

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

Ethernet reflektor

Leden 2015

Bakalant: Michal Volek

Vedoucí práce: Ing. Pavel Bezpalec PhD

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 5.1.2015



.....

Podpis bakalanta

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Volek Michal**

Studijní program: Komunikace, multimédia a elektronika
Obor: Síťové a informační technologie

Název tématu: **Ethernet reflektor**

Pokyny pro vypracování:

Navrhnete a následně v laboratorních podmínkách realizujete zařízení, které bude sloužit jako zpětná smyčka pro testování služeb na rozhraní Ethernet.

Seznam odborné literatury:

- [1] Carrier sense multiple access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE802.3-2008, prosinec 2008, ISBN 973-0-738-15796-2.
- [2] Spurgeon, Ch.: Ethernet: The Definitive Guide. O'Reilly Media. February 2000. ISBN 978-1-565-92660-8.

Vedoucí: Ing. Pavel Bezpalec, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015



prof. Ing. Boris Šimák, CSc.
vedoucí katedry



V Praze dne 9. 12. 2013



prof. Ing. Pavel Ripka, CSc.
děkan

Anotace:

Tato bakalářská práce se věnuje návrhu a realizaci zpětné smyčky na rozhraní ethernet. Pro komunikaci na ethernetovém rozhraní je využit mikrokontrolér ENC28J60 od firmy Microchip, a řízení tohoto mikročipu zajišťuje mikropočítač ATmega16 výrobce Atmel. Řídící program byl vytvořen ve vývojovém prostředí BASCOM-AVR.

Klíčová slova: Zpětná smyčka, Ethernet, ENC28J60

Summary:

This final project deals with designing and creating of loopback on Ethernet interface. The communication on Ethernet interface is solved by mikrochip ENC28J60, create by Microchip, and it is controled by ATmega16 from manufacturer Atmel. Control program was created in develop enviroment BASCOM-AVR.

Index terms: Loopback, Ethernet, ENC28J60

OBSAH

1	Úvod.....	7
1.1	Historie Ethernetu	7
1.2	ISO/OSI model	8
1.2.1	Aplikační vrstva	8
1.2.2	Prezenční vrstva	9
1.2.3	Relační vrstva.....	9
1.2.4	Transportní vrstva	9
1.2.5	Síťová vrstva	9
1.2.6	Linková vrstva.....	9
1.2.7	Fyzická vrstva	9
2	Technologie Ethernet (IEEE 802.3).....	11
2.1	LLC.....	11
2.2	MAC	11
2.2.1	CSMA/CD.....	11
2.3	Typy ethernetových rámců	12
2.4	Formát ethernetového rámce	12
2.4.1	Preamble.....	13
2.4.2	Start Frame Delimiter.....	14
2.4.3	Pole Adres	14
2.4.4	Pole délka/typ.....	14
2.4.5	Pole klientských dat	15
2.4.6	Pole Pad.....	15
2.4.7	Pole Frame Check Sequence (FCS).....	15
2.4.7.1	Výpočet CRC	15
3	Teoretický rozbor	17
3.1	Základní vlastnosti	17
3.1.1	Realizace pomocí hradel a posuvných registrů.....	17
3.1.2	Softwarové zpracování dat v mikrokontroléru.....	18
3.2	Napájení	19
3.2.1	Napájení technologií Power over Ethernet	19
3.2.2	Napájení pomocí USB konektoru	20
3.2.3	Napájení pomocí externího zdroje	20
3.2.3.1	Zdroj klasické koncepce.....	20
3.2.3.2	Spínaný zdroj	21
4	Realizace	23
4.1	Vybraný způsob realizace a napájení.....	23

4.2	Komunikační mikrokontrolér ENC28J60	24
4.2.1	Základní vlastnosti ENC28J60.....	24
4.2.2	Přehled pinů ENC28J60.....	25
4.2.2.1	Napájecí piny	25
4.2.2.2	Řídicí piny.....	26
4.2.2.3	Komunikační piny.....	27
4.2.2.4	Indikační piny	28
4.2.3	Organizace paměti ENC28J60	28
4.2.3.1	Kontrolní registry ENC28J60	29
4.2.3.2	Organizace Ethernetového Bufferu.....	31
4.2.4	SPI rozhraní.....	32
4.2.5	SPI rozhraní v ENC28J60.....	33
4.3	Řídicí mikrokontrolér ATmega16	35
4.3.1	Základní vlastnosti ATmega16	35
4.4	Vývojové sady	38
4.5	Elektrická schémata	40
4.5.1	Elektrické schéma napájecího zdroje.....	40
4.5.1.1	Určení parametrů stabilizátoru IC1	40
4.5.1.2	Výpočet parametrů transformátoru TR1	41
4.5.1.3	Výpočet kapacit kondenzátorů C1, C2	42
4.5.2	Elektrické schéma hlavní desky.....	44
4.5.3	Elektrické schéma vývojového komunikačního kitu	45
5	Program	47
5.1	Nastavení SPI komunikace	47
5.2	Select bank.....	47
5.3	Odeslání paketu.....	48
6	Oživení	50
7	Závěr.....	51
8	Seznam zkratk a symbolů.....	52
9	Použitá literatura	53
10	Seznam Obrázků.....	54
11	Seznam tabulek.....	55
12	Seznam rovnic	56
13	Seznam příloh.....	57

