

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Simulace a vizualizace soustavy nádrží

Michal Flídr

Bakalářská práce

2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal FLÍDR**
Osobní číslo: **I07415**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Název tématu: **Simulace a vizualizace soustavy nádrží**
Zadávající katedra: **Katedra elektrotechniky**

Z á s a d y p r o v y p r a c o v á n í :

Cíl:

Vytvořit aplikaci v PROMOTICu, která bude v reálném čase simulovat a vizualizovat chování zadaného systému.

Teoretická část:

- a) nastudovat numerické řešení soustavy diferenciálních rovnic metodou Runge-Kutta
- b) nastudovat a popsat možnosti SCADA/HMI prostředí PROMOTIC

Praktická část:

- a) vytvořit a odladit proceduru pro výpočet zadané soustavy diferenciálních rovnic v jazyce VB Script
- b) v prostředí PROMOTIC implementovat numerický výpočet pracující v reálném čase
- c) navrhnout vizualizaci daného systému včetně zadávání parametrů a realizovat ji v prostředí PROMOTIC

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

REKTORYS, Karel; HORÁK, Karel. Přehled užití matematiky 2. 6., přeprac. vyd. Praha : Prometheus, 1995. 874 s. ISBN 80-85849-62-3(váz.).
či jiná literatura zabývající se metodou Runge-Kutta pro řešení soustavy diferenciálních rovnic.

Vedoucí bakalářské práce:

doc. Ing. František Dušek, CSc.
Katedra řízení procesů

Datum zadání bakalářské práce:

2. prosince 2010

Termín odevzdání bakalářské práce:

13. května 2011



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Zdeněk Němec, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2011

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 4. 5. 2011

Michal Flídr

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce doc. Ing. Františku Duškovi, CSc. za cenné rady a připomínky při zpracování této práce, jeho ochotný a trpělivý přístup.

Anotace

Tato práce se zabývá vytvořením aplikace v SCADA / HMI systému PROMOTIC. Aplikace simuluje časové průběhy výšek hladin v soustavě tří vzájemně propojených nádrží v reálném čase v závislosti na průběhu přítoků a otevření vypouštěcích a přepouštěcích ventilů. Simulovaný systém je popsán soustavou obyčejných nelineárních diferenciálních rovnic. Pro numerické řešení je použita metoda Runge-Kutta.

Klíčová slova

simulace, diferenciální rovnice, Runge-Kutta, PROMOTIC

Title

Tanks system simulation and visualization

Annotation

This work focuses on creating an application in SCADA / HMI system PROMOTIC. The application simulates time tract of water level in a system of three mutual connected water reservoirs in real time in dependence on water inflow run and the opening of water transfer valves and drawdown water valves. The simulated system is described by a system of common non-linear differential equation. A method by Runge-Kutta has been used for numeric solution.

Keywords

simulation, differential equation, Runge-Kutta, PROMOTIC

Obsah

Seznam zkratk.....	8
Seznam obrázků.....	9
Seznam tabulek.....	9
1 Úvod.....	10
2 Matematický model simulovaného systému.....	11
2.1 Model soustavy.....	11
2.1.1 Rovnice kontinuity	11
2.1.2 Bernoulliho rovnice	12
2.1.3 Schéma modelu soustavy	12
2.2 Metoda Runge-Kutta	16
3 Program pro vizualizaci.....	18
3.1 SCADA / HMI.....	18
3.2 Další systémy.....	18
3.3 PROMOTIC	19
3.3.1 Editor aplikace.....	20
3.3.2 Editor obrazů	21
3.3.3 Ostatní části	23
4 Praktická část.....	27
4.1 Návrh aplikace.....	27
4.2 Vytváření aplikace.....	27
4.2.1 Logika aplikace.....	29
4.2.2 Přiřazení datové vazby.....	31
4.2.3 Načtení / Uložení souboru	32
4.2.4 Ladění programu – INFO systém	33
4.2.5 Realizace výpočtu v časovači	34
4.2.6 Trendy.....	34
5 Závěr.....	36
Literatura	37
Příloha A.....	39

Seznam zkratek

RK4	Runge-Kuttova metoda 4. řádu
SCADA	Supervisory Control And Data Acquisition
PLC	Programmable Logic Controller
HMI	Human-Machine Interface
OOP	Object-Oriented Programming
XML	Extensible Markup Language
DDE	Dynamic Data Exchange
OPC	OLE for Process Control
HTTP	Hypertext Transfer Protocol
SVG	Scalable Vector Graphics
VB	Visual Basic
VBScript	Visual Basic Script
RS232	Recommended Standard 232
TCP / IP	Transmission Control Protocol / Internet Protokol

Seznam obrázků

Obrázek 1 - Rovnice kontinuity.....	11
Obrázek 2 - Bernoulliho rovnice	12
Obrázek 3 - Schéma modelu soustavy	13
Obrázek 4 - Funkce signum.....	15
Obrázek 5 - Ukázka možné aplikace v PROMOTICu	20
Obrázek 6 - Ukázka Editoru aplikace.....	21
Obrázek 7 - Ukázka Editoru obrazu	21
Obrázek 8 - Příklad znázornění datové vazby	22
Obrázek 9 - Použití vyfoceného snímku soustavy jako pozadí	23
Obrázek 10 - Ukázka Trendu.....	24
Obrázek 11 - Ukázka Alarmu.....	25
Obrázek 12 - Rozdělené skupiny dat.....	28
Obrázek 13 - Hlavní stránka programu	29
Obrázek 14 - Okno pro zadávání proměnných.....	30
Obrázek 15 - Konfigurace datové vazby	31
Obrázek 16 - Událost onItemAfterWrite	32
Obrázek 17 - Metoda IniFileRead	33
Obrázek 18 - INFO systém.....	34
Obrázek 19 - Ukázka trendu.....	35

Seznam tabulek

Tabulka 1 - Význam veličin	13
Tabulka 2 - Význam parametrů	14
Tabulka 3 - Geometrické rozměry nádrží.....	15

1 Úvod

Tato práce se zabývá vytvořením aplikace v programu PROMOTIC, která zajistí simulaci časového chování soustavy tří spojených nádrží v reálném čase. Aplikace dále bude ve skutečném čase (například po 1 sekundě) simulovat časové průběhy výšek hladin v závislosti na aktuálních hodnotách vstupních přítoků, otevření ventilů a hodnotách výstupních výtoků. Průtoky a otevření ventilů bude možné měnit v průběhu simulace z hlavní obrazovky. Časový průběh výšek hladin bude vykreslen do grafu s možností nahlédnutí do minulosti (posledních 5 minut).

První část je věnována popisu časového chování soustavy nádrží matematickým modelem. Matematický model je tvořen obyčejnými nelineárními diferenciálními rovnicemi. Pro numerické řešení rovnic bude použita metoda Runge-Kutta. Druhá část této práce je zaměřena na základní seznámení s prostředím PROMOTIC. Jsou zde popsány nejdůležitější části tohoto SCADA / HMI systému. Poslední část obsahuje praktické vytvoření aplikace soustavy nádrží, která je zaměřena na nejdůležitější prvky systému ohledně této práce. Zejména se jedná o možnost uložit a načíst vstupní parametry nádrží ze souboru, vytvoření přehledného grafu s možností podívání se do historie a napsání VBScriptu použitého v časovači aplikace pro počítání výšek hladin z diferenciálních rovnic.

2 Matematický model simulovaného systému

Tato kapitola vychází z bakalářské práce: „Matematický model soustavy nádrží“ Bc. Viktora Smejkal [1]. Z uvedené práce bylo převzato použité schéma modelu i matematický model popisující daný systém. Výsledný model je tvořen soustavou tří nelineárních diferenciálních rovnic.

2.1 Model soustavy

Sestavení modelu soustavy vychází z hmotnostních a energetických bilancí (rovnice kontinuity, Bernoulliho rovnice). Dále budou uvedeny základní principy, ze kterých se při odvození rovnic vychází.

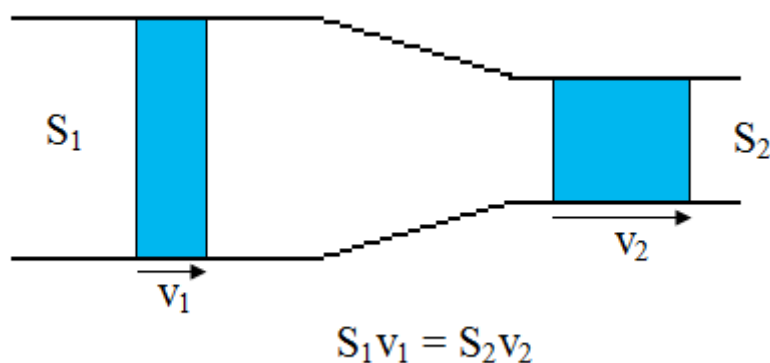
2.1.1 Rovnice kontinuity

Rovnice kontinuity pojednává o ustáleném proudění kapaliny (často se uvažuje ideální kapalina) v uzavřené trubici. Její tvar (1) popisuje vztah mezi rychlostí proudění v a průřezem S v určitém místě trubice.

$$Q_V = Sv = \text{konstanta} \quad (1)$$

kde Q_V je objemový průtok s jednotkou $\text{m}^3 \cdot \text{s}^{-1}$.

Ideální kapalina je nestlačitelná. Hustota kapaliny ρ je v každém místě trubice stejná ($\rho = \rho_1 = \rho_2$). Proto je objemový průtok Q_V v každém místě trubice stejný. Toto zjištění je zachyceno na obrázku 1. Objemový průtok Q_V je znázorněn dvěma modrými oblastmi. V místě, kde je menší obsah průřezu trubice, je větší rychlost proudění kapaliny.



Obrázek 1 - Rovnice kontinuity

Tato část textu čerpá z [2], [3].

2.1.2 Bernoulliho rovnice

Bernoulliho rovnice vyjadřuje zákon zachování mechanické energie pro ustálené proudění ideální kapaliny ve vodorovném potrubí. Celková energie proudící kapaliny je dána součtem kinetické energie a tlakové potenciální energie. Výsledná energie musí zůstat po celou dobu proudění kapaliny konstantní.

Pro kinetickou energii platí vztah:

$$E_K = \frac{1}{2}mv^2 \quad (2)$$

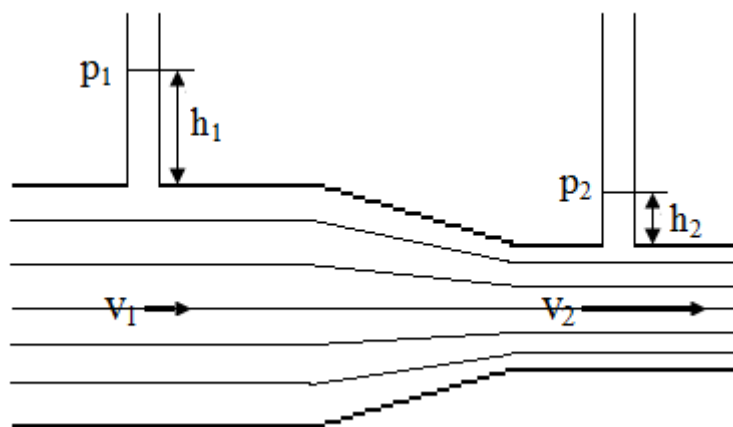
Pro tlakovou potenciální energii platí vztah:

$$E_p = Vp \quad (3)$$

Pro ideální kapalinu proudící v trubici platí zákon zachování mechanické energie:

$$E_K + E_p = \text{konstanta} \quad (4)$$

Jestliže máme na jedné straně trubici s větším obsahem průřezu, má proudící kapalina na této straně menší rychlost, ale větší tlak. Na druhé straně má trubice menší obsah průřezu. Tady je rychlost proudění větší, ale zase je zde menší tlak.



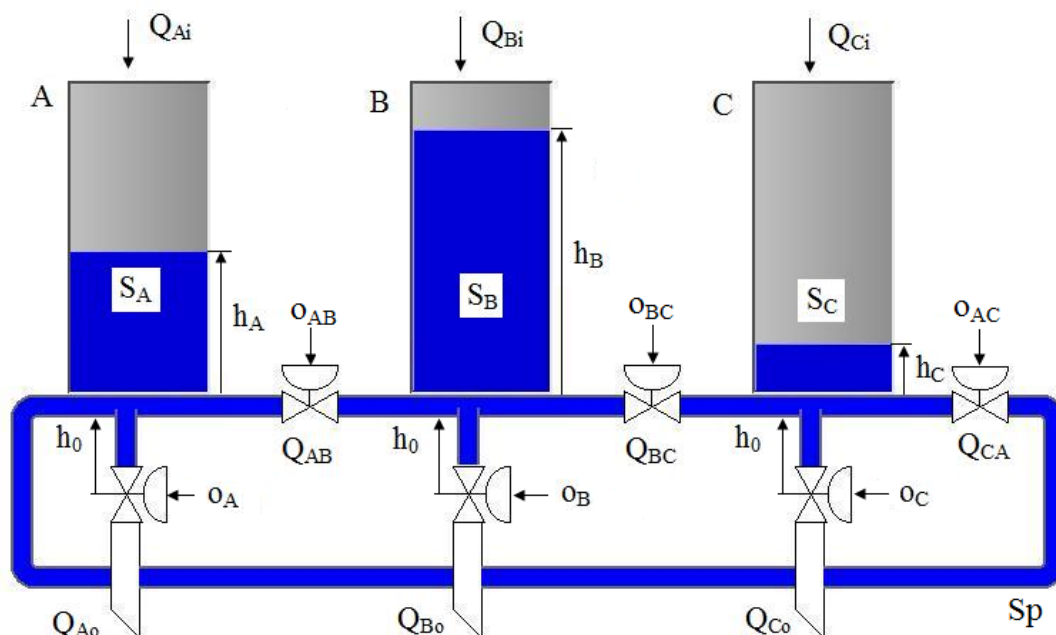
Obrázek 2 - Bernoulliho rovnice

Podrobnější vysvětlení naleznete na [4], [5].

