

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

REALIZACE MODULU ÚDRŽBY PRO PROGRAMOVACÍ STANDARD
PLC / HMI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN TICHÝ

BRNO 2015



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

REALIZACE MODULU ÚDRŽBY PRO PROGRAMOVACÍ STANDARD PLC / HMI

IMPLEMENTATION OF MAINTENANCE MODULE FOR PROGRAMMING STANDARD PLC / HMI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN TICHÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN PÁSEK, CSc.

BRNO 2015



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: Jan Tichý

ID: 158576

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Realizace modulu údržby pro programovací standard PLC / HMI

POKYNY PRO VYPRACOVÁNÍ:

Jedná se o funkci, která by kontrolovala počet pohybů ventilů a motohodiny motorů oproti zadané hodnotě. Při jejím překročení by se vyvolal alarm (nebo událost), která by informovala údržbu. Po provedené údržbě by se čítač vynuloval. Funkce modulu bude zahrnovat i informativní a alarmní informace pro údržbu. Realizace pro automaty Simatic S7-300/400. Vizualizační část by měla být pro InTouch, případně i pro WinCC flexible.

Cíle BP:

1. Popište PLC / HMI standard, pro který se má nový modul vytvořit a služby, které má modul poskytovat.
2. Navrhněte koncepci modulu údržby, generování potřebných dat a jejich přenos do stávajícího standardu.
3. Vytvořte programový kód modulu údržby pro PLC Simatic.
4. Navrhněte a realizujte aplikaci HMI pro ovládání a vizualizaci modulu.

DOPORUČENÁ LITERATURA:

Firemní dokumentace společnosti SPX s.r.o.

Termín zadání: 9.2.2015

Termín odevzdání: 25.5.2015

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

Bakalářská práce se zabývá rozбором standardu PLC/HMI firmy SPX s.r.o., do kterého se má navrhnout a implementovat modul údržby. Cílem práce je zhodnotit možnosti návrhu modulu údržby, aby se stotožňoval s požadavky firmy. V první kapitole práce jsou popsány základní informace o řídicím a informačním systému podniku. V kapitole druhé jsou popsány systémy údržby. Kapitola třetí se věnuje rozboru firemního standardu pro PLC/HMI. Předposlední se zabývá již návrhem a realizací samotného modulu údržby. Finální kapitola se věnuje popisu implementaci modulu do reálného technologického procesu.

KLÍČOVÁ SLOVA

Údržba, S88, HMI, Step7, modul údržby, WinCC Flexible 2008, InTouch

ABSTRACT

This bachelor thesis deals with the analysis of the company's standard of SPX s. r. o., where I am design and implement maintenance module. The aim of this thesis is evaluate the possibility of module design and maintenance, to make the best maintenance efficiency. The first chapter deals with the theory of management and information systems. The second chapter consists of maintenance systems. The third chapter deal with the analysis of corporate standard for PLC/HMI. Penultimate chapter is about the design and implementation of the module maintenance. The final chapter describes the module implementation into real technological process.

KEYWORDS

Maintenance, S88, HMI, SIMATIC Step7, maintenance module, WinCC Flexible 2008, InTouch

TICHÝ, Jan *Realizace modulu údržby pro programovací standard PLC / HMI: bakalářská práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2015. 64 s. Vedoucí práce byl Ing. Jan Pásek, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Realizace modulu údržby pro programovací standard PLC / HMI“ 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 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 autorského 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.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Janu Páskovi, CSc. a panu Ing. Filipu Ježkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych chtěl poděkovat celé své rodině, která mě podporovala po celé délce studia.

Brno

.....

(podpis autora)

OBSAH

Úvod	7
1 Automatizace	8
1.1 Systémy pro řízení podniku (ERP)	9
1.2 Systém řízení výroby (MES)	9
1.2.1 Řízení údržby	10
1.3 Dispečerské řízení a sběr dat (SCADA) a programovatelné logické automaty (PLC)	10
2 Údržba zařízení v průmyslové automatizaci	11
2.1 Náklady na údržbu v automatizaci	12
2.2 Vliv údržby na spolehlivost	13
3 Standard firmy SPX pro PLC	15
3.1 Výhody standardu	16
3.1.1 Procesní buňka (PC)	16
3.1.2 Modul zařízení (EM)	16
3.1.3 Ovládací modul (CM)	17
4 Tvorba modulu údržby pro PLC S7-300/400	19
4.1 PLC automaty S7-300/400	19
4.2 Tvorba modulu	19
4.3 Popis programu v PLC	20
4.3.1 Struktura programu v PLC	21
4.3.2 Organizační bloky OB1 a OB10	24
4.3.3 Datové bloky pro údržbu	24
4.3.4 Funkce volání (FC3 - Call MT)	26
4.3.5 Funkce vyhodnocování (FC13 - CM DD MT)	28
4.3.6 Flexibilita a modularita	30
4.3.7 Testování a simulace programu	31
4.4 Vizualizace - Wonderware InTouch	32
4.4.1 Softwarové vybavení prostředí Wonderware InTouch	33
4.4.2 Nastavení komunikace s PLC	33
4.4.3 Popis okna na tvorbu tagů	35
4.4.4 Popis okna pro animaci prvků	35
4.4.5 Popis scriptovacího jazyka	37
4.4.6 Popup okno pro údržbu	37

4.4.7	Popup okno tabulky pro údržbu	42
4.4.8	Footer	45
4.4.9	Simulace a ladění	47
4.5	Vizualizace - WinCC Flexible 2008	48
4.5.1	Nastavení komunikace s PLC	48
4.5.2	Popup okno pro údržbu	48
4.5.3	Popup okno tabulky pro údržbu	50
4.5.4	Simulace a ladění	52
4.6	Porovnání prostředí pro vizualizaci	53
5	Manuál pro implementaci modulu do reálného projektu	54
6	Závěr	55
	Literatura	56
	Seznam symbolů, veličin a zkratk	57
	Seznam příloh	58
A	Přílohy	I
A.1	Příložené CD	I

SEZNAM OBRÁZKŮ

1.1	Pyramida řídicího a informačního systému podniku [3]	8
2.1	Systém preventivních oprav	12
2.2	Systém diagnostické údržby	13
2.3	Systém totálně produktivní údržby (TPM)	14
3.1	Diagram fyzického modelu	15
3.2	Procesní model a jeho rozdělení	17
3.3	Datový blok pro ventily - DB10	18
4.1	Fotografie testovaného PLC	20
4.2	Struktura programu v PLC	21
4.3	Struktura ukazatele typu ANY[4]	22
4.4	Algoritmus volání FC3	27
4.5	Vyhodnocovací algoritmus FC13	29
4.6	Výpis z funkce FC13 pro ukládání dat	30
4.7	Okno v S7 - PLCSIM	31
4.8	Vývojové prostředí InTouch	32
4.9	Nastavení komunikace v System Manageru	34
4.10	Komunikační síť PLC	34
4.11	Okno pro správu tagů	36
4.12	Okno pro animaci prvků	36
4.13	Popup MT okno s Popup DD na pozadí	38
4.14	Popup okno pro potvrzení údržby	39
4.15	Příklad scriptu pro rekalkulaci adres pro okno Popup MT	40
4.16	Algoritmus výpisu okna Popup MT Table	43
4.17	Formát proměnné pro tabulku Popup MT Table	44
4.18	Popup MT Table	44
4.19	Footer	45
4.20	Zobrazení výpisu logovaných proměnných	47
4.21	Prostředí WinCC Flexible 2008	48
4.22	Popup MT pro WinCC	49
4.23	Popup DD pro WinCC	50
4.24	Algoritmus pro Popup MT Table WinCC	51
4.25	Příklad návrhu tagů pro 10. řádek, Popup MTTable	52
4.26	Okno Popup MTTable	52

SEZNAM TABULEK

4.1	Kódy paměťové oblasti [4]	22
4.2	Funkce pro tvorbu pointeru ANY	23
4.3	Struktura UDT pro datové bloky údržby	25
4.4	Tabulka vstupů a výstupů funkce FC13	28

ÚVOD

Otázka údržby v průmyslové automatizaci není žádnou novinkou. Touto otázkou se zabývají firmy, aby navrhly vhodnou strategii pro údržbu automatizačních linek. Cílem této bakalářské práce je seznámit se standardem PLC/HMI firmy SPX s. r. o., pochopit jeho strukturu a důkladně si zjistit veškeré informace o údržbě.

Znalosti o údržbě jsou aplikovány jak teoretické z knižních publikací, tak praktické od pracovníka z firmy SPX s. r. o.

Po získání těchto znalostí bylo zapotřebí si důkladně nastudovat program celého standardu v programu SIMATIC Step 7 v5.5. Bylo potřeba pochopit jejich systém programování tak, aby se firemní pracovníci v modulu údržby dobře orientovali a práce s ním byla intuitivní.

V první kapitole je stručně popsáno, co je to automatizace. Základní prvky řídicího a informačního systému podniku a zařazení řízení údržby. Také jsou tu popsány základní informace o PLC a SCADA.

Další kapitola se věnuje údržbě, její důležitosti a druhům systémů údržby. Vybrané systémy údržby jsou dále ještě rozebrány z hlediska jejich nákladnosti na realizaci a nastavování. Na závěr kapitoly je zmíněna spolehlivost, která úzce souvisí s údržbou.

Kapitola třetí představuje velice stručně standard firmy SPX s. r. o. Je zde popsána struktura standardu a modulární systém. Dále jsou zde zmíněny ventily a motory, z kterých navržený modul sbírá data a vyhodnocuje je.

Čtvrtá kapitola obsahuje tvorbu návrhu a realizaci modulu údržby. Jaké plní funkce a služby systému. Také je zde popsána nepřímá adresace v systému Step 7, která byla implementována do modulu údržby. Další podkapitoly se věnují vizualizaci modulu údržby v prostředí Wonderware InTouch a WinCC Flexible 2008.

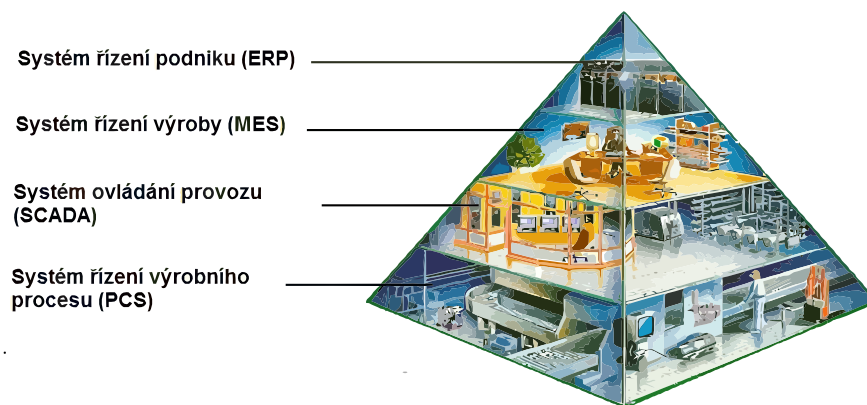
V předposlední kapitole je popsán stručný manuál, co všechno je nezbytné, aby bylo možné modul údržby přesunout a aplikovat v kterémkoliv technologickém přenosu.

Poslední kapitola se zabývá výsledky práce, jakých cílů bylo dosaženo a jaké znalosti tato práce přinesla.

1 AUTOMATIZACE

Automatizace je úzce spjata s automatizací strojů a technologických zařízení či procesů. Je tedy na místě si o tomto spojení něco uvést. Definice automatizace je podle slovníku cizích slov zavádění a používání moderních výrobních procesů, v nichž člověk pouze kontroluje automaty prováděné operace, samočinné uskutečňování mechanických pohybů a přechod od vědomé činnosti k samovolné. Automatizace je v dnešní době informačních technologií neodmyslitelnou součástí ovládní strojů, technologických procesů a zařízení ve strojírenské výrobě, potravinářství nebo jiné sféře průmyslu a výroby. Cílem automatizace je usnadnit život lidem, ulehčit člověku od fyzického a psychického zatížení a v konečném důsledku vede i ke zrychlení a zefektivnění výroby[3].

Samotná automatizace a řízení strojů a technologických procesů byla zpočátku řešena reléovými systémy a komplexní vizualizace, které dnes zobrazují a zpřístupňují data, byly řešeny obyčejnými signálkami a měřicími přístroji. Později nastala éra mikropočítačů, které oproti reléovým systémům nabízely možnosti paměti, a tedy zachování informace, které vedlo k vytvoření komplexnějších řídicích komponentů automatizace - programovatelných automatů, které postupně reléové systémy nahrazovaly. Náhradou pro "vizualizaci signálkami" a měřicími přístroji přinesl systém SCADA, který umožňuje operátorovi sledovat celý technologický proces, popř. ho přímo řídit. Tato evoluce nezasáhla jen přímo řídicí systémy strojů a technologických procesů, ale přetvořila celou strukturu výrobního podniku[3], jak zobrazuje obrázek (Obr. 1.1).



Obr. 1.1: Pyramida řídicího a informačního systému podniku [3]

1.1 Systémy pro řízení podniku (ERP)

Systémy pro řízení podniku (angl. *Enterprise resource planning* - ERP) zastupuje kancelářskou část dění ve výrobním podniku. Pracuje s objednávkami, specifikacemi surovin a výroby, požadavky na materiál a jinými záležitostmi obchodního charakteru[3].

1.2 Systém řízení výroby (MES)

Systémy řízení výroby (angl. *Manufacturing Execution System* - MES) jsou systémy vyvinuté pro operativní plánování a řízení výroby, jejichž účelem je operativně poskytovat informace pro okamžité řízení a optimalizaci výrobních procesů.

Jedná se tedy o přímý integrovaný počítačový systém, který akumuluje metody a nástroje potřebné ke zdokonalení výroby.

Na rozdíl od klasických informačních systémů pracují s aktuálními daty v reálném čase, což jim umožňuje pružně reagovat jak na nestandardní stavy ve výrobě, tak i na okamžité požadavky obchodu a přizpůsobovat výrobní proces, aby byl co nejefektivnější. Systém MES pokrývá široké spektrum vlastností, které systém může zahrnovat. Tyto informační systémy mohou obsahovat řadu funkcí, které se můžou navzájem překrývat, a naopak některé funkce se nemusí využívat.

Nevýhodou kompletních systémů MES je jejich cena, která je velmi vysoká. Proto se z celkových systémů MES vybírají jen jednotlivé funkce, kterých je nejvíce potřeba (např. systém údržby).

Funkčnosti oblasti MES:

- Krátkodobé plánování – operativní řízení výroby.
- Přidělování zdrojů a kapacit.
- Dispečerské řízení výrobních jednotek, klíčové výrobní ukazatele.
- Správa dokumentace.
- Sledování pohybu a genealogie produktu.
- Analýza výkonnosti.
- Správa lidských zdrojů.
- Ovládání výrobního procesu.
- Sběr a archivace dat.
- Řízení údržby[6].

1.2.1 Řízení údržby

Funkční oblast plánování a řízení údržby sleduje a řídí aktivity vykonávané s cílem udržovat výrobní prostředky v takovém technickém stavu, aby se předešlo neplánovaným přerušením výroby. Poskytuje rozvrhy periodické i preventivní údržby a umožňuje řídit údržbu podle skutečného stavu zařízení[3].

1.3 Dispečerské řízení a sběr dat (SCADA) a programovatelné logické automaty (PLC)

Dispečerské řízení a sběr dat (angl. *Supervisory Control And Data Acquisition - SCADA*) je systém shromažďující data v reálném čase, který v kombinaci s programovatelným automatem (angl. *Programmable Logic Controller - PLC*) tvoří nejnižší úroveň řízení výrobního podniku a umožňuje operátorovi sledovat technologický provoz (vizualizace) a zároveň do technologického provozu zasahovat prostřednictvím povelů pro řídicí systém (ovládání). SCADA systém obsahuje vstupně-výstupní hardware, regulátory, HMI, sítě, komunikace, databáze a software.

Systémy SCADA poskytují tyto informace:

- Provozně-technické.
- Poruchové.
- Statistické.
- Logistické.
- Bilanční[3].

HMI

Human Machine Interface, dříve MMI (angl. *Man Machine Interface*). Software (typicky s grafickým uživatelským prostředím – GUI), zobrazující operátorovi informace o stavu procesu a umožňující zadávat operátorské povely (příkazy)[2].

