

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra elektroenergetiky a ekologie

BAKALÁŘSKÁ PRÁCE

Integrace web serveru do technických prostředků ZAT

2018

Roman Kúhnel

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Roman KÜHNEL**
Osobní číslo: **E15B0015K**
Studijní program: **B2644 Aplikovaná elektrotechnika**
Studijní obor: **Aplikovaná elektrotechnika**
Název tématu: **Integrace web serveru do technických prostředků ZAT**
Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

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

1. Popište technické prostředky řídicího systému ZAT Sandra.
2. Popište možnou integraci web serveru do řídicího systému Sandra.
3. Optimalizujte řídicí systém Sandra.
4. Vytvořte jednoduché vzorové aplikace.

Rozsah grafických prací: podle doporučení vedoucího

Rozsah kvalifikační práce: 30 - 40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: Ing. Jana Jiříčková, Ph.D.

Katedra elektroenergetiky a ekologie

Datum zadání bakalářské práce: 10. října 2017

Termín odevzdání bakalářské práce: 7. června 2018


Doc. Ing. Jirí Hammerbauer, Ph.D.
děkan




Prof. Ing. Václav Kús, CSc.
vedoucí katedry

V Plzni dne 10. října 2017

Anotace

Předpokládaná bakalářská práce popisuje technické prostředky průmyslového řídicího systému Sandra firmy ZAT a.s.. Ukazuje aplikaci webového serveru prostřednictvím hardware Sandra a software Pertinax6 a Grafik. Dále je zde popsána implementace a zobrazení postupu práce s webovým severem, ke kterému slouží řídicí systém od společnosti ZAT a.s. s označením SandRA Z210. Součástí práce je také vzorová aplikace.

Klíčová slova

Server, web, Sandra, Pertinax, Grafik, implementace, technické prostředky

Abstract

This bachelor's thesis describes the technical means of the Sandra industrial control system from ZAT a.s.. Shows the web server application through Sandra hardware and Pertinax6 and Grafik software. Here is also described the implementation and display of the process of work with web north, which is operated by ZAT a.s. labeled SandRA Z210. Part of the thesis is also a sample application.

Key words

Server, web, SCADA, hardware, software, Sandra, Pertinax, Grafik, implementation, technical resources

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....

podpis

V Plzni dne 4.6.2018

Roman Kúhnel

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Janě Jiříčkové, Ph.D. za přínosné připomínky a metodické vedení práce. Dále bych chtěl poděkovat Ing. Zdeňku Tupému za cenné profesionální rady a konzultace z praxe. Chtěl bych také poděkovat svým nejbližším za podporu během celého studia.

Obsah

OBSAH	8
ÚVOD	9
SEZNAM SYMBOLŮ A ZKRATEK	10
ZAT- TECHNICKÉ PROSTŘEDKY	11
1.1 Řídící systémy ZAT a.s. včetně mechanického provedení jednotlivých prostředků.....	11
1.1.1 První generace.....	11
1.1.2 Druhá generace.....	12
1.1.3 Třetí generace.....	12
1.1.4 Čtvrtá generace	12
1.2 SWAROVÉ PROSTŘEDKY PRO PROGRAMOVÁNÍ	15
1.2.1 Pertinax	15
1.2.2 Popis SCADA ZAT Grafik	19
WEBOVÝ SERVER	24
2.1 POJEM WEBOVÝ SERVER.....	24
2.2 DŮVODY IMPLEMENTACE WEB SERVERU	25
2.3 POSTUP IMPLEMENTACE DO ŘS SANDRA Z210	26
MOŽNOST OPTIMALIZACE	27
3.1 APLIKOVANÉ OPTIMALIZACE.....	27
3.2 NÁVRHY KE ZLEPŠENÍ	28
UKÁZKA APLIKACE WEB SERVERU DO ŘS SANDRA	30
4.1 PŘEHLED OBRAZOVEK.....	30
4.2 POPIS FUNKČNOSTI PLNĚNÍ VODNÍ NÁDRŽE	34
4.3 UKÁZKA PROGRAMU	35
ZÁVĚR	36
SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	38
SEZNAM OBRÁZKŮ	39

Úvod

Bakalářská práce je zaměřena na představení výrobků společnosti ZAT a.s. Tato společnost prošla velkým vývojem. Již více jak 55let zajišťuje automatizaci v domácím i světovém průmyslu. Její produktové portfolio je velmi široké. Zabývá se řízením monitoringem od jednoduchých zařízení až po velké technologické celky, kterými jsou například jaderné elektrárnské bloky. To však není jediné odvětví, kde můžeme vidět práci této firmy. Produkty společnosti ZAT využívá také obsluha k monitoringu technologických celků v dopravě, plynárenství a povrchových dolech.

V současné době existuje na trhu stále větší potřeba vzdáleného dohledu na danou technologii. I díky tomuto trendu došlo ke značnému vývoji v oblasti webových serverů na ZAT. Webové servery se hojně používají jako levná varianta vizualizace pro SCADA. Webovými servery je však také vybaveno téměř každé zařízení, které se připojuje prostřednictvím sítě ethernet. Což mohou být aktivní síťové prvky jako jsou switche, IP kamery ale také PLC. Často však u těchto zařízení slouží pouze jen k parametrizaci zařízení. Pro potřeby webového klienta může však sloužit i PLC Z210 od firmy ZAT a.s., které je jedním z posledních přírůstků do velké rodiny průmyslových řídicích systémů Sandra.

Přičemž je pro implementaci webového serveru do PLC Sandra Z210 využito již dříve vyvinutého programu Grafik, který však byl značně přizpůsoben novému rozhraní a nyní nám slouží pro grafické znázornění technologie a ovládacích prvků. K úspěšnému zprovoznění automatu, je nutné naimportovat grafické objekty, včetně adres do Pertinaxu6, ve kterém se vytvoří zbývající algoritmy. Tento celek se následně nahrává do automatu. Tato aplikace bude po nahrání do PLC dostupná přes bezdrátové připojení WI-FI.

Bakalářská práce je rozdělena do čtyř sekcí, popisuje prostředky firmy ZAT, práci se SW Pertinax a Grafik. Dále popisuje aplikovaná vylepšení webového serveru a popisuje další možný rozvoj do budoucna. V závěru práce popisuje vzorovou aplikaci s využitím tohoto webového serveru na příkladu ovládání hladiny nádrže.

Seznam symbolů a zkratek

SW.....Software

HW.....Hardware

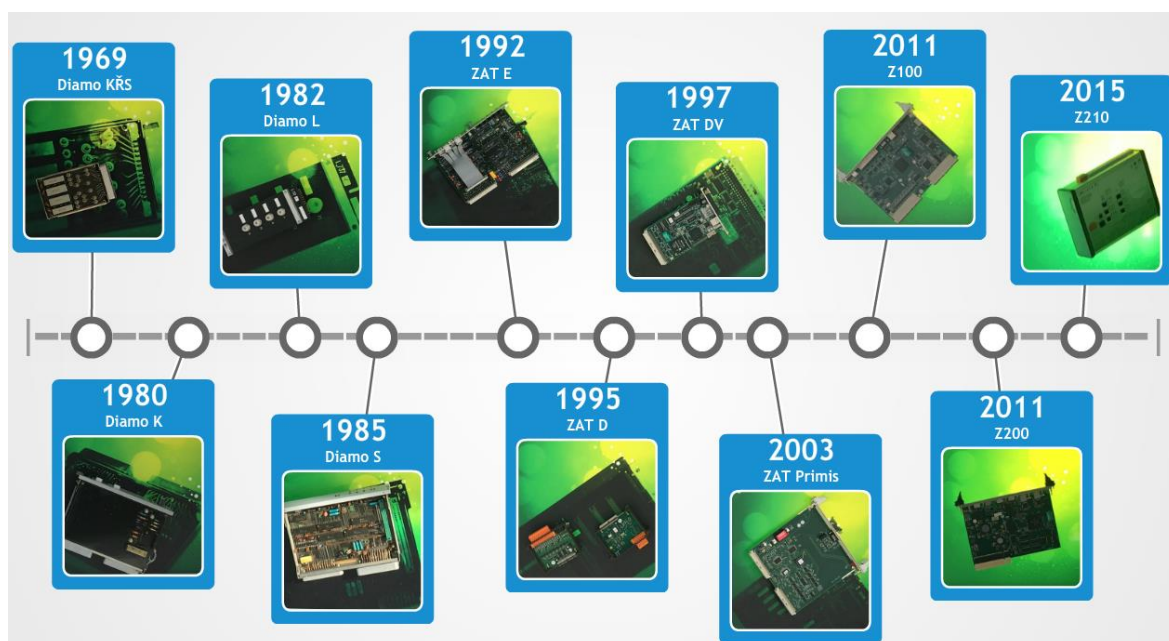
Sandra..... Safe and Reliable Automation

SCADA..... Supervisory Control And Data Acquisition

DCS.....Distributed control system

WI-FI.....Wireless Fidelity

ZAT- technické prostředky



Obrázek 1 - Časová osa historie prostředků firmy ZAT

1.1 Řídicí systémy ZAT a.s. včetně mechanického provedení jednotlivých prostředků

Společnost ZAT sídlící v **Plzni** a **Příbrami** je nejstarší českou firmou v oboru **automatizace technologických procesů** s více než padesátiletou historií. Orientuje se na vývoj, projekci, výrobu, instalaci i servis elektronických zařízení, řídicích systémů a jejich komponent a mimo jiné i na výrobu zdravotnických přístrojů.

Vše začalo v roce 1962, kdy ve Vývojové základně uranového průmyslu v Kamenné u Příbrami vzniklo středisko Automatizace. Zaměřuje se na vývoj a výrobu automatizačních prostředků a speciálních čidel pro důlní těžbu. Jde o rozvaděče na bázi relé, stykačů, spínačů a podobných elektromechanických prvků, které vykonávají základní logické funkce pro řízení důlních zařízení, jako je čerpání vod, doprava vozíků a podobně.

1.1.1 První generace

Začátek modulárních řídicích systémů přichází v roce 1969 pod značkou DIAMO. Řídicí funkce v této generaci systémů jsou vytvářeny propojováním desek realizujících základní logické operace.

Představitelem první generace byl modulární systém DIAMO KŘS založený na diodo-reléové logice. Základním prvkem bylo miniaturní jazýčkové relé, které umožnilo realizovat základní logické funkce na zásuvných deskách a pomocí jejich propojení vytvářet i složité funkce logického řízení. Rozvoj polovodičové elektroniky se promítl do nové řady řídicího systému DIAMO K. Základní funkce byly realizovány pomocí polovodičových integrovaných obvodů.

