

# **Propojení PLC SAIA s nadřazeným počítačem pomocí sítě Ethernet**

Josef Martinák

---

Bakalářská práce  
2006



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2005/2006

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Josef MARTINÁK**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Propojení PLC SAIA s nadřazeným počítačem pomocí sítě Ethernet**

### Zásady pro vypracování:

- Seznamte se s technickým zařízením použitým v laboratoři mokrých procesů pro recyklaci koželužského odpadu, především s PLC SAIA a moduly Advantech pro připojení zařízení se sériovým výstupem do sítě Ethernet.
- Seznamte se s PLC SAIA z hlediska tvorby programů, řízení technologických procesů a komunikace. Zpracujte literární rešerši na toto téma.
- Seznamte se s prostředím pro vývoj a provozování aplikací reálného času Control Web.
- Navrhněte a realizujte základní programové vybavení pro komunikaci mezi PLC a programem běžícím na nadřazeném počítači v prostředí Control Web.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Pokorný, P.: Problematika řízení při hydrolyzačním zpracování bílkovinných druhotných surovin. Disertační práce FT UTB Zlín, 2003.
2. Horných, J.: Aplikace PLC automatů pro řízení enzymatické hydrolýzy zpracování koželužských odpadů. Diplomová práce, Zlín, 1999.
3. SAIA-Burgess Electronics: Programming tools for MS Windows PG4. Vydání 26/748 E1, 1996
4. SAIA-Burgess Electronics: Hardware - Řady PCD1 a PCD2. Vydání 26/737 C5, 1998.
5. Husták, P.: Počítačové řízení rotační odparky. Diplomová práce VUT Brno, FT Zlín, 1999.
6. Moravské přístroje a.s: Elektronická příručka k programu Control Web 5, <http://www.mii.cz>.

Vedoucí bakalářské práce:

**Ing. Jan Dolinay**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

**14. února 2006**

Termín odevzdání bakalářské práce:

**16. června 2006**

Ve Zlíně dne 14. února 2006

  
prof. Ing. Vladimír Vašek, CSc.  
*pověřený děkan*



  
doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Cílem práce je realizovat připojení programovatelného automatu SAIA se sériovým rozhraním do centralizovaného řídicího systému, na bázi sítě Ethernet, s využitím Ethernet/RS konvertoru Adam-4571 firmy Advantech. Teoretická část seznamuje s použitým technickým zařízením, především PLC SAIA a s prostředím pro vývoj a provozování aplikací reálného času Control Web. Praktická část popisuje realizaci zapojení s využitím modulu ADAM-4571 a vývoj programového vybavení pro komunikaci mezi PLC a programem běžícím na nadřazeném počítači v prostředí Control Web.

Klíčová slova: programovatelný automat, SAIA PCD2, Ethernet, ADAM-4571, Control Web

## **ABSTRACT**

The aim of this work is a realization of the connection of a Programmable Logic Controller SAIA with a serial port to the centralized operating system on the base of the net Ethernet with the use of Ethernet/RS convertor ADAM-4571 made by Advantech. The theoretical part deals with the technical arrangement, above all PLC SAIA and with the environment for developing and operating realtime applications Control Web. The practical part describes the realization with the use of module ADAM-4571 and the development of the program equipment for communication between PLC and the program running on the superior computer in the environment of Control Web.

Keywords: Programmable Logic Controller, SAIA PCD2, Ethernet, ADAM-4571, Control Web

Děkuji panu Ing. Janu Dolinayovi, vedoucímu mé diplomové práce, za pomoc v průběhu jejího řešení.

# OBSAH

<b>ÚVOD</b> .....	<b>5</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>7</b>
<b>1 LABORATOŘ RECYKLAČNÍCH TECHNOLOGIÍ</b> .....	<b>8</b>
1.1 POPIS TECHNOLOGIE LABORATOŘE.....	8
1.2 CELKOVÁ KONCEPCE LABORATOŘE.....	8
<b>2 ÚVOD DO PROBLEMATIKY PLC</b> .....	<b>10</b>
2.1 ZÁKLADNÍ POJMY .....	10
2.2 ZAŘAZENÍ PROGRAMOVATELNÉHO AUTOMATU MEZI ŘÍDÍCÍ SYSTÉMY .....	11
2.3 HLAVNÍ CHARAKTERISTIKY PROGRAMOVATELNÝCH AUTOMATŮ .....	12
2.4 TYPY PROGRAMOVATELNÝCH AUTOMATŮ Z HLEDISKA VELIKOSTI A KONSTRUKČNÍHO USPŘÁDÁNÍ .....	14
<b>3 RODINA AUTOMATŮ SAIA PCD</b> .....	<b>16</b>
<b>4 PROGRAMOVATELNÝ AUTOMAT SAIA PCD2</b> .....	<b>17</b>
4.1 DODÁVANÉ TYPY VSTUPNĚ-VÝSTUPNÍCH MODULŮ .....	17
4.2 TYPY DATOVÝCH PROSTŘEDKŮ VYUŽITELNÝCH V PROGRAMECH .....	18
4.3 STRUKTURA ULOŽENÍ PROGRAMU .....	19
<b>5 VÝVOJOVÉ PROSTŘEDÍ PG4</b> .....	<b>21</b>
5.1 INSTRUCTION LIST.....	23
5.2 FUNCTION PLAN – FUPLA (FBD).....	24
5.3 GRAFTEC .....	26
<b>6 CONTROL WEB (CW)</b> .....	<b>27</b>
6.1 VÝVOJ APLIKACÍ V PROSTŘEDÍ CW.....	27
6.2 FILOZOFIE TVORBY APLIKACE V PROSTŘEDÍ CW .....	28
6.3 PODPOROVANÉ PROTOKOLY A STANDARDY .....	31
<b>II PRAKTICKÁ ČÁST</b> .....	<b>32</b>
<b>7 HARWAROVÁ ČÁST</b> .....	<b>33</b>
7.1 KONFIGURACE PCD2 PRO APLIKACI ŘÍZENÍ PRACOVNÍŠTĚ FERMENTOR.....	33
7.1.1 Analogový vstupní modul PCD2.W200 .....	34
7.1.2 Dvouhodnotový výstupní modul PCD2.A400 .....	34
7.1.3 Analogový výstupní modul PCD.W400 .....	35
7.1.4 Vstupní a výstupní signály PCD2.....	35
7.2 PŘIPOJENÍ PCD2 K PC POMOCÍ ETHERNETU .....	36
7.2.1 Konektor PGU jako obecné sériové rozhraní RS232 .....	36
7.2.2 Modul ADAM 4571-A .....	36
7.2.3 Realizace zapojení PCD2 a modulu ADAM-4571-A.....	37
<b>8 PROGRAMOVÁ REALIZACE KOMUNIKACE CW-PCD2</b> .....	<b>39</b>
8.1 INSTALACE A KONFIGURACE MODULU ADAM-4571-A .....	39
8.2 PROGRAM PLC PRO PRACOVNÍŠTĚ FERMENTOR.....	42
8.3 OVLADAČ PRO KOMUNIKACI SE STANICEMI SAIA PCD PRO SYTÉM CW .....	43
8.4 KOMUNIKACE SAIA PCD2 – CONTROL WEB .....	45
<b>ZÁVĚR</b> .....	<b>50</b>

<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>52</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>53</b>
<b>SEZNAM OBRÁZKŮ.....</b>	<b>54</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>56</b>

## ÚVOD

Na našem institutu byla vyvinuta unikátní technologie recyklace chromitého odpadu, která si již získala celosvětový ohlas. Za účelem dalšího výzkumu a zlepšování je v laboratorních podmínkách realizován model této technologie.

Z hlediska ekonomického i ekologického se pro zpracování chromočiněných odpadů jeví nejvýhodnější využití enzymatické hydrolýzy. Technologie pro toto zpracování je realizována v laboratoři. Celý proces pro zpracování kožedělných odpadů pomocí enzymatické hydrolýzy je možno rozdělit na 4 pracoviště a to: fermentace, filtrace, sušení (odparka) a recyklace. Koncepce laboratoře vychází z hierarchické struktury, kde na vrcholu je centrální počítač propojený s jednotlivými pracovišti technologie pomocí průmyslového Ethernetu.

V nedávné době bylo rozhodnuto o celkové rekonstrukci a modernizaci vybavení uvedené laboratoře. U pracovišť fermentor a filtrace bylo rozhodnuto o ponechání stávajícího vybavení založeného na řídicích PLC firmy Saia, pouze s případnými změnami na místě snímačů a akčních členů. Pro začlenění těchto zařízení do sítě Ethernet jsou použity moduly konvertoru Ethernet/RS ADAM-4571-A. [1]

Nejrozšířenějším přenosovým médiem dneška je bezesporu internet. Internetové technologie jsou ověřeným standardem s širokým rozšířením a s velkou budoucností. Z libovolného místa ve světě můžeme získat informace, které jsou k dispozici na internetu. Stále více lidí využívá pravidelně internet, málokdo si však uvědomuje, jak lze toto médium využít například při řízení technologického procesu.

Můžeme mít technologický proces, který je nutné řídit ze dvou míst. Vstupní data mohou být zadávána ze vzdáleného pracoviště a výstupní data musí být k dispozici na dalším pracovišti. Klasický způsob řešení by zahrnoval vybudování komunikačních linek (metalických, optických nebo rádiových). Nehledě na pracnost takového řešení je k vybudování těchto komunikačních linek zapotřebí vynaložit potřebné finanční prostředky. Nabízí se tedy využití Ethernetu, který je dostupný prakticky na každém pracovišti vybaveném výpočetní technikou. Náklady na zařízení nutné k začlenění technologického procesu do sítě Ethernet mohou být mnohem menší než náklady na vybudování komunikačních linek. Navíc data mohou být přístupná na kterémkoli místě připojeném na síť Ethernet, případně prostřednictvím internetu mohou být data přístupná



kdekoli na světě. Pro takovéto aplikace vyvinula firma Advantech zařízení s označením ADAM-4571. Jedná se o datovou bránu mezi rozhraním RS232/485/422 a Ethernetem.

Předmětem práce je realizovat propojení programovatelného automatu SAIA s nadřazeným počítačem pomocí sítě Ethernet a ověřit možnosti jejich vzájemné komunikace. Pro připojení bude použit Ethernet/RS konvertor ADAM-4571 firmy Advantech. Pro vytvoření programového vybavení na straně PC a jako základ celého řídicího systému bude použito programové vybavení Control Web firmy Moravské přístroje a.s.

## **I. TEORETICKÁ ČÁST**

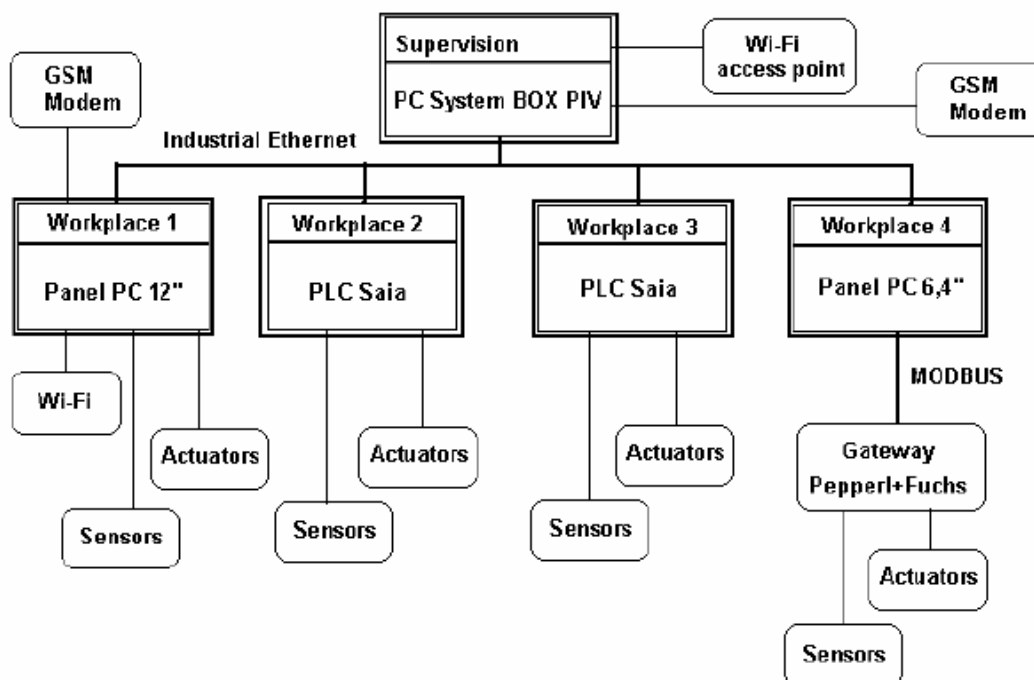
# 1 LABORATOŘ RECYKLAČNÍCH TECHNOLOGIÍ

## 1.1 Popis technologie laboratoře

Celý proces pro zpracování kožedělných odpadů pomocí hydrolyzy je možno rozdělit na 4 pracoviště a to : fermentace, filtrace, sušení (odparka) a recyklace.

Prvním krokem je chemická reakce v bioreaktoru (fermentoru). Produkt této reakce se následně řízeným způsobem filtruje a výsledný produkt suší v podtlakové odparce. Filtrační koláč z pracoviště filtrace je dále naplněn do kalolisu čtvrtého pracoviště recyklace, kde je využit k odstranění chrómu z koželužské chromité odpadní vody. Na tomto pracovišti se kapalné koželužské odpady obsahující chróm promývají přes filtrační koláč tak dlouho, až obsah chrómu poklesne pod stanovenou mez.

## 1.2 Celková koncepce laboratoře



Obr. 1. Celkové schéma řídicího systému

Celá technologie je v laboratoři rozdělena na 5 pracovišť, z nichž 4 jsou přímo spojena s technologií a jedno je nadřazené. Koncepce zapojení pak vychází z hierarchické struktury, kde na vrcholu je centrální počítač propojený s jednotlivými pracovišti technologie pomocí průmyslového Ethernetu a duplicitně i pomocí bezdrátové sítě WiFi či GSM.

Na jednotlivých technologických pracovištích je pak použito průmyslové PC s dotykovým displejem, průmyslový počítač rozměru PDA a dva PLC automaty. Na nejnižší úrovni jsou kombinovány jak standardní senzory s výstupem 4-20 mA, tak i inteligentní senzory propojené pomocí sériového rozhraní či sběrnice MODBUS v případě specializovaného systému Pepperl + Fuchs.

Stěžejní části systému jsou postaveny na technologii firmy Advantech. Na nejvyšší úrovni řízení je použit osobní počítač v průmyslovém provedení typu BOX PIV. Tento počítač má sloužit především jako nadřazené pracoviště, odkud je možno sledovat celou technologii. Při použití příslušného programového vybavení se ovšem počítá i s možností jednotlivé technologie z tohoto pracoviště přímo řídit.

Počítač je v konfiguraci odpovídající současnému standardu, tj. procesor Pentium 4, 3 GHz 512 MB RAM, pevný disk s kapacitou 160 GB, operační systém Microsoft Windows XP Professional.

Propojení mezi centrálním počítačem a jednotlivými pracovišti je realizováno průmyslovým Ethernetem. Ve fyzické vrstvě je použit Ethernet switch v průmyslovém provedení ADAM 6520-B, který podporuje přenosové rychlosti 10/100 Mbps.

Propojení tohoto počítače s pracovištěm recyklace je realizováno také duplicitně pomocí bezdrátového Ethernetu (Wi-Fi). Toto propojení je postaveno na standardních prvcích, centrální počítač vytváří přístupový bod a podřízená pracoviště se mohou připojit k tomuto bodu.