1 Úvod

Termín Ethernet reflektor neboli Ethernetové zrcadlo, označuje zařízení vytvářející smyčku mezi vysílačem a přijímačem. Jeho princip je velmi jednoduchý: vysílací strana odešle Ethernetový rámec, Ethernetový reflektor tento rámec přijme, prohodí MAC adresy a odešle rámec zpět. Podívejme se nyní, kde se vlastně technologie Ethernet vzala a jak funguje.

1.1 Historie Ethernetu

Historie typu sítí se sdíleným médiem, mezi které se řadí i Ethernet začíná v 60. letech na Havaji, kde byla vyvinuta radiová síť ALOHA, která sloužila k propojení Havajských ostrovů. Jednalo se o síť s náhodným přístupem, což znamenalo, že vysílací stanici není přiřazen časový interval vysílání, ale rámec je poslán ve chvíli, kdy se to vysílací stanici hodí.

Z tohoto principu je jasné, že na přenosovém médiu vznikají kolize. Je tedy jasné, že musí existovat kontrolní mechanismus, který když zjistí, že odvyšlání rámec nebyl přijat, nebo byl přijat s chybou, dojde k opětovnému odvyšlání daného rámce. K tomuto opakování vysílání dochází v průměru třikrát při maximální použité zátěži. Využití kanálu je tedy asi 15% [1].

Pro zlepšení těchto vlastností byla vyvinuta tzv. řízená ALOHA. Ta pracuje tak, že interval opakovaného vysílání je volen náhodně. Tuto vlastnost přebírá i technologie sítí Ethernet.

První síť využívající technologii Ethernet vznikla v laboratořích PARC¹ firmy Xerox z důvodů propojení tamních počítačů v polovině 70. let. Autory této „pilotní“ verze Ethernetu byly Bob Metcalfe a David Boggs a dosahovala rychlosti necelých 3 Mb/s. Firma Xerox nabídla v roce 1979 spolupráci společností DEC² a Intel a v roce 1980 byl Ethernet zrychlen na 10 Mb/s. Tato verze Ethernetu nesla označení DIX, dle názvů firem, které se podíleli na jejím vývoji.

Tyto společnosti se rozhodli nenechat si Ethernet jako proprietární řešení, ale zveřejnili jeho specifikace a nechali Ethernet standardizovat. Jakou standardizační autorita byla zvolena společnost IEEE³, která předala projekt Ethernet do rukou pracovní skupiny 802.3. První standard technologie Ethernet zpracovaný IEEE byl publikován roku 1985 a nesl název „*IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification*“ [3].

Vývoj šel dál, a v roce 1993 byla publikována technologie přepínání, což mělo za následek snížení počtu kolizí a zmenšení kolizních domén, tudíž i lepší využití šířky pásma přenosového media. V této době se začínal prosazovat i plně duplexní přenos a byl odstartován vývoj virtuálních sítí VLAN. Z důvodu obrovského nárůstu počtu uživatelů celosvětové sítě Internet, přestala být rychlost 10Mb/s dostatečnou a to i s využitím výše uvedených technologických vylepšení.

