studenti si budou moci vyzkoušet vzdálené řízení technologického procesu a použití reálného regulátoru, s kterým se seznámily např. v předmětu Systémy a řízení. K nekoliznímu přístupu na model se bude využívat systému *Lablink*, který umožňuje rezervování pracovní doby na úloze pro registrovaného studenta. S modelem tedy nebude moci pracovat (programovat PLC) nikdo jiný než ten, kdo má v příslušnou dobu rezervaci.

Literatura

[1] Vysoký, O. *Elektronické systémy II*. Vydavatelství ČVUT, Praha 1999

[2] John, J. *Systémy a řízení*. Vydavatelství ČVUT, Praha 1999

[3] Štecha, J *Optimální rozhodování a řízení*. Vydavatelství ČVUT, Praha 1999

[4] Havlena, V. *Odhadování a filtrace*. Přednášky k předmětu

[5] Manual *Programming Tool IEC 61131-3 WAGO-I/O-I/O PRO 32 version 2.0.0*

[6] Manual *WAGO-I/O-System750, Ethernet TCP/IP 750-342, 750-842 version 2.0.0*

[7] Manual *AXIS Communications AXIS 2100 Network Camera User´s Guide v.1.1*

[8] *WAGO Kontakttechnik - WAGO INNOVATIVE CONNECTIONS*
http://www.wago.com

[9] *Axis Communications*
http://www.axis.com

[10] Nepustil, P., Karafiát, M *Protokol TCP/IP, semestrální práce k předmětu „Počítačová technika a komunikace“*. ČVUT FEL, Praha 2000

[11] Foit, J., Hudec, L. *Součástky moderní elektroniky*. Vydavatelství ČVUT, Praha 1998

[12] Šimek, T., Burget, P. *Elektronické systémy I-přednášky*. Vydavatelství ČVUT, Praha 2001

[13] Müller, K. *Výpočetní technika a programování I*. Vydavatelství ČVUT, Praha 1999

[14] *GES Katalog 2004*, úplný katalog sortimentu GES-ELECTRONICS, Vydání XII/2003

Příloha A

Obsah přiloženého CD

Přiložený CD ROM v jednotlivých složkách obsahuje:

DiplomovaPrace – Adresář obsahuje tuto diplomovou práci ve formátu *pdf* a *doc*.

Manualy – Dokumentace k používanému hardwaru a softwaru ve formátu *pdf*.

Obrazky – Adresář obsahující veškeré obrázky použité v této diplomové práci ve formátu *jpg*, *bmp* a *cdr,* dále pak elektronická schémata vytvořená v aplikaci Orcad Family Release 9.2 (formát *dsn*) a tabulky a grafy ve formátu *xls*.

Program – Tato složka obsahuje zdrojový text programu, který jsem napsal v aplikaci WAGO-I/O-PRO 32.

Software – Obsahuje freeware software, který byl použit při realizaci modelu výtahu.