1.1.2 Druhá generace

Řídicí systémy ZAT charakterizuje nástup mikroprocesorů do automatizační techniky. To přináší naprosto nový způsob tvorby řídicích algoritmů, které už nevznikají drátovým propojením jednotlivých desek, ale programují se pomocí softwarových prostředků. Prvním systémem řízeným mikroprocesorem byl systém DIAMO L uvedený do výroby v roce 1982. Byl určený především pro potřeby uranového průmyslu a dalších provozů s náročným prostředím. Například pro řízení velkého množství technologií pro vtláčení vody do podzemních vrtů v rámci rozsáhlé hydraulické bariéry při chemické těžbě uranu.

1.1.3 Třetí generace

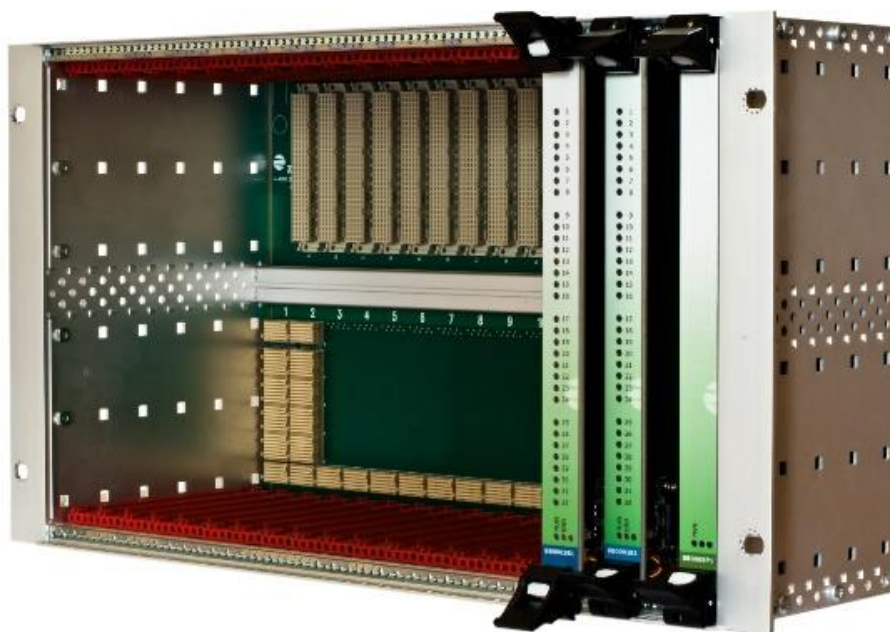
Vzniká v druhé polovině devadesátých let minulého století a je charakterizovaná postupným přechodem od osmibitových procesorů k výkonným řídicím stanicím s 32bitovými procesory. Třetí generaci reprezentuje v roce 1995 řídicí systém ZAT-D. Jde o systém s novou architekturou distribuovaných vstupně/výstupních stanic komunikujících s řídicí jednotkou pomocí sériové sběrnice standardu Profibus DP. Tato koncepce umožňuje rozmístit stanice vstupů a výstupů po řízené technologii a výrazně tak snížit náklady na kabeláž připojující snímače a akční členy k řídicímu systému. Systém ZAT-D byl dodán na řadu elektráren a tepláren, například na elektrárně.

1.1.4 Čtvrtá generace

Řídicí systém ZAT pod obchodní značkou SandRA. Jde o spojení slov Safe and Reliable Automation charakterizujících základní vlastnosti – bezpečnost a spolehlivost – této automatizační platformy. Ta zahrnuje hardwarové prostředky i softwarové nástroje pro realizaci komplexního řešení řídicího systému třídy DCS.

Představení Z200 a Z210

Řídicí stanice řady **SandRA Z200** představují otevřený řídicí systém s vysokým výpočetním výkonem a rozsáhlými komunikačními možnostmi. Variabilita hardwarové konfigurace řídicí stanice je podporovaná širokou nabídkou vstupních, výstupních a komunikačních desek. Flexibilní rozšiřování funkčnosti softwarového vybavení je umožněno díky implementovanému operačnímu systému Linux. Ve spojení s programovacím nástrojem Pertinax6.



Obrázek 2 - PLC ZAT SandRA Z200 [3]

Kompaktní stanice řady SandRA Z210 vycházejí z řady Z200 a doplňují portfolio řídicího systému SandRA o prostředky pro řízení menších technologií s malým počtem vstupů a výstupů, jako jsou regulační a předávací stanice v plynárenství, zásobníky plynu, čističky odpadních vod, tepelné výměníky, monitoring a řízení produktovodů, dohledové drážní systémy apod. Z210 jsou vhodné také na realizaci komunikační bridge pro sběr dat z rozličných koncových zařízení s různými komunikačními protokoly a jejich přenos na vzdálená dispečerská pracoviště.



Obrázek 3 - PLC ZAT SandRA Z210 [2]

Základní vlastnosti Z210:

- Rozsáhlé komunikační možnosti
 - kanály Fast Ethernet 100BasE-tX
 - kanály sériových rozhraní (softwarově volitelné Rs-232/Rs-485/Rs-422)
 - 1 kanál UsB 1.1

- Výkonný hardware
 - výkonný Procesor Freescale Power QUicc ii MPc8270 (855 MiPs@450MHz)
 - slot pro paměťovou kartu typu sd
 - zálohovaný obvod reálného času (Rtc)
 - napájení 24v dc, možnost redundantního napájení ze dvou zdrojů

- Připojení vstupů a výstupů
 - řídicí stanice řady z210 využívají pro připojení technologických signálů systém vzdálených vstupů a výstupů X20.
 - tyto vstupy/výstupy jsou pomocí komunikačního rozhraní (Ethernet) připojeny k řídicí stanici.

1.2 Softwarové prostředky pro programování

1.2.1 Pertinax

Pertinax6 je vlastní vývojový software vyvinutý společností ZAT a.s. Tento vývojový software je základním prostředkem pro tvorbu uživatelského software procesních stanic, zároveň umožňuje nahrávání aplikace a komplexní správu stanice.

Hlavními částmi integrovaného prostředí Pertinax6 jsou grafický projekční systém ZatCAD a integrované SW nástroje pro konfiguraci (Správce projektu) a komunikaci (komunikační server KoS). Prostedí zahrnuje množství dalších aplikací a funkcí.

Vlastnosti:

- Podpora projektové databáze s návazností na vizualizační úroveň
- Snadné programování aplikací pomocí grafických objektů (funkční bloky)
- Dálková parametrizace uživatelského software
- Datová komunikace prostřednictvím několika typů protokolů
- Nástroj OpenVPN pro bezpečné spojení mezi stanicemi a jejich okolím v nedůvěryhodných (nezabezpečených) ethernetových sítích

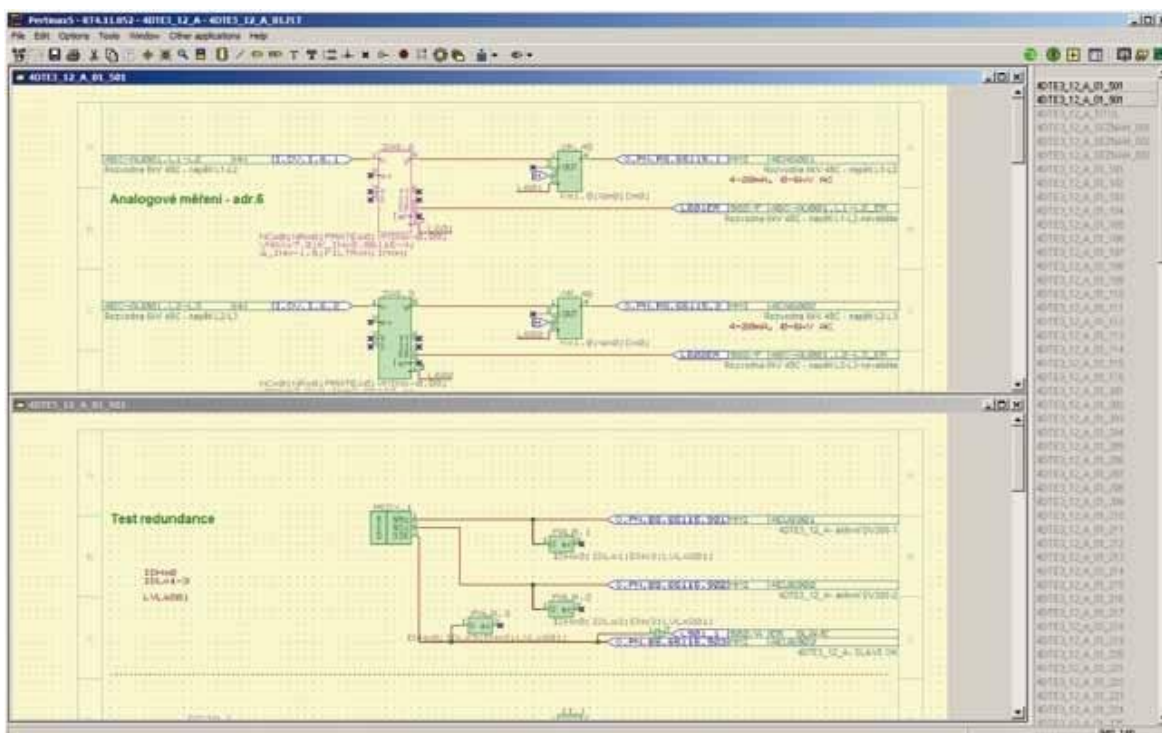
ZatCAD má tři základní pracovní režimy. Režim Editor umožňuje vytvoření žádané funkce a její kompilaci. V režimu Simulátor lze provést kontrolu funkce pomocí simulace signálů bez připojené stanice. Režim Analyzátor podporuje práci s připojenou stanicí a umožňuje nahrát SW moduly do stanice (měnit její aktuální stav) nebo číst data ze stanice (monitorovat její aktuální stav). [6]

Pracovní režimy:

- Editor – konfigurace, kreslení, překlad, křížové kontroly
- Simulátor – kontrola funkce off-line na PC
- Analyzátor – nahrávání stanice, diagnostika stanice on-line, čtení ze stanice, úprava parametrů (on-line), grafy, trendy atd.

