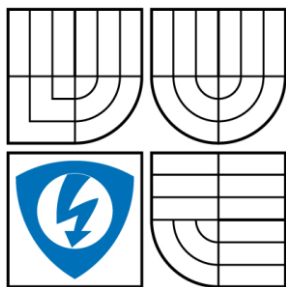


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNologií
ÚSTAV AUTOMATIZACE A MERICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODEL PRO ŘÍZENÍ VÝŠKY VODNÍ HLADINY

MODEL OF THE WATER LEVEL CONTROL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LADISLAV KAROLA

VEDOUcí PRÁCE
SUPERVISOR

Ing. LUDĚK CHOMÁT

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Ladislav Karola

ID: 106530

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Model pro řízení výšky vodní hladiny

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s technologií pro regulaci výšky vodní hladiny v nádržích. Naprogramujte do mikročipu simulaci technologického procesu tak, aby byl podobný reálnému systému výšky hladiny v nádrži. Matematický model ověřte pomocí Matlab/Simulink. Model bude řízen pomocí programovatelného automatu s vestavěným dotykovým panelem od firmy B&R a taktéž zde bude vytvořena jednoduchá vizualizace pro nastavování parametrů regulátoru. Výsledkem bakalářské práce je sestavení trenažéru technologického procesu, ověření funkčnosti, vytvoření manuálu a zadání pro model.

DOPORUČENÁ LITERATURA:

B&R www.br-automation.com

Matoušek, D.: Práce s mikrokontroléry ATMEL AVR - ATmega16. ISBN 80-7300-174-8, BEN, 2006.

Pivoňka, P.: Číslicová řídicí technika, VUT Brno, ISBN AMT101, 2003.

Ďaďo, S. & Bejček, L.: Měření průtoku a výšky hladiny. Praha, ISBN 80 7300 156 X, BEN, 2005.

Termín zadání: 8.2.2010

Termín odevzdání: 31.5.2010

Vedoucí práce: Ing. Luděk Chomát

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cílem bakalářské práce je seznámit se s problematikou řízení výšky vodní hladiny a navrhnout simulaci tohoto reálného systému.

Práce se zabývá matematickým rozbořem systému. Teoretické předpoklady jsou potvrzeny simulací v programu Matlab Simulink. Je vytvořen fyzický model obsahující mikrokontrolér ATmega16. Práce popisuje jednotlivé funkční části modelu. Pro řízení je vytvořena aplikace v Automation Studio pro programovatelný automat typu Power Panel 400 od firmy B&R. Obsahuje vizualizaci technologického procesu.

Na závěr jsou porovnány teoretické a reálné data modelu. Je uveden stručný manuál k modelu a zadání laboratorní úlohy.

Klíčová slova:

Regulace výšky vodní hladiny, Power Panel, B&R, Automation Studio, ATmega16, Simulink,

ABSTRACT

The Bachelor's Thesis deals with problem of water level control. Author design the simulation of a real technologic process.

Thesis is analyzing mathematic model of system. Mathematic model is realised in Simulink. Next point is built a model, which simulates this real technologic process. There is implemented microchip ATmega16. Thesis includes descriptions of model parts. Model is controlled with Power Panel 400 series by B&R Company. Project also contains a Visualisation of technologic process.

The last points of Thesis are comparison of theoretic data with real data, compilation of a brief manual of model and instruction for laboratory work.

Key Words:

Water Level Control, Power Panel, B&R, Automation Studio, ATmega16, Simulink,

BIBLIOGRAFICKÁ CITACE

KAROLA, L. *Model pro řízení výšky vodní hladiny*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 74 s. Vedoucí bakalářské práce Ing. Luděk Chomát.

PROHLÁŠENÍ

„Prohlašuji, že svoji bakalářskou práci na téma Model pro řízení výšky vodní hladiny jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením této projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **31. května 2010**

.....
podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu semestrálního projektu Ing. Lud'ku Chomátovi za účinnou metodickou pomoc, mnoho neocenitelných rad a především za trpělivost a vstřícnost při tvorbě a vypracování mé bakalářské práce.

V Brně dne: **31. května 2010**

.....
podpis autora

Obsah

1. ÚVOD	12
2. TEORETICKÝ ÚVOD	13
3. SIMULAČNÍ MODEL.....	15
3.1 Model Simulink	16
3.2 Regulace.....	18
3.2.1 PSD s filtrací derivační složky a omezením sumační složky	18
3.2.2 β -PSD regulátor	19
3.2.3 Feed Forward regulátor.....	21
4. REÁLNÝ MODEL	23
4.1 Fyzický vzhled modelu	23
4.1.1 Konektor MLW20	23
4.1.2 LED signalizace.....	25
4.1.3 LCD displej MC1602E-SYL/H	27
4.1.4 Vstupy poruchy do nádrže	29
4.1.5 Spínaný zdroj	30
4.1.6 DA převodník MCP4922	31
4.1.7 Mikrokontrolér ATmega 16.....	32
4.2 Implementace výpočetního algoritmu.....	34
4.2.1 Časová základna	35
4.2.2 Pravidelný výpočet v podprogramu.....	37
5. OVLÁDÁNÍ MODELU POMOCÍ PLC	43
5.1 Konfigurace projektu	43
5.2 Programová implementace.....	44
5.3 Vizualizace.....	47
6. DOSAŽENÉ VÝSLEDKY	51
7. LABORATORNÍ ÚLOHA	57
7.1 Manuál modelu nádrže.....	57
7.2 Zadání laboratorní úlohy	59
8. POUŽITÝ SOFTWARE.....	60
8.1 Matlab Simulink.....	60
8.2 AVR Studio.....	60

8.3 AVR DUDE & AVR8 Burn-O-Mat	61
8.4 Eagle	61
8.5 Automation Studio	61
9. ZÁVĚR.....	62
10. POUŽITÁ LITERATURA	64
11. PŘÍLOHY.....	66

Seznam tabulek

Tab. 1 Standardní zapojení konektoru MLW20 v laboratoři CLG	24
Tab. 2 Zapojení konektoru MLW20 v modelu nádrže	25
Tab. 3 Sloupcová LED indikace objemu.....	26
Tab. 4 Konfigurace karet v PLC.....	43

Seznam obrázků

Obr. 1 Princip ultrazvukových hladinoměrů	13
Obr. 2 Časový průběh ON/OFF regulace včetně hystereze	14
Obr. 3 Schéma modelu nádrže	15
Obr. 4 Simulační schéma modelu nádrže.....	17
Obr. 5 Odezva na skokovou změnu přítoku.....	17
Obr. 6 Struktura PSD regulátoru s filtrací derivační složky	18
Obr. 7 Výstup soustavy na skokovou změnu žádané hodnoty s PSD	19
Obr. 8 Struktura β -PSD regulátoru.....	20
Obr. 9 Výstup soustavy na skokovou změnu žádané hodnoty s β -PSD.....	20
Obr. 10 Struktura Feed Forward regulátoru	21
Obr. 11 Výstup soustavy na skokovou změnu žádané hodnoty s Feed Forward	22
Obr. 12 LED diody a LCD displej s BAR grafem	27
Obr. 13 LCD displej s informací o působící poruše.....	28
Obr. 14 Zapojení LCD displeje.....	29
Obr. 15 Schéma zapojení spínaného zdroje 5V s ochranami	31
Obr. 16 Schéma zapojení analogového vstupu s ochranou na A/D převodník	33
Obr. 17 Struktura dat na sběrnici SPI.....	41
Obr. 18 Konfigurace projektu	44
Obr. 19 Konfigurace vstupů a výstupů karet.....	45
Obr. 20 Přiřazení programů a vizualizace v procesoru	46
Obr. 21 Vizualizace nádrží.....	47
Obr. 22 Vizualizace PSD regulátor	48
Obr. 23 Vizualizace β -PSD regulátor.....	49
Obr. 24 Vizualizace Feed Forward regulátor	50
Obr. 25 Model pro řízení výšky vodní hladiny a Power Panel	52
Obr. 26 Porovnání dat pro PSD regulátor	53
Obr. 27 Porovnání dat pro β -PSD regulátor.....	54
Obr. 28 Porovnání systému pro Feed Forward regulátor	55
Obr. 29 Zapojení konektoru MLW20	58

1. ÚVOD

Hlavním cílem bakalářské práce je vytvoření modelu simulujícího reálný technologický proces regulace výšky vodní hladiny v nádrži.

V práci se budu zabývat teorií ohledně regulace výšky vodní hladiny, popisu jednotlivých komponent systému, počínaje významem samotné nádrže a konče nejčastěji používanými regulátory. Konkrétní technologický proces nejprve popíši na matematickém modelu a schématu. Následně uvedu matematické rovnice popisující chování systému. Vytvořím simulaci modelu nádrže v programu Matlab Simulink. Vytvořím vhodné regulátory a ověřím teoretické předpoklady chování systému.

Pro simulovaný technologický proces navrhnu model obsahující mikročip. Vytvořím program, který nahraji do mikročipu a navrhnu elektronické schéma, které následně realizuji a vytvořím funkční tak model pro regulaci výšky vodní hladiny. Navrhnu a vytvořím řízení v programovatelném automatu typu Power Panel. Aplikaci do ní přidám vizualizaci a prvky umožňující nastavení parametrů regulátoru.

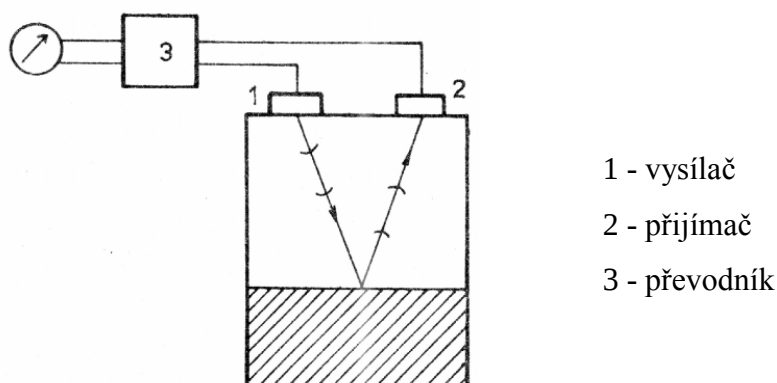
Otestuji funkčnost zařízení a porovnáím dosažené výsledky v Simulinku s výsledky dosaženými modelem řízeným programovatelným automatem. Vytvořím manuál pro obsluhu modelu a příklad zadání laboratorní úlohy pro model.

2. TEORETICKÝ ÚVOD

Regulace je definována jako udržování hodnot regulované veličiny podle daných podmínek a hodnot této veličiny zjištěných měření [1]. Úloha regulace výšky vodní hladiny v nádrži znamená udržování vodní hladiny na požadované úrovni. V praxi má nádrž nejčastěji význam zásobníku kapaliny sloužící k jejímu následnému odběru v nepravidelných intervalech a množství. Zásoba kapaliny v nádrži tyto výkyvy odběru absorbuje. Ve zvláštních případech je žadáným efektem hydrostatický tlak vodního sloupce, který je úměrný výšce vodní hladiny. Tento tlak může být využíván přímo nebo k regulaci průtokového množství. Tento princip nachází uplatnění především ve vodárnách a vodojemech. Úroveň hladiny v nádrži je snímána snímačem hladiny a přes řídicí mechanismus a akční prvek je zajištěna korekce skutečné výšky vodní hladiny.

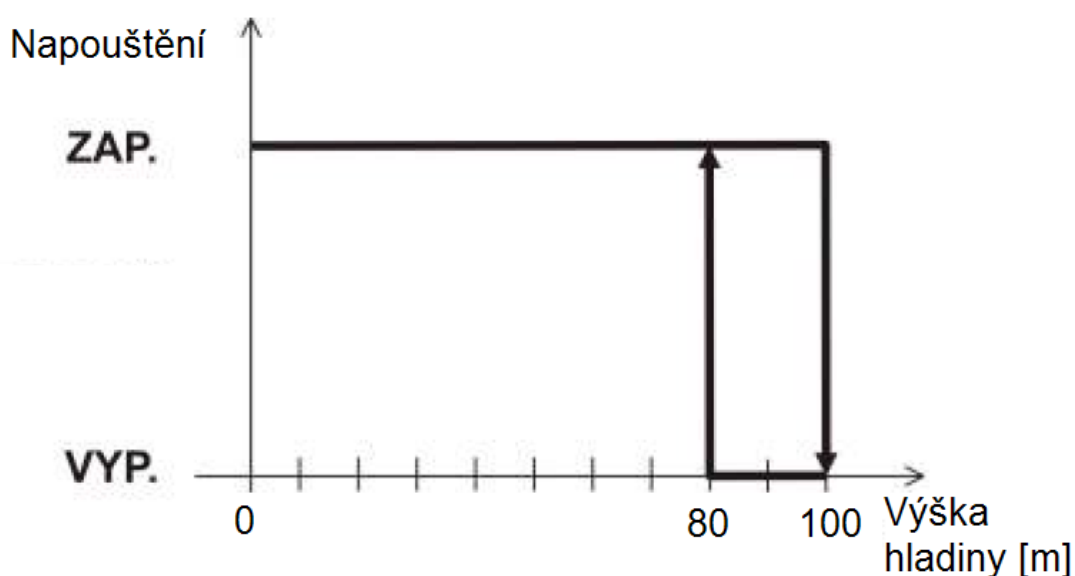
Pro účely měření výšky vodní hladiny existuje mnoho principů snímačů [2]. Nejpoužívanější snímače patří:

- Kapacitní hladinoměry, převádějící měření hladiny na měření kapacity
- Ultrazvukové snímače, měřící dobu letu ultrazvukové vlny
- Optické snímače, využívající odraz světelného záření od hladiny
- Hydrostatické hladinoměry, využívající působení hydrostatického tlaku nejčastěji na tenzometrický můstek
- Plovákové hladinoměry
- Další metody: pneumatické, konduktometrické, s ponorným tělesem



Obr. 1 Princip ultrazvukových hladinoměrů [2]

U jednoduchých technologických procesů, jako například regulace výšky hladiny je nejčastější použití dvoupolohové, tzv. ON/OFF regulace [3]. Základními parametry této regulace jsou žádaná hodnota a hystereze.

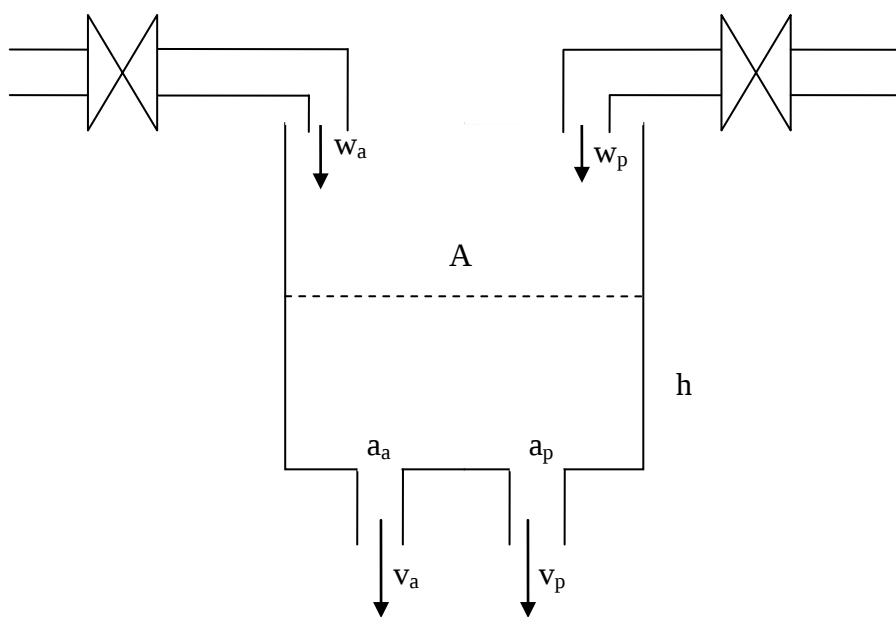


Obr. 2 Časový průběh ON/OFF regulace včetně hystereze

Pokročilejším řešením je použití PID regulátoru pro přesné řízení regulované veličiny. Toto řešení je obzvláště vhodné v kombinaci se servopohony. V aplikaci budou použity diskrétní ekvivalenty PID regulátoru. Výhoda oproti ON/OFF regulaci bude především v ustálení akčního zásahu při konstantní žádané hodnotě a konstantní působící poruše.

3. SIMULAČNÍ MODEL

Pro sestavení funkčního trenažeru technologického postupu bylo zvoleno řešení nádrže, jakožto integrátoru s několika vstupy. Tyto vstupy představují potrubí ovládaná ventilem. Dvě potrubí jsou realizována jako přítoková a dvě jako odtoková. Jedno přítokové a jedno odtokové potrubí je realizováno jako vstup poruchy vstupující do soustavy. Tato koncepce umožňuje univerzální použití a lze jí simulovat většinu reálných situací. Nádrž je vybavena snímačem výšky vodní hladiny, např. ultrazvukovým snímačem.



Obr. 3 Schéma modelu nádrže

APlocha hladiny válcové nádrže, $A = 0,255 \text{ m}^2$

hVýška vodní hladiny, $h_{\max} = 1 \text{ m}$

w_aMnožství přitékající vody

w_pMnožství přitékající vody z druhého potrubí (poruchy)

a_aPrůřez odtokového potrubí

a_pPrůřez odtokového potrubí působící poruchy

v_aMnožství odtékající vody

v_pMnožství odtékající vody působící poruchy

Objem vody v nádrži

$$V = h * A \quad [m^3] \quad (1)$$

$$V = h * A * 10^3 \quad [lit] \quad (2)$$

Množství odtékající vody [4]:

$$v_a = k * a_a \sqrt{2 * gh} \quad [m^3 s^{-1}] \quad (3)$$

$$v_p = k * a_p \sqrt{2 * gh} \quad [m^3 s^{-1}] \quad (4)$$

gGravitační zrychlení Země, $g = 9,81 \text{ ms}^{-2}$

kHydrodynamický součinitel, pro vodu $k = 0,94$

Bilanční rovnice [4]:

$$\frac{dV}{dt} = A \frac{dh}{dt} \quad (5)$$

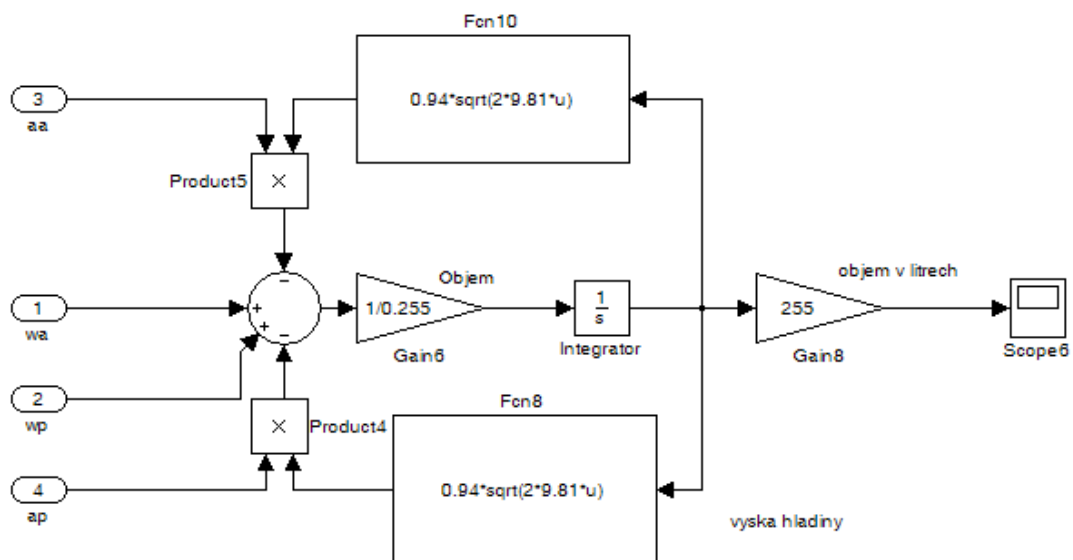
$$\frac{dV}{dt} = w_a + w_p - v_a - v_p \quad (6)$$

$$\frac{dV}{dt} = w_a + w_p - (a_a + a_p) * 0,94 \sqrt{2 * 9,81 h} \quad (7)$$

Z těchto rovnic je patrná nelinearita modelu. Zatímco přítoky jsou lineární, závislé na průtoku potrubím a nezávislé na výšce vodní hladiny. Odtok je úměrný průřezu potrubí i výšce hladiny. S rostoucí výškou hladiny roste i odtékající množství.

