



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

INOVACE LABORATORNÍ ÚLOHY - "TANKY"

LABORATORY MODEL „LIQUID TANKS“ INNOVATION

BAKALÁŘSKÁ PRÁCE

BACHELOR THESIS

AUTOR PRÁCE

AUTHOR

Antonín Škapa

VEDOUcí PRÁCE

SUPERVISOR

Ing. Václav Kaczmarczyk, Ph.D.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**
Ústav automatizace a měřicí techniky

Student: Antonín Škapa

ID: 172136

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Inovace laboratorní úlohy - "Tanky"

POKYNY PRO VYPRACOVÁNÍ:

Provedte komplexní inovaci laboratorní úlohy "Tanky", která je hlavní součástí cvičení předmětu Automatizace Procesů.

1. Seznamte se s novou koncepcí úlohy
2. Zadokumentujte navržený zdrojový kód úlohy
3. Na základě získaných poznatků proveďte úpravy stávající dokumentace k předmětu.
4. Vytvořte zkrácenou verzi dokumentace obsahující pouze přehledně sestavené výpisy kódů, jako podklad pro vyučující, kteří budou hotové práce hodnotit.

DOPORUČENÁ LITERATURA:

Pásek, J.: Automatizace procesů - Laboratorní cvičení I

Termín zadání: 5. 2. 2018

Termín odevzdání: 10. 8. 2018

Vedoucí práce: Ing. Václav Kaczmarczyk, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady



UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Tato práce se zabývá inovací laboratorní úlohy „Tanky“. Jsou zde popsány nejpoužívanější senzory a aktuátory, včetně možností jejich připojení k PLC. Je rozebrána a popsána problematika normy ANSI/ISA S88 BATCH a ANSI/ISA S95 MES, kde jsou zvláště rozebrány výkonové ukazatele KPI. Jelikož úloha „Tanky“ pracuje s modelem technologického tanku, jsou zde tyto zdokumentované včetně příkladů použití v průmyslu. Dále jsou zde objasněny programovací jazyky pro PLC, zejména SCL. Nakonec je popsáno samotné zadání úlohy včetně technologického schématu a ošetření chybových stavů.

Klíčová slova

PLC, Siemens S7, SCL, ST, Laboratorní úloha, HMI, Senzory, aktuátory, BATCH, S88, MES, S95

Abstract

This bachelor thesis deals with innovation of laboratory exercise „Tanks“. There are described the most used sensors and actuators including means of connection to PLC. The ANSI/ISA S88 and ANSI/ISA S95 standards are depicted as well with emphasis on key performance indicators. Technological tanks are documented here including industrial use examples since the laboratory exercise „tanks“ works with model of a technological tank. Furthermore there are briefly analyzed PLC programming languages, especially SCL. In the end, there is the laboratory exercise assignment with technological scheme and attention of error states.

Keywords

PLC, Siemens S7, SCL, ST, Laboratory exercise, HMI, Sensors, actors, BATCH, S88, MES, S95

Antonín Škapa. Inovace laboratorní úlohy „Tanky“. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace, 2018. 56 s., 45 s. příloh. Bakalářská práce. Vedoucí práce: Ing. Václav Kaczmarczyk, Ph.D.

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Inovace laboratorní úlohy „Tanky“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené semestrální práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

Poděkování

Děkuji vedoucímu bakalářské práce Ing.Václavu Kaczmarczykovi, Ph.D. za odborné vedení, konstruktivní kritiku a přátelský přístup. Dále bych také chtěl poděkovat Ing.Tomáši Benešlovi za neúnavné konzultace, praktické ukázky použitého softwaru a užitečné tipy zkušeného programátora.

Obsah

1	Úvod	1
2	Teoretický popis	2
2.1	Technologické procesy a procesní instrumentace.....	2
2.1.1	Rozdělení technologických procesů	2
2.2	Řízení procesů.....	4
2.2.1	Snímače a možnosti jejich připojení k PLC	4
2.2.2	Základní akční členy a možnosti jejich řízení.....	9
2.3	Řízení dávkových procesů (S88)	16
2.3.1	Modely standardu S88 [40].....	17
2.3.2	Receptury	18
2.3.3	Úrovně řízení	19
2.4	Propojení se systémy pro řízení výroby (MES – S95) [29].....	20
2.4.1	Správa výrobních zdrojů	20
2.4.2	Správa výrobních postupů	21
2.4.3	Detailní plánování výroby	21
2.4.4	Dispečerské řízení	21
2.4.5	Řízení výroby	21
2.4.6	Sběr dat	21
2.4.7	Sledování výrobků a jejich rodokmen.....	21
2.4.8	Výkonnostní analýzy [42].....	22
3	Programovací jazyky	26
3.1	STL.....	26
3.2	GRAFCET	26
3.3	LAD	27
3.4	SCL.....	27
3.4.1	Vývoj jazyka SCL.....	27
3.4.2	Deklarace proměnných.....	28
3.4.3	FOR cyklus	28
3.4.4	WHILE cyklus	29
3.4.5	REPEAT cyklus.....	29
3.4.6	Použití EXIT v SCL.....	30
3.4.7	Použití RETURN v SCL.....	30
3.4.8	Čítač v jazyku SCL	31
3.4.9	Časovač v jazyku SCL.....	33

3.4.10	Stavový automat v jazyku SCL	34
3.4.11	Komunikace MODBUS v jazyku SCL	37
4	Úloha TANKY	41
4.1	Pasterace	41
4.2	Technologické schéma tanku	42
4.3	Popis vstupů/výstupů	43
4.3.1	Napouštěcí ventil	43
4.3.2	Míchadlo	43
4.3.3	Topné těleso	43
4.3.4	Výpustný ventil	44
4.3.5	Senzory	44
4.4	Zadání úlohy	44
4.4.1	Možné poruchové stavy	45
4.5	Náhled pracoviště	46
5	PLC Simatic S7-1500	47
5.1	Rozšiřující karty	47
5.1.1	Připojení snímačů k PLC	47
5.2	PLC použité v úloze	49
5.3	Zdroj	50
5.4	AI 5/AQ 2	50
5.5	DI 16/DQ 16_1	50
5.6	DI 16/DQ 16_2	50
6	Závěr	51
	Literatura	52
	Seznam příloh	56

Obr. 39 - Technologický tank [32]	41
Obr. 40 - Technologické schéma tanku.....	42
Obr. 41 - Náhled pracoviště.....	46
Obr. 42 - Zapojení sensoru s proudovou smyčkou do PLC Siemens [46].....	48
Obr. 43 - Čtyřvodičové zapojení sensoru s proudovou smyčkou	49
Obr. 44 - Možné konfigurace PLC S7-1500 [4].....	49

Seznam tabulek

Tabulka 1 - Srovnání termočlánků [13]	5
Tabulka 2 - Porovnání kovů pro teplotní čidla	6
Tabulka 3 - Výstupy napouštěcího ventilu.....	43
Tabulka 4 - Zpětná vazba od napouštěcího ventilu	43
Tabulka 5 - Ovládání stykače míchadla a jeho zpětná vazba	43
Tabulka 6 - Výstupy ohřevu	43
Tabulka 7 - Výstupy vypouštěcího ventilu	44
Tabulka 8 - Zpětná vazba vypouštěcího ventilu.....	44
Tabulka 9 - Senzory v technologickém tanku	44

Seznam zkratek

ERP – Enterprise resource planning, Podnikový informační systém

FBD – Function block diagram, Programovací jazyk funkčních bloků

HMI – Human machine interface

HTST – High temperature short time pasteurization, Pasterizace vysokou teplotou po krátký čas

KPI – Key performace indicator, Klíčové ukazatele výkonnosti

LED – Light emitting diode, Elektroluminiscenční dioda

LTLT – Low temperature long time pasteurization, Pasterizace pro nízké teplotě po dlouhý čas

MES – Manufacturing execution system, Výrobní informační systém

MIS – Management information system, Manažerský informační systém

MTBF – Mean time between failure, Střední doba mezi poruchami

MTTR – Mean time to repair, Střední doba opravy

NC – Normally closed, Normálně zavřeno

NO – Normally open, Normálně otevřeno

OEE – Overall equipment effectiveness, Celková efektivita zařízení

OPE – Overall proces effectiveness, Celková efektivita procesu

PEE – Production equipment efficiency, Celková výkonnost produkce

PID – Proportional–integral–derivative controller, Proporciálně-intergračně-derivační regulátor

PLC – Programmable logic controller, Programovatelný logický automat

PWM – Pulse width modulation, Pulzně šířková modulace

SCL – Structured control language, Jazyk strukturovaného textu

SSR – Solid state relay, Polovodičové relé

TEEP – Total equipment effectiveness performance, Celková efektivita zařízení vztážená ke kalendářnímu času

UHT – Ultra high temperature pasteurization, Pasterizace za ultra vysoké teploty

1 ÚVOD

Cílem této bakalářské práce bylo provést inovaci laboratorní úlohy tanky do předmětu MAUP dle standartu BATCH v aktuální TIA portálu v14 SP1. Spolu s vedoucím práce jsme stanovili konkrétní cíle a vlastnosti tohoto projektu. Projekt byl průběžně testován v laboratoři na reálném PLC řady S7-1500 a HMI TP-700 Comfort. Na začátku této bakalářské práce jsou nastíněny rozdíly mezi spojitými, diskrétními a dávkovými technologickými procesy včetně příkladů použití v různých odvětvích průmyslu.

V další kapitole jsou postupně rozebrány různé typy hladinových a teplotních sensorů včetně způsobů připojení k PLC. Po senzorech jsou v této práci zpracovány nejčastěji používané aktuátory a způsoby jejich ovládání. U třífázových motorů jsou podrobně vysvětleny frekvenční měniče, jejich samotné ovládání, jako taky způsob fungování. V kapitole věnované dávkovému řízení je toto dále popsáno. Prostor je věnován zejména fyzickému, procedurálnímu a procesnímu modelu dle standartu S88 BATCH. Poté se věnují recepturám dle tohoto standardu a úrovním řízení jím zavedených. V návaznosti na standard S88 je popsáno řízení podniku a výroby MES systémem standardu S95.

V kapitole zabývající se MES systémy je vysvětleno hierarchické řízení na patra a jejich význam v řízení podniku a výroby jako takové. Manufacturing Execution System (MES) je rozebrán na jednotlivé funkční celky od řízení zdrojů, přes správu receptur až po výkonnostní analýzy. Další část je věnována rozboru programovacích jazyků používaných při programování PLC. Největší důraz je kladen na jazyk SCL, jelikož v něm je celá úloha naprogramovaná. Za příklady jsou uvedeny důležité konstrukce používané v tomto jazyku, jako i komunikační bloky.

Poté je popsána funkce technologického tanku na příkladu procesu pasterizace, který by bylo možné s tankem z naprogramované úlohy bez problému provést. Následně je popsána samotná úloha, nároky na ni kladené a možná úskalí. Je předvedeno pracoviště, na kterém budou studenti úlohu fyzicky vykonávat. Nakonec je vysvětlen pojem PLC a představeno PLC použité v úloze. K měřícím kartám PLC je názorně ukázáno připojení čtyřvodičového sensoru s proudovou smyčkou. Výsledky práce jsou poté shrnuty v závěru.

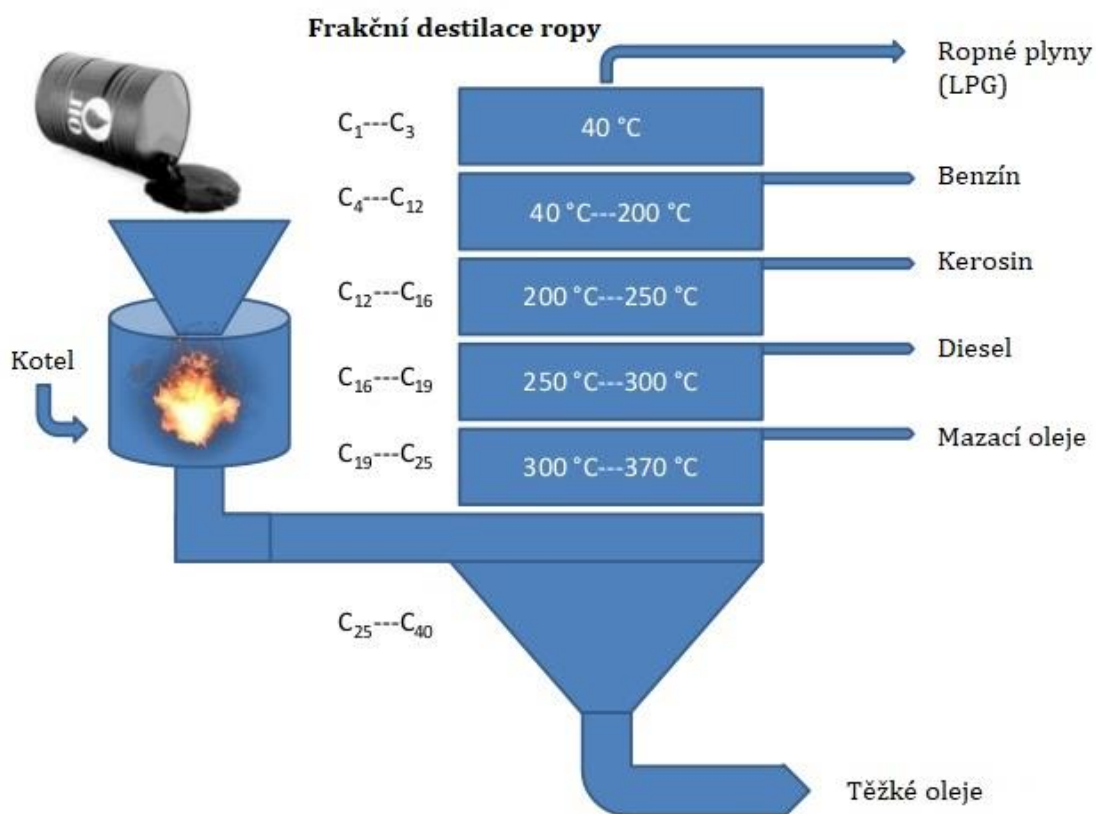
2 TEORETICKÝ POPIS

2.1 Technologické procesy a procesní instrumentace

2.1.1 Rozdělení technologických procesů

2.1.1.1 Spojité technologické procesy

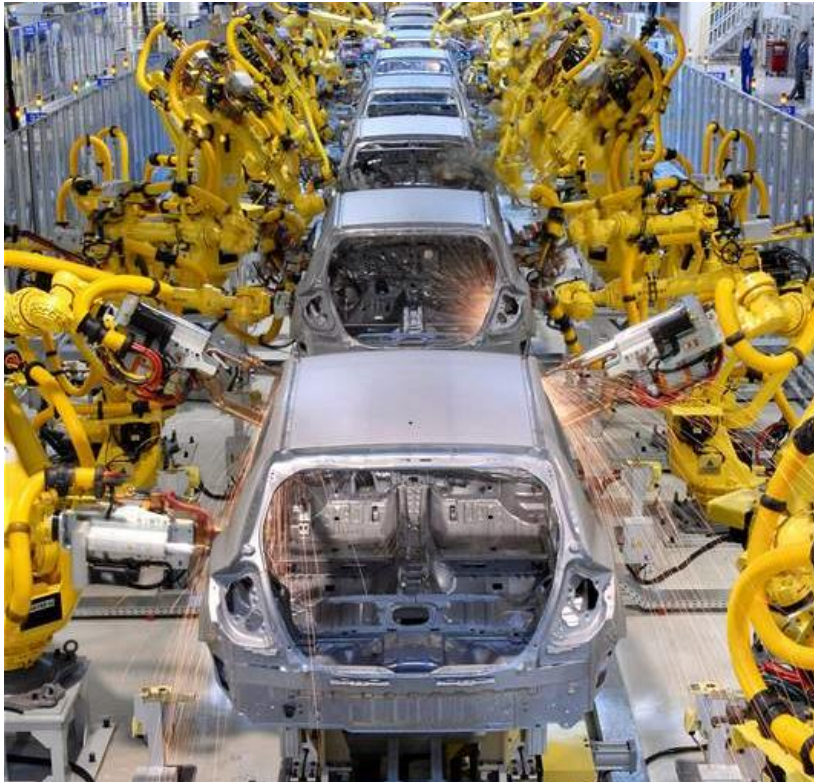
Podstatou spojitých technologických procesů je nepřetržitý tok materiálu přes různá technologická zařízení ve výrobě. Fyzický systém je často reprezentován proměnnými s plynulou a nepřerušovanou změnou v čase. Spojité procesy se nejčastěji používají v petrochemickém a chemickém průmyslu, kde je potřeba vyrábět ohromná množství produktu.