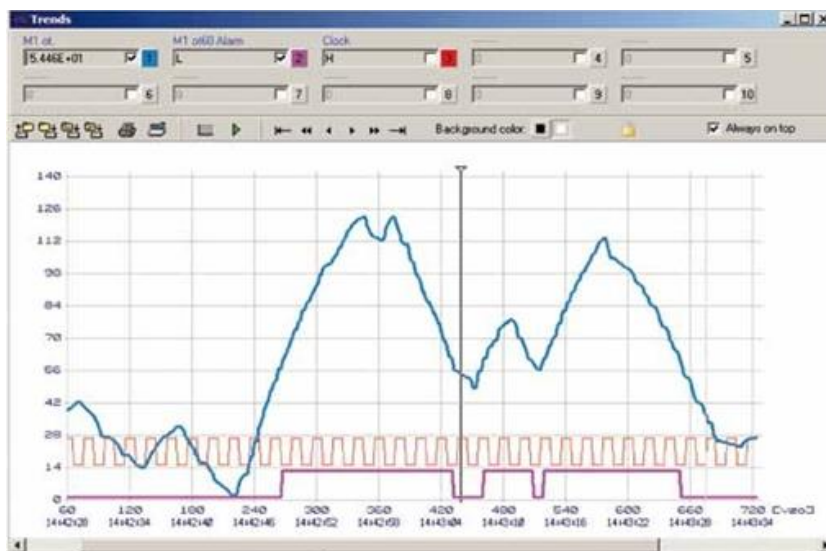
Výhody systému:

- Jednotný programovací prostředek pro procesní stanice v systémech ZAT
- Základem systému je projekt, který zahrnuje vlastnosti HW i SW
- Správa verzí systémového SW
- Rozhraní stanice definováno přes ovladače příslušných periferních zařízení
- Bez nárazová aktualizace uživatelského SW
- Zobrazení aktuálních hodnot algoritmu



Obrázek 4 – Ukázka programu Pertinax6

Integrované prostředí umožňuje sledovat živé hodnoty algoritmu v připojené stanici. V Analyzátoru se volí zobrazení aktuálních hodnot v označených bodech algoritmu. Průběh vybraných hodnot lze též zobrazit jako graf proměnných v závislosti na reálném čase. Průběh vybraných hodnot, může být také zobrazován jako graf proměnných v závislosti na reálném čase. Záznam může být také uložen do souboru pro další zpracování.



Obrázek 5 – Okno analyzátoru

Práce s aplikací:

- Konfigurace hardware
- Tvorba algoritmů
- Nahrání a spuštění řídicí desky
- Sledování systému

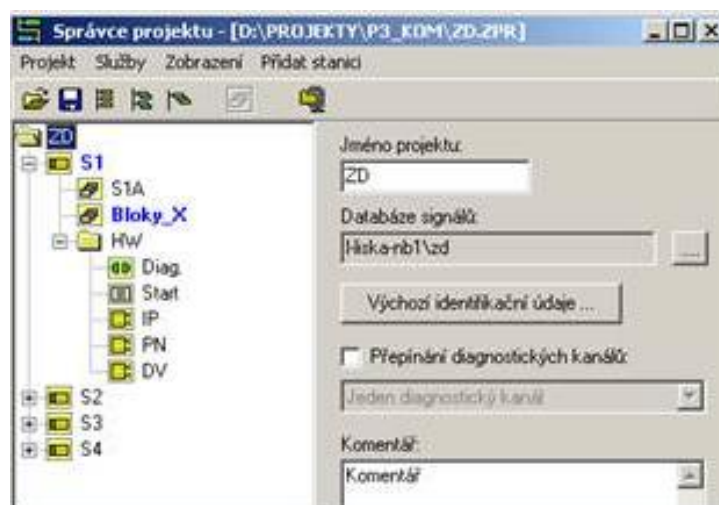
Konfigurace systému

Pro konfiguraci systému slouží Správce projektu v režimu Editor. Informace o konfiguraci systému Pertinax6 se ukládají do projektového souboru *.ZPRX.

Při spuštění systému Pertinax6 se po zobrazení úvodního okna otevírá Správce s posledním aktuálně uloženým projektovým souborem. Tato vlastnost je předem definována (viz Editor, Nastavení, karta Editor, položka *Načíst poslední projekt*). Pokud nebyl projektový soubor *.ZPRX otevřen automaticky, objeví se okno pro otevření stávajícího nebo zadání nového projektového souboru

Při konfiguraci se ve Správci projektu definují následující objekty: projekt, stanice, ovladač, úloha, systémový ovladač, startovací soubor

Projekt



Obrázek 6 – Okno správce projektu

Jméno projektu „ZD“ lze libovolně editovat, zobrazuje se ve stromové struktuře. Výchozí jméno po prvním otevření projektu: *Nový projekt*. Pod projektem jsou umístěny jednotlivé stanice „S1, S2, S3, S4“ pod kterými jsou umístěny jednotlivé úlohy a pod nimi už samotné výkresy. [5]

Zpracování uživatelských úloh

Uživatelská úloha se vytváří v režimu Editor. Úloha se skládá ze vstupních a výstupních signálů, které jsou propojeny přes tělo algoritmu. Algoritmus je záznam předpisů pro dosažení žádaného cílového chování řízeného procesu. Signály mohou být binární nebo analogové, přičemž analogové se ještě dělí na několik typů (viz dále). Algoritmus se skládá z navzájem pospojovaných funkčních bloků. Funkční blok je základní prvek algoritmu s definovanou funkcí (logickou, numerickou, sekvenční) a je graficky znázorněn svojí značkou. Vstup a výstup na spoji musí být vždy stejného typu. Nelze spojovat binární signály s analogovými nebo analogové signály různých typů navzájem.

Nahrávání a spouštění řídicí desky

Pro nahrání stanice se musí použít příkaz **Synchronizovat** (viz Analyzátor, Nástroje, Odeslat soubor). Je zobrazen seznam mazaných souborů ve stanici, je třeba jej potvrdit, zamítnout mazání nebo celou operaci zrušit.

Sledování systému

Stavové informace zobrazuje v prostředí Analyzátoru příkaz Správce projektu, Služby, **Stav systému** (je přístupný jen při označení objektu Stanice, musí být otevřena 1 úloha dané stanice). Zobrazí se údaje o systému (indikace spuštění hlavního modulu a počet spuštěných ovladačů, časování a redundance). Dále se zobrazí údaje o běžících úlohách – jejich počet a pro každou stav, mód spuštění a cyklus úlohy. Podrobný popis je v příloze E.1. - Správce projektu. [5]

1.2.2 Popis SCADA ZAT Grafik

Program GRAFIK je určen jako cenově velmi atraktivní alternativa klasických monitorovacích a řídicích systémů pro malé aplikace. V takovýchto aplikacích je zpravidla cena standardně používané vizualizace (software InTouch + odpovídající hardware) příliš vysoká. Z tohoto důvodu byl vyvinut programový systém GRAFIK.

Princip funkce

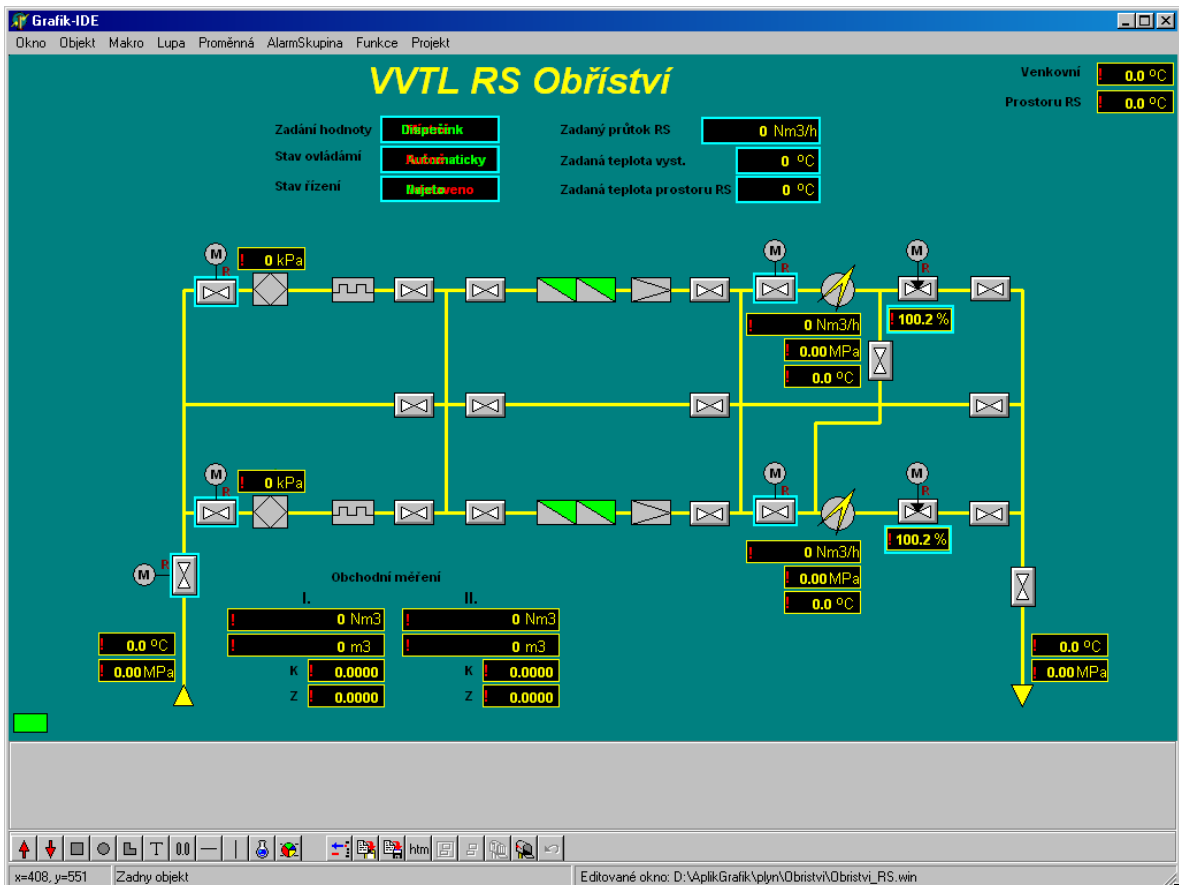
Programový systém se skládá ze dvou programů:

- Program GRAFIK-IDE představuje vývojové prostředí, v něm se tvoří databáze signálů a jednotlivé technologické obrazovky.
- Program GRAFIK-RT představuje RunTime, tedy tu část, která na základě dat, vygenerovaných programem GRAFIK-IDE provádí vlastní On-Line vizualizace a eventuální řízení.