2 ÚDRŽBA ZAŘÍZENÍ V PRŮMYSLOVÉ AUTOMATIZACI

Základním cílem údržby je zabezpečit při optimálních nákladech co nejlepší technický stav zařízení a jeho spolehlivost po čas životního cyklu. U zařízení, která plní mechanickou činnost, jako jsou např. ventily, motory atd., je údržba více potřebnější než na ostatních zařízeních. Mechanické zařízení, u kterých se provádí průběžná údržba, mohou poskytovat vyšší výkon, tím pozitivně zvýšit objem výroby a snížit náklady na údržbu. Také je zapotřebí zohlednit, zda můžeme celý závod odstavit z požadavku na údržbu jediného zařízení. V určitých výrobních procesech, kdy přerušeni výrobního procesu může znamenat až statisícové ztráty, se tyto požadavky na údržbu musí zohlednit[7].

Údržbu můžeme řadit do několika kategorií dle toho, podle jakého požadavku byla vykonávána :

- **Údržba po poruše**
 - používání zařízení probíhá až do jeho zničení či havárie. potéto situaci proběhne údržba.
- **Systém preventivních oprav (PPO)**
 - jedná se o preventivní údržbu v pevně daném časovém horizontu (PPO). V praxi dochází k plánování týdenních, měsíčních a ročních kontrol. Kontroly jsou založeny pouze na daném časovém plánování a nikoliv na stavu zařízení.
- **Systém diferencované proporcionální starostlivosti (DIPP)**
 - plánování a stanovení údržbářských úkonů probíhá na určitém základě všeobecně známých intenzifikačních faktorů, jako jsou například: difference, prevence, plánování apod.
- **Systém diagnostické údržby**
 - metodou technické diagnostiky je zjišťování aktuálního stavu zařízení. Údržba probíhá až po překročení limitu tolerance, např. když ventil překročil limit počtu pohybů. Této metody využívá i tento modul údržby, kdy hodnota opotřebování představuje překročení limitu počtu pohybů u ventilu či překročení počtu odpracovaných hodin motorů. Vyhodnocená data jsou dále zasílána do operátorského softwaru, kde se vyhodnocují.
- **Systém prognostické údržby**
 - je jakousi nadstavbou systému diagnostické údržby. Rozdíl je v tom, že hodnoty získané diagnostikou nejsou využívány jen k zjištění aktuálního stavu, ale na základě jejich hodnot jsou dále plánovány aktivity odpovědným technikům za spolehlivost.

- **Systém automatizované údržby**
 - nese často název informačního systému pro řízení údržby a je charakteristický řízením v reálném čase. Obecně bývá rozdělen do více modulů.
- **Systém totálně produktivní údržby (TPM)**
 - nejkompexnější nástroj pro zlepšení stavu zařízení za účelem maximální efektivity a kvality výroby, který bere ohled na výrobní proces.

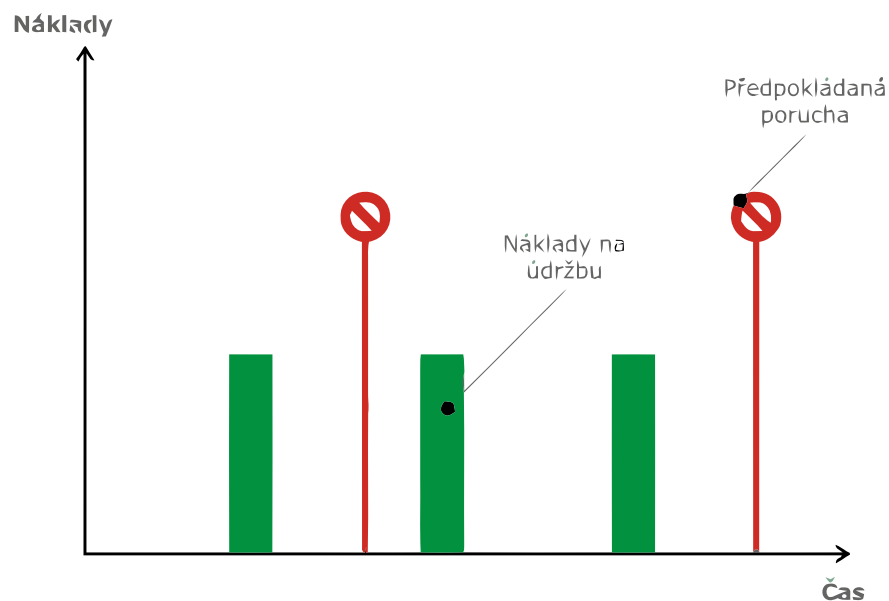
2.1 Náklady na údržbu v automatizaci

Náklady na údržbu se vztahují jak na opravy, výměny zařízení, tak i na realizaci a nastavení systému údržby. Jejich nákladnost ovlivňuje to, v jakém prostředí se zařízení nacházejí a o jak rozsáhlé systémy se jedná.

Dle typu systému údržby můžeme kvalifikovat jejich budoucí náklady.

Systém preventivních oprav (PPO)

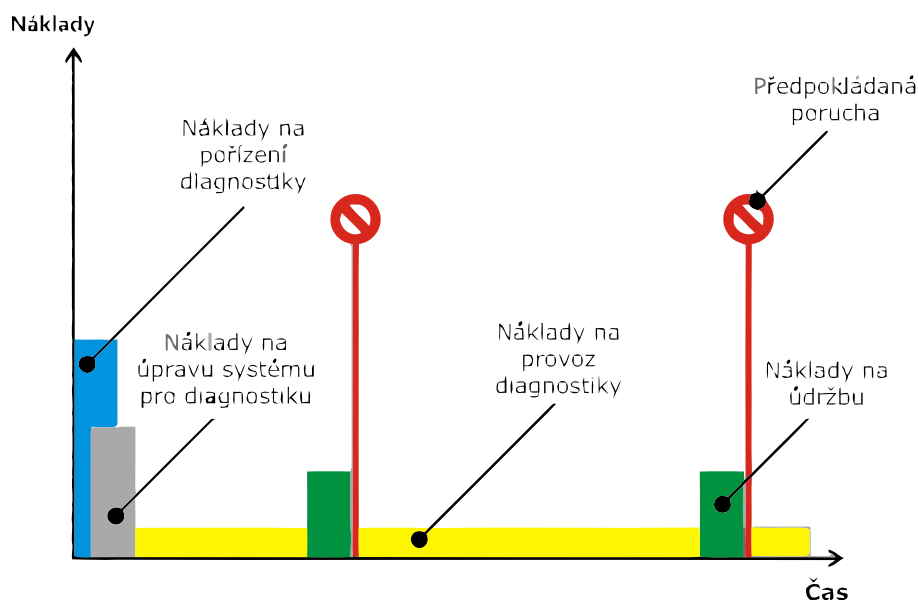
Jak lze vidět na obrázku (Obr. 2.1). Náklady na údržbu jsou pouze při vykonávání údržby za uplynulou dobu.



Obr. 2.1: Systém preventivních oprav

System diagnostické údržby

Diagnostická údržba se provádí na základě činnosti prvků, jak je popsáno výše. Náklady na počátku jsou především tvořeny programy, které hlídají činnosti prvků, a na základě jejich využívání vyhodnocují ideální dobu pro údržbu.



Obr. 2.2: System diagnostické údržby

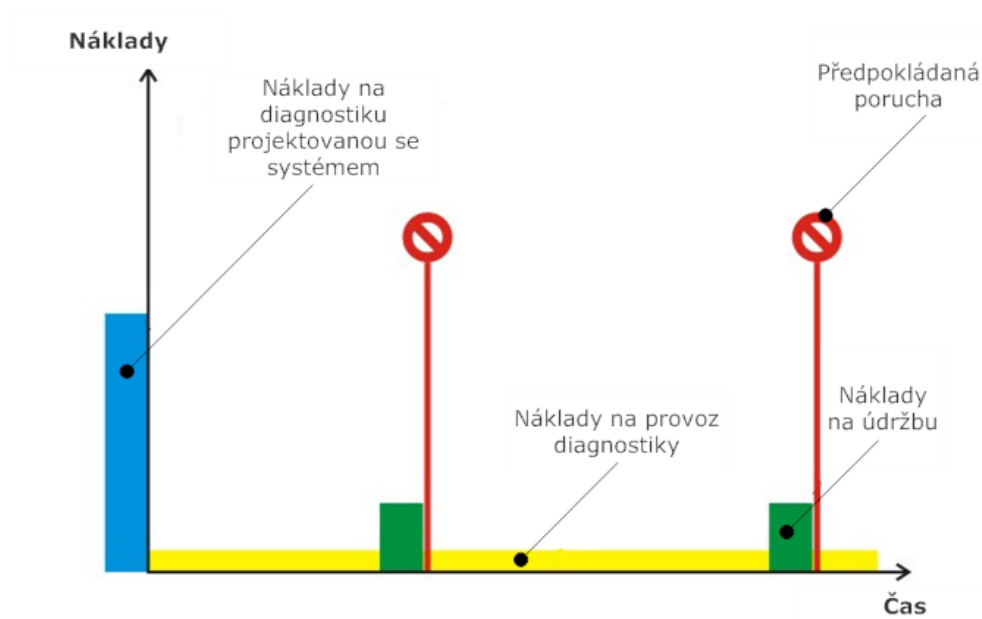
System totálně produktivní údržby (TPM)

Nejdokonalejší varianta pro efektivnost a kvalitu údržby. Vysokou nevýhodou jsou její počáteční náklady a nutnost diagnózy každého členu v procesu.

2.2 Vliv údržby na spolehlivost

Údržba úzce souvisí se spolehlivostí. Ta představuje soubor vlastností technického systému, který vyjadřuje schopnost plnit dané funkce během doby při zachování provozních parametrů zadaných dle technických podmínek. Vlastnosti technického systému ovlivňující spolehlivost jsou:

- bezporuchovost;
- životnost;
- udržovatelnost a opravitelnost;
- pohotovost[5].



Obr. 2.3: Systém totálně produktivní údržby (TPM)

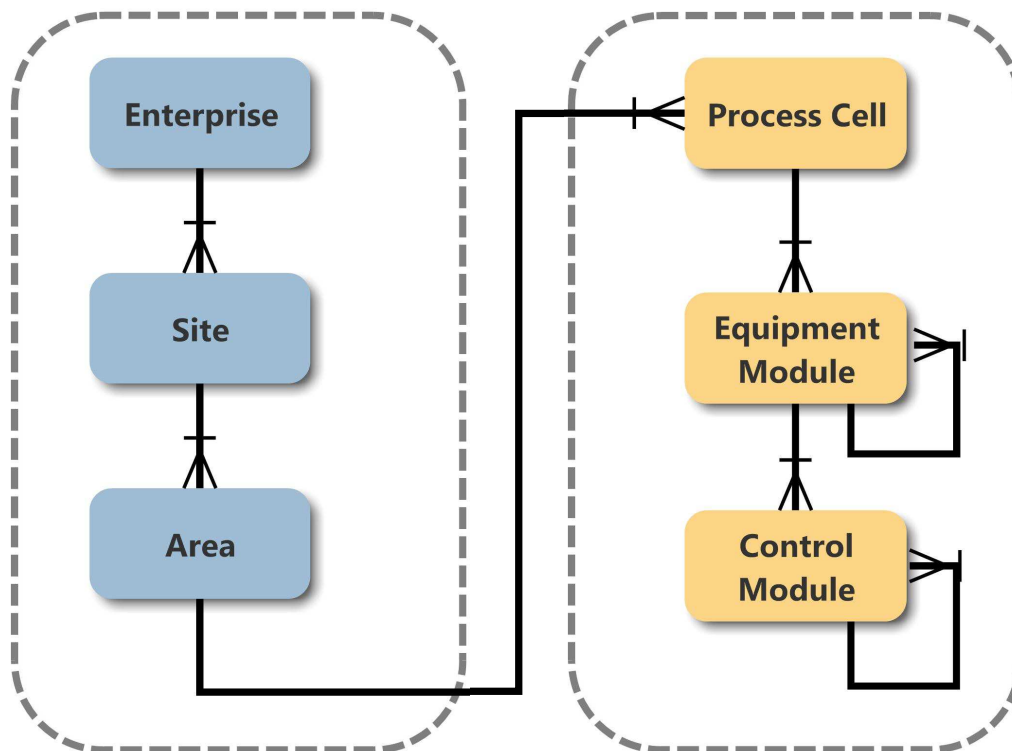
3 STANDARD FIRMY SPX PRO PLC

Tato kapitola se věnuje rozboru firemního standardu, jeho struktury a použití.

Standard firmy SPX s. r. o. vychází ze standardu *ISA - S88*. Jedná se o standard dávkovacího procesu, který neslouží pouze pro software, zařízení nebo procedury. Jedná se o cestu myšlenkového postupu při vytváření celého návrhu procesu. Rozdíly mezi těmito standardy můžeme nalézt v jejich stavbě fyzického modelu, kde fyzický model společnosti SPX s. r. o. nepracuje s jednotkou procedury (angl. *unit*). Celý systém je postavený na používání často se opakujících úloh.

Standard se rozděluje na dvě části:

- fyzický model;
- procesní model.



Obr. 3.1: Diagram fyzického modelu

Na obrázku (Obr. 3.1) lze vidět fyzický model celé struktury standardu. Na straně levé je struktura společnosti a na straně pravé fyzická struktura procesu.

3.1 Výhody standardu

Modularita

System se vytváří tvorbou bloků pro vrstvu ovládacích modulů (CM). Tyto moduly se dále používají v komplexnější vrstvě modulu zařízení (EM) a následně v nejvyšší vrstvě procesní buňky (PC). Důvodem pro tuto stavbu standardu je zvýšit rychlost tvorby aplikací a minimalizovat chyby programátorů.

Eliminace chyb

Méně složité bloky přispívají k menší pravděpodobnosti chybovosti při navrhování systému. Z důvodu možnosti přenášení do jiných projektů a používání i v jiných aplikacích se zajistí ještě menší chybovost ze strany programátorů, kteří po mnoha četných aplikacích mohou jednotlivé moduly odladit.

Efektivní komunikace a nalezení problému

Výhodou přesně daného systému je také komunikace mezi programátory, kteří mohou jasně popsat problém a následně popsát, kde se nachází.

Hierarchie systému se rozděluje na tyto části:

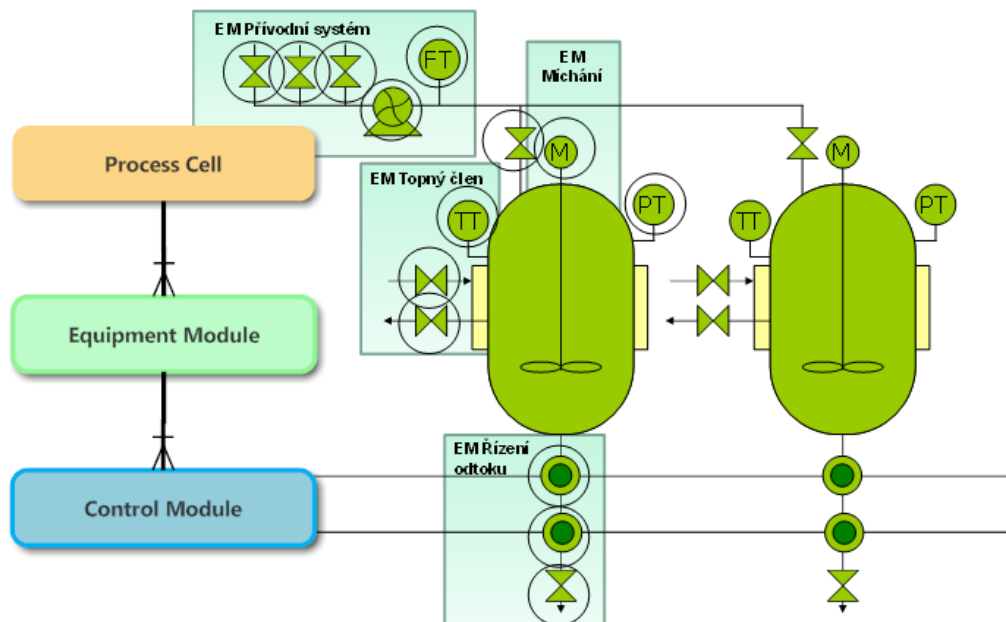
- procesní buňka (angl. *process cell* - PC);
- modul zařízení (angl. *equipment modul* - EM);
- ovládací modul (angl. *control modul* - CM).