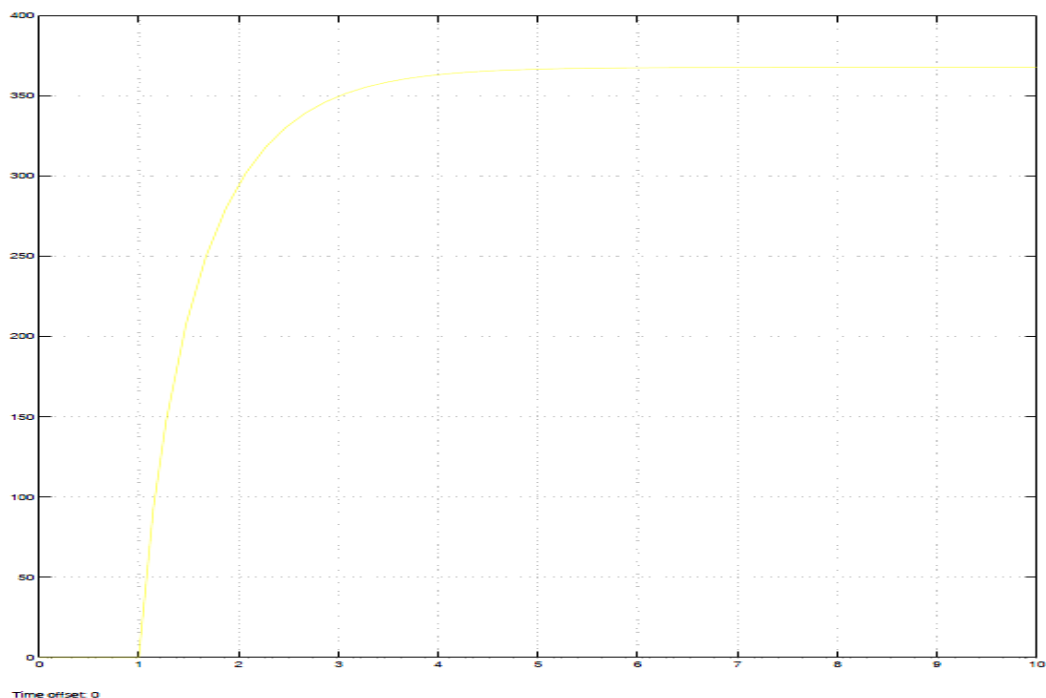
3.1 MODEL SIMULINK

Pro potvrzení teoretických poznatků a ověření vlastností soustavy je vytvořen model systému v programu Simulink.



Obr. 4 Simulační schéma modelu nádrže

Vstupy systému odpovídají parametrů v rovnici (7). Integrátor integruje výšku vodní hladiny.



Obr. 5 Odezva na skokovou změnu přítoku

Na vstup systému byl přiveden jednotkový skok w_a . Parametr průřezu potrubí a_a měl hodnotu 0,2. Z průběhu je patrná nelinearita systému. K ustálení výšky vodní hladiny dojde v momentě vyrovnání přítoku a odtoku.

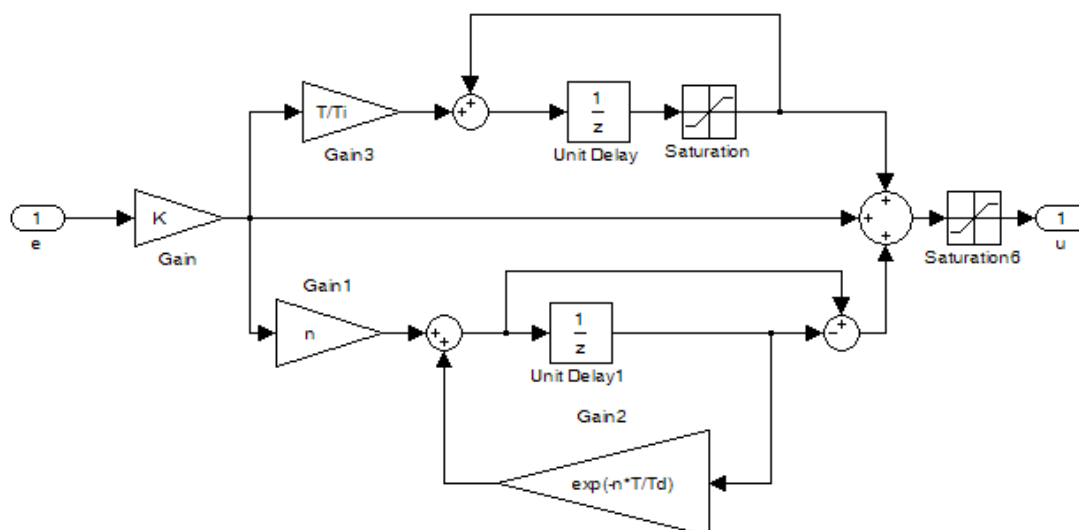
3.2 REGULACE

Pro účely regulace výšky vodní hladiny je potřeba implementovat do řídicího algoritmu regulátory. Jedná se o 3 hlavní ty PSD, β -PSD a Feed Forward regulátory [5] [6]. Perioda vzorkování $T = 1s$. tato perioda je volena vzhledem k akčním prvům v soustavě, což jsou ventily a nejsou žádoucí velmi časté a rychlé změny polohy.

3.2.1 PSD s filtrací derivační složky a omezením sumační složky

Přenos PSD regulátoru v Z transformaci:

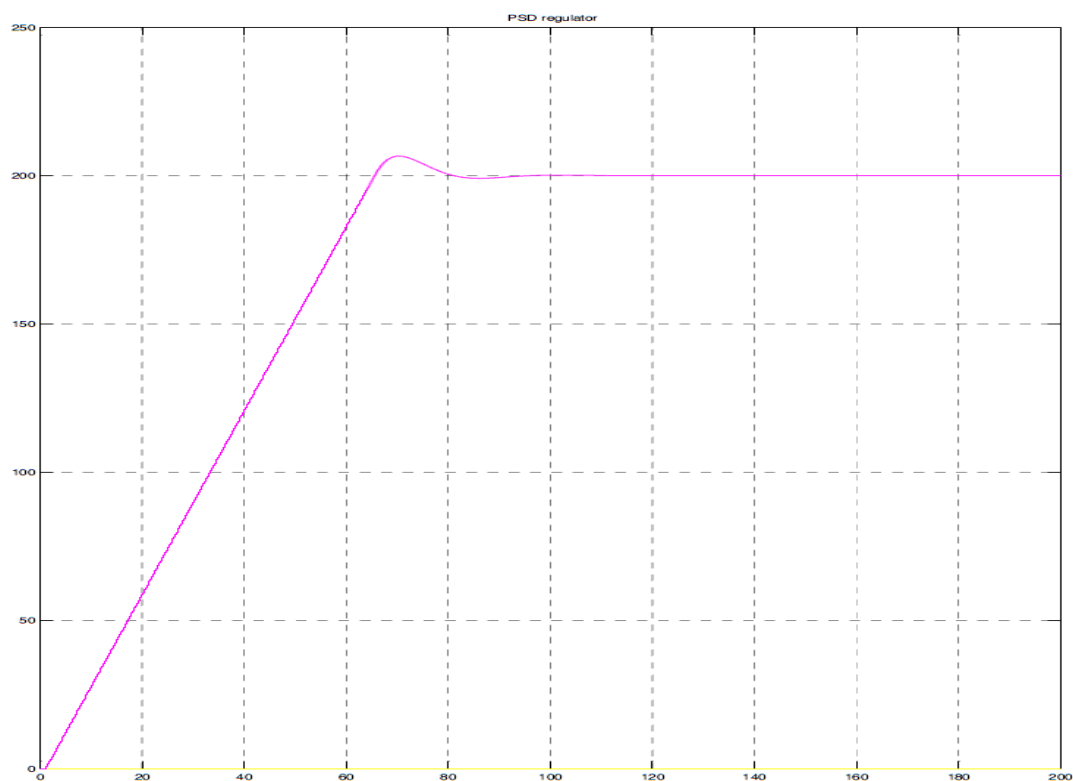
$$F_R(z) = K * \left(1 + \frac{Tz^{-1}}{T_i(1-z^{-1})} + n \frac{1-z^{-1}}{1-e^{-\frac{TN}{T_d}}} \right) \quad (8)$$



Obr. 6 Struktura PSD regulátoru s filtrací derivační složky

Parametry PSD regulátoru:

$$K = 50; T_i = 50; n = 3; T_d = 0,125; T = 1$$

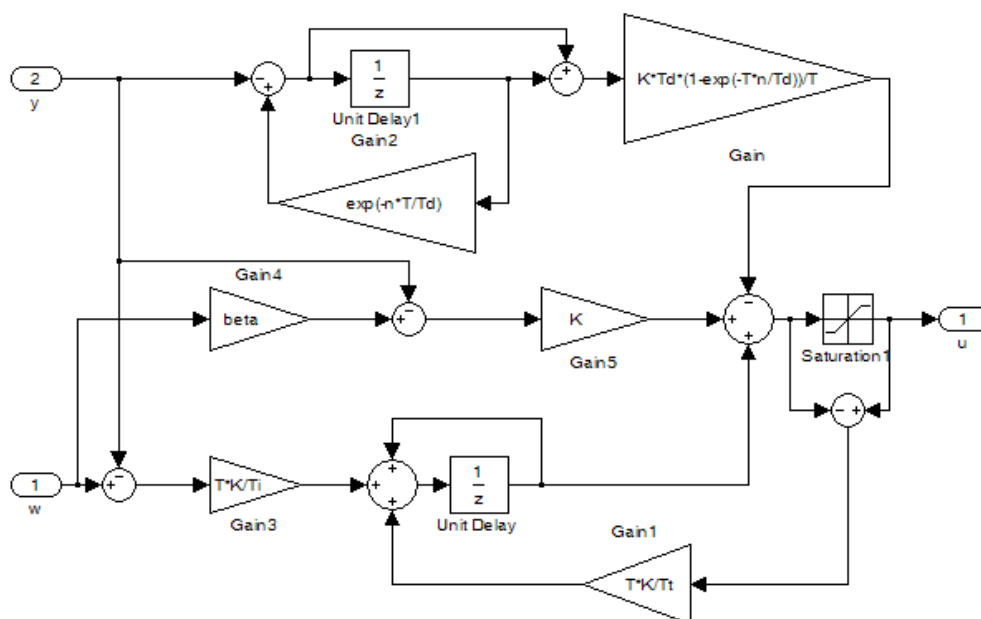


Obr. 7 Výstup soustavy na skokovou změnu žádané hodnoty s PSD

3.2.2 β -PSD regulátor

Výstup β -PSD regulátoru v Z transformaci:

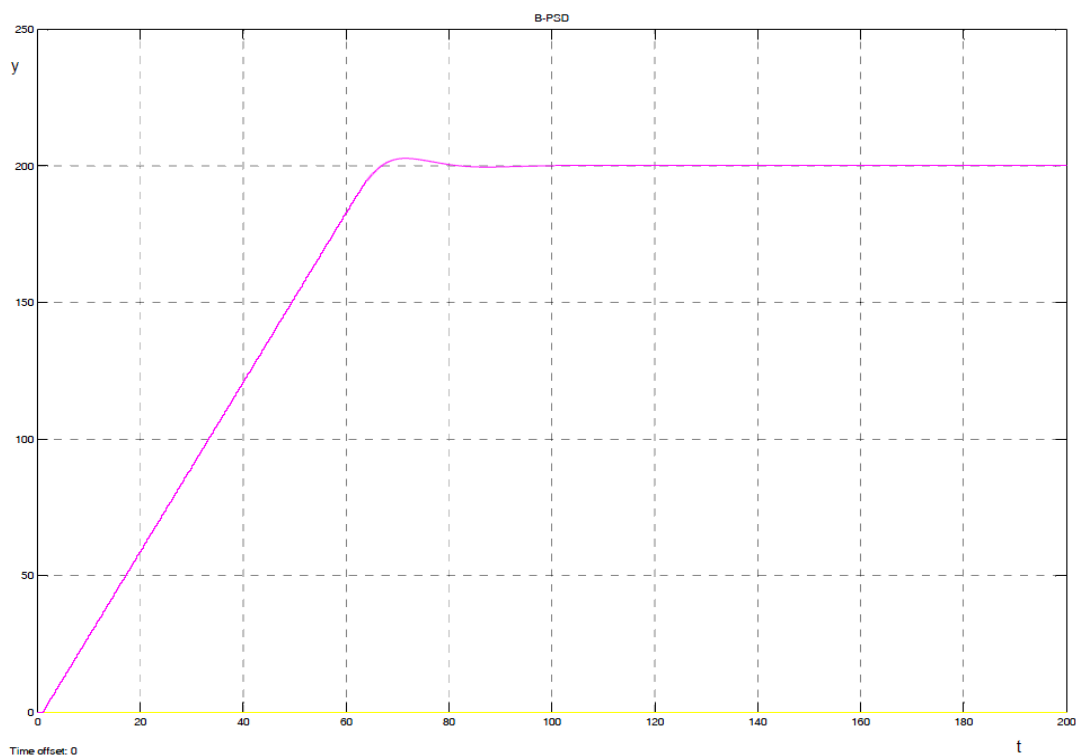
$$U(z) = K(\beta W(z) - Y(z) + \frac{Tz^{-1}}{T_i(1-z^{-1})} E(z) - \frac{T_d}{T} (1 - e^{-\frac{T_N}{T_d}}) \frac{1-z^{-1}}{1-e^{-\frac{T_N}{T_d}}} Y(z)) \quad (9)$$



Obr. 8 Struktura β -PSD regulátoru

Parametry β -PSD regulátoru:

$K = 50$; $T_i = 50$; $n = 3$; $T_d = 0,125$; $T = 1$; $\beta = 1$

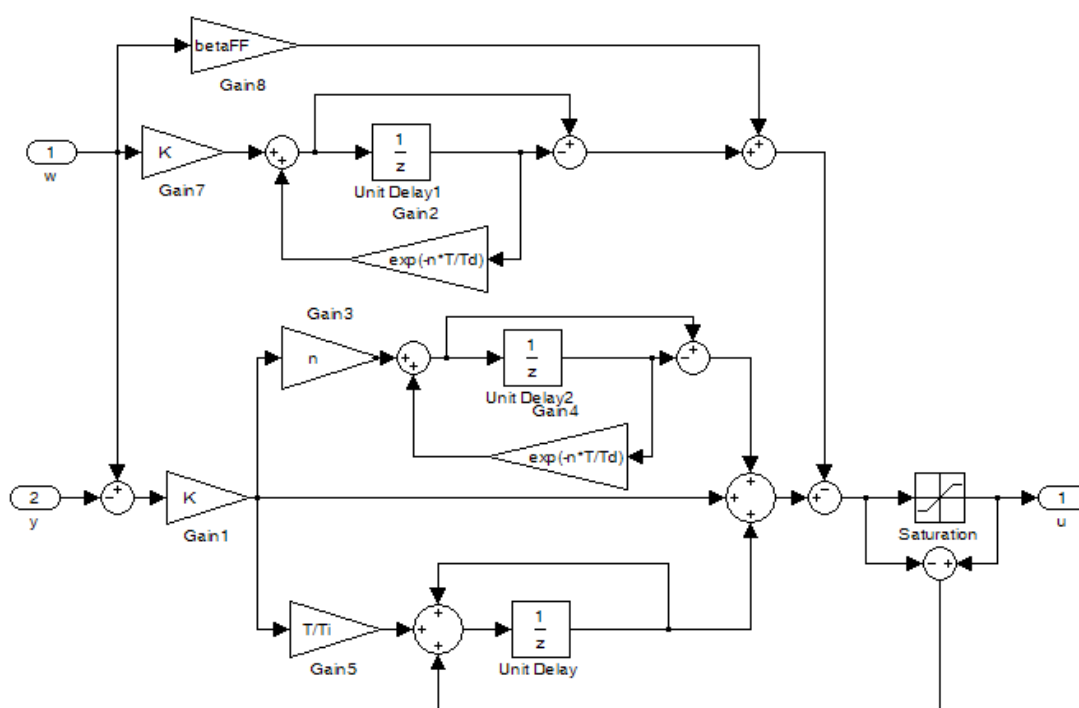


Obr. 9 Výstup soustavy na skokovou změnu žádané hodnoty s β -PSD

3.2.3 Feed Forward regulátor

Výstup Feed Forward regulátoru v Z transformaci:

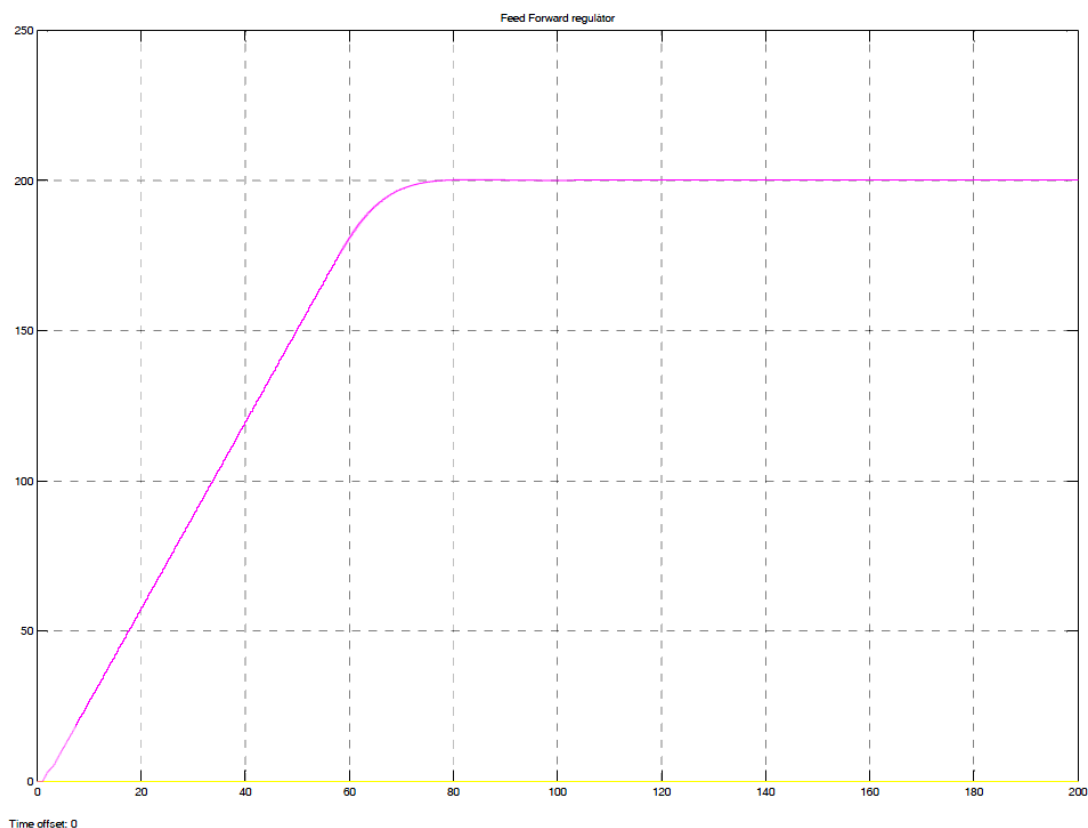
$$U(z) = U_R(z) - U_{FF}(z) = K \left((W(z) - Y(z)) + \frac{Tz^{-1}}{T_i(1-z^{-1})} E(z) - \frac{T_d}{T} \left(1 - e^{-\frac{TN}{T_d}} \right) \frac{1-z^{-1}}{1-e^{-\frac{TN}{T_d}}} (W(z) - Y(z)) \right) - K(\beta W(z) + \frac{T_d}{T} (1 - e^{-\frac{TN}{T_d}}) \frac{1-z^{-1}}{1-e^{-\frac{TN}{T_d}}} W(z)) \quad (10)$$



Obr. 10 Struktura Feed Forward regulátoru

Parametry Feed Forward regulátoru:

$$K = 50; T_i = 100; n = 3; T_d = 0,125; T = 1; \beta_{FF} = 1$$



Obr. 11 Výstup soustavy na skokovou změnu žádané hodnoty s Feed Forward

4. REÁLNÝ MODEL

Cílem modelu vodní hladiny je věrná simulace reálného procesu. Tento model simuluje reálný technologický proces spojený s úpravou pitné vody, kdy se do čisté vody přidávají příměsi jako chlór pro její dezinfekci a úpravu. Model umožňuje plnohodnotnou regulaci výšky vodní hladiny v nádrži. Největší výhodou oproti reálnému modelu je jeho snadná realizace a provoz. Není třeba zpracovávat skutečnou fyzikální veličinu. Tímto se vyhneme problémům spočívajících v rizicích s tím spojených. V našem případě nemusíme řešit složité otázky bezpečnosti práce s nebezpečnými látkami, jako například chlór, a rizika ochrany elektrických obvodů proti nežádoucím kapalinám. WEB-B8

4.1 FYZICKÝ VZHLED MODELU

Primárním okruhem procesu je hlavní potrubí, ve kterém je přiváděna čistá voda. Objem protékajícího množství je v čase proměnný, to lze ve skutečném procesu přičíst například aktuální spotřebě pitné vody. Potrubí ústí do míchací nádrže. Do míchací nádrže je rovněž přiváděna směs příměsí. Množství přitékajících příměsí je pevně dáno ventilem, který vypouští směs z hlavní nádrže. Otevření ventilu je pevně svázáno s protékajícím množstvím čisté vody v hlavním potrubí.