Centrální počítač a počítač na pracovišti recyklace jsou navíc vybaveny GSM modemy tak, aby bylo možno demonstrovat i tuto perspektivní technologii při použití v praktické řídicí aplikaci.

U pracovišť fermentor a filtrace jsou použity řídicí PLC firmy SAIA. Pro začlenění těchto zařízení do sítě Ethernet jsou použity moduly konvertoru Ethernet na RS 232/422/485 ADAM 4571-A. Jedná se o zařízení, které vytváří virtuální sériovou linku prostřednictvím sítě Ethernet. Z hlediska nadřazeného počítače se celé zapojení jeví jakoby bylo PLC připojeno přímo k sériovému portu tohoto počítače. Podporována je přenosová rychlost až 230 kb za sekundu. [1]

## 2 ÚVOD DO PROBLEMATIKY PLC

### 2.1 Základní pojmy

Programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů nebo strojů, mnohdy specializovaný na úlohy převážně logického typu (obzvláště u starších typů nebo meších systémů). Nejčastěji se označuje zkratkou PLC (Programmable Logic Controller), v německé literatuře se lze setkat s označením SPS (Speicherprogrammierbare Steuerung). Občas najdeme i označení PC (Programmable Controller). Česká zkratka, která se teprve začíná používat, je PA (Programovatelný Automat).

Původně byly programovatelné automaty navrženy k řešení úloh logického řízení, často jako přímá náhrada pevné reléové logiky. V současných aplikacích se však zvyšuje podíl úloh regulačního typu, úloh monitorování i úloh analogových měření.

Každý programovatelný automat se v podstatě skládá z centrální procesorové jednotky, systémové paměti, souboru vstupních a výstupních jednotek pro připojení řízeného systému a souboru komunikačních jednotek pro komunikaci se souřadnými i nadřazenými řídicími systémy. Jednotky programovatelného automatu jsou navzájem propojeny systémovou sběrnici.

Řídicí algoritmy jsou realizovány uživatelským programem, který může být zapsán v různých programovacích jazycích a po přeložení je uložen v uživatelské paměti programovatelného automatu. Program obsahuje posloupnost instrukcí, které procesor vykonává cyklicky.

Chování programovatelného automatu je tedy dáno v podstatě zaměnitelným programem, zatímco u reléových systémů bylo chování určeno strukturou zapojení jednotlivých komponent, která byla po realizaci reléových systémů téměř nezměnitelná. Proto se zvláště v počátcích vývoje programovatelných automatů hovořilo o tzv. volně programovatelné logice a mnoho programovatelných automatů se označovalo zkratkou FPC (Free Programmable Controller).

## 2.2 Zařazení programovatelného automatu mezi řídicí systémy

### Relé a kontakty versus programovatelné automaty

Relé, stykače a tlačítka jsou v některých případech nenahraditelné a nemá smysl se bránit jejich použití i v případech, kdy je k řízení použit programovatelný automat. Z bezpečnostních důvodů se takto realizují bezpečnostní okruhy, např. obvod CENTRAL STOP. Nemá smysl bránit se příležitostnému vytváření logických funkcí prostým propojením kontaktů, obzvláště, pokud tím ušetříme počty vstupů a výstupů programovatelného automatu a snížíme tak jeho cenu. Rozsáhlejší funkce se kontaktní a reléovou technologií již nerealizují a svěříme je důsledně programu programovatelného automatu.

### Regulátory versus programovatelné automaty

Rozdíl mezi těmito dvěma tradičními kategoriemi výrobků – mezi regulátory a programovatelnými automaty se postupně stírá. Obě dvě skupiny těchto dnes vyráběných výrobků pracují číslicově. Hranice je tak neostrá, že je mnohdy obtížné rozhodnout, zda produkt je ještě regulátorem nebo již programovatelným automatem.

Výrobci programovatelných automatů postupně expandují do aplikačních oblastí dosud vyhrazených pro regulátory, a to jak aplikacemi tradičních PLC, tak i vývojem nových výrobků, které se mnohdy odklánějí od původní koncepce PLC. Na druhé straně tradiční výrobci regulátorů naopak své programovatelné regulátory uzpůsobují i pro logické řízení.

Vyrábějí např. inteligentní regulátory, které mají přidány logické funkce a jsou schopny pracovat v různých režimech podle stavu řízeného systému, a pronikají s nimi do oblastí typických pro nasazení programovatelných automatů.

Často je tedy rozlišení mezi programovatelným automatem a regulátorem v dnešní době spíše formální a je záležitostí tradice a firemní obchodní politiky.

### Personální počítače versus programovatelné automaty

Někdy se setkáváme s přímým řízením technologických procesů standardním PC, mnohdy umístěným přímo v technologii. Toto řešení je přinejmenším riskantní a diskutabilní. Běžný počítač kategorie PC je produkt spotřební elektroniky a je konstruován pro provoz v prostředí domácností, laboratoří a kanceláří, kde obvykle funguje s vyhovující spolehlivostí. V drsných průmyslových podmínkách mnohdy selhává (bývá málo spolehlivý, je citlivý na rušení a přepětí, nemá potřebnou životnost).

Problémy vznikají už s pouhým připojením většího počtu vstupních a výstupních vodičů a s jejich odrušením.

Průmyslové počítače (IPC, IC) se někdy používají při přímém řízení strojů a technologií, někdy jen v roli inteligentního operátorského panelu nebo komunikačního adaptéru. Problémem při jejich nasazování je vysoká cena. Jsou tedy účelné jen tam, kde je zdůvodněna, zejména při archivaci a zpracování velkých objemů dat, při využití obrazovky a standardního počítačového ovládání, při využívání výkonných komunikací, při řešení geometrických a jiných výpočetně náročných úloh.

Přímé řízení počítačem je dnes účelné jen v laboratorních podmínkách, pro potřeby výuky a řízení laboratorních a modelových úloh. Je-li v průmyslových podmínkách použití PC nezbytné, pak je nutno použít průmyslový typ. Standardním řešením je použití distribuovaného systému, kdy osobní počítač je ve velínu nebo na dispečerském pracovišti a do drsného průmyslového prostředí jsou předsunuty programovatelné automaty.

Někteří výrobci nabízejí v sortimentu modulů svého programovatelného automatu i počítačový modul kompatibilní s PC. V něm lze odpovídajícími prostředky řešit úlohy příslušející počítači. (složité a rychlé výpočetní algoritmy, grafické a geometrické úlohy, zpracování a archivace velkého množství dat, databázové úlohy, výkonné komunikace, napojení do počítačové sítě).

### **2.3 Hlavní charakteristiky programovatelných automatů**

#### ***VÝHODY:***

##### ***Rychlá realizace***

Hlavní předností programovatelných automatů je možnost rychlé realizace systému. Technické vybavení nemusí uživatel vyvíjet. Stačí navrhnout a včas objednat vhodnou sestavu modulů programovatelného automatu (konfiguraci) pro danou aplikaci, vytvořit projekt, napsat a odladit uživatelský program – a pak to vše realizovat a uvést do chodu.

##### ***Spolehlivost, odolnost, diagnostika***

Technické vybavení programovatelných automatů je navrženo tak, že jsou extrémně spolehlivé i v drsných průmyslových podmínkách, jsou odolné proti rušení i poruchám, vyznačují se robustností a spolehlivostí. Programovatelné automaty bývají vybaveny i vnitřními diagnostickými funkcemi, které průběžně kontrolují činnost systému a včas zjistí případnou závadu, lokalizují ji, bezpečně ji ošetří a usnadní její odstranění.

### ***Snadná přizpůsobitelnost řešení (nekončící změny v zadání)***

Jen někdy se podaří, že první varianta řešení zůstane tou poslední a konečnou. Představy zadavatele a koncového uživatele, ale i projektanta a programátora postupně zrají, požadavky se průběžně vyvíjejí a rozšiřují. Při uvádění do provozu je třeba všechny funkce důkladně prověřit a odstranit mnohé chyby a slabá místa. Dodatečné požadavky a zadání nových funkcí vznikají i po mnohých měsících a letech provozu

### ***Schopnost komunikace***

K neopomenutelným výhodám programovatelných automatů patří jejich schopnost komunikace s nejrůznějšími systémy a zařízeními jak v podřízené úrovni, v takzvaném poli, což je oblast senzorů, měřících zařízení a akčních členů, tak i v souřadné úrovni s ostatními programovatelnými automaty či jinými řídicími systémy, a v neposlední řadě i směrem k nadřízeným systémům. Právě tato schopnost komunikace umožňuje stavbu distribuovaných nebo i hierarchických systémů řízení z nejrůznějších komponent a od různých výrobců.

### ***NEVÝHODY:***

#### ***Prodloužení odezvy***

Řídicí systém s pevnou logikou se od systémů s PLC liší v době odezvy, tj. v době za kterou zareagují výstupy na změnu na vstupech systému. V pevné logice jsou všechny logické členy trvale aktivní, algoritmus systému se realizuje paralelně a ve spojitém čase. Odezva na změnu vstupů je dána jen celkovým zpožděním logických členů v nejvyšší větvi. U integrovaných obvodů to bývají řádově nanosekundy až milisekundy, u reléových systémů jednotky, desítky, někdy i stovky milisekund.

Odezva PLC bývá delší a je dána dobou průchodu programu. Závisí na rychlosti procesoru a na délce aktivní větve programu. Typicky nabývá hodnot v řádu jednotek až desítek milisekund (někdy stovek milisekund), což pro běžné aplikace postačuje. Je však třeba s touto skutečností počítat, aby v některých případech nebyla příčinou nečekaných překvapení.

#### ***Nespojitost v čase***

Dalším důležitým znakem programovatelných automatů je časová nespojitost zpracování. Algoritmus je vykonáván cyklicky, vždy jen v určitých okamžicích. Uvnitř intervalu mezi okamžiky aktivace systém nereaguje na změny vstupních hodnot. Tuto skutečnost je třeba



respektovat při návrhu programování systému, jinak může být příčinou hazardů a chyb, ztráty krátkého vstupního impulsu, nevyhodnocení hrany signálu apod.

### ***Postupnost zpracování***

Program v PLC je vykonáván v pořadí, v jakém je zapsán, nikoli v pořadí „toku signálů“ v odpovídajícím logickém schématu. Je-li možné zapsat PLC program sousledně s tokem signálů (aby pořadí instrukcí sledovalo tok signálu směrem od vstupů k výstupům), nebývají problémy. U složitých a nepřehledných logických funkcí (a zejména v případech mnoha změn a vsuvek do programu) se to vždy nepodaří. V lepším případě je následkem prodloužení odezvy systému (k ustálení hodnoty výstupu je zapotřebí několika cyklů PLC programu). V případě nesystematického návrhu sekvenčních funkcí (se zpětnými vazbami) může být následkem i chybná funkce programu nebo jeho zdánlivě nahodilé chyby (hazardy).

## **2.4 Typy programovatelných automatů z hlediska velikosti a konstrukčního uspřádání**

### ***Mikro PLC***

Nejmenší a nejlevnější kompaktní PLC systémy (mikro PLC) nabízejí pevnou sestavu vstupů a výstupů, obvykle jen binárních, konkrétních počtech podle modelu. Uživatel se v tomto případě může rozhodnout pro jeden typ systému, který již nemůže dodatečně rozšiřovat. Svým kompaktním provedením, malými rozměry a nízkou cenou se mikro PLC řadí do kategorie spotřebního materiálu. Jejich funkční a programátorský komfort je obvykle redukován na nezbytné minimum, komunikační možnosti mnohdy chybějí.

### ***Kompaktní PLC***

Ostatní PLC v kompaktním provedení nabízejí určitou, i když omezenou variabilitu ve volbě konfigurace. Uživatel může k základnímu modulu připojit jeden nebo několik přídatných modulů z omezeného sortimentu s pevnou kombinací vstupů a výstupů. Některé kompaktní systémy se navíc vyznačují ještě vnitřní modulárností, kdy konfiguraci základního modulu lze sestavit osazením základní desky zásuvnými moduly vhodného typu, tzv. „piggybacky“. Piggyback bývá řešen jako plošný spoj s jedním nebo několika integrovanými obvody a konektorem.

### ***Modulární PLC***

Nesrovnatelně větší volnost ve volbě konfigurace poskytují modulární programovatelné automaty. Do základního rámu lze zasouvat libovolné moduly i rozšiřovací moduly

k připojení dalších rámců i stovky metrů vzdálených. Tak lze vytvářet různě strukturované distribuované systémy.

### ***Programovatelné pracovní stanice***

V nedávné době představila řada výrobců své kompaktní výrobky, které v sobě sdružují funkce PLC a operátorského panelu. Integrace funkcí a praktické konstrukční provedení a výhodný poměr cena/výkon poskytuje těmto systémům široké možnosti uplatnění – a to i tam, kde by bylo použití tradičního PLC s odděleným operátorským panelem dosud cenově nedostupné. Je otázkou diskuse, zda tyto výrobky zařadit mezi PLC, mezi operátorské panely nebo zda tvoří samostatnou kategorii. Pro odlišení zavádějí někteří výrobci označení „pracovní stanice“, někdy s přívlastkem „inteligentní“ nebo „programovatelná“.

[2]

### 3 RODINA AUTOMATŮ SAIA PCD

Pro aplikaci v laboratoři je použit programovatelný automat SAIA PCD2 švýcarské firmy SAIA-Burgess Electronics Ltd.

Modelová řada automatů SAIA PCD (Process Control Device) má společnou základní architekturu, stejnou instrukční sadu, shodný přístup k prostředkům a periferiím. Program vytvořený pro jednu verzi PCD poběží bez úprav i na všech ostatních (samozřejmě s přihlédnutím k dostupnosti prostředků, např. počtu procesorů). Jednotlivé modely se liší velikostí a stupněm modularity:

- PCD1 je kompaktní přístroj se čtyřmi pozicemi pro instalaci vstupně výstupních modulů v jedné rovině se základní deskou. V základním provedení je osazen jedním procesorem, 17 kB paměti rozšiřitelné až na 140 kB, může obsahovat 1 až 2 sériové kanály. Je úspornou verzí jinak velmi podobného modelu PCD2.
- PCD2 obdobně jako PCD1 je plochý, kompaktní, s možností osazení osmi modulů, čtyři na horním i spodním okraji základní desky. Paměť pro jediný procesor je 32 – 536 kB velká, má 1 – 4 sériová rozhraní.
- PCD4 je již v pravém smyslu modulární. Je tvořen bloky montovanými na DIN lištu. Má 1 až 2 procesory, paměť 64 – 428 kB, 1 – 45 sériových portů.
- PCD6 je tvořen moduly ve „vanách“ vkládaných do 19“ skříní. 1 až 7 procesorů má k dispozici od 256 kB do 1 MB paměti, 4 až 28 sériových kanálů.)
- Všechny modely shodně používají procesor Motorola 68340, instrukční soubor obsahuje přes 120 instrukcí včetně obsluhy komunikací a aritmetiky s plovoucí desetinnou čárkou, což je významným kritériem při realizaci číslicových regulátorů. Provedení PCD2 bylo vybráno vzhledem ke své vybavenosti odpovídající rozsahu aplikace.

## 4 PROGRAMOVATELNÝ AUTOMAT SAIA PCD2

SAIA PCD2 je programovatelný automat kompaktní konstrukce. Základní jednotka obsahuje hlavní desku se všemi aktivními prvky, za ní je sběrnicová deska s konektory pro připojení vstupně – výstupních modulů – čtyři na horním okraji a čtyři na dolním. Celek je uzavřen v ploché skřínce, jež umožňuje přístup jen ke konektoru PGU (Programming Unit) pro připojení programovacího zařízení a svorkovnicím jednotlivých modulů na stranách přístroje. Světlovody přivádějí od LED diod na plošných spojích na povrch optickou indikaci stavu základní jednotky a jednotlivých vstupů a výstupů (Obr. 19).

