

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INTELIGENTNÍ ZÁSUVKY PRO NAPÁJENÍ SPOTŘEBIČŮ V LABORATOŘI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ADAM CRHA

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

INTELIGENTNÍ ZÁSUVKY PRO NAPÁJENÍ SPOTŘEBIČŮ V LABORATOŘI

INTELLIGENT ELECTRICAL OUTLETS FOR APPLIANCES POWER SUPPLY IN LABORATORY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ADAM CRHA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2011

Abstrakt

Tato práce popisuje návrh a implementaci dálkově řízených inteligentních zásuvek. Inteligentní zásuvky jsou zařízení, které je schopné komunikovat s okolním světem. Hlavní jednotkou zařízení je mikrokontrolér ATMEL ATMega168, který obsluhuje výkonové výstupy a komunikaci s vnějším ovládacím zařízením. Je diskutována možnost rozšíření pro měření spotřeby elektrické energie každé zásuvky. K připojení do sítě Ethernet je využit modul Nano SocketLan od firmy ConnectOne. Komunikace inteligentních zásuvek a ovládacího zařízení je řešena na bázi TCP protokolu.

Abstract

This thesis describes the design and implementation of remotely controlled intelligent outlets. The designed device has ability to communicate with outside world. The device is based on ATMEL ATMega168 microcontroller, which controls high-power outputs and communication with the outer control device. The hardware was designed considering possible extensions for measuring energy consumption of each socket. Ethernet connection is provided by Nano SocketLan module manufactured by ConnectOne. TCP protocol is used for communication between intelligent outlets and outer control device.

Klíčová slova

ATMega, SPI, mikrokontrolér, Nano SocketLAN, TCP, Ethernet, protokol, PDU.

Keywords

ATMega, SPI, microcontroller, Nano SocketLAN, TCP, Ethernet, protokol, PDU.

Citace

Adam Crha: Inteligentní zásuvky pro napájení spotřebičů v laboratoři, bakalářská práce, Brno, FIT VUT v Brně, 2011

Inteligentní zásuvky pro napájení spotřebičů v laboratoři

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka

.....
Adam Crha
15. května 2011

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Václavu Šimkovi za jeho čas, který věnoval a poskytl mi odbornou pomoc při řešení této práce.

© Adam Crha, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	5
2 Přehled vlastností a typů PDU	6
2.1 Základní vlastnosti	6
2.2 Rozdělení	6
2.2.1 Stolní / skříňové	6
2.2.2 Online / Offline	7
2.2.3 Manažovatelné / Nemanadžovatelné	7
2.2.4 Měřicí / Neměřicí	8
3 Ethernet a vestavěné systémy	9
3.1 Ethernet	9
3.1.1 Fyzická vrstva	10
3.1.2 Linková vrstva	10
3.1.3 ISO/OSI Model	12
4 Konceptuální návrh	14
4.1 Blokový návrh a model situace	14
4.2 Konceptuální návrh HW	15
4.2.1 Volba mikrokontroléru	15
4.2.2 Volba ethernetového modulu	17
4.2.3 Volba měřicího modulu	19
4.2.4 Konceptuální návrh základní jednotky	20
4.3 Konceptuální návrh FW	21
4.3.1 Konceptuální návrh FW pro obsluhu periférií	21
4.3.2 Konceptuální návrh síťové komunikace	23
5 Realizace HW	27
5.1 Prostředky použité při realizaci	27
5.1.1 Návrhový systém DPS EAGLE	27
5.1.2 Prostředky pro osazení DPS	27
5.2 Základní jednotka	28
5.2.1 Popis schématu zapojení základní jednotky	28
5.2.2 DPS základní jednotky	29
5.3 Modul spínání a měření	29
5.3.1 Popis schématu zapojení spínacího a měřicího modulu	30
5.3.2 DPS spínacího a měřicího modulu	30

6	Realizace FW	32
6.1	Použité periferie	32
6.1.1	GPIO (General Purpose Input Output)	32
6.1.2	UART (Universal Asynchronous Receiver/Transmitter)	32
6.1.3	SPI (Serial Peripheral Interface)	33
6.2	Implementace firmwaru	34
6.2.1	Hlavní program	34
7	Problémy, testování a další vývoj	37
7.1	Konstrukce	37
7.1.1	Základní jednotka	37
7.1.2	Spínací a měřicí modul	37
7.2	Oživení	38
7.2.1	Programování firmware ATmega168	38
7.2.2	Propojení bloků	38
7.2.3	Testování	38
7.3	Problémy při vývoji	39
7.3.1	Problémová SPI komunikace s Nano SocketLAN	39
7.4	Rozšíření do budoucna	39
8	Závěr	41
A	Normy pro rozvod elektrické energie	46
A.1	ČSN 33 2130 – Vnitřní elektrické rozvody	46
A.1.1	Zásuvkové obvody	46
A.2	ČSN 33 2000-4-41 – Ochranná opatření před úrazem elektrickým proudem	47
A.2.1	Ochrana před nebezpečným dotykem živých částí	47
A.2.2	Ochrana před nebezpečným dotykem neživých částí	48
B	Schémata zapojení	49
B.1	Schéma zapojení základní jednotky	49
B.2	Schéma zapojení spínacího a měřicího modulu	49
C	Osazení desek plošných spojů	52
C.1	Základní jednotka	52
C.2	Měřicí a spínací modul	52
D	Fotografie prototypu	53
D.1	Fotografie prototypu zepředu	53
D.2	Fotografie prototypu zezadu	53
D.3	Fotografie prototypu shora	54
D.4	Fotografie detailu hlavní části	54

Seznam obrázků

2.1	PDU pro umístění na stůl (vlevo) [23], PDU pro montáž do skříně (vpravo)[22].	7
3.1	Ethernetový rámec.	11
3.2	Schéma ISO/OSI modelu [30].	12
4.1	Model aplikace v laboratoři.	15
4.2	Blokové schéma mikrokontroléru ATmega168 [8].	17
4.3	Pohled na Nano SocketLAN od firmy ConnectOne [13].	18
4.4	Blokové schéma hardwaru PDU.	20
4.5	Konečný automat znázorňující komunikaci mikrokontroléru(host) a Nano SocketLAN [24].	22
4.6	Struktura TCP paketu [36]	24
5.1	Vlevo horní vrstva DPS, vpravo spodní vrstva základní jednotky	30
5.2	Vlevo horní vrstva DPS, vpravo spodní vrstva spínacího a měřícího modulu	31
B.1	Schéma základní jednotky	50
B.2	Schéma spínací a měřící části	51
D.1	Pohled na prototyp PDU zepředu.	53
D.2	Pohled na prototyp PDU zezadu.	53
D.3	Pohled na prototyp PDU shora.	54
D.4	Detail hlavní části prototypu PDU.	54

Seznam tabulek

4.1	Aplikační protokol PDU.	25
4.2	Popis příkazů aplikačního protokol PDU.	25
4.3	Popis odpovědí aplikačního protokol PDU.	26

Kapitola 1

Úvod

Obyčejnou zásuvku elektrického spotřebiče najdeme dnes prakticky kdekoliv. Z důvodů bezpečnosti, nebo úspory energie dnešních „stand-by“ spotřebičů, je vhodné mít zásuvky spínatelné. Nejjednodušší řešení je použití běžného kolébkového přepínače „vypnuto/zapnuto“. Toto řešení je levné a obvykle dostačující, avšak v některých případech velmi nepraktické. Ukázkovým příkladem může být řízení přívodu napájení serveru v místě A. Srázce sítě se nachází na vzdáleném místě B. S využitím dnešních technologií je možné navrhnout vestavěný systém, který bude podporovat vzdálené a přenositelné řízení zásuvek elektrických spotřebičů. Na trhu jsou již dnes hotové výrobky, které by splňovaly podobné požadavky, nevýhodou je vysoká cena.

V této práci se budeme zabývat realizací inteligentních zásuvek pro laboratoř. Cílem je navrhnout zařízení PDU (Power Distribution Unit), dále také jako zásuvková lišta, které bude připojitelné z jednoho konkrétního místa (centrální ovládací systém, dále COS - bakalářská práce studenta Jana Martišky¹ [26]). Výsledkem bude řízení přívodu elektrického proudu v laboratoři ke každému pracovišti a jeho konkrétnímu spotřebiči (počítač, monitor, osciloskop, ...).

Obsahem práce tedy bude návrh hardwaru a implementace firmwaru PDU, následně pak návrh aplikačního protokolu pro komunikaci s COS. Absenci COS lze nouzově nahradit použitím jednoduchého TCP klienta.

Práce je členěna do několika kapitol. První tři teoreticky zaměřené kapitoly se věnují přehledu PDU na trhu, standardu Ethernet a jeho použití ve vestavěných systémech a následně konceptuálnímu návrhu systému. Další kapitoly zmiňují hardwarovou realizaci, implementaci firmwaru PDU a nakonec testování prototypu, možnosti rozšíření a závěr.

¹xmarti27@stud.fit.vutbr.cz

Kapitola 2

Přehled vlastností a typů PDU

Tato kapitola popisuje stručný, ale sjednocený přehled jednotek PDU. Diskutuje základní vlastnosti a typy PDU.

2.1 Základní vlastnosti

PDU je přístroj, který slouží k distribuci dodávky elektrické energie do několika zařízení, jehož přívod může být řízen na základě volby uživatele. Typicky se PDU využívá ve speciální skříni pro servery, kde je požadavkem mít přívod energie do serveru pod kontrolou.

Základní standard pro zařízení PDU je totožný s normou „ČSN 33 2130 – Elektrické instalace nízkého napětí – Vnitřní elektrické rozvody“ [39], která charakterizuje zásady elektroinstalace. Další norma „ČSN 33 2000-4-41 – Elektrické instalace nízkého napětí – Ochranná opatření před úrazem elektrickým proudem“ [38], definuje bezpečnosti těchto zařízení. Výtahy norem jsou uvedeny v příloze A.

PDU umožňuje přidělování energie do jednotlivých elektrických zařízení. Termín PDU se dnes používá pro řízené zásuvky, které jsou pod správou uživatele. To znamená, že uživatel může definovat, která zásuvka bude připojená do sítě elektrické energie a která ne. S obrovským pokrokem vědy a technologií je možné sledovat stav každé zásuvky, včetně spotřeby elektrické energie.

2.2 Rozdělení

PDU zařízení je možné rozdělit do kategorií na základě jejich vlastností. *Stolní/skříňové (rack), manažovatelné/nemanažovatelné, online/offline, měřící/neměřící.* Moderní systémy většinou mají tyto vlastnosti: online, manažovatelné a měřící současně.

2.2.1 Stolní / skříňové

Stolní PDU jsou konstruovány taky, aby je bylo možné umístit na téměř jakoukoliv rovnou plochu. Kupříkladu položení na stůl. Jejich podstavy jsou vybaveny protiskluzovými nožičkami. Ve většině případů jsou boxy PDU vybaveny sadou děr, které umožňují individuální přichycení závislé na situaci úchyty.

PDU pro montáž do serverové skříně jsou mnohem více rozšířené, nežli stolní. Důvodem je právě využívání řízených PDU zejména v serverových zařízeních. Tyto boxy jsou uzpůsobeny tak, aby je bylo možné nainstalovat do serverové skříně. Všechny jejich periferie

jsou vyvedeny na přední panel, aby byly dobře dostupné. Rozměry jsou uváděny v palcích. Jsou uzpůsobeny pro svislou i vodorovnou montáž. Tyto PDU jsou většinou *online* a *manažovatelné*.



Obrázek 2.1: PDU pro umístění na stůl (vlevo) [23], PDU pro montáž do skříně (vpravo) [22].

2.2.2 Online / Offline

Online PDU jsou zařízení, které mají možnost připojení do sítě ethernet. Tuto vlastnost dnes najdeme téměř v každé řízené PDU. Někdy bývají nazývány jako „zásuvky s IP adresou“. Mají obrovskou výhodu v tom, že jsou spravovatelné, či sledovatelné prakticky z jakéhokoliv místa s připojením k síti internet. Tato vlastnost tak dává dálkově řízeným PDU smysl. Typů fyzických připojení je několik. Nejrozšířenější je metalické spojení kabelem UTP a konektorem RJ45. Méně rozšířenými (dnes už prakticky nesehnatelné) jsou připojení do sítě ethernet pomocí BNC konektoru a koaxiálního kabelu. Tento typ připojení byl aktuální před několika desítkami let. Díky dnešním bezdrátovým technologiím se na trhu začínají objevovat PDU s technologií Wi-Fi. Díky připojení do sítě ethernet je možné ovládat, či sledovat stav PDU. PDU po síti pak komunikuje pomocí TCP/IP protokolu, telnetu, nebo nabízí webové ovládací rozhraní.

Offline PDU systémy nemají možnost připojení do počítačové sítě. Jejich správa je pak možná pouze z těsné blízkosti. Tyto PDU mají většinou fyzické připojení typu RS232 a jsou spravované například terminálem, či jinou obslužnou aplikací v počítači. Za offline systémy můžeme považovat i obyčejný zásuvkový rozbočovač.

2.2.3 Manažovatelné / Nemažovatelné

Manažovatelné PDU umožňují řídit PDU na základě požadavků uživatele. Nemažovatelné nikoliv. Někdy se vlastnost řízení PDU uživatelem může považovat za samozřejmost a v různých specifikacích produktu můžeme nalézt takto řízenou zásuvku s parametry jako nemažovatelnou. Důvodem je právě to, že manažovatelné PDU by měly umožnit uživateli nastavit více vlastností PDU, než pouze zapnout/vypnout. Příkladem může být nastavení zpoždění mezi zapnutím jednotlivých zásuvek. Toto je důležitá vlastnost požadovaná u spínaných zdrojů spotřebičů, jejichž nárazový proud je několikanásobně vyšší než ustálený

proud. Při sepnutí všech zásuvek v jeden okamžik by tak došlo k výpadku proudového chrániče.

2.2.4 Měřicí / Neměřicí

Měřicí PDU mají navíc vlastnost měření spotřeby elektrické energie. Tato vlastnost dnes bývá většinou nadstandardem a důvodem pro vyšší nákupní cenu. PDU obsahuje integrovaný obvod. K měření se používá transformátor napětí a transformátor proudu, nebo dělič napětí. Integrovaný obvod dokáže měřit pouze na malém rozsahu měřeného napětí, proto je nutné měřené veličiny upravit právě do tohoto rozsahu. K tomu je využito transformátorů. Vysoké napětí a elektrický proud se tak upraví na menší napěťové a proudové úrovně. Tímto způsobem dojde jak ke zmenšení úrovně napětí a proudu, tak i ke galvanickému oddělení nízkého stejnosměrného napětí od vysokého střídavého napětí. Transformované napětí a proud je sledován již zmíněným speciálním integrovaným obvodem, který je již schopný poskytnout aktuální hodnotu spotřebované energie.

Další metodou měření je měření na rezistoru. Využíváno je tak rezistorového děliče napětí. Tato metoda není úplně nejvhodnější, jelikož nevzniká galvanické oddělení od měřeného obvodu a integrovaného obvodu. Galvanické oddělení se následně musí řešit pomocí optočlenů na logických vodičích používaných pro komunikaci s integrovaným obvodem. Informaci o aktuální spotřebě dále distribuuje uživateli příslušným způsobem. Jestliže se jedná o zařízení typu *offline*, je aktuální spotřeba zobrazená na displeji instalovaném v PDU. Pokud se jedná o zařízení *online*, PDU může zasílat naměřené údaje pomocí sítě ethernet.

Kapitola 3

Ethernet a vestavěné systémy

Věda a technika se v posledních letech vyvíjí obrovskou rychlostí. Její nasazení je možné vidět opravdu všude. Personální počítač byl počátkem pro vzestup těchto technologií. Počítač dnes dosahuje obrovských škál použití. Takový počítač je možné nalézt v elektrárně na řízení složitých systémů, nebo i v domácnosti. Lze hovořit o univerzálním počítači. Použití personálního počítače je některých případech velmi nevhodné. Představte si, že na ovládání výtahu byste byli nuceni spustit počítač a nastavit, aby vás odvezl do třetího patra. Zde se staví na scénu vestavěné systémy, které se dnes používají k automatizovanému řízení periférií, kde není třeba vysokého výkonu, jak je tomu u personálního počítače. Tyto vestavěné systémy vynikají právě malými rozměry a nízkou cenou.

Vestavěný systém je typicky jednoúčelová aplikace k předem definovanému účelu. Vestavěné systémy jsou dnes prakticky všude, ačkoli je nevidíme, například v pračce nebo v mikrovlnné troubě. Takový systém většinou obsahuje mikroprocesor, který řídí činnost celého zařízení.