¹ Palo Alto Research Center

² Digital Equipment Corporation

³ Institute of Electrical and Electronic Engineers

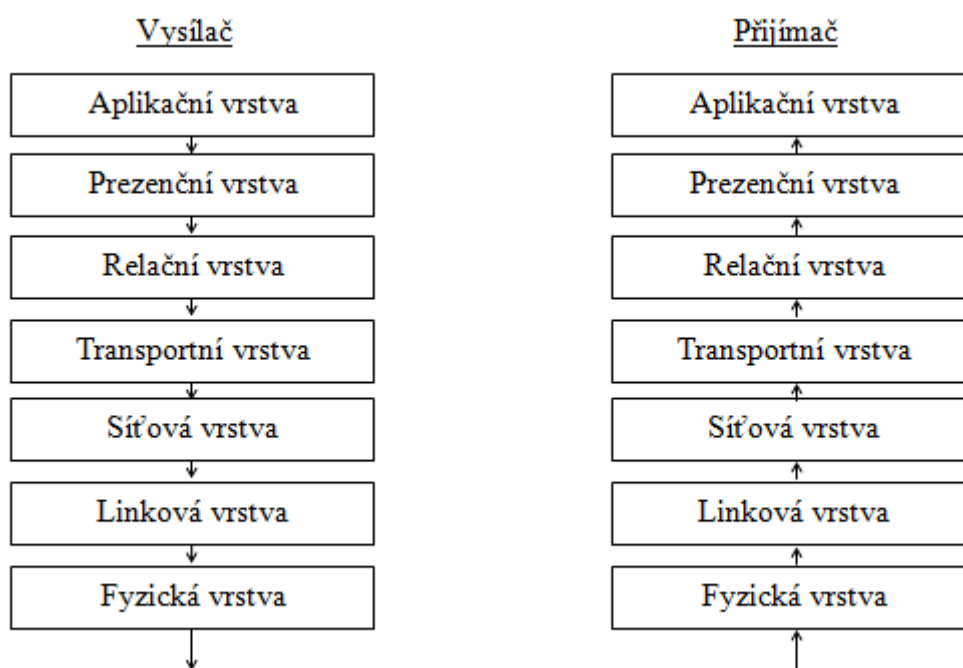
Organizace IEEE, pod kterou mezitím vývoj Ethernetu plně přešel, proto vyvíjela a v roce 1995 i standardizovala FastEthernet pod označením 802.3u. Tento standard obsahoval mimo jiné i navýšení celkové rychlosti na 100Mb/s, tedy na celý desetinásobek původního přenosového pásma. Ale z důvodu používání metody CSMA/CD se tím snížila maximální vzdálenost mezi uzly asi na desetinu (u metalického vedení asi na 100m).

V roce 1999 pak dochází k dalšímu zdesetinásobení rychlosti a společnost IEEE ratifikuje standard 802.3 na 802.3ab, který se zabývá připojením GigabitEthernetu pomocí UTP kabelu Cat 5, nebo 6. V roce 2002 pak došlo k další ratifikaci (802.3ae-2002) a k dalšímu navýšení rychlosti, tentokrát na 10Gb/s. Až do současnosti pokračuje rozšiřování této verze Ethernetu na různá přenosová média.

Další navýšení rychlosti nastalo v roce 2010 nejprve na 40Gb/s (802.3ba-2010) a v roce 2011 na 100Gb/s (802.3bg). Pro tyto rychlosti je primárně určené jako medium optické vlákno.

1.2 ISO/OSI model

Referenční model ISO/OSI je sedmivrstvý model vypracovaný organizací ISO v roce 1984. Model slouží ke znázornění zpracování dat v rámci telekomunikační sítě.



Obr. 1.1 Referenční model ISO/OSI

Tento model objasňuje proces komunikace a vše co musí data podstoupit při přenosu z jedné strany na druhou. V následujících kapitolách si stručně představíme význam jednotlivých vrstev.

1.2.1 Aplikační vrstva

Tato vrstva má za úkol komunikovat přímo s uživatelem. Přijímá od něj požadavky a ty poté přenáší k prezenční vrstvě. Typickým zástupcem aplikační vrstvy je webový prohlížeč, nebo klienti elektronické služby.

1.2.2 Prezenční vrstva

Prezenční vrstva slouží ke správné interpretaci dat. Nezapomeňme, že síťová zařízení jsou elektronická, a tudíž „rozumí“ pouze formátu logických jedniček a nul, zatímco pro uživatele je tento formát nečitelný. Je tedy nutné převádět strojový text do textu čitelného člověkem. Dalším důležitou funkcí prezenční vrstvy je komprimace dat v případě pomalého připojení k síti.

1.2.3 Relační vrstva

Relační vrstva RM IOS/OSI modelu zodpovídá za sestavování, spravování a rušení logických spojení (relací) mezi dvěma koncovými aplikacemi[2]. V rámci jedné relace pak může být navázáno několik spojení. Připojíme-li se například k webovému serveru, relační vrstva nejprve naváže spojení pro přenos html⁴ souboru a teprve poté (už v rámci jiného spojení) začne třeba se stahováním obrázků, které jsou na webové stránce zobrazeny.

