

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a**  
**biomedicínského inženýrství**

**Softwarové PLC pro operační systém QNX 6**  
**RTOS, komunikace a řízení pomocí**  
**průmyslové sítě PROFINET**

Software PLC for Operating System QNX 6  
RTOS, Communication and Control using  
Industrial Network PROFINET

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

## Zadání bakalářské práce

Student: **Jakub Sikora**  
Studijní program: B2649 Elektrotechnika  
Studijní obor: 2601R004 Měřicí a řídicí technika  
Téma: **Softwarové PLC pro operační systém QNX 6 RTOS, komunikace a řízení pomocí průmyslové sítě PROFINET**  
**Software PLC for Operating System QNX 6 RTOS, Communication and Control using Industrial Network PROFINET**

### Zásady pro vypracování:

V rámci této bakalářské práce se bude řešit implementace softwarového PLC a jeho napojení na průmyslovou komunikační síť PROFINET v rámci operačního systému QNX6 RTOS. Součástí práce bude vytvoření testovací aplikace pro softwarové PLC, která bude komunikovat s distribuovanou periferií ET200 MP IM155-5 PN ST pomocí průmyslové sítě PROFINET a komunikační karty SIEMENS CP 1616 (PC Station).

### Body zadání:

1. Konfigurace hardwarových prvků QNX6 PC Station a průmyslové sítě PROFINET.
2. Nastavení komunikace QNX6 PC Station (Siemens CP1616) s distribuovanou periferií E200MP IM155-5 PN ST a jejími I/O digitálními a analogovými moduly.
3. Vytvoření softwarového PLC pro operační systém QNX6 RTOS.
4. Napojení softwarového PLC na průmyslovou komunikační síť PROFINET.
5. Vytvoření vývojového nástroje a testovací aplikace pro softwarové PLC pomocí normy IEC 61131.
6. Vytvoření síťové vizualizace softwarového PLC a distribuované periferie ET200MP IM155-5 PN ST s I/O digitálními a analogovými moduly.
7. Zhodnocení dosažených výsledků práce.

### Seznam doporučené odborné literatury:

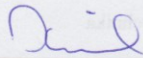
- [1] POPP, Manfred. *Industrial communication with PROFINET*. Karlsruhe (DE): PROFIBUS.  
[2] KRTEM, Rob. *The QNX cookbook : recipes for programmers*  
. Canata (CA): Parse Software Devices, 2003. ISBN 0-9682501-2-2.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

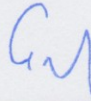
Vedoucí bakalářské práce: **Ing. Vilém Srovnal, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

**Poděkování:**

Děkuji vedoucímu bakalářské práce Ing. Vilémovi Srovnalovi, Ph.D. za odbornou pomoc a užitečné rady při zpracování bakalářské práce.

**Prohlášení:**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 6.5.2014

Podpis: 

**Prohlášení zástupce firmy Disam RT s.r.o.:**

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě dne: 6.5.2014

Podpis: 

Ing. Antonín Kutálek  
Jednatel společnosti  
Disam RT s.r.o.

## **Abstrakt**

V této bakalářské práci jsem se zabýval implementací softwarového PLC a jeho napojení na průmyslovou komunikační síť PROFINET. Implementované softwarové PLC bylo navrženo pro operační systém reálného času QNX6 RTOS. V rámci komunikace mezi jednotlivými hardwarovými moduly byl použit komunikační protokol PROFINET RT. Součástí práce bylo vytvoření aplikace, pomocí které, byla ověřena správnost funkčnosti softwarového PLC.

Navržený průmyslový komunikační systém je zobrazen prostřednictvím vizualizace na vzdáleném uživatelském dotykovém panelu, který je propojen s řídicím vestavěným PC pomocí TCP/IP rozhraní.

## **Klíčová slova**

softwarové PLC, QNX6 RTOS, průmyslová komunikační síť, komunikační protokol PROFINET, komunikační protokol TCP/IP, vizualizace, distribuovaná periferie, vestavěné PC

## **Abstract**

In this thesis I dealt with the implementation of the software PLC and its connection to industrial communication network PROFINET. The implemented software PLC was designed for real time operating system RTOS QNX6. In terms of communication between the hardware modules has been used communication protocol PROFINET RT. Part of this work was to create an application what was used for check the correctness of the functionality of the software PLC.

Designed industrial communication system is shown through the visualization of the remote user's touch panel, which is connected to the control PC via built-in TCP / IP interface.

## **Keywords**

software PLC, QNX6 RTOS, industrial communication network, communication PROFINET protocol, communication TCP/IP protocol, visualization, distributed peripherals, embedded PC

**Seznam použitých symbolů a zkratk:**

<b>CBA</b>	<b>Component Based Automation</b>
<b>DCOM</b>	<b>Distributed Component Object Mod</b>
<b>FTP</b>	<b>File Transfer Protocol</b>
<b>GSDML</b>	<b>General Station Description Markup Language</b>
<b>ID</b>	<b>Identity</b>
<b>IEC/EN</b>	<b>International Electrotechnical Commission/ European standards</b>
<b>IEEE</b>	<b>Institute of Electrical and Electronics Engineers</b>
<b>IO</b>	<b>Input/Output</b>
<b>IP</b>	<b>Internet Protocol</b>
<b>IPC</b>	<b>Inter Process Communication</b>
<b>IRT</b>	<b>Isochronous Real Time</b>
<b>ISO/OSI</b>	<b>International Organization for Standardization/Open Systems Interconnection</b>
<b>MAC</b>	<b>Media Access Control</b>
<b>PC</b>	<b>Personal Computer</b>
<b>PCI</b>	<b>Peripheral Component Interconnect</b>
<b>PLC</b>	<b>Programmable Logic Controller</b>
<b>POSIX</b>	<b>Portable Operating System Interface</b>
<b>RT</b>	<b>Real Time</b>
<b>RTOS</b>	<b>Real-Time Operating System</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>WI-FI</b>	<b>Wireless Fidelity</b>

# Obsah

<b>1. Úvod .....</b>	<b>1</b>
<b>2. Operační systém QNX6 RTOS a komunikační průmyslová síť PROFINET.....</b>	<b>2</b>
2.1. Operační systém QNX6 RTOS .....	2
2.1.1. Architektura operačního systému QNX .....	2
2.1.2. Norma POSIX a QNX6 RTOS .....	3
2.1.3. Vlákna a procesy operačního systému QNX .....	3
2.1.4. Návrh architektury průmyslového systému .....	4
2.2. Komunikační průmyslová síť PROFINET .....	6
2.2.1. Topologie distribuované průmyslové sítě PROFINET .....	6
2.2.2. PROFINET a ISO/OSI model.....	7
2.2.3. PROFINET CBA, RT a IRT .....	8
2.2.4. PROFINET IO Controller .....	11
2.2.5. PROFINET IO Device .....	12
2.2.6. PROFINET IO Supervisor.....	12
<b>3. Konfigurace komunikačních hardwarových prvků a nastavení průmyslové sítě PROFINET pro operační systém QNX 6 RTOS.....</b>	<b>14</b>
3.1. Hardwarová konfigurace a topologie průmyslové sítě PROFINET.....	14
3.2. Syntaktická analýza (PARSING) souborů hardwarové konfigurace QNX softwarové PLC pro distribuovanou periférii ET200MP IM155-5 PN ST .....	17
3.2.1. Syntaktická analýza (PARSING) konfiguračních souborů .....	18
<b>4. Softwarové PLC pro operační systém QNX 6 RTOS .....</b>	<b>19</b>
4.1. Návrhový vzor QNX softwarové PLC.....	19
4.2. Systémový modul QNX softwarové PLC .....	21
4.3. Programový modul QNX softwarové PLC .....	21
4.3.1. Testovací aplikace pro QNX softwarového PLC .....	22
4.4. Komunikační modul PROFINET RT/IRT (softwarové PLC) .....	23
4.4.1. PROFINET - ovladač pro komunikační kartu Siemens CP1616 (QNX)..	23
4.5. Komunikační modul TCP/IP a QNET (softwarové PLC).....	23
4.5.1. TCP/IP ovladač pro QNX 6 Neutrino RTOS.....	24
4.5.2. QNET ovladač pro QNX 6 Neutrino RTOS .....	24
4.6. Hardwarová konfigurace QNX softwarového PLC pomocí GSD souboru .....	25

<b>5. Vzdálená vizualizace QNX softwarové PLC a distribuované periferie ET200 MP IM155-5 PN ST .....</b>	<b>27</b>
5.1. Průmyslový operátorský panel s vestavěným PC a QNX 6.5.0 SP1 RTOS .....	27
5.2. Propojení modulu vizualizace s QNX softwarovým PLC pomocí TCP/IP rozhraní (klient/server).....	27
5.3. Grafická knihovna QT pro QNX 6 RTOS .....	28
5.4. Vzdálená grafická vizualizace řídicích a akčních hardwarových modulů pomocí knihovny QT .....	28
<b>6. Zhodnocení .....</b>	<b>30</b>
<b>7. Citovaná literatura.....</b>	<b>31</b>



# 1. Úvod

Tato práce je součástí projektu TAČR vedeného pod evidenčním číslem TA01010632 a je učena pro společnost Disam RT s.r.o, která se zaměřuje na monitorovací a řídicí systémy technologických procesů v oblasti chemického, petrochemického, ropného, důlního průmyslu a v oblasti energetiky a teplárenství.

Dílčím cílem tohoto projektu bylo vytvořit průmyslový systém, který by demonstroval možnosti komunikace prostřednictvím různých komutačních rozhraní. Ukazuje možnosti řízení distribuovaných periférií nebo jejich vizualizaci za pomoci uživatelského dotykového panelu.

V rámci mé bakalářské práce jsem se zabýval jednou z dílčích částí projektu, která byla věnována implementaci softwarového PLC a jeho komunikaci se vzdálenou distribuovanou periférií E200MP IM155-5 PN ST pomocí komunikační průmyslové sítě PROFINET. Pro testovací účely průmyslové komunikace a softwarového PLC jsem vytvořil vizualizaci, která je zobrazena na uživatelském dotykovém panelu. Prostřednictvím vizualizace může uživatel ovládat funkce softwarové PLC a sledovat vzdáleně distribuované technologické vstupy a výstupy dané periferie.

Tato bakalářská práce je rozdělena do 4 základních témat, z nichž první téma v kapitole 2 se zabývá problematikou operačního systému reálného času QNX 6 RTOS a průmyslovou komunikační sítí PROFINET. Následující téma v kapitole číslo 3 se věnuje hardwarové konfiguraci modulů navrženého průmyslového systému. Téma v kapitole číslo 4 popisuje softwarové PLC, jeho jednotlivé moduly, komunikační ovladače síťových rozhraní, konfigurace softwarového PLC a testovací program. V poslední tematické části uvedené v kapitole 5 je řešena vizualizace průmyslového systému na uživatelském dotykovém panelu s operačním systémem reálného času QNX 6 Neutrino.

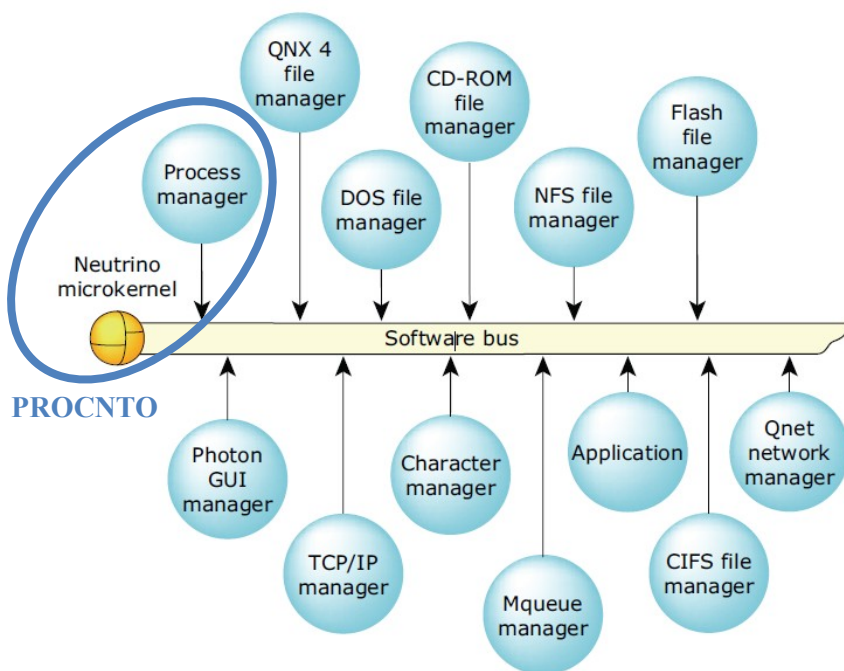
## 2. Operační systém QNX6 RTOS a komunikační průmyslová síť PROFINET

### 2.1. Operační systém QNX6 RTOS

QNX Neutrino RTOS je robustní a plnohodnotný operační systém, který dokáže vykonávat operace v reálném čase. Jeho modulární architektura a konstrukce mikro-jádra umožňuje uživatelům vytvářet vysoce spolehlivé a optimalizované systémy s nízkými náklady na vlastnictví. Tento operační systém se hlavně uplatňuje v odvětvích, kde se využívají vestavěné systémy, kde je požadována vysoká spolehlivost a bezpečnost. V rámci své architektury nabízí osvědčenou strategii přechodu z single-core do multi-core procesů. [1]

#### 2.1.1. Architektura operačního systému QNX

Operační systém QNX Neutrino se skládá z malého jádra zvaného Neutrino microkernel, které řídí skupinu procesů. Každý proces má svoje proces-ID (pid) a obsahuje jedno nebo více vláken. QNX 6 využívá softwarovou sběrnici, která umožňuje dynamicky připojit vstupní nebo výstupní moduly operačního systému, kdykoli je potřeba.



