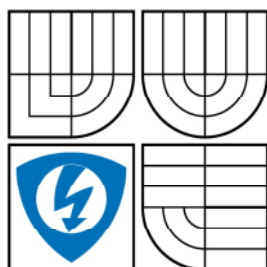


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A
KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘÍCÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

VYTVOŘENÍ KOMUNIKACE MEZI POWER PANELY

CREATING COMMUNICATION BETWEEN POWER PANEL

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

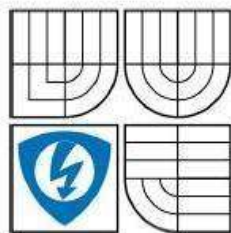
AUTOR PRÁCE
AUTHOR

JAKUB MOTL

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LUDĚK CHOMÁT

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Jakub Motl

Ročník: 3

ID: 77814

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Vytvoření komunikace mezi Power Panely

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi komunikace mezi více Power Panely od firmy B&R pomocí PVI manageru, Ethernetu, Powerlinku, CAN a RS232. Dále se seznamte se základními vizualizačními prostředky B&R. Pomocí PLC s vestavěným dotykovým panelem navrhnete aplikaci, která bude komunikovat s dalšími PLC s vestavěným dotykovým panelem. Vypracujte vzorové řešení činnosti komunikace mezi Power Panely pomocí vytvořené hry pro dva a více hráčů - hada.

DOPORUČENÁ LITERATURA:

B&R Documentation. Automation Net / PVI Online Help, AS240Ee, 1997-2003,
<http://www.br-automation.com>

Termín zadání: 9.2.2009

Termín odevzdání: 1.6.2009

Vedoucí práce: Ing. Luděk Chomát

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení:

Bytem:

Narozen/a (datum a místo):

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
 - ☐ diplomová práce
 - ☐ bakalářská práce
 - ☐ jiná práce, jejíž druh je specifikován jako
- (dále jen VŠKP nebo dílo)

Název VŠKP:

Vedoucí/ školitel VŠKP:

Ústav:

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v * :

- ☐ tištěné formě — počet exemplářů
- ☐ elektronické formě — počet exemplářů

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☐ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....
Nabyvatel

.....
Autor

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav automatizace a měřicí techniky

Vytvoření komunikace mezi Power Panely

Bakalářská práce

Studijní obor: Automatizace a měřicí technika

Student: Jakub Motl

Vedoucí práce: Ing. Luděk Chomát

Abstrakt :

Teoretická část této bakalářské práce se zabývá možnostmi komunikace mezi Power Panely od firmy B&R. Jsou zde rozebrány možnosti komunikace mezi více PLC, vizualizační prostředky, seznam produktů od firmy B&R. Dále obsahuje obecnou literaturu rozhraní RS232, Ethernet, Ethernet Powerlink, CAN, PVI manager.

Praktická část obsahuje vytvoření komunikace s využitím protokolu INA 2000 pro sériové rozhraní RS232, Ethernet s topologií sítě hvězda a vytvoření programového řešení spojeného s návrhem vizualizace hry hada.

Klíčová slova: Power Panel, PVI manager, Powerlink, Ethernet, B&R

Brno University of Technology

Faculty of Electrical Engineering and Communication

Department of Control and Instrumentation

Creating communication between Power Panel

Bachelor's thesis

Specialisation of study: Control and Instrumentation

Student: Jakub Motl

Supervisor: Ing. Luděk Chomát

Abstract :

Theoretic part of my bachelor's thesis deal with possibilities in communication between Power Panels, made by B&R company. Describes possible communication between more PLCs, visual facilities and list of B&R company products. Furthermore some technical literature, RS232 interface, Ethernet, Ethernet PowerLink, CAN, PVI manager.

The Practical part describes how to make a communication using INA 2000 protocol dedicated RS232 interface, Ethernet net with star topology. Furthermore program solution with regard to snake game visualization draft.

Key words: Power Panel, PVI manager, Powerlink, Ethernet, B&R

Bibliografická citace

MOTL, J. Vytvoření komunikace mezi Power Panely. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 65 s. Vedoucí bakalářské práce Ing. Luděk Chomát

P r o h l á š e n í

„Prohlašuji, že svou diplomovou práci na téma Vytvoření komunikace mezi Power Panely jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové 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 § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne :

Podpis:

P o d ě k o v á n í

Děkuji tímto panu Ing. Lud'ku Chomátovi za cenné připomínky a rady při vypracování bakalářské práce.

V Brně dne :

Podpis:

OBSAH

1. ÚVOD.....	9
2. TEORETICKÁ ČÁST.....	10
2.1 VIZUALIZAČNÍ PROSTŘEDKY	10
2.1.1 Power Panel	10
2.1.2 Panelware	11
2.1.3 Typy použitých power panelů.....	12
2.2 PŘEHLED PRODUKTŮ FIRMY BERNECKER&RAINER (B&R).....	15
2.2.1 Dělení produktů B&R	15
2.2.2 Automation Studio	16
2.2.3 Automation Targets	17
2.2.4 Automation Runtime.....	17
2.2.5 Automation Net.....	18
2.3 PVI MANAGER	19
2.3.1 PVI Lines	20
2.3.2 PVI Monitor	20
2.4 ROZHRANÍ RS323	21
2.4.1 Nastavitelné parametry RS232:	21
2.4.2 Řízení toku dat	22
2.4.3 Komunikace RS232	22
2.4.4 Ethernet	23
2.4.5 Master / Slave	24
2.4.6 Polarizace paketů s daty	25
2.4.7 Slot Communication Network Management	25
2.4.8 Komunikace prostřednictvím rozhraní Ethernet TCP/IP	25
2.5 ETHERNET POWERLINK.....	28
2.5.1 Parametry Ethernet Powerlink	28
2.6 CAN (CONTROLLER AREA NETWORK)	31
3. PRAKTICKÁ ČÁST	32
3.1 MOŽNOSTI KOMUNIKACE	32

3.1.1 Navázání komunikace s PVI Managerem.....	33
3.1.2 Odesílání a přijímání žádostí a odpovědí.....	34
3.2 KOMUNIKACE RS232.....	34
3.3 KOMUNIKACE ETHERNET	35
3.4 KOMUNIKACE INA2000	36
3.4.1 Knihovna AsIMA.....	36
3.5 PRAKTICKÉ ŘEŠENÍ PROPOJENÍ MEZI PLC VARIANTA I.	37
3.5.1 Struktura programu INA komunikace pro Slave1, Slave2 varianta I.:	39
3.1 PRAKTICKÉ ŘEŠENÍ PROPOJENÍ MEZI PLC VARIANTA II.....	41
3.1.1 Struktura programu INA komunikace pro Slave1, Slave2 varianta II.:	42
3.2 ZHODNOCENÍ KOMUNIKACÍ	43
3.2.1 Varianta I.	43
3.2.2 Varianta II.	44
4. HRA HAD.....	45
4.1 CÍL HRY	45
4.2 ŘEŠENÍ VIZUALIZACE	45
4.3 OVLÁDÁNÍ HRY	46
4.4 PROGRAMOVÉ ŘEŠENÍ HRY HADA.....	46
4.4.1 Popis funkcí programu hry had.....	47
5. ZÁVĚR	49
6. LITERATURA.....	50
7. SEZNAM PŘÍLOH	51

SEZNAM OBRÁZKŮ

Obr. 1 – Power Panel 4PP420.1043-75 [7]	12
Obr. 2 – Power Panel 4PP480.1043-75 [7]	13
Obr. 3 – Power Panel 4PP481.1043-75 [7]	14
Obr. 4 – Schéma B&R Automation NET	19
Obr. 5 – Schéma průchodu dat zásobníkem	24
Obr. 6 – Hierarchie stromové struktury funkčních bloků pro komunikaci přes Ethernet s využitím knihovny AsTCP	27
Obr. 7 – Začlenění Powerlinku a Automation NET do referenčního modelu ISO/OSI	29
Obr. 8 – Možnosti propojení Power Panelů	32
Obr. 9 – Zpracování žádostí PVI manageru [2]	34
Obr. 10 – Praktické propojení mezi Power Panely varianta I.	37
Obr. 11 – Praktické propojení mezi Power Panely varianta II.	41
Obr. 12 – Architektura programu pro komunikaci Ethernet	42
Obr. 13 – Nastavení komunikačních adres PLC	43
Obr. 14 – Stanice MASTER.....	44
Obr. 15 – Vývojové prostředí pro vizualizaci	46
Obrázek 16 - Funkce programu pro hru hada.....	47

SEZNAM TABULEK

Tab. 1 - signály rozhraní RS232.....	Chyba! Záložka není definována.	21
Tab. 2 - fyzická vrstva Ethernet Powerlink.....		30
Tab. 3 - přehled parametrů Ethernet Powerlink		30
Tab. 4 - parametry CAN.....		31

1. ÚVOD

V úvodu bych rád poukázal na problematiku power panelů od firmy B&R. Jelikož pod pojmem power panel si může mnoho lidí představit různé pojmy, proto bych tento pojem rád uvedl na pravou míru. Také bych chtěl vysvětlit a přiblížit možnosti komunikace mezi power panely, základní vizualizační prostředky, vývojovým prostředím Automation studio, které slouží nejen k samotnému naprogramování programů power panelu, ale i k plánování, implementaci, testování, vizualizaci, jež jsou hojně využívány v průmyslové automatizaci. V neposlední řadě bych rád přiblížil síťové moduly, používané sběrnice (CAN, Ethernet, Ethernet Powerlink) rozhraní PVI a speciální protokol INA2000.

Cílem mého projektu je mimo jiné také vypracování vzorového řešení činnosti komunikace mezi více power panely pomocí mnou naprogramované hry had, která je naprogramována tak, aby ji bylo možné hrát ve dvou či třech lidech. Hrají-li hru dva hráči, komunikují spolu dvě stanice, hrají-li tři hráči, komunikují spolu tři stanice. Lze též hrát samotnou hru při třech hráčích na jednom či dvou power panelech.

2. TEORETICKÁ ČÁST

2.1 VIZUALIZAČNÍ PROSTŘEDKY

2.1.1 Power Panel

Kompaktní inteligentní elektronické zařízení, které slouží k řízení technologických procesů, k tomu využívá vizualizačních prostředků grafické dotykové obrazovky s vysokým rozlišením. Jedná se o kombinaci řídicího systému s operátorským panelem. Toto elektronické zařízení bývá využito především v malých středně velkých systémech například pro řízení strojů technologických procesů apod. Tento prostředek slouží také pro kompletní řešení automatizačních úloh. U power panelů je zachována modularita, pomocí které máme možnost přizpůsobit konfiguraci potřebám dané aplikace.

Přístroje využívají pro komunikaci mezi moduly rozhraní CAN, x2x Link, nebo Ethernet Powerlink.

K programování těchto zařízení slouží integrované vývojové prostředí automation studio, které je popsáno v kapitole 2.2.2

V průmyslové automatizaci se uplatňují stále více metody decentralizovaného řízení, tím dochází k vzrůstu požadavků na hardwarové vybavení zejména dostatečný výkon pro vizualizaci a rychlé komunikační sběrnice. Prvním představitelem je Power Panel 15.

Power Panel 15

Existují dvě varianty systému. Obě obsahují textový displej (LCD, 4x20znaků). První varianta 16 digitálních vstupů a výstupů. Druhá 8 digitálních v/v a 4 analogové v/v.

Power Panel 21 a 41

Tyto panely jsou postaveny na stejném základním systému, liší se pouze provedením čelního panelu. Obsahují textový displej (LCD, 4x20znaků). PP41 má displej o úhlopříčce 5,7" s rozlišením QVGA (240 x 320pixelů).

Power Panel 100

Panel s dotykovým ovládáním. Displeje jsou v dvojím provedení s úhlopříčkou 5,7" LCD černobílý a 10,4"TFT barevný.

Power Panel 200

Je vybaven stejně jako PP100 procesorem Geode 266MHz, navíc má 64MB SDRAM a 256kb SRAM. Varianty displejů 5,7"LCD; 10,4"TFT; 15" TFT.

2.1.2 Panelware

Kompaktní terminál v kombinaci s kontrolním systémem jsou ideálním řešením pro úsporu místa. Je možné je využít v širokém spektru aplikací. Od jednoduchého ovládání strojů – zadávání parametrů pro danou výrobu, přes lokální obsluhu jednotlivých uzlů různých technologických celků. U těchto systémů se využívá modularity, která zajišťuje optimální konfiguraci pro danou aplikaci.

Dálkové vizualizace PanelWare – operátorské panely mohou být umístěny přesně tam, kde se zobrazují stavové zprávy a provozní údaje daného stroje a jsou snadno přístupné pro nastavení stroje.

Kompaktní operátorské panely – představují zařízení, která mají minimální požadavky na prostor. Tyto panely jsou vybaveny alfanumerickým LCD displejem a jsou ovládány pomocí sekvencí.

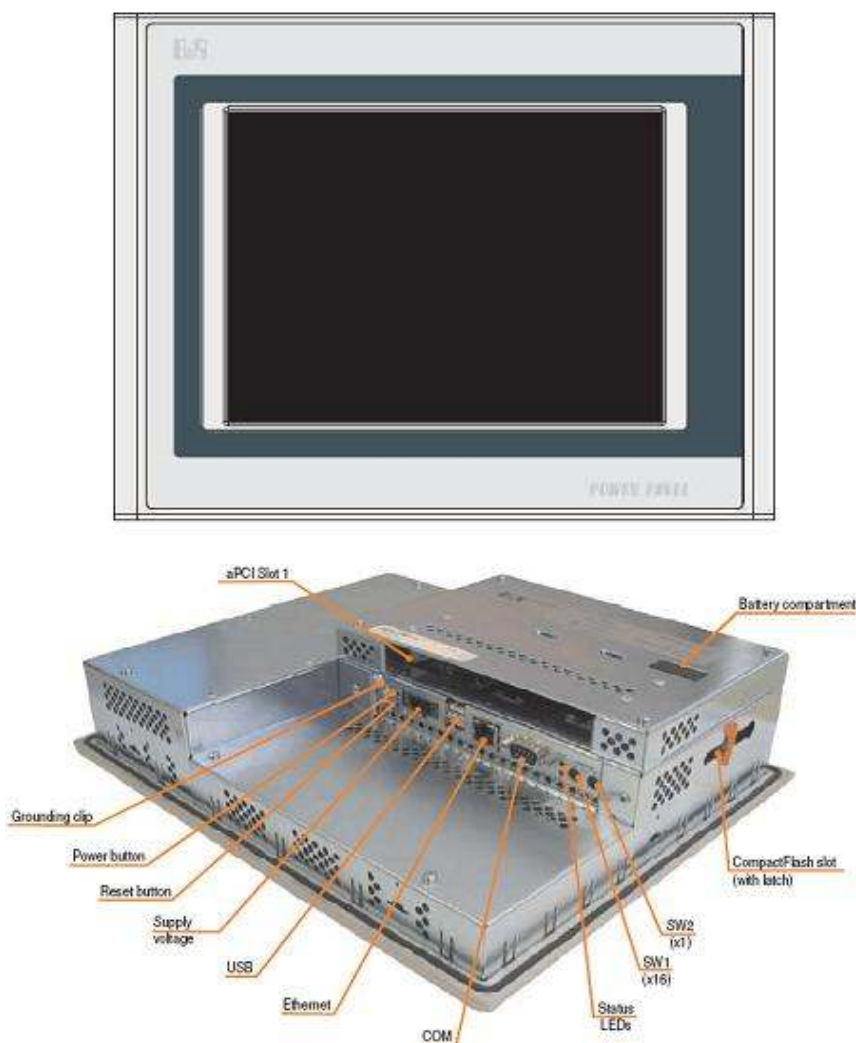
Panel controller – řadiče jsou samostatné moduly. Dokonalá konfigurace na základě požadavků, může být dosažena pomocí připojení jiných grafických panelů.