3.1 Ethernet

Trendem současnosti je rozvoj komunikace pomocí internetu. Internet je počítačová síť spojená kabelem, či bezdrátovou technologií. Bezdrátové technologie procházejí značným vývojem, oproti kabelovému rozhraní Ethernet, které je stabilní a spolehlivé a zkoumá se pouze větší šířka pásma. S rozvojem těchto síťových technologií je snaha integrovat síťové prvky přímo do vestavěných zařízení. Připojení pomocí Ethernetu umožňuje komunikaci v malých lokálních počítačových sítích, ale i v rozsáhlých sítích WAN¹ a internetu. Díky těmto technologiím je vzájemná komunikace a propojení možné na obrovské vzdálenosti.

Ethernet byl na počátku koncipován pro spojení několika počítačů méně rozsáhlých sítí. Jeho použití se natolik rozšířilo, že je rozhraní ethernet integrováno v každém personálním počítači. Ethernet je popsán standardem IEEE 802.3, který definuje elektrické a mechanické vlastnosti spojení mezi propojenými zařízeními. Tento standard popisuje princip funkce fyzické vrstvy ISO/OSI modelu.

V pokračujících podkapitolách jsou rozebrány jednotlivé vrstvy modelu ISO/OSI modelu, které je nutné implementovat v řadičích Ethernetu. Řadiče mohou být reprezentovány jako samostatné jednotky, nebo jsou integrovány jako periférie mikrokontroléru. První dvě vrstvy ISO/OSI modelu bývají většinou implementovány přímo v hardwaru řadiče [31].

¹WAN – Wide Area Network

3.1.1 Fyzická vrstva

Fyzická vrstva je první vrstva ISO/OSI modelu síťové architektury a zajišťuje fyzickou komunikaci. Základní funkcí fyzické vrstvy je spojení mezi dvěma, či více síťovými uzly. Fyzická vrstva definuje typ rozhraní, význam komunikačních elektrických signálů, napěťové úrovně těchto signálů, časování, synchronizaci apod. Tato vrstva zajišťuje obsluhu přenosových prostředků. Přenosové prostředky jsou následně spojeny s fyzickým rozhraním. Fyzické rozhraní je pak využito k přenosu jednotlivých datových informací, bitů. Tyto bity jsou převedeny na elektrický signál a přenášeny fyzickým médiem, typicky kabelem. Pro linkovou, tedy druhou vrstvu je fyzická vrstva nezbytná, jelikož udržuje fyzické spojení [28].

Hlavní funkce poskytované fyzickou vrstvou:

- Navazování a ukončování spojení.
- Spolupráce na efektivním rozložení všech zdrojů mezi všechny uživatele.
- Modulace neboli konverze digitálních dat na signály používané přenosovým médiem (a zpět) (A/D, D/A převodníky).

Velmi rozšířeným typem fyzického média je tzv. 10Base-TX, 100Base-TX a 1000Base-TX. Tento typ média využívá kabelu UTP, jehož standard je definován v normě ANSI / TIA / EIA-568-A / B . UTP kabel obsahuje 8 vodičů, které jsou po dvojicích spárovány. Spárované vodiče se nazývají jako diferenciální kroucená dvoulinka. Diferenciální kroucená dvoulinka se používá pro eliminaci rušení. Dle standardu jsou kabely rozděleny do několika kategorií, kde kategorie určuje kvalitu kabelu [27].

3.1.2 Linková vrstva

Linková vrstva je vrstva následující hned po fyzické vrstvě. Tato vrstva je rozdělena na dvě podvrstvy, které jsou následovně popsány.

MAC vrstva

MAC vrstva, neboli Media Access Control je umístěna hned nad vrstvou fyzickou. To proto, že již využívá služby, které poskytuje právě fyzická vrstva. Tato vrstva je druhou podvrstvou linkové vrstvy. MAC vrstva definuje adresní a kontrolní systém, který je potřebný k přístupu k jednotlivým zařízením, ty jsou v takzvané multipoint topologii. Multipoint topologie je realizována pomocí sítě LAN², či MAN³.

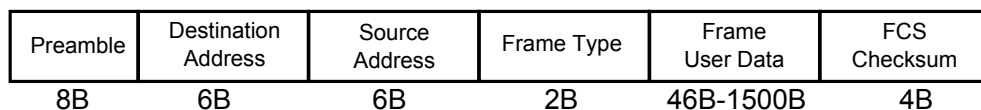
MAC vrstva poskytuje tzv. MAC adresu. Adresu tvoří 48bitů dlouhé číslo a je to jedinečný identifikátor síťového rozhraní. Tyto adresy přiřazuje asociace IEEE. Ačkoli dnešní nástroje umožňují MAC adresu změnit, v rámci jednoho síťového segmentu musí být MAC adresa unikátní. Při komunikaci je v hlavičce datového ethernetového rámce uváděna právě MAC adresa jako cílová adresa příjemce a odesílatele.

Na obrázku 3.1 můžeme vidět ethernetový rámec. Takový rámec obsahuje preambuli (preamble), která nese informaci o synchronizaci. Další položkou jsou zmiňované MAC adresy. První je přenášena cílová adresa (destination address) a za ní následuje zdrojová (source address). Další položkou je typ rámce (frame type). Poté jsou přenášeny data (user data) a nakonec kontrolní součet (FCS).

Základní funkce MAC vrstvy [28]:

²LAN – Local Area Network

³MAN – Metropolitan Area Network



Obrázek 3.1: Ethernetový rámeček.

- Fyzické adresování.
- Řízení přístupu k médiu.

Metody přístupu k médiu:

- Deterministické: je možné nastavit maximální časový interval, kdy se pracovní stanice dostane k médiu.
 - Token passing (předávání peška) - Přístup k médiu je řešen principem předávání peška.
 - Polling (řízený přístup) - řídicí arbitr určuje, který rozhoduje o přístupu k médiu.
- Stochastické (contention = soupeření): nelze zaručit časový interval, takže stanice musí čekat.
 - CSMA/CD (Carrier Sense Multiple Access/Collision Detection) - stanice naslouchá a jestliže je linka volná, může vysílat. Pokud nastane situace, že začnou vysílat dvě zařízení zároveň, vznikne kolize. Stanice vyšle kolizní zprávu (JAM PATTERN). Ostatní stanice po obdržení zprávy o kolizi budou ignorovat přijatý rámeček. Poté se situace opakuje.
 - CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) - pokud je médium volné, započne přenos dat. Neprovádí se zjišťování kolizí. Dojde-li ke kolizi, zkolidovaný rámeček se celý odvysílá a je nutné jej poslat znovu. Metoda není efektivní.

LLC vrstva

LLC vrstva⁴ je druhou podvrstvou linkové vrstvy nadřazenou podvrstvě MAC.

Základní funkce LLC vrstvy [29]:

- Přepínání protokolů vysílaných na MAC vrstvu.
- Řízení toku dat a korekce chyb.
- Není závislá na typu použité sítě.

Řízení toku:

- Jednotlivé potvrzování (Stop & Wait).
- Potvrzování s návratem (Go-Back-N).
- Potvrzování se selektivním opakováním (Selective Repeat).

⁴LLC – Logical Link Control

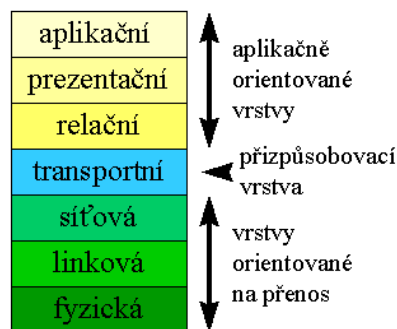
Detekce a korekce chyb:

- Kontrolní součty (CRC).
- Detekce chyb pomocí parity (jednoduchá lichá/sudá, křížová, ...).
- Hammingovy kódy.

3.1.3 ISO/OSI Model

Model ISO/OSI je referenční komunikační model označený zkratkou slovního spojení (International Standards Organization / Open System Interconnection), neboli Mezinárodní organizace pro normalizaci / propojení otevřených systémů. Je to doporučovaný model definovaný organizací ISO v roce 1983. Model rozděluje vzájemnou komunikaci mezi počítači do sedmi souvisejících vrstev.

Úkolem každé vrstvy je poskytovat služby následující vyšší vrstvě a nezatěžovat vyšší vrstvu detaily o tom, jak je služba ve skutečnosti realizována. Než se data přesunou z jedné vrstvy do druhé, rozdělí se do paketů. V každé vrstvě se pak k paketu přidávají další doplňkové informace (formátování, adresa), které jsou nezbytné pro úspěšný přenos po síti [30].



Obrázek 3.2: Schéma ISO/OSI modelu [30].

Obrázek 3.2 znázorňuje ISO/OSI model a jeho vrstvy (každá vyšší vrstva využívá funkce nižší vrstvy).

Model ISO/OSI [30]:

1. **Fyzická vrstva:** Specifikuje prostředky pro komunikaci s technickými prostředky a s přenosovým médiem rozhraní. Také definuje fyzické, elektrické, mechanické a funkční parametry týkající se fyzického propojení jednotlivých zařízení.
2. **Linková vrstva:** Specifikuje spojení mezi dvěma síťovými uzly. Řadí data z fyzické vrstvy do logických celků zvaných rámce. Seřazuje přenášené rámce. Nastavuje parametry přenosu. Detekce neopravitelných chyb. Poskytuje také synchronizaci pro fyzickou vrstvu.
3. **Síťová vrstva:** Poskytuje protokoly pro směrování a adresování dat, jejichž prostřednictvím je zajištěn přenos informací do vzdáleného cílového uzlu, který není ve stejné síti jako zdrojový uzel. V lokální síti vůbec nemusí být využita, nepoužívá-li se směrování.

4. **Transportní vrstva:** Vrstva nabízí spojově (TCP) a nespojově orientované (UDP) protokoly.

TCP - Zajišťuje přenos dat se zárukou, kde se nesmí ztratit žádná informace. Jde zejména o přenosy souborů, emailů, WWW stránek apod. TCP protokol řeší ztráty přenášených paketů, zachování jejich pořadí a odstranění duplikace.

UDP - Zajišťuje přenos dat bez záruky, kde není kritická ztráta paketu. Tento bezzárukový přenos vede zejména ke snížení kvality, opakování dotazu. Využívá se pro DNS, VoIP, streamované video, internetová rádia, online hry aj.

5. **Relační vrstva:** Úkolem vrstvy je organizovat a synchronizovat dialog relací tak dlouho, dokud je potřebný. Umožňuje vytvoření a ukončení spojení mezi relacemi, jejich synchronizaci a obnovení spojení. Dále zajišťuje zabezpečovací, přihlašovací a správní funkce.
6. **Prezentační vrstva:** Vrstva transformuje data do tvaru, který používají aplikace (šifrování, konvertování, komprimace). Formát dat se může lišit na obou komunikujících systémech, navíc dochází k transformaci pro účel přenosu dat nižšími vrstvami. Mezi funkce patří např. převod kódů, modifikace grafického uspořádání, přizpůsobení pořadí bajtů apod. Vrstva se zabývá jen strukturou dat, ale ne jejich významem, který je znám jen vrstvě aplikační.
7. **Aplikační vrstva:** Definuje komunikaci aplikací využívající počítačovou síť, například databázové systémy, elektronická pošta. Používá služby předchozích vrstev a díky tomu je izolována od problémů síťových technických prostředků.

První a druhá vrstva je realizována v řadiči Nano SocketLAN. Třetí až šestou vrstvu zajišťuje firmware mikrokontroléru iChip 2144. Sedmá vrstva je zajišťována mikrokontrolérem ATmega168, kde je implementován aplikační protokol.

Kapitola 4

Konceptuální návrh

Úkolem systému, lehce zmíněného v úvodu, je vzdálené řízení elektrických zásuvek, jakožto dálkové řízení přívodu elektrického proudu do elektrického zařízení. Jedná se tedy o vestavěný systém, který bude schopný řídit zásuvkové rozvody 230 V.

Přidělovat a kontrolovat stav přívodu proudu do elektrického zařízení manuálně je vhodné, pokud počet elektrických zařízení nepřekročí únosnou mez. Díky dnešním technologiím můžeme řídit a sledovat stav přívodu proudu vzdáleně za pomoci vhodného komunikačního rozhraní. Velmi oblíbenou technologií je počítačová síť, jelikož umožňuje připojení téměř z každého místa s připojením do sítě internet. Dnes je nejoblíbenější rozhraní ethernet, protože prošel značnou částí vývoje. Do popředí se však pomalu dostává rozhraní Wi-Fi, kterým se nebudeme v této práci zabývat. Vestavěný systém, s možností připojení do počítačové sítě, nabízí plnou kontrolu nad PDU.

Navrhovaný systém řízení a sledování zásuvek pro napájení spotřebičů je navrhován jako vestavěný systém. Práce se tedy skládá několika fází, kterými jsou návrh a realizace hardwaru, návrh a implementace firmware řídicího mikrokontroléru, návrh aplikačního protokolu a testování. V poslední fázi, testování, je předpokládáno, že budou zjištěny nedostatky a celý proces se bude opakovat od konceptuálního návrhu.

4.1 Blokový návrh a model situace

Reálný systém bude použit v laboratoři. Řízených PDU tedy bude několik. Všechny PDU v laboratoři budou zapojeny do sítě ethernet a budou řízeny z COS¹ - řídicí aplikace s GUI². Model situace tedy bude obsahovat tyto základní prvky:

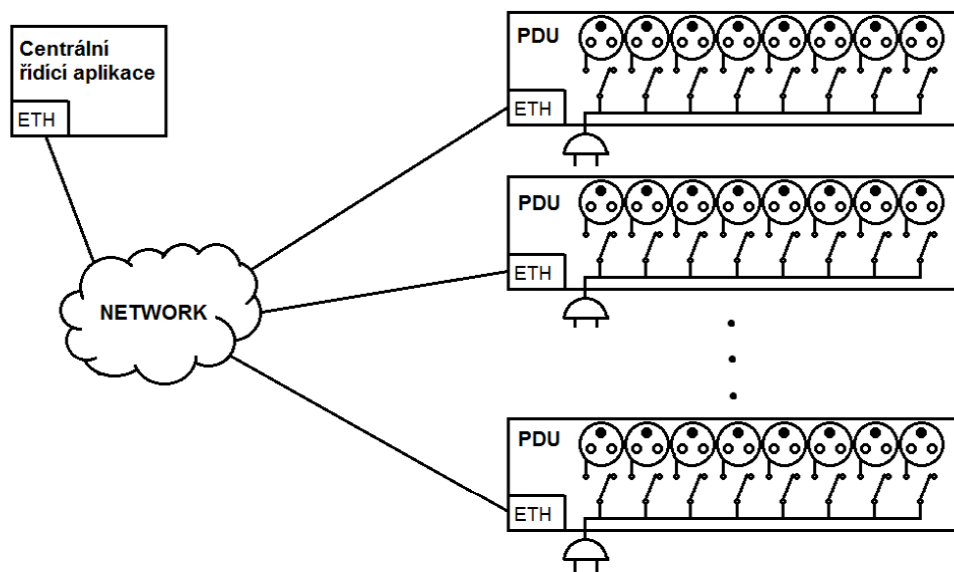
- PDU – vestavěný systém.
- Centrální řídicí počítač s vhodnou obslužnou aplikací.

Na obrázku 4.1 je zobrazen model situace zapojení v laboratoři.

Jak je zřejmé z obrázku 4.1, komunikace PDU s COS bude založena na standardu ethernet s využitím protokolu TCP/IP. Jednotlivé bloky by měly disponovat jednotným hardwarovým síťovým rozhraním a stejným komunikačním protokolem.

¹COS – Centrální Ovládací Systém

²GUI – Graphic User Interface



Obrázek 4.1: Model aplikace v laboratoři.

4.2 Konceptuální návrh HW

K navrhovanému systému je vhodné dobře zvolit základní komponenty, abychom jednak nepořídili komponentu, která nebude poskytovat potřebné vlastnosti, ale také abychom zbytečně nepořizovali komponentu, která by zbytečně měla vlastnosti, které nevyužijeme. Zařízení bude obsahovat mikrokontrolér, ethernetový modul (schopný realizovat TCP/UDP spojení) a měřící modul (rozšíření do budoucna). Dále pak ostatní potřebné součástky.

4.2.1 Volba mikrokontroléru

Mikrokontrolérů je na trhu obrovské množství, proto budeme brát ohled především na tyto důležité požadavky: kvalitní IDE (vývojové prostředí), potřebné rozhraní a periferie, podporu mikrokontroléru výrobcem a také cenu. Mikrokontrolér je vhodné zvolit uvážlivě. Požadujeme, aby zejména splňoval tyto vlastnosti:

- **SPI rozhraní** – většina hardwarových integrovaných logických obvodů komunikuje právě pomocí sériového rozhraní SPI. Toto rozhraní využijeme pro spojení s ethernetovým modulem a měřicími moduly.
- **UART rozhraní** – toto rozhraní má téměř každý počítač (RS-232), umožňuje tak připojení ladícího terminálu.
- **8-bitový registr** – k řízení zásuvek bude třeba V/V registr na relé zásuvek, v případě připojení měřících modulů bude třeba SPI chipselect.
- **I2C rozhraní** (není nutné) – možnost připojení externí paměti.
- **Další V/V signály** – připojení dodatečných řídicích signálů, například interrupt.
- Dostatečnou velikost paměti programu ≥ 8 kB.