4.1.1 Konektor MLW20

Pro účely komunikace modelu s nadřazeným prvkem v systému, což je v našem případě Power Panel firmy B&R je model vybaven konektorem MLW20. Zapojení pinů je provedeno podle dohodnutého standardu laboratoře CLG.

Tab. 1 Standardní zapojení konektoru MLW20 v laboratoři CLG

PIN	Značení	Popis
1	24V	24V z PLC z karty BR9300
2	24V	24V z PLC z karty BR9300
3	GND	GND z PLC z karty BR9300
4	GND	GND z PLC z karty BR9300
5	-	nezapojeno
6	AI_1	Analogový vstup do PLC
7	AI_2	Analogový vstup do PLC
8	AI_3	Analogový vstup do PLC
9	AO_1	Analogový výstup z PLC
10	AO_2	Analogový výstup z PLC
11	AO_3	Analogový výstup z PLC
12	-	nezapojeno
13	DI_1	Digitální vstup do PLC
14	DI_2	Digitální vstup do PLC
15	DI_3	Digitální vstup do PLC
16	DI_4	Digitální vstup do PLC
17	DO_1	Digitální výstup z PLC
18	DO_2	Digitální výstup z PLC
19	DO_3	Digitální výstup z PLC
20	DO_4	Digitální výstup z PLC

V modelu nádrže jsou použity pouze potřebné vývody a ostatní jsou nezapojeny. V následující tabulce je přehled zapojených pinů v modelu.

Tab. 2 Zapojení konektoru MLW20 v modelu nádrže

PIN	Značení	Popis
1	24V	24V z PLC z karty BR9300
2	24V	24V z PLC z karty BR9300
3	GND	GND z PLC z karty BR9300
4	GND	GND z PLC z karty BR9300
5	-	nezapojeno
6	AI_1	HLADINA Analogový vstup do PLC
7	AI_2	PORUCHA PŘÍTOK Analogový vstup do PLC
8	AI_3	PORUCHA ODTOK Analogový vstup do PLC
9	AO_1	AKČNÍ ZÁSAH Analogový výstup z PLC
10	-	nezapojeno
11	-	nezapojeno
12	-	nezapojeno
13	-	nezapojeno
14	-	nezapojeno
15	-	nezapojeno
16	-	nezapojeno
17	-	nezapojeno
18	-	nezapojeno
19	-	nezapojeno
20	-	nezapojeno

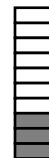
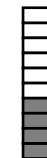
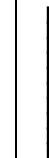
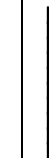
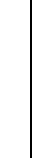
4.1.2 LED signalizace

Model nádrže obsahuje několik druhů optických indikací probíhajícího procesu. Tyto indikace jsou základním prvkem umožňujícím identifikovat aktuální stav modelu a právě probíhající procesy. Hlavní částí optické signalizace jsou skupiny informativních LED diod. Zvláštní pozornost je třeba věnovat výstražným signalizacím a signalizacím upozorňujícím na nestandardní jevy při řízení modelu.

Sloupcová indikace objemu směsi

Jednotka je tvořena deseti LED diodami zelené barvy uspořádanými horizontálně vedle sebe. Slouží k informaci o stavu směsi v nádrži. Světlo LED diod má charakter BAR grafu, s narůstajícím objemem narůstá počet rozsvícených diod. Jednotlivé diody se rozsvěcejí v lineárním rozložení na celkový objem nádrže. Rozlišení mezi jednotlivými stavy je 1/10 objemu nádrže. Stav jednotlivého ukazatele se mění při překročení střední hodnoty jeho intervalu. Podrobný výpis indikace je zobrazen v **Tab. 1**. Například pro indikaci nejnižšího stavu – spodní dioda – pro rozsah 0 až 10% nádrže se tato dioda rozsvítí po překročení hranice 5%.

Tab. 3 Sloupcová LED indikace objemu

Objem nádrže [%]	≥ 5	≥ 15	≥ 25	≥ 35	≥ 45	≥ 55	≥ 65	≥ 75	≥ 85	≥ 95
Odpovídající objem nádrže [l]	12,75	38,25	63,75	89,25	114,75	140,25	165,75	191,25	216,75	242,25
Indikace										

Krajní stavy nádrže

Pro přesnou identifikaci krajních stavů nádrže jsou implementovány 2 modré LED diody. Slouží k rychlé a jednoznačné informaci pokud je nádrž úplně prázdná nebo úplně plná.

První LED dioda (dolní) svítí, pokud je nádrž úplně prázdná – objem směsi 0 litrů. Pokud je v nádrži libovolné nenulové množství směsi, je tato dioda zhasnutá.

Druhá LED dioda (horní) svítí, pokud je nádrž úplně plná – objem nádrže 255 litrů. Při jakémkoliv menším objemu směsi v nádrži je kontrolka zhasnutá.

Indikace otevřených ventilů u nádrže

Pro lepší kontrolu akčního zásahu a působících poruch v podobě přítoku a odtoku z nádrže obsahuje model 4 LED diody a každá je přiřazena jednomu ventilu v modelu nádrže. Pokud je ventil v jiné poloze než zavřený, příslušná LED dioda svítí.

Chyba regulace

Model rovněž obsahuje indikaci nevhodného nastavení řízených veličin. Z toho lze usuzovat, že se pravděpodobně jedná o chybnou regulaci a model přišel do nepřírodního stavu. Jedná se především o stavy, kdy je nádrž plná a stále aktivní přítok nebo je nádrž prázdná a jsou stále aktivní ventily odtoku. Kontrolka na tuto situaci upozorní a je již na obsluze, jak na tuto situaci zareaguje. Nelze vyloučit případ, že zmíněná situace je žádaná a nejde o omyl či chybu v regulaci.

4.1.3 LCD displej MC1602E-SYL/H

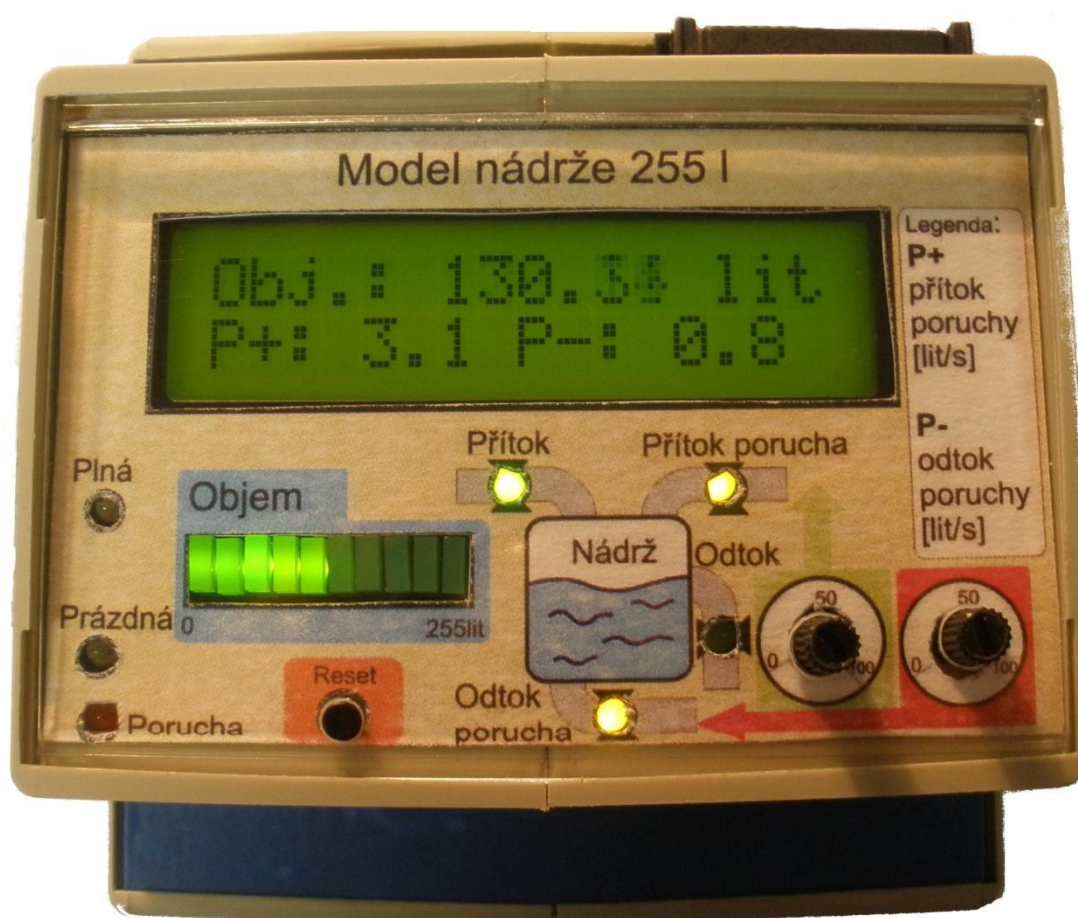
Implementovaný BAR graf a další LED signalizace nemusí vždy přesně vystihnou vnitřní stav modelu, proto je přípravek vybaven alfanumerickým LCD displejem MC1602E-SYL/H od firmy Everbouquet. Displej disponuje polem LED podsvícení a vlastním řadičem. Řadič je kompatibilní se standardním řadičem HITACHI HD44780 a umožňuje tak univerzální použití. Lze komunikovat pomocí 4 nebo 8 bitů. V zájmu úspory vodičů budeme používat pro komunikaci 4bit. Displej poskytuje rozlišení 2x16 znaků a je tedy možné zobrazit mnohem více přesných informací o modelu.

Na prvním řádku displeje je zobrazen text „Obj: XXX.XX lit“, kde XXX.XX je aktuální objem nádrže v litrech s přesností na 2 desetinné místa.



Obr. 12 LED diody a LCD displej s BAR grafem

Na druhém řádku je rozdílný text v závislosti na působení poruchy. Pokud porucha nepůsobí, je řádek použit jako sekundární BAR graf, který sloupcově zobrazuje výšku vodní hladiny, jak lze vidět na **Obr. 12**. Pokud je alespoň jedna z působících poruch nenulová (ovládání ručně pomocí otočných knoflíků - trimrů) je na spodním řádku displeje vypsána velikost působících poruch v jednotkách lit/s. Syntaxe zápisu je následující „P+: X.X P-: Y.Y“ kde X.X vyjadřuje velikost kladné poruchy a Y.Y velikost záporné poruchy působící na nádrž, oboje v jednotkách lit/s.

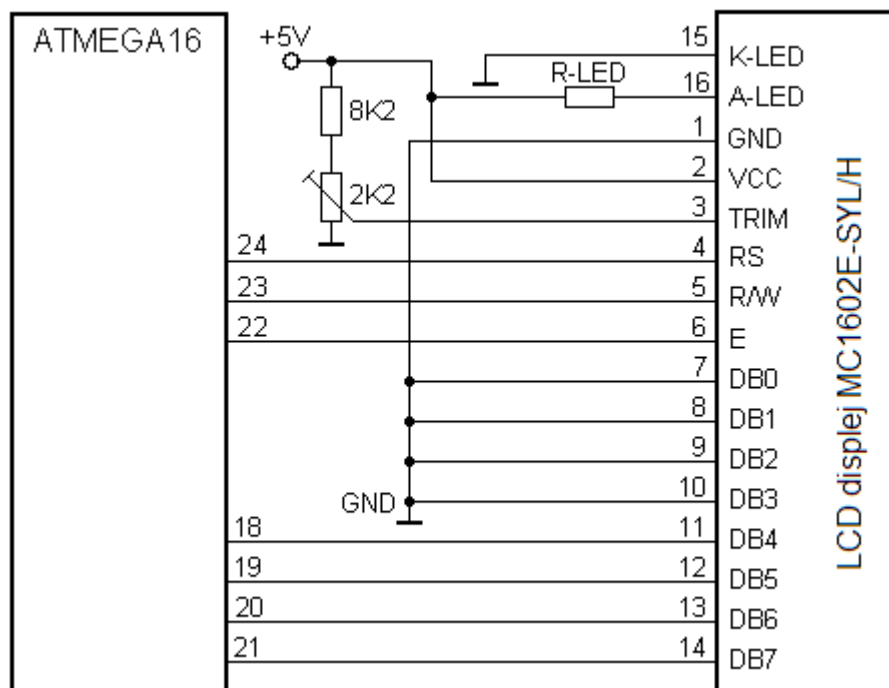


Obr. 13 LCD displej s informací o působící poruše

Kontrast displeje je nastavitelný. Volba se provádí nastavením trimru 2k2. Jas podsvícení displeje je pevně daný. Proud podsvícení je určen odporem R_{LCD} o hodnotě 6Ω a lze jeho hodnotu vypočítat:

$$I_{LCD} = \frac{U_{CC} - U_{LED}}{R_{LCD}} \quad (11)$$

$$I_{LCD} = \frac{5 - 4,2}{6} = 133,33 \text{ mA} \quad (12)$$



Obr. 14 Zapojení LCD displeje

4.1.4 Vstupy poruchy do nádrže

Model nádrže je jakožto regulovaná soustava vybaven prostředky pro modelování působící poruchy na nádrž. Tato porucha je realizována jako dvojice ventilů ovládaných ručně z modelu. V případě otevření prvního ventilu „Porucha přítok“, přitéká navíc do nádrže takové množství vody, úměrné nastavení ventilu. Snímač nastavené poruchy je odporový trimr 50kΩ připojený na napájecí napětí 5V. Nastavitelný vývod je připojen s filtračním kondenzátorem 10nF ke vstupu AD převodníku ADC1 u ATMEGA 16 a velikost přítoku poruchy je přímo-úměrná nastavené hodnotě na trimru.

$$Q_{p+} = \frac{P_{ADC+}}{80} \quad (13)$$

Q_{p+} Velikost kladné poruchy [lit/s]

P_{ADC+} 8-bit hodnota poruchy získaná z ADC

Maximální hodnota $Q_{p-max} = 3,19$ lit/s

V případě otevření druhého ventilu „Porucha odtok“, odtéká z nádrže navíc takové množství vody, úměrné nastavení ventilu a aktuální výšce hladiny. Snímač nastavené poruchy je odporový trimr $50k\Omega$ připojený na napájecí napětí 5V. Nastavitelný vývod je připojen s filtračním kondenzátorem 10nF ke vstupu AD převodníku ADC2 u ATMEGA 16 a průřez potrubí (ventilu) záporné poruchy je přímo-úměrné nastavené hodnotě na trimru.

$$a_p = P_{ADC-} * 0.002829 \quad (14)$$

a_p Průřez odtokového ventilu poruchy [m^2]

P_{ADC-} 8-bit hodnota poruchy získaná z ADC

Maximální hodnota $a_{pmax} = 0,721$ m^2

$$Q_{p-} = a_p * k\sqrt{2gh} \div 1000 \quad (15)$$

Q_{p-} Velikost záporné poruchy [lit/s]

k Hydrodynamický součinitel, $k = 0,94$ (voda)

g Gravitační zrychlení [m/s^2], $g = 9,81$ m/s^2

h Výška vodní hladiny [m]

Maximální hodnota $Q_{p-max} = 3,0$ lit/s

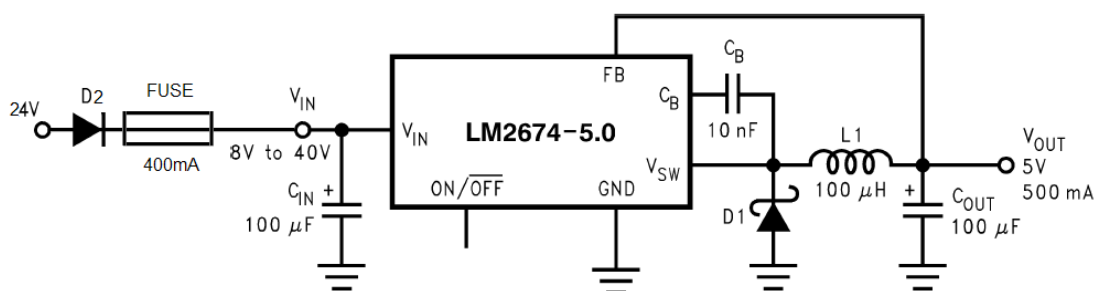
V rovnicích 1 a 3 jsou použity číselné konstanty. Jejich velikosti jsou voleny úměrně k parametrům modelu, aby maximální hodnoty obou veličin byly přibližně 3 lit/s. Realizace trimrů lze vidět na **Obr. 12** a **Obr. 13**.

4.1.5 Spínaný zdroj

Napájecí napětí přiváděné na model nádrže má velikost 24V. Tato velikost je ovšem nepřijatelná pro použitou elektroniku, obzvlášť mikrokontrolér, DA převodník a LCD displej. Z tohoto důvodu je třeba přeměnit napětí na velikost 5V.

K tomuto účely byl použit stabilizovaný zdroj napětí 5V typu 7805. Velikost napětí byla vyhovující, ovšem problém představoval úbytek napětí, který neúměrně zahříval stabilizátor a nebylo možné jej použít v plastové krabici modelu, protože by došlo k její destrukci vlivem nepřiměřeně vysoké teploty a rovněž by mohly být poškozeny další vnitřní části modelu jako například LCD displej.

Řešení se objevilo v podobě spínaného zdroje typu LM2674 [7] o frekvenci 260kHz v pouzdře SO8. Vyhovuje dostatečnou stabilitou napětí 5V, dostatečným proudem 500mA a díky vysoké účinnosti 90% netrpí nárůstem teploty. Obvod je doplněn o usměrňovací diodu D2 jako ochranu proti přepólování a tavnou pojistku 400mA jako ochranu proti nadproudu.