Obr. 1 Architektura mikrojádra [1]

Všechny služby operačního systému, kromě těch, které spravuje modul PROCNTO, jsou zpracovávány pomocí standardních procesů. Systémové procesy se výrazně neliší od jakéhokoliv uživatelsky psaného programu. Systémové procesy používají stejné veřejné API a vhodně privilegované uživatelské procesy mají k dispozici i stejné služby jádra.

Každý prvek operačního systému QNX 6 (driver, protokol stack, systém souborů, uživatelská aplikace, atd.) běží v bezpečném, paměti chráněném uživatelském prostoru

mimo jádro. Prakticky každá komponenta může selhat, v tom případě bude automaticky, pokud je to vyžadováno, restartována bez vlivu na jiné složky nebo jádro operačního systému. Žádný jiný komerční systém reálného času nenabízí tento stupeň ochrany a izolace. [1]

### 2.1.2. Norma POSIX a QNX6 RTOS

QNX Neutrino RTOS podporuje stovky příkazů, nástrojů a programovacích rozhraní standardu POSIX, které maximalizují přenositelnost kódu a opětovného použití. Toto bohaté, na standardech založené prostředí zahrnuje známé rozhraní příkazového řádku a umožňuje vývojářům rychle přenášet Linuxové, UNIXové a další open-source programy do operačního systému QNX. QNX Neutrino RTOS je certifikován v souladu s normou 1003,13-2.003.

QNX 6 Neutrino byl navržen od základu pro normy POSIX. Tento přístup eliminuje komplexní, POSIXovou adaptační vrstvu, kterou potřebují jiné RTOS, což umožňuje větší výkon a nižší spotřebu paměti pro embedded aplikace.

### 2.1.3. Vlákna a procesy operačního systému QNX

Při vytváření aplikace (v reálném čase, grafické nebo jiné), může chtít vývojář v rámci jedné aplikace spustit několik algoritmů současně. Této souběžnosti je dosaženo pomocí POSIXového vláknového model. Tento model definuje proces obsahující jeden nebo více vláken.

Vlákno se plánuje a vykonává v jádře. Proces si můžeme představit jako úložiště pro vlákna, definující adresní prostor, ve kterém se budou vlákna vykonávat.

Následující *pthread\_\** (POSIXová vlákna) knihovna funkcí nezahrnuje funkce jádra (microkernel): [1]

- *pthread\_attr\_destroy()*
- *pthread\_attr\_getdetachstate()*
- *pthread\_attr\_getinheritsched()*
- *pthread\_attr\_getschedparam()*
- *pthread\_attr\_getschedpolicy()*
- *pthread\_attr\_getscope()*
- *pthread\_attr\_getstackaddr()*
- *pthread\_attr\_getstacksize()*
- *pthread\_attr\_init()*
- *pthread\_attr\_setdetachstate()*
- *pthread\_attr\_setinheritsched()*
- *pthread\_attr\_setschedparam()*
- *pthread\_attr\_setschedpolicy()*
- *pthread\_attr\_setscope()*
- *pthread\_attr\_setstackaddr()*
- *pthread\_attr\_setstacksize()*
- *pthread\_cleanup\_pop()*
- *pthread\_cleanup\_push()*
- *pthread\_equal()*
- *pthread\_getspecific()*
- *pthread\_setspecific()*
- *pthread\_key\_create()*

- *pthread\_key\_delete()*
- *pthread\_self()*

V následující tabulce jsou uvedeny funkce POSIXových vláken, které odpovídají funkcím jádra (microkernel), což umožňuje výběr rozhraní.

POSIX call	Microkernel call	Description
<i>pthread_create()</i>	<i>ThreadCreate()</i>	Vytvoří nové vlákno.
<i>pthread_exit()</i>	<i>ThreadDestroy()</i>	Zruší vlákno.
<i>pthread_detach()</i>	<i>ThreadDetach()</i>	Odpojí vlákno (nemusí být připojeno).
<i>pthread_join()</i>	<i>ThreadJoin()</i>	Připojí vlákno čekající na exit status.
<i>pthread_cancel()</i>	<i>ThreadCancel()</i>	Zruší vlákno v příštím místě zrušení.
N/A	<i>ThreadCtl()</i>	Změní typické chování vlákna.
<i>pthread_mutex_init()</i>	<i>SyncTypeCreate()</i>	Vytvoří mutex.
<i>pthread_mutex_destroy()</i>	<i>SyncDestroy()</i>	Zruší mutex.
<i>pthread_mutex_lock()</i>	<i>SyncMutexLock()</i>	Zamkne mutex.
<i>pthread_mutex_trylock()</i>	<i>SyncMutexLock()</i>	Podmíněně zamkne mutex.
<i>pthread_mutex_unlock()</i>	<i>SyncMutexUnlock()</i>	Odemkne mutex.
<i>pthread_cond_init()</i>	<i>SyncTypeCreate()</i>	Vytvoří condition variable.
<i>pthread_cond_destroy()</i>	<i>SyncDestroy()</i>	Zruší condition variable.
<i>pthread_cond_wait()</i>	<i>SyncCondvarWait()</i>	Čeká na condition variable.
<i>pthread_cond_signal()</i>	<i>SyncCondvarSignal()</i>	Pošle signální condition variable.
<i>pthread_cond_broadcast()</i>	<i>SyncCondvarSignal()</i>	Pošle broadcastovou cond. variable.
<i>pthread_getschedparam()</i>	<i>SchedGet()</i>	Získává parametry a druh plánování vlákna.
<i>pthread_setschedparam()</i>	<i>SchedSet()</i>	Nastavuje parametry a druh plánování vlákna.
<i>pthread_sigmask()</i>	<i>SignalProcMask()</i>	Zkontroluje nebo nastavuje signální masku vlákna.
<i>pthread_kill()</i>	<i>SignalKill()</i>	Pošle signál danému vláknu.

Obr. 2 Tabulka funkcí [2]

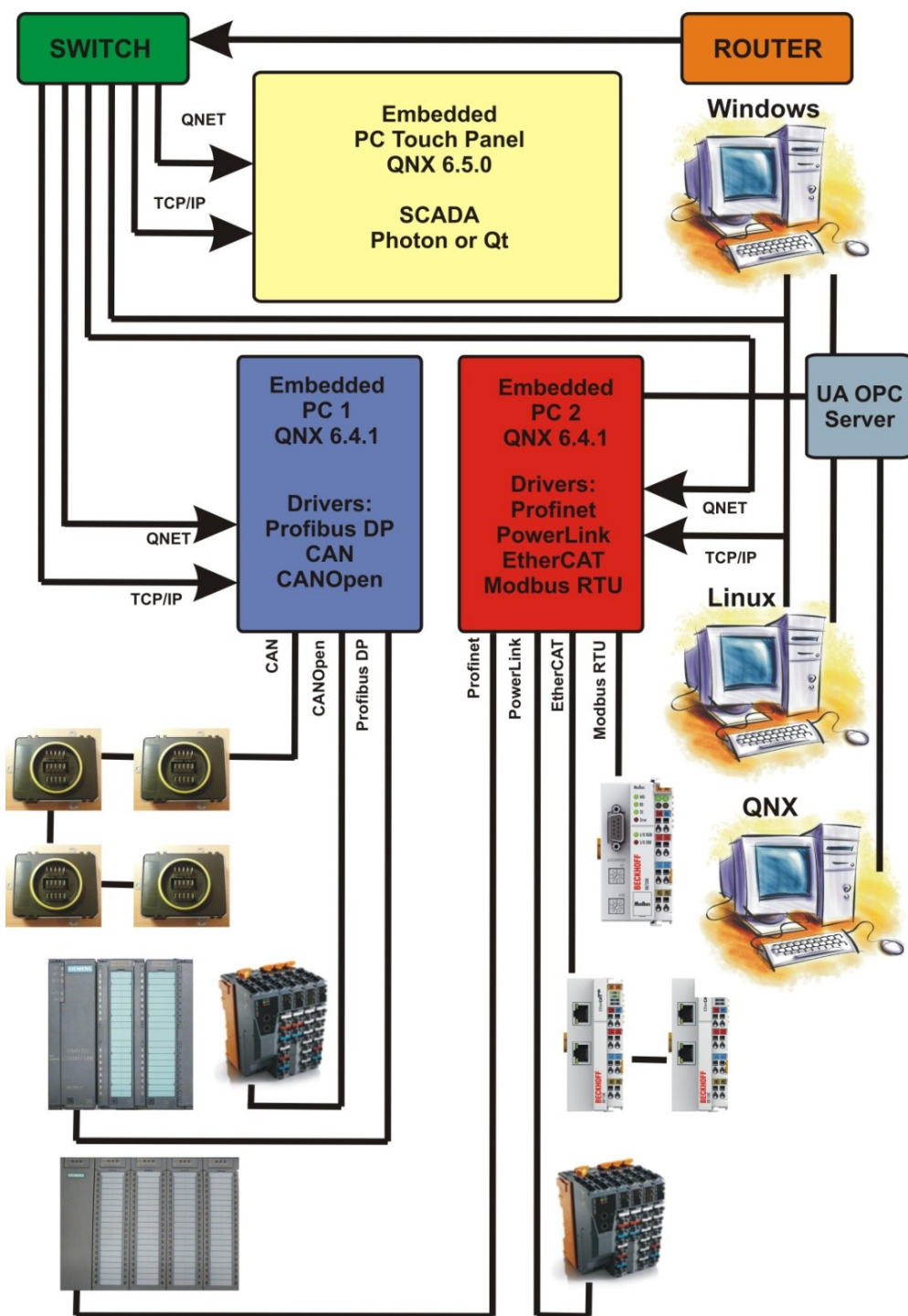
#### 2.1.4. Návrh architektury průmyslového systému

Celý průmyslový systém na následujícím obrázku je tvořen dvěma vestavěnými PC, různými typy průmyslových periferiemi, které komunikují s daným vestavěným PC v rámci komunikačního rozhraní. Jeden z vestavěných PC je rozšířen o dotykový panel určený pro vizualizaci průmyslových procesů.

V této bakalářské práci je využita jen dílčí část navrženého průmyslového systému:

- Vestavěné PC rozšířené o dotykový panel
- Vestavěné PC 2
- Průmyslové komunikační rozhraní PROFINET
- Vzdálená periférie E200MP IM155-5 PN ST

Na dotykovém panelu je zobrazena vizualizace softwarového PLC, které je spuštěno v rámci vestavěného PC 2. Vestavěné PC 2 je propojeno s distribuovanou periferií ET200MP IM155-5 PN ST a jejími I/O moduly pomocí průmyslové sběrnice PROFINET. Komunikace mezi dotykovým panelem a vestavěným PC 2 probíhá prostřednictvím standardní ethernetové sítě (protokol TCP/IP).



Obr. 3 Architektura průmyslového systému

### 2.1.5. QNX6 pro vestavěné PC

Vestavěné PC je počítač, vestavěný do nějakého systému nebo zařízení, které ovládá. Na rozdíl od osobních počítačů jsou vestavěné počítače určené pro jeden účel, tzn., že vykonávají předem definované činnosti.

V našem případě se jedná o multifunkční vestavěné PC EPS-QM57, jehož jádrem je procesor Intel Core i7 o frekvenci 2.0 GHz. Zařízení umožňuje síťové propojení pomocí dvou nezávislých LAN portů. Přes jeden z těchto portů se komunikuje s dotykovým panelem. Vestavěné PC obsahuje také dva PCI sloty, přičemž do prvního PCI slotu je zapojena komunikační karta CP1616, jenž umožňuje komunikaci v rámci průmyslové sítě PROFINET.



Obr. 4 Vestavěné PC

Ve vestavěném PC 2 je nainstalován operační systém QNX 6.4.1, v rámci kterého, je spuštěno softwarové PLC. Softwarové PLC je propojeno s průmyslovou komunikační sítí PROFINET, pomocí ovladače pro komunikační PCI kartu Siemens CP1616.