- Dostatečnou paměť dat ≥ 1 kB.
- **SMD pouzdro** – díky malému pouzdru vznikne úspora místa na DPS \rightarrow menší rozměry.
- **Vhodný počet pinů** – snadný návrh patice mikrokontroléru.
- **Nízká cena** – cena zařízení by neměla být příliš vysoká pro výrobu několik desítek kusů.

Při výběru byl brán ohled na předchozí požadavky. Nejvíce vyhovující byly mikrokontroléry firem Freescale [16] a ATMEL [9]. Osmibitové mikrokontroléry Freescale nenabízely vhodný poměr velikosti pouzdra a počtů pinů v závislosti na požadovaných perifériích. Oproti firmě ATMEL byly produkty firmy Freescale také výrazně dražší. Firma Freescale se může pyšnit svým vývojovým prostředím zvaným CodeWarrior, ATMEL ji však dohání se svým vývojovým prostředím AVR Studio.

Na základě těchto požadavků byl vybrán mikrokontrolér řady ATmega168 vyráběný firmou ATMEL, který požadované vlastnosti zcela splňuje. ATmega168 je mikrokontrolér, jehož instrukční soubor je typu RISC.

Mikrokontroléry AVR a ATmega168

Firma ATMEL už řadu let vyrábí 8-bitové mikrokontroléry řady ATmega168 s jádrem AVR a instrukčním souborem RISC. Modely se vyrábí ve čtyřech různých provedeních. A to v pouzdrě TQFP³ určeném pro SMT montáž s 32 vývody. S 32 vývody je dostupné pouzdro MLF a následně pak s 28 vývody jsou dostupná pouzdra DIP a také MLF.

K potřebám PDU využijeme mikrokontrolér řady ATmega168, který je dobře dostupný. Tento mikrokontrolér je vybaven 16 kB pamětí programu a poskytuje paměť dat o velikosti 1024 B. Tyto kapacity by měly být pro potřeby projektu dostačující.

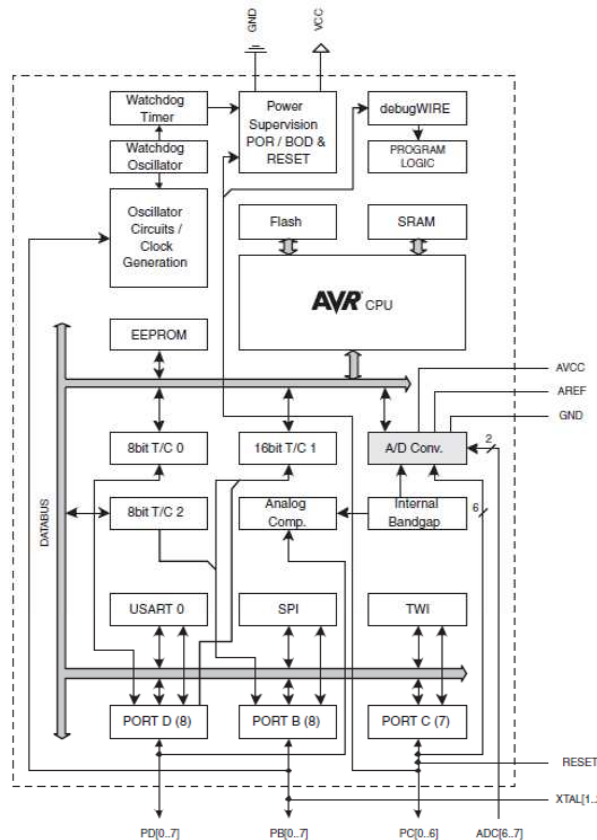
Základní parametry mikrokontroléru ATmega168:

- 8-bitový mikrokontrolér s jádrem AVR a instrukční sadou RISC.
- 4 typy pouzder, TQFP a MLF 32 pinů, TQFP a DIP 28 pinů.
- Provozní kmitočet 0 - 20 MHz při napájení 4,5 - 5 V.
- Napájecí napětí 2,7 - 5,5 V DC.
- Provozní teplota od -40°C do 85°C .
- 16 kB paměti programu, 1024 B interní SRAM paměti dat, 256 B EEPROM.
- 23 V/V programovatelných pinů, 2x sériové rozhraní UART, 1x SPI, 2x A/D převodník.
- Programovatelný pomocí JTAG.

Podrobnější specifikace mikrokontroléru ATmega168 v datasheetu výrobce [8].

Na obrázku 4.2 je možné vidět blokové schéma mikrokontroléru ATmega168.

³TQFP – Thin quad flat pack



Obrázek 4.2: Blokové schéma mikrokontroléru ATmega168 [8].

4.2.2 Volba ethernetového modulu

Ethernetový modul by měl poskytovat dobré komunikační rozhraní a jednoduché dorozumívací prostředky.

Mezi jeho vlastnosti by mělo patřit:

- SPI rozhraní – zapojení s mikrokontrolérem.
- UDP/TCP stack – pro komunikaci se vzdáleným řídicím systémem.

Nabízených produktů je mnoho, například čip ENC28J60 [15] od firmy Microchip [18], který splňuje výše uvedené požadavky. Tyto vlastnosti má také modul Nano SocketLAN [19] od firmy ConnectOne [13], který je navíc kompletně hardwarově realizován a postačí pouze připojení komunikačního interface a napájení. Mezi další kladné vlastnosti modulu patří řízení modulu AT příkazy [7].

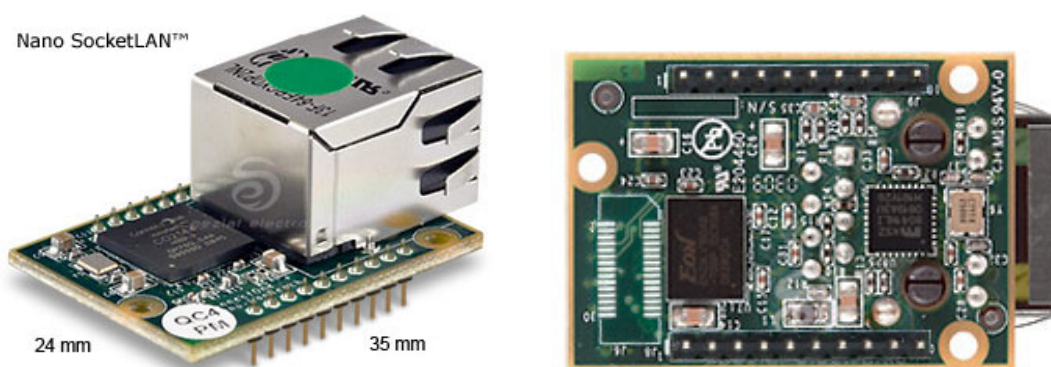
Nano SocketLAN

Nano SocketLAN je hardwarové zařízení, poskytující takzvaný most mezi sériovým rozhraním a rozhraním 10/100Mb Ethernet LAN. Jádrem modulu je mikroprocesor iChip CO2144, který podporuje TCP Stack. Nano SocketLAN je kompletně navržený modul včetně DPS.

Má kompaktní rozměry a používá jednoduchý konektor pro zasunutí do DPS se spojovaným mikrokontrolérem. Uspořádání vývodů je stejné jako u produktu iWiFi. Toho lze využít v případě použití rozhraní WiFi.

Nano SocketLAN nabízí bezpečnostní protokoly mezi aplikací a sítí. Podporuje SSL3/TLS1 protokoly pro HTTPS a FTPS. Poskytuje až deset souběžně otevřených TCP/UDP socketů. Obsahuje dvě naslouchací schránky, dva webservery, SMTP a POP3 klienty, MIME přílohy, FTP a TELNET klienty.

DPS Nano SocketLAN zahrnuje plně osazený RJ45 konektor, jak je vidět z obrázku 4.3. Firmware modulu obsluhuje TCP/IP stack a parametry síťové konfigurace jsou uloženy ve flash paměti. Jádru modulu pracuje na 1,2 V, zatímco V/V logika a napájení je koncipována na 3,3 V [19].



Obrázek 4.3: Pohled na Nano SocketLAN od firmy ConnectOne [13].

Hlavní rysy Nano SocketLAN [19]:

- Velmi malé rozměry: 38,9 x 24,88 x 20,9 mm.
- Jádru CPU ConnectOne CO2144, výrobní technologie 0,13 mikronů.
- Provozní frekvence 48 MHz.
- Napájení modulu 3,3 V DC, proudový odběr 130 mA, při stand-by režimu 30 mA.
- Provozní teplota od -40°C do 85°C .
- 2x 10 pin-header 2 mm.
- Sériová rozhraní: SPI, UART, USB.
- Řízení a komunikace mikrokontroléru s Nano SocketLAN pomocí AT příkazů.

Podporované internetové protokoly: ARP, ICMP, IP, UDP, TCP, DHCP, DNS, NTP, SMTP, POP3, MIME, HTTP, FTP a TELNET.

Podporované bezpečnostní protokoly: SSL3/TLS1, HTTPS, FTPS, RSA, AES-128/256, 3DES, RC-4, SHA-1, MD-5, MD-2.

Protokoly akcelerované hardwarem: AES, 3DES a SHA.

4.2.3 Volba měřícího modulu

Měření spotřeby elektrických spotřebičů připojených do zásuvky je v této práci pouze přípravou pro pozdější realizaci. Smyslem měření spotřeby je dát uživateli informaci o aktuální spotřebě elektrické energie. Takové měření se upotřebí například v případě, kdy uživatel zapojí spotřebiče, které budou odebírat větší proud, než na který jsou zásuvky dimenzované. Uživatel, který provádí obsluhu okamžitě zná odebíraný proud. Z těchto a více důvodů se PDU navrhuje s možností měření.

Měření se provádí pomocí speciálních integrovaných obvodů. Na tyto integrované obvody se připojí měřené napětí a měřený proud transformovaný na rozsah měřený integrovaným obvodem. Integrovaný obvod vlastní sériové rozhraní, kterým komunikuje s mikrokontrolérem.

Měřících modulů je na trhu velké množství. Nejrozšířenější jsou moduly od firem Analog Devices [5] a Cirrus Logic [12].

Výběr modulu tedy záležel na požadavcích:

- **SPI/I2C rozhraní** – bude nutné pro spojení s mikrokontrolérem.
- **Vhodné pouzdro** – zamezit komplikacím při návrhu desky plošných spojů a osazování.
- **Měření výkonu/příkonu** – některé integrované obvody měří pouze jednu veličinu.
- **Postačí měření pouze jedné fáze** – zásuvky budou pouze jednofázové.
- **Nízká cena** – protože v PDU bude umístěno 8 ks, nízká cena je velmi důležitá pro zachování nízké ceny celého PDU.

Kandidáty, kteří by splňovali výše uvedené požadavky jsou dva výrobky. ADE7763 [4] od firmy Analog Devices a CS5463 [14] od firmy Cirrus Logic. Oba mají pouzdro typu SSOP⁴. Výrobky mají téměř stejné vlastnosti. Vzhledem k ceně a lepší dostupnosti na českém trhu byl vybrán modul CS5463 od firmy Cirrus Logic.

Měřící modul CS5463 od Cirrus Logic

Tento modul je realizován v pouzdře SSOP 24 pinů. Modul je schopný měřit napětí, proud, příkon. Obsahuje jedno sériové rozhraní SPI a třívodičové pulsní rozhraní.

Základní vlastnosti modulu [14]:

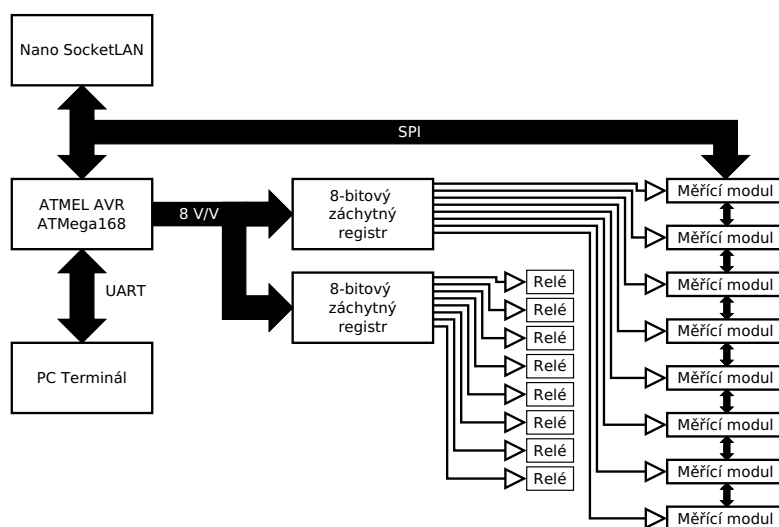
- Napájení 3-5 V DC.
- Měření jedné fáze.
- Komunikační sériové rozhraní SPI.
- Provozní teplota od -40°C do 85°C .

⁴SSOP – Shrink Small-Outline Package

4.2.4 Konceptuální návrh základní jednotky

Základní jednotka PDU bude obsahovat již zmiňované moduly. V základní jednotce přibude napájecí část, programovací rozhraní a samotný mikrokontrolér. Vstupy/výstupy mikrokontroléru budou vyvedeny na pin-headry k připojení dalšího hardwaru, například část pro spínání střídavého napětí, či část měření spotřeby elektrické energie. Takto navržená základní deska bude velmi modulární a připojené periferie bude možné libovolně připojovat a odpojovat, nebo dokonce zaměňovat. Řešení modularity se však příliš nepromítne do pořizovací ceny.

Po vybrání komponent je nutné navrhnout základní jednotku a způsob jak budou periferie propojeny a jakým způsobem spolu budou komunikovat. Na základě tohoto konceptuálního návrhu zapojení bude snažší reálný návrh zapojení a schématu.



Obrázek 4.4: Blokové schéma hardwaru PDU.

Na obrázku 4.4 je možné vidět výsledné blokové schéma navrženého PDU. Jádrem je již diskutovaný mikrokontrolér ATMEGA AVR ATmega168 v kapitole 4.2.1, v pouzdře TQFP s 32 vývody. Základní deska bude napájena 12 V stejnosměrně. Záchytné registry a měřicí modul *CS5463* vyžadují napájení minimálně 3 V, musíme proto mikrokontrolér napájet taktéž 3 V, abychom dosáhli stejných napěťových úrovní. Na základní jednotce bude tedy nutné navrhnout úpravu napájení z 12 V DC \Rightarrow 3,3 V DC.

Zapojení mikrokontroléru a ethernetového modulu:

Mikrokontrolér i ethernetový modul nabízejí více možností, jak spolu komunikovat. Je tedy na nás, jaké rozhraní zvolíme. Protože na straně mikrokontroléru budeme využívat rozhraní USART pro výpis ladících informací na terminál, pro komunikaci ethernetového modulu s mikrokontrolérem použijeme rozhraní SPI.

Zapojení mikrokontroléru a měřících modulů:

Měřicí modul nabízí pouze pro komunikaci rozhraní SPI. Zapojení bude realizováno pomocí

SPI rozhraní. Měřicí moduly jsou napájeny 3-5 V stejnosměrně. Napájet je budeme ze základní desky 3,3 V. Měřicí moduly budou umístěny na samostatné desce plošných spojů. Fyzické spojení bude pomocí kabelu.

Zapojení mikrokontroléru a ladícího terminálu:

Pro odesílání ladících informací a ladění firmwaru aplikace využijeme periférii USART na straně mikrokontroléru.

Zapojení mikrokontroléru a relé výstupů:

Zařízení bude obsahovat 8 zásuvek. Jejich stav bude závislý na 8-bitovém výstupu mikrokontroléru. Výstupy z mikrokontroléru budou zapojeny do registru a následně pak do tranzistorového pole. Tranzistorové pole bude zapojeno jako otevřený kolektor. Pro spínání zásuvek použijeme relé, jehož vstup cívky bude připojen na 12 V a výstup na kolektory tranzistorového pole. Po sepnutí tranzistoru dojde k uzemnění a uzavření okruhu cívky relé.

Zapojení mikrokontroléru a programátoru:

Mikrokontrolér bude programován pomocí JTAG využívaje rozhraní SPI.

4.3 Konceptuální návrh FW

Návrh firmware mikrokontroléru PDU závisí na zvolené platformě. V tomto případě budeme navrhovat firmware pro mikrokontroléry AVR (viz. 4.2.1). Při návrhu je nutné zohlednit principy a periférie zvoleného typu, tedy ATmega168. Firma ATMEL [9] nabízí k vývoji mikrokontroléru vývojové prostředí AVR Studio díky kterému se práce při vývoji zefektivní.

