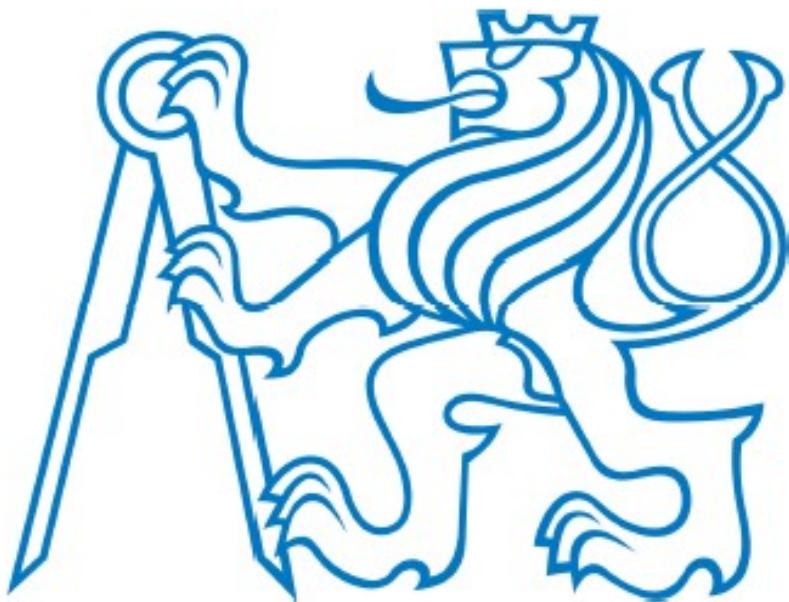


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

Fakulta elektrotechnická

DIPLOMOVÁ PRÁCE



Bc. Václav Sedláček

Embedded zařízení v IT integracích

slaboproudých technologiích

Vedoucí diplomové práce: Ing. Martin Hlinovský, Ph.D

Katedra řídicí techniky

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne.....
Podpis

Poděkování

Rád bych na tomto místě poděkoval svému vedoucímu diplomové práce, Ing. Martinu Hlinovskému, Ph. D. nejen za cenné rady, ale také za poskytnutí volnosti při realizaci vlastního řešení diplomové práce.

Dále chci poděkovat společnosti Integoo s. r. o., a to především Ing. Radimovi Matějovi, který byl nedílnou součástí celé diplomové práce. Poskytl mi cenné rady a prostředky, díky nimž jsem měl možnost celou práci realizovat.

Anotace

Práce pojednává o embedded zařízení v IT integracích slaboproudých technologií. První část je věnována typickým architekturám v IT integracích. Dále je ukázáno, jaký mají navržená Embedded zařízení vliv na změnu architektury a jaké jsou přínosy takového zařízení. Protože embleded zařízení jsou navržena na procesorech firmy Beck, jsou popsány jednotlivé řady dostupných procesorů. Dále je obsažen návrh HW a SW embleded zařízení. SW je navržen tak, aby byl do maximální možné míry modifikovatelný a rozšířitelný. Realizované embleded zařízení jsou synchronizace obsahů monitorů a rozhraní TCP – protokol Spinel. Navrhovaná zařízení komunikují pomocí protokolu TCP na rozhraní Ethernetu, a jednotlivé zprávy jsou předávány ve formátu XML. Jednotlivá zařízení jsou na konci příslušných kapitol zhodnocena.

Klíčová slova: SW, HW, TCP, XML, uml, Ethernet

Annotation

The aim of diploma thesis is the embedded device in IT integration of low voltage technology. First part includes typical architecture of IT integration. Next section explains effects and benefits of developed embedded devices on typical architecture. All processors of IPC@BECK are described in detail as embedded devices were based on them. The thesis also includes the concept of HW and SW of developed devices. SW was designed to allow the user to modify and extend its programming. Two embedded devices were realized, the first is the synchronization of monitor content and the second is TCP – Spinel protocol interface. Ethernet interface of TCP is used for communication with developed devices and particular messages are send in XML format. Conclusions of relevant chapters involve the evaluation of those devices.

Key words: SW,HW,tcp,xml,uml, ethernet

Obsah

PROHLÁŠENÍ.....	I
PODĚKOVÁNÍ.....	II
ANOTACE.....	III
ANNOTATION.....	III
Obsah.....	1
Úvod.....	2
Kapitola 1.....	3
1 DEFINICE OBORU INTEGRACE SLABOPROUDÝCH ZAŘÍZENÍ.....	3
1.1 Popis integrace slaboproudých zařízení v oboru IT.....	3
1.2 Včlenění nového zařízení do dříve popsané integrační architektury.....	8
Kapitola 2.....	11
2 POPIS PROCESORŮ IPC@CHIP FIRMY BECK.....	11
2.1 Procesory řady SC1x.....	11
2.2 Procesory řady SC2x.....	12
2.3 Procesory řady SC1x3.....	16
2.4 Procesory řady SC2x3.....	17
2.5 Popis některých vývojových kitů a dalších doplňků.....	18
2.6 Operační systém RTOS.....	20
2.7 Softwarové nástroje.....	24
Kapitola 3.....	26
3 NÁVRH HW.....	26
3.1 Volba vhodné procesorové řady.....	26
3.2 Komunikační kanály.....	27
3.3 Paměťové medium (SD karta).....	31
3.4 Napájení zařízení.....	32
3.5 Celkové zapojení HW.....	34
Kapitola 4.....	40
4 NÁVRH SW ARCHITEKTURY.....	40
4.1 Požadavky na SW architekturu.....	40
4.2 Popis modulů.....	41
Kapitola 5.....	51
5 SYNCHRONIZACE OBSAHŮ CCTV MONITORŮ.....	51
5.1 Popis řešeného problému.....	51
5.2 Volba použitého HW.....	53
5.3 SW synchronizace obsahu monitorů.....	54
5.4 Zhodnocení synchronizace obsahu monitorů.....	59
Kapitola 6.....	61
6 ROZHRANÍ TCP – PROTOKOL SPINEL.....	61
6.1 Protokol Spinel.....	61
6.2 Začlenění do IT integračních zařízení.....	64
6.3 Návrh SW rozhraní TCP – protokol spinel.....	65
6.4 Zhodnocení současného zařízení.....	71
Literatura a použité zdroje.....	73
Obsah přiloženého CD.....	74
Příloha A.....	75
HW DOKUMENTACE ROZHRANÍ TCP-PROTOKOL SPINEL.....	75

Úvod

Cílem této diplomové práce je popis integračních architektur a návrh dvou zařízení. Prvním navrhovaným zařízením je synchronizace obsahů monitorů. Druhým zařízením je rozhraní TCP-protokol Spinel. Jednotlivé kapitoly popisují postup mé práce.

V první kapitole definuji obor integrace slaboproudých zařízení. Cílem této kapitoly je uvedení do problematiky a nastavení cesty k určení požadavků na takové zařízení. Aby byl splněn cíl, ukážu, jak vypadají typické integrační architektury a jak se změní v případě, že zařadíme navrhované zařízení.

V druhé kapitole se budu věnovat procesorům firmy Beck. Tyto procesory budou použity pro realizaci navrhovaných zařízení. Protože na sobě mají tyto procesory již nainstalovaný operační systém, tak se budu také věnovat jeho popisu. V této kapitole budou také popsány SW vývojové prostředky.

Ve třetí kapitole bude rozebrán navrhovaný HW pro rozhraní TCP-Protokol spinel. Popíšu jednotlivé HW části zařízení a základní principy jejich funkce.

Ve čtvrté kapitole se budu věnovat návrhu SW. Návrh SW bude ukázán na UML analytických diagramech tříd. Ještě před vlastním návrhem si určím požadavky, které vyplynou především z první kapitoly. Jednotlivé SW moduly budu pak používat v konkrétních aplikacích realizovaných v této diplomové práci.

V páté kapitole se budu věnovat realizaci synchronizaci obsahu monitorů CCTV. Abych upřesnil, o co se jedná, tak popíšu konkretní instalaci CCTV systému. Dále se budu věnovat popisu SW tohoto zařízení a popisu SW na PC. Na závěr kapitoly krátce zhodnotím dosažený výsledek.

V poslední kapitole se budu věnovat návrhu zařízení realizující rozhraní TCP – protokol spinel. Kapitola bude obsahovat stručný popis protokolu spinel. Dále navrhnu a implementuji SW, který bude přítomen na navrhovaném zařízení a na PC. Tyto dva SW spolu budou komunikovat.

Kapitola 1

1 Definice oboru integrace slaboproudých zařízení

Tato kapitola pojednává o principech integrace slaboproudých zařízení v oboru IT. Nejdříve se soustředím na popis klasických principů integrace slaboproudých zařízení v oboru IT a shrnutí cílů takové integrace. Dále se pokusím vyjmenovat zařízení, která nejčastěji bývají předmětem integrace a jež se používají ve firmě Integoo s.r.o. Na základě dříve vyjmenovaných poznatků popíšu typickou integrační architekturu. V poslední části kapitoly popíšu, jak začlenit emleded zařízení do IT integrace slaboproudých zařízení.

1.1 Popis integrace slaboproudých zařízení v oboru IT

Samotné slovo integrace má nepřeberné množství významů, proto je nutné zúžit význam tak, aby byl vhodný pro naše potřeby. Slovo integrace (a nebo slovní spojení integrace slaboproudých zařízení) bude mít pro naše potřeby význam spojení několika zařízení dohromady v jeden funkční celek. Můžeme také říci, že takováto integrace je nástavbou nad dalšími systémy. Tato nástavba může například propojovat dva systémy, poskytovat uživatelsky přívětivá data z několika systémů, a nebo případně poskytnou funkcionality, kterou by subsystém nebyl schopen poskytnout.

Téměř každá integrace v oboru IT obsahuje server s komunikačním rozhraním ethernet. Velice často se objevují i další komunikační rozhraní v podobě sériové linky (RS232, RS485/422). Tato rozhraní slouží ke komunikaci s dalšími zařízeními. Ničím neobvyklým nebývá ani použití několika serverů. Několik serverů může být použito z mnoha důvodů, např.: vytvoření redundancy, zajišťování výpočetní náročných operací atd... V některých případech je nutné k serveru připojit diskové pole z důvodu navýšení kapacity, nebo vytvoření RAID diskového pole pro navýšení zabezpečení dat proti poruše. Další velice důležitou součástí integrační nástavby je prezentační vrstva pro uživatele, kterou si lze nejlépe představit jako grafické uživatelské rozhraní. Toto rozhraní bývá nejčastěji realizováno na PC ve výjimečných případech operátorským panelem nebo tlačítka. Toto PC s grafickou uživatelskou nástavbou budu označovat jako workstation. Realizace workstation je značně různorodá, proto se můžeme setkat s PC v provedení desktopu, dotykového panelu atd. Grafická uživatelská nástavba bývá v poslední době často realizována i na mobilních

zařízeních (mobilní telefon, laptop, ...). Poslení částí jsou integrovaná zařízení. Tímto zařízením může být cokoliv, co lze připojit přes výše vyjmenované sběrnice. Dovolím si připomenout, že takovým zařízením může být i server.

Sběrnice sériové linky (RS232, RS 485/422) bývají často realizovány jako převodníky typu ethernet-RS485/422, ethernet-RS232. Důvodem použití těchto převodníků je stále větší snaha výrobců HW serverů tyto sběrnice odstranit ze základních desek serverů. Dalším důvodem také může být ochrana drahého HW serverů, protože porucha převodníku sběrnic na základní desce serveru vede na výměnu celé základní desky a v případě, že není server zálohován dalším serverem, také k výpadku systému. Můžeme se také setkat s převodníky USB - příslušná komunikační sběrnice. Tyto převodníky podle mého názoru nejsou příliš vhodné pro průmyslové provedení, a navíc nelze čekat ochranu HW základní desky, jako u převodníku umístěného na ethernetu.

Pokud se podíváme na spolehlivost dříve uvedených zařízení, tak se ukazuje, že nejvíce problematickými jsou servery, protože výpadek serveru často znamená výpadek celého systému a nebo velké části systému. Tento výpadek nemusí být nutně způsoben poruchou HW (restart z důvodu aktualizace může trvat desítky minut).

Nejčastější součásti SW integrace v IT bývají relační databáze, ovladače sběrnic, ovladače zařízení, watchdogy, integrační servery, aplikace uživatelského rozhraní, konfigurační nástroje, operační systém.

Relační databáze

- nejčastěji používány bývají MySQL a Microsoft SQL server. Funkcí těchto relačních databází je udržování dat uživatelů a konfigurace systému.

Ovladače sběrnic

- nejčastěji se jedná o ovladače převodníků sběrnic. V případě použití převodníků pro sériové linky se starají o vytvoření virtuálního sériového portu a jeho realizaci.

Ovladače zařízení

- starají se o komunikaci a řízení zařízení připojených na dostupných sběrnicích. Velice často používají virtuální sériové porty ovladačů sběrnic. Typicky mívají komunikační rozhraní přizpůsobené pro propojení do integračního serveru (TCP, DLL knihovny)

Watchdogy

- aplikace starající se o hlídání správné funkce jednotlivých aplikací a nebo částí aplikací.

Nejčastěji se sleduje, zda je aplikace spuštěna a jestli nedošlo například k jejímu pádu. Ale používají se i pokročilejší metody hlídání aplikací (velikost alokované paměti). Standardní funkcí watchdogu by mělo být i logování informací o chybách v aplikaci.

Integrační servry

- starají se o správné propojení jednotlivých částí aplikací. Dále se starají o správnou funkci systému. Příkladem může být správa uživatelských účtů, fyzické propojení mezi subsystémy, vytvoření logických vazeb, filtrování událostí v systému, správa uživatelských deníků událostí.

Konfigurační nástroje

- jsou aplikace, které se starají o nastavení systému, případně podsystémů. Např.: nastavení uživatelských práv, nastavení komunikačních parametrů, přidání dalších zařízení do systému, konfigurace uživatelského prostředí.

Aplikace uživatelského rozhraní

- stará se o správnou interpretaci dat pro uživatele v grafické podobě. Aplikace může být realizována jako tenký klient (WEB server) a nebo tlustý klient (lokálně běžící aplikace). Aplikace by měla pro získávání dat používat integrační server.

Operační systém

- často bývají použity pro servery serverové operační systémy firmy Microsoft, a to samé platí o workstation. Důvodem použití těchto operačních systémů je – mimojiné – nejširší podpora ze strany výrobců integrovaných subsystémů.

1.1.1 Typická zařízení používaná v IT integracích

Tato část se snaží vyjmenovat zařízení, která jsou často integrována v oblasti Building Managementu. Účelem níže uvedeného seznamu je poskytnout základní přehled o zařízeních a jejich různorodosti. Ke každému typu zařízení je nutné implementovat ovladač zařízení tak, aby toto zařízení mohlo být použito v rámci integračního serveru. Nad těmito ovladači je pak vytvořena vrstva implementující logiku práce se zařízeními.

Binární vstupy/výstupy

Binární vstupy a výstupy jsou nejzákladnějšími zařízeními pro integrace. Poskytují jednoduchý způsob získávání a předávání informací okolí. Používají se např.: spínání světel jako vazby na řídicí systém, hlídaní stavu závor atd... Často se připojují přes převodníky sběrnice, ale

existují i modely s rozhraním ethernet. Používány jsou například modely řady Qido firmy Papouch s.r.o.

Analogové vstupy

Analogové vstupy se používají k měření analogových veličin. Používají se například v měření teploty a vlhkosti. Analogové vstupy obsahují A/D převodník a další logiku pro zpracování signálu. Opět často dochází k připojení přes převodník sběrnice a nebo po ethernetu.

Frekvenční vstupy/výstupy (PWM)

K frekvenčním vstupům se připojují čidla s frekvenčním výstupem, např. měření průtoku. Frekvenční výstupy se nejčastěji používají k ovládání akčních členů se vstupem PWM. Akčním členem může být např. světlo a frekvenčním výstupem s možností řízení střídy signálu. Touto střídou můžeme regulovat osvětlení.

CCTV systémy

Uzavřený televizní okruh (Closed-Circuit television) je důležitým bezpečnostním prvkem. Z pohledu integrací se na CCTV lze dívat jako na samostatný subsystém. Do integračního serveru a aplikací uživatelského rozhraní se přidávají pomocí SDK (Software Development Kit) k systémům CCTV. Případně pomocí speciálních video grabovacích karet. Pro oblast IT integrací jsou zvláště vhodné IP CCTV systémy. Prvním vstupním bodem do CCTV kamerových systému jsou servery obsahující software, který umožňuje vzdálené připojení (realizováno po síti Ethernet). V některých případech dochází k integraci samostatných kamer. Typickou úlohou integrace bezpečnostních technologií je zobrazení živého obrazu z kamery, která se nachází v oblasti vzniku události ze systémů EZS a EPZ. Moderní kamerové systémy se stávají sami zdrojem událostí, protože stále častěji obsahují možnosti vhodné video detekce.

EZS a EPS

EZS (elektrický zabezpečovací systém) slouží k zabezpečení objektů proti neoprávněnému vstupu. Klíčovým prvkem pro integraci EZS je vhodná zabezpečovací ústředna. Takováto vhodná zabezpečovací ústředna obsahuje komunikační rozhraní ethernet nebo sériovou linku. Po těchto komunikačních kanálech dochází k přenosu informace k integračnímu serveru. Integrační server musí implementovat příslušný protokol pro komunikaci s ústřednou. Toto propojení umožňuje plně ovládat systém EZS a získávat z něj potřebná data.

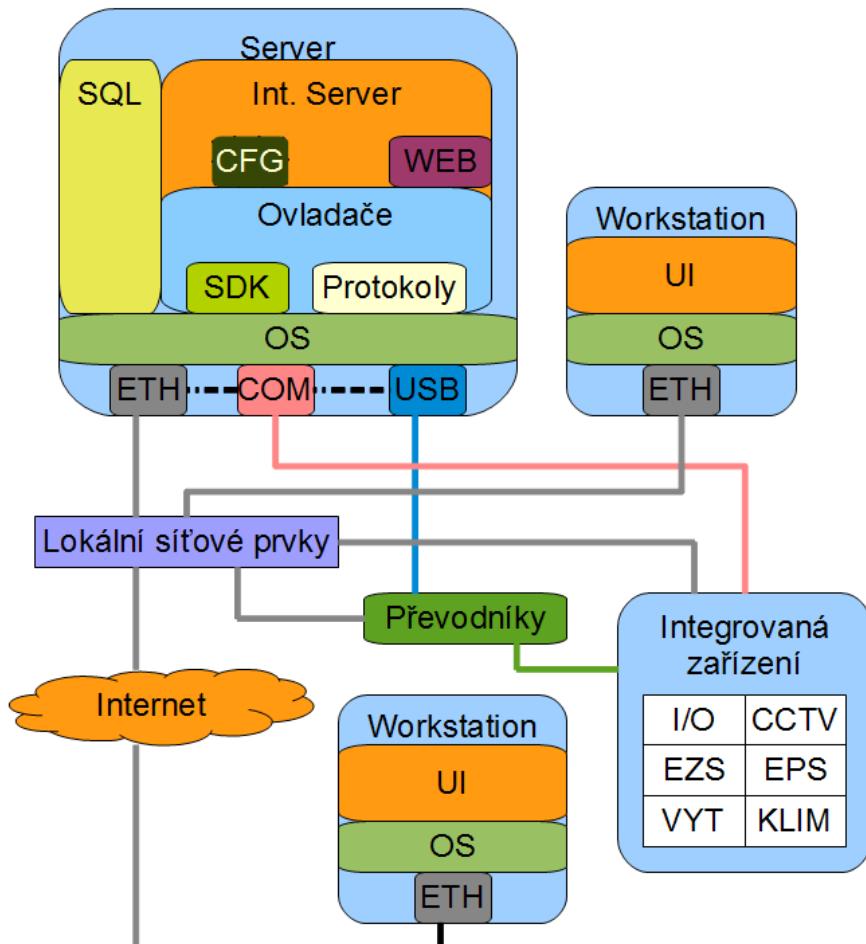
Podobně jako u EZS je to u EPS (Elektrická požární signalizace).

Vytápěcí a klimatizační jednotky

Integrace v oboru building managementu integruje také systémy vytápění a klimatizace. Tyto systémy se často ovládají z binárních, analogových a frekvenčních vstupů/výstupů, ale existují systémy, které mají pokročilejší interface se sériovým a nebo ethernetovým rozhraním. Tato rozhraní se používají k ovládání a sběru dat. Pomocí integrace lze sestavit uživatelsky přívětivé rozhraní a nebo lze technologii ovládat centrálně. Integrace by neměla řešit vlastní řízení těchto systémů, ale měla by fungovat jako nástavba.

1.1.2 Blokové schéma integrační architektury

Z výše uvedených úvah je možné sestavit blokové schéma typické integrační architektury. Základní schéma je na obrázku 1. Na obrázku 1 jsem použil několik zkratek. CFG značí konfigurační nástroje, WEB označuje WEB server, ETH rozhraní Ethernetu, COM rozhraní sériové linky, UI uživatelsky přívětivé rozhraní, OS operační systém, I/O binární, analogové a frekvenční vstupy a výstupy, VYT systém vytápění, KLIM systém klimatizace.



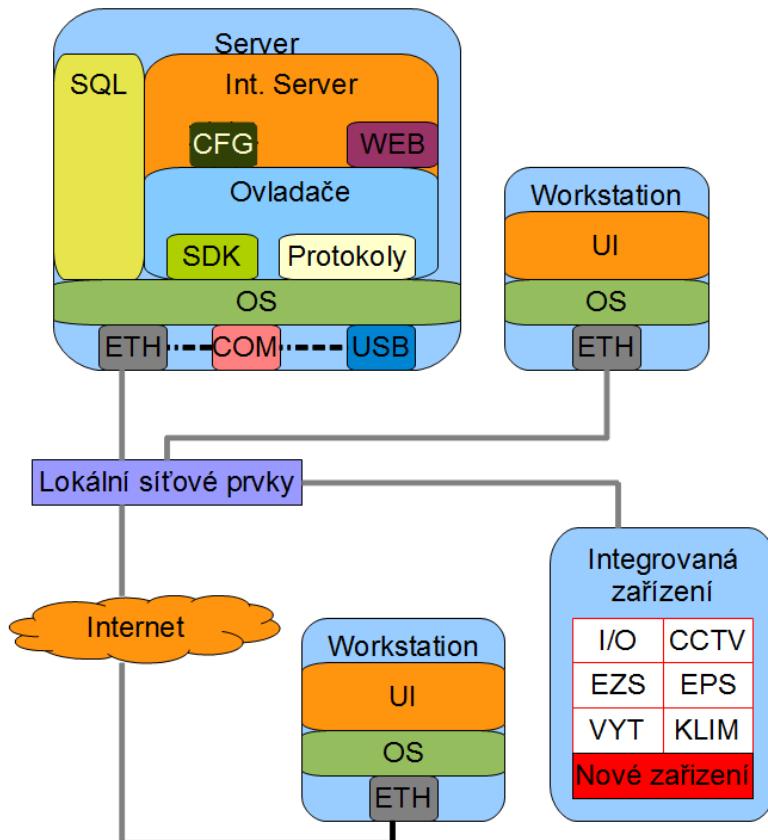
Obrázek 1 Blokové schéma integrační architektury

V blokovém schématu jsou uvedeny sítové prvky, o nichž nebylo dříve diskutováno. Tyto sítové prvky musí být poskládány a nastaveny tak, aby bylo možné propojit všechny součásti, které jsou určeny pro integraci a jsou připojeny přes rozhraní Ethernet. Na síťové prvky mohou být kladený speciální požadavky, jako např. přenos paketů multicastem. Multicast bývá využíván systémy CCTV.

Server označuje HW počítače používaného jako server, workstation označuje HW počítače (případně tabletu, operátorského panelu...) určeného pro běžného uživatele. Do systému může být připojena workstation prostřednictvím internetu.

1.2 Včlenění nového zařízení do dříve popsané integrační architektury

V předchozích částí jsem se snažil popsat problematiku integrace slaboproudých zařízení. Tento popis nezdůrazňoval význam a uplatnění nového zařízení, které je také předmětem této práce. Na toto zařízení se lze dívat jednak jako na zařízení, které má být integrováno a nebo jako na zařízení, které má v integraci pomoci. Klíčovou částí takového zařízení by mělo být rozhraní Ethernet. Význam Ethernetu je vidět z obrázku 1, protože je schopen propojit všechny součásti integrace.

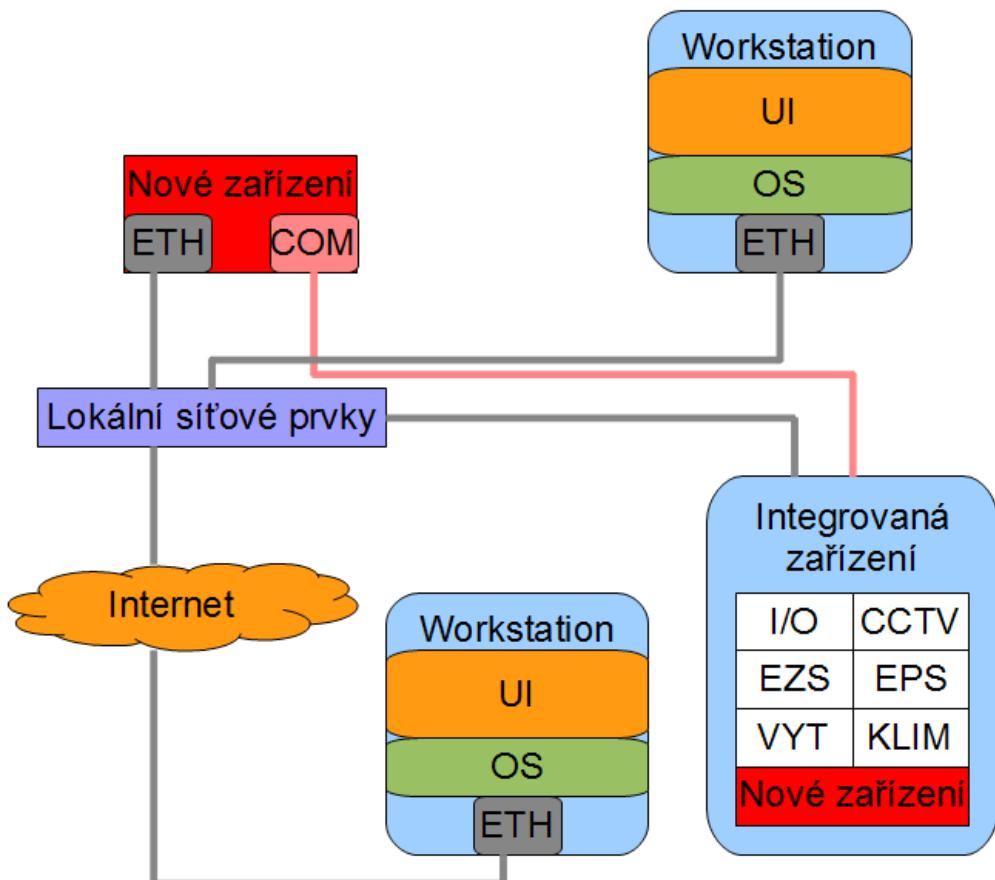


Obrázek 2 Nové zařízení jako integrované zařízení

Blokové schéma na obrázku 2 zobrazuje, jak se změní situace, pokud zařízení začleníme do naší integrace. Protože toto zařízení bude již obsahovat sériové linky, je možné odstranit části, které se týkají převodníku sběrnic, a je tak možno přistupovat k dalším zařízením pomocí tohoto zařízení. Toto zařízení pak musí implementovat nad Ethernetem aplikační vrstvu ISO/OSI modelu, která zaručí dostatečný přístup k datům z dalších systémů a jejich povelování. Z obrázku je vidět, že toto nové zařízení může také být samostatným subsystémem s nějakou speciální funkcí.

