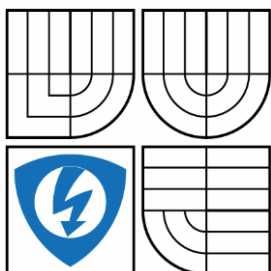


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PLC ŘÍZENÍ VÝROBY VE VÍCE NÁDRŽÍCH NA BÁZI STANDARDU S88

PLC PROCESS CONTROL OF MORE TANKS BASED ON STANDARD S88

BAKALÁŘSKÁ PRÁCE

BACHELOR THESIS

AUTOR PRÁCE

AUTHOR

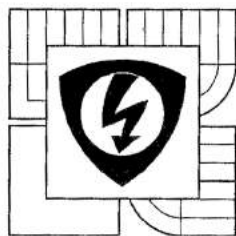
MATÚŠ DIZORZI

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAN PÁSEK, CSc.

BRNO 2016



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Matúš Dizorzi

Ročník: 3

ID: 164256

Akademický rok: 2015/16

NÁZEV TÉMATU:

PLC řízení výroby ve více nádržích na bázi standardu S88

POKYNY PRO VYPRACOVÁNÍ:

1. Vypracujte teoretický rozbor standardu ISA S88.
2. Vypracujte popis úlohy řízení procesu plnění, míchání a vypouštění tanků.
3. Navrhněte koncepci řešení zadané úlohy pomocí PLC modulárním způsobem na bázi S88.
4. Realizujte vlastní řešení programů modulů řídicího systému pro PLC.
5. Realizujte vlastní řešení pro vizualizaci a ovládání ve WinCC.
6. Ověřte funkčnost řízení procesu simulovanou demoverzí.

DOPORUČENÁ LITERATURA:

Termín zadání: 8. 2. 2016

Termín odevzdání: 23.5.2016

Vedoucí práce: Ing. Jan Pásek, CSc.

Konzultanti bakalářské práce:

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

Hlavnou úlohou tejto práce je vytvorenie riešenia úlohy PLC riadenia výroby vo viacerých nádrži na bázy štandardu ISA S88. Táto práca bude ďalej využitá ako výukový materiál v laboratórnej úlohe počas magisterského štúdia. Prvá časť práce sa zaoberá rozborom štandardu ISA S88. Druhá časť práce je venovaná rozboru vytvoreného programu od jeho najmenších častí až po kompletnú funkčnú nádrž. Záverom práce je overenie funkčnosti riadenia procesu simulovanou demo verziou – vizualizáciou vo WinCC.

Kľúčové slová

Štandard ISA S88, dávkové riadenie, dávkové procesy, PLC program, riadenie nádrží, riadiaci systém Simatic S7-300, systém BATCH.

Abstract

Main task of this work is to create a solution to PLC process control of more tanks based on ISA S88 standard. This work will be used as a study material to laboratory exercises during master's study. First part of this work deals with the ISA S88 standard. The second part deals with analysis of created program from its most basic parts to complete functional tank. Conclusion of this work is to confirm functionality of this program by simulated demo version – visualisation in WinCC.

Keywords

ISA S88 Standard, batch control, batch processes, PLC program, control of tanks, Simatic S7-300 control system, BATCH system

Bibliografická citace:

DIZORZI, M. PLC řízení výroby ve více nádržích na bázi standardu S88. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 47 s. Vedoucí bakalářské práce Ing. Jan Pásek, CSc.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma PLC řízení výroby ve více nádržích na bázi standardu S88 jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této diplomové bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **23. května 2016**

.....
podpis autora

Poděkování

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

V Brně dne: **23. května 2016**

.....
podpis autora

Obsah

1	Úvod	10
2	Štandard ISA S88	11
2.1	Proces	11
2.2	Základné modely štandardu S88	12
2.2.1	Procesný model	12
2.2.2	Fyzický model	13
2.3	Štruktúra dávkového riadenia	14
2.4	Receptúry	17
2.5	Režimy	18
2.6	Stavy.....	20
3	Proces plnenia, miešania a vypúšťania nádrže	21
3.1	Rozbor úlohy riadenia podľa štandardu ISA S88.....	22
3.2	Požiadavky zadania	23
3.2.1	Funkčný popis	23
3.2.2	Vizualizácia.....	24
3.2.3	Režimy a módy ovládania	24
4	Štruktúra programu.....	25
4.1	Tvorba ventilu	25
4.2	Tvorba motoru.....	27
4.3	Monitorovanie hladiny v nádrži	30
4.4	Vytvorenie fáz procesu	30
4.5	Strom volaní	36
4.6	Štruktúra kompletnej nádrže	37
4.7	Prevádzka a ovládanie procesu	39
4.8	Užívateľské rozhranie	41
5	Záver.....	44

Zoznam použitých obrázkov

Obr. 2.1: a) Procesný model dávkového riadenia, b) Fyzický model dávkového riadenia	14
Obr. 2.2: a) Model procedurálneho riadenia, b) Vzťah medzi fyzickým model, procesným modelom a modelom procedurálneho riadenia	16
Obr. 2.3: Rozdelenie receptúr	18
Obr. 3.1: Ilustračné schéma dvoch nádrží, zdroj [2]	21
Obr. 3.2: Programovanie typu BATCH	23
Obr. 4.1: Vývojový diagram funkcie ventilu	26
Obr. 4.2: Vstupno-výstupný parameter DATA funkcie FC_VENTIL.....	27
Obr. 4.3: Vstupno-výstupný parameter DATA funkcie FC_MOTOR.....	28
Obr. 4.4: Vývojový diagram funkcie motoru.....	29
Obr. 4.5: Parametre funkcie napúšťacej fáze	31
Obr. 4.6: Vývojový diagram napúšťacej fáze	32
Obr. 4.7: Ukážka kódu simulovania napúšťania nádrže	32
Obr. 4.8: Parametre funkcie miešacej fáze.....	33
Obr. 4.9: vývojový diagram miešacej fáze.....	34
Obr. 4.10: Parametre vypúšťacej funkcie.....	35
Obr. 4.11: Vývojový diagram vypúšťacej fáze	36
Obr. 4.12: Strom volaní funkcií	37
Obr. 4.13: Štruktúra hotovej nádrže	38
Obr. 4.14: Diagram prechodov.....	40
Obr. 4.15: Ovládací panel automatického režimu.....	41
Obr. 4.16: Ovládanie nádrže v manuálnom režime, indikátory a parametre nádrže	43
Obr. 4.17: Kompletne užívateľské rozhranie, ukážka bežiaceho programu v automatickom režime.....	43

Zoznam tabuliek

Tab. č. 2.1: Príklad implementácie režimov	19
Tab. č. 4.1: Užívateľsky definovaný typ DT_VENTIL	25
Tab. č. 4.2: Užívateľsky definovaný typ DT_MOTOR	27
Tab. č. 4.3: Užívateľsky definovaný typ DT_HLADINA.....	30
Tab. č. 4.4: Užívateľsky definovaný typ DT_FAZA	30
Tab. č. 4.5: Užívateľsky definovaný typ DT_NADRZ	37
Tab. č. 4.6: Dátový blok DB_NADRZ	38
Tab. č. 4.7: Užívateľsky definovaný typ DT_PREVADZKA.....	39
Tab. č. 4.8: Matica prechodov stavov	40

1 ÚVOD

Proces napúšťania, miešania a vypúšťania jednej či viacerých nádrží má široké uplatnenie. Využíva sa v podnikoch zaoberajúcimi sa skladovaním rôznych druhov materiálu a následne jeho spracovaním.

U procesov spracovania produktov v nádržiach sa zvyčajne jedná o dávkovú výrobu. Práve pre tento typ výroby bol vyvinutý štandard ISA S88 zaoberajúci sa jej riadením. Jedná sa o logické vyústenie procesného riadenia založeného na programovateľných automatoch PLC a pre to bol štandard ISA S88 zaradený do výuky na našej fakulte na ústave UAMT v magisterskom štúdiu. Konečnú aplikáciu výukového projektu dodala spoločnosť COMPAS s.r.o., avšak časť ukážkového riešenia bola vypísaná ako bakalárska práca, ktorej som sa stal riešiteľom, a ktorú predkladám v nasledujúcich kapitolách.

Táto úloha bude slúžiť ako laboratórne cvičenie v druhom ročníku magisterského štúdia a to v predmete MAUP – „Automatizácia procesov“. Technologicky sa jedná o napúšťanie, miešanie a vypúšťanie dvoch a viacerých nádrží. Predpokladá sa pri tom možnosť voľby priority a funkčných nadväzností jednotlivých nádrží.

Konečné riešenie bude realizované v PLC SIMATIC S7 s vizualizáciou HMI WinCC flexible. Program bude rešpektovať hierarchickú a procedurálnu štruktúru štandardu ISA S88.

Vlastná bakalárska práca je tvorená z dvoch častí, pričom v prvej časti sa zaoberá základnými modelmi a terminológiou štandardu ISA S88, podľa ktorého je tvorená druhá časť práce. V druhej časti sa nachádza rozbor kompletného programu a jeho popis, od jeho riadiacich modulov cez fáze až po kompletnú štruktúru vytvorenej nádrže. Hotový program je priložený ako príloha na CD.

2 ŠTANDARD ISA S88

Štandard ISA S88 zabezpečuje štandardnú terminológiu a modely pre „Batch Control“, teda dávkové riadenie. Modely a terminológia definovaná v tomto štandarde sa zaoberá zlepšením riadenia dávkových procesoch vo výrobných závodoch, znázorňuje rozumné použitia na návrh a riadenie týchto procesov a dá sa použiť aj mimo oboru automatizácie.

Štandard S88 sa skladá z konceptov a modelov, ktoré dokážu zlepšiť komunikáciu medzi účastníkmi tohto procesu a to tak, že znižujú časovú dobu dosiahnutia plnej produkcie nových produktov, dodávateľom umožňujú zaobstarat' potrebné prostriedky na implementáciu dávkového riadenia, umožňujú používateľom lepšie identifikovať ich potreby či znížiť náklady na automatizáciu dávkových procesov.

Cieľom tohto štandardu nie je tvrdiť, že existuje len jeden spôsob implementácie dávkového riadenia alebo nútiť používateľom zmeniť ich spôsob riešenia dávkových procesov, či zakazovať rozvoj v oblasti dávkového riadenia. Hlavnými cieľmi štandardu S88 je modularita, transparentnosť a zrozumiteľnosť a opakované použitie.

Čo sa týka modularity, tak rozdelením technologického procesu na skupinu menších a jednoduchších operácií sa dosiahne jednoduchšieho popisu a pochopenia procesu. Taktiež umožňuje stavebnicové zostavenie programu z programových modulov.

Štandardná terminológia S88 je základom pre dobrú komunikáciu členov riešiteľského tímu. Transparentnosť je daná modulárnou štruktúrou a grafickou formou programu, pričom uľahčuje komunikáciu medzi užívateľmi, dodávateľmi a subdodávateľmi.

Opakované použitie vyplýva z modularity. Prvky riadenia a prvky technológie sú štruktúrované hierarchicky do nezávislých programových modulov. Teda keď ja takýto modul vyvinutý, môže byť znovu použitý pre každý podobný technologický prvok. Opakovateľnosť je zásadné hľadisko pre zníženie nákladov na riadenie procesov.

2.1 Proces

Proces je sekvencia fyzických, chemických alebo biologických činností na premenu, premiestňovanie či uskladňovanie materiálu alebo energie. V priemyselnej výrobe sa procesy môžu definovať ako spojité, diskrétne alebo dávkové. Klasifikácia procesu závisí na tom, či sa výstup z procesu javí ako stály tok (spojitý proces), v konečných množstvách alebo častiach (diskrétny proces) alebo po konečných množstvách materiálu, teda dávkach.

V spojitom procese je materiál dodávaný v stálom (spojitom) toku zariadeniu, ktoré tento materiál spracováva. Po dosiahnutí ustálenej prevádzky nezávisí povaha procesu na dĺžke doby prevádzky. Začiatky, prechody ani zastavenia procesu neprispievajú k dosiahnutiu požadovaného spracovávania spojitého procesu.

V diskretnom procese je produkt pridelený do produkčných skupín, ktoré sú založené na základe výrobného materiálu, požiadavkách a histórie produkcie. V diskretnom procese sa pohybuje určité, predom dané, množstvo produktu ako jednotka medzi pracovnými stanicami, pričom každá jednotka si uchováva svoju jedinečnú identitu.

Dávkový proces vedie k produkcii konečného množstva materiálu, teda dávok, a to podrobením týchto dávok vstupného materiálu predom definovanému poradiu procesných činností použitím jedného alebo viacerých zariadení. Výsledný produkt sa nazýva dávka. Dávkové procesy sú nespojité procesy, to znamená, že nie sú ani diskretné, ani spojité, avšak majú vlastnosti oboch.

2.2 Základné modely štandardu S88

2.2.1 Procesný model

Štandard S88 obsahuje dva základné modely a to model procesný a model fyzický. Procesný model definuje proces z hľadiska procesných činností, pričom sa neviaže na žiadne zariadenie. Skladá sa zo štyroch hierarchicky usporiadaných úrovní a to proces, procesný úsek, procesná operácia a procesný krok.