Návrh FW můžeme rozdělit na dvě základní části:

1. Firmware mikrokontroléru řídící vstupy/výstupy, jeho periférie (SPI, UART, ...)
2. TCP/IP komunikace s COS⁵ (návrh TCP protokolu)

4.3.1 Konceptuální návrh FW pro obsluhu periférií

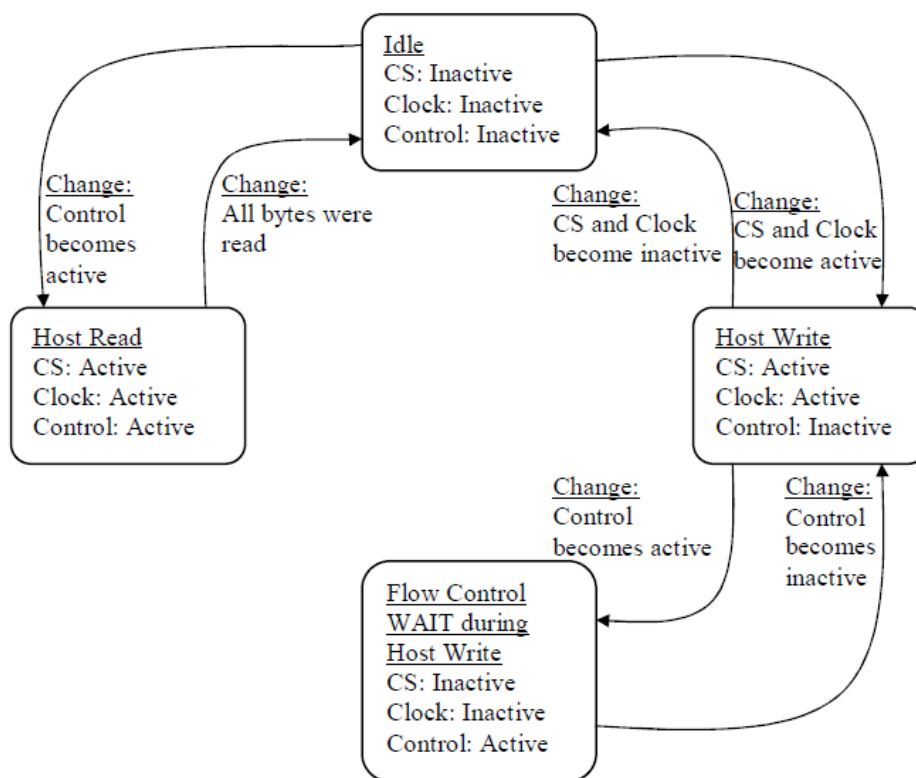
Úkolem firmware mikroprocesoru bude řízení jeho periférií a zajistit komunikaci s připojenými moduly. Nejdůležitějším úkolem firmwaru je komunikace s modulem Nano SocketLAN, který je spojen s mikrokontrolérem pomocí sériového rozhraní SPI. Dalším úkolem firmwaru bude schopnost komunikovat pomocí UART, pomocí kterého bude možnost připojit PDU k PC terminálu a sledovat jeho chování. Zásadní funkcí, která je pro smysl PDU nezbytná, je ovládání výstupního 8-bitového portu. Protože bude nutné pro zvolení měřicího modulu využívat signál ChipSelect, bude tento výstup společný s výstupem relé. Tento problém bude ošetřen dvěma zachytnými 8-bitovými registry a signálem OutputEnable.

SPI komunikace s Nano SocketLAN

Nano SocketLAN má SPI protokol na principu Příkaz - Odpověď. Mikrokontrolér bude v režimu *master* a Nano SocketLAN v režimu *slave*. Princip SPI komunikace je postaven primárně na *masterovi* a krátkých 8-bitových přenosech. *Master* vyšle znak a *slave* okamžitě odpoví taktéž znakem. Nano SocketLAN má v dokumentaci specifikovaný konečný automat [24], dle kterého bude implementace komunikace postupovat.

Na obrázku 4.5 je možné vidět konečný automat převzatý z [24]. Na obrázku je možné spatřit signál *ChipSelect (CS)*, který nastavuje *master*. Dává tak modulu informaci o tom,

⁵COS – Centrální Ovládací Systém



Obrázek 4.5: Konečný automat znázorňující komunikaci mikrokontroléru(host) a Nano SocketLAN [24].

že právě s tímto modulem bude komunikovat. Dalším signálem je *Clock*. To jsou hodiny, kterými *master* určuje frekvenci komunikace a synchronizaci mezi dvěma body. Posledním signálem na obrázku je signál *Control*, což je signál směřovaný z Nano SocketLAN do mikrokontroléru. Tímto signálem dává Nano SocketLAN informaci o tom, že je modul schopný komunikace. Označením *Host* se myslí mikrokontrolér v roli *mastera*.

Pokud čtenář bude detailněji zkoumat konečný automat 4.5, zjistí princip komunikace. *Master* sleduje signál *Control*, jakmile se nastaví do log.1, je to znamení pro *mastera*, že modul chce komunikovat. Pokud *Master* potřebuje navázat komunikaci, nastaví CS do log.0 a spustí „hodiny“ k synchronizaci. Následně začne posílat sadu znaků, které jako řetězec tvoří zprávu pro Nano SocketLAN zakončenou znakem *CR*. Na každý přijatý znak odpovídá Nano SocketLAN nesmyslným znakem. Nesmyslným znakem informuje modul *mastera*, že modul úspěšně přijal znak od *mastera*. Jakmile Nano SocketLAN přijme zprávu, které rozumí, nachystá odpověď. Příprava odpovědi trvá nějaký čas. V případě, že je odpověď připravena, Nano SocketLAN informuje *mastera* signálem *control*. V tomto případě začne *master* vysílat znaky nenesoucí pro Nano SocketLAN žádnou informaci a přijímat znaky mající nějaký význam od Nano SocketLAN.

Nano SocketLAN je řízen AT příkazy [7]. To znamená, že mikrokontrolér pošle příkaz v daném formátu, Nano SocketLAN jej provede a příslušně odpoví.

Ukázkový příklad AT příkazu vypadá následovně:

Příkaz: AT+iIPA?, odpověď bude IP adresa modulu: Odpověď: I/192.168.1.2

Takovými příkazy nakonfigurujeme modul na základě obsahu následující podkapitoly 4.3.2.

4.3.2 Konceptuální návrh síťové komunikace

Aby bylo možné zásuvkovou lištu ovládat vzdáleně pomocí sítě ethernet, musíme diskutovat problém přenosu informací. Pro přenos dat máme na výběr mezi dvěma transportními protokoly, a to TCP⁶, nebo UDP⁷. Dále se také musíme zaměřit na problematiku server/klient.

Volba TCP / UDP

Na základě znalostí obou transportních protokolů zvolíme ten nejvhodnější pro tuto aplikaci. Položíme si několik základních otázek, které budou specifické pro návrh.

- **Je důležité garantovat doručení zprávy?:**

Protože se jedná o zařízení, které musí pracovat spolehlivě a rychle, je nutné eliminovat možné problémy při přenosu. Je předpokladem, že navržený aplikační protokol bude komunikovat na principu dotaz - odpověď. V případě, že jeden z aktérů pošle požadavek, bude čekat, dokud na něj druhá strana neodpoví. Za základě znalostí UDP protokolu, víme, že nemáme garantován spolehlivý přenos a je nutné implementovat speciální opatření - ztratila se zpráva na síti, či někde jinde? TCP protokol má sice vyšší režii přenosu, ale garantuje doručení zprávy.

- **Je kritické pro síť využití protokolu s vyšší režii?**

Předpokladem je, že řídicí zprávy pro nastavení stavu zásuvek se budou zasílat pouze v případě zájmu uživatele. Vzhledem k tomu, že uživatel zadá příkaz jednou za dlouhý časový interval, vyšší režie spojení nevádí.

Závěrem této podkapitoly je volba protokolu TCP.

PDU jako klient, nebo server

Většina síťových aplikací pracuje na principu klient – server. Otázkou ale je, jaké zařízení bude server nebo klient.

Pro lepší pochopení problematiky je uveden základní princip „klient – server“: Server je stále čekající na příchozí spojení, jakmile klient požádá o spojení, server jej akceptuje a může začít komunikace. Klient je zařízení, které se většinou připojuje k serveru, v případě, že potřebuje obsloužit. Klient pošle požadavek, sever jej zpracuje a odpoví klientovi.

- **PDU klient, COS server?**

V tomto případě vzniká několik problémů - jak donutit zásuvky, aby se připojily ke správnému serveru? Co když se spojení nechtěně rozpojí? Jak by vypadala komunikace aplikačního protokolu?

- **PDU server, COS klient?**

Toto řešení se zdá být daleko vhodnější. Zásuvky budou server, čekající na spojení od klienta (COS). Klient se připojí na daný server (PDU), který bude mít vlastní ID (IP adresa). Klient pošle zásuvce požadavek na změnu stavu. Server provede obsluhu (nastaví stav zásuvek) a odpoví klientovi o provedení požadavku.

Na základě předchozí úvahy přiřazujeme PDU roli serveru a COS roli klienta.

⁶TCP – Transmission Control Protocol

⁷UDP – User Datagram Protocol

TCP protokol

Jak bylo zmíněno v závěru předchozí kapitoly, pro komunikaci PDU s COS pomocí sítě ethernet bude využito TCP transportního protokolu. TCP je spojově orientovaný protokol pro přenos dat na transportní vrstvě s doručováním se zárukou. V současnosti je zdokumentován v IETF RFC 793 [32]. TCP používá služby IP protokolu opakovaným odesláním paketů a při ztrátě paketu zajišťuje znovu doručení a přeuspořádáním přijatých paketů zajišťuje správné pořadí doručení. Tím TCP plní úlohu transportní vrstvy ve zjednodušeném modelu ISO/OSI počítačové sítě, který byl zmíněn v kapitole 3.1.3.

Aplikace posílá proud (stream) bajtů TCP protokolu k doručení po síti, TCP rozděljuje proud bajtů do přiměřeně velkých segmentů. Velikost segmentů je určena parametrem MTU⁸. TCP pak předá takto vzniklé pakety IP protokolu (3.vrstvě ISO/OSI Modelu) k přenosu ethernetem do TCP stacku na druhé straně TCP spojení. TCP služba zjistí, že nedošlo ke ztrátě dat tak, že každému paketu přidělil pořadové číslo, které se využije ke zjištění, zda data byla přijata ve správném pořadí [36].

TCP modul na straně příjemce posílá zpět potvrzovací pakety pro data, které byly úspěšně přijaty. V případě, že by se potvrzovací pakety nevrátily do vhodného časového intervalu (RTT, round-trip time), vypršel by limit odesílatele a ztracené pakety by vyslal znovu [36].

Bitů	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	zdrojový port																cílový port															
32	číslo sekvence																															
64	potvrzený bajt																															
96	offset dat				rezervováno				příznaky				okénko																			
128	kontrolní součet																Urgent Pointer															
160	volby (volitelné)																															
192	volby (pokračování)																								výplň (do 32)							
224	data																															

Obrázek 4.6: Struktura TCP paketu [36]

TCP protokol také zjišťuje, jestli přenášená data nebyla ovlivněna okolním rušením tak, že spočítá kontrolní součet odesílaného paketu. Následně je kontrolní součet vložen do hlavičky odesílaného paketu a příjemce je na základě tohoto součtu ověří, zda paket nebyl poškozen. Na obrázku 4.6 je možné vidět strukturu TCP paketu.

Návrh aplikačního protokolu PDU

Aby mohl COS komunikovat s PDU, je nutné, aby se oba dorozumívali stejným jazykem, takzvaným aplikačním (komunikačním) protokolem. V kapitole 4.3.2 bylo rozhodnuto, že PDU budou v roli serveru. To znamená, že jednotlivé PDU budou čekat na výzvu od klienta (COS). PDU nikdy sám od sebe nebude nikomu vnucovat komunikaci. Komunikaci vždy započne COS a PDU pouze bude konat příkazy a podávat zprávu o jejich provedení. Prvotní myšlenka byla navrhnout protokol binární. Nakonec aplikační protokol bude navržen jako

⁸MTU – Maximum Transmission Unit

znakový, jelikož je vhodné přenášet data (příkazy a odpovědi), kterým by mohl rozumět běžný člověk.

Komunikační protokol je znázorněn v tabulce 4.1:

Příkaz: COS→PDU	Odpověď: PDU→COS
TURNON XXXXXXXX;	ACK YYYYYYYY;
TURNOFF XXXXXXXX;	ACK YYYYYYYY;
SET YYYYYYYY;	ACK YYYYYYYY;
STATUS;	STATUS YYYYYYYY;
DELAY ZZZZ;	ACK_D ZZZZ;
STATUS_D;	STATUS_D ZZZZ;
Neznámý příkaz	UNKNOWN_COMMAND;

Tabulka 4.1: Aplikační protokol PDU.

V tabulce 4.2 jsou detailně popsány příkazy, které PDU přijímá.

COS→PDU	Popis
TURNON XXXXXXXX;	Příkazem je možné zapnout zásuvky 0-7. X (nejlevější X odpovídá zásuvce č.0) je nahrazeno čísly [0-1], kde 0 značí ignoraci a 1 značí převést zásuvky do stavu zapnuto.
TURNOFF XXXXXXXX;	Příkazem je možné vypnout zásuvky 0-7. X (nejlevější X odpovídá zásuvce č.0) je nahrazeno čísly [0-1], kde 0 značí ignoraci a 1 značí převést zásuvky do stavu vypnuto.
SET YYYYYYYY;	Příkazem je možné nastavit stav zásuvek 0-7. Y (nejlevější Y odpovídá zásuvce č.0) je nahrazeno čísly [0-1], kde 0 značí vypnout a 1 značí zapnout zásuvku.
STATUS;	Příkazem je možné zjistit stav zásuvek 0-7.
DELAY ZZZZ;	Příkazem je možné nastavit interval mezi zapnutím jednotlivých zásuvek, kde Z je číslo [0-9] a určuje interval v milisekundách.
STATUS_D;	Příkazem je možné zjistit aktuálně nastavený interval mezi zapnutím jednotlivých zásuvek.

Tabulka 4.2: Popis příkazů aplikačního protokol PDU.

Tabulka 4.3 vysvětluje význam odpovědí PDU na přijaté příkazy.

PDU→COS	Popis
ACK YYYYYYYY;	Odpověď od PDU, který přijme příkaz mění jeho stav zásuvek. Potvrzuje přijetí příkazu a stav zásuvek 0-7 po provedení příkazu. Y (nejlevější Y odpovídá zásuvce č.0) je nahrazeno čísly [0-1], kde 0 značí vypnuto a 1 značí zapnuto.
STATUS YYYYYYYY;	Odpověď od PDU na stav zásuvek. Y (nejlevější Y odpovídá zásuvce č.0) je nahrazeno čísly [0-9], kde 0 značí vypnuto a 1 značí zapnuto.
ACK_D ZZZZ;	PDU odpoví potvrzením, kde Z je číslo [0-9] a informuje o nastaveném intervalu po přijetí příkazu. Zpoždění je v milisekundách.
STATUS_D ZZZZ;	kde Z je číslo [0-9] a informuje o aktuálně nastaveném intervalu zpoždění v milisekundách.
UNKNOWN_COMMAND;	Odpověď od PDU v případě, že přijme neznámý příkaz, nebo dojde k poškození správného příkazu.

Tabulka 4.3: Popis odpovědí aplikačního protokol PDU.

Kapitola 5

Realizace HW

Tato kapitola popisuje realizaci konceptuálního návrhu hardwaru. Obsahem je tedy návrh schémat zapojení základní jednotky a jednotky pro měření a spínání zásuvek. Dále návrh desky plošných spojů (dále DPS) a seznam potřebných součástek. Navržené schémata, DPS a seznam součástek jsou obsažena v příloze práce a jsou ve formátu návrhového systému DPS EAGLE, viz. kapitola 5.1.1.

5.1 Prostředky použité při realizaci

5.1.1 Návrhový systém DPS EAGLE

Návrhu desek plošných spojů (DPS, angl.PCB) byl proveden v návrhovém systému EAGLE¹ od firmy CadSoft [11] pocházející z Německa. Návrhový systém EAGLE je velmi schopný a výkonný editor DPS, použitelný i na profesionální návrhy. Je dostupný ve free verzi s omezením na velikost DPS (100x80mm). Návrhový systém je dostupný jak pro operační systém Windows, tak i Linux.

Aplikace má v sobě několik podaplikací, které jsou schopné návrhu jednotlivých a individuálních součástek, návrhu elektrických schémat a návrhu desek plošných spojů. Podaplikace v sobě obsahují usnadňující funkce. Například kontrola správně spojených vodičů v schématu. Mezi další významné funkce patří Autorouter. Tato funkce umožňuje automaticky propojit signály na DPS. Strategie propojování vodičů je určena uživatelskými parametry (šířka spoje, izolace, apod). Navržené schémata i DPS je možné exportovat pomocí CAM procesoru ve formátech PNG, PostScript a jiné. Pro návrh byl použit dvouvrstvý návrh DPS.