3.1.1 Procesní buňka (PC)

Procesní buňka (angl. *Process cell* - PC) je soubor veškerého vybavení (zařízení), které je vyžadováno pro část procesu (úseku, uzlu, linky, apod.). Procesní buňka se skládá ze všech EM. Příkladem PC může být výroba jogurtu. Tento příklad si můžeme představit jako úsek podniku, kde se vyrábí jogurt, ve kterém se nachází výrobní linka z tanků a dopravníků.

3.1.2 Modul zařízení (EM)

Modul zařízení (angl. *Equipment modul* - EM) se skládá z několika CM nebo z dalších EM. Jako příklad může být EM pro zahřívací nebo ochlazovací systémy. Standard firmy SPX s. r. o. na rozdíl od standardu S88 pracuje pouze s EM a CM. Ve standardu S88 můžeme dále nalézt jednotku, která se sdružuje z více EM spolu s CM.



Obr. 3.2: Procesní model a jeho rozdělení

3.1.3 Ovládací modul (CM)

Ovládací modul (angl. *Control modul* - CM) je skupina modulů, které se nacházejí v nejnižší vrstvě celého fyzického diagramu. Plní pouze základní funkce. Do této skupiny patří například: ventily, motory apod. CM jsou ovládány a aktivovány nadřazenou složkou a to EM. Nejsou nijak zastoupeny v PC nebo ve fázích a vyšších složkách.

Ventil a motor

Ventil a motor je jedním z mnoha základních akčních členů, které můžeme ovládat. Na tyto členy je navržen vyvinutý modul údržby.

Veškeré ventily a motory, které se používají v systému, jsou obsaženy v data blocích (DB) v programu Step 7. Pro ventily je zde blok **DB10** a pro motory **DB11**. Na obrázku (Obr. 3.3) je vidět data blok pro ventily. Každý ventil je tvořen **UDT CM DD**, který je totožný jak pro ventily, tak pro motory. Tudíž bude představen jen jeden blok.

Nejdůležitější proměnou je zde **NHT**, která obsahuje informaci o počtu cyklů ventilu nebo počtu odpracovaných hodin motoru. S touto proměnou se dále pracuje v modulu údržby.

KAPITOLA 3. STANDARD FIRMY SPX PRO PLC

3.1. VÝHODY STANDARDU

Index	Variable Name	Data Type	Value 1	Value 2	Description
1.0	Dummy.Status.READY	BOOL	FALSE	FALSE	Ready
1.1	Dummy.Status.OSO	BOOL	FALSE	FALSE	Output State Obtained
1.2	Dummy.Status.PE	BOOL	FALSE	FALSE	Phase Enable
1.3	Dummy.Status.XY	BOOL	FALSE	FALSE	Output to Device
1.4	Dummy.Status.YI	BOOL	FALSE	FALSE	Interlock Active
1.5	Dummy.Status.X1	BOOL	TRUE	TRUE	On / Open
1.6	Dummy.Status.X2	BOOL	FALSE	FALSE	Off / Closed
1.7	Dummy.Status.MON	BOOL	FALSE	FALSE	Monitor of CM enable
2.0	Dummy.CMD.ACK	BOOL	FALSE	FALSE	Alarm Acknowledge
2.1	Dummy.CMD.FM	BOOL	FALSE	FALSE	Valve Normal Position Open / Closed (1 = open)
2.2	Dummy.CMD.H2C	BOOL	FALSE	FALSE	Use Off / Closed Feedback
2.3	Dummy.CMD.H1C	BOOL	FALSE	FALSE	Use On / Open Feedback
2.4	Dummy.CMD.SIM	BOOL	FALSE	FALSE	Simulation mode
2.5	Dummy.CMD.IM	BOOL	FALSE	FALSE	Auto / Manual Bit (1 = manual)
2.6	Dummy.CMD.VM	BOOL	FALSE	FALSE	Valve / Motor Selection (1 = Valve)
2.7	Dummy.CMD.YA	BOOL	FALSE	FALSE	Auto Activate Request
3.0	Dummy.CMD.YM	BOOL	FALSE	FALSE	Manual Activate Request
3.1	Dummy.CMD.PEB	BOOL	FALSE	FALSE	Phase Enable Bypass
3.2	Dummy.CMD.NHT_Mem	BOOL	FALSE	FALSE	For Internal use - Internal Number of Operatio
3.3	Dummy.CMD.NF_Mem	BOOL	FALSE	FALSE	For Internal use - Number Of Faults - Memory B
3.4	Dummy.CMD.YS	BOOL	FALSE	FALSE	Safety interlock from external source
3.5	Dummy.CMD.spare2	BOOL	FALSE	FALSE	
3.6	Dummy.CMD.spare3	BOOL	FALSE	FALSE	
3.7	Dummy.CMD.MON	BOOL	FALSE	FALSE	Off / On monitor of CM
4.0	Dummy.STA	INT	0	0	Status To SCADA
6.0	Dummy.MD	INT	0	0	Mode To SCADA
8.0	Dummy.NF	INT	0	0	Number of Faults
10.0	Dummy.NHT	REAL	0.000000e+000	0.000000e+000	Total Pump Running Hours / Valve Operations

Obr. 3.3: Datový blok pro ventily - DB10

4 TVORBA MODULU ÚDRŽBY PRO PLC S7-300/400

Tato kapitola se věnuje tvorbě modulu údržby, který se nachází v PLC automatu a je vizualizován v prostředí Wonderware InTouch a WinCC Flexible 2008. Čím je modul tvořen a jakým kritériím podléhá. Také je zde uveden stručný popis PLC automatů řady S7-300/400, pro které se modul vyvíjí. Aplikace v PLC využívá převážně nepřímé adresace a i té je věnována část této kapitoly.

4.1 PLC automaty S7-300/400

Modul je navržen pro PLC od výrobce Siemens, řady S7-300/400. Obě platformy PLC, S7-300 i S7-400 jsou modulární platformy používané pro řídicí účely v nejrozmanitějších automatizačních úlohách (např. v automobilovém průmyslu). Oba modely jsou optimálním řešením jak pro centralizovaná, tak pro distribuovaná řešení, podporují komunikaci MPI, PROFIBUS DP, Ethernet (PROFINet) a programují se ve vývojovém prostředí Step 7 v5.5. ve stejných programovacích jazycích.

Hlavní rozdíly jsou ve velikosti, výkonu, rychlosti, paměti, ceně a použití. Kde typ S7-400 je větší, výkonnější, disponuje větší a rychlejší pamětí, má větší možnosti modularity a rozšíření a nabízí například i redundantní konfiguraci (bezvýpadkový provoz). Nevýhodou je jeho cena, která je oproti modelu S7-300 vyšší. Naopak model S7-300 je více kompaktní, má levnější provozní náklady a disponuje výrazně nižší cenou.

4.2 Tvorba modulu

Firma SPX s. r. o. požaduje modul údržby, kterým se budou kontrolovat akční členy v technologickém procesu. Kontrolovat se bude počet jejich odpracovaných hodin nebo počet provedených cyklů, to závisí na druhu kontrolovaného akčního členu. Kontrolovaná hodnota se vztahuje k limitní hodnotě, kdy se při jejím překročení aktivuje alarm. K dispozici jsou dva druhy alarmů, které mají své vlastní set pointy¹ nezávisle na sobě. Pro ventily je to alarm údržby a pro motory je k dispozici jak alarm údržby tak alarm pro výměnu oleje, který má jiný set point než alarm údržby.

Následně vizualizovat získaná data na operátorském panelu a upozornit pověřené osoby v případě spuštění alarmu (doporučované údržbě), aby provedly servis zařízení. Hlášení však nesmí nijak ovlivnit stabilitu celého systému a nesmí bránit

¹Cílová hodnota, kterou má automatický systém řízení dosáhnout.

v přehlednosti od ostatních vážných chyb.



Obr. 4.1: Fotografie testovaného PLC

Kritéria, kterým podléhá modul údržby:

- Nezatěžovat PLC - tak aby nebyly ovlivněny důležitější funkce v PLC.
- Přenosnost - modul musí být možno přenášet do jiných technologických procesů s co nejmenšími úpravami stávajícího kódu.
- Rozšiřitelnost - modul může být bez sebemenších problémů rozšířen.

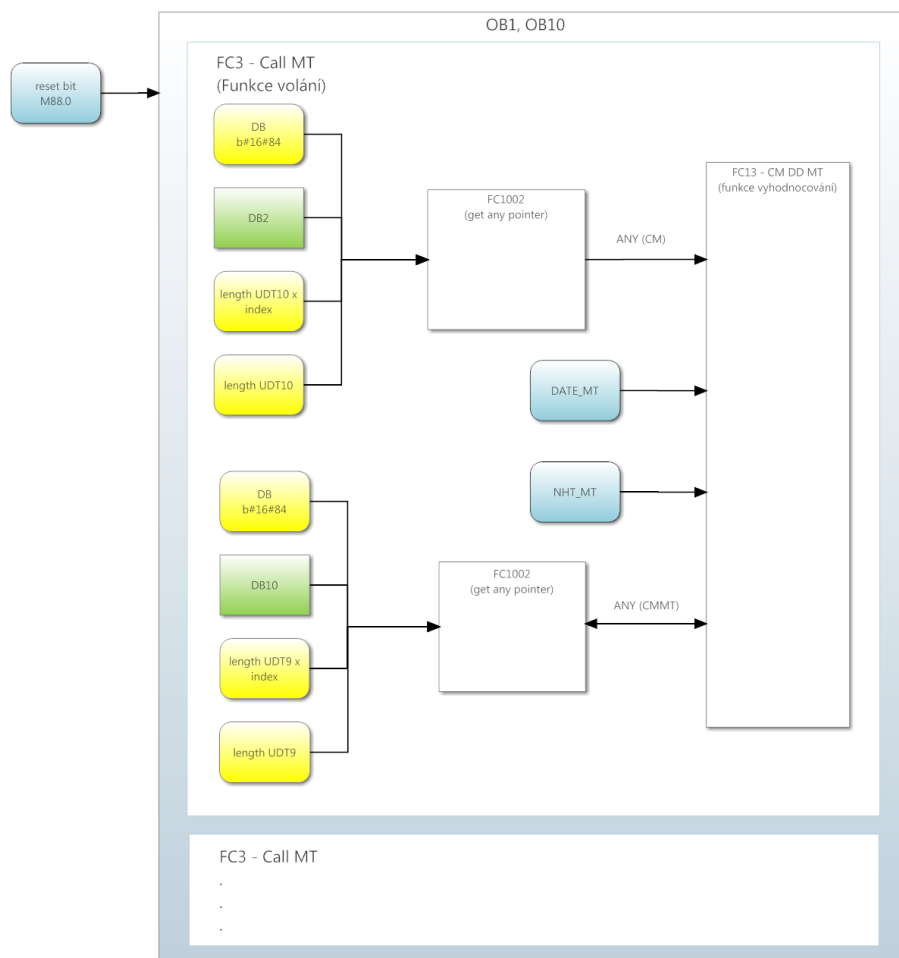
4.3 Popis programu v PLC

Program v PLC je vytvořen ve vývojovém prostředí programu Step 7 v5.5. Tento nástroj slouží k hardwarové konfiguraci PLC, návrhu síťové komunikace, testování a ovládání PLC. Podporuje programovací jazyky: LAD, S7-GGRAPH, FBD a STL.

V aplikaci jsou využívány dva typy těchto jazyků a to FBD (angl. *Function Block Diagram*) a STL (angl. *Statement List Control Language*). Jazyk STL je zvolen z důvodu jeho použití při nepřímé adresaci, viz v kapitole 4.3.1 a jazyk FBD z důvodu přizpůsobení se firemnímu standardu.

4.3.1 Struktura programu v PLC

Navržená struktura programu je koncipována dle hierarchické logiky. Jak lze vidět na obrázku (Obr. 4.2), kde volání funkce pro údržbu (FC3 - Call MT) je umístěno v organizačních blocích OB1 a OB10. Po zavolání funkce **FC3** se následně volají funkce **vyhodnocovacího algoritmu** (FC13 - CM DD MT) pro jednotlivé akční členy. Jak již bylo zmíněno výše, volání funkce pro vyhodnocování je doprovázeno nepřímou adresací za pomoci ukazatele ANY.



Obr. 4.2: Struktura programu v PLC

Přímá a nepřímá adresace paměťové oblasti v PLC

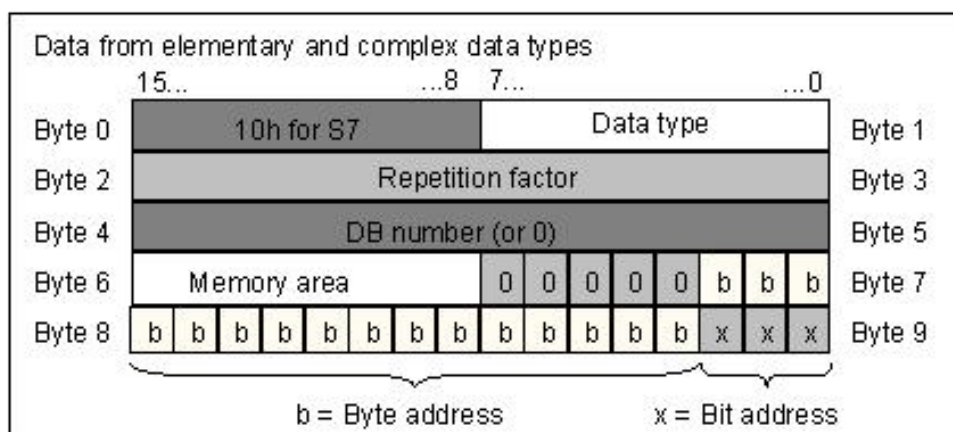
Přímá adresace je v PLC běžnou záležitostí, můžeme si ji představit jako určení přesného místa v paměti např. *Q10.0* nebo použít symbolický název, *výstup*. Tímto způsobem můžeme adresovat VSTUP, VÝSTUP, BIT, BYTE, WORD, DWORD atd.

Nepřímou adresaci umožňuje prostředí Step 7 v5.5 pomocí ukazatelů, které mají přesně danou strukturu. Tuto strukturu lze vytvořit a upravit na hodnoty dle našich potřeb s odkazem na místo v paměti. Tato data je pak možné získat přes systémovou funkci **SFC20: BLKMOV**, která má povinný vstupní parametr ukazatel zdroje a výstupní parametr cílového místa pro kopírování dat.

K dispozici jsou tři typy ukazatelů. Tyto tři ukazatelé se odlišují ve velikosti a možnosti adresace. Prvním, nejmenším ukazatelem, lze ukazovat pouze na byte a bit a tvoří ho 4 byty. Dalším lze poukázat na adresu v oblasti a též na byte a bit, jeho velikost je 6 bytů. Posledním typem je typ ANY, který má největší možnosti, jeho strukturu lze vidět na obrázku (Obr. 4.3). Tímto ukazatelem lze ukazovat jak na oblast, byte a bit, tak také na délku oblasti. Pro účely modulu byl použit právě ukazatel ANY.

Tab. 4.1: Kódy paměťové oblasti [4]

Název	Hexa kód	Příklad
Input area	b#16#81	I4.0, IB4, IW4, ID4
Output area	b#16#82	Q12.0, QB12, QW12, QD12
Bit memory area	b#16#83	M8.0, MB8, MW8, MD8
Data block	b#16#84	DB1.DBX7.0, DB1.DBB7, DB1.DBW7, DB1.DBD7
Local data (L stack)	b#16#86	L16.0, LB16, LW16, LD16



Obr. 4.3: Struktura ukazatele typu ANY[4]

Tvorba datového typu ANY

Datový typ ANY je dynamická proměnná, která uchovává hodnotu ukazatele na místo v paměti. Tak, aby bylo možné dynamicky přistupovat k datům.