[1]

2.1.3 Typy použitých power panelů

Power Panel PP420.1043-75 nabízí:

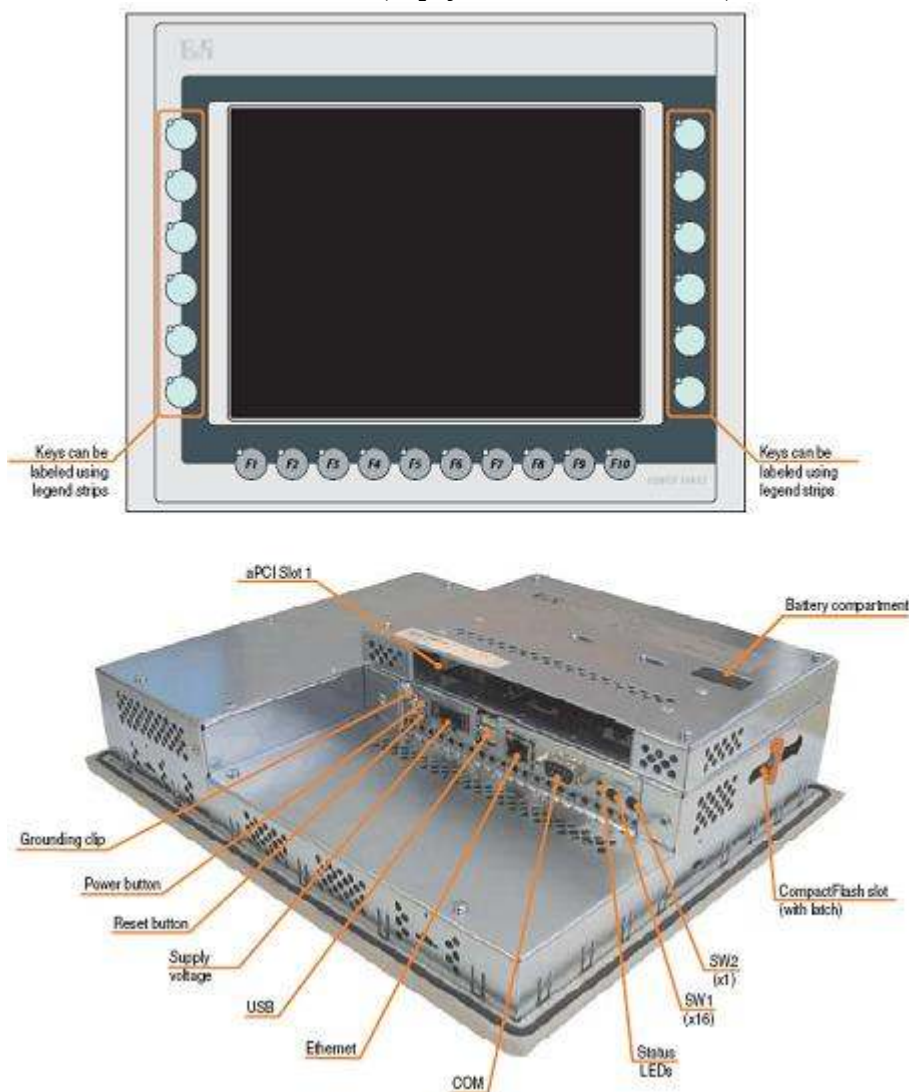
- 10,4 "VGA barevný TFT displej s dotykovou obrazovkou
- 1 aPCI slot, 128 MB SDRAM, 512 kB SRAM; Compact Flash slot - slot pro výměnné paměťové karty
- Ethernet 10/100, sériové rozhraní RS232 a 2xUSB
- na rozdíl od panelů 4PP480 a 4PP481 neobsahuje žádná funkční tlačítka
- Akumulátor 24VDC (napájecí svorka 0TB103.9)



Obr. 1 – Power Panel 4PP420.1043-75 [7]

Power Panel 4PP480.1043-75 obsahuje:

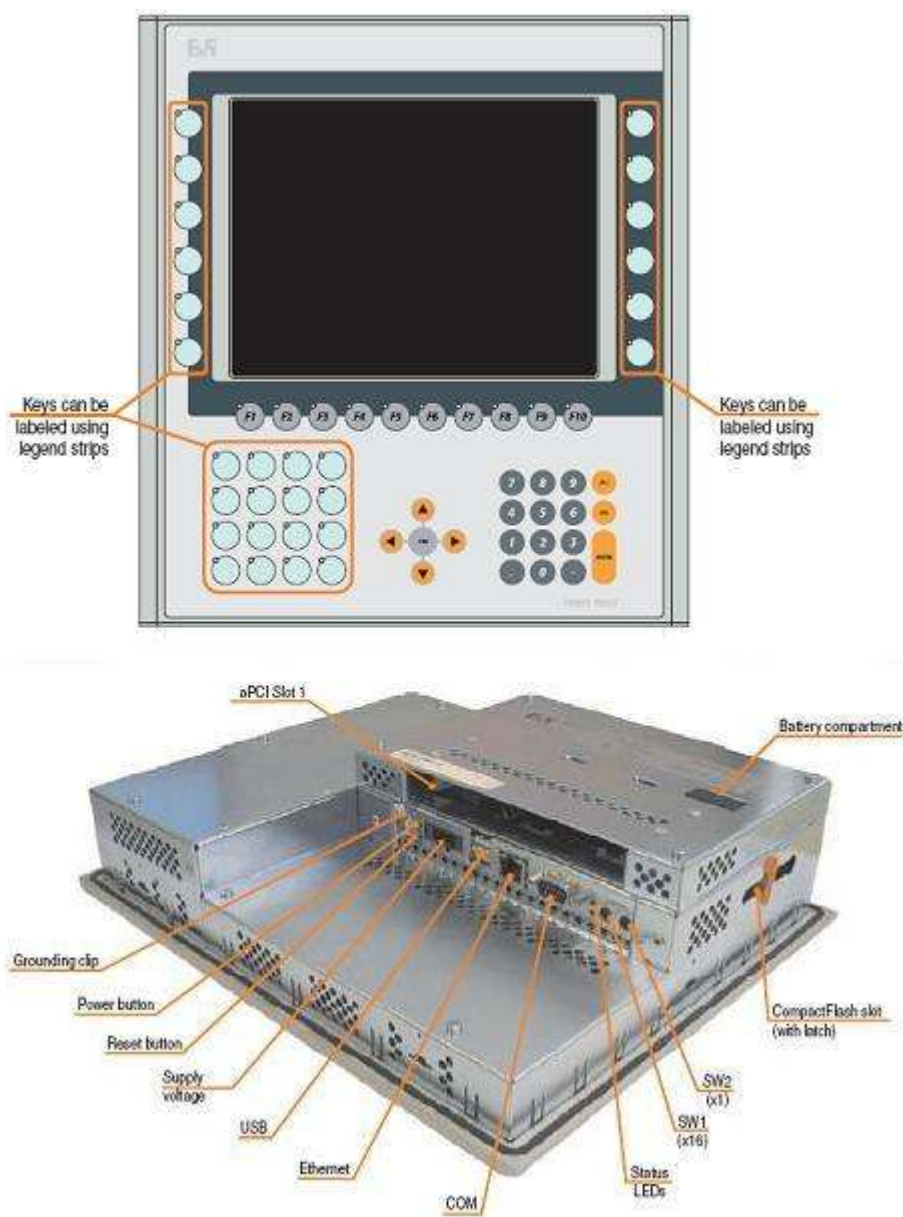
- 10,4 "VGA barevný TFT displej s dotykovou obrazovkou
- 12 funkčních tlačítek + 10 tlačítek pro snadné spouštění
- 1 PCI slot, 128MB SDRAM, 512kB SRAM,
- 1 slot pro Compact flash (výměnné karty)
- ETH 10/100, sériové rozhraní RS232, 2x USB
- Akumulátor 24VDC (napájecí svorka 0TB103.9)



Obr. 2 – Power Panel 4PP480.1043-75 [7]

Power Panel 4PP481.1043-75 obsahuje:

- 10,4 "VGA barevný TFT displej s dotykovou obrazovkou
- 10 soft tlačítek, 28 funkčních tlačítek, a 20 systémových kláves
- 1 aPCI slot, 128 MB SDRAM, 512 kB SRAM slot pro výměnné karty
- ETH 10/100; RS 232, 2xUSB
- Akumulátor 24VDC (napájecí svorka 0TB103.9)



Obr. 3 – Power Panel 4PP481.1043-75 [7]

2.2 PŘEHLED PRODUKTŮ FIRMY BERNECKER&RAINER (B&R)

2.2.1 Dělení produktů B&R

Produkty nabízející komplexní řešení pro automatizaci společnosti B&R můžeme začlenit do těchto skupin.

- Automation software
- Automation Net
- Automation Runtime
- Automation Targets

Ve skupině Automation software se nachází vývojové prostředí Automation studio, které slouží pro programování, konfiguraci a diagnostiku dané aplikace. Automatizační úlohy požadují vhodný programovací nástroj.

Automation Net slouží výhradně ke komunikaci mezi navzájem propojenými účastníky. To znamená vzájemnou výměnu dat mezi systémy. Pomocí různých sběrnic (CAN, Profibus, až po Ethernet TCP/IP), sériových linek, pro aplikace s nejvyššími požadavky na rychlost, přijatelnou časovou deterministickou odezvu, kterou nám nabízí Ethernet Powerlink.

Automation Runtime nám vytváří mezičlánek mezi Automation Software a Automation Targets. Ke spojení těchto dvou systémů slouží již zmiňovaný Automation Net.

Automation Target je hardware na kterém běží operační systém Automation Runtime.

Automation Target se dělí:

- Control Systems – řídicí systémy možnost úpravy chování jiných zařízení nebo systémů
- Panel Systems – na panely od malých kompaktních operátorských panelů po displeje s modulárními IPC
- Motion Systems – pohony a motory pro dynamické polohování

2.2.2 Automation Studio

Vývojové prostředí od společnosti B&R. V Automation Studio je možné psát, ladit programy pro všechny produkty B&R. Základem celého systému je možnost tvorby programů, vizualizace, polohové regulace, komunikace po více rozhraních. Grafické zobrazení konfigurace hardwaru a softwaru nabízí jasný přehled v projektu. Hardwarová konfigurace cílového hardwaru je rozpoznána automaticky nebo ručně.

Tvorba programů je možná v těchto standardních jazycích:

- Automation Basic
- ANSI-C
- Jazyky podle IEC 61131-3
- Strukturovaný text (ST)
- Sekvenční vývojové diagramy (SFC)
- Kontaktní schémata (LAD)

Tento výběr programovacích jazyků lze implementovat do všech cílových systémů B&R. Poskytuje velkou flexibilitu na všechny programové požadavky. U B&R jsou všechny vytvořené součásti kompatibilní pro různé typy řídicích procesorů. Jestliže změníme typ řídicího systému, vyžaduje to minimální změny v naprogramovaném zdrojovém kódu.

Editor procesních obrazovek s plnou podporou grafiky se automaticky adaptuje na používaný hardware. Obsahuje společnou datovou oblast pro řízení i vizualizaci. Možnosti Visual Components:

- 256 barev (internetový standard)
- Rastrové obrázky, tlačítka, čáry, elipsy, kruhy, vyplněné oblasti s dynamickými změnami barev, text
- Vstupní a výstupní pole (alfanumerická, číselná, sloupcové grafy, rastrové obrázky)
- Číselná klávesnice nebo klávesnice na dotykové obrazovce
- Programování nestandardních grafických výstupů funkčními bloky VISAPI
- Dynamické operace (zpracování hodnot, barev a atributů)

2.2.3 Automation Targets

Jedná se o hardwarovou platformu, na které běží operační systém Automation Runtime. Řídící jednotky jsou složeny z různých komponent, které se dají libovolně kombinovat (řídící jednotka, komunikační moduly, atd.).

2.2.4 Automation Runtime

Operační systémy používané pro programovatelné automaty B&R se nepatrně liší ve vnitřní struktuře, v závislosti na použitém konkrétním hardwaru programovatelného automatu, avšak navenek mají stejné charakteristické vlastnosti.

Tento systém je upravený speciálně pro potřeby řídicí techniky. U standardních systémů, které nejsou určeny pro řízení v reálném čase, je časový průběh jednotlivých úloh ovlivněn určen prioritou úloh. Změny ve verzi operačního systému je možno provádět ve vývojovém prostředí Automation Studio. V editoru projektu Automation Studia je uvedena stromová struktura hardwarové konfigurace aktuálního projektu. Ve stromové struktuře se nachází centrální procesorová jednotka (CPU).

Cyklické úlohy (cyclic tasks) – programy, které má programovatelný automat provádět. Uživatel zde může nadefinovat dobu cyklu. Tato doba nám udává čas, ve kterém se cyklicky vykonává program. Počet různých dob cyklu je omezen v závislosti na cílovém hardwaru. Úkolem tohoto operačního systému je zajištění a správné zpracování příslušných úloh v daném čase. Mimo doby cyklu můžeme nastavit i časovou toleranci, která nám umožňuje kontrolu nad skutečným časovým průběhem všech úloh. Doba cyklu je hodnota nastavená uživatelem pro jednotlivé cyklické třídy. Doba cyklu nepodává informaci o skutečné době průběhu, která je třeba k vykonání všech úloh obsažených v dané cyklické třídě. Definujeme-li čas, za který je úloha provedena jako t v cyklické třídě y jako Ty_t , nastavitelnou dobu cyklu třídy y jako $T_{cyklu}y$ a toleranci τ , musí platit pro všechny cyklické třídy tento vztah

$$\sum Ty_t \leq T_{cyklu}y + \tau$$

Automation Runtime lze nastavit tak, aby při nesplnění této podmínky při vykonávání programu nahlásil chybu.

Vlastnosti operačního systému Automation Runtime:

- Rozdílné nastavitelné doby cyklu s možností kontroly
- Deterministický multitasking
- Paralelní zpracování více úloh

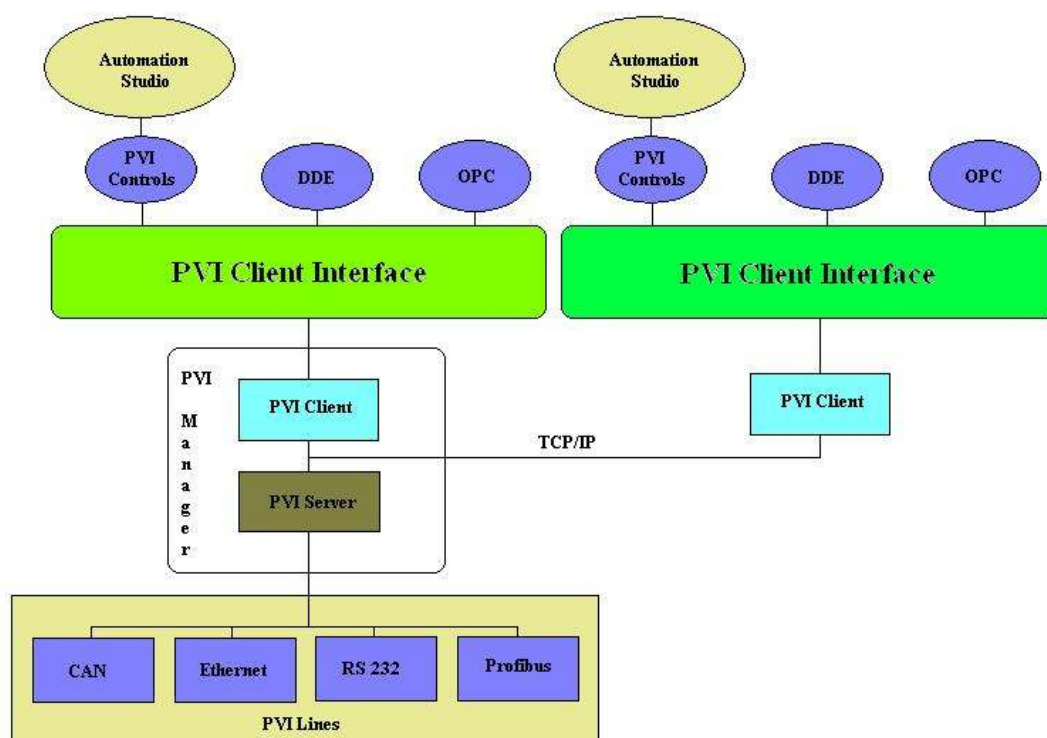
2.2.5 Automation Net

Automatizační síť slouží ke komunikaci a výměně dat mezi jednotlivými řídicími systémy (Automation Targets), Automation Studiem a dalšími I/O moduly.