## 2.2. Komunikační průmyslová síť PROFINET

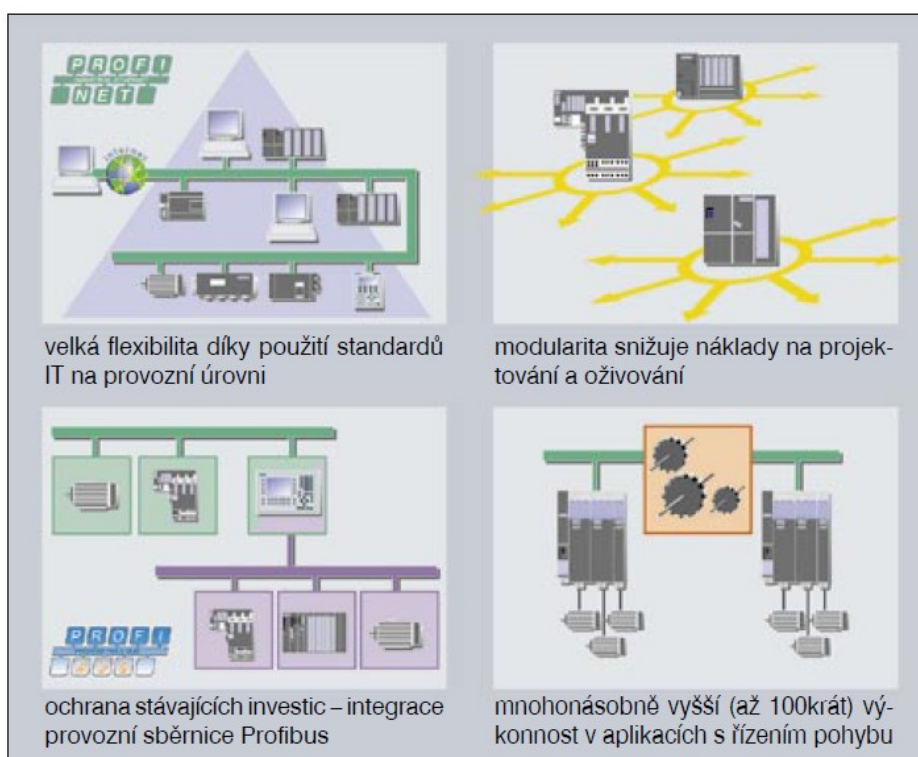
### 2.2.1. Topologie distribuované průmyslové sítě PROFINET

PROFINET je otevřený standart, který se využívá v průmyslové automatizaci. Je založen na průmyslovém ethernetu, využívá TCP/IP standardu a je kompatibilní s klasickým ethernetovým připojením. Oproti klasickému ethernetu je schopen odolávat

těžkým průmyslovým podmínkám a dokáže vydat takový výkon, který požaduje průmyslová automatizace.

### Výhody PROFINETU

- Umožňuje komunikaci v reálném čase
- Umožňuje komunikaci pomocí bezdrátové sítě WLAN
- Nabízí vysoký výkon při řízení pohybu
- Jednoduchá struktura sítě
- Jednoduše rozšiřitelná síť
- IT software



Obr. 5 Charakteristika průmyslové sítě PROFINET [3]

### 2.2.2. PROFINET a ISO/OSI model

Aby bylo možné hodnotit komunikační systém a porovnávat ho s ostatními, je třeba ho zařadit do standardizovaného modelu porovnání. Model ISO/OSI byl vytvořen v roce 1978 jako mezinárodní referenční model komunikací a byl ilustrován do 7 vrstev, které jsou znázorněny na následujícím obrázku.

ISO/OSI			
7b	PROFINET IO Services acc. to IEC 61158		PROFINET CBA (IEC 61158 Type 10)
7a	Connectionless RPC		DCOM Connection oriented RPC
6	empty		empty
5			
4	UDP(RFC 768)		TCP(RFC 793)
3	IP (RFC 791)		
2	Real-Time Enhancements acc. To IEC 61784-2 IEEE802.3, Full-Duplex, IEEE802.1Q, Prio. Tagging		
1	IEEE 802.3 100 Base TX , 100 Base FX		

Obr. 6 Model ISO/OSI [4]

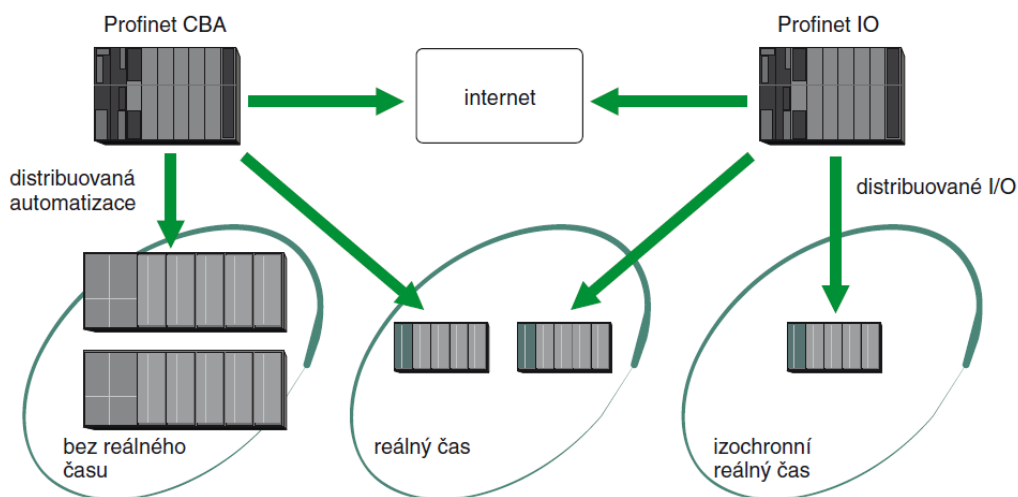
Vrstvy 1 až 7a jsou exkluzivně založeny na základě mezinárodně osvědčených standardů. Funkčnost průmyslového komunikačního rozhraní PROFINET je definována ve vrstvě 7b a řídí se standardy IEC 61158 a IEC 61784.

### 2.2.3. PROFINET CBA, RT a IRT

Architektura komunikačního systému PROFINET je modulární, takže jeho funkční schopnosti lze do jisté míry volit podle povahy dané úlohy. Zásadně je třeba rozlišovat mezi variantami: [5]

- **Profinet CBA** (*Component Based Automation*) jako variantou modulární výstavby komunikačního systému z předem připravených komponent
- **Profinet IO** (*Input/Output*) jako variantou určenou k realizaci propojení decentralizovaných periférií především v cyklickém režimu komunikace.





Obr. 7 Varianty sítě PROFINET [5]

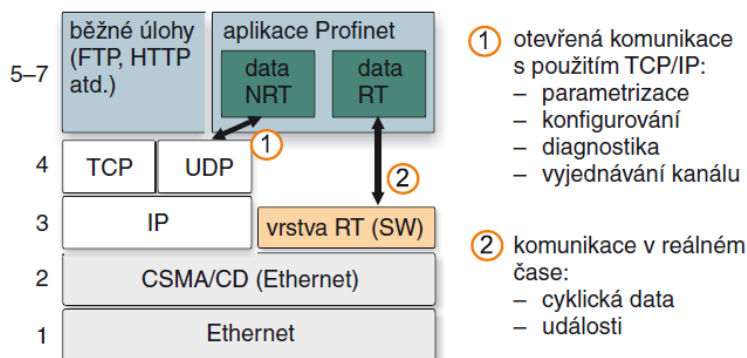
## PROFINET CBA

PROFINET CBA definuje jeden pohled na funkci automatizačního zařízení. Spočívá v tom, že celý automatizovaný celek lze rozdělit na samostatně pracující komponenty a popsat jejich funkci s použitím jazyka XML jako PCD (*Profinet Component Description*).

V databázi PCD jsou pak uloženy všechny údaje o komunikujících entitách (přístrojích, zařízeních), které jsou důležité pro komunikaci. Tvorba komunikačního systému v síti PROFINET CBA pak spočívá jen v projektování spojení s využitím databáze PCD, a nikoliv již v psaní komunikačních programů. V síti PROFINET CBA, při komunikaci bez podpory vlastností reálného času, lze využít profily automatizačních a komunikačních zařízení definované pro síť PROFIBUS. Systém PROFINET pracuje na principu producent-konzument. Producentem je vysílací a konzumentem přijímací uzel. [5]

## Komunikace v reálném čase (RT)

Využívá se pro přenos cyklických dat a alarmů. RT data mají větší prioritu než TCP/IP data. Doba cyklu výměny dat u sběrnice dosahuje rozsahu několika stovek mikrosekund. Komunikace v reálném čase se využívá v průmyslové automatizaci.



Obr. 8 Komunikační model PROFINET RT [5]

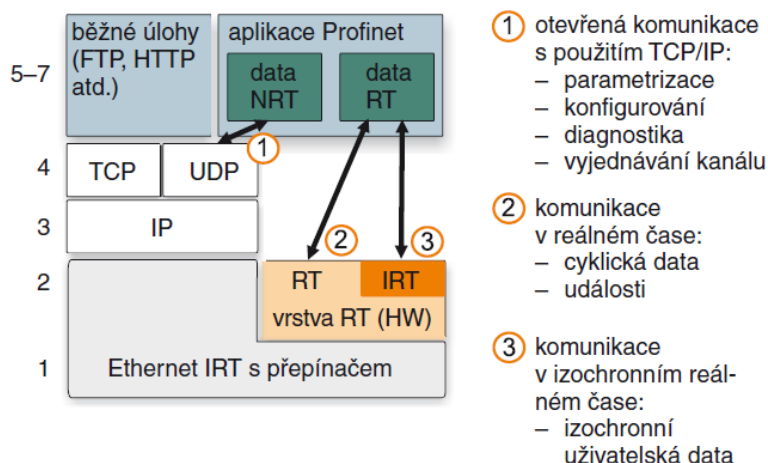
## Třídy pro komunikaci RT:

(Liší se z hlediska determinismu, nikoliv z hlediska výkonu)

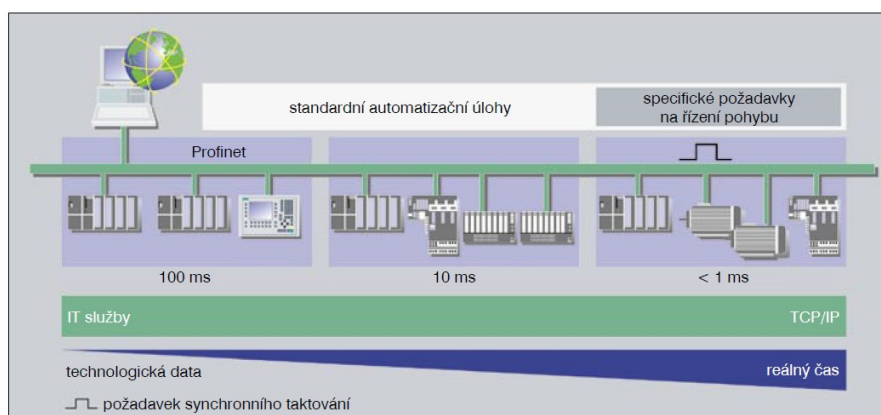
- **RT\_CLASS\_1:** Tato třída umožňuje nesynchronizovaný přenos dat v rámci jedné podsítě. Je to standardní funkce a každé zařízení PROFINET IO podporuje tuto třídu. V této třídě není definován začátek cyklu sběrnice. Lze zde využít standardní ethernetových přepínačů.
- **RT\_CLASS\_2:** Tato třída umožňuje synchronizovaný přenos dat v rámci jedné podsítě. V této třídě jsou všechna zařízení synchronizována a reagují na počáteční interval sběrnice. Tato třída vyžaduje kompatibilní spínače pro PROFINET.
- **RT\_CLASS\_3:** V této třídě je podporován izochronní přenos dat v rámci jedné podsítě.

## Izochronní komunikace (IRT)

Tento druh komunikace se využívá v úlohách s tvrdými požadavky na krátkou dobu odezvy a determinismus, to znamená, že začátek cyklu sběrnice se dodržuje s největší přesností. Cyklus výměny dat je v rozmezí od několika set mikrosekund až několik milisekund. Izochronní komunikace využívá v úlohách, ve kterých řídí motory apod.



