



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ŘÍZENÍ PROCESU TŘÍDĚNÍ PROGRAMOVATELNÝM AUTOMATEM

TITLE

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE
AUTHOR

MAREK NOVÁK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ MARADA, Phd.

BRNO 2008

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2007/08

ZADÁNÍ DIPLOMOVÉ PRÁCE

student(ka): Novák Marek

který/která studuje v **magisterském studijním programu**

obor: **Inženýrská informatika a automatizace (3902T024)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Řízení procesu třídění programovatelným automatem

v anglickém jazyce:

Control of sorting by programmable logic controller

Stručná charakteristika problematiky úkolu:

Úkol řeší třídění zboží do kontejnerů tak, aby váhy zboží v kontejnerech měli předepsanou hmotnost. Řízení zahrnuje chod celého zařízení pro vážení a třídění.

Cíle diplomové práce:

1. Podrobně zpracovat optimalizaci třídění zboží do připravených kontejnerů. Výsledné hmotnosti zboží v kontejnerech se musí co nejlépe blížit požadovaným hodnotám.
2. Zpracovat programové vybavení pro třídění k danému automatu fy Phoenix Contact.
3. Navrhnout monitorování procesu vážení a třídění pro dané vyvíjené zařízení.
4. Zpracovat programové vybavení pro vizualizaci procesu.
5. Vývojové práce kooperovat s řešením navazující části pro manipulaci se zbožím.

Seznam odborné literatury:

- [1] Šmejkal, L., Martinásková, M., PLC a automatizace, Praha: BEN, 1999.
- [2] Firemní materiály o programovatelných automatech fy Phoenix Contact.

Vedoucí diplomové práce: Ing. Tomáš Marada, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2007/08.

V Brně, dne

4. 12. 2007



Šeda

doc. RNDr. Ing. Miloš Šeda, Ph.D.
Ředitel ústavu

V. z. - F. Hvozdová

doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

Abstrakt

Tato diplomová práce se zabývá návrhem a výrobou jedné ze dvou částí zařízení pro řízení procesu třídění. Toto zařízení ke své činnosti využívá dva programovatelné automaty (dále PLC), společnosti Phoenix Contact, které jsou vzájemně propojeny.

Část řešení, kterou podrobně popisuje tato práce se týká komunikace mezi těmito automaty, návrhem optimalizačních metod třídění zboží, které je významné pro činnost celého zařízení. Další částí práce je návrh a realizace vizualizace. PLC jsou naprogramovány v prostředí PC worx a Vizualizace je vytvořena pomocí programu Visu+ od firmy Phoenix Contact.

Výsledkem je zařízení, které po nastavení parametrů pracuje samostatně a je kompletně spravováno pomocí dotykové obrazovky HDMI.

Diplomová práce vznikla při řešení výzkumného záměru Inteligentní systémy v automatizaci podporovaného MŠMT ČR pod registračním číslem MSM 0021630529.

Abstract:

My diploma thesis deals with the design and production of the first of the two parts of a device for classifying goods. This device needs two programmable logic controllers (hereafter PLCs), produced by Phoenix Contact, to complete these activities. These controllers are interconnected.

One part of the solution, which is described in my thesis in detail, deals with the communication between these controllers, the design of optimization of methods of classifying goods, which is important for the operation of the whole device and visualization realization. The PLCs are programmed in PC worx software and the visualisation has been made using the Visu+ programme produced by Phoenix Contact.

The outcome of my thesis is a device, which, after setting up of the parameters, works independently and is completely controlled by a HDMI touch screen.

This thesis has been supported by the Czech Ministry of Education in the frame of MSM 0021630529 Research Intention Intelligent Systems in Automation.

KLÍČOVÁ SLOVA

PLC, programovatelný automat, dotekový displej, komunikace, optimalizace.

KEYWORDS

PLC, programable logic controller, touch panel, communication, optimization.

Poděkování

Úvodem této práce bych rád poděkoval především vedoucímu mé práce p. Ing. Tomáši Maradovi, Phd., nejen za jeho profesionální a věcnou pomoc při řešení všech závažných problémů, ale za velmi věcné připomínky a názory, díky kterým získala moje práce podobu jakou má teď.

Dále mé díky patří p. Ing. Zdeňkovi Němcovi, který se velmi rád podělil o své cenné zkušenosti z praxe a taktéž za jeho vynikající pedagogickou úroveň.

Mé poděkování též patří celému akademickému sboru ústavu automatizace, který mi umožnil využít kvalitní a skvěle vybavené laboratoře, ale také mě skvěle připravil na praxi která bude následovat po dokončení studia.

Na závěr můj nemalý dík patří mé rodině, bez jejíž finanční a psychické podpory by tato práce nemohla vzniknout.

Obsah:

1	ÚVOD.....	11
2	Úvod do terminologie PLC.....	13
2.1	Programovatelné automaty	13
2.2	Charakteristiky PLC	14
2.2.1	Technické řešení (HW) programovatelných automatů (PLC).....	14
2.2.2	Vstupní/výstupní podsystém PLC	16
2.2.3	Speciální moduly	18
2.2.4	Silová část PLC.....	18
2.2.5	Modul CPU	19
2.3	Programování PLC	20
2.3.1	Úvod do programového vybavení PLC	20
2.3.2	Operační systém PLC	20
2.3.3	Programovací jazyky	22
3	Základní navození problému o řešené úloze.....	25
3.1	Vznik úlohy.....	25
3.2	Popis komponent.....	25
3.3	Použité komponenty	27
3.3.1	ILC 150 ETH – řídící PLC.....	27
3.3.2	IBL IL PWM/2 – pulzní generátor	28
3.3.3	IBL IL 24/48 DOR 2/W-PAC.....	28
3.3.4	TP 12 T – Dotyková obrazovka.....	29
3.3.5	HS-475 HB – servo pohon.....	29
4	Struktura programu	31
4.1	Úvod do struktury programu.....	31
4.2	Síťová komunikace obecně.....	31
4.3	Síťová komunikace prakticky – popis funkčních bloků pro komunikaci	32
4.3.1	Funkční blok IP_CONNECT.....	32
4.3.2	Funkční blok IP_USEND	33
4.3.2	Funkční blok IP_URCV.....	34
4.3.4	IP_TIME	35
5	Podrobný popis režimů	37
5.1	Struktura programu a názvů proměnných.....	37
5.2	Jednotlivé režimy slovně	38
5.2.1	Vážení s optimalizací (režim 1).....	42
5.2.2	Vážení s tříděním dle hmotnosti (režim 2)	48
5.2.3	Vážení bez optimalizace (režim 3)	50
5.2.4	Ostatní části programu	51
6	Servo-pohon HS-475 HB.....	53
7	Tvorba vizualizace	57
7.1	Vizualizace ve zkratce	57
7.2	Princip dotekové technologie displeje TP 12 T	57
7.2	Způsob práce v prostředí VISU+	58
7.3	Struktura vizualizace.....	58
7.3.1	Základní obrazovka.....	58
7.3.2	Režim vážení bez optimalizace.....	59

7.3.3	Režim vážení s optimalizací	60
7.3.4	Režim vážení s tříděním dle hmotností.....	64
8.	Závěr	67
9.	Seznam použité literatury	69

1 ÚVOD

Tato práce se zabývá návrhem a výrobou zařízení pro řízení procesu třídění. Zařízení vzniklo za účelem reprezentace v soutěži Xplore. Celé zařízení slouží k vážení a třídění zboží, v našem případě citronů a skládá se ze dvou na sobě závislých částí. K vážení je použita speciální váha, vážící na principu kinematického momentu. Na váhu jsou z podavače umísťovány citrony které jsou váženy pomocí algoritmu závislého na rozběhu a doběhu elektro-motoru. Po zvážení jsou citrony roztříděny do kontejnerů pomocí algoritmu a do kontejnerů jsou vkládány pomocí servo-pohonů a ramene váhy. Rameno se natáčí nad kontejner, do kterého bude zboží vkládáno. Zařízení je ovládáno pomocí dotykové obrazovky.

Obě části úlohy využívají pro svou činnost PLC jednotky a každé části se věnuje vlastní diplomová práce. Obecně budou části úlohy dále děleny na část vážící a část třídící. Třídící PLC je zároveň řídící. V první části úlohy, které se věnuje tato práce, se zpracovávají třídící algoritmy, vizualizace, činnost servo-pohonů a síťová komunikace. Druhá část úlohy se rovněž zabývá síťovou komunikací. Problematika speciálního vážení je vysvětlena v jiné diplomové práci. Úspěšnost úlohy, mimo jiné, závisí na síťové komunikaci mezi oběma PLC.

Zařízení pro třídění pracuje se třemi režimy (z toho 2 pro třídění). Hlavním režimem je optimalizace třídění zboží do připravených kontejnerů, přičemž zboží v kontejnerech se musí co nejlépe blížit nastaveným parametrům. Kontejnery jsou celkem čtyři. Tři jsou vyhrazeny pro zboží splňující zadané požadavky a jeden je vyhrazen pro objekty spadající mimo vymezenou hranici jakosti. Veškeré tolerance jsou nastavitelné.

Zařízení spolupracuje s navazujícím zařízením, které váží zboží a zároveň s ním manipuluje. Obě zařízení tvoří jeden celek a jsou řízena programovatelnými automaty firmy Phoenix Contact (dále PLC), které spolu navzájem musí komunikovat. PLC na kterém je zpracováno programové vybavení, je rovněž řídící. S funkcí řídícího PLC je spjata již zmíněná vizualizace. PLC obstarávající vážení je řízené a reaguje až na podněty řídícího PLC.

Algoritmus optimalizace funguje jako třídící algoritmus. Vhodná volba algoritmu optimalizace je důležitá pro bezproblémový chod celého zařízení a v programu PLC je řešena jako nezávislý funkční blok. Tohoto řešení lze velmi efektivně využít při zpracování více pracovních režimů zařízení. Další režim potom realizujeme tak, že využijeme stávající části programu a vložíme nový funkční blok, který již disponuje funkcí specifickou pro daný režim. Pracovní režimy jsou celkem tři, z nichž v jednom se provádí optimalizace a zbylé dva jsou doplňkové.

První režim je z hlediska řešení nejvýznamnější. Jeho hlavní částí je funkční blok optimalizace. Jedná se právě o řešení s optimalizací a od jeho úspěšnosti se odvíjí funkčnost celého zařízení. Jak již bylo nastíněno, srdcem režimu je funkční blok optimalizace. Ten pracuje tak že při porovnání hmotnosti váženého zboží a velikostí všech aktuálních hmotností obsahů kontejnerů vybere jeden, do kterého se zboží přidá.

Druhý režim slouží pouze k vážení zboží umístěného na váze. Ve své podstatě slouží pouze k jakési praktické demonstraci spolupráce mezi oběma PLC.

Třetí režim umožňuje obsluhu nastavit kontejnerům tolerance a zařízení potom umísťuje zboží splňující kritéria do jednoho ze tří kontejnerů. Čtvrtý kontejner je vyhrazen pro zboží, které nesplňuje kritéria nastavená obsluhou pro první tři kontejnery.

Zařízení pracující v tomto režimu prakticky vybírá zboží stejných hmotností a seskupuje je do tří skupin, kdy každá skupina splňuje parametry zadané obsluhou.

Další z nezbytných kroků řešení se zabývá řízením servo-pohonů sloužících k manipulaci se zbožím. Řešení je realizováno přídatným modulem, který má stavebnicový charakter a generuje nastavitelné kmitočty.

Posledním řešením kterým se práce zabývá je Vizualizace. Jedná se o řídicí program s jehož pomocí lze celé zařízení ovládat. Program musí rovněž spolupracovat s řídicím PLC, což je realizováno použitím OPC serveru a vyhrazením společných proměnných s PLC. Program je vytvořen pomocí softwaru VISU+ společnosti Phoenix Contact. Styk s obsluhou je realizován pomocí dotekového displeje, který zastává roli vstupně-výstupního zařízení. Součástí dotekového displeje je počítač na kterém je tento program spuštěn.

2 Úvod do terminologie PLC

Tato kapitola vychází z [1]

Programovatelné automaty (Programmable Logic Controllers PLC) se staly nejvýznamnějším řídicím prostředkem pro řízení technologických procesů, výrobních linek a strojů již během pivní poloviny 80. let. Byly odezvou na vývoj mikroelektronické technologie, který umožnil vytlačit centralizovaného řízení, reprezentované řídicími počítači a minipočítači distribuovanou řídicí technikou. Tato technika (PLC) sice zůstala na dlouhou dobu pozadu v programátorském komfortu za řídicími počítači a minipočítači, na druhé straně vykazovala nesporné výhody. Mezi ty patří spolehlivost, snazší rozdělení řídicí struktury na samostatné celky s jasně definovatelnými rozhraními, vysoká spolehlivost, nižší náklady na kabeláž. Z toho plyne rychlejší uvedení do chodu, snazší údržba, jednodušší ladění programů, modulární výstavba a tím optimalizace ceny HW, vysoká stabilita jednoduchého operačního systému, nižší nároky na kvalifikaci projekčních a inženýrských pracovníků, celkové nižší náklady na realizaci projektu, uvedení do chodu a závěrečné fáze projektu. [1]

2.1 Programovatelné automaty

Vzhledem k tomu, že PLC nenahradily jen řídicí počítače a minipočítače, ale i malou automatizaci, reprezentovanou průmyslovými regulátory, bezkontaktní logikou a reléovou logikou, bylo pochopitelné, že jedním z kategorických požadavků průmyslu (projektantů, elektroinženýrů a středních odborných pracovníků) byl především jednoduchý programovací jazyk, který by byl velmi podobný jazyku logických schémat, booleovským rovnicím, reléovým schémátům, assembleru. Díky těmto jednoduchým programovacím "jazykům" bylo poměrně jednoduché klasickou techniku logického řízení nahradit programově orientovanými a tedy nesrovnatelně flexibilnějšími řídicími systémy - programovatelnými automaty. Programovatelný automat umožňuje logické rovnice naprogramovat, zatímco předcházející bezkontaktní nebo reléová logika (nebo v dnešní době programovatelná logická pole) řeší logické rovnice fyzickým propojením logických členů. Jakákoli změna logické struktury se snadno provede změnou programu programovatelného automatu, což je podstatně jednodušší, než přepojení reléového nebo logického schéma. Odhlédneme-li od počáteční nespolehlivosti pivních programovatelných automatů (způsobené především nespolehlivostí elektronických součástek), náhrada relé a bezkontaktní logiky programovatelnými automaty byla jednoduchá a úspěšná.

V případě náhrady řídicího počítače programovatelným automatem nebyla situace pro novou technologii zdaleka tak příznivá. Pokročilejší programovatelné automaty sice vykazovaly již dostatečnou spolehlivost a rovněž organizace projekčních prací a jejich realizace byly výrazným zjednodušením oproti centralizovanému návrhu, sériovém ladění jednotlivých úloh a uvádění složitého systému do chodu, na drahé straně programátorský komfort minipočítačů se programovým prostředím PLC nahradil v plné míře až s příchodem SCADA systémů.

Výhody a nevýhody programovatelných automatů: [1]

Výhody :

- rychlé přeprogramování úlohy

- malá varieta náhradních dílů
- možnost vystavění velké hierarchické struktury dle potřeby
- flexibilita (naprojektování na míru)
- modularita (možnost rozšíření)
- hospodárnost (levné velmi malé a malé kompaktní automaty)
- vestavěná diagnostika vlastního PLC
- možnost tvorby diagnostiky vnější
- jednoduché programování
- možnost použití vyšších programovacích jazyků u nových automatů
- jednoduchý a tím spolehlivý OS reálného času
- velká nabídka kvalitních přístrojů různých výrobců B.

Nevýhody:

- nižší programátorský komfort než u minipočítačů
- vyšší cena než IPC ekvivalentního výkonu při nižším programátorském komfortu PLC
- menší flexibilita ve srovnání s IPC
- užití nedostatečné standardizovaných sériových komunikačních sběrnic pro propojení automatů do sítě
- nezbytnost hierarchické architektury při propojování do větších celků

2.2 Charakteristiky PLC

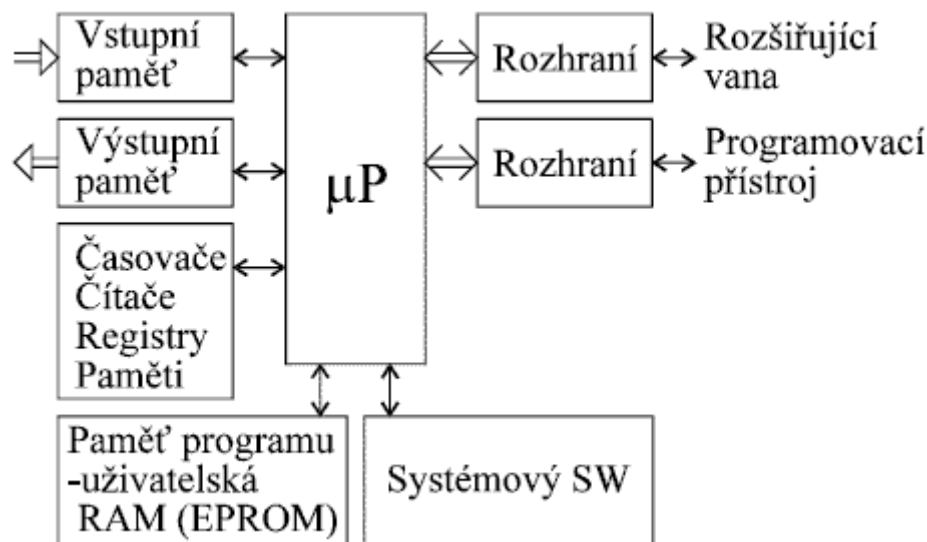
2.2.1 Technické řešení (HW) programovatelných automatů (PLC)

V době svého vzniku (konec 60. let) si programovatelné automaty kladly za úkol nahradit efektivnějším způsobem reléovou a později i bezkontaktní logiku. Proto jejich architektura vycházela z toho, že budou zpracovávat binární informaci. Jako HW jádro používaly bitové procesory. To mělo za následek, že v době velmi pomalých procesorů s 8 nebo 16 bitovým slovem (v průběhu 70. let), se jevíly bitové procesory jako velmi rychlé, kvaziparalelní řešení ve srovnání s 8 a lóbitovými procesory. Proto se na architekturu PLC kladly následující nároky:

- bitové orientovaná CPU
- bitové orientovaná paměť dat
- slovně orientovaná paměť programu
- rozhraní na programovací přístroj
- jednoduchý instrukční soubor na zpracování logických rovnic
- systém speciálních funkcí (časovače, čítače a další)

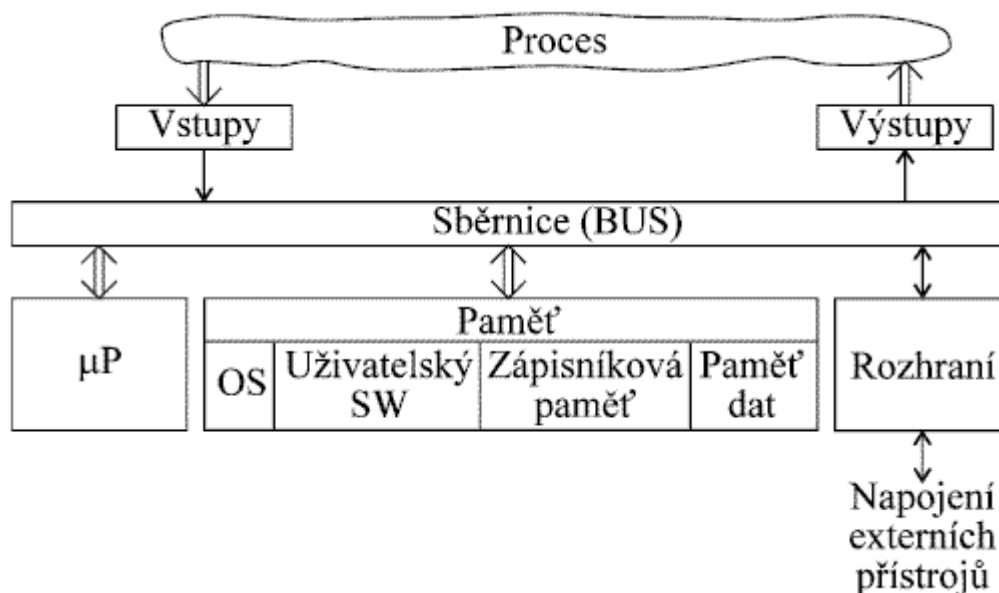
Takto zkonstruovaný PLC se do dnešní doby nezachoval. Rychlost a příznivá cena výkonných mikroprocesorů umožňuje použití slovně orientovaných mikroprocesorů i u velmi malých PLC. Přesto se blokové schéma velmi malých,

kompaktních PLC liší od architektury středních a velkých automatů, jak je patrné z **OBR 2.1** a **OBR 2.2**.