Ukazatel je vytvářen za pomoci funkce **FC_getAnyPointer** (FC1002). Jak lze vidět na obrázku (Obr. 4.3) od nultého do sedmého bitu. Do prvního bytu je vložen datový typ, který pro nás představuje hodnotu *b#16#84*, protože potřebujeme přistupovat k datovým blokům. Dalším podstatným parametrem je číslo datového bloku v 4.-5. bytu, bytová a bitová hodnota námi zvolené adresy.

Vstupní a výstupní parametry funkce pro tvorbu pointeru ANY jsou v tabulce (Tab. 4.2).

Tab. 4.2: Funkce pro tvorbu pointeru ANY

Směr	Název	Typ	Popis
Vstup	ParameterTyp	BYTE	Vstup/Výstup/Merker/DB
Vstup	DB_nr	INT	Číslo DB
Vstup	Byte_nr	DINT	Číslo bytu
Vstup	ByteCnt	INT	Počet bytů
Výstup	AnyPtr	ANY	Výsledný pointer

4.3.2 Organizační bloky OB1 a OB10

Organizační bloky OB1 a OB10 zajišťují volání funkce, která volá jednotlivé akční členy (FC3 - Call MT). Blok OB1 je blok, který probíhá cyklicky a je v něm umístěn hlavní program. Jeho priorita je nejnižší v porovnání s ostatními organizačními bloky. To znamená, že kterýkoliv jiný organizační blok může přerušit hlavní program v místě jeho vykonávání a vykonat svůj vlastní program. Program v OB1 volá funkci údržby jen za podmínky, pokud je nastaven bit pro restartování/inicializaci (M88.0.) hodnot pro údržbu.

Blok OB10 s za podmínky přerušení. V hardwarové konfiguraci PLC je nastaven tak, aby přerušení nastalo pouze jednou od nastaveného času v určitém dni, ve kterém se má opakovat. V našem případě je přerušení nastaveno tak, aby se uskutečňovalo 1x denně a mělo nízkou prioritu. Tím je zajištěno, že všechny operace prováděné v bloku nezatíží program v PLC.

4.3.3 Datové bloky pro údržbu

Datové bloky slouží k uchovávání informací. V našem případě datové bloky uchovávají informace o stavu údržby jednotlivých akčních členů.

Každému typu akčního členu (např. dvoustavové ventily, motory apod.) je přiřazen jeden datový blok údržby. Ve struktuře datového bloku je na adrese 0.0 umístěná proměnná jménem **Alarm** typu BOOL. Jedná se o pomocnou proměnnou, která má výchozí hodnotu FALSE, a při aktivaci alarmu jakéhokoliv členu se nastaví na TRUE. Dalšími položkami v DB jsou již jednotlivé akční členy, které jsou typu UDT9 (CM DD MT). Strukturu UDT9 lze vidět v tabulce (Tab. 4.3).

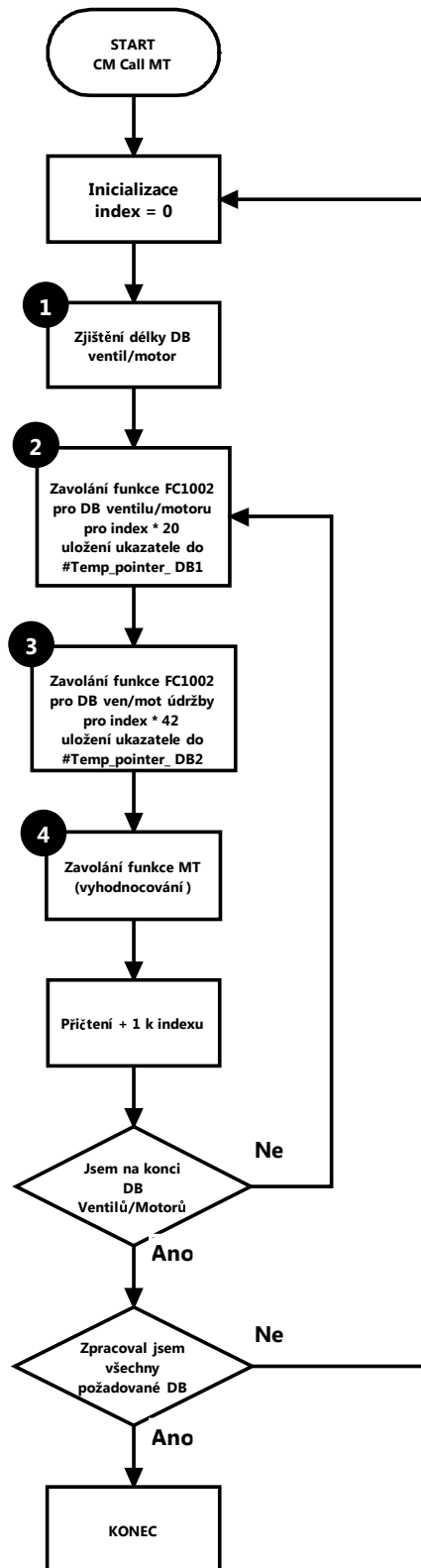
Tab. 4.3: Struktura UDT pro datové bloky údržby

Adresa	Název	Typ	Popis
0.0	NHT_LOG	REAL	NHT při předešlé údržbě
4.0	NHT_LOG_OIL	REAL	NHT při předešlé údržbě oleje
8.0	LIM_MT	REAL	Limit NHT pro nastavení alarmu údržby
12.0	LIM_OIL	REAL	Limit NHT pro nastavení alarmu oleje
16.0	C_MT	INT	Počet provedených servisů
18.0	C_DATE	INT	Počet dnů od poslední údržby
20.0	C_DATE_OIL	INT	Počet dnů od poslední výměny oleje
22.0	C_OIL	INT	Počet provedených výměn oleje
24.0	D_MT	INT	Limit dnů pro nastavení alarmu údržby
26.0	D_OIL	INT	Limit dnů pro nastavení alarmu oleje
28.0	SCADA_Date_MT	STRUCT	Datum, kdy se naposledy provedl servis
34.0	SCADA_Date_OIL	STRUCT	Datum, kdy se naposledy provedla výměna oleje
40.0	AL_CC	BOOL	Alarm údržby pro SCADU
40.1	AL	BOOL	Alarm údržby pro PLC
40.2	SET_OIL	BOOL	Povolení alarmu pro výměnu oleje
40.3	AL_CC_OIL	BOOL	Alarm výměny oleje pro SCADU
40.4	AL_OIL	BOOL	Alarm výměny oleje pro PLC
40.5	V_D	BOOL	Pro motor = FALSE, pro VENTIL = TRUE

4.3.4 Funkce volání (FC3 - Call MT)

Tato funkce zastupuje volání jednotlivých funkcí pro vyhodnocování (FC13 - CM DD MT) s různými vstupními parametry. Funkce je zodpovědná za výpočet vstupních parametrů typu ANY, které se vypočítávají za běhu programu. Jejich hodnota se odvíjí podle toho, na jakém řádku v datovém bloku je vyhodnocování prováděno. Funkce je tvořena pomocí jazyka STL. Tento jazyk se zvolil z důvodu možnosti nepřímé adresace, který tento jazyk umožňuje. Viz kapitola 4.3.1.

Uvnitř funkce je každému typu akčních členů vytvořen jeden vlastní network. Program v networku nejprve zjistí počet členů v systému daného typu, za pomoci volání systémové funkce **SFC24: TEST DB**, která vrací délku datového bloku ①. Z tohoto faktu se vypočítá počet všech členů, za předpokladu, že víme, kolika byty je tvořen jeden člen v DB. Dále se již vypočítává ukazatel ANY pro adresu v datovém bloku členu ② a potom ukazatel pro datový blok údržby ③. S těmito adresami ukazatelů a inicializačními limitními hodnotami pro NHT a dny se volá funkce pro vyhodnocování (FC13 - CM DD MT) ④. Algoritmus pro FC3 je na obrázku (Obr. 4.4).



Obr. 4.4: Algoritmus volání FC3

4.3.5 Funkce vyhodnocování (FC13 - CM DD MT)

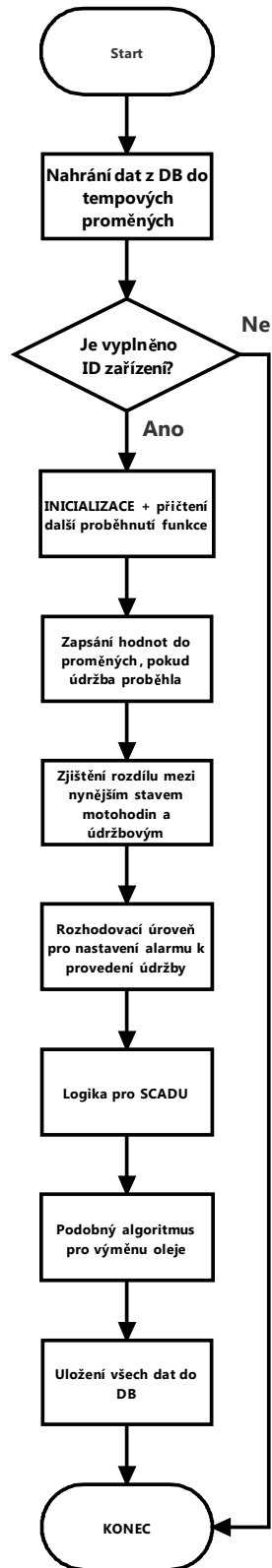
Funkce (FC13 - CM DD MT) je tvořena vyhodnocujícím algoritmem, který nastavuje alarm údržby a výměny oleje. K sepnutí alarmu dochází při překročení limitního stavu NHT (hodnota motohodin) nebo limitního stavu počtu dnů v provozu. U motorů lze také nastavit alarm pro výměnu oleje, který má jiné hodnoty set pointu než alarm údržby. Algoritmus pro vyhodnocování (Obr. 4.5) používá dva systémy údržby. Systém preventivních oprav a systém diagnostické údržby. Systém preventivních oprav je alarm, který nastává v pevně daném časovém horizontu. Například po 200 dnech.

Systém diagnostické údržby je založen na sledování stavu NHT, kdy při překročení hodnoty NHT nastane sepnutí alarmu. Oba tyto systémy jsou na sobě závislé. To znamená, že při vyhlášení alarmu a provedení údržby, se restartují oba dva. Alarm výměny oleje také podporuje stejné dva systémy jako alarm údržby.

Nastavit limitní alarmové hodnoty lze pomocí inicializace, kdy se hodnoty nastaví na hodnoty podle vstupních parametrů nebo lze nastavit hodnoty ručně ve vizualizačním prostředí, viz kapitoly 4.4.6 a 4.5.2. Strukturu vstupních a výstupních parametrů lze vidět v tabulce (Tab. 4.3.5).

Tab. 4.4: Tabulka vstupů a výstupů funkce FC13

Směr	Název	Typ	Popis
Vstup	CM	Any	Pointer control modulu
Vstup	NHT_MT	Real	Limitní hodnota NHT pro nastavení alarmu
Vstup	DATE_MT	Real	Limitní hodnota dnů pro nastavení alarmu
Vstup / Výstup	CM_MT	Any	Pointer control modulu v MT



Obr. 4.5: Vyhodnocovací algoritmus FC13

Princip získávání dat a ukládání dat ve funkci FC13

Jak lze vyčíst z tabulky (Tab. 4.3.5) vstupem jsou dvě proměnné typu ANY, které obsahují ukazatel na místo v paměti. Získáním dat z datového místa a uložení do místní proměnné *Temp_CM* a *Temp_CM_MT* je prováděno pomocí nepřímé adresace a systémového bloku **SFC20: BLKMOV**, jak je zřejmé z výpisu z programu (Obr. 4.6). Tempové proměnné pro data jsou typu UDT CM DD pro *Temp_CM* a UDT CM DD MT pro *Temp_CM_MT*. Tyto proměnné jsou stejného datového typu jako data uložená v datových blocích, na která odkazují hodnoty ukazatele ANY. Ukládání dat je prováděno obdobně, také přes blok **SFC20: BLKMOV**.

```
1 L      P##CM //nacteni bloku s~motorem/ventilem
2 LAR1
3 L      D [AR1,P#0.0]
4 T      LD      0
5 L      D [AR1,P#4.0]
6 T      LD      4
7 L      W [AR1,P#8.0]
8 T      LW      8
9
10 CALL  "BLKMOV"//Nacteni UDT Podle adress do~Temp_CM
11 SRCBLK :=#T_pointerCM
12 RET_VAL:=#T_ret_value
13 DSTBLK :=#Temp_CM
```

Obr. 4.6: Výpis z funkce FC13 pro ukládání dat

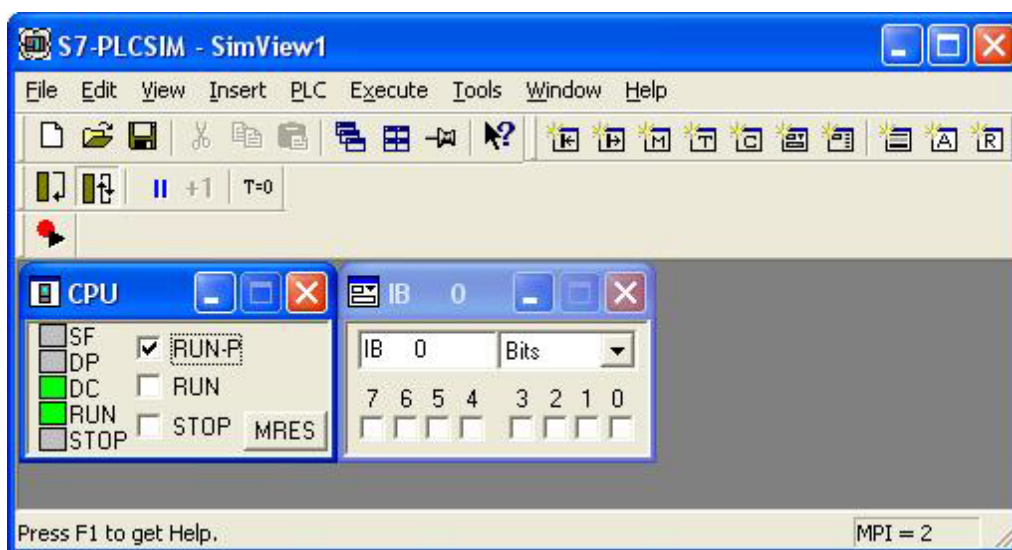
4.3.6 Flexibilita a modularita

Tento modul údržby je připraven k dalšímu rozšíření na ostatní akční členy v systému. Příkladem může být extenze na 3. stavový ventil. Pro rozšíření je nutno zkopírovat datový blok 3. stavového ventilu, vložit ho pod novým číslem DB, které ještě není obsazeno. Vložit na adresu 0.0 v datovém bloku proměnnou BOOL s názvem Alarm. Dále přepsat všechna UDT na UDT CM DD MT, ve funkci volání (FC3) zkopírovat jakýkoliv již stávající network a přepsat číslo datových bloků, pro údržbu, pro akční člen a přepsat velikost UDT v bytech akčního členu. Následuje zkopírování funkce pro vyhodnocování a změny tempové proměnné *Temp_CM* na typ odpovídající v datovém bloku.

4.3.7 Testování a simulace programu

Simulace programu byla prováděna v S7-PLCSIM (Obr. 4.7). Simulační program slouží k testování programu na simulovaném PLC, které je definováno v hardwarové konfiguraci. Program poskytuje jednoduché rozhraní pro monitorování a změnu parametrů. V mém případě jsem používal VAT tabulku (variable table) k monitorování proměnných a jejich konfiguraci.

Po potvrzení funkčnosti programu jsem posléze celý software otestoval na reálném PLC S7-300 s CPU315-2 PN/DP. Připojené na sběrnici ProfiNET.