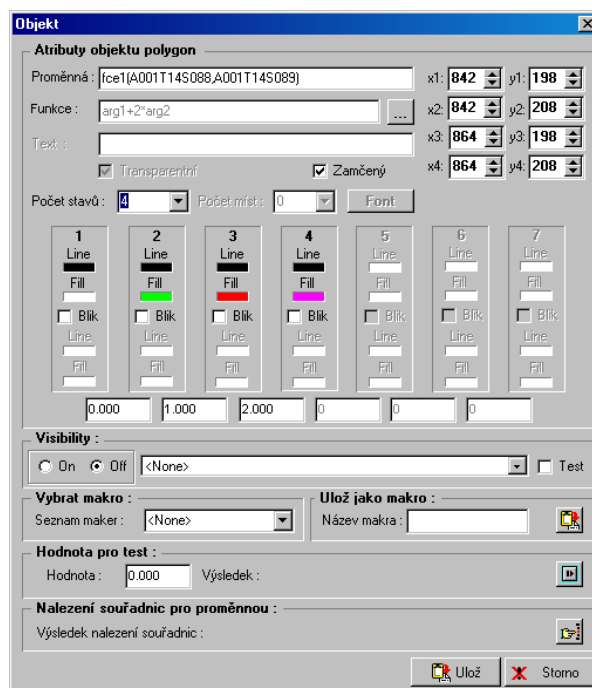
Tvorba technologických snímků je založena na principu rozmístování jednotlivých objektů předdefinovaných typů (kružnice, displej apod.) na popředí snímku. Pozadí může být (ale nemusí) tvořeno bitmapou v odpovídajícím rozlišení. Každý objekt má k dispozici sadu atributů (barva popředí, barva pozadí apod.), které lze dynamicky měnit v závislosti na aktuálních hodnotách procesních proměnných. Pro čtení a eventuální zápis hodnot z / do technologie používá standardní komunikaci DDE. Proto lze jako interface na technologii použít kterýkoliv z IO, resp. DDE serverů, dostupných pro systém InTouch. Lze tudíž komunikovat protokoly Profibus FMS, PERNET nebo MODBUS. Zvláště poslední dva jsou pro levné aplikace výhodné. Ethernetový port má dnes k dispozici každý počítač a rovněž všechny modernější řídicí desky, používané v systémech ZAT. Pokud je vzdálenost mezi technologií, resp. automatem a operátorskou stanicí velká, nebo nelze použít běžnou Ethernetovou kabeláž, lze s výhodou použít komunikaci MODBUS, která funguje prakticky po jakémkoliv v praxi používaném kabelu.

Vývojové prostředí – program GRAFIK-IDE

Vývojové prostředí slouží k tvorbě obrazovek a k vytváření databáze signálů. Obrazovky se skládají ze statického pozadí a dynamických objektů na popředí. S objekty lze pohodlně manipulovat pomocí myši (přesun objektu, změna velikosti a atributů objektu atd.). Systém podporuje následující typy objektů: obdélník, polygon, kruh, text, displej, bitmapa, analogový a binární povel, bargraf

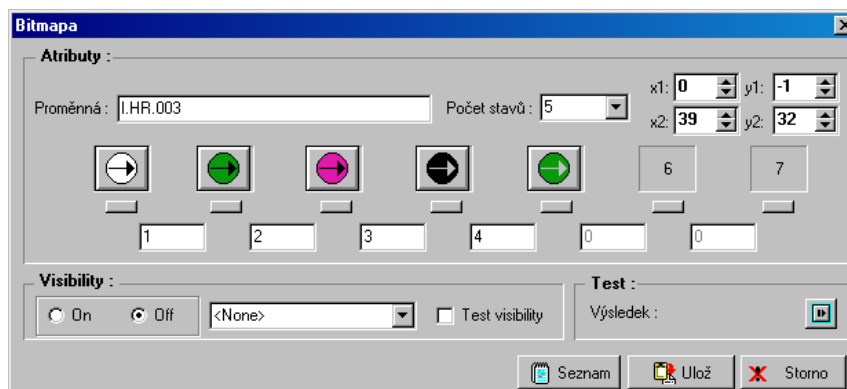


Obrázek 7 – Okno vývojového prostředí Grafik



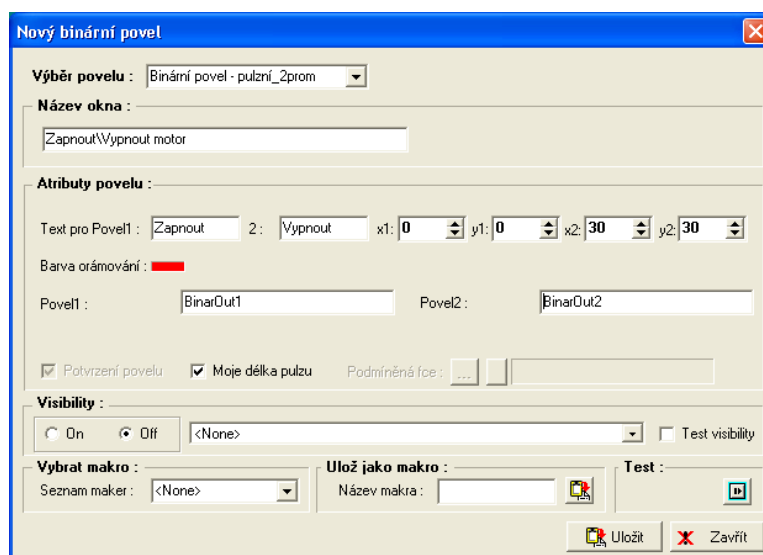
Obrázek 8 – Nastavení atributů objektů

Objekt má sadu atributů (barva čáry, barva výplně, blikání). Atributy lze přiřadit maximálně sedmi stavům dané proměnné: u analogových proměnných lze definovat stavy jejich dolních a horních mezí, pro binární mají samozřejmě význam pouze dva stavy. Na pozadí snímku lze rovněž vkládat objekt Bitmapa. Jedná se vždy o sadu předem nadefinovaných malých bitmap, které se dle stavu analogové nebo binární proměnné zobrazují na daném místě ve snímku.



Obrázek 9 – Nastavení adresy povelu

Pro možnost povelování jsou k dispozici objekty pro analogový a binární povel. Binární povel mohou být impulzní i trvalé.

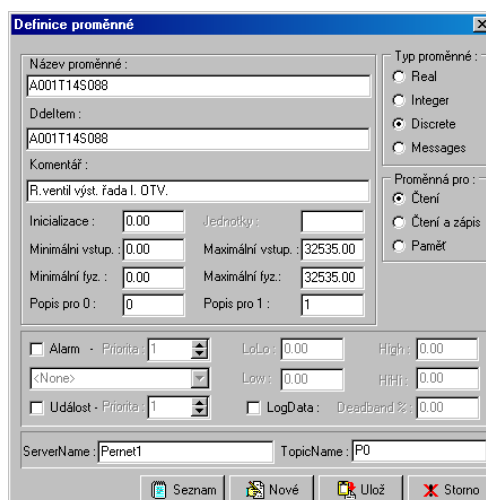


Obrázek 10 – Vlastnosti binárního povelu

Podporované typy proměnných jsou REAL, INTEGER, DISVRETE, MESSAGES, MEMORY

Proměnné mohou být pouze pro čtení nebo pro čtení i zápis, analogovým proměnným lze definovat lineární přepočtovou funkci na fyzikální hodnotu. Pro binární proměnnou lze určit popis pro stav '1' a '0'. Lze definovat počáteční hodnotu při inicializaci (než se naváže spojení s technologií), dvě meze pro analogové signály a samozřejmě adresaci signálu podle standardu DDE: Application – Topic – Item.

Příklad definice proměnné:



Obrázek 11 - Definice proměnné

Dále je možno definovat danou proměnnou jako alarmní s alarmní skupinou, prioritou 1 až 999 anebo jako událost s prioritou 1 až 999 Pro analogovou

proměnnou lze stanovit alarmní meze Low a High. Všechny proměnné jsou přehledně zobrazeny v seznamu proměnných.

Porad	Proměnná	Ben	Iconentál	Typ	HW	Server/Topic	Envid	PolICE	Alarm	Ukolici	Směrnice	Posun
16	Analog3	A0021055000	Analog vstup 3	Integer	RO	Portnet IP0	12	12	ANO	NE	1	0
17	Analog4	A0021055004	Analog vstup 4	Integer	RO	Portnet IP0	13	13	NE	NE	1	0
18	Analog5	A0021055005	Analog vstup 5	Integer	RO	Portnet IP0	33	32	NE	NE	1	0
19	Analog6	A0021055006	Analog vstup 6	Integer	RO	Portnet IP0	49	48	NE	NE	1	0
110	Analog7	A0021055007	Analog vstup 7	Integer	RO	Portnet IP0	50	47	NE	NE	1	0
111	Analog8	A0021055008	Analog vstup 8	Integer	RO	Portnet IP0	51	49	ANO	NE	1	0
112	Analog9	A0021055009	Analog vstup 9	Integer	RO	Portnet IP0	52	49	NE	NE	1	0
113	Analog10	A0011055001	Analog vstup 10	Integer	RW	Portnet IP0	5	8	NE	NE	1	0
114	Analog10	A0011055010	Analog vstup 10	Integer	RW	Portnet IP0	46	43	NE	NE	1	0
115	Analog11	A0011055011	Analog vstup 11	Integer	RW	Portnet IP0	47	44	NE	NE	1	0
116	Analog12	A0011055012	Analog vstup 12	Integer	RW	Portnet IP0	48	45	NE	NE	1	0
117	Analog13	A0011055002	Analog vstup 3	Integer	RW	Portnet IP0	6	7	NE	NE	1	0
118	Analog13	A0011055003	Analog vstup 3	Integer	RW	Portnet IP0	7	8	ANO	NE	1	0
119	Analog14	A0011055004	Analog vstup 4	Integer	RW	Portnet IP0	8	9	NE	NE	1	0
20	Analog15	A0011055005	Analog vstup 5	Integer	RW	Portnet IP0	39	38	NE	NE	1	0
21	Analog16	A0011055006	Analog vstup 6	Integer	RW	Portnet IP0	42	39	NE	NE	1	0
22	Analog17	A0011055007	Analog vstup 7	Integer	RW	Portnet IP0	43	40	NE	NE	1	0
23	Analog18	A0011055008	Analog vstup 8	Integer	RW	Portnet IP0	44	41	ANO	NE	1	0
24	Analog19	A0011055009	Analog vstup 9	Integer	RW	Portnet IP0	45	42	NE	NE	1	0
26	Binari1	A00211450001	Binar vstup 1	Discrete	RO	Portnet IP0	18	20	NE	NE	1	0
27	Binari2	A00211450002	Binar vstup 2	Discrete	RO	Portnet IP0	19	21	ANO	NE	1	0
27	Binari3	A00211450003	Binar vstup 3	Discrete	RO	Portnet IP0	20	22	NE	NE	1	0
28	Binari4	A00211450004	Binar vstup 4	Discrete	RO	Portnet IP0	23	23	NE	NE	1	0
29	Binari5	A00211450005	Binar vstup 5	Discrete	RO	Portnet IP0	14	14	ANO	NE	1	0
30	Binari6	A00211450006	Binar vstup 6	Discrete	RO	Portnet IP0	21	17	ANO	NE	1	0
31	Binari7	A00211450007	Binar vstup 7	Discrete	RO	Portnet IP0	16	16	ANO	NE	1	0
32	Binari8	A00211450008	Binar vstup 8	Discrete	RO	Portnet IP0	17	18	ANO	NE	1	0
33	Binari11	A00111450001	Binar vstup 1	Discrete	RW	Portnet IP0	15	16	NE	ANO	1	0
34	Binari12	A00111450002	Binar vstup 2	Discrete	RW	Portnet IP0	22	19	NE	NE	1	0
35	Binari13	A00111450003	Binar vstup 3	Discrete	RW	Portnet IP0	38	37	NE	NE	1	0
36	Binari14	A00111450004	Binar vstup 4	Discrete	RW	Portnet IP0	24	29	NE	ANO	1	0
37	ParCst1			Integer	PA		3	3	NE	NE	1	0
38	ParCst2			Integer	PA		4	4	NE	NE	1	0
39	ParCst3			Integer	PA		37	36	NE	NE	1	0
40	ParCst1			Discrete	PA		1	1	ANO	NE	1	0
41	ParCst10			Discrete	PA		36	36	ANO	NE	1	0
42	ParCst11		Alarm1	Discrete	PA		36	35	ANO	NE	1	0
43	ParCst2			Discrete	PA		2	2	NE	NE	1	0
44	ParCst3			Discrete	PA		11	NE	NE	NE	1	0
45	ParCst4			Discrete	PA		25	24	NE	NE	1	0
46	ParCst5			Discrete	PA		37	26	NE	NE	1	0
47	ParCst6			Discrete	PA		28	28	NE	NE	1	0
48	ParCst7			Discrete	PA		37	27	NE	NE	1	0
49	ParCst8			Discrete	PA		29	29	NE	NE	1	0
50	ParCst9			Discrete	PA		26	30	ANO	NE	1	0
51	Prout1	Prout1		Real	RO	lib.aktiva-nb17agrame182	59	NE	NE	1	0	
52	Prout2	Prout2		Real	RO	lib.aktiva-nb17agrame183	60	NE	NE	1	0	
53	Prout	A0021060000	Real vstup 1	Real	RO	Portnet IP0	56	53	ANO	NE	1	0
54	Prout	A0021060001	Real vstup 2	Real	RO	Portnet IP0	57	54	NE	NE	1	0
55	Prout	A0021060002	Real vstup 3	Real	RO	Portnet IP0	58	55	NE	NE	1	0
56	Prout	A0021060003	Real vstup 4	Real	RO	Portnet IP0	59	56	NE	NE	1	0
57	Prout1	A0011060001	Real vstup 1	Real	RW	Portnet IP0	60	57	NE	NE	1	0
58	Prout2	A0011060002	Real vstup 2	Real	RW	Portnet IP0	61	58	NE	NE	1	0
59	Prouten.00	SO		Integer	RO	Portnet IP0	70	70	NE	NE	1	0
60	Prouten.01			Discrete	PA		40	62	NE	NE	1	0