1.2.4 Transportní vrstva

Zatímco relační vrstva se stará o vytváření relací a předávání dat uživateli, transportní vrstva sama o sobě řeší to, jak jsou jednotlivá spojení v rámci relace navázána a jak probíhá přenos dat. Neřeší pak, kudy jsou data posílána ani to, jakou rychlostí k přijímací straně dojdou. Účelem transportní vrstvy je tudíž zajistit přenos dat a jejich rozdělení pro jednotlivé aplikace v rámci koncového zařízení. Třídění datových bloků (u transportní vrstvy nazývaných segmenty) je zajištěno pomocí portů. Každé spojení dostane tzv. číslo portu, a transportní vrstva si pamatuje, který proces o dané spojení požádal. Při přijetí segmentu pak podle čísla portu pošle data správnému procesu.

1.2.5 Síťová vrstva

Tato vrstva je zodpovědná za doručení dat ke správnému příjemci. Síťová vrstva má sice celkem přesnou představu o topologii sítě jako celku, ale na druhou stranu se již nezajímá o jednotlivé dílčí sekce sítě. Zařízení síťové vrstvy ví, v jaké části sítě se adresát nachází a kudy se do dané části dostat, ale detaily doručení přenechává vrstvě linkové. Dnes se již takřka výhradně pro adresaci koncových zařízení používá IP adres a datové bloky přenášené na síťové vrstvě se nazývají pakety.

1.2.6 Linková vrstva

Úkolem linkové vrstvy⁵ je především zajistit přístup k mediu a bezchybný přenos datových bloků, které se na linkové vrstvě nazývají rámce. Přístupem k mediu se rozumí zajištění vysílání dat tak, aby nedošlo ke kolizi se zprávami určenými pro jiná koncová zařízení a jejich rozlišení příjemců a vysílačů. Podrobnější popis funkcí linkové vrstvy a její vztah k Ethernetu bude popsán v kapitole 2.

1.2.7 Fyzická vrstva

Fyzická vrstva přiřazuje konkrétní fyzický signál jednotlivým bitům vysílaným na přenosové medium^[2]. Fyzická vrstva nezná pojem začátku nebo konce datového bloku.

⁴ HyperText Markup Language

⁵ Někdy též značena jako spojivá vrstva

U fyzické vrstvy se vždy jedná o časově nekonečný⁶ sled logických jedniček a nul, které tato vrstva konvertuje do správné posloupnosti bitů a přeposílá vyšší vrstvě v případě přijímače. V případě vysílače proces probíhá opačně.

⁶ Nekonečný z pohledu zařízení, které nezná pojem před zapnutím a po vypnutí.

2 Technologie Ethernet (IEEE 802.3)

Jak již bylo zmíněno v úvodu této práce, technologie ethernet pracuje na druhé vrstvě referenčního modelu ISO/OSI. Také bylo zmíněno, že tato vrstva zprostředkovává přístup k mediu a zajišťuje bezchybné (resp. s ověřením, že k chybě došlo) přenášení rámců. V této kapitole se zaměříme na to, jak tyto úlohy plní právě technologie ethernet.

Pro zajištění předchozích dvou požadavků na linkovou vrstvu se technologie ethernet dělí na dvě podvrstvy: LLC⁷ a MAC⁸.

2.1 LLC

Tato podvrstva technologie ethernet zajišťuje kompatibilitu s použitým protokolem na třetí vrstvě referenčního modelu ISO/OSI. Jinými slovy, podvrstva LLC zajišťuje, že bude-li na třetí vrstvy jakýkoli protokol, druhá vrstva bude schopna přijat formát paketů tohoto protokolu, správně tento paket zapouzdřit do ethernetového rámce a vyslat na přenosové médium.

2.2 MAC

Podvrstva MAC zajišťuje především správný přístup k přenosovému mediu a označení vysílače a přijímače pro jednoznačné určení odkud a komu je rámec určen. Dále pak zajišťuje integritu dat a rozpoznání, pokud by došlo ke změně dat při přenosu. Pro zajištění správného přístupu k mediu pak podvrstva MAC využívá mechanismu CSMA/CD⁹.

2.2.1 CSMA/CD

Přístup k mediu v rámci této metody je v podstatě velmi jednoduchý. Znázornění principu CSMA/CD je zobrazeno na stavovém diagramu na Obr. 2.1.