Uspořádání z obrázku 2 přináší několik výhod proti uspořádání z obrázku 1:

1. Odstranění převodníků sběrnic
2. Vytvoření jednotného HW a SW rozhraní pro integrační server s přístupem přes rozhraní ethernet
3. V případě výpadku integračního serveru možnost udržet nezbytné funkce zařízení, která by jinak nemohla bez integračního serveru fungovat. V případě implementace takové funkce můžeme mluvit o částečné redundanci
4. Přidaní dalších funkcí nového zařízení



Obrázek 3 Použití nového zařízení jako integračního serveru

Obrázek 3 ukazuje další způsob, jak se může nové zařízení uplatnit v integraci. Toto nové zařízení je použito jako integrační server. Při reálném použití tohoto zařízení v architektuře uvedené na obrázku 3 musíme počítat s nižšími schopnostmi oproti klasickému serveru, co se týká výpočetního výkonu a paměťové kapacity. V této architektuře nové zařízení je možno použít i v integrovanych zařízeních.

Výhody oproti architektuře z obrázku 1 a 2 jsou:

1. Levnější řešení bez HW a SW serveru
2. Použité zařízení bude mít velmi rychlý start – cca 2s
3. Stabilnější HW bez točivých součástí (disky, větráky)
4. Menší spotřeba energie
5. Realtime operační systém

Nevýhody oproti architektuře z obrázku 1 a 2 jsou:

1. Méně výpočetního výkonu a paměťové kapacity
2. Nelze začlenit SDK, která jsou stavěna pro použití v určitém operačním systému a využívají API toho operačního systému
3. Neobsahuje některé služby a aplikace, které jsou v serverových operačních systémech standardně přítomny
4. Přítomnost pouze jednoho ethernetového rozhraní o rychlosti 100Mbit/s
5. Z bodů 1-4 plyně použití pouze v jednodušších instalacích

Kapitola 2

2 Popis procesorů IPC@CHIP firmy BECK

Tato kapitola popisuje všechny dostupné řady procesorů, které lze koupit u firmy Beck IPC GmbH. Budu se snažit popsat jejich parametry a u každé řady odvodit z těchto parametrů předpokládané použití. Dále se budu věnovat popisu použitých vývojových kitů a některých doplňků. Všechny procesory se dodávají s operačním systémem RTOS. Jeho popisu se budu věnovat v předposlední části této kapitoly. Kapitolu zakončím popisem použitého SW, který je určen pro práci s procesory a jejich programováním.

Na tomto místě bych chtěl upozornit na menší nesrovnalost v použití názvu procesor. Pokud nebude uvedeno jinak, tak v této práci budu označovat slovem procesor jednotlivé členy řady embedded controleru firmy BECK.

2.1 Procesory řady SC1x

Jedná se o výkonově nejslabší řadu procesorů od firmy BECK. Řada obsahuje dva procesory SC11 a SC13. O výpočty se stará mikročip SC186 na frekvenci 40 MHz. Společné vlastnosti procesorů lze vidět v tabulce 1. Obrázek procesu lze vidět na obrázku 4.

Vlastnost	Hodnota
Mikročip	SC186/40 MHz
RAM	512 KB
Flash paměť	512 KB
Seriová linka	2xTTL
I/O	14 PIO, Intel AD-Bus
Spotřeba	Typ. 0,6W/Max. 0,8W
Operační systém	RTOS
Pouzdro	DIL32
DMA	Ano
I2C	Ano
Watchdog	Ano

Tabulka 1 Společné vlastnosti řady SC1x

Procesor SC13 navíc obsahuje Ethernet 10/100BaseT s fyzickou vrstvou.

Dříve řada SC1x obsahovala procesor SC12, ale ten se dnes již nedodává samostatně, ale lze ho najít v modulu DK40 firmy Beck.



Obrázek 4 Procesor řady SC1x (převzato z www.beck-ipc.com)

Procesor řady SC1x se svým výkonem hodí k jednoduššímu řízení, kde není potřeba vysoký výkon. Dokázal bych si ho představit v zařízení s řízením binárních I/O a ovládání přes web browser a nebo pomocí CGI povelů, případně jako jednotné rozhraní pro přístup k sériové lince pomocí Ethernetu.

2.2 Procesory řady SC2x

Jedná se o výkonově střední třídu procesorových řad firmy BECK. Řada obsahuje dva procesory SC23 a SC24. O výpočty se stará mikročip SC186EX/96MHz. Společné vlastnosti procesorů SC2x lze vidět v tabulce 2. Procesor řady SC2x lze vidět na obrázku 5.

Vlastnost	Hodnota
Procesor	SC186EX/96MHz
RAM	8 MB SDRAM
FLASH	2 MB (SC23)/ 8MB (SC24)
Seriové linky	3xTTL
Ethernet	1x10/100BaseT s fyzickou vrstvou
USB	USB1.1 (Host i device)
CAN	2xCAN2.0b
I/O	17 PIO
Spotřeba	Typ.1W / max 1,3W
Operační systém	RTOS
Pouzdro	DIL32
Hardware SPI	Ano
Hardware I2C	Ano
DMA	Ano
Watchdog	Ano

Tabulka 2 Společné vlastnosti řady SC2x

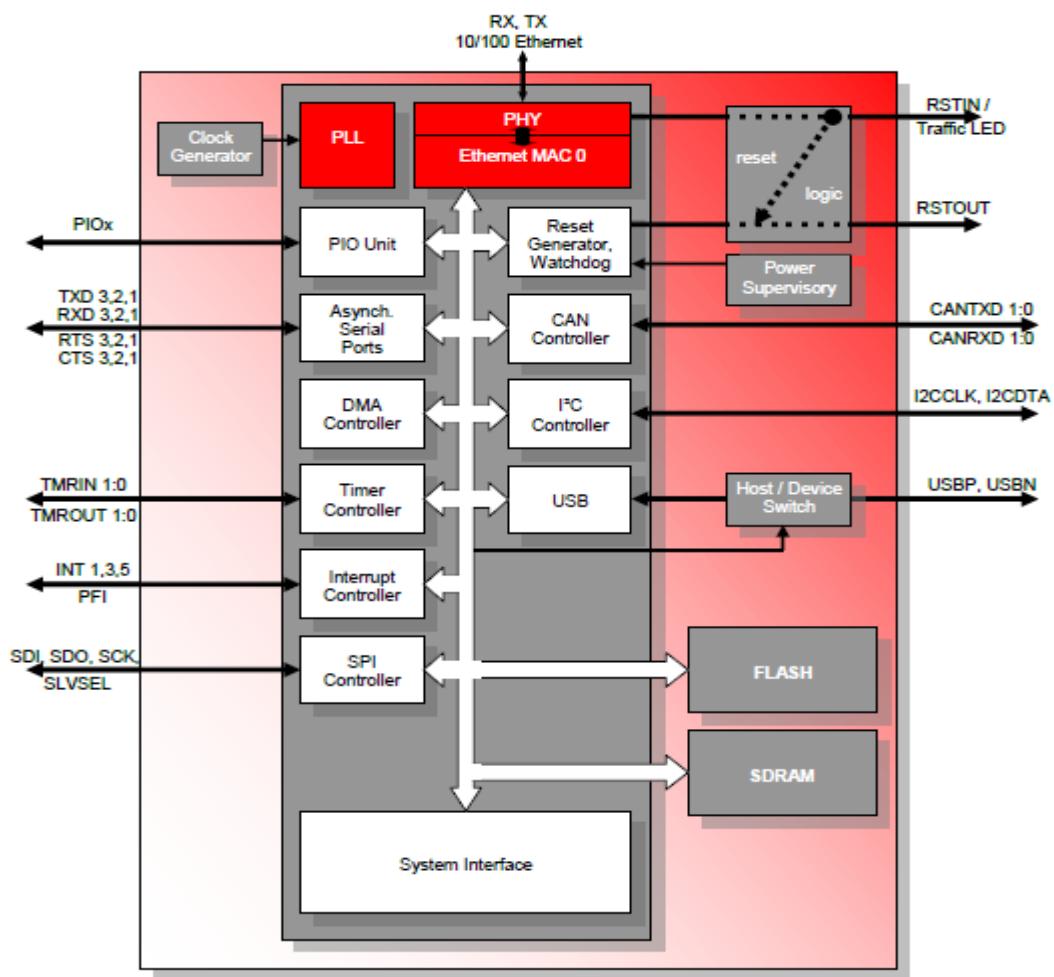
Rozdíl mezi SC23 a v SC24 je pouze ve velikosti flash paměti, viz 3. vlastnost v tabulce 2.

Protože procesor bude použit v novém zařízení, tak se jím budu zabývat do větších podrobností. Blokové schéma na obrázku 6 znázorňuje přehledně jednotlivé funkční bloky

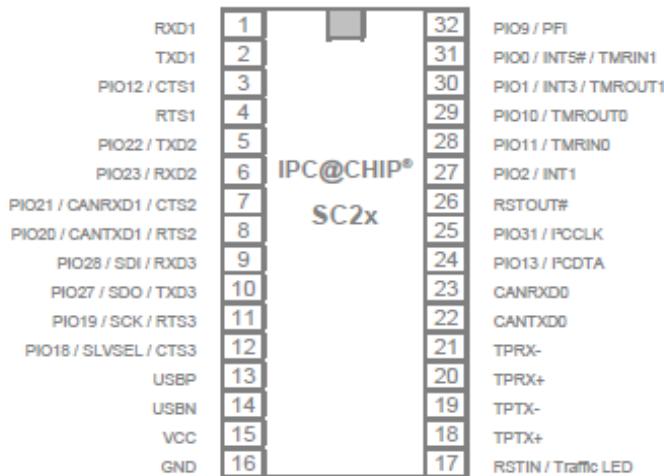
procesoru. Na obrázku 7 je zapojení pinů SC23 v pouzdro DIL32. Z obrázku 7 je vidět, že nevšechny vlastnosti vyjmenované v tabulce 2 jsou dostupné najednou. Někdy se musíme rozhodnout, co bude daný pin realizovat. Například pohledem na pin 11 v obrázku 7 se musíme rozhodnout, zda použijeme sběrnici UART3, SPI a nebo pin použijeme jako IO PIN.



Obrázek 5 Procesor řady SC2x (převzato z www.beck-ipc.com)



Obrázek 6 Blokové schéma SC2x (převzato z www.beck-ipc.com)

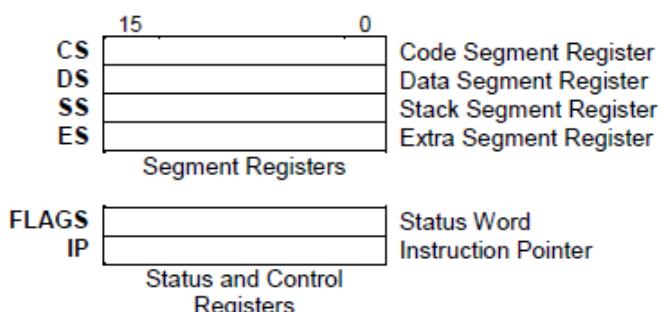


Obrázek 7 Zapojení pinů SC2x (převzato z www.beck-ipc.com)

Jádro procesoru implementuje 8086 mikroprocesor a přidává 10 instrukcí procesoru 80186. Toto uspořádání zajišťuje kompatibilitu s rodinami mikroprocesorů 8086 a 80186. Na obrázku 8 je vidět náhled na registry obecného použití. Na obrázku 9 jsou vidět registry segmentu, stavu a řídící registry.

16-Bit Register Name	8-Bit Register Name	Special Register Functions
	15 8 7 0	
AX	AH AL	Multiply/Divide
DX	DH DL	I/O Instructions
CX	CH CL	Loop/Shift/Repeat/Count
BX	BH BL	Base Registers
BP		
SI		Index Registers
DI		
SP		Stack Pointer

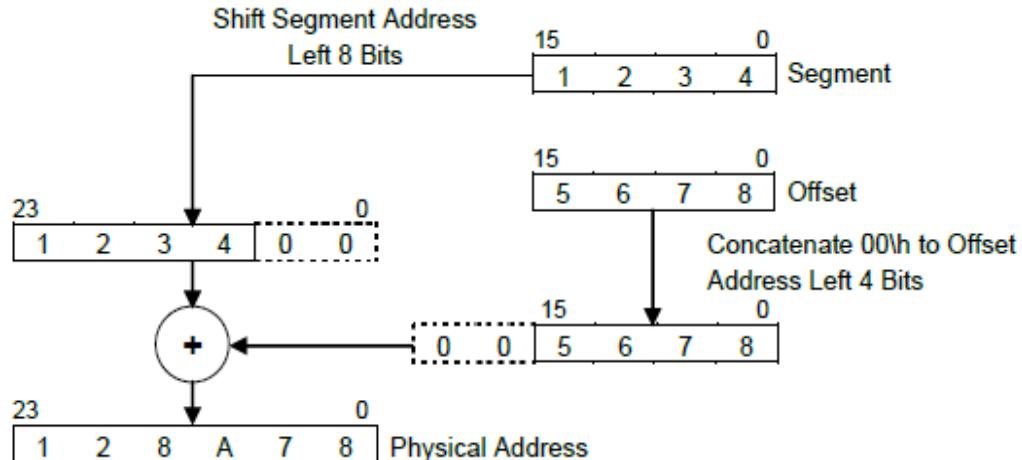
Obrázek 8 Registry obecného použití SC2x (převzato z www.beck-ipc.com)



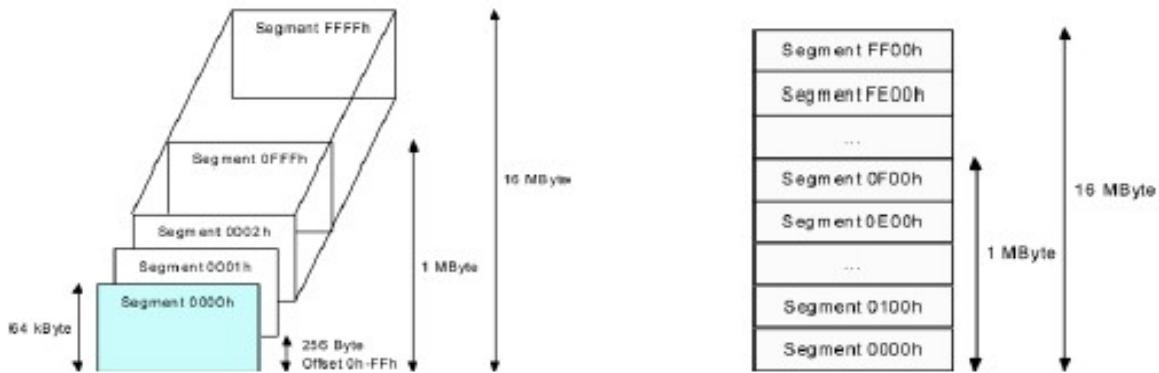
Obrázek 9 Segmentové, stavové a řídící registry SC2x (převzato z www.beck-ipc.com)

Procesor může adresovat 16MB fyzické paměti. Paměť je rozdělena do řad jednotlivých segmentů. Každý segment je tvořen za sebou jdoucí řadou bytů o maximálním počtu 64 tisíc (64Kx8bit). Pro adresaci je nutné použít dva 16-bitové ukazatele. Na obrázku 10 lze vidět, jak se provádí vygenerování 24-bitové adresy. Na obrázku 11 můžeme vidět, jak vypadá rozložení

paměťových segmentů v paměti. Z obrázku je vidět, že každý segment překrývá 256 bytů segmentu následujícího.



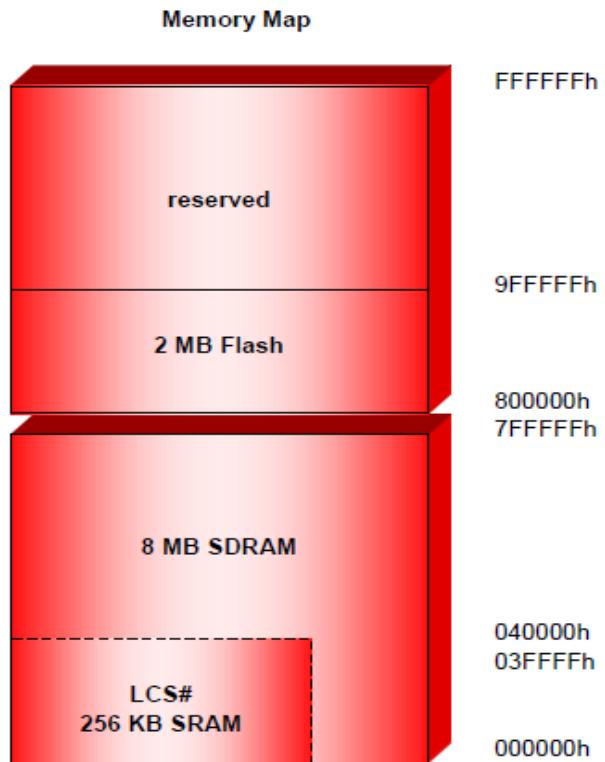
Obrázek 10 Vygenerování 24-bitové adresy (převzato z www.beck-ipc.com)



Obrázek 11 Rozložení segmentů v 24-bitové adresaci (převzato z www.beck-ipc.com)

Na obrázku 12 můžeme vidět mapu paměti SC23. Z této mapy je vidět, na kterých adresách leží jednotlivé periferie paměti.

Řada procesorů SC2x disponuje dostatečným výkonem pro všechny dostupné komunikace umístěné na procesoru. Pokud se podíváme na poměr cena/výkon, pak je tento procesor nevhodnější pro naše účely, protože disponuje dostatečným výkonem, pamětí RAM a Flash. Dokážu si ho představit i jako náhradu integračního serveru, který zprostředkovává uživatelský interface do webbrowseru na workstation a využívá javaaplety pro tvorbu příjemného uživatelského rozhraní.



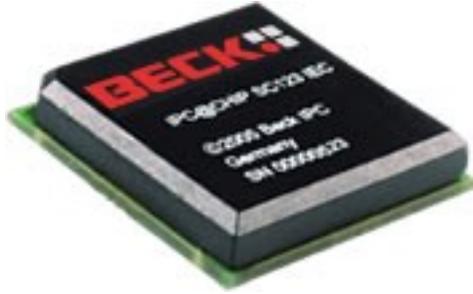
Obrázek 12 Mapa paměti vzhledem k paměťovým periferiím (převzato z www.beck-ipc.com)

2.3 Procesory řady SC1x3

Jedná se o výkonově střední třídu procesorových řad firmy BECK. Řada obsahuje dva procesory SC123 a SC143. O výpočty se stará mikročip SC186EX/96MHz. Hlavní rozdíl mezi řadami SC2x a SC1x3 spočívá v použití jiného pouzdra. Společné vlastnosti procesorů SC1x3 lze vidět v tabulce 3. Procesor řady SC1x3 lze vidět na obrázku 13.

Vlastnost	Hodnota
Procesor	SC186/96 MHz
RAM	8 MB
Flash	2 MB(SC123) / 8MB(SC143)
Servié linky	4xTTL
Ethernet	1x10/100BaseT, 1xMII(pro externí PHY)
USB	USB1.1
CAN	2 x CAN2.0b
I/O	31 PIO, Intel AD-Bus
Spotřeba	Typ. 1W / Max. 2W
Operační systém	RTOS
Pouzdro	BGA177
SPI	Ano
I2C	Ano
16-bit adresová sběrnice	Ano

Tabulka 3 Vlastnosti procesorové řady SC1x3



Obrázek 13 Procesor řady SC1x3 (převzato z www.beck-ipc.com)

Procesory řady SC1x2 by jsem použitím zařadil podobně jako procesory SC2x. Výhodou těchto procesorů je vyvedení více pinů z pouzdra, nevýhodou pak vyšší nároky na montáž procesoru řady SC1x2.

2.4 Procesory řady SC2x3

Procesory řady SC2x3 jsou nejvyšší třídou procesorů dostupných od firmy BECK. Disponují vysokým výpočetním a paměťovým výkonem. Tato řada má v současnosti jediného zástupce - SC243. O výpočty se stará mikročip Freescale MPC5200/400MHz. Výkon tohoto procesoru je 760 MIPS. Na obrázku 14 lze vidět procesor SC243 a v tabulce 4 jednotlivé vlastnosti procesoru.

Vlastnost	Hodnota
Procesor	MPC5200/400MHz
RAM	64 MB
Flash	32 MB
Servíré linky	6xTTL
Ethernet	1x10/100BaseT
USB	USB1.1
CAN	2x CAN2.0b
I/O	41 PIO
Opearacni systém	RTOS for PowerPC
Pouzdro	Modul o velikosti 60x56 mm, 240 pinů
SPI	Ano
I2C	Ano
LocalPluBus	Ano

Tabulka 4 Vlastnosti procesoru SC243

Procesory řady SC2x3 se hodí pro zobrazení grafických prvků ve spojení s VGA modulem, pro náročnější řízení, pro řízení několika motorů, velmi rychlou komunikaci atd...



Obrázek 14 Procesor SC234 (převzato z www.beck-ipc.com)

2.5 Popis některých vývojových kitů a dalších doplňků

V této části kapitoly popíšu vývojový kit DB54, modul DK40, GM01, WL03.

2.5.1 Vývojový Kit DB54

Vývojový kit DB54 je používán pro SW vývojové účely. Byť provedení kitu je velice kvalitní, tak do zákaznické instalace ho nelze zařadit kvůli jeho vysoké ceně a rozměrově většímu provedení. Na obrázku 15 je vidět vzhled vývojového kitu.



Obrázek 15 Vývojový kit DB54 (převzato z www.beck-ipc.com)

Tento vývojový kit je osazen procesorem SC23 a obsahuje tyto periferie pro testovaní: 1x10/100BaseT, 3xRS-232, 2x CAN, USB (host i device), SD kartu, Textový display 2x16 znaků, SPI, I2C s PCF8574 (I/O expander) pole pinů pro další rozšíření.

2.5.2 Modul DK40

Modul DK40 je určen pro vývojové účely a pro montáže k zákazníkovi. Tento modul je osazen procesorem SC12. Obsahuje tyto periferie: 8 I/O, RS232, 10BaseT Ethernet, 1x volně dostupná LED a napěťový měnič +15V-30V DC na 5V DC. Tento modul bude použit pro realizaci externí synchronizace monitorů.

Procesor SC12 je pinově kompatibilní s řadou SC1x. Parametry procesoru SC12 jsou

v tabulce 5. Na obrázku 16 je vidět modul DK40.

Vlastnost	Hodnota
Procesor	186/20 MHz
RAM	512 KB
Flash	512 KB
Seriové linky	2xTTL
Ethernet	10BaseT
I/O	14
Pouzdro	DIL32

Tabulka 5 Parametry procesoru SC12



Obrázek 16 Modul DK40 (převzato z www.beck-ipc.com)

2.5.3 GM01

Jedná se o GSM/GPRS modem. Pro tento modul je již připravena funkcionalita v RTOS.

V případě potřeby můžeme využít TCP/IP vnitřní TCP/IP stack. V tabulce 6 můžeme najít parametry tohoto modulu. Modem se dá ovládat i pomocí AT příkazů. Modem je na obrázku 17.

Vlastnost	Hodnota
GSM rozsahy	850/900/1800/1900 Mhz
GPRS Class	GPRS Class 10
Napájení	3,3V – 4,2V
Komunikační rozhraní	Seriové, USB
Komunikační rychlosť	300 – 460,8kbit/s
Spotřeba	Menší jak 2A

Tabulka 6 Parametry modulu GM01



Obrázek 17 Modem GM01 (převzato z www.beck-ipc.com)

2.5.4 WL03

Jedná se o bezdrátový LAN modul s podporou standardů 802.11a/b/g/n. Podporovaná šifrování jsou WEP64/128, WPA64/128, WPA-PSK, WPA2-PSK. Modul lze používat v módu infrastructure a ad-hoc. Pro tento modul existuje podpora pro RTOS. S procesorem se senzor propojuje po sběrnici SPI. Pro zvýšení dosahu Wifi signálu je možné připojit až dvě antény. WLAN modul je vidět na obrázku 18.



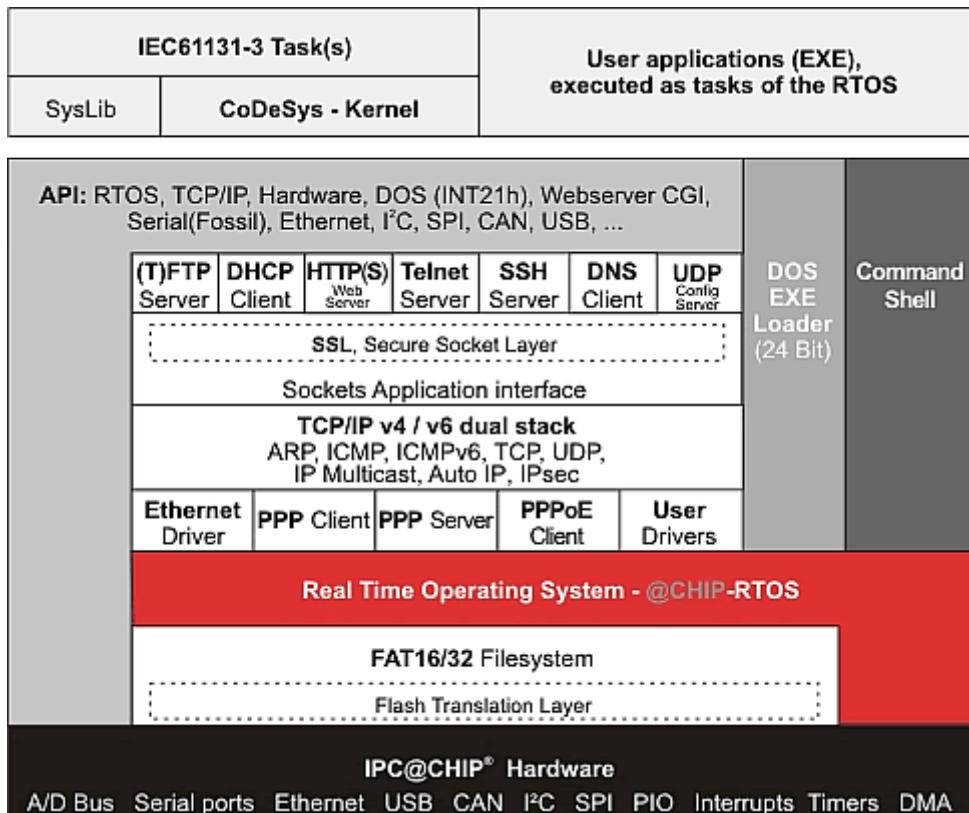
Obrázek 18 WLAN modul WL03 (převzato z www.beck-ipc.com)

2.6 Operační systém RTOS

Operační systém RTOS (Real time operation system) je již předinstalován na všech dostupných procesorech od firmy Beck. V prosinci roku 2010 existoval ve verzi 1.40. V současné době existují 2 mutace RTOS. Jedna mutace pro mikročipy kompatibilní s instrukcemi 80186. Tato mutace je označována jako RTOS. Druhá mutace je kompatibilní s procesorem MPC5200 a je označena RTOS - PPC (real-time operation systém for PowerPC). Protože aplikace v této práci budou používat pouze RTOS, tak popis RTOS-PPC bude vynechán.

Jednotlivé součásti operačního systému jsou vidět na obrázku 19. Z obrázku lze vysledovat strukturu RTOS, a jak jsou jednotlivé součásti na sobě závislé. RTOS obsahuje souborový FAT16/32. Souborový systém je oddělen od aplikací, a proto dovoluje například současný zápis do

logu, přenos souboru po ftp, přenos javaapletu do webové stránky. RTOS je operační systém podporující preemptivní multitasking. V jednu dobu může běžet několik aplikací, které mají na sobě několik tasků.



Obrázek 19 Součásti a organizace RTOS (převzato z www.beck-ipc.com)

Novou aplikaci do operačního systému můžeme dostat po sériové lince a v lepším případě pomocí ftp serveru. Základní ovládání OS se provádí v příkazové řádce (command Shell). V RTOS je možné komunikovat s command shellem pomocí sériové linky a nebo pomocí telnetu. V RTOS je předinstalován web server. Přehled všech vlastností ukazuje tabulka 7, přesný přehled pro aktuální verzi lze najít na stránce <http://www.beck-ipc.com/files/api/scxxx/software.htm>. RTOS se distribuuje v 6 variantách pro procesory řady SC1x. Jednotlivé varianty se liší v dostupnosti jednotlivých aplikací obsažených RTOS. Verze Tiny obsahuje nejméně dodatečné funkcionality, ale spotřebuje nejméně výpočetních a paměťových prostředků. Pro SC2x se dodává jedna verze označená jako FULL.