OBR 2.1: Blokové schéma velmi malého PLC. [1]

Řízení logické úrovně je nemyslitelné bez toho, aby byly k dispozici v základním vybavení každého PLC časové funkce (časovače) a funkce čítání impulsů (čítače). Proto každý PLC má tyto dvě funkce v základním programovém vybavení.

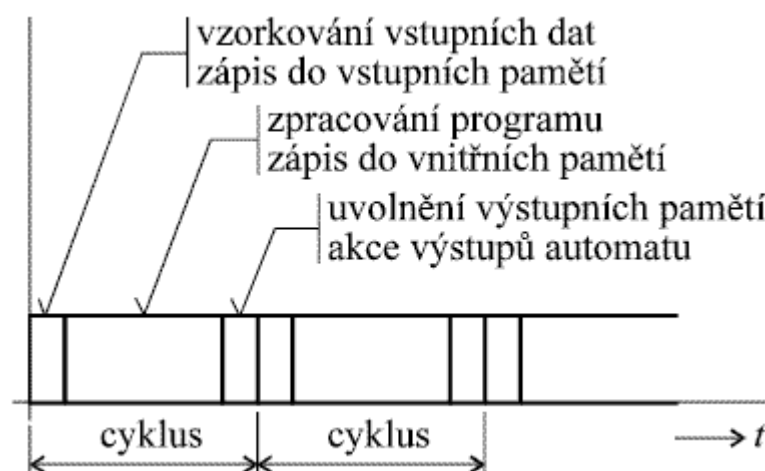


OBR 2.2: Blokové schéma standardního modulárního PLC. [1]

Je patrné, že blokové schéma standardního modulárního PLC je velmi podobné na architekturu mikropočítače. Základ tvoří vnitřní 16 nebo 32 bitová sběrnice, kolem které je modulárně vytvořen celý PLC.

Zatímco u prvních PLC s bitové orientovanou CPU byla paměť programu oddělena od paměti dat nebo naopak a pro data se používala i jiná (bitová) organizace paměti, dnešní PLC mají jednu operační paměť, ve které jsou vyhrazeny prostory pro vstupní data, výstupní data, vnitřní proměnné a paměťový prostor na vlastní program. Kromě toho jsou v paměti uloženy i funkční bloky a funkce jak systémové, tak vytvořené uživatelem.

Způsob práce, který od počátku charakterizuje PLC a odlišuje je od řídicích mikropočítačů, t.j. cyklický způsob vykonávání programu zůstal základním režimem prakticky všech PLC. Tento základní režim práce PLC je ukázán na **OBR 2.3**. [1]

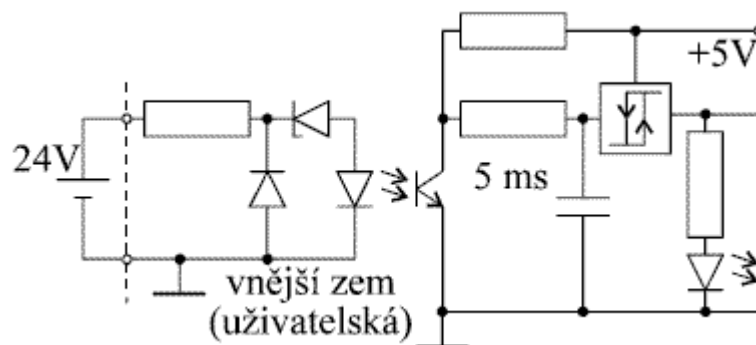


OBR 2.3: Časový diagram základní funkce PLC. [1]

2.2.2 Vstupní/výstupní podsystém PLC

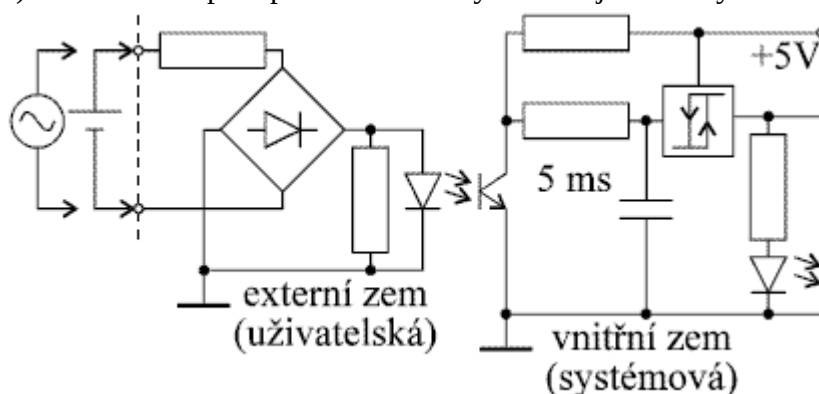
Programovatelný automat má perfektně propracovaný podsystém vstupů a výstupů. Tato vlastnost (platí i pro velmi malé PLC), která charakterizuje tyto řídicí systémy, je jednou z příčin jejich trvalé popularity. Zatímco s nákupem programovatelného automatu uživatel přirozeně nakupuje perfektně vyřešený podsystém vstupů a výstupů v pevné konfiguraci (kompaktní PLC) nebo v jednotlivých V/V modulech. V případě, že se rozhodne pro některou jinou alternativu řídicího systému (jako mikropočítač, programovatelné hradlové pole apod.), musí podsystém vstupů a výstupů řešit.

Na **OBR 2.4** je principiálně znázorněno řešení karet digitálních vstupů PLC.

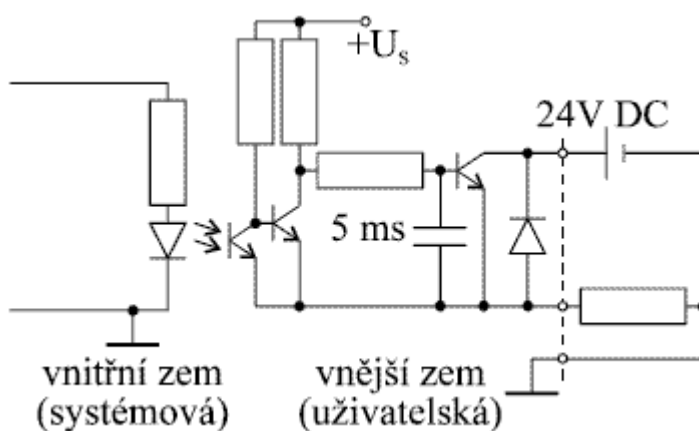


OBR 2.4: Digitální vstup. [1]

Digitální vstupy bývají i v univerzálním provedení (střídavé i stejnosměrné napětí viz **OBR 2.5**). Kromě galvanického oddělení jsou vybaveny filtrem pro odfiltrování poruchových signálů (3-10 iris RC filtr na vstupu) a diodami pro ochranu proti prepólování a proti napěťovým špičkám (**OBR 2.4**). Digitální výstupy jsou buď v nevýkonovém provedení do max. stovek mA (s tranzistorem **OBR 2.6**) nebo výkonové (s tyristorem) nebo reléové pro spínání větších výkonů stejnosměrných i střídavých.

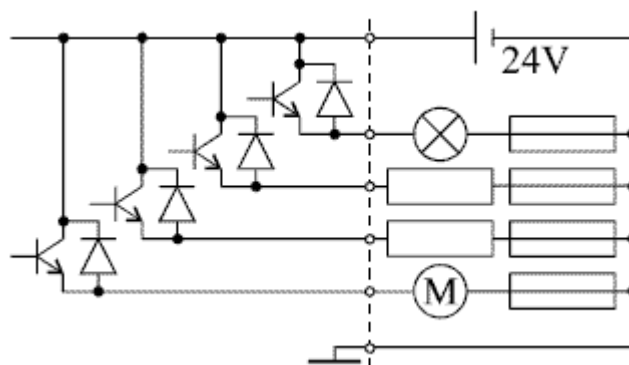


OBR 2.5: Univerzální digitální vstup. [1]



OBR 2.6: Zapojení digitálního výstupu. [1]

Kromě digitálních (též binárních) vstupů a výstupů DI/DO poskytují uživatelé jak mnohé kompaktní, tak především modulární PLC i analogové vstupy a výstupy. Analogové vstupy jsou napěťové pro stejnosměrné napětí s různou přesností A/D převodníků (typicky 12bitové), nebo proudové (4 - 20 mA), nebo proudové pro připojení Pt100. Analogové výstupy jsou v různém provedení dle požadovaného výkonu. Často se analogový výstup realizuje formou šířkově modulovaných výstupních impulsů konstantní amplitudy. [1]



OBR 2.7: Zapojení skupiny výstupů. [1]

2.2.3 Speciální moduly

U modulárních PLC najdeme velkou nabídku speciálních modulů, které rozšiřují působnost programovatelného automatu daleko za původní hranice jeho určení. Jde např. o moduly realizující až 8 regulačních smyček s volitelnými parametry PID regulátorů. Dále o moduly pro řízení hydraulických servoventilů a proporcionálních ventilů, moduly diagnostiky, vizualizační moduly pro sledování procesu na připojených provozních obrazovkách (tyto velmi drahé moduly již ztratily na významu s příchodem prostředí Windows), moduly pro měření teploty, pro regulaci teploty, a další. Zvláštní místo mezi speciálními moduly zaujímají moduly komunikační. Pomocí těchto modulů jsou automaty propojovány navzájem mezi sebou a s vyššími i nižšími úrovněmi řízení po průmyslových sériových sběrnicích. [1]

2.2.4 Silová část PLC

Do této části zahrneme jak rám programovatelného automatu, tak napájecí zdroj. Kompaktní a velmi malé PLC jsou v jiném mechanickém provedení, než modulární PLC. Napájecí zdroj, CPU, podsystém I/O, speciální moduly (komunikační rozhraní) jsou provedeny v jednom kompaktním celku bez možnosti rozšíření (jedině připojením dalšího -rozšiřujícího kompaktního modulu, viz kupř. Télémécanique TSX). Naproti tomu mechanické provedení modulárních PLC umožňuje modulární výstavbu systému, volbu její minimální až maximální varianty, připojování rozšiřujících rámců, propojování automatů navzájem mezi sebou a se sousedními úrovněmi řízení pomocí sériových sběrnic. Rám se již zcela výjimečně konstruuje robustně, s vývojem mikroelektroniky se systémy zmenšily z velikosti odpovídající dvojitému evropskému formátu na jednoduchý. Jen nejvýkonnější automaty jsou pak stále ještě v provedení 2E. Moduly

(karty) zaujímají v rámu jednu nebo 2 pozice. Provedení rámu zpravidla definuje i pozici karet. Výkonový zdroj (buď na 230V AC. nebo na 24V DC) má danou pozici, podobně jako CPU. Rovněž přídatná paměť a komunikační procesory mají přesné místo v rámu. Ostatní karty (někdy s výjimkou speciálních karet) mají polohu libovolnou. Rám je vybaven vnitřní paralelní systémovou sběrnici pro signálové a elektrické propojení modulu s CPU a pamětí. Umístění karty v rámu však zpravidla určuje i její adresu a dle tohoto fyzického uspořádání karet je třeba systém zkonfigurovat. Mechanické upevnění modulů (karet) je v provedení "plug-and-play" a klade na obsluhu minimální nároky. U předcházející generace středních a velkých PLC byl v rámu umístěn i ventilátor. Rám se zpravidla montuje na montážní lištu do rozvaděče, neboť současná generace PLC již není v takovém IP krytí, aby se umísťovala přímo do procesu. Rám zpravidla obsahuje i modul s konektory pro propojení rozšiřujících rámů. Tak je možné rozšířit počet I/O. příslušejících k dané CPU. umístěné v základním rámu. Vzdálenost bývá buď krátká (do 1m. když se jedná o rozšíření v jednom rozvaděči) nebo do 10 m. [1]

2.2.5 Modul CPU

Tento modul tvoří jádro PLC a je zpravidla k dispozici v různém provedení dle výkonu procesoru. Modul CPU je také jediným modulem, v němž se během životnosti daného PLC dá pružně reagovat na vývoj mikroelektronické technologie. Moduly CPU mají též zabudováno některé programovací nebo komunikační rozhraní pro přenos programu do paměti programu. Moduly CPU modulárních PLC musí toto programovací rozhraní obsahovat z principu, u modulárních záleží na výrobci, zda rozhraní provede jako samostatný modul nebo zda jím vybaví modul CPU. Rozdíl v ceně modulu CPU s rozhraním kupř. Profibus a bez něj je značný. [1]

Modul CPU je zpravidla vybaven:

- procesorem s 16 nebo 32. výjimečně s 64 bitovým slovem
- může být doplněn bitovým procesorem pro rychlé řešení logických rovnic (jen u velkých a některých středních PLC a u starší generace PLC)
- bitovými registry (flags)
- pamětí EPROM pro operační systém
- pamětí RAM (zálohovanou) pro program
- pamětí RAM (zpravidla zálohovanou) pro V/V data
- programovacím rozhraním
- standardizovaným rozhraním s protokolem pro sériovou komunikaci s nižší i vyšší řídicí úrovní

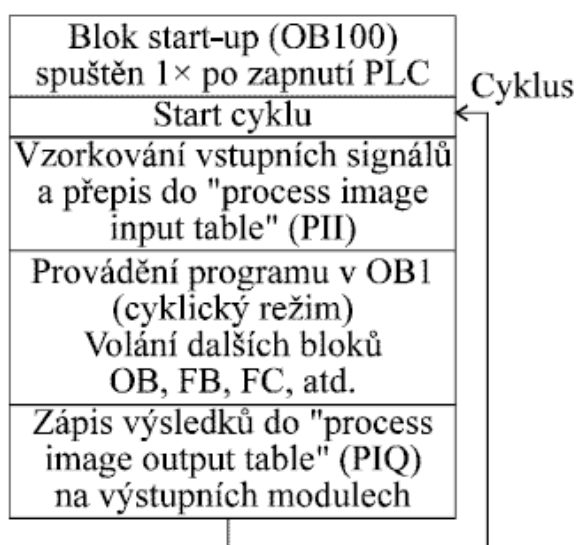
2.3 Programování PLC

2.3.1 Úvod do programového vybavení PLC

Základní režim práce programovatelných automatů je cyklický režim provádění řídicího programu. Tím se zásadně liší od počítačů pro vědecko - technické výpočty i od řídicích počítačů, minipočítačů a mikropočítačů. Je to dáno tím, že původním posláním PLC byla náhrada reléové a bezkontaktní logiky. Ta ovšem také pracuje jako paralelní řídicí systém, protože na danou kombinaci vstupních signálů (vstupní vektor) dává v libovolném okamžiku odpovídající reakci (výstupní vektor). Programovatelné automaty simulují tuto funkci rychlým sériovým zpracováním vzorku vstupního vektoru a vysláním výstupního vektoru opět v jednom okamžiku. To předpokládá, že řídicí program obsahuje všechny logické funkce mezi vstupními, vnitřními a výstupními signály, které jsou potřebné pro řízení daného technického zařízení v logickém sledu. Dále to předpokládá, že programy jsou řízeny jednoduchým operačním systémem reálného času, který zaručuje naskenování vstupního vektoru, výpočet všech logických rovnic, které popisují daný řídicí systém a vyslání výstupního vektoru na výstupní zařízení, která realizují výstupní signály. Kromě toho musí operační systém také zajistit spuštění a stop PLC a jeho vnitřní diagnostiku. [1]

2.3.2 Operační systém PLC

Operační systém je soubor programů, které spolu s dalšími vlastnostmi výpočetního systému tvoří základ pro pracovní režimy výpočetního systému. Zejména vykoná řízení a dohled nad běžícími programy. Programovatelné automaty PLC jsou vybaveny zpravidla jednoduchým, spolehlivým operačním systémem reálného času. Jednoduchost a spolehlivost operačního systému (OS) programovatelných automatů hraje významnou roli v konkurenceschopnosti PLC oproti IPC a dalším prostředkům průmyslové automatizace. [1]



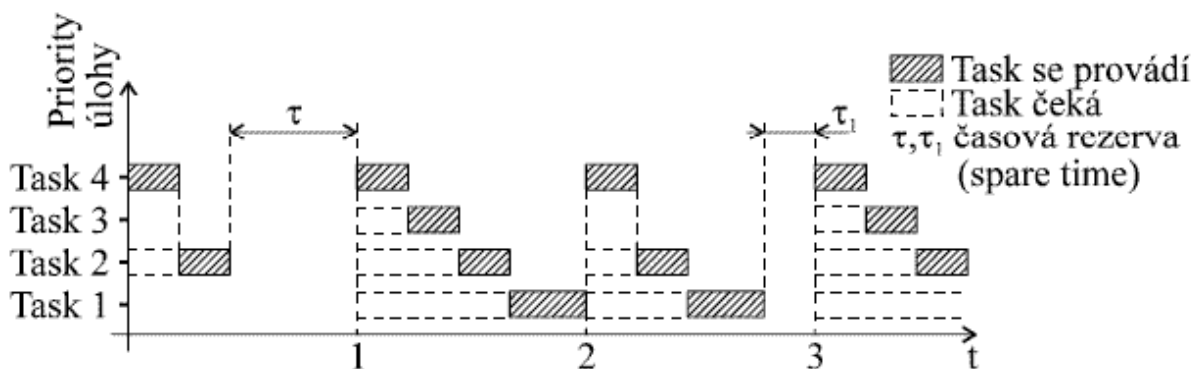
OBR 2.8: Cyklický režim PLC. [1]

Tento operační systém musí zajistit výše uvedené funkce pro provádění jak cyklického programu (**OBR 2.8**), tak reagování PLC na vnitřní i vnější události mimo cyklický režim. Režim přerušení může být parametrizován, takže časově kritické akce mohou být obslouženy mimo cyklus PLC.

Proto je v PLC implementována některá z forem mnohonásobného provádění úloh, tzv. multitasking. V současné době, kdy většina PLC má k dispozici jen jednu CPU, nejde o skutečný paralelní běh úloh na multiprocesorovém systému, ale pouze o souběžné vykonávání několika úloh na jediném procesoru. V dalším uvedeme princip jednoduchého multitaskingového systému, implementovaného často na stávajících PLC.

Jedná se o jistou jednoduchou formu multitaskingu, kdy i zde lze rozlišovat přerušení od časovače (time-driving-multitasking) a přerušení od procesu (event-driving-multitasking). Na **OBR 2.9** je znázorněna funkce PLC při time-driving-multitasking.

