

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Návrh vizualizačního systému technologického procesu v prostředí LabVIEW

BAKALÁŘSKÁ PRÁCE

Lukáš Srsen

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R001 - Elektronické informační a řídicí systémy

Pracoviště: Ústav řízení systémů a spolehlivosti

Fakulta mechatroniky a mezioborových inženýrských studií

Technická Univerzita v Liberci

Studentská 2, 461 17, Liberec 1

Vedoucí práce: Ing. Jiří Bažant

Konzultant: Ing. Jaroslav Vlach, PRECIOSA, a.s. Jablonec n.N.

Datum a místo vydání: 9.5. 2008 v Liberci

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/200 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že Technická univerzita v Liberci má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že souhlasím s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem Technické univerzity v Liberci, která má právo požadovat ode mne přiměřený příspěvek na úhradu nákladů vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury pod vedením školitele.

V Liberci 9.5. 2008

.....
Lukáš Srsen

Poděkování

Především bych chtěl poděkovat panu Ing. Jaroslavu Vlachovi, Josefu Havlíčkovi a Karlu Dusilovi z firmy PRECIOSA, a.s. za poskytnuté rady a pomoc při vypracovávání práce. Dále bych chtěl poděkovat vedoucímu práce Ing. Jiřímu Bažantovi a v neposlední řadě svým rodičům, kteří mě ve studiu a psaní této práce nemalou měrou podporovali.

Abstrakt

Náplní práce je seznámení s problematikou vizualizačních systémů a s pojmy, které s ní souvisí. Práce je prakticky zaměřena na tvorbu vizualizačního systému technologického procesu v grafickém vývojovém prostředí LabVIEW. Vizualizační systém představuje řídicí systém compactRIO, ethernet a osobní počítač. Dále se práce zabývá popisem bezdrátové technologie a otázkou jejího využití v oblasti průmyslu. Pomocí praktického testování bezdrátové technologie Wawenis popisuje vliv rušení běžného provozu na přenos informací.

Klíčová slova: Vizualizace, LabVIEW, compactRIO, fieldbus, Wawenis

Abstract

The thesis provides a comprehensive information on visual systems and related terminology. From the practical point of view it is focused on visualisation of a technological process with the graphical software LabVIEW. The visualisation system consists of a control system compactRIO, ethernet and a PC. In addition, the thesis also describes a wireless technology and its application in industry – the information transfer through a wireless technology Wawenis is tested in standard conditions of an industrial plant.

Keywords: visualisation, LabVIEW, compactRIO, fieldbus, Wawenis

Obsah

Úvod.....	8
<u>Kapitola 1.</u> Vývojové prostředí LabVIEW	9
1.1 Vývoj.....	9
1.2 Co je „VI“ a práce s ním	10
1.3 Funkce, ovládací prvky a datové typy.....	12
<u>Kapitola 2.</u> Vlastnosti komunikačních rozhraní používaných v průmyslu	17
2.1 Vývoj.....	17
2.2 Jednotlivé typy	17
2.2.1 CAN.....	17
2.2.2 Profibus	19
2.2.3 AS-i	21
<u>Kapitola 3.</u> Projekt vizualizačního systému modelového technologického procesu	23
3.1 Pojem vizualizační systém	23
3.2 Řídicí systém.....	24
3.3 Popis programu	24
3.4 Řešené problémy.....	28
<u>Kapitola 4.</u> Bezdrátové spojení v průmyslu.....	31
4.1 Vývoj.....	31
4.2 Bluetooth.....	31
4.3 IEEE 802.11	33
4.4 Wawenis.....	34
4.5 Využití technologie WAWENIS.....	35
4.5.1 Popis testovaných modulů.....	35
4.5.2 Testování teplotního čidla Wawetherm Dallas 18S20	36
Závěr	40
Literatura.....	42
Příloha A	43
Příloha B	44
Příloha C	45

Úvod

Hlavním tématem práce je vizualizace dat pomocí LabVIEW. K pojmu vizualizace se váží další důležité pojmy, které je nutné znát a dokázat je popsat. Tato práce je rozdělena na část teoretickou, ve které budou popsány důležité pojmy, a praktickou. Pro naplnění jejího obsahu bude zvoleno doplňkové téma, které je úzce spojeno s jedním z probíraných pojmů.

Prvním úkolem práce bude popsání grafického vývojového prostředí LabVIEW. Zmíněnému úkolu bude věnována první kapitola, která bude uvozena popisem vývoje daného prostředí. Dále zde bude popsáno samotné uživatelské rozhraní a práce s ním. V poslední části první kapitoly budou popsány nejčastěji používané funkce, ovládací prvky a datové typy.

Další kapitola vysvětluje pojem komunikační rozhraní. Zabývá se nejdůležitějšími vlastnostmi komunikačních rozhraní, které budou popsány u třech různých typů standardně používaných v průmyslu.

Cílem třetí kapitoly bude naplnění hlavního tématu práce vizualizace modelového technologického procesu. V první části si nejdříve vysvětlíme pojmy vizualizační a řídicí systém. Dále se práce bude zabývat podrobným popisem programu a možným řešením vyskytnutých problémů.

Poslední kapitola bakalářské práce je věnována vývoji bezdrátové technologie a základním vlastnostem, které budou popsány na třech různých typech. V rámci práce bude provedeno praktické testování zvolené bezdrátové technologie v provozu.

Kapitola 1. Vývojové prostředí LabVIEW

1.1 Vývoj

Laboratory Virtual Instrumentation Engineering Workbench (dále LabVIEW) je grafické vývojové prostředí pro měření a automatizaci. Bylo představeno firmou National Instruments (dále NI)[5.] v roce 1986 a od té doby prošlo více jak dvacetiletým vývojem. Původní záměr byl zrealizovat jednoduchý nástroj pro programování měřících přístrojů, ale z LabVIEW se po mnoha inovacích vyvinula všeobecná platforma pro návrh, měření a řízení. LabVIEW si postupem času získalo i spoustu nových uživatelů a dnes se může chlubit velmi početnou aktivní uživatelskou základnou, která pokrývá širokou řadu oborů a aplikací. V dnešní době se dá říci, že se jedná o nejpopulárnější nástroj pro sběr dat a řízení přístrojů. I s ohledem na popularitu a oblíbenost je však málo uživatelů, kteří dokáží pochopit a využít celé spektrum možností, které LabVIEW nabízí. Spousta uživatelů si pokládá otázku jestli je LabVIEW skutečný programovací jazyk. LabVIEW je plnohodnotný programovací jazyk se stejnými programovými strukturami, jaké jsou v abstraktních textových jazycích.

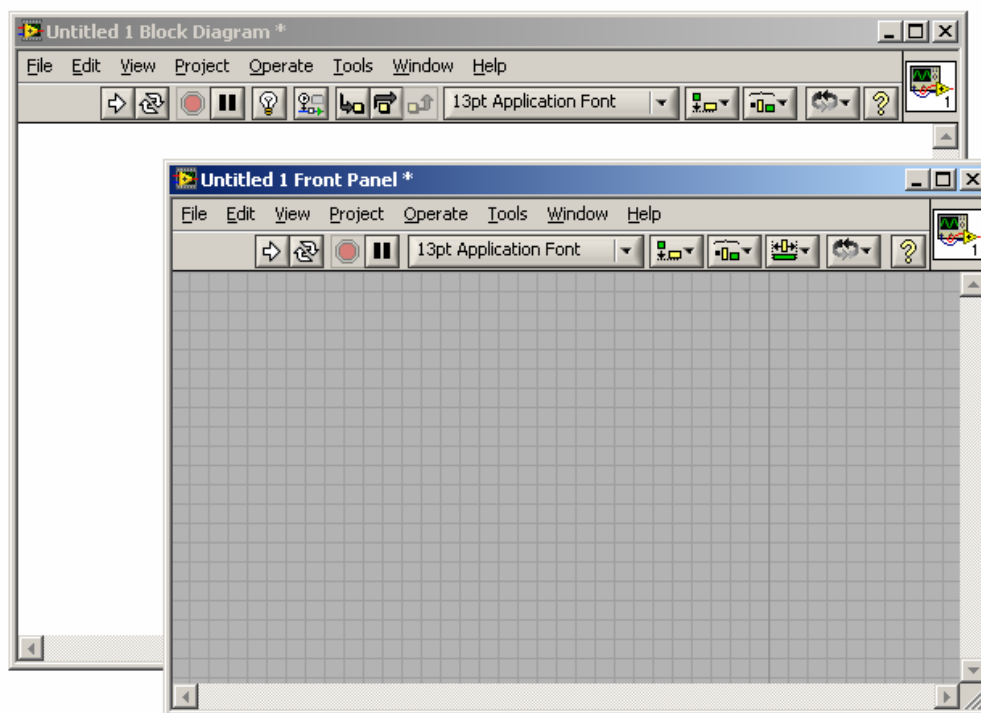
Již základní verze LabVIEW nabízí unikátní možnosti při programování paralelních procesů a pro běh programů v reálném čase. Vše umocňuje jednoduchost a rychlost jejich naprogramování. S rostoucím počtem jader v procesorech a kontrolérech s FPGA poli je tento směr programování paralelních procesů velmi vítán. Od verze 8.20 je LabVIEW rozšířeno také o objektové programování. I když se LabVIEW tváří jako nedostupné a nekompatibilní s jinými programovacími jazyky, tak opak je pravdou. Je běžné volat kód z dynamické knihovny dll nebo do sdílené knihovny integrovat externí kód a aplikace prostřednictvím softwarových technologií jako jsou např. ActiveX nebo .NET. Od verze 8.20 je zavedení knihovny dll jednodušší. Byl přidán průvodce, který pomocí hlavičkového souboru a samotné knihovny dll vygeneruje pro každou funkci obsaženou v knihovně vlastní virtuální nástroj (dále VI). Stejně lehce se dá celý postup udělat i opačně. VI lze lehce proměnit na dynamickou knihovnu dll a zpřístupnit ji programátorům ostatních programovacích jazyků.

Dalším parametrem je rychlost interpretování napsaného kódu. Jelikož je grafický kód LabVIEW kompilován přímo do strojového kódu, je jeho zpracování daleko rychlejší než u jiných interpretovaných jazyků. Během vývoje LabVIEW se

dokonce objevily nové možnosti uplatnění a to zejména v komunikaci s novými typy sběrnic. Na základě toho byl představen i FPGA modul, který umožňuje v LabVIEW napsat kód pro hradlová pole. Kód pak běží přímo v křemíku, čímž je dosaženo dosud nevídané rychlosti. Odpadá tak zdoluhavé a komplikované psaní programu v jazyce VHDL.

1.2 Co je „VI“ a práce s ním

VI je zkratka z anglického dvojslovného pojmu Virtual Instrument, v češtině virtuální nástroj. Tento pojem nahrazuje klasické slovo program, které zná každý programátor z jiných programovacích jazyků. Programy v LabVIEW svým vzhledem a činnostmi připomínají klasické přístroje, jako jsou např. osciloscopy a multimetry.



Obr.1.1 Front panel a block diagram

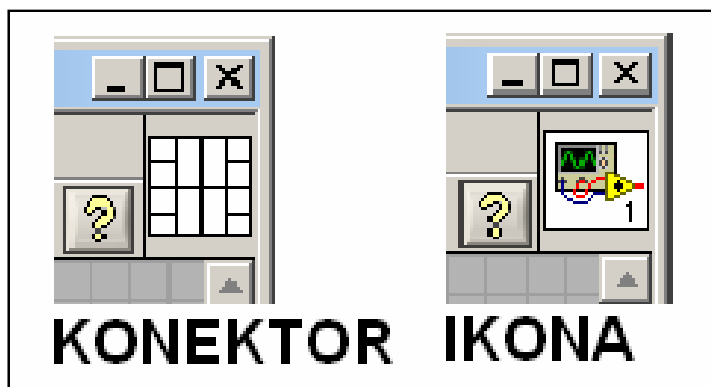
Při práci s VI využíváme dvou oddělených ploch, zobrazených na obr. 1.1. První plochou je front panel. Je to uživatelské rozhraní. Na front panel se umísťují ovládací prvky. Ty jsou přístupné při stisku pravého tlačítka myši na ploše front panelu. Vkládáme je snadným způsobem, přetažením z menu na plochu front panelu. Grafické zpracování by mělo být přehledné a umožnit uživateli se rychle orientovat při změnách v zobrazování. Front panel řídí programový kód, který zapisujeme resp. kreslíme do



Obr.1.2 Paleta nástroje

diagramu. Zápisem resp. kreslením kódu rozumíme propojení různých terminálů, které jsou umístěny na front panelu, či jiných bloků, které reprezentují funkce nebo subVI. V diagramu můžeme vkládat různé funkce, které jsou přístupné, stejně jako ovládací prvky na front panelu, při stisku pravého tlačítka myši. Stejně je i jejich umisťování, stačí blok pouze přetáhnout z menu na plochu diagramu. Při práci s bloky na front panelu a diagramu využíváme paletu tools, která je na obr. 1.2. Tato paleta je přístupná po stisku kombinace SHIFT + pravé tlačítko myši.