Obr. 1 - Kontinuální frakční destilace ropy - převzato a upraveno [37]

2.1.1.2 Diskrétní technologické procesy

Diskrétní technologické procesy se používají u výroby věcí. Díl, nebo stanovené množství dílů se pohybuje z jedné pracovní stanice do další a postupně se tak utváří hotový výrobek. Příkladem tohoto procesu může být výroba aut v automobilkách, nebo výroba elektrozařízení v manufakturách.



Obr. 2 - Diskrétní proces - výroba automobilů [38]

2.1.1.3 Dávkové technologické procesy

Dávkové technologické procesy jsou kombinací spojitých a diskretních procesů. Dávkové procesy se využívají napříč průmyslovým spektrem. Jejich použití zahrnuje výrobu sypkých hmot, čistících prostředků, barev a podobně. Dávkové procesy mají za začátku vždy určité množství vstupních materiálů, které je zpracováváno po konečný časový úsek a výsledkem je konečné množství výrobku. Toto množství vyrobené jedním cyklem dávkového procesu nazýváme také batch, neboli dávka. Řízení dávkových procesů je oproti spojitým, nebo diskretním procesům mnohem náročnější, jelikož tyto procesy mohou běžet nepřetržitě dlouho dobu, ale dávky vznikají často za sebou a přerušovaně. Nelze také oddělit jednotlivé prvky výroby, jako je tomu u diskretních technologických procesů. Pro zjednodušení řízení dávkových procesů tedy vyvstala nutnost tyto procesy normalizovat.



Obr. 3 - Dávkový proces - výroba piva [39]

2.2 Řízení procesů

2.2.1 Snímače a možnosti jejich připojení k PLC

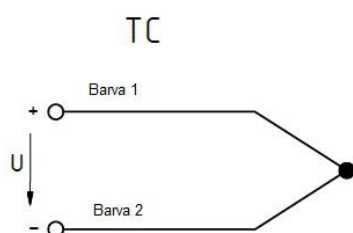
2.2.1.1 Měření teploty

Termočlánky

Termočlánky fungují na principu Seebeckova jevu objeveného roku 1821 německým fyzikem Thomasem Johannem Seebeckem. Principem Seebeckova jevu je přítomnost elektrického napětí na dvou koncích spojených drátů z různých materiálů, a to za předpokladu, že se liší teplota na místě spojovacím a na volných koncích drátu. Toto napětí nazýváme napětím termoelektrickým. Měříme vždy rozdíl mezi teplotou v místě spojení a teplotou na svorkách měřícího přístroje.

typ	měřicí rozsah [°C]	kombinace materiálů	použití
B	0-1700	Pt, 30 % Rh – Pt, 6 %Rh	pro měření vyšších teplot, nevyžaduje kompenzační vedení
E	-200-950	Ni, 10 % Cr – Cu, 45 % Ni	všeobecné použití, pro měření středních a nižších teplot
G	0-2300	W-W, 26 % Re	pro měření vysokých teplot, křehký
J*	-210-1200	Fe – Cu, 45 % Ni	pro měření vyšších teplot v redukční atmosféře
K*	-200-1250	Ni, 10 % Cr – Ni, 2 % Al, 2 %Mn, 1 % Si	všeobecné použití v oxidační atmosféře
P	0-1395	Platinel – Platinel	stabilnější, ale dražší než typy K a L
R	-50-1750	Pt, 13 % Rh – Pt	velmi přesný, pro vyšší teploty
T*	-270-400	Cu – Cu, 45 % Ni	všeobecné použití pro nižší teploty, odolný vlhkosti

Tabulka 1 - Srovnání termočlánků [13]



Obr. 4 - Schématická značka termočlánku [28]

Odporové kovové snímače teploty

V tomto typu senzorů používáme čisté kovy, aby nereagovaly s izolačním, nebo ochranným krytem a zároveň se snažíme o co největší stálost teplotního součinitele odporu α v čase.

$$R = R_0[1 + \alpha(T - T_0)] \quad (2.1)$$

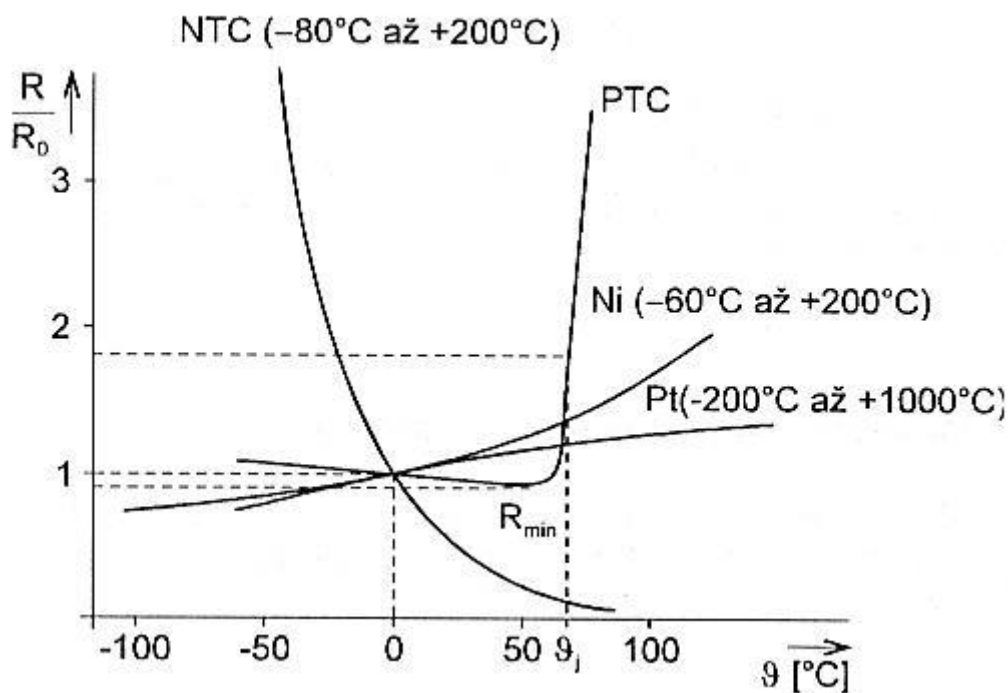
$$\alpha = \frac{R_{100} - R_0}{100R_0} \quad (2.2)$$

V majoritě případů používáme platinu, nebo nikl. Každý z těchto materiálů má své výhody a také úskalí, proto vybíráme typ kovu podle konkrétní aplikace a vyplývajících požadavků.

kov	měřicí rozsah [°C]	použití
platina	-200-850	Vysoká přesnost, dlouhodobá stálost, chemická odolnost, ale vysoká citlivost na magnetické pole
nikl	-60-150	Vyšší teplotní součinitel odporu oproti platině, vysoká citlivost
měď	-200-200 [25]	Levné, používá se nejčastěji k měření teploty vinutí motoru, běžně se nepoužívají kvůli snadné oxidaci mědi a nízké rezistivitě [20]

Tabulka 2 - Porovnání kovů pro teplotní čidla

Dle použitého kovu a odporu čidla i pojmenováváme. „Například označení Pt 100 udává, že se jedná o platinové čidlo s odporem 100 Ohmů při referenční teplotě 0°C.“ [26]



Obr. 5 - Porovnání teplotních senzorů [15]



Obr. 6 - Technologická značka čidla teploty [47]

2.2.1.2 Kapacitní měření hladiny

Kapacitní senzory se používají zejména pro kontinuální měření, ale existují i typy, které možno je použít pro měření limitních stavů. Výhodou kapacitních limitních senzorů je možnost použití i přes stěnu nádrže a tímto způsobem měřit například agresivní chemikálie.

$$C = \frac{S * \epsilon_r * \epsilon_0}{d} \quad (2.3)$$

S-plocha elektrod

d-vzdálenost mezi elektrodami

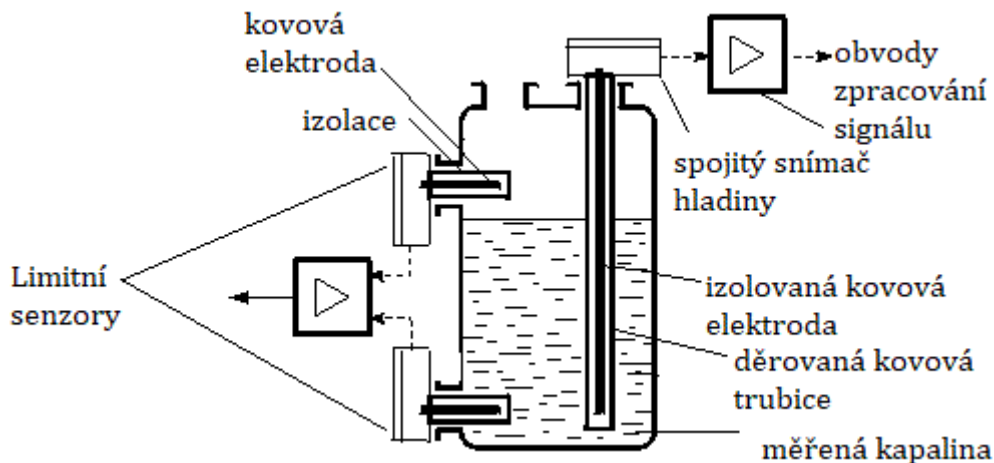
ϵ_r -relativní permitivita

ϵ_0 -permitivita

Měření zde probíhá na základě změny:

- plochy překryvu elektrod,
- vzdálenosti elektrod,
- dielektrika.

Při kontinuálním měření hladiny potřebujeme dvě elektrody. Jednu může suplovat stěna nádrže, pokud má vhodné parametry. V případě, že nemá, tak se jako elektroda používá děrovaná kovová trubice, uvnitř které se nachází teflonem izolovaná druhá elektroda a jejich přerytí se mění podle míry zaplavení. [14]

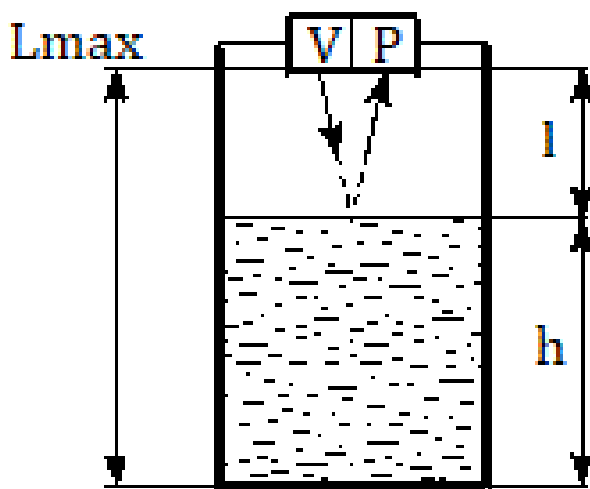


Obr. 7 -Měření hladiny kapacitními senzory. Převzato a upraveno [14]

U kapacitních senzorů bývá často kritickým prvkem spojovací vedení mezi samotným čidlem a vyhodnocovacími obvody (můstkové, rezonanční) a proto dnes již většinou výrobci umísťují vyhodnocovací obvody přímo do samotného senzoru. [14]

2.2.1.3 Ultrazvukové měření hladiny

Ultrazvukové senzory pracují na principu emise zvukové vlny a měření doby do příjmu vlny odražené od překážky (hladiny). Používají se nejčastěji pro kontinuální měření hladiny v nádržích, kde nepanují extrémní podmínky (například horké chemikálie). Při jejich použití si také musíme dát pozor na přítomnost dalších objektů v nádrži (míchadla, vnitřní kostra), nebo zvlnění měřené hladiny, které mohou dávat falešné výsledky. Jelikož rychlost šíření ultrazvuku v médiu závisí na přítomných podmínkách, používá se při potřebě přesného měření nejrůznějších korekcí, jako například korekce dle teploty prostředí a hustoty.



Obr. 8 - Měření hladiny ultrazvukovým senzorem [14]

$$l = c * \frac{t}{2} \quad (2.4)$$

Vzdálenost měřená ultrazvukovým senzorem

c-rychlost šíření ultrazvuku v daném prostředí

t-doba od emise zvukové vlny do doby přijetí odrazu

l-výsledná vzdálenost mezi hladinou a senzorem

L_{max}-vzdálenost mezi dnem a senzorem

$$h = L_{\max} - c * \frac{t}{2} \quad (2.5)$$

Výška hladiny od dna nádrže

2.2.1.4 Plovákový snímač

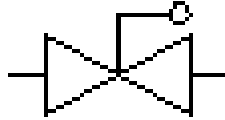
Nejčastější a nejjednodušší limitní snímače hladiny jsou snímače plovákové.

Využívají gravitačního a Archimédova zákona, kdy pohyblivá plováková část má

menší hustotu než měřená kapalina a při kontaktu se snímačem začne jeho plováčkovou část nadnášet a tím dojde ke změně její pozice a sepnutí snímače.



Obr. 9 - Plovákový snímač [31]



Obr. 10 - Technologická značka armatury ovládané plovákem [27]

2.2.2 Základní akční členy a možnosti jejich řízení

2.2.2.1 Ventily

Servo ventily

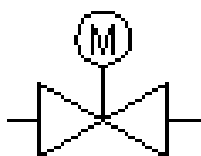
Jedná se nejčastěji o kulový ventil, který je ovládán servomotorem. Výhodou tohoto řešení je možnost spojitého nastavení ventilu, který navíc kromě nastavování pozice nespotřebovává žádnou elektrickou energii. V praxi se často používají vícecestné (2,3,4,6) ventily například pro řízení vytápění v budovách, nebo pro bazénovou technologii. V průmyslu se také těchto ventilů využívá pro řízení hydraulických systémů, a to většinou se zpětnou vazbou, čímž docílíme velice přesného dosažení požadovaného tlaku. Servoventily řídíme nejčastěji analogově- 0-10 V, nebo 4-20 mA. Zpětnou vazbu dostáváme stejným způsobem, nebo koncovými mikrospínači.



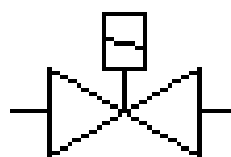
Obr. 11 - Servoventil [17]



Obr. 12 - Solenoidový ventil [16]



Obr. 14 - Technologická značka servoventilu [27]



Obr. 13 - Technologická značka solenoidu [27]

Solenoidové ventily

Solenoidové ventily pracují na principu jednoho klidového stavu a jednoho stavu pod napětím. V útroběch solenoidu se nachází cívka, která vtahuje/odpuzuje středový válec z feromagnetického materiálu, čímž otevírá/zavírá ventil.

Rozlišujeme dva základní druhy solenoidů – normálně otevřeno (NO) a normálně zavřeno (NC). Použití se pak řídí požadavky na bezpečnost a energetickou efektivitu provozu. Z principu jsou to digitálně řízené ventily s pouze dvěma stavy – plně otevřeno/zavřeno. Funkci solenoidu můžeme zpětně sledovat pomocí limitních mikrospínačů.

Klapkové ventily

U klapkových ventilů ovládá průtok pohyblivá klapka. Existují různé druhy klapek. Můžeme je dělit například podle ovládání jedním, nebo dvěma digitálními výstupy, kde na rozdíl od solenoidu zůstane dvouvstupový ventil i po odpojení napětí ve stejné poloze. Jednovstupový funguje podobně, jako solenoid, jelikož se po odpojení napětí vrací pružinou do původního stavu. U klapkových ventilů máme na

rozdíl od servo ventilů pouze zpětnou vazbu od koncových spínačů a ne průběžnou.

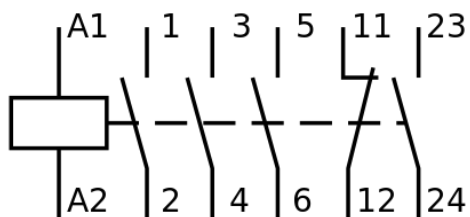


Obr. 15 - Elektricky ovládaný klapkový ventil [41]

2.2.2.2 Motory

Stykače

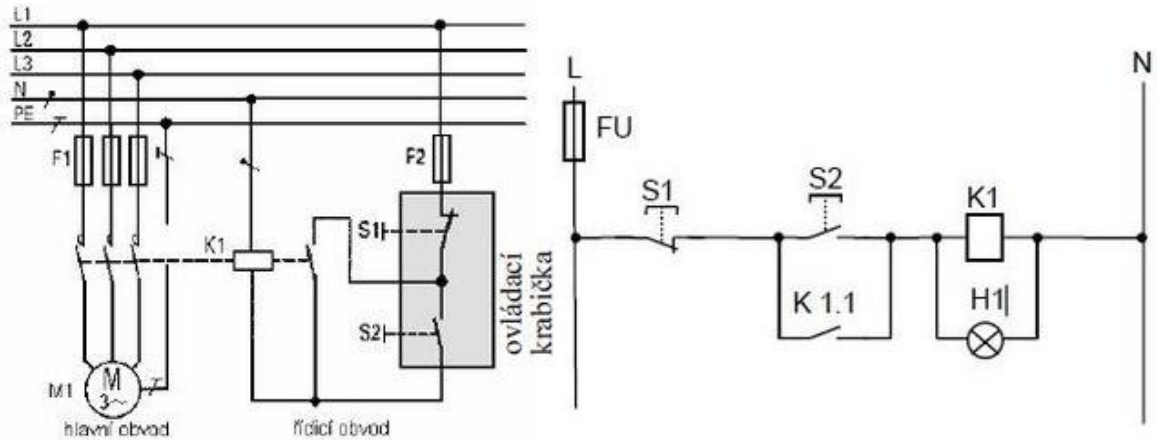
Ovládání stykačem je nejjednodušší způsob řízení motorů. Stykač je v podstatě relé dimenzované na větší zátěž. U stykačů máme obvykle výkonové kontakty a kontakty na menší proudy. Pomocné kontakty na menší proudy jsou na diagramu níže popsány čísly 12 a 24, kde 12 je NC a 24 NO. A1 a A2 je napájení spínací cívky a 1,3,5 jsou samotné spínané výkonové kontakty.