Po sejmutí krytu se zpřístupní svorkovnice pro napájení přístroje 24V= $\bar{}$ , komunikační rozhraní a kontakty relé watchdogu. Nyní lze na kontakty sběrnicové desky zasunout I/O moduly. Na hlavní desce jsou také dvě pozice k instalaci rozšiřujících piggybacků s komunikačními a zobrazovacími obvody a patice pro rozšíření paměti (RAM i EPROM). Při vypnutí napájení jsou paměti zálohovány knoflíkovým článkem (Obr. 20).

Datová komunikace mezi hlavní deskou a přídavnými moduly probíhá pomocí adresovaných kanálů. Každý I/O modul má vyhrazeno 16 adres podle pozice, na níž je instalován. Adresace začíná v levém horním rohu sběrnicové desky od bázové adresy 0 a pokračuje ve směru hodinových ručiček. Pro každý modul je bázová adresa zvyšována vždy o 16. Toto pravidlo platí vždy, nezáleží na tom, zda modul využívá 16, 8, 6 nebo jen 4 adresy. V případě potřeby připojení více než osmi modulů lze použít rozšiřující jednotku, která obsahuje sběrnici s dalšími osmi pozicemi s bázovými adresami 128 – 240.

Pozn.: Protože obvodu watchdog na hlavní desce je interně přiřazena adresa 255. nesmí být na adrese 240 instalován modul, který používá všech 16 adres.

### 4.1 Dodávané typy vstupně-výstupních modulů

- Komunikační rozhraní RS 422, RS 485, RS 232, proudová smyčka 20mA (jako piggyback)
- Hodiny reálného času + displej (piggyback)
- Dvouhodnotové vstupní a výstupní moduly – 4 až 8 vstupů nebo výstupů, 24V= $\bar{}$  nebo 250V $\sim$ , proud až 2A. Galvanicky neoddělené i optočleny oddělené vstupy, výstupy a tranzistory MOSFET nebo relé.

- Analogové vstupní a výstupní moduly se 4 až 8 kanály s rozlišením A/D, D/A převodníků 8 až 12 bitů pro signály 0 – 10V, -10 – 0V, -10 – 10V, 0 – 20mA, -20 – 0mA, -20 – 20mA, 4 – 20mA a pro přímé připojení teploměrů Pt, Ni 100, 1000.
- Čítací polohovací moduly
- Zobrazovací moduly a terminály

[3]

## 4.2 Typy datových prostředků využitelných v programech

Data a hodnoty, které program používá jsou striktně rozděleny do datových typů. Do paměti nelze přistupovat libovolně, ale vždy jen k prvku určitého typu a pořadového čísla (adresy) v rámci typu. Tabulka 1 obsahuje přehled použitelných typů dat.

Tab. 1. Přehled datových typů programovatelného automatu SAIA

Kód typu	Typ	Popis	Rozsah adres
I	Vstup (Input)	Hodnota fyzického dvouhodnotového vstupu modulu (adresa podle umístění modulu)	0..8191
O	Výstup (Output)	Slouží k nastavení nebo přečtení stavu fyzického dvouhodnotového výstupu I/O modulu (adresa podle umístění modulu). I a O sdílejí stejné adresy.	0..8191
F	Flag	Binární proměnná	0..8191
T	Časovač (Timer)	Proměnná k nastavení nebo čtení stavu časovače	0..450
C	Čítač (Counter)	Proměnná k nastavení nebo čtení stavu čítače. T a C sdílejí adresy, je možno si zvolit, co bude čítač a co časovač.	0..1599
R	Registr (Register)	Obecný číselný registr (32 bitů). Uložená hodnota může být celé nebo reálné číslo.	0..4095
K	Konstanta (K constant)	Podle přítomnosti desetinné tečky se rozlišují celá a reálná čísla.	
X	Text	Slouží k ukládání řetězců do paměti	0..7999

Kód typu	Typ	Popis	Rozsah adres
DB	Datový Blok (Data Block)	Slouží k ukládání bloků dat do paměti. Výhodné pro větší množství dat, požívají se speciální instrukce. X a DB sdílejí stejné adresy, je nutno zvolit zda půjde o řetězec nebo data.	0..7999

Uvedený rozsah adres platí vždy jen v rámci typu (kromě výjimek sdílení adres). Např.: Flag F15 je uložen jinde než registr R15. V programech lze samozřejmě využívat možnosti přiřazení symbolických jmen (např. direktivou EQU v Instruction List ). [4]

### 4.3 Struktura uložení programu

Podobně jako k datům a proměnným lze přistupovat pouze systematicky pomocí adres v rámci datového typu, program se musí zapisovat formou bloků různých typů podle určení. Tabulka 2 obsahuje přehled použitelných bloků.

Tab. 2. Přehled typů programových bloků

Kód Typu	Typ bloku	Popis	Rozsah číslování
COB	Cyklický organizační blok (Cyclic Organization block)	Části programu, umístěné v těchto blocích, jsou cyklicky prováděny (postupně se vykonává COB0, COB1, až po COB15 a znova od COB0).	0..15
XOB	Blok výjimky (Exception Organization Block)	K jednorázovému spuštění bloků tohoto typu dochází při výjimečných událostech přiřazených každému číslu XOB. Např. XOB0 se provede při výpadku napájení atd.	0..31
PB	Programový Blok (Program Block)	Blok programu (procedury). Volá se z jiného bloku a po provedení se vrací zpět.	0..299
FB	Funkční Blok (Function Block)	Funkce – použití stejné jako u PB, při volání lze navíc předávat parametry.	0..999
ST	Krok (Step)	Blok obsahující krok – dílčí část sekvence	0..1999

Kód Typu	Typ bloku	Popis	Rozsah číslování
SB	Sekvenční blok (Sequential Block)	Blok pro programování sekvenčních algoritmů (Graftec), obsahuje IST, ST a TR.	0..31
IST	Úvodní krok (Initial Step)	Provádí se na začátku sekvence definované SB.	0..1999
TR	Přechod (Transition)	Blok obsahující podmínky, na jejichž základě se rozhoduje o přechodu na další krok.	0..1999

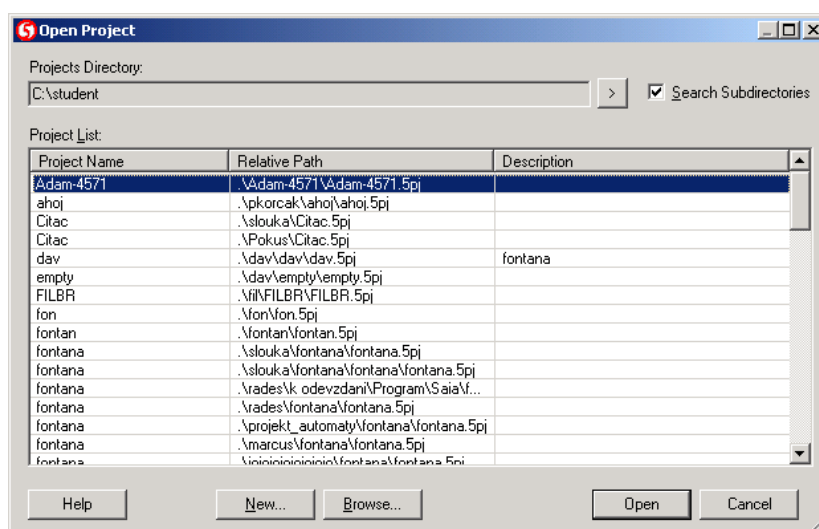
V programu se nesmějí vyskytovat bloky stejného čísla v rámci typu. [5]

## 5 VÝVOJOVÉ PROSTŘEDÍ PG4

K tvorbě programů pro rodinu programovatelných automatů SAIA PCD výrobce dodává vývojové prostředí PG4 běžící na PC s operačním systémem MS Windows 3.x, 95, NT. Jedná se o ucelený balík nástrojů pro tvorbu jednotlivých rutin v různých programovacích jazycích, jejich kompilování, správu knihovny takto vzniklých objektů, linkování programů a správu celých projektů.

### *Knihovna projektů*

Po spuštění vývojového prostředí PG4 se objeví okno knihovny projektů (SAIA Projects Library). Toto je nejvyšší úroveň, zde se k další práci vybírá jeden ze samostatných již vytvořených projektů nebo se zakládá nový.



Obr. 2. Okno knihovny projektů

Po zvolení jednoho z projektů s ním lze pracovat ve Správci projektu (SAIA Project Manager).

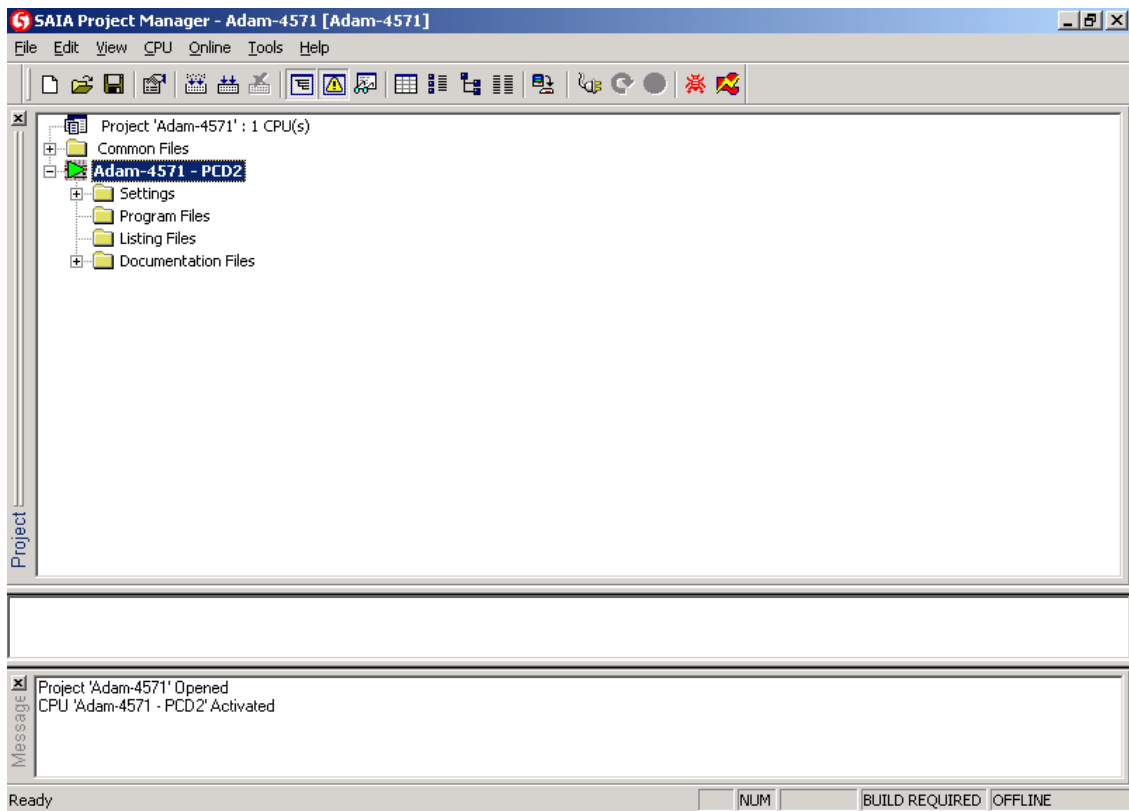
### *Správce projektu*

Okno správce projektu obsahuje seznam programových modulů obsažených v tomto projektu, z nichž se sestavuje výsledný program. V případě potřeby lze samozřejmě určitý modul ze sestavování vyjmout, aniž by se musel zcela odstranit z projektu.

Správce je jádrem PG4. Odtud lze linkovat program, připojit se on-line k PCD, nahrát do něj program a následně jej spustit, krokovat. K dispozici jsou také nástroje ke konfigurování paměťových zdrojů PCD – rozvržení prostoru RAM pro dynamické proměnné, čítače, časovače, uživatelské texty a datové bloky. Online Configurator umožňuje přímo nastavovat systémové prostředky logického automatu – vnitřní obvod

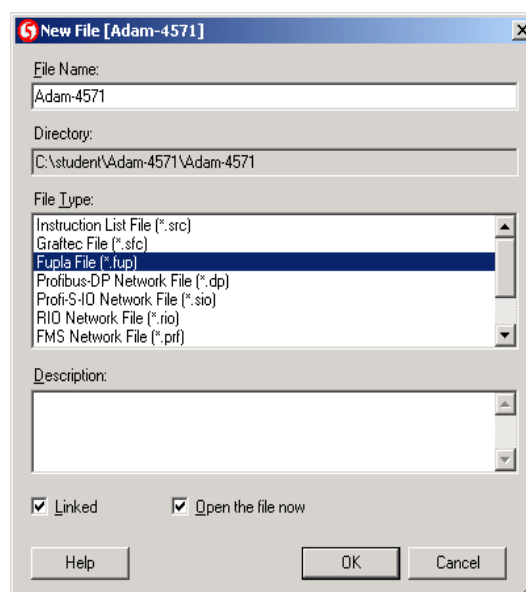


reálného času, parametry komunikačních rozhraní, případně zablokovat další přístup k paměti PCD heslem (pro zamezení neoprávněného kopírování či modifikace).



Obr. 3. Okno správce projektu

Při vytváření nového objektu se volí programovací jazyk, v němž bude program vytvořen a tím i editor, který pro něj bude používat.



Obr. 4. Výběr typu programovaného objektu

- **IL (AWL)** – Instruction List (assembler), editor AWL (textový editor)
- **FBD/LD** – Function Block Diagram/Ladder Diagram (schéma funkčních bloků), editor FUPLA (Function Plan)
- **SFC** – Graftec grafický jazyk pro programování sekvencí bloků
- **OBJ** – předem zkompileovaný objekt k zahrnutí do výsledného programu
- **RIO** – Remote IO Network
- **FMS** – Profibus FMS Network – objekty připojené ze vzdáleného zdroje v síti

### 5.1 Instruction List

Instruction List (výpis instrukcí) je forma zápisu programu přímo v základních instrukcích. K zápisu slouží obyčejný textový editor, řádky se nečíslují, komentáře se uvozují středníkem.

Bloky se specifikují úvodními s koncovými instrukcemi podle typu bloku.

Příklad:

```
XOB 16          ; Studený start – začátek bloku
...
EXOB           ; Konec bloku
```

K přiřazení symbolického označení slouží direktiva EQU.

Příklad: Registr R14 se nadále může vyjadřovat symbolem „Teplota“.

```
Teplota      EQU R 14
```

Aby byl definovaný symbol i se svým významem použitelný v jiných objektech (souborech) z nichž se sestavuje výsledný program, musí se zveřejnit direktivou PUBL.

