OBJEKTOVĚ ORIENTOVANÉ VÝVOJOVÉ SYSTÉMY

G STUDIJNÍ CÍLE

Po prostudování :

- § Seznámíte se s možnostmi objektově orientovaného systému.
- § Poznáte jeho základní stavební prvky
- § Seznámíte se s postupem tvorby měřící aplikace
- § Porozumíte principu činnosti takto vytvořené aplikace.

Ñ KLÍČOVÁ SLOVA

vizuální programování, data, objekty, metody, virtuální přístroj.

Â ČAS POTŘEBNÝ KE STUDIU

135 minut

1. CONTROL PANEL

1.1 Úvod do problematiky

Tento systém umožňuje vývoj aplikací pro přímé řízení jednotlivých strojů i celých technologií, zpracování rychlých signálů v reálném čase, modulárních distribuovaných aplikací v rámci počítačové sítě.

Je to objektově orientovaný systém, který slouží ke generování měřicích, řídicích a regulačních programů pro PC. Systém může být využit ve strojírenství, hutnictví, potravinářském a chemickém průmyslu, v energetice.

Prostředí Control Panel umožňuje vytvářet programy pomocí symbolického popisu navrhovaného systému.

Použité technologie

- § virtuální modely
- § vizuálního programování
- § technologie sdílení kódu

Výhody sdílení kódu

- § výkon
- § spolehlivost
- § vzhled a uživatelský komfort

Volba způsobu tvorby aplikace

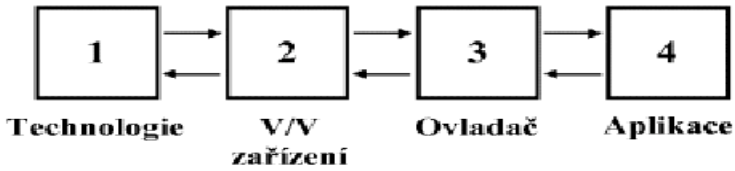
- § **vizuální programování** - spočívá ve tvorbě aplikací výhradně pomocí myši, grafických symbolů, algoritmických a vizuálních struktur
- § **textové programování** - zápisem klíčových slov v textovém editoru
- § **kombinace obou možností** - částečně vizuální a částečně textové programování

Aplikační program tvoří **data** a **objekty** (zapouzdřená data + metody) s určitými vlastnostmi. Control Panel bezprostředně nepodporuje procedurální programovací konstrukce ale pracuje jako **neprocedurální událostmi řízený systém kooperativních paralelních procesů**. To znamená, že objektový model dané technologie se chová podobně jako skutečná řízená technologie.

1.2 Základní prvky systému

- § ovladač V/V zařízení
- § vstupní, výstupní nebo virtuální kanál
- § obrazový panel
- § virtuální přístroj

Na obr. 1.1 je zobrazeno principiální schéma systému.



Obr. 1.1 Principiální schéma

1.2.1 Základní prvky měřicích a regulačních systémů

- **konstanty a proměnné** - slouží pro přenos dat uvnitř aplikace
- **ovladače vstupně / výstupních zařízení** - slouží pro propojení aplikace s reálným světem
- **vstupní, výstupní nebo obousměrné kanály** - slouží pro přenos dat mezi aplikací a reálným světem přes ovladač
- **virtuální přístroje** - realizují vlastní funkčnost aplikace, slouží k vykonávání nejrůznějších činností: zobrazování a nastavování hodnot, regulaci, archivaci a další zpracování dat ap.
- **obrazové panely** - zvláštní typ virtuálních přístrojů; slouží pro estetické a logické slučování více virtuálních přístrojů do jednoho celku
- **časovače** - zvláštní typ virtuálních přístrojů; pomocí sekvence, selekce a iterace slouží pro tvorbu časových algoritmů

1.3 Postup práce se systémem

Při práci se systémem je nejprve nutno definovat potřebné **proměnné** (příp. konstanty) a prostřednictvím **ovladačů** a **kanálů** stanovit vazby programu na reálná výstupní zařízení (pro demonstraci je možno použít i **virtuální** a **modelový** ovladač). Potom můžeme rozmístit a propojit panely, přístroje, ovládací a indikační prvky, regulátory, správce dat atd. Nemusíme se zabývat obvyklými problémy (alokace a uvolňování paměti, ukazatele...), které v hojném počtu provázejí programování v běžných programovacích jazycích.