Obr. 16 - Schéma stykače [22]

Na obrázku níže vidíme typické využití pomocných kontaktů pro takzvanou funkci samodržení stykače. Pokud máme zapojený motor dle schématu níže, tak stisknutím tlačítka S2 sepne celý stykač, čímž se sepne i pomocný kontakt K1.1 a tlačítko S2 již pro chod motoru není třeba držet. Naopak pro vypnutí motoru

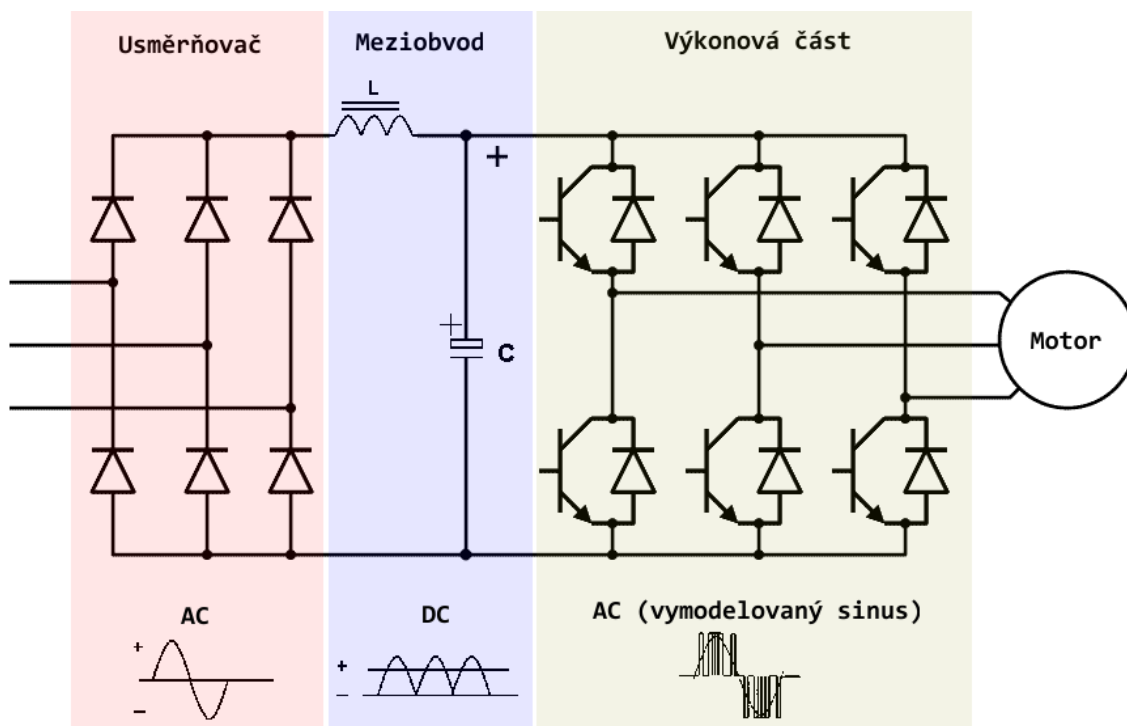
stačí stisknout rozpojovací tlačítko S1, které stykač s motorem a pomocným kontaktem vypne.



Obr. 17 - Zapojení motoru na stykač [18]

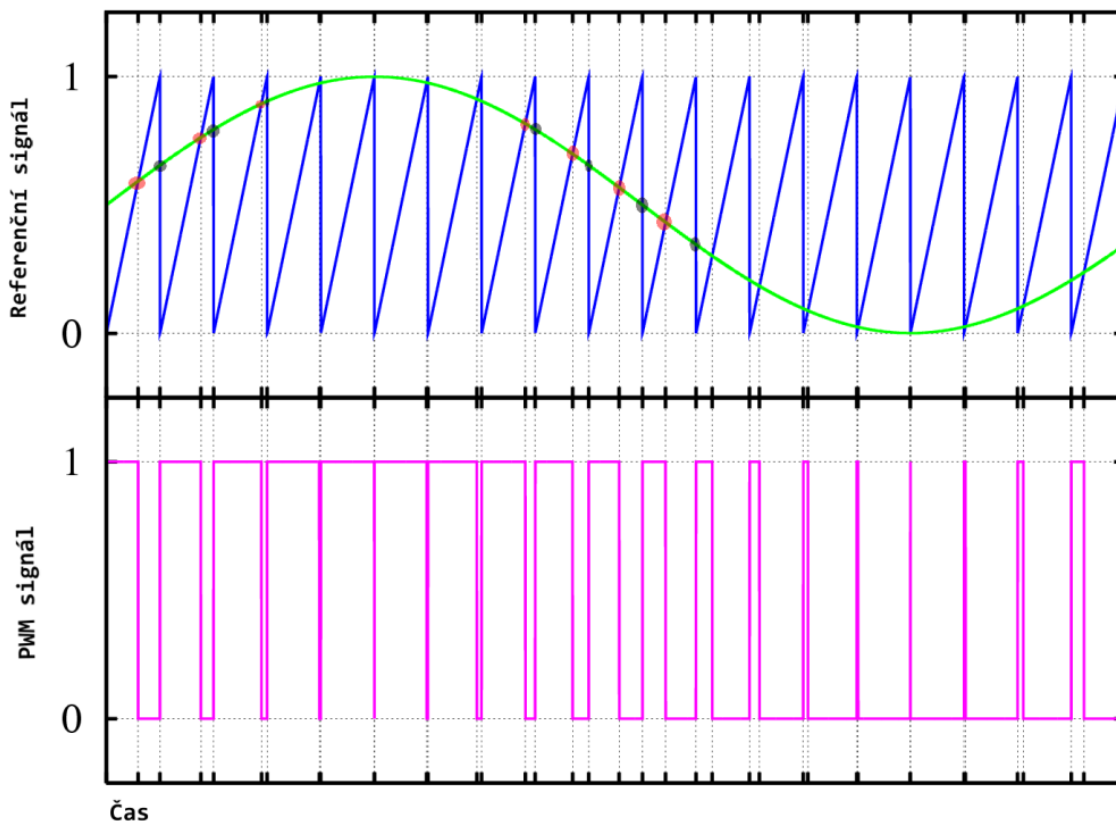
Řízení motoru frekvenčním měničem

Frekvenční měnič je zařízení, pomocí kterého regulujeme otáčky elektromotorů, a to nejčastěji třífázových. Regulace funguje na principu současné změny frekvence a napětí, kdy po dosažení požadovaných otáček pracujeme pouze se změnou frekvence. [19] Pomocí frekvenčního měniče tak můžeme plynule měnit otáčky dle potřeby, či technologické aplikace. Často frekvenční měniče používáme pro zmírnění proudových a momentových rázů při rozběhu a doběhu. Měniče můžeme dělit buď podle napájecího napětí na jednofázové a třífázové, nebo podle typu zátěže na měniče pro konstantní zátěž – High overload, nebo kvadratickou zátěž – Low overload. U měničů high overload pro kvadratickou zátěž se počítá s kvadratickým nárůstem zátěže, která je typická například pro ventilátory nebo odstředivá čerpadla. Měniče se k nadřazenému systému připojují digitálními výstupy pro základní funkce jako je start/stop, nulování poruchy, reverzace a rychlé zastavení. Pro kontinuální změnu otáček, nebo točivého momentu využíváme analogová rozhraní, popřípadě je celý měnič se všemi jeho funkcemi ovládán přes digitální sběrnici.



Obr. 18 - Blokové zapojení frekvenčního měniče [19]

Na obrázku 18. je naznačeno blokové schéma frekvenčního měniče. Růžově je naznačena usměrňovací část, která nám ze střídavého napětí udělá stejnosměrné. Z tohoto důvodu u měniče nezáleží na sledu fází, ale pouze vadí případná asymetrie. Dále je zapojen fialovou barvou značený meziobvod, ve kterém se nachází kondenzátory uchovávající energii, která je potřeba pro pokrytí proudových špiček, nebo pro uskladnění energie putující z motoru do měniče při rekuperaci. Pokud by při brždění mohlo docházet k přepětí je zařazen ještě takzvaný brzdný rezistor na kterém se přebytečná energie promění na teplo. Poslední částí je zeleně podbarvená část výkonová, ve které se nacházejí samotné tranzistory, které zajišťují spínání napětí na výstup. Celý měnič je řízen řídicími obvody, kde probíhají v reálném čase všechny výpočty potřebné pro ovládání motoru a měniče samotného, včetně bezpečnostních funkcí.



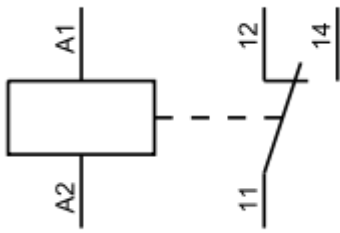
Obr. 19 - Generování PWM signálu ve frekvenčním měniči [19]

Řídící obvody generují sinusový signál o požadované frekvenci a také trojúhelníkový signál spínací frekvence tranzistorů. V průsečících obou signálů se výstupní tranzistory spínají a vypínají. Vymodelovaný PWM výstup výkonové části můžeme vidět na obrázku 19. dole. Frekvence spínání tranzistorů se, pokud je nastavitelná, volí dle technologické aplikace. Typický rozsah je jednotky, až nižší desítky kHz. Vyšší frekvence přepínání více tepelně zatěžuje měnič, kdežto nižší frekvence produkuje v motoru magnetizační hluk. [19] Inteligentní měniče umí v případě svého přetížení operativně změnit frekvenci a po ochlazení opět obnovit původní.

2.2.2.3 Topné elementy

Topná spirála

Pokud nám stačí regulace zapnuto/vypnuto (bang-bang), tak je na místě použít relé, nebo stykač, dle potřebného spínaného výkonu.



Obr. 20 - Relé [21]

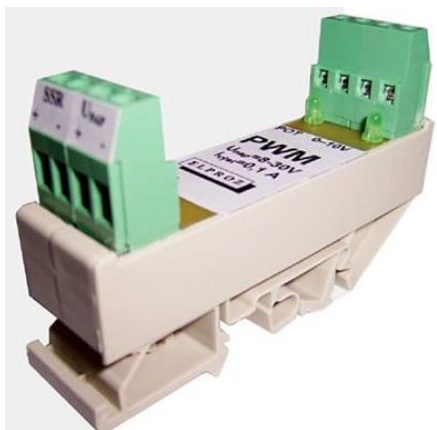
Na schématu relé se nachází spínací kontakty A1 a A2, kde po přivedení napětí na tyto kontakty se relé sepne. Kontakt 12 je NC a 14 NO.

Situace je lehce komplikovanější v případě, kdy potřebujeme výkon regulovat spojitě. V tomto případě musíme použít PWM regulátor s SSR relé. Princip fungování je podobný, jako byl nastíněn u frekvenčního měniče s tím, že můžeme použít pouze jednu fázi.



Obr. 21 - SSR relé [23]

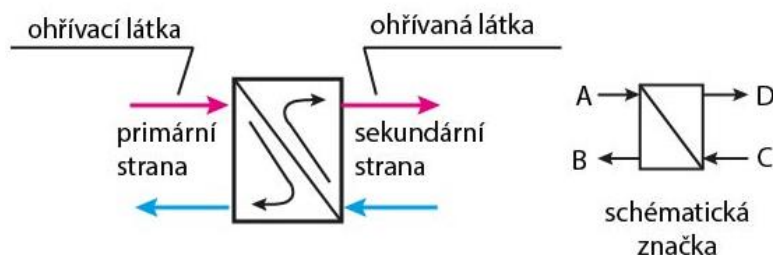
Výhodou SSR (Solid State Relay) relé je absence pohyblivých prvků, jelikož se jedná o polovodičovou součástku. SSR relé je také mnohem rychlejší v přepínání, což nám právě umožňuje jeho použití v PWM. Nevýhodou může být větší spotřeba oproti klasickému relé, nebo stykači a z toho vyplývající nutnost chlazení a odpovídající nadimenzování chladiče. Pokud digitální výstup PLC podporuje PWM, můžeme SSR relé připojit napřímo. Pokud tomu tak není, zařadíme mezi PLC a SSR relé převodník 0-10 V na PWM a řídíme z PLC analogovým výstupem.



Obr. 22 - Převodník 0 - 10 V na PWM [24]

Výměník

V případě výměníků nastávají dva typy případů. Buď máme danou teplotu topného média a potřebné vytápění technologie řídíme pomocí čerpadla a třicestného ventilu, nebo můžeme regulovat i teplotu média ve výměníku, což nám dává větší prostor pro správnou regulaci, popřípadě energetickou efektivitu. Příklad prvního typu může být datacentrum WEDOS, které chladí servery olejem (topné médium), kterým se pak přes výměník vytápí voda v plaveckém bazénu.



Obr. 24 - Značka výměníku



Obr. 23 - Deskový výměník

Příkladem druhého přístupu budiž kotel se zásobníkem teplé vody, jejíž teplotu řídíme. V zásobníku je poté uložen výměník, kterým protéká ohřívána látka, u které můžeme regulovat rychlost průtoku.

2.3 Řízení dávkových procesů (S88)

Řízením dávkových procesů a jejich standardizací se zabývá ISA-S88, v ČR norma ČSN EN 61512-1. Standard S88 popisuje systém hierarchického řízení, kde odděluje produkty, procesy a receptury. Správné pochopení a aplikace vede ke zjednodušení, standardizaci a zefektivnění výroby. Velkou výhodou je také usnadnění samotného návrhu, kdy aplikační, procesní inženýři a technologové snadno dojdou shody, pokud všichni tuto normu dodržují. Umožňuje to také

paralelizaci prací, kdy nemusí být ještě přesně daná receptura, ale již se může programovat a sestavovat daný výrobní systém. V důsledku dodržování terminologie a struktury je možné, aby na jednom zařízení pracovaly bez problému dvě firmy a nedocházelo k problémům a kolizím. Díky oddělení receptur od dalších částí programu, může vytvářet receptury i neprogramátor. Všechny tyto vlastnosti snižují časové nároky na implementaci, čímž se snižují náklady.

Hlavní body standardu S88:

- Aplikace na manuální i automatické řízení, popřípadě kombinaci obojího.
- Schopnost oddělení receptury a fyzického zařízení. Jedná se o stěžejní vlastnost tohoto standardu. Funguje to tak, že receptura řídí proces výroby a řízení poté definuje fyzické chování daného zařízení.
- Modularita procesů. Veškeré procesy a procedury dělí na menší části, což umožňuje větší flexibilitu a reprodukovatelnost jednotlivých částí jak v recepturách, tak i v zařízeních.

2.3.1 Modely standardu S88 [40]

Pokud chceme vytvořit dávkově řízený technologický proces, tak musíme začít s tvorbou modelů. Standard S88 zná tři základní modely a to procesní, fyzický a procedurální řídicí model.

2.3.1.1 Procesní model

Procesní model má čtyři úrovně dějů:

- Proces – vede k výrobě konečného množství produktu aplikací definovaných akcí na určitých zařízeních na vstupní suroviny.
- Stupeň procesu – část procesu vykonávaná nezávisle na dalších stupních. Například příprava vstupních surovin.
- Operace procesu – zde se odehrává hlavní děj daného procesu. Například ohřátí na danou teplotu po daný čas u pasterizace.
- Akce procesu – nejmenší jednotka procesů, kde kombinace jednotlivých akcí tvoří operaci.

2.3.1.2 Fyzický model

Fyzický model má úrovní sedm a je uspořádán do podobné hierarchické struktury, jako model procesní. Spojení s procesním a procedurálním řídicím modelem je ale pouze na úrovni procesního souboru, jednotky a modulu zařízení. Úrovně jsou zde následující:

- Podnik – firma řídící výrobu v jedné, nebo více lokalitách.