<i>Vlastnost OS</i>	<i>Vlastnost</i>	<i>Hodnota</i>
<i>multitasking</i>	Tasks(úlohy)	35 (SC1x)/78
	Semafora	
	Časovače	78(SC1x)/128
	Events groups	
<i>Souborový systém</i>	Mezi procesní komunikace	10(SC1X)/32 (Message exchange)
	Vnitřní flash disk	
	Vnitřní ram disk	
	Externí paměťové medium	Compact Flash, MMC, SD (1(SC1X)/2)
	Dlouhé názvy	SC2x a SC1x2
	FAT16	
	FAT32	SC2x a SC1x2
	Max. Otevřených souborů	20(SC1X)/40
<i>TCP/IP</i>	Ipv4/v6	
	TCP,UDP,ARP,ICMP,IP multicast,ICMPv6	SC2x a SC1x2
	Sokety	64(SC1X)/26
	PPP server přes RS232	
<i>TCP služby a aplikace</i>	SSL, IPSec	
	HTTP web server, FTP server, Telnet server, SSH	
	DHCP client	
	SNMP	
	UDP config server	
	Paralelní běh aplikací	Max 12 aplikací paralelně
<i>Spouštěč aplikací</i>	Příkazový řádek ve stylu MS-DOS	Přes Telnet, sériovou linku a USB
<i>API</i>	Funkce RTOS	
	TCP/IP soket rozhraní	
	Sériová komunikace s DMA	
	Dynamické stránky realizované CGI rozhraním	
	HAL (Hardware abstract layer)	
	Ethernetový driver	

Tabulka 7 Funkce RTOS

	Tiny	Small	Medium	Medium PPP	Large	Large PPP	Full
RTOS kernel	x	x	x	x	x	x	x
Serial port fossil driver	x	x	x	x	x	x	x
RTOS filesystem	x	x	x	x	x	x	x
Ext Disk				x	x	x	x
XMODEM filetransfer	x	x	x	x	x	x	x
TCP/IP Ethernet driver		x	x	x	x	x	x
TCP/IP PPP Client/server				x		x	x
Ethernet packet interface	x	x	x	x	x	x	x
TCP/IP protocol stack		x	x	x	x	x	x
DHCP client		x	x	x	x	x	x
I2C API	x	x	x	x	x	x	x
SPI API	x	x	x	x	x	x	x
Hardware API	x	x	x	x	x	x	x
CFG server		x	x	x	x	x	x
Webserver					x	x	x
SSL Webserver							x
TFTP server							x
FTP server			x	x	x	x	x
Telnet server			x	x	x	x	x
SSH server							x
SNMP MIB support	x	x	x	x	x	x	x
USB API							x
CAN API							x
IPsec API							x

Tabulka 8 Přehled funkcí pro jednotlivé varianty RTOS (převzato z www.beck-ipc.com)

2.6.1 Úlohy v RTOS

Úlohy jsou v RTOS označovány jako tasky, toto označení převezmu. Tasky jsou dvojího typu. Prvním typem jsou tasky a druhým typem DOS tasky, které vlastní a spouští tasky. Každý task má svůj vlastní kontext obsahující ukazatel na instrukci a zásobník. Každý task musí mít přiřazenu svoji prioritu vykonávání. Dostupný rozsah priorit je od 2 do 127, kde nejvyšší priorita je 2. Každý task se může dostat do 3 hlavních stavů: aktivní, blokovaný a odstavený. Všechny tyto stavů mají několik podstavů, jejichž přesnější popis lze najít v API RTOS. Každá aplikace je spouštěna ve formě DOS tasku, tyto DOS tasky jsou vytvořeny s prioritou 25.

Časová základna RTOS používá realtime interrupt na frekvenci 1000 Hz. Z toho plyne rozlišení 1ms pro časování tasků.

2.6.2 Kritické sekce v provádění multitaskového kódu

Příkladem úlohy s kritickou sekcí provádění multltitaskového kódu producent – odběratel. V terminologii RTOS se jedná o task, který vytváří nějaká data a druhý task, který data čte

a používá je k další činnosti. RTOS nabízí 3 možnosti, jak se bránit vzniku kritických sekcí. Jsou jimi semafory, zákaz přerušení, RTOS task switch lock.

Semafory se používají k ochraně kódu přistupujícímu k nějakému sdílenému prostředku. Použití semaforů minimalizuje vliv na běh RTI. Potenciální nevýhodou je inverze priorit, kdy task s vyšší prioritou čeká na uvolnění semaforu.

Zákaz přerušení je nejzákladnější metodou, jak zabránit vzniku kritické situace. Výhodou této metody je minimální zatížení systému. Nevýhodou je, že sekce kódu se zákazem přerušení musí být provedena velice rychle, protože v této době je i zakázáno RTI, a tak nemůže dojít k preemci tasku. Vykonávaní kódu v takovéto sekci by nemělo trvat déle než 50us.

Poslední možností, jak zabránit kritickým sekčím v kódu, je použití RTOS task switch lock. Zakázání plánovaní tasků. Tato metoda nepotlačuje RTI, ale zakáže dočasně (do odblokování) preemci tasků. V této sekci by kód neměl být déle než 400 us.

2.6.3 Message exchange manager

Message exchange manager poskytuje mechanismus pro meziprocesní komunikaci. Message exchange manager je instantním řešením pro známý problém producent a konsument. A to pro situace, kdy máme jednoho anebo více producentů a potřebujeme doručit data jednomu a nebo více konzumentům (v případě více konzumentů ještě nevíme, kdo přesně data dostane).

2.6.4 Event manager

Event manager poskytuje vhodný mechanismus pro řízení tasků čekajících na události. Taková událost může být vytvořena timerem a nebo jiným taskem. Task manager umožňuje vyčkávání několika tasků na konkrétní událost a nebo na kombinaci konkrétních událostí.

2.7 Softwarové nástroje

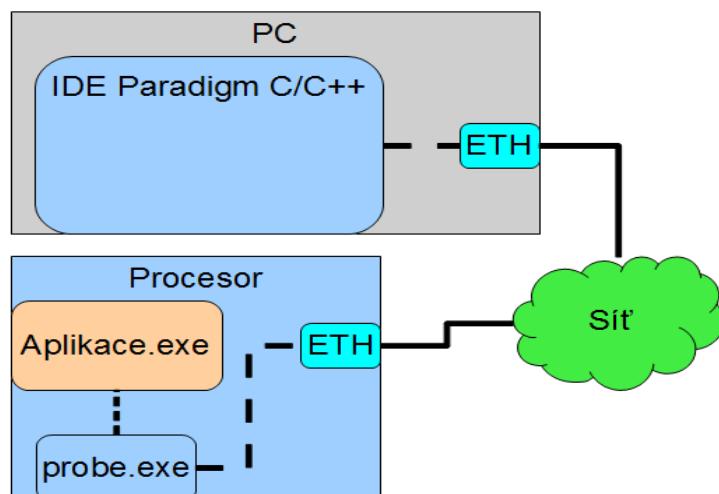
V této části kapitoly se budu snažit vyjmenovat některé SW užitečné nástroje pro konfiguraci, práci, debugování atd. Budu vyjmenovávat jen nástroje, které jsem použil a zajímavé.

2.7.1 Chiptool

Tento malý SW umožňuje konfiguraci procesorů, vyhledání procesorů, nahrání nové verze RTOS, podporuje jednoduchého FTP klienta, telnet. Tento sw se dá spouštět z příkazové řádky a vhodnými parametry ho lze využít k automatizaci výroby větších sérií. V prosinci 2010 existoval ve verzi 6.1.3.6

2.7.2 Paradigm C/C++ Beck IPC Editon Compiler for IPC@CHIP

Jedná se aplikaci kompletního IDE + překladače pro procesory IPC@CHIP firmy BECK. Obrázek 20 ukazuje, jak se provádí debugování s tímto nástrojem. Pro debugování je nutné do procesoru nahrát aplikaci probe.exe, která je dodávána s tímto SW. Aplikace probe také umožňuje nahrávat novou aplikaci do flash paměti (disk A:). Tento SW je placený a při spuštění je potřeba HW klíče. V případě cenové nedostupnosti lze použít překladač podporující 80186 instrukce. V prosinci 2010 byla aktuální verze 7.00.046.



Obrázek 20 Proces debugování s IDE Paradigm C/C++

2.7.3 ChipControl DLL

Jedná se o DLL implementující UDP protokol konfiguračního serveru. Takováto knihovna umožňuje v aplikacích třetích stran provádět konfiguraci zařízení, kterou by jinak zvládla jen aplikace chiptool. V prosinci 2010 existovala DLL ve verzi 2.1.7.0.

Kapitola 3

3 Návrh HW

Tato kapitola se zabývá návrhem HW nového zařízení, které lze zařadit do integrací diskutovaných v kapitole popisující integrace slaboproudých technologií v IT. V první části kapitoly vyberu vhodný procesor z jednotlivých procesorových řad firmy BECK. Dále se budu věnovat popisu komunikačních kanálů, jimiž jsou RS232, RS485/422, I2C, Ethernet. U každého komunikačního kanálu provedu stručný popis, zařazení do kontextu této práce a popis realizace. Dále se budu věnovat paměťovému mediu (SD karta) a jeho realizaci. V další části popíšu zdroje umístěné v zařízení a napájení zařízení. V poslední části se budu věnovat celému zapojení. Kompletní schéma, obrazec plošných spojů a tabulka s použitými součástkami jsou na přiloženém CD a v příloze.

3.1 Volba vhodné procesorové řady

Vhodnou procesorovou řadou je SC2x. Důvodem pro volbu této procesorové řady je dobrý poměr cena/výkon. Svým výkonem by měla být tato řada dostatečná pro řadu integračních činností. Obsahuje dostatek komunikačních linek, které lze v integracích použít. Nemalou výhodou tohoto procesoru je pouzdro (DIL32), které umožňuje ruční pájení. V této práci bude použit procesor SC23. Tento procesor obsahuje sice 2 MB flash oproti SC24, který má 8 MB, ale pro funkci zajištění funkcí navrhovaného zařízení jsou 2 MB flash naprosto dostačující. V případě potřeby mohou být tyto dva procesory prostou výměnou nahrazeny.

Hlavním důvodem pro zamítnutí řady SC1x je nedostatečný výkon a paměťové prostředky, což by stěží pokrývalo většinu žádaných integračních vlastností. Přesto i takovýto procesor (SC12) může najít uplatnění v integraci CCTV, jak bude ukázáno v kapitole Synchronizace obsahů monitorů.

Důvodem pro zamítnutí řady SC1x3 bylo pouzdro nevhodné pro ruční výrobu a mírně vyšší cena. Pokud by mohlo dojít k výrobě, která nebude ruční, tak by tato procesorová řada byla vhodným kandidátem, protože poskytuje stejné vlastnosti jako SC2x, ale navíc přidává více nožiček na pouzdro, což vede k odstranění blokování některých funkcí sobě navzájem.

Řada SC2x3 byla zamítnuta pro svojí vyšší cenu a vysoký výkon by v integracích nejspíš nebyl využit.

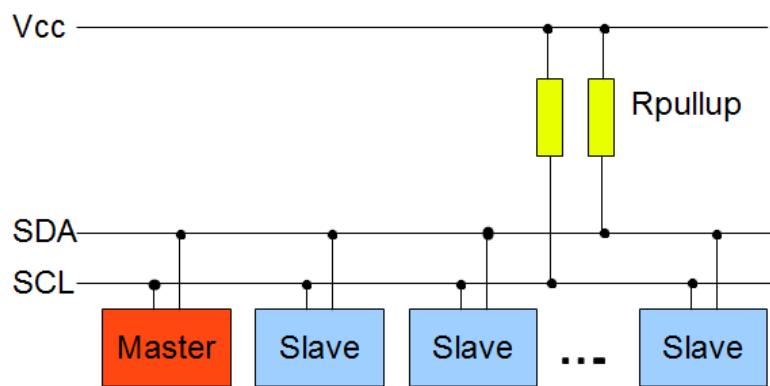
3.2 Komunikační kanály

V integracích slaboproudých zařízeních jsou komunikační kanály velice důležitou součástí zařízení. Proto jsem se rozhodl pro implementaci 4 komunikací, které by měly být schopny dostatečným způsobem zajistit všechny potřebné komunikace. Rozhodl jsem se použít komunikační linky I2C, RS232, RS485/422, Ethernet. Linky jsem seřadil podle možné vzdálenosti dvou komunikačních bodů od nejkratší po nejdelší.

3.2.1 I2C

I2C (správně označovaná jako I²C) vychází z anglického spojení Inter-Integrated Circuit. I2C je sériová multimasterová sběrnice, které se používá při komunikaci na velmi krátké vzdálenosti. Typickým příkladem je propojení periferií na desce plošných spojů, jako jsou třeba paměti, expandery vstupů a výstupů, nízko-rychlostní A/D a D/A převodníky, hodiny reálného času, další mikroprocesor atd...

I2C umožnuje propojení až 128 různých zařízení. Sběrnice je tvořena dvěma vodiči. Jeden vodič slouží pro přenos SCL hodinového signálu a druhý vodič pro přenos dat. Maximální délka vodičů je dána nejvyšší přípustnou kapacitou 400 pF. Maximální frekvence SCL je 100 kHz a 400 kHz. Tato frekvence je dána verzí I2C. Duplexní přenos není po I2C umožněn, což je dáno použitím jen jednoho vodiče pro data. Na obrázku 21 lze vidět, jak vypadá zapojení sběrnice I2C. Sběrnice obsahuje master, který zahajuje a ukončuje komunikaci a generuje SCL signál a dále pak zařízení slave, které je adresováno masterem. Pro adresování zařízení je dostupných 7 bitů. Ne všechny adresy jsou určeny k adresaci jednotlivých zařízení. Například adresa 0 je určena k vysílání broadcastu.



Obrázek 21 Zapojení sběrnice I2C

Tato sběrnice byla zařazena do navrhovaného zařízení, protože umožňuje připojit řadu periferií, které mohou být důležité pro integrace a které nebudou přítomny na desce. Sběrnice I2C

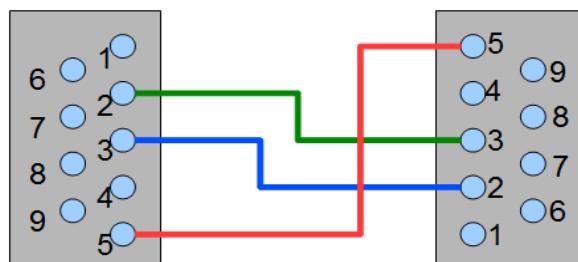
může být také použita pro komunikaci s dalším zařízením na desce plošných spojů, a tak může být rozšířena původní funkčnost pomocí modulů. Příkladem může být použití expanderů I/O výstupů, připojení znakového displeje LCD, realtime modulu atd..

I2C nebude v novém zařízení použít a, a proto byla vyvedena na konektor a byly doplněny dva rezistory Rpullup (viz. Obrázek 21) o hodnotě $4,7\text{ k}\Omega$. Procesor SC23 již obsahuje driver pro linku I2C.

3.2.2 RS232

RS232 je sériová komunikace určená pro komunikaci dvou zařízení. Poslední standard je RS-232C. V současné době je RS232 u osobních počítačů značně na ústupu, a proto se k PC doplňují různé převodníky sběrnic, které na jedné straně obsahují sériovou linku RS232. Standard RS232 pouze definuje fyzickou vrstvu modelu ISO/OSI. Tedy pouze určuje napěťové úrovně a způsob, jak přenášet sekvenci bitů. Pokud má PC sériovou linku, tak je vyvedena na konektoru D-SUB typu samec s devíti piny. Přenos dat probíhá asynchronně pomocí pevně stanovené komunikační rychlosti. Synchronizace se provádí sestupnou hranou start bitu. Data lze přenášet do vzdálenosti cca 20 m.

Sériová linka používá jiné napěťové úrovně než jsou TTL, a proto se často používají převodníky napěťových úrovní. Logická hodnota je klidový stav této sériové linky a je reprezentována napěťovou hodnotou mezi -25V a -3V . Nula je reprezentována kladnou úrovní napětí mezi hodnotami 3V a 25V . Sériová linka obsahuje další řídicí signály, ale ty jsou v dnešní době používány spíše výjimečně, proto je v návrhu počítáno pouze se signály RTS a CTS. Většina zapojení komunikuje po tří drátovém propojení, takzvaným Nullmodemem. Toto zapojení je vidět na obrázku 22.



2-RXD, 3-TXD, 5-SGND

Obrázek 22 Třídrátové propojení RS232

Tato sběrnice byla zařazena do navrhovaného zařízení, protože umožňuje základní ladění aplikace pomocí vypisování dat do sériové linky. Procesor SC23 umožňuje distribuovat command

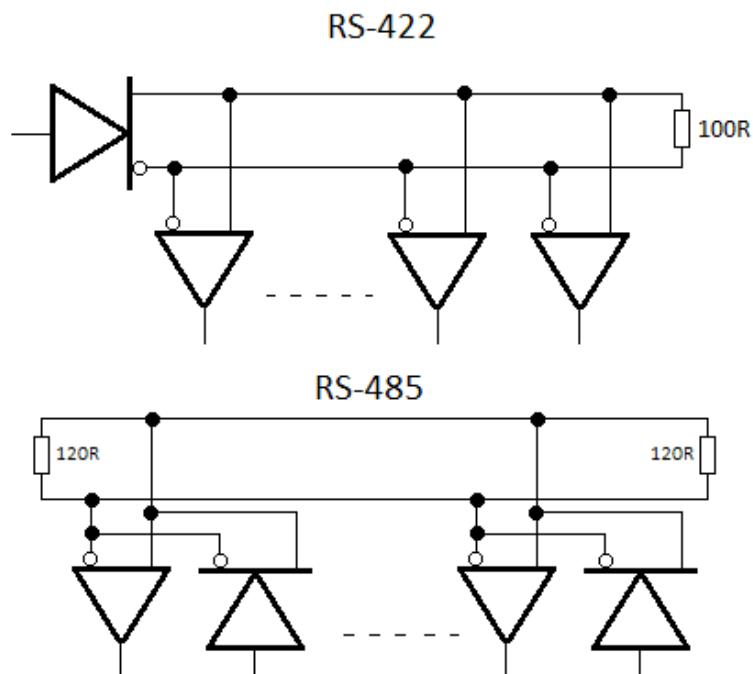
shell na sériovou linku, a tak je možné provádět nastavení po sériové lince například v případě, že není dostupná síť Ethernet. Dalším použitím může být propojení s ústřednami EZS a EPS, jednoduchá vazba na řídicí systém, realizace vzdálených I/O.

Na navrhovaném zařízení je linka RS232 realizována pomocí integrovaného obvodu SMD obvodu MAX3232CSE, který umožňuje napájení z 3,0V až 5,5V. Tento obvod obsahuje dva vysílače a dva přijímače linky RS232 a je určen pro přenosové rychlosti až 1Mbps. Pro správnou funkci je potřeba doplnit několik odporů a kondenzátorů. Přesné zapojení bude uvedeno na konci kapitoly. K převodníku jsou připojeny vodiče sériové linky UART1 procesoru SC23. Z napěťových úrovní procesoru je vytvořen signál napěťových úrovní RS232. Signál na úrovni RS232 je připojen na konektor DSUB9M. Tento konektor byl zvolen jako nejčastěji používaný.

3.2.3 RS485/422

Linky RS485/422 jsou sériové linky vhodné na přenos dat na delší vzdálenosti. Jejich topologie je sběrnicová. Linka RS485 a linka RS422 mají řadu společných vlastností. Společnou vlastností je stejný princip přenosu dat, stejná reprezentace logických stavů, dvouvodičové propojení zařízení, přenos dat na delší vzdálenosti, přenosová rychlosť kolem 200 kb/s. Rozdílnými vlastnostmi jsou jednosměrný přenos dat u RS-422 a obousměrný přenos u RS-485. Linka RS422 se typicky používá v případech, kdy potřebuje od jednoho zařízení přenést data na několik dalších zařízení, a my nečekáme od těchto zařízení odpověď. Těchto přijímačů může být až cca 10. Často se o RS422 mluví jako o průmyslovém ekvivalentu k RS232. Sériové linky RS485/422 jsou odolnější proti rušení než RS232. Odolnost rušení se dále zvýší použitím krouceného dvou-páru.

Obě sběrnice lze zapojit dvouvodičově a nebo čtyřvodičově. V našem zařízení budeme používat dvouvodičové zapojení. V tomto zapojení musíme počítat s tím, že musíme řídit přístup na sběrnici pro vysílače jednotlivých zařízení. Komunikační protokoly na RS485 často používají termíny MASTER a SLAVE, jako komunikační roli jednotlivých zařízení připojených na sběrnici. Dvouvodičové zapojení je na obrázku 23. V přenosech na delší vzdálenosti (připojení na rozdílné zdroje napájení) často musíme přidat třetí vodič jako zem (tentot vodič není v obrázku zakreslen). V obrázku jsou nakresleny zakončovací odpory, které zabraňují zákmitům, v případě propojení zařízení na delší vzdáleností.



Obrázek 23 Dvouvodičové zapojení RS485/RS422

Linka RS485/422 byla použita, protože její použití je v průmyslu velice časté. Dovoluje připojit vzdálené I/O obvody. Pro komunikaci ji často používají PLC automaty. V případě implementace jednoduchého protokolu MODBUS můžeme dostat zařízení, které lze velice dobře zapojit do stávajícího řídicího systému. Zajímavou aplikací by mohla být kombinace s Ethernetem s implementací protokolu ModBus Over Ethernet.

Hlavní součástkou pro realizaci zapojení sériové linky RS485/422 je obvod MAX3485CSA+. Tento obvod převádí binární napěťové úrovně procesoru na diferenciální napětí sériové linky RS485/422. O řízení přístupu na sběrnici se starají piny DI a RO. Výstup sériové linky je připojen na konektor RJ45. Tento konektor byl zvolen, protože často bývá používána strukturovaná kabeláž. Na desku plošných spojů navrhovaného zařízení byl dále přidán jumper pro odepínání a připínání zakončovacího odporu. Stabilita sériové linky byla dále zvýšena pomocí bias odporů. Tyto odpory slouží k stabilizaci klidové hodnoty linky v době, kdy se odpojují a připojují vysílače a přijímače.

3.2.4 **Ethernet**

Ethernet umožňuje zařízení propojit na nejdelší vzdálenosti ve srovnání s předchozími sběrnicemi. Nejednodušší variantou je propojení dvou zařízení bez dalších síťových prvků. V tomto případě musíme použít kabel zapojený do kříže, který je zapojen tak, aby byly propojeny vysílače a přijímače. Častější bývá varianta, kdy je zařízení připojeno do switche, a je tak možno realizovat

komunikaci s více zařízeními.

Fyzická vrstva Ethernetu může být realizována bezdrátově a nebo pomocí měděných vodičů. Naše zařízení bude používat fyzickou vrstvu realizovanou na měděných vodičích. Fyzická vrstva Ethernetu je definována specifikací IEEE 802.3. Ethernet je realizován jako sériová linka, která obsahuje dva protisměrné kanály, které zajišťují obousměrnou komunikaci. V návrhu zařízení je počítáno s realizací Ethernetu 10/100BaseT. Tato realizace je již podporována procesorem SC23.

Ethernet byl do navrhovaného zařízení zařazen pro svojí univerzálnost a výhradní použití v oboru IT. Ethernet umožňuje připojení do sítě Ethernet, což v případě potřeby v dnešní době umožňuje připojit se k zařízení odkudkoliv. Procesor SC23 obsahuje plnou podporu Ethernetu na úrovni HW a v API RTOS. Ethernet bude nejčastěji používán ke komunikačnímu připojení k integračnímu serveru a ke stanicím workstation. Velice zajímavou aplikací by bylo vytvoření integračního serveru na navrhovaném zařízení a realizace uživatelského prostředí využívajícího web server pro distribuci na stanice typu workstation.

HW realizace Ethernetu na navrhovaném zařízení obsahuje konektor s trfy. Tato trafa se starají o přizpůsobení spolu s několika pasívními součástkami. Takto přizpůsobený signál je přiveden na vstupy procesoru, kde je již fyzická vrstva Ethernetu realizována. Na konektoru jsou přítomny dvě diody, které se starají o signalizaci aktivity Ethernetu.

3.3 Paměťové medium (SD karta)

SD (secure digital) karta slouží jako perzistentní paměťové úložiště pro objemnější data. Na obrázku 24 je rozložení pinů SD karty. V tabulce 9 pak lze najít popis jednotlivých pinů v SPI módu, který bude realizován v navrhovaném zařízení. Existují také módy One-bit SD a Four-Bit SD, ale ty nebudou použity, a proto je zde nebudu popisovat.



Obrázek 24 Rozložení pinů SD karty

Pin	Název	Funkce v SPI móde
0	DAT3/CS	Chip Select (SS)
1	CMD/DI	Master Out Slave In (MOSI)
2	VSS1	Ground (zem)
3	VDD	Supply Voltage (Napájení)
4	CLK	SCK (Hodiny)
5	VSS2	Ground (zem)
6	DAT0/D0	Master In Slave Out (MISO)
7	DAT1/NC	Nepoužitý or nebo přerušení
8	DATA2/NC	Nepoužitý

Tabulka 9 Popis pinů SD karty v SPI móde.

Paměťovým médiem SD karty je paměť flash. SD karta dovoluje uložit od 1 MB dat až po 4 GB dat. SD karty se vyrábějí ve velikostech SD, mini SD, mikro SD. Na zařízení bude umístěn konektor, který umožňuje vložit pouze formát SD. Počet zápisů na kartu je výrobci nejčastěji udáván kolem čísla 100 000. Jednotlivé karty se liší také v rychlosti zápisu, tato změna rychlosti je označována Class X, kde X je číslo reprezentující maximální rychlosť zápisu v MB/s.

SD karta byla do zařízení použita, protože umožňuje ukládat velké objemy dat, které by se na flash procesoru (SC23 2MB) nevešly. Příkladem takových dat je např.: data loger, dočasné uložiště dat v případě výpadku integračního serveru, sledování tendencí teplot, vedení deníku událostí. Dalším příkladem použití SD karty je uložení dat, které potřebuje pro svůj běh SW uložený v paměti flash procesoru. Například to mohou být obrázky pro web server, javaaplety atd..

Realizace HW rozhraní SD karty obsahuje konektor pro SD kartu, který je připojen k procesoru SC23 SPI rozhraním. Ke konektoru SD musíme připojit napětí 3,3V a zem.

3.4 Napájení zařízení

Na zařízení budou umístěny dva zdroje napájení o hodnotách 3,3V a 5V. Ze zdroje 5V nebude v současné době napájen žádný integrovaný obvod, a toto napájení je použito pouze pro případ napájení dalších desek přes rozšiřující modul a pro připojení rezistorů pullup na sběrnici I2C. Ze zdroje 5 V jsou napájeny všechny použité integrované obvody včetně procesoru SC23. Použil jsem spínané zdroje z důvodů jejich vyššího efektivního převodu výkonu ze vstupu na výstup, menším prostorovým nárokům a nenutnosti osazení dalšího chlazení proti lineárním měničům.

Zdroj 5V je realizován integrovaným obvodem LM2594M-5.0, což je step-down měnič napětí. Tento obvod umožňuje měnit na napětí 5V až z hodnoty rozsahu vstupního napětí 8-60V stejnosměrných. Výstupní proud je až 0,5A. Tento použitý obvod dovoluje dosáhnout efektivity převodu výkonu ze vstupu na výstup až o 85% při odběru 0,5A a na výstupu 12 V stejnosměrných.