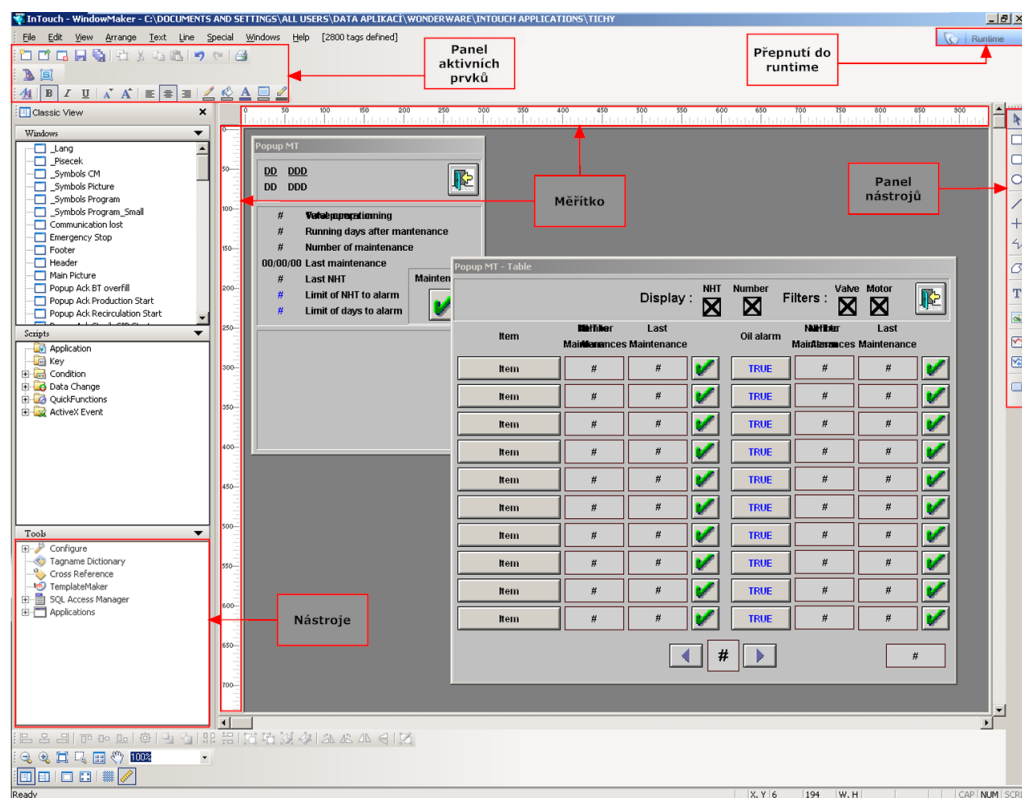


Obr. 4.7: Okno v S7 - PLCSIM

4.4 Vizualizace - Wonderware InTouch

Wonderware InTouch je výkonný systém pro vytváření aplikací pro sběr dat, vizualizaci, sledování a řízení technologických procesů. Umožňuje vytvořit grafické zobrazení výrobních technologií na PC, jejich ovládání a dynamické animace, které v reálném čase zobrazují aktuální stavy systémů[1].

InTouch patří mezi systémy SCADA/HMI (Supervisory Control and Data Acquisition / Human-Machine Interface).



Obr. 4.8: Vývojové prostředí InTouch

4.4.1 Softwarové vybavení prostředí Wonderware InTouch

Prostředí InTouch obsahuje kromě vývojové aplikace také:

- **SMC (System Management Console)** - Tento produkt centralizuje konfiguraci klíčových produktů Wonderware, stejně tak poskytuje diagnostiku a hlášení chyb. V tomto softwaru se konfiguruje DASServer, který zprostředkovává komunikaci mezi PLC a ethernetovou kartou v PC. Dále má prostřední možnost zaznamenávání logů, které se využijí při odladění chyb. Více v kapitole 4.4.9.
- **Application Manager** - Prostředí, které slouží k organizaci projektů, možnosti konfigurace projektu, exportu a importu databází proměnných a spouštění prostředí WindowsMaker a WindowsViewer.
- **WindowsMaker** - Vývojové prostředí pro vizualizaci.
- **WindowsViewer** - Runtime vyhotovené aplikace.

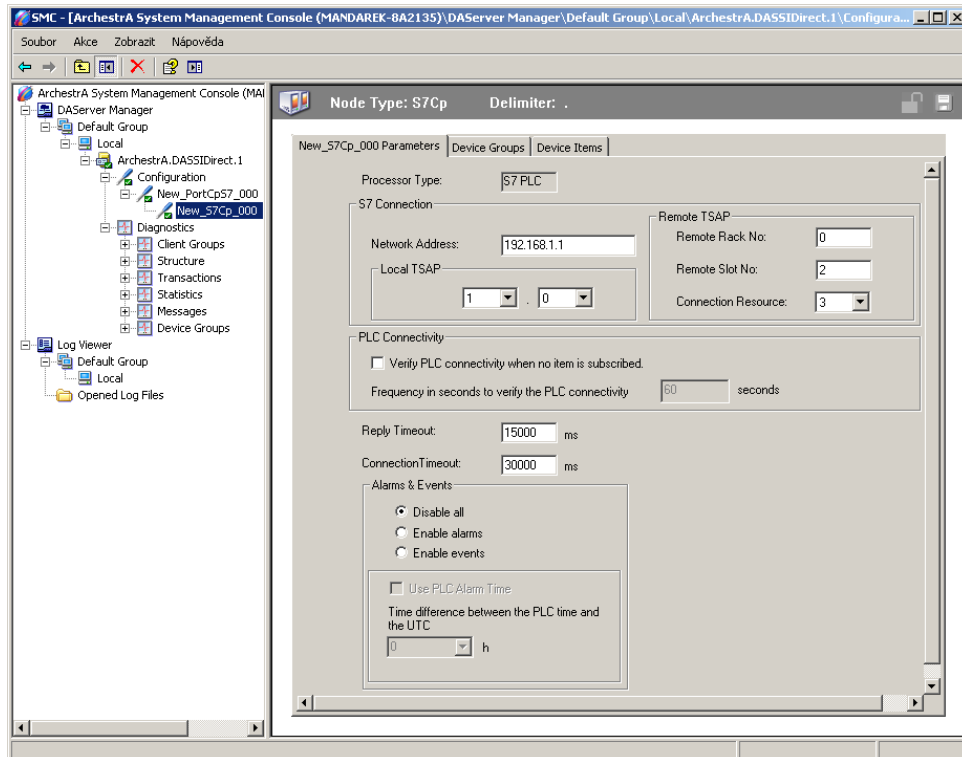
4.4.2 Nastavení komunikace s PLC

Komunikace mezi zařízeními probíhá za pomoci ProfiNETU. V následující ukázce je popsán příklad spojení PLC a prostředí InTouch.

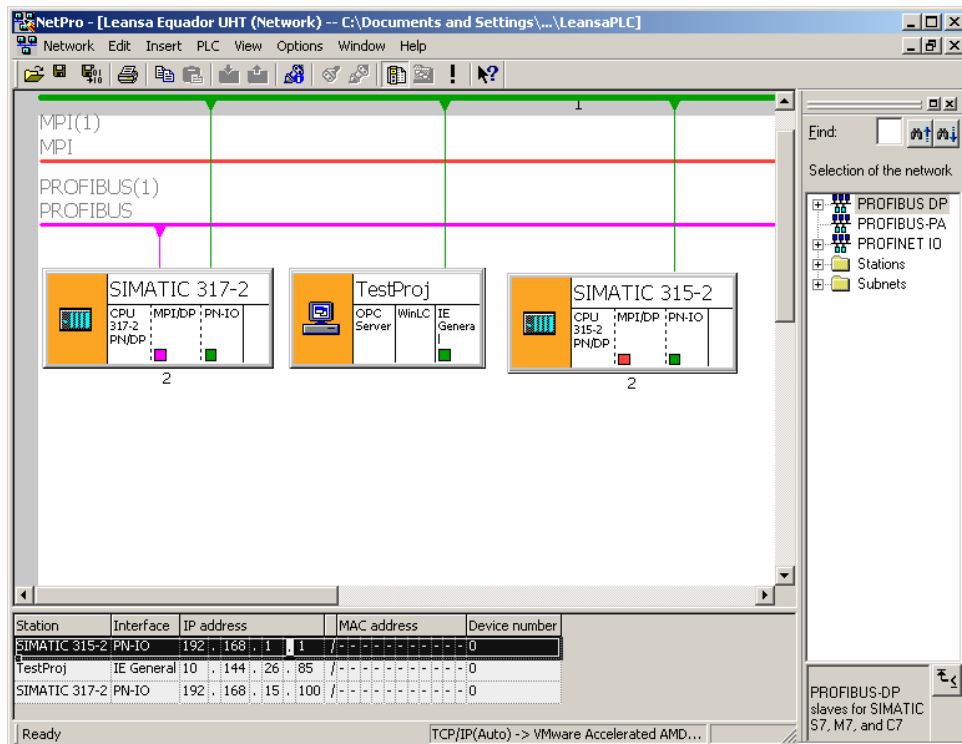
1. Otevřeme prostředí Simatic Step 7 v5.5 a vložíme Insert New Object → Industrial Ethernet a nastavíme IP adresu komunikačního procesoru.
2. V prostředí InTouch - System Management Console vytvoříme New Node Group (Default Group) → New Node (Local) → Achestra.DASSIDirect → Configuration → New Port → konečný bod, kde se nastaví IP adresa PLC automatu. Jak je patrné z obrázku (Obr. 4.9).

KAPITOLA 4. TVORBA MODULU ÚDRŽBY PRO PLC S7-300/400

4.4. VIZUALIZACE - WONDERWARE INTOUCH



Obr. 4.9: Nastavení komunikace v System Manageru



Obr. 4.10: Komunikační síť PLC

4.4.3 Popis okna na tvorbu tagů

V této kapitole bude popsán popis polí, která jsou důležitá pro tvorbu tagů a pochopení následné nepřímé adresace. Tagy se rozdělují dle typu na INT, REAL a BOOL. Dále na I/O, paměťové a nepřímé.

Nejdůležitějšími poli pro realizaci modulu jsou pole *Access Name* a *Item*, která lze nalézt u I/O tagů. Tyto dva parametry odkazují na adresu hodnot v paměti PLC a při jejich změně se dokážeme odkazovat na jiné hodnoty. To znamená, že díky změnám těchto parametrů, je možné dynamicky adresovat kterákoliv data v paměti PLC. Ve scriptovacím jazyce se tyto dva parametry změny jako dotaz na přístup atributu objektu, např. `obj.Reference = Access Name + Item`. Důležitou poznámkou je, že při přepsání reference nezískáme data okamžitě, ale s časovým zpožděním, řádově stovek milisekund. Tento fakt si je nutno uvědomit a kalkulovat s ním při tvorbě modulu.

Také nesmíme zapomínat, jakého datového typu je tag, pro REAL jsou to 4 byty, pro INT 2 byty a pro BOOL 1 bit.

Výjimečným typem tagu v InTouchi jsou *SuperTagy*, jedná se o tag, který sdružuje tagy do jedné skupiny.

Popis okna pro tvorbu tagů (Obr. 4.11):

- 1 Otevření okna *Select Tag* pro výběr z již vytvořených tagů.
- 2 Počáteční hodnota tagu.
- 3 Nastavení datového typu tagu.
- 4 Nastavení tagu pro čtení nebo čtení i zápis.

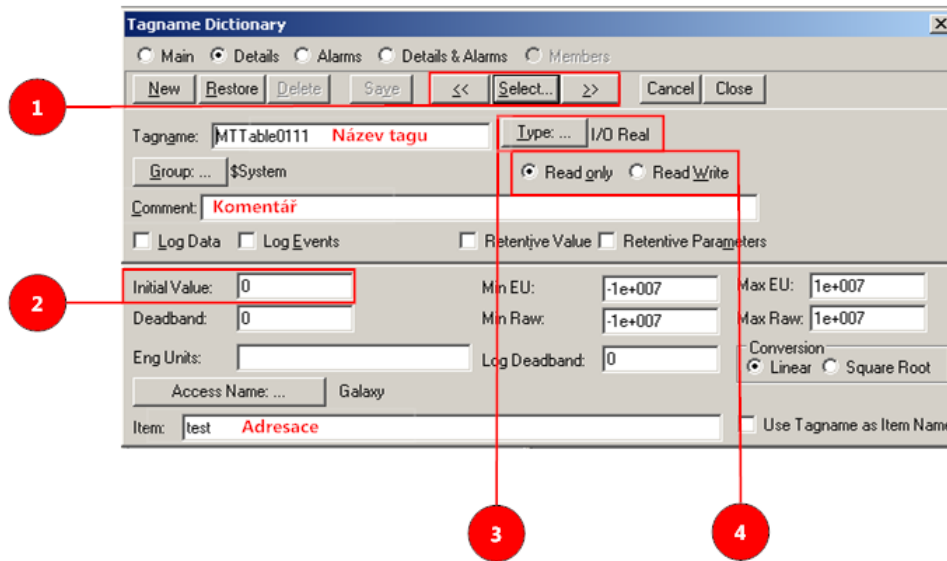
4.4.4 Popis okna pro animaci prvků

Pro animaci každého objektu slouží pole, které lze vidět na obrázku (Obr. 4.12). Pomocí tohoto pole jdou volat scripty, skládat stringy², odčítat či sčítat tagy, skrývat či zobrazovat prvky, měnit jejich hodnoty atd...

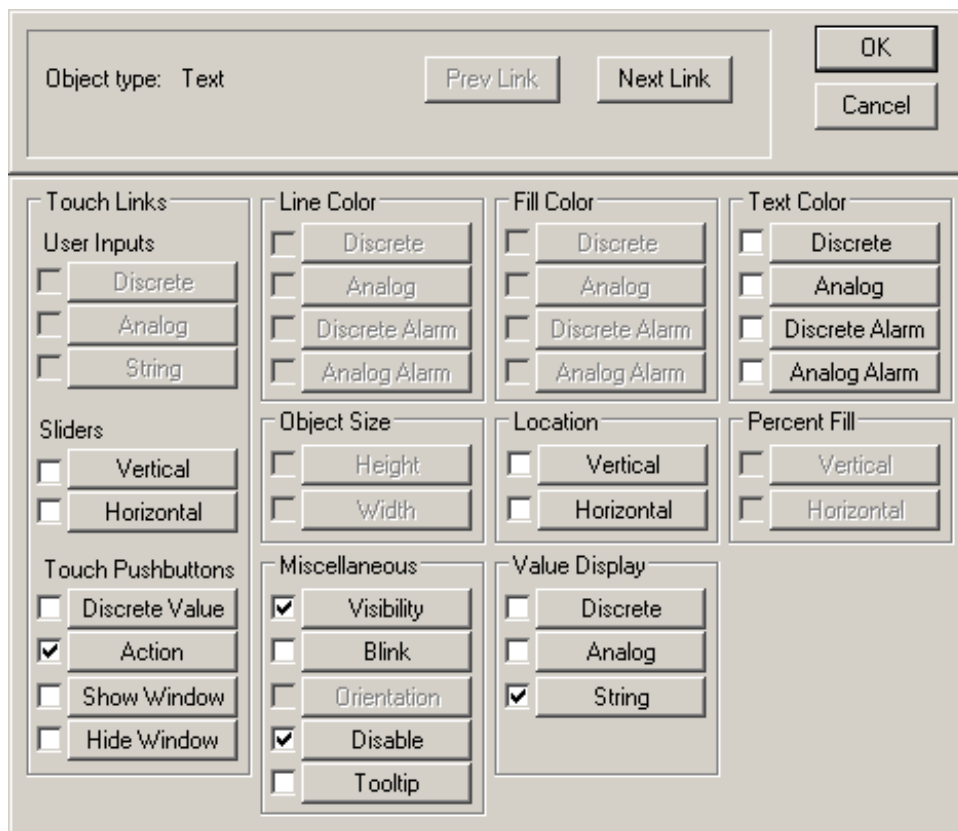
V modulu se nejvíce využívá položka *Action*, která spustí vepsaný kód uvnitř funkce. Dále *Visibility*, které slouží k viditelnosti oken. *Disable* k neumožnění přístupu a úprav hodnot, například pokud operátor nemá dostatečné oprávnění. *Value Display* k zobrazování hodnot.

Také v celém projektu nejsou použita žádná textová pole, nýbrž pole, kterým se mění hodnota stringu. To se provádí z důvodu vícejazyčné podpory modulu, kdy je potřeba měnit řetězec textu.

² Řetězec



Obr. 4.11: Okno pro správu tagů



Obr. 4.12: Okno pro animaci prvků

4.4.5 Popis scriptovacího jazyka


InTouch podporuje velice kvalitní podporu událostně orientovaných scriptů s velkorysou nabídkou vestavěných funkcí. Scriptovací jazyk firmy Invesys, která tento jazyk vyvíjí, je kombinací jazyků Visual Basic a C# s možností využít Active X a .NET.