5.1.2 Prostředky pro osazení DPS

Na osazení DPS byla použita technologie SMT². SMT je technologický postup, kdy se součástky pájejí přímo na povrch plošného spoje. Tyto součástky jsou nazývány SMD³. Na rozdíl od původní technologie, kdy se musely vývody součástek prostrčit skrz DPS a zapájet na druhé straně, jsou součástky SMD miniaturní. Vzniká tak úspora místem. Dalším kladem těchto součástek je možnost osazení z obou stran DPS. DPS tak může být daleko menší v porovnání DPS řešenou klasickou technologií. Jediná nevýhoda spočívá v tom, že

¹EAGLE – Easily Applicable Graphical Layout Editor

²SMT – Surface Mount Technology

³SMD – Surface Mount Device

součástky jsou miniaturní a je obtížné, prakticky až nemožné, využívat technologii SMT pro amatérské návrhy a je třeba speciálního osazovacího zařízení.

V této práci jsou SMT technologií vyrobeny a osazeny DPS základní jednotky a 8 kusů DPS měřicího a spínacího modulu. Použití této technologie vedlo k daleko menší velikosti desek obou modulů. Základní jednotka tak nabývá rozměrů 67x81mm, a modul měření a spínání nabývá 83x49mm i přes masivní součástky jako relé nebo transformátor napětí a transformátor proudu. Na výsledné DPS je i zelená nepájivá maska, která vede ke snadnějšímu osazování a chrání před nechtěným zkratem kovovým předmětem. Pocínování pájecích plošek taktéž usnadňuje práci při osazování. Této pomocné procedury je v práci taktéž využito.

Klasickou technologií je navržena spínací a měřicí část DPS, jelikož součástky, které nabývají větších rozměrů (transformátory, relé) nešlo minimalizovat. Signálová část DPS pro spínání a měření je osazena technologií SMT.

5.2 Základní jednotka

Základní jednotka je realizována na základě konceptuálního návrhu popsaném v kapitole 4.

5.2.1 Popis schématu zapojení základní jednotky

Na obrázku B.1 v příloze B je možné vidět schéma zapojení základní jednotky. Popis použitých modulů je popsán v kapitole 4.2.4. Hlavní částí této základní jednotky je mikrokontrolér ATmega168. Další důležitou jednotkou je Nano SocketLAN, jehož patice je taktéž umístěna na DPS. Ostatní součástky jsou nezbytné obvody pro správný běh vestavěného systému.

Napájení

Napájecí napětí celého zařízení je 12 V DC. Napájecí napětí mikrokontroléru ATmega168, Nano SocketLAN a záchytných registrů je 3,3 V DC. Nejprve je upraveno napájecí napětí 12 V DC na 5 V DC konstantním stabilizátorem napětí s označením 7805 [3] z důvodů vývoje a laborování s prototypem. Upravené napájecí napětí 5 V DC je opět stabilizováno na 3,3 V DC pomocí dalšího konstantního stabilizátoru s označením LF33 [17]. Kondenzátory C1-C6 v napájecí části slouží vyhlazení a filtraci výstupního napětí jednotlivých úrovní. Všechny integrované obvody obsahují blízko svých napájecích vývodů tzv. blokovací kondenzátory C10-C12, jejichž úkolem je zásobovat integrovaný obvod elektrickým proudem při rychlých změnách jeho odběru proudu. LED1 je informační LED dioda, která signalizuje připojení napájecího napětí.

ATmega168

ATmega je zapojená dle doporučení výrobce [8]. Jako hodiny taktující jádro mikrokontroléru je umístěn krystal o frekvenci 20 MHz, lze však použít interní krystal obvodu. K programování ATmega168 je vyveden konektor 2x3pinů JP1 a využívá standardu JTAG [34]. 8-bitový port D je vyveden do záchytného registru 74AC573 [1] (relé výstup) a dalšího záchytného registru 74HCT563 [2], který slouží jako ChipSelect (výběr slave zařízení pro komunikaci po SPI). Další vyvedenou periferií je konektor SV1, na který je vyvedeno rozhraní SPI pro připojení měřicích modulů. Na konektor je navíc vyveden signál *reset*. Na portu C jsou vyvedeny potřebné řídicí signály k řízení zápisu do záchytných registrů, I2C

signály pro připojení externí paměti E2PROM a signál přerušení od měřících modulů. Poslední vyvedená periférie mikrokontroléru je vývod 1x3 pinů k připojení sériového rozhraní UART.

Nano SocketLAN a komunikace s ATMega168

Nano SocketLAN, jak již bylo zmíněno v kapitole 4.2.2, je napájen 3,3 V. S mikrokontrolérem komunikuje přes sériové rozhraní SPI. Ke komunikaci využívá další řídicí signály, kterými dává mikrokontroléru znamení, že potřebuje obsloužit. Těmito signály jsou SPI_INT a DATA_RDY.

8-Bitový port, záchytné registry a tranzistorové pole

Jak již bylo zmíněno v předchozích kapitolách, ke spínání relé zásuvek využijeme 8-bitový port mikrokontroléru. Protože nebude tento výstupní port využit pouze ke spínání relé, ale bude mít i funkci ChipSelectu (CS), je nutné použít dva záchytné registry. Tyto registry jsou vůči jejich vstupu imunní, dokud není aktivovaný signál REG_1CK nebo REG_2CK. Poté nabudou jejich výstupy aktuálních vstupů. Takto se pak mohou signály na sběrnici měnit, aniž by ovlivnily výstup relé, nebo ChipSelect.

Protože část relé je výkonný elektromagnet, spotřebuje mnoho proudu. Mikrokontrolér dokáže manipulovat pouze s logickými hodnotami, tedy pouze s napěťovou úrovní a minimem proudu v řádech mikroampér. Napájet elektromagnet mikrokontrolérem je z těchto důvodů nemyslitelné. Vhodným řešením je využití tranzistoru v zapojení zvaném „otevřený kolektor“. Pro úsporu místa je místo 8 tranzistorů použito tranzistorové pole *ULN2803A* [20]. Je předpokládáno, že čtenář zná princip tranzistoru. Na bázi tranzistoru je připojen signál ze záchytného registru (úroveň 3,3 V), na kolektor je přivedeno +12 V (výstup z cívky) a emitor tranzistoru je spojen s zemí. Jakmile se na bázi tranzistoru objeví logický signál 3,3 V, tranzistor se otevře, a cívku relé uzemní. Tím se uzavře elektrický okruh a relé je sepnuto. Takových tranzistorů se v ULN nachází 8.

Záchytný registr ChipSelectu je stejný jako záchytný registr pro relé, ovšem rozdíl je v tom, že registr neguje svoji vstupní hodnotu. Je to z toho důvodu, že ChipSelect vybírá zařízení pomocí nízké logické úrovně.

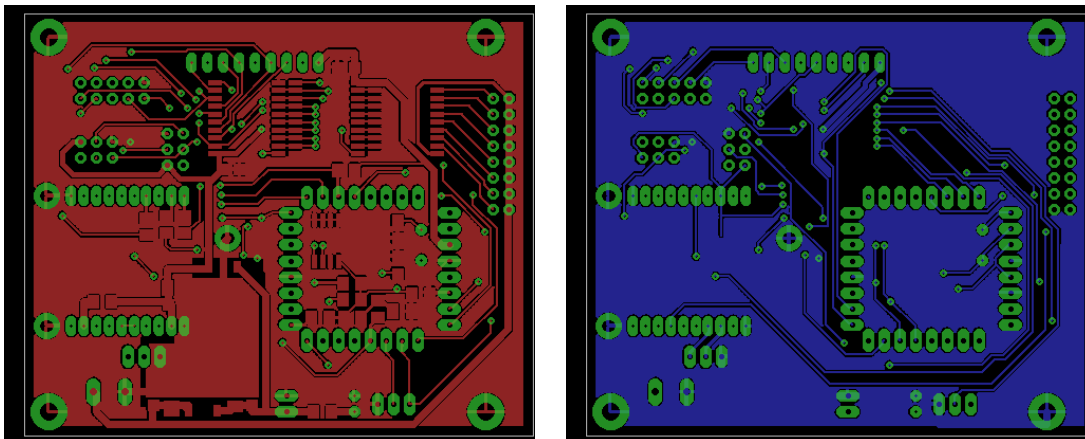
5.2.2 DPS základní jednotky

Na obrázku 5.1 je zobrazena deska plošných spojů základní jednotky. DPS je navržena jako dvouvrstvá, proto vlevo může čtenář vidět horní stranu spojů a vpravo spodní stranu spojů. Součástky jsou osazeny pouze z jedné strany. Součástky jsou tak dobře dostupné. Na straně spodní jsou pouze tištěné spoje. Napájecí konektor je tvořen svorkou *ARK500-2*, která umožňuje připojení dvou napájecích vodičů. Deska je napájena pouze 12 V stejnosměrně.

Na desce plošných spojů je „rozlitá měď“. To znamená, že zemnicí svorka má signál téměř po celé ploše desky. Rozlitá měď se tak používá jako stínění. Velikost desky odpovídá 67x81mm. Návrh desky plošných spojů byl metodicky veden na základě literatury [37]. Na obrázku C.1 v příloze C je zobrazeno rozmístění součástek.

5.3 Modul spínání a měření

Modul spínání a měření nebyl v konceptuálním návrhu příliš zmíněn, jelikož byl považován za méně důležitou součást PDU. Pouze v hardwarovém návrhu bude mít každá zásuvka



Obrázek 5.1: Vlevo horní vrstva DPS, vpravo spodní vrstva základní jednotky

vlastní desku plošných spojů, kde bude umístěno relé, transformátory proudu a napětí, a také integrovaný obvod měřící elektrickou energii.

Cílem této práce není implementace měření, ale pouze příprava pro případnou realizaci. Schéma a DPS je tedy navržena včetně měření. Osazeny však budou součástky pouze pro spínání.

5.3.1 Popis schématu zapojení spínacího a měřícího modulu

Na obrázku B.2 v příloze B je zobrazeno schéma modulu pro spínání zásuvek a měření spotřebované energie. Na modulu jsou tři galvanicky oddělené okruhy.

První okruh slouží ke spínání relé, jeho elektrický potenciál je tedy pouze 12 V DC. Tento okruh je napájen ze základní jednotky, která okruh uzavírá/spíná (výstup na konkrétní relé).

Dalším okruhem je spínaný okruh, kterým je výstup na zásuvky. Jeho napěťový potenciál je střídavý, 230 V. Toto napětí je životu nebezpečné. Je předpokládáno, že čtenář zná problematiku zásuvkových rozvodů, kde se rozlišují vodiče (u jednofázové zásuvky) na *fáze*, *nulák*, a *zem*. Modrým nulovacím vodičem teče v případě odběru (ze zásuvky) stejný proud jako vodičem fázovým (fázový má napětí proti zemi 230 V, nulový vodič 0 V). Fázový vodič je tak životu nebezpečný, proto bude rozepínán relém.

Posledním okruhem je okruh měření. Tento okruh je galvanicky oddělený od okruhu s napětím 230 V pomocí transformátoru proudu a transformátoru napětí. Měřící modul je zapojen dle doporučení výrobcem [14]. Modul má vyvedeno na konektor *JP1* 2x5 pinů sériové rozhraní pro komunikaci se základní jednotkou. Tímto konektorem je také přivedeno napájení integrovaného obvodu 3,3 V a signál interrupt. Dalším vývodem *JP2* je pulsní rozhraní pro připojení například čítače. Pro naše účely je toto rozhraní nevyužito.

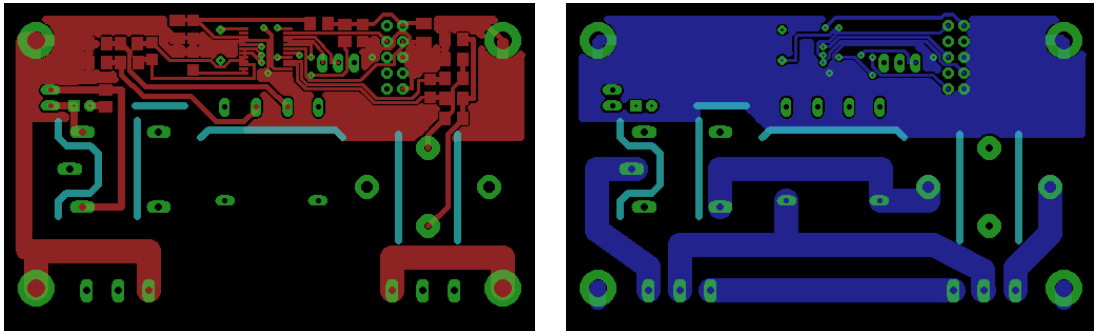
5.3.2 DPS spínacího a měřícího modulu

Na obrázku 5.2 je zobrazena deska plošných spojů spínacího a měřícího modulu. DPS je navržena jako dvouvrstvá, proto vlevo může čtenář vidět horní stranu spojů a vpravo spodní stranu spojů. Součástky jsou osazeny pouze z jedné strany. Součástky jsou tak dobře dostupné. Na spodní straně jsou pouze tištěné spoje. Pro napájení relé a současně

tak spínacího okruhu je použit konektor *PSH-2*, který umožňuje připojení dvou napájecích vodičů s napětím 12V.

Dalším přívodem napájení je konektor *ML-10*, který zajišťuje napájení měřícího modulu 3,3 V DC a poskytuje připojení sériového rozhraní SPI. Jeho vývody také obsahují signál přerušeni.

Okruh, který je spínáný, tedy střídavý s napětím 230 V je dimenzován do 10 A. Jeho spoje jsou 4mm široké a při návrhu DPS byla snaha o co největší izolační mezery mezi tištěnými vodiči. Z tohoto důvodu není v této části spínaného okruhu rozlitá měď. Rozlévání mědi tu není nutné jelikož při přívodu napájení spotřebičů není třeba eliminovat okolní rušení.



Obrázek 5.2: Vlevo horní vrstva DPS, vpravo spodní vrstva spínacího a měřícího modulu

Na desce plošných spojů je takzvaná „rozlitá měď“ pouze v nízkonapěťové části. Rozlitá měď je použita jako stínění logických signálů. V elikost desky odpovídá rozměrům 48x84mm. Návrh desky plošných spojů byl taktéž metodicky veden na základě literatury [37]. Na obrázku C.2 v příloze v příloze C je zobrazeno rozmístění součástek. V této práci není však osazena měřící část, návrh desky je pouze přípravou pro další vývoj, včetně měření.

Kapitola 6

Realizace FW

V této kapitole je popsána implementace firmwaru mikrokontroléru. V první podkapitole [6.1](#) jsou detailně diskutovány programovatelné periferie mikrokontroléru ATmega168. V dalších podkapitolách jsou popsány zajímavé konstrukce programu.

K implementaci je využit jazyk C a vývojové prostředí AVR Studio 5. Projekt je tak realizován jako projekt spustitelný v AVR Studiu. Je však možné projekt přeložit i na systémech Linux bez využití AVR Studia.

6.1 Použité periferie

Na tomto místě jsou popsány jednotlivé periferie mikrokontroléru, které byly využity při návrhu a realizaci projektu.

6.1.1 GPIO (General Purpose Input Output)

Mikrokontroléry rodiny ATmega mají programovatelné vstupy/výstupy jednotlivých portů. Každý V/V pin má svoji primární funkci. Tato funkce je přiřazena na základě toho, ke které periférii port patří (UART, SPI, ADC, ...). Každý port má konfigurační, stavové a kontrolní registry. Funkce pinů lze těmito registry softwarově změnit. Takto lze například periferie UART nakonfigurovat jako univerzální V/V pin.

Typické vlastnosti GPIO:

- Až 3 možné funkce jednoho V/V pinu.
- V/V registry umožňují:
 - Nastavení pinu jako vstup nebo výstup.
 - Zápis výstupních dat.
 - Čtení logické hodnoty z pinu.
 - Nastavení jednotlivých pinů příslušného portu.

6.1.2 UART (Universal Asynchronous Receiver/Transmitter)

Rozhraní UART je jedno z velmi rozšířených sériových rozhraní. Často používaná pro komunikaci dvou zařízení. Rozhraní umožňuje využívat protokol RS232 [\[35\]](#) pro standardní asynchronní komunikaci. Mikrokontrolér obsahuje jeden programovatelný UART modul. Ten má tyto vlastnosti:

- Plně duplexní komunikace.
- Synchronní nebo asynchronní komunikace.
- Mód vysílače nebo přijímače.
- Programově nastavitelná velikost datových rámců, režim parity a mód stopbitů.
- Detekce chyb.
- 3 rozdílné vektory přerušení.