Obr. 15 Schéma zapojení spínaného zdroje 5V s ochranami

4.1.6 DA převodník MCP4922

Komunikace mezi modelem nádrže a Power Panelem probíhá analogově v úrovních 0 - 5V. Model obsahuje 3 výstupní veličiny, které jsou přiváděny na analogové vstupy karet připojených k Power Panelu. Přenáší se informace o aktuální výšce hladiny v litrech, o aktuálním poruchovém přítoku do nádrže a aktuálním poruchovém odtoku z nádrže, oboje v litrech/s.

Výhodná situace je pouze u poruchového přítoku. Ten se ovládá ručně pomocí trimru a výstupní hodnota napětí z tohoto trimru je v rozsahu 0 - 5V a přímo udává rychlost přítoku kladné poruchy v lit/s. Situace je vyřešena přivedením výstupu z trimru nejen na vstup AD převodníku v mikrokontroléru pro výpočet, ale rovnou

také na výstupní pin modelu, odkud jsou data čteny Power Panelem a ten získává informaci o aktuální působící kladné poruše. V dalších dvou případech (výška hladiny a záporná porucha) jsou data buď interpretována pouze v mikročipu, nebo je třeba čtenou hodnotu z trimru záporné poruchy nejprve přepočítat dle rovnic (2) a (3) a následně poskytnou na výstup modelu v podobě analogové veličiny.

K účelu převedení digitálně vyjádřené veličiny na analogovou je v modelu zapojen DA převodník MCP4922 [8]. Jedná se o 12 bitový dvoukanálový DA převodník s rozhraním SPI v pouzdře DIL14. Jako referenční napětí je použito napájecí napětí. Výstup A reprezentuje výšku vodní hladiny a výstup B reprezentuje zápornou poruchu. Ke každému výstupu je ještě připojen vyhlazovací kondenzátor 20nF. DA převodník je zapojen v režimu Slave, proto se budeme vlastním přenosem zabývat v části mikrokontrolér. Schéma zapojení DA převodníku je přiloženo v příloze.

SPI (Serial Peripheral Interface) rozhraní je sériové periferní zařízení používané pro sériovou komunikaci mezi mikroprocesory a periferními zařízeními jako je v tomto případě DA převodník. Na SPI sběrnici je Master (mikročip) a jeden popřípadě více Slave (např. DA převodník). Aktivace Slave obvodu se provede pomocí pinu $\overline{SS}$ který aktivuje obvod v log 0. Dále je použit společný taktovací signál SCLK a datové vodiče MISO (Master In, Slave Out) a MOSI (Master Out, Slave In).

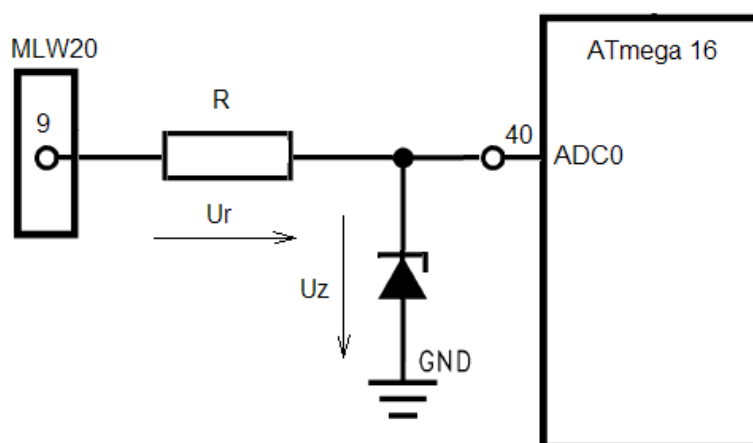
4.1.7 Mikrokontrolér ATmega 16

Hlavním prvkem modelu nádrže je jeho mikročip, který představuje veškerý výpočetní výkon a jsou v něm implementovány algoritmy chování reálného systému. Je použit mikroprocesor z řady AVR a to ATmega 16 od firmy Atmel [9] [10]. Přesné označení je ATMEGA16 - 16PU 0811K v pouzdře DIL40. Obsahuje 16kB programové flash paměti, 1kB RAM paměti a 512B EEPROM paměti. Dále nabízí 2x 8bit a 1x 16bit čítač časovač s podporou PWM, časovač Watch-Dog, JTAG, SPI a UART rozhraní a osmikanálový 10bit AD převodník s postupnou aproximací. Obsahuje čtyři 8bit vstup/výstupní porty PORTA až PORTD. Je implementován interní RC oscilátor 1MHz, ale v této aplikaci je softwarově odpojen a na vstup je připojen

externí krystalový oscilátor 16MHz udělující hodinový kmitočet. Pro retenční napětí A/D převodníku je použito napájecí napětí z vývodu A_{VCC} . A na vstup A_{REF} je připojen blokovací kondenzátor 20nF.

Schéma zapojení mikrokontroléru je uvedeno v příloze 1.

Na vstup ADC0 A/D převodníku je přiváděn analogový signál z karty B&R. Výstup na této kartě je v rozsahu -10 až 10V. Vstup A/D převodníku dovoluje vstup pouze v rozmezí 0 až 5V a značné překročení tohoto rozsahu by mohlo vést ke zničení celého mikrokontroléru. Z tohoto důvodu bylo potřeba vyvinout řešení k omezení analogové veličiny do přípustných mezí. Použila se Zenerova dioda $U_Z = 5,1 \text{ V}$ s předřadným odporem $R = 560 \Omega$.



Obr. 16 Schéma zapojení analogového vstupu s ochranou na A/D převodník

Maximální kladné napětí na vstupu ADC0 = 5,1 V

Maximální záporné napětí na vstupu ADC0 = -0,7 V

Proud diodou v závěrném směru:

$$I_R = \frac{U - U_Z}{R} \quad (16)$$

$$I_R = \frac{10 - 5,1}{560} = 8,75 \text{ mA} \quad (17)$$

Proud diodou v propustném směru:

$$I_F = \frac{U - U_D}{R} \quad (18)$$

$$I_F = \frac{10 - 0,65}{560} = 16,7 \text{ mA} \quad (19)$$

4.2 IMPLEMENTACE VÝPOČETNÍHO ALGORITMU

Hlavní program *main* spuštěný primárně v mikrokontroléru slouží především pro provedení inicializace a nastavení parametrů. Program se poté dostane do nekonečné smyčky, ve které je mikrokontrolér uváděn do režimu snížené spotřeby. Používáme režim Idle, jde o základní režim, kdy se zastaví kmitočet hodin pro procesor, ale ostatní funkce zůstávají zachovány. Jde například o přerušení, I/O porty, Watch dog, čítače-časovače...

Než se procesor uspí, je provedena inicializace. Nejdříve provedeme nastavení počátečních hodnot proměnných používaných v programu, obzvláště se jedná o proměnnou hladina. Poté provedeme nastavení vstup-výstupních portů. ATmega16 obsahuje rozhraní JTAG, které je implicitně povoleno. Rozhraní používá piny PORTC2 až PORTC5. Pro využívání těchto portů jako klasických vstup-výstupních je potřeba JTAG zakázat. Toto se provede zapsáním log 1 na bit JTD v registru MCUCSR. Mikročip obsahuje ochranu proti možnému náhodnému zakázání a proto je potřeba tuto instrukci provést 2x během 4 po sobě jdoucích strojních taktů. V mikročipu budeme využívat funkci Watch-dog a v inicializační části provedeme její nastavení a spuštění. Podle použitého krystalu nastavíme interval pro čítač Watch-dog. Zápisem log 1 na bity WDP2 a WDP0 v registru WDTCR získáme interval Watch-dog resetu přibližně 0,52s. Před povolením funkce Watch-dog preventivně vynulujeme registr Watch-dogu příkazem „WDR“ (příkaz je v Assembleru) a poté Watch-dog spustíme zápisem log 1 na bit WDE v registru WDTCR. Povolíme režim snížené spotřeby zápisem log 1 na bit SE v registru MCUCR a provedeme nastavení režimu, předdělicího poměru a vrcholu čítače pro čítač-časovač 1. Toto nastavení je uvedeno v kapitole 3.2.1.

V inicializaci A/D převodníku je nastavena reference na A_{VCC} . Je nastaveno zarovnání výsledku doleva, čímž oddělíme nejméně významné 2bity a dále budeme používat 8bit hodnotu. Zvolíme předděličku kmitočtu /128 čímž je takt převodníku 125kHz a je dosaženo rozmezí frekvence 50kHz až 200kHz kdy je převod nejpřesnější. Je nastaven kanál 0, který budeme přepínat v programu pro obsluhu přerušení. Povolíme přerušení od A/D převodníku, nastavíme Enable A/D převodníku a nastavíme kontinuální převod. Toto znamená, že další cyklus převodu je spuštěn automaticky při příchodu žádosti o přerušení. Posledním příkazem pro A/D převodník je nastavení bitu zahajujícího převod.

V inicializačním programu je následně provedena inicializace SPI rozhraní. K tomuto účelu slouží registr SPCR. V tomto registru povolíme SPI přenos (bit SPE), nastavíme zařízení jako master (bit MSTR). Dělička kmitočtu je nastavena na 16, tím je dosaženo taktu SPI rozhraní 1MHz (bit SPR1=0 SPR0=1).

Pro správnou funkci LCD displeje je potřeba jej nejprve správně inicializovat. K tomuto účelu je v knihovně lcd.h několik funkcí. Proveďte se volba šířky komunikačního portu na 4 vodiče a displej je vymazán. Následně jsou na displej nahrána data zobrazená v počátečním stavu modelu. Pokud nepůsobí porucha je na displeji údaj o výšce vodní hladiny a BAR graf indikující tuto hladinu. Pokud působí porucha je BAR graf nahrazen textem informujícím o těchto poruchách. Přesná syntaxe je uvedena v kapitole LCD displej.

Na závěr je povolena globální maska přerušení příkazem sei() a v nekonečné smyčce uspíme mikročip do režimu Idle. Nyní budou vykonávána pouze přerušení od čítačů-časovačů.

4.2.1 Časová základna

Pro práci s průtokem vyjádřeným protékajícím množstvím za jednotku času a nádrží, jakožto integrátorem, je potřeba pracovat nejen s hodnotou veličiny jako ta-

kovou, ale také s časem, po který tato veličina působí. V modelu nastává otázka realizace časové oblasti.

U ATmega16 je nejelegantnější řešení pomocí čítače-časovače. Čítač-časovač nabízí několik režimů funkce, ale pro naši potřebu je nejvhodnější provázání s hlavním kmitočtem hodin. Kmitočet hodin je nejprve dělen pomocí přednastavené předděličky kmitočtu. Odtud jdou impulsy do registru časovače a každým dalším příchozím pulzem se hodnota registrů časovače zvýší o jedničku. Hodnota se zvyšuje až do doby dosažení námi zadané hodnoty. V případě povoleného přerušení se v tento moment vygeneruje příznak přerušení. Registr čítače se vynuluje a začíná se znovu čítat.

V našem konkrétním případě jsme zvolili časový krok 0,1s. Budeme tedy požadovat, aby příznak přerušení přišel každé 0,1s a my tak měli přesný údaj o čase. U mikroprocesoru využíváme externí krystal pracující na frekvenci 16 MHz.

Pro náš případ je nejvhodnější použít čítač-časovač 1. Hlavní výhodou je 16 bitový registr. Zbývající čítače-časovače jsou pouze 8bitové a zpracovávat časový interval 0,1s by bylo komplikovanější. Nastavíme režim čítače 12 – Clear Timer on Compare Match (CTC). Nastavení se provede v registru TCCR1B nastavením bitů WGM12 a WGM13 do log 1. V tomto režimu se čítač vynuluje po dosažení hodnoty zapsané v registru ICR1. Pro získání přerušení s intervalem 0,1s prvně určíme počet taktů procesoru uplynulých během této doby.

$$N = T_{vz} * f_{kr} \quad (20)$$

N.....potřebný počet taktů procesoru

T_{vz}.....Požadovaná perioda přerušení 0,1s

f_{kr}.....frekvence krystalu mikroprocesoru

$$N = 0,1 * 16 * 10^6 \quad (21)$$

$$N = 60 * 10^5 \quad (22)$$

Dalším krokem je určení správného nastavení předděličky kmitočtu. Výhodné je volit co nejvyšší poměr, ale s ohledem na nulový zbytek po dělení. Nevhodným zvolením předdělicího poměru bychom dosáhli desetinného čísla provedených čítání. Výsledkem by byla nutnost zaokrouhlení a v konečném důsledku bude časový údaj nepřesný.

V našem případě zvolíme předděličku kmitočtu 256. Nastavení provedeme zapsáním log 1 na bit CS12 v registru TCCR1B.

$$Imp = \frac{N}{Pred} \quad (23)$$

Imp.....požadovaný počet čítání

Pred.....předdělicí poměr

$$Imp = \frac{16 \cdot 10^5}{256} \quad (24)$$

$$Imp = 6250 \quad (25)$$

Tímto jsme získali celočíselné číslo udávající počet kroků čítače pro dosažení stanoveného intervalu 0,1s. Zde je důvod, proč jsme zvolili 16bitový čítač, 8bitový by svým rozsahem nedostačoval (maximální číslo 255).

Čítač-časovač 1 obsahuje registr ICR1 pro nastavení vrcholu pro čítač. Do tohoto registru zapíšeme vypočtené číslo zmenšené o 1. Je to z důvodu čítání včetně nuly. Tuto hodnotu není třeba nijak cyklicky měnit po každém přerušení od čítače. Čítací registr je oddělený a pouze se pokaždé s touto hodnotou porovnává.

Pro správný chod je potřeba povolit globální přerušení a příslušné přerušení pro čítač-časovač. Požadované přerušení povolíme nastavením bitu TICIE1 – Input Capture Interrupt Enable v registru TIMSK – Timer/Counter Interrupt Mask Register. Globální masku přerušení povolíme příkazem sei() .

4.2.2 Pravidelný výpočet v podprogramu

Vytvořili jsme si žádost o přerušení v pravidelném intervalu 0,1s. V podprogramu pro obsloužení přerušení je umístěn výpočetní program sloužící

k aktualizaci stavu nádrže. Funkce vykonávaná v přerušení je vepsána ve funkci `irq_capture`. Ukazatel na tuto funkci vložíme pomocí funkce `timerAttach` do volání funkce v podprogramu pro obsluhu přerušení od č/č1 `InterruptCapture`.

V této funkci nejprve provedeme reset Watch Dog čítače. K tomu slouží příkaz `WDR` (pozor, příkaz je v jazyce Assembler). Čítač funkce Watch dog je nastaven na interval přibližně 0,52s. Čítač je nulován v intervalu 0,1s. Tímto zajistíme plynulý chod mikroprocesoru a nedodají tak k přetečení čítače a následnému resetu.

Příjem akčního zásahu

Akční zásah přichází analogovým signálem z řídicího Power Panelu na vstup A/D převodníku, kde je převeden na 8bit číslicovou hodnotu „prijem“. Tento akční zásah je společný pro napouštěcí i vypouštěcí ventil a je potřeba tyto veličiny od sebe oddělit. Nulová hodnota otevření obou ventilů (klidová poloha) se nachází v půlce rozsahu, tedy `0x7F` (analogově 2,5 V). Kladná velikost výchylky nad tuto hodnotu odpovídá otevření napouštěcího ventilu a záporná výchylka od této hodnoty odpovídá otevření vypouštěcího ventilu. V každém okamžiku může být otevřen pouze jeden z těchto ventilů, druhý je zavřený.

V případě akčního zásahu „Prijem“ většího než `0x7F`

$$Q_{a+} = \frac{\text{Prijem} - 0xFF}{40} \quad (26)$$

$$Q_{a-} = 0$$

Q_{a+} Velikost přítoku akčního zásahu [lit/s]

Q_{a-} Velikost odtoku akčního zásahu [lit/s]

Prijem 8-bit hodnota akčního zásahu z ADC

Maximální hodnota $Q_{a+\max} = 3,2$ lit/s

V případě akčního zásahu „Prijem“ menšího než `0x7F`

$$a_a = (0x7F - \text{Prijem}) * 0.005682 \quad (27)$$

a_a Průřez odtokového ventilu akč. zásahu [m²]

Prijem 8-bit hodnota akčního zásahu z ADC

Maximální hodnota $a_{\max} = 0,721 \text{ m}^2$

$$Q_{a-} = a_a * k\sqrt{2gh} \div 1000 \quad (28)$$

$$Q_{a+} = 0$$

Q_{a-} Velikost odtoku akčního zásahu [lit/s]

k Hydrodynamický součinitel, $k = 0,94$ (voda)

g Gravitační zrychlení [m/s^2], $g = 9,81 \text{ m/s}^2$

h Výška vodní hladiny [m]

Maximální hodnota $Q_{a-\max} = 3,0 \text{ lit/s}$

Hodnoty odtoku a přítoku poruchy

Popis získání a výpočtu těchto veličin je podrobně popsán v kapitole 4.1.4

Vstupy poruchy do nádrže.

Přepočet výšky hladiny

V této chvíli je znám přítok a odtok z nádrže a může se přepočítat nová, aktuální výška hladiny. Při výpočtu se vychází ze známých průtoků napouštění/vypouštění v jednotkách lit/s. Toto množství vynásobíme časovým interval propočtu, který činí 0,1s. Tímto získáme množství vody, o které se aktuální hladina změnila.

$$V_t = V_{t-1} + T_{vz}(Q_{p+} + Q_{a+} - Q_{p-} - Q_{a-}) \quad (29)$$

V_t Aktuální objem nádrže [lit]

V_{t-1} Objem nádrže v předchozím kroku [lit]

Nejprve se z předchozí hodnoty nádrže, přítoku a odtoku zjistí, jestli není nově vypočtená hodnota mimo povolený rozsah. To znamená, že by nádrž přetekla nebo naopak dosáhla záporných hodnot. Pokud tomu tak není, provede se přičtení přítoku a odečtení odtoku. Tímto získáme aktuální výšku hladiny, která až do příštího cyklu zůstává stejná.

Pokud se ovšem vyhodnotí, že výška hladiny plus přítok a minus odtok je více než 255litrů, musí se provést omezení. Do nádrže nelze napustit více než je její objem. Situace se vyřeší omezením přítoku na takovou hodnotu, kterou je nádrž schopna pojmout nebo dojde k přetečení nádrže a odtečení nadbytečné vody. V průmyslu je tato varianta nebezpečná a mohlo by dojít k poškození okolních objektů působením rozlité vody. Tímto je stav nádrži na maximum a výška hladiny se nastaví na 255l. Rovněž se rozsvítí LED dioda plná nádrž. Pokud se vyhodnotí předpoklad záporného objemu kapaliny v nádrži, je potřeba provést podobné kroky jako v případě přítoku. Hodnota hladiny v nádrži se nastaví na 0. Rovněž se rozsvítí LED dioda prázdná nádrž. Proveďte se omezení odtékajícího množství z nádrže. Odtékající množství bude odpovídat přítékajícímu. Toto odpovídá i logické představě, kdy odteče pouze takové množství vody, které je k dispozici.

Sloupcová indikace objemu

Po přepočtení výšky hladiny potřebujeme upravit vizuální prvky. Model obsahuje sloupcovou indikaci objemu - BAR graf podrobně popsáný v kapitole 4.1.2. Program pomocí algoritmu určí v jakém intervalu (dle **Tab. 3**) se hladina nachází a podle toho nastaví příslušné hodnoty segmentů BAR grafu.

Chybná regulace

Při regulaci výšky vodní hladiny v modelu může dojít ke dvěma krajním stavům. Prvním stavem je prázdná nádrž a snaha odpouštět více než přitéká a opačná situace, kdy je nádrž plná a máme snahu připouštět více, než odtéká. Podrobněji je situace popsána v kapitole 4.1.2. Pokud nastane jedna z těchto 2 situací, tato dioda se rozsvítí a upozorní na pravděpodobnou chybu v regulaci, kdy požadujeme po modelu fyzikálně nerealizovatelné akce.

Indikace ventilů

Pro přesnou informaci obsluhy modelu jsou implementovány informační LED diody. Vždy jedna dioda ke každému ovládanému ventilu.