Automatizační síť používají všechny součásti systému. Každá stanice v síti musí být schopna vyměňovat a zpracovávat programové objekty a procesní data nezávisle na cílovém operačním systému (Windows, operační systém řídicích systémů), používaném protokolu (INA2000, Net2000, Mininet a další).

Moderní automatizace vyžaduje sdílení dat napříč sběrnici. Automation Net je plně integrován do všech operačních systémů zařízení B&R a slouží pro vzájemnou výměnu dat mezi systémy, diagnostiku a programování.

Zajišťuje propojení různých síťových standardů do jedné sítě. Proč je to nutné? Ve spoustě firem jsou používány jiné sběrnice souběžně, pro různé aplikace jsou některé sběrnice vhodnější některé méně vhodné. Automation Net nám zajišťuje sdílení napříč těmito sběrnici. V této platformě lze komunikovat po sériové lince RS232, sběrnici CAN, Profibus, Ethernet TCP/IP, pro větší požadavky na deterministickou odezvu Ethernet Powerlink. [10][2]



Obr. 4 – Schéma B&R Automation NET

2.3 PVI MANAGER

Slouží k vizualizaci procesů, zároveň je to součást automatizační sítě (Automation Net). Vytváří společné rozhraní pro B&R průmyslové PC. To má za následek zajištění komunikace mezi průmyslovým prostředím a operačním systémem PC. Prostřednictvím knihovny DLL je možné využívat získaná data v kancelářském balíku MS Office. PVI umožňuje propojit různé architektury za pomoci DDE serveru (dynamická výměna, slouží ke sdílení proměnných mezi několika aplikacemi na jednom PC př. Excel, Acces, toto sdílení se nejvíce používá k získávání dat z PLC automatů), OPC serveru (rozhraní pro aplikace, které jsou zaměřeny na řízení a monitorování rychlých procesů, lepší a rychlejší obdoba DDE). Pomocí PVI klient/server je možné provádět vzdálené operace. PVI jako klient může přenášet data, programy z nebo do PVI serveru pomocí sítě, nebo dalších komunikačních médií. Knihovny PVI umožňují vytvářet nástroje pro komunikaci s automatem (klienty) pomocí standardních programovacích nástrojů (MS Visual C++, Borland

C++, Delphi), tím dochází k urychlení a zjednodušení aplikací pro vizualizaci a aplikací, které jsou určeny k řízení programovatelného automatu.

- PVI manager je zodpovědný za řízení všech procesních dat
- Využívá specifikace z aplikace tzn. zajišťuje, které data jsou přenášena přes, které zařízení a medium z/do stanice.
- Asynchronní úlohy musí být schopny používat síťové zpoždění a řízení zpracování různých úloh.

2.3.1 PVI Lines

Základním úkolem PVI Lines je připojení PVI objektů s PVI objekty, které jsou mimo PVI, zároveň je zodpovědný za dohled nad komunikací. Určuje komunikační protokol, který má být použit.

2.3.2 PVI Monitor

Pomůcka pro zobrazování provozních informací, také ke konfiguraci funkce diagnostiky a globálních vlastností pro PVI manager. Cyklicky generuje žádost o informace z PVI Manager.

2.4 ROZHRAŇÍ RS323

Toto rozhraní slouží pro přenos informací původně pro komunikaci dvou zařízení do maximální vzdálenosti 20 metrů. Z důvodu větší odolnosti vůči rušení je informace přenášena po propojovacích vodičích větším napětím než je standardních 5V. Přenos informací probíhá asynchronně pomocí pevně nastavené přenosové rychlosti, která se pohybuje od 9,600 do 115,200 baud a synchronizace sestupnou hranou startovacího impulsu.

Tab. 1 – signály rozhraní RS232

Popis signálů	
DCD - Data Carrier Detect	Detekce nosné. Modem oznamuje terminálu, že na telefonní lince detekoval nosný kmitočet
RXD - Receive Data	Tok z modemu (DCE) do terminálu (DTE).
TXD - Transmit Data	Tok dat z terminálu (DTE) do modemu (DCE).
DTR - Data Terminal Ready	Terminál tímto signálem oznamuje modemu, že je připraven komunikovat.
SGND - Signal Ground	Signálová zem
DSR - Data Set Ready	Modem tímto signálem oznamuje terminálu, že je připraven komunikovat.
RTS - Request to Send	Terminál tímto signálem oznamuje modemu, že komunikační cesta je volná.
CTS - Clear to Send	Modem tímto signálem oznamuje terminálu, že komunikační cesta je volná.
RI - Ring Indicator	Indikátor zvonění. Modem oznamuje terminálu, že na telefonní lince detekoval signál zvonění.

2.4.1 Nastavitelné parametry RS232:

- Rozhraní COM 1 – COM16
- Přenosové rychlosti pro sériovou komunikaci
- Přenosové rychlosti 9,600; 19,200; 38,400; 57,600; 115,200 baud

Parita

- Lichá parita – počet jedničkových bitů + 1 paritní = Liché číslo
- Sudá parita – počet jedničkových bitů + 1 paritní = Sudé číslo
- Nulová parita – paritní bit je vždy v log. 0

Časový interval (ms) – maximální čas (ms), který může uplynout mezi obdržení dvou po sobě jdoucích znaků. Tento čas se používá pro komunikaci mezi PC a cílovým systémem.

2.4.2 Řízení toku dat

Lze použít k zajištění kontinuálního přenosu dat mezi stanicemi v síti, které nepracují synchronně. Pokud vysílač pracuje s pomalejším přijímačem, nesmí dojít ke ztrátě dat. Tento typ řízení toku dat se nazývá Handshake, pracuje na hardwarové nebo softwarové úrovni.

Hardwarový Handshaking – řízení toku dat je pomocí RTS/CTS linky. Pokud je zaplněno určité procento z obdrženého bufferu, CTS (clear to send) se vypíná. Vzájemné stanice by pak měly zastavit odesílání dat, dokud se CTS opět nezapne. Pokud stanice není připravena na příjem dat, RTS se vypíná a tím je zastaveno odesílání dat.

Softwarový Handshaking – probíhá na úrovni komunikačních protokolů. Pomocí běžného datového kanálu přijímač sdělí vysílači, zda je schopen data přijmout a zpracovat. Jestliže se vysílá Handshaking pomocí Ascii znaku XON/XOF je nutné vyslat speciální sekvenci znaků, což výrazně zpomaluje přenos dat.

2.4.3 Komunikace RS232

Pro výměnu dat s externími zařízeními, které nemohou komunikovat prostřednictvím standardní sítě (operátorské panely, monitory, tiskárny, atd.) se

využívá knihovny DVFrame. Informace je přenášena pomocí rámců. DVFrame driver může být realizován jako (Master/Slave, nebo Master/Master).

Knihovna DVFrame obsahuje:

FRM_xopen() funkce pro inicializaci rozhraní. Pomocí této funkce může být inicializováno a připraveno k přenosu až 16 rozhraní. Jakmile bylo inicializováno rozhraní, mohou se používat funkce FRM_write() a FRM_read().

FRM_mode() funkce, která upravuje parametry rozhraní a to paritu, stop bit, ale také dobu nečinnosti.

FRM_read() funkce pomocí, které se čtou data a ukládají se do (obdrženého) bufferu
FRM_rbuf() funkce zajistí uvolnění obdrženého bufferu poté co byla načtena data.
pomocí fce FRM_read().

FRM_gbuf() funkce slouží pro odeslání odpovědi bufferu. Odesílá adresu bufferu.

FRM_write() funkce přenáší obdržená data do bufferu. Umožní Frame driveru posílat data do bufferu.

FRM_robuf() funkce uvolňuje buffer

FRM_close() funkce deinicializuje rozhraní

FRM_ctrl() funkce, která slouží pro ovládání rozhraní (DTR/DSR)

[5][9]

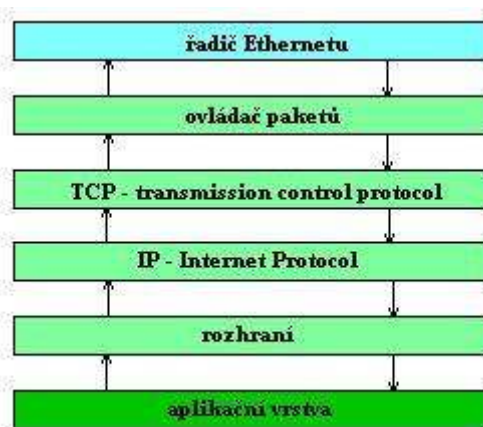
2.4.4 Ethernet

Ethernet je sériová sběrnice vyvinutá na konci 70. let firmou Xerox. Ethernet využívá původně jen jedné specifikace s přenosovou rychlostí 10 Mb/s, přenos v základním pásmu, délka segmentu je do 500 m, topologie sběrnice a padesáti-ohmový tlustý koaxiální kabel.

K původnímu Ethernetu se již delší dobu využívají i další varianty, přizpůsobené tenkému koaxiálnímu kabelu a kroucené dvoulince. Rychlý (Fast) Ethernet – řešení fyzické vrstvy umožňuje rychlost 100Mb/s. Teprve tyto vysoké

rychlosti a úpravy topologie sítí Ethernet předurčují tento systém také pro průmyslovou komunikaci.

Ethernet TCP/IP – přenášená data jsou nejdříve „komprimována“ pomocí TCP poté IP. Přijímaná data musí být opět „dekomprimována“. To zajišťuje zásobník stack (buffer). Celý proces přenášení paketů je ale časově velice náročný.



Obr. 5 – Schéma průchodu dat zásobníkem

Možnosti částečného řešení determinismu jsou:

- Použití principu master – slave
- Polarizace paketů s daty pomocí IEEE 802.1Q/802. 1p.
- Využití řízení přístupu na síť Slot Communication Network Management

2.4.5 Master / Slave

Master řídí celou komunikaci (řídící stanice), Slave představuje (stanici řízenou) může být tvořena pomocí protokolu TCP/IP nebo UDP/IP. Tím že master řídí celou komunikaci, nedochází zde ke kolizím, nevýhodou je nemožnost přímé komunikace mezi dvěma zařízeními, ale prostřednictvím nadřazené stanice. Schůdnější varianta je pomocí Slot Communication Network Management.

2.4.6 Polarizace paketů s daty

Polarizace podle doporučení IEEE 802. 1Q(802. 1p, je umožněna pomocí rozšíření hlavičky protokolu. Tok dat je pak řízený pomocí aktivních prvků přepínačů (switchů). Tím jsou přednostně přenášeny pakety s vyšší prioritou. Není však zabezpečené pořadí přenesených telegramů. Telegramy však mohou být uspořádány pomocí TCP/IP protokolu.

2.4.7 Slot Communication Network Management

Na tomto principu je založená Ethernet Powerlink firmy B&R. Jedná se o úpravu principu Master / Slave. Nedochozí zde k rozptylu časování telegramů. Každé zařízení má přesně definovanou dobu na vysílání, po kterou může vysílat data kterékoliv stanici i stanicím v síti. Ke kolizím pak nemůže dojít a šířka přenosového pásma sběrnice je optimálně využita.

2.4.8 Komunikace prostřednictvím rozhraní Ethernet TCP/IP

Pro komunikaci přes ethernetové rozhraní využíváme podobně, jako tomu bylo u rozhraní RS232 knihovnu DVFrame.h u ethernetového rozhraní se jedná o knihovnu AsTCP.h, která obsahuje řadu funkcí pro navázání komunikace a přenos dat mezi zařízeními. [3][8][9]

Knihovna AsTCP.h

Knihovna slouží k výměně dat pomocí socketů. To umožňuje jednoduchou komunikaci klient/server aplikace je naprogramována v jazyce C.

V této knihovně pracují všechny funkční bloky asynchronně.

TCP je protokol transportní vrstvy. Zajišťuje spolehlivý přenos dat, zajišťuje spojení mezi klienty a servery. Obstarává tedy to, aby byla data správně obdržena. TCP také třídí data. Jestliže jsou data obdržena v nesprávném pořadí, mohou být zpětně seřazena pomocí pořadového čísla.

Knihovna obsahuje tyto funkce:

TcpOpen() otevírá TCP socket. Tím se určuje, jestli se jedná o TCP Server nebo TCP klienta.

TcpServer() navazuje komunikaci se serverem, má definované číslo portu, které potřebuje znát TCP klient, aby došlo k navázání spojení.

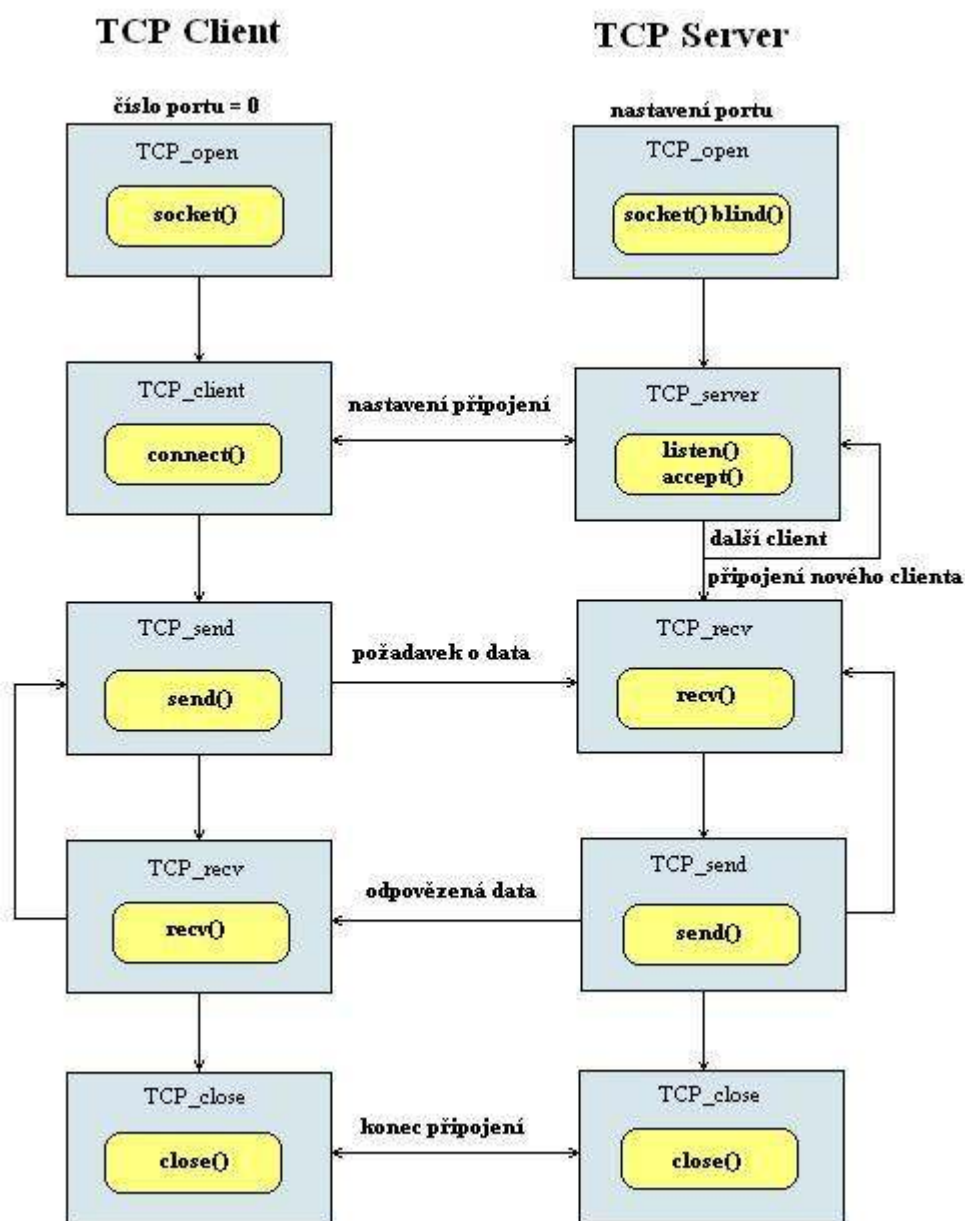
TcpClient() používá se k nastavení připojení klienta k TCP serveru.

TcpClose() ukončuje (zavírá) TCP socket

TcpSend() slouží k odesílání dat

TcpRecv() slouží pro příjem dat

TcpIoctl() umožňuje speciální nastavení, používá se pro změnu nebo načtení nastavení.