Jednotlivým úlohám (task) jsou přiřazeny priority 1 až 4. Čím vyšší číslo, tím vyšší prioritita. Úlohy 4 a 2 se vykonávají v každém cyklu, protože jde o časově kritické úlohy, které nestačí ošetřovat s delší periodou. Naopak úlohy 3 a 1 stačí ošetřovat jen každý druhý cyklus. Na **OBR 2.9** je dále vidět, že dle priority jsou tasky vykonávány v pořadí 4-3-2-1 a dále to, že doba cyklu není dostatečně dlouhá, takže task 1 se nestačí vykonat v jednom cyklu. Zbytek úlohy (task) se proto musí vykonat v následujícím cyklu, ale s prioritou nižší, než mají úlohy 4 a 2. V tomto režimu může každý uživatel psát svoji úlohu nezávisle na ostatních úlohách. Každý task využívá samostatně "svoje vstupní veličiny". Funkce přerušení je podobná i v případě přerušení od procesu. Kupř. změna vstupní veličiny způsobí start příslušného tasku. Některé multitaskové systémy umožňují libovolnému tasku využívat libovolné proměnné (vstupní, vnitřní proměnné).



OBR 2.9: Time-driving-multitasking u PLC. [1]

I při existenci přerušení zůstává ve většině případů cyklický režim funkce, znázorněný na Obr. 9. základním pracovním režimem. Doba cyklu PLC je zpravidla definována jako doba, kterou PLC potřebuje k načtení dat, vyslání dat na výstupy a zpracování 1k instrukcí (nikoli 1 instrukce, neboť program o 1 instrukci nedává smysl). Typická doba cyklu moderních PLC (k roku 2000) je 1 - 5 ms, přičemž PLC s rychlými CPU do 1 ms jsou k dispozici (Mitsubishi a další). [1]

2.3.3 Programovací jazyky

Stávající programovací jazyky PLC jsou velmi jednoduché. Odpovídají stále ještě ve velké míře svému původnímu poslání, totiž umožnit projektantům logické úrovně řízení přejít z HW řešení booleovských rovnic (pomocí reléové nebo bezkontaktní logiky) na řešení Booleovských rovnic programem pro programovatelný automat. Proto převažují tyto programovací jazyky:

A. Grafické

- jazyk reléových schémat - Ladder Diagram (LD)
- jazyk logických schémat - Function Block Diagram (FBD)
- mnemo kód (SL - Statement List)

S vývojem HW programovatelných automatů docházelo i k vývoji programovacích jazyků. Přesto v Německu a střední Evropě se používá v největší míře mnemo kód (německy AWL- Anweisungsliste) kombinovaný s programováním pomocí funkčních bloků a jazyk reléových schémat LD. Ve Francii se prosadil moderní a velmi efektivní jazyk pro popis sekvenčního systému GRAFCET, kombinovaný s LD. Na severoamerickém kontinentu se používá takřka výhradně Ladder Diagram. Před několika lety mezinárodní elektrotechnická komise IEC vydala normu, týkající se programovatelných systémů a systémů logického řízení jako takových. Klade si za úkol maximální normalizaci HW a zejména způsobu programování PLC. Je celou řadou výrobců PLC akceptována, avšak má i celou řadu kritiků a její osud není znám s ohledem na rychlý vývoj mikroelektronické technologie. Přesto zůstává nejvýznamnější iniciativou v oblasti programování PLC od vzniku této techniky.

Třetí část Normy IEC 1131-3 se zabývá programováním a programovacími jazyky. Zahrnuje všechny nejobvyklejší programovací jazyky a doplňuje je na následující množinu:

Grafické:

- LD
- FBD
- SFC

Algebraické:

- IL
- ST (Structured Text), vyšší programovací jazyk, podobný PASCALu

Sequential Function Chart (SFC) je grafický jazyk vycházející důsledně z GRAFCETu a prakticky totožný s jeho průmyslovou variantou.

Kromě toho programovací model IEC 1131-3 vychází z prvků strukturovaného programování a podporuje další programovací přístupy, které v oblasti PLC nebyly příliš používány jako povinná deklarace proměnných na počátku programu, jasně definované požadavky na funkce a funkční bloky ap., což v konečném důsledku spěje k systémově neutrálnímu programování PLC, umožňujícímu přenositelnost programu z jedné HW platformy (automatu) na druhou a to i jiného výrobce.

U PLC byl vždy kladen velký důraz na spolehlivost a bezpečnost. V dnešní době se tyto požadavky kladou na všechny řídicí systémy ve zvýšené míře. Programovatelné automaty se vyznačují vysokou odolností proti rušení (potenciálové oddělení vstupů a výstupů), definovaným chováním v případě výpadku napájení (uvedení do bezpečného stavu, automatický restart, uložení programu v nedestruktivní paměti, diagnostika vnitřní i vnější). [1]

3 Základní navození problému o řešené úloze

3.1 Vznik úlohy

Tato úloha vznikla jako soutěžní projekt do soutěže Xplore New Automation 2008. Tuto soutěž každoročně vypisuje firma Phoenix Contact za podpory ministerstva školství DE. Soutěž byla teprve v posledním soutěžním období zpřístupněna soutěžícím týmům z České Republiky. Tým řešitelů prošel výběrovým řízením, ve kterém bylo vybráno pouze 100 týmů z celého světa. Účast v soutěži je přínosem nejen z důvodu zviditelnění fakulty, ale především jako možnost získat velké zkušenosti s projektem podobné velikosti a získání drahého vybavení pro laboratoře. Cílem soutěže je, kromě nezvyklé reklamy firmy Phoenix Contact, ukázka co vše a s jakou složitostí lze s PLC provádět. Vzhledem k těmto skutečnostem, byla úloha navržena a realizována právě s ohledem na atraktivitu celého provedení.

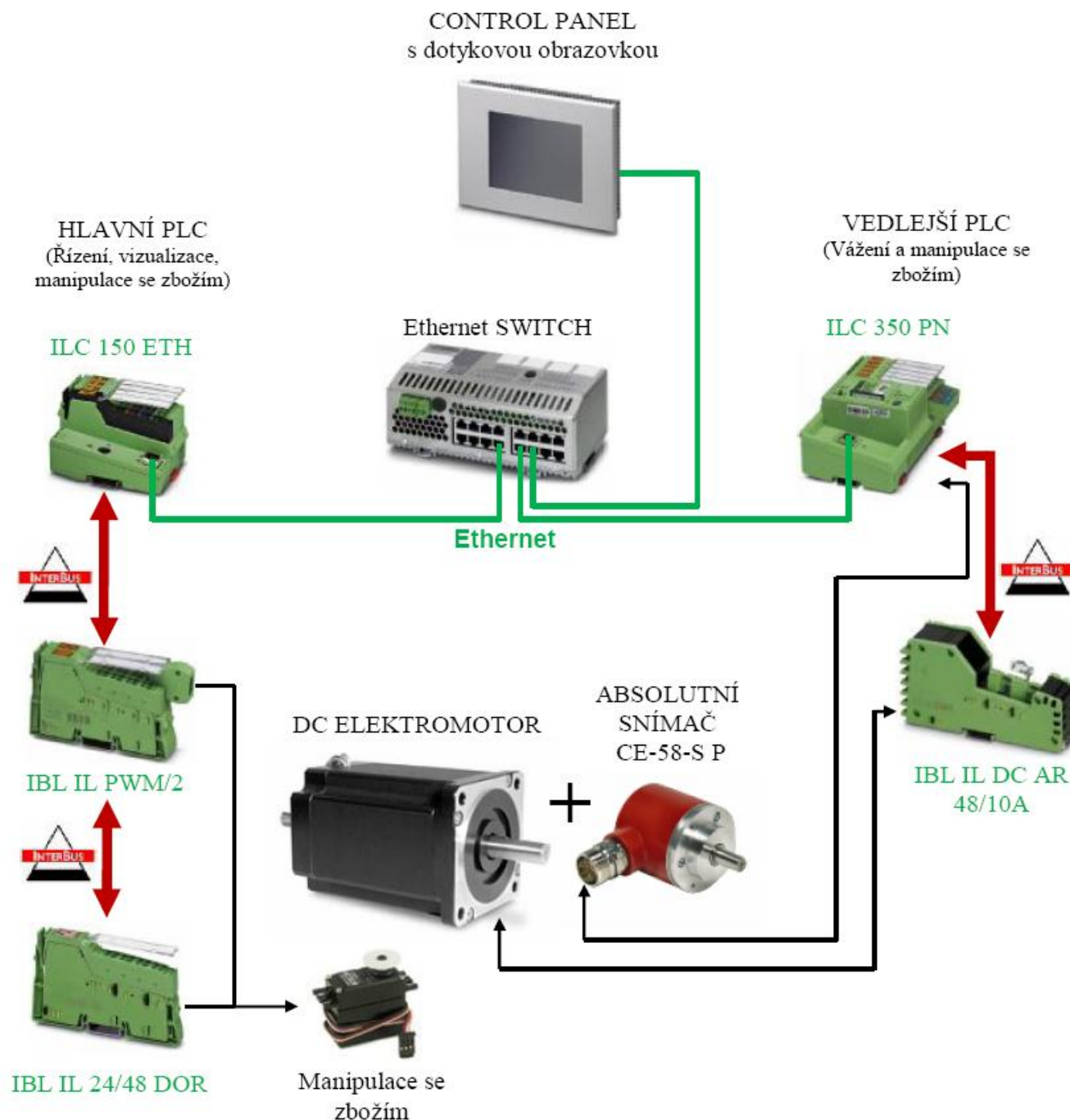
3.2 Popis komponent

Veškeré komponenty, rozšiřující primární funkce PLC, jsou objednány rovněž od firmy Phoenix Contact. Přídavné komponenty mají stavebnicový charakter, stejně jako použité PLC. Celá hierarchie propojení jednotlivých součástí je jasně zřetelná ze schématu v **OBR 3.2**. Z tohoto schématu je rovněž zřetelné, kterou částí řešení se práce zabývá.

Popis komponent se logicky týká pouze těch zařízení které spadají do rozsahu práce. Jsou to ta zařízení na jejíž řízení se podílí hlavní (řídící) PLC ILC 150 ETH. Vedlejší PLC podléhá rovněž požadavkům řídícího PLC.



OBR 3.1: Ukázka celého zařízení.



OBR 3.2: Schéma zapojení zařízení pro řízení procesu třídění, včetně použitých komponent

Na obrátku zobrazený PLC ILC 150 ovládá své rozšiřující moduly IBL IL PWM/2 a IBL IL 24/48 DOR. Moduly jsou použity pro obsluhu servo-pohonů, které jsou zobrazeny jako symbol „manipulace se zbožím“. Parametry všech zařízení jsou uvedeny v tabulce. Obecně je potřeba řídit 5 servo-pohonů (1 nakládací a 4 vykládací) generátorem impulsů o vlnové délce 0,9 až 2,1 ms. Jako generátor impulsů slouží modul PWM/2. Ten má pouze 2 výstupy. Činnost ovládání serv je rozdělena na ovládání

nakládajícího serva (1 výstup) a všech 4 vykládajících serv (2 výstup). Problém aby zároveň neběžela všechna 4 vykládající serva řeší modul 24/48 DOR. Tento modul slouží jako reléový výstup a u serv která nebudou v činnosti vypne napájení.

Pro komunikaci s dotykovou obrazovkou a druhým PLC ILC 350 je použit klasický ethernet switch.

3.3 Použité komponenty

3.3.1 ILC 150 ETH – řídicí PLC

Číslo produktu	29 85 33 0
Programovací prostředí	PC Worx 5 a vyšší

Rozměry, hmotnost a odolnost vůči teplotě

Šířka	80mm
Výška	119,8mm
Váha	285g
Pracovní teplota	-25 °C ... 55 °C

Datové rozhraní

Vnitřní sběrnice	INTERBUS
Způsoby propojení	RS-232-C, Ethernet 10Base-T/100Base-TX (10/100Mb/s)

Zdroj

Napětí zdroje	24 V
Rozsah zdroje	20.4 V DC... 30 V DC
Běžná spotřeba	200 mA
Maximální spotřeba	860 mA

Údaje o sběrnici ITERBUS

Počet připojitelných zařízení	Max. 128
Počet vstupů a výstupů	Max. 4096
Baterie	Integrovaná nabíjecí
Počet řídicích operací	8
Počet čítačů, časovačů	Závislý na datové paměti
Počet datových bloků	Závislý na datové paměti
Datová paměť	256 kB
Vyrovnávací paměť	8 kB (NVRAM)

3.3.2 IBL IL PWM/2 – pulzní generátor

Číslo produktu	27 42 61 2
----------------	------------

Rozhraní

Rozhraní	Vnitřní lokální sběrnice
Typ zapojení	Vnitřní datový konektor
Přenosová rychlost	500 kb/s

Zdroj

Napětí pro komunikaci	7.5 V DC
Běžná spotřeba proudu	100 mA (průměrná)
Napětí pro vst. a výst.	24 V DC
Zvlnění	5%
Napět'ový Rozsah zdroje	19.2 V DC ... 30 V DC

Výstupy

Počet výstupů	2
Výstupní napětí	5 V DC
Výstupní proud	10 mA (5 V), 500 mA (24 V)

3.3.3 IBL IL 24/48 DOR 2/W-PAC

Číslo produktu	28 62 97 4
----------------	------------

Rozhraní

Rozhraní	Vnitřní lokální sběrnice
Typ zapojení	Vnitřní datový konektor
Přenosová rychlost	500 kBaud

Digitální výstupy

Počet výstupů	2
Maximální proud pro jeden kanál	2A
Jmenovité výstupní napětí	48 V

Zdroj

Napětí zdroje	24 V DC
Napětí pro komunikaci	7,5 V
Napět'ový Rozsah zdroje	19.2 V DC ... 30 V DC
Spotřeba proudu	Max. 30 mA (z lokální sběrnice)
Spotřeba zdroje	Max. 0,23 W (z lokální sběrnice)

Obecný přehled

Hmotnost	48g
Teplota prostředí	(pracovní) -25 °C ... 55 °C (uskladnění) -25 °C ... 85 °C
Dovolená vlhkost	Za provozu - 75% ... 85% (při stanovené teplotě) Při uskladnění - 75% ... 85%
Tlak vzduchu	Za provozu - 70 kPa ... 106 kPa (do 3000m n. mořem) Při uskladnění - 70 kPa ... 106 kPa (do 3000m n. mořem)

3.3.4 TP 12 T – Dotyková obrazovka**Obrazovka**

Číslo produktu	29 85 19 4
Hmotnost	3,465 kg
Obrazovka	12,1" TFT aktivní
Rozlišení	800 x 600 bodů
Dotyková obrazovka	Odporová průmyslová dotyková obrazovka

Počítač

Procesor	Xscale PXA255, 400 MHz
Operační paměť	64 MB SDRAM
Paměť pro data	256 MB externí Compact Flash karta pro aplikace
Rozhraní	10/100 Mb Ethernet, 2 x USB Host 2.0

Obecný přehled

Šířka	340 mm – externí rozměr
Výška	270 mm – externí rozměr
Hloubka	5 mm – externí rozměr

Schopnost odolat okolním vlivům

Stupeň ochrany	IP65 (přední část), IP20 (zadní část)
Pracovní teplota	0 °C ... 50 °C
Teplota skladování	-25 °C ... 70 °C
Dovolená vlhkost	Pracovní - 10% ... 95% nekondenzovaná Pro uskladnění - 10% ... 95% nekondenzovaná

3.3.5 HS-475 HB – servo pohon

Napájecí napětí	4,8 V až 6 V
Převody	Plastové
Ložiska	Kuličková
Rozměry	39x20x36 mm
Hmotnost	30 g
Síla při 4,8V	44 Ncm

Rychlost posuvu (4,8V) 60° / 0,23s

4 Struktura programu

4.1 Úvod do struktury programu

Řídící program PLC je vytvořen v prostředí PC Worx. Prostředí je použitelné pouze pro PLC firmy Phoenix Contact a většinou bývá pravidlem, že každý výrobce PLC používá své vlastní programovací prostředí. Program PLC má takovou strukturu tak, že hlavní část programu běží v reálném čase (ta bude dále označována jako MAIN) a ostatní části jsou ve většině případů realizovány pomocí podprogramů a funkčních bloků (dále FB). Reálným časem se rozumí ta činnost PLC, kdy má činnost s vyšší prioritou přednost před běžně prováděnými úkony.

Všechny činnosti programu zařízení pro optimalizaci vážení běží v MAIN a jsou rozděleny na pod sektory pomocí hodnot podmínky, jež pracuje s proměnnou iKrok. Proměnná, má právě tu hodnotu, ve které fázi se běžící program nachází. Ve výsledku se celá činnost jeví jako navzájem na sebe navazující bloky (Struktura je graficky znázorněna na **OBR 4.2**, **OBR 4.3**, **OBR 4.4** a **OBR 4.5** v kapitole 4 formou vývojových diagramů). Hodnota proměnné iKrok je využita i pro displeje komentářů ve vizualizaci. Jsou to displeje ukazující status PLC, tedy informují obsluhu o právě probíhající činnosti PLC, varují obsluhu v případě že nastala chyba v komunikaci a slouží taktéž jako vodičko (např.: Stiskněte Start apod.).

FB jsou podprogramy a v našem případě jsou FB volány z pod sektorů hlavního programu. Můžeme je v zásadě rozdělit na bloky které si programátor vytvoří sám a předdefinované, ty jsou integrovány přímo v prostředí PC worx. Při vytváření FB je výhodné aby blok pracoval, pokud je to možné, nezávisle na hlavním programu. Nezávislé bloky lze poté využít v jakýchkoli nově naprogramovaných aplikacích.

4.2 Síťová komunikace obecně

Pro síťovou komunikaci jsou využívány předdefinované FB, které jsou detailně popsány v dalších podkapitolách. Je nutné aby byly FB umístěny na vhodném místě v programu například z důvodu, že reagují pouze na náběžnou hranu hodinového impulsu. V prostředí PC Worx jsou k dispozici 4 předdefinované FB sloužící pro komunikaci, z nichž každý má specifickou funkci (kontrola navázání spojení, posílání dat, příjem dat a spojení které se ukončí po definovaném čase).

Po nastavení parametrů nutných pro správnou funkci FB (IP adresa, ID, atd.), je část programu ve kterém je FB obsažen, schopen činnosti pro kterou je určen. Reakce na náběžnou hranu hodinového impulsu, která je nutná k aktivaci FB, může být realizována částí programu, nebo pomocí programátorem vytvořeného FB. Tento se chová jako oscilátor hodinových impulsů. Prakticky je využit „oscilátor“ za pomoci vlastního FB, ten má i jiné vlastnosti (dále bude zmiňován jako FB SEND). Generování impulsů je realizováno tak, že při každém zavolání FB se změní stav proměnné iStep. Proměnná iStep nabývá hodnot TRUE a FALSE, kdy hodnota TRUE spustí FB pro přenos dat.

4.3 Sít'ová komunikace prakticky – popis funkčních bloků pro komunikaci

Pro sít'ovou komunikaci jsou využívány FB IP_CONNECT (kontrola navázání spojení), IP_TIME (spojení jež se ukončí po stanoveném čase), IP_URCV (příjem dat), IP_USEND (odesílání dat). V praxi jsou využity všechny FB jenž jsou k dispozici, kromě bloku IP_TIME. Práce bloků je zmíněna níže, ale v zásadě je pouze potřeba nastavit nutné parametry pro komunikaci (IP adresa atp.) a přenos dat může být uskutečněn.

4.3.1 Funkční blok IP_CONNECT

Prvním FB jenž bude popsán je IP_CONNECT, jehož funkcí je navázání spojení. Obsahuje nastavitelné parametry EN_C s datovým typem BYTE a parametr PARTNER s datovým typem STRING. Do parametru PARTNER je nutno vložit IP adresu zařízení, se kterým chceme komunikovat. Způsob jakým je IP adresa vložena do parametru PARTNER následující: PARTNER='/IP=LXX.LXX.LXX.XXX. symboly „X“ jsou nahrazeny čísly.