Proces sa skladá z jedného alebo viacerých procesných úsekov, ktoré sú organizované ako usporiadaná skupina. Táto skupina môže byť usporiadaná sériovo, paralelne alebo kombináciou oboch. Proces sa zvyčajne skladá z niekoľkých procesných úsekov pracujúcich nezávisle na sebe. Výsledkom procesného úseku zvyčajne býva sekvencia chemických alebo fyzických zmien spracovávaného materiálu. Typickým procesným úsekom môže byť napríklad spracovávanie kvasníc, fermentácia či filtrovanie v pivovare.

Každý procesný úsek sa skladá z usporiadanej skupiny procesných operácií. Procesná operácia reprezentuje hlavné procesné činnosti. Výsledkom procesnej operácie zvyčajne býva chemická či fyzická zmena spracovávaného materiálu. príkladom procesnej operácie môže byť napríklad fermentácia piva – dávkovanie mladiny, vlastná fermentácia, presun piva do ďalších nádrží.

Každá procesná operácia môže byť ďalej rozdelená do usporiadanej skupiny procesných krokov. Tieto kroky uskutočňujú procesy požadované od procesných operácií, opisujú vedľajšie procesné činnosti, ktoré skombinovaním vytvárajú procesnú operáciu. Procesným krokom sú tie najjednoduchšie činnosti procesného modelu, ako je napríklad dávkovanie surovín.

2.2.2 Fyzický model

Fyzický model je hierarchická štruktúra fyzického vybavenie výrobného celku, zložená z podnikov, závodov, oblastí, procesných buniek, jednotiek, modulov zariadení a riadiacich modulov. Najnižšie časti fyzického modelu sú kombinované, pričom tvoria vyššie časti v hierarchickej štruktúre. Najvyššou časťou fyzického modelu je podnik, za ním nasleduje závod a oblasť.

Podnik je zložený z jedného alebo viacerých závodov. Môže obsahovať závody, oblasti, procesné bunky, jednotky, moduly zariadenia aj riadiace moduly. Podnik je zodpovedný za určenie materiálu, ktorý sa bude vyrábať, v ktorých závodoch sa tento materiál bude vyrábať, a celkovo ako sa bude vyrábať.

Závod je fyzické, geografické alebo logické zoskupenie určené podnikom a môže obsahovať všetky nižšie úrovne fyzického modelu. Podobne je oblasť fyzické, geografické alebo logické zoskupenie určené závozom, ktoré môže obsahovať zvyšné nižšie úrovne.

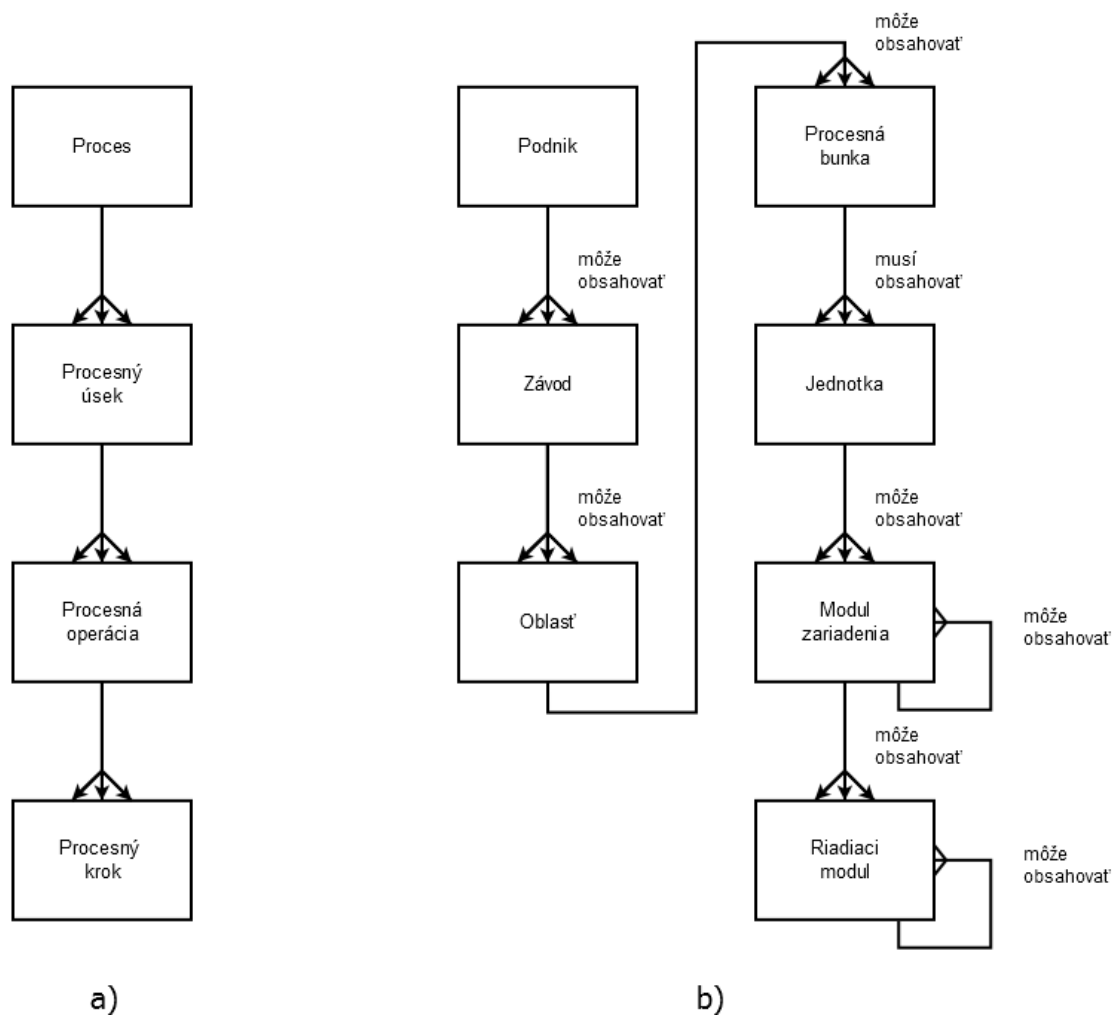
Procesná bunka obsahuje všetky jednotky, moduly zariadení a riadiace moduly požadované k výrobe jednej či viacerých dávok. Procesná bunka je klasifikovaný ako jedno-produktová alebo viac-produktová závisiac od počtu produktov plánovaných na produkciu v tejto bunke. Jedno-produktová procesná bunka produkuje rovnaký produkt v každej dávke, pričom procedúry a parametre sa môžu líšiť, napríklad kvôli rozdielnemu zariadeniu či inému výrobnému materiálu. Viac-produktová procesná bunka produkuje rozdielne produkty využitím rôznych metód produkcie či riadenia. Ďalej je možné procesnú bunku deliť podľa jej štruktúry na jedno-cestnú, viac-cestnú alebo sieťovú.

Jedno-cestná procesná bunka je skupina jednotiek, cez ktoré dávka prechádza sekvenčne. Viac-cestná procesná bunka pozostáva z niekoľkých jedno-cestných, ktoré sú usporiadané paralelne, pričom medzi nimi nedochádza k presunu materiálu. Sieťová procesná bunka obsahuje cesty, ktoré sú buď pevné, alebo premenné. Bunka s pevnými cestami používa rovnaké jednotky v rovnakom poradí, zatiaľ čo v bunke s premennými cestami sa poradie môže určiť na začiatku dávky alebo počas produkcie dávky. Bunka môže mať cesty aj úplne flexibilné, kedy dávka môže začať v akejkoľvek jednotke a viesť niekoľko ciest cez procesnú bunku.

Jednotka je tvorená z modulov zariadení a riadiacich modulov. V jednotke môže byť vykonávaná jedna alebo viacero hlavných procesných činností. jednotka kombinuje všetky potrebné fyzické a riadiace zariadenia požadované k vykonávaniu týchto činností. Zvyčajne sa sústreďuje na hlavnú časť procesného zariadenia, ako je napríklad nádrž na miešanie. Jednotky pracujú nezávisle od seba.

Modul zariadenia kombinuje potrebné fyzické procesy a riadiace zariadenia požadované k vykonaniu procesných činností, pričom sa sústreďuje na časť procesného zariadenia, ako je napríklad filter.

Riadiaci modul je tvorený zo skupiny senzorov, akčných členov, ostatných riadiacich modulov a priradených procesných zariadení. Riadiaci modul môže byť taktiež tvorený z iných riadiacich modulov.



Obr. 2.1: a) Procesný model dávkového riadenia, b) Fyzický model dávkového riadenia

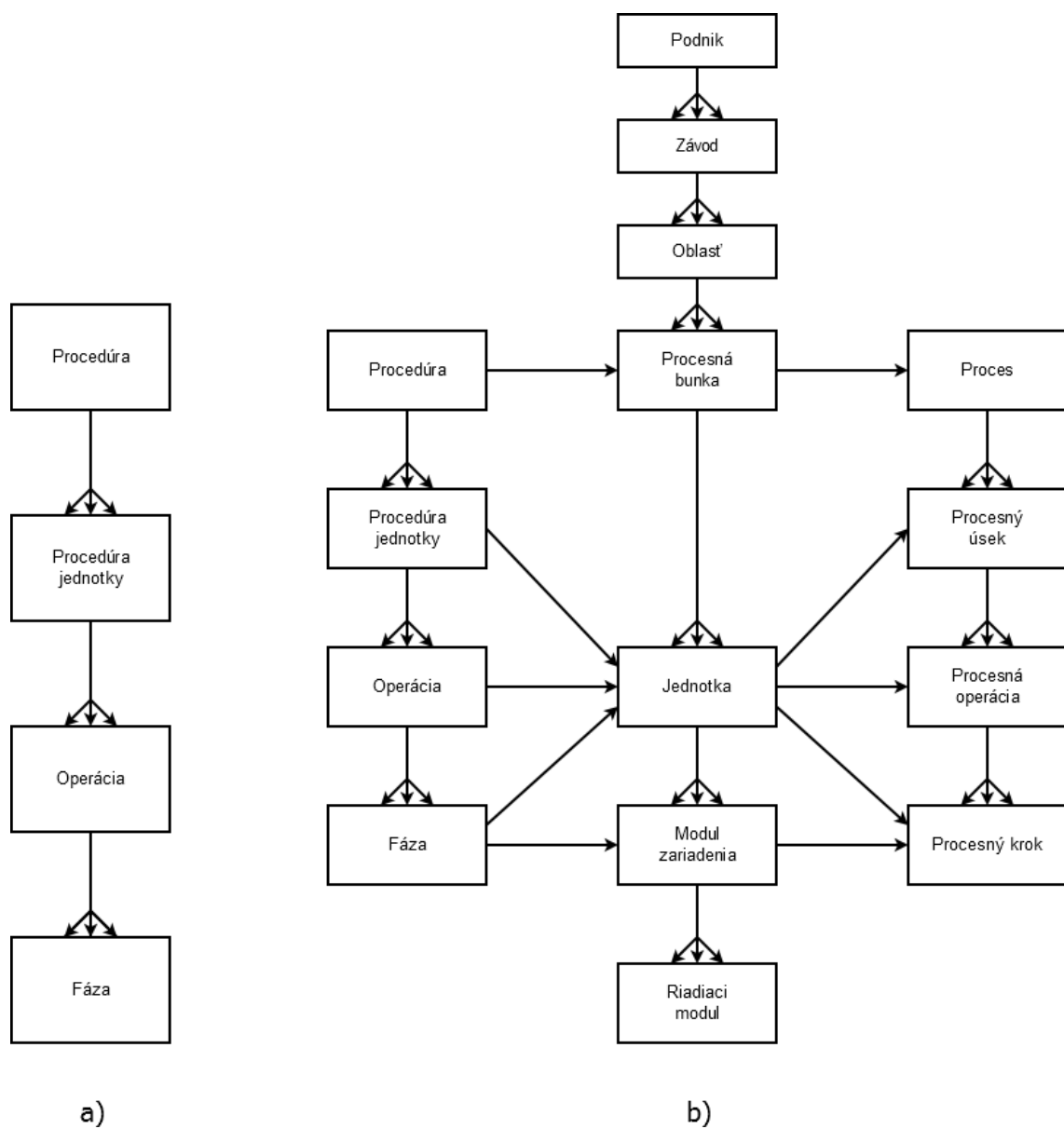
2.3 Štruktúra dávkového riadenia

Dávkové riadenie možno rozdeliť na tri druhy riadenia a to základné, procedurálne a koordinačné riadenie. Základné riadenie zahŕňa riadenie určené k zavedeniu a udržiavaniu špecifického stavu zariadenia a procesu. Spadá sem regulačné riadenie, blokovanie, monitorovanie či spracovanie výnimiek. Základné riadenie reaguje na udalosti v procese, ktoré by mohli ovplyvniť výstupy zariadení. Môže byť aktivované, deaktivované alebo modifikované operátorskými príkazmi alebo procedurálnym či koordinovaným riadením.