Scripty se dají rozdělit do těchto kategorií:

- **Application** - Scripty, které se spustí při zapnutí/vypnutí/průběhu vizualizace.
- **Key** - Aktivace scriptů při stisknutí definované klávesy.
- **Condition** - Při splnění podmínky.
- **Data Change** - Při změně dat.
- **QuickFunctions** - Funkce se vstupními a výstupními parametry.

4.4.6 Popup okno pro údržbu

Okno pro údržbu sdružuje informace, které modul údržby poskytuje. Toto okno lze vyvolat po otevření popup okna pro DD členy. To se vyvolá po kliknutí na akční člen ventilu či motoru v hlavním procesním okně.

Pro možnost zobrazení okna údržby je nutno v okně *Popup DD* zvolit záložku *General*, kde je v dolní části umístěno počet provedených údržeb a poslední údržba (Obr. 4.13). Dále je možnost kliknout na ikonu nářadí , po kterém se otevře okno s údržbou. Na tuto operaci je potřeba mít potřebné oprávnění. Pokud ikona s nářadím červeně pulzuje, znamená to, že je potřeba provést údržbu zobrazeného akčního členu.

Po otevření popup okna pro údržbou je možno vidět souhrn všech informací o údržbě. Pokud má operátor dostatečná oprávnění, může i vypnout alarm údržby nebo nastavit limit hodnot, ke kterým se vztahují alarmy. Také je možnost zapnutí či vypnutí alarmu pro výměnu oleje, tato funkce je možná pouze u motorů.

Pokud není potřebné provést údržbu a operátor klikne na ikonu pro provedení údržby, objeví se potvrzovací okno, zda-li chce opravdu údržbu provést. Viz obrázek (Obr. 4.14).

Princip realizace

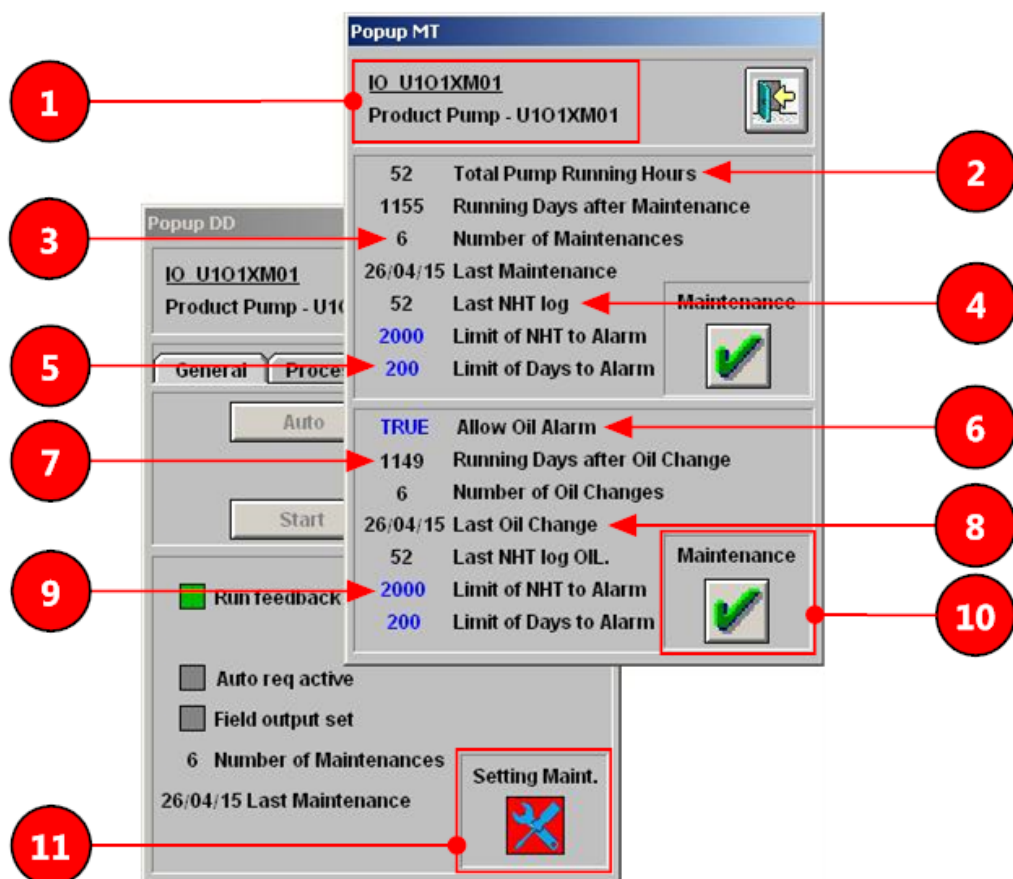
Na počátku je důležité si vhodně rozvrhnout celé okno, jak bude vypadat a které tagy se budou používat. Rozvržení by také mělo obsahovat způsob komunikace jednotlivých tagů. Tagy, do kterých není nutno zapisovat, zvolit pouze pro čtení a tagy, u kterých požadujeme úpravu hodnoty zvolit jako *read/write*. poté se již zaměřit na způsob realizace.

Dynamika tohoto okna je realizována pomocí supertagu *indMT*, kde se jednotlivým tagům mění jejich reference (viz kapitola 4.4.3). Reference tagů se přepisují již ve

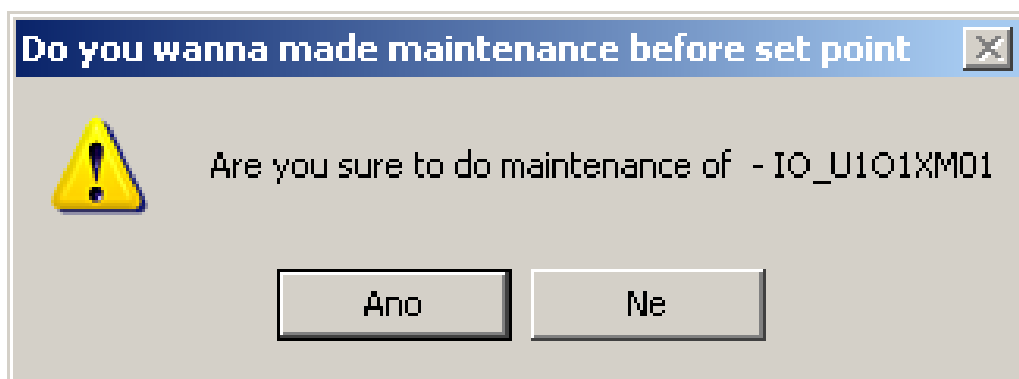
scriptu pro *Popup DD* a to z důvodu zobrazení poslední údržby a počtu provedených servisů.

Princip algoritmu výpočtu lze vidět ve výpisu z kódu na obrázku (Obr. 4.15), kde vstupem do scriptu je jméno členu. Dále script zavře již otevřená popup okna pro údržbu i pro DD členy a jméno členu se nahraje do jména nepřímého tagu. Tím tento tag odkáže na tag akčního členu, který již obsahuje informace v jakém DB se nachází a jaká je jeho adresa. Díky znalosti těchto informací, je možno vytvořit *AcessName* a adresu tak, aby se mohlo odkázat na adresu DB pro MT v PLC. Následující zápisy jsou již na principu spojování indexu a *AcessName* k přepisům referencí jednotlivých tagů. V příkladu (Obr. 4.15) lze vidět všechny tři možnosti datových typů, jak pro *BOOL*, *INT*, tak pro *REAL*.

Poslední řádek kódu slouží k odladování scriptu. Tento kus kódu vypíše hodnotu proměnné do logovacího okna, viz obrázek (Obr. 4.20).



Obr. 4.13: Popup MT okno s Popup DD na pozadí



Obr. 4.14: Popup okno pro potvrzení údržby

```
1 input - indNM AS MESSAGE;
2
3 DIM AccessName AS MESSAGE;
4 DIM index AS INTEGER;
5 DIM pos AS INTEGER;
6 DIM index_db AS INTEGER;
7 {*** Hide windows for DD ***}
8 Hide "Popup DD";
9 Hide "Popup MT";
10
11 {Update the indirect tagnames for DD}
12 indDD\STA.Name = indNM;
13
14 {Build inderect pointer names}
15 pos = StringInString(indDD\STA.Reference, "W", 1, 0);
16 index = StringToIntg(StringMid(indDD\STA.Reference, pos + 2, 0)) ...
    - 4;
17
18 {Maintenance tags}
19 index_db=StringToIntg(StringMid(indDD\STA.Reference,9,2));
20
21 {Recount DB}
22 IF index_db == 10 THEN
23 AccessName = "plc01.DB2";
24 ELSE
25 AccessName = "plc01.DB3";
26 ENDIF;
27 {Calculatign adress}
28 index = ((index/20)*42)+2;
29
30 {Bool built}
31 indMT\AL_CC.Reference = AccessName + ",X" + Text(index + ...
    40.0, "#.#");
32 {Analog}
33 indMT\D_OIL.Reference = AccessName + ",W" + Text(index + 26, "#");
34 {Analog}
35 indMT>Last_NHT.Reference = AccessName + ",REAL" + Text(index, "#");
36 {Troubleshootting}
37 {LogMessage(indMT\SET_OIL.Reference );}
```

Obr. 4.15: Příklad scriptu pro rekalkulaci adres pro okno Popup MT

Popis okna

- 1 Jméno zařízení a jeho popis.
- 2 Počet celkově odpracovaných hodin motoru nebo počtu cyklů ventilu.
- 3 Počet provedených údržeb.
- 4 Poslední hodnota odpracovaných hodin motoru nebo počtu cyklů ventilu při poslední údržbě.
- 5 Limitní hodnota, dnů po kterých se vyhlásí alarm.
- 6 Povolení alarmu pro výměnu oleje.
- 7 Počet uplynulých dnů po poslední údržbě.
- 8 Datum poslední údržby.
- 9 Limitní hodnota odpracovaných hodin motoru nebo počtu cyklů ventilu, po kterých se vyhlásí alarm.
- 10 Tlačítko pro vypnutí alarmu a vykonání údržby. Pokud ještě není potřeba provést údržbu, vyskočí okno na obrázku (Obr. 4.14) pro potvrzení kroku.
- 11 Otevření okna Popup MT.

Popis ikon a jejich význam



Ikona signalizuje, zda-li je potřeba provést údržbu. Při nutnosti údržby ikona pulzuje. Dále vyvolává script, který zaloguje provedení údržby.



Ikona signalizuje, zda-li je potřeba provést údržbu. Při nutnosti údržby ikona pulzuje. Dále vyvolává script, který otvírá okno s oknem údržby, Popup MT.



Tato ikona zavírá popup okno.

4.4.7 Popup okno tabulky pro údržbu

Jedná se o přehlednou tabulku, ve které jsou vypsané všechny ventily a motory v technologickém procesu. Operátor na operátorském panelu má tudíž přehled o všech zařízeních v systému.

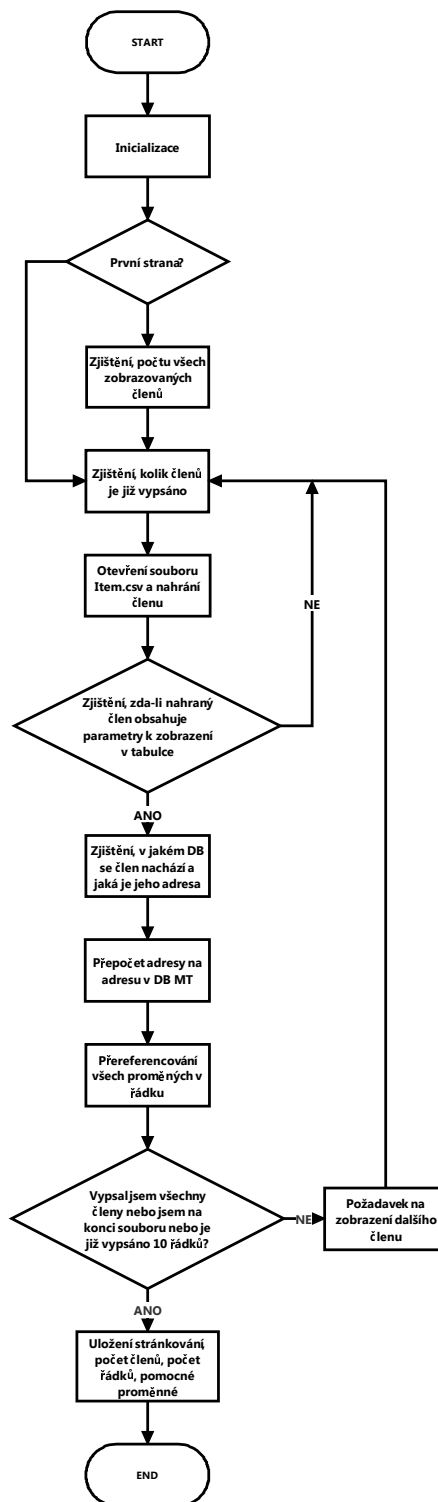
Tabulka je rozdělena do 8 sloupců a stránkování tabulky je navrženo na 10 členů. Tabulka obsahuje možnost filtrace, tudíž je možnost vypisovat pouze ventily, motory nebo vše. Také je možno měnit zobrazované informace mezi počtem provedených servisů a počtem NHT, které chybí do limitní hodnoty alarmu. Při kliknutí na jméno členu se zobrazí jeho popup okno. Vysvětlení všech polí v tabulce lze nalézt dále v kapitole 4.4.7 o popisu okna.

Princip realizace

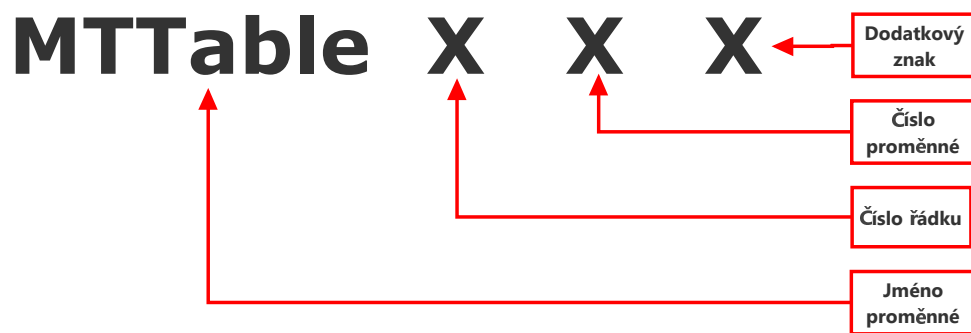
Realizace je provedena za pomoci hlavního scriptu *indMTTable*, který obstarává lokaci zařízení v projektu a nadále jejich výpis do tabulky. Script hledá v generovaném souboru *Item.csv*, který se nachází ve složce projektu. V tomto souboru se nachází všechna zařízení v projektu a script výběru zařízení, která se mají zobrazit. Po výběru zařízení zjistí nejprve adresu zařízení v DB, a poté vypočte adresu pro DB údržby. Následné změny referencí jsou obdobné jako v kapitole 4.4.6. Tyto změny referencí se následně provede pro všechny řádky tabulky. Algoritmus je popsán na obrázku (Obr. 4.16)

Každý řádek tabulky obsahuje 20 tagů ve formátu dle obrázku (Obr. 4.17).