2.1.3 Schéma modelu soustavy

Schéma modelu soustavy propojených nádrží vidíme na obrázku 3 níže. Tento obrázek popisuje tři navzájem propojené nádrže. Nádrže mohou mít jakýkoliv tvar obsahu průřezu, ale předpokládáme, že v celé výšce hladiny je obsah průřezu konstantní. Dále jsou spojeny propojovacím potrubím s odtokovými a průtokovými ventily. Nahoře jsou

naznačené přítoky, které by byly realizovány pomocí ventilů. Jednotlivý význam veličin a parametrů je popsán v tabulkách 1 a 2.



Obrázek 3 - Schéma modelu soustavy

Tabulka 1 - Význam veličin

Veličina	Rozměr	Význam
Q_{Ai}	$m^3 \cdot s^{-1}$	přítok do nádrže A
Q_{Bi}	$m^3 \cdot s^{-1}$	přítok do nádrže B
Q_{Ci}	$m^3 \cdot s^{-1}$	přítok do nádrže C
Q_{A0}	$m^3 \cdot s^{-1}$	výtok z nádrže A
Q_{B0}	$m^3 \cdot s^{-1}$	výtok z nádrže B
Q_{C0}	$m^3 \cdot s^{-1}$	výtok z nádrže C
Q_{AB}	$m^3 \cdot s^{-1}$	průtok mezi nádržemi A a B
Q_{BC}	$m^3 \cdot s^{-1}$	průtok mezi nádržemi B a C
Q_{CA}	$m^3 \cdot s^{-1}$	průtok mezi nádržemi C a A
o_A	bez rozměru	otevření výpustního ventilu z nádrže A (nabývá hodnot 0 - 1)
o_B	bez rozměru	otevření výpustního ventilu z nádrže B
o_C	bez rozměru	otevření výpustního ventilu z nádrže C
o_{AB}	bez rozměru	otevření přepouštěcího ventilu z nádrží A a B
o_{BC}	bez rozměru	otevření přepouštěcího ventilu z nádrží B a C
o_{CA}	bez rozměru	otevření přepouštěcího ventilu z nádrží C a A

Tabulka 2 - Význam parametrů

Parametr	Rozměr	Význam
S_A	m^2	plocha nádrže A
S_B	m^2	plocha nádrže B
S_C	m^2	plocha nádrže C
S_p	m^2	plocha přepouštěcího potrubí
s_v	m^2	plocha vypouštěcího ventilu
sprep	m^2	plocha přepouštěcího ventilu

Systém je dále popsán soustavou rovnic (5) – (13). Blíže viz [1].

$$Q_{Ao} = s_A o_A (2 - o_A) \sqrt{\frac{2(h_A + h_0)g}{1 - \left[\frac{s_A o_A (2 - o_A)}{S_A} \right]^2}} \quad (5)$$

$$Q_{Bo} = s_B o_B (2 - o_B) \sqrt{\frac{2(h_B + h_0)g}{1 - \left[\frac{s_B o_B (2 - o_B)}{S_B} \right]^2}} \quad (6)$$

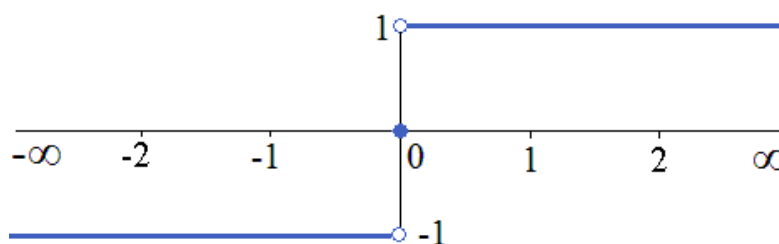
$$Q_{Co} = s_C o_C (2 - o_C) \sqrt{\frac{2(h_C + h_0)g}{1 - \left[\frac{s_C o_C (2 - o_C)}{S_C} \right]^2}} \quad (7)$$

$$Q_{AB} = \text{sign}(h_A - h_B) s_{AB} o_{AB} (2 - o_{AB}) \sqrt{\frac{2|h_A - h_B|g}{1 - \left[\frac{s_{AB} o_{AB} (2 - o_{AB})}{S_p} \right]^2}} \quad (8)$$

Proměnné s_{AB} , s_{BC} , s_{CA} ze vztahů (8) až (10) jsou plochy přepouštěcích ventilů. Všechny jsou stejně velké, proto jsem je v tabulce 2 i programu nahradil proměnnou sprep.

Funkce signum (sign) je matematická funkce, která jakémukoliv číslu přiřadí hodnotu 1, -1 nebo 0. Funkce signum je znázorněna na následujícím obrázku 4.

V intervalu: - $(-\infty, 0)$ má funkce signum hodnotu -1
- 0 nabývá hodnoty 0
- $(0, \infty)$ má funkce signum hodnotu 1



Obrázek 4 - Funkce signum

$$Q_{BC} = \text{sign}(h_B - h_C) s_{BC} o_{BC} (2 - o_{BC}) \sqrt{\frac{2|h_B - h_C|g}{1 - \left[\frac{s_{BC} o_{BC} (2 - o_{BC})}{S_p} \right]^2}} \quad (9)$$

$$Q_{CA} = \text{sign}(h_C - h_A) s_{CA} o_{CA} (2 - o_{CA}) \sqrt{\frac{2|h_C - h_A|g}{1 - \left[\frac{s_{CA} o_{CA} (2 - o_{CA})}{S_p} \right]^2}} \quad (10)$$

$$\frac{dh_A}{dt} = \frac{Q_{Ai} - Q_{Ao} - Q_{AB} + Q_{CA}}{S_A} \quad (11)$$

$$\frac{dh_B}{dt} = \frac{Q_{Bi} - Q_{Bo} - Q_{BC} + Q_{AB}}{S_B} \quad (12)$$

$$\frac{dh_C}{dt} = \frac{Q_{Ci} - Q_{Co} - Q_{CA} + Q_{BC}}{S_C} \quad (13)$$

Konstanty použité pro výpočty S_A , S_B , S_C obsažené ve vztazích (5) – (13) jsou vysvětleny v následující tabulce 3.

Tabulka 3 - Geometrické rozměry nádrží

Symbol	Hodnota	Rozměr	Význam
d_A	0.1	m	průměr válcové nádrže A
d_B	0.2	m	průměr válcové nádrže B
d_C	0.3	m	průměr válcové nádrže C
d	0.015	m	průměr přepouštěcího potrubí mezi nádržemi
d_v	0.005	m	průměr max. průřezu vypouštěcího ventilu
d_p	0.0075	m	průměr max. průřezu přepouštěcího ventilu
h_0	0.1	m	výška dna nádrže nad ventilem

Výše popsané rovnice (5) až (10) jsou odvozeny za předpokladu, že průtok kapaliny ventilem nezáleží na směru průtoku kapaliny.

Systém máme popsán soustavou obyčejných nelineárních diferenciálních rovnic. Řešením je zjištění časového průběhu výstupních veličin v závislosti na časovém průběhu vstupních veličin. Nemůžeme zde použít analytické řešení, protože pro uvedené nelineární rovnice neexistuje. Z tohoto důvodu použijeme numerické řešení. Byla vybrána nejčastěji používaná metoda Runge–Kutta.

2.2 Metoda Runge-Kutta

Runge-Kuttovy metody jsou často užívané jednokrokové metody. Nejčastěji se používá Runge-Kuttova metoda 4. řádu (RK4) [6]. Objeví-li se v literatuře řešení obyčejné diferenciální rovnice prvního řádu metodou Runge-Kutta, myslí se tím většinou právě tato „standardní“ metoda 4. řádu RK4. Pro diferenciální rovnici 1. řádu ve tvaru $\frac{dy}{dx} = f(x, y)$ pro časový interval T , jsou výpočetní vztahy RK4 dány rovnicemi (14) až (18).

$$y_{n+1} = y_n + \frac{1}{6}T(k_1 + 2k_2 + 2k_3 + k_4) \quad (14)$$

$$k_1 = f(x_n, y_n) \quad (15)$$

kde $x_n = x(t_0 + nT)$

$$x_{n+1} = x(t_0 + (n+1)T)$$

$$k_2 = f\left(x_n + \frac{1}{2}T, y_n + \frac{1}{2}Tk_1\right) \quad (16)$$

$$k_3 = f\left(x_n + \frac{1}{2}T, y_n + \frac{1}{2}Tk_2\right) \quad (17)$$

$$k_4 = f(x_n + T, y_n + Tk_3) \quad (18)$$

Z uvedených vztahů vyčteme, že další vypočítaná hodnota závisí na již spočítané hodnotě, tedy následující hodnotu vypočítáme pomocí předešlé hodnoty.

Pro soustavu obyčejných diferenciálních rovnic prvního řádu popsaných rovnicemi (11) až (13) můžeme RK4 konkretizovat ve vektorovém zápisu jako:

$$\mathbf{h}(t+T) = \mathbf{h}(t) + \frac{T}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (19)$$

$$\text{kde } \mathbf{h} = \begin{bmatrix} \mathbf{h}_A \\ \mathbf{h}_B \\ \mathbf{h}_C \end{bmatrix} \quad (20)$$

$$\mathbf{k}_1 = \mathbf{f}(x_n, \mathbf{h}) \quad (21)$$

$$\mathbf{k}_2 = \mathbf{f}(x_n + \frac{1}{2}T, \mathbf{h} + \frac{1}{2}T\mathbf{k}_1) \quad (22)$$

$$\mathbf{k}_3 = \mathbf{f}(x_n + \frac{1}{2}T, \mathbf{h} + \frac{1}{2}T\mathbf{k}_2) \quad (23)$$

$$\mathbf{k}_4 = \mathbf{f}(x_n + T, \mathbf{h} + T\mathbf{k}_3) \quad (24)$$

$$\text{kde } \mathbf{f}(\mathbf{h}, \mathbf{Q}_{Ai}, \mathbf{Q}_{Bi}, \mathbf{Q}_{Ci}, o_A, o_B, o_C, o_{AB}, o_{BC}, o_{CA}) = \begin{bmatrix} dh_A / dt \\ dh_B / dt \\ dh_C / dt \end{bmatrix} \quad (25)$$

3 Program pro vizualizaci

Pro vytvoření mé aplikace zabývající se simulací vodních nádrží, byl vybrán program PROMOTIC. PROMOTIC je komplexní SCADA systém, který nám dovoluje velké množství nastavení pro tvorbu aplikací. Informace obsažené v této části jsem převzal z nápovědy systému PROMOTIC a www stránek firmy MICROSYS [7].

3.1 SCADA / HMI

SCADA je zkratka utvořená ze slov "Supervisory Control And Data Acquisition" [8], [9]. V překladu toto sousloví znamená „řízení kontroly a získávání dat“. Tento pojem se často používá v oblasti řídicích průmyslových systémů např. monitorování a kontrola průmyslových zařízení nebo procesů. Jedná se tedy o systém shromažďující data z čidel v reálném čase a tato data posílá na centrální počítač, kde se dále zpracovávají. SCADA není plnohodnotný řídicí systémem, ale zaměřuje se spíše na oblast dispečera. Může se jednat o aplikaci (software), která funguje nad skutečným řídicím systémem. Tyto systémy jsou zpravidla programovatelné logické automaty (PLC).

HMI, anglicky "Human-Machine Interface", je modifikace původního termínu MMI (rozhraní člověk - stroj), který se používal dříve. HMI je uživatelské rozhraní mezi člověkem a strojem (zařízením) a je to většinou jeden místní stroj nebo zařízení [9], [10].

Rozhraní je velmi často graficky přívětivé pro uživatele. Dává operátorovi vědět informace o různých stavech procesů, umožňuje mu měnit hodnoty a mít nad systémem dokonalý přehled a kontrolu. Dále dává možnost zobrazit grafické průběhy požadovaných veličin, v tomto odvětví nazývané trendy. Tato zařízení mohou být spojována do jednoho kontrolního zařízení.

3.2 Další systémy

Kromě PROMOTICu se u nás v České republice často používají systémy Reliance [11] nebo Control Web [12]. Oba systémy jsou samozřejmě z kategorie SCADA / HMI.

Reliance je program určený pro monitorování a ovládání průmyslových technologií a automatizaci budov. Využívá se v mnoha oborech. Jsou to například obory: plynárenství (ve firmě E.ON Česká republika je využit pro dispečerský systém plynárenské sítě v jižních Čechách), potravinářství (Budějovický Budvar, kde monitoruje a ovládá technologie chlazení v pivovaru), řízení inteligentních budov a v dalších odvětvích [13].

Control Web se v současnosti nejvíce používá jako http server, jehož možností je generování dynamického obsahu www stránek z běžícího aplikačního programu [14].

Další možností jsou zahraniční průmyslové systémy Wonderware [15], CitectSCADA [16] a jistě mnoho dalších.

3.3 PROMOTIC

A nyní konkrétně o PROMOTICu. Tento program je už vyvíjen zhruba 20 let českou firmou MICROSYS. V jeho počátcích se jednalo o PROMOTICTM, který byl vytvořen v roce 1991 a byl uzpůsoben pro operační systém MS-DOS. Tento vizualizační a řídicí systém se uchytil na trhu a díky němu bylo vyrobeno několik stovek aplikací, nejen v České republice, ale i v zahraničí.