- Lokalita – fyzické, geografické, nebo logické uskupení tvořící jeden výrobní celek. Například areál továrny.
- Provoz – fyzické, geografické, nebo logické uskupení v rámci jedné lokality. Například jednotlivé výrobní haly v rámci areálu.
- Procesní soubor – logické seskupení zařízení potřebných k výrobě dávky. Například výrobní linka.
- Jednotka – skupina modulů zařízení a modulů řízení, ve které je realizována jedna, nebo více výrobních aktivit potřebných pro výrobu dávky. Jednotka v jeden časový okamžik zpracovává pouze jednu dávku.
- Modul zařízení – může obsahovat kromě vlastního technologického zařízení také moduly řízení.
- Modul řízení – soustava aktuátorů, senzorů a podřízených řídicích modulů.

2.3.1.3 Procedurální řídicí model

Procedurální model se vyskytuje pouze u dávkového řízení, pro který je charakteristický. U diskrétních, nebo kontinuálních technologických procesů jej nenajdeme. [40] Podobně jako procesní model, je procedurální řídicí model realizován ve čtyřech úrovních:

- Procedura – nejvyšší úroveň řízení, ve které je dána strategie výroby dávky
- Jednotková procedura – skládá se z jednotlivých operací, které v jednotce zajišťují výrobní proces.
- Operace – rozsáhlejší sekvence v procesu sestavená z jednotlivých fází. Obvykle převádí látky z jednoho fyzikálně-chemického stavu do jiného.
- Fáze – nejmenší prvek dávkového řízení. Může obsahovat jeden, nebo více příkazů a akcí.

2.3.2 Receptury

Receptura je definována jako entita, která obsahuje minimální množství údajů postačujících pro jednoznačné určení výrobních požadavků a postupů u určitého výrobku. [40] Dle normy S88 jsou definovány čtyři typy receptur:

- Generální receptura – obsahuje charakteristické informace pro zpracování produktu v obecné rovině. Například informace o nutném vybavení a surovinách.
- Místní receptura – obsahuje specifické informace pro danou lokalitu. Upřesňuje generální recepturu tak, aby byla použitelná pro zařízení na dané lokalitě fyzicky instalovaná. Můžeme také použít pro výrobu stejných výrobků na dvou různých místech jiným způsobem.

- Hlavní receptura – obsahuje informace charakteristické pro jednotku procesu. Dále upřesňuje místní recepturu o technologický postup a požadavky na řízení.
- Řídící receptura – obsahuje jednoznačný identifikátor dávky, informace o zpracování a informace generované systémem a operátorem. Dále upřesňuje hlavní recepturu o podrobné operační údaje unikátní pro každou dávku.

2.3.2.1 Součásti receptury

Každá receptura by měla mít tyto položky (u hlavní a řídicí jsou bezpodmínečně vyžadovány):

- Hlavička – administrativní údaje.
- Formule – údaje o vstupních surovinách, finálních produktech a meziproduktech receptury. Obsahuje také parametry procesu (teplota, tlak, doba reakce).
- Požadavky na zařízení – určují a omezují výběr možného zařízení pro provedení receptury.
- Procedura receptury – určuje postup provádění receptury na úrovni hlavní a řídicí receptury vyjádřenou pomocí prvků modelu procedurálního řízení.
- Další údaje – údaje o shodě s normou, údaje o balení a podobně.

2.3.3 Úrovně řízení

Norma S88 rozlišuje tři druhy řízení – základní, procedurální a koordinační.

2.3.3.1 Základní řízení

Základní řízení má za úkol dosažení a udržování určitého stavu zařízení a procesu. Základní řízení se v dávkové výrobě moc neliší od řízení kontinuálních procesů, v dávkové výrobě jsou ale obvykle větší nároky na strmost a stabilitu řízení. Na řízení dávkových procesů jsou také větší nároky ohledně flexibility – například změna chování podle operativních příkazů. Je zde také zajištěno blokování (interlocking), monitorování a obsluha výjimek, jako i cyklické diskrétní, nebo sekvenční řízení. Může zde být také ošetřeno korigování podmínek procesu. Celé řízení může být řízeno příkazy operátora, nebo procedurálním, popřípadě koordinačním řízením.

2.3.3.2 Procedurální řízení

Procedurální řízení ovládá akce týkající se zařízení tak, aby se odehrávaly v dané sekvenci. Procedurální řízení je to, co umožňuje uskutečnit dávkový proces a je tvořeno procedurálními prvky kombinovanými v hierarchické struktuře tak, aby se

provedl proces, který je definován procesním modelem. Hierarchie se skládá z procedur, jednotkových procedur, operací a fází.

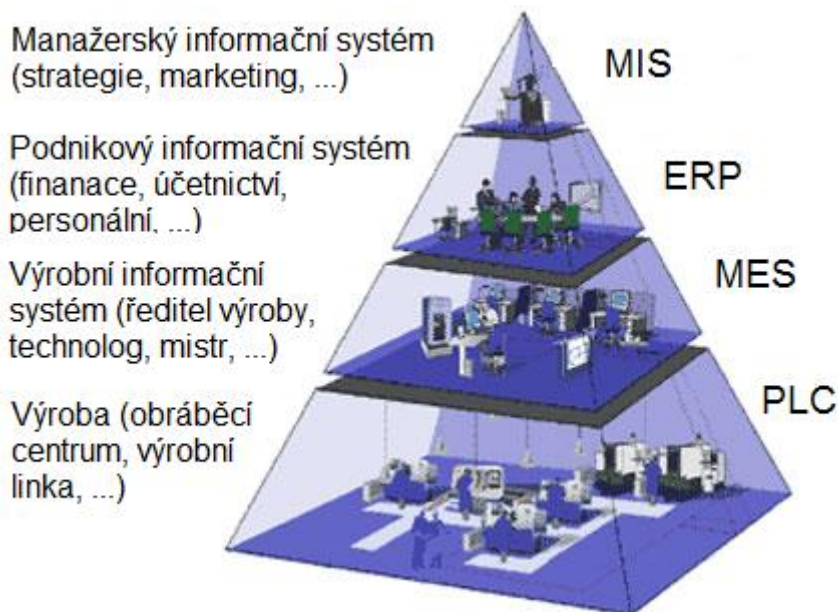
2.3.3.3 Koordinační řízení

Koordinační řízení ovládá procedurální řízení a využití jednotlivých aparátů. Na rozdíl od procedurálního řízení není strukturované podle úlohy orientované na proces. Řeší se zde například dostupnost, nebo kapacita zařízení a jeho přiřazování jednotlivým dávkám, řešení konfliktů mezi alokačními požadavky a výběr prvků procedury, které mají být realizovány.

2.4 Propojení se systémy pro řízení výroby (MES – S95)

[29]

Manufacturing Execution Systems, neboli informační systémy pro výrobu, jsou systémy tvořící vrstvu mezi podnikovými informačními systémy (ERP) a systémy pro automatizaci samotné výroby.



Obr. 25 - Hierarchie podnikových informačních systémů [29]

2.4.1 Správa výrobních zdrojů

Tato část MES systému má na starosti přidělování a sledování zdrojů a výrobních kapacit potřebných pro proces, jako například osoby, materiál, zařízení, nástroje, energie a podobně. Vychází se z aktuálního stavu a z předpokládaného vyhrazení zdrojů v budoucnu.

2.4.2 Správa výrobních postupů

Zde se provádí evidence a výměna dat s okolními systémy. Jedná se například o kusovníky materiálu, výrobní pravidla, výrobní zdroje a podobně. Všechny dostupné informace poté vytvářejí definici tvorby finálního výrobku.

2.4.3 Detailní plánování výroby

Detailní plánování výroby je stěžejní součástí výroby samotné, tak i MES systémů. V praxi se používá mnoho přístupů k plánování výroby. Například dopředné a zpětné plánování, plánování dle jednoduchých algoritmů podle priority zakázek, či složité plánování genetickými algoritmy. Na výstupu plánování je takzvaná fronta práce, která přesně určuje pořadí vykonávání jednotlivých výrobních příkazů. Fronta práce je zde připravována s důrazem na efektivitu využívání všech zdrojů, ať už se jedná o eliminaci zbytečného seřizování strojů, prostojů, nebo energetickou náročnost.

2.4.4 Dispečerské řízení

Dispečerské řízení jsou veškeré aktivity řídící tok výroby jak přiřazováním práce jednotlivým osobám a zařízením, tak také zajišťováním dostatečného množství vstupních surovin a energie. Dispečerské řízení se také zabývá sledováním aktuálního stavu výrobního procesu a operativním řešením nastalých problémů.

2.4.5 Řízení výroby

Zajišťuje provádění fronty práce ve výrobě. Úkolem řízení výroby je také kontrola zdrojů a komunikace aktuálního stavu výroby okolním systémům. Data o aktuálním stavu mohou zahrnovat například zabezpečení kontrolních kroků ve výrobě, odvody práce a podobně. Tato část MES systému je klíčová pro propojení s podnikovým informačním systémem a online kontrolou výrobního procesu.

2.4.6 Sběr dat

Zde se sbírají a ukládají výrobní data, stavy zařízení a podobné informace. Situace se zde velmi liší podle toho, zda se jedná o jednoduchou výrobu, kde se sbírá malý soubor hodnot, nebo o velkou automatizovanou linku s obrovským množstvím dat.

2.4.7 Sledování výrobků a jejich rodokmen

V tomto bloku MES systému se shromažďují a dále komunikují informace o konkrétních zdrojích použitých pro výrobu daného kusu finálního výrobku. To může zahrnovat například zaměstnance, kteří se na výrobě podíleli, materiál, který

se spotřeboval, nebo konkrétní stroje, kterými výrobek prošel. Sledování rodokmenu výrobku je důležité kvůli legislativním požadavkům v některých odvětvích a také pro řešení případných reklamací a auditů.

2.4.8 Výkonnostní analýzy [42]

Výkonnostní analýzy znamenají vyhodnocování klíčových výkonnostních ukazatelů, takzvaných KPI [29] [30]. Tyto výkonnostní ukazatele využívají výrobní podniky k vyhodnocování efektivity v jednotlivých dílčích částech výroby. Pro každý podnik a každý druh výroby, nebo zpracování jsou důležité jiné ukazatele.

2.4.8.1 OEE

Nejnámějším ukazatelem je pravděpodobně OEE [30] (overall equipment efficiency – celková efektivita řízení). Tento ukazatel nám říká, s jakou efektivitou využíváme daného stroje/zařízení/procesu. Zjednodušený výpočet OEE můžeme provést podle tohoto vzorce:

$$OEE = \text{dostupnost} * \text{výkon} * \text{úroveň kvality}$$



Obr. 26 - Kvalitativní ukazatel OEE [42]

Ztráty na produktivitě jsou způsobeny těmito faktory:

- seřizování a nastavování (plánované prostoje),
- poruchy,
- redukce rychlosti (náběhy, provozní zpomalení),
- snížení kvality (výroba zmetků).

Hodnoty faktoru OEE se standardně pohybují kolem 30–65 %. U špičkových závodů je to až 85 %. [42] Na základě neustálé potřeby pro co nejlepší hodnocení efektivity vznikly další ukazatele odvozené od OEE. Nejpoužívanější jsou popsány níže.

Použití OEE v průmyslové výrobě

Příklad použití ukazatele OEE v průmyslové výrobě s užitím technologických tanků může být například výroba zápary, která se poté destiluje na Bourbon. Postup výroby zápary je následující – smíchá se:

- žito,
- ječmen,
- rozdrčená kukuřice,
- slad.

Poté se celá směs pomalu zahřívá až na 71 °C, jelikož při této teplotě probíhá enzymatická proměna škrobů na zkvasitelné cukry. Teplotní šířka pásma, ve kterém probíhá enzymatická proměna je velice úzký. Proměna probíhá od 68 °C, ale již při 75 °C dochází k usmrcení enzymů. Po přeměně veškerých škrobů na zkvasitelné cukry se obsah tanku vypustí do kvasné nádoby. Po zkvašení je vypálen a naplněn do dubových sudů. Jednotlivé části ukazatele potom budou vypadat následovně:

- Celkový plánovaný pracovní čas – technologický tank má plánovaný nepřetržitý provoz 24/7.
- Plánovaný čas pro výrobu – po výrobě jedné dávky, která trvá 5,5 hodiny, je potřeba půl hodiny pro vyčištění tanku. Plánovaný čas výroby je tedy 22 hodin každý den.
- Skutečný čas výroby – tento vidíme jako výstup MES systému.
- Očekávaný výkon – pokud známe skutečný čas, po který byl technologický tank v provozu, můžeme jednoduše stanovit výkon (množství vyrobené zápary), který by měl vykonat.
- Skutečný výkon – skutečný výkon se od očekávaného bude lišit podle nedokonalostí ve výrobě, které zasáhly do procesu. V našem případě to bude nedokonalá regulace.
- Očekávaná kvalita – skutečný výkon opět vidíme v MES systému a dle toho očekáváme určitou kvalitu (procento přeměny škrobů na zkvasitelné cukry).
- Skutečná kvalita – pokud budeme mít na regulaci velký překmit (nad 75 °C), dojde k usmrcení enzymů a nedostatečné přeměně škrobů na cukry, což zapříčiní nižší výnosnost bourbonu.

2.4.8.2 TEEP Total effective equipment productivity [44]

TEEP, česky také celková efektivní výkonnost zařízení, je výkonový ukazatel, který ukazuje celkovou výkonnost na základě času, po který bylo zařízení přítomno ve výrobě. Jelikož je většinou přítomno neustále, tak nám ukazuje, jak dobře je zařízení využito. TEEP je jednou ze tří metrik ze kterých se skládá OEE. Jedná se o nejpoužívanější odvozený ukazatel, protože díky němu jednoduše zjistíme efektivitu zařízení vztahenou ke kalendářnímu času.

(2.6)

$$TEEP = \text{Plánované prostoje} * \text{dostupnost zařízení} * \text{výkon} * \text{úroveň kvality}$$

Nebo také:

(2.7)

$$TEEP = \text{Plánované prostoje} * OEE$$

Dostupnost

V dostupnosti zařízení započítáváme poměrný rozdíl mezi plánovaným a skutečným operačním časem.

(2.8)

$$\text{Dostupnost} = \frac{\text{Skutečný operační čas}}{\text{Plánovaný operační čas}}$$

Výkon

Ukazatel výkonu nám říká, jaký je poměr mezi plánovanou a skutečnou rychlostí výroby tím, že zachycuje ztráty rychlosti způsobené různými nedokonalostmi ve výrobě.

(2.9)

$$\text{Výkon} = \frac{\text{Skutečný výkon}}{\text{Očekávaný výkon}}$$

Kvalita

Kvalitativní ukazatel říká, jaký je podíl mezi zmetky a dobrými produkty.

(2.10)

$$\text{Kvalita} = \frac{\text{Počet dobrých produktů}}{\text{Celkový počet výrobků}}$$

2.4.8.3 PEE Production equipment efficiency [43]

PEE, česky také efektivita výrobního zařízení, je také jedním z odvozených ukazatelů. Na rozdíl od OEE, mají jednotlivé dílčí ukazatele danou váhu.