Příklad:

```
Teplota      EQU R 14
              PUBL Teplota
```

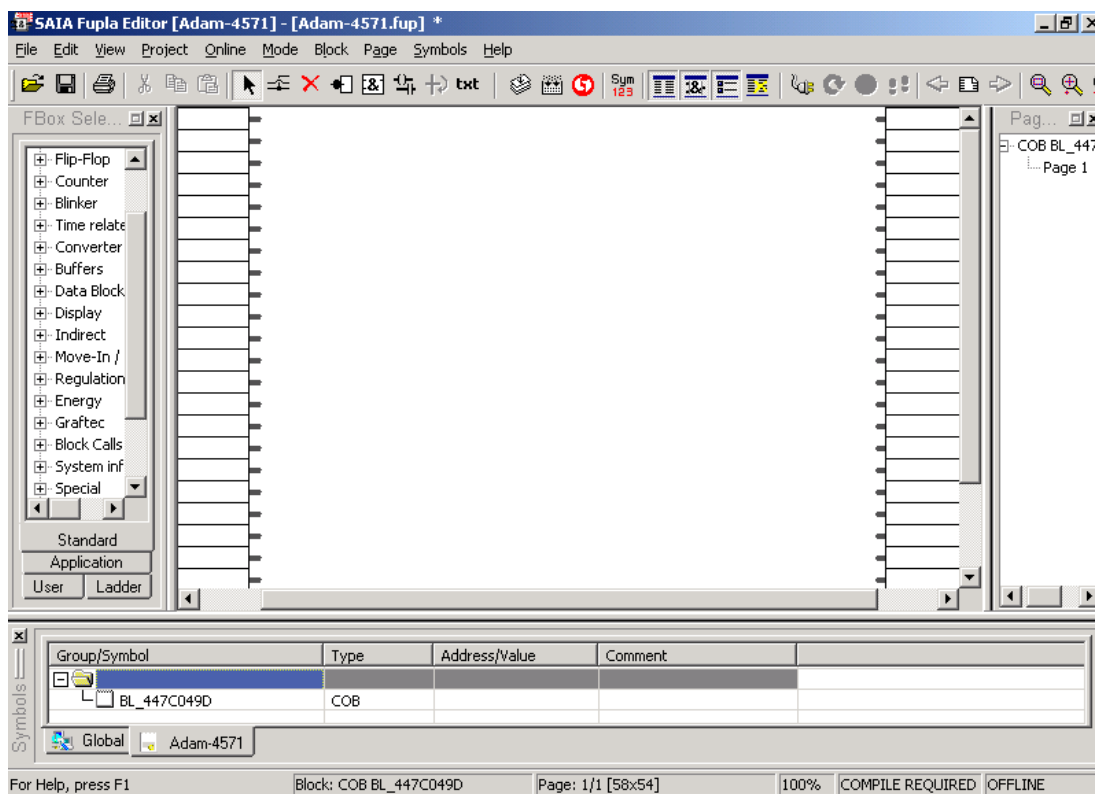
Aby mohl být v aktuálním kódu použit symbol definovaný a zveřejněný v jiném objektu, zavede se direktivou EXTN jako externí.

Příklad:

EXTN Teplota

## 5.2 Function Plan – FUPLA (FBD)

FUPLA (Function Plan) je grafický jazyk založený na makrech s výhodou dynamického přiřazování datových prostředků.



Obr. 5. Okno editoru FUPLA

Do „sběrnice“ na levém okraji strany se vepisují vstupní data bloku – konstanty, registry, flagy, vstupy, výstupy, čítače, časovače apod. a to buď absolutně, tj. kódem datového typu a číslem, nebo symbolicky. Podobně pravá „sběrnice“ obsahuje prostředky přijímající výstupní hodnoty.

Seznam všech použitých datových prostředků a jim přiřazených symbolů lze vyvolat výběrem *Resource -> Resource List..* z hlavního menu. Nebyl-li registr, flag, časovač nebo čítač definován jako externí a nespecifikuje-li se jeho číslo, bude mu přiřazeno dynamicky v okamžiku sestavování programu. Oblasti adres, ze kterých může překladač v tomto případě čerpat, se nastaví vyvoláním *Resource->Dynamic Distribution..* z menu Správce projektu (Project Manager). Datové prostředky z takto rezervovaných oblastí by neměly být použity absolutní adresací ve zbytku programu.

Program je sestaven z funkcí, reprezentovaných obdélníky s popisem, vstupy vlevo a výstupy vpravo (FBox). Předávání dat mezi nimi definují spojnice. Funkce se provádějí postupně z levého horního rohu dolů, při zachování logické návaznosti. Jsou definovány v knihovnách, dodávaných s vývojovým prostředím, v nichž je uložen jim odpovídající kód ve formě výpisu instrukcí a symbolickým vyjádřením vstupujících a vystupujících dat. Při překládání programu se použije tento kód jako makro a symboly se nahradí konkrétními čísly datových prostředků. Opět se čerpá z oblastí rezervovaných pro dynamické přiřazování.

Striktně se rozlišují číselné typy hodnot – binární, celočíselné a s plovoucí desetinnou čárkou – a to i barvou spojnic. Např. celočíselný výstup nelze přivést na vstup očekávající reálné číslo, musí se použít konverzní funkce.

Každý soubor v jazyce FUPLA může obsahovat více bloků. K vybrání jednoho z nich ke zobrazení v editačním okně slouží seznam bloků, vyvolaný z menu *FILE->Block List*.

Volba *Properties* umožňuje nastavit typ (COB, XOB, PB ...) a číslo bloku, *NEW..* zakládá nový blok.

Každý blok může mít rozsah více stran. Listování mezi nimi, přidávání nových a jejich organizaci umožňují funkce z nabídky *Page*.

Jazyk FUPLA je přehledný, programování v něm je jednoduché, ale některé nedostatky přinášejí omezení:

- Nelze použít skoky v programu, podmíněné ani nepodmíněné, což zcela znemožňuje realizaci cyklu
- Ze stejného důvodu lze podmíněně jen volat podprogramy, i jednoduchý program se tak může stát nepřehledným
- Při volání funkčních bloků (FB) jim nelze předat parametry, což je degraduje na procedury (PB)

Jako vyšší jazyk přirozeně FUPLA oproti *Instruction List* vytváří redundantní kód, to je však samozřejmá cena za používání knihoven funkcí k univerzálnímu použití.

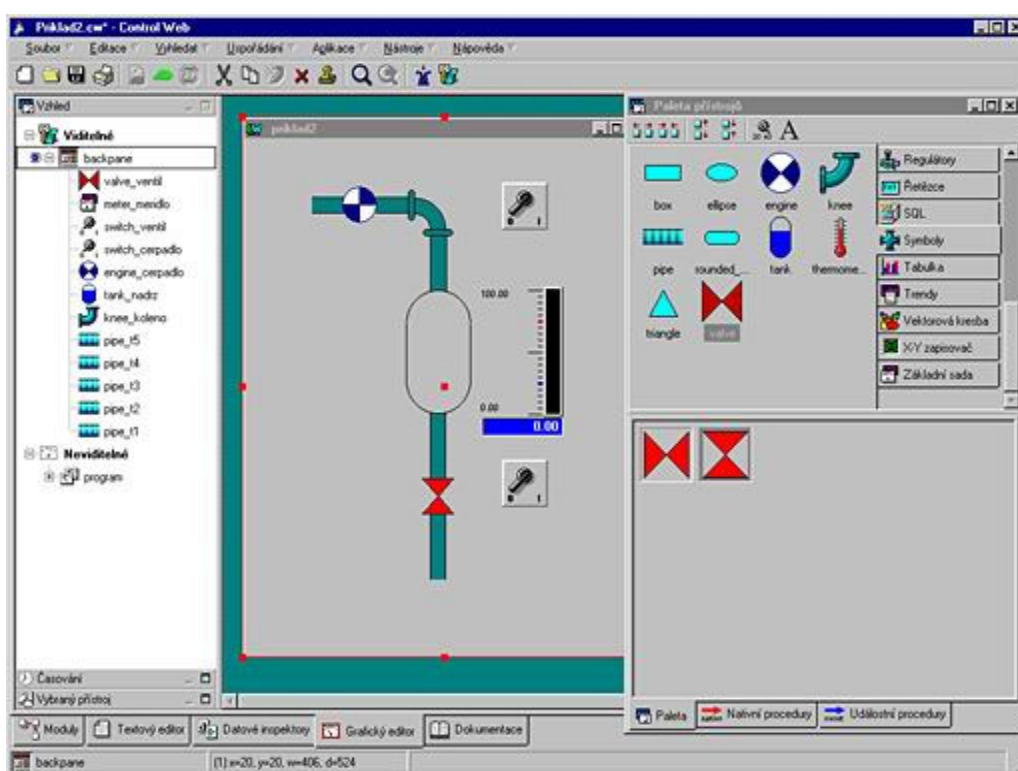
### 5.3 Graftec

Graftec je grafický jazyk pro zápis sekvenčních algoritmů. Vývojový diagram určuje postup provádění dílčích kroků a podmíněných přechodů mezi nimi. Procedury kroků (ST) i vyhodnocování přechodů (TR) jsou napsány buď v jazyce IL nebo FUPLA.

Po provedení kroku program stojí, dokud není splněna podmínka ukončení přechodu na následující krok. Z tohoto důvodu je v případě realizace číslicových regulátorů s potřebou současného provádění více procesů Graftec nevhodný. [5]

## 6 CONTROL WEB (CW)

Control Web je systém rychlého vývoje aplikací pro vizualizaci a řízení technologických procesů v reálném čase, umožňující také přímé řízení strojů a technologií nebo použití jako rozhraní člověk – stroj. Control Web je určen pro běh pod 32bitovými operačními systémy Windows na počítačích standardu PC. Požadavky systému Control Web na počítač nejsou nijak výjimečné a lze říci, že pokud na daném počítači uspokojivě pracuje operační systém a běžné kancelářské aplikace, bude na něm bez problémů pracovat i systém Control Web. Je samozřejmé, že s rostoucí aplikací rostou i nároky na počítač.



Obr. 6. Okno programu Control Web

### 6.1 Vývoj aplikací v prostředí CW

Tvorba kompletní aplikace zahrnuje kompletní tvorbu algoritmu aplikace. Tvůrce aplikace se zabývá jednak otázkami „jak ukládat a organizovat data“ a také „jakými algoritmy data zpracovat“. Vyšší pracnost a také vyšší nároky na znalosti tvůrců aplikací se pak vrací v podobě větší obecnosti a možnosti mít zcela pod kontrolou práci aplikace. Ne vždy je ale taková obecnost potřebná.

Prostředky rychlého vývoje aplikací (RAD – Rapid Application Development) jako je Control Web umožňují jednodušší vývoj aplikace. Tvůrce aplikace se zabývá jen otázkou

jak organizovat a ukládat data a algoritmy za něj řeší systém sám. Vývoj aplikace se tedy děje skládáním a parametrizováním komponent.

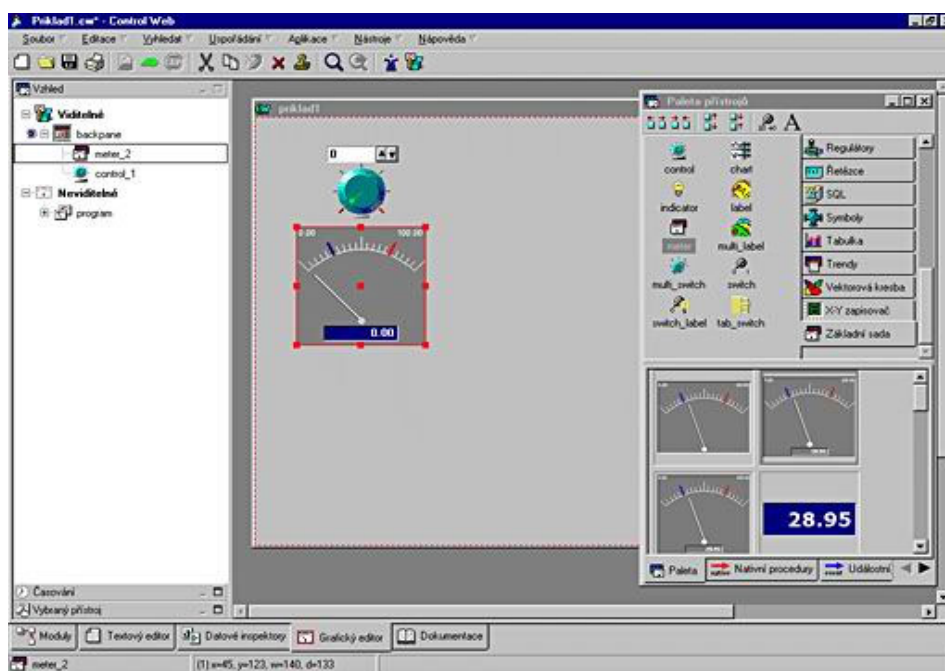
## 6.2 Filozofie tvorby aplikace v prostředí CW

K tvorbě aplikace v prostředí Control Web není třeba znalost programování, pro náročnější požadavky na aplikace je ovšem k dispozici programovací jazyk.

Tvorba aplikace:

### *Přetažení přístrojů na pracovní plochu*

Základním stavebním kamenem aplikací Control Web jsou tzv. Přístroje. Přístroj je komponenta, ze které se skládá aplikace, např. zobrazovací a ovládací prvky, alarmy atd. Hlavní výhodou aplikace CW je velké množství již hotových komponent, které v sobě zahrnují funkčnost potřebnou pro běh našich aplikací.



Obr. 7. Vložení přístroje meter z palety přístrojů

### *Definice proměnných - datové elementy.*

Datové elementy uchovávají hodnoty veličin. Může se jednat o veličiny přečtené z technologie, veličiny řídící běh aplikace a s technologií přímo nesouvisející (např. logické příznaky určující, která obrazovka je momentálně zobrazena v operátorském panelu).

### *Nastavení způsobu aktivace přístrojů.*

Všechny virtuální přístroje provádějí svou činnost v rámci své aktivace. Pokud například u přístroje *meter* (měřák) definujete výraz, který bude vyčíslen a výsledek přiřazen do nějakého datového elementu, toto vyčíslení, přiřazení se děje právě v rámci aktivace. Nebo přístroj *alarm* v rámci své aktivace testuje překročení mezních hodnot a přístroj *archiver* zapisuje data apod. Základní způsob aktivace u aplikací reálného času je periodická aktivace. Například každou sekundu chcete aktualizovat zobrazenou teplotu, nebo každou sekundu chcete testovat jestli nedošlo k překročení mezí. Někdy je ale zbytečné aktivovat přístroj periodicky když má reagovat jen na konkrétní událost. Proto je možno aktivovat přístroj, když je aktivován jiný přístroj. Např. pokud tlačítko (přístroj *switch*) nastavuje svou výstupní hodnotu, můžete uvést seznam jiných přístrojů, které budou po změně této hodnoty aktivovány (přístroj *indicator*).

### *Propojení s technologickým procesem (zápis hodnot do kanálů)*

Abychom mohli řídit nějakou skutečnou aplikaci, musí existovat propojení mezi CW a hardware. Ke komunikaci s HW zařízením slouží ovladače. Ovladač slouží v systému Control Web pro komunikaci s periferním zařízením. Pro každé konkrétní zařízení je nutné použít odpovídající ovladač. Daný ovladač se vloží do aplikace, nakonfiguruje a potom je již možné vkládat do aplikace kanál, které prostřednictvím ovladače komunikují s daným periferním zařízením. Některé ovladače jsou součástí instalace systému *Control Web*, jiné je potřeba před použitím nainstalovat.

System Control Web je navrhován jako systém nezávislý na hardware. S patřičným ovladačem komunikuje s jakýmkoli průmyslovým zařízením:

- PLC (Siemens, Mitsubishi, Omron, Teco, Allen – Bradley, SAIA..)
- I/O moduly (DataLab IO, ELSACO, ADAM.. )
- Měřicí karty (Advantech, Axiom, Tedia..)
- „virtuální“ zařízení, např. WWW server apod.

Architektura ovladačů je otevřená a dokumentovaná, každý může implementovat vlastní ovladač.

System *Control Web* je vybaven několika ovladači, které jsou určeny jak pro ladění a testování aplikací, tak pro nasazení v reálném technologickém procesu.