Obrázek 12 - Definice matematické funkce

Dále je možno definovat základní matematické funkce, které umožňují provádět výpočty s nadefinovanými proměnnými a podmíněné funkce, které se vykonají na základě splněných podmínek. [5]

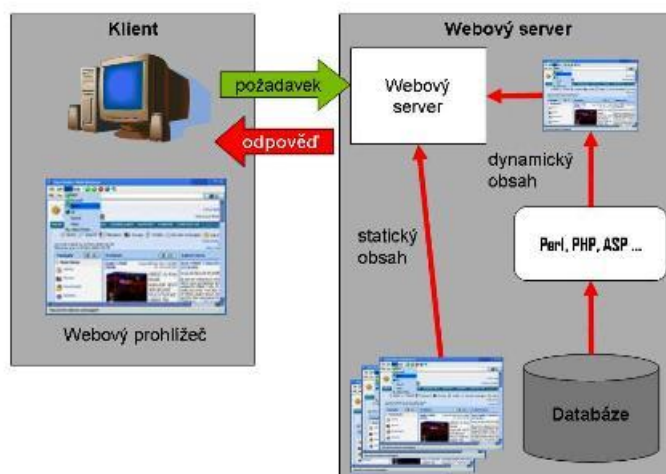
Export pro webový sever se provádí přes položku Projekt a dále Exportzip.

Webový server

2.1 Pojem webový server

Webové servery se hojně používají jako levná varianta vizualizace pro SCADA. Webovými servery je však také vybaveno téměř každé zařízení, které se připojuje prostřednictvím ethernet. Což mohou být aktivní síťové prvky jako jsou switche, IP kamery ale také PLC. Často však u těchto zařízení slouží pouze jen k parametrizaci zařízení. Každý webový server je připojen k síti a přijímá požadavky ve tvaru HTTP. Tyto požadavky vyřizuje a vrací počítači, který požadavek vznesl, vrací odpověď. Odpověď obvykle představuje nějaký HTML dokument. Může to být ale i dokument v jiném formátu – text, obrázek apod. Odpověď serveru je opět ve tvaru HTTP, je uvozena hlavičkou obsahující stavový kód, za níž následuje samotný obsah.

Nejpoužívanějším webovým serverem vůbec je Apache HTTP Server, následovaný především Internet Information Services a nginx.



Obrázek 13 - Komunikace mezi klientem a webovým serverem [9]

Webový server (program) má téměř vždy nějaké možnosti konfigurace – stanovení kořenového adresáře, z něhož je přístupný obsah poskytovaný on-line, a dále konfigurace pro každý jeho podadresář individuálně, například jaký soubor zpracovat implicitně, obsahuje-li URL pouze daný adresář, nebo v jaké časové zóně se nachází či jaké podporuje jazyky a přípony souborů.

Podporuje-li webserver dynamický obsah, je součástí i nastavení interpreterů skriptovacích jazyků, jež tento obsah zpracovávají. Součástí architektury sofistikovanějších webových serverů mohou být různé zásuvné moduly a pokročilé metody řízení požadavků.

Webový server má v zásadě dvě možnosti, jak získávat informace, které vrací klientům:

- jsou to buď předem připravené datové soubory (HTML stránky), které webový server bez změny poskytne klientovi (tzv. statický obsah)
- teprve na základě požadavku klienta jsou data shromážděna (přečtena ze souboru, databáze, nebo nějakého koncového zařízení), zformátována a připravena k prezentaci ve formátu HTML a poskytnuta webovému prohlížeči (tzv. dynamický obsah)

K dynamickému vytváření obsahu se používá celá řada různých technologií (Perl, PHP, ASP, ASP.NET, JSP, Python apod.). Statický obsah je schopný server poskytnout výrazně rychleji než dynamický. Na druhé straně pomocí dynamického obsahu lze poskytovat mnohem větší obsah informací a lze reagovat i na různé „ad hoc“ dotazy klientů. Proto se v praxi v mnoha případech oba způsoby poskytování obsahu kombinují – například cachování, node.js

Jazyk HTML je charakterizován množinou značek (tzv. tagů) a jejich vlastností (atributů).

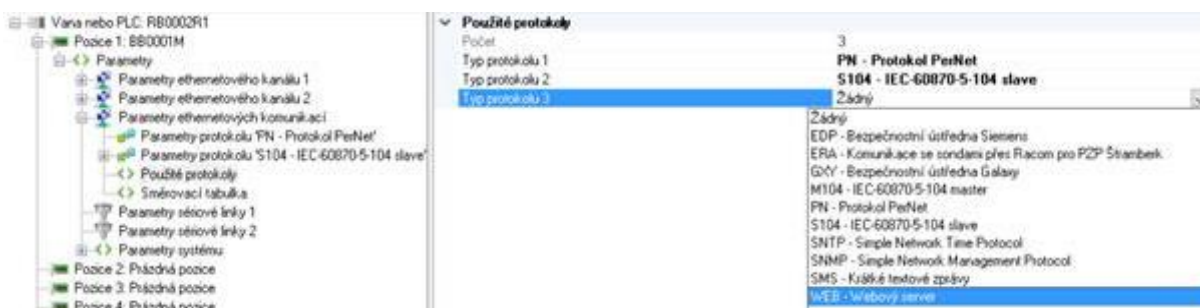
V případě použití ZAT se jedná se o ovladač, který vytvoří na PLC webový server, který zprostředkovává komunikaci mezi HTML aplikací a PLC automatem. Grafická stránka webové aplikace se vytvoří v programu Grafik. Výsledný projekt aplikace Grafik je importován do konfigurace a při nahrávání projektu odeslán do automatu.

2.2 Důvody implementace WEB serveru

Implementace webové aplikace na server, který v tomto případě obstarává řídicí automat, má nespornou výhodu v tom, že je dostupná odkudkoliv. Jednoduchost uživatelského přístupu z toho dělá pohodlný nástroj pro vzdálený dohled a řízení. Výhoda této implementace je ve zkreslování jedné aplikace, ke které se můžete připojit z více vzdálených míst.

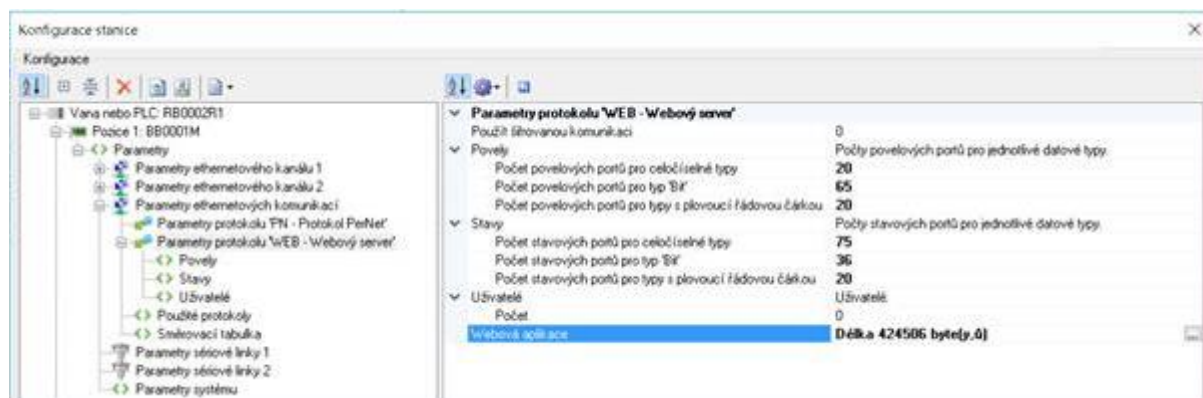
2.3 Postup implementace do ŘS Sandra Z210

V konfiguratorem jsem si zvolil ovladač tak že, v parametrech ethernetových komunikací jsem založil nový protokol a vybral WEB – Webový server



Obrázek 14 - Konfiguratorem WEB – přidání protokolu

V parametrech protokolu WEB jsem si nadefinoval počty proměnných pro povelů a stavy. Povelů jsou vstupní proměnné, které zajišťují příjem hodnot z prohlížeče. Stavy odesílají hodnoty prohlížeči. Měl jsem v možnost zvolit ze tří základních typů (od int8 až po Uint32). Při konfiguraci serveru stačilo zvolit pouze typ. Exportovaný projekt z programu Grafik se importuje na řádku **Webová aplikace...**



Obrázek 15 - Příklad konfigurace WEB serveru s nastavenými počty proměnných

Možnost optimalizace

Pojem optimalizace vede k vyšší efektivitě nebo ke snížení nároků výpočetního systému. Výpočetním systémem může být počítačový program, který je v PLC, komplexní řešení určitého problému a podobně. Například program může být optimalizován tak, aby pracoval rychleji, potřeboval pro provedení výpočtu méně operační paměti, méně systémových prostředků, případně se rychleji spustil a podobně. U optimalizace záleží na rychlosti provádění kódu, na velikosti výsledného kódu a na paměťové náročnosti.