Obr. 6 – Hierarchie stromové struktury funkčních bloků pro komunikaci přes Ethernet s využitím knihovny AsTCP

2.5 ETHERNET POWERLINK

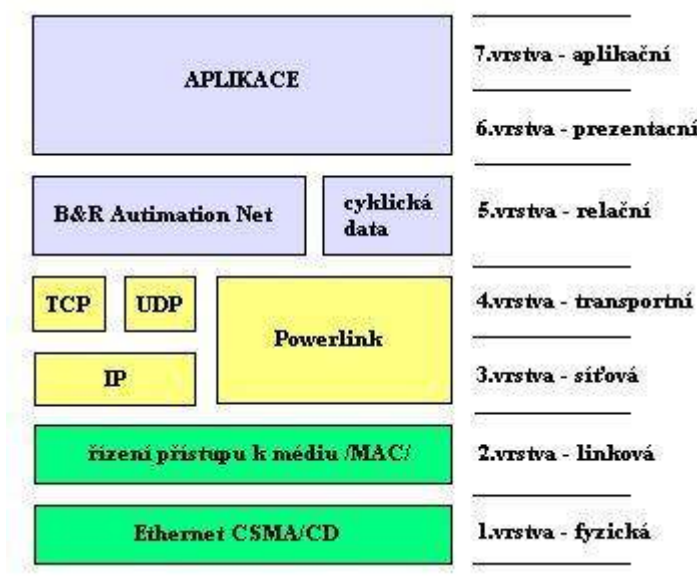
Ethernet Powerlink je deterministický real-time protokol pro standardní Ethernet. Je to otevřený protokol, který je spravován Ethernet Powerlink Standardization Group (EPSG). Byl zaveden rakouskou společností B&R v roce 2001.

Ethernet Powerlink využívá ke své komunikaci ethernetové standardy, které jsou v dnešní době hodně rozšířeny, to má za následek nižší cenu. Ethernet není dokonalý hlavně díky časovému nedeterminismu, což je způsobeno řízením přístupu na sběrnici CSMA/CD. Zařízení sleduje provoz na sběrnici, jestliže není sběrnice obsazena, zařízení začne vysílat data. Pokusí-li se vysílat dvě zařízení současně, dochází ke kolizi, pak se obě zařízení odmlčí na určitou dobu, která je definována pomocí náhodného algoritmu, kde po uplynutí určité doby se pokusí znovu odeslat svá data. Celý cyklus se opakuje tak dlouho, dokud nejsou všechna data správně přenesena. Proto také firma B&R přišla s novým řešením – Ethernet Powerlink disponuje vysokými přenosovými rychlostmi s minimálním rozptylem časových telegramů. Ethernet Powerlink, využívá k řízení přístupu na síť Slot Communication Network Management, u kterého má každé zařízení přesně danou dobu kdy může vysílat data kterékoliv stanici v síti. Tím pádem nemůže dojít ke kolizi a šířka pásma sběrnice je tak plně využita. Powerlink mimo rychlé cyklické výměny dat zajišťuje i acyklickou výměnu dat, ta ale nesmí cyklickou výměnu nijak omezit. Ethernet Powerlink je vhodný pro synchronizaci mezi v/v zařízeními.[1][6]

2.5.1 Parametry Ethernet Powerlink

Nastavitelná doba cyklu sběrnice je 400 μ s, v této periodě může být obslouženo 8 zařízení. V 500 μ s, je to asi 10 zařízení, v 1 ms je to až 30 zařízení. Z celkového maximálního počtu připojitelných zařízení, kterých může být až 254 typu slave a jeden Slot Communication Network Manager. U samotné sběrnice je doba cyklu menší než 1 μ s což v porovnání se sběrnici CAN při přenosové rychlosti 500kb/s asi 254 μ s. Používané konektory jsou RJ45 a kabeláž s kříženými žilami. Délka segmentu je asi 100m. V porovnání s kabeláží rozhraní RS232, u kterého

maximální délka segmentu dosahuje pouhých 15m. Topologie sítě u Ethernet Powerlinku může být různá, jelikož je to deterministický systém pro příklad topologie hvězdy nebo stromu. [6][9]



Obr. 7 – Začlenění Powerlinku a Automation NET do referenčního modelu ISO/OSI

Na obrázku 7 je nastíněno přiřazení Ethernet Powerlinku a Automatizační sítě firmy B&R do komunikačního modelu ISO/OSI. Jak je vidět na obrázku aplikace získává data pomocí Automation net nebo skrze rychlou cyklickou výměnu dat přes Ethernet Powerlink. Ethernet Powerlink zastřešuje transportní a síťovou vrstvu tím tak nahrazuje dvojici vrstev TCP/UDP a IP. Jak je vidět z obrázku, B&R Automation Net může využívat jak služeb Ethernet Powerlinku, tak i komunikační protokoly TCP/IP nebo TCP/UDP.

Tab. 2 – fyzická vrstva Ethernet Powerlink [6]

Délka segmentu	Max. 100 m
Topologie	Hvězda nebo strom s úrovní dvou Hubů
Kabel	Twisted Pair, S/UTP, AWG26
Typ kabelu	Cat. 5 patch cable ('crossover cable'), jeden pro všechny spoje

Tab. 3 – přehled parametrů Ethernet Powerlink [6]

Přenos	Standard Fast Ethernet, IEEE 802.3u, 100Base-TX
Přenosová rychlost	100 Mbps
Protokol	Slot Communication Network Management (SCNM)
Podporované typy komunikace	cyklická, override, acyklická
Min. Net Frame	46 bytes
Min. Net Data pro stanici	1 byte
Max. Net Data pro stanici	1500 bytů
Počet stanic v síti Powerlink	Max. 253
Doba cyklu	může být definována
Počet stanic při cyklu 400 μ s	8 cyklických + n override stanic (80 bytů net data/stanici)
Počet stanic při cyklu 800 μ s	22 cyklických + n override stanic (80 bytů net data/stanici)
Možné operační módy stanic	synchronně, asynchronně k cyklu Powerlinku
Hardwarový Watchdog	Ano
Softwarový Watchdog	Ano
Elektrická izolace mezi sběrnici a I/O	Ano

2.6 CAN (CONTROLLER AREA NETWORK)

Komunikační protokol, který byl vyvinut firmou Bosch, postupně se však rozšířil díky své nízké ceně, spolehlivosti, vysoké přenosové rychlosti do různých průmyslových aplikací. Datová komunikační síť CAN původně používala modifikované rozhraní RS 485, později bylo definováno normou ISO. Je koncipován tak, aby umožnil řízení v reálném čase, jeho přenosová rychlost dosahuje 1Mbit/s se zabezpečením přenosu proti chybám. Spolehlivost je dána tím, že tento protokol může pracovat jako multi-master, jestliže tedy dojde k poruše jednoho uzlu, zbytek sítě je schopen pracovat dál. Z hlediska determiniscence tento protokol používá náhodný přístup k médium, který řeší kolize na základě prioritního rozhodování. Komunikace mezi dvěma zařízeními probíhá pomocí datové zprávy a žádosti o data, obsahuje taktéž jakousi organizaci sítě (pozastavení komunikace, signalizaci a chybové zprávy a zprávy o přetížení). Zprávy, které jsou vysílány po sběrnici, neobsahují informaci o tom, komu jsou určeny, jsou přijímány všemi uzly (stanicemi) připojeným ke sběrnici. Zprávy obsahují identifikátor, ten nastavuje prioritu doručení. Pomocí identifikátoru je zajištěno to, že daná stanice obdrží pouze ta data, o které požádala. [9]

Tab. 4 - parametry CAN

Přenosová rychlost	125 kBit/s až 1 Mbit/s
Počet uzlů v síti	max. 30
Přenosová rychlost:	odpovídající maximální délka:
1 Mbit/s	40 m
500 kbit/s	112 m
300 kbit/s	200 m
100 kbit/s	640 m
50 kbit/s	1340 m
20 kbit/s	2600 m
10 kbit/s	5200 m

3. PRAKTICKÁ ČÁST

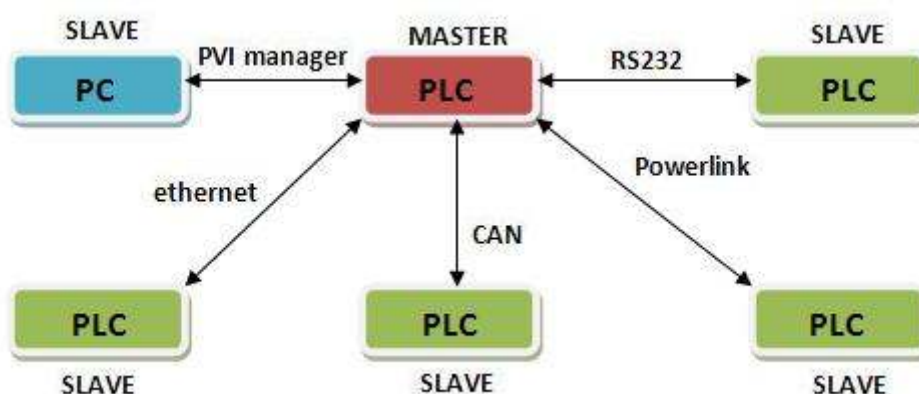
3.1 MOŽNOSTI KOMUNIKACE

Pod pojmem komunikace rozumíme souhrn řídicích činností umožňujících výměnu dat mezi komunikujícími systémy. Koordinaci komunikace nám zajišťují protokoly, které definují formální stránku komunikace. Protokoly jsou soubory pravidel, formátů procedur, které určují výměnu dat mezi dvěma či více komunikujícími zařízeními. K fyzickému propojení zařízení slouží síťové moduly a provozní sběrnice, ty zajišťují flexibilní komunikaci mezi zařízeními. Rozsáhlé komunikační možnosti jsou základním požadavkem pro všechny automatizační řešení. Nejvíce používaná komunikace je Ethernet, Ethernet Powerlink a CAN.

Dnes většina procesorů bývá vybavena integrovaným rozhraním - Ethernetem 10/100Mbps.

V průběhu řešení komunikace mezi power panely se naskytlo hned několik variant řešení.

Komunikace se musí přizpůsobit požadavkům na danou aplikaci, proto existuje množství modulů rozhraní.



Obr. 8 – Možnosti propojení Power Panelů

Obrázek 8 popisuje možnost propojení Power Panelů ve hře had, které je věnována kapitola 4.6. Topologie hvězda, obsahuje master, ve kterém je nahrán

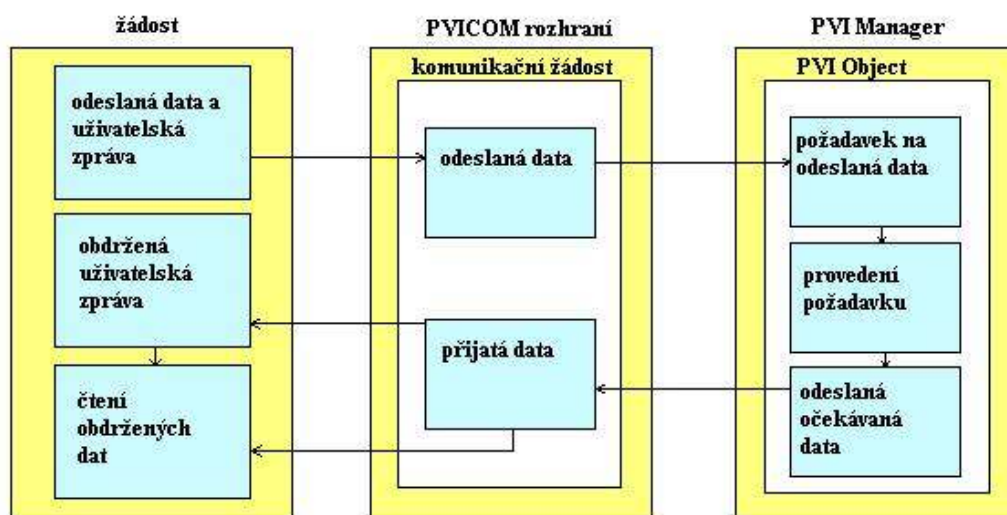
samotný program s vizualizací. Stanice slave obsahují program hry, program pro zřízení komunikace a program s názvem calib, který slouží ke kalibraci dotykové obrazovky PLC.

3.1.1 Navázání komunikace s PVI Managerem

Využívá se zde knihovna PviCom.dll. Pro navázání komunikace s PVI Managerem musíme být schopni nejdříve nastavit komunikační žádost. Komunikační žádostí nastavíme spojení mezi PVICOM a PVI Managerem. To zajistíme voláním funkce PviInitialize() nebo PviXInitialize() v PVICOM. Touto funkcí stanovíme způsob komunikace (místní nebo vzdálenou) a časový limit pro sledování (communication timeout). Každé volání určuje novou komunikační žádost – to nám umožní komunikaci PVICOM s více manažery na různých PC. Pokud navázání komunikace proběhla v pořádku, komunikační žádost můžeme deinicializovat funkcí PviDeinitialize(). PVICOM rozhraní pracuje na principu klient/server. PVI Manager představuje Server a PVICOM aplikace jako klient. Server a klient jsou na stejném zařízení, pak se jedná o komunikaci místní. Jestliže jsou na různých zařízeních, jedná se o komunikaci vzdálenou. Při místní komunikaci se používá k výměně údajů pomocí sdíleného paměťového prostoru. U vzdálené komunikace se používají sítě na principu TCP/IP. V PVICOM rozhraní umožňuje aplikaci nastavit několik komunikačních instancí (žádostí). Takto se provádí výměna dat mezi několika PVI Managery ve stejnou dobu. Jestliže chceme nastavit PVICOM komunikační žádost, která se volá pomocí funkce PviInitialize(). Typ komunikace je definován uvedením IP (inicializačního parametru), ten se nastavuje v PviInitialize. Jestliže IP není uveden (lze definovat v Automation Studio), pak se jedná o místní komunikaci. Pokud je uveden IP jedná se o komunikaci vzdálenou klient/server přes TCP/IP protokol a inicializační parametr představuje IP adresu, nebo název hostitele serveru (PVI Manager). Viz. kapitola 2.3

3.1.2 Odesílání a přijímání žádostí a odpovědí

Použitím funkcí request (žádost) lze zasílat požadavky do PVI manageru prostřednictvím komunikačních instancí a uživatelskou zprávou. Po zpracování v PVI manageru se uživateli přidělí zpráva, o tom jak byla tato žádost vykonána. O tom jak probíhá zpracování žádostí, pojednává obrázek 9.



Obr. 9 – Zpracování žádostí PVI manageru [2]

3.2 KOMUNIKACE RS232

Popis programu: V hlavičkovém souboru si vytvoříme buffer, do kterého se budou ukládat odeslaná a přijatá data. V inicializační části nejprve inicializujeme požadované sériové rozhraní (př. RS232) nastavíme jeho parametry (paritu, start a stop bit, rychlost). Poté postupně voláme funkce, které slouží pro otevření rozhraní FRM_xopen. Funkcí FRM_read přečteme frontu příchozích zpráv. Pokud obsahuje data, zkopíruje se do proměnné *Data* a smaže zprávu z fronty. Funkční bloky

obsahují i pomocný buffer, do kterého se ukládají data. Pomocí funkce FRM_write se zařadí načtená data do send bufferu. FRM_close slouží pro uzavírání spojení mezi účastníky. Pro realizaci komunikace není třeba využívat všechny funkční bloky. Samotnou komunikaci lze realizovat pomocí. Inicializační funkce, FRM_xopen, FRM_read, FRM_write, FRM_close. Viz. kapitola 2.4.3

Frame Limit (pro příjem dat)

Rámec obsahuje čísla bitů, které jsou vysílány v pořadí, aby bylo možné rozlišovat obdržená data (rámce), musí být specifikována jejich délka.

Rámcový oddělovač – zvláštní charakter odesílání konce rámců, nezáleží na délce rámce. Tímto DVFrame driver zjistí, že byla data přijata, a následovně začíná nový rámec.

Frame Limit (pro odesílání dat)

V Odeslaném rámci je definována funkce blokování volání. Ta volí buffer, který je bez zpoždění. Délka rámce v bufferu musí být menší než maximální délka bufferu.