Obr. 9 Komunikační model PROFINET IRT [5]

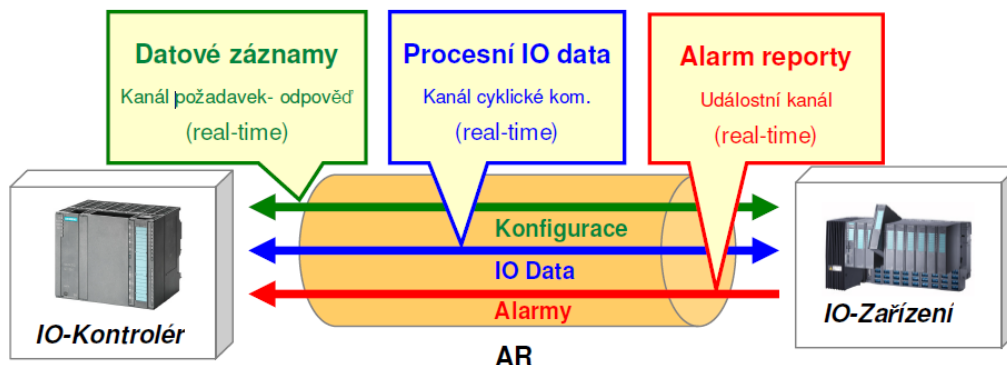


Obr. 10 PROFINET - komunikace [6]

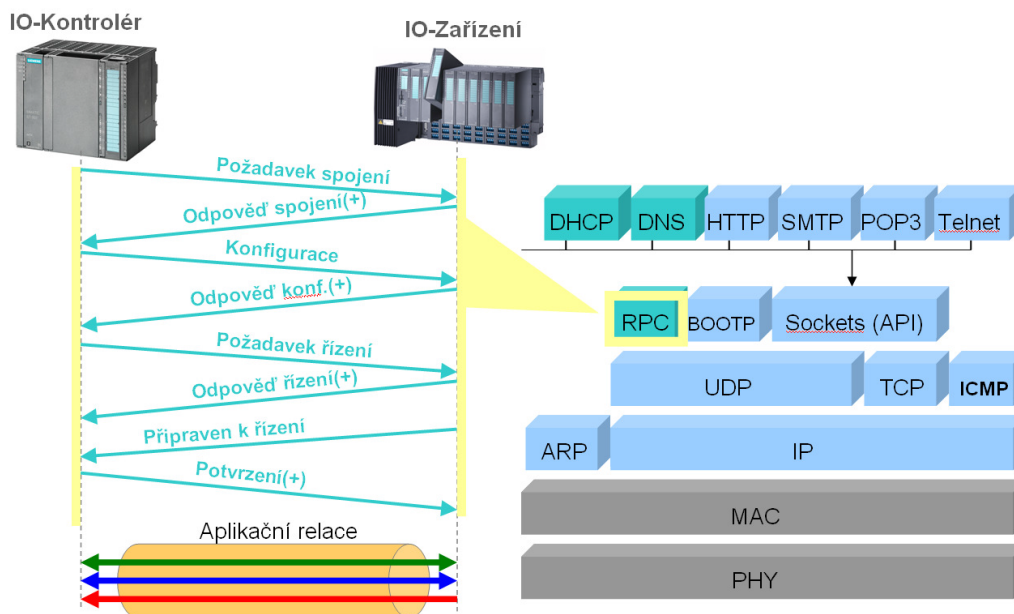
## 2.2.4. PROFINET IO Controller

Je to stanice v automatizačním systému, na kterém řídící program běží. Převážně se jedná o programovatelný logický obvod (PLC). Požaduje procesní data (vstupy z nakonfigurovaných IO-zařízení během zapnutí napájení), zpracovává řídicí program a přenáší vstupní nebo výstupní data do příslušného IO-zařízení.

Aby mohli mezi sebou komunikovat IO-kontroler a IO-zařízení, musí být provedena konfigurace systému, která bude obsahovat komunikační data. Mezi zařízeními probíhá tzv. aplikační relace, v rámci které probíhá komunikace.



Obr. 11 Aplikační komunikační relace PROFINET IO [6]



Obr. 12 Průběh navázání komunikace [6]

IO-Kontrolér v této práci představuje softwarové PLC, které je spuštěno v rámci vestavěného PC, jehož operačním systémem je QNX6 RTOS. Komunikace se vzdálenou periferií probíhá prostřednictvím komunikační karty CP1616.



Obr. 13 Komunikační karta CP1616

Komunikační karta CP1616 umožňuje počítačům s PCI slotem připojení k PROFINET IO zařízením. Je ideální pro časově kritické aplikace, které jsou v oblasti řízení pohybu. Umožňuje synchronizaci řádů stovek pohonů v daném časovém cyklu. Tato karta reprezentuje switch pracujícím v reálném čase a zajišťuje determinismus v průmyslové komunikační síti PROFINET.

### 2.2.5. PROFINET IO Device

IO-Device je decentralizované zařízení, které je připojeno přes PROFINET IO. Ve srovnání s průmyslovou sběrnici PROFIBUS, odpovídá zařízení typu slave. Zařízení jsou adresována pomocí MAC adres, případně IP adres.

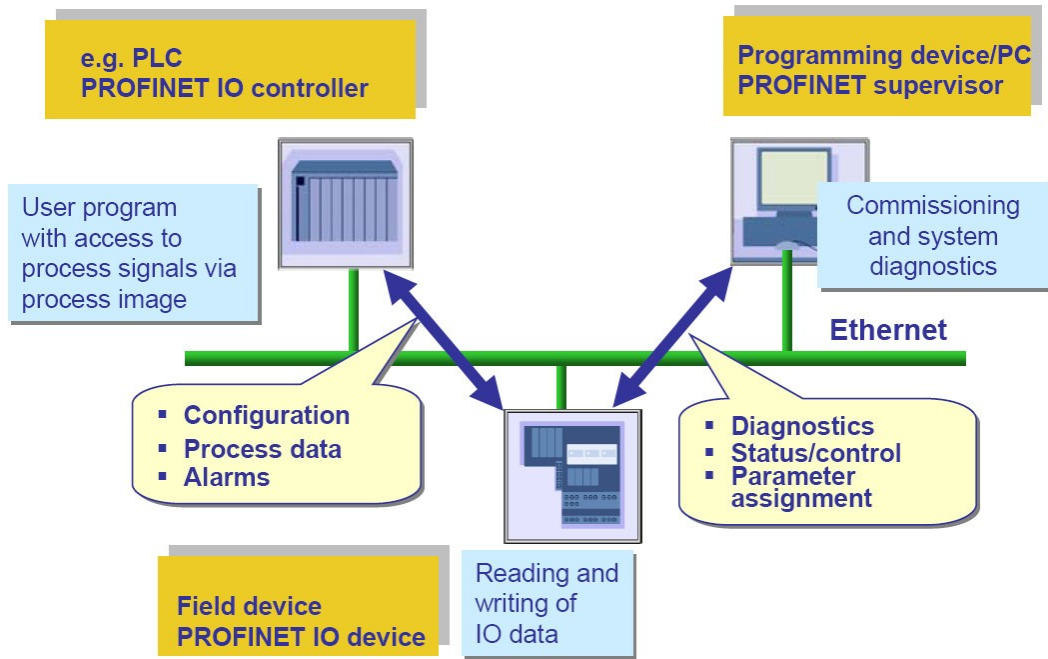
V tomto průmyslovém systému IO-Device představuje distribuovaná periférie ET200MP IM155-5 PN ST, která je rozšířená o dva moduly s analogovými vstupy/výstupy a dva moduly s digitálními vstupy/výstupy.

- AI 8xU/I/RTD/TC ST
- AQ 4xU/I ST
- DI 16x24VDC HF
- DQ 8x24VDC/2A HF

### 2.2.6. PROFINET IO Supervisor

Je určen pro spouštění a sběr diagnostických dat, které může zobrazit koncovému uživateli. IO Supervisor umí číst a zapisovat interní diagnostické údaje související s PROFINET IO nebo diagnostické údaje poskytnuté aplikací zařízení.

IO Supervisor může být programovatelné zařízení, osobní počítač nebo HMI a SCADA zařízení. V mém případě je to průmyslový operátorský panel s vestavěným PC.



Obr. 14 PROFINET IO struktura [4]

### 3. Konfigurace komunikačních hardwarových prvků a nastavení průmyslové sítě PROFINET pro operační systém QNX 6 RTOS

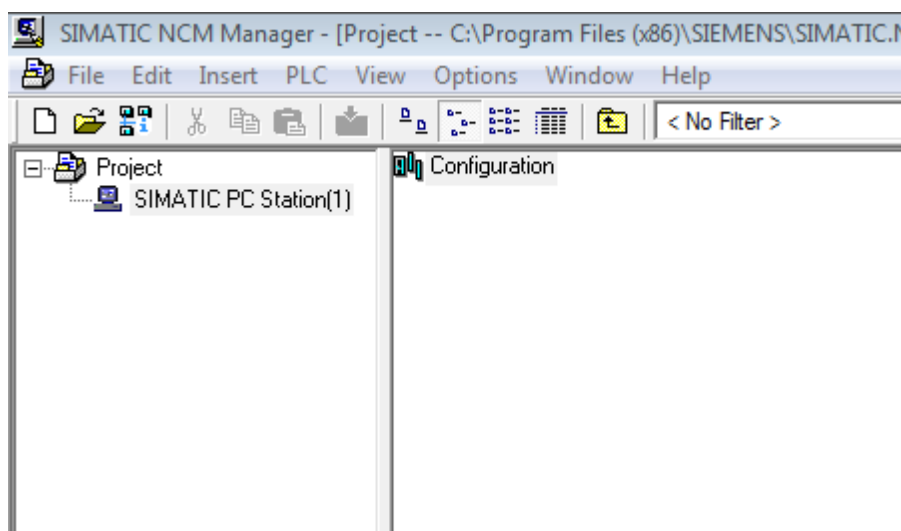
#### 3.1. Hardwarová konfigurace a topologie průmyslové sítě PROFINET

Hardwarová konfigurace je vytvářena pomocí softwarového nástroje SIMATIC NCM Manager nebo SIMATIC Step7 Manager. První jmenovaný nástroj je možno používat bez nároku na licenci, ale má omezené funkce.

Po vytvoření projektu je potřeba vložit objekt PC STATION a následně je možné provést hardwarovou konfiguraci.

Před započatím vytváření hardwarové konfigurace je potřeba doinstalovat tzv. GSDML soubor, který obsahuje data o komponentech využitých v následné konfiguraci.

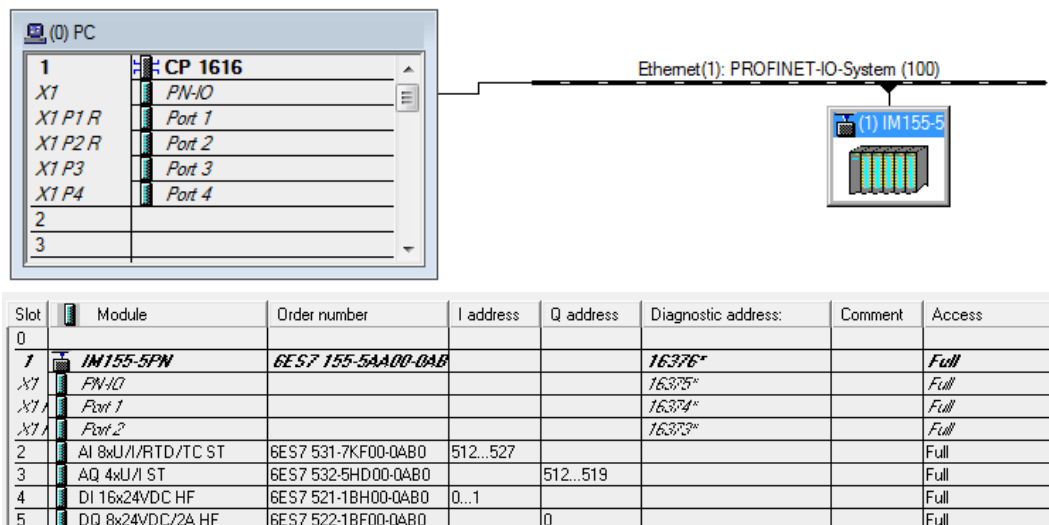
Instalace GSDML souboru: Otevřít **Configuration** → **Options** → **Install GSD File**



Obr. 15 Založení nového projektu

Jako první je do hardwarové konfigurace přidán komunikační modul CP1616. Následně je z něj vyvedena komunikační sběrnice PROFINET-IO. K této sběrnici připojit distribuovanou periférii ET200MP IM155-5 PN S a její přídatné IO moduly:

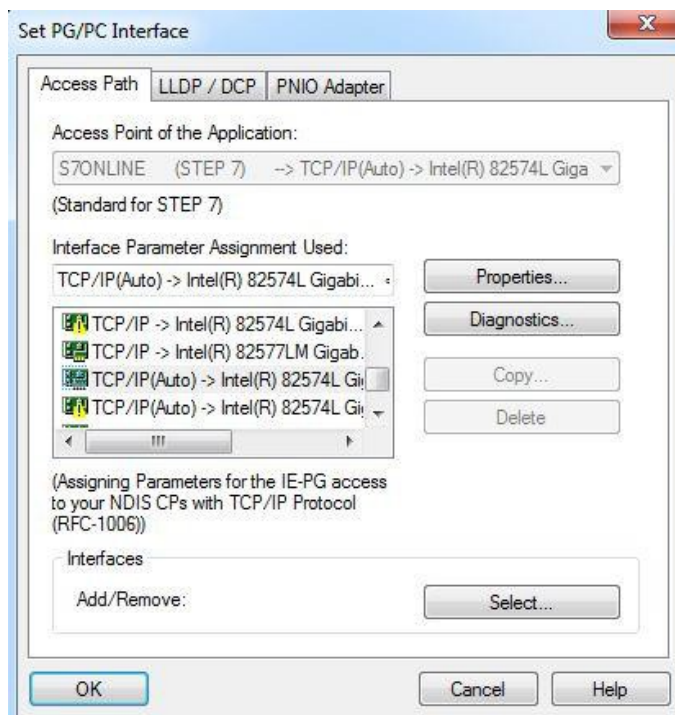
- AI 8xU/I/RTD/TC ST
- AQ 4xU/I ST
- DI 16x24VDC HF
- DQ 8x24VDC/2A HF



Obr. 16 Hardwarová konfigurace

Aby mohla být tato konfigurace nahrána do PC Stationu, musí být správně nastavený síťový adaptér PC. Postup pro konfiguraci síťového adaptéru je popsán v následujících krocích:

1. Otevřít záložku **OPTIONS** → **SET PG/PC INTERFACE**.
2. Zvolit síťovou kartu.
3. Potvrdit OK.

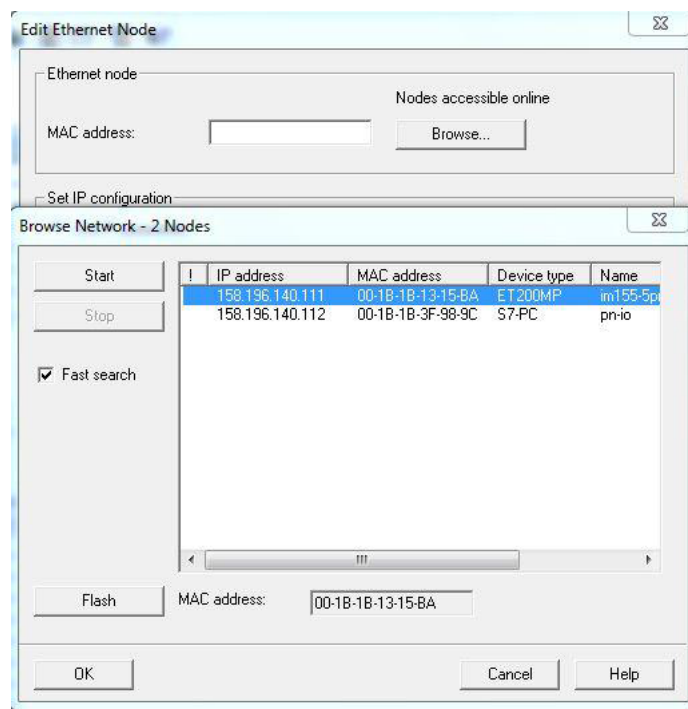


Obr. 17 Výběr síťového adaptéru

Dalším krokem je přiřazení IP adres PC STATIONU (Siemens CP1616) a periférií ET200MP. Postup pro správné přiřazení IP adres je popsán v následujících krocích:

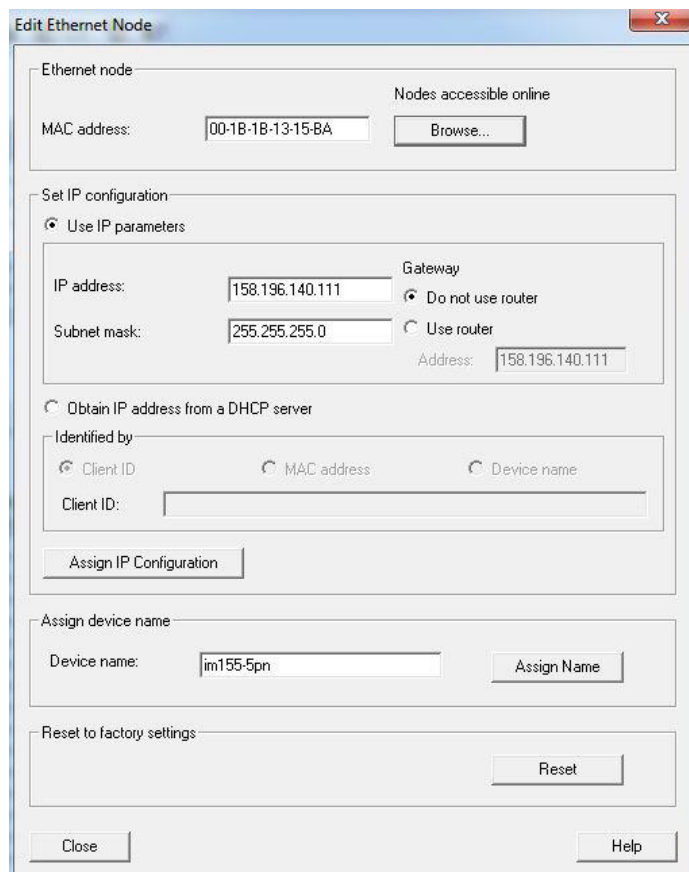
1. Otevřít záložku **PLC**→**Edit Ethernet Node**.
2. Kliknout na „**Browse**“ (pod nápisem „Nodes accessible online“).
3. Z nalezených zařízení vybrat „**ET200MP**“ (Obr. 18) a potvrdit OK.
4. Přiřadit zařízení IP adresu, masku podsítě a následně potvrdit „**Assign IP Configuration**“ (Obr. 19)
5. Pojmenovat zařízení a potvrdit „**Assign Name**“ (Obr. 19).
6. „**Close**“.

Tento postup se opakuje i pro další zařízení.



Obr. 18 Procházení sítě a výběr zařízení





Obr. 19 Přiřazení IP adresy a jména

Nyní je konfigurace dokončena. Před nahráním do PC Station je potřeba konfiguraci ještě zkompilevat.

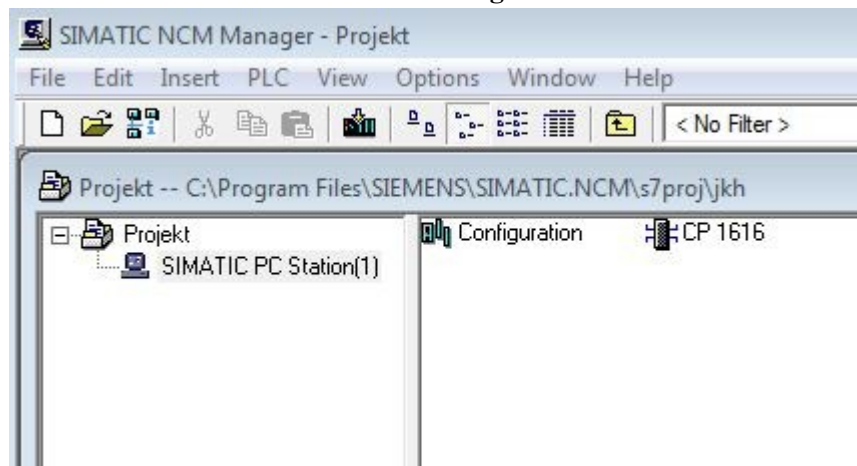
### 3.2. Syntaktická analýza (PARSING) souborů hardwarové konfigurace QNX softwarové PLC pro distribuovanou periferii ET200MP IM155-5 PN ST

Standardní konfigurace probíhá přes tzv. S7 Protokol. Ten obsahuje informace o použitém hardwaru a komunikačním rozhraní. Tyto informace jsou v našem případě např. počty vstupů/výstupu jednotlivých modulů periferie a jsou nezbytné pro softwarové PLC, které podle nich vygeneruje svou vlastní hardwarovou konfiguraci. Bohužel S7 Protokol není tzv. open-source a tudíž ho nelze nijak exportovat nebo editovat.

Přesto lze z prostředí vývojového nástroje SIMATIC Step7 Manager popř. SIMATIC NCM Manager vyexportovat tzv. konfigurační soubor, který obsahuje různé informace o grafickém zpracování konfigurace v konfiguračním prostředí, ale taky ID všech připojených periferií a jejich přídatných modulů.

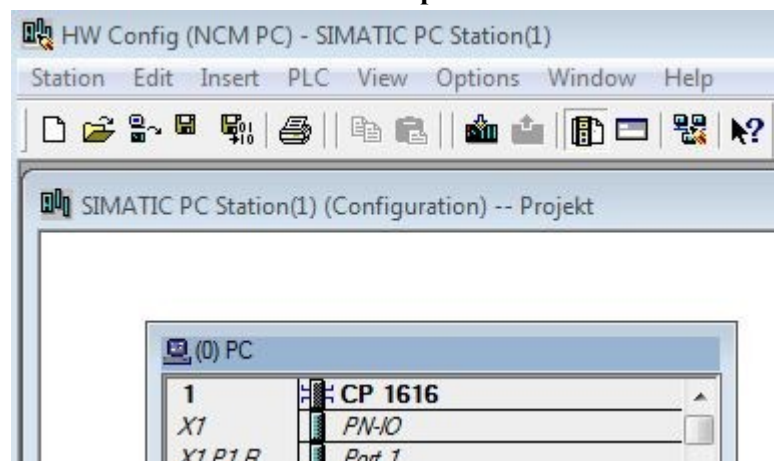
Export konfiguračního souboru se provádí následovně:

1. Otevřít SIMATIC PC Station → **Configuration**



Obr. 20 Export konfiguračního souboru

2. Kliknout na záložku **Station** → **Export**



Obr. 21 Export konfiguračního souboru

### 3.2.1. Syntaktická analýza (PARSING) konfiguračních souborů

K syntaktické analýze je potřeba vyexportovaný konfigurační soubor a GSDML soubor s daty pro danou periferii, což je v našem případě ET200MP. V konfiguračním souboru jsou zapsané ID pod názvem PN\_MODULE\_IDENTBUMBER. Bylo potřeba vytvořit dynamický program, který by dovedl provést syntaktickou analýzu konfiguračního i GSDML souboru a vyčíst si z nich data, potřebná k vytvoření hardwarové konfigurace pro softwarové PLC.

Vytvořený program otevře konfigurační soubor, ve kterém najde řádky obsahující název PN\_MODULE\_IDENTBUMBER. Z každého takového řádku vyčte ID číslo a převede ho do hexadecimálního tvaru. Všechny tyto hexadecimální ID čísla začne hledat v GSDML souboru. V případě, že program narazí na shodné ID číslo, vyhledá požadované informace o daném zařízení a uloží je. Tyto získané data softwarové PLC zpracuje a vytvoří hardwarovou konfiguraci.

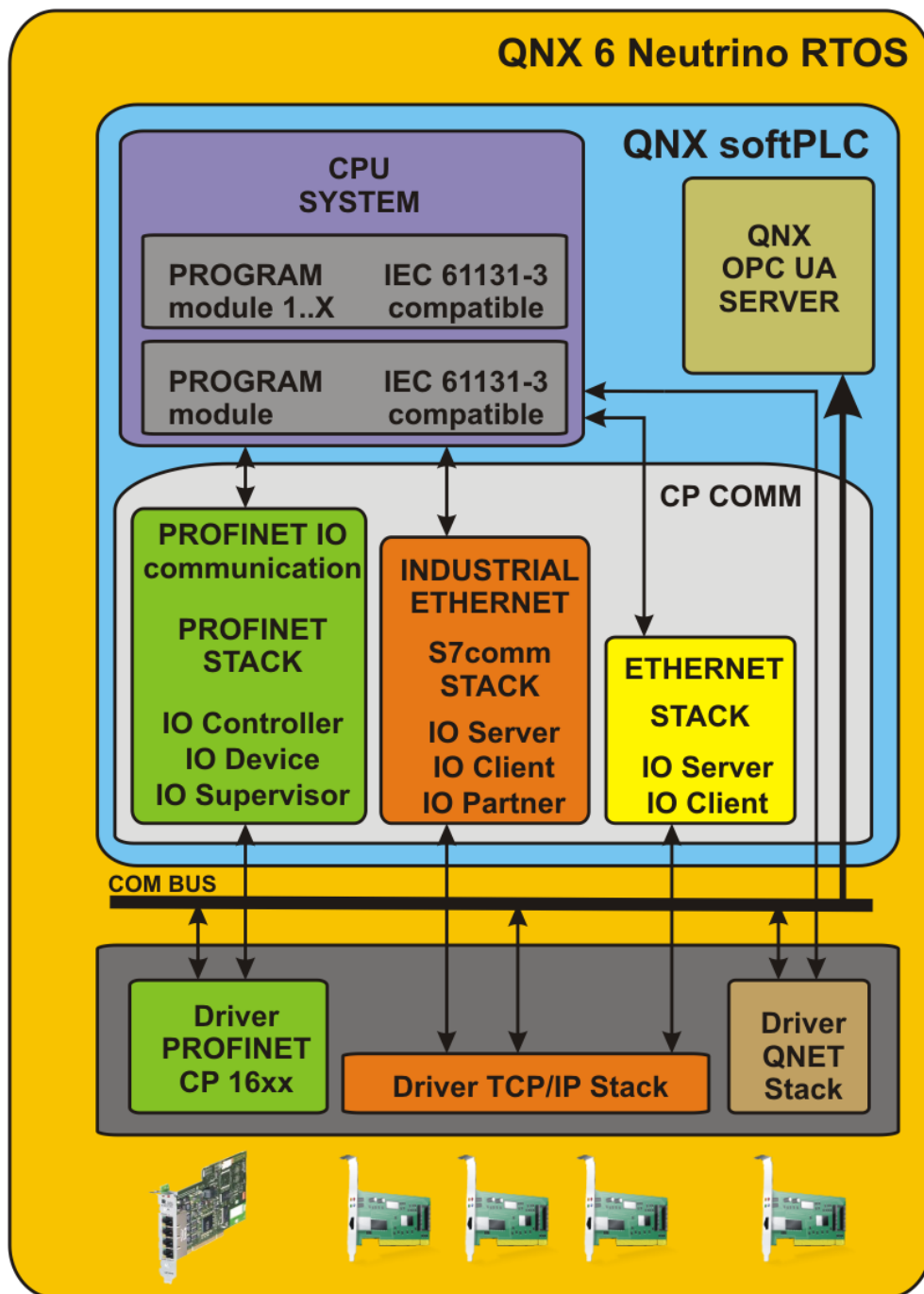
## 4. Softwarové PLC pro operační systém QNX 6 RTOS