Od ostatních systémů podobného zaměření se Control Panel liší hlavně těmito vlastnostmi:

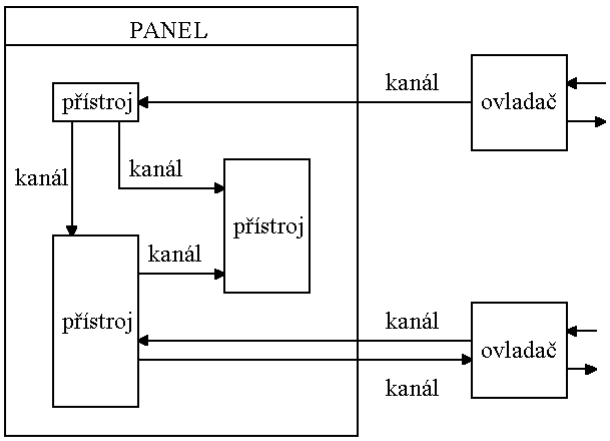
- převážná většina běžných systémů je nějakým způsobem omezena (tzn. je v nich omezen např. počet vstupně / výstupních kanálů, současně zobrazitelných údajů apod.). Control Panel nemá v principu žádná omezení. Jedinými limity jsou velikost paměti, velikost disku a rychlost počítače.
- na rozdíl od jednoduchých systémů Control Panel pracuje ve víceúlohovém prostředí s kvalitním grafickým uživatelským rozhraním. Prostřednictvím grafických ovladačů spolupracuje s různými grafickými adaptéry (např. HiColor a TrueColor kartami s vysokým rozlišením, GUI akcelerátorem s čipem S3 aj.).

- většina obdobně zaměřených systémů pracuje pouze v reálném módu procesoru 80x86. Control Panel plně využívá všech výhod chráněného módu a navíc podporuje i virtualizaci kódových a datových segmentů.
- běžné systémy často produkují pouze zdrojový kód nějakého programovacího jazyka. Aplikační program je pak nutno dodatečně přeložit standardním překladačem a sestavit jej s dodávanými knihovnami. Naproti tomu Control Panel přímo generuje událostmi řízený multitaskový program.
- Control Panel není pouhým konfigurovatelným programem. Je skutečným volně programovatelným vývojovým systémem.
- Control Panel přináší možnost vizuálního programování aplikací technologií objektového modelování.

Výrazové prostředky

- zdrojový text
- překladač
- klíčová slova
- editor aplikací

Na obrázku 1.2 je uvedeno principiální schéma měřící a řídicí aplikace v prostředí Control Panel



Obr. 1.2 Obecné schéma měřící aplikace

1.4 Ovladače vstupně/výstupních zařízení

Tyto ovladače umožňují, aby aplikace v systému Control Panel byla schopna komunikovat s libovolným zařízením v počítači (např. převodníkové karty) nebo s vnějším zařízením propojeným s počítačem (např. sériovou linkou RS-232 a podobně). Ovladač tvoří dynamicky linkovaná knihovna (DLL), která se do paměti zavede až v okamžiku jejího spuštění. Ovladače mohou být napsány v některém z

vývojových systémů podporujících tvorbu DLL pro operační systém OS/2 v 1.x, např. TopSpeed Modula-2, Pascal, C/C++.

Každému ovladači je přiřazen **popisový soubor** jeho kanálů. Tento soubor využívá překladač při dekódování zdrojového textu (je kontrolována správnost použití jednotlivých kanálů).

1.4.1 Typy ovladačů

- **Virtuální ovladač** - poskytuje řadu signálů pro demonstrační použití
- **Modelový ovladač** - slouží pro testování regulačních obvodů

Virtuální ovladač

Virtuální ovladač se neváže na žádné konkrétní fyzické zařízení. Slouží ke generování průběhu různých signálů a poskytuje různé systémové údaje (kanály 1 až 96). Tak jako každý ovladač má i virtuální ovladač svůj mapovací soubor. Standardní tvar mapovacího souboru VSOURCE.DMF je následující:

```
begin
        1 - 96  real input
        97              real output
        98              boolean output
        99              real output
        101 - 200       real output
        201 - 300       boolean output
        301 - 400       integer output
        501 - 600       boolean input
        601 - 700       string output
end
```

Kanály 1 až 96 tohoto souboru jsou vyhrazeny pro generování vstupních signálů. Typ jednotlivých kanálů je zřejmý z popisového souboru.

