VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PLC SCADA systém pro řízení větrného tunelu

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Tomáš Trenčan
AUTHOR

VEDOUCÍ PRÁCE — Ing. Jakub Arm
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Tomáš Trenčan ***ID:*** 154897

Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

PLC SCADA systém pro řízení větrného tunelu

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte rešerši SCADA systémů pro PLC firmy WAGO podporujících Modbus TCP.
2) Seznamte se s vývojovým prostředím e!COCKPIT firmy WAGO - funkce Modbus TCP server a ovládání řízené osy (PLCopen).
3) Proveďte návrh regulátoru pro model větrného tunelu na základě identifikace.
4) Vytvořte programy pro PLC, které budou provádět dvě úlohy zároveň (ovládání řízené osy a regulace větrného tunelu).
5) Vytvořte společnou vizualizaci obou procesů, která monitoruje a nastavuje důležité parametry, ve zvoleném SCADA software na základě předchozí rešerše.

DOPORUČENÁ LITERATURA:

Martinásková M., L. Šmejkal. Řízení programovatelnými automaty, vydavatelství ČVUT, Praha, 2004.

Termín zadání: 6.2.2017 ***Termín odevzdání:*** 29.5.2017

Vedoucí práce: Ing. Jakub Arm

Konzultant: Jakub Svoboda

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

Abstrakt

Předložená bakalářská práce se věnuje programovatelnému automatu (PLC) firmy WAGO a SCADA systémům. Skládá se ze dvou části, z teoretické části a z praktické části. V teoretické části jsou na zakladě poznatků z odborné literatury vymezeny pojmy ohledně SCADA systémů, PLC od firmy WAGO a vývojového prostředí e!COCKPIT firmy WAGO a rešerše volně dostupných SCADA systémů podporujících Modbus TCP. Na základě této rešerše byl vybrán SCADA systém pro vizualizaci dvou demonstračníc úloh v praktické části. Praktická část práce obsahuje PLC programy vytvořené ve vývojovém prostředí e!COCKPIT pro dvě demonstrační úlohy, větrný tunel a řízená osa a jejich společnou vizualizaci.

Klíčová slova

PLC, SCADA, WAGO, e!COCKPIT

Abstract

This work focuses on programmable logic controllers (PLC) made by WAGO and SCADA systems. It consists of two parts, theoretical part and a practical part. In theoretical part there are terms which explains SCADA systems, WAGO PLC and programming software e!COCKPIT made by WAGO and searches of free SCADA systems supporting communication through Modbus TCP. One of the SCADA systems was chosen as a visualization software for two task in practical part. Practical part consists of PLC codes made in programming software e!COCKPIT for two tasks, wind tunnel and controlled axis and their visualization.

Keywords

PLC, SCADA, WAGO, e!COCKPIT

Bibliografická citace:

TRENČAN, T. *PLC SCADA systém pro řízení větrného tunelu*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 50 s. Vedoucí bakalářské práce Ing. Jakub Arm.

Prohlášení

„Prohlašuji, že svou závěrečnou práci na téma PLC SCADA systém pro řízení větrného tunelu 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 uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **29. května 2017** .....

podpis autora

Obsah

1 Úvod ... 10
2 Rešerše SCADA systémů podporujících Modbus TCP ... 11
2.1 SCADA Laquis 4.1 ... 11
2.2 IGSS ... 12
2.3 WinTr ... 14
2.4 Promotic PmFree ... 15
2.5 FreeScada2 ... 17
2.6 AdvancedHMI v3.99s ... 17
2.7 WinLog Lite 3 ... 20
2.8 Placené SCADA Systémy ... 21
2.8.1 Control Web ... 21
2.8.2 InTouch ... 22
2.8.3 WinCC ... 22
2.9 Zhodnocení ... 23
3 PFC200 CS 2ETH RS ... 24
3.1 Základní popis ... 24
4 E!COCKPIT ... 27
4.1 Konfigurace ... 27
4.2 Programování ... 27
4.3 Vizualizace ... 28
5 Řízení větrného tunelu ... 29
5.1 Popis modelu ... 29
5.2 Identifikace soustavy ... 30
5.3 Návrh regulátoru ... 31
5.4 PLC program ... 32
5.5 Implementace regulátoru ... 33
6 Řízení osy ... 35
6.1 Popis modelu ... 35
6.2 Identifikace soustavy ... 36
6.3 Návrh regulátoru ... 37
6.3.1 Filtr žádané hodnoty ... 39
6.4 PLC program ... 40
6.5 Implementace regulátoru ... 41

7 Vizualizace 45
8 Závěr 47

Seznam obrázků

Obrázek 1: Uživatelské prostředí SCADA Laquis 12
Obrázek 2: Grafický editor IGSS 13
Obrázek 3: Výsledná vizualizace ve WinTr Runtime 15
Obrázek 4: Uživatelské prostředí Promotic 16
Obrázek 5: Uživatelské prostředí AdvancedHMI 19
Obrázek 6: Komunikační blok AdvancedHMI ve Visual Studiu 19
Obrázek 7: Uživatelské prostředí WinlogLite3 21
Obrázek 8: Kontrolér PFC200 CS 2ETH RS [5] 24
Obrázek 9: Komponenty kontroléru PFC200 [5] 25
Obrázek 10 Blokové schéma zapojení modelu větrného tunelu 29
Obrázek 11 Skoková změna žádané hodnoty otáček větráku v čase 30
Obrázek 12 Regulační schéma modelu s větrným tunelem 31
Obrázek 13Teoretický graf závislosti vyregulování žádané hodnoty v čase z matematického modelu 32
Obrázek 14 Ukázka kódu - měření otáček větráku 32
Obrázek 15 Nastavený blok PID regulátoru v programu 33
Obrázek 16 Změřená závislost vyregulování žádané hodnoty otáček větráku v čase 34
Obrázek 17 Blokové schéma zapojení modelu řízené osy 35
Obrázek 18 Změřená závislost skokové změny žádané hodnoty polohy v čase 36
Obrázek 19 Regulační schéma modelu řízené osy 37
Obrázek 20 Teoretický graf závislosti rychlosti motoru na čase při regulaci polohy z matematického modelu 38
Obrázek 21 Teoretický graf závislosti vyregulování polohy v čase z matematického modelu .. 38
Obrázek 22 Ukázka kódu - Změna směru a filtrování vstupní hodnoty PS regulátoru 40
Obrázek 23 Ukázka kódu - Měření rychlosti z polohy 41
Obrázek 24 Nastavený blok PD regulátoru v programu 41
Obrázek 25 Nastavený blok PID regulátoru v programu 42
Obrázek 26 Závislost žádané vstupní hodnoty PS regulátoru na čase při regulaci polohy 43
Obrázek 27 Změřená závislost rychlosti na čase při regulaci polohy 43
Obrázek 28 Změřená závislost regulace polohy 44
Obrázek 29 Společná vizualizace pro obě úlohy 45

Seznam tabulek

Tabulka 1: Srovnávací tabulka SCADA systémů 23
Tabulka 2: Popis komponentů kontroléru PFC200[5] 26
Tabulka 3 Použité vstupy a výstupy PLC v modelu větrného tunelu 30
Tabulka 4 Použité vstupy a výstupy PLC v modelu řízené osy 36
Tabulka 5 Proměnné posílané z PLC do SCADA systému přes Modbus TCP 45

1 ÚVOD

Tato bakalářská práce se věnuje přehledu volně dostupných SCADA systémů podporujících komunikaci přes Modbus TCP a jejich využití v praxi. Dané téma bylo vybráno, neboť se autor práce o PLC již delší dobu zajímá a výběr správného SCADA systému pro PLC program může rozhodnout o výsledku projektu.

Na začátku je důležité obecné seznámení s PLC, prozkoumání jednotlivých hardwarových částí, vstupních a výstupních karet a jejich funkcí a možnosti jeho programování. Také bylo nutné seznámit se s připravenými modely a upravit si je tak, aby splňovaly zadání úloh.

Prvním z cílů práce je seznámení se s PLC PFC200 CS 2ETH RS a vývojovým prostředím e!COCKPIT firmy WAGO, které bude použito pro vytvoření PLC programu pro model větrného tunelu a model řízené osy.

Druhým cílem je rešerše volně dostupných SCADA systémů podporujících Modbus TCP z pohledu uživatele-programátora. Výsledkem této rešerše bude jejich srovnání dle předem daných kritérií - zprovoznění SCADA systému, navázání komunikace PLC přes Modbus TCP, robustnosti komunikace, možnosti grafických objektů, možnosti a obtížnosti psaní scriptu, dostupnosti návodů a tutoriálů. Nejlepší SCADA systém bude použit pro společnou vizualizaci demonstračních úloh.

Třetím cílem je návrh regulátorů pro model větrného tunelu a model řízené osy. Po změření soustav bude navržen regulátor otáček pro model větrného tunelu a regulátory polohy a rychlosti pro model řízené osy. Požadavek na návrh regulátoru otáček je co nejrychlejší vyregulování žádané hodnoty. Požadavek na regulátory pro model řízené osy je pozvolný rozjezd a pozvolný dojezd. Druhým požadavkem je co nejrychlejší dosažení hodnoty žádané polohy. Tyto regulátory budou implementovány do PLC programu.

Čtvrtým cílem je napsání PLC programu pro obě úlohy ve vývojovém prostředí e!COCKPIT v libovolném jazyku, kterým dané vývojové prostředí disponuje. Výsledkem programů by měl být větrák, otáčející se rychlostí zadanou uživatelem a jezdec po řízené ose, který se přesune do polohy zadané uživatelem.

Posledním cílem je vytvoření vizualizace ve volně dostupném SCADA systému na základě výsledku rešerše. Tato vizualizace bude společná pro obě úlohy a umožní tak uživateli řídit obě úlohy zároveň na jednom přehledném panelu.

2 REŠERŠE SCADA SYSTÉMŮ PODPORUJÍCÍCH MODBUS TCP

V této kapitole jsou popsány SCADA systémy umožňující komunikaci přes Modbus TCP. Zaměřuji se na volně dostupné(free), open-source a webové formy. Placené systémy jsou zde alespoň zmíněny.

SCADA (Supervisory Control And Data Acquisition), tzn. supervizní řízení a sběr dat. SCADA tedy není plnohodnotným řídicím systémem, ale zaměřuje se spíše na úroveň supervizora. Zpravidla je to software fungující nad skutečným řídicím systémem založeným na PLC (programovatelný logický automat) nebo jiných HW zařízeních. HMI je zkratka pro Human–Machine Interface, tzn. Rozhraní mezi člověkem a strojem. [1]

.

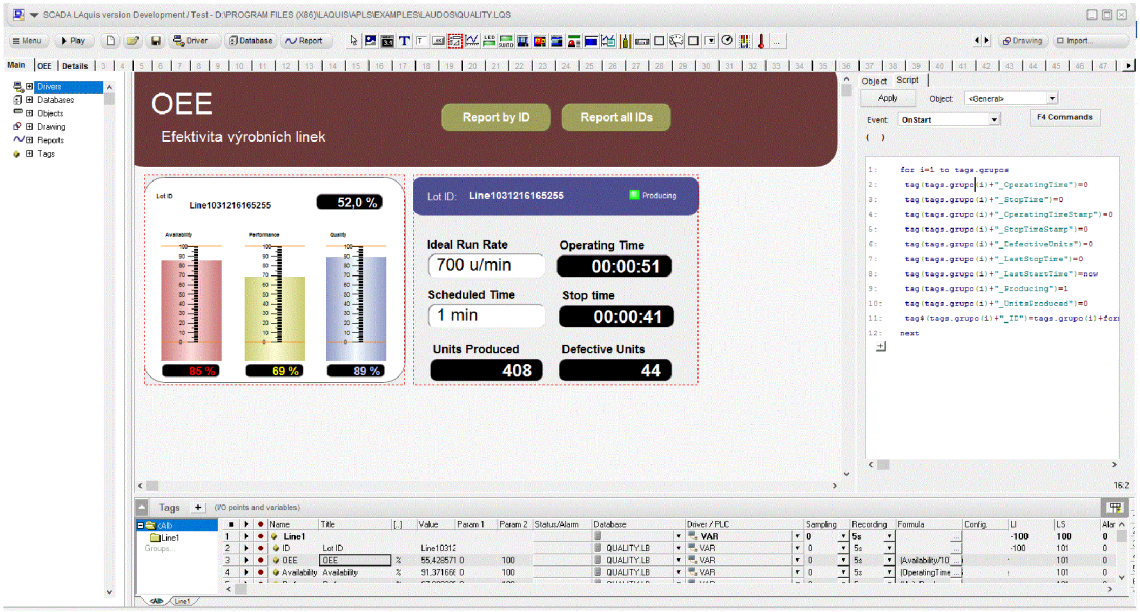
2.1 SCADA Laquis 4.1

Tento systém, vyvinutý brazilskou firmou LCDS je propracovaný avšak velice jednoduchý na ovládání. Díky velmi obsáhlému návodu a instruktážnímu videu je vhodný i pro úplné začátečníky i přesto, že i anglická verze je pořád trochu v portugalštině. Lze jej pořídit pouze na operační systém Windows. Přímo v programu je několik možností pro komunikaci jako Modbus RTU, Modbus TCP, Advantech, Alnet, Omron, DF1 a další. Také je možnost si další drivery přidat přes scripty, externí DLL a OPC. Zprovoznení komunikace přes Modbus TCP je otázkou sekund. Stačí u dané proměnné zadat IP adresu DHCP serveru a číslo TCP portu. Takže lze mít v projektu tagy z několika zařízení přes různé způsoby komunikace. Adresa registru proměnných se zapisuje v hexadecimálním tvaru s prefixem pro datový typ (0x0000 pro BOOL, W40x0000 pro WORD atd.), takže je nutné si adresy registrů, které jsou v e!COCKPIT zadané v decimálním tvaru, převádět. Tagům(proměnným) lze nastavit několik vlastností jako minimumální/maximální hodnota, úprava hodnoty matematickými operacemi nebo změnou o jiné tagy(např. vynásobení jiným tagem atd.), alarmy, které nastanou když proměnná dosáhne určité hodnoty a ukládání do databáze. Tyto proměnné se pak používají v objektech HMI, kde se zároveň vytváří script pro daný objekt. Scriptovací jazyk je jednoduchý jazyk na bázi VB script určený pro SCADA systémy. Scripty se tradičně píší pro jednotlivé akce (OnStart, OnGroup, OnKeyDown atd.) pro jednotlivé objekty. SCADA Laquis umožňuje také importovat vlastní objekty a také je kreslit v grafickém editoru. Každému objektu lze nastavit stav, při kterém nastanou takže se z nich dá vytvořit i slušná řada animací pro reálnější vizualici. SCADA Laquis zaujme také možností si vytvářet 3D animace, což ostatní free SCADA systémy neumí. Vytvořit si automatizovaný model celé továrny zní sice pěkně, avšak vizualizace se stává méně přehlednou, takže si tuto možnost většina uživatelů nezvolí. Grafická kvalita 3D animací také není zrovna oslnivá a vytváření 3D animací je složitější a zdlouhavější

než vytváření klasických animací. Data z tagů lze ukládat a zpracovávat do tabulek a různých grafů. Je zde na výběr několik šablon, jak pro tabulky tak pro grafy, které se dají volně upravovat. Reporty jsou ukládány pouze do vlastního formátu, možnost ukládání dat do excellu zde však chybí a tak si je do něj uživatel musí kopírovat manuálně z tabulek SCADY Laquis.