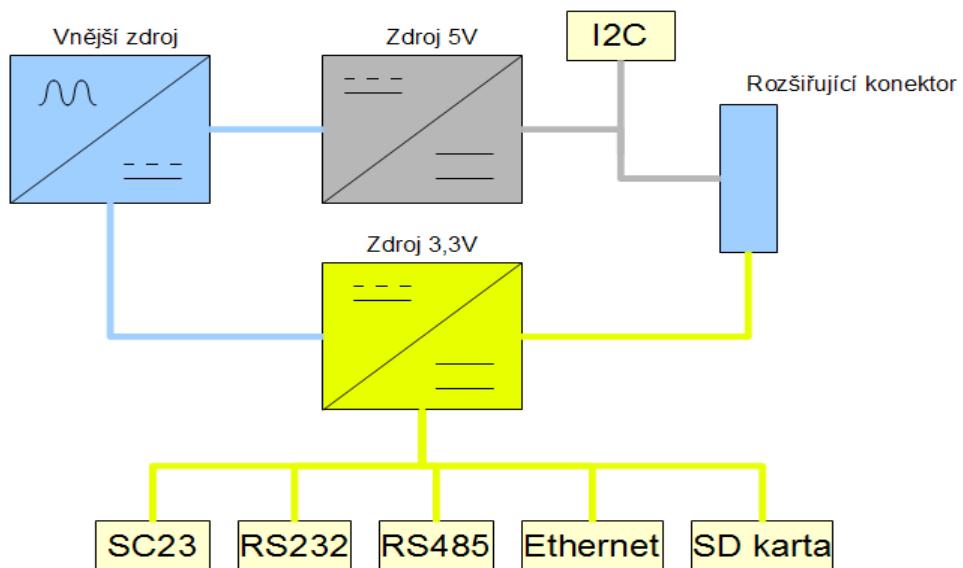
Pro svoji funkci potřebuje integrovaný obvod LM2594M-5.0 zapojení několika pasívních

součástek. Těmito součástkami jsou dva kondenzátory, pro filtraci výstupního napětí. Dále je obsažena cívka a shottkyho dioda pro realizaci stepdown měniče. Zapojení vychází z katalogového zapojení. K měniči je přivedeno napětí z konektoru přes několik filtračních prvků

Zdroj 3,3V je realizován integrovaným obvodem LM2672M-3.3, v tomto případě se opět jedná o stepdown měnič napětí. Rozsah vstupního napětí měniče se pohybuje v rozmezí 8V až 40V. Maximální povolený výstupní proud je 1A. Efektivita převodu výkonu ze vstupu na výstup tohoto obvodu při vstupním napětí 12V stejnosměrných a odběru 1A dosahuje hodnoty 86%.

Obvod LM2672M-3.3 potřebuje pro svojí činnost podobné součástky (ale jiné parametry) jako obvod LM2594M-5.0 použitý ve zdroji 5V. Vstupní napětí je přivedeno ke zdroji stejně jako v případě zdroje 5V.

Na obrázku 25 je blokové schéma napájení zařízení. V obrázku je také nakreslen vnější zdroj napájení z 230V střídavých na 12V stejnosměrných. Mezní a doporučené parametry, které by měl splňovat vnější zdroj, jsou uvedeny v tabulce 10.



Obrázek 25 Blokové schéma napájení zařízení

Parametr	Minimální hodnota	Doporučená	Maximální
Výstupní $U_{ef}[V]$	8	12	36
Maximální Výstupní proud $I_{ef} [A]$	2	2	

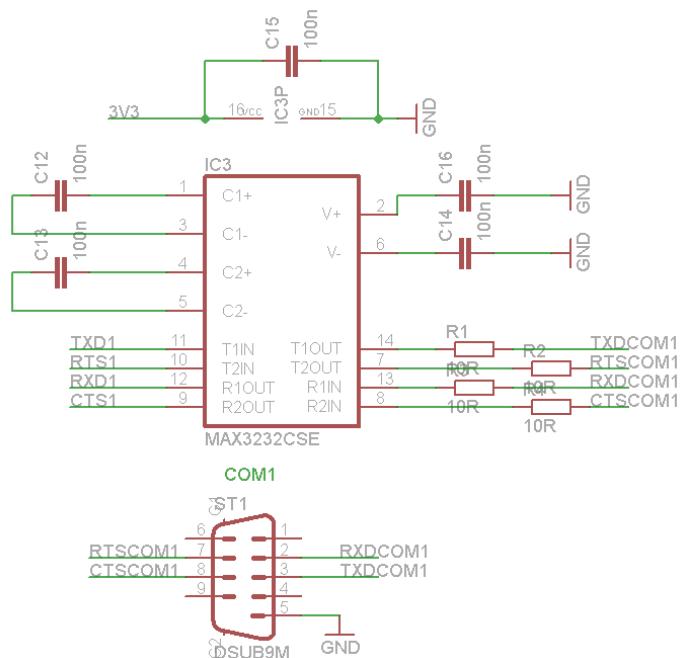
Tabulka 10 Mezní a doporučené parametry vnějšího napájecího zdroje

3.5 Celkové zapojení HW

V této části kapitoly nejdříve ukážu schémata jednotlivých funkčních bloků. Tyto funkční bloky dále spojím v blokovém schématu. Výkres kompletního schématu, výkres plošných spojů, seznam použitých součástek jsou obsaženy v příloze a na přiloženém CD ve formátu programu Eagle verze 5.10.

3.5.1 Schéma zapojení RS232

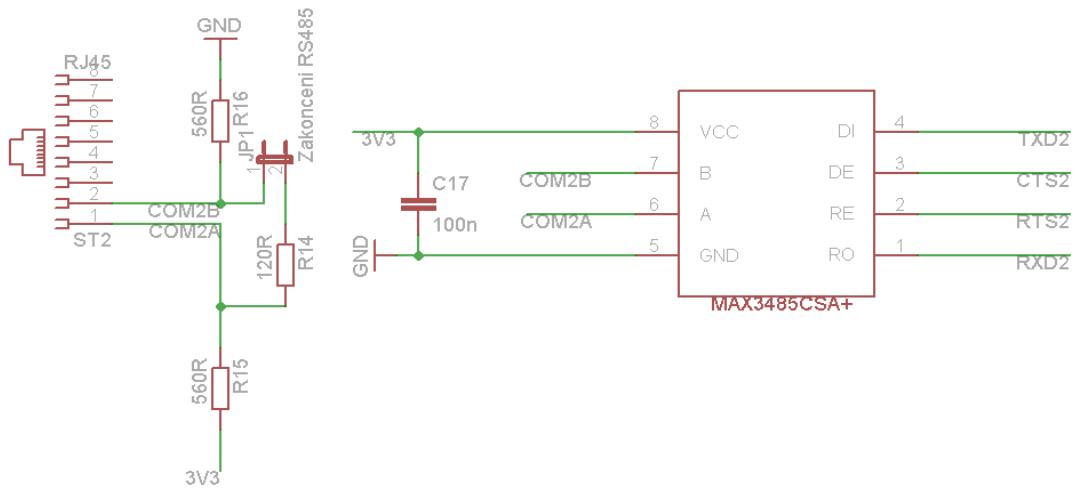
Na obrázku 26 je zapojení linky RS232, které obsahuje konektor a obvod MAX3232CSE. Toto zapojení se používá pro linku rs232 velice často, a proto není potřeba dalšího popisu.



Obrázek 26 Schéma zapojení RS232

3.5.2 Schéma zapojení RS485/422

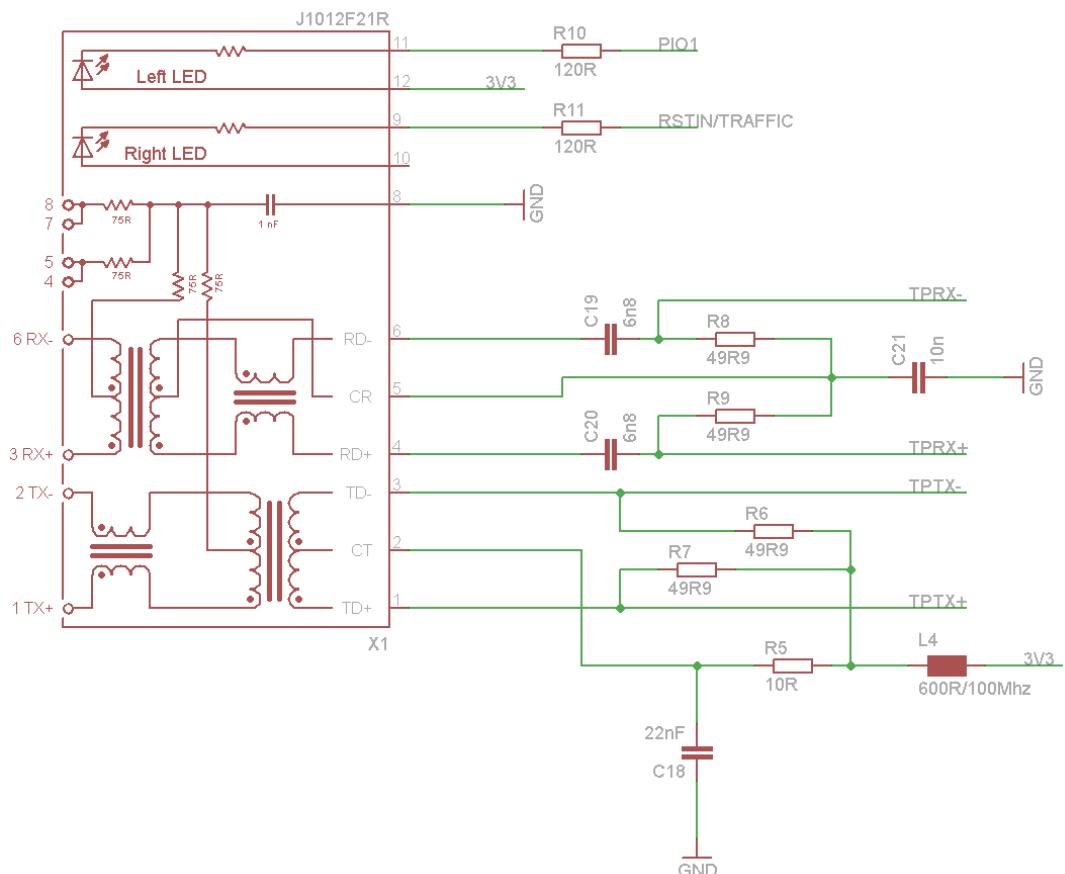
Schéma zapojení linky RS485/422 je vidět na obrázku 27. Zapojení obsahuje R16 a R17 jako bias odpory. R14 je zakončovací odpor, který je možno rozpojit propojkou JP1. Kondenzátor C17 je použit pro filtraci činnosti převodníku sběrnice MAX34855CSA+. Jako konektor je použit RJ45.



Obrázek 27 Zapojení linky RS485/422

3.5.3 Schéma zapojení Ethernetu

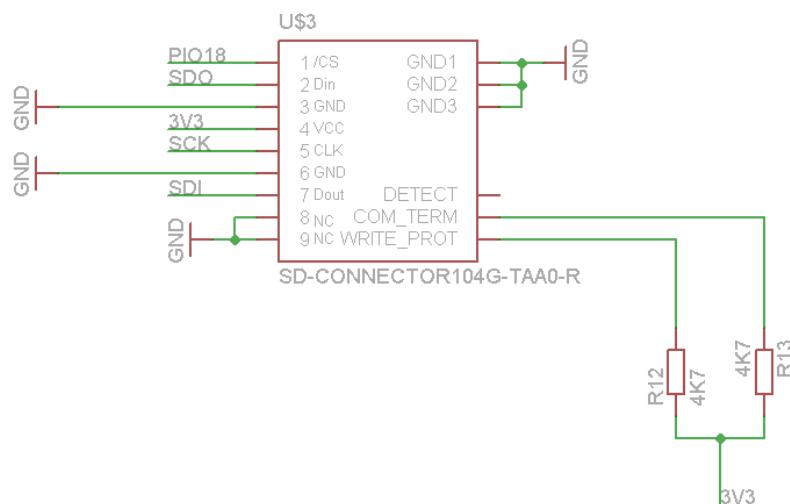
Na obrázku 28 je schéma zapojení Ethernetu. Hlavním stavebním prvkem je konektor RJ45 obsahující trafo přizpůsobení Ethernetu a led diody pro signalizaci aktivity. O přizpůsobení a o filtraci se dále stará několik pasivních součástek.



Obrázek 28 Zapojení ethernetu

3.5.4 Schéma zapojení SD konektoru

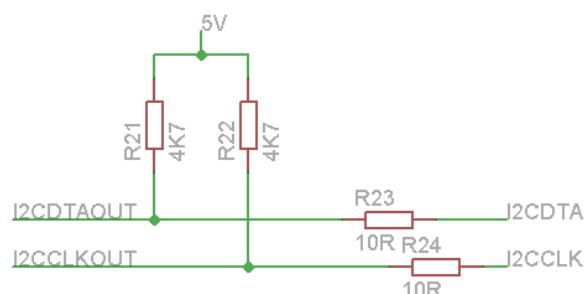
Zapojení konektoru pro SD kartu se nachází na obrázku 29. Ke konektoru jsou přivedeny signálové vodiče komunikačního kanálu SPI a signál pro chip select a napájení 3,3V.



Obrázek 29 Zapojení SD konektoru

3.5.5 Schéma zapojení I²C

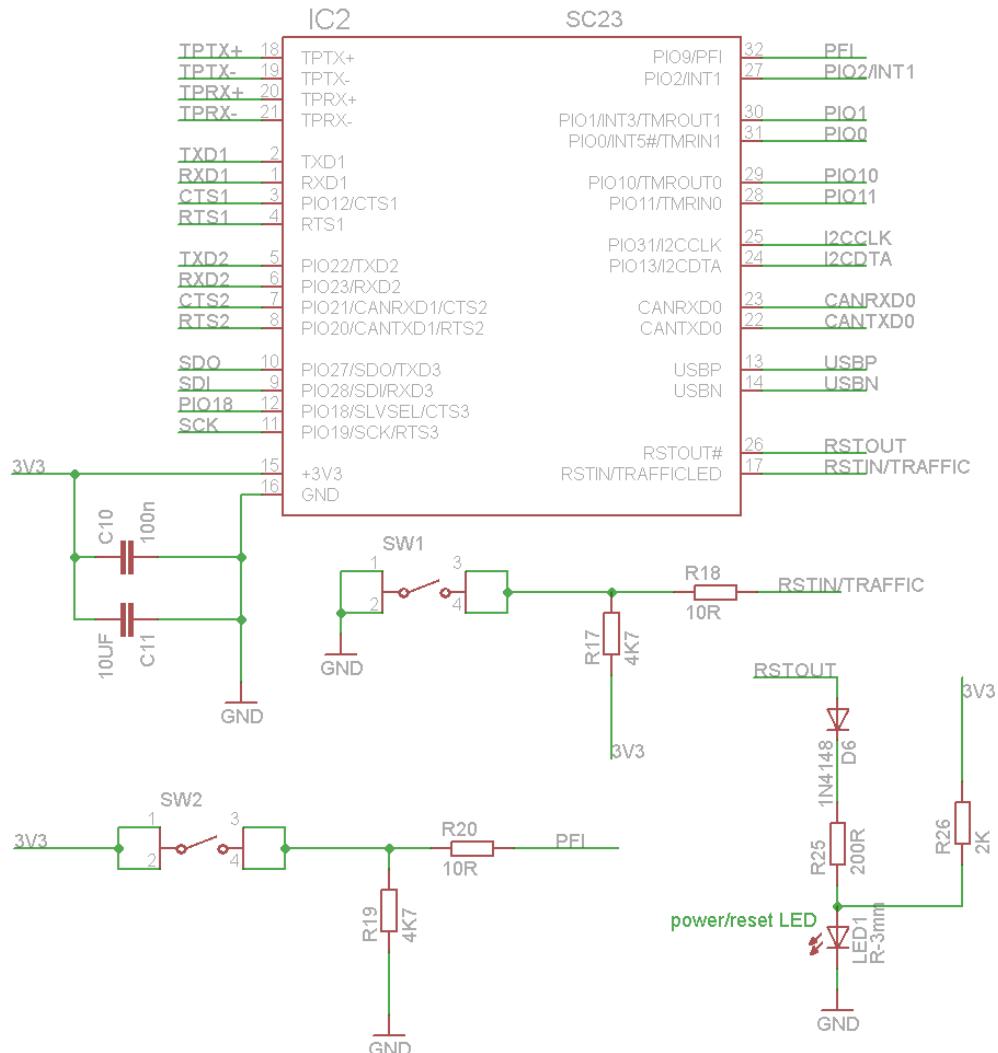
Zapojení I²C je vidět na obrázku 30. I²C není v současné době v navrhovaném zařízení použita, a proto je pouze vyvedena na rozšiřující konektor. R21 a R22 slouží jako pullup rezistory.



Obrázek 30 Zapojení I²C

3.5.6 Schéma zapojení procesoru SC23

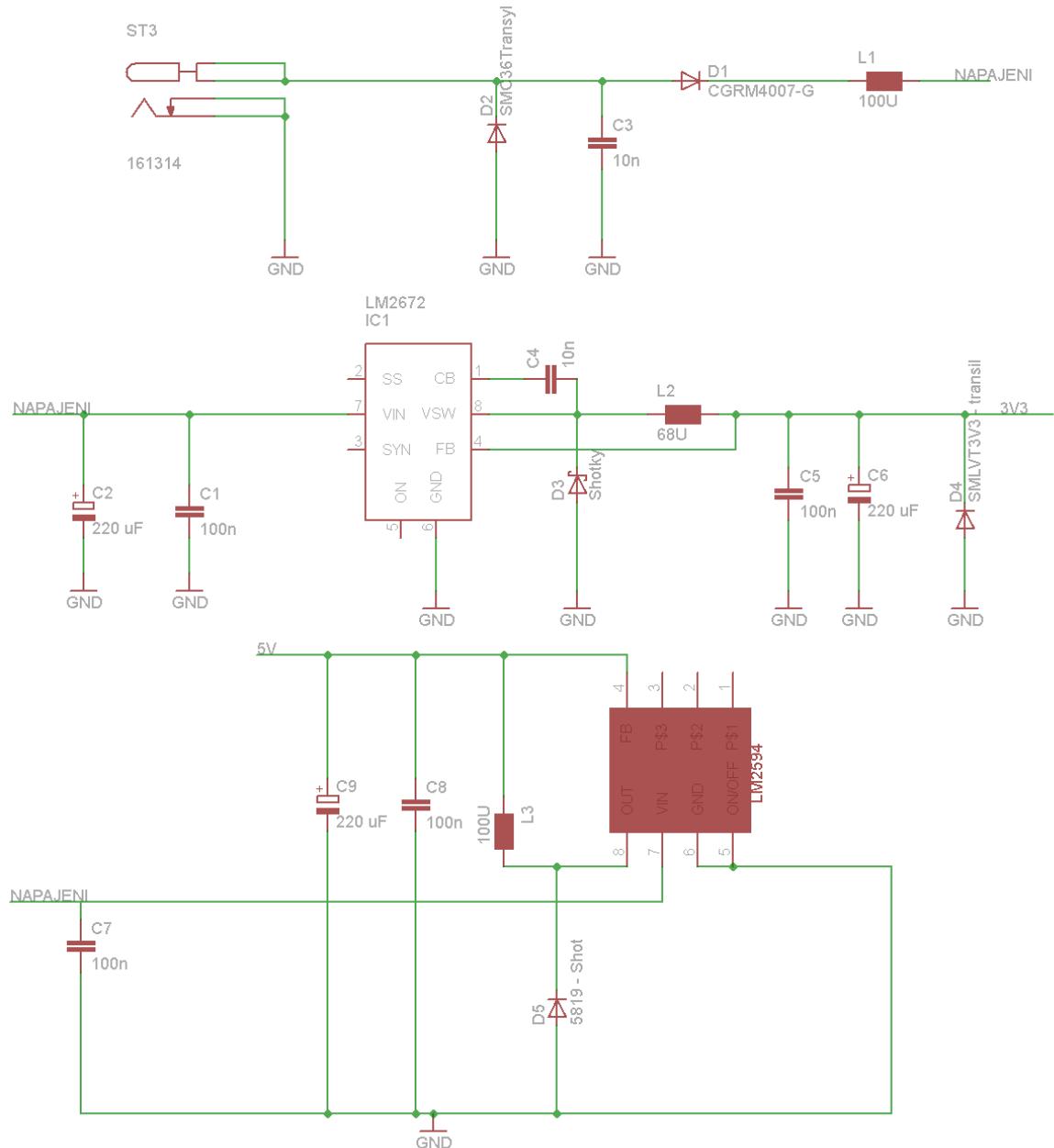
Schéma zapojení na obrázku 31 obsahuje tyto hlavní části: procesor SC23, dva spínače SW1 a SW2 a signalizační diodu LED1. Spínač SW1 provádí reset zařízení. Spínač SW slouží pro simulaci přerušení PFI. LED1 slouží pro signalizaci připojeného napájení a pro signalizaci resetu, když je její svit tlumený.



Obrázek 31 Schéma zapojení procesoru SC23

3.5.7 Schéma zapojení spínaných zdrojů 5V a 3V

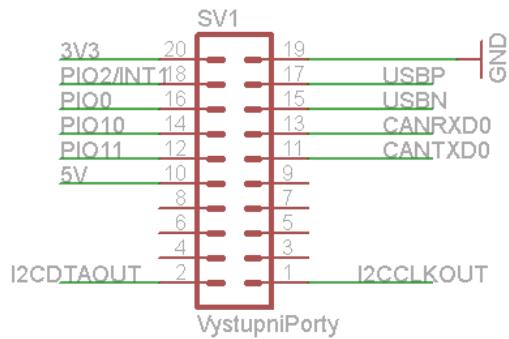
Na obrázku 32 se nachází schéma zapojení spínaných zdrojů 5V a 3V stejnosměrných. Zapojení obou integrovaných obvodů vychází z katalogových zapojení. Vstupní napětí na tyto měniče je přivedeno z konektoru ST3 přes součástky, které mají bezpečnostní (ochrana proti napěťovým špičkám a prohození polarity napětí) a filtrování funkce.



Obrázek 32 Schéma zapojení spínaných zdrojů 5V a 3V

3.5.8 Schéma zapojení rozšiřujícího konektoru

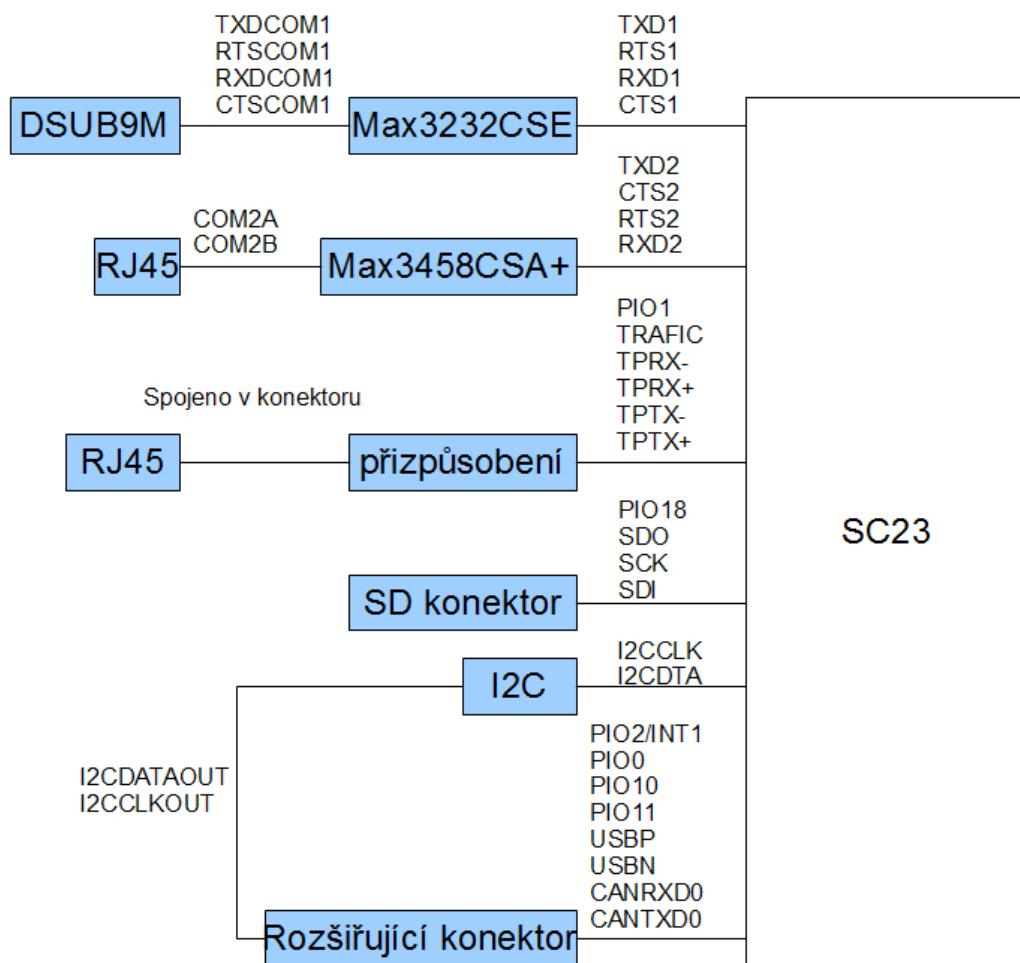
Rozšiřující konektor má za úkol připojení dalších rozšiřujících částí. Co všechno je připojené na tento konektor, je vidět na obrázku 33. Byl připojen každý nepoužitý pin a výstupy obou napájecích zdrojů. Pro vazbu na další zařízení má sloužit především sběrnice I²C.



Obrázek 33 Schéma zapojení rozšiřujícího konektoru

3.5.9 Blokové schéma propojení jednotlivých bloků

V tomto blokovém schématu není zohledněno napájení, protože to již bylo ukázáno na obrázku 25. V blokovém schématu je zdůrazněno, jaké signály připojení obsahuje, tak aby bylo jasné, jak jsou dané bloky propojeny. Centrem blokového schématu je procesor SC23, který ovládá a používá všechny ostatní funkční bloky. Blokové schéma je znázorněno na obrázku 34.



Obrázek 34 Blokové schéma propojení jednotlivých bloků

Kapitola 4

4 Návrh SW architektury

V této kapitole se budu věnovat návrhu SW architektury. Tato architektura musí splňovat požadavky, které budou vyjmenovány v první části kapitoly. Po vyjmenování požadavků na SW architekturu se budu věnovat popisu jednotlivých modulů. Tyto moduly by měly pokrýt základní potřeby v IT integracích.

4.1 Požadavky na SW architekturu

Vstupem pro tvorbu požadavků jsou předchozí kapitoly popisující prostředí, v jakém může být zařízení nasazeno, a jaký je dostupný HW. Soupis požadavků by měl být obecnějšího stylu, protože zařízení nebude obsahovat pouze funkcionality předepsanou zadáním této diplomové práce.

Obecné požadavky:

- Návrh jednotlivých modulů by měl být udělán tak, aby byl co nejméně závislý na konkrétní implementaci SW.
- Rozhraní jednotlivých modulů by mělo být přehledné.
- Návrh modulů by měl umožnit rozdistribuování jednotlivých částí aplikace na více zařízení.
- Jednotlivé moduly nesmí být psány tak, aby nešly použít v jiném projektu.
- Pokud bude použita nějaká forma komunikace, měla by umožňovat formu echa, které netestuje pouze funkčnost komunikace, ale i živost modulu.
- Komunikace s nadřazeným systémem musí poskytovat přehledné, co možná nejvíce jednotné rozhraní.
- Jednotlivé moduly by měly být přeložitelné pro všechny řady procesorů firmy BECK
- SW musí být modulární.
- Jednotlivé moduly musí být mezi sebou snadno kombinovatelné.

Z kapitoly popisující integraci slaboproudých zařízení v IT plyne zařazení několika modulů do SW architektury. U těchto modulů se předpokládá, že budou v budoucnu použity a nebo přímo budou použity pro potřeby této práce. Moduly specifické konkretní aplikaci budou popsány u jejich realizace.