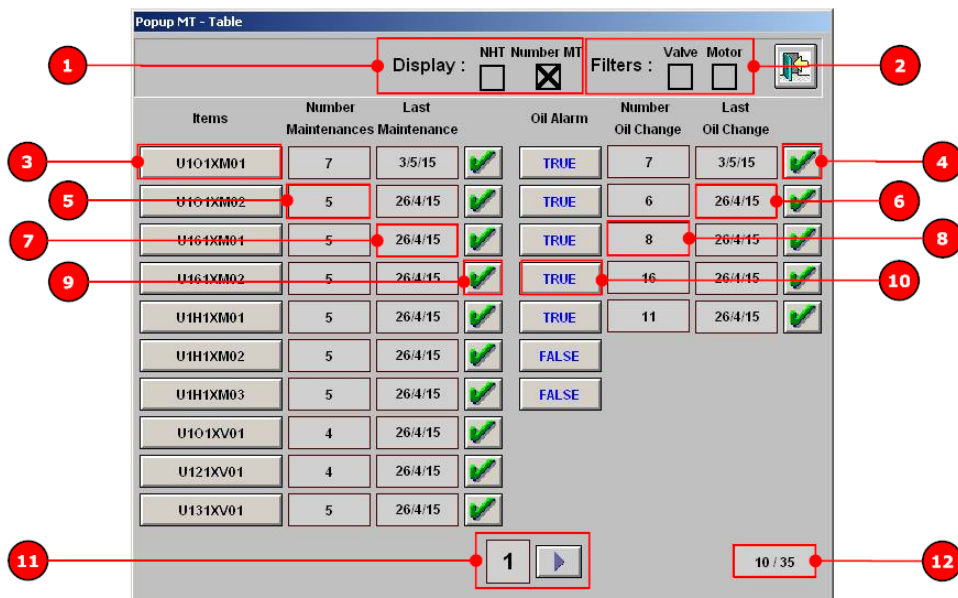
Z důvodu možného opakování každého řádku se každý řádek seskupil do vyššího celku - tzv. Smart Symbolu. Díky jemuž se usnadní tvorba řádku.



Obr. 4.16: Algoritmus výpisu okna Popup MT Table



Obr. 4.17: Formát proměnné pro tabulku Popup MT Table



Obr. 4.18: Popup MT Table

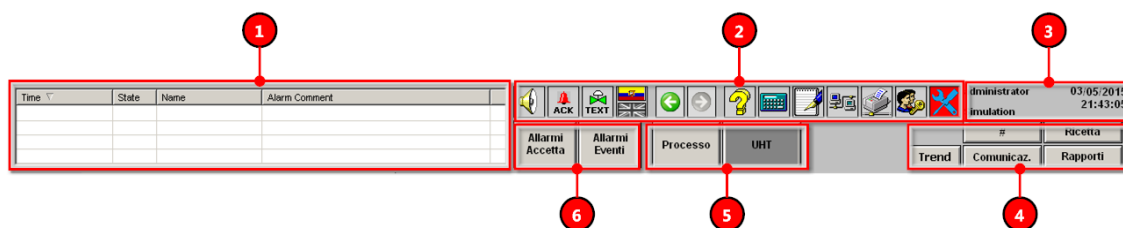
Popis okna (Obr. 4.18)

- 1 Možnosti zobrazení celkového počtu údržeb nebo počtu odpracovaných hodin motoru nebo cyklů ventilu do limitní hodnoty alarmu.
- 2 Filtrování výpisu zařízení.
- 3 Název zařízení s možností kliknutí a zobrazení okna *PopupDD*.
- 4 Ikona pro možnost vypnutí alarmu a provedení logování výměny oleje.
- 5 Počet provedených údržeb, toto pole se mění na základě zvoleného zobrazení dle bodu 1.
- 6 Datum poslední výměny oleje.
- 7 Datum poslední údržby.
- 8 Počet provedených výměn oleje, toto pole se mění na základě zvoleného zobrazení dle bodu 1.
- 9 Ikona pro možnost vypnutí alarmu a provedení logování údržby.
- 10 Možnost povolení alarmu výměny oleje.
- 11 Stránkování okna s šipkou na přesun na další stranu.
- 12 Počet zobrazených akčních členů z celkově možných.

4.4.8 Footer

Jedná se o patičku, která je nepřetržitě zobrazená na operátorském panelu. Obsahuje základní navigační přehledové okno a výpis alarmů, jak lze vidět na obrázku (Obr. 4.19).

V tomto okně byla realizována ikona pro zobrazení okna *Popup MT Table*. Ta signalizuje, pokud v systému existuje nějaký člen, u kterého je nutné provést údržbu.



Obr. 4.19: Footer

Popis okna (Obr. 4.19)

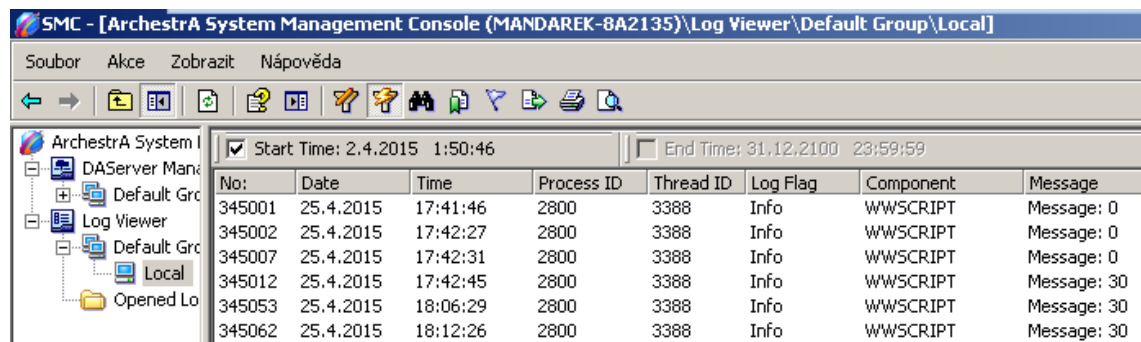
- 1 Okno pro výpis alarmů.
- 2 Panel základních funkcí. Popsáno níže.
- 3 Okno pro zobrazení přihlášeného uživatele, mód systému, datum a čas.
- 4 Panel funkcí.
- 5 Zobrazení procesu, proces nebo UHT (Ultra high temperature).
- 6 Panel pro okna alarmů.

Popis ikon a jejich význam

- | | |
|---|--|
|  | mute/unmute; |
|  | alarmy; |
|  | zobrazení textových polí; |
|  | výběr jazyka; |
|  | zpět/vpřed; |
|  | nápověda; |
|  | kalkulačka; |
|  | zobrazení poznámkového bloku; |
|  | status komunikace; |
|  | tisk okna operátorského panelu; |
|  | okno s přihlášením uživatelů; |
|  | ikona signalizuje, zda-li je potřeba provést údržbu. Při nutnosti údržby ikona pulzuje. Dále vyvolává script, který otvírá okno s oknem údržby |
|  | |

4.4.9 Simulace a ladění

Simulace a ladění programu se provádělo v programech IntouchViewer a SMC. Kde IntouchViewer slouží k zobrazení aplikace v reálném čase a za pomoci SMC je možné vypisovat jednotlivé logy proměnných. Příklad, jak logovat proměnné, lze nalézt na obrázku (Obr. 4.15), výpis z programu zalogované proměnné je na obrázku (Obr. 4.20).



The screenshot shows the SMC - [Archestra System Management Console (MANDAREK-8A2135)\Log Viewer\Default Group\Local] window. The interface includes a menu bar (Soubor, Akce, Zobrazit, Nápověda), a toolbar with various icons, and a tree view on the left showing the project structure. The main area displays a table of log entries with the following data:

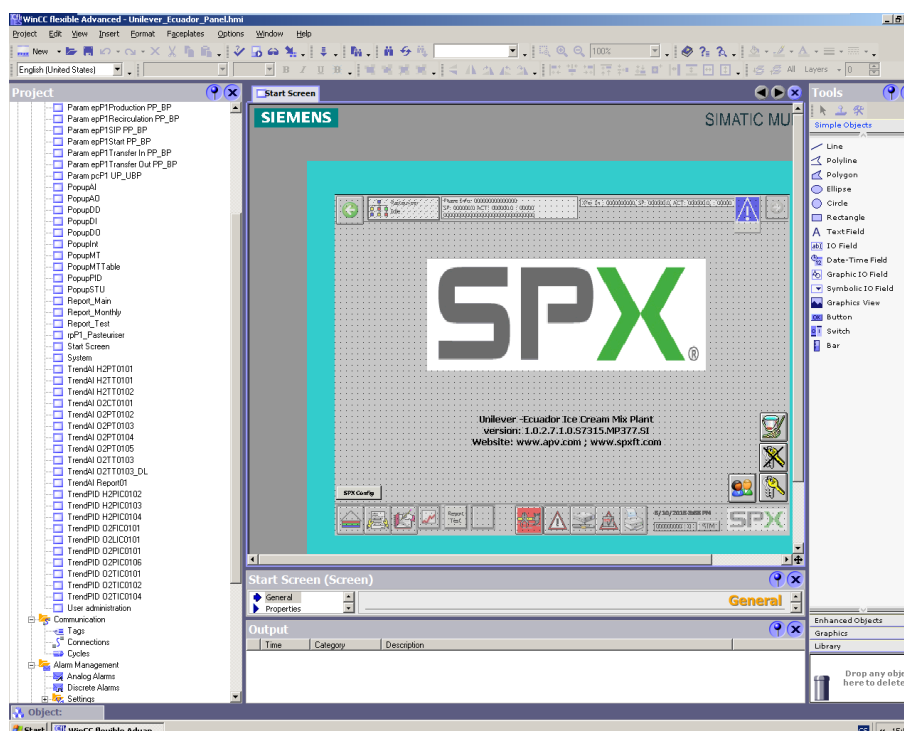
No:	Date	Time	Process ID	Thread ID	Log Flag	Component	Message
345001	25.4.2015	17:41:46	2800	3388	Info	WWSCRIPT	Message: 0
345002	25.4.2015	17:42:27	2800	3388	Info	WWSCRIPT	Message: 0
345007	25.4.2015	17:42:31	2800	3388	Info	WWSCRIPT	Message: 0
345012	25.4.2015	17:42:45	2800	3388	Info	WWSCRIPT	Message: 30
345053	25.4.2015	18:06:29	2800	3388	Info	WWSCRIPT	Message: 30
345062	25.4.2015	18:12:26	2800	3388	Info	WWSCRIPT	Message: 30

Obr. 4.20: Zobrazení výpisu logovaných proměnných

4.5 Vizualizace - WinCC Flexible 2008

Na rozdíl od SCADA systému InTouch se jedná o software, vyvinutý pro jednoduché programování HMI operátorských panelů. Výhodou tohoto softwaru je jednoduché propojení se Step 7 v5.5, kde odpadá velké množství problémů, například v tvorbě tagů. Při návrhu modulu je nutné, že HMI panely jsou méně výkonné než SCADA systémy, a tak nemohou disponovat tolika funkcemi, které by omezily chod aplikace.

Tak jako v prostředí InTouch se dají ve WinCC Flexible 2008 psát scripty v obdobném jazyce, velice podobném VB a C.



Obr. 4.21: Prostředí WinCC Flexible 2008

4.5.1 Nastavení komunikace s PLC

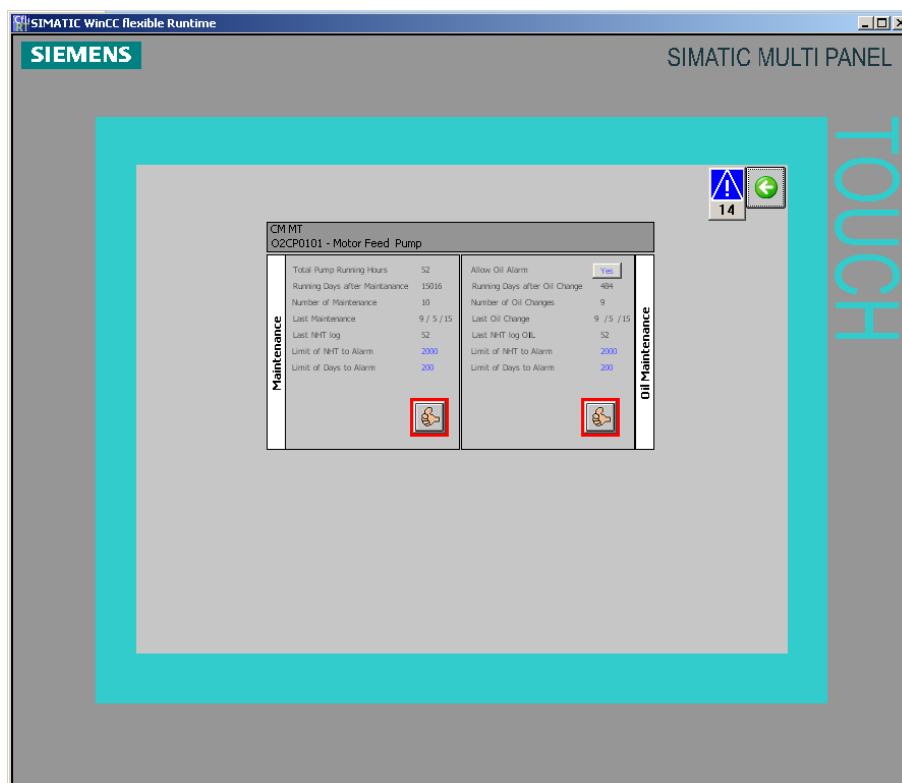
Pro komunikaci s PLC byla nastavena stejná IP adresa a stejnou maskou podsítě jako v kapitole 4.4.2, jen pro HMI panel se zvolila imaginární IP adresa v síti.

4.5.2 Popup okno pro údržbu

Popupové okno údržby pro WinCC Flexible 2008 je rozvrženo stejným způsobem jako okno v prostředí InTouch a obsahuje i stejné funkce. Viz kapitola 4.4.6. Liší se

pouze rozdílnými ikonami a principem realizace.

Okno se otevírá za pomoci ikony nářadí v okně na obrázku (Obr. 4.23). Po stisknutí tlačítka se spustí script, který vypočítá, na jaké adrese se umísťuje zobrazovaný člen, a zobrazí okno údržby. Po otevření okna údržby jsou již předadresovány všechny nepřímo adresované tagy, které vyčítají hodnoty z DB údržby.

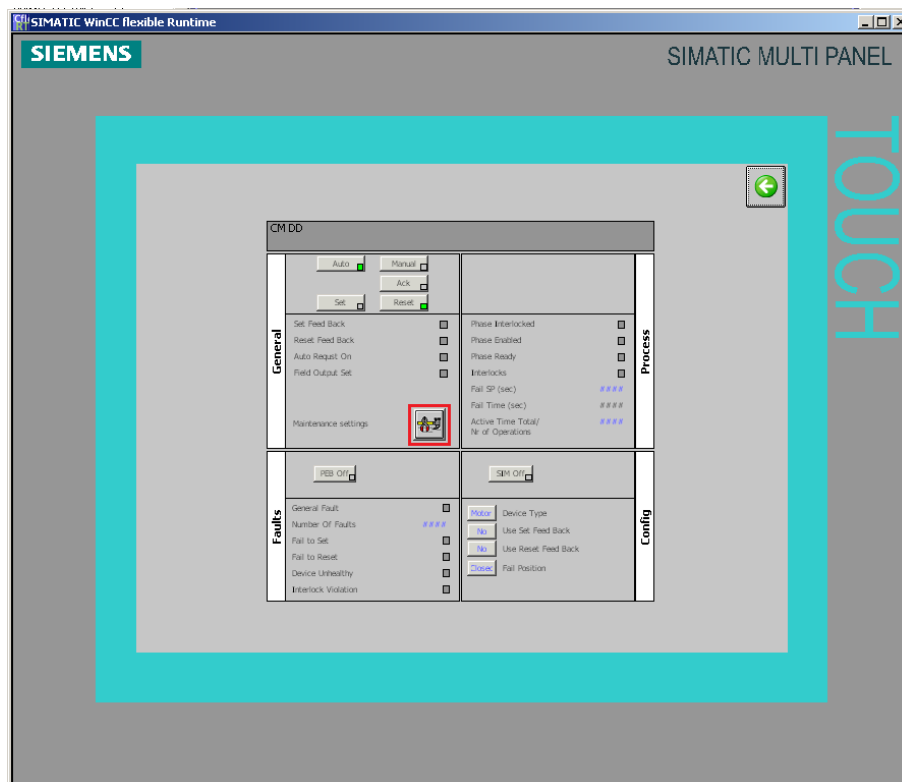


Obr. 4.22: Popup MT pro WinCC

Princip realizace

Z důvodu rozdílných prostředí WinCC Flexible 2008 a InTouch je rozdílná i samotná realizace vizualizace. Nepřímá adresace ve WinCC Flexible 2008 probíhá pomocí tagů ve tvaru: CM_INT_XX DB [CM_DB] DBW [CM_Int_Index_XX], kde vzorová proměnná obsahuje hodnotu v datovém typu INT a její adresa v PLC je závislá na parametrech CM_DB a $CM_Int_Index_XX$. Tyto dva tagy jsou též datového typu INT a na základě jejich hodnot se dotazuje na místo v paměti PLC. Na obrázku (Obr. 4.25) lze vidět podobný návrh tagů pro okno Popup MT Table.