***Standardně dodávané ovladače:***

Pro demonstrační a ladicí účely slouží:

Virtuální Ovladač - není spojen se žádným konkrétním fyzickým zařízením. Slouží ke generování různých signálů.

Modelový ovladač - slouží jako náhrada reálné soustavy při testování správné funkce řídicího programu, zvláště pak regulačních obvodů.

Simulační ovladač (Dummy) - nahradí reálný ovladač pro testování a ladění aplikace.

Simulační ovladač (SimBuf) - k testování a ladění aplikací, má nadefinováno několik kanálů, které generují průběhy funkce sinus. Tyto průběhy lze různě ovlivňovat.

Pro běžné použití v reálných aplikacích:

Ovladač DDE Client systému Control Web - pracuje jako DDE klient a slouží k propojení aplikace s libovolným DDE serverem.

Ovladač ASCDRV5 pro komunikaci přes standardní sériové rozhraní počítače - slouží pro ASCII komunikaci přes sériové rozhraní počítače

TCP/IP ovladač - CWNETDRV - pro spojení několika aplikací *Control Web* na různých počítačích v rámci sítě.

***Nasazení aplikace***

Control Web existuje ve 2 verzích: vývojové a runtime.

Vývojová verze systému *Control Web* slouží k tvorbě a testování aplikací. Ačkoliv aplikace může zcela plnohodnotně pracovat i v prostředí vývojové verze, typicky je ale před nasazením přeložena do podoby '.cwx' souboru, který je určen pro Runtime verzi systému. Aplikaci v podobě CWX již není možno modifikovat, změny je možné provádět jen ve vývojové verzi v souborech '.cw'. Runtime verze vyžaduje aplikaci ve formátu '.cwx', připravenou ve vývojové verzi. Tento formát není textový, ale obsahuje přeloženou podobu aplikace. Aplikaci běžící v runtime verzi tak není možné žádným způsobem modifikovat ani získat její zdrojovou podobu. Tím jsou chráněny investice aplikačních firem do vývoje aplikací.

### 6.3 Podporované protokoly a standardy

#### *Podpora otevřených protokolů*

- ASCII komunikace po sériové lince - Znakový protokol využívá velké množství jednoduchých zařízení.
- OPC Data Access - stále vzrůstá množství OPC serverů
- DDE / NetDDE, FastDDE - zachování zpětné kompatibility s DDE servery
- GSM modemy, SMS zprávy
- HTTP přístup k WWW serverům
- Modicon Modbus, Modbus/TCP

#### *Podpora otevřených standardů*

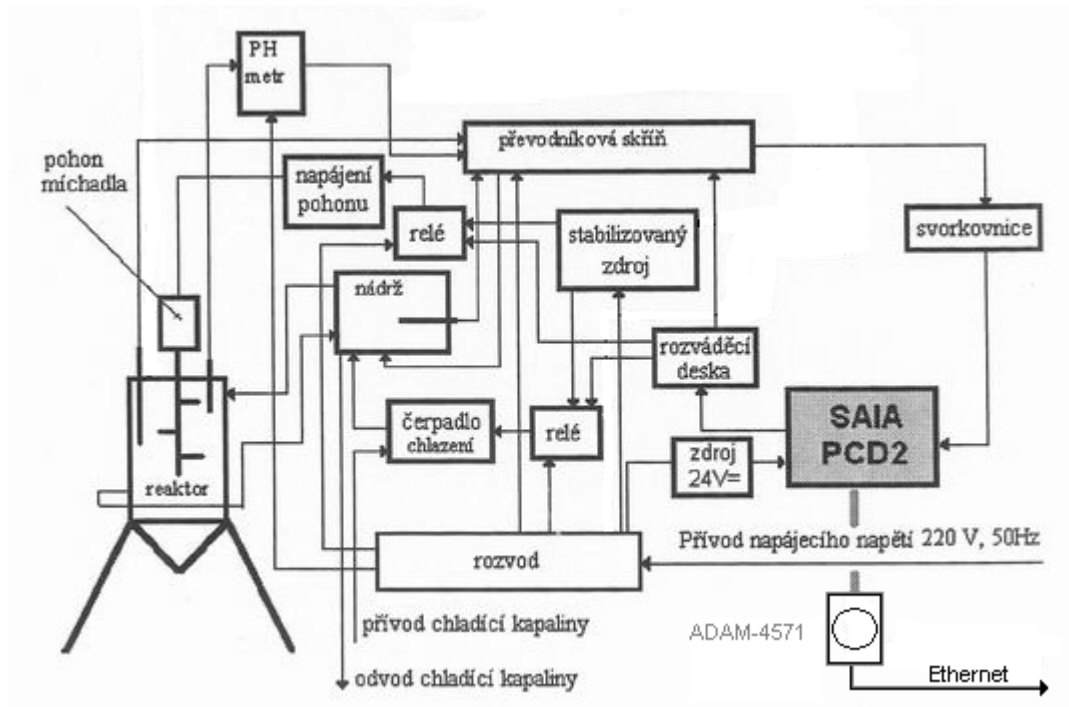
- Široká interoperabilita díky podpoře standardních protokolů a formátů dat
- TCP/IP, HTTP, HTML (Ethernet, WiFi, dial-up, ...)
- ODBC / SQL
- COM / ActiveX
- OPC (OLE for Process Control)
- GSM / GPRS
- DDE, NetDDE

[6]

## **PRAKTICKÁ ČÁST**

## 7 HARWAROVÁ ČÁST

Tato část práce se bude zabývat návrhem a realizací zapojení pro komunikaci mezi PLC a programem běžícím na nadřazeném počítači v prostředí Control Web pomocí sítě Ethernet, konkrétně pro pracoviště fermentor v laboratoři recyklačních technologií.



Obr. 8. Celkové schéma pracoviště fermentor

### 7.1 Konfigurace PCD2 pro aplikaci řízení pracoviště fermentor

Řízení technologického procesu na pracovišti fermentace klade tyto nároky na vstupní a výstupní kanály programovatelného automatu:

- měření teploty v reaktoru analogový vstup 0 – 10 V
- měření teploty v nádrži temperační jednotky analogový vstup 0 – 10 V
- měření pH analogový vstup 0 – 10 V
- spínání topení v nádrži TTL výstup
- otevírání solenoidového ventilu chlazení TTL výstup
- řízení otáček míchadla analogový výstup 0 – 10 V

Tyto požadavky na I/O kanály pokrývá následující výběr modulů.

### 7.1.1 Analogový vstupní modul PCD2.W200

Technické parametry:

- 8 kanálů pro signály 0 – 10 V
- bez galvanického oddělení
- číselná reprezentace (rozlišení): 10 bitů (0 - 1023)
- doba A/D převodu < 50  $\mu$ s
- nepřesnost  $\pm 1$  bit
- vstupní odpor: 200 k $\Omega$  / 0,15 %

Kontakty 0 – 7 svorkovnice modulu jsou kladné póly analogových vstupů E0 – E7, kontakt 9 je společná zem.

[3]

Pro řízení procesu fermentace je instalován jeden modul W200 s osmi vstupy (na 3. pozici, bázová adresa 32).

### 7.1.2 Dvouhodnotový výstupní modul PCD2.A400

Technické parametry:

- 8 MOSFET tranzistorových výstupů
- bez galvanického oddělení
- výstupní proud 5..500 mA bez ochrany proti zkratu
- pracovní režim: spíná kladné napětí na výstup 5 – 32 V=
- úbytek napětí  $\leq 0,5$  V při 0,5 A
- zpoždění výstupu: při spínání 10  $\mu$ s, při rozpínání 50  $\mu$ s

Kontakty 0 – 7 jsou svorkovnice modulu jsou kladné póly výstupního signálu, kontakt 9 je společná zem a na svorku 8 se přivádí napájecí napětí.

[3]

Modul A400 je instalován na 1. pozici, bázová adresa 0.

### 7.1.3 Analogový výstupní modul PCD.W400

Technické parametry:

- 4 kanály s ochranou proti zkratu
- výstupní rozsah 0 – 10 V
- číselná reprezentace (rozlišení): 8 bitů (0 - 255)
- doba převodu D/A převodu < 5  $\mu$ s
- zatěžovací odpor  $\geq 3 \text{ k}\Omega$
- nepřesnost 1 %  $\pm 50 \text{ mV}$

Kontakty 0, 2, 4, 6 jsou kladné póly analogových výstupů A0 – A3.

[3]

Modul W400 je instalován na 2. pozici, bázová adresa 16.

### 7.1.4 Vstupní a výstupní signály PCD2

Vstupní a výstupní signály jsou přivedeny na jednotlivé moduly takto:

Tab. 3. Vstupní a výstupní signály PCD2

Signál	Modul	Kontakt na svorkovnici modulu
Teplota v reaktoru	PCD2.W200 (3. pozice, bázová adresa 32)	0
Teplota v nádrži	PCD2.W200 (3. pozice, bázová adresa 32)	1
pH v reaktoru	PCD2.W200 (3. pozice, bázová adresa 32)	2
Spínání topení v nádrži	PCD2.A200 (1. pozice, bázová adresa 0)	0
Spínání chlazení	PCD2.A200 (1. pozice, bázová adresa 0)	1
Pohon míchadla	PCD2.W400 (2. pozice, bázová adresa 16)	0

## 7.2 Připojení PCD2 k PC pomocí Ethernetu

### 7.2.1 Konektor PGU jako obecné sériové rozhraní RS232

Konektor PGU (Programming Unit) slouží k připojení programovacího zařízení, případně nadřazeného pracoviště, v tomto případě PC s vývojovým prostředím a programem Control Web. Toto rozhraní je možné použít obecně jako sériové rozhraní kanálu č.0 pro připojení jiného periferního zařízení.

Pro začlenění PCD2 do sítě Ethernet je použit modul konvertoru Ethernet na RS 232/422/485 ADAM-4571-A (Obr. 21).

### 7.2.2 Modul ADAM 4571-A

Modul ADAM-4571 je externí vstupně/výstupní modul, sloužící jako převodník sběrnice RS 232/482/422 na Ethernet (EDG – Ethernet Data Gateway) pro realizaci vzdáleného komunikačního portu. Disponuje jedním sériovým rozhraním, které je přes Ethernet zcela transparentně připojeno do PC jako virtuální sériový port. Zařízení připojené na modul ADAM-4571-A, mapované na virtuální porty počítače, se budou chovat stejně jako zařízení připojená přímo na standardní komunikační port.

Charakteristiky:

- Na jednom počítači vybaveném operačním systémem Windows 95/98/NT/2000/ME/XP lze namapovat až 255 portů.
- Podporuje standard 10/100 T-Base Ethernet.
- napájení 10 – 30 V=
- montáž na DIN lištu, panel, nebo na sebe (piggyback)
- Používá protokol TCP/IP
- Přenosová rychlost 300 bps - 230 Kbps
- Délka slova 5,6,7,8 bitů
- Parita (odd, even, none, space, mark)
- počet stop-bitů (1, 1,5 a 2)
- Snadná diagnostika pomocí LED diod

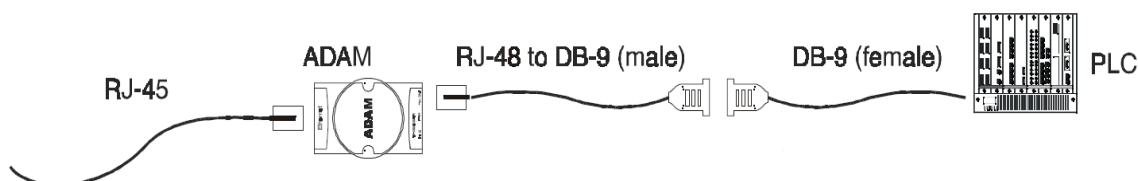
Na přední straně modulu jsou čtyři LED diody, které slouží k diagnostice stavu modulu. Tabulka 4 obsahuje přehled a význam jednotlivých diod.

Tab. 4. Význam LED diod modulu ADAM-4571

LED	Barva	Stav	Význam
Status/Power	Červená	ON	Pracuje(1x za sekundu)
		OFF	Nepracuje
	Zelená	ON	Napájení zapnuto
		OFF	Napájení vypnuto
Speed/Link	Červená	ON	Rychlost 100 Mbps
		OFF	Rychlost 10 Mbps
	Zelená	ON	Připojení k síti
		OFF	Chyba připojení
TX/RX (Ethernet)	Červená	ON	Vysílání dat (Ethernet)
		OFF	Data nebyla odeslána
	Zelená	ON	Příjem dat
		OFF	Data nebyla přijata
TX/RX (port 1)	Červená	ON	Vysílání dat (sériový port)
		OFF	Data nebyla odeslána
	Zelená	ON	Příjem dat
		OFF	Data nebyla přijata

### 7.2.3 Realizace zapojení PCD2 a modulu ADAM-4571-A

Modul ADAM-4571-A má na horní části modulu jedno sériové rozhraní vybavené konektorem typu RJ-48, na spodní části je standardní konektor RJ-45 pro připojení sítě Ethernet. Pro připojení sériového rozhraní RS232 je nutné použít kabel dodávaný spolu se zařízením, který je na jedné straně vybaven konektorem RJ-48 a na druhé konektorem DB-9. [7]



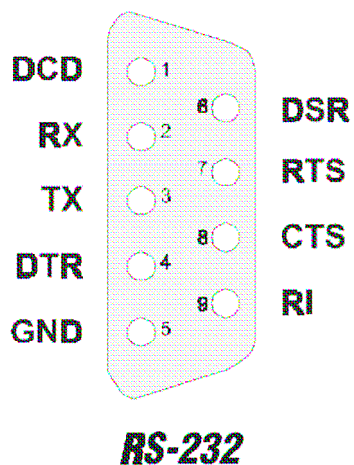
Obr. 9. Zapojení modulu ADAM-4571-A



Tabulka 5 obsahuje přehled jednotlivých pinů konektoru RJ-48 a jejich význam.

Tab. 5. Význam pinů konektoru RJ-48

Čísli PINU	Signál	Význam
1	DCD	Ochranná zem
2	RX	Příjem dat
3	TX	Vysílání dat
4	DTR	Nezapojeno
5	GND	Signálová zem
6	DSR	Připojeno PGU
7	RTS	Požadavek vysílání
8	CTS	Povolení vysílání
9	RI	Napájení programátoru



Obr. 10. Význam jednotlivých pinů konektoru DB-9

## 8 PROGRAMOVÁ REALIZACE KOMUNIKACE CW-PCD2

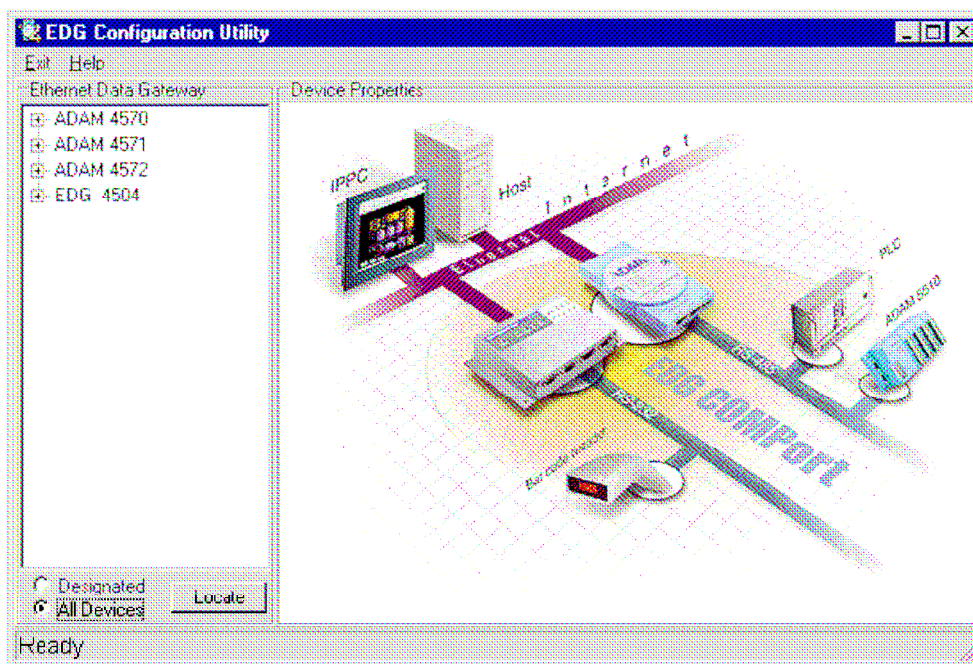
Tato část práce se zabývá programovým vybavením pro komunikaci mezi PLC a programem běžícím na nadřazeném počítači v prostředí Control Web.