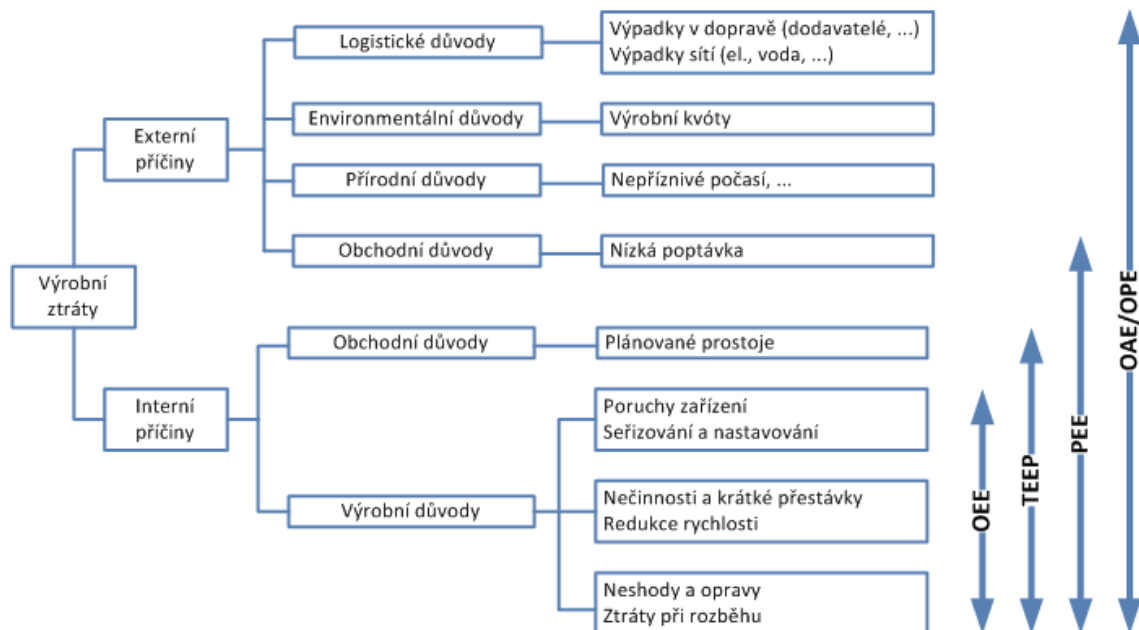
(2.11)

$$PEE = (dostupnost)^{k1} * (výkon)^{k2} * (kvalita)^{k3}$$

k1, k2, k3 – váhové koeficienty (0-1), $\Sigma k_i=1$

2.4.8.4 OPE Overall production effectiveness [43]

OPE, česky také celková efektivita produkce, je ukazatel, vycházející z OEE, ale upravuje se pro každé odvětví tak, aby přesně vyhovoval jejím požadavkům, a proto se jeho definice liší. Ze všech ukazatelů zahrnuje největší spektrum možných ztrát v celém výrobním procesu. [43]

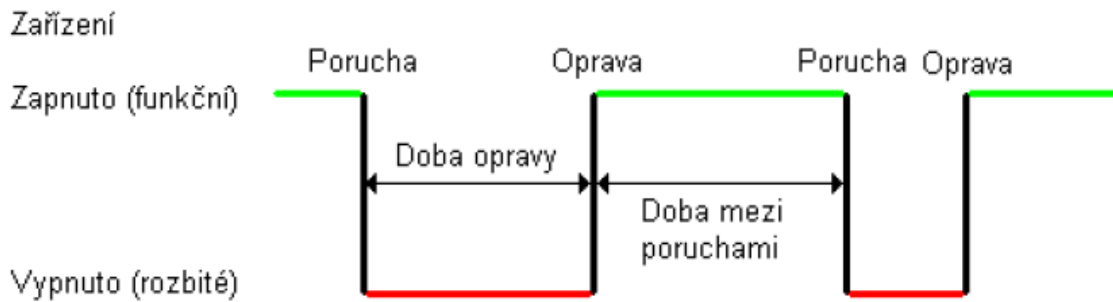


Obr. 27 - Porovnání ukazatelů KPI v MES systému [43]

2.4.8.5 MTBF – Mean time between failures

MTBF – česky také střední doba mezi poruchami je aritmetický průměr času mezi jednotlivými poruchami. Vypočítat střední dobu mezi poruchami můžeme tímto vzorcem:

$$MTBF = \frac{\sum \text{časů mezi poruchami}}{\text{počet poruch}} \quad (2.12)$$



Obr. 28 - Kvalitativní ukazatele MTBF a MTTR [42]

2.4.8.6 MTTR – Mean time to repair

MTTR – česky také střední doba opravy je aritmetický průměr času potřebného k uvedení porouchaného stroje do provozuschopného stavu. Střední dobu opravy můžeme vypočítat podle následujícího vzorce:

$$MTTR = \frac{\sum \text{času opravy}}{\text{počet poruch}} \quad (2.13)$$

3 PROGRAMOVACÍ JAZYKY

V TIA Portálu 14 můžeme programovat v jazycích Statement list (STL), Structure control language (SCL), Ladder diagram (LAD), Function block diagram (FDB) a GRAPH. Každý z nich se hodí pro konkrétní typ úlohy. V semestrální práci jsem použil jazyky STL, GRAPH (Grafcet) a LAD. Naproti tomu v bakalářské práci již byl použit pouze jazyk SCL. [2]

3.1 STL

Statement list-můžeme přeložit jako seznam instrukcí. Jedná se o textový jazyk přizpůsobený pro použití v PLC. Používá mnemotechnické instrukce jako například A-AND, L-LOAD atd. Jedná se o nejnižší úroveň programování, ale přesto můžeme u mnoha aplikací napsat v tom jazyce kód nejrychleji a nejefektivněji. Jedná se o jazyk normy IEC 61131-3 a standartu IEC 60848. [1] [2]

3.2 GRAFCET

Grafcet je grafický programovací jazyk, který je nejvhodnější pro programování sekvenčních dějů, jelikož zde definujeme pouze úkony v aktuálním stavu a podmínky pro přechod do stavu jiného. Po vykonání posledního stavu se program opět vrací do stavu prvního a začíná znova. Princip fungování vychází z Petriho sítí. Jedná se o jazyk normy IEC 61131-3 a standartu IEC 60848. [3] [1] [11]

3.3 LAD

Ladder diagram-takzvané spínačové zobrazení vychází z principu elektrických schémat v liniovém zapojení, proto je program v tomto jazyku je asi nejlépe čitelný a pochopitelný pro uživatele, který se s PLC ještě nesetkal. Pro složité programy s mnoha větvemi ale není tolik vhodný, jelikož se program stává nepřehledným. Jedná se o jazyk normy IEC 61131-3 a standartu IEC 60848. [1] [11]

3.4 SCL

Pokročilý strukturovaný jazyk na bázi Pascalu. Program píšeme ve formě strukturovaného textu s použitím metod vyšších programovacích jazyků jako například deklarací, podmínek, knihoven a dalších. Na druhou stranu používáme také prostředky jazyků LAD A STL jako například vstupy/výstupy, časovače, bitová paměť a další.

3.4.1 Vývoj jazyka SCL

Programovací jazyk Structured text byl vydán roku 1993 v normě IEC 61131-3, jejíž cílem bylo standardizovat programovací jazyky používané pro programování PLC. Structured text jako velmi intuitivní a výkonný programovací jazyk využívá typických operací jako je: logické větvení, smyčky a další. [8] Programy v SCL mohou být psány v jakémkoli textovém editoru, a protože jejich jasná a přehledná syntaxe připomíná věty, tak je následné testování programu a celkové porozumění kódu velmi snadné. Díky tomu je tento jazyk vhodný pro komplexnější úlohy, kde by při použití jiných jazyků utrpěla celková přehlednost kódu, nebo by vůbec nebylo možné zadaný úkol efektivně realizovat.

Jak se postupně celý obor programování PLC přirozeně vyvíjí, tak je zde čím dál větší důraz na multiplatformnost použitých programovacích jazyků a hardwaru. Z tohoto důvodu nabývá jazyk SCL na čím dál větší oblibě, jelikož drtivá většina výrobců automatizační techniky tento jazyk podporuje. a navíc není problém například provádět online změny v programu bez zastavení PLC. [10] Díky tomu jsou firmy schopny v případě potřeby měnit programovatelné automaty od jednoho výrobce za programovatelné automaty druhého výrobce znatelně snáz, popřípadě doladovat program za chodu linky. [8]

Z podstaty SCL plyne, že programátoři, kteří se již setkali s jazyky jako například C, Python atd. mají předpoklad se naučit SCL velice rychle. Navíc v TIA Portal 14 je mnoho již předpřipravených funkcí zajišťujících například komunikaci, nebo matematické funkce. Je zde také možné propojovat funkční bloky napsané

v různých programovacích jazycích, což dává programátorovi velkou volnost a není tak nucen držet se jednoho jazyku.

3.4.2 Deklarace proměnných

SCL umožňuje deklaraci proměnných, podobně jako například v jazyce C, nebo symboly v jazyce LAD. Všechny jména proměnných musí začínat písmenem, ale další symboly již mohou obsahovat i čísla a některé speciální znaky. [8] Velice důležité je to, že Structured text není takzvaně „case - sensitive“, což znamená, že nerozeznává velké písmeno od malého. V rámci zachování programátorských konvencí, je ale dobré se držet zažitých pravidel pro pojmenovávání funkcí a proměnných. Proměnné můžeme deklarovat pomocí následujících klíčových slov:

- **VAR/END_VAR** Začátek/konec deklarace proměnných
- **VAR_GLOBAL** Globální proměnná
- **VAR_INPUT** Vstupní proměnná
- **VAR_OUTPUT** Výstupní proměnná
- **VAR_IN_OUT** Vstupně výstupní proměnná
- **VAR_ACCESS** Přímý přístup do paměti
- **VAR_EXTERNAL** Tímto klíčovým slovem se odkazujeme na globální proměnnou. Toto většinou není potřeba, jelikož to za programátora udělá překladač. Nejčastější použití je při odkazu například do knihovny nebo mimo aktuální rozsah překladu.
- **VAR_TEMP** Dočasná proměnná
- **AT** Pomocí tohoto klíčového slova můžeme definovat, kde přesně v paměti se bude proměnná nacházet
- **CONSTANT** Neměnná konstanta
- **RETAIN** Proměnná, která zůstane v PLC uložena i po jeho vypnutí a vypnutí napájení.

[8]

3.4.3 FOR cyklus

Typická konstrukce FOR cyklu vypadá následovně:

```
”  
FOR count := initial_value TO final_value BY increment DO  
    <statement>;  
END_FOR;
```

„[6]

Initial_value-východzí hodnota od které počítáme FOR cyklus

Final_value – hodnota do které běží FOR cyklus

Increment – velikost kroku

<statement> - vykonávaný kód

Protože FOR cyklus má omezený, dopředu daný počet opakování, tak se používá tam, kde přesně takhle vlastnost může být výhodná, či přímo klíčová. Například v modelovém příkladu, kde máme za úkol namočit výrobek do kyseliny 10krát, tak přesně zde bude FOR cyklus výhodný.

3.4.4 WHILE cyklus

Oproti FOR cyklu, má WHILE cyklus formálně jednodušší zápis:

”

```
WHILE [boolean expression] DO
<statement>;
END_WHILE;
```

„[7]

Boolean expression – True/false výraz při jehož splnění se bude cyklus provádět

<statement> - Samotný kód cyklu

WHILE cyklus se oproti FOR cyklu používá tam, kde je žádoucí, aby se výraz v cyklu opakovat až do splnění zadané podmínky. Pokud si zde opět pomůžeme modelovým příkladem z průmyslu, tak to bude tento: máme za úkol vkládat opakovaně odporový drát do lázně s kyselinou, dokud jeho odpor nedosáhne požadované úrovně. Z důvodu změny teploty, koncentrace kyseliny a dalších faktorů nebude potřebný počet cyklů vždy stejný, proto zde bude WHILE cyklus optimálně použit.

3.4.5 REPEAT cyklus

Posledním standartním typem cyklu je REPEAT cyklus:

”

```
REPEAT
<statement>;
UNTIL [boolean expression]
END_REPEAT
```

„[5]

<statement> - Samotný kód cyklu

Boolean expression – True/false výraz při jehož splnění se cyklus **přeruš**

REPEAT cyklus funguje přesně opačným způsobem než WHILE cyklus. Všimněme si zde toho, že podmínka je zde až za prováděným kódem, a proto se kód za všech okolností provede minimálně jednou. Modelovým příkladem by zde mohlo být načítání stavu zásobníku, dokud nenačteme nenulové, kladné, celé číslo. V tomto případě musíme provést načtení minimálně jednou, a právě pro tento případ bude REPEAT cyklus ideálním kandidátem.

3.4.6 Použití EXIT v SCL

Pokud potřebujeme ukončit cyklus při určité podmínce i bez dokončení cyklu dle původně zadané podmínky, tak se nám výborně hodí příkaz EXIT, který zapouzdříme do IF podmínky.

”

```
WHILE current_level <= setpoint DO
Valve_Open := TRUE;
IF Sensor_Fault = TRUE THEN
Valve_Open := FALSE;
EXIT;
END_IF;
END_WHILE;
```

„[8]

Na modelovém příkladu výše je znázorněn kód pro WHILE cyklus, který má za úkol držet otevřený ventil, dokud není dosaženo požadované hladiny, nebo nezaznamenané chybu senzoru. V případě chyby senzoru ventil okamžitě zavřeme a vystoupíme z WHILE cyklu.

3.4.7 Použití RETURN v SCL

RETURN se na rozdíl od EXIT používá uvnitř funkce, nebo funkčního bloku. [8] Používáme ho obvykle, pokud potřebujeme z funkce vystoupit před jejím koncem a obvykle je, jako EXIT, zapouzdřený v IF podmínce.

”

```
FUNCTION_BLOCK Valve_Control
VAR_INPUT
Tank_Level, Max_Level : REAL;
END_VAR
VAR_OUTPUT
Valve_Open : BOOL;
Out_Msg : STRING;
END_VAR
IF Tank_Level > Max_Level THEN
```

```

Valve_Open := FALSE;
Out_Msg := 'Tank Level is Full';
RETURN;
ELSE
Valve_Open := TRUE;
END_IF;
IF Tank_Level > Max_Level * 0.8 THEN
Out_Msg := 'Tank Level is High';
ELSE
Out_Msg := 'Tank Level is Normal';
END_IF;
END_FUNCTION_BLOCK

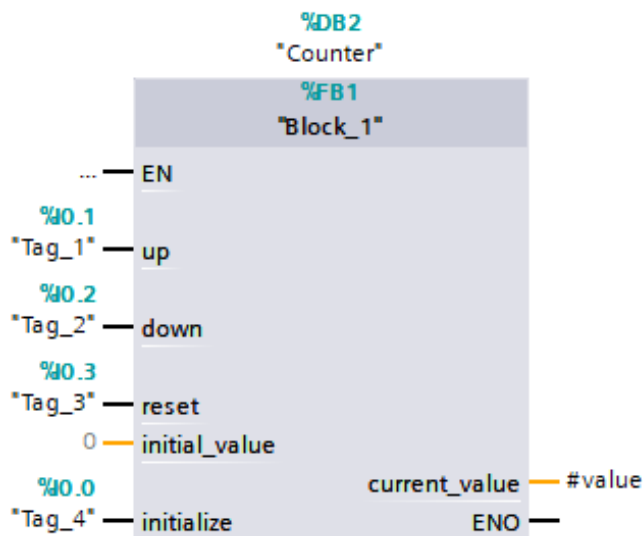
```

„[8]

Na modelovém programu výše je napsán funkční blok, který otevírá a zavírá ventil podle stavu hladiny v nádrži. Pokud je nádrž naplněna, tak se ventil uzavře, je vyhozena hláška o naplnění nádrže a blok je ukončen. V případě, že se tak nestane, tak je stav hladiny dále zhodnocen a je vyhozena odpovídající hláška.

3.4.8 Čítač v jazyku SCL

V příkladu níže je ukázka použití čítače v jazyku SCL. Jedná se o modelový příklad, kdy pomocí čítače počítáme počet vyrobených výrobků na dvou výrobních linkách. PLC je připojeno na výrobní linky, které dávají při projetí vyrobeného výrobku linkou číslo 1. signál logická 1 na vstup I.0.1, naopak při zaznamenání výrobku linkou číslo 2. vidíme logickou 1 na vstupu I0.2. Můžeme tak v proměnné *value* sledovat, zdali obě linky mají stejnou výkonnost. První linka totiž k číslu v čítači jedničku přičítá a druhá odečítá. Vstup I0.3 slouží k vynulování čítače po proběhlé várce a logická 1 na vstupu I0.4 nám čítač znovu připraví nahráním startovací hodnoty. Na obrázku 29. můžeme vidět funkci samotného čítače přidanou do bloku OB1, který se spouští automaticky při zapnutí PLC.



Obr. 29 - Hotový blok Counter

Na obrázku 30. jsou proměnné samotné funkce čítač napsané v programovacím jazyku SCL. Jelikož v našem příkladu potřebujeme hodnotu přičítat i odečítat je použit čítač CTUD(Counter-Up/Down). Pokud se to hodí naší aplikaci, můžeme také použít čistě přičítací, nebo odečítací čítače, jejichž syntaxe je podobná.

Block_1				
	Name	Data type	Default value	Retain
1	Input			
2	up	Bool	false	Non-retain
3	down	Bool	false	Non-retain
4	reset	Bool	false	Non-retain
5	initial_value	Int	0	Non-retain
6	initialize	Bool	false	Non-retain
7	<Add new>			
8	Output			
9	current_value	Int	0	Non-retain

Obr. 30 - Proměnné bloku Counter

```
"Counter1".CTUD(CU:=#up,
                CD:=#down,
                R:=#reset,
                LD:=#initialize,
                PV:=#initial_value,
                CV=>#current_value );
```

3.4.9 Časovač v jazyku SCL

Modelový příklad časovače ukazuje, jakým způsobem můžeme pomocí jednoho časovače řídit více událostí. V našem případě se jedná o postupné zapínání třech LED diod, každé po pěti sekundách. Výstup LED diod je ve formě struktury, na kterou už můžeme přímo navázat odpovídající fyzické výstupy. V příkladu vidíme použitý časovač TONR, jehož vstupy jsou IN-povolovací bit časovače, RESET-resetování časovače, PT-čas, po jehož uplynutí se výstup Q překlápí do stavu logická jedna a nakonec ET-aktuální uběhlý čas od spuštění časovače.