3.3 KOMUNIKACE ETHERNET

Popis programu: V inicializační části se nejprve volá funkce TCP_open zde se určí, kdo bude komunikovat, jestli server nebo klient a nastaví se číslo portu, se kterým bude klient nebo server spojen. Poté funkcí TCP_send se nastaví adresa bufferu, na kterou se posílají data. Pro přijetí zprávy se používá funkce TCP_recv. Server může “běžet” pořád u klienta se spojení uzavírá pomocí funkce TCP_close. V bufferu se přijímají data souřadnice nebo proměnné, které se odesílají ze serveru do klienta a naopak. Pro zvolení typu komunikace (RS232, Ethernet, atd.) se využívá třída task, která je v případě zvolení komunikace přes Ethernet aktivní nebo neaktivní. Stejně je tomu tak pro komunikaci RS232. Hierarchie stromové struktury a volání funkcí se navzájem liší. viz. Obrázek 6. Kapitola 2.4.4

3.4 KOMUNIKACE INA2000

Tento komunikační protokol je obsažen v každé verzi Automation Studia. Pracuje na principu client/server. Poskytuje funkce pro čtení a zápis dat. Tento protokol umožňuje komunikaci pomocí rozhraní RS232, Ethernet, Ethernet Powerlink, CAN. Pro naši komunikaci využíváme knihovnu AsIMA. Naprogramovaná komunikace je vytvořena v programovacím jazyce Automation Basic. Možnosti mnou naprogramované komunikace s využitím komunikačního protokolu INA2000. Power panely mohou komunikovat přes ethernetové rozhraní, sériovou linku RS232 a sběrnici CAN. CAN sběrnici nebylo možné prakticky ověřit ve škole, jelikož škola nemá na tuto komunikaci požadovanou licenci. Typ použité komunikace se nastavuje v datovém objektu, ten je pro každé použité rozhraní jiný např. pro komunikaci přes ethernet je to datový modul1 -> station1.dat, pro komunikaci přes RS232 je to datová modul2 -> station1. To co obsahuje tento datový modul, je popsáno v kapitole **struktura programu přes INA komunikaci (RS232)** viz. níže.

Nastavení komunikace přes Ethernet Powerlink probíhá tak, že v položce Physical View v sub slot pravým tlačítkem vybereme insert a do prázdného slotu přidáme kartu 3IF7xx podle používané karty. Poté v hardwarové konfiguraci vybereme verzi ethernet powerlinku (ethernet, V1, V2). Komunikaci přes INA2000 s využitím ethernet powerlinku nastavíme v datovém objektu obdobně jako ethernet.

3.4.1 Knihovna AsIMA

- IMAnit – voláním této funkce se vyhodnocují údaje z datového objektu. Nezbytná paměť pro komunikaci se vytváří v dočasném prostoru. Jestliže dojde k chybě, pak nedochází k uvolnění paměti. IMAnit je inicializační funkce, která musí být volána před ostatními funkcemi v knihovně AsIMA.
- IMComm – pomocí této funkce se zřizuje skutečná komunikace mezi klientem a serverem. IMComm se volá v cyklické části programu. Doba cyklu a zařazení do třídy task pro tuto komunikaci > 10ms.

- IMAClose – touto funkcí uzavíráme spojení mezi serverem/klientem a současně dochází k dealokaci paměti. Jestliže chceme opětovně zřídit komunikaci klient/server, musíme volat funkci v inicializační části jako první.

3.5 PRAKTICKÉ ŘEŠENÍ PROPOJENÍ MEZI PLC VARIANTA I.



Obr. 10 – Praktické propojení mezi Power Panely varianta I.

Na obrázku 10 máme tři power panely, které jsou propojeny pomocí ethernetového rozhraní a rozhraní RS232.

Master obsahuje tyto položky v třídě task:

- program hada vizualizaci a program kalibrace obrazovky kvůli zpřesnění ovládání dotykové obrazovky, který se provádí pouze jednou a to v úvodu po znovu nahrání programu do power panelu.

Slave1 obsahuje tyto položky v třídě task:

- programu hada, vizualizaci, kalibraci – stejně jako Master
- datový objekt modul2 s názvem station1, ve kterém jsou uloženy parametry INA komunikace.

"[ROUTING_PATH]" – název oddílu, musí být v závorkách,
"/SECTION=INTERFACE" – definuje parametry pro směřování požadované cesty (které zařízení se má připojit ke kterému).

"/ CN = IF1" směřování k serveru pro komunikaci RS232.

"[PV_LIST_1]" název oddílu, musí být v závorkách

"/SECTION=PV" definuje sekci pro proměnné programu, které mají být odesílány.

"/SV=READ_WRITE /SERVPV=cervi:start /CLNTPV=cervi:start"
zápis pro deklarování proměnné, která má být sdílena mezi masterem a slavem.

- ima_task program tvoří tři funkce a to: **IMAINit** – ta inicializuje INA2000 Manager (vyhodnocuje a v konfiguračním datovém objektu alokuje paměť). **INAcmm** provozuje INA2000 manager jako klient (obsahuje komunikační procesy). **IMAclose** uzavírá INA2000 manager (deinicializace komunikace a uvolňuje přidělenou paměť).

Slave2 obsahuje tyto položky v třídě task:

- programu hada, vizualizaci, kalibraci
- ima_task – stejně jako Slave1
- Datový objekt modul1 s názvem station1 ve kterém jsou uloženy parametry INA komunikace.

"[ROUTING_PATH]" – název oddílu, musí být v závorkách,
"/SECTION=INTERFACE" – definuje parametry pro směrování požadované cesty (které zařízení se má připojit ke kterému).

"/ CN = IF5.1 " směrování k serveru pro komunikaci Ethernet (.1) - je node number, toto číslo je specifické pro každý power panel a je zapsán v hexadecimální soustavě). Node number lze nastavit na každém power panelu většinou ze zadní strany u napájení a slotů pro komunikační rozhraní nastavuje se pomocí dvou 16-ti polohových přepínačů. Pro komunikaci musí být v celé síti jedinečný. Pro zjednodušení sledování hodnoty *node number* je ve vizualizaci (*Visu2*) umístěn numeric, do kterého se vypisuje aktuální *node number* power panelu.

"[PV_LIST_1]" název oddílu, musí být v závorkách

"/SECTION=PV" definuje sekci pro proměnné programu, které mají být odesílány.

"/SV=READ_WRITE /SERVPV=cervi:start /CLNTPV=cervi:start"
zápis pro deklarování proměnné, která má být sdílena mezi masterem a slavem2. Tato proměnná start je používána u tlačítka start a pomocí něj se startuje celá hra.

3.5.1 Struktura programu INA komunikace pro Slave1, Slave2 varianta

I.:

- **ima_task** - **ima_task.var**: obsahuje funkční bloky pro zřízení komunikace. **ima_task.ab**: program pro zřízení komunikace. V inicializační části je obsažena funkce IMAinit. viz (knihovnaAsIMA). Dále pak v cyklické části funkce IMAcomm a IMAClose. viz (knihovnaAsIMA)
- **var_task** - **var_task.var**: zde se definují jména proměnných a jejich datové typy, které jsou využívány při komunikaci, pro příklad score hadů, nastavení proměnných pro ovládání hada (tlačítek) atd. **var_task.ab**: program obsahuje inicializační a cyklickou část. V inicializační se pracuje s názvy proměnných a jejich přidělených nebo nastavených hodnot. Pro příklad ovládání tlačítka v části .var si nadefinujeme proměnnou nastav_tlacitko a její datový typ (INT) v programové části .ab pak definujeme hodnoty pro čtyři směry, kde každý směr má různou hodnotu. Takže nastav_tlacitko :=1, nastav_tlacitko := 2, nastav_tlacitko := 3, nastav_tlacitko := 4.
- **modul2: station1.dat**: jedná se o datový objekt pro komunikaci přes sériovou linku RS232, ve kterém jsou uloženy parametry pro INA

komunikaci. Pro příklad komunikace pro ovládání tlačítek a výpisu score:

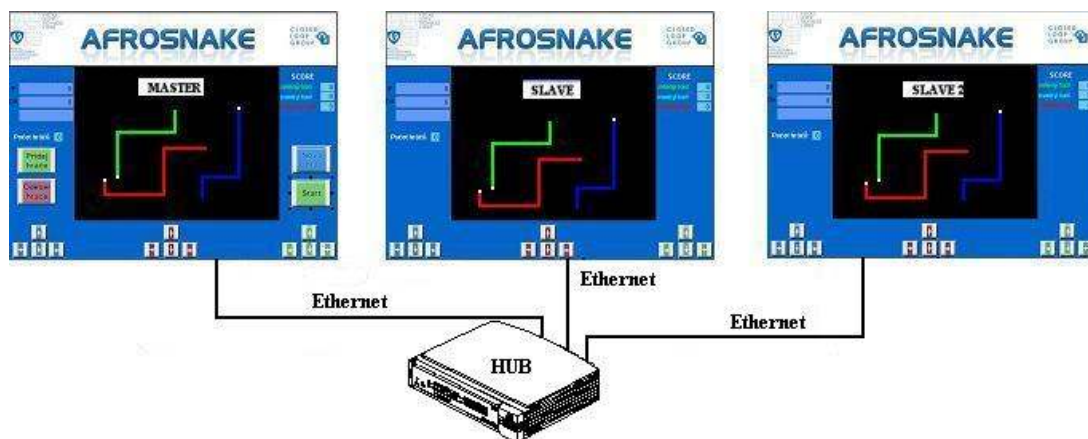
```
" [ROUTING_PATH] "  
"/SECTION=INTERFACE"  
"/CN=IF1" pro komunikaci přes RS232  
;"CN=IF3.4" pro komunikaci přes CAN
```

```
" [PV_List_1] "  
"/SECTION=PV"  
"/SV=READ_WRITE /CLNTPV=var_task:nastav_tlacitko"  
"/SV=READ_WRITE /CLNTPV=var_task: score1"
```

- **modul1: station1.dat:** jedná se o datový objekt pro komunikaci přes ethernet, ve kterém jsou uloženy parametry pro INA komunikaci. Pro příklad komunikace pro ovládání tlačítek a výpisu score:

```
" [ROUTING_PATH] "  
"/SECTION=INTERFACE"  
"CN=IF5.44" pro komunikaci přes Ethernet. Argument (.44) je adresa (Node  
number, DA) PLC v Hexa a musí být v síti jedinečná.  
;"CN=IF7.44" pro komunikaci přes Ethernet Powerlink. Argument (.44) je adresa  
(Node number, DA) PLC v Hexa a musí být v síti jedinečná  
" [PV_List_1] "  
"/SECTION=PV"  
"/SV=READ_WRITE /CLNTPV=var_task:nastav_tlacitko"  
"/SV=READ_WRITE /CLNTPV=var_task: score1"
```

3.1 PRAKTICKÉ ŘEŠENÍ PROPOJENÍ MEZI PLC VARIANTA II.



Obr. 11 – Praktické propojení mezi Power Panely varianta II.

Na obrázku 11 máme tři power panely, které jsou propojeny pomocí ethernetového rozhraní s využitím hubu.

Master

- obsahuje samotný program hada s vizualizací.

Slave1

- program hada s vizualizací
- obsahuje datový objekt modul1 s názvem station1, kde je nastavena adresa zařízení ke kterému se má slave1 připojit "/ CN = IF5.1 "
- ima task viz. kapitola 3.5.1

Slave2

- program hada s vizualizací
- obsahuje datový objekt modul1 s názvem station1, kde je nastavena adresa zařízení ke kterému se má slave2 připojit "/ CN = IF5.1 "
- ima task viz. kapitola 3.5.1

3.1.1 Struktura programu INA komunikace pro Slave1, Slave2 varianta II.:

- **modul1: station1.dat:** Zařízení Slave1 a Slave2 obsahují stejný datový objekt. Modul1 je datový objekt pro komunikaci přes ethernet, ve kterém jsou uloženy parametry pro INA komunikaci. Pro příklad komunikace pro ovládání tlačítek a výpisu score:

" [ROUTING_PATH] "

"/SECTION=INTERFACE"

"CN=IF5.1" pro komunikaci přes Ethernet. Argument (.1) je adresa (Node number, DA) PLC v Hexa a musí být v síti jedinečná.

;"CN=IF7.1" pro komunikaci přes Ethernet Powerlink. Argument (.1) je adresa (Node number, DA) PLC v Hexa a musí být v síti jedinečná

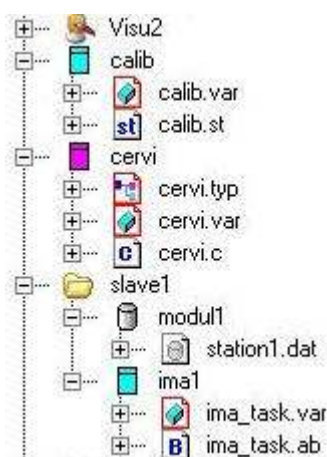
" [PV_List_1] "

"/SECTION=PV"

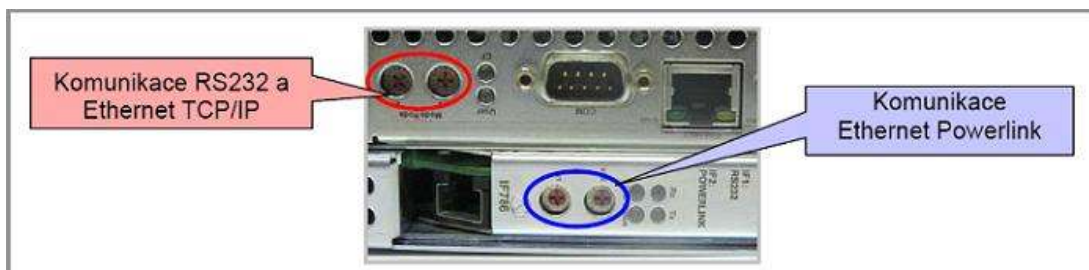
"/SV=READ_WRITE /CLNTPV=var_task:nastav_tlacitko"

"/SV=READ_WRITE /CLNTPV=var_task: score1"

- **ima_task - ima_task.var:** obsahuje funkční bloky pro zřízení komunikace. **ima_task.ab:** program pro zřízení komunikace.



Obr. 12 – Architektura programu pro komunikaci Ethernet



Obr. 13 – Nastavení komunikačních adres PLC

3.2 ZHODNOCENÍ KOMUNIKACÍ

3.2.1 Varianta I.

Tato možnost umožňuje komunikaci mezi MASTEREM a SLAVEM1 prostřednictvím ethernetového rozhraní, mezi MASTEREM a SLAVEM2 je komunikace řešena pomocí sériového rozhraní RS232 viz. Obr.10

Ethernetová komunikace z hlediska implementovatelnosti má výhodu oproti komunikaci vytvořené přes RS232 v možnosti propojení více zařízení s využitím hubu. U rozhraní RS232 jsme schopni propojit pouze dvě stanice. Další výhoda komunikace přes Ethernet je jeho nepoměrně vyšší rychlost než u rozhraní RS232.

Ve vytvořené vzorové aplikaci hry had lze využít libovolné komunikační rozhraní. Vzhledem k tomu, že se sdílejí pouze informace o stisku tlačítek a score, není komunikace náročná na přenosovou rychlost. Problém by mohl nastat v přenášení obrazu mezi PLC, obraz vyžaduje sdílení většího množství informací. Zároveň by byl kladen větší požadavek na přenosovou rychlost. Rozhraní RS232 by pak nebylo schopno komunikace, vznikaly by velké časové odezvy a tím tak nemožnost ovládání hry z důvodu desynchronizace mezi stanicemi. Hráč, který komunikuje po ethernetovém rozhraní, by měl výhodu zejména ve vykreslování hada oproti hráči, který by komunikoval přes rozhraní RS232.

3.2.2 Varianta II.

V tomto řešení je propojení realizováno pomocí ethernetového rozhraní, kde stanice SLAVE1 a SLAVE2 jsou propojeny s MASTEREM pomocí Ethernetu prostřednictvím hubu viz. Obr.11. Nedochozí zde k časovým odezvám a desynchronizaci vlivem přenosové rychlosti použitého média. Každá stanice SLAVE musí mít nastavenou komunikační adresu PLC, ta udává adresu MASTERU ke které se připojí ostatní účastníci komunikace.