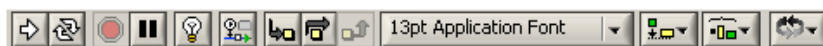
Pokud chceme VI použít v jiném VI, nazýváme ho subVI. Je důležité nastavit jeho vstupy a výstupy, tak aby byly v nadřazeném VI viditelné. To se provádí pomocí ikony, která je umístěna v pravém horním rohu front panelu a diagramu. Ikonu si lze dle libosti upravit. Pokud na ní na front panelu klikneme pravým tlačítkem myši objeví se



Obr. 1.3 Detail Ikony a Conectoru

pop-up menu, ve kterém vybereme položku show conector. Místo ikony se zobrazí konektor. Jeho vzhled můžeme ovlivnit pomocí volby patterns. Označením pole na konektoru a následným kliknutím levým tlačítkem myši na objekt na front panelu provedeme spojení. To provedeme pro všechny objekty, které chceme používat v nadřazeném VI. Zvykem je umísťovat do levé části vstupní signály a do pravé části signály výstupní. LabVIEW nabízí i tzv. express VI, které je konfigurovatelné pomocí dialogového okna. V diagramu se zobrazuje jako modrý obdélník, ve kterém se zobrazuje příslušná ikona a název. Stejně jako každá ikona má své vstupy a výstupy.

Samotnou aplikaci lze spustit v samotném LabVIEW nebo ji zkompileovat a provozovat samostatně. Spuštění aplikace se provádí pomocí tlačítek umístěných v horní části front panelu a diagramu, zobrazených na obr. 1.4. Pokud je spouštěcí šipka



















Obr. 1.4 Spouštěcí tlačítka

přerušená je v aplikaci chyba a nelze ji spustit. Pokud šipku stiskneme automaticky se nám daná chyba zobrazí. Pokud je aplikace bez chyb šipka je nepřerušená a při jejím stisku se aplikace spustí. Jestliže chceme aplikaci spustit cyklicky použijeme ikonu se stočenými šipkami. Když nemáme v aplikaci ukončovací tlačítko, lze ji zastavit stisknutím ikony stop. Pro odlaďování můžeme umístit do aplikace breakpointy nebo zobrazovače aktuálních stavů a nebo aplikaci i krokovat pomocí série ikon se šipkou a obdélníkem.

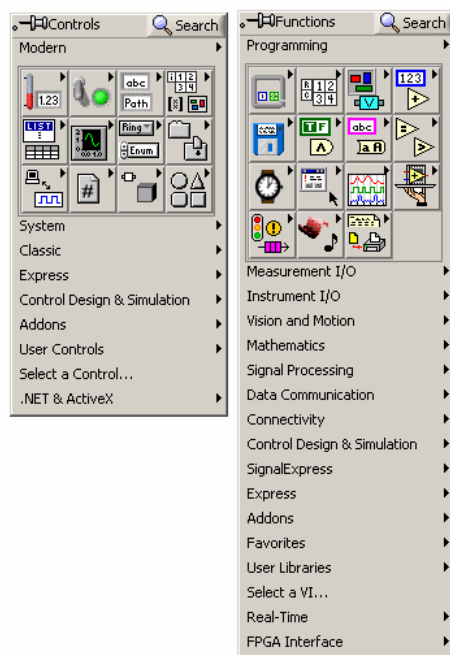
1.3 Funkce, ovládací prvky a datové typy

Jak bylo zmíněno v předchozí kapitole v LabVIEW se pracuje na dvou možných plochách. Každý ovládací prvek umístěný na front panelu je v diagramu reprezentován tzv. terminálem. Každý terminál představuje programovou proměnou, které je navíc v grafickém prostředí LabVIEW přiřazena příslušná barva. Jednotlivé terminály jsou od sebe odlišeny nejen barvou, ale i tloušťkou obvodové stěny a umístěním šipky. Pokud se jedná o řídicí terminál je šipka vpravo a obvodová stěna je ohraničena tlustou čarou. Naopak je to u terminálu indikačního, kde je obvodová stěna tvořena tenkou čarou a šipka je umístěna v levé části terminálu. Pokud v diagramu použijeme blok, který má více vstupů či výstupů, je každý označen šipkou a příslušnou barvou, dle datového typu. Při spojování jednotlivých terminálů či bloků je vodič reprezentován stejnou barvou

Control	Indicator	Data Type
		Single-precision, floating-point numeric
		Double-precision, floating-point numeric
		32-bit signed integer numeric
		32-bit unsigned integer numeric
		Boolean
		String
		Cluster
		Path

Obr. 1.5 Používané datové typy

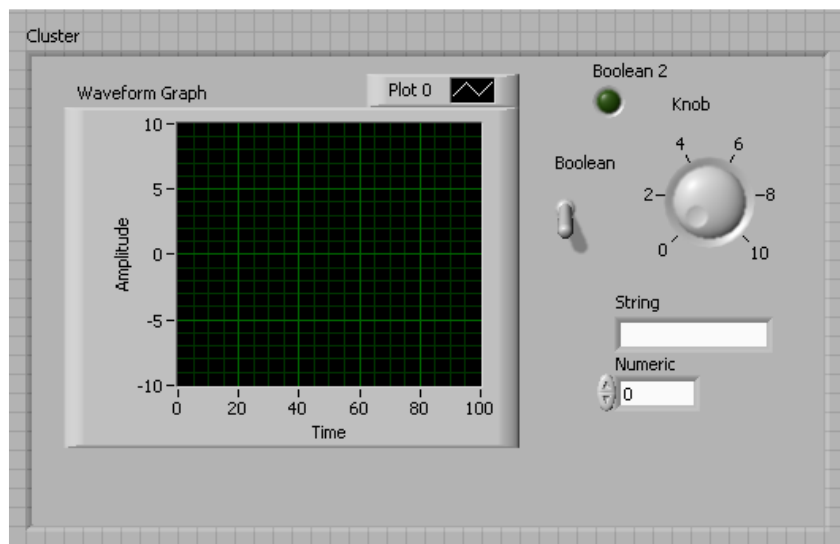
jako výstup a vstup, který spojuje. Je možné spojit i různě barevné terminály. Před vstupem jinak barevného terminálu je pak červené kolečko, které označuje změnu datového typu. Této vlastnosti však nejde využít u všech terminálů či bloků. Seznam použitých datových typů a jim příslušných barev je zobrazen v tabulce na obr. 1.5. Dále proměnné a signály dělíme na skalární a vektorové. LabVIEW umožňuje až 5-ti rozměrné vektorové proměnné či signály.



Obr.1.6 Knihovny controls a functions

Knihovna s ovládacími prvky (obr. 1.6), anglicky Controls, obsahuje veškeré prvky používané na front panelu. Knihovna obsahuje další podknihovny. Podknihovna Numeric obsahuje prvky numerického typu. Dále je dělíme na digitální (digital control a digital indicator), analogové prvky lineární (tank, thermometer, vertical a horizontal

slide), analogové prvky kruhové (knob, dial, meter, gauge) a prvky pracující s barvou (color box, color ramp). U všech prvků můžeme měnit jejich vlastnosti, většinou se od sebe liší pouze vzhledem. Můžeme jim přiřazovat i fyzikální jednotky, které vyvoláme



Obr. 1.7 Ukázka ovládacích prvků

pomocí Show Unit Label. Další podknihovnou je knihovna Boolean. Knihovna sdružuje všechny objekty s charakteristikou boolean. Prvky jako jsou Switch nebo Button je lépe použít pro controler a prvky jako jsou LED nebo Light na indicator. Navíc v módu control lze nastavit režim přepínače pomocí volby Mechanical Action. Podknihovna String and Table obsahuje objekty string control a string indicator, u kterých lze nastavit formát textu (normal display, codes display, password display a hex display). Objekt Table je dvourozměrné pole proměnných string. Další podknihovnou je Array and Cluster. Objekt typu array je při umístění na front panel prázdný. Pro práci s polem je důležité osadit ho prvkem požadovaného typu. Typ Cluster je struktura obsahující položky různého typu. LabVIEW navíc rozlišuje cluster pouze číselného typu a cluster s položkami obecného typu. Při vložení na front panel je stejně jako objekt typu array prázdný a pro práci s ním je nutného ho naplnit požadovanými typy. V knihovně jsou i dva předdefinované clustery Error In a Error Out. Podknihovna, která slouží pro práci s grafy, se nazývá Graph. Obsahuje spoustu typů grafů, jak pro práci ve 2D tak i 3D. Nejčastěji se používají tři typy grafů Waveform Chart, Waveform Graph a XY Graph. První dva se liší pouze podle zpracování hodnot a to tak, že Waveform Chart zpracovává data po jednom jak přicházejí na vstup, na rozdíl od Waveform Graph, který pracuje s bloky dat. Výhodou XY Graph je možnost nastavení i osy X. Knihovna

Pro samotné zpracování signálu v diagramu používáme různé funkce. Knihovna Functions (obr. 1.5), která je přístupná přes pravé tlačítko myši na ploše digramu, obsahuje veškeré dostupné funkce, které LabVIEW nabízí. První podknihovnou, o které se zmíním, je Structures. Jedná se o cykly jako jsou např. while, for, ale i podmíněné smyčky jako je např. case nebo event. Dále obsahuje MathScript node nebo Formule node, kde je možno zapisovat vzorce podle syntaxe programovacího jazyku C. Následující podknihovny Array, Boolean, Numeric, Cluster and Variant, String a Waveform slouží k práci s daty a zpracování signálů od ovládacích prvků umístěných na front panelu. Ve směs umožňují nastavovat, upravovat hodnoty zobrazované nebo snímané z ovládacích prvků. Například v knihovně boolean jsou umístěny veškeré logické funkce And, Or, Xor, Not, ale i funkce pro převod logické pravdy či nepravdy na numerické hodnoty 1 resp. 0. V podknihovně Numeric nalezneme veškeré aritmetické operace a funkce na převod typů např. mnou později hojně využívanou



funkci Flatten To String a k ní opačnou funkci Unflatten From String. Další neméně důležitou podknihovnou je Timing. Zde jsou obsaženy veškeré funkce pro práci s časem jako např. zpoždění, časová známka, zjištění systémového času. Posledními podknihovnami o kterých se zmíním jsou Protocols a FPGA interface. FPGA interface je podknihovna, která slouží pro práci s FPGA polem resp. umožňuje volání aplikace, která s FPGA polem komunikuje, dále dovoluje měnit veškeré nadefinované vstupní hodnoty ve volané aplikaci a přijmout data z této aplikace. Poslední podknihovnou je Protocols. Tato podknihovna je součástí podknihovny Data Communication, která navíc ještě obsahuje prvky sdílené, lokální a globální proměnné. V podknihovně Protocols jsou umístěny veškeré typy protokolů a funkce k jejich používání. Knihovna Functions obsahuje spoustu dalších podknihoven pro práci s měřicími přístroji, pro zápis dat do souborů, pro zobrazování uživatelských zpráv, pro práci se signálem nebo podknihovnu pro práci s real-time aplikacemi. A v neposlední řadě, stejně tak jako u knihovny Controls, obsahuje i User Libraries, kam si uživatel může ukládat vlastně vytvořené funkce a mít tak k nim rychlejší přístup při programování.

Kapitola 2. Vlastnosti komunikačních rozhraní používaných v průmyslu

2.1 Vývoj

Technologie průmyslových komunikačních rozhraní se objevila v 80. letech minulého století. Často je označována anglickým slovem fieldbus, které nejlépe vystihuje význam těchto systémů. Hlavní rozšíření přišlo počátkem let 90. Jako první standardní řešení byla sériová sběrnice BITBUS, která byla představen firmou Intel.