Ačkoliv SCADA Laquis není úplně bez chyb, stále je to velmi dobrý software na vytváření jednoduchých vizualizací. Vyniká převážně svou jednoduchostí, jak při psaní scriptu, tak při vytváření grafických objektů.

Odkaz na stažení: http://laquisscada.com/index-3.html



Obrázek 1: Uživatelské prostředí SCADA Laquis

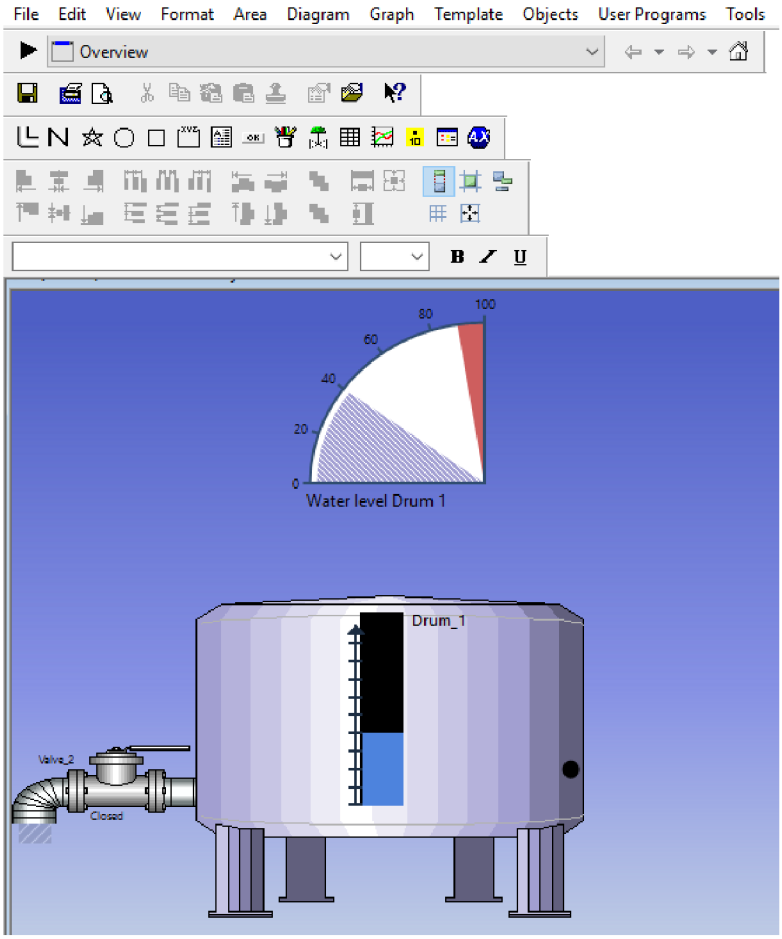
2.2 IGSS

Dánská společnost Shneider Electric už několik let vyvíjí SCADA systém IGSS. Nyní nabízí také verzi IGSS FREE50, která je zdarma a je plně funkční s omezením na 50 tagů. Pro větší projekty to není dostačující, ale pro moje demonstrační úlohy to stačit bude. IGSS se dělí na dvě části, Runtime a Design. V Designu se vytváří, objekty, funkce a celková vizualizace. Runtime složí pro operátory, kteří monitorují a vytváří reporty. Tento systém je pouze pro systém Windows a na tvorbu webových aplikací je třeba mít .NET Framework pro připojení k IGSS serverům. Ihned po vytvoření projektu IGSS nabídne konfigurační tabulku, kde si mimo jiné, pro většinu uživatelů nepodstatné nastavení lze zvolit typ komunikace a také jejich zprovoznění. IGSS podporuje přes 80 komunikačních driverů což je podstatně víc, než ostatní free SCADA systémy. Také je tu možnost si místo nastavení komunikace hodnoty simulovat ve vestavěném PLC simulátoru. Lze si vybrat z mnoha objektů, kromě běžných monitorovacích jako jsou

tlačítka, grafy a tabulky jsou tu i objekty na vizualizaci jako motory, trubky, pumpy a další. Rozhodně si je z čeho vybírat, ale IGSS postrádá vlastní grafický editor, ve kterém by se objekty daly kreslit. Každý objekt má širokou řadu detailních nastavení, takže váš koláčový graf může vypadat přesně podle vašich představ. Taktéž si lze k objektu hned přiřadit limity, alarmy, inicializační hodnotu a jednoduchý kód ve Visual Basic pro různé stavy daného objektu. IGSS celkově působí velice propracovaně, avšak je složité a nepřehledné. Help se určitě snaží pomoci pochopit všechny možné funkce, ale začátečníkovi i tak dá spoustu práce, než se k něčemu dopracuje. Velký důraz se klade na monitorování a reporty. Reporty lze ukládat jak při provedení nějakého eventu (stisk tlačítka, dosažení určité hodnoty nějaké proměnné nebo když nastane alarm) ale i real time. Reporty se ukládají buď do Microsoft Excell nebo na servery IGSS.

IGSS bych doporučil těm, kteří dělají velký projekt a plánují monitorovat větší množství proměnných, potenciálně využít všech 50. Pokud jde uživateli v projektu spíše o řízení doporučil bych mu jiný program, neboť IGSS vyniká spíše v monitorování a alarmech. Taktéž pro malé demonstrační úlohy je tento SCADA systém až moc složitý.

Odkaz na stažení: http://igss.schneider-electric.com/products/igss/download/free-scada.aspx



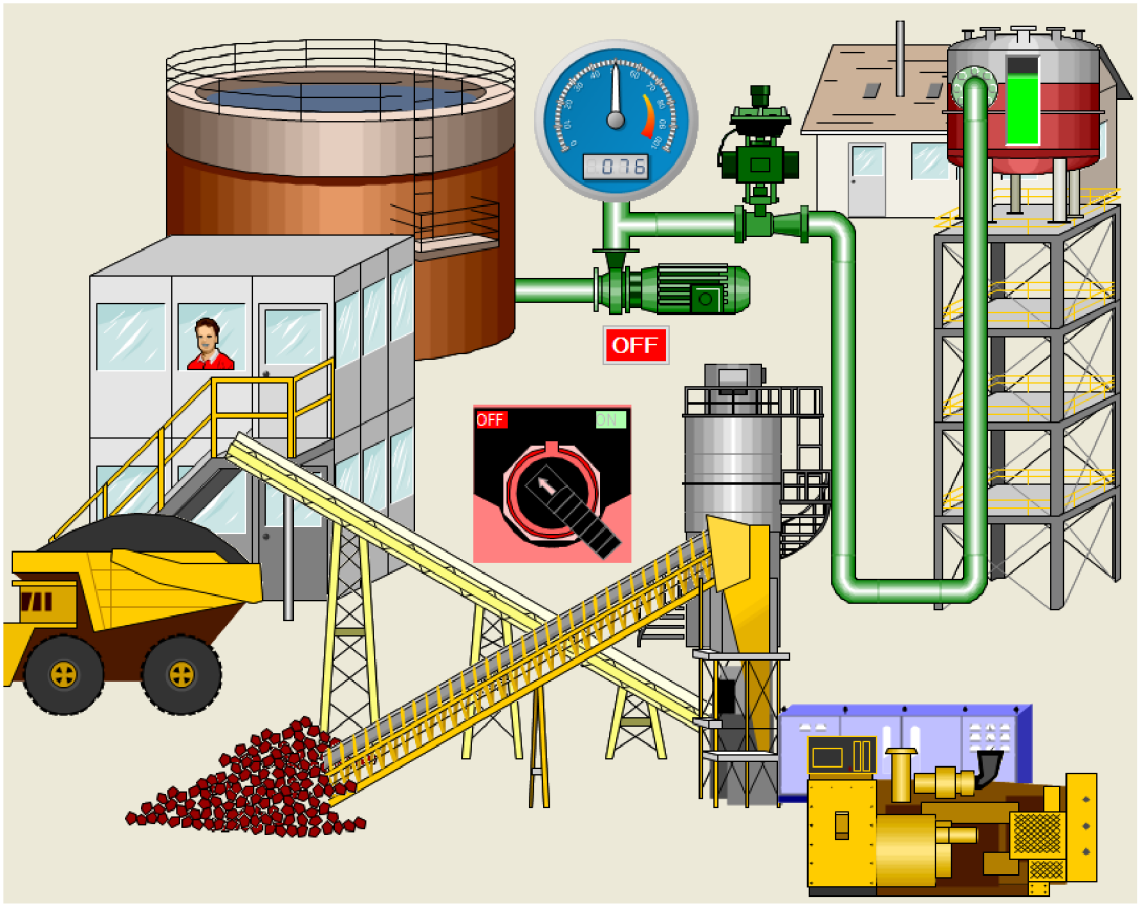
Obrázek 2: Grafický editor IGSS

2.3 WinTr

Turecká společnost Fultek nabízí jejich SCADA systém WinTr zdarma až pro 24 tagů. Pro větší projekty to není dostačující, ale pro moje demonstrační úlohy to stačit bude. WinTr se dělí na dvě části – WinTr Runtime a WinTr Developement. Ve WinTr Developement se projekt vytváří a ve WinTr Runtime se poté spouští, což může být výhodné u spouštění větších, náročnějších projektů, kdy takto není za potřebí otevírat celý projekt a nahrávat do něj všechny jeho součásti (vysvětleno níže). Systém je pouze pro operační systém Windows. Projekt ve WinTr Developement se dělí na tři oddělené části, které se při přeložení spojí v jednu. V Connection Manageru se vytvoří komunikace a tagy. Ve Screen Designeru se vytvoří objekty, kterým se přiřadí dané tagy. A jako poslední je zde script writer, ve kterém se píše script. Tento koncept tak umožňuje mít více scriptů pro jeden projekt a vždy nahrát ten potřebný bez přepisování toho původního. Taktéž lze mít více vizualizací a jen ze Screen Designeru nahrát tu co uživatel zrovna potřebuje. Zprovoznění komunikace přes Modbus TCP je velmi jednoduché. Stačí do nastavení Modbus sítě zadat IP adresu vašeho DHCP serveru a číslo TCP portu pak si přídat jednu nebo více stanic, do kterých se přidávají tagy. U jednotlivých tagů se dá pouze přiřadit jejich adresa registru a sledovat a zapisovat jejich hodnota. Kromě Modbus TCP WinTr podporuje komunikaci ještě pro Omron, Mewtocol(Panasonic), Profinet, S7 Mpi, S7 Ppi, OPC Client a Modbus RTU. Scada nabízí širokou řadu objektů potřebných k monitorování tagů a vytváření vizualizace. Mezi ně patří různé druhy grafů, trubek, zobrazovacích prvků, zobrazení alarmů a eventů atd. Také si lze vytvořit vlastní objektý přímo v grafickém designeru, avšak pouze jednoduché objekty. Uživatel je zde limitován pouze na čáru, obdélník a kruh. Také si lze každý objekt modifikovat, co se týče barvy, velikosti, popisů atd. K objektům lze přidat i funkce OnMouseDown a OnMouseUp a jednoduché animace jako změna polohy a viditelnost. Script writer je přehledný a jednoduše navržený. Lze psát v jazicích Visual Basic script a C# script. Taky je tu možnost výběru, jestli script poběží synchronně, nebo asynchronně což programátorovi dává více možností. Seznam tagů a objektů dává možnost přesné deklarace, aniž by programátor pořád musel nahlížet do Screen Designeru nebo Connection Manageru a deklarace z tohoto seznamu sama určí o jaký datový typ se jedná. Po zkompletování všech částí projektu se pracuje ve WinTr Runtime. Zde už je vytvořená aplikace v chodu a operátor může používat její řídící nástroje, nebo tlačítka na tvorbu reportů.

WinTr je přehledný a jednoduchý SCADA systém. Jeho omezení na 24 tagů je pro menší projekty zcela dostačující. Oproti ostatním ničím nevyniká, ale taky ničím nezklame.

Odkaz na stažení: http://www.scadasoftware.net/ScadaSoftwareDownload.html



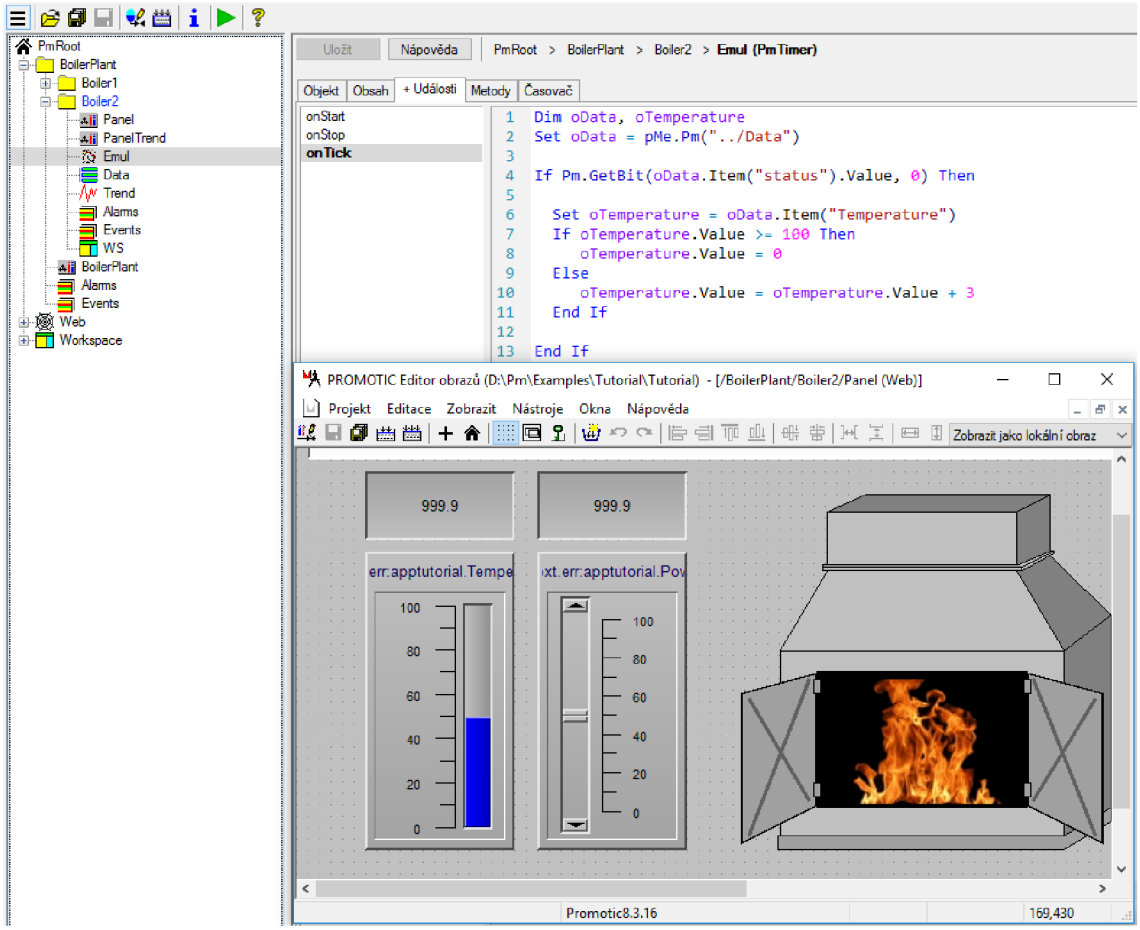
Obrázek 3: Výsledná vizualizace ve WinTr Runtime

2.4 Promotic PmFree

Promotic PmFree je free verze komplexního SCADA objektového programu pro tvorbu aplikací na řízení a sledování technologických procesů v průmyslu. Od placené verze se liší pouze množstvím proměnných a grafických objektů. Je tedy možno mít 30 proměnných a 10 grafických objektů. Vyvíjí ho česká firma Microsys sídlící v Ostravě. Promotic je tím pádem kompletně v češtině. Promotic pracuje na systémech Windows. Při tvorbě aplikace lze zvolit, aby aplikace byla zároveň WEB serverem a nabízela HTML a XML stránky. Struktura softwaru se dělí na editor aplikace a editor obrazů. V editoru aplikace se definuje stromová struktura objektů. Do aplikace se zde vkládají jednotlivé části jako objekty. První objekt, který zde musíme vložit je objekt pro komunikaci. Promotic podporuje veškeré nejpoužívanější drivery jako Modbus TCP, Modbus RTU, Siemens, IEC8705, Allen Bradley, Omron, Wago a mnoho dalších. Zprovoznění komunikace bylo bezproblémové a výsledná komunikace byla velmi robustní. Dalším objektem je script. Promotic používá na vytváření scriptu jazyk Visual Basic Script. Script se píše pro jednotlivé eventy (onStart, onStop, onTick) což usnadňuje práci a činí scrip přehlednějším u větších aplikací. V návodu je pěkně popsané jak script vytvořit a jak s objekty pracovat. Součástí systému je objekt trendů. Zde se ukládají hodnoty ať už real-time nebo když nastane event/alarm.