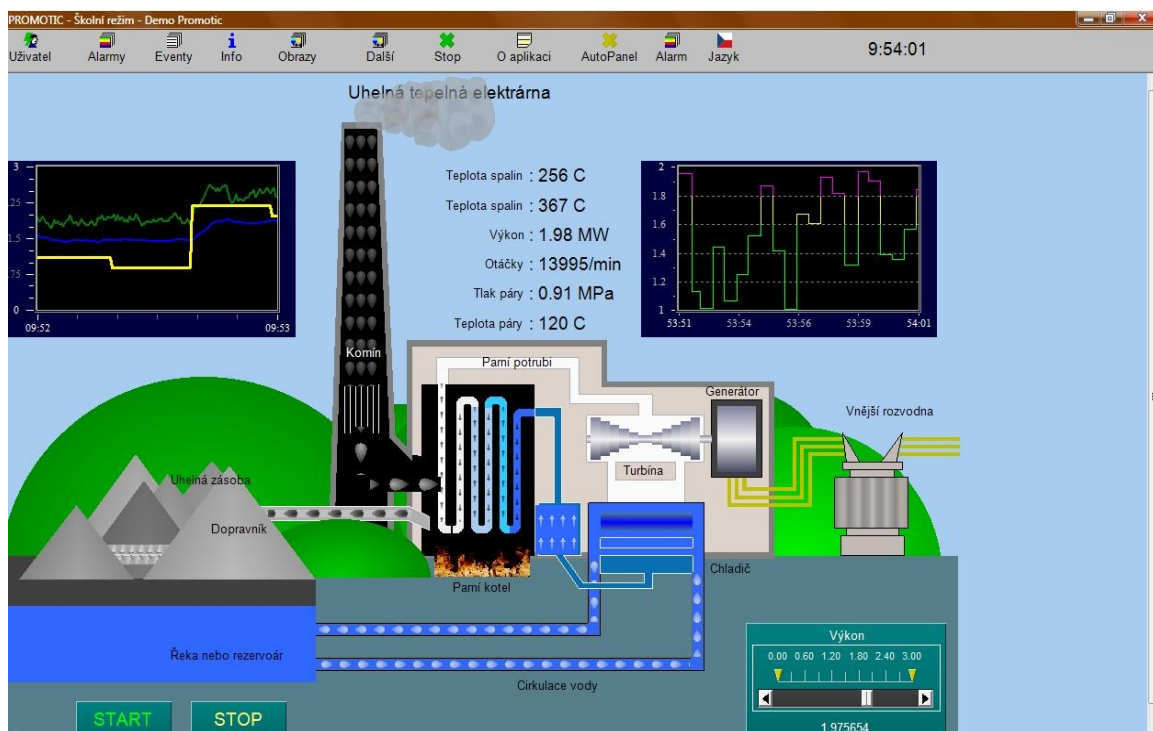
Aplikace byly nasazeny především v oblasti energetiky (tepelné a vodní elektrárny), hutního a chemického průmyslu (žihací pece, koksovny), měření a regulace odběru energie, ale třeba také v potravinářském průmyslu a v ekologii (emisní monitoring, ČOV¹). Dá se tedy říci, že tento systém je vhodný pro jakoukoliv oblast průmyslu.

V pozdějších letech byly vytvořeny novější verze. Jedná se o verze PROMOTIC 4, PROMOTIC 2000 (rok 2000), PROMOTIC 6, PROMOTIC 7 až po dnešní aktuální verzi PROMOTIC 8.0.8 (PROMOTIC 8 vytvořen v roce 2009). Tato poslední verze je vhodná pro novější operační systémy Windows 7/Vista/XP. Je zde přehledně zpracované grafické prostředí. Části systému jsou logicky rozděleny a vše je doplněno obsáhlou nápovědou v českém jazyce s různými příklady a ukázkami použití.

PROMOTIC je založen na objektově orientovaném programování (OOP - Object-oriented programming). PROMOTIC má otevřenou architekturu a podporuje většinu standardních softwarových rozhraní. Ať už se jedná o nejrozšířenější databázové systémy (Access, MS SQL), či jiná rozhraní např. XML, ActiveX, DDE, OPC.

S tímto programem můžeme také své aplikace ovládat i s pomocí internetového prohlížeče. Celá komunikace je postavena na HTTP protokolu, který není překážkou pro většinu firewallů, a díky propojení s komunikačními servery máme skoro stejné možnosti obsluhy aplikace, jako na PC, kde běží reálně naše aplikace. Na další straně na obrázku 5 vidíme, jak by mohla vypadat ukázka aplikace v tomto prostředí. Celá tato podkapitola vychází ze [17] a [18].

¹ Čistička odpadních vod



Obrázek 5 - Ukázka možné aplikace v PROMOTICu

PROMOTIC se skládá z těchto částí:

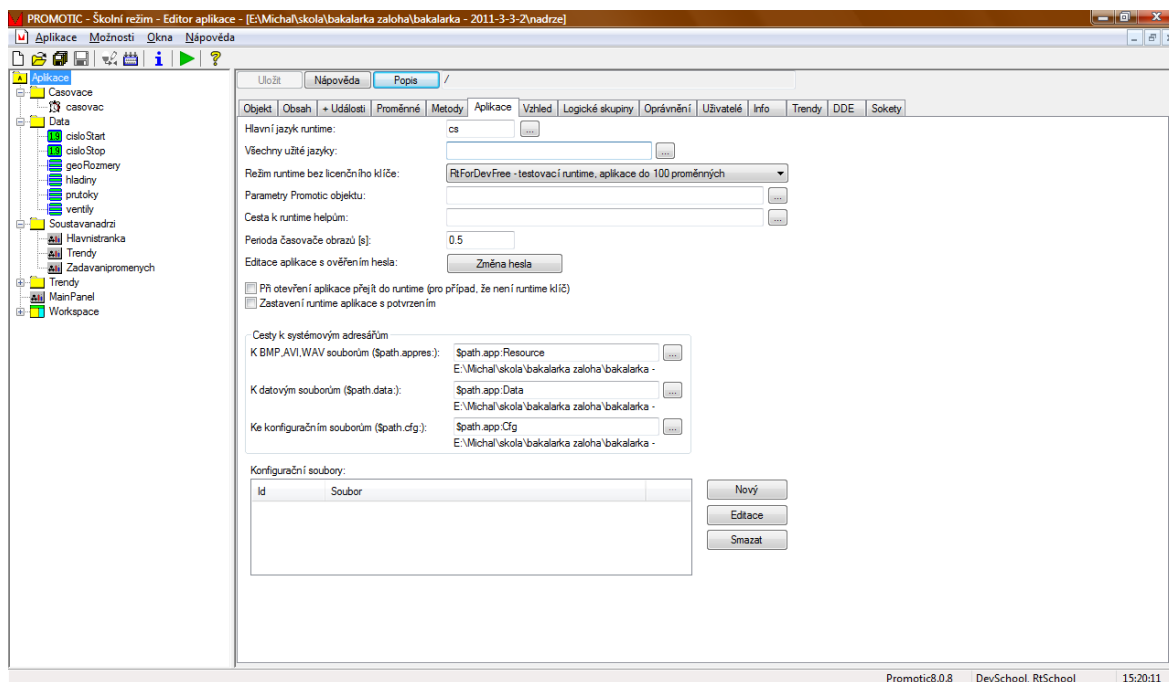
- Editor aplikace,
- Editor obrazů.

Program dále obsahuje část zabývající se trendy, alarmy a eventy², část ohledně komunikace s externími zařízeními, jazykové verze, internet.

3.3.1 Editor aplikace

Editor aplikace (obrázek 6) je základní část pro tvorbu aplikace v systému PROMOTIC. Zde vytváříme stromovou strukturu objektů, které postupně upravujeme (měníme jim jejich nastavení). Objektům můžeme přiřazovat datové typy (například double, string). Každý objekt má událost, kam zapisujeme algoritmy (scripty). Toho využíváme při přístupu k vlastnostem objektů nebo metodám. Pro zápis již zmíněných algoritmů slouží jazyk Visual Basic, přesněji řečeno jeho scriptová úprava VBScript. To nám umožní vytvořit aplikaci přesně podle našich představ. Nejsme vázáni pouze na určitou oblast zaměření, ale máme nástroj pro neomezené možnosti vývoje aplikace.

² události

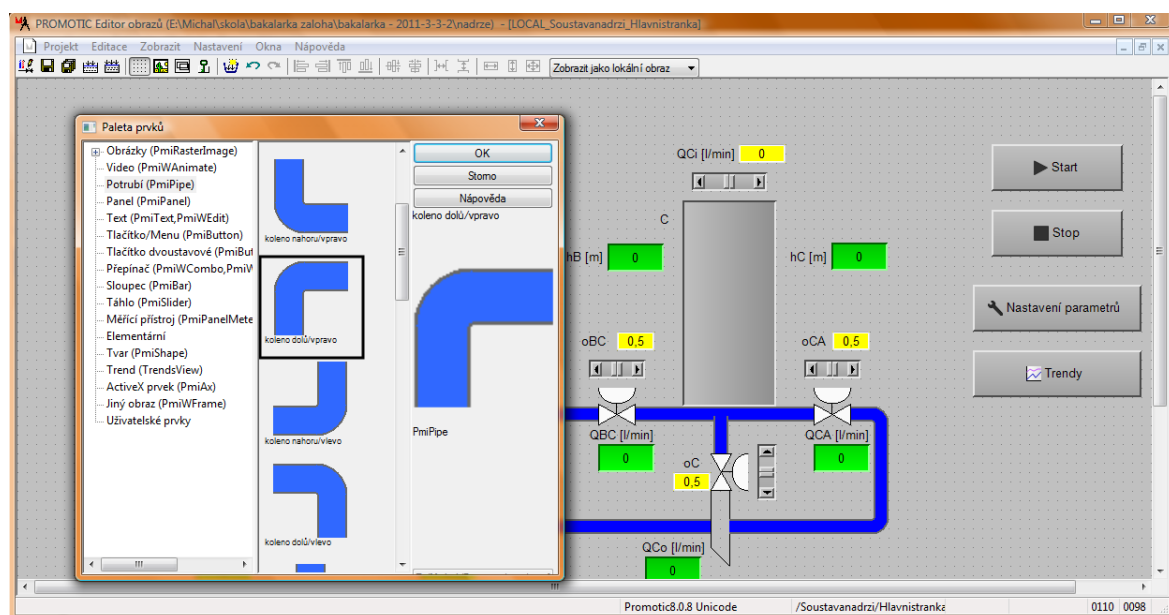


Obrázek 6 - Ukázka Editoru aplikace

Do této části se řadí ještě INFO systém. Je to informační a diagnostický systém. INFO systém slouží k odladění aplikace. Umožňuje nám prohlížení všech proměnných a důležitých informací. Díky části PROMOTIC Web můžeme tento systém také využít při vzdáleném ladění aplikace přes síť internet.

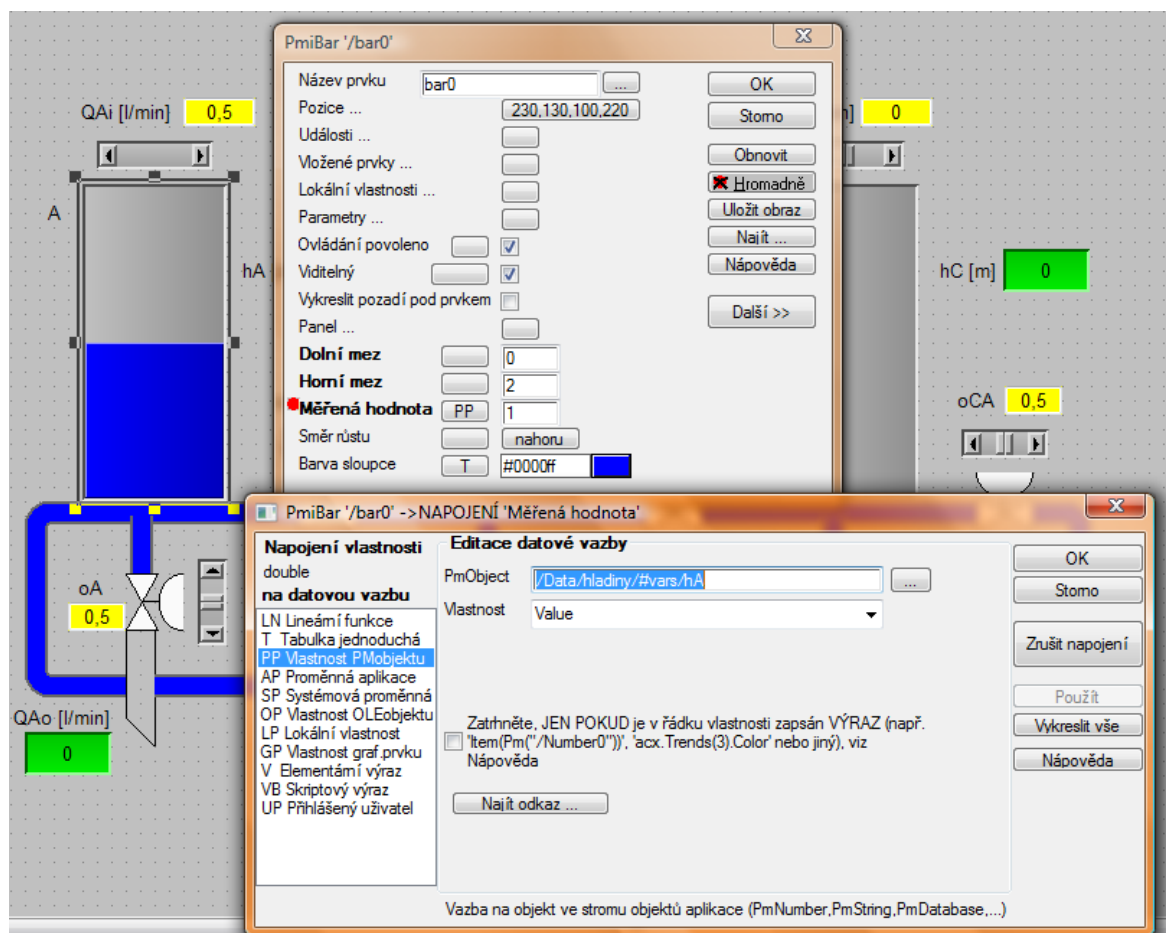
3.3.2 Editor obrazů

Slouží k vytváření grafického obrazu (obrázek 7) pomocí grafických prvků.