$Q_{a+} > 0$	Přítok ON
$Q_{a-} > 0$	Odtok ON
$Q_{p+} > 0$	Přítok porucha ON
$Q_{p-} > 0$	Odtok porucha ON

SPI přenos

V kapitole 4.1.6 je popsán D/A převodník, který převádí číslicovou hodnotu hladiny a záporného odtoku poruchy na analogovou veličinu přiváděnou na výstup modelu. Pro komunikaci mezi mikročipem a D/A převodníkem je užívána SPI sběrnice, kde mikročip představuje Master a D/A převodník Slave. Konfigurace SPI linky je popsána v kapitole 4.2 v části věnující se inicializaci. V pravidelném přerušení je prováděn přenos dat mezi oběma zařízeními.

Přenos se zahájí zápisem dat do registru SPDR. Data jsou odesílána ve dvou 8 bitových rámcích. V prvním rámci je kromě nejvyšších 4 bitů číslicové hodnoty obsaženo nastavení D/A převodníku, především zvolený kanál.

Rámec 1							
$\overline{A/B}$	0	1	1	D11	D10	D9	D8
bit 15				bit 8			

Rámec 2							
D7	D6	D5	D4	D3	D2	D1	D0
bit 7				bit 0			

$\overline{A/B}$ Volba kanálu
0 - kanál A
1 - kanál B

D11- D0 12bit data

Obr. 17 Struktura dat na sběrnici SPI

Přenos dat pro oba kanály probíhá postupně za sebou, nejprve jsou odeslána data pro kanál A (hladina) a poté pro kanál B. Při přenosu dat je nejprve do registru SPDR nahrán rámec 1 pro kanál A. Tímto se automaticky spustí odeslání. Při dokon-

čení přenosu je nasetován bit SPIF v registru SPSR. V okamžiku dokončení přenosu zapíšeme do registru SPDR rámec 2 pro kanál A a zahájíme jeho odeslání. Opět testujeme bit SPIF. Pokud je i tento rámec odeslán jsou odeslána omeletní data pro kanál A a data potvrdíme sestupnou hranou signálu $\overline{SS}$. Nyní se postup obdobně opakuje ovšem s jinými daty pro kanál B.

5. OVLÁDÁNÍ MODELU POMOCÍ PLC

Samotný model nádrže není schopen práce, je potřeba jej připojit k řídicímu přístroji. V tomto případě je řídicím prvkem PLC Power Panel 400 od firmy B&R [11]. Power Panel obsahuje ovládací karty pomocí níž je model ovládán. Panel disponuje dotykovou obrazovkou VGA s postranními tlačítky.

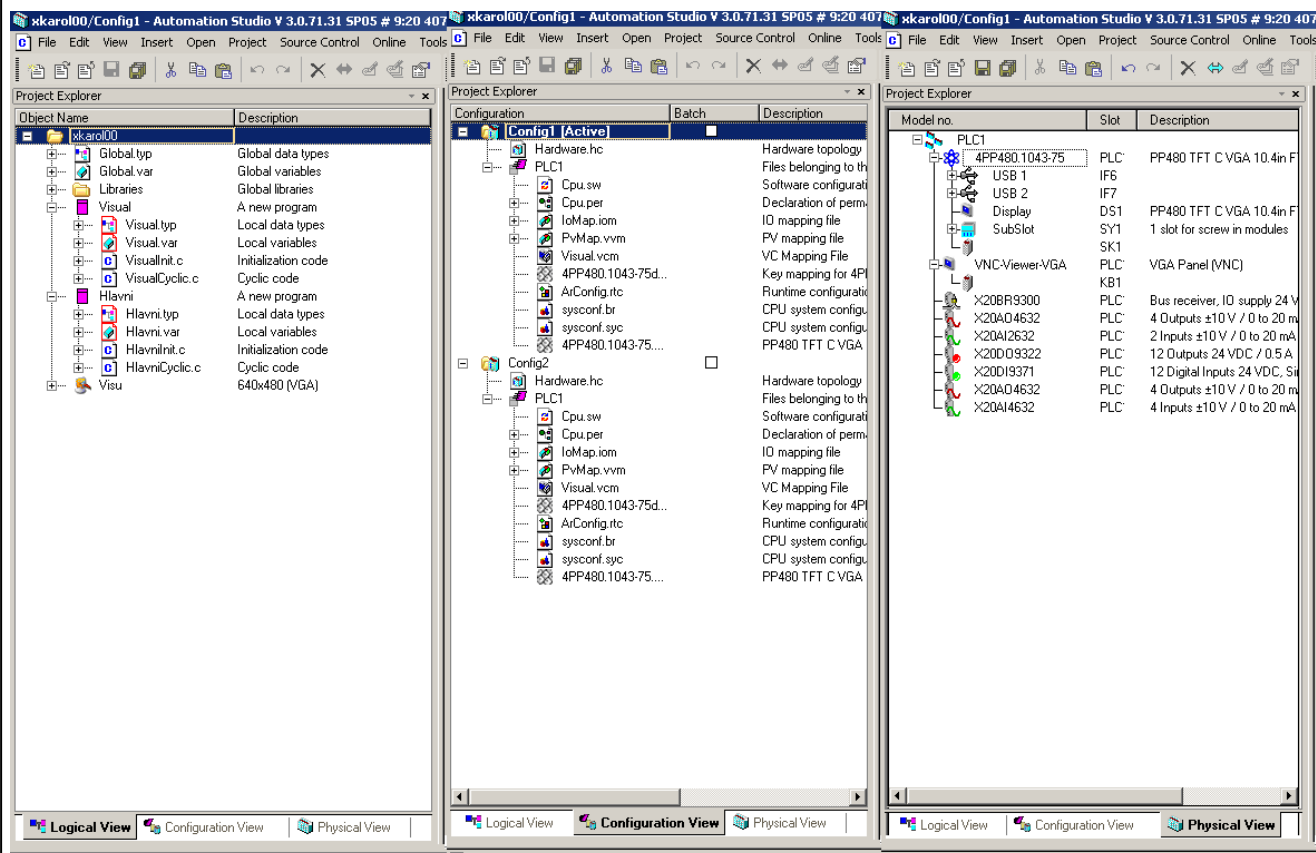
5.1 KONFIGURACE PROJEKTU

Program je nakonfigurován pro Power Panel DA11 s IP 147.229.76.19 a obsahuje druhou konfiguraci Config2 pro stanici DA09. Dále bude popsána práce na pouze na stanici DA11. Zde je použit Power Panel 480 označení 4PP480.1043-75 Rev B5. Panel je vybaven VGA dotykovou obrazovkou s postranními tlačítky. Obsahuje komunikační kartu X2X Link IF768.

Ke komunikační kartě X2X jsou připojeny moduly napájení, digitálních a analogových vstupů a výstupů. Moduly použité ve stanici DA11 seřazeny vzestupně podle pozice v konfiguraci:

Tab. 4 Konfigurace karet v PLC

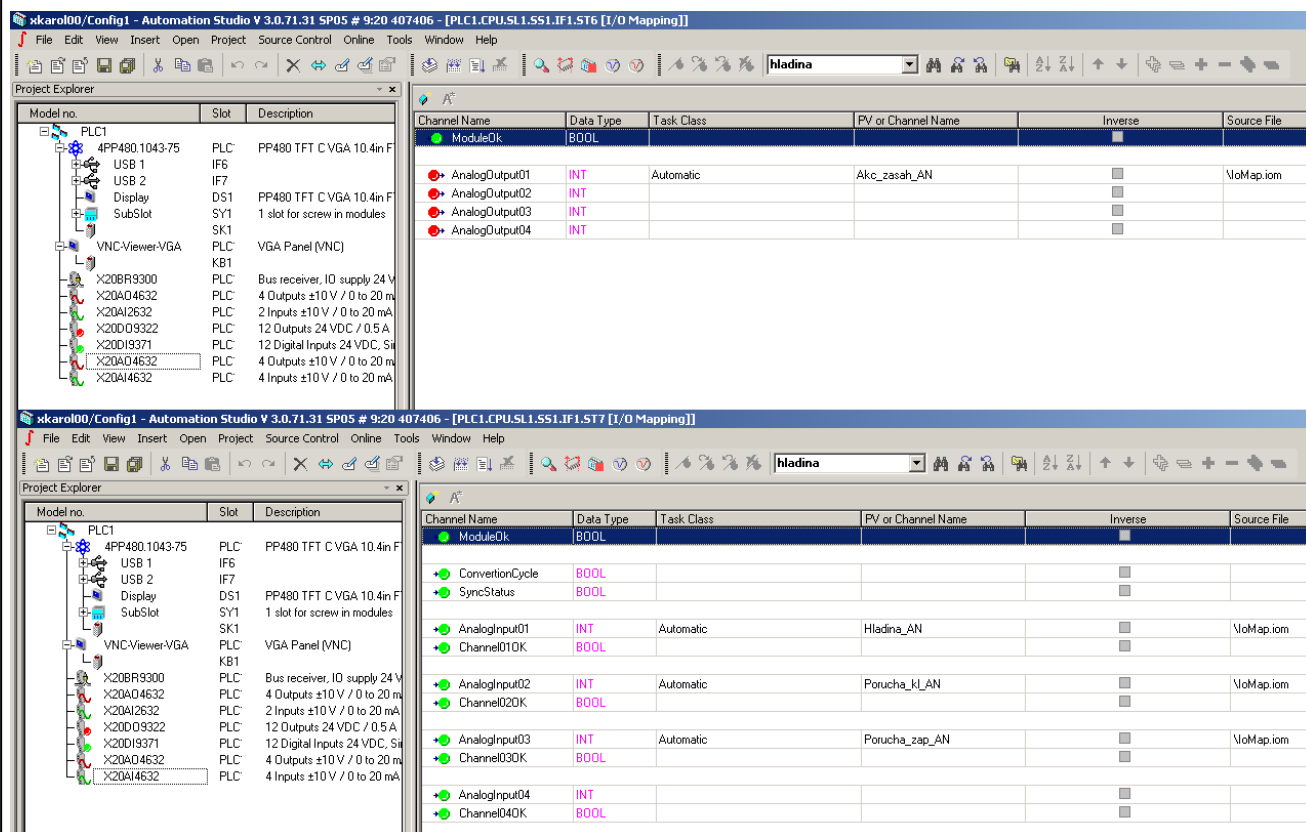
Slot	Označení karty	Popis
0	X20BR9300	Komunikační karta X2X
1	X20AO4632	4x analogový výstup
2	X20AI2632	2x analogový vstup
3	X20DO9322	12x digitální výstup
4	X20DI9371	12x digitální vstup
5	X20AO4632	4x analogový výstup
6	X20AI4632	4x analogový vstup



Obr. 18 Konfigurace projektu

5.2 PROGRAMOVÁ IMPLEMENTACE

V projektu model nádrže jsou používány pouze analogové vstupy/výstupy u posledních dvou karet. Ostatní karty jsou zapojeny pro ostatní projekty a aplikace a proto nebyly odpojeny a jsou obsaženy i v tomto projektu. Analogová karta X20AO4632 ve slotu 5 na výstupu AnalogOutput01 obsahuje data Akc_zasah_AN, což je akční zásah vstupující do modelu. Naopak analogová karta X20AI4632 ve slotu 6 na vstupu AnalogInput01 obsahuje data Hladina_AN. Na vstupu AnalogInput02 jsou data Porucha_kl_AN informující o velikosti přítoku poruchy. Na vstupu AnalogInput03 jsou data Porucha_zap_AN informující o velikosti odtoku poruchy. Detailně je nastavení vidět na následujícím obrázku:



Obr. 19 Konfigurace vstupů a výstupů karet

Důležitou činností při zpracování analogové veličiny je její normalizace na veličinu odpovídající technologickému procesu. Při normalizaci je důležité znát rozsah analogové veličiny a celočíselné hodnoty přiřazené k této veličině. Příklad normalizace analogové veličiny snímané na vstup analogové karty

$$Porucha_kl = ((float)Porucha_kl_AN) * 6.2 / 32768; \quad (30)$$

Porucha_kl_AN je číselná hodnota typu INT odpovídající analogové hodnotě na vstupu karty. Při velikosti napětí 10V je hodnota čísla 32768, při hodnotě 0V je číslo 0 a při -10V je hodnota -32768. Technologický proces poskytuje rozsah pouze 0 až 5V. Hodnotě 5V odpovídá veličina odtoku o velikosti 3,1 [lit/s]. Extrapolací této závislosti do úrovně 10V získáme parametr 6,1 použitý v rovnici. Obdobně postupujeme i pro ostatní veličiny. Výpis kódu včetně normalizace pro všechny analogové signály je uveden v příloze 4.

V projektu jsou vytvořeny 2 programy zpracováváné v procesoru. Projekty jsou psány v jazyku ANSI C, který nepatří mezi normované jazyky pro použití v PLC ale Automation Studio jej podporuje.

První program je Visual, který je podpůrný a je pouze pro grafickou část aplikace. Běží v cyklu 100ms a tím zajišťuje plynulejší a vyhlazenější zobrazení proměnných hodnot, než by tomu bylo u programu s dlouhou periodou.

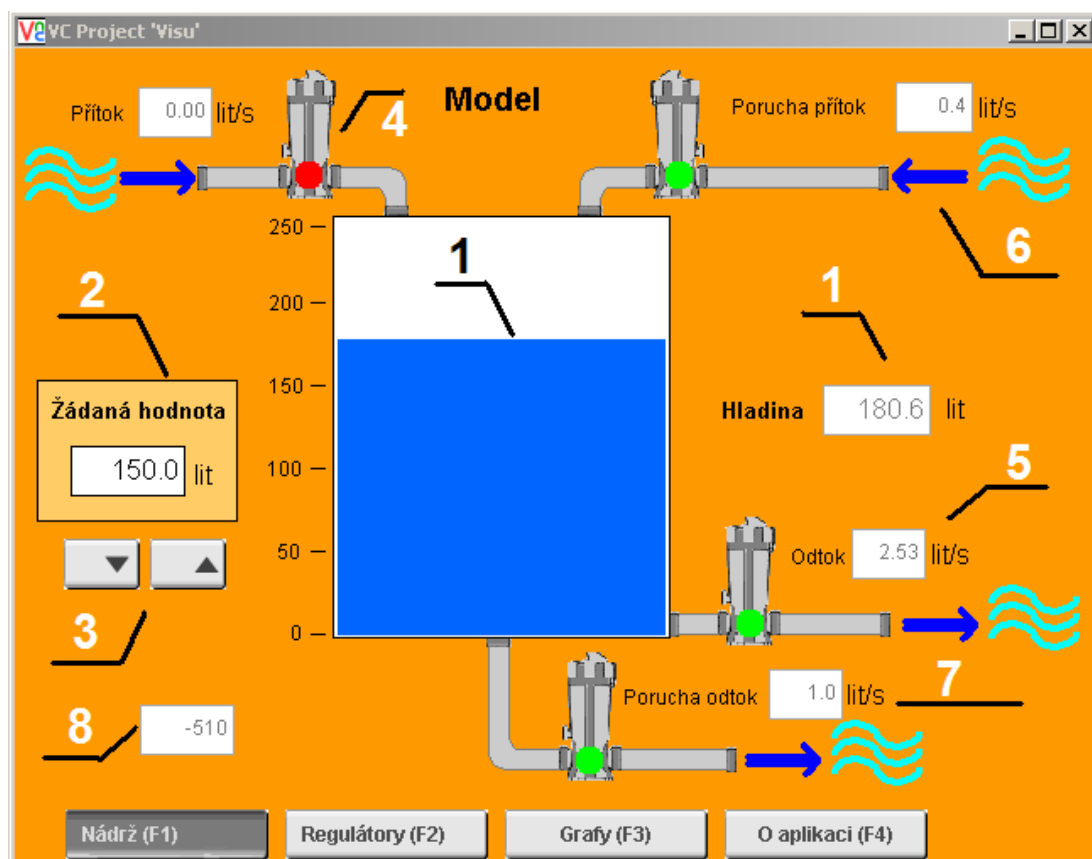
Další je Program Hlavní. V tomto programu jsou implementovány algoritmy regulátorů PSD s filtrací derivační složky, β -PSD a Feed Forward. Volba regulátoru se provádí ve vizualizaci, kde je zobrazeno jeho vnitřní schéma včetně možnosti úpravy parametrů regulátorů. Tento program je vykonáván v pravidelném cyklu 1s. tento interval je zvolen s ohledem na charakter akčních členů, kterými jsou ventily. Příliš rychlé a časté polohování by u reálného procesu znamenalo enormní zátěž a nutnost časté výměny těchto ventilů. Na konci kódu je akční zásah normalizován a zapsán to proměnné provázané s analogovou kartou.

Object Name	Version	Transfer ...	Size (byt...	Source	Source File	Description
CPU						
Cyclic #1 - [10 ms]						
Cyclic #2 - [20 ms]						
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
Visual	1.00.0	UserROM	904	Visual	\Cpu.sw	A new program
Cyclic #5 - [200 ms]						
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
Hlavní	1.00.0	UserROM	7724	Hlavní	\Cpu.sw	A new program
Cyclic #8 - [10 ms]						
Data Objects						
No Data Objects						
Visualisation						
Visu	1.00.0	UserROM	1488	Visu	\Cpu.sw	640x480 (VGA)
Binary Objects						
Library Objects						
fileio	2.95.0	UserROM	28000			
dataobj	2.95.0	UserROM	14264			
runtime	2.95.0	UserROM	31452			
visapi	3.35.0	UserROM	27164			
Configuration Objects						
asfw	1.00.0	SystemR...	321112		\Cpu.sw	
arconfig	1.00.0	SystemR...	8296		\Cpu.sw	
ashwd	1.00.0	SystemR...	10088		\Cpu.sw	
sysconf	2.95.0	SystemR...	61012		\Cpu.sw	
iomap	1.00.0	UserROM	13444		\Cpu.sw	

Obr. 20 Přiřazení programů a vizualizace v procesoru

5.3 VIZUALIZACE

Vizualizace slouží k informačním, ale i řídicím účelům. Jsou zde zobrazeny veškeré informace, které jsou zobrazovány na modelu nádrže a navíc informace týkající se řízení. Vizualizace slouží také k ovládání aplikace, zde je volena žádaná výška vodní hladiny, volba regulátoru a nastavení jeho vnitřních parametrů. Vizualizace se skládá z několika stránek, které obsahují určitou část parametrů systému.