### 4.1. Návrhový vzor QNX softwarové PLC

Návrhový vzor softwarového PLC, který byl sestaven v rámci této bakalářské práce, vychází z tradičního návrhu hardwarového PLC, kde jsou softwarově emulovány jeho některé prvky a součásti. Vlastní funkcionalita softwarového PLC, jak je tomu u tradičního PLC, byla zachována. Důraz je kladen hlavně na kompatibilitu softwarového PLC s normou IEC 61131-3 a jeho vybavení z hlediska komunikačních možností a zachování determinismu v celém řídicím systému. Aby bylo možné provozovat softwarové PLC, je nutné zajistit vhodné deterministické prostředí, ve kterém bude spuštěno. Tím to prostředím byl zvolen operační systém reálného času QNX 6 Neutrino.

Cílem návrhu softwarového PLC je využít maximálně výkon hostitelského systému (vestavěné PC s operačním systémem QNX 6.4.1), pro náročné výpočetní úlohy, se zachováním všech výhod stávajících PLC. Pro zajištění maximální výkonu komunikace a zároveň její determinovanosti bylo zvolené přenosové médium Ethernet společně s deterministickými hardwarovými síťovými prvky určenými pro práci v reálném čase. Jako průmyslový přenosový protokol byl zvolen PROFINET, který je diskutován v kapitole číslo 2.2.

Aby softwarové PLC mohlo komunikovat v rámci průmyslové sítě PROFINET, je nutné zajistit fyzické propojení mezi vestavěným PC a deterministickou komunikační sítí PROFINET. Toto spojení, jak je popsáno v kapitole 2.2.4., je zajištěno pomocí PCI komunikační karty CP1616 od firmy Siemens. Aby operační systém reálného času QNX 6 Neutrino, byl schopen komunikovat s touto kartou (CP1616) musí být při startu (nebo před spuštěním softwarového PLC) operačního systému, také zaveden její ovladač. Softwarové PLC v rámci sítě PROFINET se pak konfiguruje, jako IO Controller (master), ale mohou nastat situace, kdy je potřeba emulovat funkci chování softwarového PLC jako IO Device (slave), pokud nechci použít PN/PN Coupler. Poslední možností, jak lze konfigurovat komunikační rozhraní v rámci průmyslové sítě PROFINET je funkce IO Supervisor. Všechny zmíněné nastavení chování komunikačního rozhraní jsou popsány v kapitole 2.2. Dalším způsobem komunikace softwarového PLC je průmyslový ethernet. Pro tento způsob komunikace se využívá standardních síťových ovladačů, které poskytuje operační systém reálného času QNX 6.4.1. Nad těmi to ovladači, jak pak implementován S7 protokol, který složí pro konfiguraci a komunikaci se zařízeními firmy Siemens, které jsou uvedené v této bakalářské práci. V rámci této komunikace může softwarové PLC fungovat jako IO Server, IO Client a IO Partner. Dalším druhem komunikace, kterou softwarové PLC poskytuje je tradiční TCP/IP komunikace, kde nabízí možnost konfigurovat softwarové PLC, jako IO server nebo IO client a nebo obojí zároveň.



Obr. 222 Návrhový vzor softwarového PLC pro operační systém QNX 6 Neutrino

Posledním způsobem komunikace, kterou softwarové PLC poskytuje je komunikace pomocí QNETu. Tento způsob komunikace je blíže popsán v kapitole 4.3.6. Využívá se hlavně z důvodu flexibility návrhu architektury, kdy umožňuje softwarovému PLC běžet na různých počítačích, které jsou součástí sítě QNET a distribuovat výkon celého systému v rámci deterministické sítě. Typickým příkladem může být OPC UA server nebo klient, který poskytuje nebo získává data z technologie.

## 4.2. Systémový modul QNX softwarové PLC

Systémový modul softwarového PLC v sobě integruje programové moduly, které jsou plně kompatibilní se standardem POSIX pro unixové systémy a také s mezinárodním standardem IEC 61131-3 pro programovatelné logické automaty. Systémový modul se skládá z programových vláken, které mají různou prioritu, časový cyklus a funkcionalitu. Základním programovým modulem je hlavní organizační blok (HOB 1) nebo také ho můžeme z programového hlediska nazvat hlavní vlákno softwarového PLC, které je cyklickým programem. Softwarové PLC může obsahovat také další programová pracovní vlákna, která jsou označována jako vedlejší organizační bloky (VOB), které slouží hlavně k vykonávání časového přerušení v širším časovém měřítku nebo periodického přerušení. Jedná se tedy o periodické programy. Poslední oblastí vedlejších organizačních bloků jsou událostmi řízené programy. Mezi tuto skupinu bloků patří zejména časového a hardwarového přerušení a synchronní a asynchronní chybové stavy. Obecně lze tedy organizační bloky rozdělit na hlavní organizační blok, organizační bloky přerušení a chybové organizační bloky. Speciální skupinou organizačních obloků jsou programy, které se vykonávají při spouštění softwarového PLC (studený, teplý a horký restart softwarového PLC).

Hlavní organizační blok (HOB 1) může dále obsahovat systémové funkce (SFC) a systémové funkční bloky (SFB). Mimo těchto systémových funkcí a bloků jsou k dispozici také předdefinované funkce a funkční bloky nebo uživatelsky definované funkce a funkční bloky.

Každý vykonávaný programový blok má svoji prioritu, kterou získá na základě svoji funkcionality. Například HOB 1 má prioritu 1, tedy nejvyšší možnou prioritu. Časová přerušení mají prioritu 2-10, cyklická přerušení 10-15, hardwarová přerušení 16-20 a chybová přerušení 20-30. Je nutné poznamenat, že tyto priority nekorespondují přímo s prioritami v operačním systému reálného času QNX 6 Neutrino, který nabízí 1-63 priorit, ale vytváří speciální skupinu priorit pro softwarové PLC, které využívá priority vláken operačního systému. Důvodem je přehlednost a modularita v rámci operačního systému, kdy na jednom vestavěném PC může běžet hned několik softwarových PLC s různou funkcionalitou a výpočetními nároky. Aby byly jednotlivé softwarové PLC na sobě nezávislé z hlediska hardwarových prostředků, využívá se adaptivní partitioning, který vyčlení jednotlivým softwarovým PLC na přesně daný časový úsek cyklu operačního systému dané hardwarové a systémové prostředky. Toto lze provést i pro jiné aplikace, které v daný čas souběžně běží na společném hardwaru společně se softwarovými PLC. Tím je zaručeno, že žádná jiná aplikace nebo systémový prostředek nezapříčiní pád operačního systému a ani jednotlivých softwarových PLC zařízení.

## 4.3. Programový modul QNX softwarové PLC

Programový modul softwarového PLC je vlákno, které vykonává svoji funkcionalitu a je vytvářeno systémovým modulem. Jednotlivé programové moduly mají svoji prioritu, podle účelu využití. Každý programový modul splňuje mezinárodní standard IEC 61131-3 a je možné ho programovat pomocí 7 programovacích jazyků:

- Ladder diagram (LD), grafický programovací jazyk
- Function block diagram (FBD), grafický programovací jazyk
- Structured text (ST), textový programovací jazyk

- Instruction list (IL), textový programovací jazyk
- Sequential function chart (SFC), programovací jazyk který využívá elementy pro sekvenční a paralelní zpracování chodu programu
- ANSI C programovací jazyk
- ANCI C++ programovací jazyk

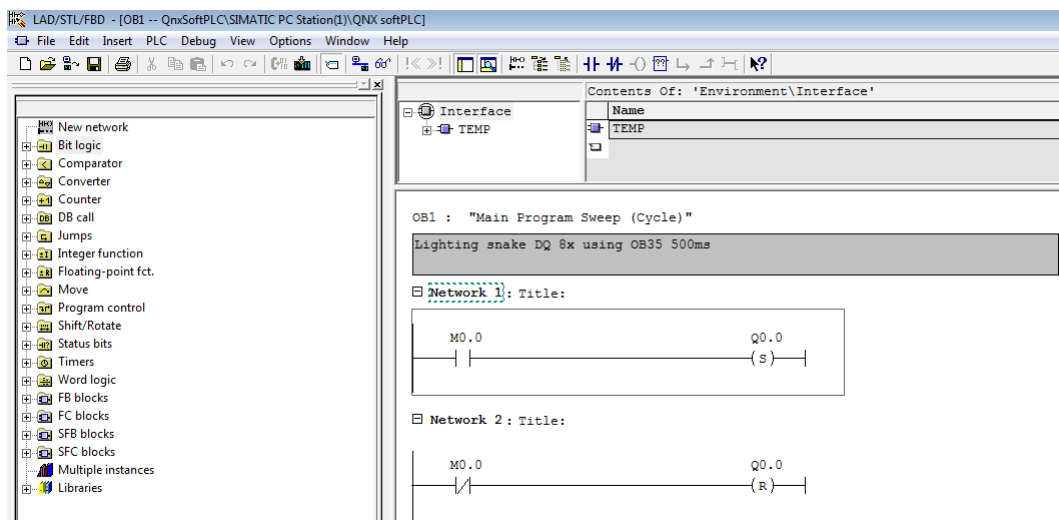
V zadání této bakalářské práce je vytvoření vývojového nástroje pro softwarové PLC. Na základě konzultace s příjemcem této bakalářské práce bylo dohodnuto, že vytvoření takového to nástroje je nad rámec této bakalářské práce a pokud to bude možné, tak bude v této práci použito prostředí Step7 firmy Siemens, které se ve firmě zadavatele využívá. Momentálně je na trhu velké množství vývojových prostředí pro logické programovatelné automaty a každý výrobce dodává s produktem své vlastní vývojové prostředí.

Synchronizaci jednotlivých vláken softwarového PLC je řízena pomocí systémového modulu a synchronizačních prvků, které nabízí operační systém reálného času QNX 6 Neutrino, potažmo mikro-jádro operačního systému.

Komunikace mezi programovým modulem a vývojovým prostředím je zajištěna pomocí S7 komunikačního protokolu, který je definován ve stacku pro komunikaci v rámci průmyslového ethernetu. Aby byla umožněna komunikace, je nutné mít spuštěné softwarové PLC na vestavěném PC a mít toto zařízení připojeno do komunikační sítě kde se nachází počítač s vývojovým prostředím SIMATIC STEP 7 Manager nebo SIMATIC NCM manager.

#### 4.3.1. Testovací aplikace pro QNX softwarového PLC

Pro testovací účely softwarového PLC byla zvolená jednoduchá aplikace světelného hada, který je tvořen 8 LED diodami digitálního výstupního modulu AQ 4xU/I ST v rámci distribuované periferie ET200M IM 155-5. Jednotlivé LED diody se postupně co 500ms rozsvěčují, jedna po druhé a v okamžiku, kdy dosáhne světelný had svého konce, tak se vrací opět zpět. Aplikace využívá jeden hlavní organizační blok HOB 1, jeden cyklický blok přerušení OB 35 a inicializační blok OB100 pro nastavení parametrů během tepleho restartu systému softwarového PLC.