Toto rozhraní je v projektu využito pro připojení ladícího terminálu. Pro běžný chod PDU není připojení terminálu nutné.

Náhled implementace UART

Základní funkce jsou implementovány v souboru *usart.c*. Tento soubor obsahuje funkce pro inicializaci a obsluhu periferie. Funkcí pro inicializaci UART je

```
USART_Init(unsigned int baud),
```

kde parametrem je index odvozený z tabulky 19-9 v manuálu [8] v závislosti na požadované přenosové rychlosti a taktování mikrokontroléru. V tomto souboru jsou také implementovány dvě funkce,

```
void USART_TransmitByte(unsigned char data)
```

a

```
void USART_TransmitString(char *string, bool crlf_enable),
```

kteří slouží pro odeslání jednoho znaku, nebo celé řetězce (využívají funkce pro odeslání znaku). Funkci pro odeslání znaku je předáván v parametru celý 8-bitový znak. Funkce nic nevrací. Funkci pro odeslání znaku jsou předávány dva parametry, kde prvním je ukazatel na posílaný řetězec a druhým parametrem je hodnota typu *bool*, která vkládá za konec řetězce znaky <CR><LF>. Taktéž tato funkce nic nevrací.

6.1.3 SPI (Serial Peripheral Interface)

SPI je sériové periferní rozhraní. Používá se pro komunikaci mezi mikrokontroléry a jinými integrovanými obvody (EEPROM paměti, displeje, ...). V SPI koncepci zapojení hraje roli *Master*, který je spojený s ostatními *slave* zařízeními pomocí sběrnice. Všechny zařízení tak využívají stejné vodiče pro komunikaci. Adresace se pak provádí pomocí speciálních vodičů (ChipSelect – CS nebo SlaveSelect – SS), které při logické nule informují *slave* zařízení o žádosti komunikace.

Základní charakteristické vlastnosti SPI:

- Master:
 - Dává informaci o synchronizaci (generuje hodinový signál).
 - Určuje se kterým zařízením bude komunikovat – signálem CS/SS.
- Slave:
 - Vysílá pokud přijímá synchronizační hodiny a signál CS/SS je v log.0.
- Plně duplexní, 3-vodičový datový přenos.

- Programově nastavitelná rychlost přenosu.
- Programově nastavitelná polarita hodin, LSB/MSB datový přenos.

SPI rozhraní je použito pro komunikaci s Nano SocketLAN a později pak pro komunikaci s měřicími moduly.

Náhled implementace SPI

Základní funkce jsou implementovány v souboru *spi.c*. Tento soubor obsahuje funkce pro inicializaci a obsluhu periferie. Funkcí pro inicializaci SPI je `char SPI_Init(char tx_mode, char ck_divider, char sck_mode, char lsb_first)`, kde parametry nastavují režimy přenosu. Návrátovou hodnotou je stav kontrolního registru periferie. V tomto souboru jsou také implementovány další podpůrné funkce. Nejdůležitější je však funkce, která odešle a zároveň přijme znak.

`char SPI_TransmitByte(char data)`

slouží k odeslání jednoho znaku. Funkci pro odeslání znaku je předáván v parametru celý 8-bitový znak. Funkce vrací přijatý znak od zařízení *slave*. Dále jsou v tomto souboru implementovány funkce které mají specifickou funkci pro komunikaci se vzdáleným zařízením.

Abstraktní popis implementace komunikace s Nano SocketLAN je rozebrán v kapitole [4.3.1](#).

6.2 Implementace firmwaru

Základním jádrem programu je soubor *main.c*. PDU ke správnému chodu využívá právě tento firmware. Firmware zajišťuje komunikaci s Nano SocketLAN, který je mostem do sítě ethernet. Pomocí tohoto mostu získává příkazy přijaté pomocí ethernetu. Na základě těchto příkazů provede příslušnou operaci (nastavení stavu PDU) a vytvoří příslušnou zprávu, kterou předá k odeslání po síti Ethernet.

6.2.1 Hlavní program

Hlavní program je implementován v souboru *main.c*. Na začátku programu jsou spuštěny veškeré inicializace použitých periférií.

- `USART_Init(baudIndex)`; – inicializace rozhraní USART.
- `REG_Init()`; – inicializace 8-bitového výstupního portu.
- `SPI_Init(1,0,1,0)`; – inicializace SPI.

Dále se zavolá funkce `server()`, která popisuje chování PDU jakožto serveru. Funkce plní smysl konečného automatu. Konečný automat na základě svého stavu zasílá AT příkazy Nano SocketLAN modulu.

Prvním stavem automatu je stav INIT. V tomto stavu se provede nastavení Nano SocketLAN modulu pomocí AT příkazů (IP z DHCP, uzavření všech otevřených soketů, ...).

Po provedení inicializace přejde automat do stavu OPEN_SOCKET, kdy je LAN modulu zaslán AT příkaz `AT+iLTCP:20116,1<CR>` pro otevření naslouchacího TCP soketu na portu 20116 a je definován maximální počet připojených klientů číslem 1. Na příkaz modul odpoví číslem otevřeného TCP soketu `I/010<CR><LF>`. V případě neúspěšného otevření soketu

odpoví I/ERROR <číslo chyby><CR><LF>, že nastala chyba a pokus o otevření socketu se opakuje. Po provedení tohoto příkazu jsme uvedli zařízení do role serveru a přejde automat do dalšího stavu.

Následujícím stavem je LISTEN_SOCKET. V tomto stavu každou sekundu je zasílán Nano SocketLAN modulu AT příkaz AT+iLSST:010<CR>, kterým se dotazuje, zda se připojil k socketu nějaký klient. LAN modul odpoví I/(-1)<CR><LF> v případě, že na socket není připojen žádný klient. Pokud se na socket připojí klient, dostaneme odpověď od NANO SocketLAN ve tvaru I/(000)<CR><LF>, kde číselná hodnota označuje číselné označení aktivního otevřeného socketu, v tomto případě 0. Bohužel Nano SocketLAN nevlastní žádný doplňkový signál, který by dokázal informovat o připojení klienta. Z tohoto důvodu je implementováno dotazování každou vteřinu. Po detekci připojení klienta můžeme přejít do dalšího stavu.

Nyní se nacházíme ve stavu, kdy je připojený nějaký klient a chystá se PDU poslat nějaký příkaz. Tento stav je nazván WAIT_FOR_DATA. Nano SocketLAN je už v tomto případě vybavený hardwarovým signálem, který přepnutím od log.1 informuje o tom, že od klienta přišly nějaká data a mikrokontrolér si je může vyzvednout, nedává však informaci o tom, zda se klient odpojil a program by tak mohl uváznout v neustálém čekání na data. Tento stav je řešen smyčkou, která čeká na signál, že přišly data. Pokud se data v bufferu nevyskytnou do 10 sekund, je implementováno chování, jako by data přišly. Následuje přechod do dalšího stavu.

Dalším stavem je READ_DATA, kdy je Nano SocketLAN zaslán příkaz AT+iSRCV:000, kde číselnou hodnotou je číslo aktivního otevřeného socketu. Tímto příkazem je požadováno po LAN modulu, aby zaslal mikrokontroléru obsah TCP bufferu. Pokud dostane I\0, modul tím značí, že buffer je prázdný. Mikrokontrolér se tak vrátí do předchozího stavu WAIT_FOR_DATA, kdy čeká na data. V případě, že odpovědí je text I/ERROR (203)<CR><LF>, znamená to, že klient se odpojil. Použije se tedy číslo aktivního otevřeného socketu a pomocí AT příkazu AT+iSCLS:<číslo aktivního socketu><CR> se aktivní socket zavře. Následně se přejde do stavu LISTEN_SOCKET, kdy se opět čeká na připojení klienta. Pokud mikrokontrolér přijme zprávu, odpovídající tabulce 4.2, kde je definován aplikační protokol, je provedeno zpracování příkazu a zajištěn přechod do dalšího, posledního stavu.

Posledním stavem je WRITE_DATA. V do tohoto stavu se automat dostane pouze v případě, že odpovídá centrálnímu ovládacímu systému aplikačním protokolem v tabulce 4.3. Na základě přijaté předchozí zprávy se provede příslušná akce, například nastavení stavu registrů, vytvoří se příslušná odpověď a pomocí AT příkazu

AT+iSSND%:<číslo aktivního socketu>,<počet znaků zprávy>:<zpráva> se odešle příslušná <zpráva>. Po úspěšném odeslání se automat vrátí zase do stavu WAIT_FOR_DATA.

Ukázka principu funkce implementovaného automatu

Dotaz na IP:

Příkaz: AT+iIPA?<CR>

IP adresa modulu:

Odpověď: 10.129.99.50<CR><LF>I/OK<CR><LF>

Uzavření TCP socketu s číslem 10 při inicializaci:

Příkaz: AT+iSCLS:010<CR>

TCP socket 10 není otevřen:

Odpověď: I/ERROR (200)<CR><LF>

Uzavření TCP socketu s číslem 11 při inicializaci:

Příkaz: AT+iSCLS:011<CR>

TCP socket 11 není otevřen:

Odpověď: I/ERROR (200)<CR><LF>

Otevření naslouchacího TCP socketu, port 20116,max počet klientů 1:

Příkaz: AT+iLTCP:20116,1<CR>

TCP socket otevřen, číslo naslouchacího socketu je 10:

Odpověď: I/010<CR><LF>

Dotaz na aktivní socket(je někdo připojen?):

Příkaz: AT+iLSST:010<CR>

Vrací -1, nikdo připojen není:

Odpověď: I/(-1)<CR><LF>

Dotaz na aktivní socket(je někdo připojen?):

Příkaz: AT+iLSST:010<CR>

Vrací 0, klient je připojen, číslo aktivního socketu je 0:

Odpověď: I/(000)<CR><LF>

Čekání na HW signál o přijetí příkazu:

Cekam...

Přijmi zprávu z TCP bufferu s číslem aktivního socketu 0

Příkaz: AT+iSRCV:000<CR>

Přijata zpráva SET 10101010; :

Odpověď: I/13:SET 10101010;<CR><LF>

Odeslání odpovědi COS:

Příkaz: AT+iSSND\%:000,13:ACK 10101010;<CR>

Čekání na HW signál o přijetí příkazu:

Cekam...

...

Kapitola 7

Problémy, testování a další vývoj

Navržený systém byl fyzicky vyroben, realizován a vyzkoušen. Tato kapitola popisuje postup oživení jednotlivých modulů a průběh testování.

7.1 Konstrukce

Základem konstrukce řešení projektu jsou desky plošných spojů a jejich osazení. Pokud nám to okolnosti dovolují, je doporučeno součástky sázet postupně. A to tak, že nejdříve sázíme součástky, které konstrukčně nebrání k osazení dalších, větších součástek.

7.1.1 Základní jednotka

Pozice součástek na DPS je znázorněna v příloze [C.1](#). Postup při osazování DPS byl následující. Jako první součástky byly osazeny smd rezistory, keramické, tantalové kondenzátory. Poté byly osazeny stabilizátory IC1 a IC2. Po připojení stejnosměrného napájecího napětí 12 V je otestováno výstupní napětí stabilizátorů. Na výstupu IC1 by mělo být napětí 5 V, na výstupu IC2 3,3 V.

Jestliže naměříme uvedené hodnoty, můžeme pokračovat v osazení dalších integrovaných obvodů IC3, IC4, IC5. Pro mikrokontrolér ATmega je vyrobena univerzální patice s precizními vývody. ATmega tedy osadíme na speciální DPS. Posledním krokem je osazení pinheadrů, ML konektorů.

Po osazení byla deska zkontrolována mikroskopickou USB kamerou, zda při pájení nezůstaly zbytky cínu, které by mohly zapříčinit zkratu a očistit je technickým lihem.

7.1.2 Spínací a měřicí modul

Součástky spínacího a měřicího modulu jsou na desce rozmístěny dle obrázku [C.2](#). Postup je shodný jako u základní jednotky. Nejdříve jsou osazeny SMD součástky, pak jsou osazeny klasické a rozměrné součástky. Je třeba dát pozor na osazování měřicího modulu, jehož pouzdro je SSOP20 a má vývody velmi blízko u sebe.

Pro naše účely budou osazeny pouze dva okruhy, a to spínaný a spínatelný. Testování bylo napřed provedeno bez osazení relé, pouze s osazenou informační diodou. Následně pak bylo připojeno relé a zapojen spínatelný okruh (230 V).

7.2 Oživení

V této podkapitole je popsán postup zprovoznění a oživení prototypu.

7.2.1 Programování firmware ATMega168

Firmware byl vyvíjen a implementován ve vývojovém prostředí AVR Studio 4.18. Jako programátor byl využit USB programátor PRESTO od českého výrobce ASIX [6]. Programování bylo prováděno pomocí rozhraní JTAG. K překladu v AVR Studiu je nutné použít projektový soubor *IntelligentOutlets.aps*. Protože k překladu využíváme otevřenou knihovnu AVR lib [10], je nezbytné tuto knihovnu doinstalovat. Aby bylo AVR Studio schopné překladu, je potřebné také doinstalovat překladač pro mikrokontroléry AVR, WinAVR [21] je překladač pro operační systém Windows.

7.2.2 Propojení bloků

Pro mikrokontrolér je navržena speciální patice. Mikrokontrolér tak stačí pohodlně zasunout do precizní patice. Toto řešení tak umožňuje snadnou výměnu mikrokontroléru.

Pro Nano Socket LAN je taktéž navržena patice, kde jsou použity dva speciální 10-ti pinové pin-headry o rozteči 2mm. Nano SocketLAN tak stačí také snadno zasunout do patice a zajistit šroubky.

Pro propojení základní jednotky a spínacího a měřicího modulu jsou využity dva plošné kabely. Jeden kabel o 16 vodičích je na základě vzdálenosti k jednotlivým modulům roztrhán po párech a příslušně zkrácen ke každému modulu. Pomocí každého páru vodičů je pak přiváděno napájení jednotlivých relé. Po každém páru tak teče 30mA o napětí 12 V.

Další plošný kabel o počtu vodičů 10 je využit jako sběrnice pro komunikaci s měřicími moduly. Tento kabel je složen z více plošných kabelů, kde 6 vodičů je vedeno jako sběrnice SPI, reset a napájení a další 4 vodiče jsou použity individuálně ke každému modulu jako ChipSelect a signál přerušení.

7.2.3 Testování

Systém byl v průběhu vývoje testován průběžně. Nejprve byly implementovány funkce pro operaci s výstupním portem a záchytnými registry, které byly otestovány připojením testovacích LED diod. Tyto LED diody pak sloužili prakticky po celou část vývoje firmwaru. Správnost výstupu registrů byla otestována pomocí LED diod. Jakmile byl firmware implementován, bylo připojeno 8 ks relé.

Po implementaci práce s výstupním portem byly implementovány funkce komunikačního rozhraní UART pro připojení terminálu. Protože vestavěný systém nemá žádný standardní výstup, tyto funkce nám posloužily pro výpis textových informací. Jako terminálová aplikace byl použit software Herkules [25] od firmy HW-group. Tento software je plně kompatibilní s Windows 7. Umí jak sériový terminál, tak TCP/UDP klient/server, což posloužilo k testování TCP komunikace.

Po připravení testovacího rozhraní byla snaha implementovat komunikaci s modulem Nano SocketLAN, kdy komunikace probíhá přes rozhraní SPI a pomocí AT příkazů. Při vývoji tak bylo využito právě debugovacího terminálu a pro jasné pochopení principu komunikace byl využit logický analyzátor Fabulous [33] používající port LPT. Správnost komunikace s LAN modulem byla otestována logickým analyzátozem a výpisy na ladící terminál.

Po otestování správnosti komunikace se síťovým mostem mohla být implementována konfigurace TCP serveru a následně pak aplikačního TCP protokolu. K testování TCP komunikace i výpisu debugovacích informací nám opět sloužila aplikace Herkules.

Nakonec bylo testováno vzdálené spojení s COS, který byl navržený a implementovaný studentem Janem Martiškou.

7.3 Problémy při vývoji

Při jakémkoliv výzkumu, vývoji se vyskytnou vždy nějaké problémy. Někdy menší, někdy větší. V projektu se jeden vyskytl větší problém, který projekt výrazně ohrozil. Popis problému je uveden v následující podkapitole.

7.3.1 Problémová SPI komunikace s Nano SocketLAN

Princip komunikace s Nano SocketLAN je uveden v kapitole 4.3.1. Je možné si všimnout z obrázku 4.5, že Nano SocketLAN používá HW signál *SPI_INT*, na obrázku jako signál Control, k informování *mastera* o tom, zda *master* může zapisovat/číst. Protože zápis příkazu do LAN modulu je umožněn jen v případě, pokud je *SPI_INT=0*, je tento signál sledován. Po přijetí příkazu LAN modul připravuje odpověď. Jakmile je odpověď připravena k odeslání, je signál *SPI_INT=1*. Po vyprázdnění bufferu s odpovědí je signál opět nastaven *SPI_INT=0*. Problém vzniká v době, kdy po odeslání příkazu či při odeslání odpovědi nezmění Nano SocketLAN stav signálu *SPI_INT* a není ochoten přijímat další příkazy.