Virtuální ovladač poskytuje mimo jiné následující signály a služby:

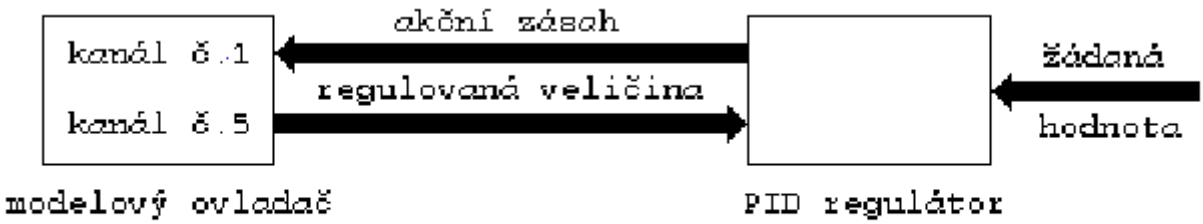
kanál č.1 - harmonický signál v závislosti na čase
kanál č.2 - trojúhelníkový signál v závislosti na čase
kanál č.3 - náhodný signál
kanál č.4 - pilový signál v závislosti na čase
kanál č.5 - obdélníkový signál v závislosti na čase
kanál č.6 - signál sinus v závislosti na počtu čtení
kanál č.7 - trojúhelníkový signál v závislosti na počtu čtení
kanál č.8 - náhodný signál
kanál č.9 - pilový signál v závislosti na počtu čtení
kanál č.10 - obdélníkový signál v závislosti na počtu čtení
kanál č.11 - počet sekund od začátku dne s přesností na setiny

Modelový ovladač

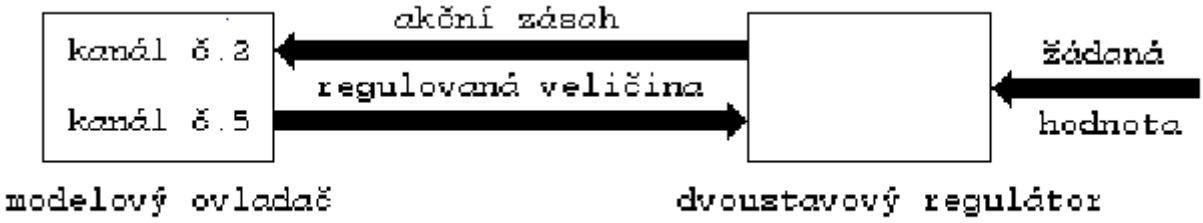
Modelový ovladač slouží jako náhrada reálné soustavy při testování správné funkce řídícího programu, zvláště pak regulačních obvodů. Tak jako každý ovladač má i modelový ovladač svůj mapovací soubor. Standardní tvar mapovacího souboru MODEL.DMF je následující:

```
begin
  1 longreal output
  2 boolean  output
  3 boolean  output
  4 boolean  output
  5 longreal input
end
```

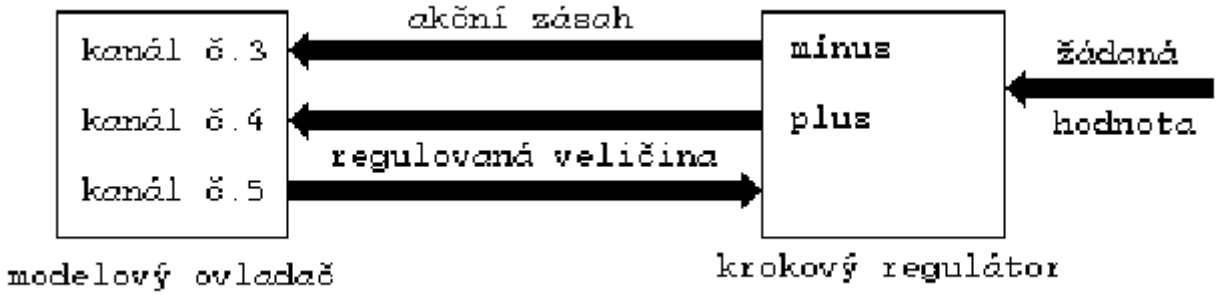
Protože modelový ovladač slouží pro testování jak PID regulátorů (obr. 1.3), tak i pro testování dvoustavových regulátorů (obr. 1.4) a krokových regulátorů (obr. 1.5), můžeme na jeho vstupy připojit akční zásahy všech tří regulátorů. Schéma zapojení je následující:



Obr. 1.3 PID regulátor



Obr. 1.4 Dvoustavový regulátor



Obr. 1.5 Krokový regulátor

Regulovaná veličina se počítá podle :

$$y(k) = + [b_1 u(k-1) - a_1 y(k-1)] + [b_2 u(k-2) - a_2 y(k-2)] + \ldots \quad (1.1)$$

Typické hodnoty periody vzorkování pro regulaci v technologických procesech:

- § průtok > 1s
- § tlak > 5s
- § hladina > 10s
- § teplota > 20s

1.5 Další prvky systému

Kanály

Slouží pro přenos dat mezi ovladači a přístroji a pro přenos dat mezi přístroji navzájem. Každý kanál se musí vztahovat k určitému ovladači. Pro komunikaci mezi přístroji je výhodnější kanály registrovat u virtuálního ovladače.

Panely

Přístroj panel je určen ke shromažďování více přístrojů do jednoho objektu, který může být zobrazován, skrýván nebo minimalizován (je-li panel v okně) podle zadaných booleovských podmínek.

Přístroj panel může zobrazovat na svém podkladu jakékoliv DataView, které je podporováno systémem Control Web, Control Panel, např. bitmapové nebo vektorové obrázky, texty, grafy, tabulky, databáze, HTML dokumenty, animované FLC soubory ap.

Přístroje

Přístroje jsou objekty, které zpracovávají data získaná z jednotlivých kanálů nebo na kanály určitá data vysílají. Mohou tato data zobrazit, archivovat nebo generovat různé výstupní signály, dále mohou nastavovat kanály na požadované

hodnoty, popř. regulovat a řídit technologické procesy. Některé přístroje mají pouze informativní či estetický význam (nápisy, ikony ...) .

2. VIRTUÁLNÍ PŘÍSTROJE

2.1 Obecný popis

Všechny přístroje mají v seznamu uvedeny jako první v pořadí tzv. standardní parametry. Jsou to parametry, jejichž význam je naprosto totožný pro všechny přístroje, v jejichž syntaxi jsou uvedeny. Jsou to parametry:

```
timer = timer_name;
owner = owner_name;
position = integer, integer, integer, integer;
win_disable = [ identifier, . . . ];
win_title = string;
access = integer;
tab_select = integer;
send_same_data;
```

2.2 Dělení přístrojů

- obecné přístroje - crt, draw, program, table, trend, trend_viewer
- systémové - file, iterator(časovač), keyboard, panel, selector, sequencer
- základní zobrazovací a ovládací - chart, control, demultiplexer, indicator, label, meter, multiplexer, switch
- databázové přístroje - alarm, archiver, backup, recipe
- regulátory - pid_regulator, boolean_regulator, step_regulator
- řetězcové přístroje - string_control, string_display, string_switch
- zvukové přístroje - sound, multi_sound
- přístroje pro práci s buffery - buffer_convertor, buffer_display, buffer_sumator
- matematické přístroje - integrator, time_integrator
- http server - httpd
- neuronová síť - neural_net
- ladící přístroje - monitor

2.2.1 Obecné

crt

Časové zobrazení funkce dvou proměnných f(x,y) vyjádřených pomocí numerických výrazů x a y do jednoho grafu.

draw

Zobrazení vektorové kresby složené ze základních geometrických prvků (přímek, elips, obdélníků apod.). Rozměry jednotlivých prvků mohou být svázány s výrazy a mohou se tedy s časem měnit.

program

Přístroj program realizuje obecně zapsaný algoritmus. Průběh algoritmu je definován řídicími příkazy (podmínky, cykly, apod.). Mimo řídicí příkazy je k dispozici několik výkonných příkazů (např. přehrání WAV souboru).

V jeho rámci je možné vyčíslovat libovolné výrazy a jejich výsledky přiřazovat datovým elementům (proměnným, kanálům a prvkům polí proměnných či kanálů).