Dnes je vývoj v tomto směru o poznání dál. Je celá škála nově navržených a akceptovaných řešení, která jsou optimalizována pro různé třídy úloh, ale i obecně pro většinu úloh z oblasti průmyslové automatizace a sběru dat. Velkou výhodou průmyslových komunikačních rozhraní je číslicová komunikace, která umožňuje detekci chyb přenášených dat prostřednictvím detekčních kódů a následného opakování přenosu.

Další možnost je získání několika různých hodnot měřené veličiny resp. veličin z jediného uzlu. Vzhledem k obvyklým topologiím nabízejí komunikační rozhraní jednodušší a kratší kabeláže a vyšší flexibilitu při rozšiřování či změnách. Komunikační rozhraní mají i své nevýhody. Hlavní je omezená dostupnost hodnot měřených veličin, které jsou vzhledem k vlastnostem komunikace diskretizovány v čase. Další nevýhodou je u některých typů komunikačních rozhraní vyšší pořizovací cena. Do práce byly vybrány jedny z nejznámějších průmyslových komunikačních rozhraní CAN, Profibus a AS-i. Vzhledem k rozsahu práce bylo snahou vystihnout jen ty nejhlavnější vlastnosti.

2.2 Jednotlivé typy

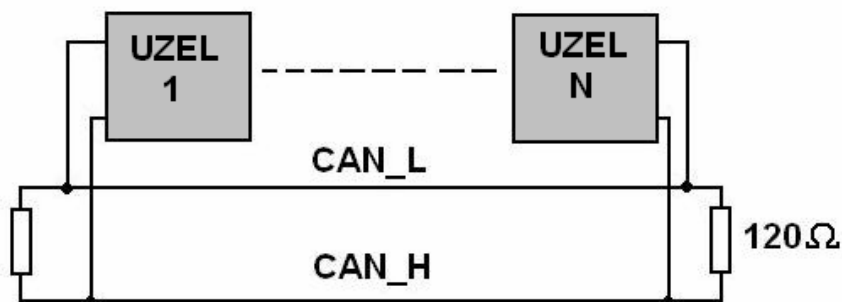
2.2.1 CAN

Controller Area Network zkr. CAN je sériový komunikační protokol, který byl původně vyvinut firmou BOSCH pro nasazení v automobilovém průmyslu. K velkému rozšíření sběrnice CAN pomohlo i samotné implantování tohoto komunikačního protokolu do integrovaných obvodů, zejména do mikroprocesorů, již ve výrobním procesu. Samotná specifikace definuje pouze protokol linkové vrstvy. Ostatní vrstvy mají mnoho variant např. fyzická vrstva má specifikace ISO 11898, ISO 11519 a J1939 atd.. Aplikační vrstvy pak známe CANopen, DeviceNet atd..

CAN je navržen tak, aby umožnil distribuované řízení systému v reálném čase

s přenosovou rychlostí až 1Mb/s a to při vysokém zabezpečení přenosu vůči vnějšímu rušení. Jedná se o protokol typu multi-master, kde je pro řízení přístupu k médiu použita sběrnice s náhodným přístupem, která řeší kolize na základě prioritního rozhodování. Jednotlivé zprávy neobsahují žádnou informaci o cílovém uzlu, kterému jsou určeny, ale jsou přijímány všemi uzly na sběrnici. Každá zpráva obsahuje identifikátor, který definuje obsah a prioritu zprávy. Zpráva, která má větší prioritu v případě kolize, je doručena přednostně.

V dnešní době se pro realizaci fyzické vrstvy využívá norma ISO 11898. Podle této normy sběrnice obsahuje dva vodiče označované CAN_H a CAN_L. Podle standardu protokolu CAN jsou definovány dvě vzájemně komplementární hodnoty bitů a to dominantní a recesivní. Pro normu ISO 11898 jsou tyto hodnoty definovány

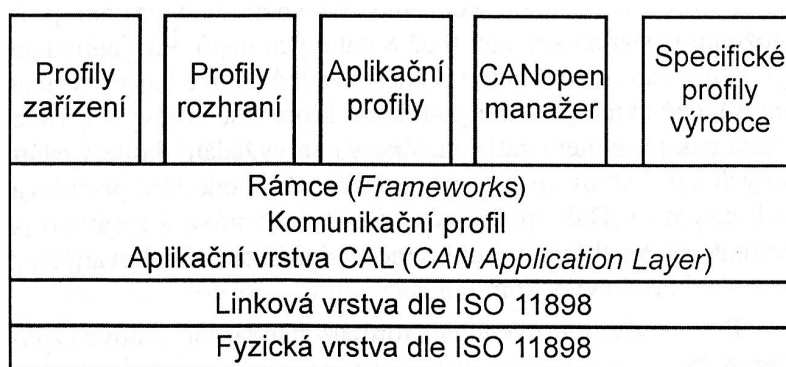


Obr. 2.1 Spojení jednotlivých uzlů

rozdílovým napětím na vodičích CAN_H a CAN_L (obr. 2.1). Recesivní úroveň představuje napětí 0V a pro dominantní úroveň hodnotu rozdílu 2V. Na samotnou sběrnici může být teoreticky připojeno neomezený počet uzlů. Při ohledu na zachování jejich statických a dynamických parametrů norma uvádí jako maximum 30 uzlů. Maximální délka při zachování maximální přenosové rychlosti 1Mb/s je 40m. Pro jiné přenosové rychlosti norma maximální vzdálenost neudává. Linková vrstva protokolu CAN je tvořena dvěma podvrstvami MAC a LLC. Podvrstva MAC (Medium Access Control) má na starosti řízení přístupu, kódování dat, vkládání doplňkových bitů do komunikace, detekci a hlášení chyb a potvrzování správně přijatých zpráv. Druhá vrstva LLC (Logical Link Control) filtruje přijaté zprávy a hlásí přetížení sběrnice. K odhalení chyb jsou použity základní mechanismy, které jsou v činnosti současně. Jedná se o monitoring, kdy se porovnává vysílaná hodnota s hodnotou na sběrnici. Další mechanismus je CRC kód. Následným mechanismem je vkládání bitů, které znamenají,

že , při vysílání pěti stejných bitů je vložen bit opačné úrovně. Kontrola formátu zprávy, kontroluje zda je na dané pozici nepovolená hodnota. Posledním mechanismem je potvrzení přijetí zprávy.

Závěrem bych se ještě zmínil o vrstvě aplikační. Zde existují desítky protokolů, které nejsou navzájem kompatibilní. Nejrozšířenějším protokolem v oblasti průmyslové automatizace je CANopen. Struktura tohoto standardu je relativně složitá, ve zjednodušené formě ji ukazuje obr. 2.2.



Obr. 2.2 Zjednodušená forma CANopen [2.]

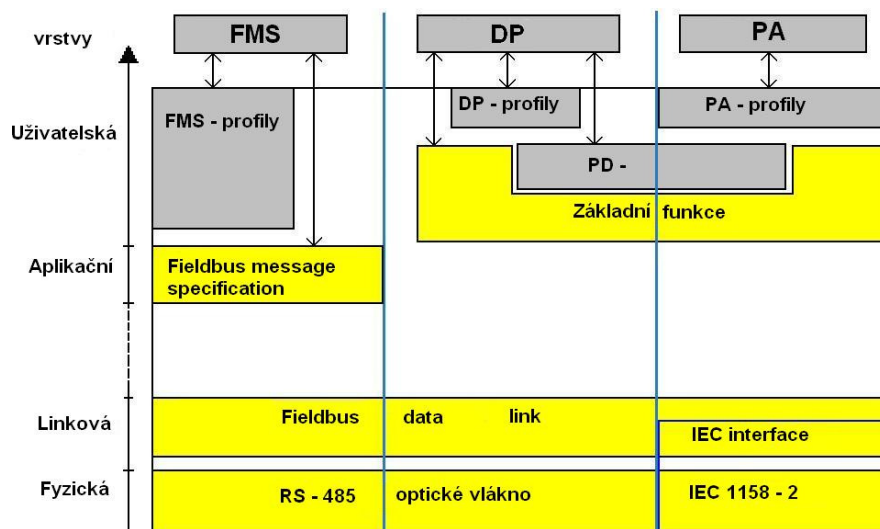
2.2.2 Profibus

Profibus je zkratka z anglického pojmu PROcess Field BUS. Tento typ komunikačního rozhraní je v současnosti nejrozšířenější v oblasti distribuovaného průmyslového řízení. Hlavní zásluhu na tom má včasná standardizace (EN 50170, EN 50254 a IEC 61158) a hlavně značná podpora evropských průmyslových gigantů.

Komunikační rozhraní Profibus vychází, jako většina, z upraveného modelu OSI. Definuje fyzickou, linkovou, aplikační a uživatelskou vrstvu protokolu. Základní struktura standardu Profibus je znázorněna na obr. 2.3.

Dnes známe tři varianty Profibusu. První je Profibus DP. Je nejrozšířenější a nabízí jednoduché funkce pro komunikaci, konfiguraci a řízení provozu na síti. Standardně je provozována na EIA/TIA 485 nebo s převodníky i na optických vláknech. O poznání složitější je další varianta Profibus FMS. Její složitost spočívá v komunikaci v heterogenním prostředí. Fyzická vrstva opět využívá EIA/TIA 485 nebo optické vlákno. Třetí variantou je Profibus PA. Má stejné funkce jako varianta Profibus DP. Fyzická vrstva je však realizována proudovou smyčkou podle standardu IEC 61158-2

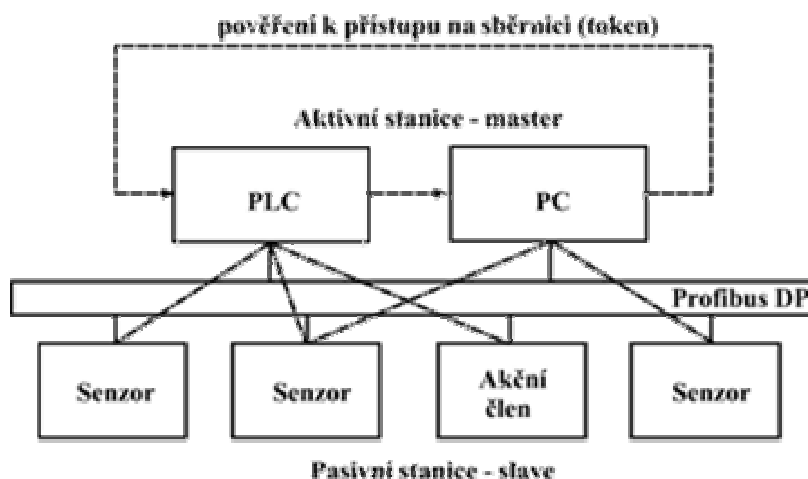
a umožňuje nasazení i v explosivním prostředí.



Obr. 2.3 Základní struktura Profibusu [2.]

Při samotném popisu vlastností fyzické vrstvy byla vybrána verze Profibus PD. Tato varianta je v praxi nejrozšířenější. Samotná sběrnice je stíněná, lze ji i vynechat v závislosti na vlivu elektromagnetického rušení, párovaný kroucený kabel. Na jeho koncích jsou umístěny terminátory. Na sběrnici lze připojit až 32 stanic v každém segmentu a při použití opakovacího je možné rozšíření až na 127 stanic. Maximální délka sběrnice je závislá na rychlosti přenosu. Pro rychlost přenosu 12Mb/s je maximální délka sběrnice 100m. Stanice se připojují pomocí konektoru D-Sub, který je běžně známý z analogových monitorů. Linková vrstva tvoří první jádro komunikačních rozhraní. Řídí veškerý přístup na sběrnici, definuje formáty zprávy, zabezpečuje přenos a základní časové parametry komunikace.