Příčinou byly přeslechy na rozhraní SPI. V prvotním konceptuálním návrhu se uvažovalo napájet mikrokontrolér 5 V DC a NanoSocketLAN 3,3 V DC. Napěťové úrovně na komunikačním rozhraní upravit pomocí zenerových diod. Po dlouhém laborování s prototypem bylo zjištěno, že zenerovy diody jsou na úpravu napěťových úrovní velmi nevhodné, jelikož způsobují šum a tím tvoří přeslechy na rozhraní SPI. Problém byl vyřešen napájením všech logických obvodů a mikrokontroléru 3,3 V. Protože bylo řešení problému odhaleno velmi pozdě, v návrhu je stále zachován stabilizátor napětí 5 V DC, který by se mohl v budoucnu vyloučit.

7.4 Rozšíření do budoucna

Celý systém je navržen velmi univerzálně a na základě aktuálních požadavků. ATMega168 svými paměťovými prostředky postačila pro vývoj projektu s dostatečnou rezervou. Hůře je na tom však se svými periferiemi, které byly v projektu téměř všechny využity. Omezily tak různé možnosti rozšíření při vývoji. Nano SocketLAN je velmi výkonné zařízení. Bylo také zjištěno, že měřící moduly pro každou zásuvku zvlášť zaberou mnoho místa. Pro minimalizaci celé PDU by bylo vhodné realizovat měření spotřeby pouze na všech zásuvkách současně.

Systému by v budoucnu bylo možné přidat následující vlastnosti:

- Využití webservru ke konfiguraci, který nabízí Nano SocketLAN.
- Osazení a implementování měřících modulů.
- Rozšířit aplikační TCP protokol o možnosti dotazů na spotřebu dané zásuvky.

- Přidání 7 segmentového displeje pro zobrazení aktuální spotřeby celé PDU (vlastnost komerčních produktů).
- Univerzální DPS základní jednotky pro 8-12-16 zásuvek.
- Odebráním patice mikrokontroléru a snaha i minimalizaci rozměrů.
- V případě potřeby snadná přestavba na Wi-Fi, vyžaduje však úpravu programu.

Kapitola 8

Závěr

V této bakalářské práci je diskutována problematika vestavěných systémů a možnosti vzdáleného řízení a monitorování. Ethernetové služby jsou dnes velmi rozšířené i ve vzdálené správě nejen vestavěných systémů. Využití ethernetových služeb se k tomuto účelu stává populární a atraktivní možností, jak jednoduše tyto služby využít.

Dokument ukazuje kompletní návrh PDU kompletně od začátku. Ukazuje rozbor použitých technologií, následně pak výběr platformy, jednotlivých komponent a součástí až po implementaci firmwaru PDU a praktickou realizaci PDU. Protože je návrh od samého začátku, je velmi důležité, aby při stavbě takového zařízení byl správný postup při návrhu hardwaru, současně softwaru, aby jednotlivé součásti byly kompatibilní. Výsledkem práce je funkční, otestované PDU schopné pracovat téměř bez žádných nedostatků vůči zadání.

Základní jednotka 5.2 byla navržena tařka univerzálně a je ji možno využít jako podobný vestavěný systém se síťovým rozhraním bez velkých nároků na úpravu. Na univerzální vstupy/výstupy je možné připojit jiné periferie, než tomu jsou původně určené. Po úpravě zdrojového kódu firmwaru je možné řídit, či sledovat další systémy a periferie s možností připojení do sítě ethernet. Navržený systém tak poskytuje vhodný most mezi počítačovou sítí a vestavěným systémem, který je možné modifikovat k jiným účelům. Základní jednotka se dá také použít jako vývojový kit k Nano SocketLAN modulu.

Finální prací na projektu byla jeho praktická a fyzická realizace. Po realizaci se ukázaly dostatky i nedostatky návrhu. Výsledné zařízení pak bylo možno porovnat s komerčními produkty. Podařilo se tak zkonstruovat amatérské zařízení, v ceně zhruba 4 000 Kč s DPH dosahující téměř stejných funkčních vlastností jako profesionální PDU startující na ceně 14 000 Kč s DPH.

V příloze D a na přiloženém CD jsou k dispozici fotografie a obrázky zhotoveného PDU.

Literatura

- [1] *74AC573 Data Sheet: 8-Bit Latch register*. [Online; navštíveno 13.12.2010].
URL <http://pdf1.alldatasheet.com/datasheet-pdf/view/21737/STMICROELECTRONICS/74AC573.html>
- [2] *74HCT563 Data Sheet: 8-Bit Latch register, inverted*. [Online; navštíveno 13.12.2010].
URL <http://www.alldatasheet.com/datasheet-pdf/pdf/155778/PHILIPS/74HCT563.html>
- [3] *7805 Data Sheet - 1.0 A Positive Voltage Regulator*. [Online; navštíveno 11.12.2010].
URL http://www.gme.cz/_dokumentace/dokumenty/934/934-051/dsh.934-051.1.pdf
- [4] *ADE7763 Data Sheet: Single-Phase Active and Apparent Energy Metering IC*.
[Online; navštíveno 10.12.2010].
URL http://www.analog.com/static/imported-files/data_sheets/ADE7763.pdf
- [5] Analog Devices. [Online; navštíveno 10.12.2010].
URL <http://www.analogdevices.com/>
- [6] Asix. [Online; navštíveno 4.2.2010].
URL <http://asix.cz/>
- [7] *AT+i Programmer's Manual for iChip CO2128*. [Online; navštíveno 2.12.2010].
URL http://www.spezial.cz/pdf/ATi_Programmers_Manual_8_32.pdf
- [8] *ATMEGA48/88/168 Data Sheet - 8-bit Microcontroller with 4/8/16K Bytes In-System Programmable Flash*. [Online; navštíveno 2.12.2010].
URL http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf
- [9] Atmel. [Online; navštíveno 30.11.2010].
URL <http://www.atmel.com>
- [10] AVRlibc: Knihovna v jazyce C pro mikrokontroléry AVR. [Online; navštíveno 1.2.2011].
URL <http://www.nongnu.org/avr-libc/>
- [11] CadSoft online. [Online; navštíveno 1.4.2011].
URL <http://cadsoft.de/>
- [12] Cirrus Logic. [Online; navštíveno 10.12.2010].
URL <http://www.cirruslogic.com/>

- [13] ConnectOne. [Online; navštíveno 2.12.2010].
URL <http://www.connectone.com>
- [14] *CS5463 Data Sheet: Single Phase, Bi-directional Power/Energy IC*. [Online; navštíveno 10.12.2010].
URL http://www.cirrus.com/en/pubs/proDatasheet/CS5463_F2.pdf
- [15] *ENC28J60 Data Sheet - Stand-Alone Ethernet Controller with SPI Interface*. [Online; navštíveno 2.12.2010].
URL <http://www.alliedelec.com/Images/Products/Datasheets/BM/MICROCHIP/383-0184.PDF>
- [16] Freescale Semiconductor. [Online; navštíveno 30.11.2010].
URL <http://www.freescale.com>
- [17] *LF33 Data Sheet - 3.3V Voltage Regulator*. [Online; navštíveno 11.12.2010].
URL http://www.gme.cz/_dokumentace/dokumenty/934/934-038/dsh.934-038.1.pdf
- [18] Microchip. [Online; navštíveno 30.11.2010].
URL <http://www.microchip.com/>
- [19] *Nano SocketLAN Data Sheet - SPI/Ethernet bridge*. [Online; navštíveno 2.12.2010].
URL http://connectone.com/media/upload/Nano_Socket_LAN_DS.pdf
- [20] *ULN2803A Data Sheet: Tranzistorové pole*. [Online; navštíveno 13.12.2010].
URL <http://www.datasheetcatalog.org/datasheet/SGSThompsonMicroelectronics/mXssxrt.pdf>
- [21] WinAVR: překladač pro mikrokontroléry AVR. [Online; navštíveno 1.2.2011].
URL <http://winavr.sourceforge.net/>
- [22] Stránky výrobce a prodejce PDU - firma APC. 2011, [Online; navštíveno 23.11.2010].
URL <http://www.apc.com/>
- [23] Stránky zahraničního prodejce PDU. 2011, [Online; navštíveno 23.11.2010].
URL <http://www.kvmchoice.com/>
- [24] ConnectOne: SPI Implementation Notes for iChip CO2128, [Na vyžádání od firmy Spezial electronic].
URL <http://www.spezial.cz>
- [25] Group, H.: Hercules terminal. [Online; navštíveno 15.2.2011].
URL http://www.hw-group.com/products/hercules/index_cz.html
- [26] MARTIŠKA, J.: *Správa elektrických zásuvek v laboratoři přes dotykový panel*. Bakalářská práce, FIT VUT v Brně, 2011.
- [27] ODVÁRKA, P.: Ethernet. 2000, [Online; navštíveno 25.11.2010].
URL <http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=1&clanekID=17>

- [28] ODVÁRKA, P.: Fyzická a linková vrstva ISO OSI. 2000, [Online; navštíveno 24.11.2010].
URL <http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&clanekID=19>
- [29] PETERKA, J.: Linková vrstva - II. 1992, [Online; navštíveno 25.11.2010].
URL <http://www.earchiv.cz/a92/a219c110.php3>
- [30] PETERKA, J.: Sedm vrstev ISO/OSI. 1996, [Online; navštíveno 27.11.2010].
URL <http://www.earchiv.cz/a96/a625k150.php3>
- [31] PETERKA, J.: Ethernet. 1997, [Online; navštíveno 24.11.2010].
URL <http://www.earchiv.cz/a97/a716k150.php3>
- [32] POSTEL, J.: TRANSMISSION CONTROL PROTOCOL. RFC 793, 1981, [Online; navštíveno 20.1.2011].
URL <http://www.ietf.org/rfc/rfc793.txt>
- [33] WALLE, B.: Fabulous Logic Analyzer. [Online; navštíveno 1.2.2011].
URL <http://tfla-01.berlios.de/>
- [34] Wikipedie: JTAG — Wikipedie: Otevřená encyklopedie. 2009, [Online; navštíveno 19.1.2011].
URL http://cs.wikipedia.org/wiki/Joint_Test_Action_Group
- [35] Wikipedie: RS-232 — Wikipedie: Otevřená encyklopedie. 2009, [Online; navštíveno 19.1.2011].
URL <http://cs.wikipedia.org/w/index.php?title=RS-232&oldid=3976185>
- [36] Wikipedie: TCP — Wikipedie: Otevřená encyklopedie. 2011, [Online; navštíveno 19.1.2011].
URL http://cs.wikipedia.org/wiki/Transmission_Control_Protocol
- [37] ZÁHLAVA, V.: *Návrh a konstrukce desek plošných spojů*. Praha, CZ: ben, první vydání, 2010, ISBN 978-80-7300-266-4.
- [38] ČSN-33-2000-4-41: *Elektrické instalace nízkého napětí: Ochranná opatření pro zajištění bezpečnosti - Ochrana před úrazem elektrickým proudem*. Praha: Český normalizační institut, 2007, 68 s.
- [39] ČSN-33-2130: *Elektrotechnické předpisy: Vnitřní elektrické rozvody*. Praha: Český normalizační institut, 1983, 40 s.

Seznam příloh

- A** Normy pro bezpečnost a rozvod elektrické energie [38] a [39].
- B** Schémata zapojení.
- C** Osazení desek plošných spojů.
- D** Fotografie realizovaného projektu.
- E** Datový nosič se zdrojovými soubory a podklady pro konstrukci. Podrobný obsah je vypsán v souboru README v kořenovém adresáři CD.

Dodatek A

Normy pro rozvod elektrické energie

Pro rozvod elektrické energie je třeba splnit normy pro elektrické rozvody ČSN 33 2130 [39] a pro bezpečnost proti úrazu elektrickým proudem ČSN 33 2000-4-41 [38].

A.1 ČSN 33 2130 – Vnitřní elektrické rozvody

Podkapitola obsahuje citace normy ČSN 33 2130/Zásuvkové obvody [38].

A.1.1 Zásuvkové obvody

- Zásuvkové obvody se zřizují pro připojení spotřebičů vidlicí do zásuvky. Na zásuvkové obvody lze podle potřeby pevně připojit jednoúčelové spotřebiče pro krátkodobé použití do celkovéh příkonu 2000V.A.
- Základní požadavky na umístění, osazení a užití zásuvek jsou uvedeny v ČSN 33-2180 a jsou doplněny požadavky v následujících 7 odrážkách.
- Zásuvky musí mít ochranný kolík připojený na ochranný vodič. Jednofázové zásuvky se doporučuje připojit tak, aby ochranný kolík byl nahoře a nulový (střední) vodič, aby byl připojen na pravou dutinku při pohledu z předu – viz. též ČSN 33-2180.
- Zásuvky musí být voleny podle napětí a proudové soustavy. Při použití dvou napěťových soustav musí být zásuvky vždy zásuvky nezáměnné.
- Zásuvky musí vyhovět požadavkům:
 - ČSN EN 60309-1 ed.3, nebo
 - ČSN EN 60309-2 ed.3, pokud je požadována zaměnitelnost.

Zásuvky se jmenovitým proudem nepřesahujícím 16A musí odpovídat příslušným národním normám. (Národní vzory zásuvek pro ČR jsou uvedeny v IEC-TR 60083:2006)

- Jednofázové zásuvky. Na jeden zásuvkový obvod lze připojit nejvýše 10 zásuvkových vývodů (vícenásobná zásuvka se považuje za jeden zásuvkový vývod), přičemž celkový instalovaný příkon nesmí překročit 3680V.A. při jistění 16A(2300V.A. při jistění 10A).

- Zásuvky s dvojitými svorkami se doporučuje připojovat smyčkováním.
- Vícenásobná zásuvka je určena pro připojení na jeden obvod a nesmí se připojit na dva různé obvody, ani se nesmí přerušit propojení obou zásuvek.
- Trojfázové zásuvky. Na jeden trojfázový obvod lze připojit několik trojfázových zásuvek na stejný jmenovitý proud. Trojfázové zásuvky o různém jmenovitém proudu se nesmějí zapojovat do stejného obvodu.
- Dimenzování a jistění zásuvkových obvodů. Vedení zásuvkových obvodů se jistí jističi, nebo pojistkami, nebo jiným jisticím prvkem se jmenovitým proudem odpovídajícím nejvýše jmenovitému proudu zásuvky; vedení musí mít takový průřez, aby bylo předřazeným jisticím prvkem jistěno proti přetížení i zkratu. Všechny svorky, kterými vedení zásuvkových obvodů prochází, musí být dimenzovány aspoň na jmenovitý proud jisticího prvku, kterým je obvod jistěn. Jisticí prvek v zásuvkovém obvodu jistí pouze rozvod k zásuvkám(vedení) a nejistí obvykle připojený spotřebič.
- Zásuvkové obvody do 20A musí mít doplňkovou ochranu tvořenou proudovým chráničem s vybavovacím reziduálním proudem, nepřekračujícím 30mA v souladu s ČSN 33-2000-4-41 ed.2. Toto opatření se vztahuje i na trojfázové zásuvky připojené na obvod s jistěním do 20A.
- U předchozí odrážky není nutno uplatňovat u zásuvek nepřístupných laické veřejnosti a zásuvek pro speciální druh zařízení(například zařízení kancelářské a výpočetní techniky velkého rozsahu nebo pro chladicí a mrazicí zařízení potravin velkého objemu, tj. zásuvky pro napájení zařízení, jehož nežádoucí vypnutí by mohlo být příčinou značných škod – viz. ČSN 33-2000-4-41 ed.2).
- Pro elektrické potřebiče, u nichž je to výrobcem určeno v návodu k montáži, se zřizuje samotný zásuvkový obvod.

A.2 ČSN 33 2000-4-41 – Ochranná opatření před úrazem elektrickým proudem

Tato podkapitola cituje normy ČSN 33 2000-4-41/Zásuvkové obvody [38], které jsou důležité k dodržení bezpečnosti před úrazem elektrickým proudem.