Tyto hodnoty se ukládají na disk počítače ve formátech Dbase, MySQL, Oracle nebo FireBird. Hodnoty je možno poté zobrazovat v grafické nebo tabulkové podobě a dále je analyzovat. Posledním nezbytným objektem je objekt obrazu. Zde se přepíná do editoru obrazů. V editoru obrazu se tvoří veškerá grafika. Lze zde nalézt obrovský výběr panelů, tlačítek, displayů a také nespočet průmyslových objektů jako potrubí, povrchy, komíny, čerpadla a mnoho dalších. Jednotlivým objektům se zde v jejich editaci přiřazuje proměnná a taktéž k nim jde rovnou psát scripty pro jednotlivé události (onStart, onMousePress...). Tyto scripty jdou mimo jazyk Visual Basic Script psát i v jazyce JavaScript. Také tu lze nastavit klasické atributy grafických objektů jako barva, meze, viditelnost a další. Grafický editor je v Promotic velice propracovaný, ale 10 grafických objektů postačí pouze na menší projekty.

Odkaz na stažení: http://www.promotic.eu/cz/promotic/download/download.htm



Obrázek 4: Uživatelské prostředí Promotic

2.5 FreeScada2

Tento OpenSource projekt je založen na platformě .NET 3.0 a C# a běží na Windows. Vývoj tohoto projektu skončil v roce 2012, takže už nejspíš nelze očekávat novou verzi. Cílem bylo vytvořit OpenSource SCADA systém pro veřejnost a také pro komerční využití. První verze byly zaměřeny pro domácí použití, které nevyžadovalo plně funkční systém, ale stačilo pouze na jednoduchou vizualizaci. Poslední verze už se plně funkčnímu SCADA systému docela podobá, ale je pořád plná nedodělávek a bugů. Narozdíl od většiny OpenSource SCADA systémů je FreeScada2 dostupná i jako instalační aplikace, takže na její spuštění nejsou potřeba žádné další programy. Pro .NET vývojáře, kteří si chtějí v systému něco doplnit nebo upravit je ovšem dostupný i její kód. FreeScada2 se dělí na Designer a Runtime. V Designeru se aplikace vytváří a Runtime slouží na vizualizaci, sledování alarmů a vytváření reportů. FreeScada2 nabízí pouze dvě možnosti komunikace a to Modbus TCP a spojení s OPC serverem. Navázání komunikace přes Modbus TCP proběhlo úspěšně avšak nebylo vždy robustní. Taktéž proměnné na určitých adresách nekomunikovaly. Grafický editor obsahuje pramálo možností. Je zde možno navrhnout základní geometrické tvary a polynom. Knihovna s nástroji obsahuje pouze pár tlačítek, posuvníků a textových oken. Všem objektům lze měnit jen základní vlastnosti jako barva, viditelnost a velikost. Navázání proměnných na grafické objekty není zrovna nejelegantnější. Změny určitých vlastností jde spojit s různými proměnnými což přináší jisté možnosti. Script se píše v embedded jazyce založeném na syntaxích Pythonu. Při psaní scriptu se využívá vestavěných knihoven, což je pro běžného uživatele nepohodlné a komplikované.

OpenSource project FreeScada2 je viditelně nedodělaný a nenabízí možnosti potřebné k tvorbě náročnějších a větších SCADA apliakcí. Pro jednoduché výpisy alarmů v domácích podmínkách však postačí.

Odkaz na stažení: https://sourceforge.net/projects/free-scada/?source=navbar

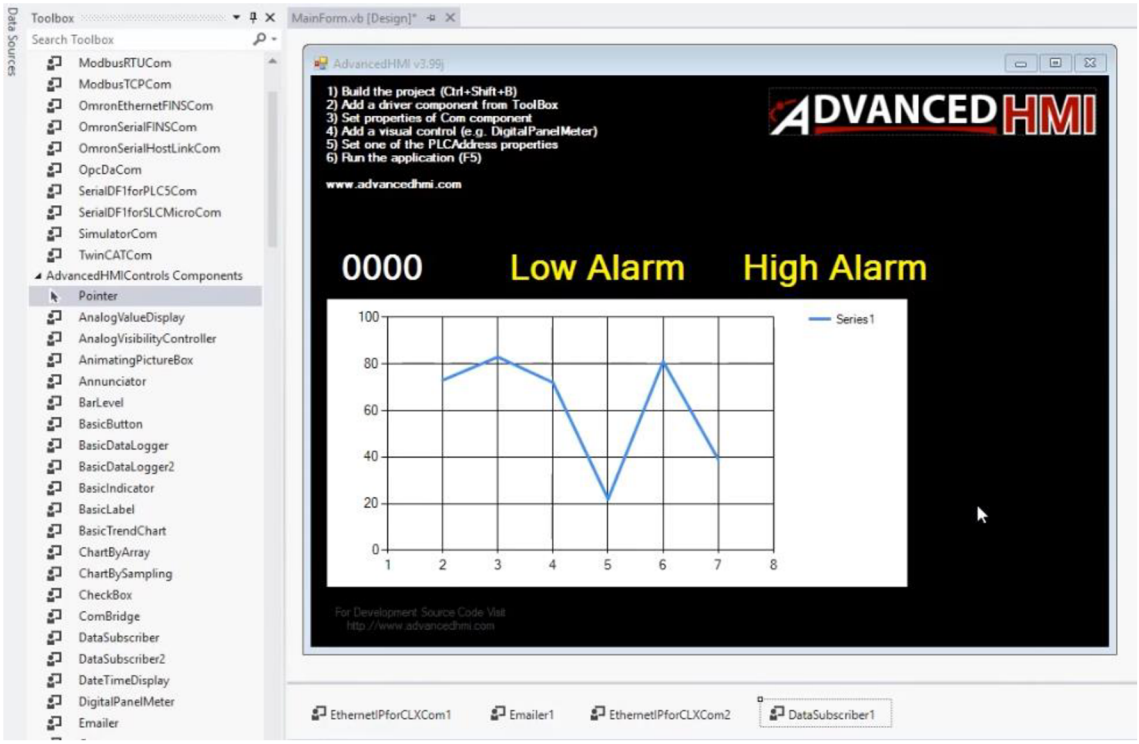
2.6 AdvancedHMI v3.99s

AdvancedHMI je jeden z nejpopulárnějších OpenSource SCADA systémů. Tento software začal vyvíjet Archie Jacobs ze společnosti Manufacturing Automation ve vývojovém prostředí Microsoft Visual Studio. Projekt se lidem zalíbil a brzy se spousta lidí začala podílet na jeho dalším vývoji a rozšíření. Software je založený na platformě Microsoft .NET a celý kód je volně přístupný. Veškeré distribuované aplikace vytvořené v AdvancedHMI musí poskytnout vývojový kód konečnému uživateli a veškerou grafiku lze používat pouze v AdvancedHMI projektech. Tento systém je pouze pro Windows. Celá SCADA nemůže běžet sama o sobě. Jako téměř všechny OpenSource projekty

potřebuje i AdvancedHMI software na přeložení souborů s jejím kódem. Proto je potřeba před spuštěním stáhnout Microsoft Visual Studio. Momentálně podporovaná verze je MVS 2015 community edition, která je zdarma ke stažení na stránkách Microsoft. Ve Visual Studiu se pak spustí solution programu a přeloží se (Build Solution). Drivery pro komunikaci jsou v aktuální verzi dostupné pro AllenBradley, Modbus TCP, Modbus RTU, Omron a TwinCAT. Daný driver stačí přetáhnout z ToolBoxu Visual Studia do přeloženého projektu. Zprovoznění komunikace přes Modbus TCP je velice snadné. Do možností přidaného driveru se napíše IP adresa DHPC serveru a proměnné bez problému reagují přes jejich adresy. Před adresu je však nutné připsat ještě číslo, značící operaci, kterou bude proměnná vykonávat z tabulky Modbusových funkčních kódů. Komunikace je velmi robustní. Grafické objekty se do projektu přiřazují tak jako všechno ostatní, a to přetažením z ToolBoxu Visual Studia. Je zde k dispozici pár standartních displayů, ukazatelů a tlačítek jako u všech SCADA systémů. Pro .NET programátory je tu samozřejmě možnost udělat si jakýkoliv grafický objekt se jim zlíbí, ale běžný uživatel si musí vystačit s hrstkou těch, které jsou tu nachystány. Objektům lze měnit barvu, velikost a jiné drobné úpravy. Je tu také možnost dokoupit si grafické balíčky za pár dolarů na webových stránkách AdvancedHMI. Script se v AdvancedHMI píše k jednotlivým objektům, a to v jazyce Visual Basic. Psaní kódu je zjednodušeno tím, že se provádí přímo ve Visual Studiu, což je velmi kvalitní vývojové prostředí. Uživatel tak využije možnosti debuggingu, automatické nabídky výběru funkcí pro daný objekt a jiných vymožeností, které jiné než OpenSource SCADA systémy běžně nemají. Pro vytváření databází se používají databázové soubory, které jsou po naprogramování schopné sbírat data a nahrávat je na server nebo do souboru.

AdvancedHMI je na poměry OpenSource SCADA systémů výborný program avšak pro běžného uživatele zaostává za těmi s vlastním uživatelském prostředím přehledností, možností komunikace a možnostmi grafických objektů. Pro .NET programátory je to však svět téměř neomezených možností. Pro začátečníky je zde návod jak s tímto softwarem začít pracovat a taktéž je od vývojáře Archieho dostupných několik instruktážních videí.

Odkaz na stažení: https://sourceforge.net/projects/advancedhmi/?source=directory



Obrázek 5: Uživatelské prostředí AdvancedHMI

Data	
(ApplicationSettings)	
(DataBindings)	
Tag	
Design	
(Name)	**MomentaryButton1**
GenerateMember	True
Locked	False
Modifiers	**Friend**
Focus	
CausesValidation	True
Layout	
Anchor	Top, Left
Dock	None
Location	**81; 152**
Margin	3; 3; 3; 3
MaximumSize	0; 0
MinimumSize	0; 0
Padding	0; 0; 0; 0
Size	**75; 110**
Misc	
ButtonColor	**Blue**
LegendPlate	**Large**
OutputType	**MomentarySet**
SuppressErrorDisplay	False
PLC Properties	
ComComponent	**ModbusTCPCom1**
MaximumHoldTime	**3000**
MinimumHoldTime	**500**
PLCAddressClick	**21486**
PLCAddressVisible	**21486**

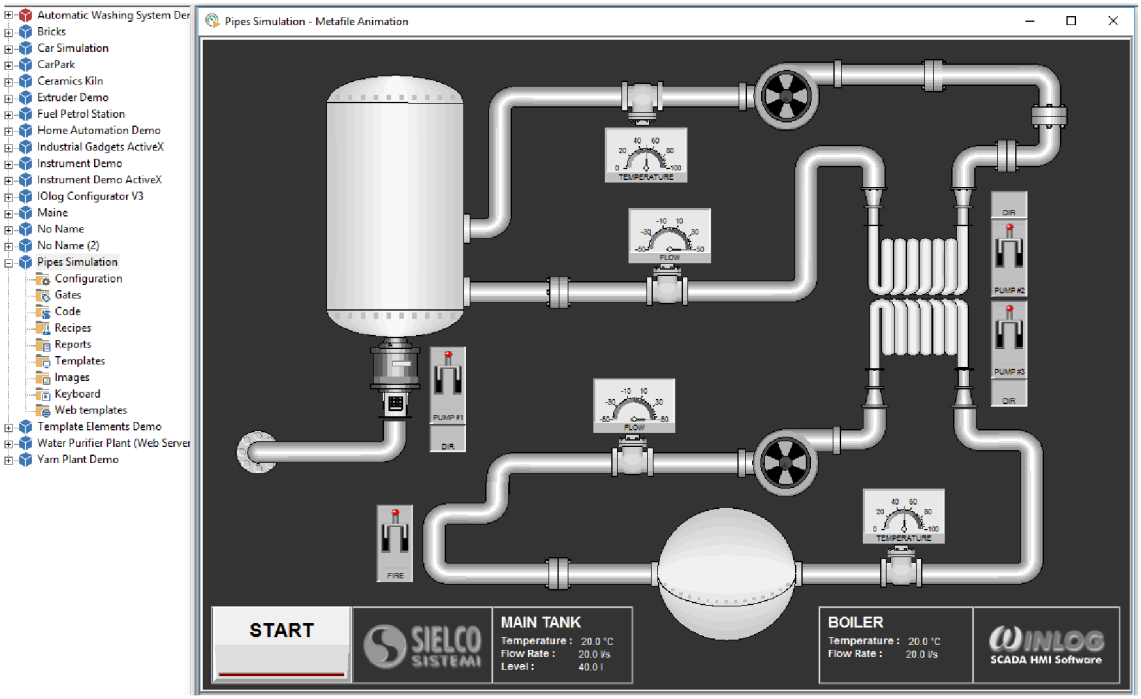
Obrázek 6: Komunikační blok AdvancedHMI ve Visual Studiu

2.7 WinLog Lite 3

Winlog Lite je free verzí placeného SCADA systému Winlog Pro vyvinutý italskou společností Sielco Sistemi. Na rozdíl od Winlog Pro je Winlog Lite omezen pouze na 24 tagů a omezenou grafickou knihovnu. Winlog Lite lze spustit ve full a demo módu. Úplně free je pouze demo mode a ten se od full modu liší tím, že lze komunikovat s externími zařízeními pouze po dobu patnácti minut. Po této době se komunikace přeruší a musí se manuálně obnovit. Tímto omezením je tento SCADA systém nepoužitelný pro komerční použití. Winlog je podporován pro Windows včetně Windows Embedded a Windows server. Interface totohle SCADA systému je velmi jednoduchý a přehledný. Celý projekt je rozebrán do pár záložek, které se vytvoří po založení projektu. Komunikace se nastavuje v záložce configuration. Zprovoznění komunikace přes Modbus TCP nedělalo žádné problémy. Tagy se definují v záložce gates. Pro přehlednost jsou rozděleny na numerické(analogové), digitální, stringy, eventy/alarmy a compoundy. Compoundy obsahují složené datové typy na změnu barvy nebo změnu polohy daného objektu. U každého tagu lze nastavit řadu věcí jako adresa registru pro komunikaci s Modbus Slave, minimální/maximální hodnota, různé matematické operace upravující hodnotu tagu a další. Vzhled aplikace a přiřazování tagů ke grafickým objektům se provádí v záložce Templates nebo Web Templates. Web template vytváří webovou aplikaci a přeložený projekt je pak možno ovládat přes webový prohlížeč. WinLog ani zde neubývá na jednoduchosti. Veškerá grafika působí tak prostě jak to jen jde. U grafických objektů v knihovnách, které Winlog nabízí je přímo možnost kliknout na editace, která otevře Microsoft Paint. Taktéž je možnost si vytvářet objekty vlastní a vložit je do složky s projektem. Script je psán v jednoduchém jazyce založeném na Visual Basic. V helpu jsou podrobně popsány všechny možnosti tohoto jazyka, takže i programátor začátečník nemá problém svou vizualizaci přivést k životu.