Seznam modulů:

- TCPServer – realizace komunikace na Ethernetu
- TCPClient – realizace komunikace na ethernetu
- BinaryDriver – modul binárních vstupu a výstupů
- AnalogDriver – modul analogových vstupů a výstupů
- TemperatureDriver – ovladač pro čtení dat z teploměrů
- EventLog – deník události
- CGIWrapper – modul pro práci s cgi skripty
- CmdXmlServer – obsluha povelu ve formátu xml
- ChipControl – modul pro zjištění a řízení stavu procesoru
- CfgModul – uložení konfigurace specifické pro aplikaci
- TimeSync – synchronizace času

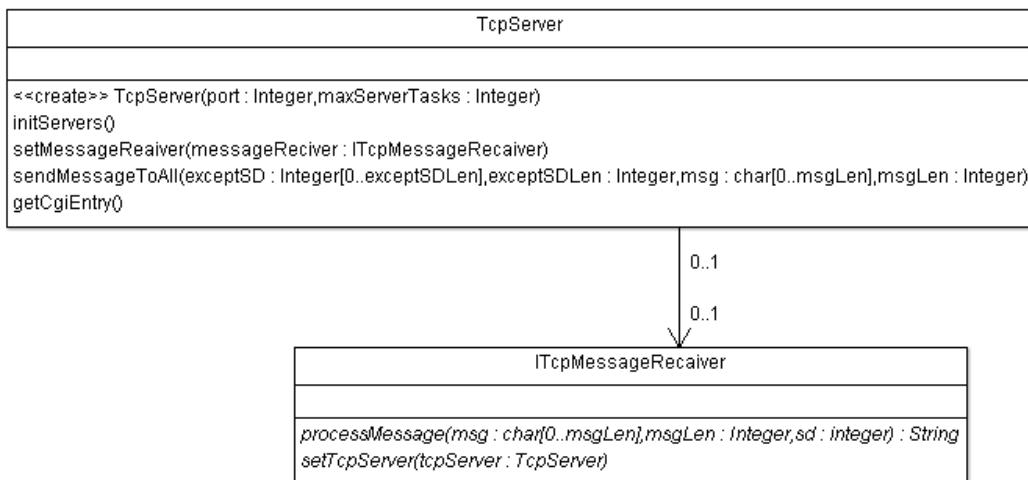
4.2 Popis modulů

V této části kapitoly se budu věnovat detailnějšímu popisu modulů a jejich analytickému modelu. Analytické modely mohou obsahovat moduly a třídy, které nebudou nijak podrobně specifikovány. Tyto moduly a třídy budou stručně popsány v popisech modulů. Zdrojové kódy v jazyku C/C++ modulů jsou na přiloženém CD.

4.2.1 TCPServer

TcpServer je modul sloužící pro realizaci připojení na Ethernetu protokolem TCP. Vytváří sokety pro TCP klienty. Tento modul nemá za úkol starat se, jak bude odpovězeno a komu bude odpovězeno. Analytický diagram tříd je uveden na obrázku 35. Na obrázku je vidět třída *TCPserver* a třída *ITcpMessageRecaiver*. *TcpServer* realizuje připojení TCP. V konstruktoru přejímá dva argumenty *port* a *maxServerTasks*, který definuje maximální počet připojených klientů. Funkce *sendMessageToAll* se používá pro odeslání informace všem připojeným klientům a můžeme ji předat i informaci o tom, komu informace nemá být odeslána. Funkce *getCgiEntry* vrací ukazatel na strukturu popisující reakci na CGI. Implementovaná cgi funkce bude vracet v současné době pouze výpis připojených klientů. Funkce *setMessageRecaiver* se stará o přiřazení instance s implementací abstraktní třídy *ITcpMessageRecaiver*. Implementace abstraktní třídy *ITcpMessageRecaiver* se stará

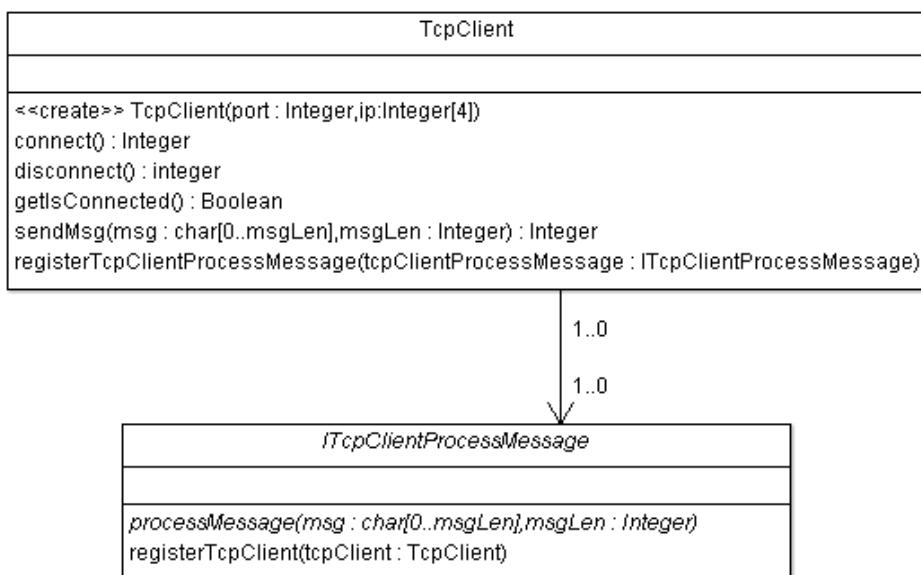
o zpracování informací z přijaté TCP serverem a vytvoření případné odpovědi.



Obrázek 35 Analytický diagram tříd pro TCPSever

4.2.2 TCPClient

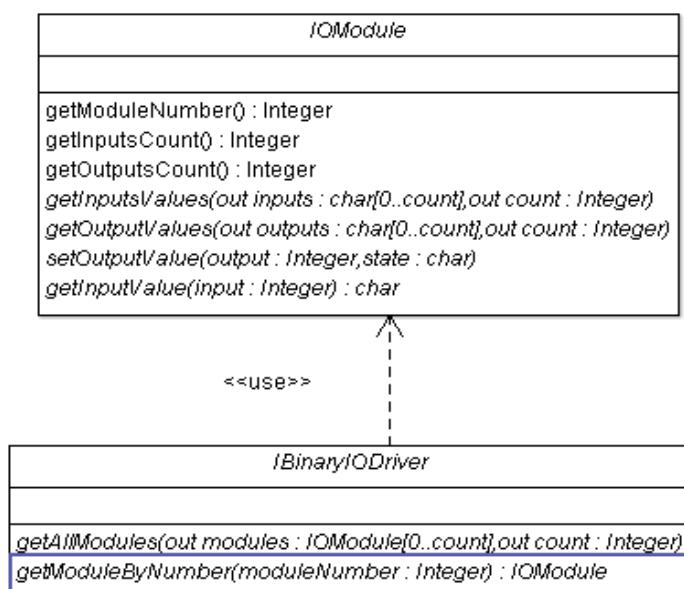
TCP klient je modul, který realizuje komunikaci na síti Ethernet protokolem TCP. Jeho úlohou je připojení k TCP serverům. Analytický diagram tříd je zobrazen na obrázku 36. Konstruktor třídy *TcpClient* přejímá dva argumenty port a adresu IP ve formátu ipv4. Pro připojení a odpojení serveru slouží funkce *connect* a *disconnect*. Pro poslání zprávy TCP serveru slouží funkce *sendMsg*. Funkce *registerTcpClientProcessMessage* slouží k registraci instancí s implementací abstraktní třídy *ItcpClientProcessMessage*, která určuje chování klienta.



Obrázek 36 Analytický diagram tříd pro TCPClient

4.2.3 BinaryDriver

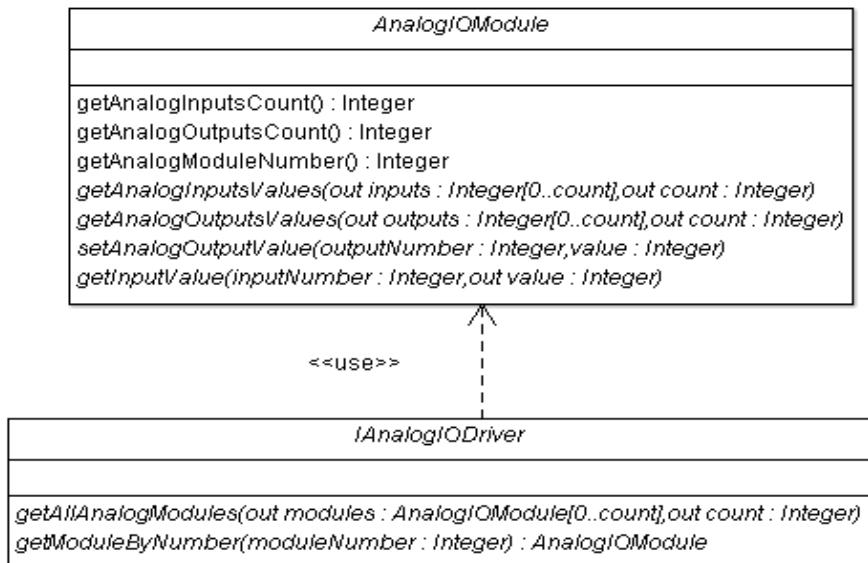
Úkolem Binary driveru je sjednotit přístup k binárním vstupům a výstupům, proto modul BinaryDriver pouze poskytuje rozhraní k binárním vstupům a výstupům a vlastní implementaci necházá až na použití daného zařízení. Obrázek 37 ukazuje analytický model modulu BinaryDriver. V modelu jsou obsaženy dvě abstraktní třídy. *IOModule* slouží pro realizaci binárního modulu vstupů a výstupů konkrétního typu. Implementace *IBinaryDriver* může být výrobní řada IO modulů od jednoho výrobce. Další možné použití těchto dvou abstraktní tříd je implementace *IBinaryIODriver* jako komunikační linky a *IOModule* budou jednotlivé moduly připojené na této komunikační lince. Z názvů funkcí je patrné, k čemu dané funkce slouží.



Obrázek 37 Analytický diagram tříd pro modul BinaryDriver

4.2.4 AnalogDriver

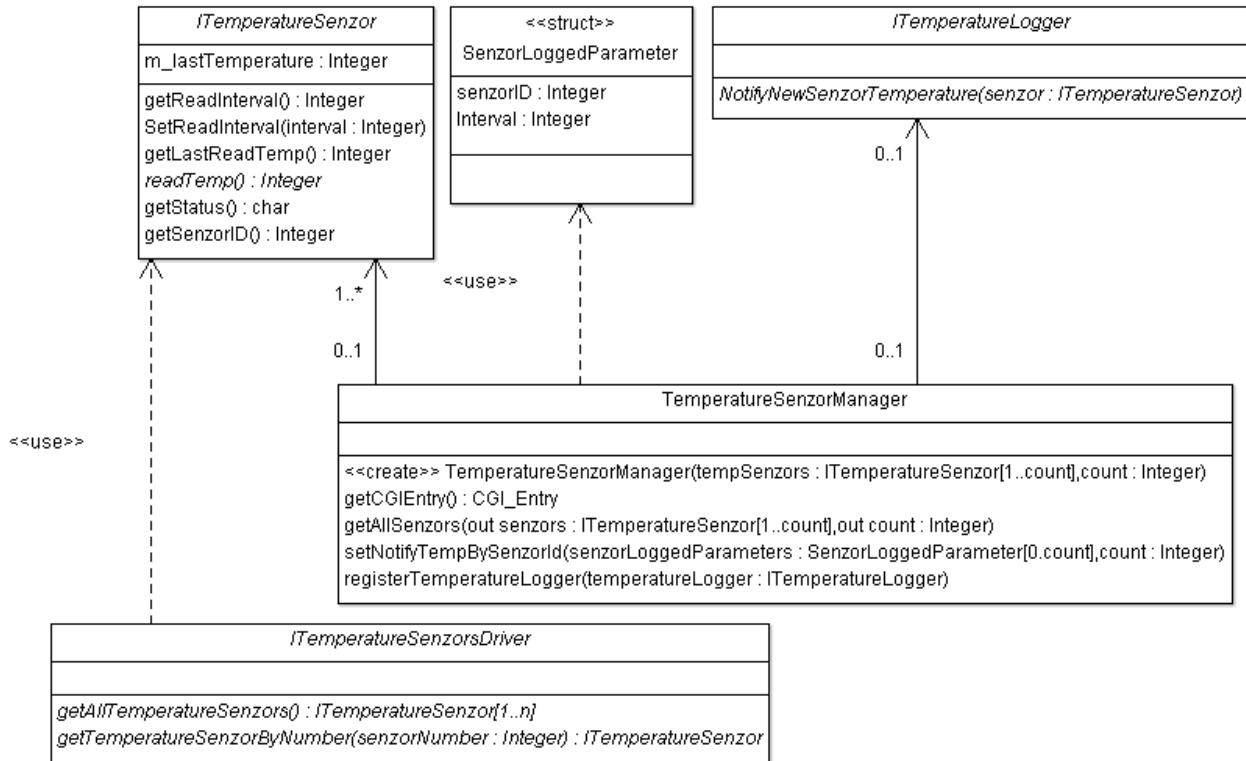
Analog driver se stará o ovládání analogových vstupů a výstupů. Návrh analogového driveru musí počítat i s frekvenčními vstupy a výstupy, protože ty jsou obdobou analogových hodnot. Obrázek 38 ukazuje analytický diagram tohoto modulu. Struktura modelu je analogická k modelu BinaryDriver. Třída *AnalogIOModule* reprezentuje analogový modul vstupů a výstupů. Abstraktní třída *IAnalogIODriver* reprezentuje skupinu modulů. Opět je tento modul pouze předpisem pro konkretní implementaci.



Obrázek 38 Analytický diagram tříd pro modul AnalogDriver

4.2.5 TemperatureDriver

Tento modul slouží pro čtení teploty, která je velice žádanou veličinou v building management řešení, ale i na jiných místech. Přestože je teplota analogová veličina a jako taková by mohla být realizována AnalogDriverem, bude tento modul řešit i několik dalších funkcí, které by samotný AnalogDriver nebyl schopen pokrýt, a proto bude tento driver realizován zvlášť. Na obrázku 39 je analytický model modulu TemperatureDriver. Model se skládá ze tří tříd a jedné struktury. Abstraktní třída `ITemperatureSenzor` reprezentuje konkrétní senzor teploty. Implementace této abstraktní třídy se skládá pouze z doplnění funkce `readTemp`, která je závislá na daném senzoru a v případě potřeby nemusí být přiřazena instanci třídy `TemperatureSenzor`. `ITemperatureLogger` slouží pro logování hodnot měřených teplot. Použití abstraktní třídy pro logování teploty značně rozvazuje ruce, protože nemusí docházet jen k logování do deníku událostí, ale například může docházet k postupnému vyhodnocování teplotních výkyvů v klimatizovaných místnostech atd... Struktura `SenzorLoggedParametr` se používá pro nastavení toho, který senzor se má logovat. Jednotlivé části propojuje instance třídy `TemperatureSenzorManager`, jejíž konstruktor přejímá senzory, které má třída používat. Třída obsahuje funkci `getCGIEntry` sloužící pro přehledné zobrazení všech hodnot přiřazených senzorů. Abstraktní třída `ITemperatureSenzorDriver` poskytuje jednotné rozhraní pro přístup ke všem senzorům ze skupiny zařízení.



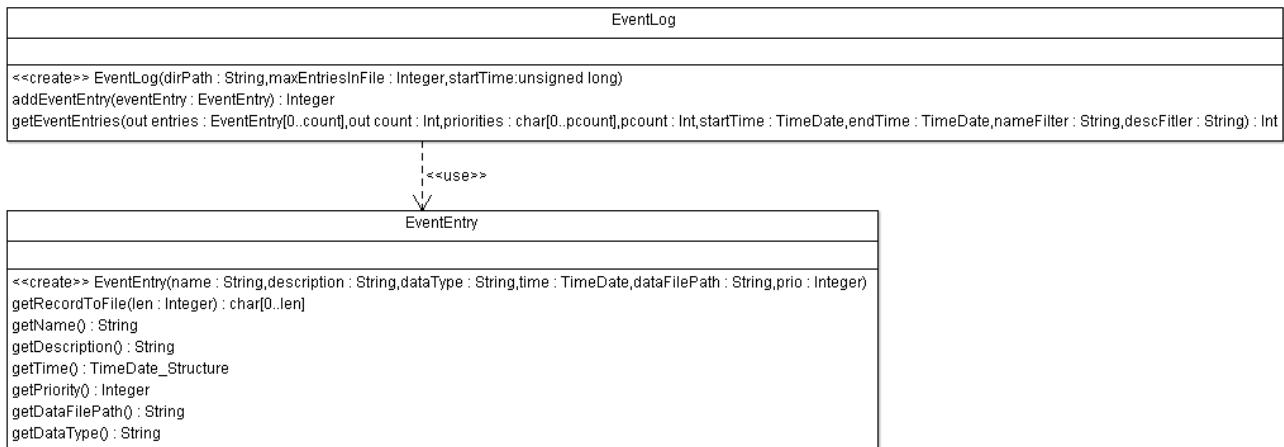
Obrázek 39 Analytický diagram tříd pro modul TemperatureDriver

4.2.6 EventLog

Event log slouží pro logování informací na perzistentní úložiště. Aplikace může logovat spoustu informací, příkladem může být logování teploty, změny binárních vstupů, chyb v komunikaci. EventLog musí obsahovat informace o tom, kdy došlo k dané události, název události, popis události, priorita události a případně nějakou formu ukazatele na data, která je kvůli případným větším nárokům vhodné ukládat mimo hlavní paměťovou oblast EventLogu. Protože RTOS na procesoru SC23 obsahuje souborový systém podobný tomu, jaký je v operačním systému DOS, budou se jednotlivé záznamy EventLogu ukládat na tento souborový systém a ukazatelem na přidružená data bude cesta k souboru s těmito daty. Záznamy jsou ukládány do souboru, který ve svém názvu obsahuje datum prvního záznamu. Tato struktura uložení umožňuje rychlejší vyhledávání v záznamu. S vyhledáváním v EventLogu není v současné době počítáno, protože při velkém množství dat, které můžeme ukládat například na SD kartu, lehko narazíme na výkonové omezení procesoru. Proto pro distribuci EventLogu bude použit FTP server, který je již obsažen v RTOS.

Na obrázku 40 je analytický diagram tříd pro modul EventLog. V tabulce 11 je formát jednoho záznamu. EventLog se skládá ze dvou tříd. Třída *EventLog* se stará o zápis do souboru v adresáři, ke kterému je přidána cesta v konstruktoru. V konstruktoru třídy *EventLog* je předáván

také parametr určující, kolik maximálně může být záznamů v jednom souboru. Do event přece jen byla přidána funkce pro vyhledávání záznamů, ale nebude použita. Třída *EventLog* nese vlastní záznam určený pro zápis do logu. Třída obsahuje funkci *getRecordToFile*, která vrací záznam v binární podobě, tak jak se má zapsat do souboru logu. Dále obsahuje funkce, které vrací jednotlivé hodnoty parametrů.



Obrázek 40 Analytický diagram tříd pro modul EventLog

Délka v B	1	1	1	2	1	1	1	1	1	1	1	1	1	NameL	DescL	DTL	FilePathL	1	
Název	SHX	Ver	NameL	DescL	DTL	FilePathL	yr	mn	dy	hr	min	s	Prio	STX	Name	Desc	DT	FilePath	ETX

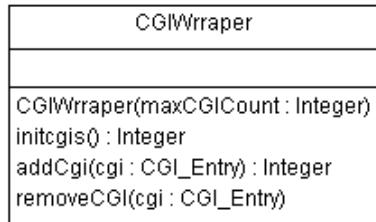
SHX	Start of Header
ver	Verze záznamu
NameL	Délka jména události
DescL	Delší popis události
DTL	Délka názvu typu dat
FilePathL	Délka cesty k souboru s daty
yr	rok vzniku události
mn	měsíc vzniku události
dy	den vzniku události
hr	Hodina vzniku události
min	Minuta vzniku události
s	Sekunda vzniku události
Prio	Priorita události
STX	Start of Text
Name	Název události
Desc	Popis události
DT	Typ dat události
FilePath	Cesta k souboru s daty
ETX	End of Text

Tabulka 11 Formát jednoho záznamu v event logu

4.2.7 CGIWrapper

Jedná se o modul, který vytváří wrapper pro funkce realizující CGI. CGI je zkratka Common Gateway Interface. Jedná se o protokol propojení externích aplikací s webovým serverem.

CGI lze také využít pro vytvoření dynamického obsahu web stránek. Na obrázku 41 je analytický model modulu CGIWrapper. Modul CGIWrapper se skládá pouze z jedné třídy. Jejím hlavním úkolem je zjednodušení poměrně složité registrace a odregistrace funkce realizující CGI. Konstruktor třídy CGIWrapper přebírá parametr určující maximální počet CGI funkcí přidělených na aplikaci.

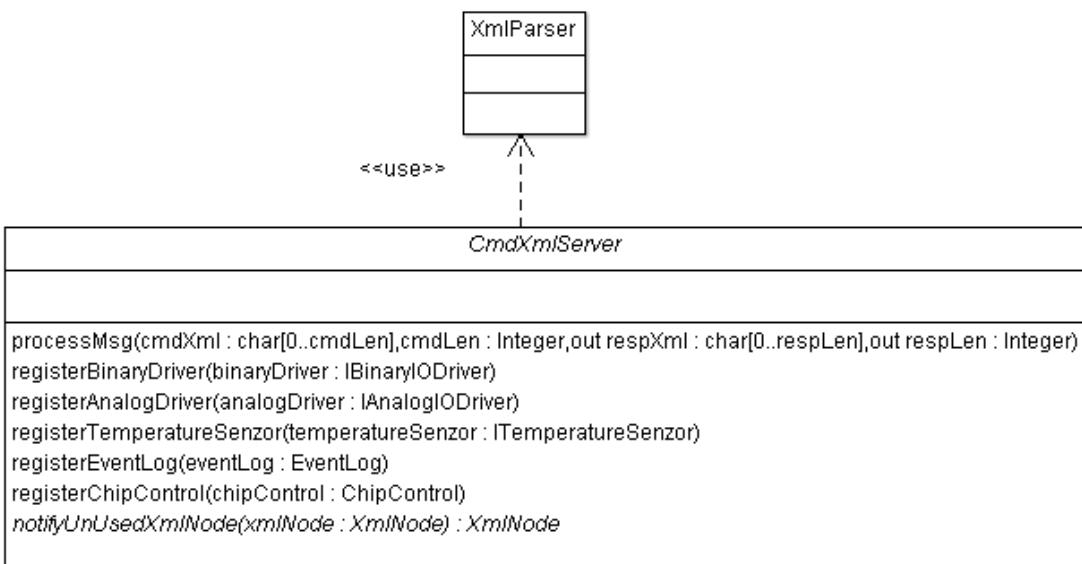


Obrázek 41 Analytický diagram tříd pro modul CGIWrapper

4.2.8 CmdXmlServer

Modul CmdXmlServer má za úkol zpracovávat příkazy a odpovídat na ně ve formátu XML, a tak vytvořit jednotné rozhraní pro příkazy. Extensible Markup Language (XML) je obecný značkovací jazyk, který je vhodný pro výměnu dat mezi aplikacemi. XML je založeno na jednoduchém textovém formátu, a proto je případně čitelný i textovým editorem. Jazyk XML není závislý na platformě, velice snadno se implementuje, protože každý moderní programovací jazyk podporuje zpracování XML. Jedinou nevýhodou XML jsou vyšší nároky na výpočetní výkon. Tato nevýhoda by neměla být problémem v integracích IT. Výhodou v integracích IT je všeobecná znalost jazyka XML, který dokáže implementovat do svých aplikací i začínající programátor velice rychle.

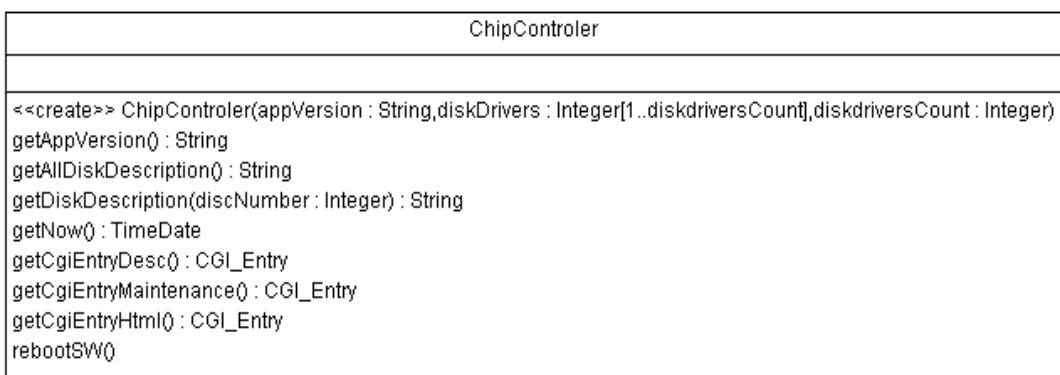
Analytický diagram tříd je na obrázku 42. Modul CmdXmlServer obsahuje třídu *XmlParser* a třídu *CmdXmlParser*. Xml parser byl převzat od Franka Vanden Berghena ze stránek <http://www.applied-mathematics.net/>. Tento xml parser je vhodný pro použití v Embedded zařízení. *XmlParser* je používán třídou *CmdXmlServer*, která se stará o implementaci vlastního povelovacího serveru. *CmdXmlServer* obsahuje třídy pro registraci jednotlivých objektů, které může povelovat a používat jejich data. Jako možnost rozšíření příkazů byla přidána virtuální funkce *notifyUnUsedXmlNode*, která se v případě implementace volá, pokud se při zpracování narazí na xml uzel, který není definován pomocí registrovaných objektů registračními funkcemi. Funkce *ProcessMsg* slouží pro zpracování příkazu ve formátu XML.



Obrázek 42 Analytický diagram tříd modulu CmdXmlServer

4.2.9 ChipControl

Modul ChipControl se stará o řízení procesoru. Na obrázku 43 je zobrazen analytický diagram tříd. Tento diagram se skládá z jedné třídy, která slouží jako fasáda pro komplikovanější programové konstrukce. Do budoucna bude nejspíš třída *ChipController* rozšířena o další funkce. Z názvu jednotlivých funkcí je vidět, k čemu dané funkce slouží. Výjimkou jsou jen tři funkce vracející strukturu pro zpracování cgi. Funkce *getCgiEntryDesc* slouží pro vrácení informace o procesoru, běžících úlohách, verze aplikace, velikosti disku atd. Funkce *getCgiEntryMaintenance* slouží pro ovládání procesu například k nastavení hodin, sw reset atd.. Funkce *getCgiEntryHtml* slučuje tyto informace do HTML stránky.

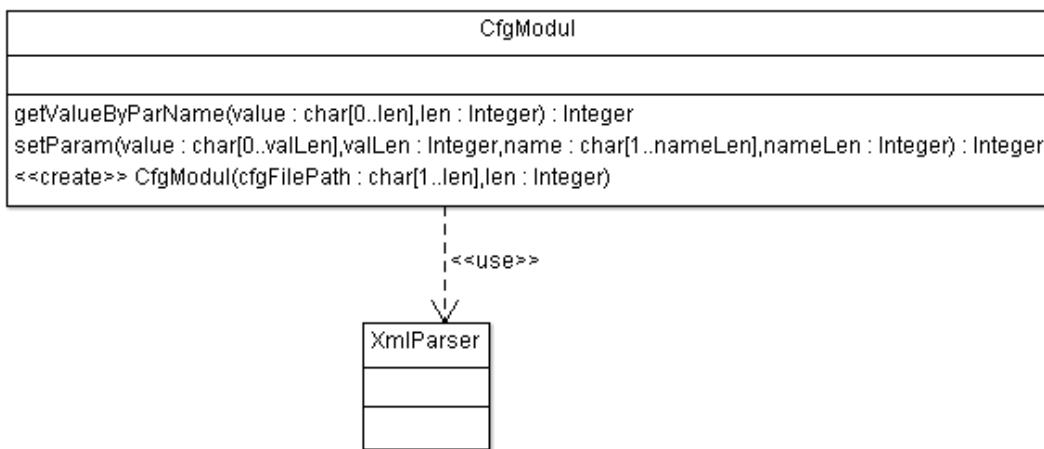


Obrázek 43 Analytický diagram tříd modulu ChipControl

4.2.10 CfgModul

CfgModul je modul pro ukládání a čtení konfigurace aplikace. Typickým příkladem může být uložení o připojení k serveru (port, IP adresa), konfigurace sériové linky, uložení připojení modulů v driveru pro binary. Pro uloženou konfiguraci by mělo platit, že je čitelnou a editovatelnou pro člověka. Z toho důvodu pro uložení konfigurace bude použit formát XML.