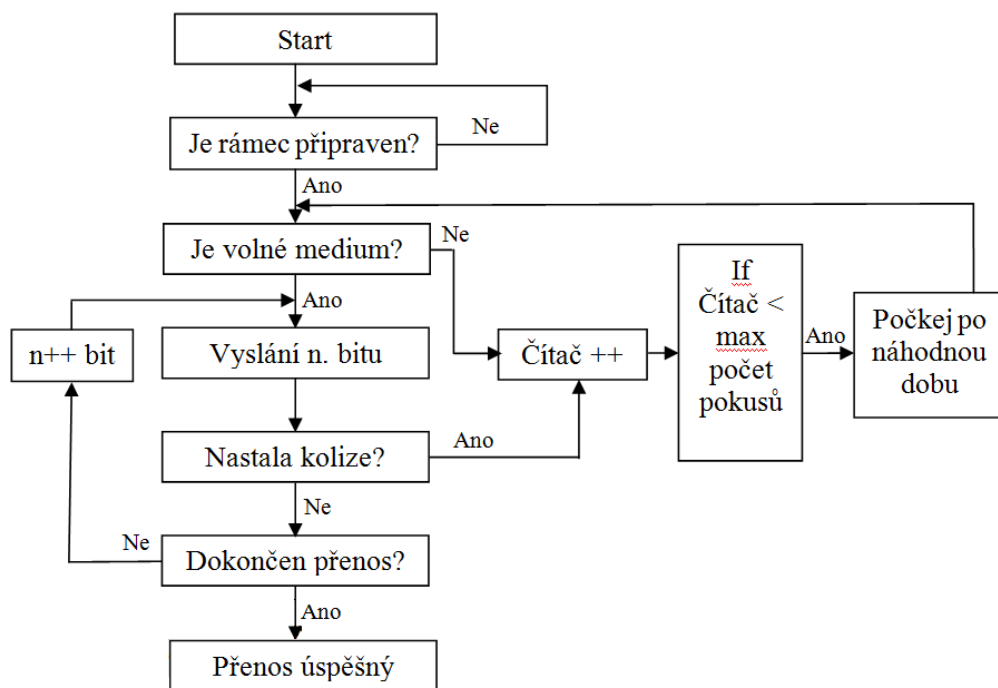
CSMA/CD se nejprve zeptá, zda je rámec připraven k odeslání. Pokud ano, pak je prověřeno, zda už vysílá někdo jiný, nebo zda je médium volné. Je-li přenosový kanál volný, pak je prověřena detekce kolize a nakonec se CSMA/CD dotazuje, zda byl přenos úspěšně dokončen.

Při detekování kolize, nebo ve chvíli, kdy vysílá jiná stanice, je inkrementován čítač, který počítá množství pokusů k odeslání. Po otestování, zda počet pokusů nedosáhl maximálního množství je pak odeslání zdrženo o určitou dobu. Tato doba je volena náhodně. Vyplývá to z toho, že všechny stanice na sdíleném mediu detekují kolizi ve stejnou chvíli, a kdyby byla chvíle odeslání přesně definována a nastala u všech stanic ve shodně, pak by došlo k další kolizi.

⁷ Logical Link Control

⁸ Media Access Control

⁹ Carrier Sense Multiple Access with Collision Detection



Obr. 2.1 Stavový diagram CSMA/CD

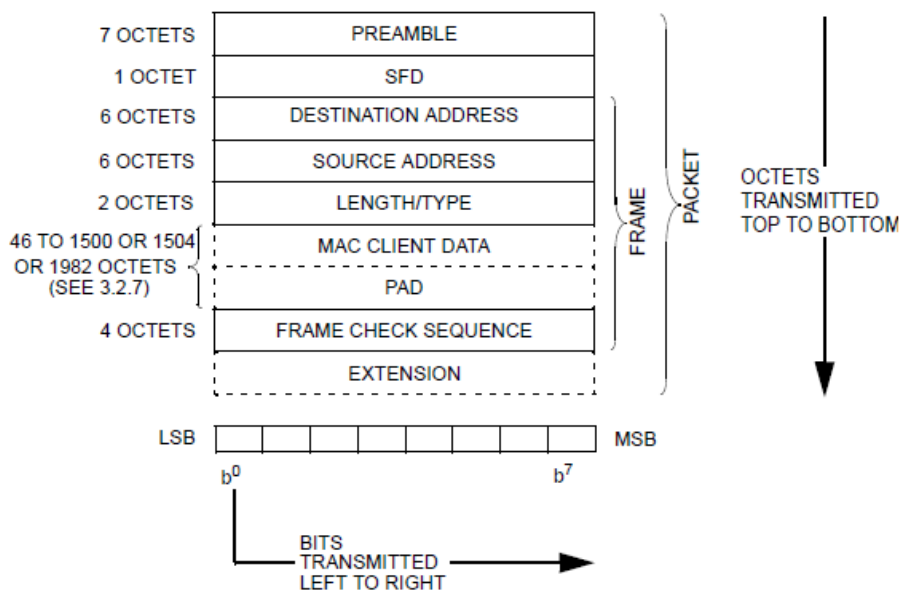
2.3 Typy ethernetových rámců

Standard technologie ethernet definuje tři druhy rámců: Základní rámeček, envelope rámeček a Q-rámeček. Základní rámeček a envelope rámeček používají stejný formát, rozdíl je pouze v maximální velikosti přenášených dat. Zatímco základní rámeček má maximální velikost klientských dat 1500 bytů, u envelope rámečku to může být až 1982 bytů. Envelope rámeček úmyslně umožňuje zahrnout dodatečné prefixy a sufixy požadované zapouzdřovacími protokoly vyšší vrstvy[3].