Pokud není stanoveno jinak, celý algoritmus programu (od počátku do konce) proběhne vždy v jednom časovém kroku, tedy se shodnými naměřenými hodnotami na kanálech.

Je však možné průběh programu pozastavit na určitou dobu, případně do splnění podmínky. V následujícím systémovém časovém kroku (nikoliv tedy časovém kroku přístroje program) je otestována podmínka běhu, a je-li splněna, program pokračuje od místa za podmínkou, ne opět od počátku. V tomto případě už jsou hodnoty čtených kanálů znovu změřeny.

Významnou vlastností programu je schopnost volat OCL metody přístrojům, jež tyto metody definují. V rámci algoritmu programu je tedy možné přímo komunikovat s jinými přístroji prostřednictvím hodnot předávaných jako argumenty OCL metod.

Výrazové prostředky přístroje program:

Obecná struktura programu

Návěští
Příkaz "if"
Příkaz "loop"
Příkaz "while"
Příkaz "repeat until"
Příkaz "pause"
Příkaz "wait"
Příkaz "stop"
Příkaz "sound"
Příkaz přiřazení
Příkaz "move"
Volání OCL metod

table

Dynamická výměna dat s tabulkovým kalkulátorem a jeho kompletní ovládání prostředky Control Panel. Přístroj table umožňuje přímo zapisovat data do buněk tabulky, vyvolávat přepočet tabulky a zpětně číst data z buněk do systému. Tabulka zobrazuje "živá" data a případně i jejich grafické vyjádření.

trend

Umožňuje sledování zobrazování a archivování historických trendů analogových signálů. Data archivuje do standardního databázového souboru typu "*.DBF"

trend_viewer

Zobrazování, prohlížení a tisk historických trendů a statistických hodnot analogových signálů. Přístroj je nečasovatelný a jeho činnost lze řídit prostřednictvím OCL metod.

2.2.2 Systémové

file

Umožňuje pracovat s obecným diskovým souborem. Veškerá funkčnost je realizována pouze prostřednictvím volání OCL metod.

keyboard

Slouží pro asynchronní zpracování nadefinovaných událostí od klávesnice.

panel

Přístroj panel je určen ke shromažďování více přístrojů do jednoho objektu, který může být zobrazován, skrýván nebo minimalizován (je-li panel v okně) podle zadaných booleovských podmínek.

2.2.3 Základní zobrazovací a ovládací přístroje

chart

Zobrazení výsledky více numerických výrazů pomocí čar v jednom grafu, s možností přiřazení těchto hodnot do jiných proměnných (výstupních kanálů).

control

Nastavení právě jedné numerické hodnoty do proměnné (výstupního kanálu) pomocí knoflíku, rolovací lišty, numerického řádku nebo pomocného okna.

demultimplexer

Zobrazení výsledku libovolného numerického výrazu pomocí ručkového přístroje, grafu, popř. displeje. Hodnota výrazu je nejen zobrazena, ale také vyslána na výstupní proměnné (kanály), které mají splněnu podmínku výběru.

indicator

Zobrazení výsledku logického výrazu a jeho případné přiřazení do proměnné nebo vstupního kanálu.

label

Zobrazení libovolné ikony nebo textu.

meter

Zobrazení výsledku numerického výrazu pomocí ručkového přístroje, sloupcového indikátoru, grafu a číslicového zobrazovače, s možností přiřazení výsledku do proměnné nebo výstupního kanálu.

multi_label

Jako *label*, ale přístroj může podle stanovených podmínek měnit svůj vzhled i polohu.

multimlexer

Zobrazení hodnoty numerického výrazu pomocí ručkového přístroje, grafu, popř. displeje. Přístroj vybere jeden z výrazů pomocí zadaných podmínek.

switch

Nastavení logické hodnoty do proměnné nebo výstupního kanálu typu boolean pomocí tlačítka.