Diagram analytického modelu tříd je na obrázku 44. Modul konfiguračního nastavení se skládá ze dvou tříd. Třída *XmlParser* byla již dříve popsána. Zde je tato třída pro realizaci ukládání a čtení konfiguračního souboru ve formátu XML. Třída *CfgModul* obsahuje konstruktor, který přejímá cestu ke konfiguračnímu souboru. Dále obsahuje funkce pro čtení a pro nastavování parametru. Funkce *setValueByParName* nastaví hodnotu konfiguračního parametru a v případě, že parametr neexistuje, tak parametr vytvoří. CfgModul není určen pro velice časté ukládání a čtení dat, protože jednotlivé parametry jsou identifikovány podle názvu, a to znamená časově náročnější vyhledávání.



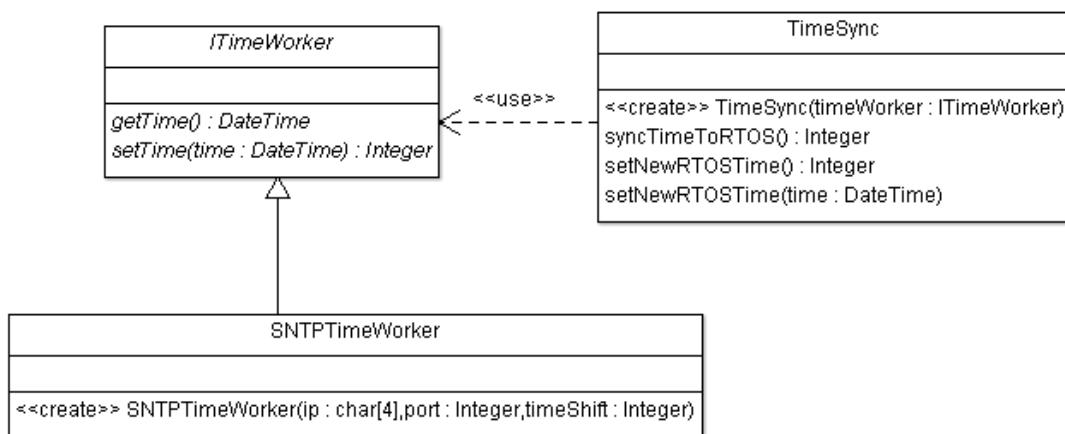
Obrázek 44 Analytický diagram tříd modulu CFGModul

4.2.11 TimeSync

Jedná se o modul synchronizace času. Jeho úlohou je nastavit správný reálný čas v RTOS. Protože v navrhovaném zařízení není obvod reálného času, bude pro synchronizaci použit protokol SNTP. Přesto by modul měl do budoucna poskytovat rozhraní pro získání a nastavení pro přesnější protokoly (např.: NTP) a pro obvod reálného času. SNTP je protokol, který se vyznačuje jednoduchou implementací, ale menší přesností, protože neuvažuje zpoždění paketů v síti a nepamatuje si stav předchozí komunikace. Pro naše účely je synchronizace času pomocí protokolu SNTP naprostě dostačující.

Analytický model modulu TimeSync je na obrázku 45. Základem modelu jsou třídy

ITimeWorker a *TimeSync*. Třída *SNTPTimeWorker* je konkrétní implementací abstraktní třídy *ITimeWorker*. Třída *TimeSync* se stará o aktualizaci času v RTOS. K získání správného času používá abstraktní třídu *ITimeWorker*.



Obrázek 45 Analytický model modulu TimeSync

Kapitola 5

5 Synchronizace obsahů CCTV monitorů

V této kapitole se budu nejdříve věnovat popisu problému, který má dané zařízení řešit. Dále vysvětlím, co si představit pod pojmem synchronizace obsahů CCTV monitorů. Protože integrační modul může být realizován na platformě PC i na platformě s použitím procesoru SC23 a nebo SC12, pokusím se obhájit volbu modulu DK40 s procesorem SC12. Dále popíšu realizovaný SW. Realizovaný SW se skládá z části realizující vlastní synchronizaci obsahů monitorů a z části realizující připojení na jednotlivých klientech. Na závěr kapitoly zhodnotím vytvořené řešení, které bylo použito na SSÚD Rudná. Toto zařízení je příkladem integrace slaboproudých technologií IP CCTV systému a embedded zařízení.

5.1 Popis řešeného problému

Základní úlohou synchronizace obsahů monitorů je pamatovat si informaci, která kamera je zobrazena na kterém monitoru. Protože pro každou kameru existuje číslo, které je v rámci celého CCTV systému unikátní, tak kamera bude reprezentována celým kladným číslem. Stejně je to i u monitorů, které budou také reprezentovány celými kladnými čísly. Toto zařízení je důležité, protože samotné CCTV tuto informaci spolehlivě neposkytují a nebo jí neposkytuje vůbec.

Daná situace je složitější, protože informaci o tom, na jakém monitoru je zobrazena jaká kamera, bude potřeba předat více aplikacím, které navíc běží na odděleném HW (v terminologii integrací workstation). Protože máme více aplikací, které mohou měnit zobrazenou kameru na určitém monitoru, tak může nastat souběh požadavků, který je nutné správně synchronizovat, tak aby nevznikala nekonzistentní data v synchronizaci obsahů monitorů.

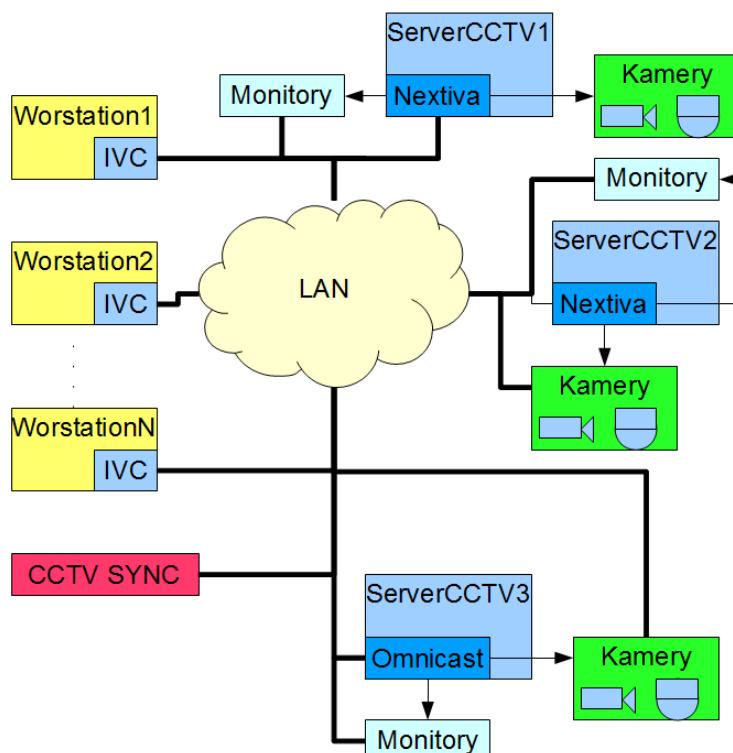
Dalším faktorem komplikujícím situaci je vyřešení a definování stavů, které mohou nastat při reálném nasazení aplikace. Například výpadek jednoho a nebo všech CCTV serverů. Kompletní výpadek napájení. Výpadek napájení synchronizace obsahů monitorů.

Obrázek 46 zobrazuje blokové schéma reálného CCTV systému, ve kterém byla nasazena synchronizace obsahů monitorů. Ve schématu jsou tři CCTV servery. ServerCCTV1 a ServerCCTV2 jsou osazeny softwarem výrobce Verint a značky Nextiva. ServerCCTV3 je osazen softwarem výrobce Genetec značky Omnicast. Z těchto serverů dochází k řízení jednotlivých kamer a monitorů a k nahrávání záznamů. Na síti je připojeno několik worksation. Workstation obsahují SW Integoo Video Clinet (IVC), které slouží pro zobrazení a ovládání jednotlivých CCTV serverů.

Z těchto video klientů dochází také k přepínání jednotlivých kamer na určené monitory. Z toho důvodu jsou tito klienti připojeni k synchronizaci monitorů. Důležité je si uvědomit, že k synchronizaci monitorů nejsou připojeny CCTV serverové softwary. CCTV systémy pouze realizují přepnutí kamer na určený monitor. Jednotlivé součásti jsou propojeny sítí LAN.

Abych mohl lépe vysvětlit, v čem spočívá použití synchronizace obsahů monitorů, tak si na chvíli představme, že synchronizace obsahů monitorů není přítomna. V takové situaci může chtít například Workstation1 zobrazit otočnou kameru s číslem 1 na monitor s číslem 10. Protože CCTV systémy Nextiva a Omnicast neposkytují informaci dalším workstation o výše uvedeném přepnutí, tak například workstation2, který přepne ovládání klávesnice na monitor označený 10, nemá informaci o tom, co se na monitoru vyskytuje za číslo kamery. V případě námi navolené otočné kamery nastává největší problém, protože nejsme schopni touto kamerou pohybovat. To znamená, že operátor CCTV systému musí provést znova přepnutí té samé kamery na ten samý monitor.

Dalším důsledkem vynechání synchronizace monitorů je nepřehlednost zobrazených kamer na jednotlivých monitorech, opět z důvodu nedokonalosti CCTV softwarů. Sice je možné získat informace o zobrazených kamerách na jednotlivých CCTV monitorech, ale při 26 monitorech v dostatečné vzdálenosti to znamená poměrně značné úsilí. Z toho důvodu lze synchronizaci monitorů použít i pro informování o všech kamerách, které jsou zobrazeny na jednotlivých monitorech.



Obrázek 46 Blokové schéma CCTV s připojenou synchronizací obsahů monitorů

Předpokládejme, že je CCTV synchronizace obsahů monitorů přítomna. V tomto případě, pokud se workstation1 pokusí přepnout otočnou kameru s číslem 1 na monitor s číslem 10, okamžitě jsou informovány všechny ostatní workstation. Následně na tuto akci se rozhodne workstation2 přepnout svůj kontext ovládání na monitor 10. Při přepnutí zjistí, že na monitoru 10 je kamera 1, což je otočná kamera a může kameru ovládat.

Další situací je, kdy dvě a více workstation mají přeplý kontext ovládaní na stejný monitor. V takové situaci se jedna z těchto workstation rozhodne změnit kameru na daném monitoru a okamžitě jsou informovány všechny ostatní workstation o této změně ze synchronizace obsahu monitorů.

Z výše uvedeného plyne, že synchronizaci obsahu monitorů může také nazvat jako vzdálený paměťový registr se řízeným přístupem ke změně dat.

5.2 Volba použitého HW

Pro realizaci fyzického zařízení se nabízí tři řešení - jedno řešení je použití HW typu PC, nebo použití navrhovaného zařízení s procesorem SC23 a nebo již hotového HW modulu DK40 s procesorem SC12.

Níže jsou uvedené výhody a nevýhody jednotlivých HW koncepcí, které jsou vztaženy k realizaci synchronizace obsahů monitorů

Výhody HW typu PC:

- Jednoduší realizace a ladění SW
- Obrovské množství výpočetního výkonu a prostředků

Nevýhody HW typu PC:

- Cena HW
- Vyšší šance nestability operačního systému
- Použitý OS windows, který není operačním systémem reálného času
- Dlouhý reboot systému
- HW obsahující často poruchové součástky (paměti RAM, mechanika HDD, větráky)

Výhody zařízení s SC23

- Nižší pořizovací cena

- Dostatek výpočetního výkonu a prostředků
- OS reálného času
- Neobsahuje mechaniku HDD
- Reboot cca 2s

Nevýhody zařízení s SC23

- Nové zařízení, které nebylo v době nasazení dostatečně otestováno

Výhody modulu DK40 s procesorem SC12

- Dostatečný výpočetní výkon a prostředky
- OS reálného času
- Neobsahuje pohyblivé mechanické součásti
- Reboot cca 1,5 s

Nevýhody modulu DK40 s procesorem SC12

- Menší dostupnost na trhu
- Vyšší pořizovací cena

Nakonec byl zvolen modul DK40, protože poskytuje dostatečný výkon pro uložení jednoduché informace a řízení komunikace, jedná se dostatečně otestovaný a stabilní HW a výhodou je velice rychlý reboot. Zařízení s procesorem SC23 bylo zavrženo, protože v době realizace nebylo dostatečně otestováno. Modul DK40 byl popsán v kapitole popisující jednotlivé řady procesorů firmy Beck.

5.3 SW synchronizace obsahu monitorů

SW synchronizace obsahů monitorů se skládá z SW dynamických knihoven pro PC a z aplikace pro modul DK40. Tyto dvě části SW spolu komunikují pomocí rozhraní Ethernet protokolem TCP, který nese informaci ve tvaru XML. Tabulka 12 ukazuje jednotlivé povely, které jsou posílány z SW na PC do SW na modulu DK40. Tabulka 13 ukazuje povely posílané z aplikace na modulu DK40 do SW na PC.

Název povelu	Povel	Parametry	Popis
Echo	<echo>Text</echo>	Text – text, který bude vrácen v odpovědi na echo	Slouží pro testování komunikace. Tímto echem je prověřena kompletní sekvence pro zpracování dotazu v SW na DK40
GetGrant	<getGrant></getGrant>	Žádné	Dotaz na povolení přístupu k synchronizaci monitorů. Tento příkaz je posílaný před každým zásahem a nebo čtením přidělených kamer na monitorech. Před echem se neposílá.
ReturnGrant	<returnGrant></returnGrant>	Žádné	Vrací grant. Po vrácení grantu není možno číst ani zapisovat do obsahu monitorů.
GetAllMonitorContent	<getAllMonitorContent></getAllMonitorContent>	Žádné	Tento slouží k dotazu na stav všech monitorů. Aby mohl být příkaz proveden, musí být vlastněn grant.
monitor	<monitor id="Monitor"> Kamera</monitor>	Monitor – číslo monitoru Kamera – číslo kamery	Zapisuje přepnutí kamery na monitor do registru monitorů. Pro provedení toho příkazu je nutné vlastnit grant

Tabulka 12 Povely pro synchronizaci obsahů monitorů

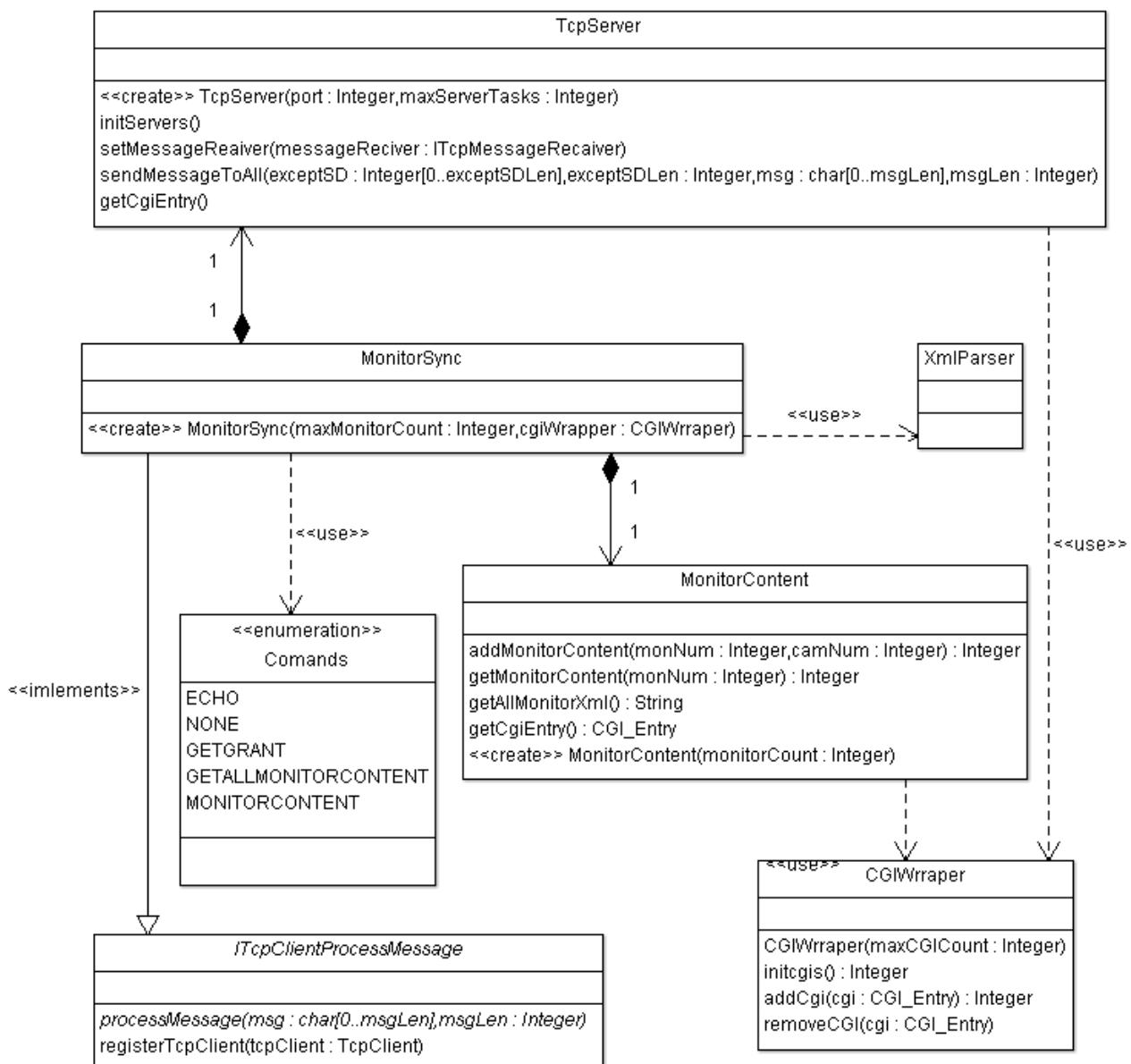
Název odpovědi	Odpověď	Parametry	Popis
Echo	<echo>Text</echo>	Text – text, který byl přijat v povelu Echo	Odpověď na povel Echo
GetGrant	<getGrant>Value</getGrant>	Value – hodnota 0 pro nepřidělení grantu a hodnota 1 pro přidělení grantu	Odpověď na dotaz GetGrant
MonitorContens	<MC><M id="Monitor"> ">Kamera</M>...<M id="Monitor"> ">Kamera</M></MC>	Monitor – číslo monitoru Kamera – číslo kamery zobrazené na monitoru	Odpověď na povel GetAllMonitorContent. Obsahuje aktuální obsah synchronizace obsahů monitorů
Monitorcallbeck	<monitorcallbeck id="Monitor">Kamera</monitorcallbeck>	Monitor - číslo monitoru na němž byla provedena změna kamery Kamera – číslo přepnute kamery	Tato odpověď je přijata v případě změny obsahu monitorů z jiné workstation, tato odpověď není posílána workstation, jež provedla přepnutí

RetVal	<code><RetVal>Value</RetVal></code>	<code>Value – 1 operace proběhla v pořádku. 0 operace se nepodařila</code>	kamery na monitor. Odpověď na povel monitor
--------	---	--	--

Tabulka 13 Odpovědi od synchronizaci pro SW na PC

5.3.1 Aplikace SW v modulu DK40

Analytický model aplikace pro modul DK40 je na obrázku 47. Obrázek obsahuje softwarové moduly, které byly již popsány v kapitole o SW architektuře. Dále obsahuje třídy *MonitorSync* a *MonitorContent* a výčet *Commands*.



Obrázek 47 Analytický model třídy synchronizace obsahu monitorů

Třída *CGIWraper* se používá pro registraci CGI funkcí instancí tříd *MonitorContent*

a *TcpServer*. CGI funkce třídy *MonitorContent* poskytuje informace o zobrazených kamerách a monitorech. CGI funkce třídy *TcpServer* zobrazuje informaci o připojených TCP klientech a přenesených datech.

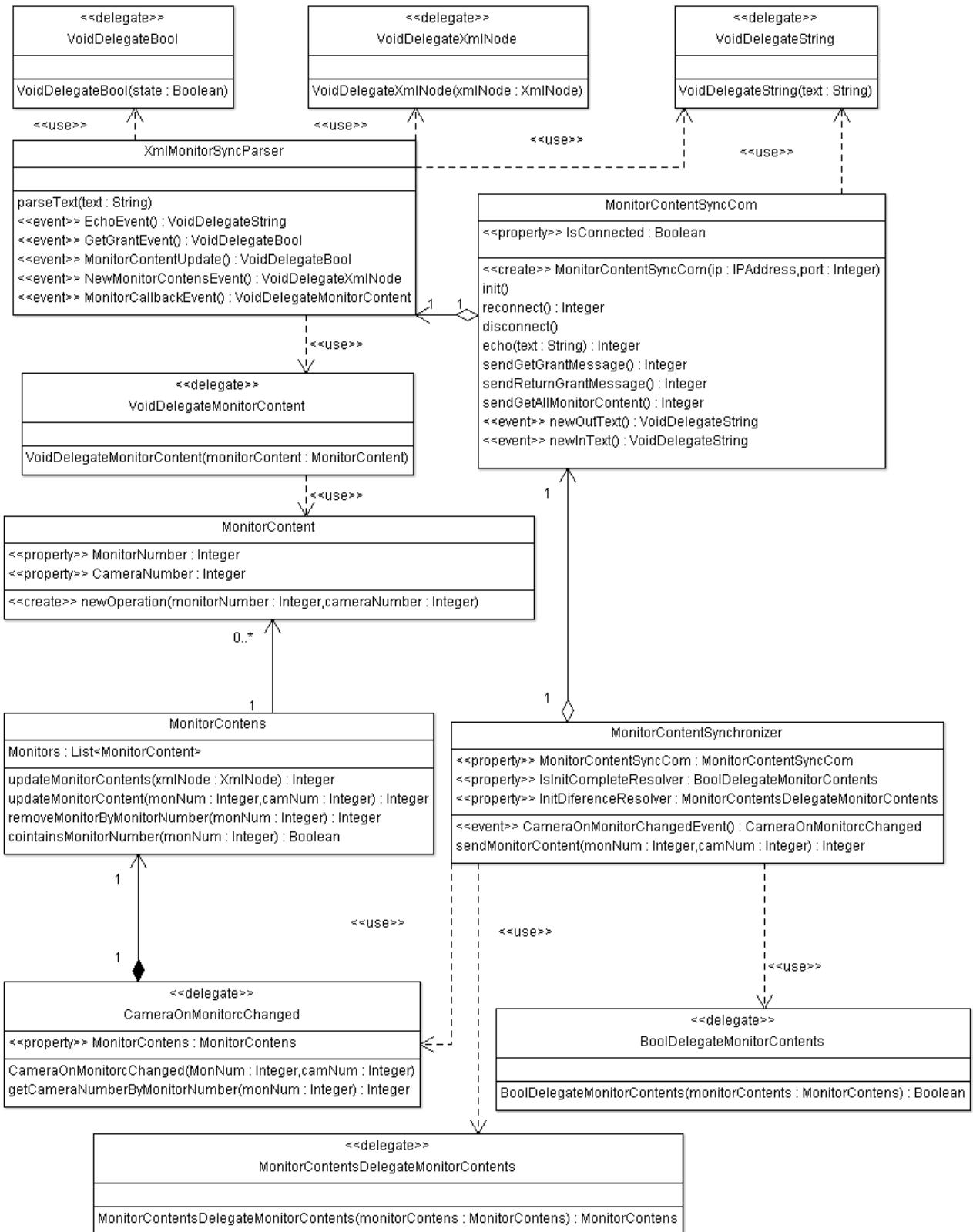
MonitorContent realizuje registr kamer připojených na monitor a poskytuje veřejné metody pro změnu a získání dat z tohoto registru. Konstruktor třídy *MonitorContent* přebírá parametr o velikosti registru.

MonitorSync je třída, která zpracovává vlastní povely z *TCP Server*. Pro zpracování dat z *TCP Server* je nutné implementovat metody abstraktní třídy *ITcpClientProcessMessage*. Povely převádí pomocí instance třídy *XmlParser* na literály výčtu *Commands*. Instance třídy *MonitorSync* řídí přístup k *MonitorContent* pomocí přidělování grantu pro přístup, tak aby nebyla narušena konzistence dat. Grant může být jednomu klientovi přidělen maximálně po dobu 3s. Pokud není grant v čas vrácen, dojde k automatickému vrácení.

5.3.2 DLL knihovna pro PC pro připojení k synchronizaci obsahů monitorů

Pro připojení k synchronizaci obsahu monitorů je nutné vytvořit knihovnu DLL (v terminologii .NET Assambly). Tato knihovna musí poskytnout rozhraní do vyšších vrstev SW a měla by co možná do nejvyšší míry odstínit od vlastní implementace. Pro realizaci byl zvolen jazyk C# technologie .NET framework firmy Microsoft.

Na obrázku 48 je analytický model tříd knihovny pro připojení k synchronizaci obsahu monitorů. Pro připojení synchronizace obsahu monitorů na vyšší vrstvy programu slouží třídy *MonitorContentSyncCom* a *MonitorContentSynchronizer*. Instance třídy *MonitorContentSyncCom* slouží pro realizaci vlastního připojení k synchronizaci obsahů monitorů. Jednotlivá data jsou přenášena v textovém formátu XML, pro parsování příkazů se používá instance třídy *XmlMonitorSyncParser*. Pokud je došlá odpověď od synchronizace obsahu monitorů v pořádku, je vyvolána příslušná událost, která signalizuje příšlou odpověď a případně nese i potřebná data. Instance třídy *MonitorContentSyncCom* se předává instanci třídy *MonitorContentSynchronizer*, která se používá pro přístup k obsahu synchronizace obsahu monitorů a zároveň řízení logiky komunikace. Na tuto třídu dojde k navázání vizualizace dat a další zpracování ve vyšších vrstvách. Třídy *MonitorContent* a *MonitorContents* slouží pro přenos a udržování informace o kamerách zobrazených na monitorech.