### 8.1 Instalace a konfigurace modulu ADAM-4571-A

K instalaci a konfiguraci je zapotřebí programové vybavení dodávané spolu s modulem. Obsahuje dva programy, konfigurační program (EDG Configuration Utility) a program pro mapování komunikačních portů připojených modulů na virtuální porty počítače (Port Mapping Utility).

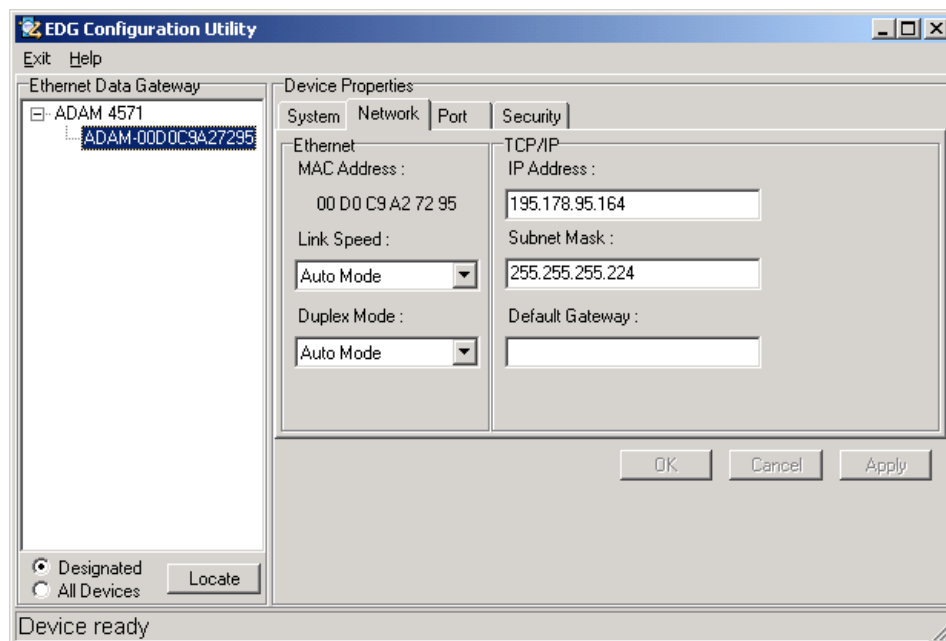
Po připojení modulu k síti Ethernet je nutné nejprve nakonfigurovat komunikační parametry pro jednotlivé porty RS-232/485/422. K tomu je určen konfigurační program.

#### *Krok 1. Konfigurační program (EDG Configuration Utility)*



Obr. 11. EDG Configuration Utility – vyhledávání modulu

Pro vyhledání modulů připojených k síti poskytuje konfigurační program funkci *Locate*. Můžeme vyhledat všechny moduly, případně zvolit skupinu modulů (např. ADAM-4570S/4571S) nebo jenom konkrétní zařízení, v našem případě ADAM-4571. Program dokáže lokalizovat pouze zařízení na stejné části místní sítě, nedokáže najít zařízení nacházející se za routrem nebo bránou. Po nalezení modulu můžeme nastavit jeho parametry.



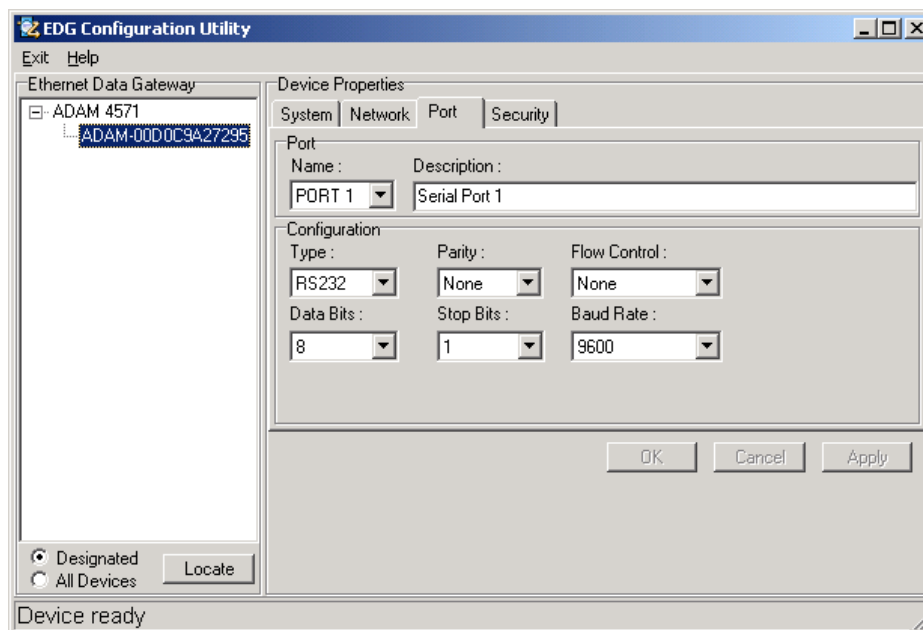
Obr. 12. EDG Configuration Utility – nastavení parametrů

V kolonce *Ethernet Data Gateway* jsou zobrazena všechna nalezená zařízení. Číslo následující za jménem modulu je tzv. MAC adresa. MAC adresa pomáhá konfiguračnímu programu identifikovat a vyhledat zařízení, tato adresa je pevně nastavena již z výroby a není třeba ji nijak nastavovat.

Kolonka *Device Properties* obsahuje čtyři záložky, které nám umožňují nakonfigurovat komunikační parametry.

- Záložka *Network* (Obr. 12) – V závislosti na nastavení sítě nastavíme IP adresu modulu a masku podsítě, hodnoty *Link Speed* a *Duplex Mode* se nastavují automaticky, je však možné je změnit.
- Záložka *Port* (Obr. 13) – Slouží k nastavení parametrů sériového portu modulu. Zde můžeme zvolit, který port chceme využívat (pokud se jedná o modul s více sériovými porty), případně přidat krátký popis v délce maximálně 128 znaků. Vybereme typ sériového rozhraní z nabídky *Type*, je zde k dispozici volba komunikačního protokolu RS-232, RS-485 nebo RS-422. Dále můžeme zvolit komunikační rychlosti (*Baud Rate*) - 300 b/s až 230 kb/s, délku slova (*Data Bits*) - 5, 6, 7, 8 bitů, použitou paritu (*Parity*) - odd, even, none, space, mark a počet stop-bitů (*Stop Bits*) - 1, 1,5 nebo 2 stop-bity.
- Záložka *System* – Můžeme pojmenovat zařízení, přidat jeho popis a nastavit heslo.

- Záložka *Security* – Zadáním IP adres je možné omezit přístup k modulu, můžeme umožnit přístup jakékoli IP adrese, nebo jej povolit jen vybraným adresám (maximálně 32 adres), případně zadat jedinou IP adresu, ze které lze měnit nastavení.



Obr. 13. EDG Configuration Utility – nastavení portu

Po nastavení parametrů lze opustit konfigurační program a spustit program pro mapování komunikačních portů připojených modulů na virtuální porty počítače.

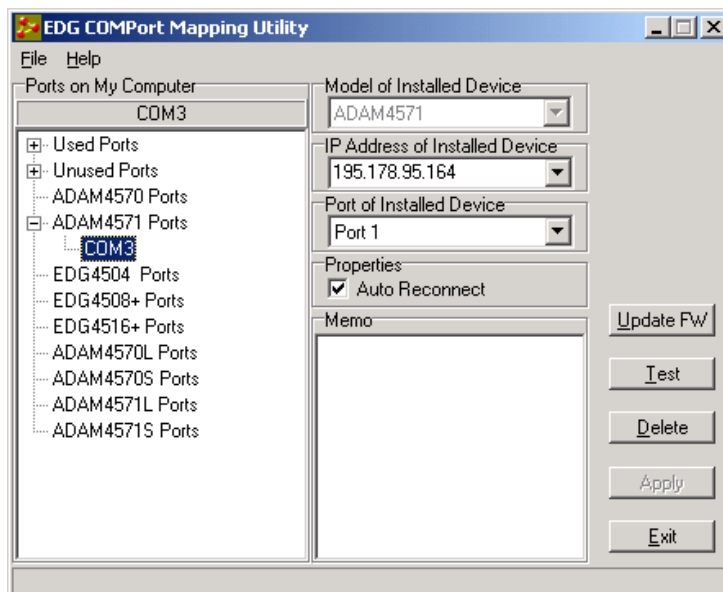
### ***Krok 2. Program mapování komunikačních portů (COMPort Mapping Utility)***

Program zobrazuje tři typy portů: využití porty, nevyužití porty a porty zařízení EDG.

Z nabídky nevyužitých portů vybereme virtuální port který chceme namapovat (v našem případě COM3), zvolíme druh modulu a zadáme IP adresu modulu, kterou jsme přiřadili v konfiguračním programu (EDG Configuration Utility) a potvrdíme tlačítkem *Add*. Mezi zařízeními se objeví vybraný modul namapovaný na námi zvolený port (Obr 14). Nyní se bude zařízení připojené na modul ADAM-4571, mapované na virtuální port COM3, chovat stejně jako by zařízení bylo připojené přímo na standardní komunikační port.

Okno programu obsahuje také tlačítka *Test* a *Upgrade FW*.

- Tlačítko *Test* spustí test signálů (RTS►CTS, DTR►RI, DTR►DSR, DTR►DCD, mezi dvěma porty) a komunikačních parametrů.
- Tlačítko *Upgrade FW* slouží k updatu firmwaru.



Obr. 14. EDG COMPort Mapping Utility

## 8.2 Program PLC pro pracoviště fermentor

Abychom byli schopní zjistit a zobrazit námi požadované hodnoty z analogových vstupů a nastavit hodnoty analogového výstupu jednotlivých modulů PCD2 v programu Control Web musíme navrhnout základní programové vybavení ve vývojovém prostředí PG4.

### *Čtení analogových vstupů*

K získání hodnot z A/D převodníku v modulu PCD2.W200 slouží v jazyce FUPLA funkce PCD2.W2 z množiny „Analog Module“ s nastavením báze adresy modulu (v našem případě je báze adresy 32). Obdélník reprezentující tuto funkci může mít až 8 výstupních kanálů, v našem případě postačí 3. Jednotlivé kanály odpovídají jednotlivým analogovým vstupům od báze adresy 32. Funkce tak v každé otočce cyklu PLC запиše do pomocných registrů hodnoty získané z A/D převodníku vstupního modulu.

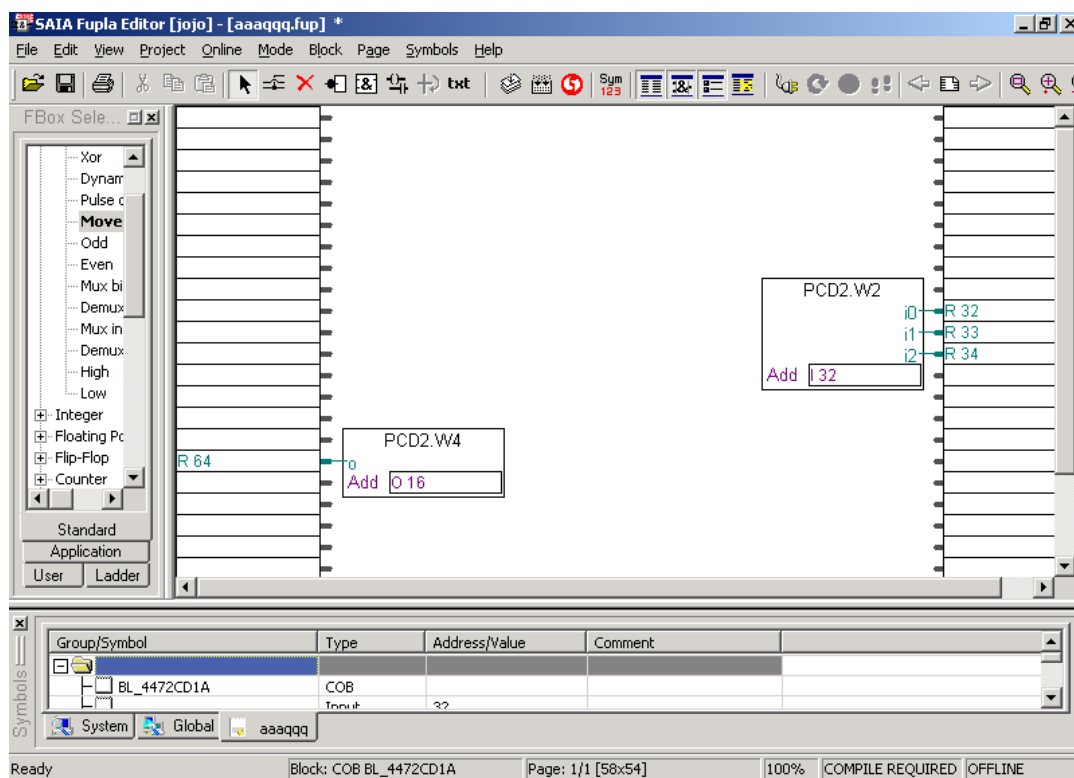
### *Zápis analogového výstupu*

K přivedení hodnoty do D/A převodníku modulu PCD.W400 opět existuje v jazyce FUPLA funkce z množiny „Analog Module“ do které запиšeme báze adresy modulu (v našem případě 16). Námi požadovanou hodnotu zadáme do pomocného registru a funkce PCD2.W4 pak v každé otočce cyklu přivede námi zadanou hodnotu do D/A převodníku modulu.

K pomocným registrům budeme přistupovat přímo z nadřazeného PC pomocí programu Control Web.

### Digitální výstupy

K ovládání digitálních výstupů není zapotřebí žádné programové vybavení v jazyce FUPLA, program Control Web dokáže zapisovat hodnoty na digitální výstupy přímo pomocí kanálů.



Obr. 15. Ovládání A/D a D/A převodníku v jazyce FUPLA

### 8.3 Ovladač pro komunikaci se stanicemi SAIA PCD pro systém CW

Ovladač SAIA slouží pro komunikaci mezi jednou nebo více stanicemi SAIA PCD a systémem Control Web. Ke své činnosti ovladač potřebuje programovou podporu firmy SAIA-Burgess. Tato podpora je zvláštní součástí instalace ovladače a instaluje se samostatně. Programovou podporu není třeba instalovat, pokud se již na počítači vyskytuje např. v rámci programovacích nástrojů PG4.