Po nastavení parametru EN_C do hodnoty TRUE (tuto hodnotu můžeme také přepínat mezi dvěma stavy například pomocí snímače, spínače atd.), se FB IP_CONNECT pokusí navázat spojení se zařízením, jehož adresa je vepsána do parametru PARTNER. Ip. adresy které jsou pro PLC vyhrazeny jsou: 192.168.0.12 pro ILC 350 a 192.168.0.20 pro ILC 150. Parametr PARTNER tedy v programu pro PLC ILC 150 nastavíme na hodnotu PARTNER='/IP=192.168.0.12'. FB s nastavenými parametry vypadá takto:

IP_CONNECT_1(EN_C:=TRUE,PARTNER:='/IP=192.168.0.12')

Jednou z požadovaných výstupních hodnot FB IP_CONNECT je číslo spojení ID (v celém tvaru potom: IP_CONNECT_1.ID). Číslo je unikátní a plní funkci identifikace spojení. Je využito dalšími bloky souvisejícími s komunikací (odesílání a příjem dat, doposud jsme se bavily pouze o navázání spojení). Číslo ID je nutné zadat aby se komunikující zařízení identifikovala. Pokud by bylo zapotřebí komunikovat s více zařízeními současně, využilo by se více FB IP_CONNECT z nichž by každý vygeneroval své jedinečné ID.

FB IP_CONNECT má k dispozici i funkci VALID, při jejím použití jsme schopni určit zda je spojení mezi zařízeními úspěšné či nikoli. Tato možnost je využita pro signalizaci o navázaném spojení. Stav parametru VALID je zaznamenáván do proměnné xEthKomunikaceNavazana, která je pomocí dvou stavového zobrazovače (symbol LED diody) zobrazena na dotykovém panelu.

Název	Datový typ	Směr dat	Popis
EN_C	BOOL	Vstup	Při nastavení tohoto vstupu do hodnoty TRUE, se provede pokus navázání spojení s IP adresou nastavenou na vstupu „PARTNER“. Nicméně, To bude možné jen tehdy, když zavoláte FB obou komunikujících partnerů, takže oba partneři žádají o komunikační linku. EN_C = TRUE Spojení bude navázáno nebo udržováno. EN_C = FALSE - Spojení je ukončeno.

PARTNER	STRING	Vstup	Tento vstup specifikuje systém se kterým bude probíhat komunikace. Specifikace probíhá vložení IP adresy partnera. Během operace s blokem nesmí být měněn obsah „PARTNER“.
VALID	BOOL	Výstup	Tento výstup ukazuje, zda je komunikace stabilní. Výstup zůstane v hodnotě „TRUE“ po dobu existující komunikace.
ID	INT	Výstup	Tento parametr jedinečně identifikuje spojení. Při několikanásobném spojení, může být použit ID parametr k rozlišení mezi nimi. Oprávněnost ID je spjata s hodnotou TRUE parametru VALID.

Stručný popis použitých parametrů FB IP_CONNECT.

4.3.2 Funkční blok IP_USEND

FB IP_USEND slouží k odesílání dat a je vložen do FB SEND. FB SEND je blok vytvořený za účelem zjednodušení komunikace se spolupracujícím PLC, resp. odesílání a přijímání příkazů a dat. FB SEND mimo jiné obsahuje i FB IP_URCV a jednoduchý generátor vzestupných hran. Generátor vzestupných hran funguje tak, že při každém zavolání FB SEND je změněna proměnná „xVzestupnaHrana“ na její znegovanou hodnotu. Proměnná xVzestupnaHrana má datový typ BYTE, z čehož vyplývá že každé druhé zavolání FB SEND je doprovázeno náběžnou hranou, která může být použita k „aktivaci“ FB_USEND a FB_URCV. To že vzestupná hrana doprovází každé druhé zavolání FB SEND nemá prakticky žádný neblahý vliv na příjem a posílání dat. Vyplývá to z podstaty pracovního režimu PLC v reálném čase. Aktivní funkce právě pracujícího FB se provádí tak rychle, že generátor hodinových impulsů popsaný výše se chová téměř jako by čekal až si FB pro přenos a příjem dat „zažádá“ o vzestupnou hranu.

Vlastní posílání dat je prováděno pomocí parametru REQ, který musí při odesílání detekovat vzestupnou hranu. Ta je realizována již zmíněnou hodnotou proměnné xVzestupnaHrana. Dále je nutné nastavit parametr ID, s číslem spojení a vlastní přenášenou hodnotu. Přenos příkazů a dat mezi zařízeními, je realizována pomocí parametru SD_1, který může disponovat libovolným datovým typem. Parametr pracuje s proměnnou stWrite, která dále spolupracuje se vstupy stWrite.wCommand a stWrite.wValue používaných v FB SEND. Do těchto vstupů jsou vepsány příkazy a hodnoty které mají být odeslány. Jsou realizovány pomocí čísla v datovém typu word. Přijímací zařízení, čili PLC obsluhující vážící mechanismus, si již číslům přidělí vlastní funkční význam.

Potvrzení úspěšného odeslání dat lze ověřit parametrem FB_USEND.DONE. Pokud má DONE hodnotu TRUE, byla data úspěšně odeslána. V této návaznosti, tedy úspěšnosti odeslání všech dat (viz funkce parametru DONE), je FB SEND připraven na příjem dat. Příjem dat je realizován pomocí FB IP_URCV, který je popsán v další kapitole.

Název	Datový typ	Směr dat	Popis
REQ	BOOL	Vstup	S pozitivní hranou přivedenou na vstup, dáte impulz k přenosu dat. Blok je uzavřen po dobu přenosu a nelze přerušit, nebo resetovat.
ID	INT	Vstup	Během založení spojení je používán FB "IP_CONNECT". Jedinečné ID je přiděleno komunikačnímu spojení. ID je použito k informování vysílacího bloku "IP_USEND" o komunikujícím zařízení, která data budou odesílána. Můžete spojit vstup "ID" přímo s výstupem "ID" bloku "IP_CONNECT".
DONE	BOOL	Výstup	Tento výstup ukazuje zda blok úspěšně odeslal všechna data. Můžete jen vyhodnotit úspěch odesílání během periody se kterou náběžná hrana se vyskytuje na výstupu
SD_1	ANY	Vstup/Výstup	Toto datové slovo není odesíláno v datové oblasti TCP/IP nebo UDP protokolu. Na čistě ethernetovém rozhraní s TCP/IP nebo UDP komunikaci, například via patice, datová oblast obsahuje pouze čistá data. Data jsou posílána jako bytový tok. Ne Intel/Motorola konverzace.

Stručný popis použitých parametrů FB IP_USEND.

4.3.2 Funkční blok IP_URCV

FB IP_URCV slouží pro příjem dat a příkazů od vázícího PLC. Podobně jako IP_USEND je i IP_URCV vložen do FB SEND. FB IP_URCV je rovněž závislý na hodnotě parametru ID bloku IP_CONNECT.

FB IP_URCV se uvede v činnost, jakmile FB IP_USEND odešle svá data. Po odeslání dat, se změní hodnota proměnné „iStep“ v FB SEND z 0 na 1. Aby bylo přijímání dat možné musí mít parametr EN_R bloku IP_URCV hodnotu TRUE. Další nutné parametry jsou, kromě zmíněného ID, parametry RD_1 a NDR.

Parametr RD_1 slouží k přenosu dat (Obdoba SD_1 v předchozí kapitole), má libovolný datový typ a pracuje s proměnnou stRead. Ta dále pracuje s proměnnými stRead.wCommand a stRead.wValue. Jejich hodnoty jsou přiděleny vstupům wReadCommand a wReadValue Bloku SEND. Všechny zmíněné vstupy a proměnné patří do FB SEND. Při příjmu dat od spolupracujícího PLC, se porovnávají data přijímaného příkazu (wReadCommand) a data přijímaných hodnot (wReadValue). Jakmile mají požadovanou velikost, poté co spolupracující PLC ukončí činnost vážení, ukončí hlavní program činnost FB SEND.

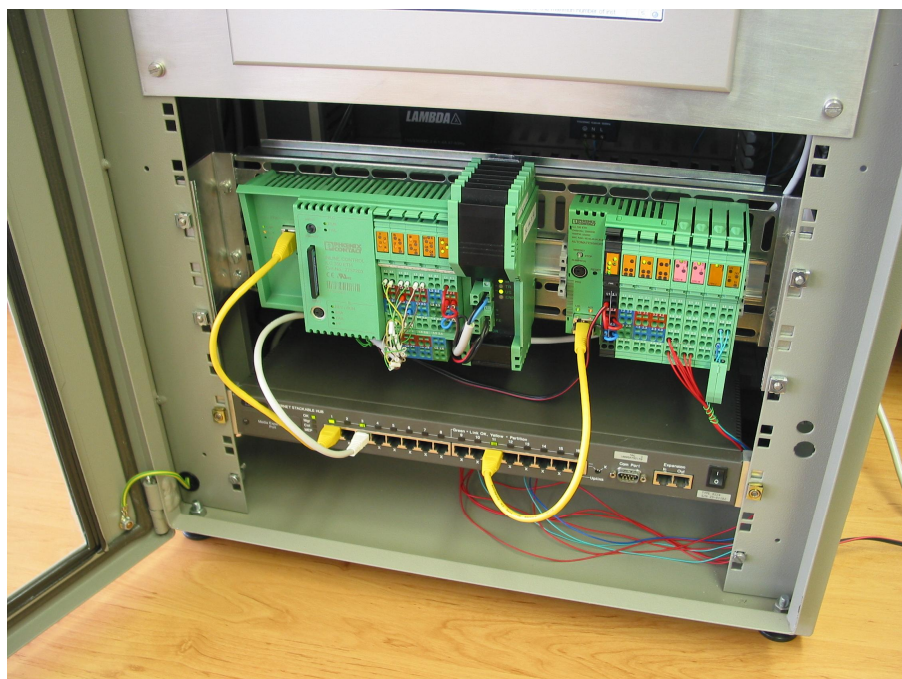
Parametr NDR monitoruje zda byla všechna odeslaná data úspěšně přijata.

Název	Datový typ	Směr dat	Popis
EN_R	BOOL	Vstup	Tento vstup je použit pro přípravu na přijetí EN_R = TRUE Přijímaná data jsou dostupná EN_R = FALSE Přijímaná data nejsou dostupná
ID	INT	Vstup	Během navázání spojení používá „IP_CONNECT“, jedinečné číslo ID je přiděleno každé komunikaci spojení. Toto ID informuje blok „IP_URCV“ o komunikujících partnerech od něhož jsou data přijímaná Vstup „ID“ může být přímo spjat s výstupem „ID“ bloku „IP_CONNECT“
NDR	BOOL	VSTUP	Tento výstup ukazuje zda blok úspěšně přijal všechna odeslaná data. Úspěšnost přenosu může být ohodnocena během probíhajícího náběžné hrany.
RD_1	ANY	Vstup/Výstup	Velikost a název proměnné „RD_1“ nesmí být změněna během používání. Jestliže byla zpráva úspěšně přijata, NDR je nastaven na hodnotu TRUE na dobu jednoho cyklu. Během tohoto času, mohou být data RD_1 kopírovány pro další zpracování. Jestliže je NDR=FALSE, jsou hodnoty RD1 neplatné.

Stručný popis použitých parametrů FB IP_URCV.

4.3.4 IP_TIME

Pomocí FB IP_TIME lze synchronizovat systém času a přenosu dat. Slouží k ukončení spojení po předem stanoveném časovém intervalu. V úloze však nebyl tento blok použit.



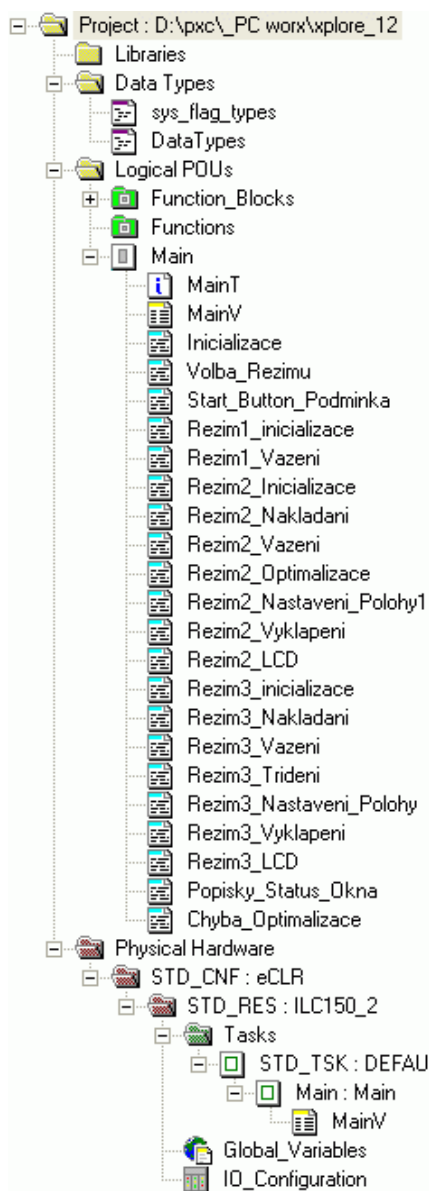
OBR 4.1: Hardwarové propojení obou PLC. Řídící PLC je na obrázku umístěn vpravo.

5 Podrobný popis režimů

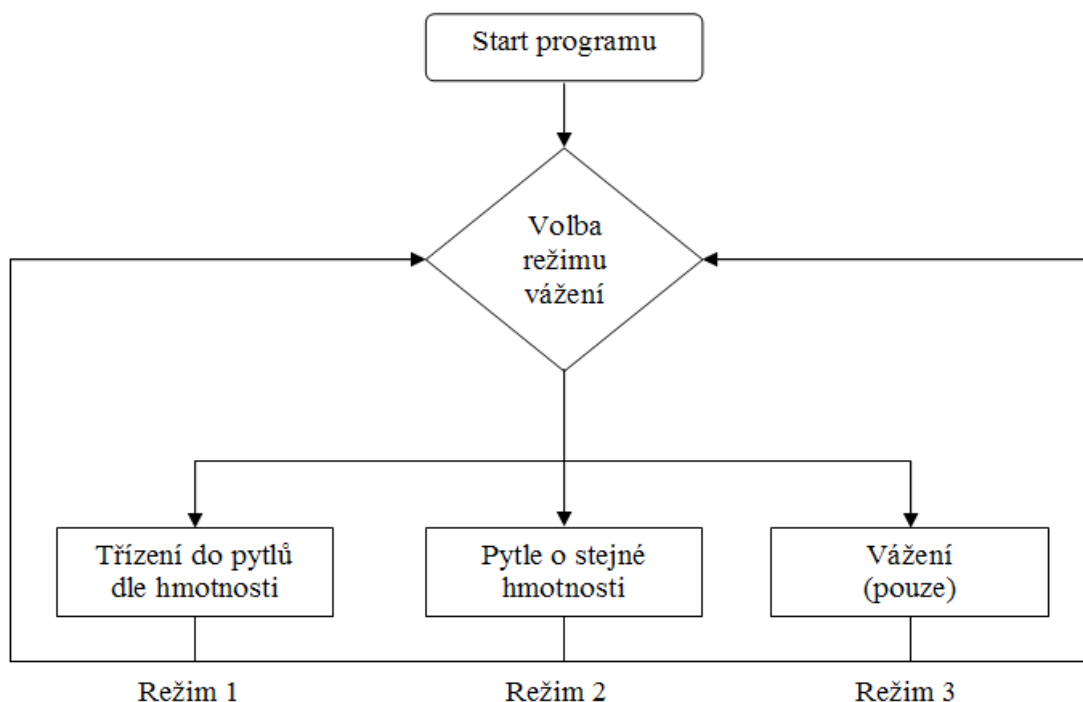
5.1 Struktura programu a názvů proměnných

Řídící program PLC je strukturován tak, že všechny činnosti programu běží v MAIN (čili v hlavním cyklu) Formou pod úloh. Pod úlohy jsou realizovány pomocí podmínek pracujících s proměnnou iKrok. Proměnná iKrok tedy jistým způsobem definuje v jaké fázi se MAIN nachází, resp. která část pod úlohy právě probíhá.

V této kapitole budou podrobně popsány všechny části MAIN a při popisování konkrétní pod úlohy bude zmíněn i stav proměnné iKrok. Na **OBR 5.5.** je zobrazena struktura programu a pomyslná návaznost všech pod úloh.



OBR 5.5: Struktura programu v PLC.



OBR 5.1: Blokové schéma vývojového diagramu inteligentního vážení a třídění zboží.

Proměnné které jsou v programu použity jsou deklarovány takovým způsobem, aby bylo na první pohled jednoznačné jakým datovým typem disponují. Datový typ proměnné symbolizuje malé písmeno vložené před název proměnné. Celý mechanismus je pochopitelný na příkladu. Například proměnná „iCislo“ má název „Cislo“ a zároveň je jednoznačně určen datový typ Integer. Analogicky proměnné s datovými typy Word a Byte mají před svým názvem malé písmeno „w“ a „b“ nebo „x“ atp. (písmeno „x“ symbolizuje dva možné stavy datového typu byte).

5.2 Jednotlivé režimy slovně

Celý systém řešení se skládá ze tří režimů činnosti. Z hlediska programu jsou režimy spuštěny podle stavu v jakém se nachází proměnná iRezim. Proměnná nabývá tří hodnot a pracuje s podmínkou. Vybraný režim je aktivní do té doby, dokud není zvolen režim jiný, nebo dokud není aktivní režim ukončen. Ukončení je provedeno nastavením proměnné iRezim do hodnoty 0.

Před samotným zvolením režimů je nutné aby program provedl některé nutné operace, jež přímo nesouvisí s pracovními režimy, ale jsou důležité z hlediska funkčnosti celého programu. Jsou to Inicializace, Volba režimu, Start button podmínka, popisky status okna a chyba optimalizace. Všechny operace budou popsány níže.

Inicializace

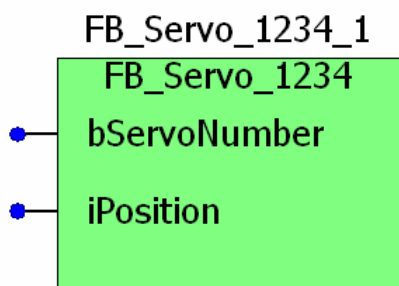
Do této pod úlohy je vcházeno po každé změně režimu a při odchodu z jakéhokoli režimu. Pokud není program právě ve zvoleném režimu (pod úloze), cykluje

ve smyčce mezi inicializací a volbu režimu. V tomto cyklu program čeká na signál obsluhy pro výběr režimu.

Na začátku inicializace je hodnota proměnné iKrok=0 a po dokončení všech podmínek inicializace je iKrok=1. V inicializaci dochází k nastavení komunikace pomocí FB IP_CONNECT a FB SEND které byly podrobně popsány v kapitole 3. Dále je nutné smazat aktuální hodnoty proměnných iOptKontejner, iNamerenaHmotnost, iHmotnostZboziContainer1 až 4 xCheckFullContainer1 až 4 a xFullContainer1 až 4.

Proměnná iOptKontejner plní funkci ukazatele na kontejner který bude plněn v následujícím kroku. iNamerenaHmotnost je proměnná do které je vkládána hmotnost váženého zboží umístěného na váze, přičemž hodnota je získána pomocí FB SEND. Zbylé čtveřice proměnných slouží k určení, do jaké míry jsou kontejnery zaplněny. Konkrétně xCheckFullContainer a xFullContainer značí který kontejner je plný, resp. zda potřebuje vyprázdnit (dvě různé čtveřice pro dva odlišné režimy třídění). Proměnné iNamerenaHmotnost1 až 4 obsahují aktuální hodnoty hmotností všech čtyř kontejnerů.

Po resetování zmíněných hodnot je provedeno nastavení polohy všech servo-pohonů do nulté polohy. Tato činnost je realizována pomocí FB Servo_1234. Blok je vytvořen pro usnadnění práce se servo-pohony.



OBR 5.2: Zobrazení FB Servo_1234 v jazyce reléových schémat.