A.2.1 Ochrana před nebezpečným dotykem živých částí

- **Ochrana izolací:** Spočívá v zajištění takové izolace el. zařízení, která znemožní dotyk živé části, případně proniknutí nebezpečného napětí na povrch a na přístupné živé části chráněného předmětu.
- **Ochrana kryty nebo přepážkami:** Kryty nebo přepážky jsou určeny k tomu, aby zabráňovaly jakémukoliv dotyku živých částí. Živé části musí být uvnitř krytu nebo za přepážkou, které zajišťují krytí minimálně IP 2X kromě případů, jako jsou některé objímky žárovek, otvory zásuvek nebo pojistek a pod. V těchto případech musí být provedena vhodná opatření proti nahodilému dotyku živých částí a musí se zajistit, pokud je to proveditelné, aby si osoby byly vědomi toto, že se otvorem mohou dotknout živých částí. Tam kde je nutné otevřít kryty, odstranit přepážky, nebo odstranit část krytů, musí to být možno pouze:

- s použitím klíče nebo nástroje
 - po samočinném odpojení živých částí
 - kde vnitřní přepážka, zajišťující krytí alespoň IP 2X brání dotyku živých částí a taková přepážka je odstranitelná pouze pomocí klíče nebo nástroje.
- **Ochrana zábranami:** Zábrany jsou určeny k tomu, aby zabránily nahodilým dotykem živých částí, nikoliv však úmyslnému dotyku záměrným obejitím přepážky. Zábrany musí bránit:
 - neúmyslnému přiblížení osoby k živým
 - nahodilému dotyku živých částí během činnosti nekrytým zařízení v běžném provozu.

Zábrany mohou být odstraněny bez použití klíče nebo nástroje, ale musí být zajištěny tak, aby se zabránilo jejich neúmyslnému odstranění. Dále se musí rozlišit, zda do těchto prostor mají přístup pouze pracovníci s požadovanou způsobilostí nebo pouze laici.

A.2.2 Ochrana před nebezpečným dotykem neživých částí

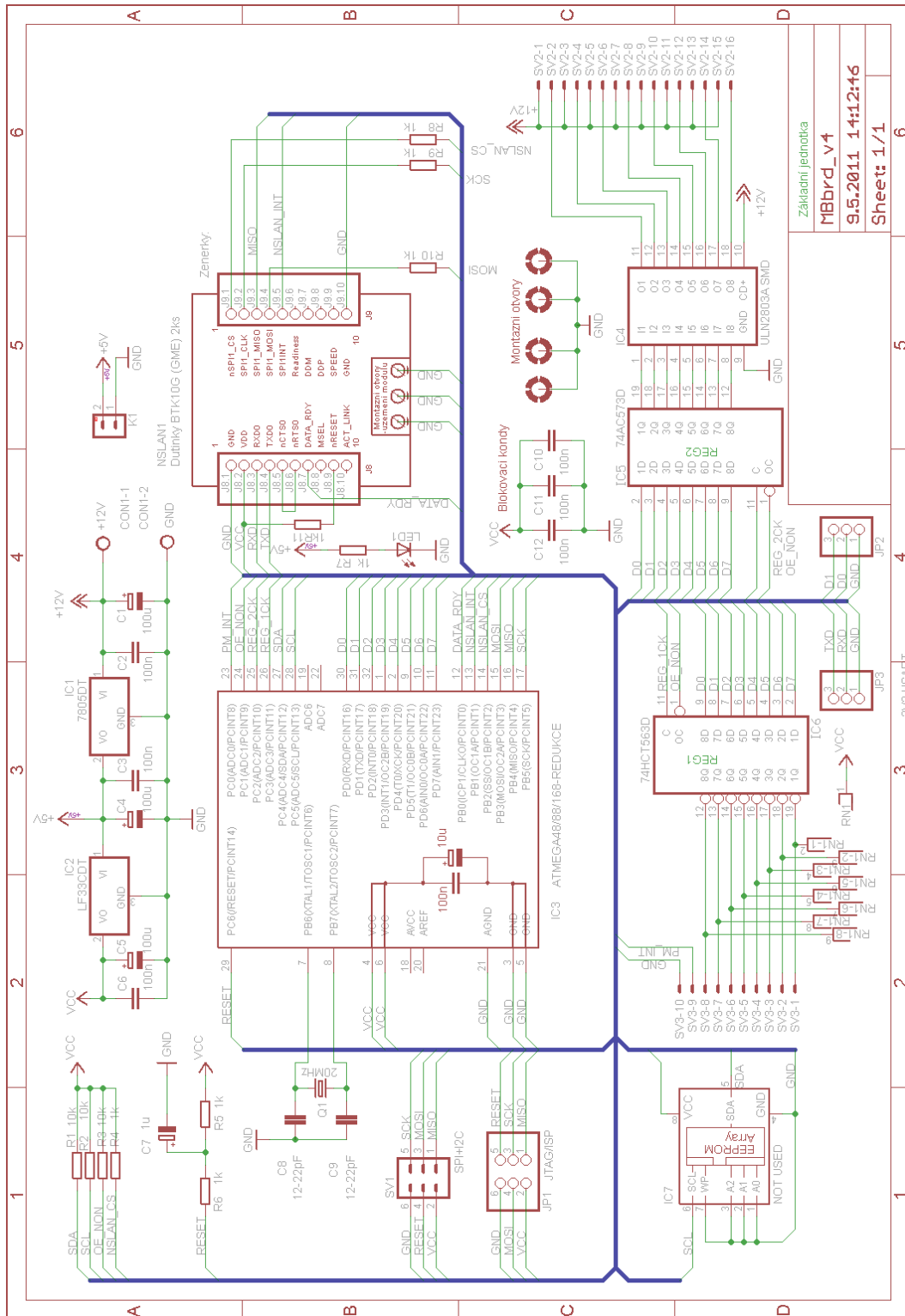
- **Ochrana nulováním:** V případě porušení izolace, protéká poruchový proud nulovacím vodičem k uzlu zdroje a fázovým vodičem zpět k místu poruchy. Nová norma váže použitelnost této ochrany především na dodržení vypínacích dob a na vyrovnání potenciálu v okolí chráněných předmětů spojováním.

Dodatek B

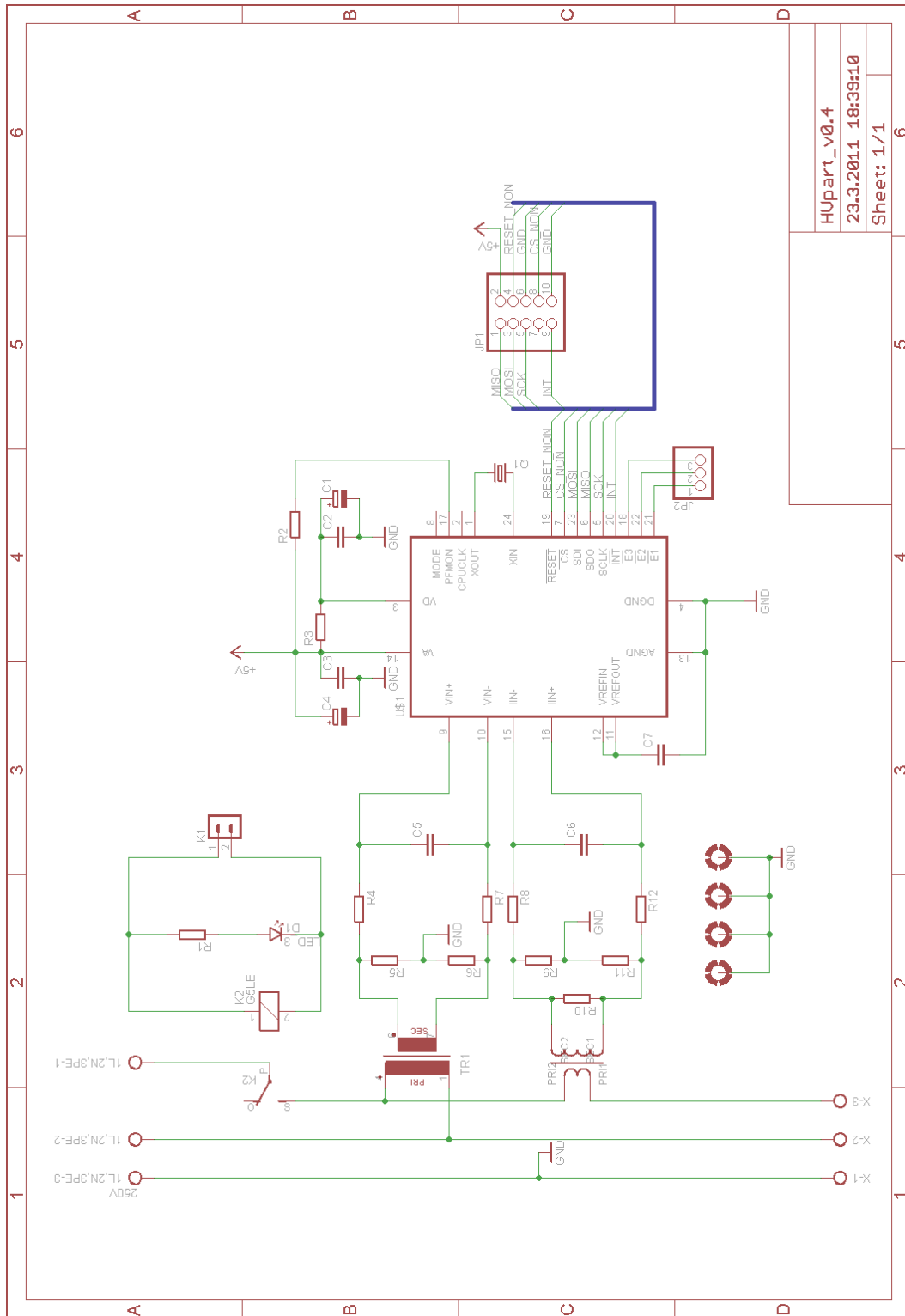
Schémata zapojení

B.1 Schéma zapojení základní jednotky

B.2 Schéma zapojení spínacího a měřicího modulu



Obrázek B.1: Schéma základní jednotky



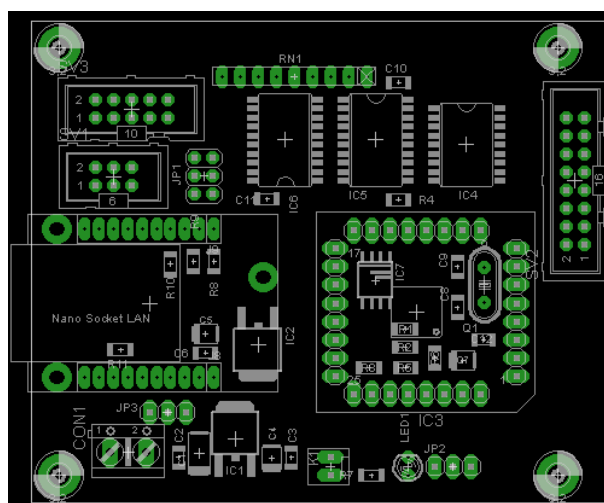
HUpart_v0.4	6
23.3.2011 18:39:10	5
Sheet: 1/1	6

Obrázek B.2: Schéma spínací a měřící části

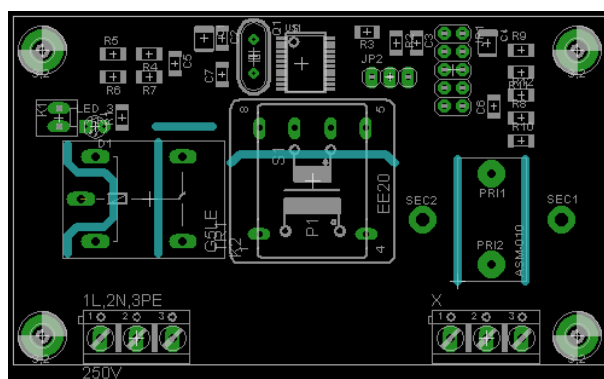
Dodatek C

Osazení desek plošných spojů

C.1 Základní jednotka



C.2 Měřicí a spínací modul



Dodatek D

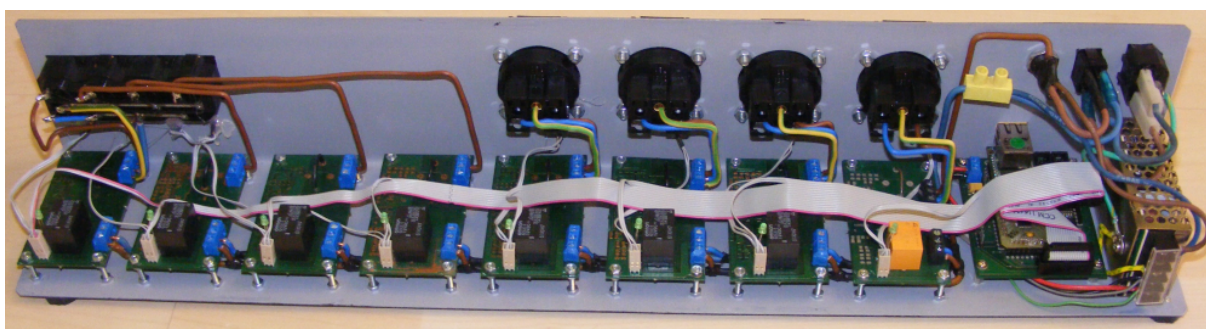
Fotografie prototypu

D.1 Fotografie prototypu zepředu



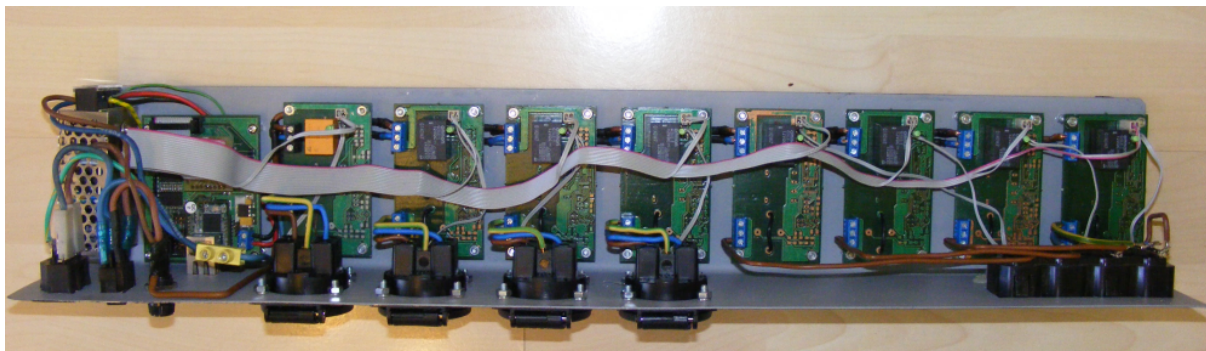
Obrázek D.1: Pohled na prototyp PDU zepředu.

D.2 Fotografie prototypu zezadu



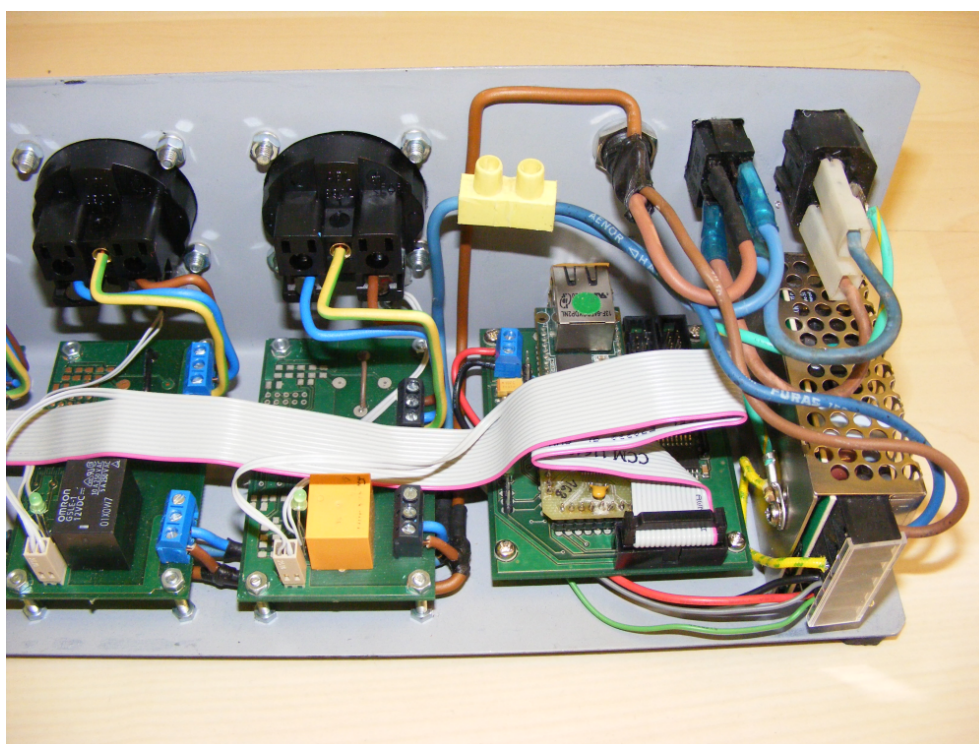
Obrázek D.2: Pohled na prototyp PDU zezadu.

D.3 Fotografie prototypu shora



Obrázek D.3: Pohled na prototyp PDU shora.

D.4 Fotografie detailu hlavní části



Obrázek D.4: Detail hlavní části prototypu PDU.