Pomocí ovladače je možno číst nebo zapisovat data z registrů (R), časovačů (T), čítačů (C), binárních výstupů (O) a příznaků (F). Je možno přečíst nebo zapsat časový údaj (K) a dále je možno přečíst hodnoty binárních vstupů (I), registr displeje (D) a stav CPU (S). Všechny tyto hodnoty se přiřadí kanálům prostřednictvím definice v souboru parametrů ovladače. Ovladač má určitou množinu kanálů vyhrazenou pro speciální informace, proto doporučuje se, aby definované kanály začínaly od čísla 100. Registry (R) stanice je možno

interpretovat různým způsobem. Jeden registr může obsahovat jednu, dvě, nebo čtyři číselné hodnoty různých typů, které se mapují do jednoho, dvou nebo čtyř kanálů (vždy typu real). Dále je možno s jedním registrem pracovat jako s jedním kanálem typu boolean nebo s 32 kanály typu boolean.

Ovladač využívá dva textové soubory, mapovací soubor (přípona .dmf) a parametrický soubor (přípona .par).

### ***Mapovací soubor***

Pomocí souboru mapujeme kanály do námi zvolených intervalů. Ovladač předává informace o stavu komunikace prostřednictvím zvláštních kanálů, které jsou mapovány do intervalu 1 až 100. Dojde-li k chybě komunikace, je vygenerována výjimka ovladače a z jednotlivých kanálů je možno zjistit různé údaje, které blíže specifikují chybu. Kanál č.1 rozlišuje, jedná-li se o chybu při čtení nebo při zápisu. Skupina kanálů č.10 až 18 doplňuje údaje o chybě při čtení resp. 20 až 28 při zápisu. Všechny numerické kanály v souboru DMF musí být typu real. Ovladač automaticky provádí nezbytné konverze na ostatní typy použité v komunikaci se stanicí.

### ***Parametrický soubor***

Pomocí souboru parametrů je možno nastavit parametry komunikace a definovat kanály ovladače, popřípadě nastavit další doplňující parametry. Celý soubor parametrů je rozdělen do několika sekcí. Název sekce je uveden v hranatých závorkách. V sekci jsou uvedeny za jménem a znakem "=" jednotlivé parametry. Na jednom řádku může být definován jeden parametr.

- Sekce [ComPort]: V této sekci se nastavují parametry komunikace. Com (určuje jméno komunikačního rozhraní), Protocol (definuje použitý komunikační protokol), Baudrate (udává komunikační rychlost), Timeout (udává dobu čekání na odpověď stanice). Hodnota pro Timeout se nastavuje v milisekundách, parametrem RTS se nastavuje způsob chování signálu RTS.
- Sekce [Channels]: V této sekci se definují bloky kanálů ovladače. Způsob zápisu parametrů je:

Block = Station, ChannelFrom, ChannelTo, Media, Offset, [Subtype], [Bidirect]

Blok je definován adresou stanice (Station), intervalem kanálů (ChannelFrom, ChannelTo), oblastí ve stanici (Media – R, T, C, F, I, O), adresou prvního elementu ve

stanici odpovídající prvnímu bloku kanálu (Offset) a pro oblast registrů R ještě datovým typem (SubType – uint, int, uint16, uint8 atd.). V případě použití obousměrných kanálů je možno definovat parametr bidirect, který zamezí vzniku kolizí při současném čtení i zápisu na stejný kanál - jsou vytvořeny dva zvláštní bloky, jeden pro čtení, druhý pro zápis.

- Sekce [Options]: Tato sekce má jediný parametr - FinalizeAfterEachBlock. Pomocí tohoto parametru se zadává, jestli budou data předána do aplikace po každé komunikaci (hodnota parametru = true) nebo až po dokončení přenosu všech bloků, které se řeší v rámci daného časového kroku. Implicitní nastavení je false.

Tabulka 6 obsahuje přehled některých datových typů ovladače.

Tab. 6. Přehled datových typů ovladače SAIA PCD

Media	Typ	Počet kanálů	Popis
R	uint	1	32-bitová hodnota bez znaménka
R	int	1	32-bitová hodnota se znaménkem
R	float	1	32-bitová real hodnota
R	uint16	2	dvě 16-bitové hodnoty bez znaménka
R	uint8	4	čtyři 8-bitové hodnoty bez znaménka
R	bool	1	jedna boolean hodnota v jednom registru
R	bits	32	32 boolean hodnot v jednom registru
F, I, O		1	boolean hodnota

Kompletní výpis jednotlivých souborů ovladače použitých v našem programu je uveden v příloze P I.

#### 8.4 Komunikace SAIA PCD2 – Control Web

Vytvořený program v prostředí CW pokrývá potřeby přímého řízení a monitorování z nadřazeného PC pro pracoviště fermentor. PCD2 nebylo připojeno přímo na pracoviště fermentor. Pro naše potřeby testování komunikace mezi PLC a programem Control Web bylo na svorky analogových vstupů přivedeno testovací napětí které bylo možné měnit, výstupní napětí bylo měřeno multimetrem a jako kontrolu stavu digitálních výstupů jsem využil LED diod na povrchu PCD2.





### 3. Konfigurace souborů ovladače

V souboru DMF si mapujeme kanály do námi zvolených intervalů. Například zápis *100 - 150 real bidirectional* nám určuje, že všechny kanály v intervalu 100 – 150 budou numerické kanály. Pro tento interval můžeme pak definovat více bloků (v souboru PAR), jeden interval nemusí odpovídat jednomu bloku.

Všechny numerické kanály musí být v aplikaci (a v souboru DMF) typu *real*. Ovladač automaticky provádí nezbytné konverze na ostatní typy použité v komunikaci se stanicemi. Parametr *bidirectional* zaručuje, že definovaný blok je určen pro čtení i pro zápis.

Soubor PAR nastavuje parametry komunikace a definuje kanály ovladače. Parametry komunikace nastavíme v závislosti na hardwaru zapojení (viz. výpis souboru PAR v Příloze P I). Definice kanálů se provádí v sekci [Channels] pomocí jednotlivých bloků kanálů.

Kanály se definují v jednotlivých blocích, přičemž hodnota počátečního kanálu v bloku odpovídá adrese prvního elementu ve stanici. Hodnota dalšího kanálu pak odpovídá ve stanici elementu na následující adrese.

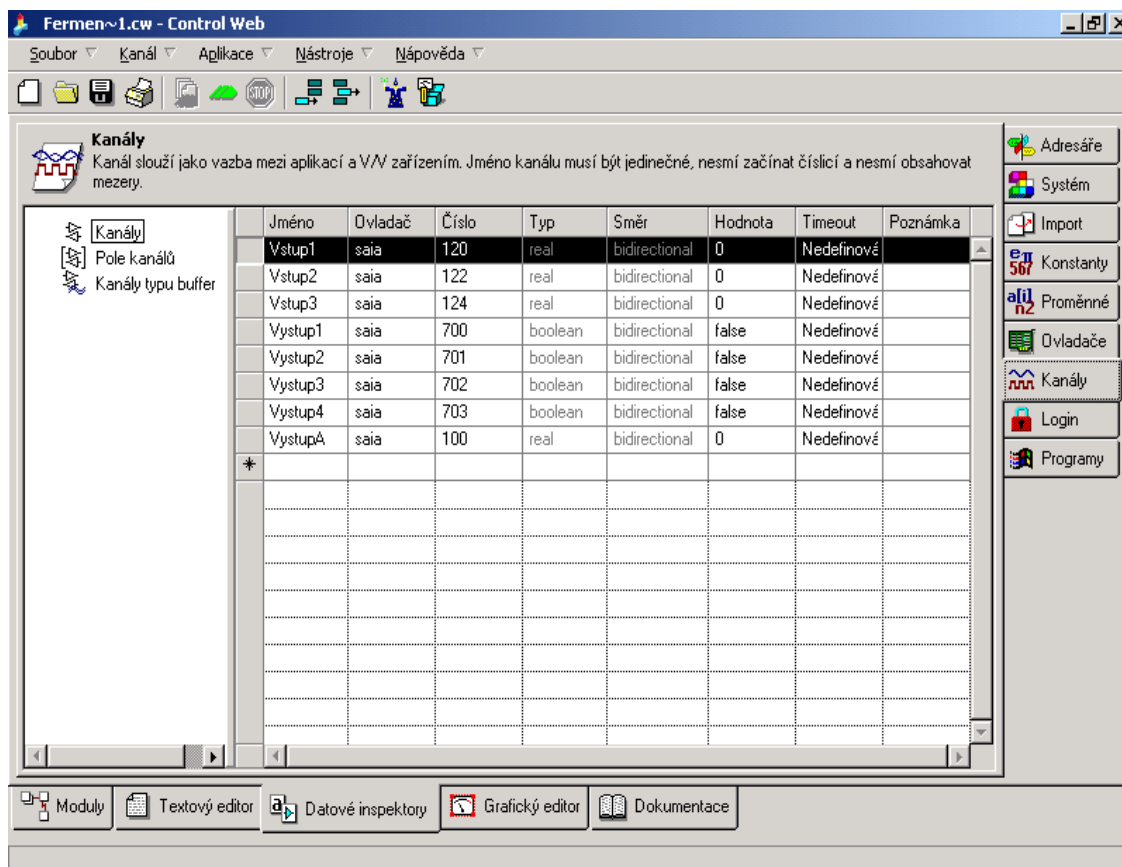
Konkrétně pro náš případ analogového vstupu:

Hodnota vstupu *teplota v reaktoru* je po A/D převodu uložena ve stanici do registru R32 (Obr. 15).

Definujeme si tedy blok kanálů: *Block = 01, 120, 130, R, 32, uint16*. To znamená, že hodnota kanálu 120 bude odpovídat registru R 32 ve stanici. V našem případě ovšem nebude hodnota dalšího kanálu, který má představovat registr R 33 (hodnotu dalšího analogového vstupu) 121 ale 122, protože datový typ *uint16*, který definujeme v bloku zabírá dva kanály, jak je vidět z tabulky 6.

### 4. Vytvoření kanálu v prostředí CW

Aby mohla aplikace v prostředí CW využít bloky kanálů definované v souborech ovladače je nutné vytvořit konkrétní kanály přímo aplikaci. V záložce datové inspektory – kanály je nutné zadat jméno kanálu a jeho číslo. Pro *vstup teplota v nádrži* je využít následující kanál: *Vstup1, 120*. Nastavení dalších kanálů nezbytných pro celou aplikaci je patrné z obrázku 17.



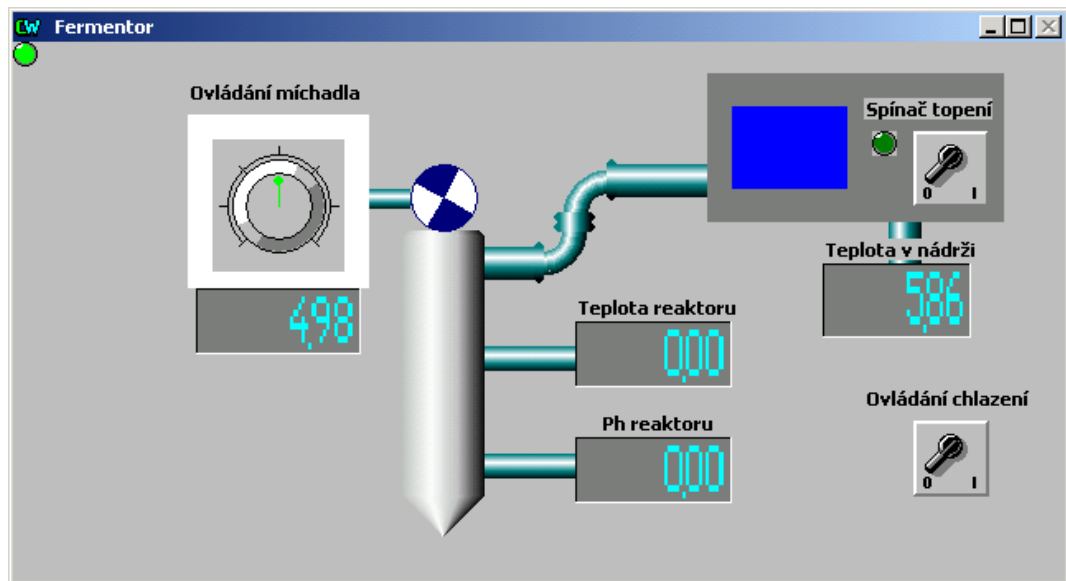
Obr. 17. CW – vytvoření kanálů

### 5. Přiřazení kanálů a aktivace přístrojů

Všechny virtuální přístroje provádějí svou činnost v rámci své aktivace. Pokud tedy u přístroje *meter* nastavíme v inspektoru přístroje (záložka *Expression*), jako hodnotu kterou má zobrazovat kanál Vstup1, bude hodnota kanálu Vstup1 zobrazena přístrojem *meter* právě v rámci aktivace. Základní způsob aktivace u aplikací reálného času je periodická aktivace. V inspektoru přístroje v záložce *Timer* je tedy nutné nastavit časový interval ve kterém se bude zobrazovaná hodnota aktualizovat. Vhodným intervalem může být například 0,5 sekundy, k aktualizaci dojde tedy dvakrát za vteřinu, což je naprosto dostačující.

### 6. Spuštění aplikace

Nyní už máme nadefinovány všechny nezbytné parametry pro komunikaci, můžeme tedy spustit aplikaci v prostředí Control Web.



Obr. 18. Okno běžícího programu fermentor

Pozn.: Testovací napětí bylo vždy přivedeno jen na jednu svorku analogového vstupního modulu PCD.W200, v tomto případě pouze jako vstup *teplota v nádrži*. Otestovány byly postupně všechny tři vstupy.

## ZÁVĚR

Předmětem této práce bylo provést propojení programovatelného automatu SAIA PCD2 s programem běžícím na nadřazeném počítači s ohledem na využití tohoto zapojení v systému řízení enzymatické hydrolýzy koželužných odpadů, konkrétně na pracovišti fermentor.

Úvodní teoretická část se zabývá obecnou problematikou využití programovatelných automatů (PLC) v procesech řízení v porovnání s ostatními možnostmi řešení. Pro PLC hovoří především jejich odolná konstrukce, důležitá v drsných průmyslových podmínkách a také modularita, která umožňuje sestavení přístroje, jehož vlastnosti odpovídají rozsahu a charakteru aplikace.

Dále byl středem zájmu konkrétní programovatelný automat SAIA PCD2. Bylo nutné nastudování všech vlastností PCD2 a to jak z hlediska hardwarové konfigurace, tak i programového vybavení. Jako nejvhodnější prostředek k vývoji programu pro řízení daného procesu se jeví zápis pomocí grafického jazyku FUPLA, který vyniká přehledností a snadným přístupem k tvoření programu. V případě složitějších programů ale přináší spoustu omezení a nutných kompromisů jenž vyplývají z filozofie jazyka.

Jako programové vybavení pro nadřazený počítač bylo vybráno vývojové prostředí Control Web. Jedná se o systém rychlého vývoje aplikací pro vizualizaci a řízení procesů v reálném čase. Seznámení se s principem tvoření programu a postupem navázání komunikace s periferním zařízením bylo základem návrhu aplikace běžící na nadřazeném PC. Bylo nutné seznámit se s ovladačem pro systém Control Web pro komunikaci se zařízením SIAA PCD.