FB Servo_1234

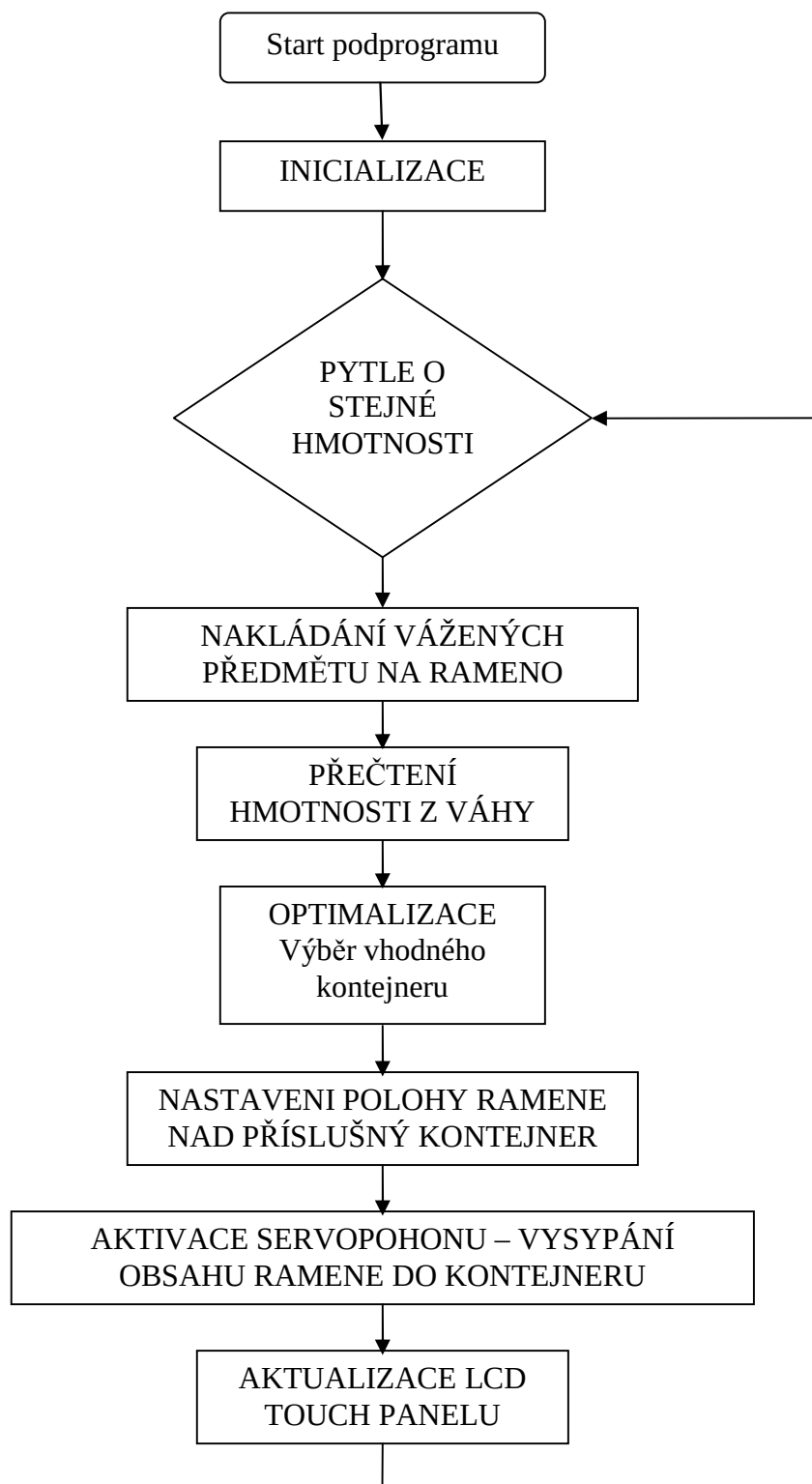
Jedná se o FB který je vytvořen pro usnadnění práce se servo-pohony. FB je zobrazen na **OBR 5.2**. Pro obsluhu servo-pohonů je využíván rozšiřující modul PLC – IB IL PWM/2 a IBL IL 24/48 DOR. Funkčně by pro řízení serva stačil modul IB IL PWM/2, má ovšem pouze dva výstupy jenž nepokryjí činnost 5-ti servo-pohonů. Modul IB IL PWM/2 tedy ovládá nakládací servo a čtveřici všech vykládacích servo-pohonů. Výběr aktivního serva ze zbylé čtveřice pro vykládání, je realizován modulem IBL IL 24/48 DOR. Modul pracuje jako reléový výstup a při výběru konkrétně zvoleného serva (nebo serv) vypne napájení ostatních nepotřebných servo-pohonů.

Pohyb serv je uskutečněn odesláním čísla v datovém typu WORD přímo na vstup IB IL PWM/2. Číslo obsahuje hodnotu délky impulzu, podle níž se vybrané servo natočí do polohy kterou definuje frekvence (délka impulzu). Servo-pohonům a jejich funkci se podrobně věnuje kapitola 6. Poté co servo-pohon vyhodnotí přijímanou frekvenci, natočí se do požadované polohy.

Pro nastavení frekvence serva je použita proměnná iPosition, která pomocí 21 stavů podmínky nastaví potřebnou frekvenci. Stavů podmínky, jsou rozpočítány po procentech mezi délkami pulzů 0,9ms až 2,1ms, s nimiž servo pracuje, právě na zmíněných 21 stavů. Na vstup iPosition bloku Servo_1234 tedy odešlu číslo 1 až 27 a

tím nastavím o jakou hodnotu se serva pootočí. Hodnoty vypočtených frekvencí jsou přenášeny pomocí proměnné „dwFreqPWM.W0“ která přímo pracuje se vstupy modulu IBL IL 24/48 DOR. (vstup iPosition je zobrazen na **OBR 5.2**)

To, který pohon je aktivní, je řešeno proměnnou bServoNumber (viz funkce modulu IBL IL 24/48 DOR). Proměnná bServoNumber nabývá čtyř stavů, podle toho jaké servo bude aktivováno. Serva jsou aktivní podle dekády binárního čísla. První servo bude aktivní na hodnotu 1 proměnné bServoNumber, druhé reaguje na hodnotu 2, třetí na hodnotu 4 a čtvrté na hodnotu 8. Pokud chci např.abý bylo aktivní servo 1 a 4 pošlu do proměnné bServoNumber číslo 9. (vstup bServoNumber je zobrazen na **OBR 5.2**)



OBR 5.3: Blokové schéma vážení s optimalizací (režim 1). Cílem je požadavek mít kontejnery o stejné požadované hmotnosti.

5.2.1 Vážení s optimalizací (režim 1)

Pod úloha vážení s optimalizací vychází z vývojového diagramu zobrazeného na **OBR 5.3**.

Jedná se o režim činnosti, ve kterém je hlavním pracovním prvkem FB sloužící k optimalizaci výběru kontejnerů. FB je umístěn na vhodném místě v programu. Princip optimalizace je takový, že prostřednictvím hmotnosti zboží na váze a obsahu v kontejnerech vybere ten kontejner, do kterého bude vážené zboží vloženo.

Kontejnery jsou plněny na předepsanou hmotnost, která je nastavitelná obsluhou. Cílem je snaha o to, aby se výsledné hmotnosti kontejnerů nelišily od předepsané hmotnosti. Předepsaná hmotnost má nastavitelné tolerance. Celý režim je možno ovládat klávesami START a STOP na dotykové obrazovce. Kontejnery jsou čtyři z nichž tři slouží pro ukládání zboží pro předepsanou hmotnost a čtvrtý slouží pro mimořádnostní zboží. Optimalizace je lépe popsána ve vlastní kapitole.

Zvolení režimu se provádí v pod úloze, která je nazvána „volba_rezimu“. V této části je proměnná iKrok=1 a pokud není zvolen žádný režim je hodnota iKrok přepsána na hodnotu 0 a program se vrací do inicializace.

Při zvolení režimu optimalizace je hodnota iKrok=20 a je spuštěna inicializace prvního režimu.

Inicializace režimu 1

Po zvolení prvního režimu a ještě před samotným nakládáním je provedena inicializace režimu. Jedná se o nastavení počátečních podmínek pro podprogram optimalizace. Počátečními podmínkami jsou: vymazání paměti poslední naměřené hmotnosti (iNamerenaHmotnost), nastavení servo-pohonů do počáteční polohy a nastavení počáteční polohy ramene. Nastavení počáteční polohy je realizováno odesláním příkazu a dat pomocí FB SEND. Jako příkaz je odesláno číslo 2 proměnné wWriteCommand a jako hodnota dat je odesláno číslo 5 proměnné wWriteVakue. Pokud je vážící rameno ve správné poloze, vážící PLC pošle zpět stejné hodnoty, program splní podmínku a může pokračovat na další krok. Po splnění podmínek této části programu je proměnná iKrok=21.

Nakládání

Nakládání je realizováno pomocí prvního servo-pohonu určeného pro nakládání a je spuštěno jakmile je hodnota iKrok=21. Podmínka nakládání je spjata s polohou ramene, ta je získána od vážícího PLC pomocí komunikace. Celý proces nakládání má časový limit 2 sekundy, do té doby je vážené zboží bez problémů naloženo na rameno váhy (nějakou dobu trvá než nakládací mechanismus naloží zboží na rameno).

Pro pohyb tunakládacího serva není využit FB Servo_1234, ale je přímo odeslána hodnota polohy pomocí proměnné dwFreqPWM.W1 která, podobně jako ve FB Servo_1234 obsahuje hodnoty vypočtených frekvencí. Po splnění všech podmínek se proměnná iKrok nastaví na hodnotu 22.

Vážení

Po dokončení nakládání (proměnná iKrok má hodnotu 22) je spuštěno vážení. Vážení proběhne jakmile vážící PLC přijme požadavek na vážení od řídicího PLC. To je realizováno pomocí FB SEND s nastavenými parametry wWriteCommand=1 a

wWriteValue=0. Poté co vážící PLC zváží zboží umístěné na rameni váhy, odešle tuto hmotnost řídicímu PLC v podobě čísla. Řídicí PLC přijme data opět pomocí FB SEND.

Při přijímání dat jsou využity parametry wReadCommand a wReadValue. Parametr wReadCommand má při přijímání dat stejnou hodnotu jako wWriteCommand, ale parametr wReadValue musí být oproti parametru wWriteValue odlišný, tedy $\neq 0$. Pokud je hmotnost nulová (na rameni nebylo v době vážení žádné zboží), program se vrátí zpět na inicializaci nastavením hodnoty iKrok=20. Pokud vše proběhne v pořádku, je do proměnné wNamerenaHmotnost nastavena hodnota umístěná v proměnné wReadCommand (hodnota hmotnosti váženého zboží). Program pokračuje splněním všech podmínek a nastavením hodnoty iKrok na hodnotu 23 do dalšího kroku (pod úlohy).

Optimalizace

Jakmile je vážení ukončeno následuje spuštění pod úlohy s FB obsahujícím algoritmus optimalizace. Při spuštění optimalizace musí mít proměnná iKrok hodnotu 23. V této části programu je zavolán pouze FB Optimalizace jehož výstupním parametrem je i „iContainer“. Hodnota parametru iContainer je umístěna do proměnné iOptKontejner, která je právě ukazatelem na kontejner, který bude v příštím kroku plněn. Po té co proměnná iOptKontejner získá hodnotu parametru iContainer, je hodnota iKrok přepsána na hodnotu 24 a část programu optimalizace je ukončena. Pro důkladnější pochopení optimalizace, bude nyní popsán FB OPTIMALIZACE.

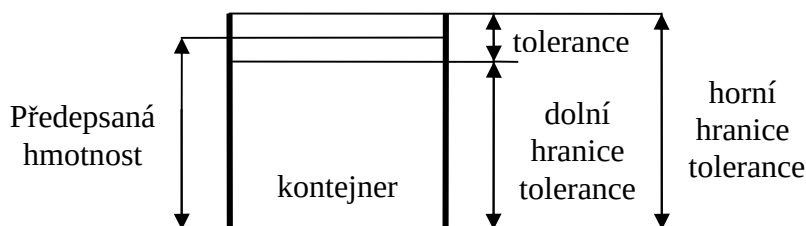
FB OPTIMALIZACE

Optimalizace pracuje způsobem porovnávání hmotnosti právě váženého zboží s hmotnostmi v kontejnerech, resp. tou hmotností kterou lze do vybraného kontejneru ještě přidat. Pakliže je hmotnost váženého zboží menší než hmotnost která v kontejneru zbývá, bude do tohoto kontejneru zboží vloženo. Zároveň je primárně vybrán nejvíce plný (hmotný) kontejner. To zda je kontejner plný je dáno jeho horní hranicí tolerance, což je předepsaná hmotnost kontejneru + tolerance (viz. **OBR 5.4**).

Pokud je vážené zboží větší než volné místo v kontejneru, je hmotnost porovnána s dalším (druhým nejvíce hmotným) kontejnerem. Pokud již celková hmotnost kontejneru přesáhne dolní hranici tolerance hmotnosti, je kontejner označen za plný a je nutná jeho výměna. Práce celého zařízení pokračuje dokud nejsou plné všechny kontejnery (tedy 3 ze 4).

Výše zmíněný algoritmus pracuje se třemi ze čtyř kontejnerů. Čtvrtý kontejner je určen pro zboží, které je mimo hmotnostní jakost (parametry jakosti jsou nastavitelné). Čtvrtý kontejner se společně s tolerancí (**OBR 5.4**) starají o předcházení hazardního stavu.

Hazardní stav by mohl nastat například v případě, že by všechny kontejnery čekaly na zboží s nesmyslně malou hmotností a úloha by nemohla být ukončena (Nesmyslně malé hodnotě je přiřazena nižší jakost – jedno z možných východisek).



OBR 5.4: Systém plnění kontejneru s tolerancí.

FB OPTIMALIZACE má 5 výstupních a 5 vstupních parametrů. Vstupními parametry jsou `iWeight`, `iContainerTolerantion`, `iRequiredWeight`, `iZboziMin`, `iZboziMax`. Parametry `iContainerTolerantion`, `iRequiredWeight`, `iZboziMin`, `iZboziMax` jsou nastavitelné obsluhou. Do parametru `iWeight` je vložena hodnota `wNamerenaHmotnost` a s touto hodnotou jsou porovnávány hmotnosti kontejnerů. V parametru `iContainerTolerantion` je vložena hodnota `tolerance`.

Velikost všech kontejnerů je jednotná a hodnota velikosti je zadána pomocí parametru `iRequiredWeight`, na **OBR 5.4** je značen jako „Předepsaná hmotnost“. Parametry `iZboziMin` a `iZboziMax` slouží pro vymezení jakosti výrobku. Výrobek bude mít správnou jakost, pokud bude mít hodnotu spadající mezi `iZboziMin` a `iZboziMax`.

Výstupní parametry jsou `iContainer`, již zmíněn v úvodu a `xIsFullContainer1` až 4. Tyto parametry značí který kontejner je plný, tedy který kontejner již svou hmotností překročil dolní hranici tolerance.

Část kódu FB OPTIMALIZACE pro pochopení jeho funkce

FB OPTIMALIZACE má všechny proměnné lokální, pouze proměnné `iHmotnostZboziContainer1` až 4 jsou ve všech FB globální. Je to z důvodu jednoduchosti, aby FB „viděl“ aktuální hmotnost všech kontejnerů. V první fázi je zjištěno který kontejner je nejvíce zaplněn.

Ukázkové Podmínky jsou stejné pro všechny kontejnery, liší se jen indexem.

```
IF iHmotnostZboziContainer1 >= (iRequiredWeight - iContainerTolerantion)
    THEN
        xIsFullContainer1:=TRUE;    */značí že je kontejner plný.
    ELSE
        xIsFullContainer1:=FALSE;   */značí že je kontejner prázdný.
END_IF;
```

Dále je zjištěno do kterého kontejneru se naposledy zváženy předmět vleze.

```
IF ((iHmotnostZboziContainer1 + iWeight) <= (iRequiredWeight + iContainerTolerantion)) and
xIsFullContainer1 = FALSE
    THE
        xVlezeDoContainer1 := TRUE;    */ Proměnná značí zda je v kontejneru dostatek místa.
    ELSE
        xVlezeDoContainer1 := FALSE;
END_IF;
```

Vlastní výběr kontejneru který bude v příštím kroku plněn.

```

IF (xPomocna and xVlezeDoContainer1 and
  ((iHmotnostZboziContainer1 >= iHmotnostZboziContainer2) or (NOT xVlezeDoContainer2)) and
  ((iHmotnostZboziContainer1 >= iHmotnostZboziContainer3) or (NOT xVlezeDoContainer3)))
THEN
  iNejobsazenejsiContainer := 1;  */Přímo číslo plněného kontejneru (pro druhý 2 atd.)
  xPomocna := False;  */pomocná proměnná, slouží k tomu, aby se vybral právě jeden
END_IF;                  kontejner

```

Čtvrtý kontejner, jelikož je plněn mimo jakostním zbožím, je v podmínce porovnáván pouze s proměnnou xVlezeDoContainer4 a s hodnotami iZboziMin a iZboziMax.

Pokud nastane takový stav že by se vážené zboží nemohlo vložit do žádného kontejneru, např. pokud jsou všechny kontejnery již plně obsazené, bude hodnota iNejobsazenejsiKontainer nastavena na 0. Takto se obsluha dozví o tom že nastal nevhodný stav.

Nastavení polohy

Dalším krokem po optimalizaci je nastavení polohy ramene. Proměnná iKrok má při spuštění této pod úlohy hodnotu 24. Při nastavování polohy využijeme hodnotu ukazatele na kontejner, nebo-li hodnotu proměnné iOptKontejner. Pokud má proměnná iOptKontejner hodnotu různou od nuly (vše je v pořádku), je odeslán příkaz wWriteCommand=2 a hodnota wWriteValue je shodná s hodnotou proměnné iOptKontejner. Jakmile Rameno dosáhne požadované polohy, odešle vážící PLC nazpět data stejných hodnot, jako bylo odeslána. Rameno má vyklápěcí polohy nastaveny tak, aby se zastavilo nad příslušným kontejnerem. Po splnění podmínek se proměnná iKrok přepíše na hodnotu 25 a program pokračuje do další podmínky.

Pokud je proměnná iOptKontejner=0 (např. jsou již plně všechny kontejnery), je proměnná iKrok nastavena na hodnotu 40 a spustí se pod úloha „Chyba_optimalizace“. V této části programu je proces zastaven nastavením xStartPossible do hodnoty FALSE (Proměnná bude popsána později) a hodnoty iRezim a iKrok jsou nastaveny na hodnotu 0 (běh režimu je násilně ukončen). Program od této doby čeká na nové zvolení režimu. (obsluha je o tomto stavu informována na status displeji a vyzvána k odchodu do hlavního menu).