Profibus kombinuje dvě úrovně komunikace. Jelikož umožňuje typ komunikace multi-master, je nutné jednotlivým zařízením typu master dát pověření promluvit. Toto zajišťuje spojení Token Ring (obr. 2.4). Mastery jsou spojeny v kruhu a mezi nimi obíhá token, který uděluje právo oslovit zařízení typu slave. Zařízení slave smí pouze naslouchat a pokud je vyzváno, tak odpovědět. Celý proces přístupu je deterministický, zaručuje se doba jednoho cyklu. V aplikační a uživatelské vrstvě je celá řada protokolů. Nejpoužívanější je samozřejmě varianta DP, která je nasazována na nižších úrovních řízení pro komunikaci mezi inteligentní zařízení. Aplikační data jsou přenášena mezi jednotlivými řídicími uzly a jejich podřízenými. V nově představené verzi protokolu



Obr.2.4 Spojení stanic typu master[2.]

DPV2 existuje i možnost komunikace přímo mezi podřízenými na bázi modelu publisher-subscriber. Varianta FMS se používá pro komunikaci na vyšší úrovni řízení. Spojuje mezi sebou řídicí systémy a systémy vizualizace a sběru dat. Cílem této sběrnice není dosažení minimální doby reakce, ale především jednotné komunikace. Jednotnou komunikace či heterogenní prostředí rozumíme schopnost, kdy se vzájemně nekompatibilní systémy spolu dorozumívají. Komunikační model je založen na konceptu virtuálního zařízení. Komunikace pak probíhá na základě relací mezi těmito zařízeními. V rámci aplikační vrstvy jsou definovány různé skupiny služeb, sloužící pro správu virtuálního zařízení, adresáře objektů a proměnných. Profibus PA slouží především pro komunikaci mezi řídicími uzly a inteligentními senzory. Profil PA definuje jednotlivé třídy zařízení, takže je zajištěna snadná konfigurace a záměnnost produktů.

2.2.3 AS-i

Celým názvem Actuator Sensor Interface. Jak už název napovídá jedná se o komunikační rozhraní pro nejnižší úroveň řízení. Jedná se o velmi levné, úsporné, rychlé a deterministické komunikační rozhraní. Fyzická vrstva není nijak definována, a však celková délka nesmí přesáhnout 100m. S použitím opakováčů, maximálně však jen dvou, lze maximální délku zvýšit na 300m. Jako médium se používá jeden standardizovaný kabel. Jednotlivé uzly se na kabel připojují krempováním, odpadají tak konektory. Kabel je využíván jak pro datovou komunikaci, ale také pro napájení napětím 24V, jehož zdroj je připojen přes definovanou indukčnost. Skokové změny

v odběru jednotlivých uzlů generují napěťové impulsy obou polarit, které jsou superponovány na napájecím napětí a přijímány ostatními uzly. Pro zajištění synchronizace se využívá kódování Manchester. Na linkové vrstvě je použito řízení Master – slave. Jednotlivé uzly typu slave jsou periodicky oslovovány. Oslovování buď obsahuje příkaz pro jednotlivé uzly nebo výstupní data. Celý přenos je zabezpečen paritou a dalšími prvky vyplývajícími ze zvoleného kanálového kódování. Doba jednoho cyklu je 5ms při plném zatížení 31 účastnických stanic. Původně standard AS-i podporoval pouze digitální vstupy a výstupy, ale ve verzi 2.0 byl rozšířen i podporu analogových signálů. Jelikož existuje velmi silná a standardizovaná vazba mezi AS-i a Profibusem, jsou tyto dva standardy používány současně.

Kapitola 3. Projekt vizualizačního systému modelového technologického procesu

Modelovým technologickým procesem budeme rozumět pouze signály a informace, simulované pomocí funkcí vývojového prostředí LabVIEW. Na základě modelového příkladu bude v dané kapitole popsána vizualizace simulovaných informací, které budou z řídicího systému přenášeny pomocí sítě ethernet. Vytvořené „VI“ jsou umístěny na CD-ROMu, který je součástí práce. Aplikace pro vizualizaci dat slouží jako modelový příklad, pro daný řídicí systém, který je možné nadále rozpracovat pro následné použití v praxi.

3.1 Pojem vizualizační systém

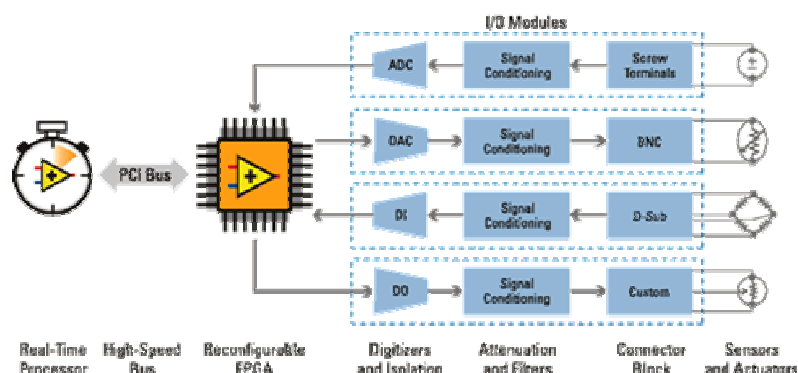
„Vizualizace, neboli zviditelnění, technologického procesu je považováno za vyšší formu řízení“[1.]. Člověku dává možnost do technologického procesu určitým způsobem zasahovat, sledovat, reagovat na situace, ale i poznat jeho zákonitosti a vlastnosti a zálohovat jeho průběh. To vše umožňuje pozdější optimalizaci technologického procesu a eliminaci ztrát.

Základním cílem vizualizace je co nejlépe znázornit, nejlépe graficky, co možná největší množství dat. Vizualizace, zvláště grafická, je pro člověka nejlepší a nejjednodušší způsob, jak poznat a případně se naučit potřebné informace. Tento cíl se přenesl i do průmyslového prostředí. Hlavním úkolem dřívější obsluhy stroje, bylo nutné znát přesné nastavení a podrobnou funkci stroje. V době kdy tento problém přebírají řídicí systémy, je však spíše požadováno, aby daná obsluha znala technologii obsluhované výroby. A právě k tomu dopomáhá samotná vizualizace technologického procesu.

Samotným pojmem vizualizační systém rozumíme soubor technických a programových prostředků potřebných pro vizualizaci řízeného technologického procesu. Technické vybavení tvoří tři základní prvky. První je řídicí systém, který je vybavený komunikačním rozhraním. Druhým prvkem je samotná komunikační linka, po které jsou přenášeny data. A posledním prvkem je nejčastěji osobní počítač. Důležité vlastnosti programového vybavení jsou grafické znázornění a archivace dat a stavů. Další vlastnosti jsou např. vytváření protokolu o průběhu procesu, možnost zpětného volání průběhu děje a popsání řízeného děje formou receptur.

3.2 Řídicí systém

Každý technologický proces je řízen řídicím systémem. Řídicím systémem rozumíme systém, který se stará o zpracování informací, rozhodování a řízení. V dřívější době se používaly jednodeskové mikropočítače obohacené o pár přidávaných obvodů. Většinou funkci řídicího systému zastupovaly 8-bitové mikropočítače. Ty však stačily pouze pro řízení malých procesů. Při návrhu náročnějších procesů, bylo potřeba



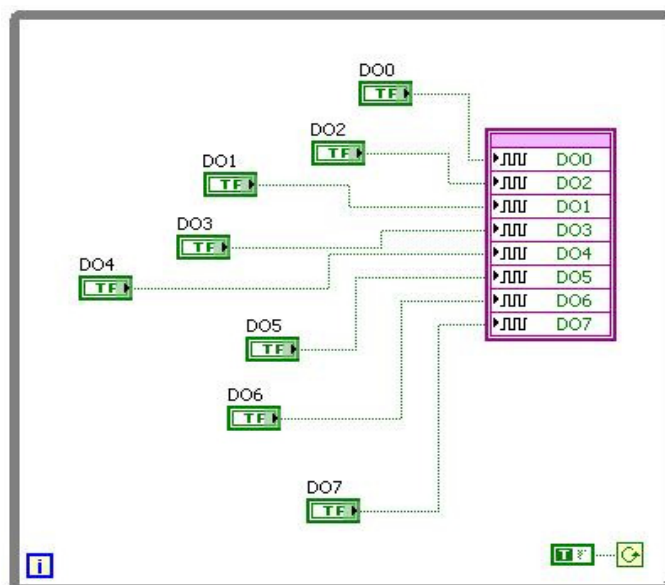
Obr. 3.1 Vnitřní struktura řídicího systému compactRIO[5.]

zvýšit požadavky i na řídicí systémy. Proto vznikly tzv. systémy PLC (angl. Programmable Logic Controller). Jejich výhoda spočívala v lepší možnosti jejich programování a kompaktnosti. Standardně byly vybaveny komunikačním rozhraním (např. RS 232C). Nároky na jednotlivé řídicí systémy však nadále stoupaly. Proto byly shrnuty ty nejlepší vlastnosti PLC systémů a osobních počítačů a byl vyvinut systém označovaný PAC (angl. Programmable Automation Controller).

Pro praktickou část práce je použit systém compactRIO (dále cRIO) od NI. Jedná se o systém PAC. Skládá se ze dvou hlavních částí (obr. 3.1). První část je tvořena embedded procesorem reálného času. Taktovací frekvence je 200MHz. Přes rychlostní PCI sběrnici je procesor spojen s programovatelným polem FPGA. FPGA pole je druhá část systému cRIO. Slouží jako prostředník, který komunikuje s přídatnými moduly se vstupními i výstupními signály. Přídatných modulů je na trhu celá řada. Dokonce je i možné u NI zakoupit tzv. KIT balíček, který umožňuje sestavit jakýkoliv modul se vstupní i výstupními signály.

3.3 Popis programu

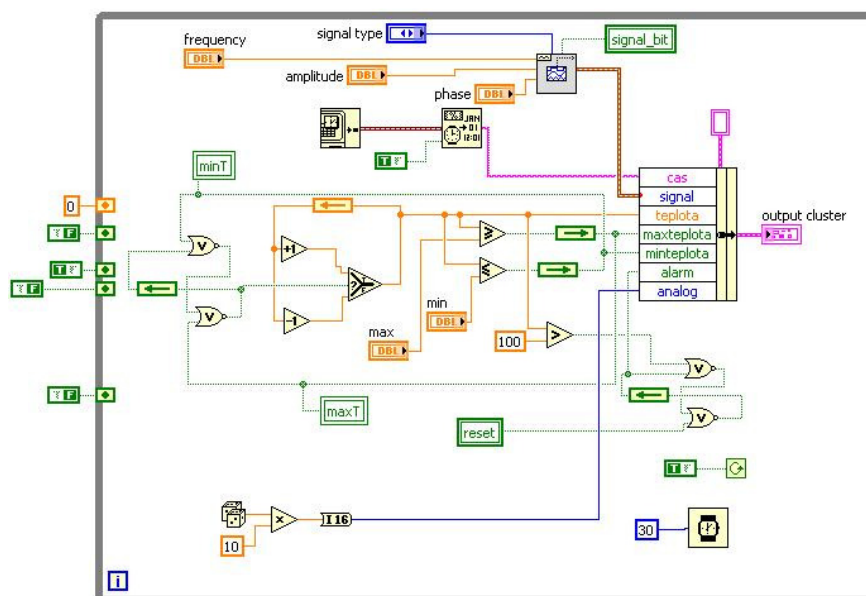
Úkolem práce je program pro vizualizaci modelového technologického procesu



Obr..3.2 Block diagram FPGA aplikace

v LabVIEW. Jeho součástí mělo být programování řídicího procesu i vizualizační části. Samotná práce je rozdělena na tři části. Všechny je spojuje ucelený projekt, který umožňuje vytvářet grafické vývojové prostředí LabVIEW.