Winlog Lite v praxi nenalezne využití. Nutnost manuální obnovy komunikace každých 15 minut z něj dělá skutečně pouze demo verzi, sloužící k vyzkoušení funkcí plné, placené verze.

Odkaz na stažení: https://www.sielcosistemi.com/en/download/public/winlog_lite.html



Obrázek 7: Uživatelské prostředí WinlogLite3

2.8 Placené SCADA Systémy

2.8.1 Control Web

SCADA systém Control Web vyvíjí už přes 15 let česká firma Moravské přístroje. Využívá se pro jakékoliv vizualizační a řídící aplikace využívající sběr dat a jejich vyhodnocení. Systém je zcela nezávislý na hardwaru, podporuje veškeré standardy pro průmyslové automaty. Otevřená rozhraní ovladačů umožňuje také vytvořit vlastní ovladač. Control Web je oproti konkurenčním systémům ze zahraničí cenově atraktivní, proto je mimo velké firmy a jejich obsáhlých aplikací používán také ve školách a ve výzkumu. Control Web se stále více používá pro malé aplikace. Čím dál častěji se stává, že je systém složen z více částí, které spolu potřebujou komunikovat. Proto každá malá jednotka má k dispozici Ethernet, Wi-Fi, Bluetoth, USB a může obsahovat HTTP server a také webový klient kam může posílat SMS zprávy, emaily a komunikovat přes GPRS a rádiové mosty. Kromě toho, že Control Web zvládá veškerou funkčnost SCADA systémů lze jej použít také jako firemní WWW server, automatický registrační a aktivační server s webovým a SMS rozhraním a integrující systém pro laboratoře. Ceny nejnovější verze Control Web 7 činí 21 700 Kč bez DPH pro vývojovou verzi a dalších 6 500 Kč bez DPH pro Runtime verzi sloužící k distribuci aplikačníc programů . [2]

2.8.2 InTouch

Tento SCADA systém společnosti Wonderware je vyvíjen již od roku 1987 a je široce používán v chemickém, strojním, papírejnském a ropném průmyslu. Prodalo se již přes 300 000 licencí. InTouch lze použít na všech řídících systémech od různých společností. Má podporu přes 800 zařízení. Komunikaci je možné vytvořit přes OPC servery nebo přes Wonderware I/O server nebo přes jiné servery. Na komunikaci s databázemi InTouch využívá moduly ADO/ODBC, .NET, SQL, SPC a další. I v novějších verzích pak zůstává podpora DDE protokolů pro starší aplikace. Dále WinCC využívá Wonderware SuiteLink pro rychlé síťové komunikaci, pracující na bázi TCP/IP. Přesná cena systému InTouch není volně dostupná. Cenu vypočítají při dotazu přes formulář registrovaný zákazníkům na míru podle jejich požadavků. Podle ceníků z roku 2010 se základní verze InTouch pohybovala okolo 30 000 – 40 000 Kč bez DPH. [3]

2.8.3 WinCC

Tento SCADA systém společnosti Siemens se používá pro náročné aplikace v různých oblastech průmyslu. WinCC vyniká tím, že automaticky přebírá kód, který vytvoří programátor v TIA Portal což ušetří několik kroků práce. Stejně jako ostatní placené SCADA systémy i WinCC podporuje takřka všechny formy komunikace. Především pak PROFIBUS/PROFINET, se kterými Siemens pracuje nejčastěji. Nechybí však ani Modbus, OPC a další. V paralelních aplikacích se data sdílejí přes OLE, DDE a OPC. WinCC nabízí i funkce navíc, které jsou dodávány jako softwarové nadstavby a kupují se zvlášť. Jsou mezi nimi například vzdálené přístupy, nadstavby pro přístup ke komrimovaným historickým datům, databázové nadstavby, pro práci se servery a zálohování stanic. Stejně jako Wonderware, ani Siemens nezveřejňuje ceník svých produktů. Starší verze se prodávají okolo 20 000 – 60 000 Kč. [4]

2.9 Zhodnocení

Zde jsou poznatky z rešerše shrnuty do Srovnávací tabulka(Tabulka 2). SCADA systémy podporující Modbus TCP jsou zde obodovýny podle zadaných kritérií: Zprovoznění SCADA systému, Navázání komunikace s PLC přes Modbus TCP, Robustnost komunikace, Možnosti grafických objektů a kvalita animací, Možnosti a obtížnost psaní scriptu, Dostupnost návodů a tutoriálů a počet použitelných proměnných. 5 bodů je nejvíce a 1 je nejméně.

Tabulka 1: Srovnávací tabulka SCADA systémů

	Instalace	Komunikace	Robustnost	Grafika	Script	Návody	Proměnné
Laquis	5	5	5	2	4	5	Neomezeně
IGSS	5	5	5	5	2	3	50
WinTr	5	4	1	4	5	4	24
Promotic	5	5	5	5	4	5	30
FreeScada2	5	3	3	1	2	1	Neomezeně
AdvancedHMI	3	4	5	2	5	4	Neomezeně
WinLog	5	5	1	4	3	4	24

3 PFC200 CS 2ETH RS

Obrázek 8: Kontrolér PFC200 CS 2ETH RS [5]

3.1 Základní popis

Kontrolér 750-8202 neboli PFC200 CS 2ETH RS je automatizační zařízení, které vykonává řídící úkoly PLC. Je vhodný pro připojení na DIN lištu a vyčnívá pro jeho široké využití. Lze k němu připojit všechny dostupné I/O moduly WAGO-I/O-SYSTEM 750 (série 750 a 753), které mu umožňují interně zpracovávat analogové a digitální signály z automatizačního prostředí nebo posílat tyto signály do jiných zařízení přes jeho různá rozhraní.

Automatizační úkoly jsou vykonávány ve všech kompatibilních jazycích standardu IEC 61131-3 pomocí WAGO-I/O-PRO nebo vývojovým prostředím e!COKCPIT. Pro programování CODESYS aplikací kontrolér poskytuje 16 MB programové paměti (flash) a 64 MB datové paměti (RAM) pod CODESYS 2 a 64 MB programové a datové paměti pod e!RUNTIME.

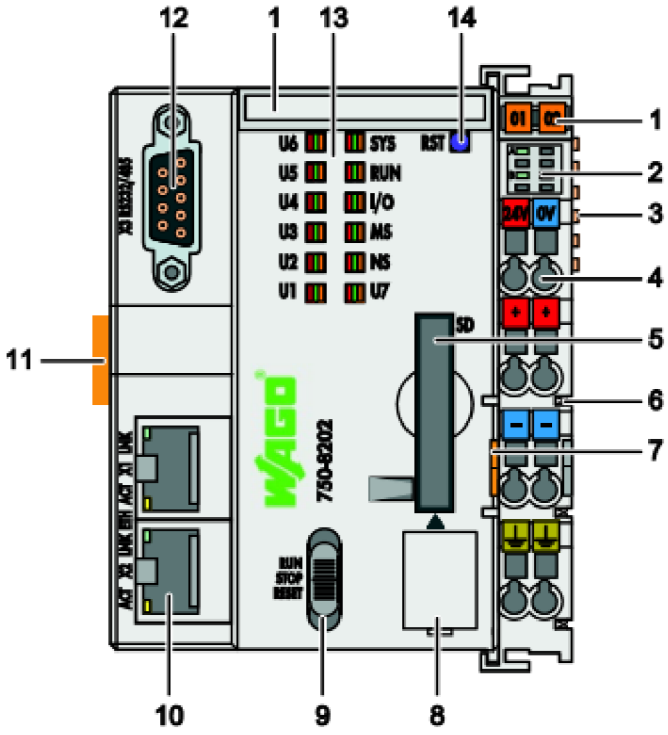
Dvě Ethernetové rozhraní a integrovaný přerušitelný switch umožňuje zapojení dvou oddělených sítí se společnou MAC adresou a IP adresou pro každé rozhraní. Taktéž je lze propojit sériově se společnou MAC adresou a IP adresou pro obě rozhraní. Obě tato rozhraní podporují 10BASE-T/100BASE-TX, plný i poloviční duplex a Auto-MDI(X).

Na výměnu dat jsou zde implementovány sběrnicové obvody:

- Modbus TCP Master/Slave
- Modbus UDP Master/Slave
- Modbus RTU Master/Slave (přes RS-232 nebo RS-485)

V kontroléru jsou všechny vstupní signály ze snímačů jsou zkombinovány. Po připojení kontroléru jsou všechny I/O karty ve sběrnici rozpoznány a je z nich vytvořen výsledný obraz. Analogová data a data speciálních modulů jsou odesílána jako WORD nebo BYTE a digitální data jsou odesílána bit po bitu. V počátečním stavu je v kontroléru nainstalován firmware založený na bázi Linuxu. Spolu s ním jsou v něm předinstalovýny aplikační programy a řada různých podpůrných programů:

- SNMP server/client
- Telnet server
- FTP, FTPS server
- SSH server/client
- Web server
- NTP client
- BootP a DHCP client
- DHCP server
- DNS server



Obrázek 9: Komponenty kontroléru PFC200 [5]

Tabulka 2: Popis komponentů kontroléru PFC200[5]

1	Mini-WSB
2	LED indikátory napájení
3	Datové kontakty
4	CAGE CLAMP přívod napájení
5	Slot pro paměťovou kartu
6	Napájení I/O modulů
7	Releasing strap
8	Servisní rozhraní
9	Přepínání režimu
10	Ethernetové připojení
11	Zamykání
12	Sériové rozhraní
13	LED indikátory systému
14	Reset

Rozhraní Ethernetu X1 a X2 kontroléru jsou propojeny 3-portovým switchem, přičemž třetí port je připojen k CPU. Rozhraní X1 a X2 můžou operovat buď ve switch modu nebo jako oddělené síťové rozhraní. Změny mohou být prováděny během runtimu. Switch mode se aktivuje automaticky během prvního spuštění a konfigurace je nastavena na DHCP. Pro rozhraní X1 lze nastavit fixní IP adresu a nemá vliv na mód, který byl předtím nastaven. Síť je zabezpečena uživatelským jménem a heslem. [5]

4 E!COCKPIT

e!COCKPIT je vývojové prostředí společnosti WAGO s moderním ovládáním a programováním podle normy IEC 61131-3 na osvědčené platformě CODESYS 3, která poskytuje jednoduché a všestranné vytváření aplikací. Toto integrované vývojové prostředí umožňuje každou automatizační úlohu z hardwarové komunikace, programování, simulace a vizualizace až po uvedení do provozu v jednom softwarovém balíčku. [6][7]

4.1 Konfigurace

e!COCKPIT používá funkci Fast fieldbus configuration. Tento nástroj má několik využití. Komplexní sítě lze jednoduše plánovat, konfigurovat a programovat z "network view". Kromě klasických síťových protokolů jako Modbus, jiné fieldbusy jako Profibus, CAN a CANopen lze implementovat pomocí několika kliků myši. Drag & Drop funkce umožňuje snadné a intuitivní přidávání prvků a funkce Copy & Paste umožňuje rychlé duplikování celých síťových větví. e!COCKPIT řeší veškerou hardwarovou konfiguraci za uživatele. Funkce Scan rozpozná a nastaví všechna PLC WAGO připojená k počítači včetně karet vstupů, karet výstupů a případně dalších modulů. [6][7]

4.2 Programování

Vývojové prostředí e!COCKPIT se zaměřuje na automatizační techniku, a proto používá průmyslový standard CODESYS 3. Tato platforma umožňuje programovat v šesti různých jazycích standardu IEC 61131-3(Ladder Diagram, Function Block Diagram, Structured Text, Instruction List, Continuous Function Chart, Sequential Function Chart). CODESYS 3 tyto jazyky umožňuje kombinovat různě mezi sebou. e!COCKPIT nabízí taktéž portfolio předpřipravených knihoven s často používanými funkcemi ke zkrácení vývojového času. Toto prostředí používá také moderní technologie a aktuální standardy jako CSS a HTML5 sloužící uživateli k ovládání zařízení pomocí tabletů nebo smartphonů. CODESYS 3 prochází neustálým vývojem, což jej činí velice bezpečnou investicí. Programy vytvořené v CODESYS 3 lze opětovně použít v e!COCKPIT, ovšem hardwarová konfigurace se přenést nedá a musí se znovu vytvořit. [6][7]

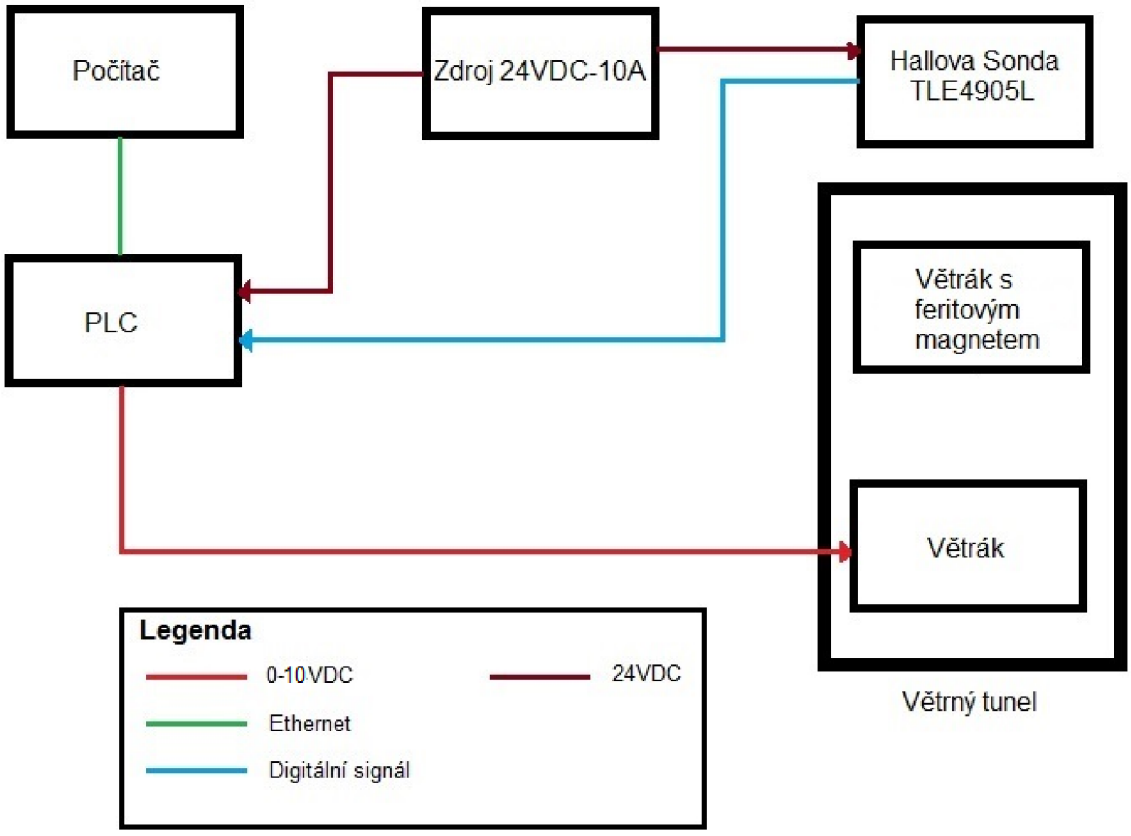
4.3 Vizualizace

e!COCKPIT využívá vizualizační funkce CODESYS 3. To umožňuje použití předdefinovaných vizualizačních prvků a příkazů pro simulaci, řízení a monitorování systémů. Webové připojení CODESYS-Web-Visu také umožňuje sledování a řízení visualizací ve webovém prohlížeči nebo na externím zařízení přes CODESYS-HMI. Integrovaný vizualizační editor umožňuje přímý přístup k proměnným. e!COCKPIT implementoval systém Drag & Drop, aby urychlil vývojový proces a zmodernizoval vývojové prostředí. Není závislý na jednom jazyku nebo systému, ale využívá jak Unicode tak moderní standardy jako HTML5 a CSS. [8]