Procedurálne riadenie usmerňuje činnosti zamerané na zariadenie do usporiadanej sekvencie tak, aby sa vykonala činnosť zameraná na proces. Je charakteristické v dávkových procesoch. Jedná sa o riadenie umožňujúce zariadeniu vykonávať práve tento dávkový proces a pozostáva z procedurálnych elementov skombinovaných

v hierarchickej štruktúre. Táto štruktúra sa skladá z procedúr, procedúr jednotiek, operácií a fáz. Procedúra je najvyšší level tejto štruktúry a definuje stratégiu na vykonanie hlavných procesných činností, ako je napríklad výroba dávky. Procedúra jednotky je zložený z usporiadaného súboru operácií, ktoré vykonávajú súvislú produkčnú sekvenciu v rámci jednotky. Predpokladá sa len jedna aktívna operácia v tom istom čase, ktorá sa vykonáva až do konca v jednej jednotke. Operácia definuje hlavné procesné sekvencie v materiáli spracovávaného z jedného stavu do druhého, zvyčajne zahŕňajúc fyzickú či chemickú zmenu. Fáza je najmenší element procedurálneho riadenia, ktorý môže vykonávať procesne orientovanú úlohu. Fáza môže byť rozdelená do menších častí, ako sú kroky a prechody a môže vykonávať jeden alebo viac príkazov, napríklad nastavenie, odstránenie či zmeny alarmov, čítanie premenných, zmenu konštanty a iné. Cieľom fázy je spôsobiť a definovať procesne orientovanú činnosť, pričom logika a súbor krokov tvoriace fázu sú špecifické od zariadenia.

Posledným riadením je riadenie koordinačné. Toto riadenie usmerňuje, inicializuje a modifikuje uskutočnenie procedurálneho riadenia, pričom využíva triedy (entity) zariadenia. Je rozličné v čase, tak isto ako aj procedurálne riadenie, avšak nie je štruktúrované na špecifickú procesne orientovanú činnosť. Príkladom koordinovaného riadenia je alokácia dávok do zariadení, výber procedurálnych elementov na uskutočnenie.



Obr. 2.2: a) Model procedurálneho riadenia, b) Vzťah medzi fyzickým model, procesným modelom a modelom procedurálneho riadenia

2.4 Receptúry

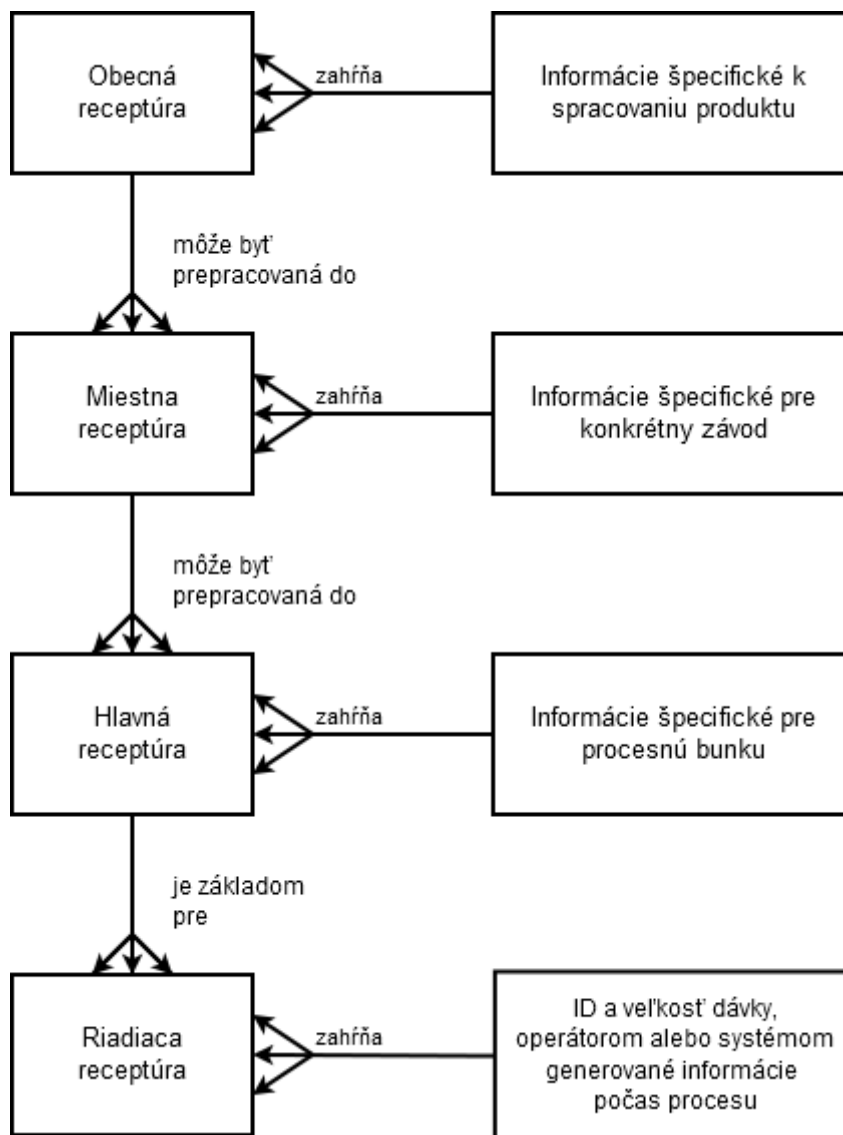
V podniku sa zvyčajne nachádzajú štyri druhy receptúr. Receptúra je subjekt obsahujúci minimálne množstvo informácií, ktoré jednoznačne definuje výrobné požiadavky pre špecifický produkt. V podstate sa jedná o opis produktu a spôsobu jeho výroby. Rozdelenie receptúr je zobrazené na *obr. 2.3*.

Obecná receptúra slúži ako základ pre nižšie-úrovňové receptúry. Je vytvorená bez špecifickej znalosti zariadenia procesnej bunky, ktorá bude k výrobe produktu použitá. Identifikuje výrobné materiály, ich množstvá a požadované spracovanie. Je prenositeľná od miesta výroby do akéhokoľvek závodu.

Miestna receptúra je špecifická pre konkrétny závod. Zvyčajne je odvodená od obcej receptúry tak, aby splnila podmienky konkrétneho závodu. Zaisťuje úroveň podrobnosti potrebnej k dlhodobej produkcii produktu. Je prenositeľná, avšak je zameraná na používanie v určitom závode a teda je prenositeľná len v tomto závode.

Hlavná receptúra je zameraná na procesné bunky alebo podmnožinu zariadení procesných buniek. Musí byť dostatočne prispôbená na vlastnosti zariadenia tak, aby sa zaistilo správne spracovanie dávky. Môže obsahovať informácie špecifické k produktu vyžadované na detailne plánovanie, ako sú napríklad vstupné informácie procesu alebo požiadavky zariadení. Je prenositeľná medzi rozličnými procesnými bunkami.

Riadiaca receptúra začína ako kópia špecifickej verzie hlavnej receptúry a po čase je modifikovaná podľa potreby tak, aby bola špecifická pre jednu dávku. Obsahuje procesné informácie špecifické k produktu na výrobu konkrétnej dávky alebo produktu. Nie je prenositeľná.



Obr. 2.3: Rozdelenie receptúr

2.5 Režimy

Entity zariadenia a procedurálne prvky môžu mať režimy. Režim môže byť založený na procedurálnych prvkoch využívajúcich základné riadiace funkcie v závislosti na hlavnej riadiacej charakteristike triedy. Riadiace moduly môžu obsahovať základné riadiace funkcie a majú manuálny a automatický režim, jednotka spracováajúca procedurálne riadenie môže mať aj režim semi-automatický.

Režim vyhodnocuje ako triedy zariadenia a procedurálny prvky reagujú na príkazy a ako fungujú. V prípade procedurálnych prvkov režim vyhodnocuje spôsob akým sa procedura bude vyvíjať a čo tento vývoj môže ovplyvniť. V prípade riadiaceho

modulu, ako je napríklad automatický ventil so základnými riadiacimi funkciami, režim vyhodnocuje mechanizmus použitý k poháňaniu ventilu a čo alebo kto môže manipulovať so zmenou jeho stavu.

Pre procedurálne prvky režim vyhodnocuje spôsob ako sa chovajú prechody. V automatickom režime prechody nasledujú bez prerušenia vtedy, keď nastane podmienka prechodu. V semi-automatickom režime požaduje procedúra manuálne potvrdenie po splnení podmienky prechodu. Preskočenie či znova vykonanie jedného alebo viacerých prechodov bez zmeny ich poradia je dovolené. V manuálnom režime sú procedurálne prvky a ich poradie špecifikované manuálne.

Procedurálne prvky môžu zmeniť režimy. Táto zmena nastane ak sú splnené požiadavky podmienenej logiky pre zmenu a to buď vnútornou logikou, alebo vonkajším príkazom, napríklad generovaným iným procedurálnym prvkom alebo operátorom. Režim sa zmení len vtedy ak sú podmienky na jeho zmenu splnené. Zmena režimu v jednom type procedurálneho prvku môže zapríčiniť korešpondujúce zmeny v ostatných typoch, napríklad nastavenie jednotkovej procedúry do automatického režimu z režimu manuálneho zapríčiní nastavenie všetkých nižšie-úrovňových procedurálnych prvkov tiež do režimu automatického.

Tab. č. 2.1: Príklad implementácie režimov

Režim	Popis	Príkazy
Automatický	Prechody v procedúre sú vykonávané bez prerušenia keď sú splnené príslušné podmienky	Operátor môže pozastaviť postup, ale nemôže vynútiť prechod
Semi-automatický	Prechody v procedúre sú vykonávané pomocou manuálnych príkazov keď sú splnené príslušné podmienky	Operátor môže pozastaviť postup alebo presmerovať vykonávanie do príslušného miesta. Prechody nemôže vynútiť.
Manuálny	Procedurálne prvky v procedúre sú vykonávané v poradí špecifikovaným operátorom	Operátor môže pozastaviť postup alebo vynútiť prechod

2.6 Stavy

Triedy zariadení a procedurálne prvky môžu mať stavy. Stav úplne špecifikuje aktuálnu podmienku triedy zariadenia a procedurálneho prvku. Množstvo možných stavov a príkazov a ich názvov sa môže líšiť.

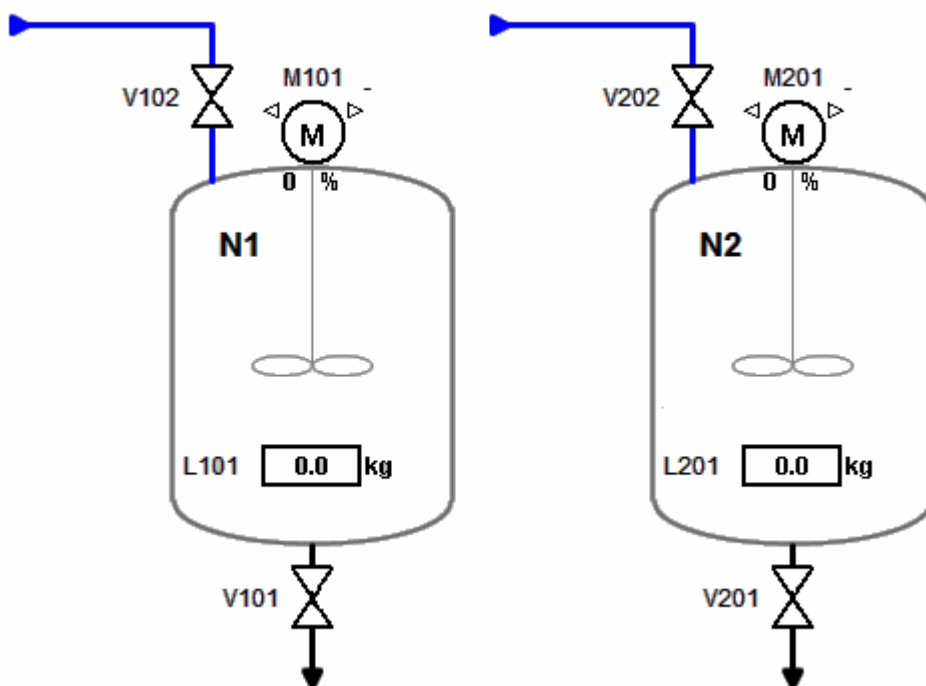
Procedurálne prvky môžu zmeniť stav. Táto zmena nastane v prípade splnenia požiadavky podmienenej logiky a to buď vnútornou logikou, alebo vonkajším príkazom generovaným iným procedurálnym prvkom alebo operátorom.

Zmena stavu v jednom type procedurálneho prvku môže zapríčiniť korešpondujúce zmeny v ostatných typoch. Napríklad nastavenie jednotkovej procedúry do určitého stavu môže zapríčiniť to, že všetky procedurálne prvky v tejto jednotke sa tiež nastaví do toho istého stavu.

3 PROCES PLNENIA, MIEŠANIA A VYPÚŠŤANIA NÁDRŽE

Program, ktorý v tejto práci riešim, sa zaoberá napúšťaním, miešaním a vypúšťaním jednej alebo viacerých nádrží. Ilustračné schéma je na *obr. 3.1*. Proces v nádržiach pozostáva z troch fáz a to napúšťacej fáze, miešacej fáze a vypúšťacej fáze. Pre každú nádrž prebieha proces zvlášť a nezávisle od ostatných nádrží. Napúšťacia fáza inicializuje proces a teda nádrž sa začína plniť. Nasleduje miešacia fáza a proces je ukončený fázou vypúšťacou. Po jeho skončení je možné proces opakovať.