První část je pojmenována názvem FPGA. Tato aplikace se výhradně stará o

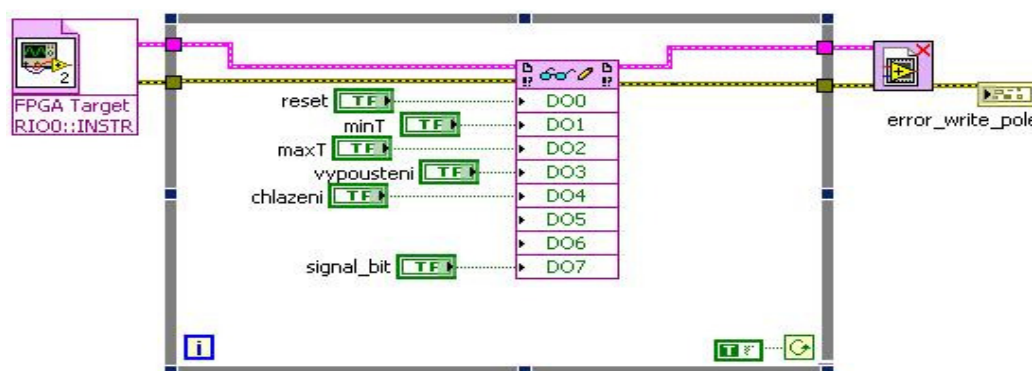


Obr. 3.3 Řídící smyčka v block diagramu aplikace Server

přenos signálů mezi procesorem a přídavnými moduly. Není nijak složitá a její vytvoření nezabere více než pár minut práce. Nejdéle trvá její samotné přeložení. Jelikož se grafický kód překládá do VHDL jazyka pro programovatelné pole. Aplikaci tvoří samostatná smyčka, která běží neustále (obr. 3.2). Ve smyčce jsou umístěny bloky typu boolean, které jsou pojmenovány stejně jako jim příslušné výstupy. Aplikaci FPGA spouští samostatná druhá aplikace, která je nazvána Server.

Server běží pod platformou LabVIEW v samotném řídicím systému cRIO. Na tuto aplikaci jsou kladeny vysoké nároky. Svým způsobem musí být natolik inteligentní, aby rozpoznala vážnost vzniklé chyby a sama se jí pokusila opravit. Není totiž vhodné časté přerušování, z důvodů odstavení části provozu resp. stroje, které by mělo za následek výpadek výroby.

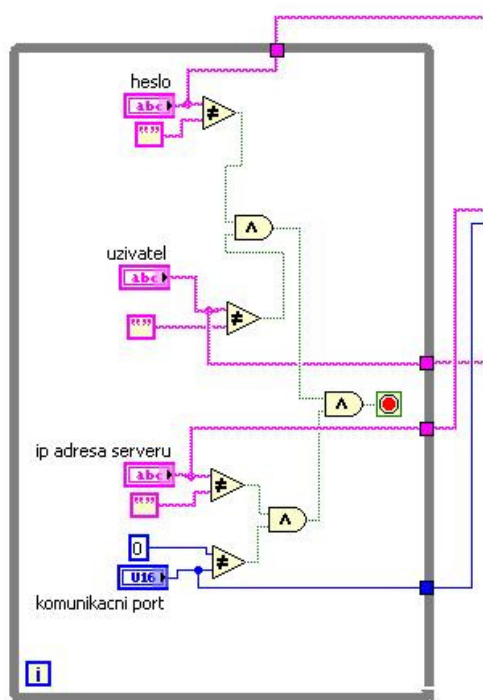
V aplikaci Server běží nezávisle na sobě tři smyčky. První z nich se stará o řídicí funkce pro samotný technologický proces (obr. 3.3). Je zde uvedena logika, která zvyšuje a snižuje teplotu v rozmezí, které zadává uživatel. Defaultně je zde nastaveno rozmezí od -5 do 85°C. Na simulaci teploty je virtuálně připojeno čidlo, které je nahrazeno proměnnou Alarm. Ta je spuštěna vždy, když teplota přesáhne 100°C. Alarm se dá vyresetovat pomocí proměnné Reset. Dále je zde logika náhodného čísla pro simulaci analogového signálu např. napětí a simulace spojitého analogového průběhu např. sinusové vlny. A poslední je získání přesného času. Všechny tyto proměnné jsou



Obr. 3.4 Spouštěcí smyčka pro FPGA aplikaci v block digramu aplikace Server

svedeny do clusteru, se kterým se dále pracuje v komunikační smyčce. Druhá smyčka je využita pro otevření a přepsání hodnot v aplikaci FPGA (obr. 3.4). Tato smyčka běží neustále. Pro předchozí dvě smyčky je nejdůležitější synchronizace předávání dat. O tomto pojmu resp. problému, který se vyskytl i v dané práci a jeho možném řešení se

zmíníme v následující kapitole 3.4 Řešené problémy. Poslední smyčka, která běží v řídicím systému, se stará o samotný přenos informací pro vizualizační část. Pro komunikaci byl použit ethernet a komunikační protokol TCP. V TCP smyčce [A], která komunikuje se samotnou vizualizační částí, je navíc přidána logika, která definuje tři stavy spojení. První stav je “odpojeno“. V tomto stavu smyčka setrvává do doby než jí osloví vizualizační část. Do stavu “odpojeno“ se smyčka dostane při načtení příznaku s hodnotou konec. Při prvním spuštění je nutné zkontrolovat, jestli je příznak nastaven na hodnotu konec. Důvod jeho zavedení si popíšeme později v následující kapitole 3.4 Řešené problémy. Další stavy jsou odvozeny od uživatelského oprávnění. Z důvodů, že každý podnik si své technologie velmi pečlivě chrání, bylo do aplikace zahrnuto i jednotlivá oprávnění. Oprávnění typu administrátorských účtů, v práci pojmenováno Admin, umožňuje do aplikace zasahovat ve formě změny vstupních dat, např. měnit meze horního limitu teploty, ruční ovládání výpustě nádrže nebo chlazení. Opakem jsou účty uživatelské, v práci vystupující jako User. Ty dovolují pouze nahlížení na aktuální hodnoty v procesu a jeho stav. V modelovém příkladě, který je naprogramován pro



Obr. 3.5 Smyčka kontroly v block diagramu aplikace Klient

účely bakalářské práce, byla spíše větší pozornost věnována právě přenosu dat mezi aplikací Server a vizualizací. Všechny tři smyčky spolu navzájem komunikují prostřednictvím lokálních proměnných. Vzájemné propojení pomocí čar, není

použitelné jelikož by se smyčky staly navzájem závislé. To by znamenalo, že spuštění další smyčky nastane pouze v případě, že smyčka předchozí byla již dokončena. Poslední aplikace, která řeší vizualizační část, nese označení Klient. Aplikaci Klient tvoří samostatná smyčka [B]. Ke spuštění této aplikace je nutné vyplnit položky IP adresy vzdáleného řídicího systému, číslo komunikačního portu, na kterém řídicí systém komunikaci naslouchá, a v neposlední řadě ochranné údaje v podobě hesla a uživatelského jména (obr. 3.5). Pokud jsou tyto údaje správně vyplněny začíná samotná autorizační komunikace. Oslovení vzdáleného řídicího systému a čekání na jeho vyjádření. Pokud aplikace Klient dostane kladnou odpověď v přesně stanoveném formátu, zahájí samotnou vizualizaci. Nejprve je vždy odeslán startovací znak “s”. Důvod jeho zavedení si popíšeme opět později v následující kapitole 3.4 Řešené problémy. Pak následuje buď odeslání nových vstupních dat nebo rovnou samotný příjem. Tento výběr záleží na zvoleném uživatelském účtu vyplněném při spuštění aplikace Klient. Aplikace Klient je ukončena stiskem tlačítka s popisem ukončení spojení. Po tomto stisku se odešle ukončovací zpráva, která je následně potvrzena přijetím hodnoty “odpojeno”. Na základě ukončení spojení aplikace Server přejde opět do stavu odpojeno, kdy naslouchá, kdo jí ke komunikaci vyzve. Nedílnou součástí aplikace Klient je Front panel, který obsahuje zobrazovací a kontrolní prvky. Front panel je zobrazen v příloze C. Na panelu jsou umístěny dva prvky typu cluster, prvek data, pro zobrazení informací přijatých z aplikace Server, a prvek zápis, který umožňuje měnit hodnoty v aplikaci Server. Další skupinou prvků jsou položky autorizačního procesu a dvojice prvků indikátor chyby a ukončovací tlačítko.

Pro veškeré testování a odlaďování chyb ve výše uvedených aplikacích bylo využito nástrojů, které nabízí grafické vývojové prostředí LabVIEW. Obrázek 1.4 v kapitole 1. nám detailně zobrazuje veškerá spouštěcí tlačítka, která umožňují snadné a jednoduché ovládání daných nástrojů. Uvedené nástroje napomohly k odhalení chyb, které jsou řešeny v následující kapitole.

3.4 Řešené problémy

Při řešení bakalářské práce se vyskytlo spoustu problémů. Prvním zarážejícím faktorem bylo samotné vývojové prostředí LabVIEW, které se využívá pro všechny vrstvy řízení. Na první pohled se prostředí zdálo velmi neintuitivní a na orientaci dosti náročné. Nejhorším faktorem pro práci s vývojovým prostředím LabVIEW je přenaučit

se přemýšlet v psaném kódu a začít uvažovat v jeho grafickém vyjádření. Po nějakém čase práce a zvyknutí si na styl programování resp. kreslení v tomto prostředí byl pohled na tento systém zcela odlišný.

Druhým vážnějším problémem, který se v práci řešil, byla samotná forma komunikace mezi aplikací Klient a Server. Jelikož je pro spojení mezi těmito aplikacemi použito TCP protokolu, který v LabVIEW dokáže zpracovávat data pouze ve formátu string, bylo nutné všechny přenášené informace resp. proměnné přetypovat do požadovaného formátu. Jak už bylo zmíněno v předchozí kapitole 3.3 Popis programu, všechny informace byly svedeny do datového typu cluster, který musí být definován přesně podle vstupních dat. Nadále se tento datový typ pomocí funkce s názvem Flatten To String, převedl na typ string. Po přenesení požadovaným směrem, buď Klient - Server nebo Server – Klient, dojde k opětovnému rozbalení pomocí inverzní funkce s názvem Unflatten From String. U této funkce je nutné nastavit na jaký formát se daný vstupní string má rozbalit. Toto definování se realizuje přivedením konstanty datového typu, která je stejného formátu jako odesílaný cluster, na vstupní svorku s názvem Type.

Zvolené řešení předchozího problému vedlo bohužel k dalšímu závažnějšímu problému. Chyba, která často nastávala při ukončení spojení, vznikla právě u funkce Unflatten From String. Vždy to bylo tím, že aplikace Klient svoji činnost ukončila a aplikace Server se i přesto snažila nadále přijímat další informace. Docházelo tedy k rozbalování nějakých náhodných dat. Tento problém byl vyřešen zavedením vysílaného příznaku hodnoty “konec“. Aplikace Server příznak rozpozná dříve, než dojde k vykonání samotné funkce Unflatten From String a sama komunikaci také ukončí. Zároveň tak byla upravena i možnost, jak se dostat do výchozího stavu komunikace, kdy aplikace Server naslouchá, jestli ji někdo vyzve ke spojení. Pokud by však došlo k nečekanému pádu aplikace Klient, klíčový příznak by se nestihl odeslat a aplikace Server by o této chybě, vzniklé na druhé straně, neměla zdání. Proto byl do celého komunikačního procesu přidán ještě synchronní znak “s“, který je vždy vysílán před samotnými daty aplikací Klient. Při neobdržení tohoto synchronizačního stavu, se aplikace Server chová jako kdyby obdržela příznak konce. To vše napomáhá k tomu, aby aplikace Server nebyla ohrožena chybou, která by vznikla aplikací Klient.

Posledním problémem, který se v rámci práce vyskytl, je problém předávání dat mezi aplikacemi Server a FPGA. Zde je nejsložitější dbát správné synchronizace.

Rychlost křemíkových hradel je příliš obrovská a aplikace Server na tuto změnu nestačil reagovat. V samotné vizualizaci už byl signál velice zkreslen. Daný problém je řešitelný, ale v rámci práce mu nebylo věnováno tolik času, kolik by si vyžadoval. Jedná se o velmi složitou problematiku, která není ani v dnešní době nějak příliš kvalitně popsána. Jedno z možných řešení je použití zásobníků, které jsou postupně vybírány. Na radu byl problém vyřešen umístěním generátoru sinusového průběhu do aplikace Server. I tak je synchronizace aplikací Klient a Server nelehkou záležitostí.

Kapitola 4. Bezdrátové spojení v průmyslu

Cílem výše uvedené kapitoly je nalézt alternativu ke komunikačním rozhraním, dále tyto možnosti stručně charakterizovat a pokusit se otestovat jejich využití při praktických úlohách.