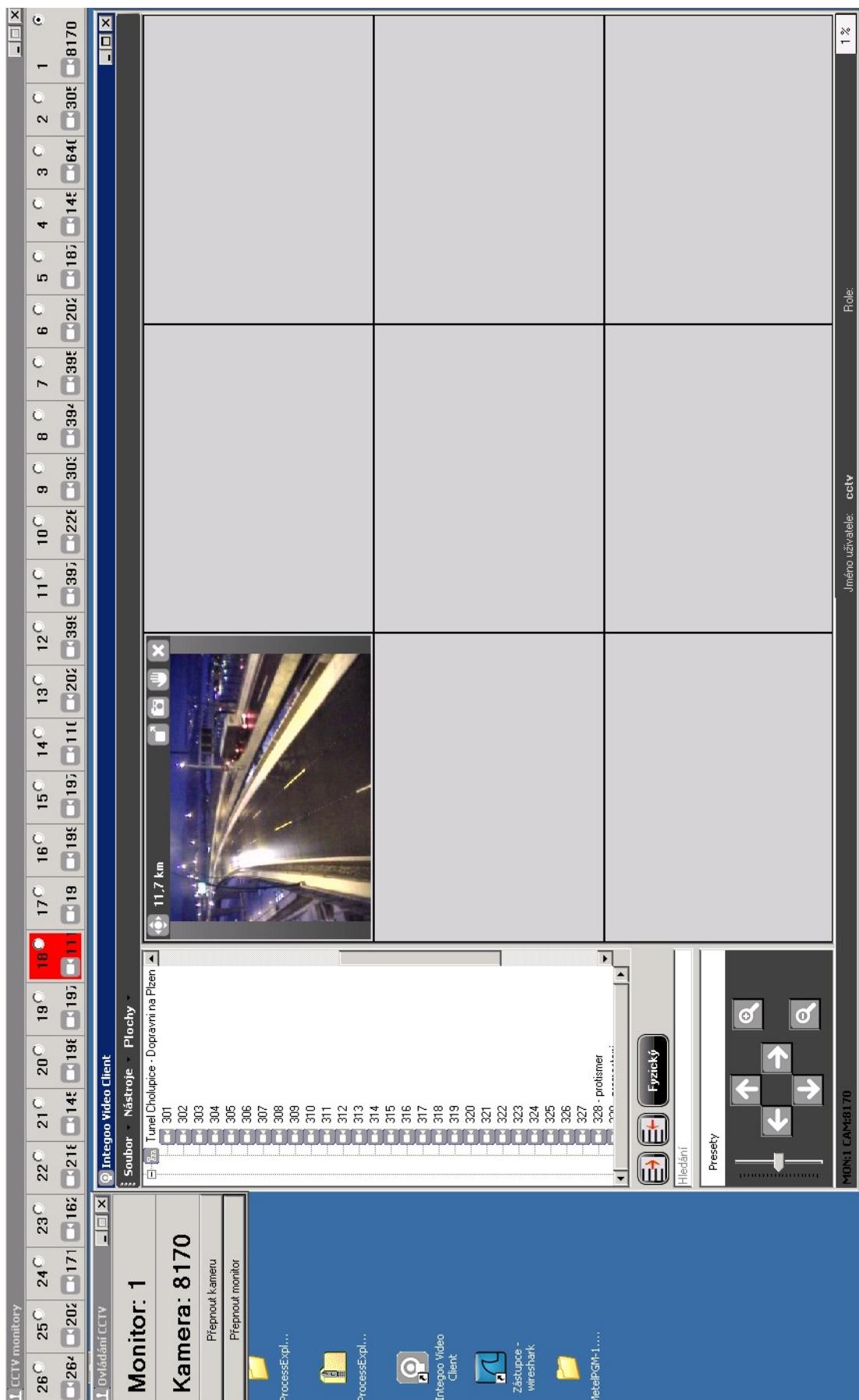
Obrázek 48 Analytický model tříd DLL knihovny připojení k synchronizaci obsahů monitorů

5.4 Zhodnocení synchronizace obsahu monitorů

Zařízení z této kapitoly bylo použito na centrálním dispečinku dopravy pro nově vybudovaný okruh v Praze na Rudné. V době psaní této diplomové práce nebylo potřeba provést jediný dodatečný zásah do SW a dokonce nebyl potřeba ani žádný restart tohoto zařízení. Na obrázku 49 je aplikace Integoo Video Client s připojením k synchronizaci obsahu monitorů.

Touto synchronizací obsahu CCTV monitorů je řízeno celkem 26 monitorů na dispečinku a 4 monitory v tunelech Cholupice a Lochkov. K synchronizaci obsahu monitorů se připojuje 7 aplikací Integoo Video Client a 2 služby starající se o alarmové monitory a vazbu na řídicí systém.

Jedinou nevýhodou takového zařízení je nemožnost měnit kamery na monitorech z aplikací a nebo HW, které nepodporuje připojení k synchronizaci obsahu monitorů. Pokud bychom připojili zařízení, které nelze připojit k synchronizaci obsahu monitorů, pak nejsou informována další zařízení, a může tak dojít k nekonzistenci dat.



Obrázek 49 Intego Video Client připojený k synchronizaci obsahů monitorů

Kapitola 6

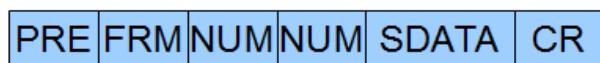
6 Rozhraní TCP – Protokol Spinel

V této kapitole popíšu realizace rozhraní TCP – protokol Spinel. V první části se budu věnovat popisu protokolu spinel české firmy Papouch. Dále se budu zabývat použitím tohoto rozhraní a začleněním do integračních technologií. Součástí toho popisu bude i vysvětlení přínosu. Předposlední částí kapitoly bude návrh SW založený na modulech popsaných v SW architektuře. Tento návrh obsahuje část aplikace na navrhovaném zařízení a DLL knihovnu pro implementaci na PC. V poslední části zhodnotím současný výsledek tohoto návrhu.

6.1 Protokol Spinel

Tato část kapitoly stručně popisuje protokol spinel firmy Papouch. Přesný popis protokolu je umístěn na stránkách www.papouch.com a na přiloženém CD. Zde budou popsány nejdůležitější aspekty protokolu, tak aby mohl být navržen SW realizující protokol spinel a všechna jeho rozšíření.

Protokol Spinel může podporovat až 255 formátů. Formáty 0 až 96 jsou určeny pro ASCII kódování a 96 až 255 pro binární kódování. ASCII formát nebude implementován, a proto ani popsán. Důvodem k tomuto rozhodnutí je vyšší spotřeba výpočetních prostředků. Dokumentace protokolu spinel doporučuje binární formát 97 pro všechna zařízení, pokud není důvod použít jiný formát. Binární formát 97 používají všechny I/O zařízení firmy Papouch. Na obrázku 50 je obecný formát pro binární kódování. Pokud dojde k přerušení komunikace, tak do 5s dojde k přerušení příjmu. Jednotlivé buňky v obrázku značí byty s výjimkou SDATA, kde je formát určen znakem FRM. Takový formát jednoho rámce dovoluje definici vlastního formátu dat bez toho, aby došlo k narušení činnost zařízení, která mají jiný formát.



PRE – Prefix (detekce začátku rámce)

FRM – Číslo formátu

NUM – počet bytů dat

SDATA – Data, jejichž formát je dán bytem FRM

CR – Zakončovací byte rámce (hodnota 0x0D)

Obrázek 50 Obecný formát pro binární kódování

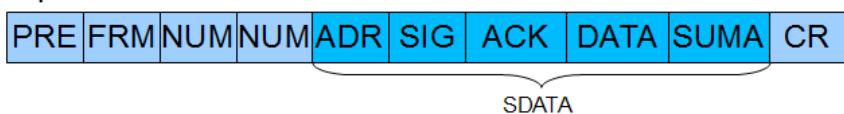
6.1.1 Binární formát 97 protokolu spinel

Protokol spinel s binárním formátem 97 je formát, který bude implementován v navrhovaném zařízení. Na obrázku 51 je znázorněn formát dotazu a odpovědi formátu 97. Formát 97 splňuje obecný formát z obrázku 50. SDATA v dotazu obsahuje adresu dotazovaného zařízení, podpis, instrukci, data a kontrolní součet. V odpovědi SDATA obsahuje adresu odpovídajícího zařízení, podpis poslaný v dotazu, znak potvrzení, data a kontrolní součet.

Dotaz:



Odpověď:



PRE – Prefix (detection of the start of the frame)

FRM – Frame number

NUM – number of bytes of data

ADR – Device address

SIG - Signature

INST - Instruction

ACK – Confirmation of the query

DATA – Data instruction

SUMA – Control sum

CR – Frame end byte (value 0x0D)

Obrázek 51 Binární formát 97 protokolu spinel

Délka dat označená v obrázku 51 NUM je 16-bitová hodnota. Hodnotu NUM je vypočítána jako součet bytů od následujícího bytu až po CR (včetně). Pokud je hodnota NUM menší jak 5, tak je dotaz považovaný za neplatný (ACK neplatná data).

ADR je jednobytová adresa, která určuje jednoznačně zařízení mezi ostatními na jedné komunikační lince. Tuto adresu posílá zařízení v odpovědi. Adresa 0xFF se používá pro odeslání všem zařízením umístěným na dané komunikační lince. Na dotaz s adresou 0xFF zařízení neodpovídají. Adresa 0xFE opět přijímají všechna zařízení, ale na tuto adresu odpovídají. To znamená, že na komunikační lince musí být pouze jedno zařízení.

Byte SIG slouží pro jednoznačné odlišení konkrétního dotazu a následné odpovědi.

INST je kód instrukce, která má být provedena. Pokud zařízení danou instrukci nepodporuje, tak je vrácen ACK neplatný kód instrukce. Jednotlivé kódy instrukcí pro I/O zařízení firmy papouch lze najít na stránkách www.papouch.com

Byte ACK slouží pro informování o zpracování dotazu. Jednotlivé kódy ACK bytu a jejich význam je v tabulce 14.

DATA obsahují data instrukce a nebo data odpovědi. Pokud se délka dat nerovná hodnotě NUM v dotazu, je v odpovědí ACK neplatná data.

SUMA kontrolní součet sloužící ke správnosti přenosu dat po komunikační lince. Pokud kontrolní součet neodpovídá, tak se na zprávu neodpovídá. Výpočet kontrolního součtu je proveden příslušných bytů instrukce:

$$\text{SUMA} = 255 - (\text{PRE} + \text{FRM} + \text{NUM} + \text{SDATA})$$

Tabulka 15 obsahuje příklady nejčačtěji používaných instrukcí. Tyto instrukce budou realizovány i v navrhovaném zařízení (ale návrh SW musí umožňovat jednoduché rozšíření o další instrukce).

Hodnota ACK (Hexadecimálně)	Význam
00	Vše v pořádku
01	Jiná chyba
02	Neplatný kód instrukce
03	Neplatná data
04	Nepovolen zápis/Přístup odmítnut
05	Porucha zařízení
0C	Automaticky vyslaná instrukce – přenos dat
0D	Automaticky vyslaná instrukce – Změna logické hodnoty na univerzálním vstupu
0E	Automaticky vyslaná instrukce – Kontinuální měření
0F	Automaticky vyslaná instrukce - ostatní

Tabulka 14 Význam kódů ACK binárního formátu 97 protokolu spinel

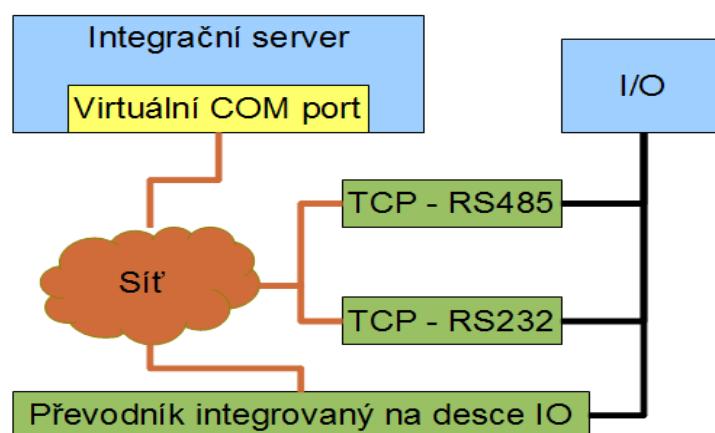
Příkaz	Rámec protokolu Spinel bin. formát 97
Sepnutí relé 2 a 6, rozepnutí relé 5 a 4, ostatní relé bezezměny, adresa zařízení 0x1, příkaz z podpisem 0x2. Odpověď je potvrzení příkazu	Příkaz: 0x2A,0x61,0x0,0x8,0x01,0x2,0x20,0x82,0x86,0x05,0x04,0x0D Odpověď:0x2A,0x61,0x0,0x4,0x01,0x02,0x0,0xD
Čtení vstupů na zařízení s adresou 0x1 a podpis 0x2. Odpověď je vstup 2,7,8 jsou v log.1 a ostatní v log.0	Příkaz: 0x2A,0x61,0x0,0x4,0x1,0x2,0x31,0xD Odpověď: 0x2A,0x61,0x0,0x5,0x1,0x2,0x0,0xC2,0xD
Čtení spojité veličiny ze vstupu 1 a 3 ze zařízení s adresou 1 a podpisem 0x2. Odpověď jsou hodnoty 0x1234 na vstupu 1 a 0x89AB na vstupu 3	Příkaz: 0x2A,0x61,0x0,0x6,0x1,0x2,0x51,0x5,0x15,0xD Odpověď: 0x2A,0x61,0x0,0xB,0x1,0x2,0x0,0x1,0x12,0x34,0x3,0x89,0xAB,0xE7,0xD

Tabulka 15 Příklady instrukcí binárního formátu 97 protokolu Spinel

6.2 Začlenění do IT integračních zařízení

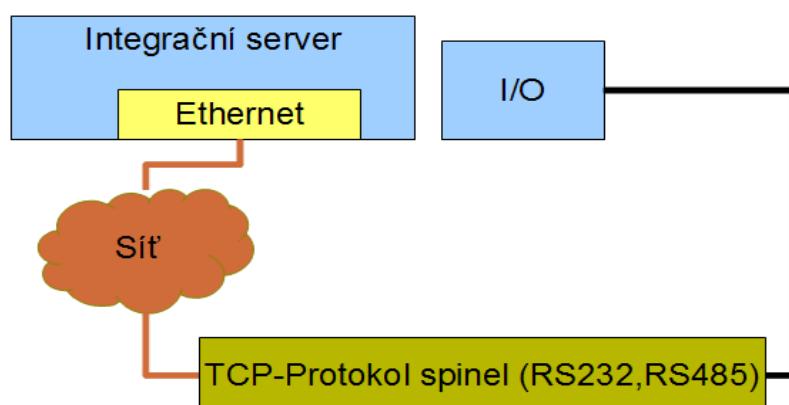
Na obrázku 1 v první kapitole byla ukázaná typická IT integrační architektura. Na obrázku 2 jsem ukázal, jak se změní architektura, pokud zařadíme nové zařízení. V této části ukážu, co konkrétního se změní, pokud použijeme rozhraní TCP – Spinel realizované na navrhovaném zařízení.

Obrázek 52 ukazuje blokové schéma systému s binárními I/O komunikující protokolem spinel bez použití navrhovaného zařízení. Z obrázku je vidět, že pro komunikaci s obvody I/O je nutný převodník sběrnic. Pro převodníky na sériové linky se typicky používá virtuální sériový port, který poskytuje iluze sériového rozhraní na PC. Metoda virtualizace sériového portu nedovoluje připojení více zařízení k převodníku sběrnic.



Obrázek 52 Blokové schéma zapojení binárních I/O bez použití navrhovaného zařízení

Obrázek 53 ukazuje blokové schéma připojení binárních I/O s použitím navrhovaného zařízení TCP-Protokol spinel. V porovnání s obrázkem 52 je vidět, že místo převodníků sběrnic je zapojeno navrhované zařízení, které funguje jako rozhraní TCP – Protokol Spinel. Toto zapojení umožňuje připojit několik integračních serverů.



Obrázek 53 Blokové schéma zapojení binárních I/O s použitím navrhovaného zařízení

Po porovnání blokových schemat z obrázku 52 a 53 nalézáme jediný rozdíl v odstranění převodníku, který dovoluje pouze vytvoření jediného komunikačního kanálu. Vytvoření více komunikačních kanálů bychom mohli řešit vytvořením aplikace, která bude komunikovat s převodníky sběrnic, a data dále předávat po Ethernetu, podobně jako navrhované zařízení.

O výhodách použití navrhovaného HW bylo podrobně diskutováno v kapitole popisující HW zařízení. Významnými výhodami navrhovaného zařízení jsou nižší pořizovací náklady proti PC, použití RTOS místo tradičních operačních systémů, reboot zařízení v řádech jednotek sekund.

6.3 Návrh SW rozhraní TCP – protokol spinel

Rozhraní TCP – protokol spinel se skládá ze dvou aplikací. Jednou částí je aplikace pro navrhované embeded zařízení. Tato část byla napsána v jazyku C++ a přeložena do nativního kódu instrukcí 80186. Další částí rozhraní TCP-spinel je DLL knihovna (Assembly) napsaná jazykem C# v technologii Microsoft .Net Framework.

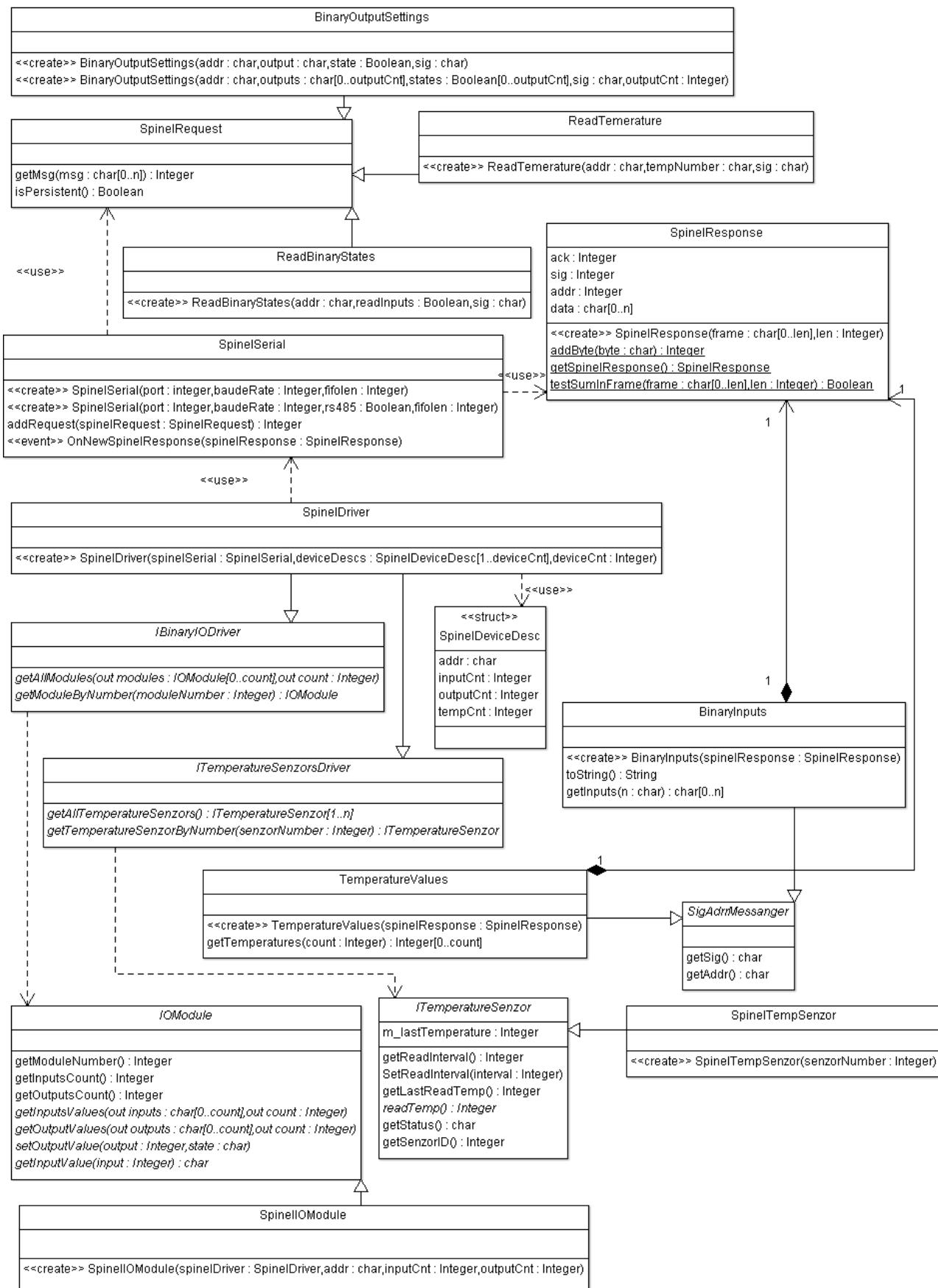
6.3.1 SW aplikace v navrhovaném zařízení

Pro realizaci rozhraní byl vytvořen SW modul Spinel. Tento Modul se stará o implementaci protokolu spinel tak, aby mohl být volně navázán na již navržené moduly, které jsem provedl v návrhu SW architektury.

Obrázek 54 obsahuje analytický model tříd modulu spinel. Významnou třídou modelu je *SpinelDriver*. Tato třída dědí definice abstraktních tříd *IBinaryDriver* a *ITemperatureSenzorDriver*. Toto dědění umožní propojení s povelovacím serverem. V konstruktoru třídy *SpinelDriver* je předána instance třídy *SpinelSerial* a struktury popisující jednotlivá zařízení připojená na komunikační lince. Instance třídy *SpinelSerial* se stará o komunikaci na sériové lince (RS232 i 485). Konstruktor této třídy přejímá parametry sériové linky a maximální délku fronty, sloužící jako vyrovnávací pamět pro zařazené příkazy.

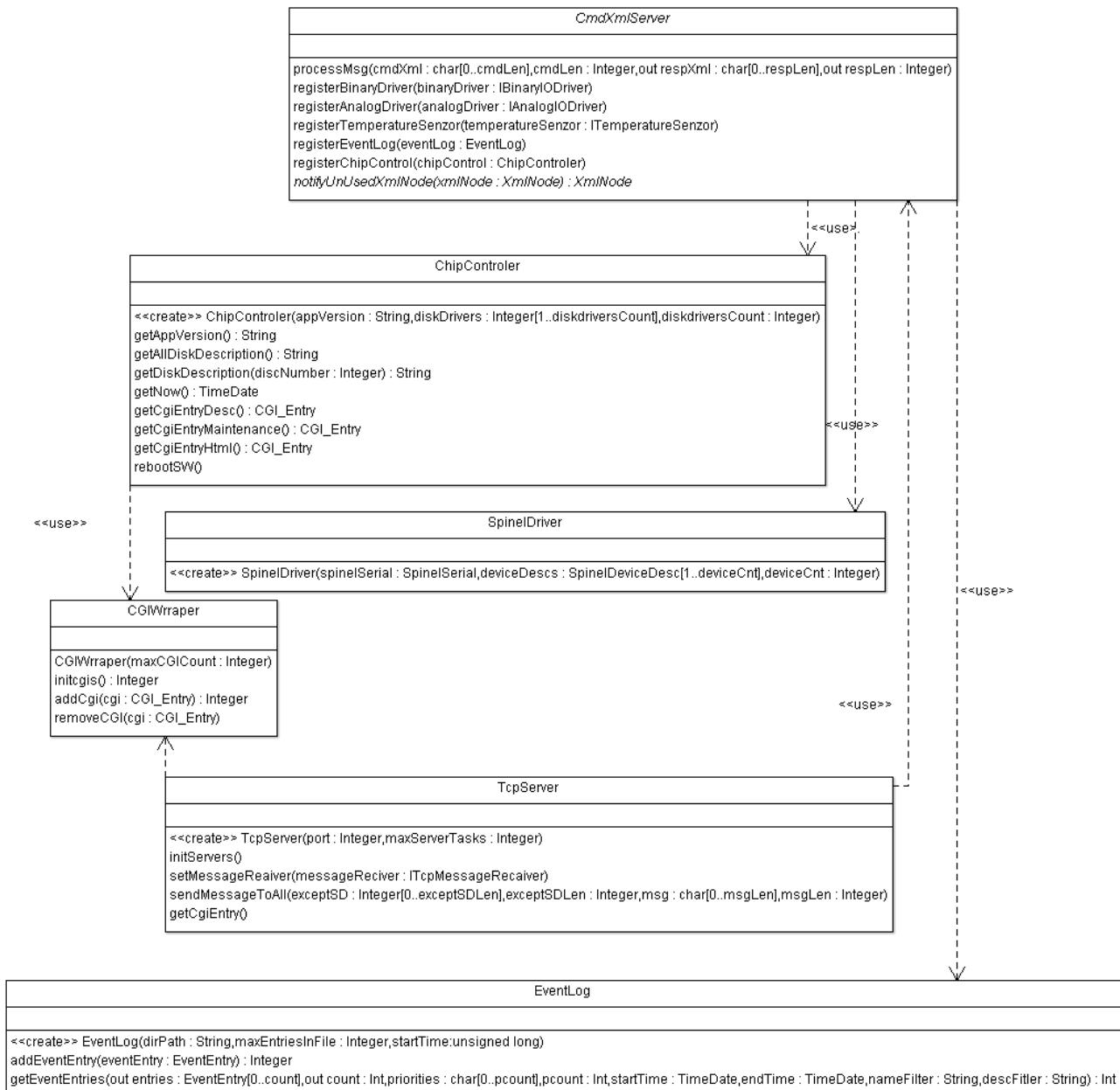
Třída *SpinelIOModul* dědí od *IOModule* a slouží k reprezentaci jednoho IO modulu na komunikační lince. Prostřednictvím této třídy dochází ke čtení a ovládání jednotlivých vstupů.

Třída *SpinelTempSenzor* dědí od třídy *ITemperatureSenzor*. Tato třída slouží pro čtení hodnoty z teplotního senzoru.



Obrázek 54 Analytický model tříd modulu spinel

Třída *SpinelRequest* slouží k definici příkazu poslaného jako příkaz spinel, poskytuje metodu *getMsg*, která vrací data pro odeslání. Od třídy *SpinelRequest* dědí třídy příkazů *BinaryOutputSettings*, *ReadTemperature*, *ReadBinaryStates*. *BinaryOutputSettings* slouží k řízení binárních výstupů. *ReadTemperature* slouží ke čtení teploty. *ReadBinaryStates* realizuje příkaz pro čtení stavů vstupů a výstupů.



Obrázek 55 Analytický model tříd aplikace rozhraní TCP- Spinel

Třída *SpinelResponse* vytváří jednotné rozhraní pro odpovědi binárního formátu 97 protokolu spinel. Dál se stará o vytváření vlastní instance. K tomu používá statické metody *getSpinelResponse*, která vrací instanci odpovědi, pokud byla zadána platná data protokolu spinel prostřednictvím statické metody *addByte*. Odpověď *SpinelResponse* inicializuje v konstruktoru

příslušné specifické odpovědi *BinaryInputs*, *TemperatureValues* dle hodnoty SIG. *BinaryInputs* reprezentuje odpověď stavů vstupů a výstupů. *TemperatureValues* obsahuje informace o hodnotách teploty na jednotlivých teploměrech.

Obrázek 55 obsahuje analytický model tříd aplikace rozhraní TCP-Spinel. V tomto modelu jsou obsaženy třídy, které byly popsány jak v této kapitole, tak v těch dřívějších. Instance třídy *CmdXmlServer* registruje instance *SpinelDriver*, *ChipControler*, *EventLog*. Instance třídy *CGIWrapper* se stará o registraci CGI metod instancí *TcpServer*, *ChipControler*. Jednotlivé příkazy pro instanci *CmdXmlServer* jsou předávány z instance třídy *TcpServer*. V obrázku 55 jsou zahrnuty i moduly, které nejsou nezbytně nutné pro funkci rozhraní TCP- spinel. Jedná se o moduly *ChipControler*, *EventLog*, *CGIWrapper*. Tyto moduly jsem zahrnul, abych ukázal jednoduché propojení těchto modulů a rozšiřitelnost aplikace.

Tabulka 16 ukazuje příkazy pro rozhraní TCP-Spinel a tabulka 17 odpovědi. Příkazy a odpovědi jsou realizovány instancí třídy *CmdXmlServer*. Každý příkaz a odpověď jsou uvozeny ve tvaru:

<STX>příkaz/odpověď<ETX>

Hodnota <STX> je 02h a hodnota <ETX> je 03h. Z toho plyne, že odpovědi budou ve tvaru ASCII.