Celý script je tvořen objektem jménem $SmartTag('xxx')$, díky tomuto objektu je možné přistupovat k proměnné jen za pomoci jejího stringového názvu. Pomocí tohoto lze vytvořit libovolný string a dotazovat se takto na tag.



Obr. 4.23: Popup DD pro WinCC

4.5.3 Popup okno tabulky pro údržbu

Význam tohoto okna je obdobný jako v programu InTouch, který je v kapitole 4.4.7.

Rozdíly mezi okny:

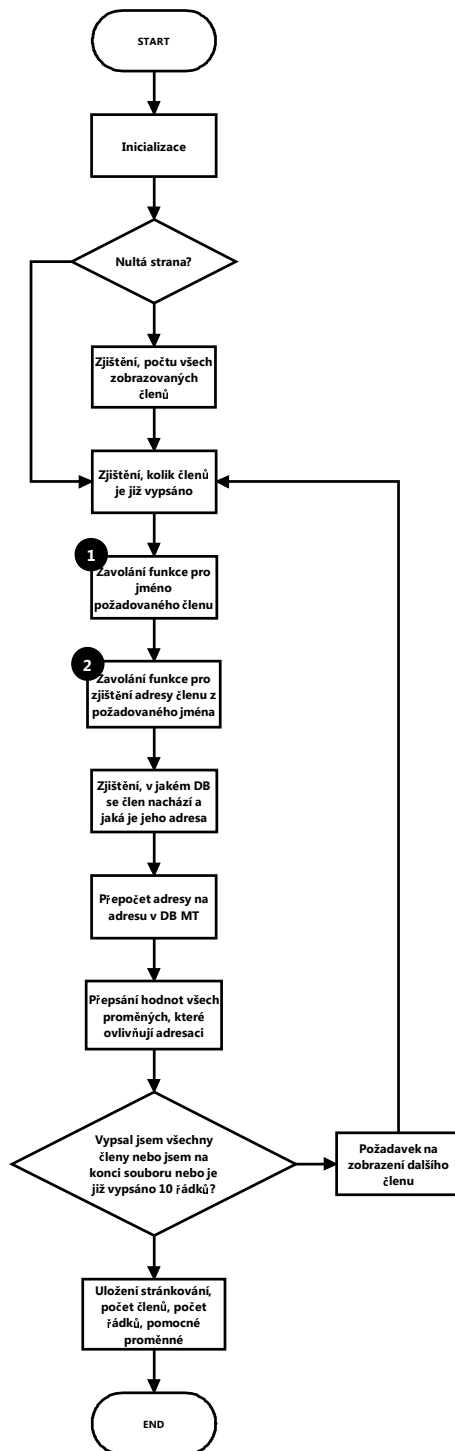
- Při požadavku přejít o stránku zpět se operátor dostává na nultou stránku.
- Číslování stránek je od nulté stránky, je to způsobené navrženým algoritmem.

Princip realizace

Výčet všech členů v systému je realizován pomocí tří funkcí, první již vytvořené $CMDB_OffsetID$ **1**, která za pomoci vstupního parametru jména akčního členu vrací jeho adresu v hodnotě bytů. Druhá funkce $CMDB_OffsetID_MT$ **2** je vytvořena za pomoci kopie první funkce a její následné editace. Při zavolání funkce se vstupním číselným parametrem vrací jméno akčního členu nebo string *end*. Třetí funkce volá obě dvě funkce a stará se o vypsání informací o akčních členech do tabulky. Rozepsaný algoritmus je na obrázku (Obr. 4.24).

Každý řádek tabulky je tvořen několika poli. Pro prvky, které mají funkci další spuštění akce (např. Item, Oil Alarm), jsou použity *Buttons*. Pro pole, která zobrazují hodnoty jsou použity *Faceplates*, které obsahují několik *I/O Fields*. To z důvodu

přehlednosti a rychlejší tvorby každého řádku. Například Faceplate pro datum poslední výměny oleje je tvořen 4 tagy, a to pro den, měsíc, rok, povolení oleje a textovými separátními znaky „/“.



Obr. 4.24: Algoritmus pro Popup MT Table WinCC

Name	Display name	Connection	Data type	Address	Array
MTTable_90		<Internal tag>	String	<No address>	1
MTTable_91		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_91]	1
MTTable_910		PLC01	Real	DB [MT_DB_9] DBD [MT_Index_910]	1
MTTable_911		PLC01	Real	DB [MT_DB_9] DBD [MT_Index_911]	1
MTTable_912		PLC01	Real	DB [MT_DB_9] DBD [MT_Index_912]	1
MTTable_913		PLC01	Real	DB [MT_DB_9] DBD [MT_Index_913]	1
MTTable_914		PLC01	Real	DB [MT_DB_9] DBD [MT_Index_914]	1
MTTable_915		<Internal tag>	Float	<No address>	1
MTTable_92		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_92]	1
MTTable_93		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_93]	1
MTTable_94		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_94]	1
MTTable_95		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_95]	1
MTTable_96		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_96]	1
MTTable_97		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_97]	1
MTTable_98		PLC01	Int	DB [MT_DB_9] DBW [MT_Index_98]	1
MTTable_99		PLC01	Byte	DB [MT_DB_9] DBB [MT_Index_99]	1
MTable_AlmMtr		PLC01	Bool	DB 3 DBX 0.0	1
MTable_AlmVlv		PLC01	Bool	DB 2 DBX 0.0	1

Obr. 4.25: Příklad návrhu tagů pro 10. řádek, Popup MTable

Item	Number Maintenances	Last Maintenance	Oil Alarm	Number Oil Maintenances	Last Oil Maintenance
O2XV0117	3	10/5 /15			
O2XV0118	3	10/5 /15			
O2XV0119	3	10/5 /15			
O2XV0120	3	10/5 /15			
O2XV0122	3	10/5 /15			
O2XV0123	3	10/5 /15			
O2AV0124	3	10/5 /15			
H2AV0101	3	10/5 /15			
H2AV0104	3	10/5 /15			
O2CP0101	10	9/5 /15	Yes	9	9/5 /15

Obr. 4.26: Okno Popup MTable

4.5.4 Simulace a ladění

Simulace vyhotovené aplikace byla prováděna ve WinCC Runtime simulátoru, protože nebylo možné otestovat vizualizaci na reálném HMI.

Ladění scriptů probíhalo za pomoci programu Microsoft Visual Studio 2010.

4.6 Porovnání prostředí pro vizualizaci

Oba typy vizualizačních prostředí mají své výhody i nevýhody. V celkovém měřítku jsou možnosti prostředí InTouch širší oproti WinCC Flexible 2008. Proto budu porovnávat nedostatky WinCC oproti prostředí InTouch, musíme brát v úvahu, že WinCC je pouze HMI software.

- Velice složité odladování složitějších scriptů.
- Při jakékoliv změně programu je nutné znovu kompilovat program a následně znovu otevřít aplikaci runtime.
- Při nepřímé adresaci je nutno vytvořit 3x více tagů.
- Chybí několikanásobné podmínky například pro visibility nebo změny barvy.
- Není možné k zobrazovanému tagu přičíst například plus 1 nebo zobrazit rozdíl dvou tagů.
- Pomalejší systém na rozsáhlé projekty.
- Faceplaty nemohou vyvolávat událost na globální úrovni.
- Nedostačující nápověda a dokumentace k nepřímé adresaci.

Tyto rozdíly jsou vypsány z doposud získaných zkušeností.

5 MANUÁL PRO IMPLEMENTACI MODULU DO REÁLNÉHO PROJEKTU

Tato kapitola obsahuje stručný návod k přenesení modulu do jakéhokoliv projektu. Následující manuál je vytvořen pro DD akční členy.

Přenos do PLC automatu S7-300/400:

1. Překopírovat bloky **UDT9**, **FC3**, **FC13**, **FC1002** v PLC.
2. Kopírovat DB pro ventily a motory DD a tyto bloky uložit pod neobsazeným číslem. (Výchozí DB2 a DB3)
3. Na **adresu 0.0** v každém překopírovaném **DB** vložit proměnnou typu **BOOL**, **Alarm**.
4. Ve funkci **FC3** přepsat čísla DB. Nastavit inicializační limitní hodnoty pro NHT a dny.
5. Ve funkci **FC13** přepsat v networku 18 a 19, **DB** pro proměnnou **Alarm** na adrese 0.0.
6. Vytvořit OB, které se bude volat **1x denně** a zakomponovat do něho volání funkce FC3.
7. Vložit volání funkce **FC3** do bloku **OB1**, kód zkopírovat ze vzorového projektu.

Přenos do Wonderware InTouch:

1. Provést kopii oken **Footer**, **Popup DD**, **Popup MT**, **Popup MT Table**.
2. Překopírovat funkce **indDD**, **indMTRefresh**, **indMTTable**.
3. Překopírovat tagy **indMT**, **MTTable*** a **txtTag707-735**.
4. Překopírovat položky, **txtTag707-735** ze souboru **StaticText.csv**.

Přenos do WinCC Flexible 2008:

1. Provést kopii oken **Template**, **PopupDD**, **PopupMT**, **PopupMTTable**.
2. Překopírovat funkce **CM_MT_Popup**, **MT_Popup_AL**, **MT_Table**, **MT_Table_Refresh**.
3. Udělat kopii funkce **CMDB_OffsetID**, přejmenovat ji na **CMDB_OffsetID_MT**. V **casu CM_DD_Mtr** a **CM_DD_Vlv** smazat vše ostatní. Na konec každého **casu** přidat položku **Case Else DB_Offset = „end“**. Dále přemístit **DB_Offset =** před jméno akčního členu a přebývající hodnotu smazat. poté očíslovat **case** od 0 do konečné položky.
4. Překopírovat tagy **MTTable***.

6 ZÁVĚR

Cílem bakalářské práce bylo, seznámit se s problematikou údržby, nastudovat si potřebné materiály k orientaci ve firemním standardu pro PLC/HMI firmy SPX s. r. o. a realizovat modul údržby. Modul údržby implementovat do PLC a vizualizovat data údržby pro prostředí Wonderware InTouch a WinCC Flexible 2008. Modul přitom nesmí ovlivnit funkce PLC, nezatěžoval celý systém a vizualizační návrh byl uživatelsky přívětivý.

Realizovaný modul údržby obsahuje dva systémy údržby. Systém preventivní údržby a systém diagnostické údržby. Tyto dva systémy údržby jsou implementovány v podobě algoritmu napsaném v jazycích FBD a STL. Modul údržby v PLC se volá 1x denně a lokace všech akčních členů k údržbě probíhá za běhu programu pomocí funkce FC3 (kapitola 4.3.4). Následná funkce vyhodnocování je volána se vstupními parametry ukazatelů adres akčních členů, FC13 (kapitola 4.3.5). Funkce vyhodnotí data a uloží je do datových bloků údržby. Datové bloky údržby jsou tvořeny UDT9 pro každý akční člen v systému.

Vizualizace je realizována ve SCADA systému Wonderware InTouch, ve kterém jsou implementovány dvě popup okna pro vizualizaci dat údržby z PLC. Okno *Popup MT* je možné zobrazit pro akční členy ventilů a motorů. Obsahuje všechny informace o údržbě, které jsou dostupné v DB. V okně je možné měnit limitní parametry pro alarm a vypínat alarm údržby. Přístup k tomuto oknu je možný přes *Popup DD*. Druhé okno, *Popup MT Table* je přehledná tabulka, která vypisuje všechny členy v systému u kterých je hlídána údržba. Z okna jde velice jednoduše otevřít jakýkoliv DD člen v systému. Pro lepší orientaci uživatele byla do tabulky přidána funkce filtrování a možnost změny výpisu informací. Tyto možnosti lze ovládat za pomoci přepínačů.

Stejná vizualizace byla vytvořena i pro systém WinCC Flexible 2008. Do tohoto prostředí byli implementovány také dvě popup okna, která mají stejnou funkci, jak je popsáno výše.

Zadání bakalářské práce bylo splněno a všem požadavkům firmy bylo vyhověno. V následující fázi čeká modul údržby další vývoj, aby bylo možné modul využít pro reálné projekty. V prostředí InTouch by bylo vhodné optimalizovat modul údržby za pomoci snížení tagů. Počet tagů v systému určuje cenu programu a stanovuje jeho zátěž. Prostředí WinCC se musí optimalizovat také z důvodu velkého množství tagů. To bylo způsobeno nevhodným počátečním návrhem modulu.

LITERATURA

- [1] PANTEK (CS) S.R.O. Pantek .Wonderware software. [online], 2007-2009, [cit. 2015-05-1].
URL <<http://tagstore.org>>
- [2] BALDA, P.: Informační a řídicí systémy I. [online], 04. 2007, [cit. 2015-05-1].
URL <http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-08_SCADA_HMI.pdf>
- [3] PÁSEK, J.: *Programovatelné automaty v řízení technologických procesů*. Brno: Fakulta elektrotechniky a komunikačních technologií, 2007, 138 str.
- [4] Siemens: *SIMATIC Programming with STEP 7*. 05 2010, 6ES7810-4CA10-8BW0.
- [5] SKŘIVÁNEK, M.; POLÍVKA, E.: *Provozní spolehlivost a údržba strojů: Určeno [též] stud. na školách techn. a ekon. směru*. Praha: 2. vyd. Praha: SNTL, 1976, 250 str.
- [6] SVĚTLÍK, V.: MES (Manufacturing Execution Systems) Informační systémy zaměřené na přímou výrobu. IT System. [online], 2001, [cit. 2015-05-1].
URL <<http://www.automatizace.cz>>
- [7] VOŠTOVÁ, V.; HELEBRANT, F.; JERÁBEK, K.: *Provoz a údržba strojů*. Praha: Vydavatelství ČVUT, Vyd. 1., 2002, ISBN 80-01-02531-4, 124 str.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

PPO Systém preventivních oprav

DIPP Systém diferencované proporcionální starostlivosti

TPM Systém totálně produktivní údržby

MES (Manufacturing Execution System) -systém řízení výroby

SCADA (Supervisory Control And Data Acquisition) -dispečerské řízení a sběr dat

PLC (Programmable Logic Controller) - Programovatelný logický automat

DB (Data blok) - datový blok

OB (Organization blok) - organizační blok

FC Funkce

UDT (User Defined Type) - uživatelem definovaný typ

PC (Process cell) - procesní buňka

EM (Equipment modul) - modul zařízení

CM (Control modul) - ovládací modul

SMC Arcestra System Management Console

UHT (Ultra high temperature) - velmi vysoká teplota

FBD (Function Block Diagram) - jazyk funkčních bloků

STL (Statement List Control Language) - strojový kód

VB Visual Basic

SEZNAM PŘÍLOH

A Přílohy	I
A.1 Přiložené CD	I

A PŘÍLOHY

A.1 Přiložené CD

Na přiloženém CD je uložena elektronická verze bakalářské práce a výpis z programu Step7, Wonderware InTouch a WinCC Flexible 2008. CD také obsahuje soubor Práce.zip, ve kterém jsou uloženy veškeré obrázky, zdrojové kódy pro L^AT_EX a další soubory.

```
/
├── PLC
│   ├── PLC_FC3(pdf)
│   ├── PLC_FC13(pdf)
│   ├── PLC_OB1(pdf)
│   ├── PLC_DB2(pdf)
│   ├── PLC_DB3(pdf)
│   └── PLC_UDT9(pdf)
├── Wonderware InTouch
│   ├── InTouch_indMTRefresh(pdf)
│   └── windows(zip)
├── WinCC Flexible 2008
│   ├── WinCC_MT_Table(pdf)
│   ├── WinCC_MT_Table_Refresh(pdf)
│   ├── WinCC_Popup MT Table(pdf)
│   ├── WinCC_Popup MT(pdf)
│   └── WinCC_CM_MT_Popup(pdf)
├── BPxtichy25(pdf)
└── Práce(zip)
```