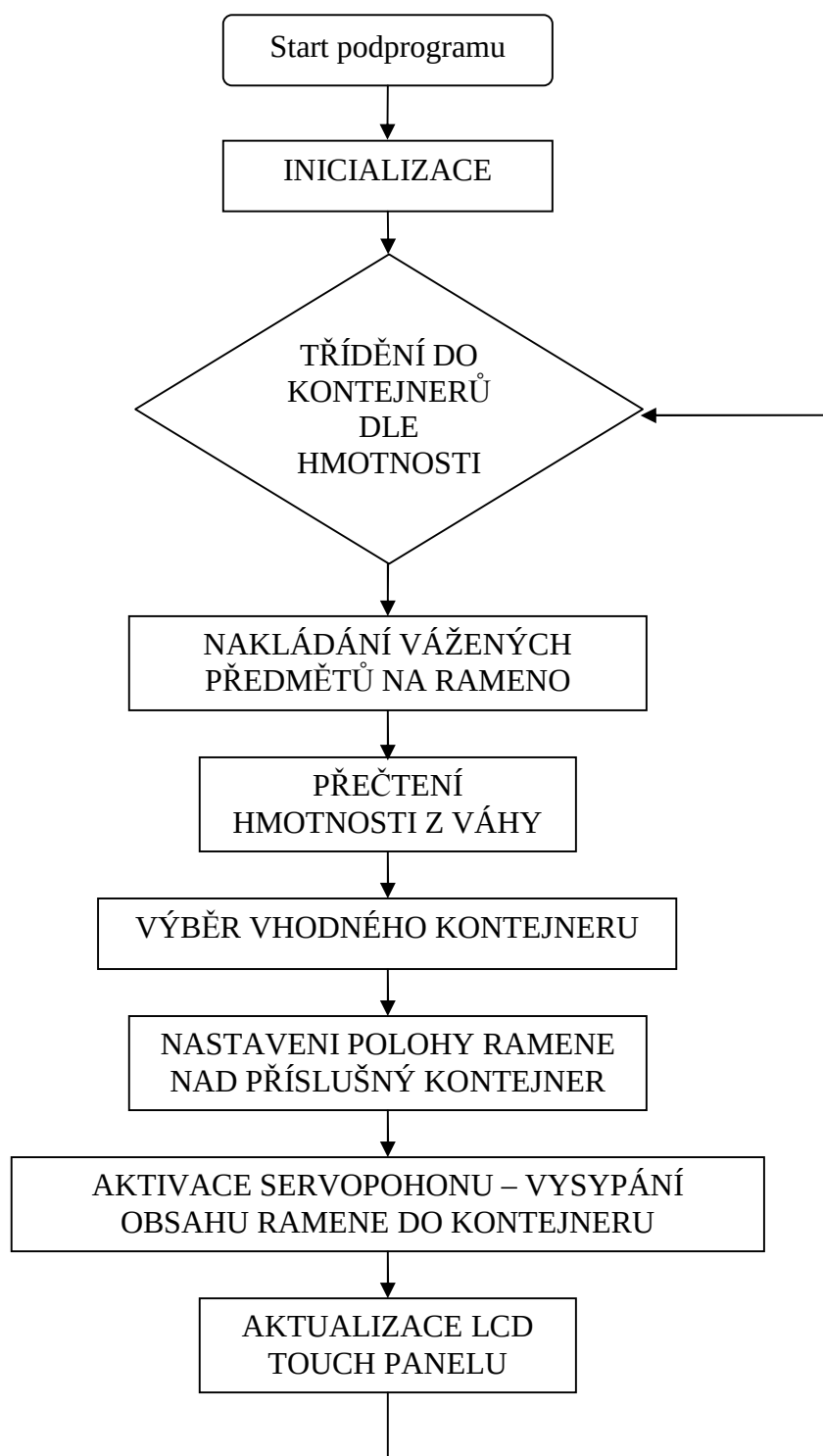
Vyklápění

Vyklápění pro svou činnost využívá zbylé čtyři servo-pohony. Pro spuštění pod úlohy vyklápění musí mít proměnná iKrok hodnotu 25. V pod úloze je nejdříve spuštěn časovač, který je nastaven na hodnotu 2 sekund. Je to výsledovaný čas potřebný pro ustálení ramene nad kontejnerem do kterého bude zvážené zboží umístěno. Poté je aktivováno servo, umístěné nad kontejnerem. Pro výběr serva je využito podmínky pracující s proměnnou iOptKontejner. Na základě ukazatele je tedy vybrán i servo-pohon. Jakmile je servo vybráno, je spuštěn FB SERVO_1234. Servo-pohon se nejdříve otočí o maximální hodnotu od klidového stavu (proměnná iPosition=21). Po dokončení manipulace se servem (za cca 1 sek.), je spuštěn opět FB SERVO_1234, ale nyní s hodnotou proměnné iPosition=0, tedy servo se vrátí zpět do nulové hodnoty. Po obou

úkonech je opět spuštěn časovač s dobou 2 sek. Po dokončení práce časovače nastavení hodnoty iKrok na hodnotu 26.

Konec

Po ukončení vyklápění se aktualizují hodnoty hmotností kontejnerů spuštěním poslední podmínky režimu optimalizace. Pro její spuštění musí být proměnná iKrok=26. Pro aktualizace hodnot kontejnerů je využita podmínka proměnné iOptKontejner. Pomocí iOptKontejner je vybrán kontejner a k jeho aktuálnímu stavu je přičtena hodnota váhy zboží které bylo zváženo, tedy hodnota proměnné wNamerenaHmotnost. Po dokončení program přejde na začátek cyklu a režim je poté opakován. Opakování je provedeno nastavením proměnné iKrok=20. Režim lze rovněž kdykoli ukončit volbou ZPĚT na dotykové obrazovce. Rovněž lze ovlivňovat chod režimu pomocí tlačítek STOP a START. Část programu starající se o možnost zastavení a znovu spuštění právě probíhajícího režimu, je popsán níže.



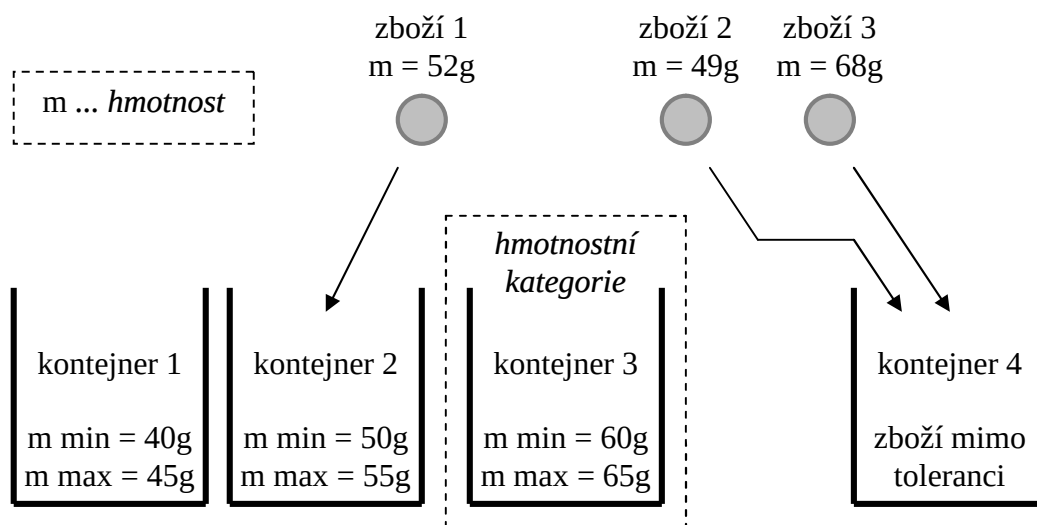
OBR 5.5: Blokové schéma vážení s tříděním dle hmotnosti (režim 2). Cílem je třídit zboží stejné hmotnosti.

5.2.2 Vážení s tříděním dle hmotnosti (režim 2)

Režim pro vážení s tříděním dle hmotnosti využívá do značné míry zdrojový kód předchozího režimu. Rozdíly jsou v hodnotách proměnné iKrok, která má hodnoty o 10 jednotek vyšší (k aktivaci režimu dojde při hodnotě iKrok=30). O jeho odlišnou funkci se stará FB TRIDENI umístěný na místě kde byl v předchozím režimu umístěn FB OPTIMALIZACE. Z výše uvedeného důvodu nebude znovu popsána funkce všech jednotlivých částí režimu, které jsou pro tento případ téměř shodné, ale pouze činnost FB TRIDENI.

Funkcí režimu pro třídění dle hmotnosti je členění váženého zboží do hmotnostních kategorií. Každá kategorie má obsluhou nastavitelné parametry. Jsou to: nastavení velikosti tolerance hmotnostních kategorií a velikosti hmotností kterou kategorie pojmu. Kategorií je v našem případě myšlen kontejner, který je pro jednu hmotnostní kategorii vyhrazen. Režim má tolik kategorií, kolik má k dispozici kontejnerů, resp. n-1 kontejnerů, viz níže.

Kategorie jsou tři plus jedna, která je určena pro zboží které nelze přidělit žádné z předchozích tří kategorií. Funkce režimu je zobrazena na **OBR 5.6**. Pokud hmotnost zboží spadá do mezí tolerance hmotnostní kategorie, je do této kategorie (kontejneru) zboží vloženo. Pokud zboží do žádné ze tří kategorií nepatří, je vloženo do čtvrtého kontejneru k tomu účelu určenému.



OBR 5.6: Na obrázku je jasně patrné, jakým způsobem režim třídí zboží.

FB TRIDENI

FB TRIDENI je značně jednodušší než FB OPTIMALIZACE. Blok má 12 vstupů a 5 výstupů. Vstupy jsou iMinimum_Kontejneru_1 až 3, iMaximum_Kontejneru_1 až 3, což jsou parametry pro vymezení tolerancí. Další vstupy jsou iKapacita_Kontejneru_1 až 4, které slouží pro vymezení hmotnostního obsahu kontejnerů. Do parametru iWeight je vložena naměřená hmotnost zboží a slouží, podobně jako v FB OPTIMALIZACE, k porovnávání aktuálních hmotností kontejnerů. Poslední vstup je iTrideni_Tolerance, tento parametr má funkci tolerance. Všechny parametry jsou nastavitelné obsluhou, jak

již nastínil fakt že se jedná o vstupy. Tolerance se jako jediný parametr nastavuje pro všechny kontejnery stejně.

Výstupy jsou, obdobně jako u FB OPTIMALIZACE, parametry xFullContainer1 až 4 a iContainer. Parametr iContainer má funkci ukazatele na kontejner který se bude plnit a parametry xFullContainer1 až 4 značí zda a které kontejnery jsou již plně obsazeny.

Při „zavolání“ FB je napřed zjištěno, zda a jaké kontejnery jsou již obsazeny a poté algoritmus porovná hmotnost zboží na váze s tolerancemi kontejnerů a jejich skutečnou hmotností. Nejvhodnější demonstrací funkce je ukázka části kódu. Část kódu je podobná pro všechny kontejnery, liší se pouze indexem u proměnných iHmotnostZboziContainer, iKapacita_Kontejneru a xFull_1.

```

IF (iHmotnostZboziContainer1 >= (iKapacita_Kontejneru_1 - iTrideni_Tolerance))
    THEN
        xFull_1 := TRUE;           */Pokud je xFULL=TRUE, kontejner je plný.
    ELSE
        xFull_1 := FALSE;
END_IF;

```

Po zjištění zda je možno do kontejnerů stále přidávat zboží je proveden vlastní výběr kontejneru. Část kódu je podobná, nyní jen pro tři kontejnery, jelikož čtvrtý kontejner je vyhrazen pro „nevyhovující“ zboží a obsahuje jiné podmínky.

```

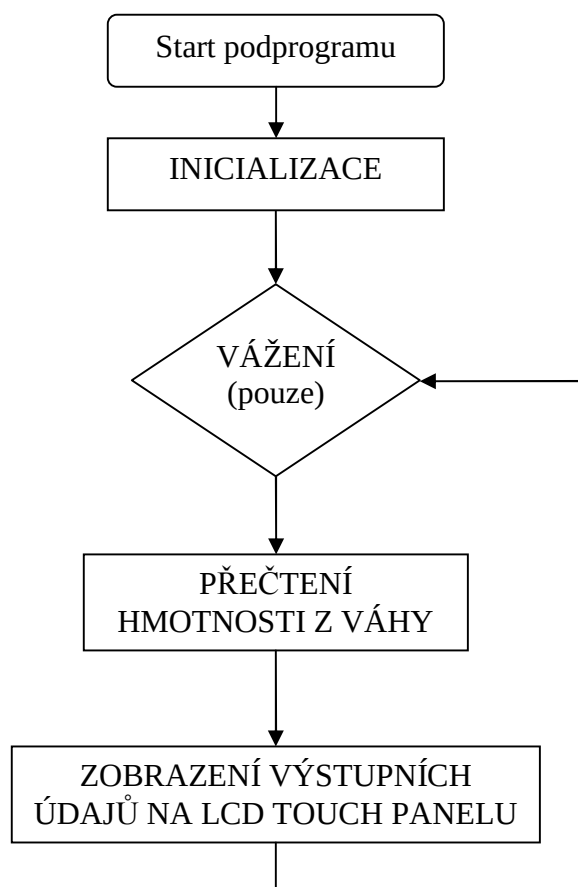
IF (iWeight >= iMinimum_Kontejneru_1) and (iWeight <= iMaximum_Kontejneru_1) and
    (iHmotnostZboziContainer1 < (iKapacita_Kontejneru_1 + iTrideni_Tolerance)) and
    (xFull_1 = FALSE) and (xPomocna = FALSE)
    THEN
        xPomocna := TRUE;
        iContainer := 1;
END_IF;

```

Proměnná xPomocna zaručuje to, že se provede pouze jedna ze všech čtyř podmínek (pokud je již vybrána jedna podmínka, jiné se neprovedou).

Čtvrtý kontejner se v zásadě liší jen v tom, že místo porovnávání zbylého místa a mezí iMinimu_Kontejneru a iMaximum_Kontejneru, porovnává minima a maxima všech kontejnerů a své zbylé místo.

V případě že jsou všechny kontejnery plné, nebo je vstupní hmotnost nulová, FB vrátí hodnotu nula, jako v případě FB OPTIMALIZACE.



OBR 5.7: Blokové schéma vážení bez optimalizace (režim 3). Cílem je pouze vážení zboží, bez nakládání ze zásobníku a vykládání do kontejnerů.

5.2.3 Vážení bez optimalizace (režim 3)

Tento režim byl do zařízení implementován hlavně z důvodu demonstrace funkce váhy, resp. zařízení pracujícího s vážícím PLC. Vlastnost režimu již není definována pouze pomocí FB, ani nejsou využity pod úlohy programu, jako je tomu u předchozích dvou režimů. Využívá pouze lehce zmodifikovaných pod úloh pro inicializaci a vážení, použitých z těla programu prvního režimu.

Po přepnutí do tohoto režimu je obsluha zařízení vyzvána, aby umístila předmět na rameno váhy (libovolný objekt, který chceme zvážit). Po stisku start je těleso zváženo a vážící proces se zastaví. Celý proces vážení lze opakovat.

Jediná zásadní změna oproti původním částem programu je přepínání proměnné xStart z hodnoty TRUE do hodnoty FALSE při inicializaci (iKrok=10). Je to z důvodu aby se vážení provedlo pouze jedenkrát a nové vážení bylo opět odstartováno až po stisku START.

5.2.4 Ostatní části programu

Kromě částí programu pro režim optimalizace, celkové inicializace a řešení pro zbylé dva režimy, je v části MAIN ještě několik částí programu, které budou nyní popsány.

Volba režimu – detailní popis

Do pod úlohy VOLBA REŽIMU se běžící program dostane, jakmile je proměnná iKrok = 1 a je to část programu, která se pravidelně spouští po inicializaci. Při spuštění volby režimu jsou nastaveny proměnných xStartPossible a xStart na hodnotu FALSE. Proměnné a jejich závislost na běh celého programu bude popsán v části START_BUTTON_PODMÍNKA.

Volba režimu je provedena pomocí podmínky iRezim, která nabývá hodnot 1 až 3 a po jejich volbě je každé hodnotě proměnné iRezim přidělena hodnota iKrok o deset vyšší. Tzn. pro režimy 1, 2 a 3 jsou hodnoty proměnné iKrok=10, 20 a 30.

START_BUTTON_PODMÍNKA

Tato podmínka umožňuje spouštění, případně přerušení činnosti právě probíhajícího režimu. Hlavními prvky, které se starají o tuto funkci, jsou proměnné xStartPossible a xStart. Proměnná xStart je umístěna na začátcích podmínek všech tří režimů. Proměnná xStartPossible je pomocnou proměnnou, aby nešlo proměnnou xStart nastavit na hodnotu TRUE (aktivovat), pokud by obsluha nastavila nevhodné hodnoty. Zamezí se tak například tomu, aby nebylo možné režim spustit s nesmyslnými hodnotami, které nastavila obsluha.

Při nastavení proměnné xStartPossible=TRUE a po splnění výše zmíněných podmínek je i hodnota xStart přepnuta do hodnoty TRUE. Režimy čekající na splnění proměnné xStart jsou nyní spuštěny do doby, dokud není proměnná xStart=FALSE, tzn. dokud není stisknuta klávesa STOP.

Popisky status okna

Je pod úloha, ve které je formou podmínek se stavem proměnné iKrok přiřazován textový význam všem krokům MAIN. Dochází zde tedy k tomu, že se všem pod úlohám přiřadí textový význam, který je zobrazován ve status displeji na dotekové obrazovce. Displej vede obsluhu zařízení jednotlivými kroky, případně o tom, ve které fázi se program či režim nachází. Texty jsou ukládány do hodnoty sText, datového typu STRING.

6 Servo-pohon HS-475 HB

Servo vyžaduje stálý střídavý obdelníkový signál o napětí 3-5 V mezi náběžnými hranami. Délky pulsů jsou od 0,9ms do 2,1ms, 1,5ms je střed. Obnova signálu je 50 Hz (20ms)

Pracovní režim serva je v rozmezí 4,8V až 6V. Při tomto napětí je servo schopno vyvinout sílu 4,4 kg/cm (44Ncm) při 4,8V a 5,5 kg/cm (55Ncm) při 6V. S rychlostí posuvu 0,23s/60° (4,8V) a 0,18/60° (6V). Cena serva je okolo 500kč v roce 2008.

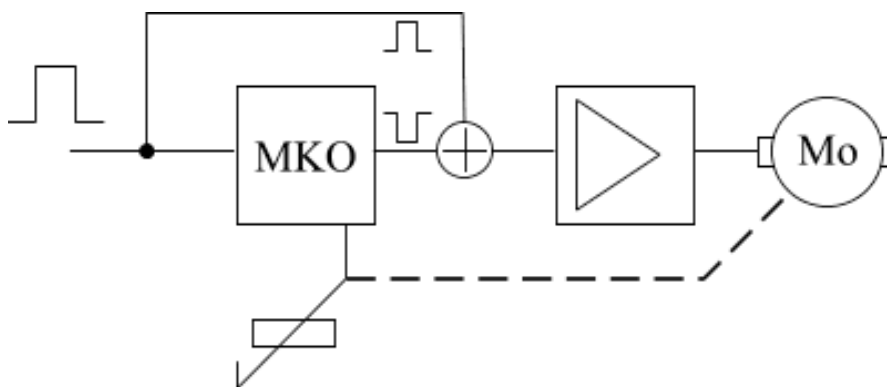


OBR.:6.1: servo-pohon HS-475HB s ukázkou převodů.[2]

Jak pracuje servo

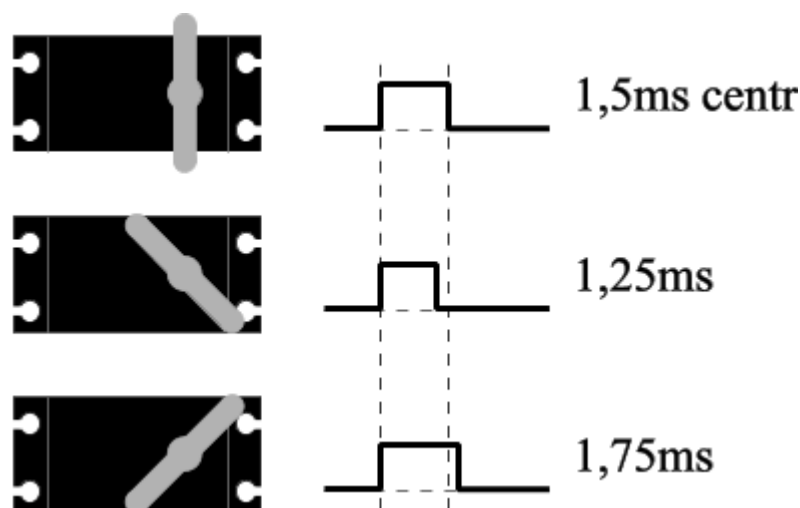
Tato kapitola vychází z [3].