Obrázek 7 - Ukázka Editoru obrazu

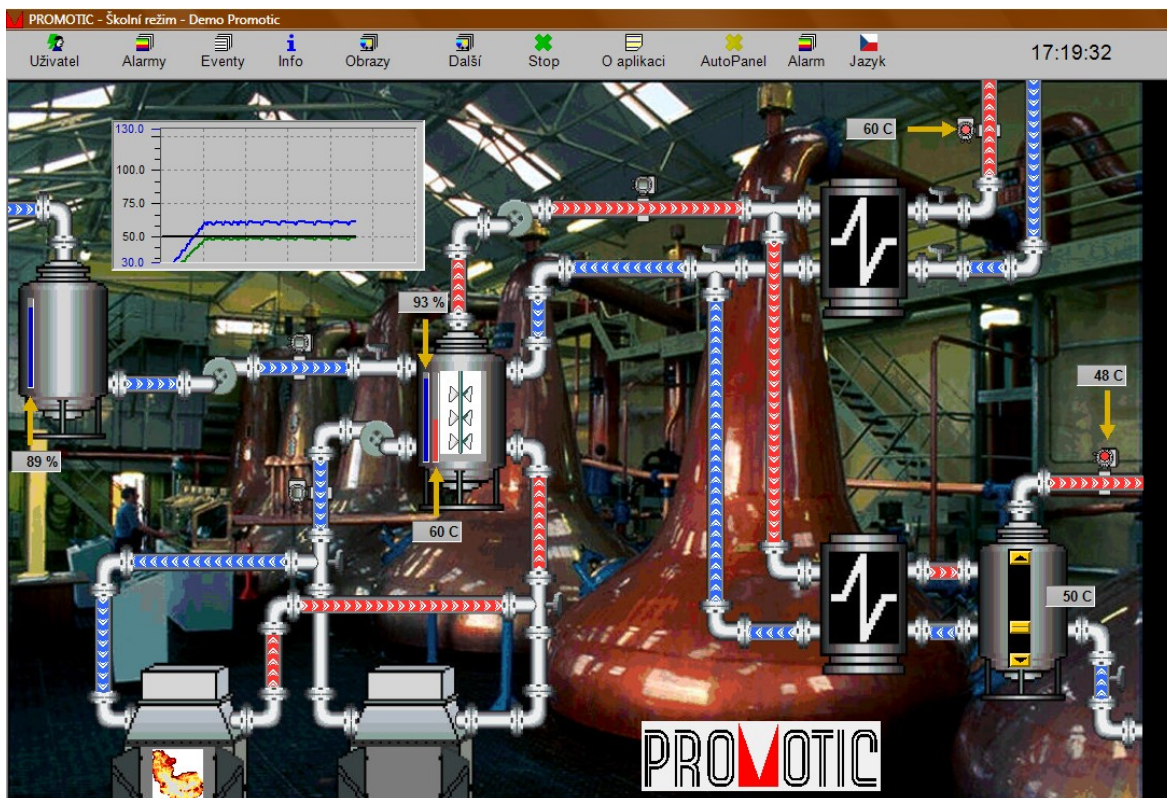
Pomocí těchto grafických objektů můžeme vytvořit libovolný vzhled aplikace, ať se jedná o zadávání dat, řízení nebo o vizualizaci dat v trendech. U těchto objektů můžeme některé jejich vlastnosti napojit datovou vazbou. Mezi tyto vlastnosti patří velikost, hodnota a barva. Název datová vazba vyjadřuje provázání proměnné s vlastností grafického prvku. Například, když klikneme na jeden z dvou trojúhelníků u táhla, tak se nám změní hodnota o jednu jednotku. Na obrázku 8 vidíme další možnost. Máme zde svázanou výšku hladiny h_A s objektem bar0 (sloupec). Jestliže se nám změní výška h_A , tak se nám tato změna projeví na zobrazení výšky hladiny v nádrži.



Obrázek 8 - Příklad znázornění datové vazby

Pro projektanta je připraveno několik předdefinovaných typů oken podobných standardním oknům z Windows, ale projektant si může vytvořit vlastní okna s pevnou velikostí, pevnou pozicí, bez okenních lišt. Dále má možnost si předvolit, kolik oken může být současně otevřeno.

Editor obrazů také podporuje tvorbu vlastních grafických prvků. V některém z programů vektorové SVG grafiky (Illustrator, CorelDraw nebo Inkscape) si tedy můžeme nakreslit vlastní grafický prvek, který následně využijeme při tvorbě naší aplikace. Jak vidíme na obrázku 9, můžeme použít jako pozadí pro naši aplikaci nakreslený obrázek nebo skutečně vyfocený snímek. To usnadní práci operátorovi v ovládání aplikace. Dosáhneme tím také větší přehlednosti a lepší orientace v programu.



Obrázek 9 - Použití vyfoceného snímku soustavy jako pozadí

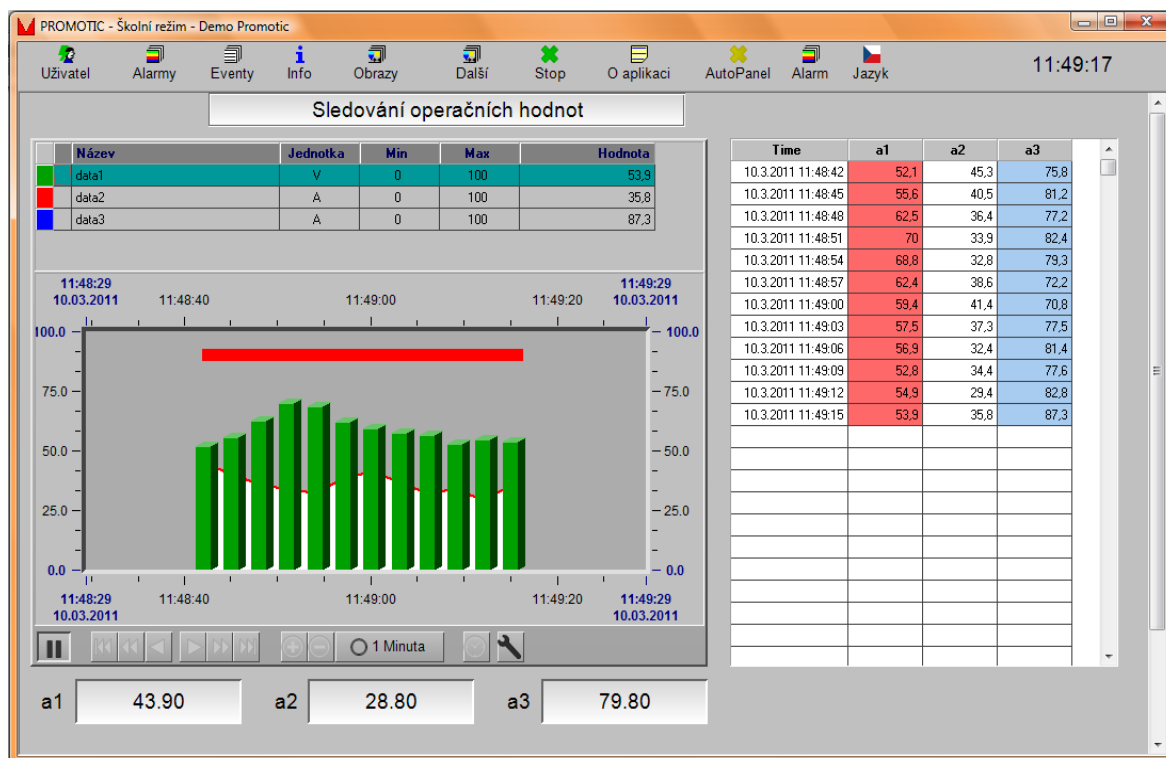
3.3.3 Ostatní části

Trendy

Součástí PROMOTICu je systém trendů. Tento výkonný nástroj slouží k tomu, aby ukládal hodnoty námi zvolených proměnných buď na pevný disk v počítači, nebo do paměti. Hodnoty vybraných veličin jsou ukládány s časovým údajem, takže jsme později schopni zpětně dohledat naměřená data v daný okamžik. PROMOTIC tedy umožňuje zobrazit historii naměřených dat.

Tyto hodnoty můžeme zobrazovat normálně ve formě grafu jako časové průběhy veličin (obrázek 10) nebo v přehledné tabulce.

Data můžeme ukládat v několika různých formátech. Máme na výběr z MS SQL, MySQL, binární soubor a Access.

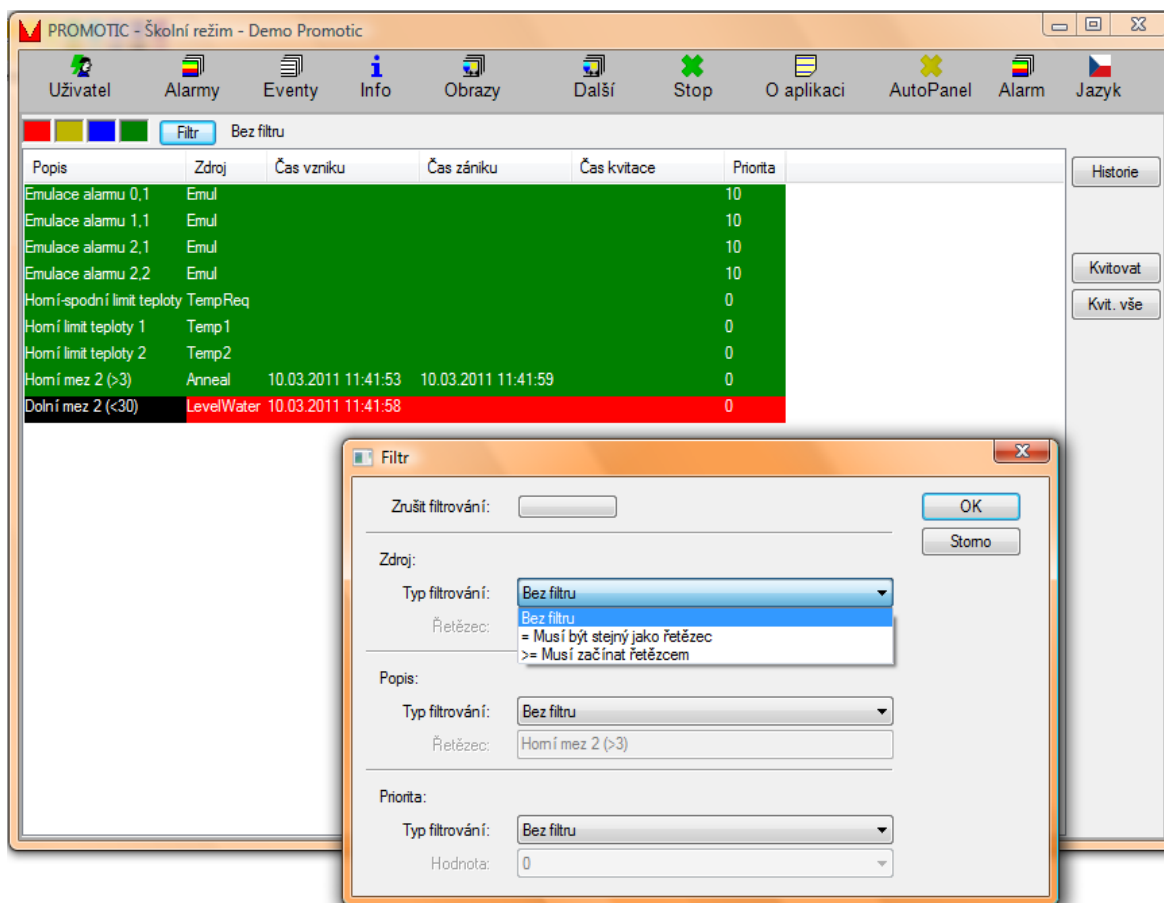


Obrázek 10 - Ukázka Trendu

Alarmy a Eventy

Nyní blíže popíšu alarmy a eventy. Jsou to částí systému PROMOTIC, které zachycují a spravují technologické děje.

Alarm (obrázek 11) je zachycený děj (např. porucha ventilu, přetečení nádrže). Je proto potřeba, aby se o něm operátor dozvěděl. Alarm nám dává upozornění, které by za normálních okolností nemělo nastat. Může se jednat o překročení mezí dané veličiny, ztrátu komunikace s daným zařízením, poruchu čerpadla a podobně. PROMOTIC obsahuje systém alarmů, který zaznamenává alarmové stavy v alarmních skupinách, a tyto stavy archivuje. Nabízí prohlížení alarmů se zvolenými filtracemi i zvukovým signálem. Později máme možnost prohlížet si jejich historii.



Obrázek 11 - Ukázka Alarmu

Event je jednodušší děj než alarm. Může to být třeba operátorská událost, kterou je potřeba pouze zaznamenat. Event je zpráva o provedení určité činnosti (zahájení, ukončení určitého procesu). Tato zpráva je ukládána na disk do souboru pro pozdější prohlížení. Oproti alarmům, se eventy pouze archivují.

Komunikace s okolím

Jestliže vytváříme aplikaci, která požaduje získávat data z externího zdroje, systém PROMOTIC nám to umožňuje. Máme-li aplikaci zaměřenou třeba na monitorování nějakého technologického procesu, většinou potřebujeme získat data ze zdroje informace nebo naopak uložit data do tohoto zdroje. Jako zdroj těchto dat může být použita databáze dříve naměřených hodnot, zásuvná karta do počítače, PLC automat nebo i server připojený do sítě internetu. Zařízení k sobě bývají nejčastěji připojena přes sériovou linku (na PC COM1, COM2 standard RS232) nebo přes ethernet (UTP kabeláž).

PROMOTIC obsahuje mnoho komunikačních ovladačů pro automaty (namátkou třeba ovladače pro Siemens Static, Telemecanique od firmy Schneider-Electric, Direc Logic firmy Koyo Electronics a mnoho dalších), protokoly pro přenos v sítích GSM a radiových sítích, komunikační protokol Modbus nebo M-Bus.

Pro vytváření aplikací v síti internetu (intranetu) slouží komunikační rozhraní s protokoly HTTP a TCP/IP.