Užívateľ je schopný program prevádzkovať v režime manuálnom a režime automatickom. V manuálnom režime má užívateľ úplnú kontrolu nad všetkými procesmi v nádržiach, je teda schopný ovládať akýkoľvek riadiaci modul hociktorej nádrže. V automatickom režime užívateľ stráca schopnosť riadiť riadiace moduly, avšak môže ovplyvňovať stavy automatického režimu. Automatický režim sa drží preddefinovanej sekvencie príkazov kedy môže buď napúšťať všetky užívateľom definované nádrže v rovnaký čas, alebo sekvenčne za sebou v užívateľom definovaných časových intervaloch. Užívateľ môže ovplyvniť automatický režim zmenou jeho stavu a to tak, ako definuje *tab. č. 4.8*.



Obr. 3.1: Ilustračné schéma dvoch nádrží, zdroj [2]

3.1 Rozbor úlohy riadenia podľa štandardu ISA S88

Pre jednoduchšie pochopenie štandardu S88 je vhodné sa vrátiť trocha do histórie automatického riadenia technologických procesov. Technologický proces sa skladá z krokov, ktoré na seba nadväzujú a sú vykonávané za podmienok predpísaných technologom, teda procesným inžinierom. Sú to procesné kroky, v ktorých sa vykonávajú konkrétne činnosti a sledujú sa stavy veličín v danom procesnom kroku, napríklad napúšťanie nádrže po polovicu jej maximálnej hladiny. Činnosti a stavy veličín sú sprostredkované elektrickými zariadeniami ako sú pohony a senzory, z čoho vyplýva, že základnými členmi projekčného tímu je procesný inžinier a elektrikár. Každý rozumie svojmu oboru. Procesný inžinier musí vymyslieť a navrhnuť procesné kroky a špecifikovať činnosť a stavy pre jednotlivé kroky. Všetky tieto informácie musí následne predať elektrikárovi, ktorého úlohou bude funkcie týchto krokov zrealizovať.

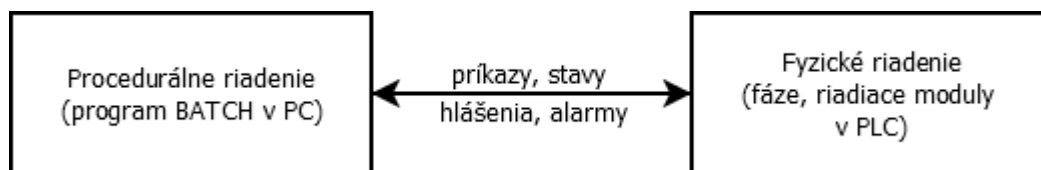
V minulosti elektrikár používal relé na realizáciu nadväznosti krokov, činností a stavov. Vytvárali sa zložité logické reléové siete. Potom nastúpila éra programovateľných automatov a relé, ako elektromechanické zariadenie, zmizlo z logických sietí. Nezmišľovali však reléové siete, tie totiž zostali na monitoroch programovacích zariadení, teda PC, v podobe programovateľných logických prvkov. Tento spôsob programovania sa nazval LADDER diagram. Neskôr sa rozšírili aj iné verzie programovania ako FBD (Function Block Diagram), IL/STL (Instruction/Statement List), SFC (Sequential Function Chart) či ST (Structured Text). Práve v jazyku STL je písaný program tejto práce. Vznikom týchto jazykov vznikol aj nový „typ“ elektrikára a to ako programátor týchto zariadení teda PLC.

Bol to veľký, takmer revolučný, skok v projektovaní priemyselných riadiacich systémov. Čo sa však zachovalo, bol nevyhnutný prenos všetkých informácií týkajúcich sa krokov, činností a stavov od procesného inžiniera k programátorovi PLC. Na presnosti a jednoznačnosti týchto informácií závisel čas tvorby a odlaďovania programu, uvádzania zariadenia do funkčnej prevádzky a náklady na skúšobnú prevádzku. Spracovávanie týchto informácií (funkčné popisy, špecifikácie) stále predstavujú obrovskú a časovo náročnú prácu pre procesných inžinierov a vo výsledku značne ovplyvňujú cenu projektov. Tento problém sa stal motívom na vytvorenie programového štandardu BATCH, pre ktorý bol vytvorený vyššie zmienený medzinárodný štandard S88.

Programovanie typu BATCH spočíva v tom, že program určený pre technologický proces sa rozdelí na programovanie krokov a programovanie činností a stavov. Kroky sa týkajú procesu, teda majú rozhodovací charakter a riešia kedy sa aký krok uskutoční, čím začne, čím skončí, ako dlho potrvá, čo sa v ňom stane a aký krok bude za ním nasledovať. Kroky nie sú nijak elektricky prepojené s technickým zariadením v procese. Na programovanie týchto procesných krokov sú vytvorené softwarové prostredia v PC, v ktorých môže jednoducho programovať sám procesný inžinier.

Každému kroku obecné pripadajú nejaké činnosti a stavy, ktoré už sú elektricky spojené s procesom, napríklad aby sa tank plnil je potrebné otvoriť ventil. Preto musia

byť naprogramované v PLC. Týmto programom v PLC sa v štandarde S88 hovorí fáze. Zjednodušený výklad programovania typu BATCH je znázornený na obr.3.2.



Obr. 3.2: Programovanie typu BATCH

Spoločnosť COMPAS poskytla pre výučbu študentov na FEKT kompletnú aplikáciu projektu „napúšťanie, miešanie a vypúšťanie nádrží“, ktorá zahŕňa program v prostredí BATCH aj program potrebných fáz v PLC. Študenti si teda môžu vyskúšať vytváranie úloh napúšťania nádrží v prostredí BATCH, vidia ako jednoducho a elegantne je možné dávkové procesy vytvárať a ovládať, avšak nevedia ako funguje štruktúra prepojenia programu v PC a v PLC, nevidia nadväznosti medzi krokmi a fázami, čím sa celá úloha pre nich stáva netransparentná. Práve z tohto dôvodu je aj táto práca riešená v jazyku STEP7 a nie v prostredí BATCH. Študenti si takto jednoducho osvoja vytvorené fáze v jazyku STEP7 a následne vytvoria v podstate náhradu prostredia BATCH, čím získajú základné znalosti o tom ako to funguje.

Tento spôsob programovania dávkových procesov v jazyku STEP7 využívajú aj menšie firmy a podniky a to buď z dôvodu finančného, alebo preto, že sa nezaoberajú dostatočne veľkými procesmi a činnosťami. Prístroje a príslušenstvo potrebné na vytvorenie programu v STEP7 je nákladovo lacnejšie ako prístroje a príslušenstvo potrebné na vytvorenie programu v prostredí BATCH. Taktiež sa BATCH využíva prevažne na väčšie procesy a činnosti, čo neplatí v tomto prípade.

3.2 Požiadavky zadania

3.2.1 Funkčný popis

1. Štart programu
2. Začne sa napúšťať nádrž číslo 1
3. Po desiatich sekundách sa začne napúšťať nádrž číslo 2
4. Pri napustení nádrže na hladinu úrovne 40 sa spustí miešanie
5. Dosiahne sa maximálna hladina v nádrži
6. Obsah nádrže sa mieša ďalších 10s
7. Po uplynutí tejto doby sa obsah nádrže začne vypúšťať
8. Pri vypúšťaní sa mixuje po hladinu úrovne 40
9. Vypne sa miešanie
10. Vypúšťanie končí dosiahnutím minimálnej hladiny v nádrži.

3.2.2 Vizualizácia

1. Technologický obraz – animácia nádrží
2. Stĺpcový diagram skutočnej a žiadanej hodnoty

3.2.3 Režimy a módy ovládania

Celý riadiaci program musí byť prevádzkovaný v automatickom (AUT) a manuálnom režime (MAN). V automatickom režime musí byť rešpektované štandardné požiadavky na stavy READY, RUN, HELD, DONE, STOPPED, ABORTED a na príkazy, ktorými sa tieto stavy uskutočňujú ako sú napríklad START, PAUSE, HOLD, RESUME, STOP. Jednotlivé zariadenia a stroje musia byť schopné prevádzky v manuálnom režime tak, aby ich prevádzka bola bezpečná pre obsluhu aj vlastné zariadenie. Manuálna prevádzka je možná len keď je celý proces prepnutý do režimu MAN. Naopak, automatická prevádzka je možná len vtedy, keď všetky zariadenia sú v režime AUT.

V automatickom režime ovládajú zariadenie fáze. Preto je tu podmienka, že ventil sa môže otvoriť alebo zatvoriť len v prípade, že fáza beží, teda fáza ovláda ventil. Ako stav fáze tu slúži indikácia či je fáza aktívna, teda či beží, alebo už svoju funkčnosť dokončila. Aby fáza bola schopná ovládať zariadenie pomocou svojich parametrov, musí byť zariadenie v automatickom režime.

Režimy fáz sú opäť automatické a manuálne. Proces, ktorý je nadradený fázam sa môže spustiť len vtedy ak sú všetky fázy a zariadenia v automatickom režime. U fázy napúšťania a vypúšťania sa fáza ukončí akonáhle dosiahne aktuálna hladina požadovanú, predom stanovenú, hodnotu. Táto hodnota je zadaná ako žiadaná hodnota v SCADA. Vo chvíli kedy sa má miešacia fáza vypnúť, musí motoru poslať bit na vypnutie a nastaviť nulové otáčky.

4 ŠTRUKTÚRA PROGRAMU

Program sa tvoril z úplných základov, kedy zo začiatku bolo potrebné vytvoriť kódy pre základné funkcie programu, teda jeho riadiace moduly – ventily a motory. Následne sa program rozšíril o fázu a ovládanie.

4.1 Tvorba ventilu

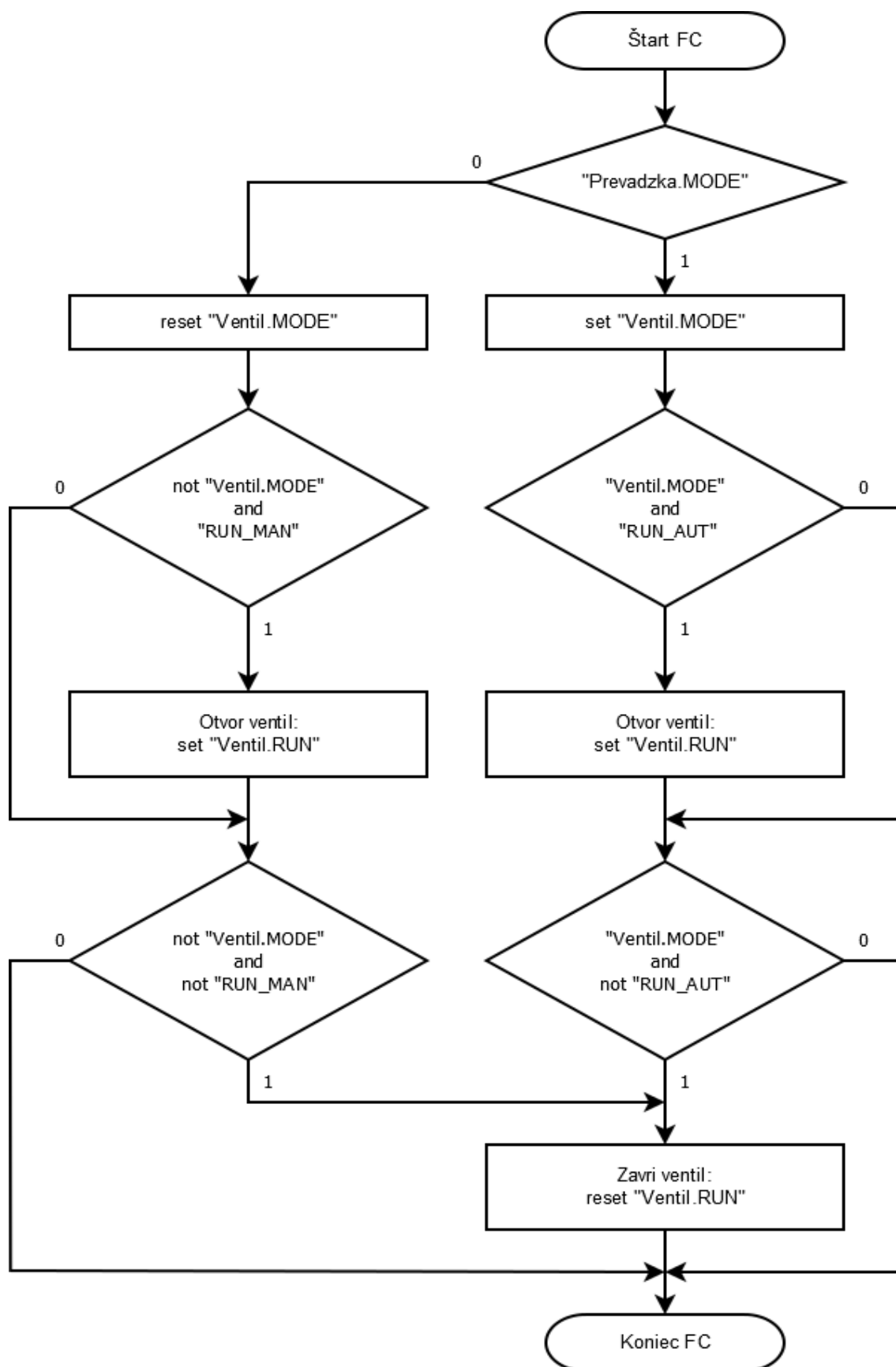
Pri vytvorení ventilu sa vychádza z jeho ovládania a jeho komunikácie s ostatnými modulmi. Ventil bude prevádzkovaný v dvoch režimoch, teda je potrebné mať informácie v akom režime sa nachádza a či je ventil aktívny alebo neaktívny. Teda celkom je potrebné mať 2 stavové informácie: MODE a RUN. Aby sa ventil do príslušného módu dostal, musí prijať povel MODE a aby sa aktivoval či deaktivoval musí prijať povel RUN_AUT alebo RUN_MAN podľa toho v akom režime sa nachádza. Z týchto požiadaviek na stavy a príkazy ventilu sa vytvorí užívateľský definovaný typ UDT s názvom „DT_VENTIL“. Štruktúra tohto UDT je uvedená v tab. č. 4.1.