Q-rámeček má potom od dvou předchozích jinou maximální velikost přenášených dat a jiný formát. Maximální velikost přenášených dat je u tohoto typu rámečku 1504 bytů. Ohledně formátu má tento rámeček mezi pole zdrojová adresa a pole velikost/typ vloženu čtyř bytovou 802.1Q hlavičku, která umožňuje identifikovat až 4096 možných virtuálních LAN.

2.4 Formát ethernetového rámečku

Vzhledem k malým rozdílům ve formátech rámců výše uvedených typů, omezím se pouze na popis základního typu ethernetového rámečku. Ethernetový rámeček se skládá ze sedmi bloků. Prvním blokem je preamble, za níž následuje start frame delimiter, cílová adresa, zdrojová adresa, typ a délka dat, samotná uživatelská data a nakonec frame check sequence. Volitelnou položkou je pole extensit, které je přidáno pouze v případě, že se jedná o poloduplexní komunikaci 1000Mb/s. Velikost jednotlivých bloků, a to jak jsou přenášený, popisuje Obr. 2.2.



Obr. 2.2 Struktura Ethernetového rámce¹⁰

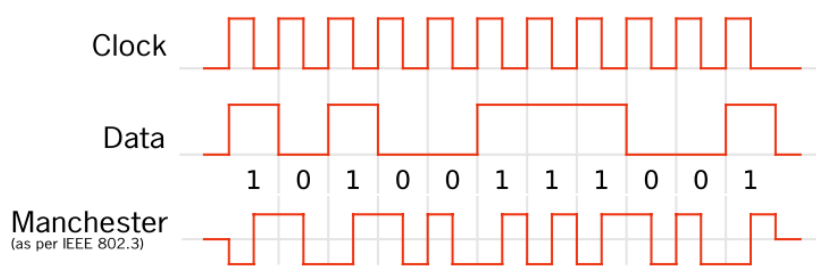
Je třeba dodat, že jeden oktet je 8 bitů, tudíž oktet je stejný termín jako byte. Termín byte budu používat ve zbytku práce. Všimněme si pořadí posílání jednotlivých bitů, které je voleno od nejnižšího bitu k nejvyššímu bitu.

2.4.1 Preambule

Preambule je sedmi bytové pole, které umožňuje časovým obvodům správnou synchronizaci s přicházejícím paketem. Jedná se o takovou sekvenci jedniček a nul:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

Pro přenos na mediu (při rychlostech 10Mb/s) je používán linkový kód Manchester, jehož průběh je v závislosti na hodinovém signálu a datech zobrazen na obrázku Obr. 2.3



Obr. 2.3 Průběh linkového kódu Manchester¹¹

Pro standard IEEE 802.3 je generuje linkový kód Manchester logická operace XOR mezi hodinovým signálem a původními daty. Tento způsob generování linkového kódu je výhodný, a to z důvodu realizace pouze za pomoci hradel, což umožňuje vysokou symbolovou rychlost.

¹⁰ [3] – str. 53

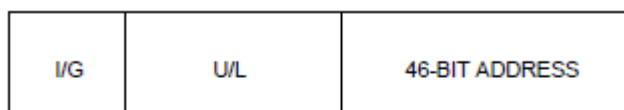
¹¹ Převzato z: http://en.wikipedia.org/wiki/File:Manchester_encoding_both_conventions.svg

2.4.2 Start Frame Delimiter

Pole Start Frame Delimiter (SFD) je takováto sekvence jedniček a nul: 10101011. V podstatě se jedná o osmí byte preamble, vyjma posledního bitu, který je jedničkou. Počítá se s tím, že při zahájení vysílání SFD je jsou již vysílač a přijímač synchronizováni a je možné začít vysílat. Ukončení synchronizační části nám indikují dva bity log. „1“ za sebou.

2.4.3 Pole Adres

Pole adres zahrnuje cílovou i zdrojovou adresu ethernetového rámce a každá z obou adres je 48 bitů velká. Formát adresy je zobrazen na Obr. 2.4.