5 ŘÍZENÍ VĚTRNÉHO TUNELU

5.1 Popis modelu

Model větrného tunelu se skládá z tunelu, na jehož obou koncích jsou větráky. Jeden větrák je napájen z analogové výstupní karty PLC napětím 0-10VDC. Druhý větrák napájen není a má na sobě přilepený feritový magnet. Z vnější strany tunelu je pak přidělaná Hallova sonda TLE4905L, která slouží ke snímání otáček a indukované napětí posílá do PLC jako digitální signál.



Obrázek 10 Blokové schéma zapojení modelu větrného tunelu

Po prostudování modelu větrného tunelu bylo nutné vyměnit větrák za silnější model. Větrák, který byl původně v tunelu využit, dokázal roztočit větrák s magnetem až při vyšším napájecím napětí. Taktéž Hallova sonda musela být vyměněna, neboť předešlá sonda měla ulomené vývody. Magnet taktéž musel být nahrazen silnějším. Po výměně těchto komponentů jej bylo možné připojit do vstupních a výstupních karet PLC. Vstupy a výstupy jsou vypsány v tabulce 3.

Tabulka 3 Použité vstupy a výstupy PLC v modelu větrného tunelu

Proměnná	Adresa	Datový typ	Popis
Vetrak	DI: %IX1.0	BOOL	Impulsy z Hallovy sondy
Out	AO: _QW4	WORD	Napájení větráku 0-10VDC

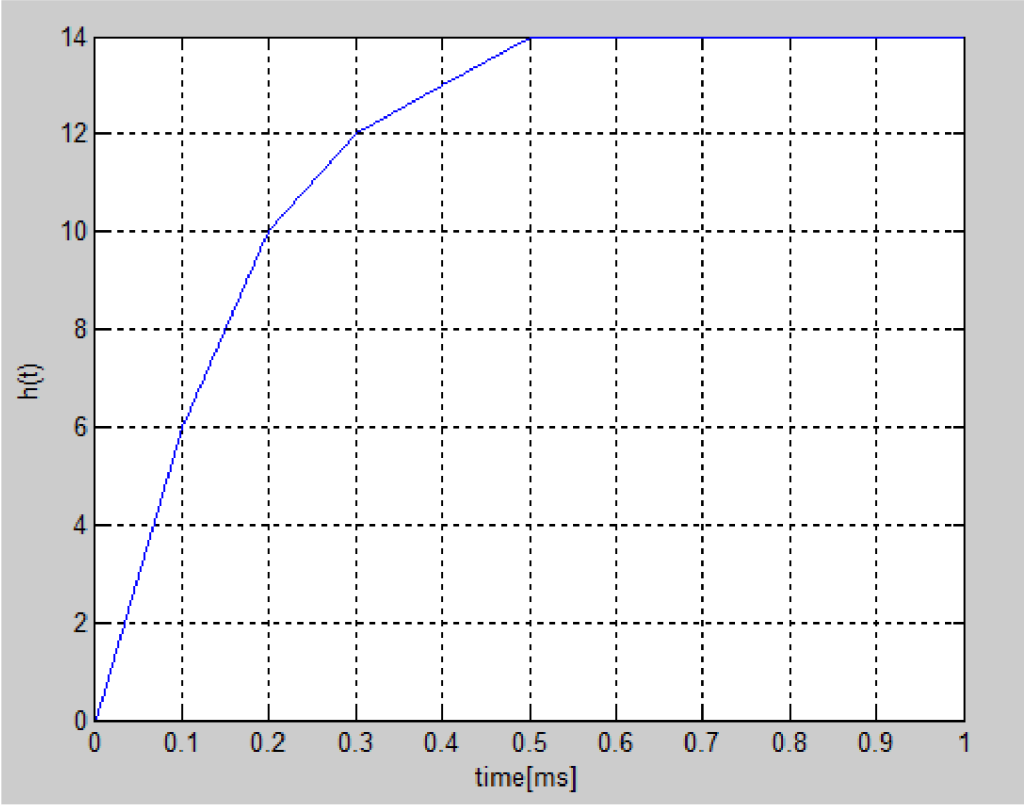
5.2 Identifikace soustavy

Soustavu lze považovat jako diskrétní systém prvního řádu bez dopravního zpoždění. Avšak pro identifikaci jsme soustavu uvažovali jako spojitý systém prvního řádu:

$$F(s) = \frac{K}{T_1 s+1} \tag{1}$$

Dopravní zpoždění $T_d = 4{,}5s$ vzniká pouze při spuštění ventilátoru z důvodu velké hmotnosti magnetu, která zpomalí roztočení měřícího větráku. Pro zjednodušení nebylo dopravní zpoždění v návrhu uvažováno. Dopravní zpoždění by šlo vyřešit nahrazením feritového magnetu, který byl k dispozici, magnetem neodymovým, který má při podobné účinnosti mnohem menší rozměry a hmotnost

Dynamiku soustavy prvního řádu lze idenfitikovat pomocí odezvy na jednotkový skok, přičemž soustava musí být izolována od všech rušivých okolních vlivů a musí být v ustáleném stavu. Odezva na jednotkový skok je na obrázku 11.



Obrázek 11 Skoková změna žádané hodnoty otáček větráku v čase

Z přechodové charakteristiky lze určit parametry soustavy. Zesílení K je rovno jednotkovému skoku a hodnota časové konstanty T_1 je rovna času, kdy je jednotkový skok v 63% ustálené hodnoty. Tudíž má naše soustava parametry:

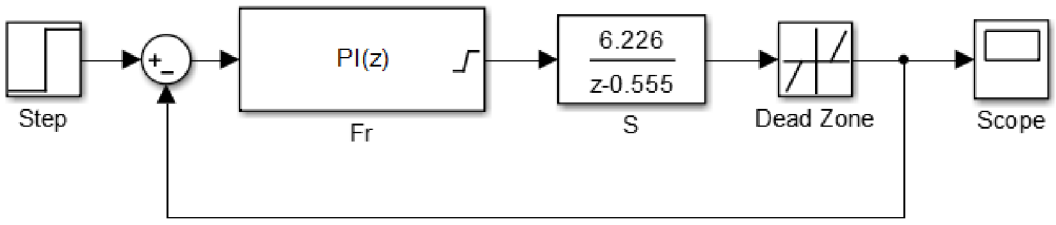
$$F(s) = \frac{14}{0{,}17s+1}$$

PLC však pracuje v cyklech, proto se musí soustava diskretizovat s vzorkovací periodou $T_{vz} = 0{,}1s$. Diskretizovaná soustava má pak parametry:

$$F(z) = \frac{6{,}226}{z-0{,}555}$$

5.3 Návrh regulátoru

Pro návrh regulátoru byl vytvořen matematický model v Matlabovském modulu Simulink.

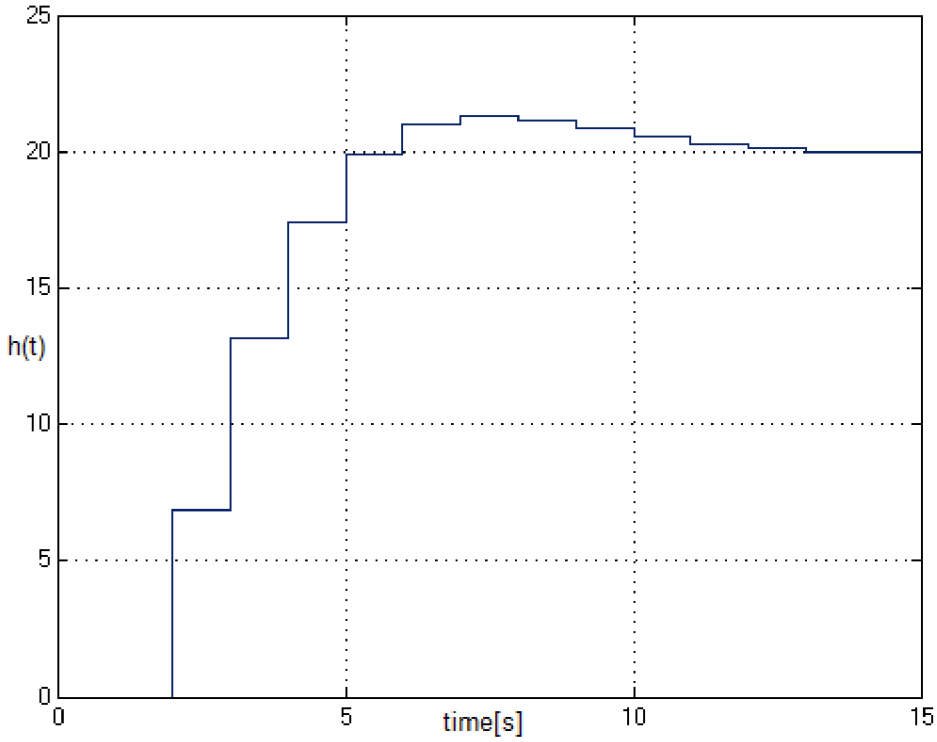


Obrázek 12 Regulační schéma modelu s větrným tunelem

Požadavkem na regulátor bylo co nejrychlejší vyregulování žádané hodnoty. To nejlépe splňoval PS regulátor s přenosem:

$$F_r = 0{,}05 * \left(1 + \frac{0{,}67}{z-1}\right) = \frac{0{,}5z-0.0165}{z-1}$$

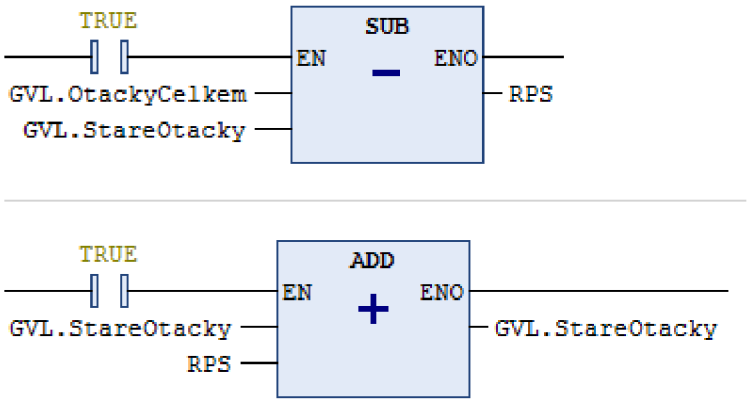
Musí se počítat také s mrtvou zónou, která se nuluje hodnotu až do reálných minimálních otáček tj. RPS = 6 ot/s. Teoretický průběh vyregulování žádané hodnoty RPS = 20 ot/s je vykreslen na obrázku 13.



Obrázek 13Teoretický graf závislosti vyregulování žádané hodnoty v čase z matematického modelu

5.4 PLC program

PLC Program je psán v jazyce Ladder Diagram a operuje ve dvou časových smyčkách. První smyčka má interval 1ms a má nejvyšší prioritu. Tato smyčka slouží ke snímání otáček v podobě logických nul pomocí čitače. Druhá smyčka má interval 1s a slouží k vyhodnocování otáček za sekundu (RPS) a také jako smyčka pro PS regulátor. Počítání otáček za sekundu je řešeno částí kódu na obrázku. Jelikož je vyhodnocovací smyčka sekundová, tak stačí od celkových otáček z čitače odečíst otáčky z minulých cyklů.

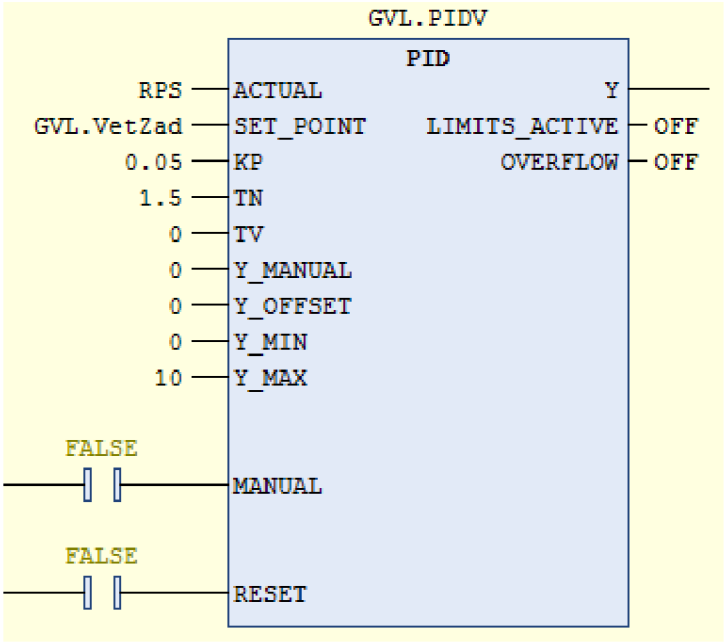


Obrázek 14 Ukázka kódu - měření otáček větráku

Otáčky RPS jsou poté přivedeny na vstup regulátoru jako aktuální hodnota. Výstupem regulátoru je 0-10VDC. Tato hodnota je přes 15ti bitový A/D převodník posílána z analogové výstupní karty PLC do větráku (0-10V = 0-32 768). Větrák pracuje v rozsahu napětí 0-24VDC, proto je hodnota z PLC převáděna přes měnič PWM.