Dnešní modelářská serva obsahují elektromotor s převodovkou a řídicí elektroniku. Zjednodušené zapojení elektroniky je na blokovém schématu (**OBR 6.2**). Do vstupu přichází řídicí impuls, který spustí monostabilní klopný obvod, ten vygeneruje impuls o délce odpovídající momentální poloze serva a opačné polarity než je vstupní řídicí impuls. Tyto dva impulsy se porovnají a výsledkem je rozdílový impuls, který po zesílení přes můstkový spínač způsobí roztočení elektromotoru jedním nebo druhým směrem. Elektromotor přes převodovku otáčí výstupní hřídeli a současně i potenciometrem, který působí jako zpětná vazba do monostabilního klopného obvodu. Směr otáčení je takový, že impuls generovaný monostabilním klopným obvodem se svojí délkou přibližuje délce vstupního řídicího impulsu a až jsou oba impulsy stejně dlouhé, elektromotor se zastaví. Servo dosáhlo polohu, která odpovídá momentálně přijímanému řídicímu impulsu.



OBR 6.2: Blokové schéma zapojení motoru a monostabilního klopného obvodu.[3]

Dnes používaná serva pracují s kladnými řídicími impulsy o délce 1-2ms. Délce impulsu 1,5ms odpovídá střední poloha serva, 1ms je jedna a 2ms druhá krajní poloha. Mechanické provedení serva může být takové, že servo je schopno pohybu v rozsahu o něco větším než 180 stupňů, ale není to pravidlem. Většina servo-pohonů má na koncích rozsahu pohybu mechanické blokování, na což je potřeba dávat pozor, protože při dojetí na doraz se výrazně zvýší proud odebíraný servem a může to skončit tím, že shoří elektronika serva. [3]

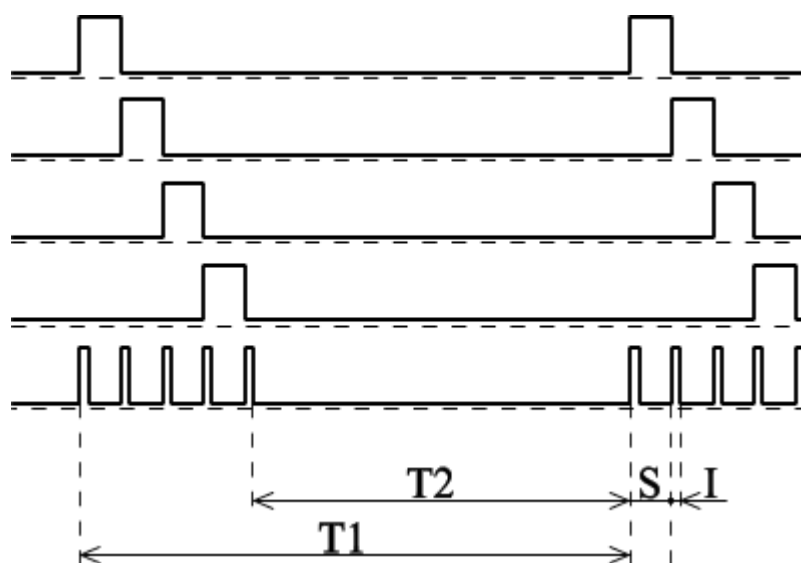


OBR 6.3: ukázka natočení serva v závislosti na délce pulsu. [3]

Řízení serva

Jak už bylo o něco výše napsáno, dnes používaná serva pracují s kladnými řídicími impulsy o délce 1-2ms. Tyto impulsy se opakují většinou s frekvencí kolem 50Hz. Rychlost opakování není kritická, pokud budou řídicí impulsy přicházet méně často, pouze se zpomalí rychlost pohybu serva. Velikost impulsů je rovná velikosti napájecího napětí serva, což je 4,8-6V (napájení čtyřmi články NiCd $4 \times 1,2V = 4,8V$ nebo suchými články $4 \times 1,5V = 6V$).

Na **OBR 6.4** je vidět jak vypadají impulsy na výstupu modelářského přijímače (první čtyři průběhy). Řídicí impulsy pro jednotlivá serva se opakují přibližně v rytmu 50Hz. Pátý průběh ukazuje, jak vypadá výstupní signál kodéru modelářského vysílače pro 4 serva. Podle typu kodéru zůstává konstantní buď T1 a skupiny impulsů se opakují v pravidelném rytmu (kolem 50Hz), nebo je konstantní mezera mezi skupinami impulsů T2 (většinou 10-15ms). S je délka řídicího impulsu pro jedno servo (1-2ms), I je oddělovací impuls (0,25-0,3ms). [3]



OBR 6.4: Impulsy na výstupu modelářského přijímače. [3]



OBR 6.5: Použití servo-pohonů v úloze. Manipulace se zbožím, umístěným na váze.

7 Tvorba vizualizace

7.1 Vizualizace ve zkratce

Vizualizace obecně je grafické prostředí sloužící k informování a ovládání zařízení pro které je navržena. V případě zařízení pro optimalizaci vážení je vizualizace vytvořena v prostředí VISU+ od společnosti Phoenix Contact. Program Vizualizace je spuštěn formou programu na dotykovém displeji. Dotykový displej plní funkci univerzálního vstupně-výstupního a rovněž informativní zařízení, tak že při doteku aktivní plochy obrazovky se na toto místo nastaví kurzor. Kurzoru je poté přidělena funkce. Obsluha je o stavu běžící aplikace informována formou grafických schémat, nebo číselně.

7.2 Princip dotekové technologie displeje TP 12 T

Tato kapitola vychází z [4].

Displej TP 12 T využívá pro dotykovou obrazovku rezistivní technologii. Její základ tvoří pružná membrána, která je na povrchu displeje. Tato průhledná membrána je zevnitř pokryta velmi tenkou a rovněž průhlednou vodivou vrstvou z kovu. Naproti pružné membráně je další vodivá průhledná vrstva, která je ale pevná.

Obě vrstvy vzájemně odděluje velmi tenká vzduchová mezera. Kvůli pružnosti horní membrány je třeba mezeru mezi oběma vrstvami pojistit rastrem izolačních podpěr umístěných ve vzduchové mezeře. Obě vodivé plochy jsou samozřejmě připojeny k řídicímu a vyhodnocovacímu modulu.

Princip činnosti je nasnadě. Při doteku displeje se horní pružná vrstva prohne a v daném místě se vodivě spojí s vrstvou spodní. Mezi vrstvami potom začne procházet elektrický proud. Řídicí jednotka pak na základě velikosti jednotlivých proudů automaticky vypočítá polohu bodu doteku. Hlavní výhodou tohoto řešení displeje je především to, že k doteku lze použít prakticky cokoli. Může to být špička prstu, a to třeba i v rukavici, tužka, speciální pero nebo v podstatě jakýkoli jiný předmět vhodného tvaru a velikosti.

Jde tu v podstatě jen o propojení obou vrstev vyvinutím tlaku na horní vodivou vrstvu. Vrchní vodivá vrstva, tedy ta, které se dotýkáme, je pod napětím. Spodní vrstva funguje jako obraz vrchní vrstvy a její funkcí je určení polohy doteku. Existují dva základní typy rezistivních displejů.

První z nich představuje tzv. čtyřvodičové řešení, kde se počítá poloha bodu podle odporů od krajů vodivých desek. U druhého pětivodičového typu se poloha stisku určuje pomocí proudů mezi horní deskou a čtyřmi kontakty v rozích spodní podkladové desky.

Rezistivní dotekové displeje se používají především v průmyslových a veřejných aplikacích. To proto, že k ovládání lze použít opravdu cokoli, v případě velkého displeje pro ovládání stavební mechaniky třeba i násadu lopaty. Výhodná je nízká výrobní cena a jednoduchost provedení.

Displeje se vyznačují vysokou rychlostí odezvy a lze u nich dosáhnout i velké přesnosti určení polohy. Nevýhodou je to, že se takovýto displej používáním dříve či později vymačká. Navíc je tu náchylnost ke zničení displeje ostrým předmětem.

K nevýhodám patří také snížení průhlednosti displeje vlivem kovových vrstev. Displeje najdeme třeba na veřejných místech – vyhledávače spojů na nádraží, interaktivní informační panely apod. [4]

7.2 Způsob práce v prostředí VISU+

V prostředí VISU+ se pracuje způsobem objektového programování. Tento princip je velmi podobný např. prostředí Delphi. Nejvhodnější způsob programování je využívání předdefinovaných prvků, jako jsou tlačítka, displeje atp. a přiřazovat jim funkci. Přiřazenou funkcí mohou být nejen změny proměnných, ale i interakce s různými druhy klávesnic, spojením s grafickými mechanismy, využitím skriptovacího jazyka VBscript, atd. Zároveň tyto funkce mohou být spuštěny paralelně. Tzn. při stisku tlačítka se provede např. změna hodnoty proměnné, provede se skript a otevře se nové okno.

Spolupráce aktivních prvků vizualizace s PLC, resp. s proměnnými z PLC je realizováno pomocí OPC serveru. OPC server běží v počítači který je součástí dotykového displeje. Pro bezproblémový chod se serverem OPC se na dotekovém displeji musí nastavit několik parametrů. Jedním z nejdůležitějších parametrů je nastavení IP adres PLC se kterými bude vizualizace pracovat. V PLC se proměnná, která má být ve vizualizaci viditelná nastaví jako OPC (to se provede při deklaraci proměnné). Tato volba je k dispozici v nastavení proměnných v prostředí PCworx. Program vizualizace se programuje externě a před vlastním odesláním programu s vizualizací do dotekového displeje, je nutno vizualizaci naprogramovat na počítači se softwarem VISU+ a dále pomocí OPC konfiguratoru nastavit parametry nutné pro bezproblémovou komunikaci. Nutné parametry jsou již zmíněné IP adresy PLC a IP adresa virtuálního OPC serveru, na kterém běží VISU+.

7.3 Struktura vizualizace

Vizualizace je vytvořena ze čtyř obrazovek. Základní obrazovka, obrazovka pracující v režimu optimalizace, režim vážení s tříděním dle hmotnosti a obrazovka s režimem třídění bez optimalizace. Mezi režimy se dá přecházet pomocí hlavní obrazovky.

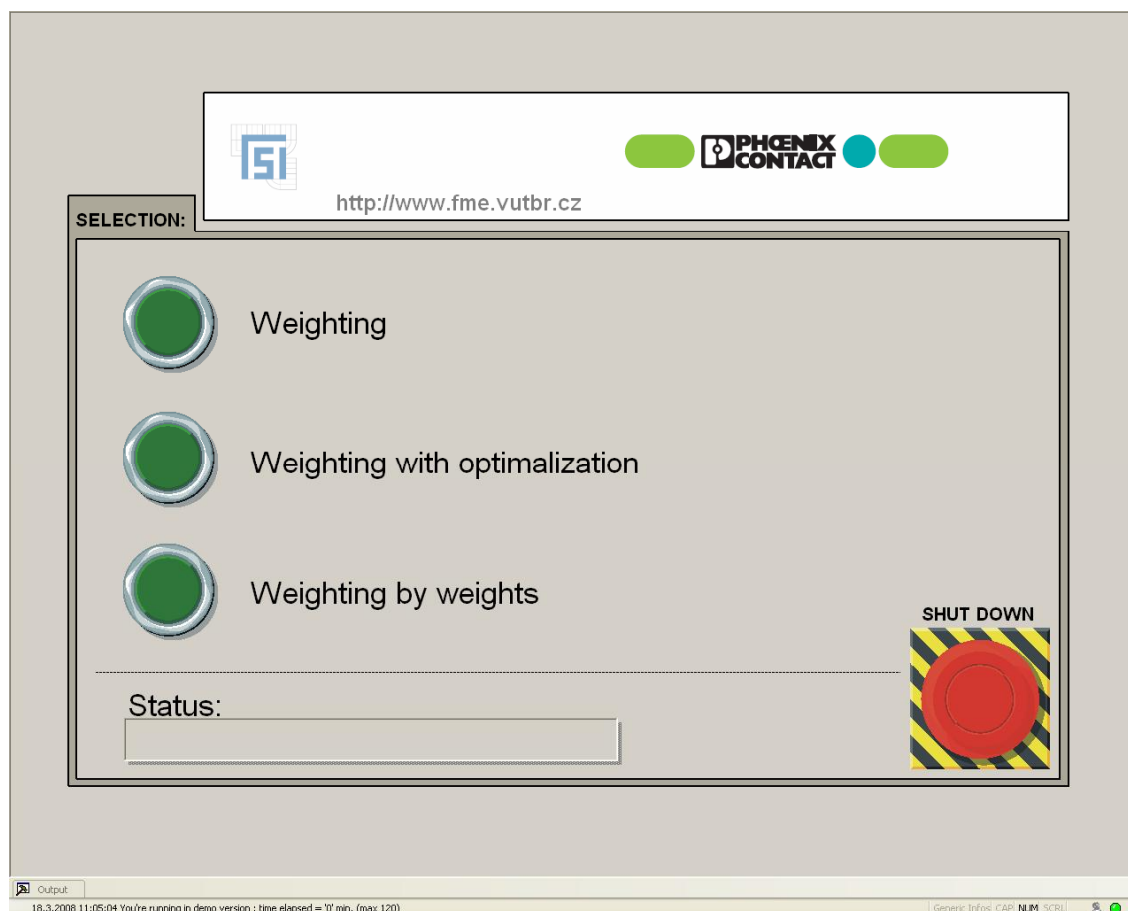
7.3.1 Základní obrazovka

Na základní obrazovce je možnost výběru režimu. Tato obrazovka je spuštěna automaticky po startu dotykového displeje. Při stisku tlačítka zvoleného režimu je změněna hodnota proměnné iRežim z hodnoty 0 na hodnotu 1 až 3 (dle zvoleného režimu). Všechny proměnné pracující se softwarem VISU+ mají nastavenou funkci OPC komunikace.

Kromě tlačítek na volbu režimu je na obrazovce vidět i displej s nápisem „status“. Displej slouží jako informativní textové pole o stavu obou PLC. V tomto případě informuje o tom, zda spolu PLC úspěšně navázaly spojení, případně zda spojení nebylo přerušeno.

Posledním nepopsaným prvkem je tlačítko central stop. Po stisku tohoto tlačítka se ukončí běžící aplikace a obsluha se dostane na plochu operačního systému počítače

dotykového displeje. Zde je možné provádět nastavení IP adres PLC, nastavení IP adresy OPC serveru apod. Pokud by zde nebyla volba pomocí tlačítka central stop, úloha by se vždy musela vypínat pomocí počítače, na kterém byl projekt vytvořen a pomocí něhož byl projekt nahrán do dotykového displeje. Z tohoto počítače je možné úlohu rovněž spustit. Spouštění úlohy probíhá automaticky po zapnutí displeje, nebo poklepáním na zástupce. Na **OBR 7.1** je vidět úvodní obrazovka dotykového displeje.



OBR 7.1: Úvodní obrazovka dotykového displeje.

7.3.2 Režim vážení bez optimalizace

Tento režim je spuštěn po zvolení volby „Weighting“ na hlavní obrazovce. Stisk tlačítka provede změnu proměnné iRežim na hodnotu 1, paralelně s touto změnou se změní aktuální obrazovka, na obrazovku zvoleného režimu.

Na obrazovce jsou ovládací prvky START a BACK. Tlačítko START spouští vážení změnou proměnné xStartPossible z hodnoty FALSE na hodnotu TRUE. Po dokončení jednoho cyklu vážení je proměnná xStartPossible přepnuta zpět na hodnotu FALSE a pro nové vážení je nutno znovu stisknout tlačítko START. Volba BACK pracuje s proměnnou iRežim. Po stisku tlačítka BACK je hodnota iRežim nastavena do hodnoty 0 a zároveň je spuštěna úvodní obrazovka. Aktuální vážící cyklus pouze dojde do konce (tato operace trvá necelou sekundu a je kratší než odezva displeje, resp. OPC serveru.).

Dalšími prvky jsou prvky informačního charakteru POSITION, WEIGHT, STATUS a kontrolka WEIGHTING. Informační displej POSITION ukazuje aktuální polohu ramene, tato funkce je zařazena spíše pro efekt a demonstraci, jak rychlé odezvy je schopen OPC server. Je vyzkoušeno že displej je dostatečně rychlý, aby rychlé změny plynule zvládl a rychlost zobrazování „brzdí“ OPC server. Displej WEIGHT ukazuje naměřenou hodnotu hmotnosti. Displej pracuje s proměnnou iNamerenaHmotnost. Kontrolka WEIGHTING signalizuje zda je vážení spuštěno a pracuje s proměnnou xStart, takže svítí pokud je proměnná xStart=TRUE (pokud program běží). Displej STATUS zobrazuje informační komentáře o stavu zařízení, chybová hlášení, případně vodítka pro obsluhu. Displej STATUS pracuje s proměnnou sText. Proměnná pracuje v podmínce s proměnnou iKrok a podle stavu hodnoty iKrok je proměnné sText přiřazen komentář v datovém typu STRING.



OBR 7.2: Obrazovka režimu vážení bez optimalizace.

7.3.3 Režim vážení s optimalizací

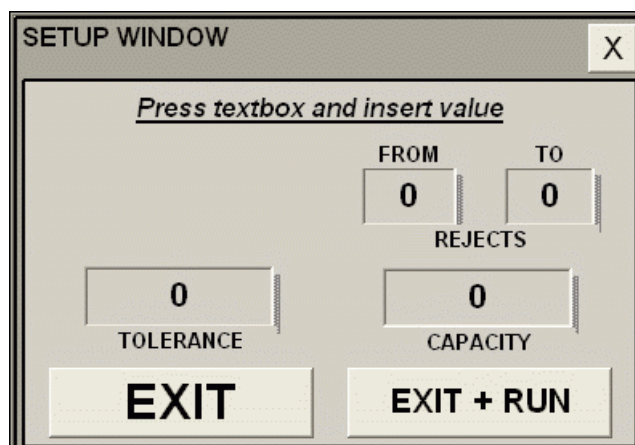
Obrazovka režimu vážení s optimalizací je spuštěna po stisknutí tlačítka „Weighting with optimization“ umístěného na hlavní obrazovce. Při stisku tlačítka je hodnota proměnné režim změněna na hodnotu 2 a je spuštěna obrazovka režimu. Zobrazovače a ovládací prvky umístěné na obrazovce režimu jsou následující

Zobrazovače

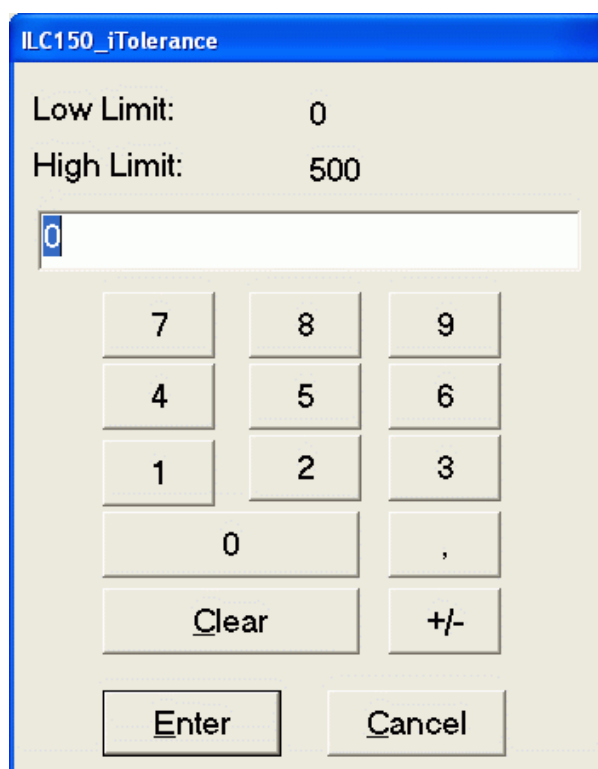
- **ACTUAL WEIGHT:** Displej zobrazuje hmotnost naposledy naměřeného zboží. Pod tímto displejem je umístěna ručička, zobrazující reálné natočení ramene. Hodnota hmotnosti pracuje s proměnnou `iNamereaHmotnost` a velikost natočení ramene pracuje s proměnnou `iPosition`.
- **REJECT:** Položka reject slouží ke stanovení minimální a maximální hmotnosti váženého zboží, které ještě nebude vloženo do čtvrtého kontejneru jako mimojakostní výrobek. Jedná se tedy o stanovení hmotnostní jakosti zboží. Hodnota je nastavitelná při stisku tlačítka **SETUP**. Celkem jsou nastavitelné 3 hodnoty, Reject, kapacity a tolerance. Maximální a minimální hmotnost zboží pracuje s proměnnými `iZboziMin` a `iZboziMax`.
- **TOLERANCE:** Jedná se rovněž o nastavitelnou hodnotu, která pracuje s proměnnou `iTolerance`. Parametr slouží jako tolerance kapacity kontejnerů. Kontejnery jsou tak plněny na určitou nastavenou hmotnost, plus mínus hodnota nastavené tolerance. Popis práce s tolerancí, viz kap.5.2.1.
- **NEXT FILLING CONTAINER:** Zobrazuje hodnotu kontejneru, do kterého bude umístěno zboží. Zobrazovač pracuje s proměnnou `iOptKontejner`.
- **STATUS:** Zobrazovač, který byl zmíněn již dříve. Na jeho displeji je textovou formou zobrazován stav zařízení, výskyt chyb a pokyny obsluze. STATUS zobrazovač pracuje s proměnnou `iText`.
- **Kontrolky RUN a STOP:** Tyto kontrolky signalizují stav režimu, resp. Zda právě běží, či nikoli. Pracují v interakci s tlačítkem **START/STOP**, resp. Se stejnou proměnnou `xStartButton`. Když je `xStartButton=TRUE` svítí **RUN** a v opačném případě svítí **STOP**.