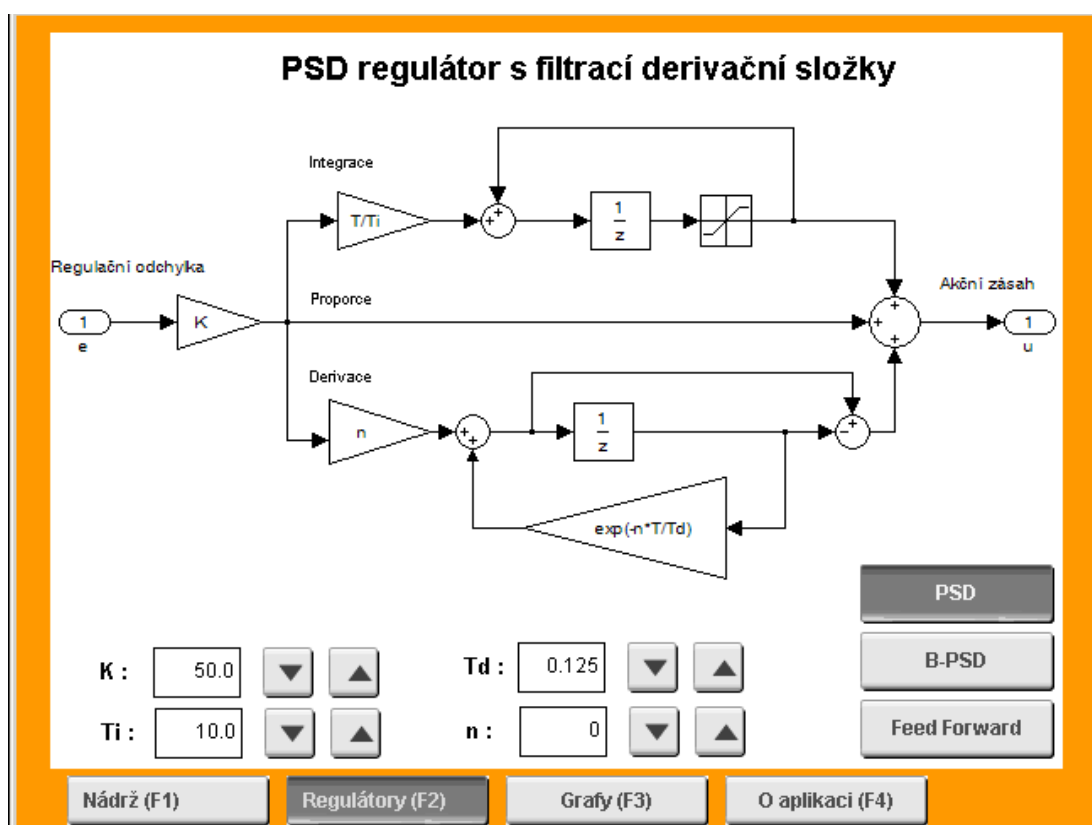


Obr. 21 Vizualizace nádrž

Legenda:

- 1 - Objem vody v nádrži v litrech
- 2 - Žádaná hodnota objemu v nádrži, po stisku se vyvolá číselné zadávání
- 3 - Inkrementální změna žádaného objemu v nádrži, krok činí 5 litrů
- 4 - Přítokové potrubí ovládané akčním zásahem z regulátoru, zobrazena hodnota průtoku v potrubí v litrech za sekundu. Pokud je průtok nenulový, svítí ventil zeleně, v opačném případě červeně

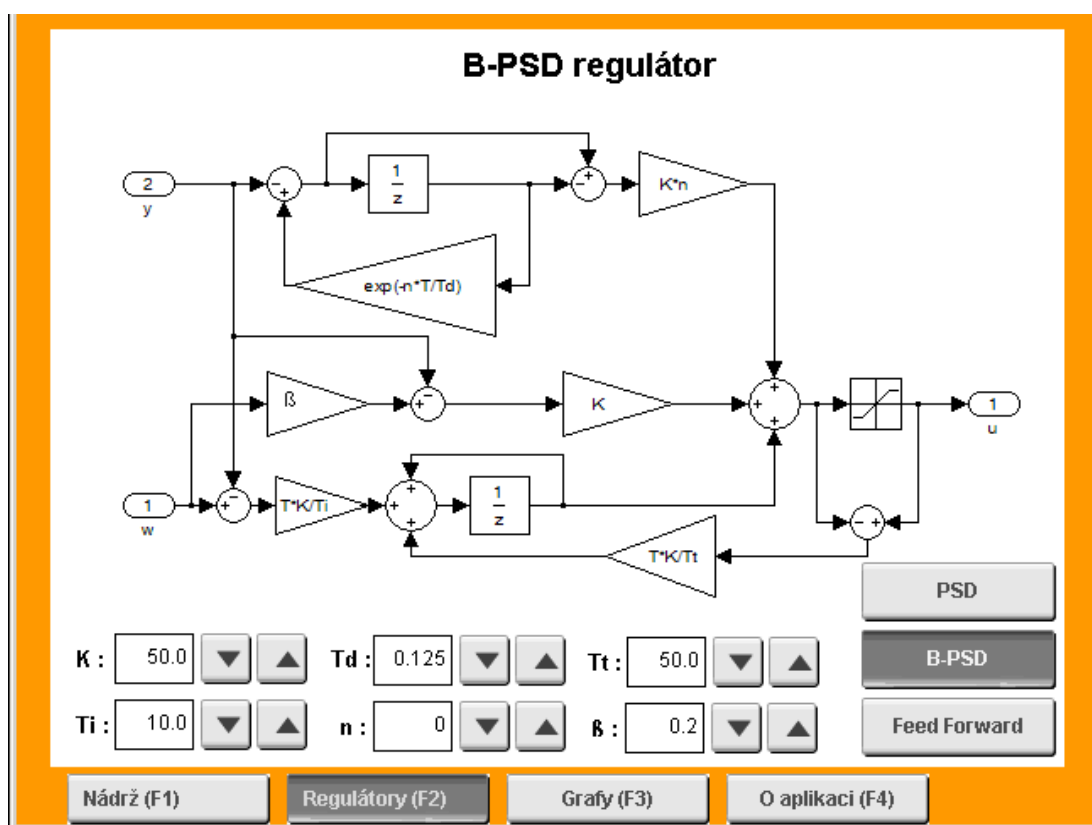
- 5 - Odtokové potrubí ovládané akčním zásahem z regulátoru, zobrazena hodnota průtoku v potrubí v litrech za sekundu. Pokud je průřez ventilu nenulový, svítí ventil zeleně, v opačném případě červeně
- 6 - Přítokové potrubí poruchy ovládané ručně na modelu, zobrazena hodnota průtoku v potrubí v litrech za sekundu. Pokud je průtok nenulový, svítí ventil zeleně, v opačném případě červeně
- 7 - Odtokové potrubí poruchy ovládané ručně na modelu, zobrazena hodnota průtoku v potrubí v litrech za sekundu. Pokud je průřez ventilu nenulový, svítí ventil zeleně, v opačném případě červeně
- 8 - Velikost akčního zásahu na výstupu regulátoru



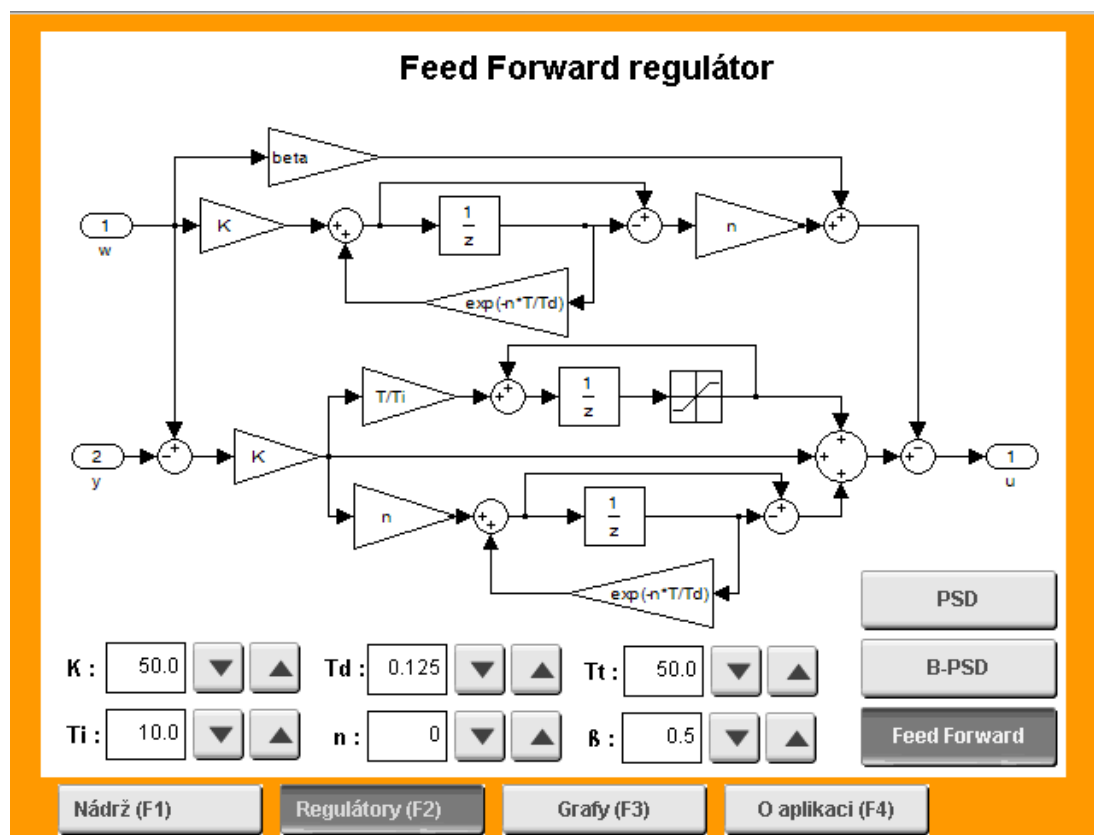
Obr. 22 Vizualizace PSD regulátor

Volba regulátoru se provádí tlačítky v pravé části. Při této volbě je aktivní PSD regulátor. Na obrázku je zobrazeno vnitřní schéma. V dolní části je možná změna parametrů regulátoru. Je možno použít přiřazená tlačítka pro inkrementální změnu

hodnotu nebo stiskem daného pole je vyvolána numerická klávesnice a lze pomocí ní hodnotu zadat. Situace je obdobná i pro další regulátory zobrazené na **Obr 23** a **Obr 24**. Po přepnutí regulátoru se aktivuje zvolený regulátor, je zobrazena jeho struktura a lze měnit parametry. Při změně regulátoru není realizováno beznárazové přepínání a dochází ke skoku akčního zásahu.



Obr. 23 Vizualizace β -PSD regulátor



Obr. 24 Vizualizace Feed Forward regulátor

6. DOSAŽENÉ VÝSLEDKY

Výsledkem bakalářské práce je navržení, sestavení naprogramování a otestování modelu pro regulaci výšky vodní hladiny.

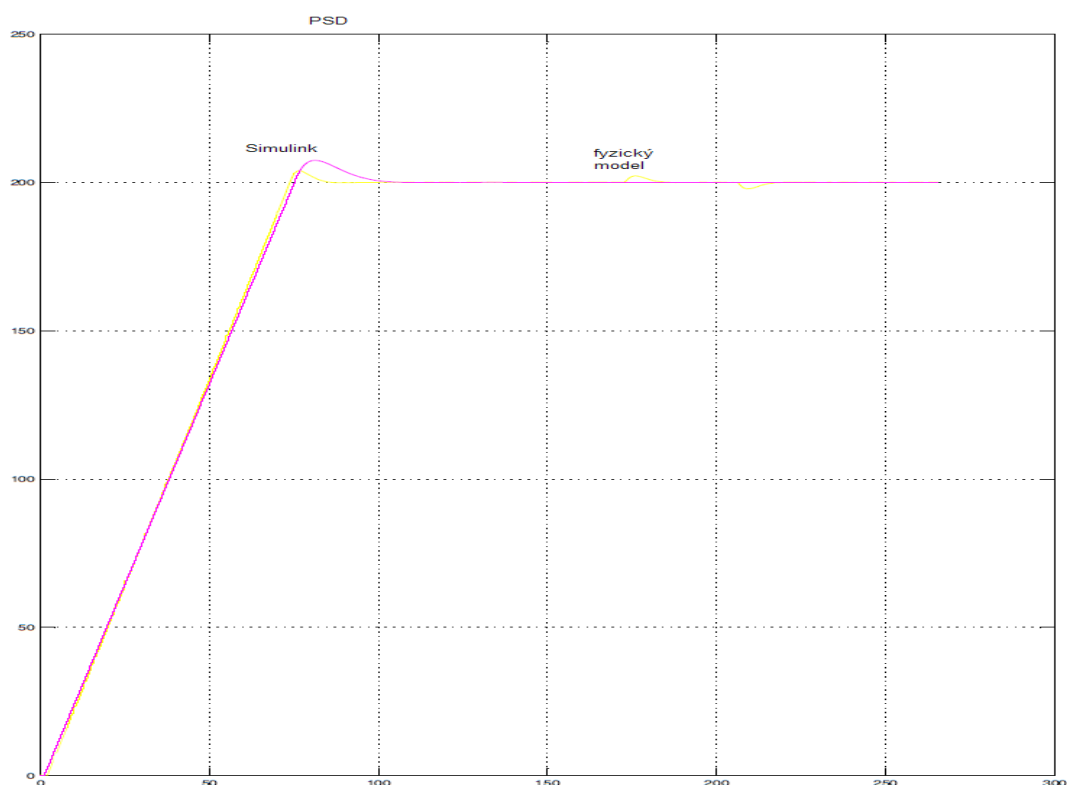
Model je umístěn do krabičky WEB B8 umístění na DIN liště. Obsahuje mikročip ATmega16 doplněný o D/A převodník MCP4922. Pro změnu napětí z 24V na 5V je vybaven spínaným zdrojem v SMD provedení s vysokou účinností. Obsahuje LED indikaci a LCD displej 2x 16 znaků pro zobrazení stavu nádrže. Je vybaven prvky pro simulaci působící poruchy, a to jak kladné tak i záporné. Pro spojení s řídicím systémem je vybaven konektorem MLW20 dle dohodnutého standardu CLG laboratoře. Data jsou přenášena analogově. Model přijímá analogová data nastavení poruchy na vstup A/D převodníku v ATmega16. Dalším vstupem do tohoto převodníku je analogová hodnota akčního zásahu generovaná řídicím systémem. Použité analogové karty v PLC poskytují rozsah napětí -10V až 10V což značně převyšuje parametry A/D převodníku. Z tohoto důvodu je před vstupem do A/D převodníku zapojena ochrana omezující toto napětí na hodnotu bezpečnou pro A/D převodník. Z modelu vystupují analogová data pro PLC s informací o objemu nádrže, velikosti kladného a záporného přítoku poruchy. Data přepočítávána v mikročipu jsou následně SPI sběrnici přenesena do D/A převodníku. Odtud jsou v analogové podobě přenášeny na výstup modelu.

Pro účel řízení modelu byl vytvořen regulační systém. K tomuto účelu byl použit programovatelný automat Power panel 400 od firmy B&R. V aplikaci jsou implementovány základní regulátory PSD, β -PSD a Feed Forward. Obsahuje také vizualizační část, ve které je řízený proces nejen zobrazen, ale lze měnit požadované hodnoty včetně nastavení jednotlivých regulátorů.



Obr. 25 Model pro řízení výšky vodní hladiny a Power Panel

Pro zpracování dat z reálného modelu a jejich porovnání se simulačním modelem bylo použito modelu vytvořeného v Simulinku. Pomocí komponent v Toolboxu VUT tools byla získána data ze skutečného modelu a mohly být vzájemně porovnány.



Obr. 26 Porovnání dat pro PSD regulátor

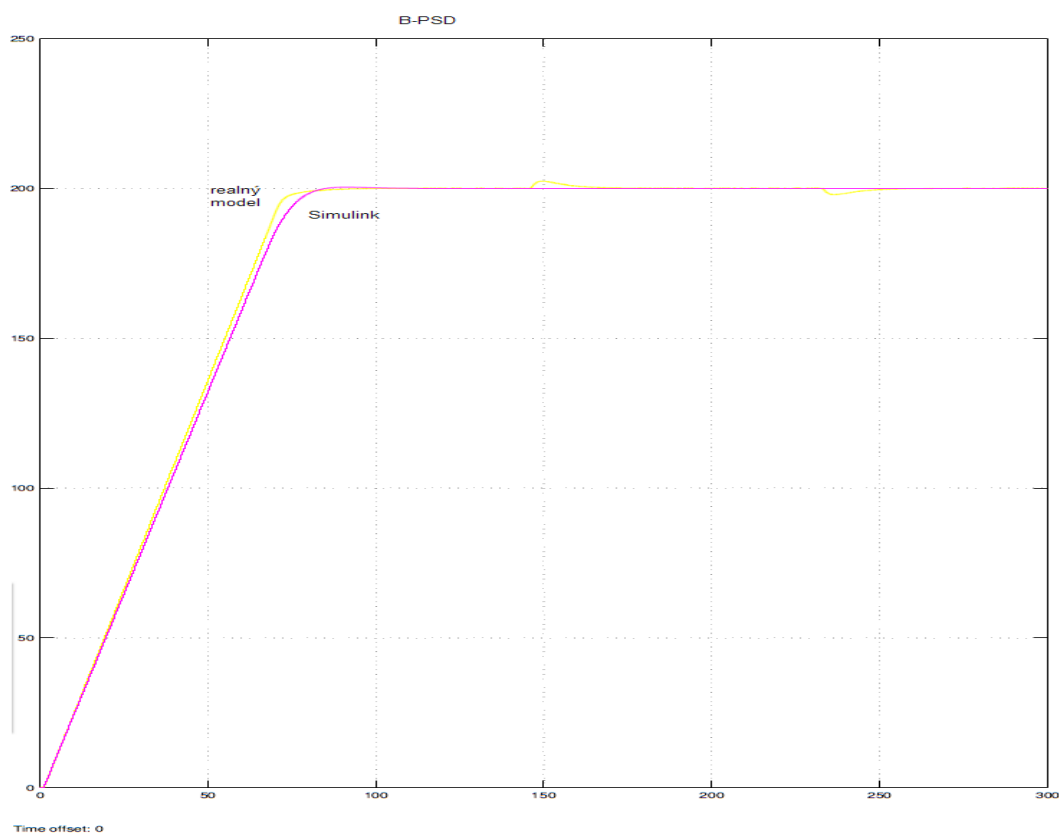
Na reálný model skokově zapůsobil poruchový přítok 1,4 lit/s v čase 180. V čase 210 zapůsobil na soustavu poruchový odtok 1,4 lit/s v okamžik počátku působení poruchy. Regulátor poruchu rychle a přesně vykompenzoval.

Parametry PSD regulátoru v PLC:

$$K = 50; T_i = 5; n = 0; T_d = 0,125; T = 1$$

Parametry PSD regulátoru v Simulinku:

$$K = 50; T_i = 20; n = 3; T_d = 0,125; T = 1$$



Obr. 27 Porovnání dat pro β -PSD regulátor

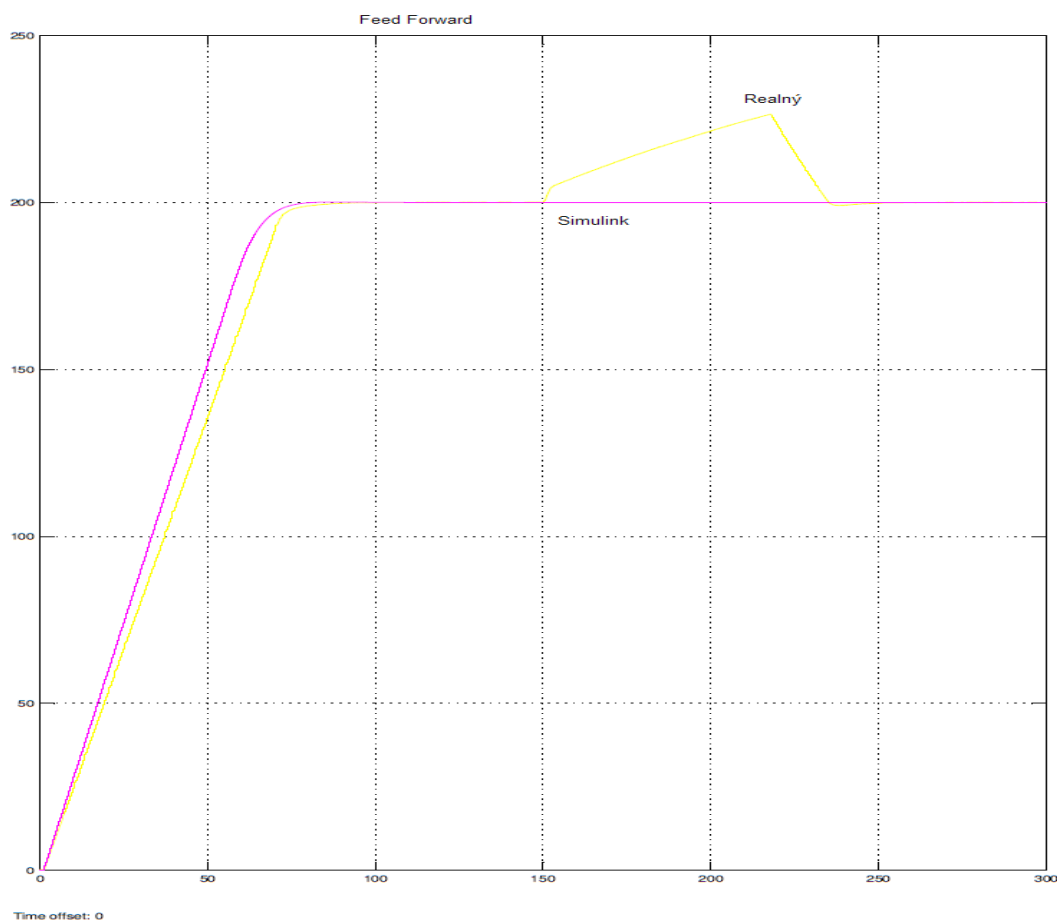
Na reálný model skokově zapůsobil poruchový přítok 1,5 lit/s v čase 145. V čase 235 zapůsobil na soustavu poruchový odtok 1,3 lit/s v okamžik počátku působení poruchy. Regulátor poruchu rychle a přesně vykompenzoval.