Pod kódem pro samotný časovač se nachází povel pro restartování časovače po uplynutí 4*5 sekund. Toto nám ve spolupráci s následující konstrukcí CASE zajistí, že LED diody se budou postupně rozsvěcovat s časovým intervalem 5 sekund a následně zhasnou a celý proces se bude opakovat.

Následně přepočítáváme čas na hodnotu int, kterou dělíme pěti sekundami (čas mezi sepnutím dvou LED diod). Funkce FLOOR nám zajistí zaokrouhlení hodnoty na celé číslo směrem dolů a jelikož tato funkce má na vstupu typ real, musíme podělenou hodnotu nejdříve na tento typ převést.

Ihned poté následuje konstrukce CASE zajišťující samotné zapínání a vypínání podle čísla přítomného v proměnné led.

```
"IEC_Timer_0_DB".TONR(IN:=#enable,
    R:=#reset,
    PT:=#time2go,
    Q=>#out,
    ET=>#time);
IF #time > (#time2go * 4) THEN
    #reset := 1;
;
END_IF;

#led := FLOOR(INT_TO_REAL(TIME_TO_INT(#time) / TIME_TO_INT(T#5S)));
CASE #led OF
    1:
        #output_led."1" := 1;
        #output_led."2" := 0;
        #output_led."3" := 0;
;
    2:
```

```

#output_led."1" := 1;
#output_led."2" := 1;
#output_led."3" := 0;
;
3:
#output_led."1" := 1;
#output_led."2" := 1;
#output_led."3" := 1;
;
;
ELSE
#output_led."1" := 0;
#output_led."2" := 0;
#output_led."3" := 0;
;
END_CASE;

```

timer				
	Name	Data type	Default value	Retain
1	▼ Input			
2	enable	Bool	false	Non-retain
3	time2go	Time	T#0ms	Non-retain
4	▼ Output			
5	out	Bool	false	Non-retain
6	time	Time	T#0ms	Non-retain
7	▼ output_led	Struct		Non-retain
8	1	Bool	false	Non-retain
9	2	Bool	false	Non-retain
10	3	Bool	false	Non-retain
11	▼ InOut			
12	reset	Bool	false	Non-retain
13	▼ Static			
14	<Add new>			
15	▼ Temp			
16	led	Int		

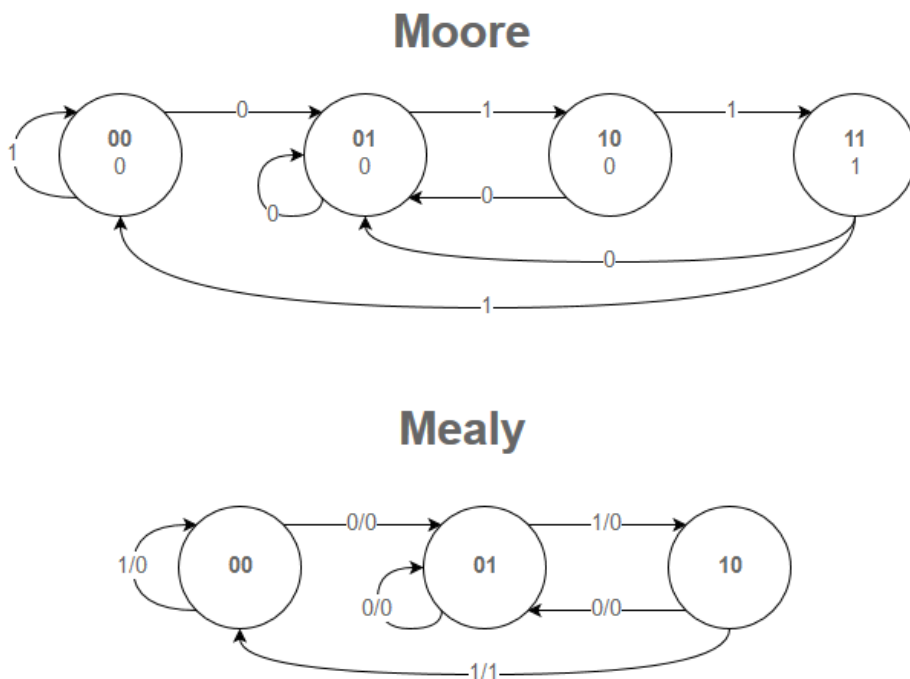
Obr. 31 - Proměnné bloku Timer

3.4.10 Stavový automat v jazyku SCL

Stavový automat je teoretický model jednoduchého počítače, který může nabývat předem daný konečný počet jednotlivých stavů. Stavové automaty se vyznačují, že

jediná použitá paměť je jejich aktuální stav. Známe dva typy základních stavových automatů, a to Mealyho a Moorův. [12]

Na obrázku 32. shora vidíme diagram Moorova automatu. Výstupy jsou určeny pouze vnitřním stavem a nezávisí na vstupním stavu. Nad šipkou vidíme, do jakého stavu vedou jednotlivé proměnné. Reakce na vstup je u Moorova automatu vidět až v následujícím kroku.



Obr. 32 - Stavový automat Moore/Mealy[9]

Na obrázku 32. dole vidíme diagram Mealyho automatu. Nad šipkou vidíme, do jakého stavu vedou jednotlivé kombinace proměnných. Mealyho reakce na vstup je okamžitá.

3.4.10.1 Modelový příklad

V modelovém příkladu vidíme stavový automat, který detekuje přítomnost dvou jedniček po sobě. Pokud kdykoliv v průběhu procesu přijde na vstup nula, stavový automat se vrací do výchozího stavu. Pokud již přišly dvě jedničky po sobě a přijde další, tak stavový automat zůstává ve stavu nula.

```

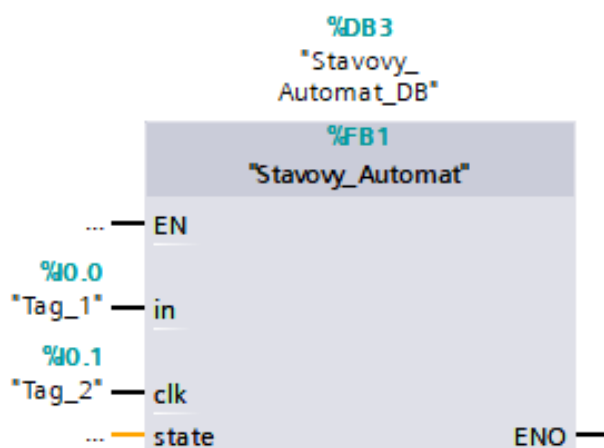
CASE #state OF
  0: IF (#in=1) THEN #state := 1;
      ;
      END_IF;
      ;
  1: IF (#in = 1) THEN #state := 2;
      ;
      ELSE #state:=0;
      END_IF;
      ;
  2: IF (#in = 1) THEN #state := 2;
      ;
      ELSE
        #state := 0;
      END_IF;
      ;
ELSE
  #state := 0; // Statement section error=reset to state 0
  ;
END_CASE;

```

Stavovy_Automat						
	Name	Data type	Default value	Retain	Accessible f...	Writa...
1	Input				<input type="checkbox"/>	<input type="checkbox"/>
2	in	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	clk	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Output				<input type="checkbox"/>	<input type="checkbox"/>
5	output	Bool	false	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	InOut				<input type="checkbox"/>	<input type="checkbox"/>
7	state	Int	0	Non-retain	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Obr. 33 - Proměnné bloku stavový automat

Na obrázku 33. se nachází seznam proměnných funkce Stavovy_automat v SCL, reprezentující blok stavového automatu. Celý stavový automat je zapouzdřený v konstrukci CASE, která zajišťuje přepínání stavů podle proměnné state reprezentující aktuální stav. Na začátku každého stavu sledujeme náběžnou hranu signálu clk a pokud ji detekujeme a zároveň je na prvním vstupu jednička, tak se posouváme do dalšího stavu.



Obr. 34 - Hotový blok stavový automat

Na obrázku 34. výše je použití bloku stavového automatu v hlavním bloku OB1. Na oranžově znázorněném pinu state je vyveden aktuální stav stavového automatu, který vychází z přivedených signálů I0.0(in) a I0.1(clk).

3.4.11 Komunikace MODBUS v jazyku SCL

V PLC Siemens můžeme dobře využít standart průmyslové komunikace MODBUS přes rozhraní PROFINET. V TIA Portal 14 na to jsou určeny předpřipravené funkce MB_SERVER a MB_CLIENT.

3.4.11.1 Klient

Na kódu níže se nachází funkce MODBUS klienta napsaná v SCL. Pro její obsluhu je výhodné použít připravenou datovou strukturu MB_CLIENT. Skládá se ze základních proměnných potřebných pro jeho běh a to jsou:

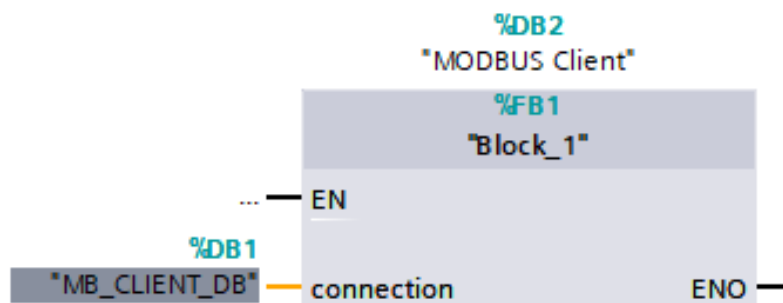
- REQ-Je vyžadováno spojení.
- DISCONNECT-Pokud je v logické 1- spojení se přeruší.
- MB_DATA_ADDR-Kombinace této proměnné a proměnných MB_DATA_LEN a MB_MODE určuje, jaký kód MODBUS funkce použije pro přenos.
- MB_MODE – Mód přenosu.
- MB_DATA_LEN-Délka dat.
- DONE-Přenos hotov.
- BUSY-Přenos probíhá.
- ERROR-Kód chyby, pokud se vyskytla.
- STATUS-Aktuální status MODBUS klienta.

- MB_DATA_PTR-Pointer ukazující na data, která se mají přijmout, nebo odeslat.
- CONNECT-Údaje určující, ke kterému hostiteli se má připojit.

```
"MB_CLIENT_DB" (REQ:=#connection.REQ,
                DISCONNECT:=#connection.DISCONNECT,
                MB_MODE:=#connection.MB_MODE,
                MB_DATA_ADDR:=#connection.MB_DATA_ADDR,
                MB_DATA_LEN:=#connection.MB_DATA_LEN,
                DONE=>#connection.DONE,
                BUSY=>#connection.BUSY,
                ERROR=>#connection.ERROR,
                STATUS=>#connection.STATUS,
                MB_DATA_PTR:=P#M1000.0 WORD 500,
                CONNECT:=#connection.TCON.ID);
```

1	▼	Input				
2	■	<Add new>				
3	▼	Output				
4	■	<Add new>				
5	▼	InOut				
6	■	▼ connection	MB_CLIENT			
7	■	▼ Input				
8	■	REQ	Bool	Activates the requested transmission if T		
9	■	DISCONNECT	Bool	Initiates a disconnect operation		
10	■	MB_MODE	USInt	Specifies the type of request: read, write		
11	■	MB_DATA_ADDR	UDInt	Specifies the starting address of the data		
12	■	MB_DATA_LEN	UInt	Specifies the number of bits or words to b		
13	■	▼ Output				
14	■	DONE	Bool	Instruction finished without error		
15	■	BUSY	Bool	Modbus transaction in progress		
16	■	ERROR	Bool	Instruction finished with error		
17	■	STATUS	Word	Detailed error information		
18	■	▼ InOut				
19	■	MB_DATA_PTR	Variant	Reference to the local source or destinati		
20	■	CONNECT	Variant	Reference to the connection parameters		
21	■	▶ Static				
22	▼	Static				

Obr. 35 - Proměnné bloku MODBUS client



Obr. 36 - Hotový blok MODBUS client

3.4.11.2 Server

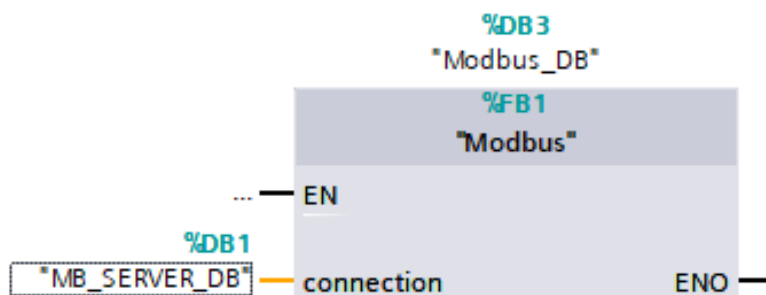
Na obrázku 38. níže se nachází funkce MODBUS serveru napsaná v SCL. Pro její obsluhu je výhodné použít připravenou datovou strukturu MB_SERVER. Skládá se ze základních proměnných potřebných pro jeho běh a to jsou:

- DISCONNECT-v případě přivedení logické jedničky server odpojí veškerou probíhající komunikaci a další již nenaváže.
- NDR-Příznak nově zapsaných dat.
- DR-Příznak nově čtených dat.
- ERROR-Chyba v nastavení, či komunikaci.
- STATUS-Aktuální status MODBUS serveru.
- MB_HOLD_REG-Datový registr, do kterého má daný server přístup. Můžeme nastavit jednotlivé části paměti, nebo celou paměť.
- CONNECT-Detaily nastavení komunikace-Můžeme například přijímat komunikaci pouze z určité IP adresy a tím zvýšit bezpečnost.

Modbus				
	Name	Data type	Default value	Retain
5	▼ InOut			
6	■ ▼ connection	MB_SERVER		
7	■ ▼ Input			
8	■ DISCONNECT	Bool		Non-retain
9	■ ▼ Output			
10	■ NDR	Bool		Non-retain
11	■ DR	Bool		Non-retain
12	■ ERROR	Bool		Non-retain
13	■ STATUS	Word		Non-retain
14	■ ▼ InOut			
15	■ MB_HOLD_REG	Variant		
16	■ CONNECT	Variant		
17	■ ▼ Static			
18	■ ▶ TCON	TCON		
19	■ ▶ TSEND	TSEND		

Obr. 37 - Proměnné bloku MODBUS server

```
"MB_SERVER_DB" (DISCONNECT:=#connection.DISCONNECT,
                NDR=>#connection.NDR,
                DR=>#connection.DR,
                ERROR=>#connection.ERROR,
                STATUS=>#connection.STATUS,
                MB_HOLD_REG:="DB1".MODBUS,
                CONNECT:=#connection.TCON.ID);
```



Obr. 38 - Hotový blok MODBUS server

4 ÚLOHA TANKY

V průmyslu se většina operací odehrává v technologických tancích. Ať už se jedná o potravinářství, farmaceutický průmysl, nebo chemickou výrobu, ve všech těchto případech firmy těží ze snadné sanitace, zdravotní nezávadnosti, univerzálnosti a chemické odolnosti nerezových technologických tanků. Většina tanků má vypouštěcí a napouštěcí ventily, otvory pro čidla a sanitaci, popřípadě míchadla, ohřev a chlazení.



Obr. 39 - Technologický tank [32]

Technologický tank v úloze se dá použít k rozličným účelům ve výrobě. Nejčastějším použitím v potravinářství bude pravděpodobně pasterace.