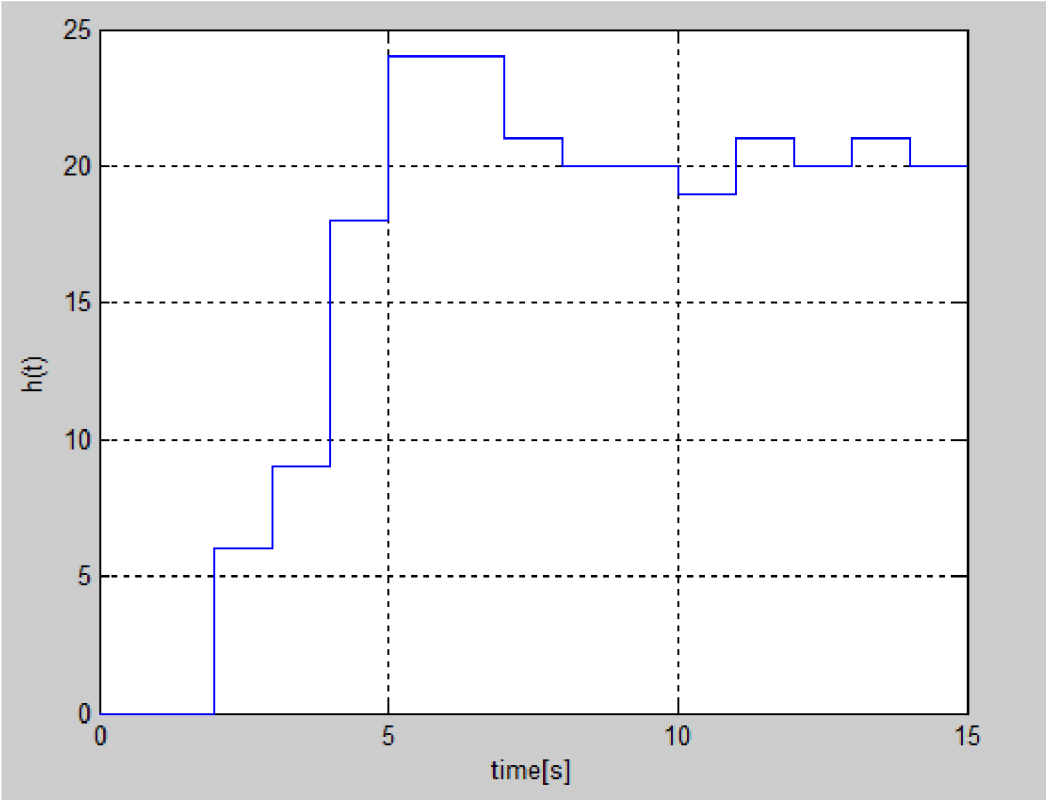
5.5 Implementace regulátoru

V PLC byl regulátor přidán do smyčky s intervalem 1s. Kvůli obrácené hodnotě integrační složky v rovnici použitého bloku PSD regulátoru je zadána hodnota TN = 1,5s.



Obrázek 15 Nastavený blok PID regulátoru v programu

Změřená odezva na změnu žádané hodnoty RPS = 20ot/s je zobrazena na obrázku 16.

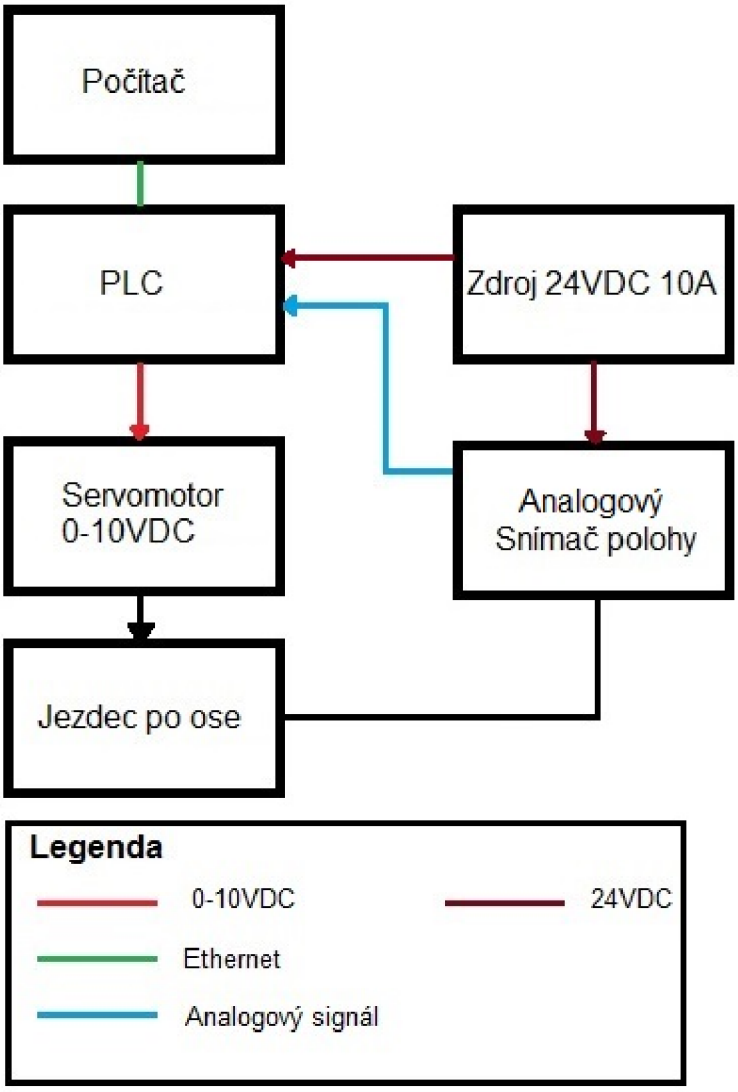


Obrázek 16 Změřená závislost vyregulování žádané hodnoty otáček větráku v čase

6 ŘÍZENÍ OSY

6.1 Popis modelu

Model se skládá s jezdce, motoru M42x40/l s 14ti bitovým A/D převodníkem a analogového snímače polohy s 15ti bitovým A/D převodníkem. Motor je ovládán z analogové výstupní karty PLC 0-10VDC (0-10V = 0-16 384). Analogový snímač polohy posílá hodnoty do PLC přes analogovou vstupní kartu. Nulová poloha je nastavena na hodnotu převodníku 9 160 a maximální na 37 684. Celková vzdálenost, po které se jezdec může pohybovat je 485mm. Blokové schéma se nachází na obrázku 17. Vstupy a výstupy PLC se nachází v tabulce 4.



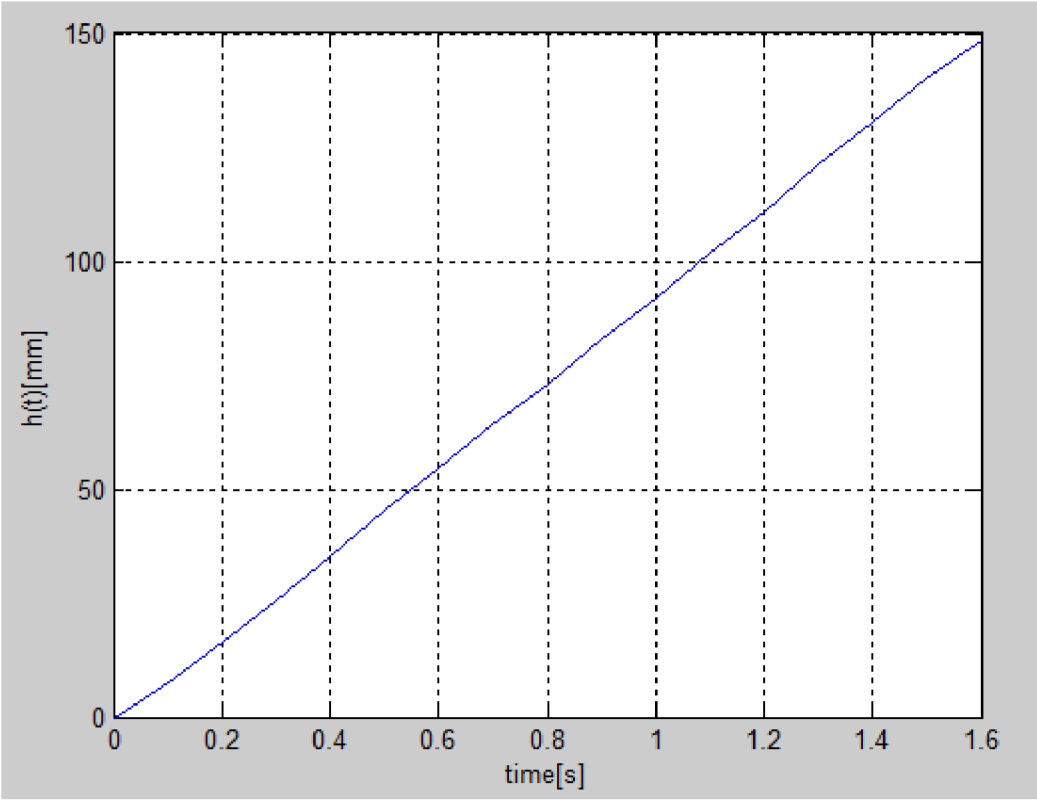
Obrázek 17 Blokové schéma zapojení modelu řízené osy

Tabulka 4 Použité vstupy a výstupy PLC v modelu řízené osy

Proměnná	Adresa	Datový typ	Popis
Poloha	AI: %IW3	WORD	Hodnota aktuální polohy ze snímače
Speed	AO: _QW3	WORD	Napájení motoru 0-10VDC
Swap	DO: %QX0.0	BOOL	Změna směru chodu motoru

6.2 Identifikace soustavy

Prvním krokem k identifikaci soustavy bylo proměření dynamiky polohy, protože narozdíl od rychlosti ji můžemě přesně měřit analogovým snímačem polohy. Byla tedy změřena skoková změna žádané hodnoty, která je na obrázku 18.



Obrázek 18 Změřená závislost skokové změny žádané hodnoty polohy v čase

Na zjištění přenosové funkce tohoto systému byla použita Matlabovská funkce minim, což je kriteriálni funkce pro hledání shody přechodové charakteristiky a funkce fminsearch, která hledá nejmenši chybu řešeni pomocí kriteriálni funkce. Minim vrací kvadratickou chybu, kterou algoritmus fminsearch minimalizuje a tím identifikuje parametry soustavy.

Výsledný celkový spojitý systém má pak přenos:

$$F_1(s) = \frac{94{,}43}{0.02182\,s^2 + s}$$

Tento systém si rozdělíme na:

$$F_1(s) = \frac{94{,}43}{0.02182\,s + 1} \cdot \frac{1}{s}$$

Derivací dostaneme soustavu pro rychlost:

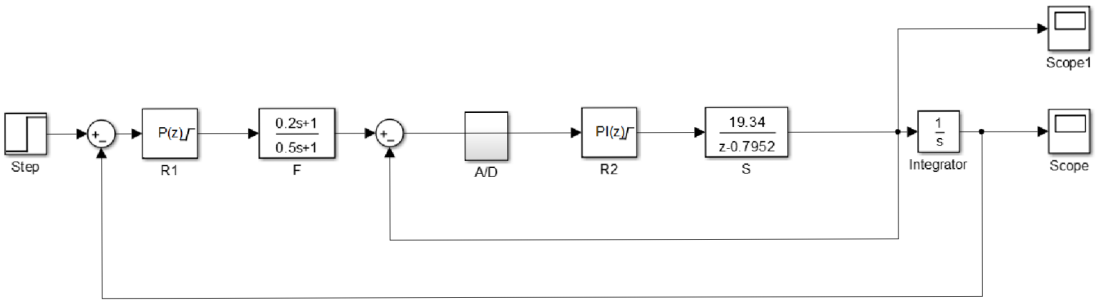
$$F_2(s) = \frac{94{,}43}{0.02182\,s + 1}$$

Systém je opět diskrétní, protože PLC operuje v cyklech. Musíme přenos tedy diskretizovat s vzorkovací periodou $T_{vz} = 0{,}005s$. Výsledný systém pro rychlost má pak přenos:

$$F_2(z) = \frac{19{,}34}{z - 0{,}8}$$

6.3 Návrh regulátoru

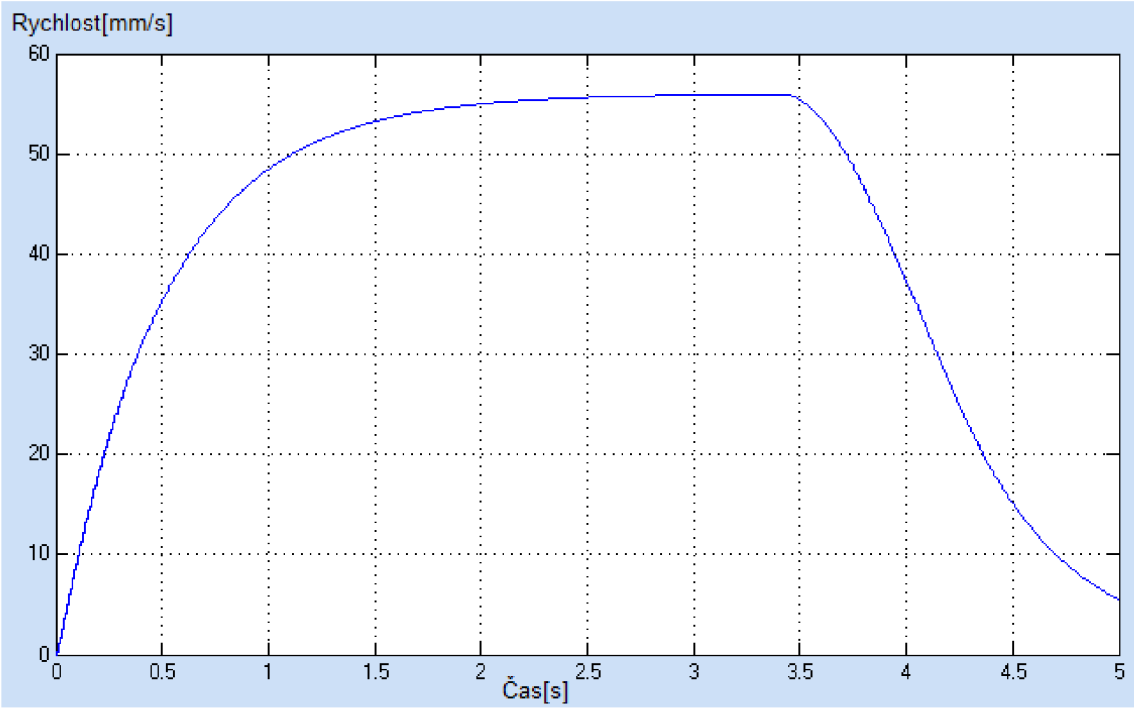
Pro náš systém je třeba použít kaskádové regulace. Jedná se o typický systém s pomocnou regulovanou veličinou. Regulační schéma je na obrázku 19.