Parametry β -PSD regulátoru v PLC:

$K = 50$; $T_i = 10$; $n = 0$; $T_d = 0,125$; $T = 1$; $\beta = 0,2$

Parametry β -PSD regulátoru v Simulinku:

$K = 50$; $T_i = 20$; $n = 3$; $T_d = 0,125$; $T = 1$; $\beta = 0,2$



Obr. 28 Porovnání systému pro Feed Forward regulátor

Parametry Feed Forward regulátoru v PLC:

$$K = 50; T_i = 10; n = 0; T_d = 0,125; T = 1; \text{betaFF} = 0,5$$

Parametry Feed Forward regulátoru v Simulinku:

$$K = 50; T_i = 100; n = 3; T_d = 0,125; T = 1; \text{betaFF} = 0,5$$

Data fyzického modelu jsou zobrazena žlutě. V čase 150s je na vstup soustavy přivedena skoková změna poruchy způsobující přítok do nádrže 3,1 lit/s. Akční zásah není schopen plně vykompenzovat přítok a dochází k nežádoucímu plnění nádrže. V čase 220s je na vstup soustavy zaveden jednotkový skok poruchového odtoku 1,9 lit/s v okamžik počátku působení. Tento odtok částečně kompenzuje poruchový přítok natolik, že je již schopen akční zásah plně vykompenzovat poruchu.

Technologický proces v Simulinku a reálný model tvořený modelem nádrže a řídicím PLC, se až na výjimky shodují. Nejlepší shoda je v počátečním strmém náběhu charakteristiky, ovšem i zde je pozorovatelná nepřesnost. Odchyly lze přisoudit rušení indukovanému na analogové kanály použité při komunikaci.

7. LABORATORNÍ ÚLOHA

Model nádrže je navržen pro použití v laboratoři ve spojení s PLC Power Panel. V této kapitole je vytvořen manuál pro obsluhu modelu, ve kterém je stručně popsán princip modelu, jeho realizace a pokyny k obsluze. V další části je popsána jednoduchá laboratorní úloha, ve které je model použit.

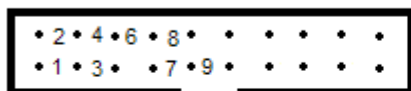
7.1 MANUÁL MODELU NÁDRŽE

Model nádrže slouží k simulaci reálného technologického procesu regulace výšky vodní hladiny v nádrži.

Nádrž je tvaru válce o ploše dna $A = 0,255\text{m}^2$ a výšce nádrže $h = 1\text{m}$. Celkový objem nádrže činí 255 litrů. Nádrž obsahuje přítokové potrubí, kterým do nádrže přitéká množství vody označené Q_{a+} . Toto množství je regulováno ventilem ovládaným akčním zásahem z regulátoru. Maximální průtok je 3,1 lit/s. Druhým přítokovým potrubím do nádrže přitéká množství vody označené jako kladná porucha vstupující do systému označené Q_{p+} . Toto množství je regulováno ventilem ovládaným ručně z modelu. Maximální průtok je 3,1 lit/s. Model obsahuje 2 odtoková potrubí. Prvním je ovládáno akčním zásahem, jehož průtokové množství Q_{a-} je funkcí průřezu ovládaného ventilu a výšky vodní hladiny. Druhé potrubí představuje záporný vstup poruchy do systému. Průřez ventilu je ovládán ručně z modelu. Protékající množství je funkcí průřezu ovládaného ventilu a výšky vodní hladiny. Schéma nádrže je zobrazeno na **Obr 3**. Matematické rovnice vztahů jednotlivých veličin a pro přepočet modelu jsou uvedeny v rovnicích (1) až (7).

Model pracuje s mikrokontrolérem ATmega16 s krystalem 16MHz. Perioda vzorkování a výpočtu vnitřních stavů probíhá periodicky každých 100ms. Na obrázku **Obr. 29** je zobrazeno zapojení komunikačního MLW konektoru. Napájecí napětí 24V je přivedeno do spínaného zdroje a dále je používáno napětí 5V. Této hodnoty také dosahují 3 analogové výstupy vystupující z modelu. Analogový vstup do modelu obsahuje ochranu proti překročení napětí, ale pracovní rozsah vstupního A/D převodníku je pouze 0 až 5V.

MLW20



1,2 - napájení 24V

3,4 - GND

6 - Analogový vstup akční zásah

7 - Analogový výstup Hladina

8 - Analogový výstup Kladná_porucha

9 - Analogový výstup Záporná_porucha

Ostatní nezapojeno

Obr. 29 Zapojení konektoru MLW20

Uživatel má k dispozici ovládací prvky nastavující průřez potrubí působící poruchy. Levým otočným trimrem (zelený) je nastavován přítok poruchy Q_{p+} . Poloha úplně vlevo znamená přítok 0. Poloha úplně vpravo způsobí přítok poruchy 3,1 lit/s. V případě nenulového nastavení je na displeji zobrazen znak „P+“ a údaj o velikosti poruchy. Zároveň svítí LED dioda „Přítok porucha“. Pravým otočným trimrem (červený) je nastavován odtok poruchy Q_{p-} . Poloha úplně vlevo znamená průřez ventilu 0. Poloha úplně vpravo nastaví průřez ventilu na maximum, společně s výškou vodní hladiny tomu odpovídá odtok poruchy Q_{p-} . V případě nenulového nastavení je na displeji zobrazen znak „P-“ a údaj o velikosti poruchy. Zároveň svítí LED dioda „Odtok porucha“.

LED BAR graf zobrazuje poměrnou velikost výšky hladiny vzhledem k maximální. Rozlišení je 10% maximální hodnoty a jednotlivé sametky jsou aktivovány ve střední hodnotě svého rozsahu. LED dioda Prázdná svítí, pokud je nádrž zcela prázdná a dioda Plná svítí, pokud je nádrž zcela plná. LED Porucha svítí, pokud je nádrž plná a stále dochází k napouštění nebo je nádrž zcela prázdná a stále jsou otevřené ventily odtoku. Tato funkce zahrnuje ventily akčního zásahu i poruchy. LCD displej na prvním řádku zobrazuje informaci o objemu vody v litrech s platností na 2 desetinné místa. Pokud působí na soustavu porucha je na druhém řádku zobrazen již zmíněný text s velikostí působící poruchy s platností na 1 desetinné místo. Pokud porucha nepůsobí je přes celou šíři druhého řádku zobrazen BAR graf obdobný jako v případě LED BAR grafu.

7.2 ZADÁNÍ LABORATORNÍ ÚLOHY

1. Dle schématu konektoru MLW20 připojte model k řídicímu automatu Power Panel.
2. Obě poruchy nastavte na nulu. Na spodním řádku displeje se musí objevit BAR graf. Tlačítkem „Reset“ proved'te reset modelu na nulové počáteční podmínky
3. Ve vizualizaci přejděte do záložky „Regulátory“. Zvolte PSD regulátor
4. Ponechejte pouze P složku regulátoru, vyřaďte sumační a derivační složku
5. Přejděte do záložky „Nádrž“ a stiskem pole „Žádaná hodnota“ aktivujte numerickou klávesnici. Zadejte hodnotu 200. V záložce grafy sledujte průběh akčního zásahu a žádané hodnoty na čase.
6. Postupně nastavujte na modelu hodnoty poruchy a pozorujte vliv na žádanou hodnotu.
7. Změňte žádanou hodnotu na 50 a opět pozorujte průběhy výstupů.
8. Přejděte do položky „Regulátory“ a přiřaďte regulátoru integrační část.
9. Opakujte měření podle bodů 5 až 7.
10. Přejděte do položky „Regulátory“ a přiřaďte regulátoru derivační část.
11. Opakujte měření podle bodů 5 až 7.
12. Přejděte do položky „Regulátory“ a zvolte regulátor β -PSD.
13. Opakujte měření podle bodů 5 až 7. V položce „Regulátory“ měňte parametry β -PSD regulátoru a sledujte chování systému.
14. Přejděte do položky „Regulátory“ a zvolte Feed Forward regulátor.
15. Opakujte měření podle bodů 5 až 7. V položce „Regulátory“ měňte parametry Feed Forward regulátoru a sledujte chování systému.
16. Vzájemně porovnejte výsledky jednotlivých regulátorů s ohledem na velikost prvního překmitu, dobu ustálení přechodového děje a vyregulování poruchy.

8. POUŽITÝ SOFTWARE

Při tvorbě bakalářské práce bylo použito několik programů, pomocí kterých byl vytvořen simulační model, napsán a naprogramován kód do mikrokontroléru, vytvořeno schéma včetně DPS a vytvořen program pro ovládání a vizualizaci v Power Panelu.

8.1 MATLAB SIMULINK

Při vývoji simulačního modelu byla použita verze MATLAB R2009b 7.9.0.529. Pro zpracování a porovnání výsledku simulace a dat z PLC v grafu byla použita verze R2008b s přídatnými komponentami v Simulinku „VUT tools“ od laboratoře CLG.

8.2 AVR STUDIO

Použitá verze AVR Studio 4.15 Build 623. Toto vývojové prostředí slouží přímo k tvorbě aplikací pracujících v mikročipech řady AVR. Je jedno z nejvíce používaných vývojových prostředí pro tuto činnost. Jednou z předností tohoto prostředí je možnost přímé volby mikročipu, pro který aplikaci tvoříme, a tím nám v mnohém zjednoduší práci při tvorbě programu. Samozřejmostí je nepřeberné množství knihoven, které velmi usnadňují psaní programu. V tomto projektu jsou použity hlavně knihovna vstupu a výstupu *io.h*, *interrupt.h*, *math.h*, *timer.h*, *string.h*, *lcd.h*, *lcd-conf.h*, *global.h*. Použité knihovny dostupné mimo WinAVR jsou použity Procyon AVRlib [12]. Tato vyšší verze programu nově umožňuje psát program nejen v assembleru, ale rovněž v jazyku C. Implementovaný GCC překladač zkompiluje zdrojový text přímo do souboru s příponou *.hex*.

AVR Studio je produkt přímo od firmy Atmel. Po registraci je volně stažitelný a připraven k instalaci. Nalézt jej můžete např. na adrese: http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725. Na této stránce jsou neustále aktualizovány nové verze programu a jejich případné doplňky.

8.3 AVR DUDE & AVR8 BURN-O-MAT

Po kompilaci projektu potřebujeme strojový kód přehrát do programátoru obsahujícího náš mikrokontrolér. Jelikož AVR Studio 4 nepodporuje použitý programátor, použijeme volně šiřitelný software AVRDUDE, který použitý typ programátoru podporuje. Tento program je ke stažení například zde:

<http://savannah.nongnu.org/projects/avrdude/>

Program nevyžaduje instalaci, avšak jedná se o čistě konzolovou aplikaci. Pro příjemnější grafické rozhraní toho programu existuje nástavba pod názvem AVR8 Burn-O-Mat 1.4.2b. Program je ke stažení například na stránkách autora:

http://www.brischalle.de/avr8_burn-o-mat_avrdude_gui/avr8_burn_o_mat_avrdude_gui_en.html

8.4 EAGLE

Pro tvorbu schématu a DPS byl použit program Eagle 5.6.0. Hlavní deska (dolní) je navržena jako dvojvrstvá a rozlitou zemí. Signalizační deska (horní) obsahující LED a trimry poruchy je jednovrstvá s rozlitou zemí. Schémata i desky vytvořené v tomto programu byly následně vyexportovány do formátu pdf.

8.5 AUTOMATION STUDIO

Pro tvorbu kompletní části ovládání i vizualizace na Power Panelu bylo použito Automation Studio 3.0.71.31. Hardwarová konfigurace, připojené karty, programy a nastavení jsou uvedeny v kapitole 5.

9. ZÁVĚR

Hlavním cílem bakalářské práce bylo seznámit se se základními vlastnostmi v problematice řízení výšky vodní hladiny. Jsou popsány základní technologické rysy od účelu vodní nádrže, přes snímače vodní hladiny až k používaným regulátorům. V práci je řešena vodní nádrž o maximálním objemu 255 litrů. Obsahuje 2dvě přítoková potrubí a 2 odtoková potrubí. Jeden pár potrubí slouží pro realizaci akčního zásahu a druhý pár slouží pro simulaci působící poruchy na vstupu soustavy. V teoretickém rozboru je popsána nádrž včetně grafického schématu na **Obr. 3** a obsahuje matematický popis modelu. Teoretický rozbor problému je ověřen pomocí modelu v programu Simulink. Získané výsledky odpovídají teoretickému předpokladu.

Takto navržený model technologického procesu byl implementován do mikrokontroléru ATmega16. Pomocí časovače 1 je generováno pravidelné přerušení, ve kterém je prováděn přepočet nádrže. Systém je vzorkován s periodou 100ms. Na modelu jsou implementovány 2 potenciometry pro nastavení kladné a záporné poruchy působící na soustavu. Analogové vstupy jsou přiváděny na zabudovaný A/D převodník v mikrokontroléru. Analogový signál akčního zásahu odesílaný z PLC obsahuje navíc ochranu proti překročení pracovního napětí A/D převodníku, by mohlo způsobit jeho zničení. Model je vybaven LCD displejem zobrazujícím aktuální objem v litrech a v závislosti zda na soustavu působí porucha, zobrazuje velikost kladné a záporné poruchy v litrech za sekundu. Pokud je působící porucha nulová je zobrazen BAR graf. Model je vybaven skupinou LED indikací. Jsou indikovány činnosti jednotlivých ventilů, krajní stavy nádrže a porucha regulace. Je obsažen BAR graf informující o výšce vodní hladiny. Mikročip komunikuje pomocí sběrnice SPI s D/A převodníkem MCP4922. Model poskytuje 3 analogové výstupy informující o výšce vodní hladiny, velikosti kladné a záporné poruchy. Napájení modelu poskytuje spínaný zdroj LM2674 doplněný o nadproudovou ochranu tavnou pojistkou a ochranou proti přepólování napájecího napětí.

Pro řízení sestaveného modelu je vytvořena aplikace pro Power Panel řady 400. Jsou implementovány regulátory PSD, β -PSD a Feed Forward. Je vytvořena vizualizace zobrazující stav technologického procesu včetně grafů a obsahuje prvky pro změnu vnitřních parametrů regulátorů. V porovnání výsledků získaných z modelu v simulinku a z modelu pomocí mikročipu řízeného programovatelným automatem výsledky odpovídají, ale shoda není stoprocentní. Odchylka je způsobena především chybou A/D převodníků a indukovaným rušením. Na závěr je vytvořen stručný manuál k modelu a vytvořeno zadání pro laboratorní úlohu.

Během práce na modelu nádrže jsem získal spoustu užitečných zkušeností, týkajících se především použití mikrokontroléru v praxi, jeho programováním a komunikací s okolím. Naučil jsem se pracovat s celou řadou vývojových programů, jejich znalost se určitě projeví v budoucí činnosti.

10. POUŽITÁ LITERATURA

- [1] KOVÁŘ, J., Regulační technika I, [online], poslední úpravy 26.6.2006, [cit. 2010-05-25], dostupné z:
<http://www.spszl.cz/modules/wfdownloads/visit.php?cid=8&lid=51>
- [2] Měření výšky vodní hladiny, [online], [cit. 2010-05-25], dostupné z:
<http://www.kvetakov.net/down.php?file=./UEIKMT/2/ls/ISAME/MTVKap7.pdf>
- [3] ON/OFF a PID regulace, [online], [cit. 2010-05-25], dostupné z:
http://logitron.cz/podpora/ON_OFF_PID_regulace.pdf
- [4] PIVOŇKA, P., Výpočetní technika v automatizaci, Skriptum VUT Brno, 2003
- [5] PIVOŇKA, P., Comparative Analysis in Implementation Discrete PID Controllers, VUT Brno
- [6] PIVOŇKA, P., Číslicová řídicí technika, Skriptum VUT Brno, 2003
- [7] Datasheet LM2674, National Semiconductor, [online], February 2005, [cit. 2010-05-25], dostupné z:
http://www.gme.cz/_dokumentace/dokumenty/935/935-022/dsh.935-022.1.pdf
- [8] Datasheet MCP4922, MICROCHIP, [online], 2004, [cit. 2010-05-25], dostupné z: http://www.gme.cz/_dokumentace/dokumenty/321/321-017/dsh.321-017.1.pdf
- [9] MATOUŠEK, D., Práce s mikrokontroléry ATMEL. 4. díl, AVR ATmega16, Praha, BEN, ISBN 80-7300-174-8, 2006

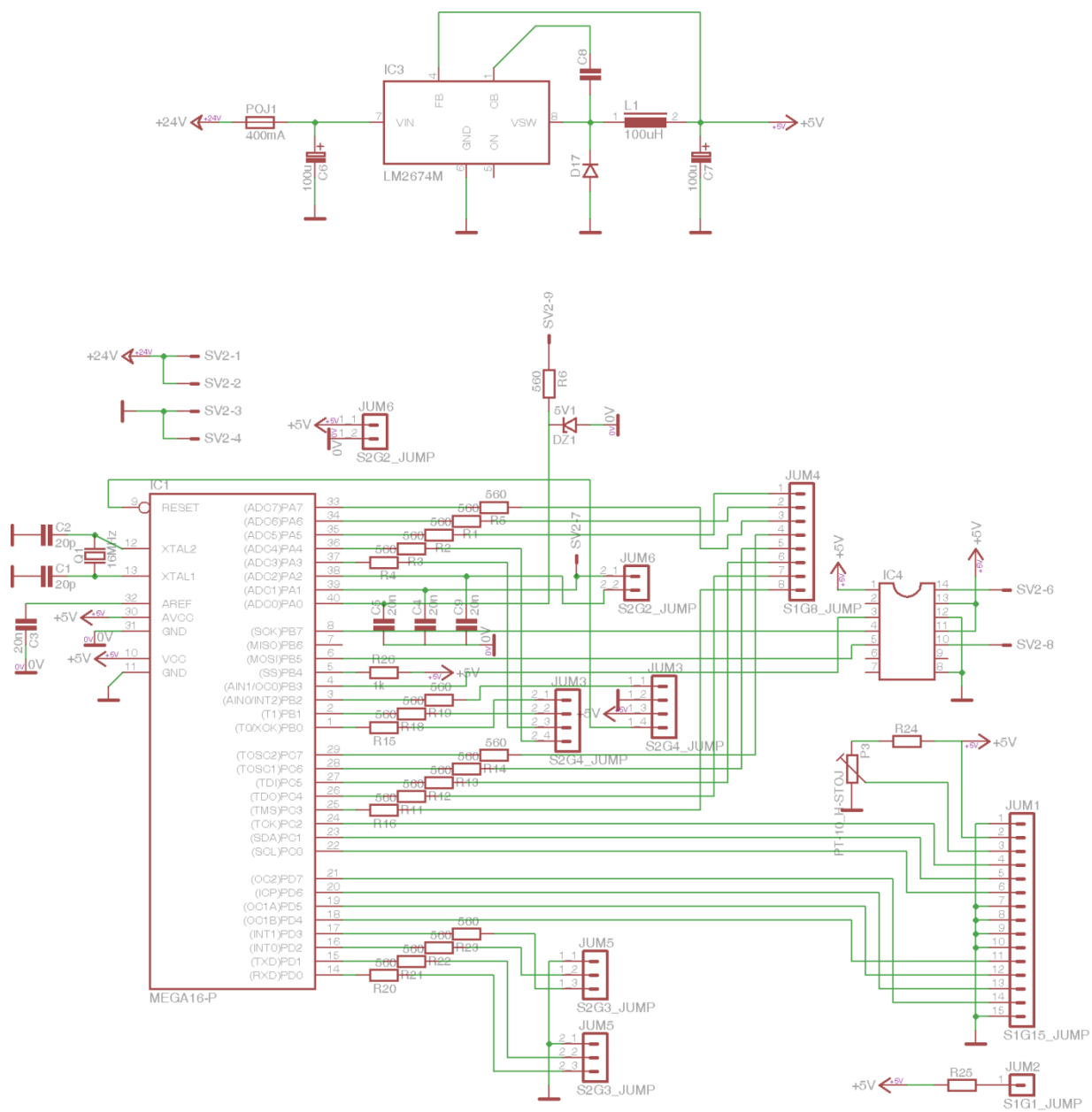
- [10] Datasheet ATmega16, ATMEL CORPORATION, [online], Rev. 2466E-AVR-10/02, [cit. 2010-05-25], dostupné z:
http://www.datasheetcatalog.org/datasheets/restul/219541_DS.pdf