Bezpečnost

PROMOTIC nabízí „zaheslování“ chráněného obsahu aplikace, do kterého se dostane pouze oprávněný uživatel. Normální uživatel se dostane jen do spuštěné aplikace, nedostane se do editoru aplikace. Jedná se o případy, kdy si firma udržuje tzv. „know how“.

Dále také obsahuje bezpečnostní komponentu SafeWDog (WatchDog³), která slouží pro zvýšení stability a bezpečnosti běžící aplikace. V některých případech si přejeme, aby aplikace v PROMOTICu běžela nepřetržitě, ale je umístěna na vzdáleném místě, kam obsluha dochází jen minimálně. Kdyby nastala porucha, chceme, aby se systém sám restartoval. Pro takové účely slouží již zmiňovaný WatchDog. Jedná se o malý program, který běží na pozadí a hlídá naši aplikaci. Když se aplikace nečekaně „zasekne“, WatchDog tuto aplikaci restartuje, nebo restartuje rovnou celý operační systém.

V neposlední řadě má projektant možnost zablokovat kritické klávesy ve Windows.

Internet

PROMOTIC dovoluje prohlížení i ovládání aplikace prostřednictvím standardního webového prohlížeče (Internet Explorer, Firefox, Opera). Zobrazuje obrazy v podobě plně automaticky generovaných dynamických HTML stránek, včetně zpětného ovládání a dodatečných zásahů. Tyto dynamické stránky obrazů se automaticky vytvářejí ve vývojovém prostředí při překladu obrazu a jsou uchovány v datovém souboru aplikace. To znamená, že z aplikace vytvořené a později ovládané na jednom PC, můžeme velmi snadno udělat aplikaci síťovou. Na vzdáleném PC jsme schopni sledovat v reálném čase průběhy trendů, zobrazovat alarmy a eventy dané aplikace. Celý přenos je postaven na HTTP protokolu.

Díky části Web server zajišťuje PROMOTIC přenosy dat a jejich zabezpečení. Web server je aplikace, která poskytuje webové stránky (soubory HTML) na internet, které můžeme poté jednoduše prohlížet.

Jazykové verze

Ještě bych zmínil jazykové verze programu. PROMOTIC používá znakovou sadu Unicode, díky tomu lze vytvářet aplikace v libovolném národním jazyce. Uvedme například češtinu, angličtinu, francouzštinu a ruštinu.

³ hlídací pes

4 Praktická část

Praktická část této práce popisuje vytvoření aplikace pro simulaci a vizualizaci soustavy nádrží v systému PROMOTIC.

4.1 Návrh aplikace

Aplikace má simulovat v reálném čase chování soustavy spojených nádrží. Aplikace by šla realizovat v jakémkoliv SCADA / HMI systému (např. Reliance, Wonderware), ale pro moji práci byl vybrán systém PROMOTIC, protože Univerzita Pardubice na něj vlastní školní licenci. V tomto systému jsou už některé nástroje pro tvorbu takto zaměřené aplikace předpřipraveny. Hlavní myšlenka této práce je ta, že máme vypočítat následující hodnoty výšek hladin v nádržích z hodnot minulých. Numerický výpočet soustavy rovnic závisí na zadaných vstupních hodnotách přítoků, otevření ventilů a geometrických rozměrech nádrží, ale i na minulých hodnotách výšek hladin. Výpočet se provádí metodou Runge-Kutta RK4 jako VBScript, který voláme v události objektu časovače. Časovač má předem přesně daný pevný krok, a to 1 sekundu. Řešení, které popisuje matematický model soustavy nádrží, je upraveno tak, aby popisovalo skutečné chování daného systému.

Za běhu programu bude možné měnit vstupní přítoky a výstupní odtoky nádrží, dále také velikosti otevření ventilů mezi jednotlivými nádržemi. V průběhu simulace nepůjdou měnit geometrické rozměry nádrží. Tyto proměnné půjdou měnit pouze v zastavené aplikaci. Pro proměnné musíme nastavit správné datové vazby. Před spuštěním aplikace budeme mít možnost načíst hodnoty přítoků a rozměry nádrží. Tento úkol vyřešíme pomocí metody `IniFileRead` a předem definovaných INI souborů. Kdykoliv za chodu aplikace můžeme hodnoty přítoků a geometrické rozměry nádrží uložit do souboru. K tomu použijeme metody `IniFileWrite`.

Časové průběhy výšek hladin budou přehledně zobrazeny do grafu. Systém PROMOTIC má pro tuto možnost implementovanou část souborů Trendů. Trendování je ukládání hodnot námi zvolených veličin na pevný disk do počítače s časovým údajem. Posléze si můžeme tyto hodnoty nechat zobrazit graficky nebo tabulkově jako časové průběhy veličin. Jelikož ukládáme i časovou stopu, máme možnost prohlížení i historie.

4.2 Vytváření aplikace

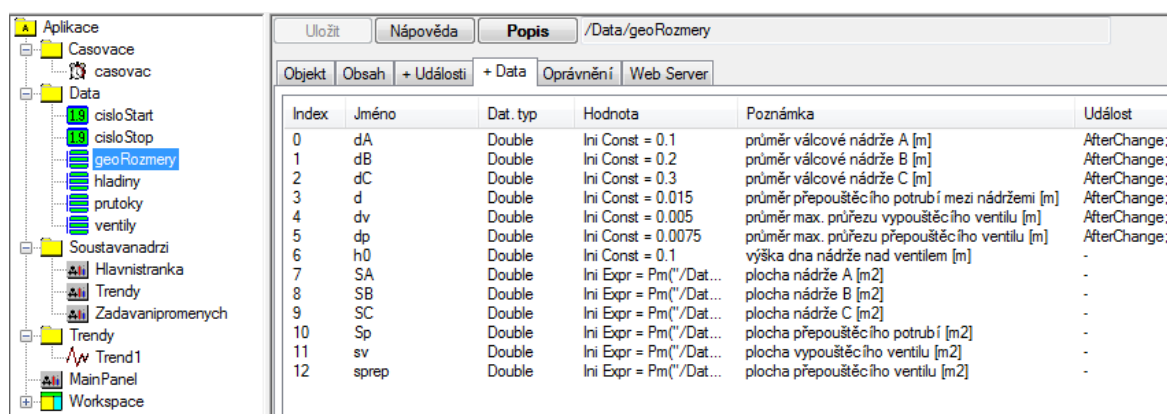
Vytváření aplikace můžeme rozdělit na tři základní části. První část se zabývá vytvářením stromové struktury objektů a jejich následnou úpravou. Nazývá se Editor aplikace. Druhá část se jmenuje Editor obrazů. Slouží k vytváření grafického obrazu pomocí grafických prvků (PmPanelů).

Třetí oblast vlastně „vdechne“ aplikaci život. Definujeme zde to, co má objekt udělat každou sekundu. Co se stane, když uživatel klikne na určité tlačítko, nebo stiskne nějakou klávesu.

V praxi je toto rozdělení na tři části jen formální. Nemusíme ho přesně dodržovat. Často se druhá a třetí část prolínají nebo určitá část zcela chybí. Podrobný popis vytváření aplikace si můžete přečíst v nápovědě programu, zejména v kapitole Základní učebnice, ze které jsem i já čerpal.

Ve stromové hierarchii vytvoříme složku *Soustavanadrzi* pro obrazovou editaci a složku *Data*, kde budou uchovány všechny objekty *PmData* pro proměnné. Ve složce *Soustavanadrzi* mám vytvořené 3 objekty *PmPanel*. Každý z těchto objektů znázorňuje vizuální stránku aplikace. Na 1. stránce nazvané Hlavní stránka (obrázek 13) je znázorněna soustava tří propojených nádrží s prvky pro ovládání aplikace. Jsou tu 3 vodní nádrže, propojovací potrubí, ventily, táhla, tlačítka pro řízení a samozřejmě také číselné hodnoty znázorňující výšky hladin, přítoky a odtoky nádrží. Celá druhá stránka graficky zobrazuje časové průběhy výšek hladin. Poslední stránka obsahuje editační okna pro zadávání hodnot. Jsou zde i tlačítka pro uložení a načtení hodnot do/ze souboru. Tyto vizuální stránky upravujeme v Editoru obrazu. Práce v této části je velice jednoduchá. Jen si vždycky vybereme potřebný prvek z Palety prvků, umístíme ho na pracovní plochu, nastavíme u něj potřebné vlastnosti a přiřadíme k němu požadovanou vazbu.

Jak vidíme na obrázku 12, data jsem rozdělil do logických celků. Data, ve významu parametrů modelu, jsou skupina obsahující geometrické rozměry nádrže (*geoRozmery*) a data týkající se časově proměnných veličin na skupiny obsahující informace o hladinách (*hladiny*), informace o průtocích (*prutoky*) a skupinu obsahující informace o ventilech (*ventily*).



Index	Jméno	Dat. typ	Hodnota	Poznámka	Událost
0	dA	Double	Ini Const = 0.1	průměr válcové nádrže A [m]	AfterChange;
1	dB	Double	Ini Const = 0.2	průměr válcové nádrže B [m]	AfterChange;
2	dC	Double	Ini Const = 0.3	průměr válcové nádrže C [m]	AfterChange;
3	d	Double	Ini Const = 0.015	průměr přepouštěcího potrubí mezi nádržemi [m]	AfterChange;
4	dv	Double	Ini Const = 0.005	průměr max. průřezu vypouštěcího ventilu [m]	AfterChange;
5	dp	Double	Ini Const = 0.0075	průměr max. průřezu přepouštěcího ventilu [m]	AfterChange;
6	h0	Double	Ini Const = 0.1	výška dna nádrže nad ventilem [m]	-
7	SA	Double	Ini Expr = Pm("/Dat...	plocha nádrže A [m2]	-
8	SB	Double	Ini Expr = Pm("/Dat...	plocha nádrže B [m2]	-
9	SC	Double	Ini Expr = Pm("/Dat...	plocha nádrže C [m2]	-
10	Sp	Double	Ini Expr = Pm("/Dat...	plocha přepouštěcího potrubí [m2]	-
11	sv	Double	Ini Expr = Pm("/Dat...	plocha vypouštěcího ventilu [m2]	-
12	sprep	Double	Ini Expr = Pm("/Dat...	plocha přepouštěcího ventilu [m2]	-

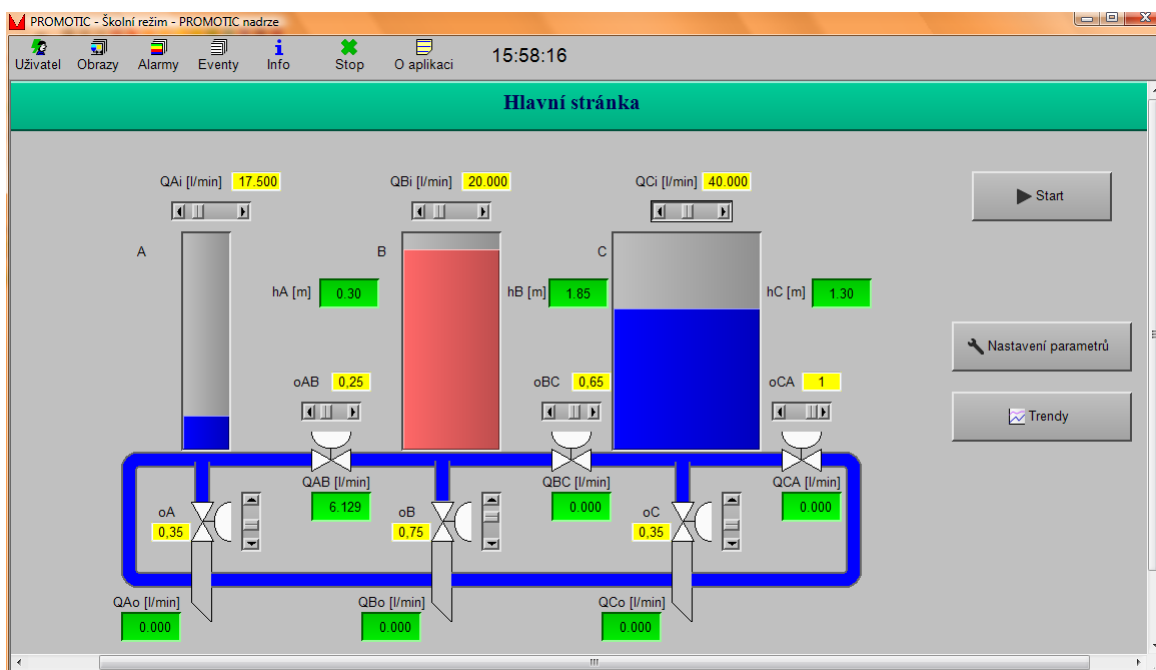
Obrázek 12 - Rozdělené skupiny dat

Například skupina dat *geoRozmery* obsahuje proměnné *dA*, *dB*, *dC*, *d*, *dv*, *dp*, *h0*, *SA*, *SB*, *SC*, *Sp*, *sv* a *sprep*. Ve vytváření proměnných záleží na pořadí, a proto si musíme dávat pozor na jejich název a pořadové umístění, aby se nám nestalo, že v nějakém výpočtu potřebujeme znát hodnotu proměnné, která by byla definována až pod ní. Každé položce přiřadíme datový typ *double*, hodnotu a datovou vazbu. Význam jednotlivých proměnných můžeme nalézt v tabulkách 2 a 3. U každého objektu máme možnost vyplnit kolonku *Popis* a u proměnných kolonku *Poznámka*, kam krátce napíšeme, o čem daná proměnná vypovídá. Těmito komentáři si můžeme osvěžit naši paměť, kdybychom

aplikaci delší čas nepoužívali nebo by se k aplikaci dostal někdo jiný. V tomto případě mu tyto komentáře usnadní pochopení dané problematiky. Složka *Data* ještě obsahuje dvě další proměnné *cisloStart* a *cisloStop*. Tyto objekty PmNumber slouží pro spuštění a zastavení aplikace tlačítky Start a Stop. Ještě jsem vytvořil složku *Trendy*, která se zabývá zobrazením výšek hladin s možností podívat se do minulosti, a složku *Casovac*, ve které je objekt Timer. V jeho události se po 1 sekundě vypočítávají ony výšky hladin. Podrobnější vysvětlení bude provedeno v následujících podkapitolách.