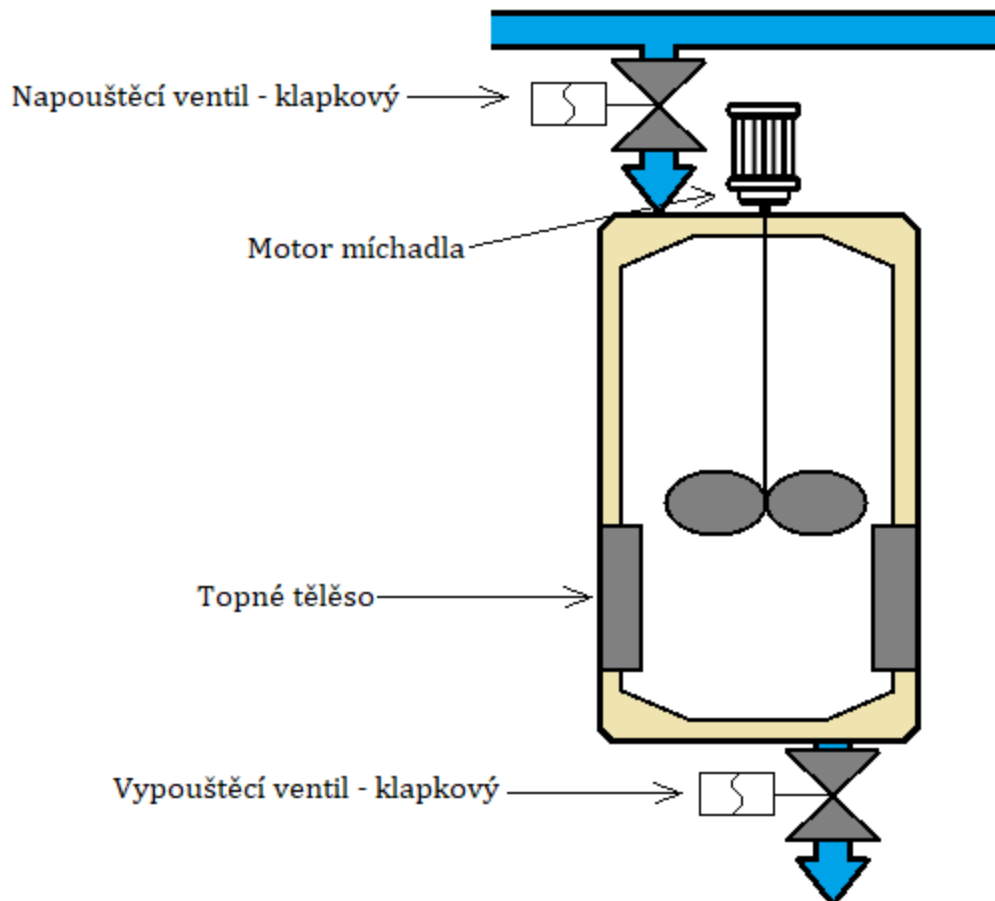
4.1 Pasterace

Pasterace, nebo také pasterizace je proces, kterého se využívá v potravinářském průmyslu, kde ve výrobcích ničí mikroorganismy šířící závažná onemocnění a snižující trvanlivost. [33] Tento proces je založen na poznatcích Louise Pasteura z 19. století, který zjistil, že krátkodobým zvýšením teploty zničí nesporeující patogenní mikroorganismy. [34] Při pasterizaci nedochází ke sterilaci. [34] Teplota a čas pasterizace se uzpůsobuje dle dané potraviny. Máme 5 základních druhů pasterizace: [33]

- Dlouhá pasterizace (LTLT)-nízká teplota po dlouho dobu-např. 30minut/63 °C.

- Šetrná pasterizace (HTST)- vyšší teplota po krátkou dobu-např. 20 vteřin/72 °C.
- Vysoká pasterizace (UHT)-výlučně u mléka a smetany pro dlouhou skladovatelnost- např. 1 vteřina/150 °C.
- Pasterizace ozářením – u zeleniny, masa a koření, kde nelze použít pasterizaci teplotou, tak se používá gama záření.
- Pasterizace horkou parou – používá se hlavně na likvidaci choroboplodných zárodků v mase.

4.2 Technologické schéma tanku



Obr. 40 - Technologické schéma tanku

V tomto technologickém tanku dochází k napouštění zadaného množství vstupní suroviny přes napouštěcí ventil a poté je obsah tanku zahřívám topným tělesem na danou teplotu po určité době. Během operace je obsah tanku také míchán míchadlem. Po ukončení operace je obsah vypouštěcím ventilem vypuštěn.

4.3 Popis vstupů/výstupů

4.3.1 Napouštěcí ventil

Jako napouštěcí ventil je použit klapkový ventil se dvěma vstupy – otevřít/zavřít. Oproti solenoidovým ventilům máme výhodu setrvání v aktuálním stavu i bez přivedeného napětí. Zpětnou vazbu potom máme od koncových spínačů ve formě logické jedničky, nebo nuly. Díky zpětné vazbě můžeme v programu ošetřit chybové stavy – například klapka se zasekne uprostřed pohybu. U tohoto ventilu máme ještě třetí zpětnou vazbu – obecnou chybu.

Otevřít ventil	doValveFill_Open	Bool	%Q260.0
Zavřít ventil	doValveFill_Close	Bool	%Q260.1

Tabulka 3 - Výstupy napouštěcího ventilu

Ventil otevřen	diValveFill_Open	Bool	%I220.0
Ventil zavřen	diValveFill_Close	Bool	%I220.1
Obecná chyba	diValveFill_Error	Bool	%I220.2

Tabulka 4 - Zpětná vazba od napouštěcího ventilu

4.3.2 Míchadlo

Míchadlo pohání třífázový elektromotor, který spínáme stykačem přes digitální výstup PLC. V případě potřeby jemnější regulace by bylo vhodné použít frekvenční měnič místo stykače, což by ale podstatě změnilo program.

Stykač motoru	doMixer_Run	Bool	%Q260.4
Zpětná vazba motoru	diMixer_Run	Bool	%I220.6

Tabulka 5 - Ovládání stykače míchadla a jeho zpětná vazba

4.3.3 Topné těleso

Jako topné těleso je v této úloze použita elektrická spirála, kterou spínáme stykačem přes digitální výstup PLC. Jednoduchou úpravou lze použít i tepelný výměník tak, že místo stykače elektrické spirály budeme spínat čerpadlo výměníku.

Stykač motoru	aoHeatPower	aoHeatPower	aoHeatPower
Zapnutí jednotky ohřevu	doHeatingEnable	doHeatingEnable	doHeatingEnable

Tabulka 6 - Výstupy ohřevu

4.3.4 Výpustný ventil

Vypouštěcí ventil je opět klapkový ventil se dvěma vstupy – otevřít/zavřít. I zde využíváme výhodu setrvání v aktuálním stavu i bez přivedeného napětí. Zpětná vazba nám zde také stačí od koncových spínačů ve formě logické jedničky, nebo nuly. Pokud bychom potřebovali kontinuální zpětnou vazbu, bylo by nutné použít servoventil, který má kontinuální zpětnou vazbu na analogový vstup PLC. Zpětnou vazbou zde také, jako i u napouštěcího ventilu, hlídáme chybové stavy.

Otevřít ventil	doValveDrain_Open	Bool	%Q260.2
Zavřít ventil	doValveDrain_Close	Bool	%Q260.3

Tabulka 7 - Výstupy vypouštěcího ventilu

Ventil otevřen	diValveDrain_Open	Bool	%I220.3
Ventil zavřen	diValveDrain_Close	Bool	%I220.4
Obecná chyba	diValveDrain_Error	Bool	%I220.5

Tabulka 8 - Zpětná vazba vypouštěcího ventilu

4.3.5 Senzory

V technologickém tanku popsaném za začátku čtvrté kapitoly se nachází dva senzory. Prvním je senzor teploty obsahu a druhým senzor výšky hladiny.

Senzor teploty	aiWaterTemp	Int	%IW242
Senzor hladiny	aiWaterLevel	Int	%IW240

Tabulka 9 - Senzory v technologickém tanku

4.4 Zadání úlohy

Naprogramujte řízení technologického tanku popsaného na začátku čtvrté kapitoly s napouštěním, vypouštěním, ohřevem a mícháním tak, aby bylo v souladu s požadavky na řízení dle standardů S88 a S95. Součástí práce bude ovládání veškerých funkcí pomocí vizualizace na panelu HMI TP-700 Comfort. V základní verzi se počítá s jedním tankem, ale ve skutečné továrně jich může být instalováno mnoho. Samotné PLC je propojeno s učitelským počítačem, na kterém běží program simulující reálný tank, pomocí S7-Communication, což znamená, že veškeré vstupy i výstupy PLC se chovají přesně tak, jako by byly fyzicky připojeny k reálnému tanku. Není proto třeba řešit nic navíc. Jednou z hlavních výhod tohoto řešení je, že můžete rovnou v průběhu programování vytvářený test oproti „reálnému“ tanku testovat. Ohřev tanku je řešený topnou spirálou, která se bude řídit PID regulátorem, který musíte správně nastavit. Pokud vyučující nestanoví jinak, tak je nutné regulátor navrhnout tak, aby držel zadanou teplotu po dobu

minimálně deseti sekund – až poté se teplota považuje za vyregulovanou. Napouštěcí a vypouštěcí hladiny, jako i požadovaná teplota a čas míchání se bude zadávat z vizualizace na HMI.

4.4.1 Možné poruchové stavy

Při provozu technologického tanku mohou nastat nenadálé situace. Je potřeba mít v programu toto ošetřeno a při poruše dostat tank do bezpečného stavu.

4.4.1.1 Napouštěcí ventil

- Ventil vyhodí chybový stav – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.
- Pokud nemáme žádný signál z hladinového čidla – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.

4.4.1.2 Vypouštěcí ventil

- Ventil vyhodí chybový stav – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.
- Pokud nemáme žádný signál z hladinového čidla – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.

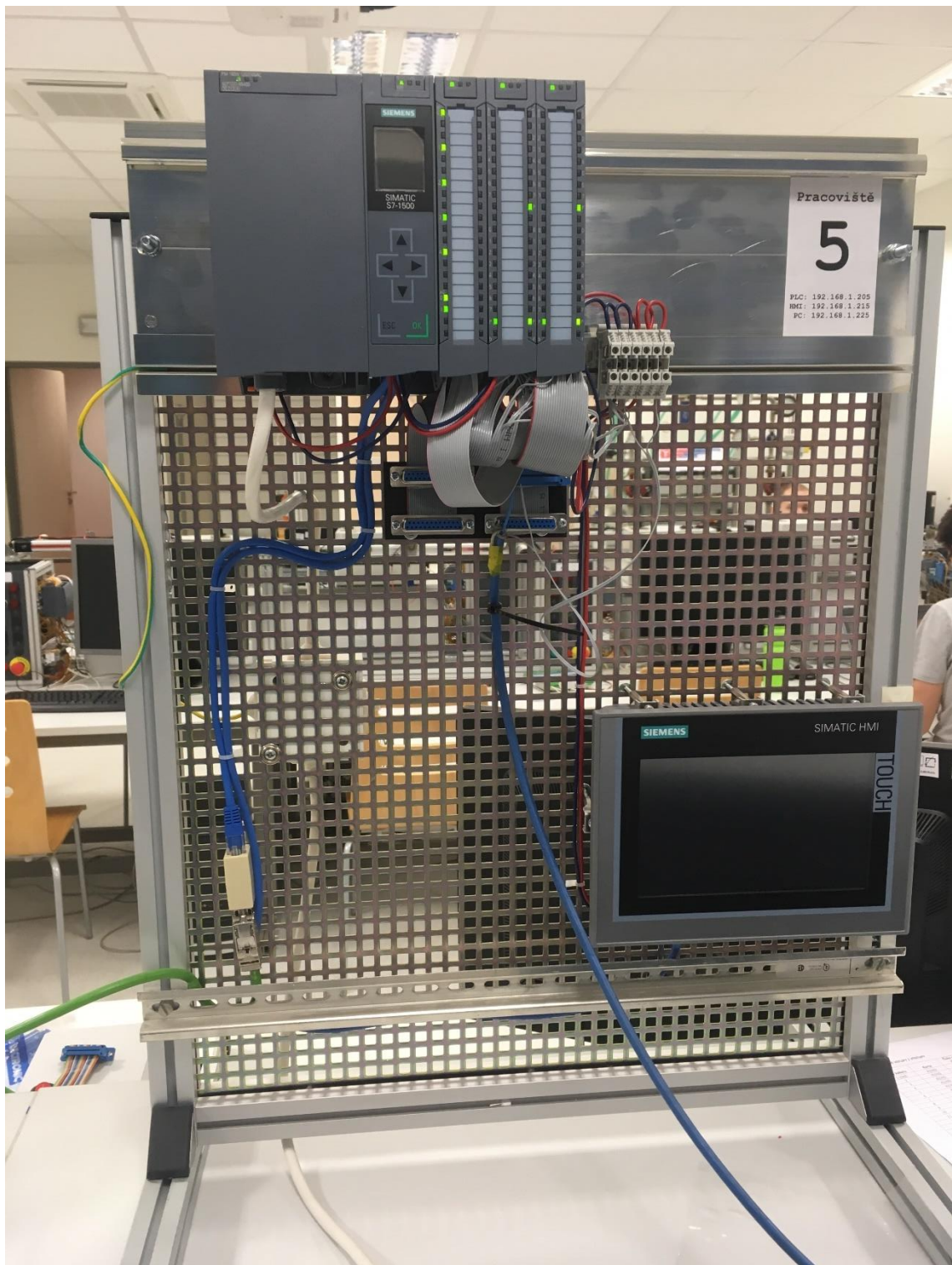
4.4.1.3 Míchadlo

- Pokud dáme povel pro stykač míchadla, ale přesto zpětná vazba od motoru nehlásí očekávaný stav – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.

4.4.1.4 Ohřev

- Pokud dáme povel pro stykač topného tělesa, ale přesto zpětná vazba z teplotního snímače po určité době toto nereflektuje – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.
- Pokud nemáme žádný signál z teplotního čidla – přerušení veškeré činnosti a přivolání obsluhy. Obsluha může po kontrole chybový stav vymazat a znovu spustit proces.

4.5 Náhled pracoviště



Obr. 41 - Náhled pracoviště

5 PLC SIMATIC S7-1500

Programovatelný automat Siemens Simatic S7-1500 byl řídicím systémem v předchozí úloze a zůstane zachován i v úloze inovované. Jedná se o výkonný, modulární automat, plně kompatibilní s TIA PORTAL 14. Důležitou vlastností oproti starším automatům je podpora jazyku GRAPH(Grafcet) a SCL ve kterém byla naprogramováno řízení jednotlivých fází předcházející úlohy. [2]

5.1 Rozšiřující karty

Programovatelný automat jako celek je tvořený hlavní procesorovou jednotkou, zdrojem a jednotlivými přídatnými kartami. Díky této modularitě si můžeme sestavit PLC podle nároků jednotlivých aplikací.

5.1.1 Připojení snímačů k PLC

Každý senzor, nebo aktuátor, který chceme používat musí být nějakým způsobem připojen k PLC. Kritériem výběru zde bývá cena, univerzálnost, jednoduchost použití, stabilita připojení a odolnost proti vnějším vlivům. Limitní senzory mají většinou přímo integrovaný převodník na digitální stavy log 1 a log 0. Oproti tomu senzory pro kontinuální měření nejčastěji připojujeme přes analogové rozhraní- buď proudové, nebo napěťové (0-10 V, 0(4) -20 mA).

5.1.1.1 0-10 V

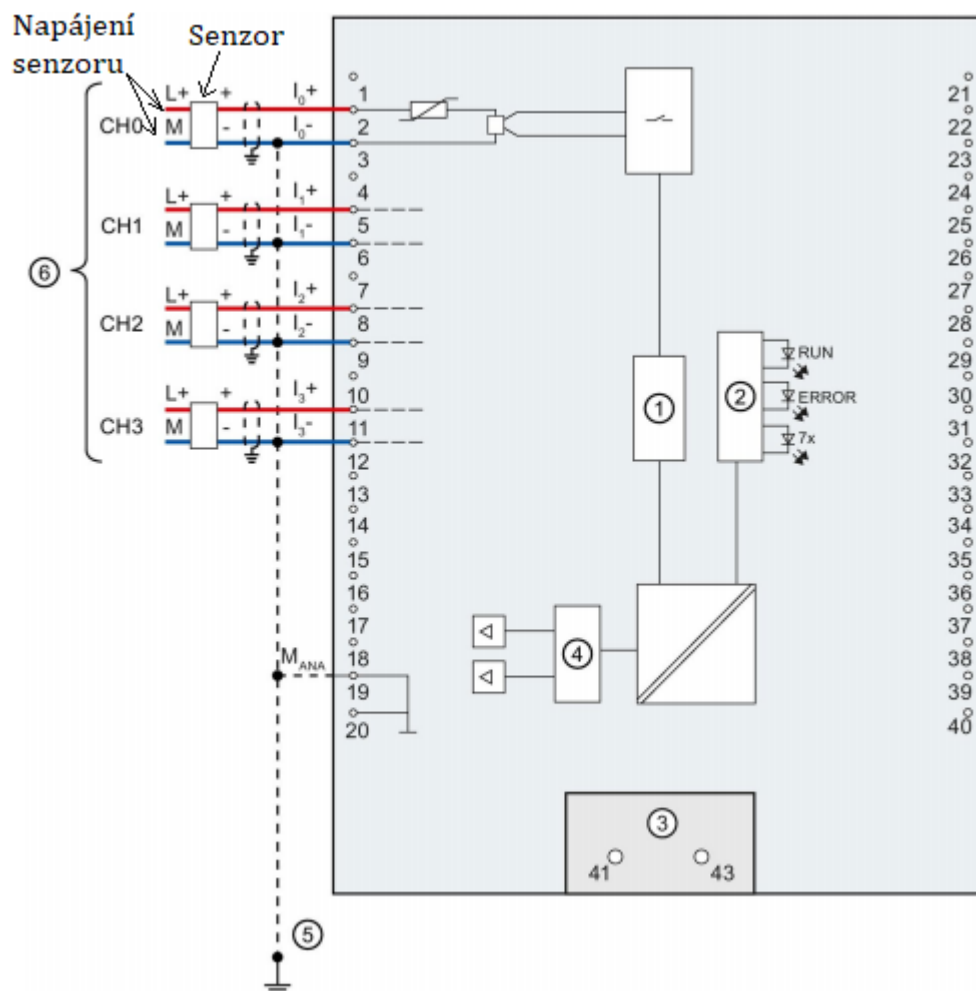
Napěťový vstup 0-10 V je nejčastěji používaný pro zpracování signálů z analogových senzorů a v podstatě každé zařízení v průmyslové praxi ho podporuje. Jelikož veškeré analogové signály jsou náchylné k rušení, zde to není výjimkou. Výkonové zařízení jako například motory, nebo relátka mohou indukovat do přívodních vodičů nežádoucí napětí a signál tak znehodnotit. Problémem je také odpor přívodního vedení v případě delší vzdálenosti.