3.1 Aplikované optimalizace

Doplnění pohledu pro hardwarové IO karty. Diagnostika PLC tak může být provedena prostřednictvím webového serveru. Výhodou je zobrazení informací na jednom místě a jejich grafická přehlednost. Není tak nutné ověřovat stav stavových LED jednotlivých jednotek, kde jsou navíc často, vzhledem k fyzickým rozměrům, zkracovány informační texty.

Zpracování OPEN VPN – OpenVPN je volně dostupný software, který dokáže vytvořit šifrovaný VPN tunel mezi hostitelskými stanicemi. S využitím architektury klient-server je schopný zajistit přímé spojení mezi počítači za NATem a to bez potřeby NAT jakkoli konfigurovat. OpenVPN umožňuje ověřit navazované spojení pomocí sdíleného klíče (anglicky pre-shared key), digitálního certifikátu nebo uživatelského jména a hesla. OpenVPN využívá knihovny OpenSSL pro zašifrování dat a řídicích kanálů i pro šifrování a autentizaci s možností využít všechny šifrovací algoritmy obsažené v OpenSSL. Další možností zabezpečení přenášených dat je použití HMAC (označováno jako HMAC Firewall).

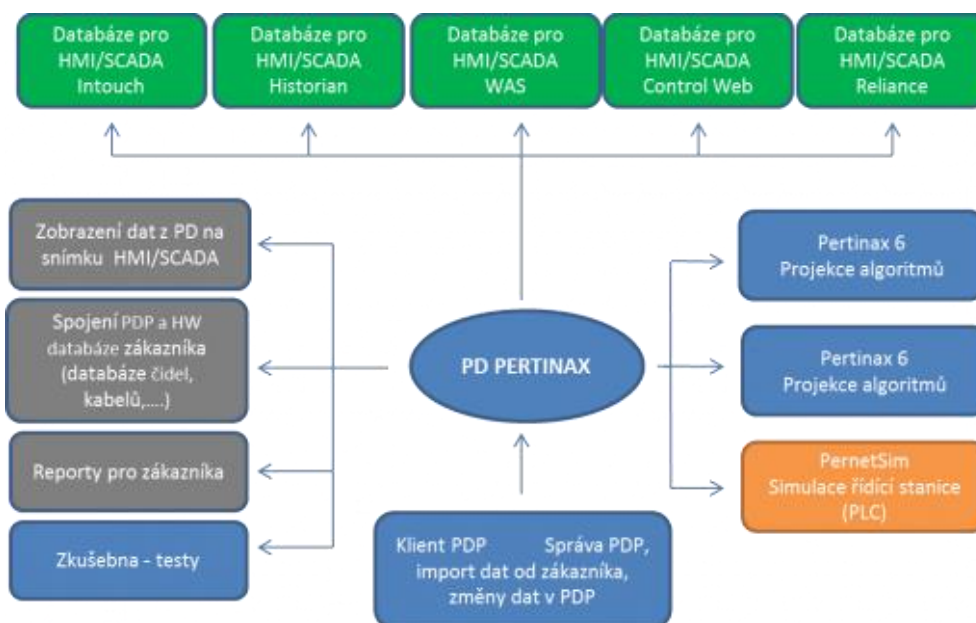
Výhodou je vytvoření zabezpečeného vzdáleného spojení pomocí virtuální privátní sítě, a to bez použití jakéhokoliv dalšího podpůrného hardware, což přináší nižší nároky na místo v rozvaděči a také zajímavou cenovou úsporu.

Vzdálené zabezpečené přehrávání aplikace PLC prostřednictvím webserveru Toto řešení přináší dvě hlavní výhody. Tou první je, že pro přehrávání software nejsou nutné speciální technické prostředky (počítač vybavený vývojovým software), druhou výhodou je, že přehrávání je možností přehrávání aplikačního software vzdáleně, tedy bez nutnosti fyzické přítomnosti u stanice (PLC).

3.2 Návrhy ke zlepšení

V současné době je již do Pertinax6 implementována vazba na projektovou databázi PDP, která slouží jako centrální místo pro centrální správu proměnných, které jsou jednak přímo spjaty s hardwarovými IO signály, ale také se signály komunikovanými.

Projektová databáze Pertinax (také Projekční databáze Pertinax, dále PD Pertinax) je databázový systém vyvinutý a používaný pro účely správy a archivace údajů o proměnných v oboru řídicích algoritmů používaných v řídicích stanicích (PLC) v řídicích systémech dodávaných společností ZAT. Systém PD Pertinax je vybudován při použití platformy MS SQL Server a využívá způsob přístupu klient/server. Jednou z hlavních idejí sledovaných při vývoji PD Pertinax byla myšlenka možného propojení PD Pertinax se systémy HMI/SCADA používanými ve společnosti ZAT, kterými jsou především produkty firmy Wonderware. Proto byly do PD Pertinax postupně zavedeny nejprve databázové objekty umožňující ukládat údaje v podobách přístupných systémům InTouch, Historian a WAS. Posléze byly přidány také objekty pro další systémy Control Web a Reliance (obr.1). Tímto způsobem lze kompletně popsat nejen všechny vlastnosti vlastnosti zobrazované proměnné v daném systému HMI/SCADA (název proměnné, komentář proměnné, výstražné meze, inicializační hodnoty, minimální a maximální hodnoty atd.), ale také její komunikační návaznosti směrem k proměnným použitým v algoritmech řídicích stanic (ItemName, AccessName, ServerName atd.). [7]



Obrázek 16 - Schéma propojení databáze projektu [8]

Při realizovaném těsném propojení nástrojů pro vypracování návrhu aplikačních programů pro řídicí stanice (PLC) s používanými systémy HMI/SCADA – Webserver je dosahováno úplné konzistence dat v celém projektu řídicího systému. Ta přináší zefektivnění práce a eliminaci chyb nejen při základním návrhu a realizaci řídicího systému, ale zejména při jeho údržbě a následných úpravách.

Responzivní web design je způsob stylování HTML dokumentu, které zaručí, že zobrazení stránky bude optimalizováno pro všechny druhy nejrůznějších zařízení (mobily, notebooky, netbooky, tablety atd.). Především díky vlastnosti Media Queries, která je zahrnuta ve specifikaci CSS3, lze rozpoznat vlastnosti zařízení, na kterém je stránka prohlížena a přizpůsobit tak samotnou stránku a její obsah. Responzivní web design má tři základní úrovně: Flexibilní struktura, Flexibilní obrázky a Media Queries. [10]

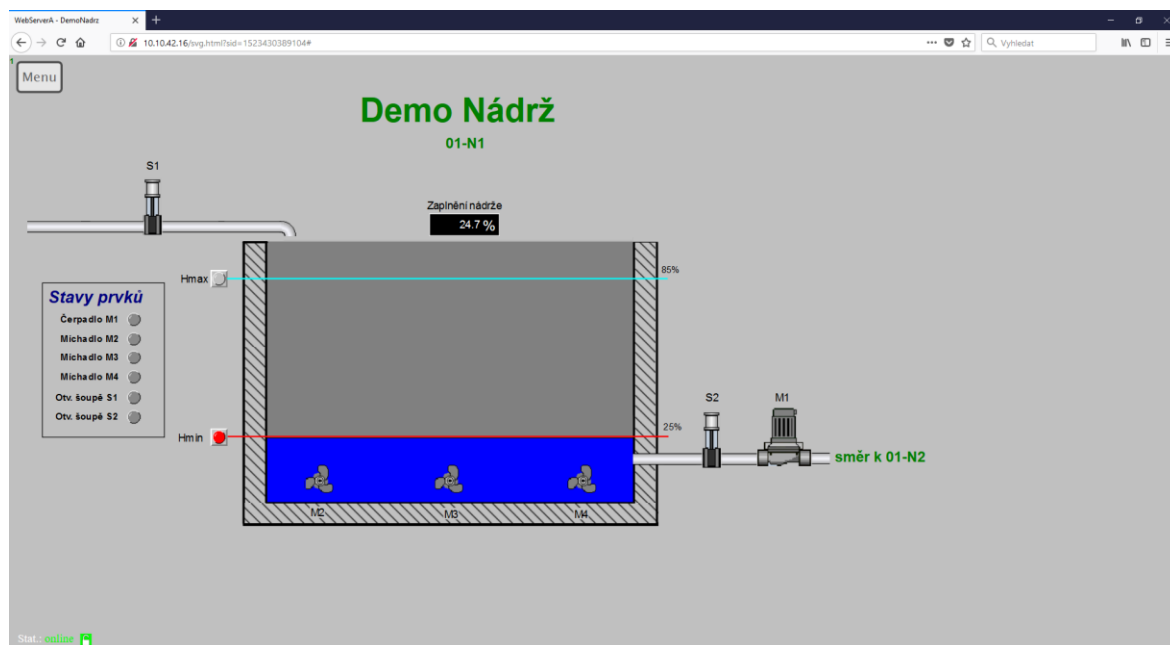
Flexibilní struktury se dosahuje pomocí procentních šířek. Jednotlivé šířky elementů tak nejsou zadávány v pixelech, nýbrž v procentech. Takto připravená struktura pak reaguje na různé šířky nejrůznějších zařízení. Technika flexibilních obrázků zajistí, že obrázky se budou přizpůsobovat stejně tak, jako samotná struktura. Aby tohoto bylo dosaženo, neuvádí se šířka a výška obrázku uvnitř tagu. Aby nedocházelo k přetékání, nastylují se všechny obrázky. Media Queries se považují za poslední úroveň responzivního web designu. Jsou to pravidla, díky kterým lze měnit stylování dokumentu v závislosti na šířce obrazovky zobrazovaného zařízení. Následující stylování (červené pozadí celého dokumentu) bude uplatněno pouze tehdy, pokud šířka prohlížeče na použitém zařízení bude v rozsahu od 660px do 780px. Jako přínosy bych uvedl jednotou aplikaci pro různá zobrazovací zařízení.