4.2.1 Logika aplikace

Spustíme-li aplikaci, zobrazí se nám Úvodní obrazovka. Kliknutím na tlačítko pro vstup do aplikace se dostaneme na obrazovku Hlavní stránka (ilustrativní obrázek 13). Pomocí táhel zde můžeme nastavovat hodnoty přítoků, otevření přepouštěcích a vypouštěcích ventilů. Tyto změny lze samozřejmě provádět i za běhu aplikace.



Obrázek 13 - Hlavní stránka programu

Více možností nastavení se skrývá pod tlačítkem Nastavení parametrů (obrázek 14). Objeví se stránka, kde opět můžeme nastavit hodnoty přítoků a otevření ventilů. Tyto parametry tady mají připsané jednotky i význam a jsou zde přehledně zobrazeny v levém bloku tabulek pod sebou. V pravém bloku zadáváme geometrické rozměry nádrží. Tyto rozměry můžeme zadat pouze před spuštěním aplikace, protože měnit geometrické rozměry skutečných nádrží při měření v reálném světě je nemožné, tudíž i v této simulaci je to zakázáno. Dosáhl jsem toho tak, že celou tabulku jsem sloučil do jednoho prvku. Když je stisknuté tlačítko Start, tak vlastnost Enabled (Povolení / zakázání ovládání) tohoto prvku se nastaví na hodnotu false a geometrické rozměry nejdou nastavovat. Pro úplnost jsou pod Geometrickými rozměry uvedeny aktuální hodnoty výšek hladin jednotlivých nádrží.

Zadávání proměnných

Vstupní průtoky

Q_{Ai} = [l/min] vstupní průtok pro nádrž A

Q_{Bi} = [l/min] vstupní průtok pro nádrž B

Q_{Ci} = [l/min] vstupní průtok pro nádrž C

Geometrické rozměry

d_A = [m] průměr válcové nádrže A

d_B = [m] průměr válcové nádrže B

d_C = [m] průměr válcové nádrže C

d = [m] průměr přepouštěcího potrubí mezi nádržemi

d_v = [m] průměr max. průřezu vypouštěcího ventilu (pro všechny stejný)

d_p = [m] průměr max. průřezu přepouštěcího ventilu (pro všechny stejný)

h₀ = [m] výška dna nádrže nad ventilem

Ventily (v rozsahu 0 - 1)

o_A = hodnota otevření vypustného ventilu o_A

o_B = hodnota otevření vypustného ventilu o_B

o_C = hodnota otevření vypustného ventilu o_C

o_{AB} = hodnota otevření vypustného ventilu o_{AB}

o_{BC} = hodnota otevření vypustného ventilu o_{BC}

o_{CA} = hodnota otevření vypustného ventilu o_{CA}

Aktuální výšky hladin

h_A = [m] aktuální výška hladiny v nádrži A

h_B = [m] aktuální výška hladiny v nádrži B

h_C = [m] aktuální výška hladiny v nádrži C

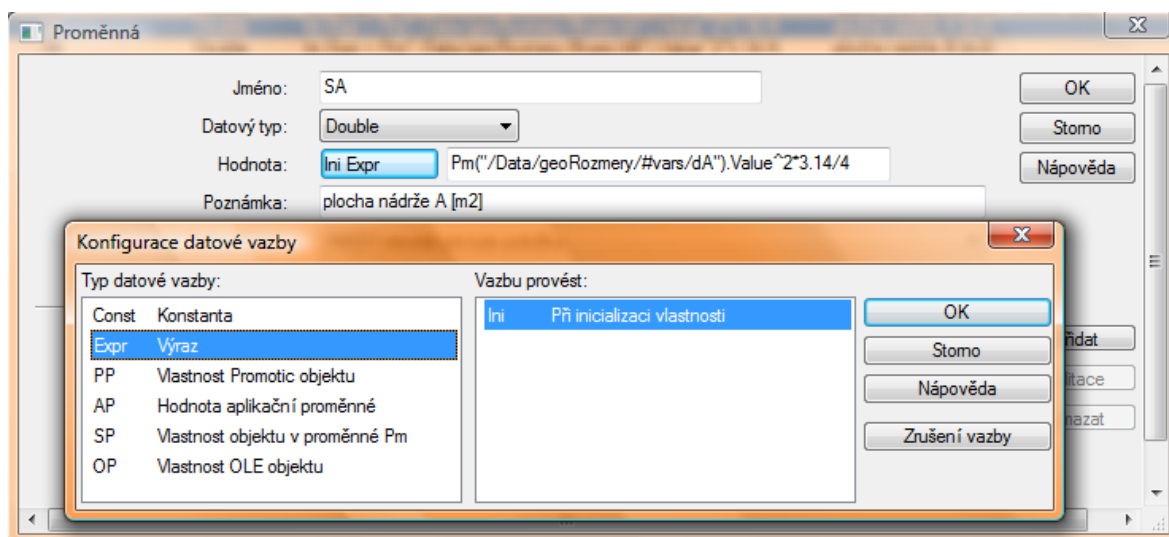
Obrázek 14 - Okno pro zadávání proměnných

Na tomto displeji jsou ještě 4 významná tlačítka. Jsou dána do dvojic pod Vstupní průtoky a pod Geometrické rozměry. Pokaždé jedno tlačítko slouží pro načtení hodnot ze souboru a druhé pro uložení hodnot do souboru. Načtení dat pro rozměry nádrží ze souboru funguje opět pouze při zastavené simulaci. Je proto použita stejná vlastnost tlačítka Enabled, jako tomu bylo u sloučeného prvku Geometrické rozměry.

Aby nám aplikace fungovala korektně, musíme zajistit ještě některé logické vazby, které jsou realizované jako jednoduché skripty. Pro spuštění aplikace použijeme tlačítko Start. Po jeho stisku se nám rozběhne časovač a v něm výpočet soustavy rovnic. Toto tlačítko nám po stisku zmizí a objeví se tlačítko Stop. K takovému chování dochází díky objektům *cisloStart* a *cisloStop*. *CisloStart* a *cisloStop* jsou objekty s datovým typem boolean. První zmiňovaný má nastavenou počáteční hodnotu na 1, druhý má počáteční hodnotu nastavenou na 0. V událostech tlačítka Start máme napsaný krátký script, který zajistí, že po stisku Start se hodnota *cisloStart* změní na 0 a hodnota *cisloStop* naopak na 1. Vlastnost viditelnosti tlačítka Start je navázána datovou vazbou také na *cisloStart*. Když je hodnota *cisloStart* 1, tlačítko je vidět. Když je hodnota 0, tlačítko už vidět není. Tlačítko Stop je propojeno s *cisloStop*. Princip je stejný, jen jsou obrácené hodnoty. Při stisku tlačítka Stop se zastaví objekt časovač a s ním i výpočet rovnic. Systém je možné opět spustit tlačítkem Start. Výpočet bude pokračovat ze zastavených hodnot jako v reálném systému.

4.2.2 Přřazení datové vazby

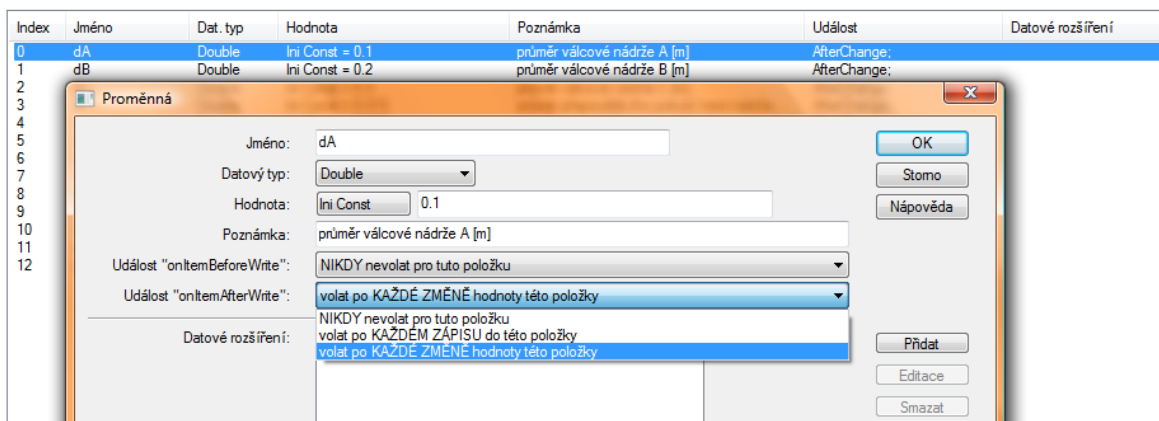
Program PROMOTIC umožňuje každému prvku přidat datovou vazbu. Pod pojmem datová vazba se rozumí spojení prvků. Když změníme vlastnost daného prvku, tak se nám současně změní vlastnost jiného prvku, aniž bychom pro to museli napsat zvláštní script, který bychom museli následně spustit. Jak již bylo řečeno, každé proměnné ve složce *Data* můžeme přiřadit datovou vazbu. Pro proměnné, u kterých chceme provést pouze inicializaci na začátku spuštění programu na určitou hodnotu, je to jednoduché. Jako typ datové vazby zvolíme Konstantu. Poněkud složitější je to pro proměnné, které nejsou definovány jako Konstanta, ale jako Výraz. Výraz musí být zapsán v syntaxi jazyka VBScript. V této práci se jedná hlavně o proměnné *SA*, *SB*, *SC*. Plochy nádrží se totiž počítávají z průměrů jednotlivých nádrží. Plocha nádrže se vypočítá podle definičního vztahu $S_A = \frac{\pi d_A^2}{4}$. Pro tyto případy zvolíme typ datové vazby Výraz. Vazbu chceme opět provést při inicializaci vlastností. Jak vidíme na obrázku 15, do kolonky Hodnota vepíšeme námi požadovaný výraz.



Obrázek 15 - Konfigurace datové vazby

Jelikož máme možnost měnit před startem programu parametry nádrže (například načíst jiné hodnoty geometrických rozměrů nádrže), musíme zajistit, aby se nám také pokaždé změnily ony plochy. Pro tuto situaci slouží událost `onItemAfterWrite`⁴. Máme tři možnosti, jak tuto událost volat. Na obrázku 16 je zachycen případ, kdy tuto událost voláme po každé změně hodnoty této položky.

⁴ co se má provést s položkou po zápisu



Obrázek 16 - Událost onItemAfterWrite

Na výběr máme mezi voláním po každém zápisu do této položky a již zmíněným voláním po každé změně hodnoty této položky. Chceme, aby se nám událost volala po každé změně hodnot průměrů nádrží. Proto zvolíme právě tuto možnost. Kdežto kdybychom zvolili možnost po zápisu, tak by se nám událost volala jen po zápisu. V záložce událost na kartě onItemAfterWrite musíme ještě dopsat krátký skript. V tomto případě vypadá asi takto:

```
Dim pi, dA 'založí 2 proměnné pi a dA
pi = 3.14
dA = Pm("/Data/geoRozmery/#vars/dA") 'do dA přiřadíme cestu, kde je
'uložená skutečná proměnná dA
Pm("/Data/geoRozmery/#vars/SA").Value=pi*(dA^2)/4 'do vlastnosti
'hodnota u proměnné SA se přiřadí hodnota plochy nádrže
```

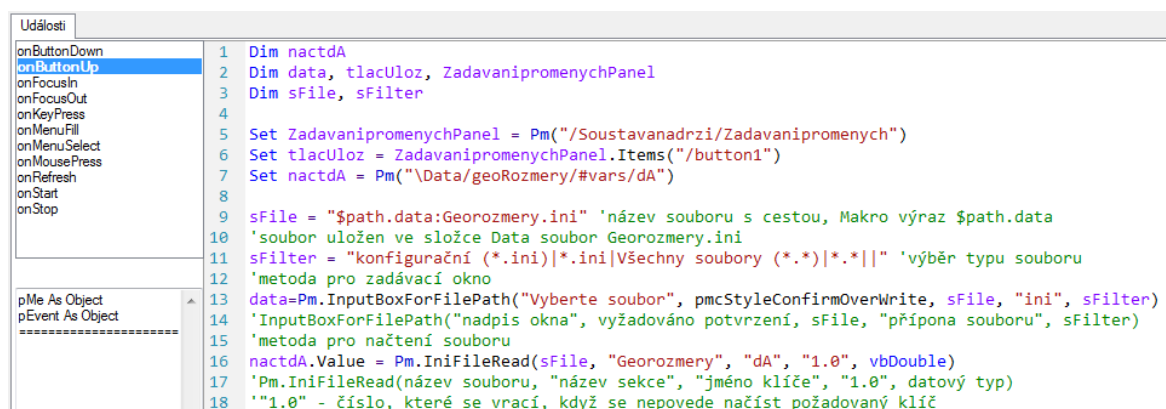
4.2.3 Načtení / Uložení souboru

Jelikož v jednom bodu zadání je napsáno, že před spuštěním aplikace půjdou načíst hodnoty geometrických rozměrů ze souboru, musíme vyřešit tento problém. Pro jednoduchá nastavení souboru používá PROMOTIC standardní INI soubory systému Windows. INI soubor obsahuje tzv. sekce a klíče. Název sekce je uveden v hranatých závorkách. Jednotlivé sekce mohou obsahovat několik klíčů. Název klíče pokaždé začíná na novém řádku a pomocí znaku „rovná se“ do něj přiřazujeme daný obsah. Klíče spadají vždy do příslušné sekce, která byla vytvořena výše. Sekce končí buď vytvořením další sekce, nebo koncem souboru. INI soubor pro geometrické nádrže má následující tvar:

```
[Georozmery]
dA=0,1
dB=0,2
dC=0,3
d=0,015
dv=0,005
dp=0,0075
h0=0,1
```


Tento soubor obsahuje sekci s názvem Georozmery, která obsahuje 7 klíčů. Jak je patrné z ukázky, hodnoty klíčů jsou stejné jako v tabulce 3. Obdobně by vypadal i INI soubor pro vstupní přítoky.