Příloha B

Ukázka programu

PLC_PRG:

```
IF global THEN globalni_rozh; END_IF;

(*LEVY VYTAH*)
L_pam_tlacitka;
L_pozadavek[L_pamet+1]:=-1;

IF L_pozice=-1THEN L_led_up:=FALSE; L_led_down:=FALSE; L_dokonceno[0]:=TRUE; END_IF;

IF L_cidlo_patro0 THEN L_pozice:=0; L_display_low:=FALSE; L_display_high:=FALSE; L_pozice0:=TRUE;
L_pozice1:=FALSE; L_pozice2:=FALSE; L_pozice3:=FALSE; END_IF;
IF L_cidlo_patro1 THEN L_pozice:=1; L_display_low:=TRUE; L_display_high:=FALSE; L_pozice0:=FALSE;
L_pozice1:=TRUE; L_pozice2:=FALSE; L_pozice3:=FALSE; END_IF;
IF L_cidlo_patro2 THEN L_pozice:=2; L_display_low:=FALSE; L_display_high:=TRUE; L_pozice0:=FALSE;
L_pozice1:=FALSE; L_pozice2:=TRUE; L_pozice3:=FALSE; END_IF;
IF L_cidlo_patro3 THEN L_pozice:=3; L_display_low:=TRUE; L_display_high:=TRUE; L_pozice0:=FALSE;
L_pozice1:=FALSE; L_pozice2:=FALSE; L_pozice3:=TRUE; END_IF;

IF L_pozice>L_pozadavek[L_i] AND L_dokonceno[L_i-1] THEN L_led_down:=TRUE; L_smer:=1; L_zbrzdeni;
END_IF; (*L_vetrak:=FALSE je v brzdi_L*)
IF L_pozice<L_pozadavek[L_i] AND L_dokonceno[L_i-1] THEN L_vetrak:=TRUE; L_smer:=2; L_led_up:=TRUE;
END_IF;

IF L_pozice=0 AND L_pozadavek[L_i]=0 THEN L_namiste; L_led_patro0:=FALSE; L_led_patro0_up:=FALSE;
L_dokonceno[L_i]:=TRUE; END_IF;
IF L_pozice=1 AND L_pozadavek[L_i]=1 THEN L_namiste; L_led_patro1:=FALSE; L_led_patro1_up:=FALSE;
L_led_patro1_down:=FALSE; L_dokonceno[L_i]:=TRUE; END_IF;
IF L_pozice=2 AND L_pozadavek[L_i]=2 THEN L_namiste; L_led_patro2:=FALSE; L_led_patro2_up:=FALSE;
L_led_patro2_down:=FALSE; L_dokonceno[L_i]:=TRUE; END_IF;
IF L_pozice=3 AND L_pozadavek[L_i]=3 THEN L_namiste; L_led_patro3:=FALSE; L_led_patro3_down:=FALSE;
L_dokonceno[L_i]:=TRUE; END_IF;

(*PRAVY VYTAH*)
R_pam_tlacitka;
R_pozadavek[R_pamet+1]:=-1;

IF R_pozice=-1 THEN R_led_up:=FALSE; R_led_down:=FALSE; R_dokonceno[0]:=TRUE; END_IF;

IF R_cidlo_patro0 THEN R_pozice:=0; R_display_low:=FALSE; R_display_high:=FALSE; R_pozice0:=TRUE;
R_pozice1:=FALSE; R_pozice2:=FALSE; R_pozice3:=FALSE; END_IF;
IF R_cidlo_patro1 THEN R_pozice:=1; R_display_low:=TRUE; R_display_high:=FALSE; R_pozice0:=FALSE;
R_pozice1:=TRUE; R_pozice2:=FALSE; R_pozice3:=FALSE; END_IF;
```

```
IF R_cidlo_patro2 THEN R_pozice:=2; R_display_low:=FALSE; R_display_high:=TRUE; R_pozice0:=FALSE;
R_pozice1:=FALSE; R_pozice2:=TRUE; R_pozice3:=FALSE; END_IF;
IF R_cidlo_patro3 THEN R_pozice:=3; R_display_low:=TRUE; R_display_high:=TRUE; R_pozice0:=FALSE;
R_pozice1:=FALSE; R_pozice2:=FALSE; R_pozice3:=TRUE; END_IF;

IF R_pozice>R_pozadavek[R_i] AND R_dokonceno[R_i-1] THEN R_led_down:=TRUE; R_smer:=1; R_zbrzdeni;
END_IF;
IF R_pozice<R_pozadavek[R_i] AND R_dokonceno[R_i-1] THEN R_vetrak:=TRUE; R_smer:=2; R_led_up:=TRUE;
END_IF;

IF R_pozice=0 AND R_pozadavek[R_i]=0 THEN R_namiste; R_led_patro0:=FALSE; R_led_patro0_up:=FALSE;
R_dokonceno[R_i]:=TRUE; END_IF;
IF R_pozice=1 AND R_pozadavek[R_i]=1 THEN R_namiste; R_led_patro1:=FALSE; R_led_patro1_up:=FALSE;
R_led_patro1_down:=FALSE; R_dokonceno[R_i]:=TRUE; END_IF;
IF R_pozice=2 AND R_pozadavek[R_i]=2 THEN R_namiste; R_led_patro2:=FALSE; R_led_patro2_up:=FALSE;
R_led_patro2_down:=FALSE; R_dokonceno[R_i]:=TRUE; END_IF;
IF R_pozice=3 AND R_pozadavek[R_i]=3 THEN R_namiste; R_led_patro3:=FALSE; R_led_patro3_down:=FALSE;
R_dokonceno[R_i]:=TRUE; END_IF;

(*PROSTREDNI VYTAH*)
M_pam_tlacitka;
hodnota_patra;
IF M_pozice=-1 THEN M_led_up:=FALSE; M_led_down:=FALSE; M_dokonceno[0]:=TRUE; END_IF;

IF M_cidlo_patro0 THEN M_pozice:=0; M_display_low:=FALSE; M_display_high:=FALSE; M_pozice0:=TRUE;
M_pozice1:=FALSE; M_pozice2:=FALSE; M_pozice3:=FALSE; END_IF;
IF M_cidlo_patro1 THEN M_pozice:=1; M_display_low:=TRUE; M_display_high:=FALSE; M_pozice0:=FALSE;
M_pozice1:=TRUE; M_pozice2:=FALSE; M_pozice3:=FALSE; END_IF;
IF M_cidlo_patro2 THEN M_pozice:=2; M_display_low:=FALSE; M_display_high:=TRUE; M_pozice0:=FALSE;
M_pozice1:=FALSE; M_pozice2:=TRUE; M_pozice3:=FALSE; END_IF;
IF M_cidlo_patro3 THEN M_pozice:=3; M_display_low:=TRUE; M_display_high:=TRUE; M_pozice0:=FALSE;
M_pozice1:=FALSE; M_pozice2:=FALSE; M_pozice3:=TRUE; END_IF;

IF M_pozice>M_pozadavek[M_i] THEN M_led_down:=TRUE; END_IF;
IF M_pozice<M_pozadavek[M_i] THEN M_led_up:=TRUE; END_IF;
pohyb;

IF M_pozice=0 AND M_pozadavek[M_i]=0 THEN M_led_up:=FALSE; M_led_down:=FALSE; M_led_patro0:=FALSE;
M_led_patro0_up:=FALSE; M_dokonceno[M_i]:=TRUE; END_IF;
IF M_pozice=1 AND M_pozadavek[M_i]=1 THEN M_led_up:=FALSE; M_led_down:=FALSE; M_led_patro1:=FALSE;
M_led_patro1_up:=FALSE; M_led_patro1_down:=FALSE; M_dokonceno[M_i]:=TRUE; END_IF;
IF M_pozice=2 AND M_pozadavek[M_i]=2 THEN M_led_up:=FALSE; M_led_down:=FALSE; M_led_patro2:=FALSE;
M_led_patro2_up:=FALSE; M_led_patro2_down:=FALSE; M_dokonceno[M_i]:=TRUE; END_IF;
IF M_pozice=3 AND M_pozadavek[M_i]=3 THEN M_led_up:=FALSE; M_led_down:=FALSE; M_led_patro3:=FALSE;
M_led_patro3_down:=FALSE; M_dokonceno[M_i]:=TRUE; END_IF;

rozhodni;
```