Obr. 14 – Stanice MASTER

4. HRA HAD

Hra je vytvořena ve vývojovém prostředí Automation Studio s využitím programovacího jazyka Ansi C a knihovny Visapi. Hra je naprogramována pro tři hráče. V základním módu hrají dva hráči, třetí hráč lze přidat kliknutím na tlačítko přidej hráče.

4.1 CÍL HRY

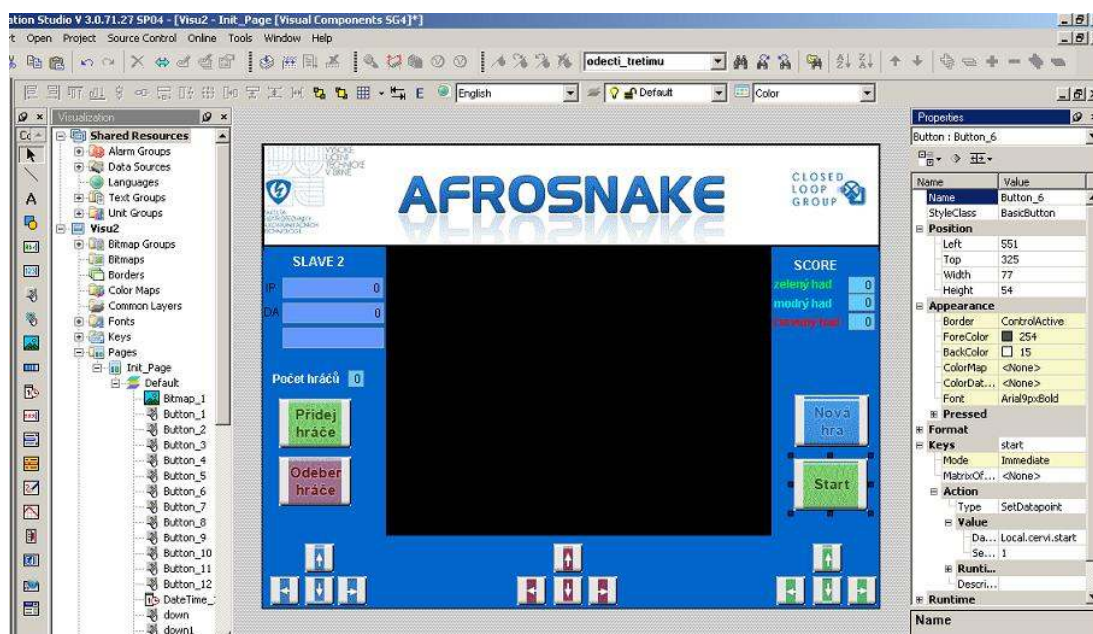
Cílem hry je obsazení co největšího území bez toho aniž by jeden hráč narazil do druhého, třetího hráče nebo do zdi. Každý hráč má tři životy, jestliže narazí, odečítá se mu život a vítězí ten, který má co největší počet životů tedy score.

4.2 ŘEŠENÍ VIZUALIZACE

K naprogramování a propojení programu a vizualizace je nutné využití knihovny visapi. Hrací plocha, kterou představuje DrawBox slouží k zobrazování funkcí obsažené v knihovně visapi na displeji. Ve vizualizaci jsou dále použity tlačítka, pomocí kterých je udáván směr hada. Každý had má přiřazen barvu ovládacích prvků stejných jako je barva samotného hada. Tlačítka jsou pevně spjata s programovou částí hada, obsahují proměnné určitého datového typu v závislosti na potřebách dané aplikace a jsou vázány pomocí virtuálních klíčů k programu. Tlačítkům jsou nastaveny různé parametry a to je mimo jiné: pozice, vlastnosti, akce, typ, bitmapy, text, hodnotu proměnné, atd. Ve hře je nejčastěji definovanou hodnotou virtuální klíč, dále pak action – type v našem případě (SetDatapoint), který po stisknutí tlačítka mění nastavenou hodnotu proměnné na námi definovanou, dále je to value – data zde se nastavuje název deklarované proměnné programu (lokální, globální). Podobné vlastnosti a nastavování parametrů jako je to u tlačítek je i u numeriků. Numeric je použit pro zobrazování a výpis score jednotlivých hráčů. Vizualizace nahraná v každém komunikujícím zařízení obsahuje IP adresu, kterou lze nastavovat pomocí TouchPadu (AlphaPadQVGA), DA (destination adress) udává informaci o komunikačních adresách PLC a pole s aktuálním časem a datem.

4.3 OVLÁDÁNÍ HRY

Hru ovládáme nejprve nastavením počtu hráčů (dva nebo tři) pomocí tlačítka přidej hráče, odeber hráče. Po pomocí tlačítka start odstartujeme hru a ovládáme hady pomocí vytvořených kurzorových šipek uzpůsobených pro ovládání na dotykové obrazovce. Dojde-li ke ztrátě všech tří života jednotlivých hadů, hra končí. K opětovnému spuštění hry slouží tlačítko nová hra.



Obr. 15 – Vývojové prostředí pro vizualizaci

4.4 PROGRAMOVÉ ŘEŠENÍ HRY HADA

Hra je naprogramována pomocí programovacího jazyka Ansi C s využitím knihovny Visapi. V hlavičkové části programu se definuje počet řádků (*RADKY*) a sloupců (*SLOUPCE*) 60 x 80, startovní souřadnice *SX,SY* pro hady 1,2,3. Dále se definují životy (*ZIVOTY*) pro každého hada tři, počáteční směr (*SMER*) hadů, *VELIKOST_CTVERECKU* = 5. Výpis do vizualizace probíhá pomocí pixelu velikost políčka, po kterém postupuje had je 5x5 pixelů. Hrací plocha je tvořena pomocí pole `int hrste[RADKY][SLOUPCE];` Vyhodnocování score hadů závisí na tom, kdo narazil do druhého hráče nebo do zdi, každý had a zeď (hranice hracího pole) mají

přiřazenou hodnotu, která se ukládá do proměnných *int ZNACKA1 = 2; int ZNACKA2 = 3; int ZNACKA3 = 4; int ZNACKA_ZED = 1;*

4.4.1 Popis funkcí programu hry had

```
0060 void hriste_init(void);
0061 void polozctverecek(int x, int y, int i);
0062 void vykresli_hady(void);
0063 void pohybuj_hady(void);
0064 void delay(void);
0065 void ramecek(void);
0066 void odedti_druhemu(void);
0067 void odedti_tretimu(void);
0068 void odedti_zivot(int i, int znacka, char *jmeno);
0069 void prohra(void);
0070
```

Obrázek 16 - Funkce programu pro hru hada

V **inicializační části** programu je volána nová hra, hadům jsou přiřazeny životy a score.

V **cyklické části** se nejdříve nastaví nová hra, jestliže je stisknuto tlačítko nová hra je splněná podmínka (*new_hra == 1*), pak se nastaví každému hadu jeho startovní souřadnice (*SX,SY*), značka (*ZNACKA*), směr (*SMER*) a nakonec barva hada. Po nastavení parametrů je volána funkce *hriste_init(); ramecek()*;

Stisknutím tlačítka start tím splnění podmínku, která zavolá funkce *vykresli_hady(); a pohybuj_hady()*;

Funkce ramecek: v této funkci dochází k nastavení pracovní plochy vizualizace, jestliže je splněna podmínka *ready* dojde k výpisu a propojení programu vizualizace pomocí *Status_Attach*, kde je nastavena cesta do *DrawBoxu* kam se vykreslují hadi a rámeček s hrací plochou. Rámeček je tvořený pomocí čtyř obdélníků o tloušťce čtyř pixelů, které jsou k sobě “poskládány”, tvoří tak okrajovou hranici *DrawBoxu*, zeď (překážku) celého hřiště.

Funkce *hriste_init*: v paměti vytváří hřiště, reprezentovaného dvourozměrným polem. Uvnitř hřiště inicializuje hodnotu jednotlivých políček na “0” a jinak na hodnotu makra *ZNACKA_ZED* (“1”).

Funkce *polozctverecek*: explicitně vloží do vizualizace do DrawBoxu pozici všech hráčů. Na inicializační souřadnice vloží čtverečky o předem určené barvě o velikosti 5x5 pixelů.

Funkce *vykresli_hady*: hadi se vykreslí do DrawBoxu pomocí této funkce. Na začátku se vykreslí dva hadi. Při splnění podmínky (*pridejhrace* == 1) kliknutím ve vizualizaci na tlačítko přidej hráče, dojde k vykreslení hada třetího.

Funkce *pohybuj_hady*: umožňuje nastavení skutečného směru hada. Pomocí proměnných *nastav_tlacitko*, do které se ukládají hodnoty ze čtyř směrů řešeny pomocí přepínače switch. Podle nastaveného směru se nastaví nové souřadnice hada a zakreslí se do vizualizace pomocí funkce *polozctverecek* a rovněž do dvourozměrného pole reprezentujícího hřiště.

Funkce *odecti_druhemu*: zajišťuje pravidla odečítání životů ostatních hadů. Pokud některý had narazí do druhého přímo, dojde k odečtení života.

Funkce *odecti_tretimu*: zajišťuje pravidla odečítání životů ostatních hadů. Pokud některý had narazí do třetího přímo, dojde k odečtení života.

Funkce *odecti_zivot*: sníží počet životů o “-1” tomu hadovi, který poruší známá pravidla a to: narazí do zdi, do jiného hráče nebo sám do sebe.

Funkce *prohra*: vyhodnocuje počet ztracených životů a tím score jednotlivých hráčů.

Zdrojový kód hry hada je popsán na 700 řádcích, zároveň s programem pro komunikaci je přiložen na nosiči přiloženému v bakalářské práci.

5. ZÁVĚR

Před vlastním návrhem vizualizace, programové části hry had a řešení komunikace jsem se musel podrobně seznámit s vývojovým prostředím Automation Studio. Zdrojem informací mi byla především B&R dokumentace a help Automation Studia.

Nejdříve jsem naprogramoval hru had společně se základy vizualizace, kde jsem funkčnost programu ověřil nejdříve na simulátoru poté na power panelu, ve druhé fázi jsem zprovoznil komunikaci mezi více power panely. V průběhu řešení komunikace mezi power panely se naskytlo hned několik variant řešení.

Jako nejvhodnější jsem zvolil komunikaci s využitím komunikačního protokolu INA 2000, která umožňuje komunikaci po RS232, Ethernet, Ethernet Powerlink, CAN.

Vzhledem k tomu, že se sdílí pouze informace o stisku tlačítek a score, není komunikace náročná na přenosovou rychlost, z tohoto důvodu je možné použití libovolného rozhraní.

Po odladění všech součástí programu a komunikace jsem upravil vizualizaci do finální podoby. Konečné řešení splňuje všechny body zadání.

Bakalářská práce by mohla být dále rozšířena o přenos obrazu mezi Masterem a Slavem prostřednictvím komunikací Ethernet Powerlink, CAN a pro komunikaci mezi více PLC a PC přes PVI manager.

6. LITERATURA

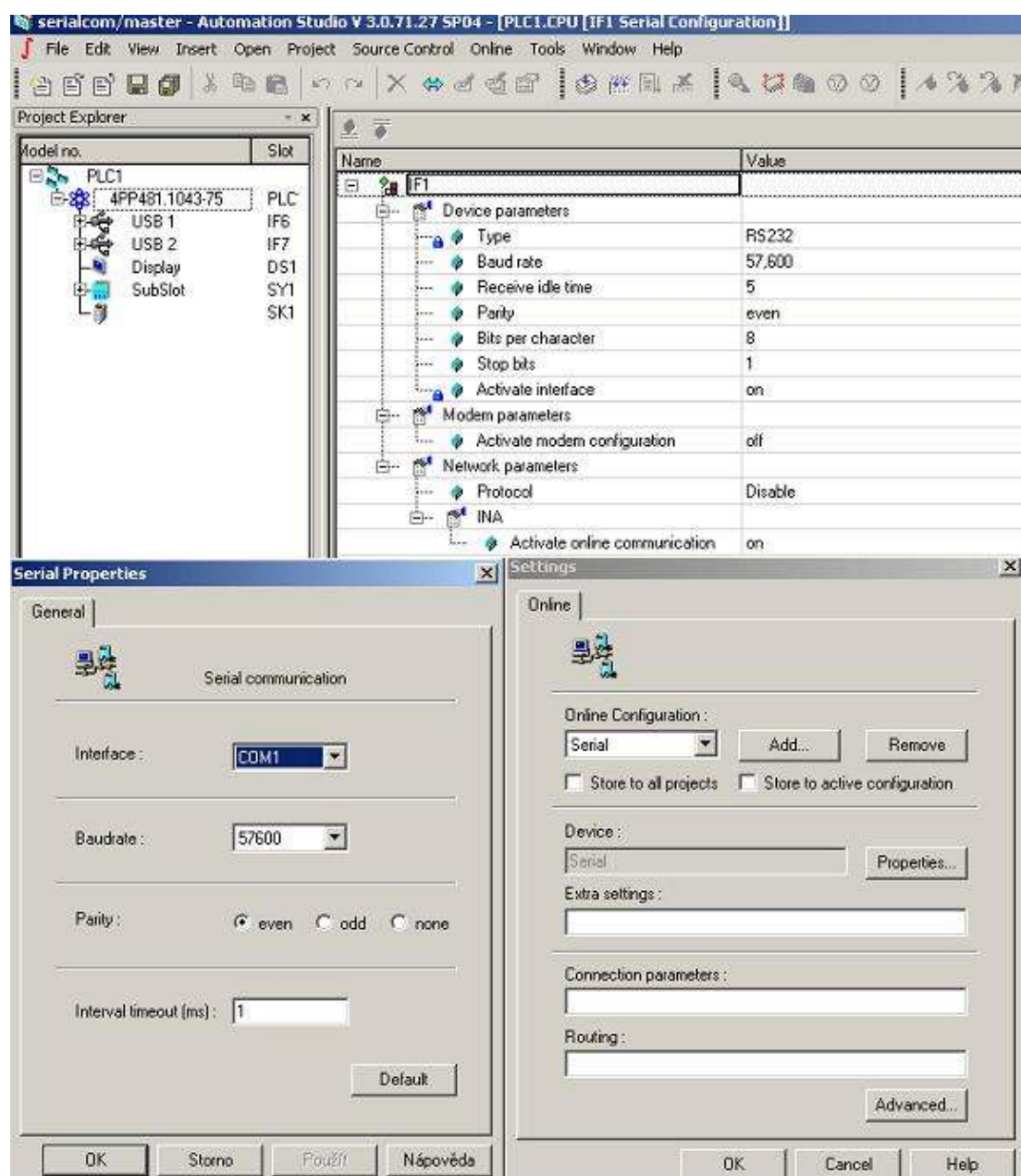
- [1] B&R Automation [online], 2009 <http://www.br-automation.com/>
- [2] Automation Net / PVI Online Help
- [3] Zezulka F.: Prostředky průmyslové automatizace, Brno 2009
- [4] Pásek J.: Programovatelné automaty v řízení technologických procesů Brno 2007
- [5] Technická literatura rozhraní RS232 <http://hw.cz/rs-232>

- [6] Ethernet Powerlink Standardization Group [online], 2005
<http://www.ethernet-powerlink.com/>
- [7] Datasheet použitých power panelů,
http://www.br-automation.com/cps/rde/xchg/br-productcatalogue/hs.xsl/products_90942_ENG_HTML.htm
- [8] Ethernet documentation
<http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Ethernet.html>
- [9] Help Automation Studio
- [10] Automation StudioTM System Introduction. B&R Automation,
- [11] Automation StudioTM Programming. B&R Automation,
- [12] Visual Components Introduction. B&R Automation,