Tlačítka

- **SETUP:** Pomocí tlačítka je spuštěno okno kde se nastavují hodnoty parametrů **REJECT**, **TOLERANCE** a **CAPACITY**, viz obr.14. Hodnoty se editují po stisku políčka s jejich zobrazovanou hodnotou. Při stisku políčka se zobrazí numerická klávesnice, na které se požadovaná hodnota navolí. Numerická klávesnice je zobrazena na **OBR 7.3**. Funkci **SETUP** lze spustit i za běhu režimu, při stisku klávesy se režim zastaví (proměnná se přepne na hodnotu `xStartButton=FALSE`) a je zobrazeno okno nastavení **SETUP WINDOW**. Při jeho opouštění lze běh režimu buď přímo spustit, nebo se lze pouze vrátit o úroveň výš. Při opouštění **SETUP WINDOW** je rovněž spuštěn Vbscript, který znemožní spuštění úlohy při nastavení nulových, nebo nesmyslných parametru (například přehození hodnot **FROM** a **TO**).



OBR 7.3: Nastavení hodnot vybraných parametrů pro režim vážení s optimalizací.



OBR 7.4: Ukázka numerické klávesnice, s jejíž pomocí se deklaruje většina nastavitelných parametrů. Položky Low Limit a High Limit jsou nastavitelné programátorem.

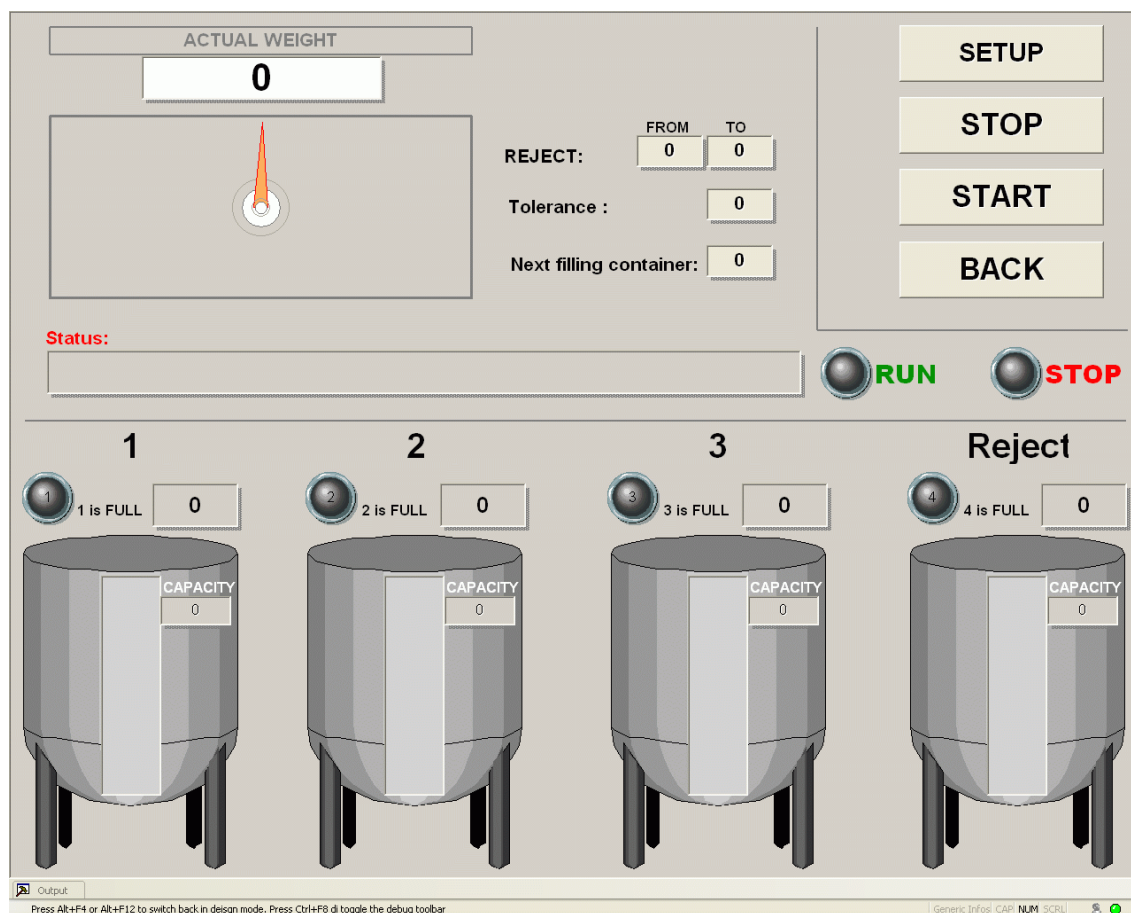
- START/STOP: Tlačítka ovládající běh režimu. Hodnota proměnné iStartButton je při manipulaci s tlačítky TRUE při stisku START a FALSE při stisku STOP. Úlohu lze takto v jakékoli fázi zastavit a opět spustit. Při jakémkoli spouštění činnosti režimu tlačítkem START je zároveň spuštěn skript na kontrolu nastavených podmínek, obdobný jako při opouštění SETUP WINDOW.

- BACK: Při stisku tlačítka back je hodnota proměnné iRezim nastavena na hodnotu 0. Zároveň je i hodnota iStartButton nastavena na hodnotu FALSE. Pokud je stisknuta klávesa BACK při činnosti některého servo-pohonu, je z důvodu špatného vynulování časovače činnost serv dokončena a program je ukončen až po této operaci. Kdyby tak nebylo mohlo by se snadno stát že by servo, při znovu spuštění režimu, nenajelo na svou první pracovní polohu (například by nevyložilo zboží apod.).

Zobrazovače kontejnerů

U každého symbolu kontejneru jsou čtyři zobrazovací prvky. Kontrolka ukazující zda je kontejner plný, displej aktuální hmotnosti kontejneru, Zobrazovač vizuálně ukazující jakou má kontejner hmotnost a displej CAPACITY.

- CAPACITY: Displej zobrazuje jak velkou kapacitou hmotnosti kontejner disponuje. Hodnota je pro všechny kontejnery stejná. Displej pracuje s proměnnou iWeight.
- Signalizace plnosti kontejneru: Pokud se kontejner stane plným, rozsvítí se kontrolka. Prakticky: Je-li hodnota xCheckFullContainer (s indexem příslušného kontejneru) přepnuta z hodnoty FALSE na hodnotu TRUE. Právě s touto proměnnou pracuje zobrazovací kontrolka. Svítí při stavu TRUE a nesvítí při stavu FALSE. Stav proměnné xCheckFullContainer jsou podrobně popsány v kapitole 5.2.1.
- Aktuální hmotnost kontejneru: K zobrazování tohoto údaje slouží dva displeje zobrazující vizuální a číselný údaj. K oběma zobrazením je použita proměnná iHmotnostZboziContainer (s indexem příslušného kontejneru). K vizuálnímu zobrazení je navíc použita hodnota iWeight, za účelem grafického znázornění „stropu“ (z důvodu věrohodného grafického zobrazení).



OBR 7.5: Obrazovka režimu vážení s optimalizací.

7.3.4 Režim vážení s tříděním dle hmotností

Obrazovka režimu vážení s tříděním dle hmotností je aktivována po stisku tlačítka „Weighting by weights“ které je umístěno na hlavní obrazovce. Při stisku tlačítka je hodnota proměnné iRežim změněna z hodnoty 0 na hodnotu 3 a je spuštěna obrazovka režimu. Záměrně se obrazovka příliš neliší od obrazovky předchozího režimu a to z důvodu příjemnější obsluhy. Zobrazovače a ovládací prvky umístěné na obrazovce režimu jsou následující

Ovládací tlačítka STOP, START, BACK a zobrazovače RUN, STOP, ACTUAL WEIGHT. Zobrazovač aktuální pozice ramene, TOLERANCE, WEIGHT, CAPACITY, kontrolka plnosti a NEXT FILLING CONTAINER jsou shodné jako u předchozího režimu. Pouze kontrolka plnosti pracuje s jinou proměnnou (v tomto případě xFullContainer1), ale její funkce zůstala stejná jako v obrazovce předchozího režimu.

Tlačítko SETUP spouští okno ve kterém je možnost nastavení několika parametrů. Oproti předchozí obrazovce je zde ale možnost nastavení jiných parametrů. Jsou to parametry FROM, TO a CAPACITY. Ty se nastavují pro tři kontejnery. Pro čtvrtý kontejner je nastavitelná pouze hodnota CAPACITY. Výraznou změnou při nastavování parametrů je pouze možnost nastavení kapacity všech kontejnerů.

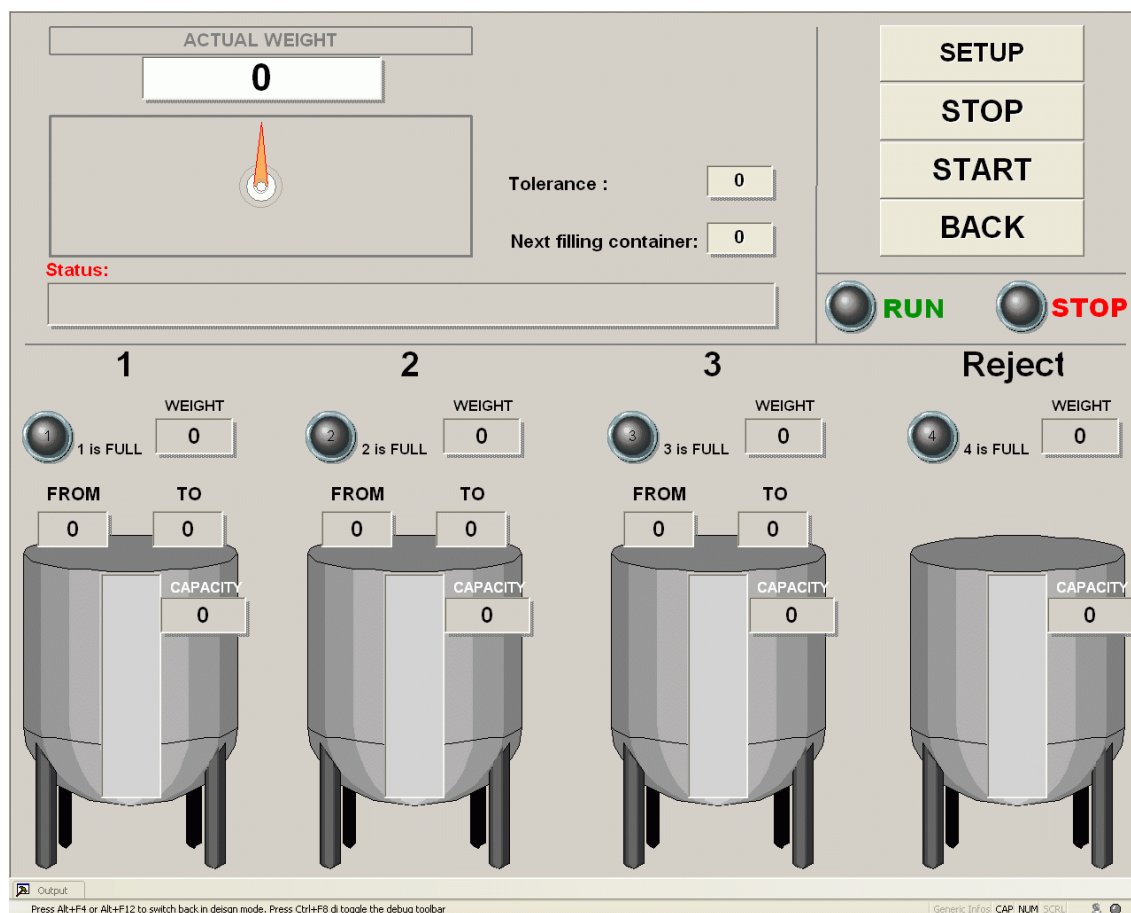
The image shows a graphical user interface window titled "SETUP WINDOW" with a close button (X) in the top right corner. Inside the window, there is a instruction: Press textbox and insert value. The window is divided into four sections for container setup:

- Container 1 - setup:** Contains two input fields labeled "FROM" and "TO", both with the value "0". Below them is a label "CAPACITY" and a larger input field also containing "0".
- Container 2 - setup:** Similar to Container 1, with "FROM" and "TO" fields set to "0", a "CAPACITY" label, and a larger input field set to "0".
- Container 3 - setup:** Similar to Container 1, with "FROM" and "TO" fields set to "0", a "CAPACITY" label, and a larger input field set to "0".
- Container 4 - setup:** Contains a label "REJECT" and a larger input field labeled "CAPACITY" set to "0".

At the bottom of the window, there are two large buttons: "EXIT" on the left and "EXIT + RUN" on the right.

OBR 7.6: Okno *SETUP WINDOW*, sloužící pro stanovení nastavitelných hodnot režimu vážení dle hmotnosti.

Hodnoty FROM a TO, které jsou zobrazeny u všech kontejnerů s výjimkou čtvrtého slouží pro vymezení hmotnostní jakosti tříd výrobků. Tomuto problému se detailně věnuje kapitola 5.2.2



OBR 7.7.: Ukázka obrazovky pro režim vážení s tříděním dle hmotnosti.

8. Závěr

Diplomová práce Řízení procesu třídění programovatelným automatem se detailně zabývá praktickým zpracováním výroby zařízení pro optimalizaci vážení, resp. jednou z jeho dvou částí.

Celé zařízení slouží k vážení a třídění zboží. Skládá ze dvou na sebe navzájem závislých celků, spolupracujících PLC, řídicího a řízeného, které si mezi sebou vyměňují data a příkazy týkající se ovládání váhy a naměřených hmotností. O propojení zařízení se stará klasický ethernet switch. Řídící část, tedy řídicí PLC, zároveň slouží k třídění. Část pro vážení ke své činnosti využívá speciální otočnou váhu, jež pracuje s kinematickým momentem. Moment je zjištěn ze spotřeby proufu při rozběhu a doběhu elektro-motoru váhy.

Každému celku se věnuje vlastní diplomová práce. Tato práce se zabývá výhradně svým řešeným problémem a to řídicím PLC a jeho rozšířeními. Rozsah této diplomové práce se věnuje zmíněné komunikaci mezi PLC, vytvořením algoritmu na třídění váženého zboží a jejich uspořádávání do kontejnerů.

Prakticky se jedná o realizaci programového vybavení pro PLC Phoenix Contact, včetně práce se servo-pohonem, výměny dat mezi PLC a vytvoření uživatelsky příjemné vizualizace. Zařízení pro třídění je ovládáno pomocí PLC systému s rozšiřujícími moduly pro ovládání servo-pohonů a dotykovou obrazovkou stejného výrobce, sloužící pro styk s obsluhou. V počítači dotykové obrazovky je spuštěn program vizualizace, který je naprogramován pomocí objektového programování a visual basic script v prostředí Visu+. Prostor Visu+ je pro dotykovou obrazovku k dispozici od výrobce. Data mezi obrazovkou a PLC jsou sdílána pomocí OPC serveru. PLC je naprogramováno v prostředí PCWorx který je rovněž k dispozici od výrobce.

PLC plní v zařízení funkci řídicího a výpočetního faktoru. Dotyková obrazovka má funkci univerzálního vstupně výstupního grafického zařízení. Pro úspěšné vyřešení zadaného problému bylo nutné pečlivé seznámení s veškerým použitým zařízením, což spotřebovalo podstatnou část času potřebného pro vlastní řešení projektu.

Výsledkem obou prací je zařízení pracující zcela samostatně a spolehlivě. Veškeré ovládání, nastavování parametrů a zobrazování údajů je řešeno pomocí dotykové obrazovky s intuitivním ovládáním. Pomyslným základem úspěšného fungování celého zařízení je úspěšné navázání komunikace mezi řídicím a řízeným PLC. S chybnou komunikací nelze zařízení ze své podstaty provozovat. Další podstatnou částí řešení bylo vymyšlení vhodného algoritmu na třídění zboží, který byl označen jako optimalizace, ovládání servo-pohonů a seznámení se s jejich činností. V průběhu řešení byla úloha rozšířena o další dva režimy činnosti, z nichž jeden je použit pro jiný způsob třídění. Třetí režim je implementován pro demonstraci komunikace mezi váhícím a řídicím PLC.

Zařízení bylo vyrobeno pro účast v soutěži Xplode new automation 2008. Tým řešitelů prošel výběrovým řízením, a účastnil se v soutěži jako jeden ze sta vybraných soutěžních týmů z celého světa. Celkové řešení projektu bylo úspěšné a tím pádem školní laboratoř získala zařízení o hodnotě 3000 €.

Svou atraktivitou, použitým zařízením a prací v týmu úloha velmi obohatila mé dosavadní profesní zkušenosti a rozšířila obzory používání automatizace v praxi.

Diplomová práce splnila všechny body zadání. Zařízení je plně funkční po softwarové stránce. Hardwarově pouze chybí umístit držáky servo-pohonů, výklopné zařízení na rameno váhy a zásobník zboží.

9. Seznam použité literatury

- [1] Doc. Ing. František Zezulka, CSc, Ing. Zdeněk Bradáč, Ing. Petr Fiedler, Ing. Pavel Kučera, Ing. Radek Štohl. *Programovatelné automaty*. [PDF dokument]. Brno: Fakulta elektrotechniky a komunikačních technologií, Říjen 2003 [cit. 1.5.2008]. Dostupný z:
<http://www.vaeprosyst.cz/Dokumentace/Programovatelne_automaty/Programovatelne_automaty-Skripta_FEKT_VUT_Brno.pdf>
- [2] Hitec RCD USA [online]. 2007 [cit. 18.4.2008]. Dostupné z:
<<http://www.hitecrcd.com/servos/show?name=HS-475HB>>
- [3] Vlastikd – Bastlír & spol, Jak funguje... modelářské servo, [online]. 24.7.2004, [cit. 18.4.2008]. Dostupné z:
<<http://vlastikd.webz.cz/bastl/serva.htm>>
- [4] Snášel, Jaroslav. Poslechně váš dotek. *Mobility* 2007, ročník 9, č.7, s. 38-42. ISSN 1212-9879.