I/G = 0 INDIVIDUAL ADDRESS
I/G = 1 GROUP ADDRESS
U/L = 0 GLOBALLY ADMINISTERED ADDRESS
U/L = 1 LOCALLY ADMINISTERED ADDRESS

Obr. 2.4 Formát adresy¹²

První bit, nám určuje, zda se vysílání bude unicast, nebo multicast. V případě, že první bit je „1“, pak se jedná o multicastové vysílání, a v opačném případě o unicastové vysílání. Daná skupina při multicastovém vysílání se nadále určuje vyšší vrstvou ISO/OSI modelu. Technologie ethernetu zná i pojem broadcastu, jehož adresa je FF-FF-FF-FF-FF.

Globální a lokální dělení adresy je historický přežitek z doby, kdy si administrátoři sítě mohli přiřazovat adresy svévolně. V tomto případě bylo nutno zajistit, aby při přenosu nedošlo ke kolizi s jinou stanicí. Dnes jsou veškeré adresy přidělovány globálně a jsou „neměnné“¹³.

Pole cílové adresy určuje adresu cílové stanice v rámci sítě. Zda se jedná o unicastové, multicastovou nebo broadcastovou adresu bylo popsáno výše na této straně. Zdrojová adresa pak určuje zdrojovou stanici, která posílá MAC rámec.

Ačkoli adresa není hierarchická, tudíž se podle ní nedá určit kde která stanice v síti fyzicky leží, je možné ji rozdělit po třech bytech na dvě stejné části. První část adresy určuje výrobce zařízení¹⁴, druhá je poté zcela v kompetenci tohoto výrobce²¹.

2.4.4 Pole délka/typ

Pole délka/typ je dvou bytové a v závislosti na číselné hodnotě má dva významy:

- Hodnota pole je menší nebo rovna 1500 (05DC hexadecimálně), pak se jedná o základní typ rámce.

¹² [3] – str.55

¹³ Měnit ji lze, ale nedoporučuje se.

¹⁴Viz <http://standards.ieee.org/develop/regauth/oui/oui.txt>

- b) Hodnota pole je větší nebo rovna 1536 (0600 hexadecimálně), pak se pole délka/typ nazývá polem druh interpretace klientského protokolu MAC. Pole délka/typ a Typ interpretace se nemohou v rámci vyskytovat najednou^[3].

Pokud je velikost přenášených dat menší, než je minimální velikost (46 bytů), pak jsou do pole Pad doplněna data do minimální velikosti.

2.4.5 Pole klientských dat

Toto pole obsahuje uživatelská data, která jsou určena k přenesení po mediu. Velikosti tohoto pole byly pospány v kapitole 2.3. Pokud by objem přenášených dat měl přesáhnout tyto hodnoty, dojde na MAC podvrstvě k fragmentaci dat, což znamená, že data přesahující tuto hodnotu budou odříznuta a přenesena v jiném rámci.

2.4.6 Pole Pad

Jak již bylo popsáno výše, pole Pad slouží k dosažení minimální velikosti přenášených dat. Protože FCS pole (o kterém bude zmíněno v podkapitole 2.4.7) zahrnuje do svých výpočtů pole adres a klientských dat, tak výpočet Pad pole probíhá podle následující rovnice:

$$PadSize = [0, minFrameSize - (clientsDataSize + 2AddressSize + 48)] \quad [bit] \quad (1)$$

To že obsah Pad může být rovno 0, je jasné, ve chvíli kdy uživatelská data budou dostatečně velká (druhá část rovnice je záporná). Ohledně druhé části, je jasné, že minimální hodnota rámce (počítáno od pole adres po FCS) musí být rovna rozdílu minimální velikosti rámce a dat v rámci obsažených.

Budeme-li uvažovat, že hodnota *clientDataSize* je rovna 0, pak do minimální hodnoty velikosti rámce se započítají hodnoty velikosti zbylých polí ethernetového rámce. Odtud pak parametr *AddressSize* velikosti adres (zdrojová a cílová), a 48 bitů (6 bytů) za pole délky/typ (2 byty) a pole FCS (4 byty).

2.4.7 Pole Frame Check Sequence (FCS)

Posledním polem Ethernetového rámce je pole FCS. Toto pole je 4 bytové a hlídá integritu přenášených dat. Výpočet zahrnuje pouze cílovou a zdrojovou adresu, pole délka/typ a klientská data (včetně Pad pole). FCS pro kontrolování, zda došlo k chybě při přenosu, využívá 32 bitový cyklický redundantní kód (CRC). Tento kód umí pouze odhalovat chyby, ale neumí je opravovat.