Ukázka aplikace web serveru do ŘS Sandra

Aplikace je určena jako webové HMI rozhraní pro jednoduchou monitorovací a ovládací technologii kde používám řídicí systém společnosti ZAT a.s. Lze použít klienty mobilních zařízení, průmyslové panely nebo jakýkoliv prohlížeč dle specifikace HTML5.

Připojení k aplikaci může proběhnout několika způsoby. Avšak jedno mají všechny společné. A to že se musíte připojit do sítě automatu. To může proběhnout vzdáleným připojením přes VPN nebo na místě přes Wi-Fi či kabel.

4.1 Přehled obrazovek

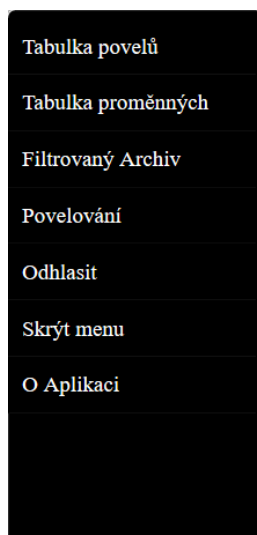


Obrázek 17 - Přehled technologie

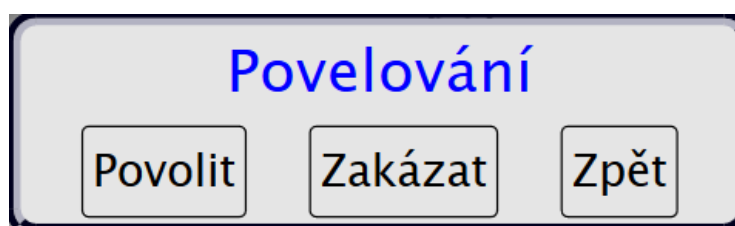
Obrazovka poskytuje informace o stavu naplnění nádrže a její jednotlivých částí. Graficky jsou znázorněny stavy šoupat a motorů pro regulaci hladiny. Vizuální informace je doplněna i analogovou hodnotou. Manipulace s nádrží je umožněna pouze po povolení povelování.

Stav nádrže – napuštěno, vypuštěno, mezi Stav

Probíhající proces – napouštění, vypouštění.

**Obrázek 18 - Lišta menu**

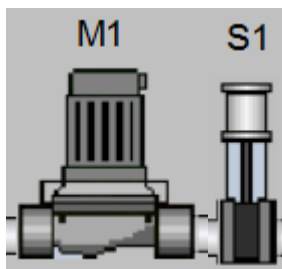
Po stisknutí tlačítka menu se vyvolá v pravé části přehledové menu v kterém je možnost otevření několika další oken jakou je například tabulka povelů, povelování, informace o aplikaci atd..

**Obrázek 19 - Povelovací okno**

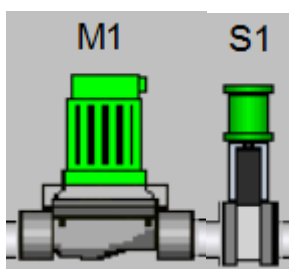
Přes tlačítko menu a můžeme dostat do povolení povelování, které je potřebné k ovládání jednotlivých prvků.

**Obrázek 20 - Ovládací okno**

Po stisknutí jednotlivého prvku se vyvolá ovládací okno s možností otevření, zavření v případě šoupěte a v případě motoru je zde zapnout či vypnout.

**Obrázek 21 - Vypnutý prvek**

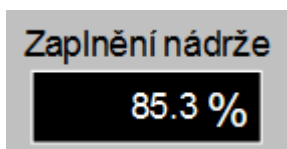
Na obrazovce je šedivou barvou znázorněn prvek vypnutý nebo zavřené šoupě

**Obrázek 22 - Zapnutý prvek**

Na obrazovce je zelenou barvou znázorněn prvek zapnutý nebo otevřené šoupě

**Obrázek 23 - Přehled stavů**

Na úvodní obrazovku jsem i umístil přehled jednotlivých stavů prvků. Zelená barva značí prvek zapnut či otevřen. Šedá barva značí prvek vypnut či zavřen.



Obrázek 24 - Zaplnění nádrže

Nad nádrží je údaj o zaplnění nádrže, kde jsem nastavil maximální hodnotu na 85 % a minimální hodnotu 25%.



Obrázek 25 - Kontrolka dosažení hladiny

Po dosažení horní či dolní hranice hladiny dojde k pročervenání kontrolky stavu hladiny

Zobraz záznamů 5

Hledat/Filtrovat:

č.	Prvek	KKS	Typ	Komentář	Min	Max	Hodnota	Item
0	BinPovel	NA_M3_ZAP	B	Povel zapni M3	0	1	0	B.57
1	BinPovel	NA_M4_ZAP	B	Povel zapni M4	0	1	0	B.59
2	BinPovel	NA_S1_OTV	B	Povel otevři S1	0	1	0	B.49
3	BinPovel	NA_S2_OTV	B	Povel otevři S2	0	1	0	B.51
4	BinPovel	NA_M1_ZAP	B	Povel zapni M1	0	1	0	B.53

Zobrazují 1 až 5 z celkem 6 záznamů

Předchozí Další

Zpět

Obrázek 26 - Seznam povelů

Přes tlačítko v menu „seznam povelů“ se nám otevře tabulka ve které vidíme aktuální stavy vy komunikovaných signálů.

Zobraz záznamů 5

Hledat/Filtrovat:

Prvek	KKS	Typ	Komentář	Min	Max	Hodnota	Item
Bargraf	NA_Zapl_01-N1	R	Stavv zaplnění nádrže 01-N1	0	100	22.96798706	R.16
Bitmapa	NA_Stav_S1	I	Stavv soupete N1	0	2	0	I.65
Bitmapa	NA_Stav_S2	I	Stavv soupete S2	0	2	0	I.66
Bitmapa	NA_Stav_M1	I	Stavv pohonu M1	0	2	0	I.67
Bitmapa	NA_Stav_Hmax	I	Stavv pohonu Hmax	0	2	0	I.73

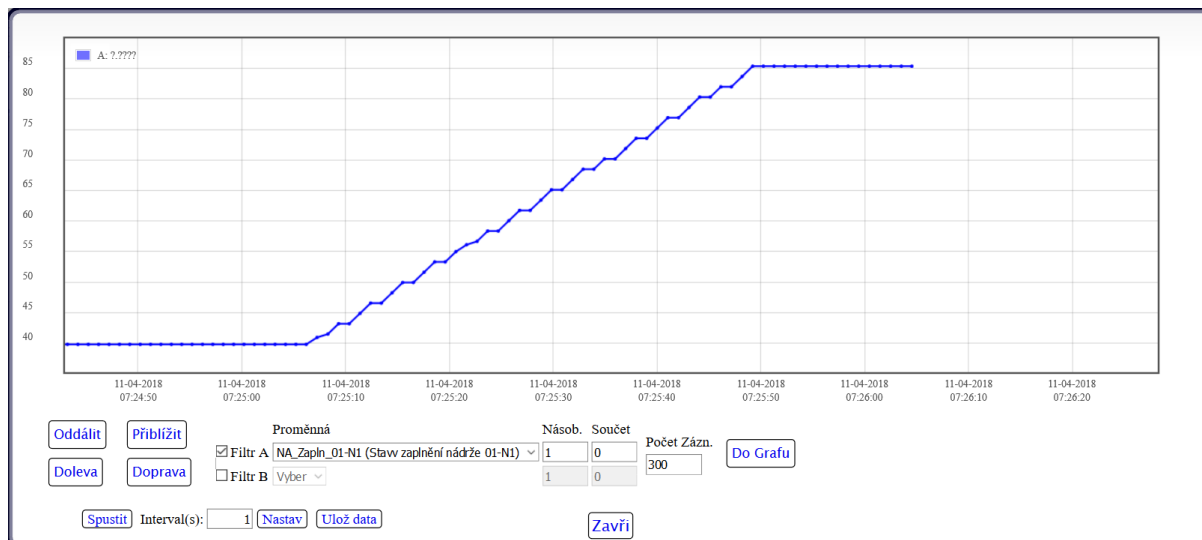
Zobrazují 1 až 5 z celkem 24 záznamů

Předchozí Další

Ulož

Obrázek 27 - Seznam proměnných

Přes tlačítko v menu „seznam proměnných“ se nám otevře tabulka ve které vidíme aktuální stavy proměnných stavů



Obrázek 28 - Graf změny hladiny

Na grafu je znázorněno plnění nádrže vodou v závislosti na čase.