2.2.4 Databázové přístroje

alarm

Slouží pro zpracování nadefinovaných alarmů a poruch. Podle definice chování při jednotlivých vzniklých stavech může akusticky upozorňovat obsluhu, tisknout na tiskárnu a archivovat vzniklé stavy do DBF souboru podobně jako u přístroje *archiver* i aktivovat nějakou jinou činnost.

archiver

Archivuje požadovaná data do standardního databázového souboru typu "*.DBF".

backup

Archivuje v definované periodě požadovaná data aplikace do souboru. Při startu aplikace může poslední zálohovaná data nastavit na příslušné datové elementy.

journal

Pořizuje záznamy o všech sledovaných změnách v aplikaci a archivuje je do standardního databázového DBF souboru. Strukturu záznamu lze uživatelsky podle potřeby upravovat.

recipe

Receptura může pracovat ve dvou režimech - vstupním nebo výstupním. Ve výstupním režimu jsou data z receptury nastavována na výstupní datové elementy

nebo jsou data načítána ze standardního databázového souboru typu DBF. Ve vstupním režimu jsou data načítána do položek záznamů receptury nebo jsou data zapisována do databázového souboru.

2.2.5 Přístroje pro práci s buffery

buffer_convertor

Přístroj umožňuje kopírovat hodnoty z dataelementů buffer do dataelementů array a naopak.

buffer_display

Zobrazení dat v kanálech a proměnných typu buffer.

buffer_sumator

Jednoduché operace pro práci s kanály a proměnnými typu buffer. Přístroj poskytuje součet bufferů a násobení bufferu konstantou.

2.2.6 Matematické přístroje

integrator

Přístroj vypočítává numerický integrál zadané diferenciální rovnice.

time_integrator

Přístroj vypočítává numerický integrál zadané diferenciální rovnice s časovou proměnnou.

2.2.7 Neuronová síť

neural_net

Vrstevnatá perceptronová neuronová síť s adaptačním algoritmem Back-Propagation. Umožňuje adaptaci sítě v průběhu její aktivní činnosti, změnu parametrů adaptace pomocí kanálů a změnu topologie sítě v průběhu učení příp. aktivní činnosti. Trénovací množina může být modifikována kdykoliv během činnosti sítě.

2.3 Funkční bloky programu

- Settings - blok parametrů aplikace
- Import - definuje další moduly, které jsou pro běh aplikace nezbytné
- Network
- Const

- Var
- Driver - syntaxe zápisu ovladačů v textovém editoru
- Channel - Kanál je v aplikačním programu používán obdobně jako proměnná. Od proměnné se však liší tím, že zprostředkovává vazbu aplikačního programu na skutečné signály připojené na vstupně/výstupní zařízení.
- Timer - Časovače jsou základními stavebními prvky při vytváření algoritmických struktur programu.
- Instrument - Blok přístrojů zdrojového textu aplikace obsahuje definice jednotlivých virtuálních přístrojů a jejich parametry. Jedná se o "výkonnou" část aplikace, která zajišťuje vlastní funkčnost aplikace.
- Login - Sekce login je určena pro definici přístupových práv uživatelů k jednotlivým částem běžící aplikace.

2.3.1 Timer - blok časovačů

Sequencer

Zajišťuje deterministické pořadí vykonávání v něm zaregistrovaných přístrojů v rámci svého časového kroku. Všechny časovače mohou být ovládány dalšími časovači. Tato vlastnost vzájemného vnořování a závislostí zaručuje nekonečné množství kombinací a variant vytvořených algoritmů, pouze pomocí tří základních logických struktur.

Selector

Výběr umožňuje libovolný počet větvení programu, která se mu do jeho struktury zadávají v podobě jména a logického výrazu, který vyjadřuje podmínku. Při splnění podmínky se zavolají všechny přístroje mající v položce timer uveden název selectoru a jméno této konkrétní podmínky:

```
timer = selector_name.case_name;
```

Používání více přístrojů selector se stejným časovým krokem je velice neefektivní. V tomto případě je vhodné všechny takto časované přístroje sdružit do jediného přístroje typu selector s více podmínkami.

Iterator

Jde o časovač, který bude cyklicky obvolávat podřazené přístroje, dokud nebude splněna podmínka definovaná parametrem exit. Celá smyčka bude až do splnění výstupní podmínky prováděna v jediném časovém kroku.

2.4 Výrazové prostředky programu

Základním cílem každého algoritmu je podle Jacksonovy teorie převést vstupní data na výstupní. Aby mohl program s daty pracovat, musí být tato data někde uložena. V systému Control Panel existuje několik možností - datových elementů, jak uchovávat a přenášet data.