Obr. 233 Testovací program světelného hada pro QNX softwarové PLC v SIMATIC STEP 7 Manageru

#### **4.4. Komunikační modul PROFINET RT/IRT (softwarové PLC)**

Komunikační modul průmyslového rozhraní PROFINET je implementován nad protokolem PROFINET IO, který slouží pro komunikaci mezi zařízením *IO Controller* (PLC, softPLC, aplikace, atd.) a jeho přidruženými distribuovanými periferiemi - *IO Devices* (akční členy, senzory, měniče, vstupně výstupní periferie, atd.). Podrobný popis průmyslového komunikačního protokolu se nachází v kapitole 2.2.

Komunikační modul PROFINETu implementuje jak synchronní komunikaci PROFINET RT, tak isochronní komunikaci PROFINET IRT. Implementovaný PROFINET stack pro operační systém QNX 6 Neutrino umožňuje konfiguraci zařízení ve třech možných režimech a to IO Controller, IO Device a IO Supervisor. Softwarové PLC pracuje v režimu IO controller, ale za jistých podmínek lze upravit nastavení tak, aby se v síti PROFINET chovalo jako IO Device zařízení. Tato konfigurace však není možná v rámci konfigurace SIMATIC STEP 7 manageru.

V této práci je využíván průmyslový protokol PROFINET RT pro komunikaci se vzdálenou periferií ET200MP IM155-5 řady S7 1500 od společnosti Siemens. Aby bylo možné propojit PROFINET komunikační stack s okolními zařízeními je potřeba instalovat do vestavěného PC PCI komunikační kartu CP1616, což je 4 portový komunikační switch pracující v reálném čase. Při spouštění komunikačního modulu, je nutné pro tuto komunikační kartu zavést ovladač v rámci operačního systému QNX 6.4.1 Neutrino.

##### **4.4.1. PROFINET - ovladač pro komunikační kartu Siemens CP1616 (QNX)**

Komunikační ovladač pro síťovou kartu Siemens CP1616 se zavádí buď při startu operačního systému nebo při startu softwarového PLC. Každá komunikační karta CP1616 obsahuje firmware, který dle své verze podporuje danou funkcionalitu PROFINETU. Komunikační karta, která byla použita v rámci této bakalářské práce, obsahuje firmware verze 2.5.2 od společnosti Siemens.

#### **4.5. Komunikační modul TCP/IP a QNET (softwarové PLC)**

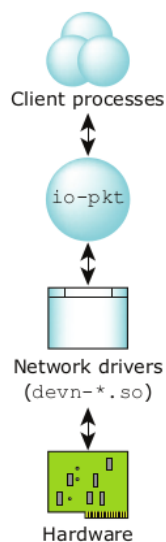
Komunikační modul založený na průmyslovém ethernetu nebo klasickém ethernetu využívající TCP/IP komunikační stack, lze rozdělit do dvou skupin komunikace a to komunikaci, která využívá S7 komunikační protokol a klasickou komunikaci pomocí standardního TCP/IP rozhraní. Rozdíl je v tom, že první typ komunikace do TCP rámce přidá vlastní implementaci S7 protokolu. Speciální typem komunikace pro systémy s operačním systémem QNX je vlastní deterministická síť nazývaná QNET .

Pomocí QNETu lze propojovat jednotlivé síťové uzly, sdílet hardwarové prostředky, distribuovat výkon celého systému a propojit více softwarových PLC na různých vestavěných zařízeních. Jednou z možností, která je nastíněná v rámci návrhu architektury softwarového PLC je distribuce dat na vzdálená zařízení pomocí OPC UA serveru. OPC UA server může běžet na jiném vestavěném PC a právě pomocí QNETu může být integrován v rámci softwarového PLC. OPC UA server umožňuje napojení OPC UA klientu, kteří běží na jiných platformách operačních systémů (Windows, Linux, Mac OS, FreeBSD, atd.)

#### 4.5.1. TCP/IP ovladač pro QNX 6 Neutrino RTOS

Termín TCP / IP jsou prakticky dva různé protokoly: TCP a IP. Vzhledem k tomu, že tyto protokoly byly použity tak často společně se TCP / IP stal standardem terminologie v dnešním Internetu. V podstatě se TCP / IP týká síťových komunikací, kde je TCP přenos využíván k doručování dat přes IP síť.

Aby mohl být TCP/IP protokol použit, je potřeba použít následující softwarové komponenty na obrázku:



Obr. 244 SW komponenty pro TCP/IP [1]

**io-pkt** – Manažer poskytující podporu pro dynamicky zaváděné síťové moduly. Zahrnuje plně vybavený TCP / IP stack odvozený od NetBSD kódové základny.

**devn-\*** - Síťový manažer, který vytváří rozhraní s hardwarem.

#### 4.5.2. QNET ovladač pro QNX 6 Neutrino RTOS

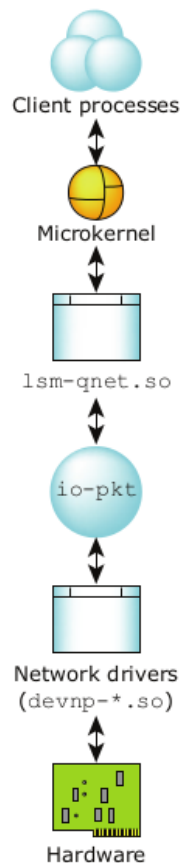
QNET je komunikační protokol pro distribuované sítě. Pomocí QNETu je možné vytvořit tzv. Neutrino nativní síť.

Neutrino nativní síť je skupina vzájemně propojených pracovních stanic, pracujících pouze pod operačním systémem QNX Neutrino. V této síti může program transparentně přistupovat k libovolným zdrojům (soubor, zařízení nebo proces) na jiném uzlu (počítače nebo pracovní stanice) v místní podsíti.

Ve většině případů není potřeba konkrétní ovladače pro hardware. Na příklad pro realizaci lokální sítě pomocí ethernetového hardwaru nebo pro realizaci TCP/IP sítí, které vyžadují IP zapouzdření. V těchto případech je základem *io-pkt*\* a TCP/IP vrstva je dostatečná pro rozhraní s vrstvou QNET, která slouží vysílání a příjem paketů.

Aby mohl být QNET protokol použit, je potřeba použít následující softwarové komponenty:





Obr. 255 SW komponenty pro QNET [1]

**io-pkt\*** - Manažer poskytující podporu pro dynamicky zaváděné síťové moduly.

**devn-\***, **devnp-\*** - Síťový manažer, který vytváří rozhraní s hardwarem.

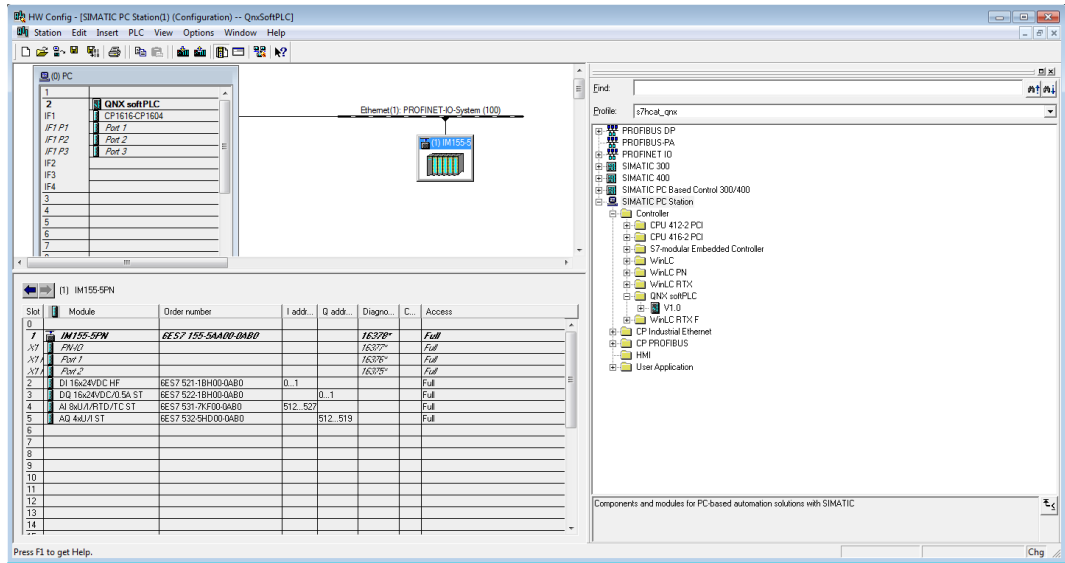
**lsm-qnet.so** - Nativní správce sítě pro implementaci QNET protokolů.

#### 4.6. Hardwarová konfigurace QNX softwarového PLC pomocí GSD souboru

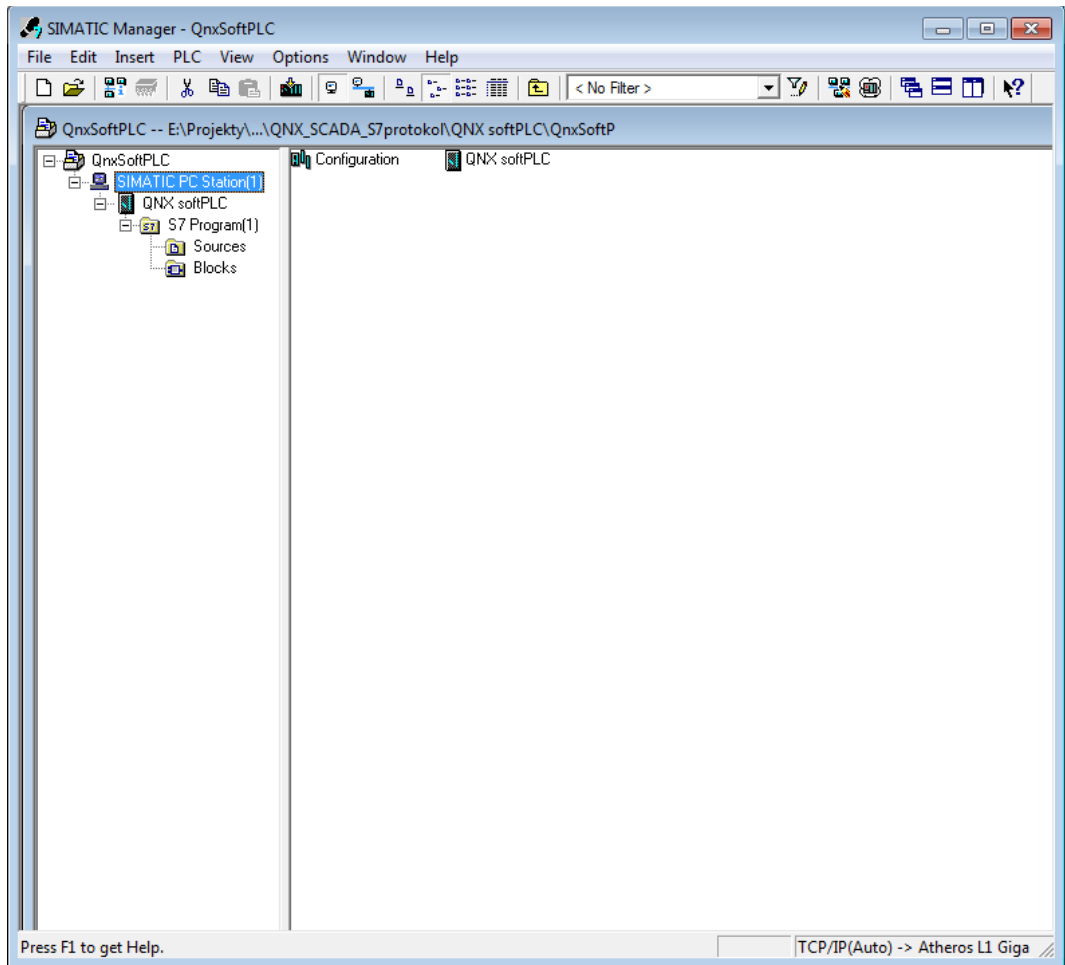
Hardwarová konfigurace softwarového PLC je vytvářena pomocí softwarového nástroje SIMATIC NCM Manager nebo SIMATIC Step7 Manager. Po vytvoření projektu je potřeba vložit objekt PC STATION a následně je možné provést hardwarovou konfiguraci. Před zahájením vytváření hardwarové konfigurace je potřeba doinstalovat GSD soubor, který obsahuje data o softwarovém PLC využívaném v následné konfiguraci.

Instalace GSD souboru: Otevřít **Configuration** → **Options** → **Install GSD File**

V hardwarové konfiguraci PC Station přidáme objekt softwarového PLC pro operační systém QNX, dále nadefinujeme průmyslovou komunikační síť PROFINET, nakonfigurujeme její parametry a přidáme vzdálenou periferie ET200MP IM155-5 a její jednotlivé digitální a analogové vstupně výstupní moduly. Podrobný popis konfigurace komunikace PROFINET a distribuované vzdálené periferie ET 200MP je uveden v kapitole 3.1, která se zabývá obecnou konfigurací objektu PC Station.