4.1 Vývoj

S velmi rychlým pokrokem ve vývoji informačních a komunikačních technologií a zejména pak v okamžiku, kdy se na trhu objevila první mobilní zařízení, která přímo pracovala s digitálními daty (elektronické diáře, kapesní počítače, notebooky atd.), vyvstala i potřeba vzájemné bezdrátové datové komunikace mezi jednotlivými zařízeními. Na řadě je i otázka proč těchto technologií nevyužít i v průmyslu. Ideální představa, kdy technik přijde do provozu se svým mobilním zařízením, ať už je to PDA nebo notebook, a bezdrátově se připojí k jakémukoliv řídicímu systému, je už za dveřmi, ne-li už tyto dveře otvírá. Je však otázkou jestli se bezdrátových technologií nedá využít i jinak. Představme si problém, kdy nelze použít běžně dostupné kabelové snímače a právě zde nastupuje myšlenka použití bezdrátového přenosu. Dnes je na trhu už několik bezdrátových technologií, které jsou určeny pro přenos informací v průmyslovém prostředí. V průběhu zpracování bakalářské práce bylo možné seznámit se se dvěma bezdrátovými technologiemi určenými pro průmysl. První byla technologie Zigbee. O této technologii se hodně píše, spousta firem se hlásí k její podpoře, ale opak je zřejmě pravdou. Bohužel v praxi je její využití minimální. A když se podařilo na internetu najít přídatný modul pro cRIO, firma NI nebyla schopna říci, jestli cRIO tuto technologii opravdu podporuje. Druhou technologií určenou pro průmysl je Wawenis. Tato technologii je zahrnuta i do této bakalářské práce a to z důvodu možnosti zapůjčení bezdrátových měřicích čidel. Dále práce obsahuje technologii Bluetooth a IEEE 802.11 označovanou Wi-Fi.

4.2 Bluetooth

Počátky se váží na rok 1998. V tomto roce společnosti Ericsson, IBM, Intel, Nokia a Toshiba založili společnost s názvem Bluetooth Special Industry Group (zkráceně SIG). K zavedení bezdrátové technologie vedl velký nárůst kabelů s různými typy konektorů, to znamenalo značnou nekompatibilitu jednotlivých zařízení.

Postupem času a vývoje byly do standartu Bluetooth přiřazeny i obecně dosažitelné body pro data řeč a postupy pro jednoduchou stavbu dynamicky restructurovatelných specializovaných datových sítí.

Název je odvozen ze jména dánského krále Harald Bluetooth. Ten využil svých diplomatických vlastností k tomu, aby válčící kmeny ukončily rozepře a zasedly ke

Class 1	Nejvýkonější, teoretický dosah až 100m ve volném prostoru
Class 2	Běžně používaná v přenosných zařízeních (mobil,PDA), dosah do 10m
Class 3	Dnes už minimálně, dosah do 1m

Obr. 4.1 tabulka tříd bezdrátové technologie Bluetooth

společnému stolu. Právě této analogie bylo využito pro název technologie Bluetooth, která má sloužit ke snadnější komunikaci mezi různými typy zařízení.

Specifikace je definována standardem IEEE 802.15.1. Sítě vytvořené pomocí Bluetooth spadají do skupiny PAN (Personal Area Network). Bluetooth rozdělujeme podle výkonu do tří tříd, které jsou uvedeny v tabulce na obr. 4.1.

Nejpoužitelnější ke konci roku 2006 je Bluetooth 1.2. Nejmodernější je Bluetooth 2.0 EDR (Enhanced Data – Rate), kde se používá nová modulace $\pi/4$ DQPSK. Ta zvyšuje datovou propustnost až na 6Mbit/s. Je použito tzv. bezlicenční pásmo 2,4GHz (přesněji 2,402 – 2,480 GHz), často označované ISM (industrial, scientific and medical). K přenosu využívá metody FHSS (Frequency-hopping spread spectrum). Během jedné sekundy dojde k šestnáctistému skokům (přeladění) mezi 79 frekvencemi s rozestupem 1MHz. Je zde myšleno i na ztrátu vyslané informace. Pokud dojde ke ztrátě informace během přenosu, je odeslána znovu, ale na jiné frekvenci. Kvalita přenosu tudíž závisí i na tom kolik je v dané lokalitě volných frekvencí právě v pásmu 2,4GHz. Pokud jsou v prostoru nějaké překážky prudce klesá dosah. To nemá za následky ani tak skokové výpadky sítě jako navyšování počtu špatně odeslaných paketů. Při přenosu je možno provozovat jak symetrický tak i asymetrický přenos. Rychlost přenosu se pohybuje okolo 700kbit/s. Bluetooth umožňuje dvoubodové a mnohabodové spojení. Jedna stanice vždy musí plnit úlohu „Master“. „Master“ dokáže obsloužit až 7

stanic se statusem „Slave“. Je možné aby jedna stanice měla status i „Master“ i „Slave“, to umožňuje budovat tzv. pikosítě. Na ploše o průměru 10m jich lze provozovat 10. Zařízení se identifikují pomocí BT_ADDR(je to vlastně MAC adresa).

Bluetooth využívá tři základní protokoly. První je L2CAP. Jde o protokol, který je používán uvnitř rozhraní. Posílá síťové pakety po HCI (Host Controller Interface), u nehostovaných systémů přímo na Link Manager. Přenáší data mezi různými vyššími protokoly. Rozděluje a slučuje pakety. Poskytuje jednotnou správu přenosu a správu kvality. Dalším protokolem je BNEP, který se používá pro dopravu paketů v síti PAN. Třetím protokolem je RFCOMM. Vychází z L2CAP. Využívá se při vytváření emulovaného sériového portu RS 232.

4.3 IEEE 802.11

Mnoho uživatelů tohoto spojení ho zná pod označením Wi-Fi. Tento název je však označení sdružení výrobců hardwaru. Jeho zavedení má zpřehlednit a autorizovat hardware pro bezdrátový přenos podle standardů IEEE 802.11. Velmi často je však tento název spojován ze standardem IEEE 802.11b. Ten byl uveden v roce 1999. Ještě dříve byl standard IEEE 802.11a. Jeho zavedení do praxe však bylo složitější a tudíž zpožděno.

Wi-fi je řazená do WLAN (Wireless Local Area Network). Samotný standard IEEE 802.11 nestačil, pro svoji malou datovou propustnost. Proto se rozšířil o IEEE 802.11a, která zvládne až 54Mb/s a IEEE802.11b s rychlostí 11Mb/s. Nároky na bezdrátovou komunikaci se stále rozšiřují a proto byly vytvořeny další standardy s označením IEEE 802.11c-j. Jde vesměs o rozšiřující normy. Např. standard IEEE 802.11g je rozšiřující pro IEEE 802.11b. Umožňuje přenosovou rychlost až 54Mb/s.

Standard 802.11 je provozován na bezlicenčním pásmu 2,4GHz. Toto pásmo je však už velmi přeplněné a tak se přešlo i na pásmo 5GHz (IEEE 802.11a). Je zde použita metoda DSSS (Direct Sequence Spread Spectrum). DSSS vysílač přeměňuje tok dat (bitů) na tok symbolů, kde každý symbol reprezentuje skupinu jednoho či více bitů. Za použití modulační techniky QPSK (Quadrature Phase Shift Keying), vysílač moduluje nebo násobí každý symbol pseudonáhodnou šumovou sekvencí (na tzv. čip). Tato operace uměle zvětšuje použitou šířku pásma v závislosti na délce sekvence. DSSS dělí pásmo na 14 kanálů po 22 MHz, které se částečně překrývají (pouze tři z nich se nepřekrývají vůbec).

Bezdrátová lokální síť (buňka) může pracovat ve dvou konfiguracích (Basic Service Set, BSS). V nezávislé konfiguraci (ad hoc) mezi sebou stanice komunikují přímo a není třeba instalovat žádnou podpůrnou infrastrukturu. Taková konfigurace je mimořádně vhodná pro náhodná uspořádání (trvajících podle potřeby hodiny, ale i měsíce), avšak nehodí se pro rozsáhlá řešení. Konfigurace infrastruktury, s distribučním systémem, předpokládá existenci přístupového bodu (Access Point, AP), který funguje jako základnová rádiová stanice a současně datový most. Přístupový bod je centrem každé WLAN a je nepohyblivý. Tato konfigurace předpokládá propojení s distribučním systémem, tedy klasickou LAN (nejčastěji Ethernet). Každá koncová stanice (klient) si najde "svůj" přístupový bod (na základě pravidelně vysílaného signálu od AP) a komunikuje pouze s ním. Klienti WLAN používají síťový adaptér, který obsahuje vysílač, přijímač, anténu a hardware.

Identifikace ve wi-fi je úzce spojena z bezpečností. Wi-fi je obecně považována za velmi lehce prostupnou a její zabezpečení je hodnoceno jako malé. Nejlepším krokem je zkombinovat co možná nejvíce možných prvků. Buď samotným skrytím SSID, ověřování MAC adresy, využití šifrovacích klíčů WEP nebo standardu 802.1X. Posledně jmenovaný způsob ochrany je dnes považován za nejbezpečnější. Jde v podstatě o to, že nově připojené zařízení musí nejdříve projít ověřením identifikace pomocí ověřovacího serveru. Ten většinou provozuje třetí strana (např. radius server). Až po jeho vyjádření je zařízení povolena komunikace.

4.4 Wawenis

Tuto technologii na trh přinesla francouzská firma Coronis Systems. Byla vyvinuta přímo pro požadavky průmyslu. Slouží k přenosu informací mezi bezdrátovými moduly a čidly. Bezdrátová technologie Wawenis se na první pohled jeví jako dokonale upravená technologie Bluetooth pro použití v průmyslu.

Wawenis může pracovat na frekvencích 433MHz, 868MHz a 915MHz. Rychlost přenosu se pohybuje v rozmezí 4.8 – 100 kbps, ve většině případů kolem 10 kbps. Tato technologie pracuje stejně jako Bluetooth, využívá přenosové metody FHSS (Frequency-hopping spread spectrum). Jako modulace se používá GFSK (Gaussian Frequency Shift Keying). Jedná se o Gaussovskou modulaci s frekvenčním klíčováním, při které je modulátoru FSK předřazena dolní propust gaussovského typu. Pravoúhlý signál se po průchodu touto propustí přemění na spojitý signál ve tvaru Gaussovy

křivky, a tím se omezí šířka pásma potřebná pro přenos. Podle použitého vysílače a jeho vyzařovacím výkonu, který je 25mW nebo 500mW, je určena vysílací vzdálenost. U slabšího vysílače je to teoreticky až 1km, u silnější verze je vzdálenost čtyřikrát delší, tedy až 4km ve volném prostoru.

Firma si v přenosové oblasti vyvinula svůj vlastní komunikační protokol. Komunikace probíhá pomocí hexadecimálního kódu odesílaného v tzv. rámcích. Uživatel může využít dodávané dynamické knihovny dll nebo si celý proces naprogramovat ve svém programovacím jazyce zcela od začátku sám. Veškeré informace které se týkají komunikační oblasti jsou dostupné a firma Macroweil, která v České republice zastupuje firmu Coronis Systems, je spolu s moduly a dynamickými knihovnami dll dodává.

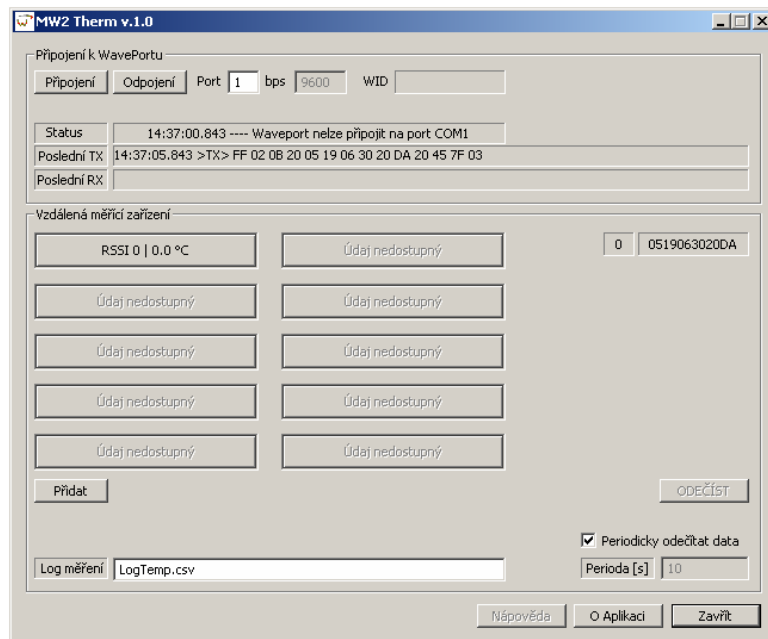
Technologie Wawenis umožňuje jak bodové spojení, tak i spojení více modulů. Moduly se dají řadit do tzv. stromů, hvězd či různých jejich kombinací. Hlavním prvkem sítě je vždy WawePort, který je připojen k osobnímu počítači, notebooku nebo PDA. WawePort lze připojit přes USB, RS 232 nebo i Compact Flash.