V Editoru obrazu na obrazovce Zadávání proměnných jsem vytvořil 2 tlačítka. Jedno s názvem Načíst ze souboru, druhé se jmenuje Uložit do souboru. Význam tlačítek odpovídá jejich názvu. Pro načítání souboru jsem do události `onButtonUp`⁵ dopsal script (obrázek 17) obsahující metodu `IniFileRead`. Tato metoda obstará to, že se nám dané hodnoty načtou ze souboru Georozmery do jednotlivých proměnných v aplikaci.



Obrázek 17 - Metoda `IniFileRead`

Obdobně by vypadal skript pro uložení hodnot do souboru, jen s tím rozdílem, že by byla použita metoda `IniFileWrite`. Pro úplnost ještě uvedu, že text programu obsahuje metodu `InputBoxForFilePath`. Hlavní funkce této metody je zobrazit zadávací okno. Uživatel může v tomto okně změnit umístění daného souboru i jeho název.

4.2.4 Ladění programu – INFO systém

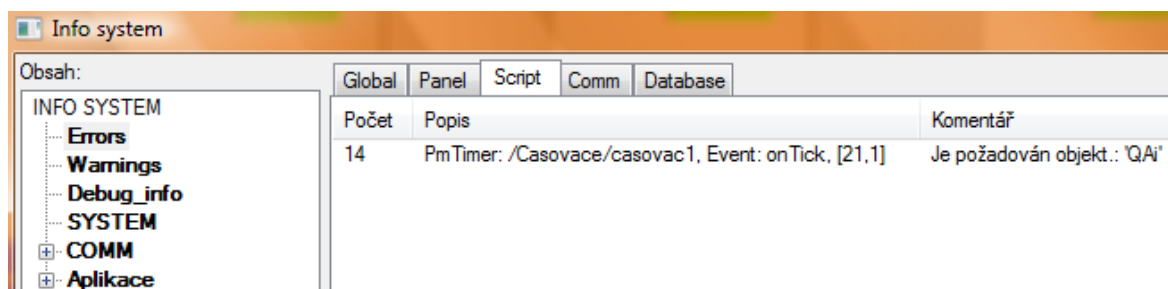
Jelikož aplikace obsahuje několik kratších skriptů a jeden rozsáhlý skript v časovači, potřeboval jsem mít možnost tyto skripty ladit. PROMOTIC standardně podporuje debugger⁶ MS Office FrontPage. Mě se ale po doinstalování programu FrontPage do počítače s operačním systémem Windows Vista ladění skriptů nepodařilo zprovoznit. Zkusil jsem ještě Visual Studio také od firmy Microsoft, ale taktéž se mi krokování v debuggeru nepovedlo zprovoznit.

PROMOTIC obsahuje komponentu INFO systém (obrázek 18). Je to souhrn informací ve speciálním okně, které nám systém dává při spuštěné aplikaci a my si je můžeme prohlížet. Například nám ukáže, jestli náš časovač již běží, jaké jsou hodnoty výšek hladin, jak velké jsou přítoky a odtoky, ale také nám podá zprávu, jestli náš napsaný skript obsahuje chybu. Když dojde k chybě při provádění skriptu, skript se nedokončí, ale aplikace nadále běží. V INFO systému se dozvíme informaci, v které události objektu

⁵ událost se projeví až po povolení stisku tlačítka myši

⁶ ladící program

došlo k chybě. INFO systém nám napíše, na jakém řádku došlo k chybě a obdržíme i krátký komentář, co by mohlo být špatně. Není to tedy moc komfortní, ale pro jednoduché ladění to stačí.



Obrázek 18 - INFO systém

4.2.5 Realizace výpočtu v časovači

Stěžejní část této práce spočívá ve vytvoření procedury pro výpočet zadané soustavy obyčejných nelineárních diferenciálních rovnic v jazyce VBScript. Tento numerický výpočet pracující v reálném čase musíme implementovat do prostředí PROMOTIC. V události onTick časovače vytvoříme skript počítající metodou Runge-Kutta diferenciální rovnice. Vložení kódu do události onTick zajistíme, že na každé tiknutí časovače se nám tato událost volá. Perioda tikání je nastavena na 1 sekundu. Úplný zdrojový kód je uveden v příloze. Tady jen nastíním způsob řešení. Na začátku si deklarujeme potřebné proměnné a přiřadíme jim cestu ke stromové struktuře v Editoru aplikace. Vstupní přítoky ještě vydělíme číslem 60000, protože přítoky nezadáujeme v rozměru $\text{m}^3 \cdot \text{s}^{-1}$ ale v $\text{l} \cdot \text{min}^{-1}$, ale vztahy (11) až (13) počítají s rozměrem $\text{m}^3 \cdot \text{s}^{-1}$, proto tento převod. Podmínkou zjistíme, jestli systém běží a tudíž jestli máme spustit časovač a s ním i výpočet. V této podmínce se uskuteční celý numerický výpočet. Nejdříve přepíšeme v syntaxi VBScriptu pomocné výpočty pro rovnice (5) až (10). Dále zapíšeme rovnice (11) až (13) ve formě vektoru. Zapíšeme i následující rovnice (21) až (24). Posléze přiřepíšeme konkretizovanou rovnici (19) pro všechny 3 výšky hladin. Průtoky mezi nádržemi a výstupní průtoky tentokrát vynásobíme 60000 a hodnoty spočítaných proměnných přiřadíme do skutečných proměnných ve stromu aplikace. Nakonec ještě použijeme metodu Run, která nám zajistí zápis hodnot do historie.

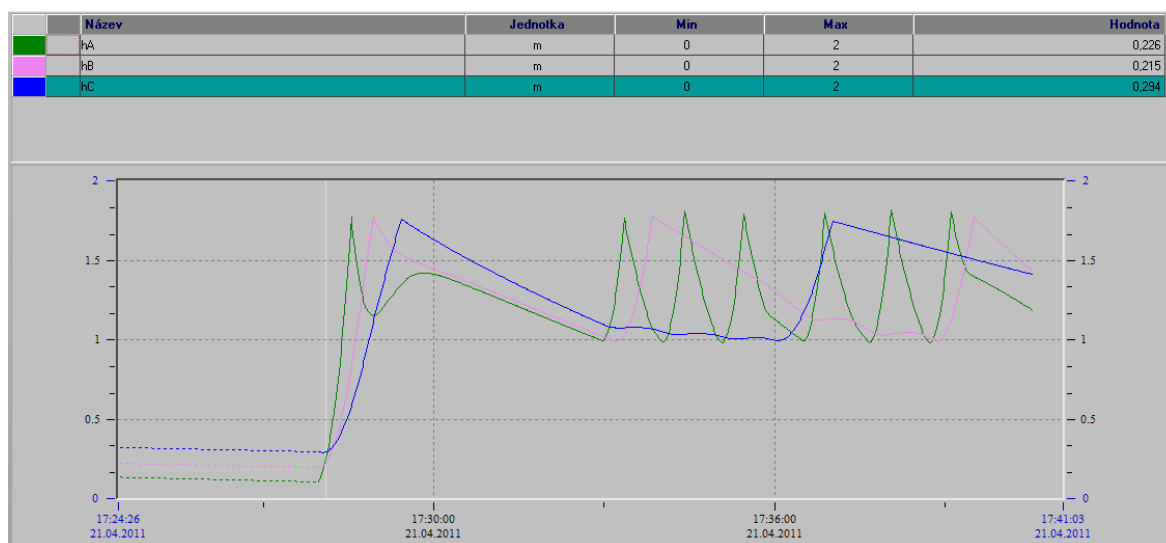
Ve skriptu je ještě přidána podmínka, že kdyby byly výšky hladin větší než 2 m, tak jsou tyto výšky nastaveny na 2 m. Vyjadřujeme tím, že nemáme nádrže s nekonečně velkou výškou, ale že nám po dosažení určité hodnoty kapalina odtéká odtokovým potrubím.

4.2.6 Trendy

Jednou z částí zadání, bylo vytvořit graf zobrazující aktuální hodnoty výšek hladin v nádržích. PROMOTIC pro tuto část využívá systém Trendů. „Trendování“ je ukládání měřených hodnot daných veličin do historie pro pozdější zpracování. Tento děj má dva stavy. Tyto stavy jsou zaznamenávání dat a pozdější zpracování těchto dat. První stav je vlastní ukládání dat do námi požadovaného souboru v paměti. Druhý stav je zpracování

těchto dat. Pro ukládání dat si můžeme zvolit výstupní formát souboru. PROMOTIC standardně ukládá hodnoty do souboru s formátem Promotic binary file cyclic. Tento formát je ryze interní a není čitelný žádným jiným programem. Já jsem tedy raději použil, jako výstupní formát pro ukládání dat, databázový soubor Access (Database Access backups). Tento typ souboru se hodí pro ukládání menšího objemu dat. Data jsou uložena přehledně v tabulce. Pro zpracování dat nejsme tedy vázáni pouze na program PROMOTIC, ale uložené hodnoty si můžeme nechat zobrazit třeba v programu Microsoft Office Access.

Pro vyobrazení časových průběhů výšek hladin jsem použil objekt TrendsView, a to typ Prohlížeč trendů s hlavičkou. Tento typ má výhodu v tom, že nad grafem vidíme číselné hodnoty aktuálních výšek s nastavenými minimálními a maximálními hodnotami. Také zde máme možnost vypnout zobrazení zvolené výšky hladiny. Můžeme se posouvat po časové ose do minulosti. V systému PROMOTIC lze takovýto trend vytvořit velice snadno, a to v Editoru obrazu. Dále do zdrojového kódu v události onTick v časovači přepíšeme metodu Run. Tato metoda nám zajistí, že po spuštění aplikace se nám do souboru group0.mdb budou každou sekundu ukládat hodnoty výšek hladin s časovým údajem výpočtu. Na obrázku 19 vidíme, jak by mohl případně takový graf zobrazující výšky hladin hA , hB , hC vypadat.



Obrázek 19 - Ukázka trendu

5 Závěr

Cílem bakalářské práce bylo vytvořit aplikaci v systému PROMOTIC, která simuluje časové chování zadané soustavy v reálném čase. Nejdříve bylo nutné nastudovat numerické řešení soustavy obyčejných nelineárních diferenciálních rovnic, kterými je popsán model soustavy tří navzájem propojených nádrží. Tyto rovnice byly dále realizovány jako numerický výpočet, který pomocí metody Runge-Kutta 4. řádu počítá výšky hladin v jednotlivých nádržích. Tvar rovnic výpočtu je upraven tak, aby byl snadno zapsatelný v jazyce VBScript. Výpočet je závislý na hodnotách vstupních přítoků, výstupních průtoků a na otevření přepouštěcích ventilů.

Aplikace byla vytvořena v prostředí PROMOTIC, proto jsem se také musel seznámit s tímto systémem a musel jsem se v něm naučit používat potřebné funkce a důležité vlastnosti (např.: napojení datové vazby na proměnné a objekty, využívat výhody INFO systému atd.). Práce v programu PROMOTIC není nijak zvláště složitá a pro vypracování zadané práce se tento program ukázal jako vhodný pomocník. Kdybychom měli podobnou aplikaci realizovat v některém ze standardních programovacích jazyků (třeba v jazyce C), určitě by nám tvorba takovéto aplikace zabrala mnohem více času a stála by nás více námahy než v tomto případě. Všechny cíle práce byly splněny a výsledná aplikace odpovídá zadaným požadavkům na daný systém.

Aplikace byla navržena takovým způsobem, že kdybychom výpočet v časovači nahradili sběrem dat ze skutečného zařízení (třeba bychom měřili skutečné přítoky a výtoky vodní nádrže), tak by aplikace jen s malými úpravami mohla nadále fungovat. Aplikace by mohla být použita ve velině jako informační systém nebo i jako řídicí systém pro operátora, který by tímto systémem ovládal zařízení zabývající se touto tematikou.

Aplikaci jsem rozšířil o řízení výšek hladin v nádržích. Tyto výšky hladin nám samy pulzují mezi minimálními a maximálními hodnotami a jejich velikost můžeme zadat pomocí INFO systému. Tato nadstavba by mohla být užitečná na reálném zařízení.

Literatura

- [1] SMEJKAL, Viktor. *Matematický model soustavy nádrží*. Pardubice, 2010. 40 s. Bakalářská práce. Univerzita Pardubice, Fakulta elektrotechniky a informatiky, Studijní obor Informační technologie.
- [2] *Proudění* [online]. c2010 [cit. 2011-04-12]. Ustálené proudění tekutin. Dostupné z WWW: <<http://www.sszdra-karvina.cz/bunka/fy/02prou/prus.htm>>.
- [3] JANDORA, Radek. *Neoficiální internetová fakulta všeho možného* [online]. 2000, poslední změna 29.01.2004 [cit. 2011-04-12]. Hydrodynamika. Dostupné z WWW: <<http://radek.jandora.sweb.cz/f06.htm>>.
- [4] Bernoulliho rovnice. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 9. 6. 2003, last modified on 9. 4. 2011 [cit. 2011-04-12]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Bernoulliho_rovnice>.
- [5] *Proudění* [online]. c2010 [cit. 2011-04-14]. Proudění tekutin. Dostupné z WWW: <<http://sszdra-karvina.cz/bunka/fy/02prou/prbr.htm>>.
- [6] REKTORYS, Karel. et al. *Přehled užití matematiky II*. 6. přepracované vydání. Praha : Prometheus, 1995. 25.3 Obecná jednokroková metoda, s. 459-462. ISBN 80-85849-62-3.
- [7] *MICROSYS - SCADA PROMOTIC* [online]. c2011 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.promotic.eu/>>.
- [8] SCADA. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 4. 10. 2010, last modified on 14. 3. 2011 [cit. 2011-04-12]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/SCADA>>.
- [9] BALDA, Pavel. *Vendulka* [online]. 2007-04-24 [cit. 2011-04-12]. SCADA a HMI systémy. Dostupné z WWW: <http://vendulka.zcu.cz/Download/Free/IRS1/IRS1-08_SCADA_HMI.pdf>.
- [10] *SCADA/HMI systém Reliance* [online]. c2011 [cit. 2011-04-12]. Co znamená SCADA/HMI?. Dostupné z WWW: <<http://www.reliance.cz/cs/products/what-does-scada-hmi-mean>>.
- [11] *SCADA/HMI systém Reliance* [online]. c2011 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.reliance.cz/cs/main>>.
- [12] *Moravské přístroje a.s. :: Hlavní stránka* [online]. c2003, 8.4.20011 [cit. 2011-04-12]. Dostupné z WWW: <<http://www.mii.cz/>>.
- [13] *SCADA/HMI systém Reliance* [online]. c2011 [cit. 2011-04-14]. SCADA/HMI systém Reliance 4. Dostupné z WWW: <<http://www.reliance.cz/cs/products/reliance4-scada-hmi-system>>.