Pro úschovu dat, která se nebudou při běhu aplikačního programu měnit, jsou určeny **konstanty**. Pro data, která se za běhu aplikačního programu mění, případně pro úschovu veličin nepřipojených na hardware jsou v sytému k dispozici **proměnné**.

Kanál je datový element, který je spojen s určitým ovladačem. Bez definice ovladače nelze s kanály pracovat. Rozdíl mezi proměnnou a kanálem je tedy takový, že kanál musí být připojen na ovladač, zatímco proměnná ne. Jinak jsou tyto datové elementy pro použití v programech rovnocenné.

Každý proces vytváření programů sebou přináší dvě základní oblasti problémů. První z nich je vytvoření samotného algoritmu programu, druhá jeho komunikace s okolím.

V systému Control Panel lze algoritmus programu "poskládat" pomocí grafického editoru a tří základních prvků, kterým budeme říkat **časovače**. Jde o **sekvenci**, **selekci** a **iteraci**. Tyto časovače pak vytváří samotnou strukturu programu, která spouští jiné prvky, tzv. **přístroje**

2.5 Činnost systému při přenosu dat pomocí ovladače

Systém časování v programu Control Panel pracuje následujícím způsobem. Nejprve jsou zpracovány požadavky na čtení hodnot z kanálů u těch virtuálních přístrojů, které mají být řešeny v rámci právě probíhajícího časového kroku. Po získání vstupních dat proběhne vlastní činnost těchto přístrojů (zobrazení hodnot, testování podmínek, archivace atd.). Pak následuje zápis hodnot do výstupních kanálů ovladačů. Je-li ukončen zápis, systém znovu kontroluje, zda u některých přístrojů nenastal časový krok a činnost se opakuje.

Systém provádí optimalizaci požadavků na čtení i zápis hodnot kanálů. Tak je zaručeno, že v rámci jednoho časového kroku je hodnota kanálu čtena nebo zapisována pouze jednou. Zapisuje-li několik virtuálních přístrojů v jednom časovém kroku do téhož výstupního kanálu, bude akceptována poslední zapsaná hodnota. Jestliže ovladač není schopen po dobu danou parametrem *time_step_limit* poskytnout požadovaná data, pracuje virtuální přístroj s hodnotou z minulého časového kroku. Jakmile má ovladač data k dispozici, data jsou předána do systému a další přístroje už mohou pracovat s novými hodnotami.

Při startu aplikace by tato vlastnost nevyhovovala, proto systém při prvním čtení na data z ovladače čeká. Toto ale neplatí pro virtuální přístroje, které mají definován absolutní časový krok (s ofsetem). Vyžaduje-li aplikace, aby se pracovalo vždy s aktuálními hodnotami, je možno nastavit jeden z parametrů systému tak, aby se vždy čekalo na data z ovladače (time_step_limit). Po zápisu do kanálů ovladače systém

čeká na potvrzení o úspěšnosti této operace. Dokud toto potvrzení nedostane, nedojde k dalšímu testování časových kroků.

U přístrojů, které nejsou časovány (např. tlačítko) se jejich požadavky okamžitě zařazují mezi požadavky ostatních přístrojů, a to opět v pořadí nejprve čtení, potom zápis.

2.6 Činnost systému při spuštění a zastavení aplikace

Při spouštění aplikace jsou nejprve vytvořeny instance veškerých použitých objektů a tyto jsou zapojeny do patřičných struktur. Pak je každý objekt inicializován na své počáteční hodnoty. Jako poslední jsou inicializovány přístroje typu *backup*, aby jimi nastavené datové elementy již nemohly být jinými přístroji ovlivněny.

To, jestli inicializované přístroje mají aktivovat své "receivers", lze definovat nastavením parametru **skip_init_outputs** v sekci **settings**.

Poté jsou již jednotlivé přístroje aktivovány podle svého časového kroku, algoritmu úlohy a také normálně přijímají události. Jako vůbec první je samozřejmě aktivován přístroj jména startup. Je-li takovýmto přístrojem program, pak běží výhradně až do případného přerušení na instrukcích **wait** nebo **pause**.