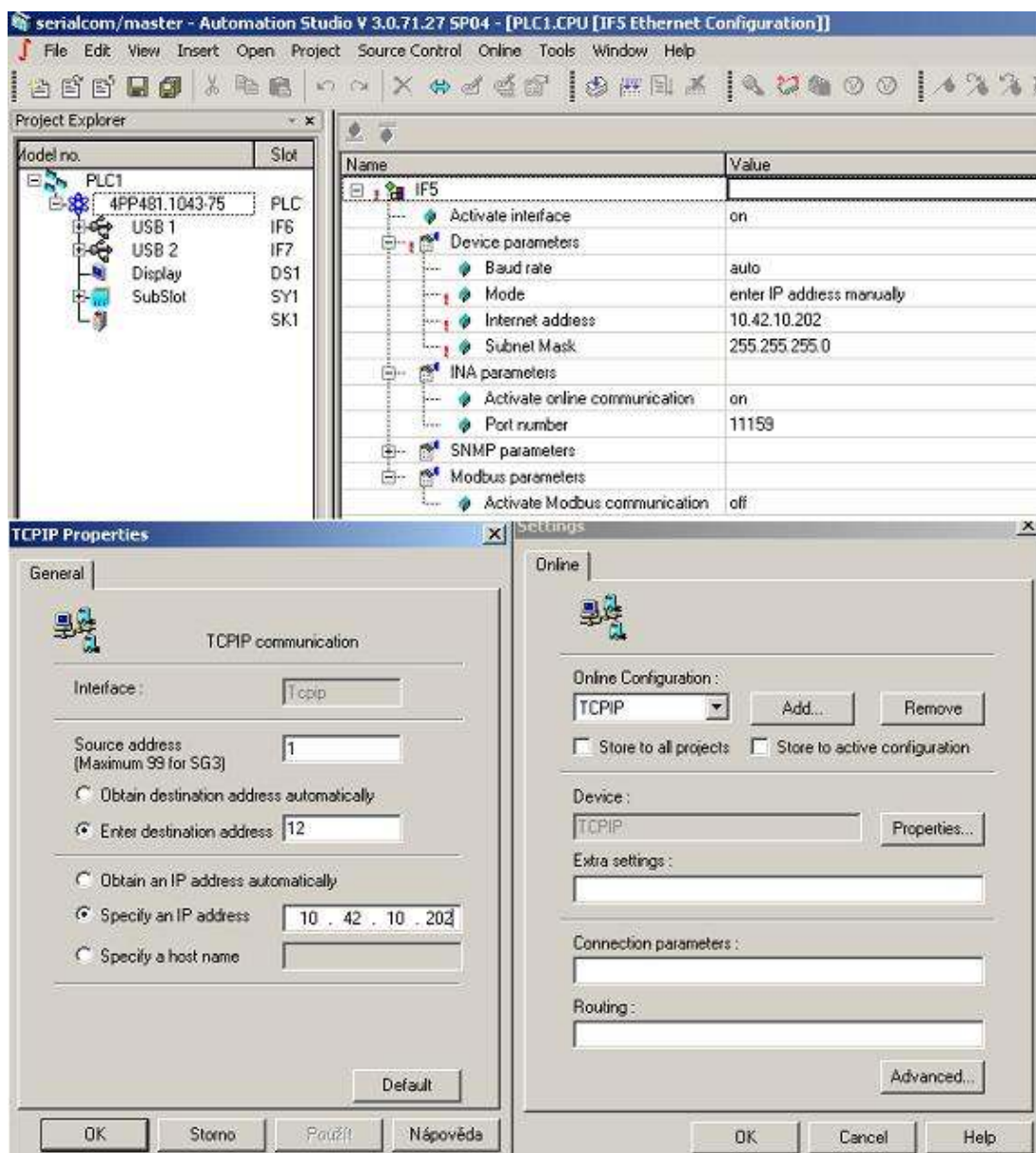
7. SEZNAM PŘÍLOH

- Příloha 1 konfigurace sériového rozhraní RS232
- Příloha 2 konfigurace Ethernetového rozhraní
- Příloha 3 nastavení třídy task pro Ethernet (SLAVE1)
- Příloha 4 nastavení třídy task pro RS232 (SLAVE2)
- Příloha 5 nastavení třídy task pro Master
- Příloha 6 proměnné pro Ethernetovou komunikaci ima.task.var
- Příloha 7 program Ethernetové komunikace ima.taks.ab
- Příloha 8 program komunikace RS232 ima.taks.ab
- Příloha 9 datový objekt modul1 pro ethernetovou komunikaci
- Příloha 10 datový objekt modul2 pro rozhraní RS232
- Příloha 11 proměnné pro kalibraci PLC calib.var
- Příloha 12 program kalibrace PLC calib.st
- Příloha 13 proměnné pro hru had cervi.var
- Příloha 14 propojení Power Panelu v laboratoři

Příloha 1 – konfigurace sériového rozhraní RS232



Příloha 2 – konfigurace Ethernetového rozhraní



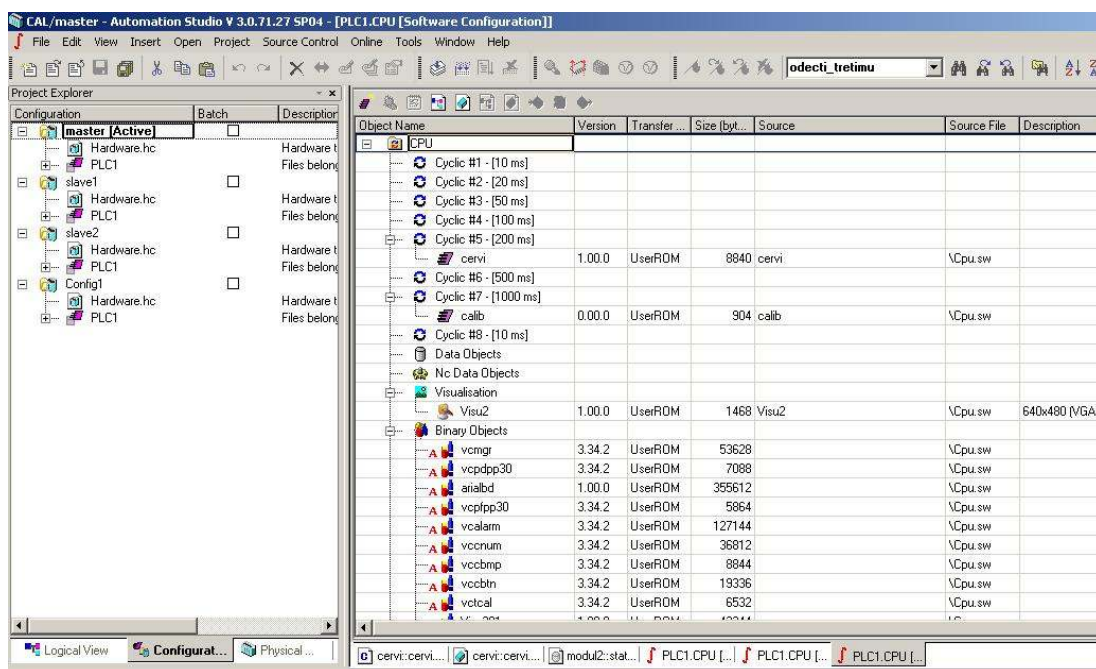
Příloha 3 – nastavení třídy task pro Ethernet (SLAVE1)

Object Name	Version	Transfer ...	Size (byt...)	Source	Source File	Description
Cyclic #1 - [10 ms]						
Cyclic #2 - [20 ms]						
ima1	1.00.0	UserROM	1396	slave1.ima1	\Cpu.sw	
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
Cyclic #5 - [200 ms]						
cervi	1.00.0	UserROM	8840	cervi	\Cpu.sw	
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
calb	0.00.0	UserROM	904	calb	\Cpu.sw	
Cyclic #8 - [10 ms]						
Data Objects						
modul1	1.00.0	UserROM	560	slave1.modul1	\Cpu.sw	
Nc Data Objects						
Visualisation						
Visu2	1.00.0	UserROM	1428	Visu2	\Cpu.sw	640x480 (VGA)
Binary Objects						
vcntrtf	3.34.2	UserROM	273684		\Cpu.sw	
vcctext	3.34.2	UserROM	8600		\Cpu.sw	
vcvbtn	3.34.2	UserROM	19336		\Cpu.sw	
anial	1.00.0	UserROM	370612		\Cpu.sw	
vcctcal	3.34.2	UserROM	6532		\Cpu.sw	
Visu202	1.00.0	UserROM	2448		\Cpu.sw	
Visu203	1.00.0	UserROM	1673544		\Cpu.sw	
vcvnum	3.34.2	UserROM	36812		\Cpu.sw	

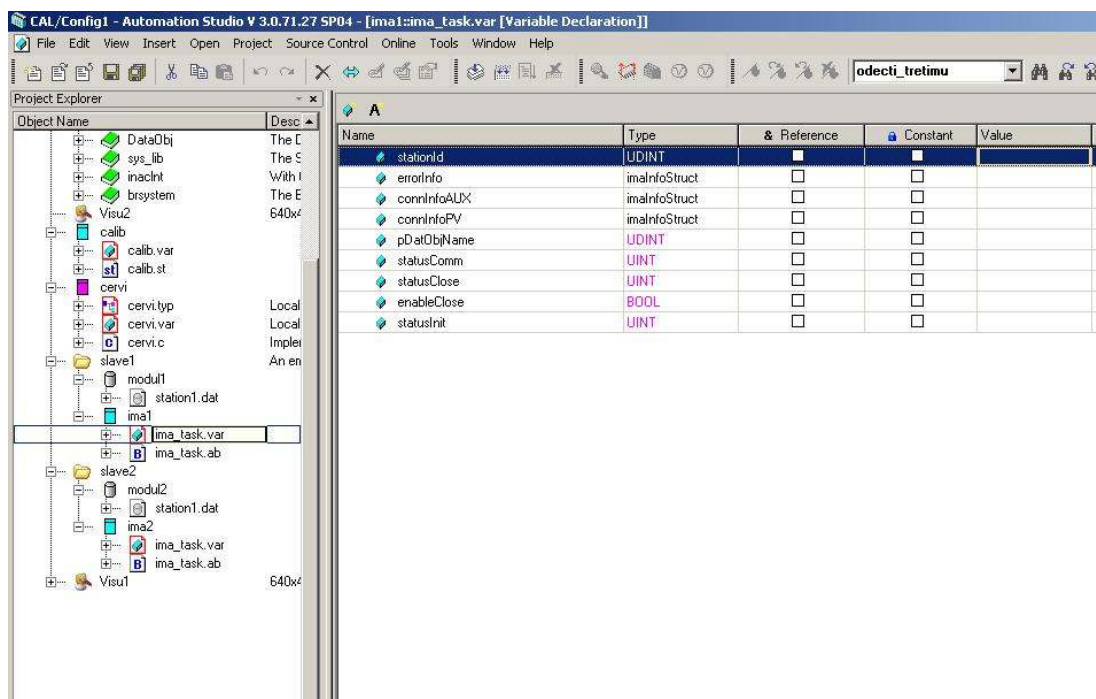
Příloha 4 – nastavení třídy task pro RS232 (SLAVE2)

Object Name	Version	Transfer ...	Size (byt...)	Source	Source File	Description
Cyclic #1 - [10 ms]						
Cyclic #2 - [20 ms]						
ima2	1.00.0	UserROM	1396	slave2.ima2	\Cpu.sw	
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
Cyclic #5 - [200 ms]						
cervi	1.00.0	UserROM	8840	cervi	\Cpu.sw	
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
calb	0.00.0	UserROM	904	calb	\Cpu.sw	
Cyclic #8 - [10 ms]						
Data Objects						
modul2	1.00.0	UserROM	560	slave2.modul2	\Cpu.sw	
Nc Data Objects						
Visualisation						
Visu2	1.00.0	UserROM	1448	Visu2	\Cpu.sw	640x480 (VGA)
Binary Objects						
vcvbk	3.34.2	UserROM	6092		\Cpu.sw	
vcvdp30	3.34.2	UserROM	7088		\Cpu.sw	
vcntrtf	3.34.2	UserROM	273684		\Cpu.sw	
vcvtext	3.34.2	UserROM	8600		\Cpu.sw	
anial	1.00.0	UserROM	370612		\Cpu.sw	
vcctcal	3.34.2	UserROM	6532		\Cpu.sw	
vcvbtn	3.34.2	UserROM	19336		\Cpu.sw	
Visu203	1.00.0	UserROM	1673544		\Cpu.sw	

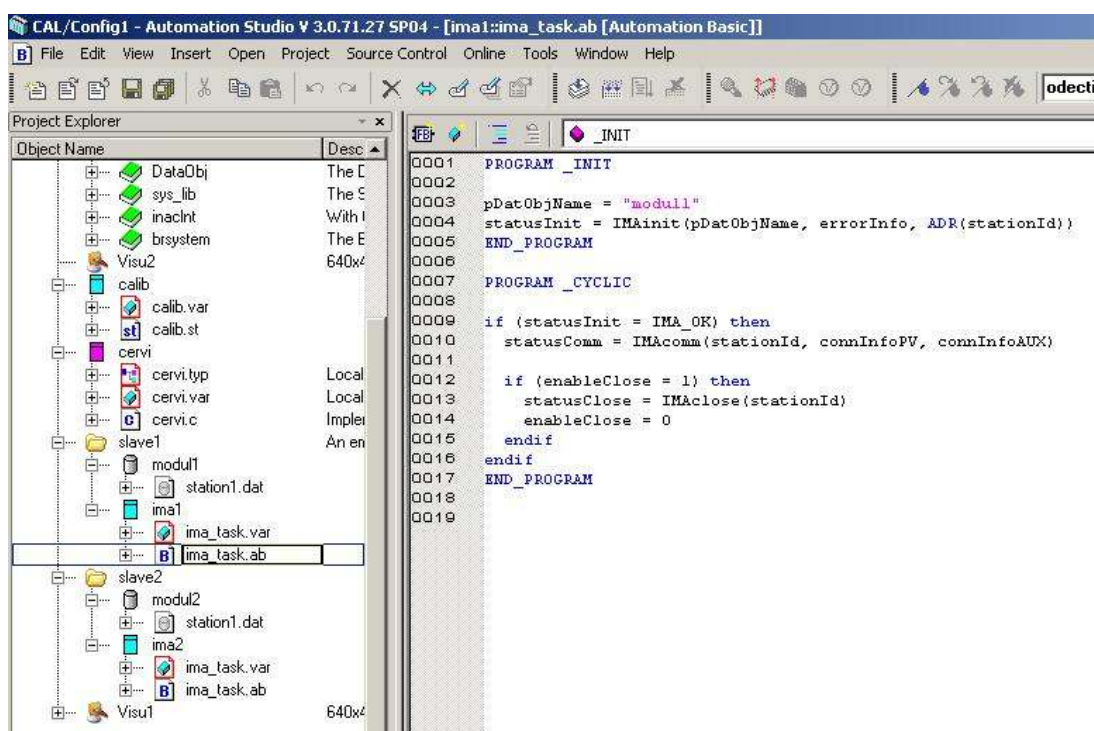
Příloha 5 – nastavení třídy task pro Master



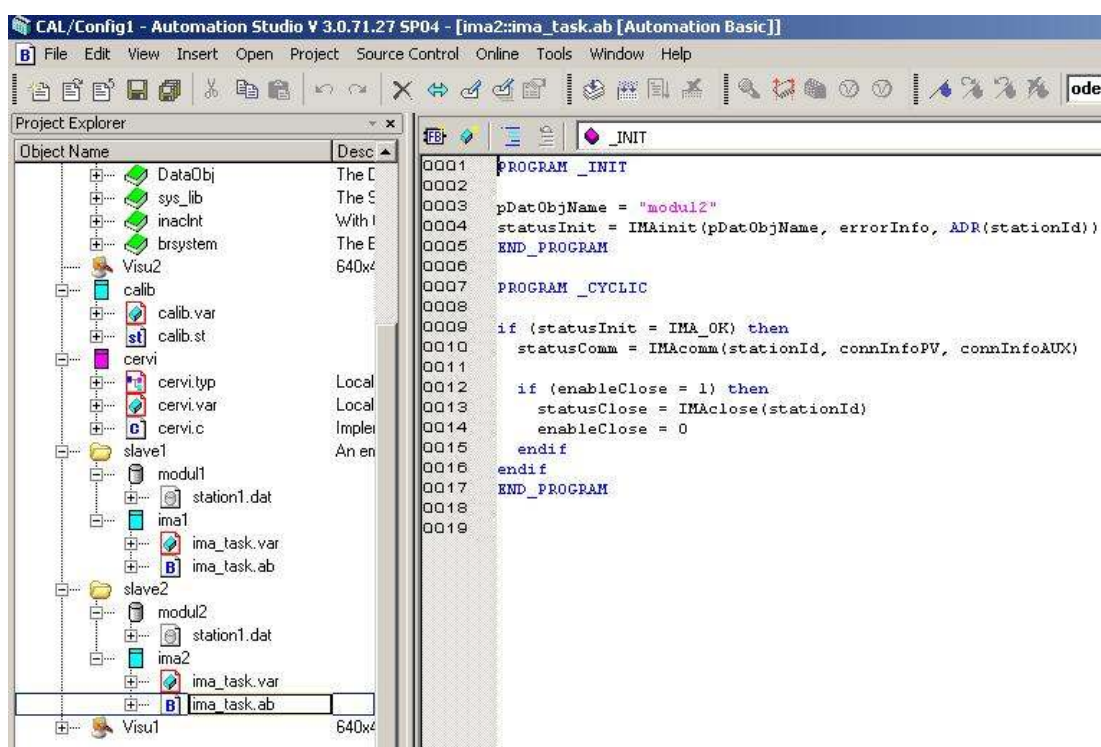
Příloha 6 - proměnné pro Ethernetovou komunikaci ima.task.var