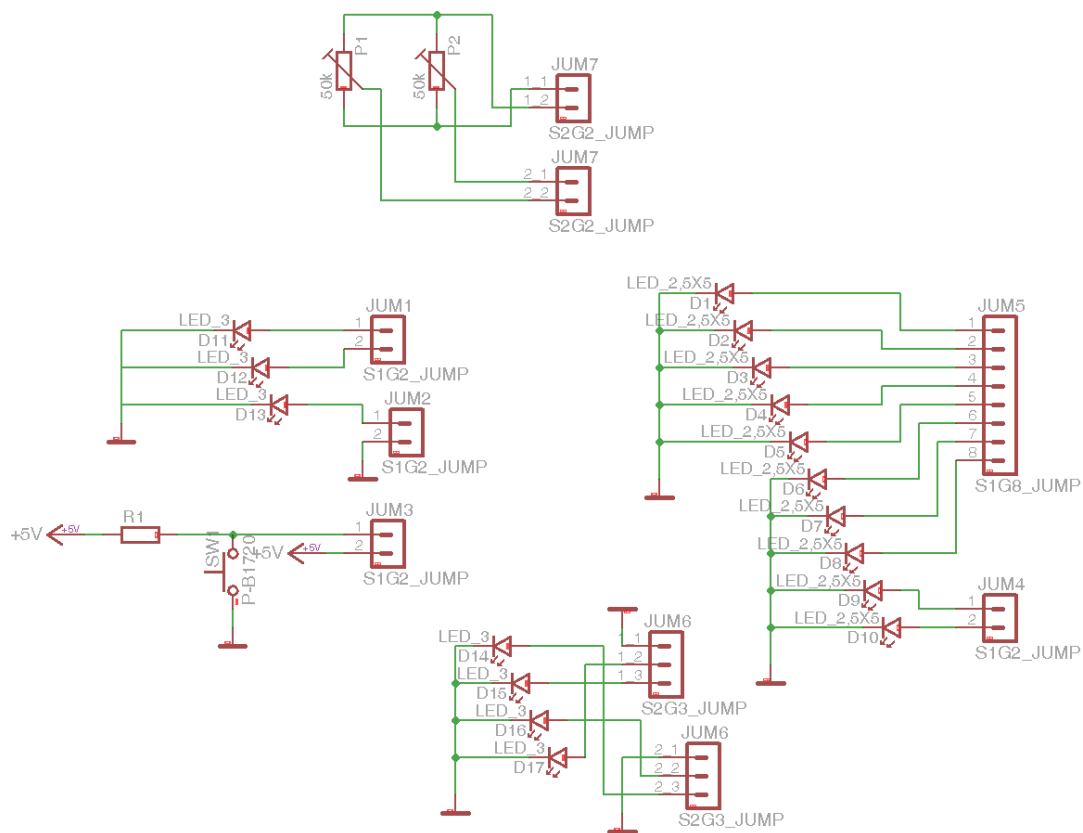
- [11] B&R Documentation, [cit. 2010-05-25], dostupné z: www.br-automation.com

- [12] Procyon AVRlib, [online], Rev. 20061029, [cit. 2010-05-25], dostupné z:
<http://www.mil.ufl.edu/~chrisarnold/components/microcontrollerBoard/AVR/avr-lib/docs/html/index.html>

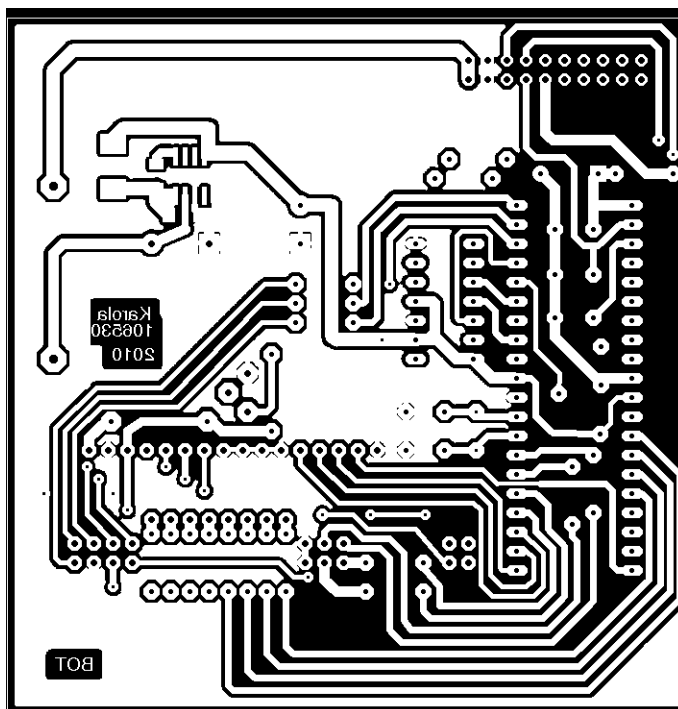
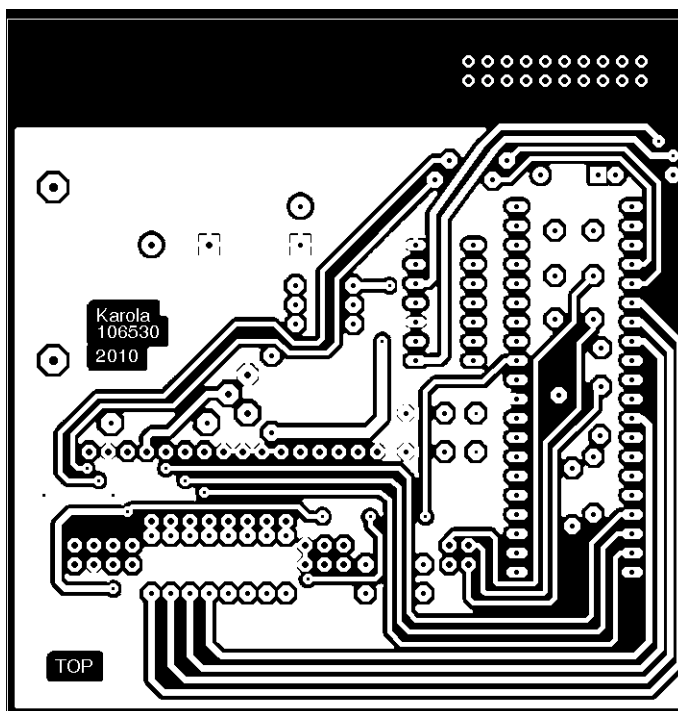
11. PŘÍLOHY

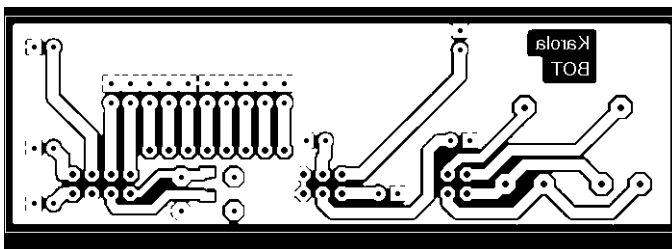
11.1 ELEKTRICKÉ SCHÉMA ZAPOJENÍ MODELU





11.2 DESKY PLOŠNÝCH SPOJŮ MODELU NÁDRŽE





11.3 ZDROJOVÝ KÓD MIKROKONTROLÉRU ATMEGA16

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <math.h>
#include <lcd.h>
#include <timer.h>
#include <string.h>
#include "global.h"
#include "lcdconf.h"

#define LED_STAV_PORT PORTD
#define LED_ON          0
#define LED_FULL        1

int          nadrz, nadrz2, odtok_ax, odtok_bx, adc_ch, prijem, prijem_kl, prijem_zap, na-
drz_spi, odtok_spi, rychl_nap_porx;
//unsigned int
float        hladina, rychl_nap2, prurez_a, prurez_b, rychl_nap, nadrz1, rychl_nap_por, od-
tok_a, odtok_b;
char         radek1[17]={"Obj.: 255.00 lit"};
char         radek2[17]={"P+:10.1 P-:10.1 "};
u16          BarGNadrz;

void BAR_G (void);
void spi_prenos (void);
void irq_capture (void);
void nadrzToChar (void);
void poruchaToChar (void);

int main (void) {

DDRC  = 0b11111111;
DDRB  = 0b10111111;
DDRA  = 0b11111000;
DDRD  = 0b11111111;

nadrz1 = 0.0;
prijem = 0;
prijem_kl = 0;
prijem_zap = 0;
nadrz_spi = 0;
prurez_a = 0.0;
prurez_b = 0.0;
MCUCSR |= ( 1 << JTD ); // JTAG disabled
MCUCSR |= ( 1 << JTD );
MCUCR  |= ( 1 << SE ); // sleep enabled

WDTCSR |= ( 1 << WDP2 ) | ( 0 << WDP1 ) | ( 1 << WDP0 ); // WatchDog 0,52s
asm ("WDR"); // WatchDog Reset
//----- CC 1 -----

TCCR1B |= ( 1 << WGM13 ) | ( 1 << WGM12 ); // mode 12 - CTC
TCCR1A |= ( 0 << WGM11 ) | ( 0 << WGM10 );

// 100ms @ 16MHz
ICR1 = 6250 - 1;
// top pro citac
// preruseni co 0.1s @1MHz
TCCR1B |= ( 1 << CS12 ) | ( 0 << CS11 ) | ( 0 << CS10 ); // clk / 256

timerAttach(TIMER1INPUTCAPTURE_INT,irq_capture);

//----- ADC -----
ADMUX |= ( 1 << REFS0 ) | ( 0 << REFS1 );
ADMUX |= ( 1 << ADLAR );
ADCSRA |= ( 1 << ADPS2 ) | ( 1 << ADPS1 ); // predelicka
ADCSRA |= ( 1 << ADEN ) | ( 1 << ADIF ) | ( 1 << ADSC ); // enable
```

```

ADMUX =      0b01100000;
            // kanal ADC0
adc_ch  =      10;

SPCR =      0b01010001;
            // init SPI
PORTB |=      0b00001000;

sei();
                                                    // globalni maska preruseni

lcdInit();
lcdInit();

TIMSK |= ( 1 << TICIE1 );
// povoleni preruseni IC
ADCSRA |= ( 1 << ADSC );

lcdGotoXY(0, 0);
lcdPrintData (radek1, 16);
lcdGotoXY(0, 1);
lcdPrintData (radek2, 16);
while( 1 )
    do {
        asm ("sleep");
        //sleep
    }

while( 1 );
return 0;

}

//----- Preruseni CC 1 -----
void irq_capture (void)
{
    cli();
    asm ("WDR");
    if      (prijem > 127) {
        rychl_nap = (float)(prijem - 128) / 40.0 / 10;
        prurez_a = 0.0;
    }
    else {
        rychl_nap = 0.0;
        prurez_a = (float)(127 - prijem) * 0.0005682;
    }

    prurez_b = (float)prijem_zap * 0.0002829;

    hladina = nadrz1 / 255.0;
    odtok_a = prurez_a * 0.97 * sqrt(2*9.81*hladina);
    odtok_b = prurez_b * 0.97 * sqrt(2*9.81*hladina);
    rychl_nap_por = (float)prijem_kl / 80.0 / 10;

    //----- Prepocet nadrze -----
    if (((nadrz1 + rychl_nap - odtok_a - odtok_b + rychl_nap_por) < 255) && ((nadrz1 + rychl_nap - odtok_a - odtok_b + rychl_nap_por) > 0)) {
        nadrz1 = nadrz1 + rychl_nap - odtok_a - odtok_b + rychl_nap_por;
        nadrz2 = (int)(nadrz1 * 100);

        //OCR2 = (int)(nadrz);
        PORTB &= 0b11111110;
        PORTB &= 0b11111101;
    }
    else {
        if ((nadrz1 + rychl_nap - odtok_a - odtok_b + rychl_nap_por) >= 255) {
            nadrz1 = 255;
            //OCR2 = 255;
            PORTB |= ( 1 << PORTB1 );
        }
    }
}
// plna

```

```

        }
        else if ((nadrz1 + rychl_nap - odtok_a - odtok_b + rychl_nap_por) <= 0) {
            nadrz1 = 0;
            //OCR2 = 0;
            PORTB |= (1 << PORTB0);

            // prazdna
        }
    }
//----- Indikace spatne regulace -----

255 if (((nadrz1 == 0) && ((prurez_a + prurez_b) > 0)) || ((nadrz1 ==
&& ((rychl_nap - odtok_a - odtok_b + rychl_nap_por) > 0)))
PORTB |= (1 << PORTB2);
else
PORTB &= 0b11111011;
if ((rychl_nap - odtok_a - odtok_b + rychl_nap_por) < 0 && (nadrz1 < 0.01)) {
    nadrz1 = 0;
    OCR2 = 0;
    PORTB |= (1 << PORTB0);
    // prazdna
}
//----- Indikace LED -----
if (prurez_a > 0)
    sbi(NADR_STAV_PORT, NADR_STAV_R_MINUS);
else
    cbi(NADR_STAV_PORT, NADR_STAV_R_MINUS);
if (prurez_b > 0)
    sbi(NADR_STAV_PORT, NADR_STAV_P_MINUS);
else
    cbi(NADR_STAV_PORT, NADR_STAV_P_MINUS);
if (rychl_nap > 0)
    sbi(NADR_STAV_PORT, NADR_STAV_R_PLUS);
else
    cbi(NADR_STAV_PORT, NADR_STAV_R_PLUS);
if (rychl_nap_por > 0)
    sbi(NADR_STAV_PORT, NADR_STAV_P_PLUS);
else
    cbi(NADR_STAV_PORT, NADR_STAV_P_PLUS);

spi_prenos();
nadrzToChar();
lcdGotoXY(0, 0);
lcdPrintData(radek1, 16);
lcdGotoXY(0, 1);
if ((prijem_kl != 0) || (prijem_zap != 0)) {
    poruchaToChar();
    lcdPrintData(radek2, 16);
}
else
    lcdProgressBar((int)nadrz1, 255, 16);

sei();
}

//----- Preruseni ADC -----
ISR(ADC_vect)
{
    cli();
    switch (adc_ch) {
        case 0:
            ADMUX = 0b01100001;
            // kanal ADC1
            adc_ch = 1;
            prijem_zap = ADCH;
            break;

        case 1:
            ADMUX = 0b01100010;
            // kanal ADC2
            adc_ch = 2;
            prijem = ADCH;
            break;
    }
}

```

```

        case 2:
            ADMUX      = 0b01100000;
            // kanal ADC0
            adc_ch      = 0;
            příjem_kl = ADCH;
            break;

        case 10:
            ADMUX      = 0b01100001;
            // kanal ADC1
            adc_ch      = 1;
            příjem      = ADCH;
            break;
    }
    BAR_G();
    sei();
}

void BAR_G (void)
{
    //----- Bar graf -----
    if (nadrz2 >= 1275)
        PORTA |= 0b00001000;
    else
        PORTA &= 0b11110111;

    if (nadrz2 >= 3825)
        PORTA |= 0b00010000;
    else
        PORTA &= 0b11101111;

    if (nadrz2 >= 6375)
        PORTA |= 0b00100000;
    else
        PORTA &= 0b11011111;

    if (nadrz2 >= 8925)
        PORTA |= 0b01000000;
    else
        PORTA &= 0b10111111;

    if (nadrz2 >= 11475)
        PORTA |= 0b10000000;
    else
        PORTA &= 0b01111111;

    if (nadrz2 >= 14025)
        PORTC |= 0b10000000;
    else
        PORTC &= 0b01111111;

    if (nadrz2 >= 16575)
        PORTC |= 0b01000000;
    else
        PORTC &= 0b10111111;

    if (nadrz2 >= 19125)
        PORTC |= 0b00100000;
    else
        PORTC &= 0b11011111;

    if (nadrz2 >= 21675)
        PORTC |= 0b00010010;
    else
        PORTC &= 0b11101111;

    if (nadrz2 >= 24225)
        PORTC |= 0b00001000;
    else
        PORTC &= 0b11110111;

```

```

}

void spi_prenos (void) {
    PORTB  &=      0b11110111;

    nadrz_spi =      (int) (nadrz1 * 16.059);
    SPDR      =      ((nadrz_spi>>8) | 0b00110000);
    while     ( (SPSR & 0b10000000) != 0b10000000);
    SPDR      =      nadrz_spi;
    while     ( (SPSR & 0b10000000) != 0b10000000);

    PORTB |=      0b00001000;
    PORTB |=      0b00001000;
    PORTB |=      0b00001000;

    PORTB  &=      0b11110111;

    odtok_spi =      (int) (odtok_b * 5980.0);

    SPDR      =      ((odtok_spi>>8) | 0b10110000);
    while     ( (SPSR & 0b10000000) != 0b10000000);
    SPDR      =      odtok_spi;
    while     ( (SPSR & 0b10000000) != 0b10000000);

    PORTB |=      0b00001000;
}

void nadrzToChar (void) {
    nadrz      =      (int) (nadrz1      *      100);

    if          ((nadrz/10000) ==      0)
        radek1[6] =      0x20;
    else
        radek1[6] =      (nadrz/10000)+48;
    if          (((nadrz/1000-(nadrz/10000)*10) ==      0) && (radek1[6] ==
0x20))
        radek1[7] =      0x20;
    else
        radek1[7] =      (nadrz/1000-(nadrz/10000)*10)+48;
    //radek1[7]
    radek1[8] =      (nadrz/100-(nadrz/1000)*10)+48;
    radek1[10]=      (nadrz/10-(nadrz/100)*10)+48;
    radek1[11]=      (nadrz-(nadrz/10)*10)+48;
}

void poruchaToChar (void) {
    odtok_ax =      (int)(odtok_a*100);
    odtok_bx =      (int)(odtok_b*100);
    rychl_nap_porex =      (int)(rychl_nap_por*100);

    if          ((rychl_nap_porex/100) ==      0)
        // P+
        radek2[3] =      0x20;
    else
        radek2[3] =      (rychl_nap_porex/100)+48;
    radek2[4] =      (rychl_nap_porex/10-(rychl_nap_porex/100)*10)+48;
    radek2[6] =      (rychl_nap_porex-(rychl_nap_porex/10)*10)+48;
    radek2[7] =      0x20;

    if          ((odtok_bx/100) ==      0)
        radek2[11]=      0x20;
    else
        radek2[11]=      (odtok_bx/100)+48;
    radek2[12]=      (odtok_bx/10-(odtok_bx/100)*10)+48;
    radek2[14]=      (odtok_bx-(odtok_bx/10)*10)+48;
    radek2[15]=      0x20;
}

```