Obr. 266 Hardwarová konfigurace projektu QNX softPLC v SIMATIC Step 7 Manageru



Obr. 277 Konfigurace projektu QNX softPLC v SIMATIC Step 7 Manageru

## 5. Vzdálená vizualizace QNX softwarové PLC a distribuované periferie ET200 MP IM155-5 PN ST

### 5.1. Průmyslový operátorský panel s vestavěným PC a QNX 6.5.0 SP1 RTOS

Pro vizualizaci systému byl vybrán dotykový panel PPC-5152-D525. Velikost zobrazovací plochy je 15" a rozlišení 1024x768. Součástí dotykového panelu je vestavěný PC. Tento PC pohání procesor Intel Atom D525 o frekvenci 1.8Ghz. Operační paměť je 1Gb s možností rozšíření až na 4Gb

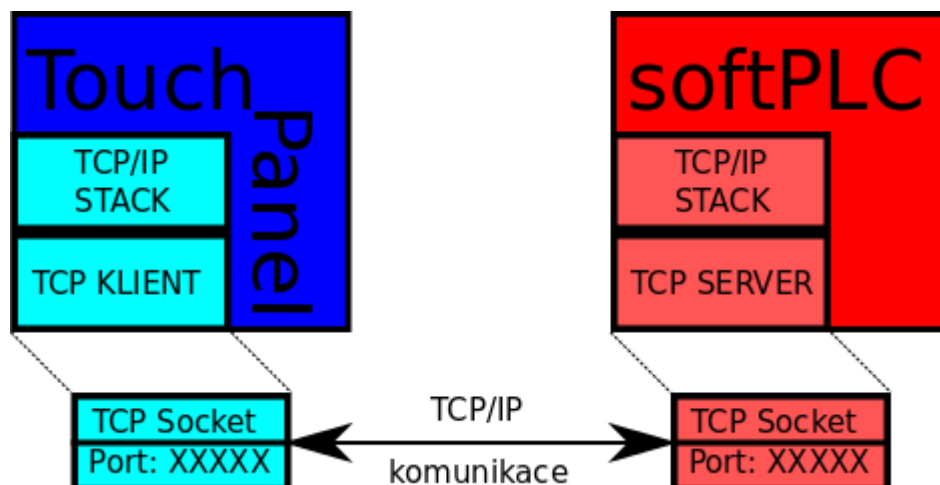


Obr. 28 Průmyslový operátorský panel

K síti lze panel připojit prostřednictvím ethernetu (TCP/IP) nebo bezdrátové sítě WiFi. Operačním systémem vestavěného PC je QNX 6.5.0 SP1 RTOS.

### 5.2. Propojení modulu vizualizace s QNX softwarovým PLC pomocí TCP/IP rozhraní (klient/server)

Komunikace mezi uživatelským dotykovým panelem a softwarovým PLC probíhá pomocí TCP/IP protokolu. V tomto průmyslovém systému představuje dotykový panel tzv. klienta a softwarové PLC server.



Obr. 29 Komunikace klient-server

Softwarové PLC jako server posílá klientovi informace jako ID číslo, ID číslo komunikačního procesoru, data o režimech, chyby a data vstupů/ výstupů vzdálené periferie. Tyto všechny informace lze následně vyčíst z vizualizace na dotykovém panelu.

### 5.3. Grafická knihovna QT pro QNX 6 RTOS

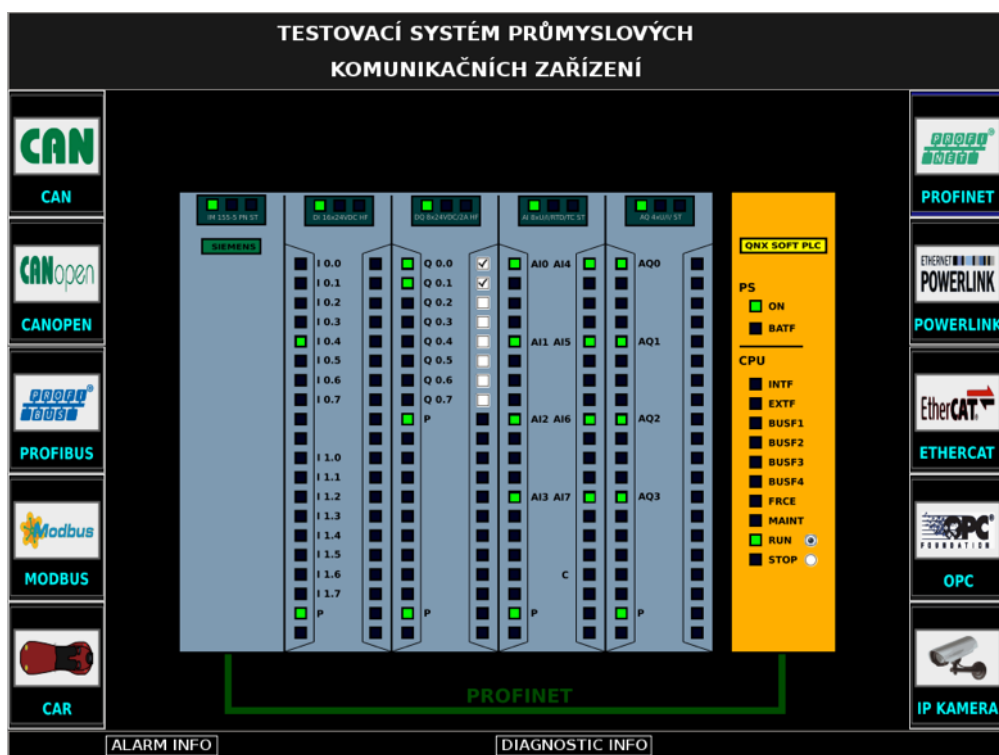
Jedná se o grafickou knihovnu, která je určena, jak pro malé vestavěné systémy, tak pro velké grafické servery na PC. Tato grafická knihovna může být použita pro různé typy platforem (Linux, Windows, Mac OS, iOS, Android, QNX, VxWorks, atd.). Grafická knihovna Qt je dílem firmy Trolltech, později Nokia a dneska firmy Digia. Jedná se o plně objektovou knihovnu napsanou pomocí programovacího jazyka C++. V rámci testovací grafické SCADA aplikace pro průmyslovou sběrnici PROFINET pod operačním systémem QNX 6.5.0 SP1 RTOS byla implementována grafická knihovna pro dvě verze Qt. Jedná se o cross-kompilaci verze Qt 4.8.3 a Qt 5.2.1 pro operační systém reálného času QNX 6 Neutrino.

Grafická knihovna Qt 4 (4.8.3) je určena až po verzi QNX 6.5.0. Pro verzi Qt 5 (5.2.1) však přichází s úplně novým konceptem realizace a napojení na grafický ovladač pod operačním systémem QNX 6. Aby bylo možné grafickou knihovnu Qt 5 spustit pod operačním systémem QNX, je požadována minimální verze QNX 6.5.0 SP1 a vyšší verze společně s rozšiřující softwarovou strukturou „QNX UI Core Graphics Framework“, která je vyvíjena společností QNX ve spolupráci s firmou Digia. Tato rozšiřující knihovna je dostupná na požádání pro zákazníky společnosti QNX.

### 5.4. Vzdálená grafická vizualizace řídicích a akčních hardwarových modulů pomocí knihovny QT

Grafický systém uvedený níže na obrázku zobrazuje fyzicky připojenou distribuovanou periferii SIEMENS ET200MP IM155-5 PN ST, která je přes průmyslovou komunikační sběrnici PROFINET napojena na komunikační kartu Siemens CP1616, která je součástí vestavěného PC s operačním systémem reálného času QNX 6.4.1. Komunikace je realizována přes komunikační server nebo softwarové PLC a dále předávána do Qt SCADA systému. Testovací grafický systém je možné rozšířit i o jiné periferie v rámci daného průmyslového komunikačního protokolu.

Grafické objekty jsou vytvářeny pomocí vektorové grafiky. Předdefinovaná zařízení na průmyslové sběrnici lze měnit za chodu a testovat jejich stav a chování dle žádané interakce. V rámci tohoto modelu byly testovány prvky typu komunikační řídicí modul vzdálené periferie, vzdálené digitální a analogové vstupy a výstupy daných průmyslových zařízení. Grafický systém informuje o vzniklých alarmech v rámci komunikace a také v reálném čase diagnostikuje jednotlivá připojená zařízení a předává zprávu do grafického vizualizačního systému.



Obr. 30 Testovací SCADA systém pro vybrané průmyslové zařízení a komunikační protokoly – PROFINET

## 6. Zhodnocení

Bakalářská práce se zaměřuje na popis návrhu softwarového PLC a jeho napojení na průmyslovou síť PROFINET. Návrhový vzor softwarového PLC implementuje různé typy průmyslových protokolů, které se využívají v průmyslové automatizaci. Jedná se zejména o protokoly využívající jako komunikační médium ethernet. Navrhovaná metodika řešení softwarového PLC je popsána pomocí blokových částí, které tvoří celkový model průmyslového IO kontroleru. Návrhový vzor softwarového PLC je koncipován tak, aby splňoval jak softwarové tak i hardwarové podmínky pro distribuci jednotlivých dílčích modulů v různých částech průmyslových technologií.

Z obecného hlediska, lze výsledný aplikační kód softwarového PLC použít i ve složitých vestavěných zařízeních, obsahujících operační systém QNX 6, který je v průmyslových aplikacích určen pro řízení v reálném čase.

Základní části textu bakalářské práce popisují:

- Operační systém reálného času QNX 6 Neutrino
- Průmyslovou komunikační síť PROFINET
- Hardwarovou konfiguraci jednotlivých komunikačních prvků sítě PROFINET
- Návrhový vzor softwarového PLC a jeho softwarových modulů
- Vzdálený uživatelský SCADA systém pro různé typy platform
- Testovací aplikace pro softwarové PLC
- Testovací aplikace hardwarových a softwarových prvků pro SCADA systémy

Mezi hlavní přínosy bakalářská práce patří:

- Návrhový vzor pro softwarové PLC pro operační systém reálného času QNX 6 Neutrino
- Implementace napojení softwarového PLC na průmyslovou síť PROFINET a průmyslový ethernet
- Implementace propojení distribuované vzdálené periferie ET200MP IM 155-5 ST se softwarovým PLC
- Implementace syntaktické analýzy GSD a GSDML souborů
- Implementace napojení softwarového PLC na daný SCADA systém
- Vestavěný SCADA systém pro monitorování a komunikaci se vzdálenou periferií připojenou přes průmyslovou síť PROFINET

Z prezentovaných výsledků této bakalářské práce je zřejmé, že všechny stanovené cíle, tak jak jsou definovány v zadání této práce, byly splněny.

## 7. Citovaná literatura

1. **Co., QNX Software Systems.** *QNX Neutrino RTOS*. [Dokument PDF] Terence Matthews Crescent : QNX Software Systems Co., 2010.
2. **Pajer, Jaroslav.** *Operační systém QNX*. [Diplomová práce] Praha : České Vysoké Učení Technické, 2003.
3. **Rostislav, Kosek.** AUTOMA. *AUTOMA*. [Online] 4 2005.  
[http://www.odbornecasopisy.cz/index.php?id\\_document=30419](http://www.odbornecasopisy.cz/index.php?id_document=30419).
4. **Eric, Gory.** *Presentation of PROFINET IO and related services*. [Presentation] Lisnagry, Co. Limerick : PROFIBUS International, 2010.
5. **Zezulka, František a Hynčica, Ondřej.** AUTOMA. *AUTOMA*. [Online] 5 2008.  
<http://www.odbornecasopisy.cz/prumyslovy-ethernet-viii:-ethernet-powerlink-profinet-37288.html>.
6. **Dovica, Martin.** *Testování implementace sběrnice PROFINET do systémů Simotion*. [Diplomová práce] Ostrava : VŠB-TU, 2010.
7. **Manfred, Popp.** *Industrial communication with PROFINET*. Germany : PROFIBUS Nutzerorganisation e.V., 2007.
8. **Gory, Eric.** *Presentation of PROFINET IO and related services*. [Prezentace] místo neznámé : PROFIBUS International, 2010.
9. **P., Herout.** *Učebnice jazyka C*. České Budějovice : Nakladatelství Kopp, 1998. ISBN 80-85828-21-9.
10. **S., Prata.** *Mistrovství v C++*. Praha : Computer Press, 2001. ISBN 80-7226-339-0.
11. SIEMENS. *Manuály a dokumentace firmy Siemens*. [Online]  
<http://www.siemens.com>.
12. Digia. *Manuály a dokumentace firmy Digia*. [Online] <http://www.qt-project.org>.
13. PPC-5152-D525. *IEI Integration Corp*. [Online] 2013.  
<http://www.ieiworld.com/index.aspx>.
14. QNX. *Manuály a dokumentace firmy QNX Software Systems*. [Online]  
<http://www.qnx.com>.