Příkaz	Formát	Popis
Vrácení vstupů	<getinputs> </getinputs>	Vrátí stavy všech dostupných binárních vstupů
Vrácení výstupů	<getoutputs> </getoutputs>	Vrátí stavy všech binárních výstupů
Vrácení teplot	<gettemperatures></gettemperatures>	Vrátí hodnoty všech teploměrů
Nastavení výstupu	<setoutput drv="DrvName" mod="ModulNum" out="OutputNum"> State </setoutput>	Nastaví binární výstup. <i>DrvName</i> – název driveru <i>ModulNum</i> – číslo modulu <i>OutputNum</i> – číslo výstupu <i>State</i> – stav sepnutí výstu (0,1)

Tabulka 16 Příkazy a odpovědi rozhraní TCP-Spinel

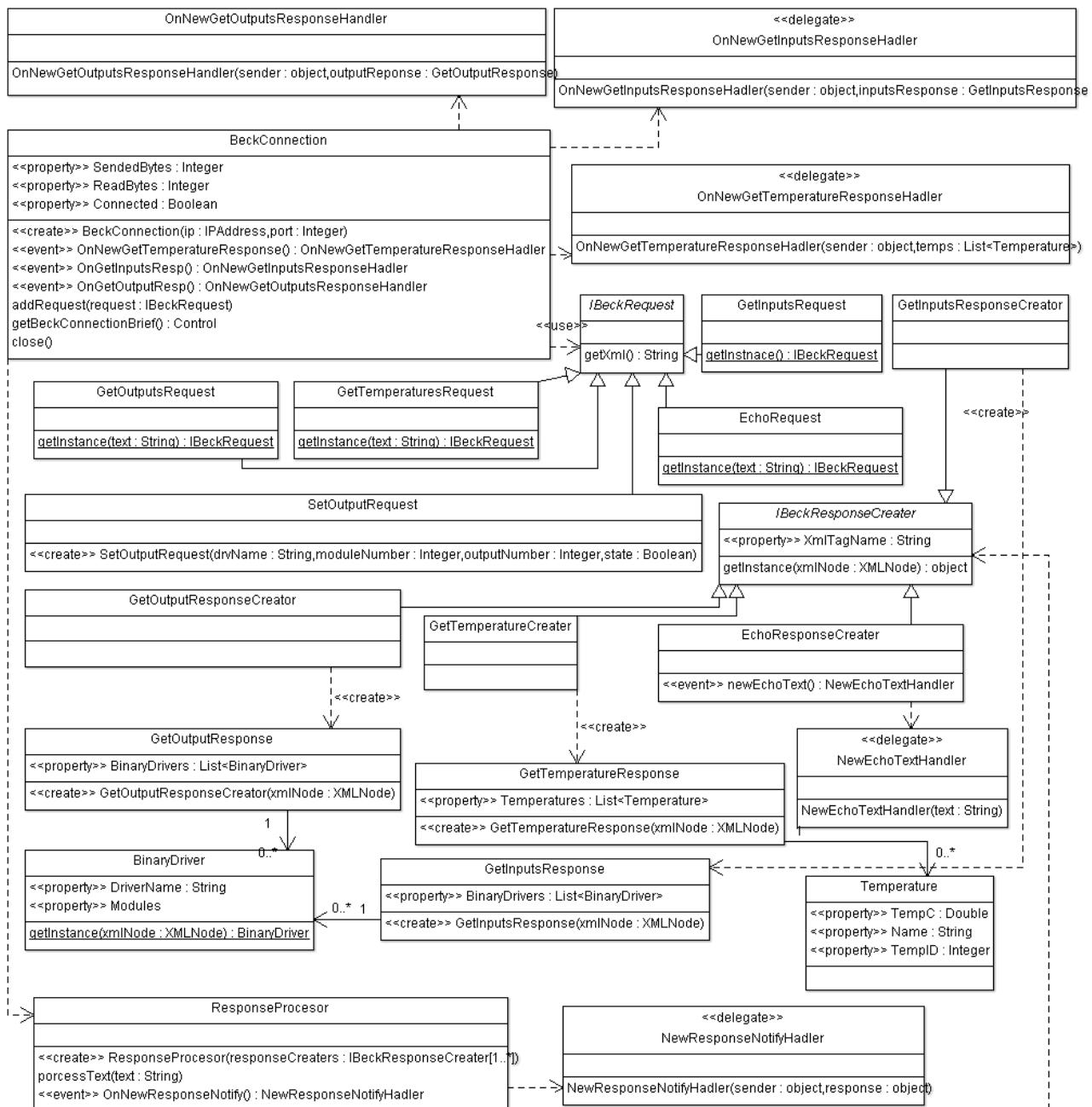
Odpověď	Formát	Popis
Vrácení vstupů	<binInputs> <drv name="DrvName"> <mod number="ModNum"> <i>BinaryStates</i> </mod> ... </drv> ... </binInputs>	Vrací hodnoty binárních vstupů. V odpovědi může být vráceno několik driverů vstupů. V Driveru vstupů může být vráceno několik modulů. <i>DrvName</i> – název driveru binárních vstupů <i>ModNum</i> – číslo modulu <i>BinaryStates</i> – textový řetězec obsahující znaky 1 a 0
Vrácení výstupů	<binOutputs> <drv name="DrvName"> <mod number="ModNum"> <i>BinaryStates</i> </mod> ... </drv> ... </binOutputs>	Vrací hodnoty binárních výstupů. V odpovědi může být vráceno několik driverů výstupů. V Driveru výstupů může být vráceno několik modulů. <i>DrvName</i> – Název driveru binárních výstupů <i>ModNum</i> – číslo modulu <i>BinaryStates</i> – textový řetězec obsahující znaky 1 a 0
Vrácení teplot	<tempSenzors> <senzor name="TempName"> <i>Teplota</i> </senzor> ... </tempSenzors>	Vrací hodnotu teplot na všech dostupných teploměrech Uzel tempSenzors může obsahovat několik uzelů senzor. <i>TempName</i> – Název teploměru <i>Teplota</i> – Hodnota teploty ve stupních Celsia. Formát je xxx.yy Například 26.6

Tabulka 17 Odpovědi rozhraní TCP-Spinel

6.3.2 SW knihovny DLL pro PC

Na obrázku 56 je analytický model knihovny pro připojení k rozhraní TCP – Spinel. Knihovna je navržena tak, aby byla jednoduše použitelná. Proto je zde třída *BeckConnection*, která jejíž funkce je odvozena od návrhového vzoru Fasáda, která odstínuje od vnitřní implementace.

Konstruktor třídy *BeckConnection* přejímá parametry IP adresy zařízení a portu aplikace s *CmdXmlServerem*. Instance třídy *BeckConnection* vrací grafický prvek přehledu stavu I/O, teplot a stavu komunikace prostřednictvím metody *getBeckConnectionBrief*. Události definované v třídě *BeckConnection* slouží pro informování nástavbových objektů o stavu binárních I/O a hodnotách teplot. Metoda *AddRequest* přidá do fronty požadavků požadavek, který implementuje rozhraní *IBeckRequest*. Současná implementace třídy *BeckConnection* posílá dotazy na stavy binárních I/O, teplot, echo automaticky. Metodu *AddRequest* lze v současné implementaci použít pouze pro spínání binárních vstupů.



Obrázek 56 Analytický model tříd knihovny DLL knihovny připojení k rozhraní TCP – Spinel

Všechny požadavky musí implementovat rozhraní *IBeckRequest*. *GetOutputsRequest* slouží pro získání stavů binárních výstupů. Třída *GetInputsRequest* reprezentuje dotaz čtení stavů binárních vstupů. *GetTemperatureRequest* představuje dotaz na stavy teploměrů. Pro nastavení výstupu slouží instance třídy *SetOutputRequest*.

ResponseProcessor je třída sloužící pro zpracování odpovědí od rozhraní TCP-Spinel. Metoda *processText* přejímá parametr přijatého textu z komunikační linky a provede zpracování. Pro zpracování příkazů používá instance tříd implementující *IBeckResponseCreator* a vytvářející příslušné objekty, které reprezentují odpověď.

6.4 Zhodnocení současného zařízení

Povedlo se realizovat zařízení, které ukazuje principy a nastoluje cestu dalšímu vývoji. Analytické modely SW pro PC a navrhované zařízení dovolují snadnou implementaci a neomezují v dalším rozšíření aplikace, které budou nutné, pokud má být zařízení schopno komerčního nasazení.

Dalšími možnými rozšířenými jsou vytvoření plnohodnotného webového serveru, který umožní konfiguraci zařízení a jeho jednoduché ovládání. Automatické zjištění zařízení implementující protokol spinel v binárním formátu 97. Rozšíření povelovacího serveru o příkazy zápisu událostí do logu událostí.

Navržená SW architektura dovoluje implementovat dynamické stránky (stránky s nestatickým obsahem), které mohou poskytnout dostatečné uživatelské rozhraní řízení aplikace. Příkladem může být využití knihovny JQuery nebo JavaApletu pro realizaci pokročilého uživatelského rozhraní.

Současně realizované funkce jsou čtení stavů binárních vstupů, výstupů, čtení teplot, nastavení binárních výstupů. Všechny tyto funkce lze ovládat pomocí realizované knihovny DLL pro PC.

Závěr

Povedlo se realizovat dvě Embedded zařízení, která lze použít v oboru integrace slaboproudých technologií. Obě tato zařízení splňují požadavky z kapitoly definující obor IT integrací slaboproudých technologií.

Prvním zařízením je synchronizace obsahu monitorů, což je zařízení konkrétní funkce bez obecnějšího použití. Toto zařízení již několik měsíců bezchybně funguje na centrálním dispečinku

Rudná nového Pražského okruhu, kde se k zařízení připojuje několik aplikací Integoo Video Client. Od instalace zařízení nebylo ani jednou restartováno a ani nebylo potřeba upravovat HW nebo SW.

Druhým zařízením je rozhraní TCP – Spinel. Realizace tohoto zařízení se skládá z HW a SW. SW splňuje zadání této diplomové práce, ale pro instalace u zákazníků bude potřeba se soustředit na dopracování některých funkcionalit, které jsem rozebíral v příslušné kapitole. V konečné podobě zařízení přinese úsporu nákladů v podobě ušetřené ceny za HW průmyslového PC. Dalším přínosem navrhovaného zařízení je SW architektura, která umožňuje snadné rozšíření o další funkce, drivery I/O atd. V neposlední řadě je výhodou navržený přenos informace protokolem TCP ve formátu XML, a zařízení je tak velice snadno implementovatelné do SW třetích stran.

Přednosti obou zařízení oproti použití PC jsou nepoužití pohyblivých mechanických součástí, především harddisků a větráků, které jsou přítomny v PC a trpí častou poruchovostí, nižší spotrebou elektrické energie. Nezanedbatelnou výhodou je také reboot zařízení v řádech jednotek sekund.

Literatura a použité zdroje

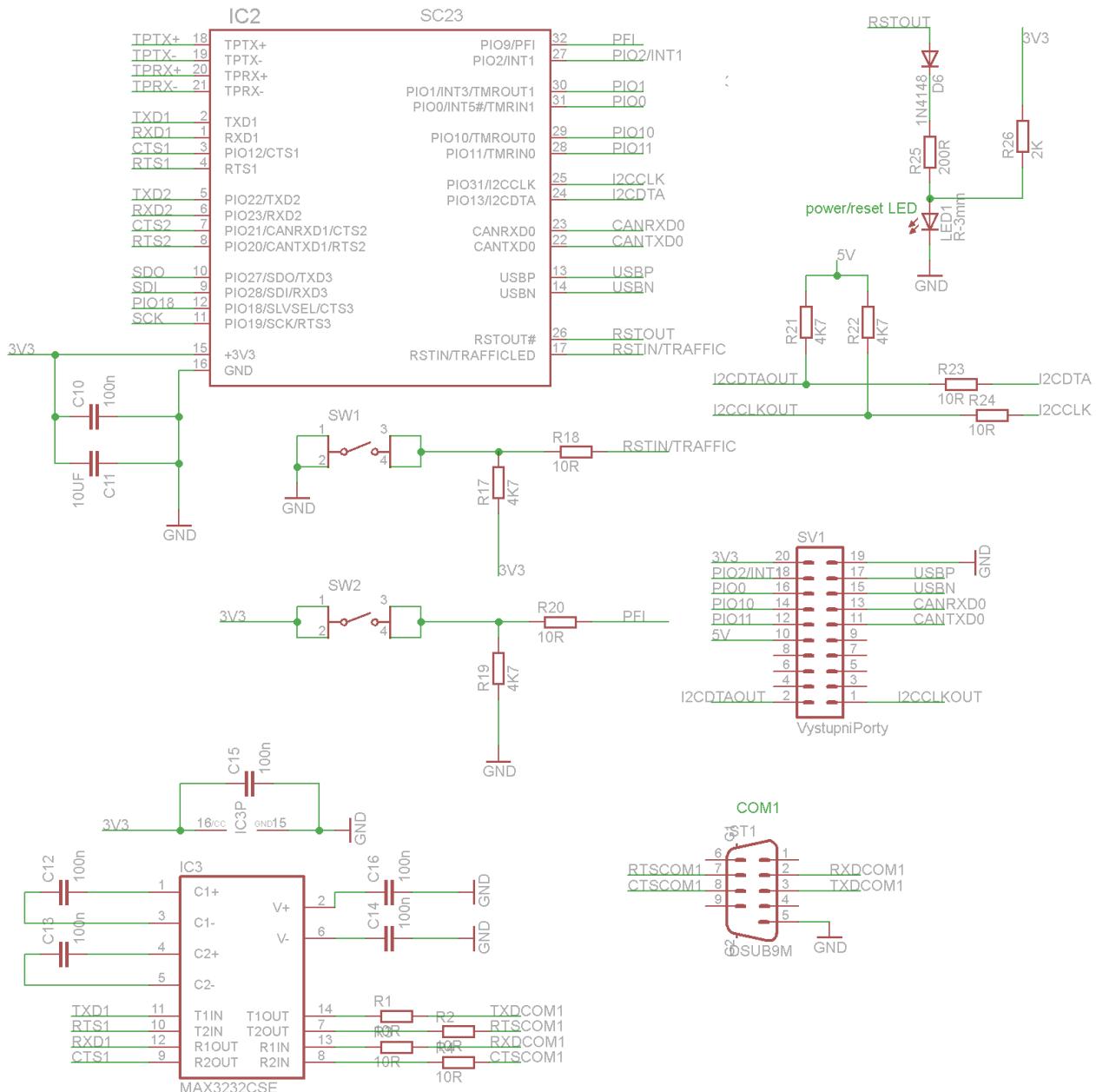
- [1] <http://www.beck-ipc.com> – stránky výrobce procesorů IPC@CHIP.
- [2] Jim Arlow, Ila Neustadt – UML 2 a unifikovaný proces vývoje aplikací
- [3] Rudolf Pecinovský – Návrhové vzory
- [4] Maxim – Datasheets k obvodům řady MAX232 a řady MAX485
- [5] National Semiconductor – Datasheet k obvodu LM2672M-3.3 a k LM2594
- [6] Franka Vanden Bergheh – <http://www.applied-mathematics.net/> - XML parser
- [7] Stránky firmy Papouch – <http://www.papouch.com>

Obsah přiloženého CD

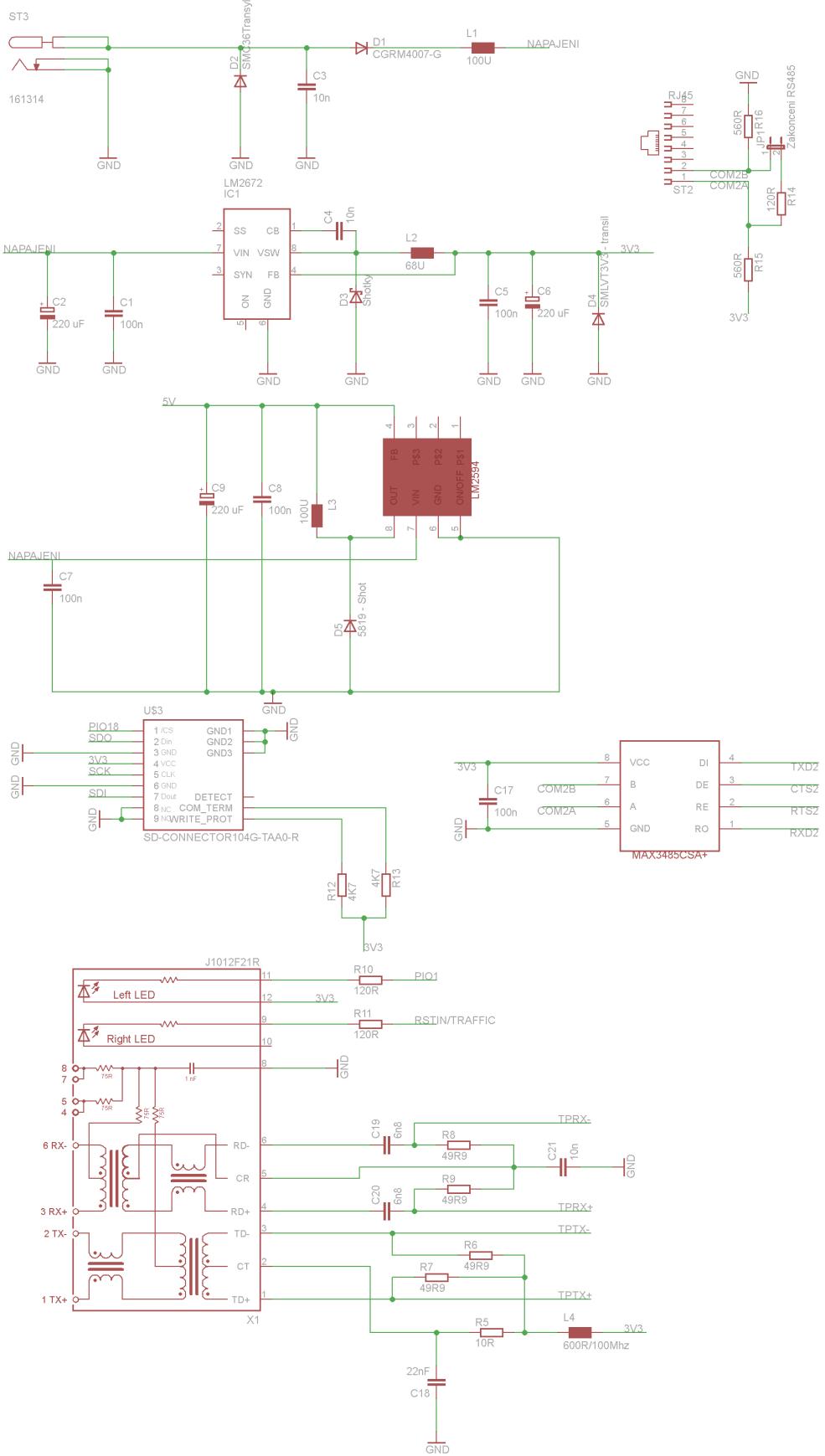
- Elektronická podoba této práce DP_2010_Sedlacek_Vaclav.pdf
- Složka datasheets
 - Datasheets součástek použité v navrhovaném zařízení
- Složka Výroba
 - Výrobní podklady DPS
- Složka Eagle
 - Projekt schémat a DPS v programu eagle
- Složka SW
 - SW projekty synchronizace obsahu monitorů (pro PC i pro DK40)
 - SW projekt rozhraní TCP – Spinel (pro PC i pro SC23)
- Složka Papouch
 - Materiály ze stránek firmy Papouch

Příloha A

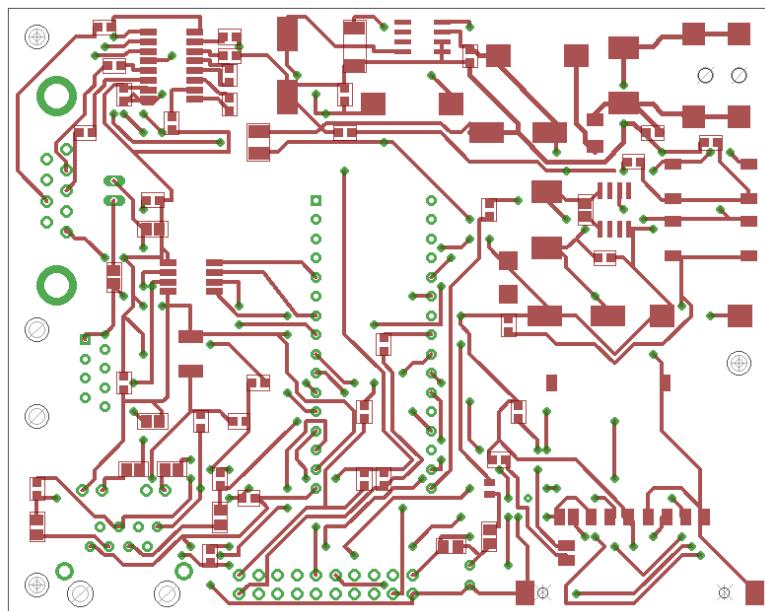
HW dokumentace rozhraní TCP-protokol spinel



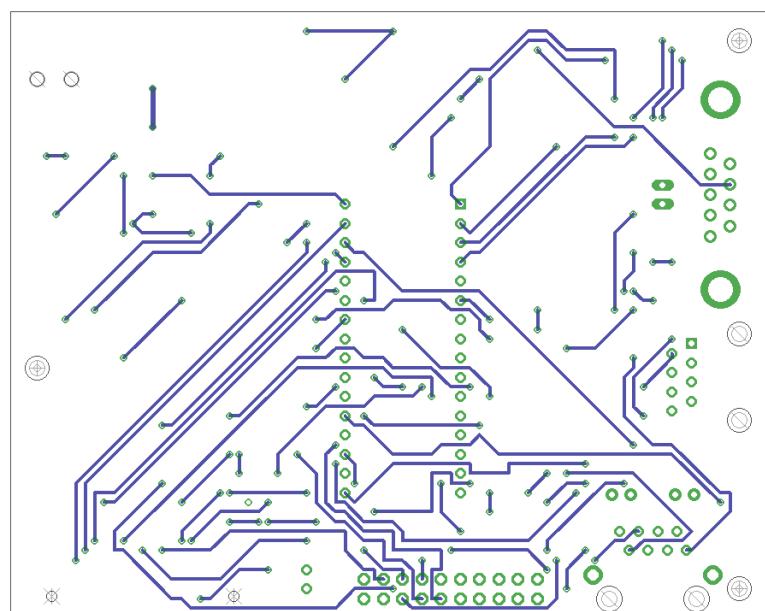
Obrázek A.1a Schéma zapojení rozhraní TCP – Spinel



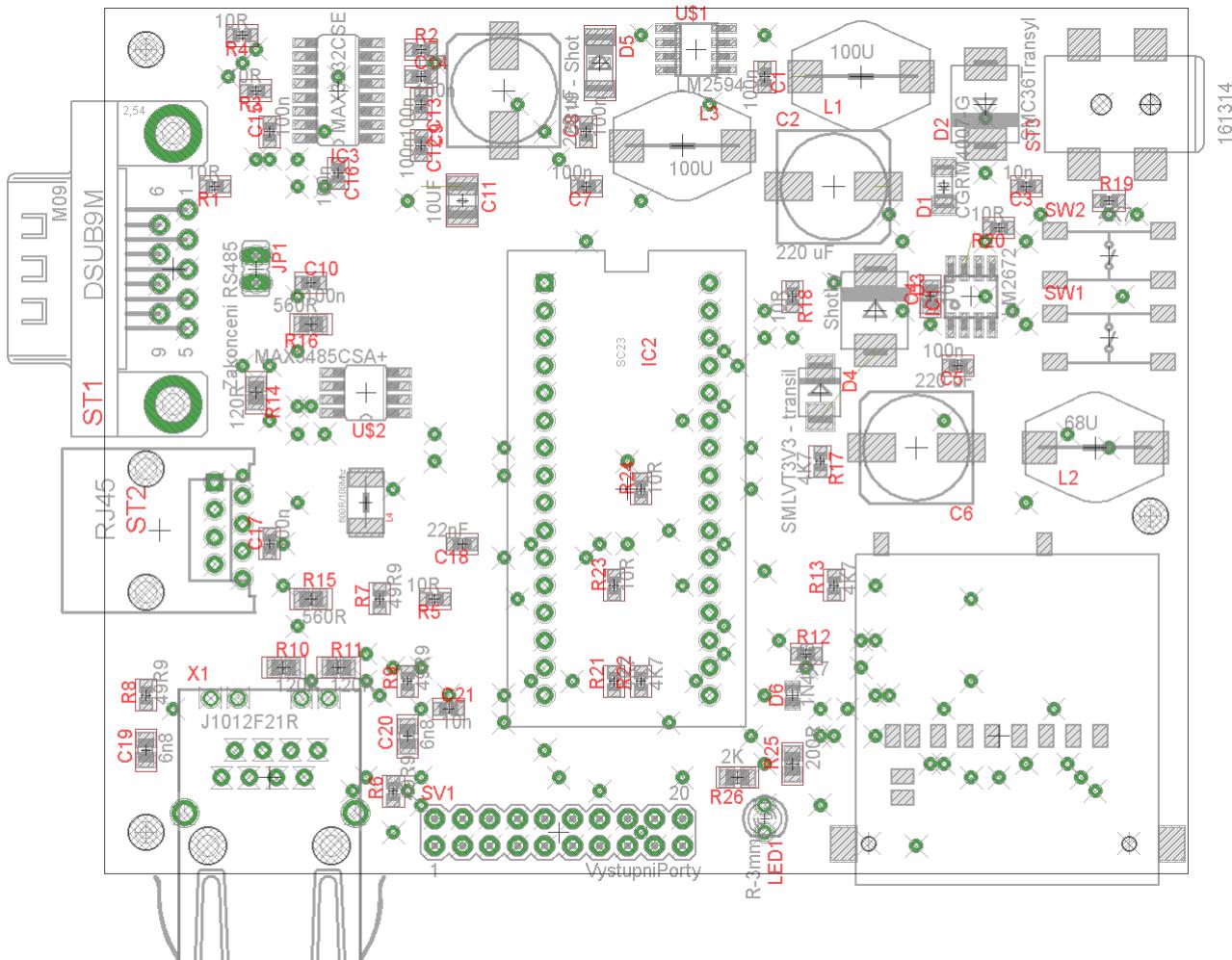
Obrázek A.1b Schéma zapojení rozhraní TCP – Spinel



Obrázek A.2 DPS strana TOP



Obrázek A.3 DPS strana BOTTOM



Obrázek A.4 Rozmístění součástek strana TOP

Součáska	Popis	Umístění	Kusů
LM2672M-3.3	Spinaný zdroj 3,3V lout 1A, SMD	Zdroj	1
Kondenzátor 100n keramický	100NF / 50V / +-10% / 0603 / X7R	C1,C5,C7,C8,C10,C12,C13,C14,C15,C16,C17	11
Elyt. Kon 220 uF	CAPACITOR, SMD, 25V, 220UF	C2,C6,C9	3
Cívka 100uH	case L	L1,L3	2
1N4007 SMD	Usměrňovací dioda	D1	1
SM15T36CA	Transil	D2	1
Kondenzátor keramicky 10n		C3,C21	2
Shottkyho dioda	Schottky dioda 60V 3A DO214AB/SMC	D3	1
cívka 68uH	case L	L2	1
transil	TVS -3V3 -DO214AA -600W	D4	1
spinany zdroj 5V LM2594	Spinaný stabilizátor 5V@0,5A, max. Uin 40V		1
Shottkyho dioda PRLL5819	DIODE, SCHOTTKY, 1A, SOD-87	D5	1
Kondenzátor, 10uF	SMD 1210	C11	1
232 sběrnice		IC3	1
Odpor 10R	0603	R1,R2,R3,R4,R5,R18,R20,R23,R24	9
485 sbernice	8 SO		1
RJ 45 zasuvka	Pouzita pro rs485	ST2	1
Ethernet konektor		X1	1
keramicky kondenzátor 22nF	0603	C18	1
Ferit	FERRITE, SMD 600OHM, 3A, 1812	L4	1
odpor 49R9	0603	R7,R6,R8,R9	4
kondenzátor keramicky 6n8	0805	C19,C20	2
odpor 120R	Rezistor SMD 0805 120R Ohm	R10,R11,R14	3
sd konektor		U\$3	1
odpor 4K7	0603 1%	R12,R13,R17,R19	4
	napajec9 konektro	ST3	1
patice dil32 pro sc23			1
Odpor 560R	0603 1%	R15,R16	2
Mikrospinac		SW1,SW2	2
Kolík lámací lišta – zvýšená		SV1	1
1K rezistor	0603	R21,R22	2
Unicerzalni dioda 1N4148		D6	1
200R odpor	0805	R25	1
LED cervena	3mm	LED1	1
Odpor 2K	0805	R26	1

Tabulka A.1 Seznam použitych součástek