Tab. č. 4.1: Užívateľsky definovaný typ DT_VENTIL

Adresa	Názov	Typ	Počiatočná hodnota	Popis premennej
0.0		STRUCT		
+0.0	STAV	STRUCT		
+0.0	MODE	BOOL	FALSE	indikuje režim MAN [0] / AUT [1]
+0.1	RUN	BOOL	FALSE	indikuje beh ventilu, zatvorený [0] / otvorený [1]
=2.0		END_STRUCT		
+2.0	PRIKAZ	STRUCT		
+0.0	MODE	BOOL	FALSE	nastavenie režimu, MAN [0] / AUT [1]
+0.1	RUN_AUT	BOOL	FALSE	režim AUT, zatvorenie [0] / otvorenie [1] ventilu
+0.2	RUN_MAN	BOOL	FALSE	režim MAN, zatvorenie [0] / otvorenie [1] ventilu
=2.0		END_STRUCT		
=4.0		END_STRUCT		

Základný kód ventilu je ďalej tvorený v programovej funkcii FC „FC_VENTIL“, ktorá ako parameter obsahuje práve tento vytvorený UDT (vid' obr. 4.2). To umožní použitie vyššie definovaných premenných v tejto programovej funkcii. Parameter má názov DATA a je to vstupne-výstupný parameter IN_OUT.

Je potrebné aby program ventilu bol schopný manipulovať s ventilom v oboch režimoch, teda v manuálnom aj automatickom režime a v týchto režimoch daný ventil aktivovať či deaktivovať na základe prijatia hodnôt premenných vyššie definovaných. Štruktúru programu ventilu zobrazuje vývojový diagram na obr. 4.1.



Obr. 4.1: Vývojový diagram funkcie ventilu

"FC_VENTIL"

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
DATA	DT_VENTIL	0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Obr. 4.2: Vstupno-výstupný parameter DATA funkcie FC_VENTIL

4.2 Tvorba motoru

Podobne ako pri správnej funkčnosti ventilu, aj funkčnosť motora vychádza z požiadaviek na jeho ovládanie a komunikáciu s ostatnými modulmi. Je teda potrebné mať informácie o stave motora, jeho činnosti. To zaisťujú stavové informácie MODE a RUN. Aby motor mohol zmeniť svoj režim na požadovaný, musí prijať povel MODE.

Tab. č. 4.2: Užívateľsky definovaný typ DT_MOTOR

Adresa	Názov	Typ	Počiatočná hodnota	Popis premennej
0.0		STRUCT		
+0.0	STAV	STRUCT		
+0.0	MODE	BOOL	FALSE	indikuje režim MAN [0] / AUT [1]
+0.1	RUN	BOOL	FALSE	indikuje beh motoru, vypnutý [0] / zapnutý [1]
=2.0		END_STRUCT		
+2.0	PRIKAZ	STRUCT		
+0.0	MODE	BOOL	FALSE	nastavenie režimu, MAN [0] / AUT [1]
+0.1	RUN_AUT	BOOL	FALSE	režim AUT, vypnutie [0] / zapnutie [1] motoru
+0.2	RUN_MAN	BOOL	FALSE	režim MAN, vypnutie [0] / zapnutie [1] motoru
=2.0		END_STRUCT		
+4.0	SPEED_SP	INT	0	nastavenie otáčok motora
+6.0	SPEED	INT	0	aktuálne otáčky motora
+8.0	MIX_TIME	S5TIME	S5T#0MS	aktuálna doba miešania
+10.0	TIME_POM	TIME	T#10S	nastavenie doby miešania
=14.0		END_STRUCT		

Na aktiváciu či deaktiváciu motoru musí prijať príslušný povel a to buď RUN_AUT alebo RUN_MAN. Ďalej je potrebné motoru nastaviť jeho otáčky a dobu činnosti. Na nastavenie správnych otáčok slúži SPEED_SP, na zobrazovanie aktuálnych otáčok slúži SPEED. Doba činnosti motoru je zaznamenávaná v MIX_TIME. Podľa týchto požiadaviek bol následne vytvorený užívateľsky definovaný typ „DT_MOTOR“, ktorého štruktúra je zobrazená v *tab. č. 4.2*.

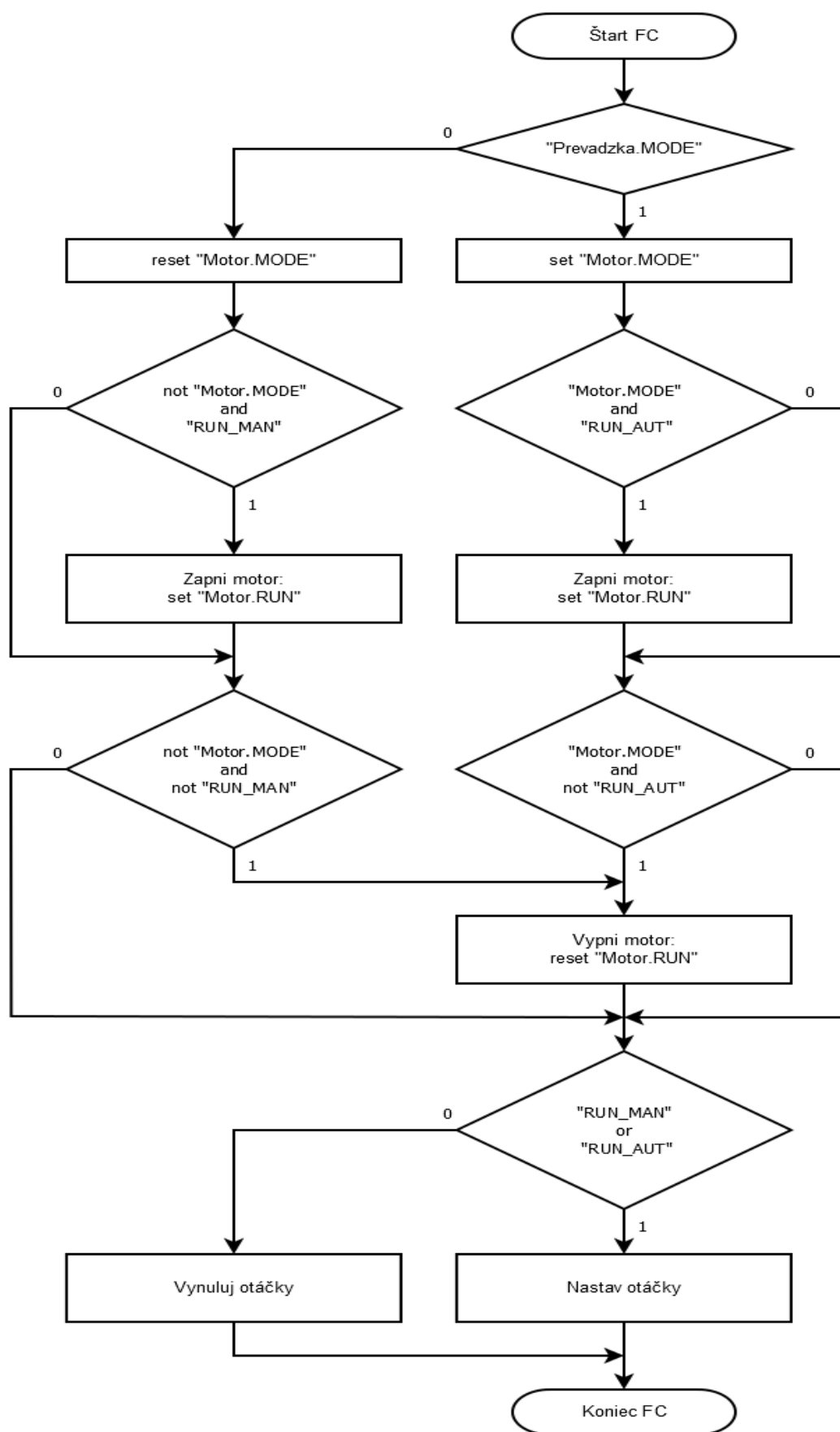
Kód programu motora je písaný v programovej funkcii „FC_MOTOR“, ktorej parametrom je vyššie definovaný UDT „DT_MOTOR“ – vid’ *obr. 4.4*. Názov tohto parametru je DATA a je deklarovaný ako vstupno-výstupný parameter, pretože jeho premenné sa v programe budú zapisovať aj čítať. Program musí vedieť ovládať motor v oboch režimoch, musí byť schopný jeho zapnutia či vypnutia, nastavenia požadovaných otáčok v oboch režimoch.

"FC_MOTOR"

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
DATA	DT_MOTOR	0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Obr. 4.3: Vstupno-výstupný parameter DATA funkcie FC_MOTOR

Vývojový diagram programu je možné vidieť na *obr. 4.4*. Ako je možné vidieť, diagram obsahuje funkcie na zapnutie a vypnutie motora v režime automatickom aj v režime manuálnom. Ďalej funkcie na nastavenie požadovaných otáčok motora a vynulovanie otáčok motora.



Obr. 4.4: Vývojový diagram funkcie motoru

4.3 Monitorovanie hladiny v nádrži

Vlastný riadiaci program reaguje na tri stavy hladiny:

- aktuálna hodnota hladiny
- horný limit hladiny
- dolný limit hladiny

Aby bol program schopný rozohnať horný alebo dolný limit hladiny, je potrebné vytvoriť stavové informácie HLADINA_SP_UP a HLADINA_SP_DOWN. Na detekciu aktuálnej hladiny v nádrži slúži premenná HLADINA. Tieto tri stavové informácie vytvárajú užívateľsky definovaný typ „DT_HLADINA“, ktorého štruktúra je zobrazená v tab. č. 4.3.

Tab. č. 4.3: Užívateľsky definovaný typ DT_HLADINA

Adresa	Názov	Typ	Počiatková hodnota	Popis premennej
0.0		STRUCT		
+0.0	HLADINA	INT	0	aktuálna hodnota hladiny
+2.0	HLADINA_SP_UP	INT	0	horný limit hladiny
+4.0	HLADINA_SP_DOWN	INT	0	dolný limit hladiny
=6.0		END_STRUCT		

4.4 Vytvorenie fáz procesu

Na komunikáciu programu medzi rozličnými fázami procesu počas celého priebehu procesu, je potrebné vytvoriť tri stavové informácie o fáze: RUN, DONE a POM. Informácie nesené v premennej RUN súvisia s behom fáze, teda či je fáza aktívna, alebo nie. Premenná DONE slúži na identifikáciu skončenia fáze a premenná POM je využitá na komunikáciu medzi nasledujúcimi fázami. Z týchto premenných bol následne vytvorený užívateľsky definovaný typ „DT_FAZA“ zobrazený v tab. č. 4.4.

Tab. č. 4.4: Užívateľsky definovaný typ DT_FAZA

Adresa	Názov	Typ	Počiatková hodnota	Popis premennej
0.0		STRUCT		
+0.0	STAV	STRUCT		
+0.0	RUN	BOOL	FALSE	indikuje beh fáze, neprebíha [0] / prebieha [1]
+0.1	DONE	BOOL	FALSE	indikuje koniec fáze [1]
+0.2	POM	BOOL	FALSE	slúži na komunikáciu medzi fázami
=2.0		END_STRUCT		
=2.0		END_STRUCT		

Napúšťacia fáza je fáza, ktorá celý proces napúšťania, miešania a vypúšťania nádrží inicializuje. Je preto potrebné aby táto fáza bola schopná ovládať napúšťací ventil a to v oboch režimoch, práve na to slúži parameter „*DATA_VENTIL_NAP*“ typu „*DT_VENTIL*“ obsahujúci práve príkazy a informácie o napúšťacom ventile. Napúšťacia fáza je tvorená v programovej funkcii „*FAZA_NAP*“, ktorá obsahuje tri parametre. Prvý parameter je vyššie zmienený, druhým parametrom je „*DATA_HLADINA*“ obsahujúci informácie o aktuálnom stave hladiny a posledným parametrom je „*DATA_FAZA_NAP*“ nesúci informácie o tejto fáze. Parametre sú na nasledujúcom obrázku.