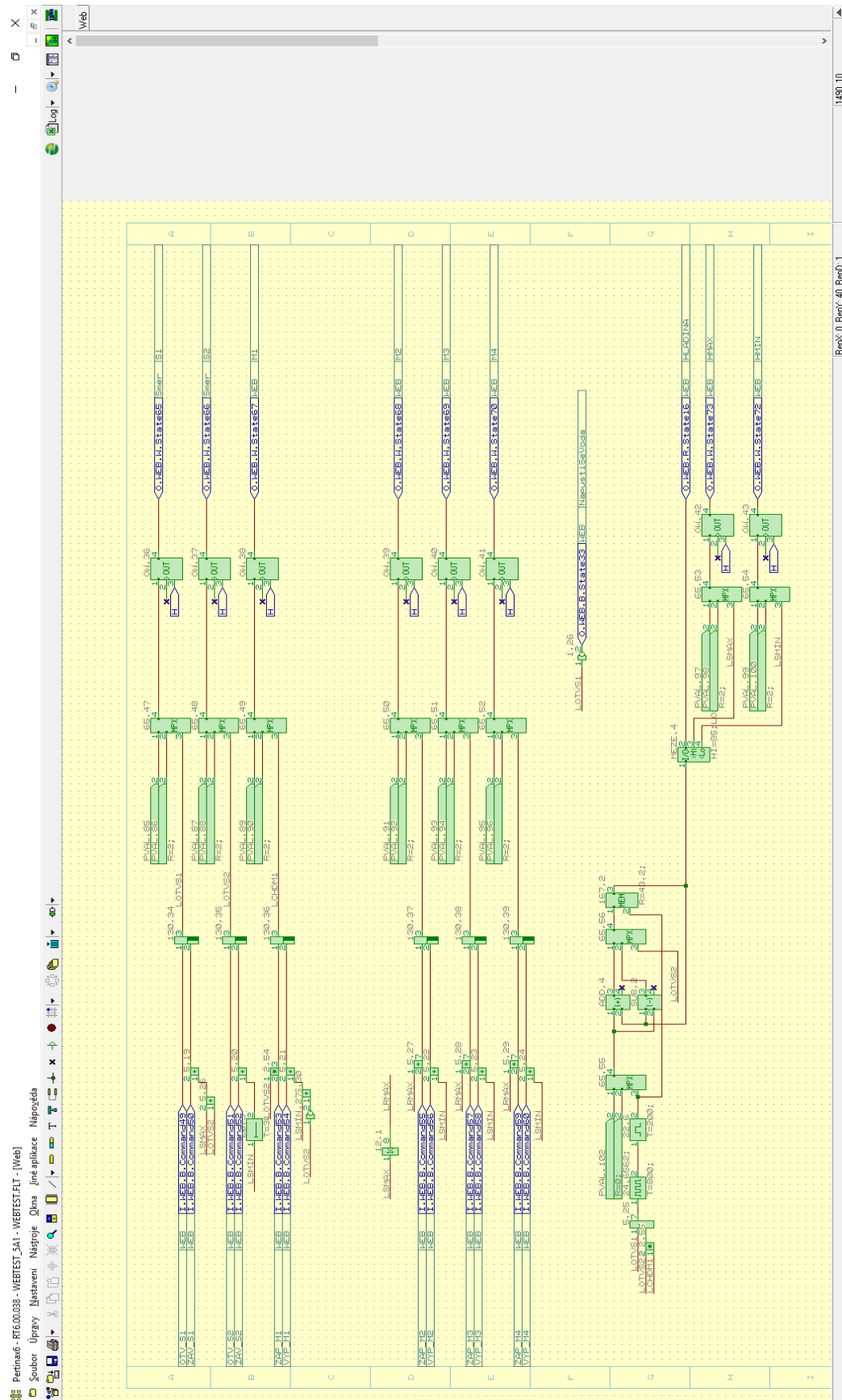
4.2 Popis funkčnosti plnění vodní nádrže

Při provozu vodní nádrže mohou nastat dva stavy.

Jedním ze stavu je napouštění, při kterém musí být splněny následující podmínky. Nádrž není naplněna k horní hranici maxima 85 % objemu nádrže. Nedochozí k vypouštění nádrže, při kterém by bylo otevřeno šoupě S2 a zapnuté odvodní čerpadlo M1. Po kliknutí na šoupě S1 se otevře ovládací okno, ve kterém zadáme povel kliknutím na tlačítko zapnout, povel na otevření se otevře šoupě S1 a začne se nádrž plnit. Vypnutí napouštění může nastat ve dvou případech buď dojde k dosažení horní hranice a je poslán povel na uzavření šoupěte S1 a tím pádem se zapnou automaticky i míchadla M2, M3, M4 anebo je možnost při napouštění kliknout ještě jednou na šoupe a zadat povel na zavření. Aktuální hodnota napouštění je zobrazena nad obrázkem nádrže.

Druhým stavem je vypouštění, při kterém musí být plněny následující podmínky. Nádrž není vypuštěna k dolní hranici minima 25 % objemu nádrže. Nedochozí k napouštění nádrže, při kterém je otevřeno šoupě S1. Pro zapnutí vypouštění nádrže se musí kliknout na šoupě S2 a zadat povel na otevření. Po otevření šoupěte S2 se musí kliknout na odvodní čerpadlo M1 a zadat povel na zapnutí. Po splnění těchto podmínek dojde k vypouštění nádrže, které se dá ukončit vypnutím čerpadla anebo automatickým vypnutím od polohy dolní hladiny vody při kterém dojde k automatickému zavření šoupěte S2 a vypnutí čerpadla M1.

4.3 Ukázka programu



Obrázek 29 - Algoritmus kreslený pomocí funkčních bloků

Závěr

Cílem této bakalářské práce bylo seznámit se s integrací webového serveru do prostředků ZAT. Konkrétně do automatu ZAT SandRA Z210 včetně bezdrátového připojení.

První kapitola práce představuje celkové portfolio výrobků společnosti ZAT a.s. Od prvních automatů řady první generace, kde se jednalo spíše o reléový systém než o automat v pravém slova smyslu. Výraznější zlom nastal ve druhé generaci s příchodem mikroprocesorů do automatizační techniky. S příchodem třetí generace řídicích systémů přišel automat ZAT-D, který dokázal přejít z 8bitových procesorů na výkonnou stanicí s 32bitovými procesory. Tento systém byl v hojné míře nasazován na velkou řadu elektráren a tepláren nejen v ČR a některých provozech slouží ještě nyní. V současnosti se nasazuje čtvrtá generace řídicích systémů SandRA, která zahrnuje hardwarové prostředky i softwarové nástroje pro realizaci komplexního řešení řídicího systému třídy DCS, u které je možnost využití webového serveru pro vzdálený dohled nad technologií. V druhé polovině této kapitoly jsem se zaměřil na softwarové prostředky, pomocí kterých jsem docílil vytvoření aplikace. Jedním z nich byl program Grafik, ze kterého nám vzniká grafický podklad pro vykreslování hotové aplikace. Jako druhý nám posloužil Pertinax6, ve kterém vznikly všechny algoritmy chování konečné aplikace. Jsou zde popsány postupy práce jednotlivými programy, včetně importů a exportu hlavních částí.

Druhá kapitola představuje pojem webový server. Jsou zde popsána místa, na kterých se můžeme s webovým serverem setkat. Jsou zde také uvedeny jeho výhody a nevýhody. Z toho vyplývá také důvod, proč se webové servery implementují do dalších zařízeních. Dále jsme si přiblížili postup, kterým byl implementován webový sever do automatu ZAT SandRA Z210.

Třetí kapitola je zaměřená na možnosti optimalizace, která je rozdělena na dvě základní části. Do první části optimalizace jsem zařadil přizpůsobení a optimalizaci stávající aplikace a do druhé možnosti optimalizace s nutným časovým prostorem. Je nutné neustále vyvíjet a zdokonalovat prostředky, se kterými pracujeme.

V praktické části mé bakalářské práce jsem Vám představil jednoduchou aplikaci pro ovládání hladiny nádrže, na které lze demonstrovat využití všech výše uvedených prostředků.

Seznam literatury a informačních zdrojů

- [1] ZAT a.s. *Katalog výrobků: Z210* [online]. Příbram VI: ZAT a.s, 2015 [cit. 2018-05-28]. Dostupné z: www.zat.cz
- [2] SandRA Z210. In: *Katalog výrobků: Z210* [online]. Příbram: ZAT, 2015 [cit. 2018-05-28]. Dostupné z: www.zat.cz
- [3] SandRA Z210. In: *Katalog výrobků: Z200* [online]. Příbram: ZAT, 2015 [cit. 2018-05-28]. Dostupné z: www.zat.cz
- [4] ZAT a.s. SM72 - Řízení návrhu - aplikační SW. 2008
- [5] ZAT a.s. PERTINAX6 - UŽIVATELSKÁ PŘÍRUČKA. 2013
- [6] ZAT a.s. In: *Pertinax6:* [online]. Příbram: ZAT, 2018 [cit. 2018-05-28]. Dostupné z: www.zat.cz
- [7] ZAT a.s. In: *Projektová databáze Pertinax:* [online]. Příbram: ZAT, 2018 [cit. 2018-05-28]. Dostupné z: www.zat.cz
- [8] ZAT a.s. In: *Projektová databáze Pertinax* [online]. Příbram: ZAT, 2018 [cit. 2018-05-28]. Dostupné z: www.zat.cz
- [9] Webový server. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2018, 14.5.2018 [cit. 2018-05-28]. Dostupné z: www.wikipedia.org
- [10] Responzivní web design. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 3.4.2018 [cit. 2018-05-28]. Dostupné z: <https://cs.wikipedia.org>

Seznam obrázků

OBRÁZEK 1 - ČASOVÁ OSA HISTORIE PROSTŘEDKŮ FIRMY ZAT.....	11
OBRÁZEK 2 - PLC ZAT SANDRA Z200 [3].....	13
OBRÁZEK 3 - PLC ZAT SANDRA Z210 [2].....	14
OBRÁZEK 4 – UKÁZKA PROGRAMU PERTINAX6	16
OBRÁZEK 5 – OKNO ANALYZÁTORU	17
OBRÁZEK 6 – OKNO SPRÁVCE PROJEKTU	18
OBRÁZEK 7 – OKNO VÝVOJOVÉHO PROSTŘEDÍ GRAFIK.....	20
OBRÁZEK 8 – NASTAVENÍ ATRIBUTŮ OBJEKTŮ	21
OBRÁZEK 9 – NASTAVENÍ ADRESY POVELU	21
OBRÁZEK 10 – VLASTNOSTI BINÁRNÍHO POVELU	22
OBRÁZEK 11 - DEFINICE PROMĚNNÉ	22
OBRÁZEK 12 - DEFINICE MATEMATICKÉ FUNKCE.....	23
OBRÁZEK 13 - KOMUNIKACE MEZI KLIENTEM A WEBOVÝM SERVEREM [9].....	24
OBRÁZEK 14 - KONFIGURÁTOR WEB – PŘIDÁNÍ PROTOKOLU.....	26
OBRÁZEK 15 - PŘÍKLAD KONFIGURACE WEB SERVERU S NASTAVENÝMI POČTY PROMĚNNÝCH.....	26
OBRÁZEK 16 - SCHÉMA PROPOJENÍ DATABÁZE PROJEKTU [8]	28
OBRÁZEK 17 - PŘEHLED TECHNOLOGIE.....	30
OBRÁZEK 18 - LIŠTA MENU.....	31
OBRÁZEK 19 - POVELOVACÍ OKNO	31
OBRÁZEK 20 - OVLÁDACÍ OKNO	31
OBRÁZEK 21 - VYPNUTÝ PRVEK.....	32
OBRÁZEK 22 - ZAPNUTÝ PRVEK	32
OBRÁZEK 23 - PŘEHLED STAVŮ	32
OBRÁZEK 24 - ZAPLNĚNÍ NÁDRŽE	33
OBRÁZEK 25 - KONTROLKA DOSAŽENÍ HLADINY	33
OBRÁZEK 26 - SEZNAM POVELŮ.....	33
OBRÁZEK 27 - SEZNAM PROMĚNNÝCH	33
OBRÁZEK 28 - GRAF ZMĚNY HLADINY	34
OBRÁZEK 29 - ALGORITMUS KRESLENÝ POMOCÍ FUNKČNÍCH BLOKŮ.....	35