4.5 Využití technologie WAWENIS

4.5.1 Popis testovaných modulů

Pro testování v bakalářské práci byla zapůjčena dvě měřící čidla a sběrný port firmy Coronis Systems. Port nese označení WawePort a je připojitelný přes rozhraní USB. Samotná komunikace probíhá po komunikačním portu COM. Jeho vysílací výkon je 25mW.

Dále bylo zapůjčeno teplotní čidlo Dallas 18S20 s označením WaweTherm. Teplotní rozsah čidla je -55 až 125°C, s chybou měření $\pm 0,5^\circ\text{C}$ s přesností na 0,5°C. Posledním zapůjčeným modulem bylo analogové čidlo. Bohužel k tomuto čidlu nebylo příliš mnoho dostupných informací ohledně jeho technického provedení a popisu. Až po dlouhém testování bylo zjištěno, že se jedná o analogové čidlo pro měření proudu. Čidlo umožňuje snímat proud 4-20mA. Bohužel nebylo zjištěno s jakou chybou či přesností toto čidlo měří. Jak již bylo napsáno výše, firma ke svým produktům dodává kompletní přehled použitého komunikačního protokolu s podrobným popisem a dokonce i dynamické knihovny dll. Firma Macroweil dokonce ještě přidává základní startovní aplikace, kde si je možné komunikaci vyzkoušet. To vše umožňuje uživateli sestavit si

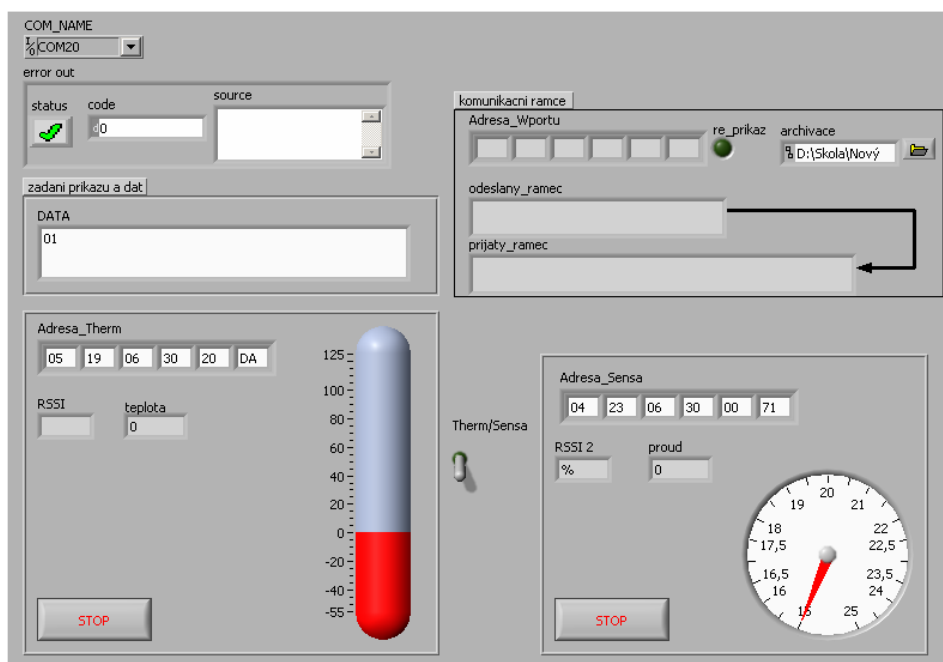


Obr. 4.2 Uživatelské rozhraní aplikace MW2 Therm

vlastní program a optimalizovat ho pro svoji momentální potřebu.

4.5.2 Testování teplotního čidla Wawetherm Dallas 18S20

Pro toto testování byla použita nejdříve startovní aplikace firmy Macroweil označenou MW2 Therm (obr. 4.2). Bohužel se nepodařilo zjistit v čem je přesně aplikace naprogramována a jestli využívá dodávané knihovny dll či celý komunikační



Obr. 4.3 Uživatelské rozhraní aplikace Wawenis

proces obstarává sama. K porovnání výsledků naměřených první aplikací, byla naprogramována aplikace v programovém prostředí LabVIEW 8.2.1. Aplikace je pojmenována Wawenis (obr. 4.3). Kvůli všeobecnému použití byla volena cesta ošetření celého komunikačního procesu.

Aplikace je tvořena na část tzv. „inicializace“, kdy je osloven WawePort s požadavkem navrácení jeho komunikační adresy. Tato adresa sice není nutně potřeba ke komunikaci, ale slouží k identifikaci při používání více Waweportů. V další části už je oslovován přímo vzdálený modul, v daném případě teplotní čidlo Dallas 18S20. Nejdříve je načteno tzv. RSSI (Received Signal Strength Indicator level), což je zkratka společnosti Coronis Systems. Pod touto zkratkou se ukrývá kvalita přenosu. Je uváděna v rozmezí 0x00 - 0x2F, to odpovídá 0-100%. V optimálním režimu je tato hodnota 96 –

MSB								LSB							
Most Significant Byte								Least Significant Byte							
b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
S	S	S	S	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴

Obr. 4.4 Ohodnocení bitů teploty [10.]

92%. V druhém kroku druhé části je zjišťována aktuální teplota. Teplota je přijata ve velikosti 2 bajtů. Horní polovina vyššího bajtu je použita pro znaménko. Nula značí nezápornou teplotu a samé jedničky v této oblasti znázorňují znaménko mínus před

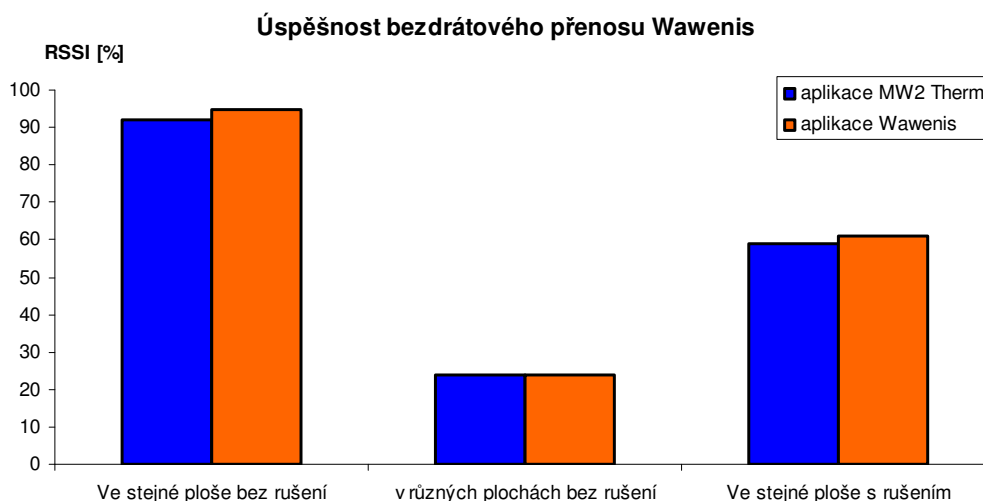
Temperature	Binary value (MSB First)				Hexadecimal value
+125°C	0000	0111	1101	0000	0x07D0
+85°C	0000	0101	0101	0000	0x0550
+25°C	0000	0001	1001	0000	0x0190
0°C	0000	0000	0000	0000	0x0000
-10.125°C	1111	1111	0101	1110	0xFF5E
-55°C	1111	1100	1001	0000	0xFC90

Obr. 4.5 Tabulka ukázky převedených teplot [10.]

měřenou teplotou. Dalších 12 bitů je ohodnoceno podle obr. 4.4. Ukázky převedených teplot jsou na obr. 4.5.

Při testování se práce zaměřuje na kvalitu signálu, tedy na hodnotu RSSI. Testování bylo prováděno ve třech různých podmínkách. První test byl provedl za

podmínek, že Waweport i testované čidlo byly ve stejné rovině. Toto měření nebylo ovlivněno provozním rušením a vzdálenost Waweportu a čidla byla okolo 25m. Druhý test byl uskutečněn tak, že Waweport a čidlo byly umístěny v různých rovinách. Tentokrát vzdálenost byla okolo 20m a opět bez vlivu provozního rušení. Ve třetím testu bylo čidlo umístěno do blízkosti pracovního stroje později i do odstíněného rozvaděče. Waweport se nacházel ve stejné rovině ve vzdálenosti 20m. U každého



Obr. 4.6 Úspěšnost bezdrátového přenosu Wawenis

měření bylo provedeno minimálně 20 hodnot RSSI. Tyto hodnoty byly následně zprůměrovány a vykresleny do grafu, který je zobrazen na obr. 4.6.

Na měřené hodnoty se od sebe dosti liší. Nejhorší výsledek resp. kvalita síly signálu přenosu nastává tehdy, je-li Waweport umístěn v jiné rovině než samotné čidlo. Signál se tudíž velmi špatně šíří vertikálním směrem. Signál byl při měření natolik slabý, že nebylo nutné ani defakto možné měřit tento stav i s provozním rušením. Opakem této situace je umístění teplotního čidla a Waweportu do stejné roviny. V této situaci je přenos téměř bezchybný. Pokud přidáme vliv provozního rušení je výsledek asi o třetinu horší. Při pohledu i na námi navíc zjišťovanou teplotu, zjistíme, že změřené výsledky se neliší. Z toho plyne, že kvalita síly signálu nemá příliš značný vliv na zkreslování hodnot. Ani výskyt chyb měření či špatně přijatých dat není nijak velký jako u vertikálního směru bez rušení. Celkovou slabinou celé bezdrátové technologie rozhodně není kvalita přenosu ale jednoznačně rychlost přenosu naměřených dat. Aplikace MW2 Therm zaznamenávala hodnoty periodicky po 10 sekundách. Oproti

tomu aplikace Wawenis dokázala zkrátit čas na polovinu a údaje zaznamenávala každých 5 sekund. Po stránce naprogramování přenosu je to nejmenší čas, kterého lze dosáhnout. Další možností je zkusit přenastavit vhodněji různé parametry přenosu a samotného teplotního čidla. Tato cesta nebyla v práci, z časových důvodů, více dopodrobna prozkoumána. Problém rychlosti přenosu není až tak důležitý u měření teploty, ale pokud vezmeme v úvahu měření analogového průběhu jakéhokoliv signálu s velkou frekvencí, tak už čas zaujímá velmi důležitou roli. Pokud nás zajímá pouze průběh signálu, můžeme využít vnitřní paměť měřících čidel. Hodnoty se pak předávají jako balík.

I když technologie Wawenis je určitě jednou ze spolehlivějších technologií pro bezdrátový přenos informací v průmyslové automatizaci, její použití je ovlivněno časovou náročností komunikace jednotlivých zařízení. Z osobních zkušeností při měření teploty se prokázala jako zdatný konkurent kabelových spojení a při snaze firmy dále tuto technologii rozvíjet, je možné, že i kabelové spojení v mnoha ohledech předčí.

Závěr

Téma práce vzniklo ve spolupráci s Oddělením realizace vývoje elektro firmy PRECIOSA, a.s., na základě její dlouholeté spolupráce s Fakultou mechatroniky a mezioborových inženýrských studií. Společnost PRECIOSA se snaží využít všechny své možnosti pro dosažení lepších a efektivnějších výsledků své činnosti, proto zvolila i cestu zadávání bakalářských a diplomových prací. Dává tím možnost studentům získávat nové užitečné poznatky při tvorbě dané práce a zároveň studenti svojí činností, pomáhají společnosti k aplikaci dané problematiky do praxe.