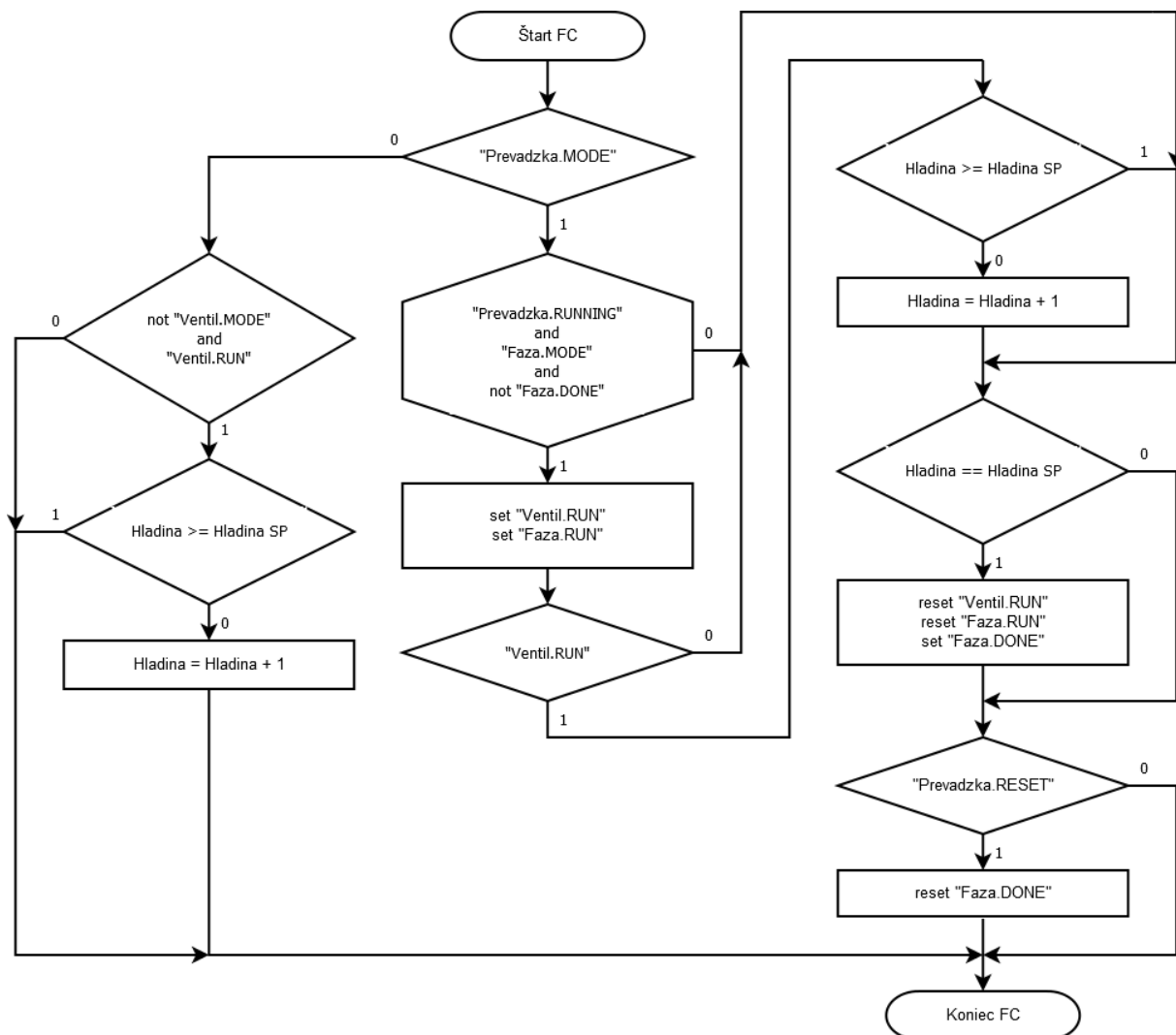
"FAZA NAP"

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
DATA_FAZA_NAP	DT_FAZA	0.0	
DATA_VENTIL_NAP	DT_VENTIL	2.0	
DATA_HLADINA	DT_HLADINA	6.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Obr. 4.5: Parametre funkcie napúšťacej fáze

Program napúšťacej fáze je tvorený tak, že obsahuje funkcie na napúšťanie hladiny v oboch režimoch. Fáza je schopná ovládať napúšťací ventil, ktorým koriguje stav hladiny v nádrži. Napúšťacia fáza svoju činnosť ukončuje po dosiahnutí maximálnej hladiny. Nasledujúci obrázok zobrazuje vývojový diagram napúšťacej fáze.

Manuálny režim fáze je reprezentovaný napúšťacím ventilom, kedy v manuálnom režime užívateľ ventil otvorí, čím zapne napúšťaciu fázu v manuálnom režime a pri zatvorení ventilu túto fázu vypne. Napúšťacia fáza sa automaticky vypne po dosiahnutí maximálnej hladiny aj v manuálnom režime.



Obr. 4.6: Vývojový diagram napúšťacej fáze

Zaujímavosťou v tejto fáze je práve simulácia napúšťania nádrže, teda postupné zvyšovanie aktuálnej hladiny v nádrži pri otvorenom napúšťacom ventile. Zdrojový kód simulácie napúšťania hladiny je zobrazený na obr. 4.7. Podobne bude vytvorená aj simulácia pri vypúšťaní nádrže použitá vo vypúšťacej fáze. Práve informácie o tejto simulácii sa uchovávajú v parametri „*DATA_HLADINA*“.

```

L    #DATA_HLADINA.HLADINA
L    #DATA_HLADINA.HLADINA_SP_DOWN

<=I
JC    S1
L    #DATA_HLADINA.HLADINA

L    1
-I
T    #DATA_HLADINA.HLADINA

```

Obr. 4.7: Ukážka kódu simulovania napúšťania nádrže

Miešacia fáza je tvorená v programovej funkcii „FAZA_MIX“. Táto funkcia obsahuje 5 parametrov, vďaka ktorým zabezpečuje komunikáciu s predošlou aj nadchádzajúcou fázou. Týmito parametrami sú: parameter „DATA_FAZA_NAP“ nesúci informácie o predošlej fáze, „DATA_FAZA_MIX“ nesúci informácie o aktuálnej fáze, „DATA_FAZA_VYP“ nesúci informácie pre nasledujúcu fázou, „DATA_HLADINA“ obsahujúci informácie o aktuálnom stave hladiny, „DATA_MOTOR“ slúžiaci na ovládanie motora.

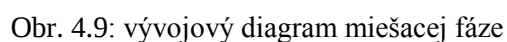
"FAZA_MIX"

Name	Data Type	Address	Comment
IN		0.0	
CASOVAC	Timer	0.0	
CAS	S5Time	2.0	
OUT		0.0	
IN_OUT		0.0	
DATA_FAZA_NAP	DT_FAZA	4.0	
DATA_FAZA_MIX	DT_FAZA	6.0	
DATA_FAZA_VYP	DT_FAZA	8.0	
DATA_MOTOR	DT_MOTOR	10.0	
DATA_HLADINA	DT_HLADINA	24.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Obr. 4.8: Parametre funkcie miešacej fáze

Program motora obsahuje funkcie na zapnutie a vypnutie motora, nastavenie otáčok motora v oboch režimoch. V automatickom režime sa proces riadi preddefinovaným postupom, kedy si miešacia fáza sníma aktuálny stav hladiny a pri hodnote hladiny 40 fáza zapína motor. Zároveň bežia obe fázy, teda napúšťacia a miešacia. Po skončení napúšťacej fáze dostane miešacia fáza signál o jej skončení a pustí sa časovač, ktorý určuje dobu miešania. Túto dobu si môže užívateľ zmeniť na ľubovoľnú hodnotu avšak preddefinovaná doba miešania je 10 sekúnd. Po uplynutí tejto doby sa vyšle signál nasledujúcej fáze, signalizujúci jej začiatok. Teda zároveň beží fáza miešacia a fáza vypúšťacia. Miešacia fáza sa ukončí po dosiahnutí požadovanej hodnoty

Manuálny režim miešacej fáze je tvorený prostredníctvom ovládania motora. Kedy užívateľ zapnutím motora zapína miešaciu fázu a jeho následným vypnutím vypne túto fázu.



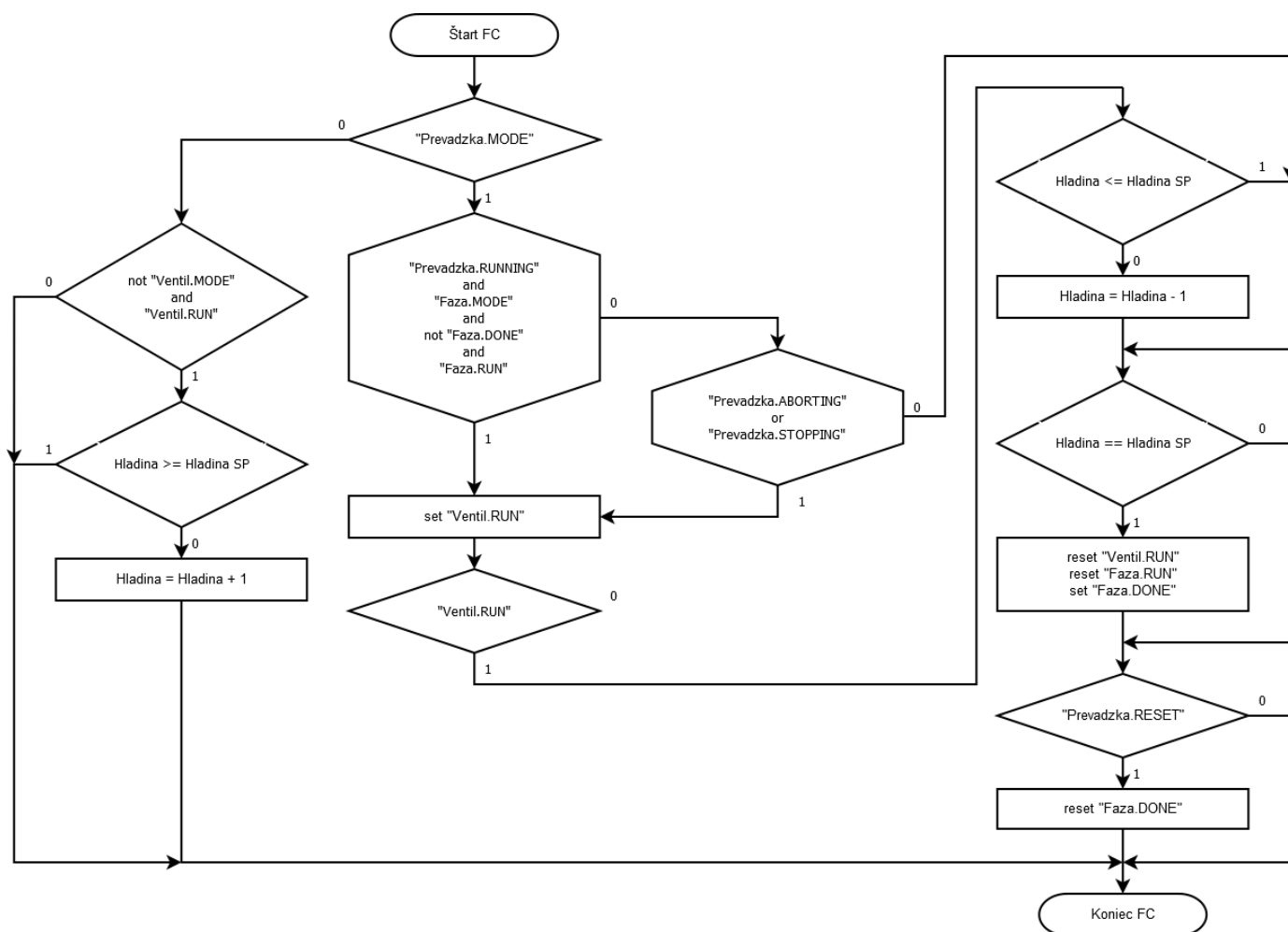
Poslednou fázou je fáza vypúšťacia. Program pre túto fázu je tvorený v programovej funkcii „FAZA_VYP“, ktoré má 4 parametre, a to „DATA_FAZA_NAP“ nesúci informácie o napúšťacej fáze, „DATA_FAZA_VYP“ nesúci informácie o aktuálnej fáze, „DATA_HLADINA“ na kontrolu stavu hladiny, „DATA_VENTIL_VYP“ na ovládanie vypúšťacieho ventilu.

"FAZA_VYP"

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
DATA_FAZA_NAP	DT_FAZA	0.0	
DATA_FAZA_VYP	DT_FAZA	2.0	
DATA_VENTIL_VYP	DT_VENTIL	4.0	
DATA_HLADINA	DT_HLADINA	8.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Obr. 4.10: Parametre vypúšťacej funkcie

Program je tvorený z funkcii slúžiacich na vypustenie obsahu nádrže. Fáza je schopná ovládať vypúšťací ventil. Vývojový diagram pre vypúšťaciu fázu je možné vidieť na obr. 4.11. Vypúšťacia fáza začína svoju prevádzku po prijatí signálu od predošlej fázy, kedy otvorí vypúšťací ventil a obsah nádrže sa pomaly vypúšťa. Po dosiahnutí minimálnej hladiny fáza ventil zatvorí a ukončuje svoju prevádzku. Ukončením tejto fázy sa ukončuje aj proces pre konkrétnu nádrž. Manuálny režim vypúšťacej fázy je tvorený prostredníctvom vypúšťacieho ventilu. Užívateľ otvorením vypúšťacieho ventilu zapne vypúšťaciu fázu v manuálnom režime a naopak, zatvorením ventilu fázu vypne. Vypúšťacia fáza sa automaticky ukončuje po dosiahnutí minimálnej hladiny v nádrži aj v režime manuálnom.