- [14] *Moravské přístroje a.s. :: Hlavní stránka* [online]. c2003, 4.10.2010 [cit. 2011-04-12]. Moravské přístroje a.s. :: Co je Control Web?. Dostupné z WWW: <<http://www.mii.cz/art?id=380&cat=146&lang=405>>.
- [15] *Wonderware Home* [online]. c2011 [cit. 2011-04-11]. Dostupné z WWW: <<http://global.wonderware.com/EN/Pages/default.aspx>>.
- [16] *SCADA, Industrial & Facilities Management Software - Citect Home* [online]. c2010 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.citect.com/>>.
- [17] *PROMOTIC* [online]. 2011 [cit. 2011-04-12]. Co je PROMOTIC?. Dostupné z WWW: <<http://www.promotic.eu/cz/promotic/scada-pm.htm>>.
- [18] *PROMOTIC* [online]. 2011 [cit. 2011-05-02]. Obsah dokumentace PROMOTIC. Dostupné z WWW: <<http://www.promotic.eu/cz/promotic/scada-pmdoc.htm>>.

Příloha A

Zdrojový kód obsluhy metody onTick časovače (výpočet diferenciálních rovnic):

```
Dim HlavnistrankaPanel, tlacStart, cislo 'deklarace proměných
Dim ohA, ohB, ohC 'proměnné použité pro vykreslení grafů
Dim hA, hB, hC, h, xhA, xhB, xhC 'výšky hladin
Dim h0, dp, dv, d, dC, dB, dA, g 'geometrické rozměry nádrží
Dim SA, SB, SC, Sp, sv, sprep
Rem sprep = sp -> plocha prepousteciho ventilu
Dim oA, oB, oC, oAB, oBC, oCA 'otevření ventilů
Dim QA0, QB0, QC0, QAB, QBC, QCA, QAi, QBi, QCi 'výstupní odtoky a
'průtoky
Dim dt, k1(2), k2(2), k3(2), k4(2) 'proměnné pro výpočet Runge-Kutta
'k1(2) - pole o 3 položkách, indexy od 0, 3 položky protože jsou 3
'hladiny
'k1, k2, k3, k4 - protože RK4 chce 4 konstanty

Set HlavnistrankaPanel = Pm("/Soustavanadrzi/Hlavnistranka")
'zpřístupnění Hlavní stránky
Set tlacStart = HlavnistrankaPanel.Items("/button0") 'zpřístupnění
'tlačítka Start
Set cislo = Pm("/Data/cisloStart") 'napojení na objekt cisloStart

if (cislo.Value = 0) then 'je-li stisknuté tlač. Start
g = 9.81 'gravitační zrychlení
dt = 1 'časový interval, krok nastaven na 1 sekunda
h0 = Pm("\Data/geoRozmery/#vars/h0") 'přiřazení proměnných ke skutečným
'objektům
SA = Pm("\Data/geoRozmery/#vars/SA")
SB = Pm("\Data/geoRozmery/#vars/SB")
SC = Pm("\Data/geoRozmery/#vars/SC")
Sp = Pm("\Data/geoRozmery/#vars/Sp")
sv = Pm("\Data/geoRozmery/#vars/sv")
sprep = Pm("\Data/geoRozmery/#vars/sprep")

hA = Pm("\Data/hladiny/#vars/hA")
hB = Pm("\Data/hladiny/#vars/hB")
hC = Pm("\Data/hladiny/#vars/hC")

QAi = Pm("\Data/prutoky/#vars/QAi")/60000 'dělení 60000 kvůli přepočtu na
'1/min, 1 m3 -> 1000 l, 60s -> 1 min, 1 m3/s = 60000 l/min
QBi = Pm("\Data/prutoky/#vars/QBi")/60000
QCi = Pm("\Data/prutoky/#vars/QCi")/60000
oA = Pm("\Data/ventily/#vars/oA")
oB = Pm("\Data/ventily/#vars/oB")
oC = Pm("\Data/ventily/#vars/oC")
oAB = Pm("\Data/ventily/#vars/oAB")
oBC = Pm("\Data/ventily/#vars/oBC")
oCA = Pm("\Data/ventily/#vars/oCA")
```

```

Rem POMOCNÉ VÝPOČTY výtoků a průtoků pro diferenciální rce
QA0 = sv*oA*(2-oA)*sqr((2*(hA+h0)*g)/(1-((sv*oA*(2-oA))/SA)^2))
QB0 = sv*oB*(2-oB)*sqr((2*(hB+h0)*g)/(1-((sv*oB*(2-oB))/SB)^2))
QC0 = sv*oC*(2-oC)*sqr((2*(hC+h0)*g)/(1-((sv*oC*(2-oC))/SC)^2))
Qab = sgn(hA - hB)*sprep*oAB*(2-oAB)*sqr((2*Abs(hA - hB)*g)/(1-((sprep*oAB*(2-oAB))/Sp)^2))
Qbc = sgn(hB - hC)*sprep*oBC*(2-oBC)*sqr((2*Abs(hB - hC)*g)/(1-((sprep*oBC*(2-oBC))/Sp)^2))
Qca = sgn(hC - hA)*sprep*oCA*(2-oCA)*sqr((2*Abs(hC - hA)*g)/(1-((sprep*oCA*(2-oCA))/Sp)^2))
k1(0)= (QAi - QA0 - Qab+QCA)/SA 'dhA/dt
k1(1)= (QBi - QB0 - Qbc+QAB)/SB
k1(2)= (QCi - QC0 - Qca+QBC)/SC

'QA0, QB0, QC0, Qab, Qbc, Qca závisí ve výpočtech na hA, hB, hC
'proto nové výpočty POMOCNÝCH VÝPOČTŮ
xhA=hA+dt*k1(0)/2 'vzoreček (22) podle RK4, dt = T, xhA abych si
'nepřepsal hA
QA0 = sv*oA*(2-oA)*sqr((2*(xhA+h0)*g)/(1-((sv*oA*(2-oA))/SA)^2))
'znova spočítat přítoky a výtoky
xhB=hB+dt*k1(1)/2
QB0 = sv*oB*(2-oB)*sqr((2*(xhB+h0)*g)/(1-((sv*oB*(2-oB))/SB)^2))
xhC=hC+dt*k1(2)/2
QC0 = sv*oC*(2-oC)*sqr((2*(xhC+h0)*g)/(1-((sv*oC*(2-oC))/SC)^2))
Qab = sgn(xhA - xhB)*sprep*oAB*(2-oAB)*sqr((2*Abs(xhA - xhB)*g)/(1-((sprep*oAB*(2-oAB))/Sp)^2))
Qbc = sgn(xhB - xhC)*sprep*oBC*(2-oBC)*sqr((2*Abs(xhB - xhC)*g)/(1-((sprep*oBC*(2-oBC))/Sp)^2))
Qca = sgn(xhC - xhA)*sprep*oCA*(2-oCA)*sqr((2*Abs(xhC - xhA)*g)/(1-((sprep*oCA*(2-oCA))/Sp)^2))
k2(0)= (QAi - QA0 - Qab+QCA)/SA
k2(1)= (QBi - QB0 - Qbc+QAB)/SB
k2(2)= (QCi - QC0 - Qca+QBC)/SC

xhA=hA+dt*k2(0)/2 'vzoreček (23) podle RK4
QA0 = sv*oA*(2-oA)*sqr((2*(xhA+h0)*g)/(1-((sv*oA*(2-oA))/SA)^2))
xhB=hB+dt*k2(1)/2
QB0 = sv*oB*(2-oB)*sqr((2*(xhB+h0)*g)/(1-((sv*oB*(2-oB))/SB)^2))
xhC=hC+dt*k2(2)/2
QC0 = sv*oC*(2-oC)*sqr((2*(xhC+h0)*g)/(1-((sv*oC*(2-oC))/SC)^2))
Qab = sgn(xhA - xhB)*sprep*oAB*(2-oAB)*sqr((2*Abs(xhA - xhB)*g)/(1-((sprep*oAB*(2-oAB))/Sp)^2))
Qbc = sgn(xhB - xhC)*sprep*oBC*(2-oBC)*sqr((2*Abs(xhB - xhC)*g)/(1-((sprep*oBC*(2-oBC))/Sp)^2))
Qca = sgn(xhC - xhA)*sprep*oCA*(2-oCA)*sqr((2*Abs(xhC - xhA)*g)/(1-((sprep*oCA*(2-oCA))/Sp)^2))
k3(0)= (QAi - QA0 - Qab+QCA)/SA
k3(1)= (QBi - QB0 - Qbc+QAB)/SB
k3(2)= (QCi - QC0 - Qca+QBC)/SC

xhA=hA+dt*k3(0) 'vzoreček (24) podle RK4, už není 1/2
QA0 = sv*oA*(2-oA)*sqr((2*(xhA+h0)*g)/(1-((sv*oA*(2-oA))/SA)^2))

```



```

xhB=hB+dt*k3(1)
QB0 = sv*oB*(2-oB)*sqr((2*(xhB+h0)*g)/(1-((sv*oB*(2-oB))/SB)^2))
xhC=hC+dt*k3(2)
QC0 = sv*oC*(2-oC)*sqr((2*(xhC+h0)*g)/(1-((sv*oC*(2-oC))/SC)^2))
Qab = sgn(xhA - xhB)*sprep*oAB*(2-oAB)*sqr((2*Abs(xhA - xhB)*g)/(1-((sprep*oAB*(2-oAB))/Sp)^2))
Qbc = sgn(xhB - xhC)*sprep*oBC*(2-oBC)*sqr((2*Abs(xhB - xhC)*g)/(1-((sprep*oBC*(2-oBC))/Sp)^2))
Qca = sgn(xhC - xhA)*sprep*oCA*(2-oCA)*sqr((2*Abs(xhC - xhA)*g)/(1-((sprep*oCA*(2-oCA))/Sp)^2))
k4(0)= (QAi - QA0 - Qab+QCA)/SA
k4(1)= (QBi - QB0 - Qbc+QAB)/SB
k4(2)= (QCi - QC0 - Qca+QBC)/SC

hA = hA + (k1(0)+2*k2(0)+2*k3(0)+k4(0))*dt/6 'přepsání vztahu (19), nové
'hA = staré hA + konstanty
hB = hB + (k1(1)+2*k2(1)+2*k3(1)+k4(1))*dt/6
hC = hC + (k1(2)+2*k2(2)+2*k3(2)+k4(2))*dt/6

if (hA > 2) then 'podmínky, aby výšky hladin nebyly větší než 2 metry
hA = 2
End if

if (hB > 2) then
hB = 2
End if

if (hC > 2) then
hC = 2
End if

Pm("\Data/hladiny/#vars/hA").Value = hA 'zaktualizování hladin, vypočtené
'hA uložíme do objektu
Pm("\Data/hladiny/#vars/hB").Value = hB
Pm("\Data/hladiny/#vars/hC").Value = hC

'mám nové hladiny, tak provedu znova výpočet
QA0 = sv*oA*(2-oA)*sqr((2*(hA+h0)*g)/(1-((sv*oA*(2-oA))/SA)^2))
QB0 = sv*oB*(2-oB)*sqr((2*(hB+h0)*g)/(1-((sv*oB*(2-oB))/SB)^2))
QC0 = sv*oC*(2-oC)*sqr((2*(hC+h0)*g)/(1-((sv*oC*(2-oC))/SC)^2))
Qab = sgn(hA - hB)*sprep*oAB*(2-oAB)*sqr((2*Abs(hA - hB)*g)/(1-((sprep*oAB*(2-oAB))/Sp)^2))
Qbc = sgn(hB - hC)*sprep*oBC*(2-oBC)*sqr((2*Abs(hB - hC)*g)/(1-((sprep*oBC*(2-oBC))/Sp)^2))
Qca = sgn(hC - hA)*sprep*oCA*(2-oCA)*sqr((2*Abs(hC - hA)*g)/(1-((sprep*oCA*(2-oCA))/Sp)^2))

Pm("\Data/prutoky/#vars/QA0").Value = QA0*60000 'zveřejnění a zase *60000
'převedení na m3/s
Pm("\Data/prutoky/#vars/QB0").Value = QB0*60000
Pm("\Data/prutoky/#vars/QC0").Value = QC0*60000

```

```

Pm("\Data/prutoky/#vars/QAB").Value = Qab*60000
Pm("\Data/prutoky/#vars/QBC").Value = Qbc*60000
Pm("\Data/prutoky/#vars/QCA").Value = Qca*60000

    pMe.Pm("\Trendy/Trend1").Run 'metoda Run použita pro zápis dat do
'historie, uchová všechny hodnoty proměnných ve složce
'Trendy/Trend1/záložka data s časem okamžiku volání 'této metody, tedy
'hodnoty proměnných thA, thB, thC
End if

```