2.6.1 Absolutní a relativní časování, posun časování

Je–li přístroj nebo časovač časován hodnotou (přímým zadáním časového kroku), je možné zvolit **absolutní časování**. Absolutní časování zaručuje aktivaci přístroje v periodě dané časovým krokem počítaným nikoliv od startu aplikace, ale od půlnoci dne startu aplikace. Pokud je změněn systémový čas, je následující časový krok stanoven opět k půlnoci aktuálního dne. Pouze v případě, kdy jeden den (24 hodin) je celočíselným násobkem periody časování, je doba aktivace přístroje shodná ve všech dnech.

Mimo prostého zvolení absolutního časování je navíc možno zvolit **posun časování** menší, než je perioda. Např. má-li být přístroj aktivován vždy v páté sekundě každé minuty, bude perioda časování 60 a posun 5.

Časový interval udávající posunutí se zapisuje na řádek klíčového slova **timer** za údaj o periodě časování a je oddělený čárkou. Např. časování výše zmíněného přístroje bude zapsáno následovně:

```
meter budík
  timer = 60, 5;
  ...
end_meter;
```

Nebude–li parametr za čárkou uveden, přístroj bude časován relativně. Je–li požadováno pouze absolutní časování vzhledem k půlnoci, je nutné uvést posun časování 0.

2.7 Skupina přístrojů pro energetiku

Umožňuje sledování a řízení odběru elektrické energie. Zastřešujícím přístrojem je "*power_instrument*", funkce jsou rozčleněny do několika přístrojů.

Supply_metter

umožňuje zobrazovat odběr činné a jalové složky elektrické energie, přepočítávat vstupní impulsy na kWh a zjišťovat účiník. Vstupem jsou dva reálné výrazy - činná a jalová složka.

Maximum

umožňuje zadat hodnotu maxima pro probíhající a následující měsíc, sledovat množství odebrané energie v probíhajícím maximu, předpoklad pro celé maximum a dobu trvání maxima.

Přístroj má 3 výstupní kanály - hodnota nastaveného maxima, aktuální odběr v maximu, odběr v aktuální hodině.

Section

Sekce je část rozvodné sítě elektrické energie, do které lze regulovat přívod elektrického proudu.

Maximum_graph

zobrazí odběr elektřiny v probíhajícím maximu a zároveň zobrazí předpoklad odběru za dobu trvání celého maxima.

Diagram

Lze zadat odběrový diagram pro probíhající a následující měsíc včetně snížení pro jednotlivé regulační stupně.

2.8 Možnosti rozšíření systému

DDK (Device Development Kit)

Slouží pro vývoj a tvorbu ovladačů V/V zařízení v podobě DLL knihoven.

Postup

- vytvoření dialogového okna ovladače
- vygenerování projektu a zdrojových textů pro zvolený programovací jazyk (projektový, definiční a implementační soubor)
- doplnění implementací funkcí a procedur umožňující ovládání konkrétního zařízení
- překlad a linkování ovladače
- vytvoření mapového a případně parametrického souboru

S KONTROLNÍ OTÁZKY TEORETICKÉ

1. (1 bod) Z jakých základních prvků se skládá systém Control Panel ?
2. (2 body) Jaký je postup práce se systémem a jak vypadá principiální schéma měřící aplikace ?
3. (2 body) K čemu slouží V/V ovladače ?
4. (2 body) Co představuje modelový a virtuální ovladač ?
5. (4 body) Jaká je funkce virtuálního přístroje? Popište základní katagorie těchto přístrojů.
6. (2 body) Jaké funkční bloky programu máme k dispozici ?
7. (2 body) Vysvětlete pojem „časování“ aplikace.

_ SHRNUTÍ

Nové poznatky:

- objektově orientovaný systému
- tvorba měřící aplikace
- prostředky dostupné pro tvorbu aplikace
- modelové a virtuální ovladače
- časování vytvořené aplikace
- přístroje dostupné pro energetiku

Nové pojmy :

vizuální programování, data, objekty, metody, virtuální přístroj.

Ñ KLÍČ K TEORETICKÝM OTÁZKÁM

1. Kapitola 1.2
2. Kapitoly 1.3, obr. 1.2
3. Kapitola 1.4
4. Kapitola 1.4.1
5. Kapitola 2.2
6. Kapitola 2.3
7. Kapitoly 2.3.1, 2.6

$ AUTOKONTROLA

Pokud jste získali minimálně 10 bodů z teoretických otázek, můžete pokračovat dále ve studiu. V opačném případě si ve zkráceném čase příslušné kapitoly znovu nastudujte.