Obr. 4.11: Vývojový diagram vypúšťacej fáze

4.5 Strom volaní

Aby bol program spustiteľný a vykonával svoju funkciu správne, je potrebné vyššie vytvorené funkcie volať. Všetky vytvorené ventily sa volajú z programovej funkcie „*CALL VENTIL*“, podobne vytvorené motory sú volané z funkcie „*CALL MOTOR*“. Následne sa tieto dve funkcie volajú z bloku OB1. Fáze procesu sú volané z funkcie „*CALL FAZE*“, kde sa volá napúšťacia, miešacia a vypúšťacia fáza každej nádrže. Táto funkcia sa volá z bloku OB35 a nie z bloku OB1 a to práve pre z dôvodu simulovania napúšťania a vypúšťania hladiny v nádrži. Blok OB35 má periódu volania 100ms, čo zabezpečuje dostatočné aktualizovanie aktuálnej hodnoty simulovanej hladiny v týchto fázach. Po vytvorení ďalšej nádrže je potrebné zavolať každý riadiaci modul a fázu tejto nádrže. Pre užívateľa to znamená pripísanie volacích funkcií do vyššie zmienených funkcií „*CALL VENTIL*“, „*CALL MOTOR*“ a „*CALL FAZE*“. Kompletný strom volaní je zobrazený na nasledujúcom obrázku.

```

OB1
└─> CALL FC1 (FC PREVADZKA)
└─> CALL FC4 (CALL VENTIL)
      └─> CALL FC2 (FC VENTIL - ventil nap 1)
      └─> CALL FC2 (FC VENTIL - ventil vyp 1)
      └─> CALL FC2 (FC VENTIL - ventil nap 2)
      └─> CALL FC2 (FC VENTIL - ventil vyp 3)
      . . .
└─> CALL FC5 (CALL MOTOR)
      └─> CALL FC3 (FC MOTOR - motor 1)
      └─> CALL FC3 (FC MOTOR - motor 2)
      . . .

OB35
└─> CALL FC9 (CALL FAZE)
      └─> CALL FC6 (FAZA NAP - tank 1)
      └─> CALL FC7 (FAZA MIX - tank 1)
      └─> CALL FC8 (FAZA VYP - tank 1)
      └─> CALL FC6 (FAZA NAP - tank 2)
      └─> CALL FC7 (FAZA MIX - tank 2)
      └─> CALL FC8 (FAZA VYP - tank 2)
      . . .

```

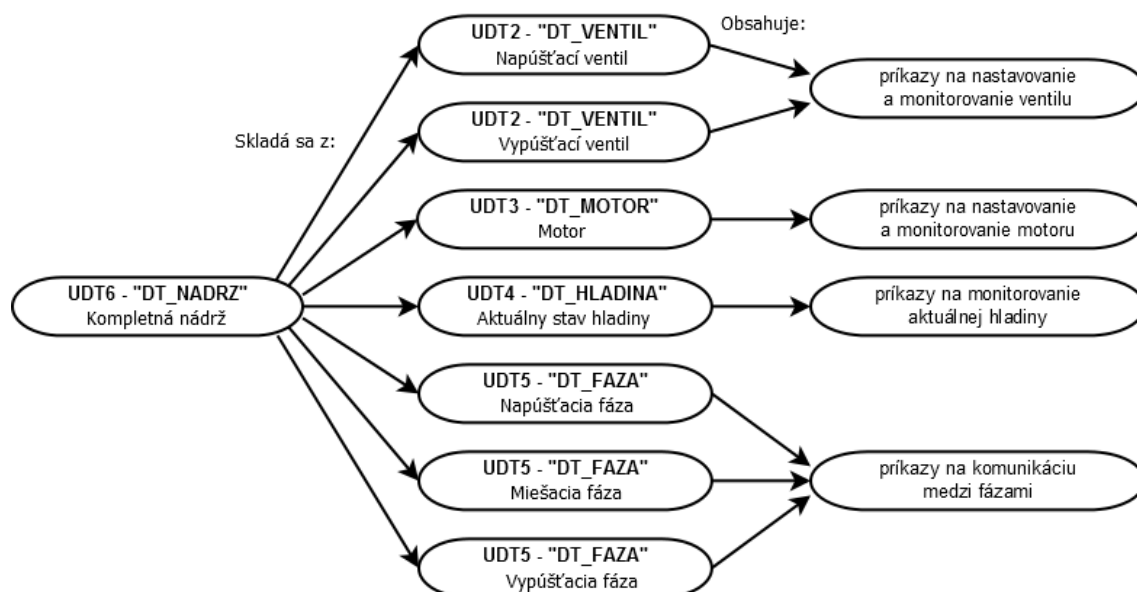
Obr. 4.12: Strom volaní funkcií

4.6 Štruktúra kompletnej nádrže

Aby bolo možné poskladať kompletnú nádrž z už vytvorených riadiacich modulov a fáz, bol vytvorený UDT „DT_NADRZ“. Tento UDT obsahuje všetky ostatné UDT vytvorené a popísané vyššie. Celková štruktúra je zobrazená v nasledujúcej tabuľke. Grafické zobrazenie štruktúry nádrže je zobrazené na obr. 4.13. Nádrž je zložená z dvoch ventilov – napúšťacieho a vypúšťacieho, z motoru a troch fáz – napúšťacej, miešacej a vypúšťacej.

Tab. č. 4.5: Užívateľsky definovaný typ DT_NADRZ

Adresa	Názov	Typ	Počiatočná hodnota	Popis premennej
0.0		STRUCT		
+0.0	VENTIL_NAP	"DT_VENTIL"		napúšťací ventil nádrže
+4.0	VENTIL_VYP	"DT_VENTIL"		vypúšťací ventil nádrže
+8.0	MOTOR	"DT_MOTOR"		motor nádrže
+22.0	FAZA_NAP	"DT_FAZA"		napúšťacia fáza
+24.0	FAZA_MIX	"DT_FAZA"		miešacia fáza
+26.0	FAZA_VYP	"DT_FAZA"		vypúšťacia fáza
+28.0	HLADINA	"DT_HLADINA"		aktuálna hladina v nádrži
=34.0		END_STRUCT		



Obr. 4.13: Štruktúra hotovej nádrže

Následne je vytvorený dátový blok „DB_NADRZ“ obsahujúci premenné typu „DT_NADRZ“. To znamená, že každá premenná v tomto bloku je vlastne jedna kompletná nádrž. Pre ukážku je v tabuľke nižšie zobrazená štruktúra dátového bloku obsahujúca práve tri nádrže. Ak by užívateľ chcel pridať do programu ďalšie nádrže, jednoducho pridá ďalšie premenné.

Tab. č. 4.6: Dátový blok DB_NADRZ

Adresa	Názov	Typ	Počiatočná hodnota	Popis premennej
0.0		STRUCT		
+0.0	TANK1	"DT_NADRZ"		nádrž č. 1
+34.0	TANK2	"DT_NADRZ"		nádrž č. 2
+68.0	TANK3	"DT_NADRZ"		nádrž č. 3
=102.0		END_STRUCT		

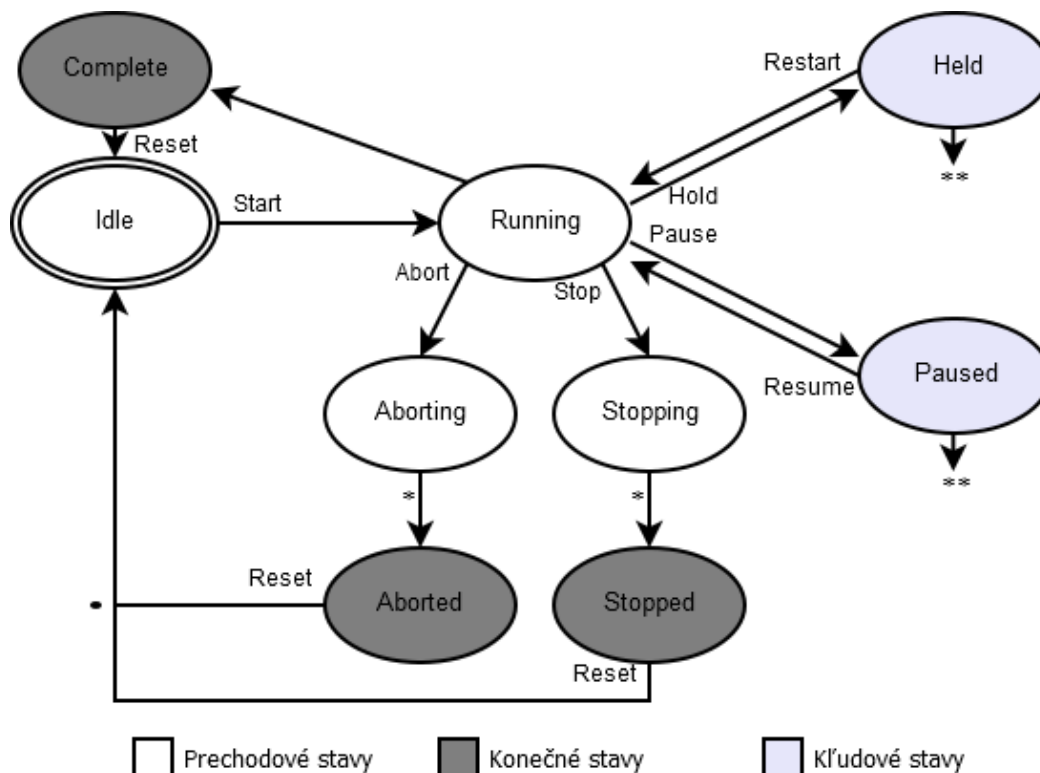
4.7 Prevádzka a ovládanie procesu

Aby program fungoval v režime manuálnom alebo v režime automatickom a bolo možné jednoduché prepínanie medzi týmito režimami, bol vytvorený UDT „DT_PREVADZKA“. Tento UDT obsahuje premenné slúžiace na zmenu režimov a stavov programu. Následne, aby bolo možné tieto premenné použiť v programe, bol vytvorený dátový blok „DB_NADRZ“, obsahujúci premennú typu „DT_PREVADZKA“. Štruktúru vyššie zmieneného užívateľského definovaného typu aj dátového bloku zobrazuje nasledujúca tabuľka.

Tab. č. 4.7: Užívateľsky definovaný typ DT_PREVADZKA

Adresa	Názov	Typ	Počiatočná hodnota	Popis premennej
0.0		STRUCT		
+0.0	PRIKAZ	STRUCT		štruktúra obsahujúca príkazy na zmeny stavov
+0.0	Start	BOOL	FALSE	
+0.1	Stop	BOOL	FALSE	
+0.2	Hold	BOOL	FALSE	
+0.3	Restart	BOOL	FALSE	
+0.4	Abort	BOOL	FALSE	
+0.5	Reset	BOOL	FALSE	
+0.6	Pause	BOOL	FALSE	
+0.7	Resume	BOOL	FALSE	
+1.0	Mode	BOOL	FALSE	
=2.0		END_STRUCT		
+2.0	STAV	STRUCT		štruktúra obsahujúca indikátory aktuálnych stavov
+0.0	Idle	BOOL	FALSE	
+0.1	Running	BOOL	FALSE	
+0.2	Complete	BOOL	FALSE	
+0.3	Paused	BOOL	FALSE	
+0.4	Held	BOOL	FALSE	
+0.5	Stopping	BOOL	FALSE	
+0.6	Stopped	BOOL	FALSE	
+0.7	Aborting	BOOL	FALSE	
+1.0	Aborted	BOOL	FALSE	
+1.1	Mode	BOOL	FALSE	
=2.0		END_STRUCT		
=4.0		END_STRUCT		

V automatickom režime je schopný užívateľ ovplyvňovať stav procesu tak ako zobrazuje nasledujúci obrázok.



Obr. 4.14: Diagram prechodov

* prechod do ďalšieho stavu nastáva automaticky práve vtedy, keď program môže byť zastavený bez akéhokoľvek porušenia, v tomto prípade prejde do nasledujúcich stavov po vyprázdnení obsahu nádrže

** v stavoch *HELD* a *PAUSED* môžu byť použité príkazy *Abort* a *Stop*, avšak pre priehľadnosť blokového diagramu nie sú zobrazené, všetky možné prechody medzi stavmi zobrazuje matica prechodov stavov v tab. č. 4.8.