Páteří celého systému řízení enzymatické hydrolýzy koželužných odpadů je síť Ethernet. Pro začlenění programovatelného automatu do systému je použit externí vstupně/výstupní modul ADAM-4571, sloužící jako převodník sběrnice RS 232/482/422 na Ethernet. Bylo realizováno zapojení – propojení PLC s modulem a nadřazeným PC. PLC je připojeno k sériovému portu modulu a ten pomocí síťového kabelu k PC. Nezbytnou součástí zapojení bylo nakonfigurování modulu pro komunikaci mezi PLC a PC, k tomu jsem využil programové vybavení dodávané spolu s modulem. Dodávané prostředí je velmi intuitivní a přehledné a umožňuje během krátkého času nastavit parametry komunikace.

V prostředí Control Web jsem navrhl schéma pracoviště fermentor a s využitím dodávaného ovladače pro komunikaci mezi programem Control Web a PLC SAIA PCD2 nadeřinoval parametry komunikace a především kanály prostřednictvím kterých ovladače komunikují s daným periferním zařizením.

Tato práce představuje komplexní řešení připojení programovatelného automatu SAIA PCD2 se sériovým rozhraním s nadřazeným PC pomocí sítě ethernet, včetně návrhu a realizace základního programového vybavení nezbytného pro jejich vzájemnou komunikaci.

**SEZNAM POUŽITÉ LITERATURY**

- [1] DOLINAY, Jan, VAŠEK, Vladimír. *Nové vybavení laboratoře recyklačních technologií*. XXX. ASR '2005 Seminar, Instruments and Control, Ostrava, 2005. 7.s
- [2] MARITNÁSKOVÁ, Marie. *Řízení programovatelnými automaty*. Skriptum ČVUT, Praha 1998. 157 s.
- [3] SAIA-Burgess Electronics. *Hardware – Řady PCD1 a PCD2*. Vydání 26/737 E6, 1998.
- [4] HORNYCH, Jan. *Aplikace PLC automatů pro řízení enzymatické hydrolýzy zpracování koželužných odpadů*. 1999. 55 s. Diplomová práce.
- [5] SAIA-Burgess Electronics. *Programming Tools for MS Windows PG4*. Vydání 26/748 E1, 1996.
- [6] *Control Web 5* [online]. [2004] [cit. 2006-05-25]. Dostupný z WWW: <<http://www.mii.cz/cat?id=94&lang=405>>.
- [7] *ADAM-4571/4571L/4571S Users manual*. Taiwan: Advantech, 2002. 66 s.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

Wi-Fi	Bezdrátová technologie
GSM	Technologie založená na mobilních sítích
PDA	Kapesní počítač
PLC	Programmable Logic Controller – programovatelný automat
MODBUS	Otevřený protokol pro komunikaci různých zařízení
RS 232/222/485	Standardy sériového rozhraní
PC	Osobní počítač
PCD	Process Control Device – programovatelný automat firmy SAIA
PCD2	Modelová řada automatů PCD
LED	Svítivá dioda
piggyback	Modul pro nasunutí na základní desku PLC
I/O	Vstupně/výstupní
PG4	Vývojové prostředí pro automaty SAIA
IL	Instruction List – forma zápisu programu jako výpis funkcí
FUPLA	Grafický programovací jazyk
FBox	Obdélník reprezentující v jazyce FUPLA funkci
CW	Control Web
RAD	Rapid Application Development - Prostředky rychlého vývoje aplikací
PGU	Programming Unit – konektor pro připojení programovacího zařízení
A/D	Analogově/digitální
D/A	Digitálně analogový
EDG	Ethernet Data Gateway – brána mezi rozhraním Ethernet a RS



**SEZNAM OBRÁZKŮ**

Obr. 1. Celkové schéma řídicího systému.....	8
Obr. 2. Okno knihovny projektů.....	21
Obr. 3. Okno správce projektu.....	22
Obr. 4. Výběr typu programového objektu.....	22
Obr. 5. Okno editoru FUPLA.....	24
Obr. 6. Okno programu Control Web.....	27
Obr. 7. Vložení přístroje meter z palety přístrojů.....	28
Obr. 8. Celkové schéma pracoviště fermentor.....	33
Obr. 9. Zapojení modulu ADAM-4571.....	37
Obr. 10. Význam jednotlivých pinů konektoru DB-9.....	38
Obr. 11. EDG – Configuratin Utility – vyhledávání modulu.....	39
Obr. 12. EDG – Configuratin Utility – nastavení parametrů.....	40
Obr. 13. EDG – Configuratin Utility – nasatvení portu.....	41
Obr. 14. EDG COMPort Mapping Utility.....	42
Obr. 15. Ovládání A/D a D/A převodníku v jazyce FUPLA.....	43
Obr. 16. CW – vložení ovladače.....	46
Obr. 17. CW – vytvoření kanálů.....	48
Obr. 18. Okno běžícího programu fermentor.....	49
Obr. 19. Programovatelný automat SAIA PCD2.....	59
Obr. 20. Základní deska PLC SAIA PCD2.....	59
Obr. 21. Modul ADAM –4571-A.....	60
Obr. 22. Schéma pracoviště fermentor v prostředí CW.....	60

**SEZNAM TABULEK**

Tab. 1. Přehled datových typů prostředků programovatelného automatu SAIA.....	18
Tab. 2. Přehled typů programových bloků.....	19
Tab. 3. Vstupní a výstupní signály PCD2.....	35
Tab. 4. Význam LED diod modulu ADAM-4571.....	37
Tab. 5. Význam pinů konektoru RJ-48.....	38
Tab. 6. Přehled datových typů ovladače PCD2.....	45

## SEZNAM PŘÍLOH

P I Soubory ovladače SAIA PCD pro Control Web

P II Obrazová dokumentace

## **PŘÍLOHA P I: SOUBORY OVLADAČE SAIA PCD PRO CW**

Mapovací soubor ovladače SAIA PCD pro CW:

```
begin
  1 real input
  10 real input
  11 real input
  12 string input
  13 real input
  14 real input
  15 real input
  16 real input
  17 boolean output
  18 boolean output
  20 real input
  21 real input
  22 string input
  23 real input
  24 real input
  25 real input
  26 real input
  27 boolean output
  28 boolean output
  100 - 150 real bidirectional
  200 - 263 boolean bidirectional
  500 - 515 boolean bidirect
  520 - 599 real bidirect
  600 - 615 boolean input
  700 - 715 boolean bidirectional
  900 - 917 real bidirectional
  1010 - 1017 boolean bidirectional
  1100 - 1107 boolean input
  1200 - 1207 boolean input
end.
```

Parametrický soubor ovladače SAIA PCD pro CW:

[ComPort]

Com = COM3

Protocol = p800

SBusMode = Parity

BaudRate = 9600

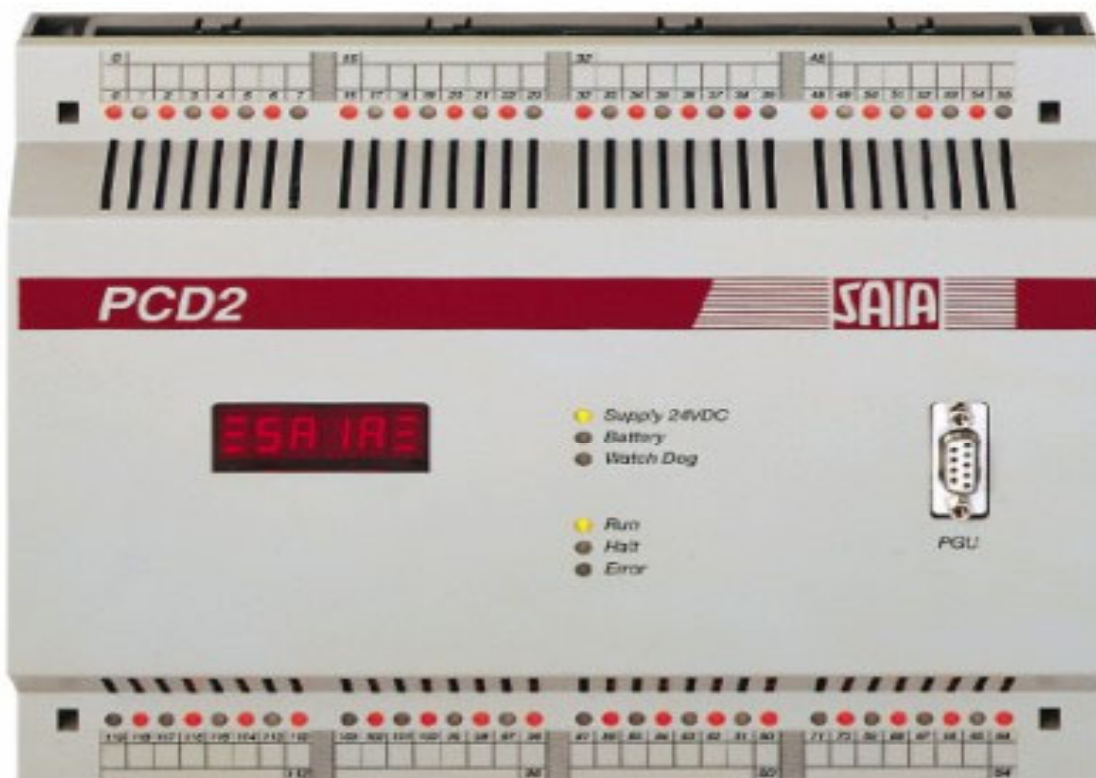
Timeout = 200

RTS = enable

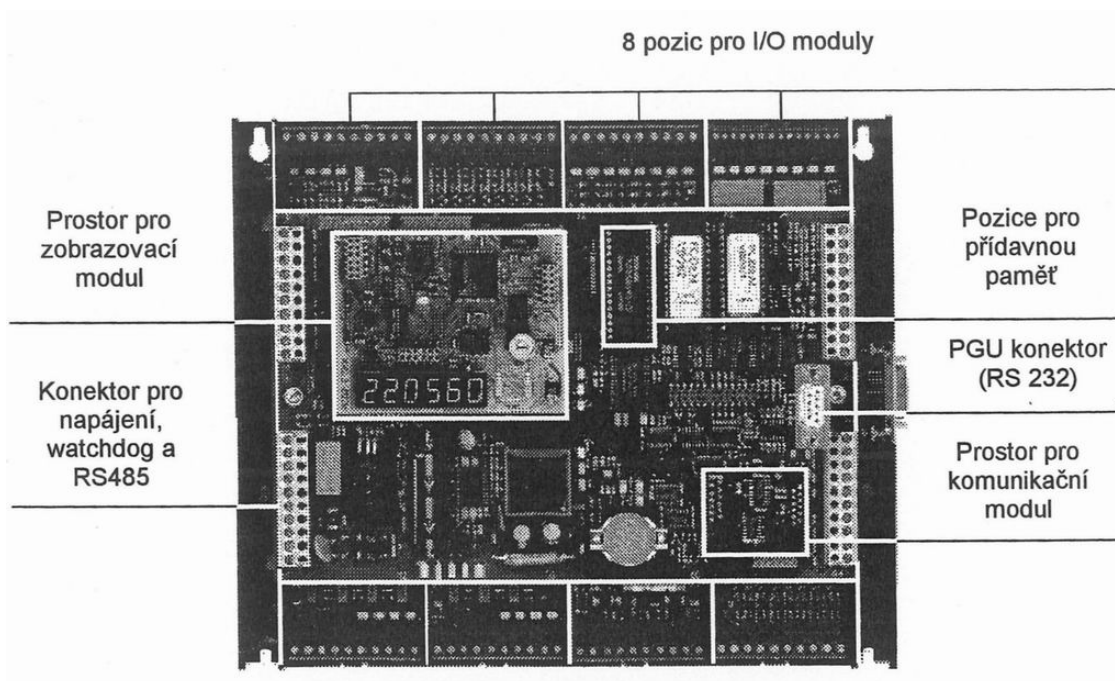
[Channels]

```
;-----  
;Block = Station, ChannelFrom, ChannelTo, Media, Offset, [Subtype], [Bidirect]  
;-----  
Block = 01, 100, 105, R, 64, uint8  
Block = 01, 120, 130, R, 32, uint16  
Block = 01, 500, 515, F, 0  
Block = 01, 520, 525, R, 10, uint16  
Block = 01, 600, 607, I, 16  
Block = 01, 700, 707, O, 0, bidirect  
;Block = 01, 901, 901, D, 0  
;Block = 01, 910, 917, K, 0, bidirect  
;Block = 01, 1010, 1017, O, 0, bidirect  
;Block = 01, 1100, 1107, I, 32  
;Block = 01, 1200, 1207, F, 0
```

## PŘÍLOHA P II: OBRAZOVÁ DOKUMENTACE



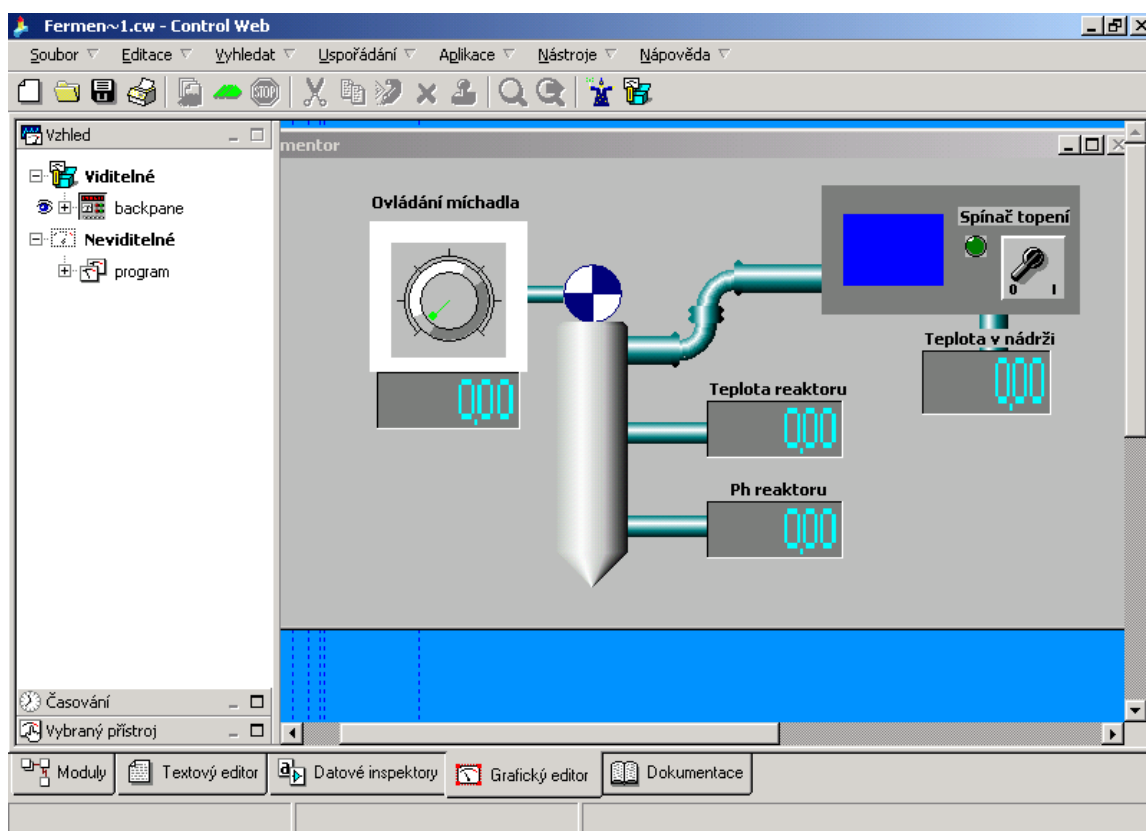
Obr. 19. Programovatelný automat SAIA PCD2



Obr. 20. Základní deska PLC SAIA PCD2



Obr. 21. Modul ADAM-4571-A



Obr. 22. Schéma pracoviště fermentor v prostředí CW