Obrázek 19 Regulační schéma modelu řízené osy

První byl navržen regulátor rychlosti. Z matematického modelu byl v Simulinku zvolen PS regulátor s přenosem:

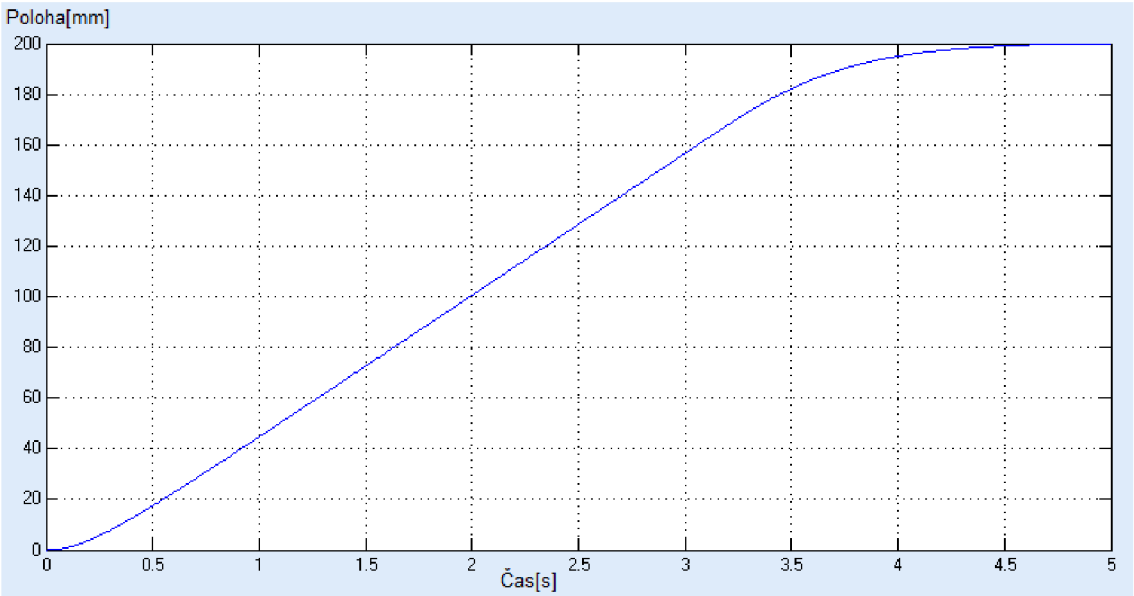
$$R_2(z) = 0.0008 * \left(1 + \frac{0{,}005 \cdot 65}{z-1}\right) = \frac{0{,}000\ 8z - 0.000\ 54}{z-1}$$



Obrázek 20 Teoretický graf závislosti rychlosti motoru na čase při regulaci polohy z matematického modelu

Na regulátor polohy pak už stačí pouze P regulátor. V matematickém modelu bylo navrženo optimální zesílení K=1,5. Výsledný regulátor polohy má pak přenos:

$$R_1(z) = 1{,}5$$



Obrázek 21 Teoretický graf závislosti vyregulování polohy v čase z matematického modelu

6.3.1 Filtr žádané hodnoty

Požadavkem na návrh regulátoru je pozvolný rozjezd a dojezd jezdce do žádané polohy. Tento požadavek lze vyřešit filtrem žádané hodnoty, který byl implementován do matematického modelu v Simulinku. Filtr žádané hodnoty má tvar:

$$F_w(s) = \frac{\alpha T_1 s+1}{T_1 s+1} \tag{2.0}$$

Kde α je filtrační činitel, T_1 je časová konstanta filtru. Filtr pracuje s periodou vzorkování T_s, která bude v našem případě shodná s časovou smyčkou PLC programu, ve které bude filr umístěn.

Pro implementaci filteru do PLC programu musíme z přenosu dostat rovnici pro vstup y(k) , které bude vstupem PS regulátoru. Rovnici rozdělíme na:

$$F_W(s) = \frac{\alpha T_1 s+1}{T_1 s+1} = \frac{\alpha T_1 s}{T_1 s+1} + \frac{1}{T_1 s+1} \tag{2.1}$$

Vyjádříme si diskrétní ekvivalenty obou přenosů:

$$F_{WD}(z) = L^{-1}\left\{\frac{\alpha T_1 s}{T_1 s+1}\right\} + L^{-1}\left\{\frac{1}{T_1 s+1}\right\} = \frac{1-z^{-1}}{1-e^{-\frac{T_S}{T_1}}z^{-1}} + \frac{\left(1-e^{-\frac{T_S}{T_1}}\right)z^{-1}}{1-e^{-\frac{T_S}{T_1}}z^{-1}} \tag{2.2}$$

Zavedeme substituci:

$$a = e^{-\frac{T_S}{T_1}} \tag{2.3}$$

Výsledný přenos rozdělíme na vstupní část Y(z) a výstupní část U(z) a vyjádříme stavovou proměnnou E(z):

$$F_{WD}(z) = \frac{\alpha+(1-a-\alpha)z^{-1}}{1} \cdot \frac{1}{1-az^{-1}} = \frac{Y(z)}{U(z)} = \frac{Y(z)E(z)}{E(z)U(z)} \tag{2.4}$$

Porovnáním získáme výsledné rovnice pro výstup a stavovou proměnnou:

$$y(k) = \alpha e(k) + (1-a-\alpha)e(k-1) \tag{2.5}$$
$$e(k) = u + ae(k-1) \tag{2.6}$$

Koeficienty byly zvoleny experimentálně s důrazem na opravdu pozvolný rozjezd a pouze drobný překmit, který filtr způsobuje. $\alpha = 0{,}4$, $T_1 = 0{,}5s$. [9]

Výsledná rovnice filtru je pak:

$$F_{(s)} = \frac{0{,}2s+1}{0{,}5s+1}$$

6.4 PLC program

PLC program je psán jak v jazyce Ladder Diagram tak v jazyce Structured Text, abychom demonstrovali možnost e!COCKPIT kombinovat různé jazyky mezi sebou. V jazyce LD jsou psány dvě POU. Jedno se nachází ve smyčce s intervalem 5ms a nachází se v něm regulátory a převody datových typů. Druhé POU je ve smyčce s intervalem 100ms a slouží k měření rychlosti z polohy. Jiný způsob měření rychlosti v modelu není implementován. Třetí POU je psáno v jazyce ST a obsahuje nastavování rychlosti z PS regulátoru do motoru přes A/D převodník, ošetřování chybových stavů, určení nulové polohy, potlačování šumu snímače polohy a obracení směru chodu motoru v závislosti na aktuální a žádané poloze. POU v jazyce ST je nadřazeno tomu v jazyce LD, protože se v něm nastavují hodnoty pro regulátory, které jsou v podřazeném POU.

Jelikož směr motoru nelze udávat záporným napětím, ale pouze kladným napětím a přehozením bitu z digitální výstupní karty musely být použity dva shodné regulátory polohy, každý pro jiný směr. Vstupem regulátoru rychlosti je pak výstup toho regulátoru polohy, v jehož směru se má jezdec pohybovat. Tento vstup je však ještě modifikován filtrem žádané hodnoty, vycházejícím ze vzorců 1.5 a 1.6. Do vzorců byly dosazeny následující proměnné z PLC programu:

y(k) = GVL.PS_Regulator.SET_POINT
u(k) = GVL.P_Regulator.Y a GVL.P_Regulator2.Y
e(k) = GVL.ek
e(k-1) = GVL.ek2
a = q
α = alfa

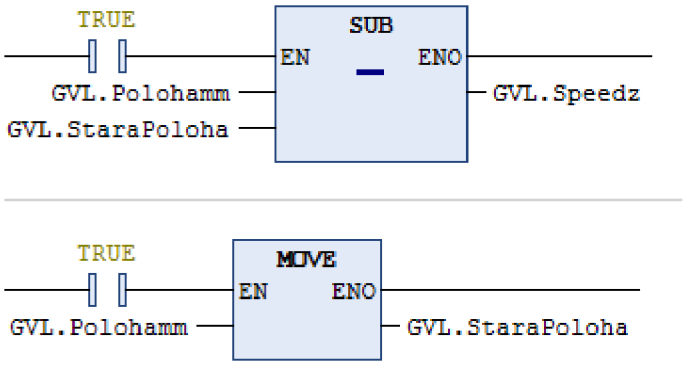
```
IF (GVL.Polohamm > GVL.ZadanaPoloha) THEN
Swap := FALSE;
GVL.ek := GVL.P_Regulator2.Y+q*GVL.ek2;
GVL.PS_Regulator.SET_POINT := alfa*GVL.ek+(1-q-alfa)*GVL.ek2;
ELSIF (GVL.Polohamm < GVL.ZadanaPoloha) THEN
Swap := TRUE;
GVL.ek := GVL.P_Regulator.Y+q*GVL.ek2;
GVL.PS_Regulator.SET_POINT := alfa*GVL.ek+(1-q-alfa)*GVL.ek2;
ELSIF (GVL.Polohamm = GVL.ZadanaPoloha) THEN
GVL.PS_Regulator.SET_POINT := 0;
END_IF
```

Obrázek 22 Ukázka kódu - Změna směru a filtrování vstupní hodnoty PS regulátoru

Tento problém by se dal řešit i jedním regulátorem rychlosti a zavedením fiktivních maximálních a minimálních hodnot výstupu regulátoru. Taktéž by šlo problém vyřešit

zavedením fiktivní aktuální polohy, ale možnost využít dvou regulátorů je nejelegantnějším řešením.

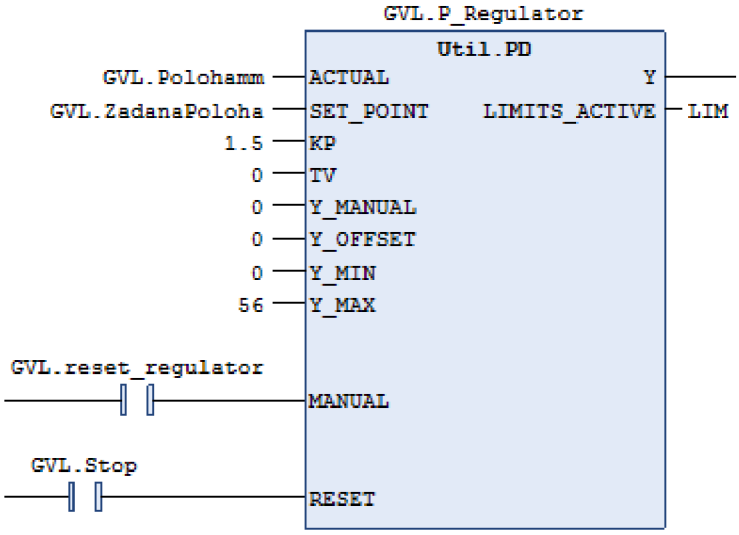
Měření rychlosti spočívá v přiřazení aktuální polohy na konci cyklu do pomocné proměnné a její odečtení od aktuální polohy na začátku následujícího cyklu. Perioda těchto cyklů byla vybrána 100ms, tudíž se výsledná hodnota musí vynásobit deseti, aby regulátor dostával hodnotu v řádech mm/s. Rychlejší perioda by znamenala menší přesnost a pomalejší, například sekundová, by znamenala pomalejší změnu hodnoty rychlosti pro PS regulátor.



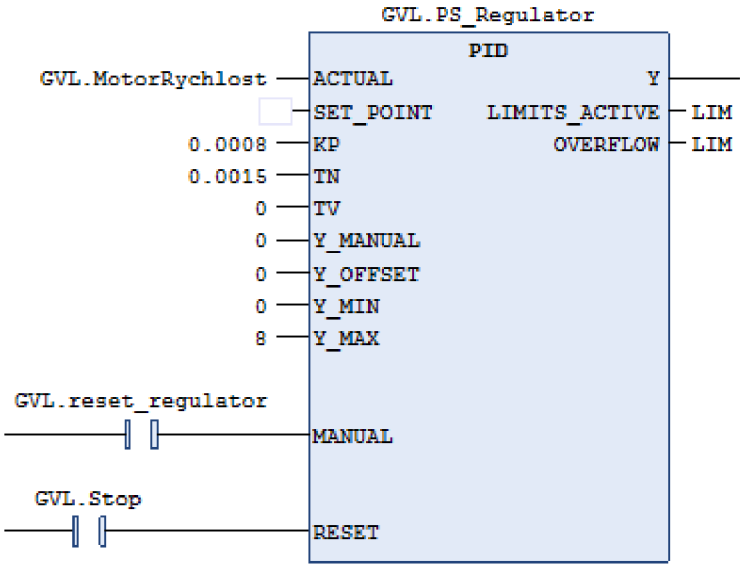
Obrázek 23 Ukázka kódu - Měření rychlosti z polohy

6.5 Implementace regulátoru

Regulátory byly implementovány do smyčky s intervalem 5ms, kde se nachází většina kódu. Byly pro ně využity vestavěné bloky z knihovny prostředí e!COCKPIT. Pro P regulátor byl použit PD regulátor s nulovou derivační složkou a pro PS regulátor byl použit PID regulátor s nulovou derivační složkou.



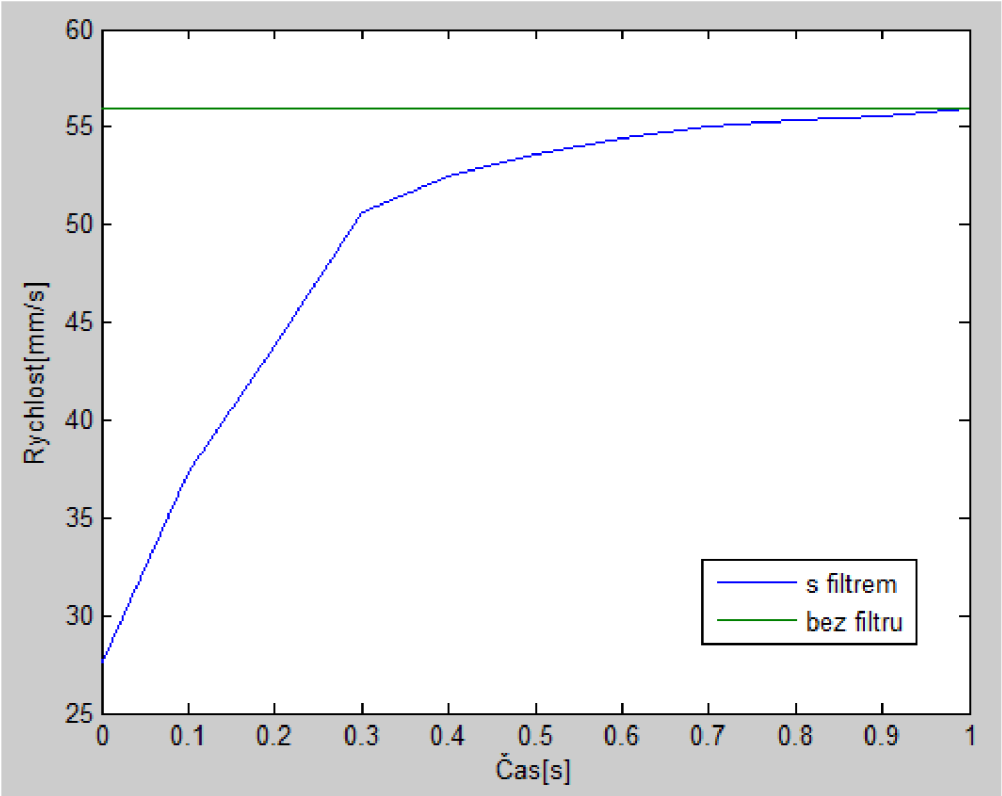
Obrázek 24 Nastavený blok PD regulátoru v programu