5.1.1.2 0(4) -20 mA

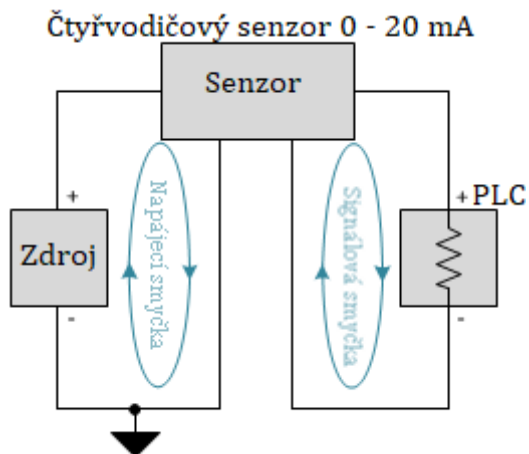
U proudové smyčky přenášíme signál pomocí velikosti protékajícího proudu, kdy u uspořádání 4-20 mA můžeme jednoduše také detekovat přerušení obvodu. Proudová smyčka je mnohem odolnější proti rušení, což je zejména u delších vedení stěžejní vlastnost. Na druhou stranu toto vše je vykoupeno o něco menší mírou kompatibility a podpory mezi výrobcí a větší cenou těchto senzorů a měřících karet.

Čtyřvodičové připojení proudové smyčky k PLC Siemens

Měřící kanál vstupní karty má čtyři piny. Při čtyřvodičovém zapojení připojujeme kladný pól signálové smyčky na druhý pin a záporný pól za třetí pin. Na piny L a M na senzoru připojujeme napájecí smyčku senzoru.



Obr. 42 - Zapojení senzoru s proudovou smyčkou do PLC Siemens [46]



Obr. 43 - Čtyřvodičové zapojení senzoru s proudovou smyčkou

5.2 PLC použité v úloze

CPU 1512C-1 PN-Jednotka má dva Profinet konektory, kapacitu 250 kB pro program a 1 MB pro data, což pro naše účely bohatě postačuje.



Obr. 44 - Možné konfigurace PLC S7-1500 [4]

5.3 Zdroj

Typ PS-60 W-24VDC – Stabilizovaný zdroj napájející soustavu samotného PLC a
přídavných karet přes zadní sběrnici.

5.4 AI 5/AQ 2

Karta analogových vstupů a výstupů. Obsahuje 5 kanálů analogových vstupů a 2
kanály analogových výstupů.

5.5 DI 16/DQ 16_1

Karta digitálních vstupů a výstupů. Obsahuje 16 kanálů digitálních vstupů a 16
kanálů digitálních výstupů.

5.6 DI 16/DQ 16_2

Druhá karta obsahuje stejně, jako předešlá karta digitálních vstupů a výstupů 16
kanálů digitálních vstupů a 16 kanálů digitálních výstupů.

6 ZÁVĚR

Hlavním cílem této bakalářské práce bylo vypracovat dokumentaci programu v PLC k předmětu MAUP podle aktuálního TIA Portálu 14 SP1. [Příloha 1] Dokumentace úlohy je připojena jako příloha. Připojen je také stručný výpis kódu pro potřeby opravování vypracovaných projektů vyučujícím. [Příloha 2] Jakýmsi bonusem pro studenty, kteří budou tuto úlohu vypracovávat je „tahák“ se základní syntaxí jazyka SCL, který bude na pracovišti jako součást úlohy. [Příloha 3] Zadání se mi podařilo i přes problémy s některými částmi programu splnit. Aktuální vývojové prostředí se velmi liší od toho, ve kterém je navržena prapůvodní úloha, kterou jsem přepracovával v rámci semestrální práce, a proto musely být některé instrukce upraveny, či nahrazeny obdobnými. Při tvorbě semestrální práce předcházející této bakalářské práci jsem si osvěžil učivo předmětu BPGA, které se mi při zpracovávání velmi hodilo. Jednotlivé bloky byly v semestrální práci psány jazyky LAD, STL a GRAFCET podle toho, který byl pro daný účel nejvhodnější.

Na rozdíl od toho je možné pro PLC v této bakalářské práci psát celý program pouze v programovacím jazyku SCL, který mi po plném osvojení vyhovuje mnohem více než předcházející jazyky, které byly v různých ohledech více limitovány. Oproti tomu jazyk SCL disponuje vším pro komfortní napsání celého programu a odpadá tímto rozhodování, který programovací jazyk pro ten konkrétní blok zvolit.

K programu v PLC byla vytvořena i vizualizace v HMI, která s PLC komunikuje přes Ethernet. HMI se skládá ze šesti různých obrazovek-výchozí obrazovky, která podává základní informace o procesu. Můžeme odtud řídit jednotlivé aktuátory jako například ventily, míchadlo a zahřívání. Všechny můžeme zapnout/vypnout a přepínat do manuálního/automatického módu. V dalších obrazovkách již ovládáme jednotlivé fáze, popřípadě parametry procesu.

V teoretické části této bakalářské práce popisují nejčastěji používané senzory a aktuátory pro technologické tanky. Jsou zde také důkladně rozebrány standardy S88 BATCH a S95 MES včetně KPI ukazatelů. Nejpoužívanější programovací jazyky jsou stručně zadokumentovány, ale největší důraz je kladen na jazyk SCL, ve kterém je celá úloha napsána. K jazyku SCL je uvedeno množství příkladů použití, ať už nejčastěji používaných konstrukcí, či komunikačních bloků, tak také stavového automatu. Je zde také napsáno samotné zadání úlohy „Tanky“ včetně ošetření možných chybových stavů

Literatura

- [1] PLC Simatic S7-1500 Manual [online] [cit. 2017-12-25]. Dostupné z: https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/EN/software_complete_en.pdf
- [2] PLC Simatic S71200, S7-1500 Programming guideline[online] [cit. 2017-12-25]. Dostupné z: http://www1.siemens.cz/ad/current/content/data_files/automatizacni_systemy/mikrosystemy/simatic_s71200/programming-guideline-for-s71200-s71500_2014-09_en.pdf
- [3] S7-GRAPH V5.3 for S7-300/400 Programming Sequential Control Systems [online] [cit. 2017-12-26]. Dostupné z: https://cache.industry.siemens.com/dl/files/630/1137630/att_28560/v1/Graph_7_e.pdf
- [4] Tisková zpráva S7-1500[online] [cit. 2017-12-26]. Dostupné z: <http://www.siemens.cz/press/nove-kompaktni-ridici-jednotky-rady-simatic-s7-1500>
- [5] *Structured Text Tutorial to Expand Your PLC Programming Skills* [online]. [cit. 2018-05-20]. Dostupné z: <http://www.plcademy.com/structured-text-tutorial/#repeat-loops>
- [6] *Structured Text Tutorial to Expand Your PLC Programming Skills* [online]. [cit. 2018-05-10]. Dostupné z: <http://www.plcademy.com/structured-text-tutorial/#for-loops>
- [7] *Structured Text Tutorial to Expand Your PLC Programming Skills* [online]. [cit. 2018-05-10]. Dostupné z: <http://www.plcademy.com/structured-text-tutorial/#while-loops>
- [8] K. HO, Anthony. *Structured Text Programming* [online]. **2012** [cit. 2018-05-10]. Dostupné z: <https://pdhonline.com/courses/e334/e334content.pdf>
- [9] *State machines* [online]. [cit. 2018-05-10]. Dostupné z: https://www.norwegiancreations.com/wp-content/uploads/2017/03/moore_mealy.png

- [10] *Structured Text (ST) Programming Guide Book* [online]. [cit. 2018-05-10].
Dostupné z: <http://dl.mitsubishielectric.com/dl/fa/document/manual/plc/sh080368e/sh080368eh.pdf>
- [11] PÁSEK, Jan a Václav KACZMARCZYK. *Automatizace procesů Laboratorní cvičení I*. BRNO: VUT FEKT [cit. 2018-05-10]
- [12] *What is a state machine?* [online]. [cit. 2018-05-10]. Dostupné z: <https://www.techopedia.com/definition/16447/state-machine>
- [13] *Termočlánky* [online]. [cit. 2018-07-13]. Dostupné z: <https://e-konstrukter.cz/prakticka-informace/termoclanky-konstrukce-vyuziti-a-ucinnost>
- [14] *Měření výšky hladiny* [online]. [cit. 2018-07-13]. Dostupné z: <http://uprt.vscht.cz/kminekm/mrt/F4/F4k44-hlad.htm>
- [15] *Srovnání teplotních čidel* [online]. [cit. 2018-07-13]. Dostupné z: https://automatizace.hw.cz//files/styles/full/public/story_automat/10535/poro_v_dotyk.jpg
- [16] *Solenoidový ventil* [online]. [cit. 2018-07-13]. Dostupné z: <https://www.indiamart.com/proddetail/uflow-solenoid-valve-14201317788.html>
- [17] *Servoventil* [online]. [cit. 2018-07-13]. Dostupné z: https://www.ohrej.se/pic_zbozi/ic82343af0553.jpg
- [18] *Stykače a jejich zapojení* [online]. [cit. 2018-07-13]. Dostupné z: https://eluc.kr-olomoucky.cz/uploads/images/11833/content_UC2_6130_schema1.jpg
- [19] *Zapojení a základní nastavení frekvenčního měniče* [online]. [cit. 2018-07-16].
Dostupné z: <https://www.mylms.cz/text-zapojeni-a-zakladni-nastaveni-frekvencniho-menice/>
- [20] *Měření teploty - kovové odporové senzory teploty* [online]. [cit. 2018-07-16].
Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/mereni-teploty-kovove-odporove-senzory-teploty.html>
- [21] *Schéma relé* [online]. [cit. 2018-07-13]. Dostupné z: https://cs.wikipedia.org/wiki/Elektromagnetick%C3%A9_rel%C3%A9#/media/File:Relay-IEC.svg

- [22] *Schéma stykače* [online]. [cit. 2018-07-13]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/thumb/e/eb/Kontakt.svg/440px-Kontakt.svg.png>
- [23] *SSR relé* [online]. [cit. 2018-07-13]. Dostupné z: https://www.hwpro.cz/oc/image/cache/catalog/produkty/HW0534_1-800x600.jpg
- [24] *Převodník 0-10V na PWM* [online]. [cit. 2018-07-13]. Dostupné z: <https://cdn.myshoptet.com/usr/www.elproz.cz/user/shop/big/1447.jpg?58ac4f1e>
- [25] *Snímače teploty* [online]. [cit. 2018-07-16]. Dostupné z: <http://skriptum.wz.cz/autom/teplota.htm>
- [26] BRZOBOHATÝ Lukáš: *Měření teploty*. Brno, 2011. 19s., CD. FSI VUT v Brně, Ústav strojírenské technologie, Vedoucí práce Ing. Kamil Podaný, Ph.D.
- [27] *Značky pro kreslení dokumentace ve vytápění* [online]. [cit. 2018-07-16]. Dostupné z: <https://vytapani.tzb-info.cz/teorie-a-schemata/351-znacky-pro-kresleni-dokumentace-ve-vytapani>
- [28] *Plášťový termočlánek* [online]. [cit. 2018-08-03]. Dostupné z: <http://www.sensit.cz/shop/plastovy-termoclanek-s-pripojenym-vedenimprumer-1-mm/termoclanek1mm/terkab.html>
- [29] *Co je MES - Výrobní informační systém* [online]. [cit. 2018-07-16]. Dostupné z: <http://www.mescentrum.cz/o-projektu/co-mes>
- [30] *Slovník pojmů* [online]. [cit. 2018-07-16]. Dostupné z: <http://www.mescentrum.cz/slovník-pojmu>
- [31] *Plovákový spínač* [online]. [cit. 2018-07-17]. Dostupné z: <https://switchstore.cz/obchod/plovakove-spinace/snimac-vysky-hladiny-plovakovy-hladinovy-spinac-220v-50w-1a-pp12h/>
- [32] *MHT-8000 Míchací a homogenizační nádrž 8000 L* [online]. [cit. 2018-07-18]. Dostupné z: <http://eshop.czechminibreweries.com/cs/product/mht-8000/>
- [33] *Pasterizace - co o ní možná nevíte* [online]. [cit. 2018-07-18]. Dostupné z: <https://www.frigomat.cz/pasterizace-ni-mozna-nevite/>
- [34] *Pasterace* [online]. [cit. 2018-07-18]. Dostupné z: <https://cs.wikipedia.org/wiki/Pasterace>

- [35] *Nerezové deskové výměníky* [online]. [cit. 2018-07-18]. Dostupné z: <https://www.regulus.cz/cz/nerezove-deskove-vymeniky>
- [36] VOBEJDA, L. *Dávkové řízení modelu destilační kolony*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 71s. Vedoucí diplomové práce byl Ing. Radek Štohl, Ph.D
- [37] *Fractional distillation of petroleum* [online]. [cit. 2018-07-18]. Dostupné z: <https://www.slideshare.net/ajmalarbab/fractional-distillation-of-petroleum>
- [38] Podívejte se, jak se na Slovensku vyrábějí vozy Kia. *Www.idnes.cz* [online]. [cit. 2018-07-18]. Dostupné z: https://auto.idnes.cz/podivejte-se-jak-se-na-slovensku-vyrabeji-vozy-kia-fs9-/automoto.aspx?c=A080916_151719_automoto_fdv
- [39] *Pacovské strojířny* [online]. [cit. 2018-07-16]. Dostupné z: <http://www.minipivovary-pacovske.cz/cz/homepage/default/5/projekty>
- [40] *Co lze nalézt v normě ANSI/ISA-88.01* [online]. [cit. 2018-07-23]. Dostupné z: http://automa.cz/cz/casopis-clanky/co-lze-nalezat-v-norme-ansi/isa-88-01-2001_10_33691_1736/
- [41] *Hugo Valves* [online]. [cit. 2018-07-24]. Dostupné z: <http://cz.hugovalves.com/butterfly-valve/>
- [42] HERZÁN, Jan Bc. *Systém na sledování výroby*. Praha, 2005. Diplomová práce. ČVUT. Vedoucí práce Prof. Dr. Ing. Zdenek Hanzálek.
- [43] *OEE a odvozené ukazatele* [online]. [cit. 2018-07-16]. Dostupné z: <http://www.mescentrum.cz/clanky/mes-mom/133-oeo>
- [44] *Understanding TEEP* [online]. [cit. 2018-07-16]. Dostupné z: http://www.sylution.com/knowledge_base/TEEP_Definition.htm
- [45] *Texas Instruments: 0-20 mA current loop* [online]. [cit. 2018-07-16]. Dostupné z: http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/446068?0-20mA-and-4-20mA-current-loop#
- [46] *Siemens S7-1500 datasheet* [online]. [cit. 2018-07-16]. Dostupné z: https://cache.industry.siemens.com/dl/files/676/109478676/att_895885/v1/s71500_cpu1512c_1_pn_manual_en-US_en-US.pdf
- [47] Rekonstrukce chlazení: Technologické schéma. Dostupné také z: Interní neveřejný dokument

Seznam příloh

Příloha 1. ŠKAPA, A. *Inovace laboratorní úlohy - "Tanky"-nová dokumentace*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. Vedoucí bakalářské práce Ing. Václav Kaczmarczyk, Ph.D..

Příloha 2. ŠKAPA, A. *Inovace laboratorní úlohy "Tanky"-výpis kódů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. Vedoucí bakalářské práce Ing. Václav Kaczmarczyk, Ph.D..

Příloha 3. ŠKAPA, A. *Inovace laboratorní úlohy "Tanky"-základní syntaxe SCL*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. Vedoucí bakalářské práce Ing. Václav Kaczmarczyk, Ph.D..