Příloha 7 - program Ethernetové komunikace ima.taks.ab



Příloha 8 - program komunikace RS232 ima.taks.ab



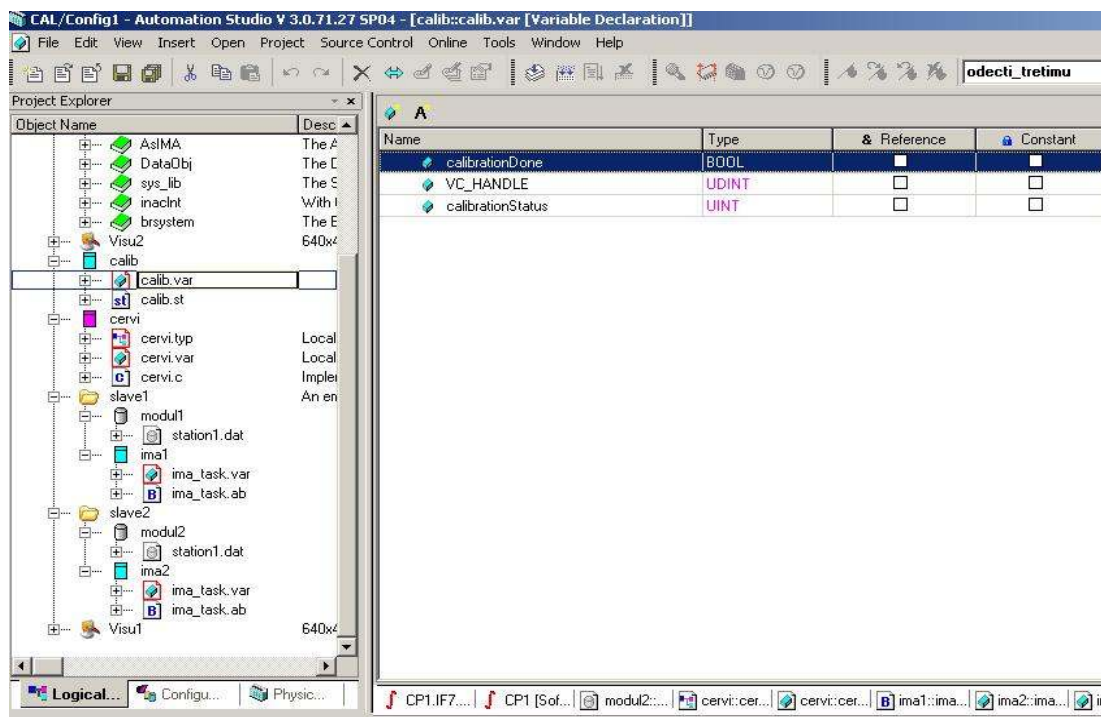
Příloha 9 - datový objekt modul1 pro ethernetovou komunikaci

```
0001 ;-----
0002 ; IMA configuration data object for STATION1
0003 ;-----
0004
0005 "{ROUTING_PATH}" ;section name must be in brackets and is free to choose
0006 "/SECTION=INTERFACE" ;defines a section for routing path definition;
0007 ;must follow the section name
0008 ; routing path definition:
0009 ;
0010 ; /CN name of the routing path stands in data object itself;
0011 ;
0012 ; /CNPV name of the routing path is stored in a STRING-PV;
0013 ; if a dynamic PV is used, than * must stand at the beginning of the name,
0014 ;
0015 "/CN=IFS.1" ;routing-path to server (from IF3 of IPC2000 via CAN to the node #4)
0016 ;ATTENTION: node number must be written in HEXADECIMAL !
0017 ;"/CN=IF1"
0018
0019 "{PV_LIST_1}" ;section name must be in brackets and is free to choose
0020 "/SECTION=PV" ;defines a section for process variable (PV) definition;
0021 ;must follow the section name
0022
0023 ; optional definitions which are common for the whole section:
0024 ;
0025 ; /CONPV condition-PV (UINT) (default: no condition, PV-List will be processed without condition),
0026 ;
0027 ; /CONVAL value of condition-PV, if the PV-List is to be processed at the moment,
0028 ; (default: 0),
0029 ;
0030 ; /MINT minimum time period for the processing of the PV-List [ms] (UINT) (default: 0)
0031 ; if 0 : as often as possible,
0032 ; if <> 0 : PV-List will be processed after time elapses the defined period,
0033 ;
0034 ; /MAXT maximum time period for the processing of the PV-List [ms] (UINT) (default: 0)
0035 ; used only for WRITE service,
0036 ; if 0 : value of the client-PV will be sent to server, only if the value has changed,
0037 ; if <> 0 : value of the client-PV will be sent to server in every case after the defined time period;
0038 ; if the value has changed and the defined time has not elapsed yet, than the
0039 ; value will be sent immediately after that,
0040 ;
0041 ;"/CONPV=var_task:condition_1 /CONVAL=1 /MINT=0 /MAXT=0 "
0042
0043 ; process variable definition:
0044 ;
0045 ; /SV defines a PV-service; must be the first parameter in the line;
0046 ; READ: value of server-PV will be copied to the client-PV,
0047 ; WRITE: value of client-PV will be copied to the server-PV,
0048 ; READ_WRITE: functions as READ, but if the value of PV at the client has changed,
0049 ; than WRITE will be done,
0050 ;
0051 ; /CLNTPV name of the client-PV (must be provided);
0052 ; if a dynamic PV is used, than * must stand at the beginning of the name,
0053 ;
0054 ; /SERVPV name of the server-PV (default: name of the client-PV);
0055 ; if a dynamic PV is used, than * must stand at the beginning of the name,
0056 ;
0057 ; /EVPV name of the counter-PV (UINT); only used for READ and READ_WRITE service;
0058 ; if the value of the server-PV has changed, than the counter-PV will be incremented;
0059 ; (default: no counter-PV);
0060 ; dynamic PV may not be used;
0061 ;"/SV=READ_WRITE /CLNTPV=PV_BYTE /EVPV=var_task:byte_link_cnt"
0062
0063 "/SV=READ_WRITE /SERVPV=cervi:gg /CLNTPV=cervi:gg"
0064 "/SV=READ_WRITE /SERVPV=cervi:nastav_tlacitko /CLNTPV=cervi:nastav_tlacitko"
0065 "/SV=READ_WRITE /SERVPV=cervi:nastav_tlacitko2 /CLNTPV=cervi:nastav_tlacitko2"
0066 "/SV=READ_WRITE /SERVPV=cervi:score1 /CLNTPV=cervi:score1"
0067 "/SV=READ_WRITE /SERVPV=cervi:score2 /CLNTPV=cervi:score2"
0068 "/SV=READ_WRITE /SERVPV=cervi:score3 /CLNTPV=cervi:score3"
0069
```

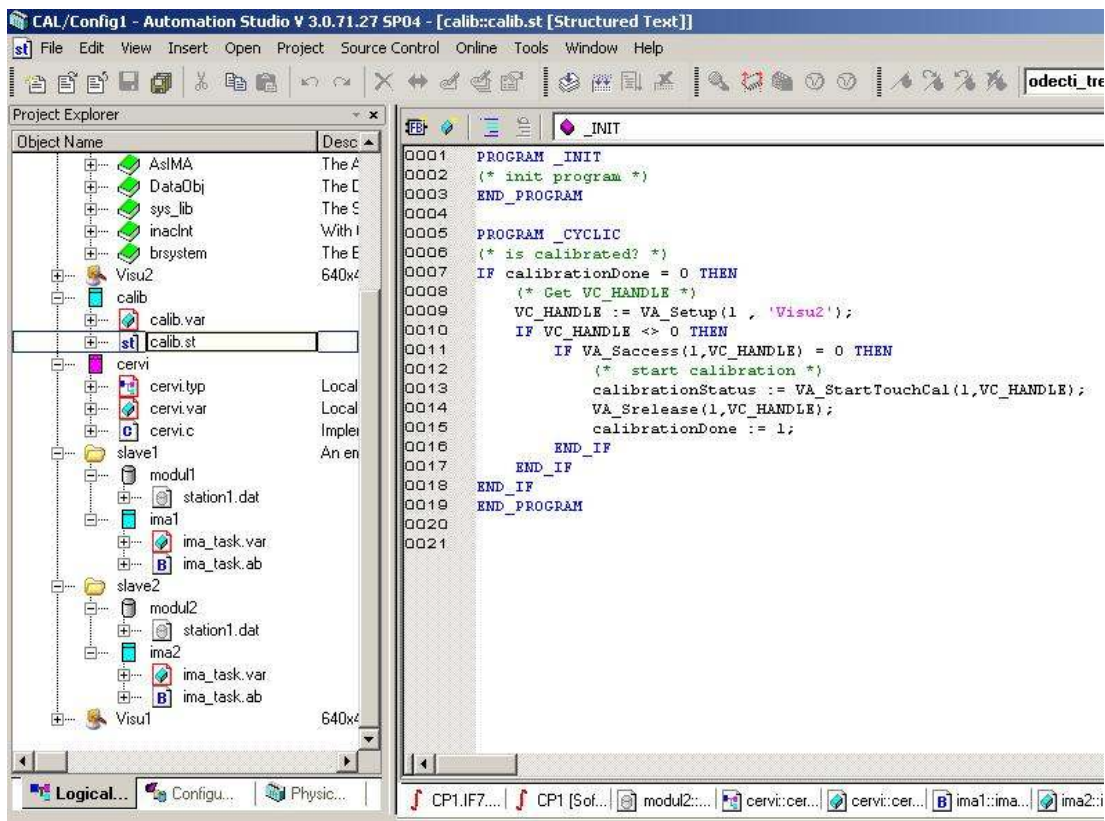
Příloha10 - datový objekt modul2 pro komunikaci RS232

```
0001 ;-----
0002 ; IMA configuration data object for STATION1
0003 ;-----
0004
0005 "[ROUTING_PATH]" ;section name must be in brackets and is free to choose
0006 "/SECTION=INTERFACE" ;defines a section for routing path definition;
0007 ;must follow the section name
0008 ; routing path definition:
0009 ;
0010 ; /CN name of the routing path stands in data object itself;
0011 ;
0012 ; /CNPV name of the routing path is stored in a STRING-PV;
0013 ; if a dynamic PV is used, than * must stand at the beginning of the name,
0014 ; /CN=IFS.C" ;routing-path to server (from IF3 of IPC2000 via CAN to the node #4)
0015 ;ATTENTION: node number must be written in HEXADECIMAL !
0016
0017 "/CN=IF1"
0018
0019 "[PV_LIST_1]" ;section name must be in brackets and is free to choose
0020 "/SECTION=PV" ;defines a section for process variable (PV) definition;
0021 ;must follow the section name
0022
0023 ; optional definitions which are common for the whole section:
0024 ;
0025 ; /CONPV condition-PV (UINT) (default: no condition, PV-List will be processed without condition),
0026 ;
0027 ; /CONVAL value of condition-PV, if the PV-List is to be processed at the moment,
0028 ; (default: 0),
0029 ;
0030 ; /MINT minimum time period for the processing of the PV-List [ms] (UINT) (default: 0)
0031 ; if 0 : as often as possible,
0032 ; if <> 0 : PV-List will be processed after time elapses the defined period,
0033 ;
0034 ; /MAXT maximum time period for the processing of the PV-List [ms] (UINT) (default: 0)
0035 ; used only for WRITE service,
0036 ; if 0 : value of the client-PV will be sent to server, only if the value has changed,
0037 ; if <> 0 : value of the client-PV will be sent to server in every case after the defined time period;
0038 ; if the value has changed and the defined time has not elapsed yet, than the
0039 ; value will be sent immediately after that,
0040 ;
0041 ; "/CONPV=var_task:condition_1 /CONVAL=1 /MINT=0 /MAXT=0 "
0042
0043 ; process variable definition:
0044 ;
0045 ; /SV defines a PV-service; must be the first parameter in the line;
0046 ; READ: value of server-PV will be copied to the client-PV,
0047 ; WRITE: value of client-PV will be copied to the server-PV,
0048 ; READ_WRITE: functions as READ, but if the value of PV at the client has changed,
0049 ; than WRITE will be done,
0050 ;
0051 ; /CLNTPV name of the client-PV (must be provided);
0052 ; if a dynamic PV is used, than * must stand at the beginning of the name,
0053 ;
0054 ; /SERVPV name of the server-PV (default: name of the client-PV);
0055 ; if a dynamic PV is used, than * must stand at the beginning of the name,
0056 ;
0057 ; /EVPV name of the counter-PV (UINT); only used for READ and READ_WRITE service;
0058 ; if the value of the server-PV has changed, than the counter-PV will be incremented;
0059 ; (default: no counter-PV);
0060 ; dynamic PV may not be used;
0061 ; "/SV=READ_WRITE /CLNTPV=PV_BYTE /EVPV=var_task:byte_link_cnt"
0062
0063 "/SV=READ_WRITE /SERVPV=cervi:gg /CLNTPV=cervi:gg"
0064 "/SV=READ_WRITE /SERVPV=cervi:nastav_tlacitko /CLNTPV=cervi:nastav_tlacitko"
0065 "/SV=READ_WRITE /SERVPV=cervi:nastav_tlacitko2 /CLNTPV=cervi:nastav_tlacitko2"
0066 "/SV=READ_WRITE /SERVPV=cervi:score1 /CLNTPV=cervi:score1"
0067 "/SV=READ_WRITE /SERVPV=cervi:score2 /CLNTPV=cervi:score2"
0068 "/SV=READ_WRITE /SERVPV=cervi:score3 /CLNTPV=cervi:score3"
0069
```

Příloha11 - proměnné pro kalibraci PLC calib.var



Příloha12 - program kalibrace PLC calib.st



Příloha13 - proměnné pro hru had cervi.var

CAL/Config1 - Automation Studio V 3.0.71.27 SP04 - [cervi:cervi.var [Variable Declaration]]

File Edit View Insert Open Project Source Control Online Tools Window Help

Project Explorer

Object Name Desc

- AsiMA The A
- DataObj The C
- sys_lib The S
- inacInt With I
- brsystem The E
- Visu2 640x4
- calib
- calib.var
- calib.st
- cervi
- cervi.typ Local
- cervi.var Local
- cervi.c Imple
- slave1 An en
- modul1
- station1.dat
- ima1
- ima_task.var
- ima_task.ab
- slave2
- modul2
- station1.dat
- ima2
- ima_task.var
- ima_task.ab
- Visu1 640x4

Logical... Configu... Physic...

CP1.IF7... CP1 [Sof... modul2:... cervi:cer... cervi:cer... ima1::ima... ima2::ima... ima1::ima... ima2::

Name	Type	& Reference	Constant	Value
nastav_tlacitko2	USINT	☐	☐	0
score2	INT	☐	☐	0
score1	INT	☐	☐	0
start	UINT	☐	☐	0
new_hra	INT	☐	☐	1
VC_HANDLE	UDINT	☐	☐	0
ready	UINT	☐	☐	0
firstcycle	USINT	☐	☐	1
Status_Attach	UINT	☐	☐	0
fill	UINT	☐	☐	0
x1	UINT	☐	☐	0
x2	UINT	☐	☐	0
color	UINT	☐	☐	0
y1	UINT	☐	☐	0
y2	UINT	☐	☐	0
score3	USINT	☐	☐	0
gg	INT	☐	☐	0
nastav_tlacitko	USINT	☐	☐	0
odebertrace	USINT	☐	☐	2

Příloha14 - propojení Power Panelu v laboratoři