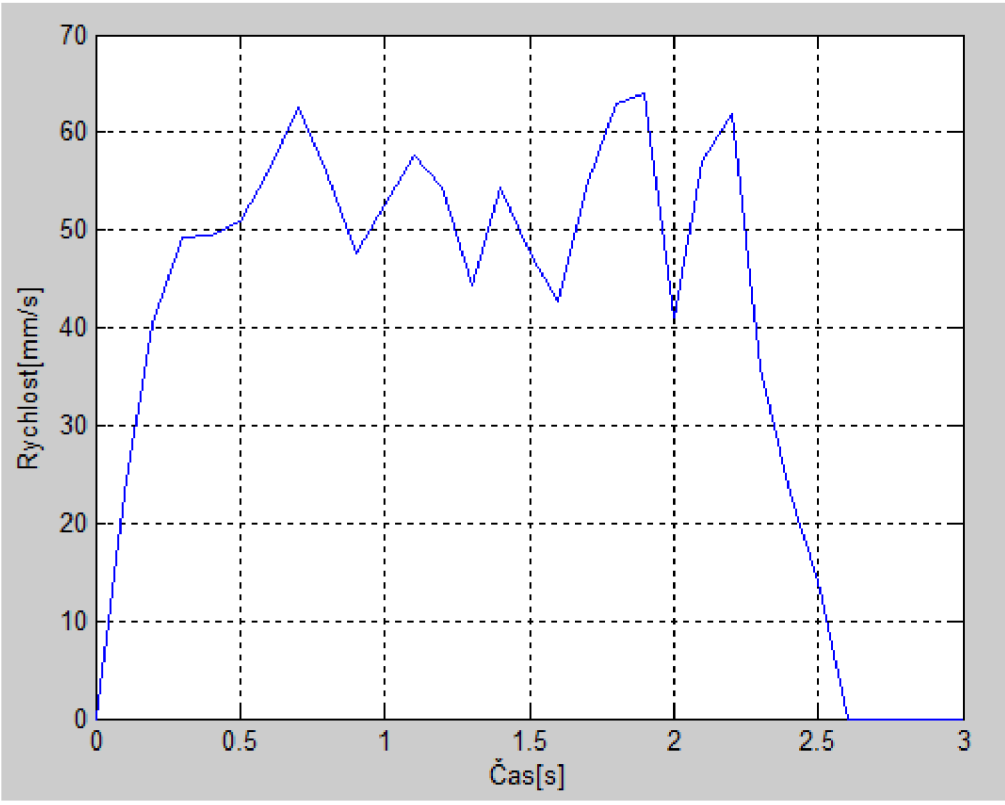
Obrázek 25 Nastavený blok PID regulátoru v programu

Koeficienty regulátorů byly nastaveny podle návrhu. Hodnota Y_MAX značící maximální výstup byla nastavena na 56(mm/s) pro P Regulátor jakožto změřená nejvyšší možná rychlost, kterou se je jezdec po ose schopen pohybovat. Pro PS Regulátor je tato hodnota 8(V). Motor začíná reagovat až na hodnotě vyšší než 2V. Tato hodnota je poté přičtena k výstupu regulátoru. Při výstupu regulátoru GVL.PS_Regulator.Y = 0 jsou do motoru z PLC posílány 2V čímž se motor stále ještě nepohybuje. Při maximálním výstupu regulátoru GVL.PS_Regulator.Y = 8 je do motoru z PLC posíláno 10V a motor jede na plné otáčky.

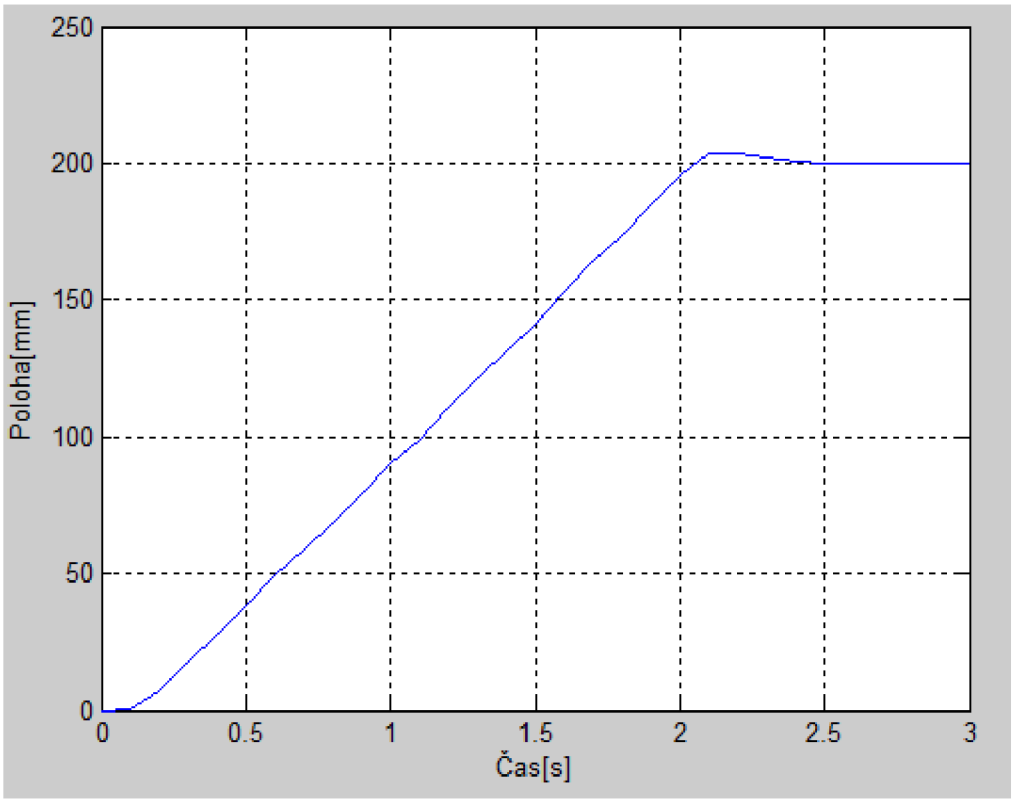
Regulátory byly vyzkoušeny pro změnu žádané polohy z 0 na 200mm a teoretické závislosti byly porovnány s naměřenými.



Obrázek 26 Závislost žádané vstupní hodnoty PS regulátoru na čase při regulaci polohy



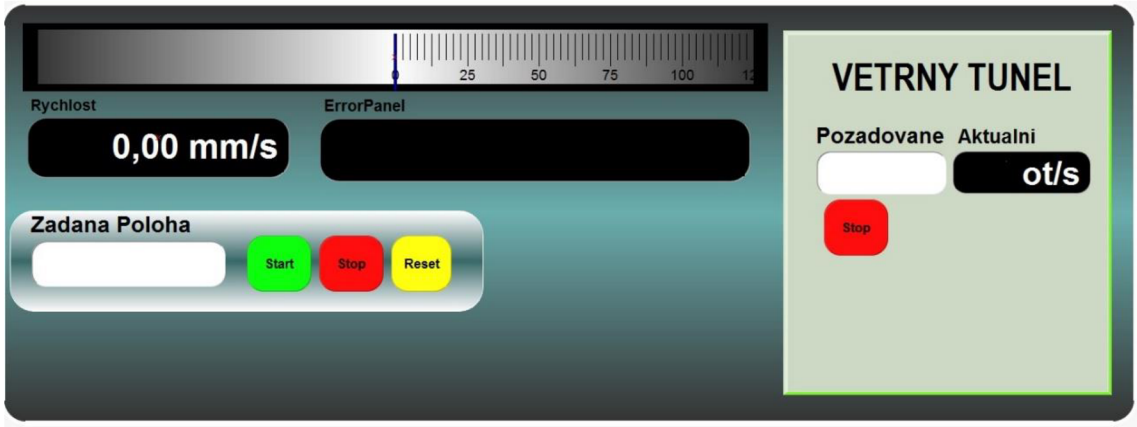
Obrázek 27 Změřená závislost rychlosti na čase při regulaci polohy



Obrázek 28 Změřená závislost regulace polohy

7 VIZUALIZACE

Vizualizace je vytvořena ve SCADA systému SCADA Laquis. Cílem bylo vytvořit jeden panel pro ovládání obou úloh současně. Volil jsem hlavní panel pro řízenou osu s malým barevně odlišeným přídavným panelem pro větrný tunel.



Obrázek 29 Společná vizualizace pro obě úlohy

V horní části panelu je znázorněna aktuální poloha jezdce. Pod ní se nachází aktuální rychlost motoru v milimetrech za sekundu a ErrorPanel, který vypisuje chyby, pokud nějaká nastane. Pokud uživatel nastaví polohu mimo omezení modelu, na panelu se objeví hláška, že se tak stalo a jezdec se nerozjede. Kolonka Zadana Poloha slouží uživateli k zapsání žádané polohy jezdce. Tlačítko Stop zastaví motor v pohybu. Motor do chodu poté uvede tlačítko Start. Tlačítko reset uvede jezdec do nulové polohy.

Panel větrného tunelu je umístěn vpravo od panelu pro řízenou osu. Obsahuje kolonku pro zadávání požadovaných otáček uživatelem a zobrazení aktuálních otáček větráku. Tlačítko Stop větrák vypíná.

Přes Modbus TCP jsou z a do PLC posílány následující proměnné, které jsou propojeny s grafickými objekty ve vizualizaci.

Tabulka 5 Proměnné posílané z PLC do SCADA systému přes Modbus TCP

Název	Datový typ	Adresa	Popis
AktPoloha	WORD	2	Aktuální poloha jezdce
ZadPoloha	WORD	32 008	Žádaná poloha zadávána uživatelem
Stop	BOOL	32 768	Zastavení motoru
Rychlost	REAL	4	Rychlost motoru v mm/s
MVetZad	WORD	32 001	Žádané otáčky zadávány uživatelem
RPS	WORD	3	Aktuální hodnota otáček větráku v ot/s

Adresy registrů proměnných, do kterých se hodnoty zadávají z vizualizace začínají na adrese 32 000. Pro adresy registrů proměnných pouze pro čtení začínají na adrese 0. Adresa proměnných datového typu bool (coil) začíná na adrese 32 768.

8 ZÁVĚR

Prvním cílem bakalářské práce byla rešerše volně dostupných SCADA systémů. Vybráno bylo sedm nejpopulárnějších free systémů, které byly zhodnoceny dle daných kritérií. Nejlépe z těchto systémů obstála SCADA Laquis, která byla také použita pro společnou vizualizaci obou programů.

Důležitou součástí práce bylo seznámení se s modelem větrného tunelu a řízené osy a jejich modifikací pro naši úlohu a také seznámení s PLC od firmy WAGO PFC200 CS 2ETH RS a jejich vzájemných propojením. Model řízené osy byl bez problému připraven k použití včetně všech jeho součástí. Model větrného tunelu musel být upraven pro požadavky zadaní.

Dalším krokem bylo seznámení se s vývojovým prostředím e!COCKPIT firmy WAGO. Po prozkoumání funkcí prostředí e!COCKPIT, byl napsán PLC program v jazyce Ladder Diagram, který slouží k zápisu hodnot veličin v čase. Tuto potřebu by měl řešit vizualizační prvek Trend, který se nepodařilo zprovoznit. Tento program posloužil k proměření soustav modelů. Z naměřených hodnot byla provedena identifikace přechodových funkcí daných systémů.

Pro soustavu modelu větrného tunelu byl navržen PS regulátor. Vyregulování žádané hodnoty je možné pro otáčky v rozsahu 16 – 27ot/s. Pro nižší otáčky je regulace zatížena chybou způsobenou tíhou magnetu. Model by bylo možné vylepšit výměnnou feritového magnetu, třeba za neodymový. Taktéž se jedná o turbulentní proudění vzduchu a měřící větrák nezachytne vše, což se u velmi nízkých otáčkách projevuje více a regulovaná veličina tím pádem velmi kmitá. Rychlejší a kvalitnější regulace by mohlo být dosaženo implementací více magnetů do modelu, které by zpřesnili měření otáček a umožnili by regulátor zapojit do rychlejší časové smyčky, než je nynější s intervalem 1s.

Pro soustavu modelu řízené osy byl navržen P regulátor pro polohu a PS regulátor pro rychlost. Vyregulování žádané hodnoty polohy je bezproblémové, šum potenciometru, který měří polohu jezdce však musí být řešen podmínkou v programu, která vypíná jeho funkci při dosažení žádané hodnoty polohy. Rychlost má díky filtru žádané hodnoty plynulý náběh a plynulý dojezd. Vlivem filtru však vzniká drobný překmit v rozmezí 1-3%. Modelu chybí možnost spolehlivě měřit otáčky motoru. Přepočet rychlosti ze snímače polohy není ideální a proto se může zdát, že hodnota rychlosti kmitá, ikdyž je na pohled rovnoměrná.

Vizualizace vyvinutá v SCADA systému SCADA Laquis byla vytvořena přehledně a pro co nejjednodušší používání. Uživatel má možnost řídit oba modely zároveň na jednom barevně odlišeném panelu.

Oba programy jsou použitelné v praxi pro skutečné větrné tunely a zařízení využívající řízenou osu na stejném principu. Díky jednoduchosti kódu v jazycích LD a STL jsou taktéž lehce modifikovatelné a umožňují uživateli přidání dalších možností při změně některého z modelů.

Literatura

[1] VLACH, Jaroslav. *Řízení a vizualizace technologických procesů*. 1. vyd. Praha: BEN - technická literatura, 1999. ISBN 80-86056-66-X.

[2] Mii: Moravské přístroje, a.s. [online]. [cit. 2016-14-12]. Dostupné z: http://www.mii.cz/art?id=380&cat=146&lang=405

[3] Pantek: *Výrobní inteligence v průmyslové automatizaci* [online]. [cit. 2016-14- 12]. Dostupné z: http://www.pantek.cz/index.php

[4] Siemens: *Vizualizační software* [online]. [cit. 2016-14-12]. Dostupné z: http://www1.siemens.cz/ad/current/index.php?vw=0&ctxnh=7698ba9221&ctxp=home

[5] *Wago-I/O-system 750 manual*. Wago [online]. WAGO Kontakttechnik GmbH & Co., 2016 [cit. 2017-01-05]. Dostupné z: http://www.wago.us/download.esm?file=%5Cdownload%5C00375232_0.pdf&name=m07508202_xxxxxxxx_0en.pdf

[6] *e!COCKPIT: Vývojové prostředí zaměřené na uživatele* [online]. Brno: WAGO-Elektro spol. s r.o., 2015, , 1 [cit. 2017-05-01]. Dostupné z: http://www.wago.cz/aktuality/tiskove-zpravy/press-releases-dateils-22272.jsp

[7] *e!COCKPIT Engineering Software — invitation to discover* [online]. WAGO Kontakttechnik GmbH & Co., 2015, , 1 [cit. 2017-05-01]. Dostupné z: http://global.wago.com/en/products/new-items/overview/engineering-software.jsp

[8] *e!COCKPIT manual*. Wago [online]. WAGO Kontakttechnik GmbH & Co., 2016 [cit. 2017-01-05]. Dostupné z: http://www.wago.us/download.esm?file=%5Cdownload%5C00375232_0.pdf&name=m07508202_xxxxxxxx_0en.pdf

[9] VELEBA, Václav. *Číslicová řídicí technika:počítačová cvičení*. VUT v Brně, skriptum, 2005

SEZNAM ZKRATEK

SCADA - Supervisory Control And Data Acquisition

HMI – Human machine interface

PLC – Programmable logic controller

LD – Ladder diagram – programovací jazyk

ST – Structured text – programovací jazyk

DI – digital input

DO – digital output

AI – analog input

AO – analog output

ot/s – otáčky za sekundu

mm – milimetr

s – sekunda

V – volt

Seznam příloh

Příloha 1. CD s elektronickou verzí bakalářské práce, zdrojovými kódy PLC programu a vizualizací.