Tab. č. 4.8: Matica prechodov stavov

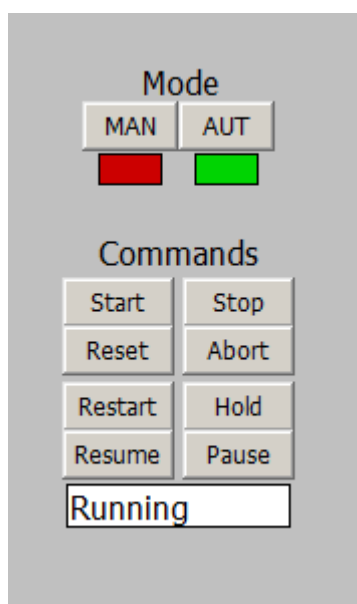
Príkaz		Start	Stop	Hold	Restart	Abort	Reset	Pause	Resume
Počiatočný stav	Konečný stav	Matica prechodov stavov							
Idle	x	Running	x	x	x	x	x	x	x
Running	Complete	x	Stopping	Held	x	Aborting	x	Paused	x
Complete	x	x	x	x	x	x	Idle	x	x
Paused	x	x	Stopping	Held	x	Aborting	x	x	Running
Held	x	x	Stopping	x	Running	Aborting	x	x	x
Stopping	Stopped	x	x	x	x	Aborting	x	x	x
Stopped	x	x	x	x	x	Aborting	Idle	x	x
Aborting	Aborted	x	x	x	x	x	x	x	x
Aborted	x	x	x	x	x	x	Idle	x	x

4.8 Uživatelské rozhranie

Uživatelské rozhranie je vytvorené v programe WinCC flexible a slúži na jednoduchú obsluhu jednej alebo viacerých nádrží a to v režime manuálnom alebo režime automatickom.

Užívateľ je schopný prepínať medzi automatickým a manuálnym režimom tlačidlami MAN a AUT, pričom činnosť týchto režimov je znázornená farebnými indikátormi. Pri červenom indikátore je režim nečinný, teda vypnutý a pri zelenom je režim aktívny.

V automatickom režime má užívateľ k dispozícii osem tlačidiel slúžiacich na zmenu stavov podľa tab. č. 4.8. Ovládací panel spolu s tlačidlami na prepínanie režimu programu je zobrazený na nasledujúcom obrázku.



Obr. 4.15: Ovládací panel automatického režimu

Príkaz **Start** slúži na spustenie programu. Po prepnutí do automatického režimu program sa automaticky nastaví do stavu *Idle* a čaká na prijatie toho príkazu. Po prijatí mení svoj stav na *Running* a začína proces napúšťania, miešania a vypúšťania nádrží. Príkaz Start môže používateľ použiť len v stave *Idle*.

Príkazom **Reset** je možné program po správnom dokončení znovu spustiť. Program prechádza po správnom dokončení, teda po vyprázdnení obsahu všetkých nádrží, do stavu *Complete*. Prijatím príkazu Reset prechádza do stavu *Idle* a čaká na opätovné spustenie užívateľom. Príkaz je použiteľný len v stavoch *Complete*, *Aborted* alebo *Stopped*.

Príkazom **Abort** užívateľ zastaví akúkoľvek činnosť programu. Program prechádza do stavu *Aborting*, kedy sa otvoria vypúšťacie ventily a obsah nádrží sa začne

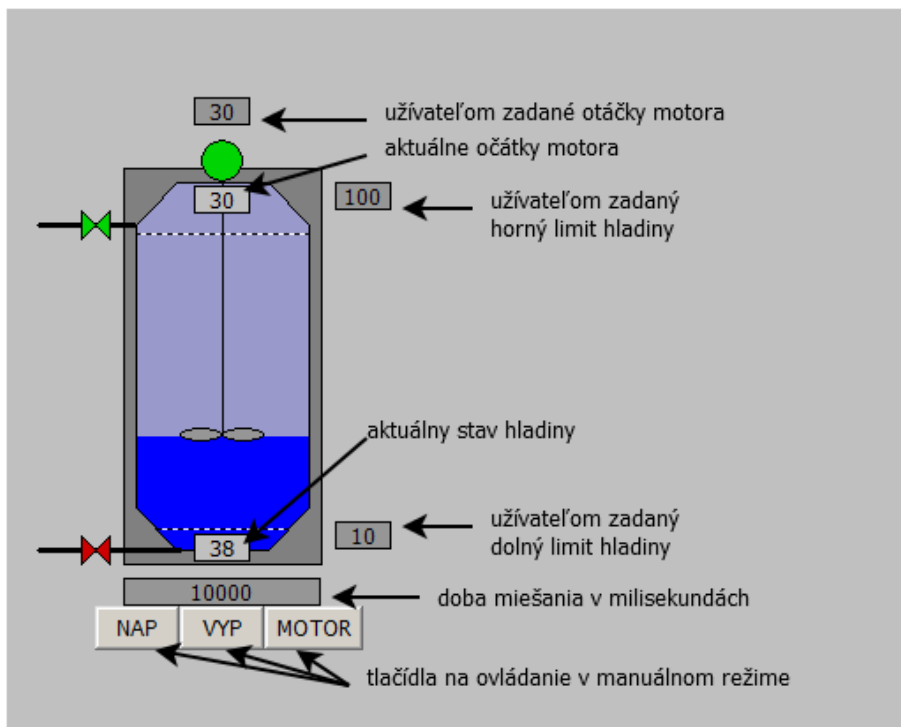
vyprázdňovať. Po vyprázdnení všetkých nádrží program prejde do stavu *Aborted*, kedy užívateľ môže použiť príkaz *Reset* na zmenenie stavu na *Idle*. Tento príkaz užívateľ môže použiť v stavoch *Running*, *Paused*, *Held*, *Stopping* a *Stopped*. Príkaz slúži na nútené ukončenie programu.

Príkaz **Stop** slúži na korektné ukončenie programu. Program po prijatí tohto príkazu prechádza do stavu *Stopping*, kedy vyprázdni zvyšný obsah nádrží. Po vyprázdnení program prechádza do stavu *Stopped*, kedy užívateľ príkazom *Reset* nastaví program do stavu *Idle*. Príkaz *Stop* je použiteľný v stavoch *Paused*, *Held* a *Running*.

Hold pozastaví program. Užívateľ tento príkaz využije napríklad pri poruche jedného nádrže. Po prijatí tohto povelu programom prejde program do stavu *Held*, v ktorom zostane pokým užívateľ nepoužije príkaz *Restart*. Po odstránení poruchy používateľ použije *Restart* a program sa vráti do bežiaceho stavu *Running*, kedy pokračuje v prevádzke. Príkaz *Hold* je použiteľný len v stave *Running* a *Paused*. Tento príkaz sa používa na dlhodobšie zastavenie programu. **Restart** slúži na opätovný štart programu. Je použiteľný len v stave *Held*.

Príkaz **Pause** slúži na krátkodobé pozastavenie programu. Po prijatí príkazu program prejde do stavu *Paused*, kedy čaká na prijatie príkazu *Resume*. Po použití príkazu *Resume* sa program vráti do bežiaceho stavu *Running* a pokračuje v prevádzke. Príkaz *Pause* je možné použiť iba počas prevádzky systému, teda v stave *Running*. Príkaz **Resume** je využitý na opätovné spustenie programu po prechode do stavu *Paused*, preto je použiteľný len v tomto stave.

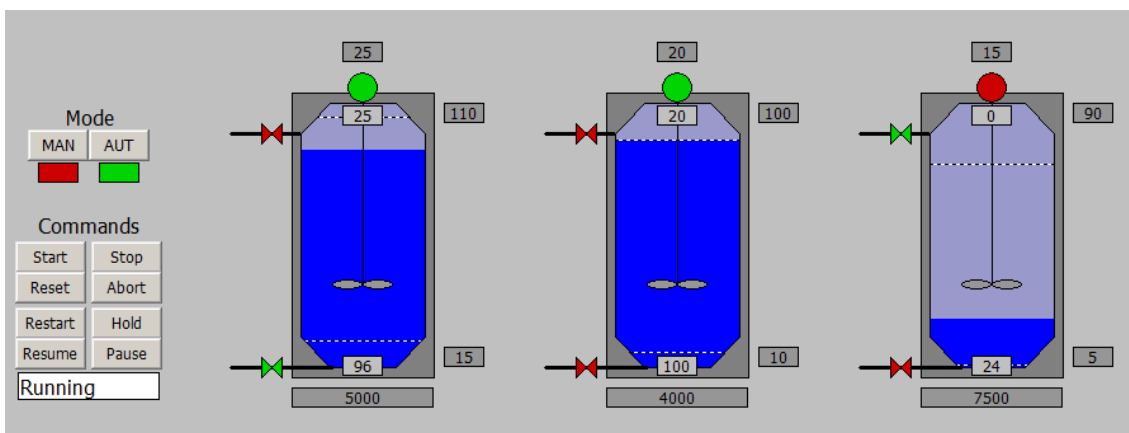
V manuálnom režime má užívateľ k dispozícii tri tlačidlá a to tlačidlo NAP pre ovládanie napúšťacieho ventilu, tlačidlo VYP pre ventil vypúšťací a tlačidlo MOTOR na spustenie a vypnutie motora. V tomto režime má užívateľ dovolené operovať s týmito riadiacimi prvkami tak, ako uváži za vhodné, teda pri ich ovládaní v manuálnom režime nie je obmedzovaný. Ventily a motory sú vo vizualizácii zobrazené indikátormi, ktoré sú farebne odlíšené. Červená farba indikátora znamená zatvorený ventil alebo vypnutý motor, zatiaľ čo zelená farba znamená opak, teda otvorený ventil a zapnutý motor.



Obr. 4.16: Ovládanie nádrže v manuálnom režime, indikátory a parametre nádrže

V oboch režimoch je potrebné pred spustením zadať parametre nádrží. Do týchto parametrov spadajú minimálna a maximálna hladina v nádrži a otáčky motora. Užívateľ taktiež môže zadať dobu miešania obsahu nádrže v miešacej fáze, avšak tento parameter je preddefinovaný na desať sekúnd, teda tento parameter nie je potrebné zadávať pre správnu funkčnosť programu.

Vo vizualizácii sa ďalej nachádzajú indikátory slúžiace na sledovanie aktuálnej hladiny v nádrži a aktuálnych otáčok motora. Prehľadný popis prvkov vizualizácie a ich funkcie sú zobrazené na obr. 4.16. Celkové užívateľské rozhranie počas behu programu v automatickom režime je zobrazené na obr. 4.17.



Obr. 4.17: Kompletné užívateľské rozhranie, ukážka bežiaceho programu v automatickom režime

5 ZÁVER

Ako už bolo zmienené na začiatku práce, táto práca je určená ako príkladný materiál k štúdiu pre laboratórne cvičenie výuky programovania dávkového procesu v systéme BATCH. Konkrétne sa jedná o systém COMES BATCH, spoločnosti COMPAS s.r.o. V tomto systéme budú študenti vytvárať dávky na základe receptúr a tie sa zasa budú tvoriť pomocou operácií a fáz. Fáze, ktoré tu sú použité sú súčasťou systému COMES BATCH a ich analýza, vzhľadom na ich zložitosť, je mimo časový rámec semestra.

Študenti by boli postavený pred úlohu pracovať s fázami, pričom by nevedeli čo sa pod pojmom fáza prakticky skrýva. Pracovali by s posledným článkom reťazca, pričom by nevedeli z čoho vychádzajú predošlé články, od riadiacich modulov ako sú ventily a motory až po fázu a celkový riadiaci program predstavujúci riadenie procedúry. Vo všetkých častiach riešenia sú zohľadňované zásady dané štandardom ISA S88. Študent, ktorý takýmto spôsobom program vytvorí a oživiť môže plynule a bez prekážok prejsť od svojich fáz k použitiu fáz od systému COMES. Vďaka tejto kontinuite naplní poskytnutý systém COMES BATCH svoj zmysel v plnom rozsahu.

Od veci nie je aj zmienka, že ako som v rešeršiach zistil, mnohé firemné štandardy pre programovanie PLC pre technologické procesy sú postavené na podobnej myšlienke, teda využitia predností štandardu ISA S88 bez toho, aby sa musel kúpiť profesionálny systém BATCH, ktorého cena je stále relatívne vysoká. Medzi výhody, ktoré sa však oproti systému BATCH stratia patrí odľahčenie práce procesného inžiniera pri vytváraní detailného funkčného popisu a tiež transparentnosť daná grafickou formou programovania systému BATCH.

Overenie funkčnosti procesu riadenia prebiehalo v zhotovenej simulovanej demo verzii, teda prostredníctvom hotovej vizualizácie v HMI WinCC flexible. Pre účely možnosti vytvoriť ďalšie funkčnú hotovú nádrž bola táto demo verzia testovaná troma nádržami, tak ako je možné vidieť na obr. 4.11, aj keď v zadaní laboratórnej úlohe, pre ktorú je táto práca zhotovená sú len dve nádrže.

Literatúra

- [1] [ANSI/ISA-88.01, 1995] ANSI/ISA-88.1, 1995, Batch Control, Part 1: Models and Terminology, ISA The Instrumentation, Systems and Automation Society, USA 1995, ISBN 155617-562-0. Dostupné z: <http://www.gmpu.com/GAMP/ISA-88.pdf>
- [2] PÁSEK J., BRAUN V.: Automatizace procesů II, elektronické učební texty
- [3] PÁSEK J., ŠTOHL R.: Automatizace procesů – Laboratorní cvičení, elektronické učební texty

Zoznam príloh

Príloha 1. CD obsahujúce zdrojové kódy programu a vizualizácie