Téma bakalářské práce bylo zvoleno na základě současného využívání vývojového prostředí LabVIEW k tvorbě virtuálních nástrojů, které se využívají pro řízení technologických procesů a dále i k využití vizualizace dat. Právě zmíněná problematika byla hlavním tématem bakalářské práce. I když je práce spíše teoretického charakteru a ve větší míře se věnuje popisům a sumarizaci pojmů spojených s vizualizací dat, tak její praktická část může určitou měrou pomoci při realizaci vizualizace pro momentálně řešený nebo v budoucnu řízený technologický proces. Doplňkové téma práce zaměřené na bezdrátové technologie bylo zvoleno z důvodu nalezení alternativy ke klasickým kabelovým spojením, respektive ke komunikačním rozhraním, jelikož u speciálních případů je obtížné jejich použití.

LabVIEW je pro realizaci řídicích, měřicích a regulačních aplikací jeden z nejlepších nástrojů dostupných na trhu. Pro uživatele je přehledné a lehké, téměř vše se dá zvládnout pomocí pohybu myši. K samotnému programování napomáhá barevné rozlišení datových typů. Jedinou slabinou daného prostředí je jeho využití ve vizualizaci dat. Programátorům nenabízí až tak dokonale propracované prvky pro zobrazování.

Pojem komunikační rozhraní je označení pro samotné datové sběrnice, které spojují jednotlivé prvky. Jejich použití je v dnešním průmyslu velmi běžné. V práci jsou probírané i bezdrátové technologie, které by v budoucnu mohly dnes standardně používané komunikační rozhraní nahradit. Při testování bezdrátové technologie Wawenis se však ukázala hlavní nevýhoda a to časová náročnost. Pokud by se jednalo o procesy méně časově náročné nasazení bezdrátové technologie Wawenis by nic nebránilo.

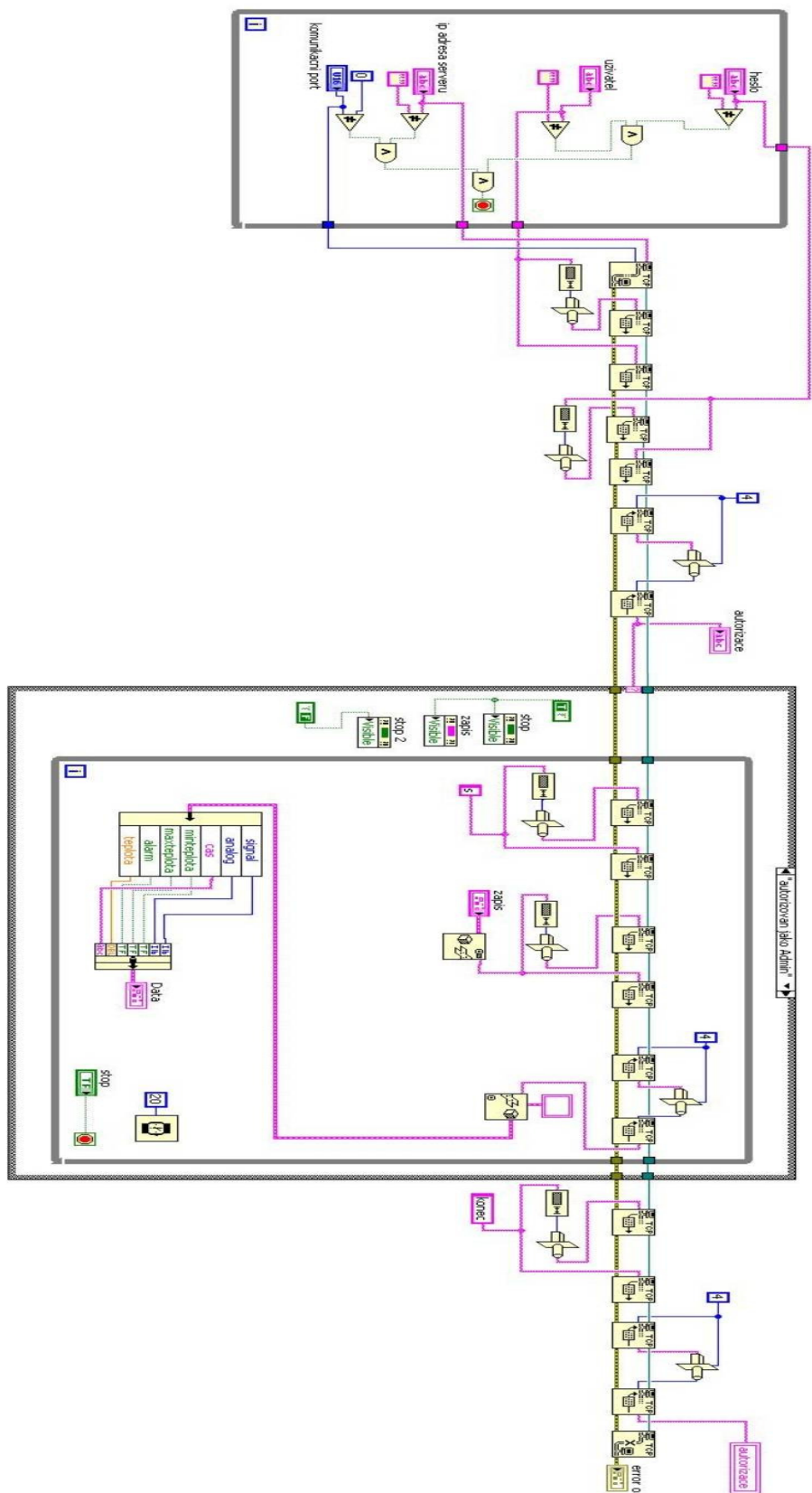
Hlavním úkolem práce bylo vytvořit vizualizační aplikaci pro modelový technologický proces. Modelový příklad byl opravdu jednoduchého charakteru a využití

řídícího systému nebylo značné. Výběr řídícího systému byl ovlivněn možnostmi jeho praktického použití v práci. Řídící systém compactRIO byl zvolen na základě vzájemné dohody s firmou PRECIOSA, a.s.. Tento řídící systém splňuje všechny nároky kladené na řízení těch nejnáročnějších procesů. Všechny funkce jsou dostupné z grafického vývojového prostředí LabVIEW. Hlavní činnost práce byla směřována na přenos a zobrazení informací. V práci je tento úsek detailně popsán a jsou zde vysvětleny a vyřešeny veškeré vyskytnuté problémy.

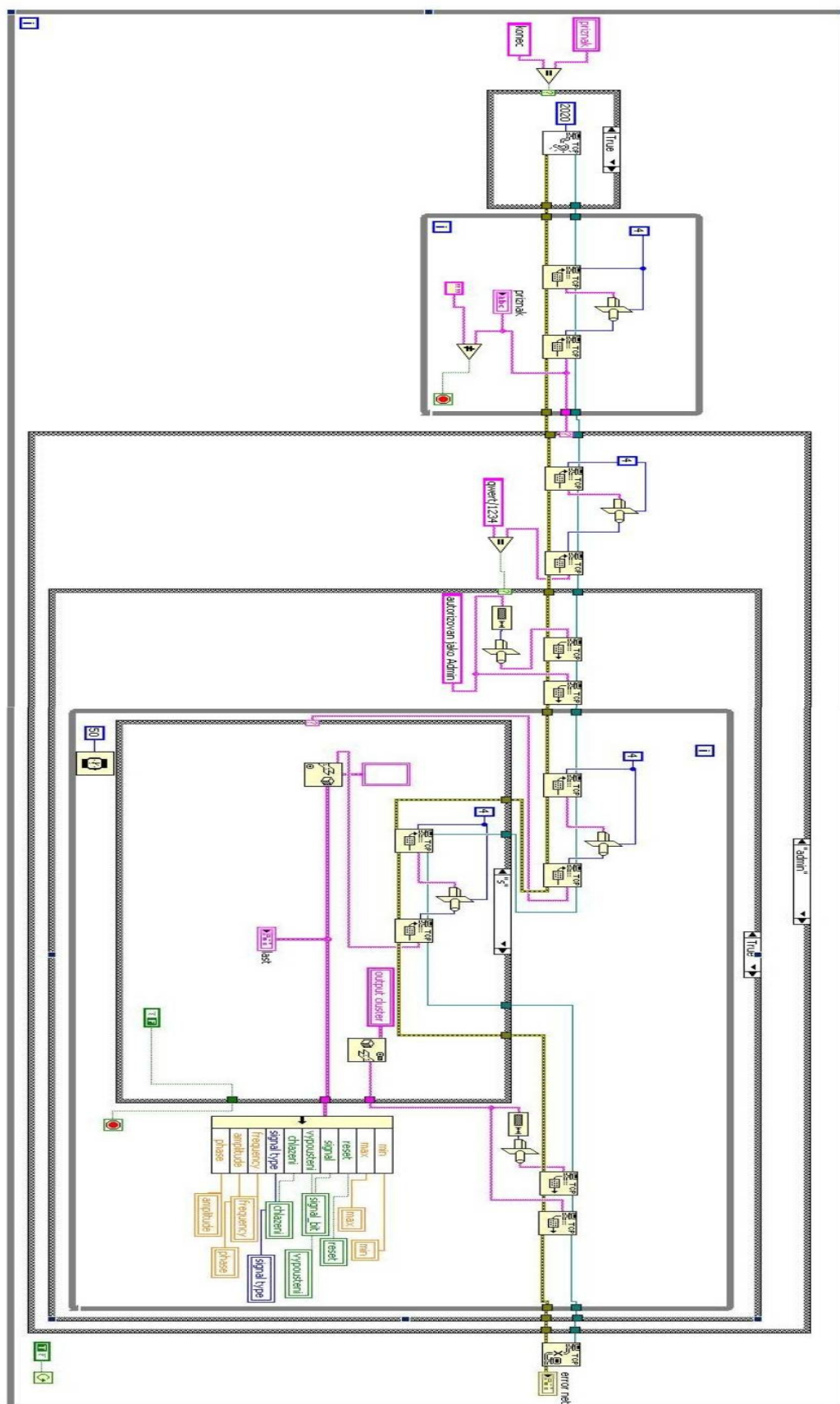
Literatura

- [1.] VLACH, Jaroslav. *Řízení a vizualizace technologických procesů*. 1. vyd. Praha : BEN - technická literatura, 1999. 159 s. ISBN 80-86056-66-X.
- [2.] KOCOUREK, Petr, NOVÁK, Jiří. *Přenos informace*. 1. vyd. Praha : ČVUT, 2006. 164 s. ISBN 80-01-02892-5.
- [3.] Více než 20 let inovací LabVIEW. *Automatizace*. říjen 2007, roč. 50, č. 10, s. 673-674.
- [4.] KMIT, Pavel. PAC - další generace řídicích automatů. *Automatizace*. březen 2005, roč. 48, č. 3, s. 188-189.
- [5.] *National Instruments* [online]. c2008 [cit. 2008-03-10]. Dostupný z WWW: <<http://www.ni.com/>>.
- [6.] NOVÁK, Jiří, KOCOUREK, Petr. *Fieldbus* [online]. c1998 [cit. 2008-03-10]. Dostupný z WWW: <<http://fieldbus.feld.cvut.cz/>>
- [7.] *AS-interface Česká republika* [online]. [2004] [cit. 2008-03-10]. Dostupný z WWW: <<http://as-interface.cz/index-form.html>>.
- [8.] SVOBODA, Jiří. Principy a perspektivy technologie bluetooth. *Sdělovací technika* [online]. 2004, roč. 2004, č. 8 [cit. 2008-03-10]. Dostupný z WWW: <<http://www.stech.cz/articles.asp?ida=357&idk=288>>.
- [9.] *Wavecard - Waveport User Manual*. [s.l.] : [s.n.], 2006. 70 s.
- [10.] *WaveTherm module User handbook*. [s.l.] : [s.n.], [2006?]. 65 s. Version 1.
- [11.] *WaveSense modules User manual*. [s.l.] : [s.n.], [2006?]. 49 s. Version 1.
- [12.] VOTRUBEC, Radek. *LabVIEW for Windows*. [s.l.] : [s.n.], 2000. 47 s.
- [13.] *Wikipedie : otevřená encyklopedie* [online]. 2008 [cit. 2008-03-10]. Dostupný z WWW: <<http://cs.wikipedia.org>>.

Příloha A



Příloha B



Příloha C