Algoritmus probíhá tak, že vysílací stanice při odesílání dat vypočítá hodnotu CRC (jak je počítána, o tom budu mluvit níže) a tu odešle za daty v poli FCS. Stanice, která přijímá ethernetový rámec, si „odstříhne“ datovou část od FCS, z datové části vypočítá hodnotu CRC a tu poté porovná s hodnotou přijatou. V případě že dojde ke shodě obou hodnot CRC, rámec je prohlášen za úspěšně přijatý. Jsou-li hodnoty přijatého a vypočteného CRC různé, pak je rámec zahozen a je vyžádán znovu.

2.4.7.1 *Výpočet CRC*

Jak již bylo řečeno, technologie ethernet používá 32-bitový CRC (CRC-32). Hodnota CRC je obecně počítána dle rovnice (2).

$$\frac{D(x) \cdot x^{31}}{G(x)} = R(x) + r(x) \quad (2)$$

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (3)$$

Kde $D(x)$ je polynom dat o stupni $x^{(n-1)}$, kde n je velikost dat v bitech, $G(x)$ je tzv. generující polynom CRC, v případě technologie ethernet se jedná polynom 32. Stupně, jehož přesná hodnota je vyjádřena rovnicí (3). $R(x)$ je poté výsledek dělení polynomů a $r(x)$ je zbytek po daném dělení, což je hodnota CRC, která je přenášena. Polynom $D(x)$ je násoben hodnotou x^{31} , z důvodu povýšení tohoto polynomu na minimální hodnotu 32-bitů, aby výsledkem dělení byl polynom stupně ≤ 31 .

Protože matematický výpočet je časově náročný, používá se pro kalkulaci hodnoty CRC kombinace logické operace XOR a posuvných registrů. Pro hardwarovou implementaci se používá tzv. Galoisův LFSR¹⁵, který generuje právě zbytek po dělení dvou polynomů.

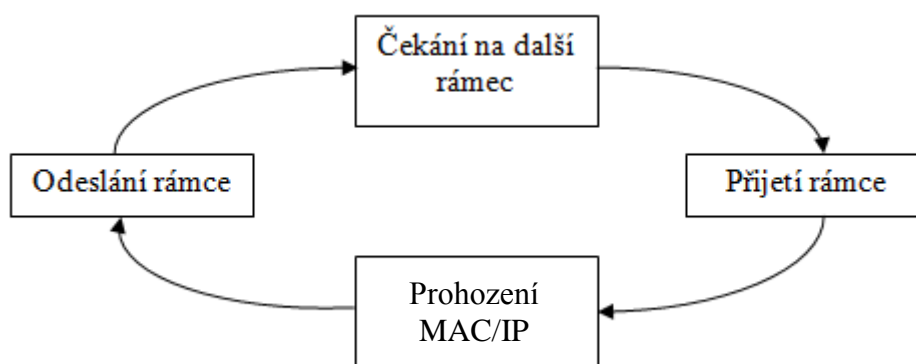
¹⁵ LFSR – Lineární Zpětnovazební Posuvný Registr

3 Teoretický rozbor

Na následujících stranách se budu zabývat základními vlastnostmi, které by měl Ethernet/IP reflektor mít a co k samotné realizaci bude třeba.

3.1 Základní vlastnosti

Základním principem a funkcí Ethernet/IP reflektoru je odesílání ethernetového rámce, příp. IP paketu zpět ke stanici, která tento rámec vyslala. Jedná se tedy o smyčku, která pouhým prohozením IP adres odešle nezměněnou zprávu zpět k odesílateli. Stavový diagram Ethernet reflektoru bude vypadat takto:



Obr. 3.1 Stavový diagram Ethernet reflektoru

Proces lze tedy rozdělit do čtyř částí: Čekání na rámec, Přijetí rámce, Prohození MAC/IP adres a odeslání rámce. Čekání na rámec je funkcí danou jednoznačně standardem IEEE 802.3 a metodou vysílání a přijímání rámců CSMA/CD. O přijetí a odeslání rámce se bude starat jeden komunikační mikrokontrolér. Blok prohození MAC adres může být v zásadě realizován dvěma způsoby:

- a) pomocí hradel a posuvných registrů,
- b) softwarové zpracování v mikrokontroléru.

Nyní si rozebereme tyto dvě varianty co do rychlosti zpracování dat, časové a finanční náročnosti a obtížnosti řešení.

3.1.1 Realizace pomocí hradel a posuvných registrů

Nejprve se budeme zabývat vlastnostmi této varianty co do rychlosti zpracování dat. Celková doba zpoždění při zpracování dat je zde dána pouze zpožděním na jednotlivých hradlech a v posuvných registrech. Celková doba zpoždění se tedy vypočítá podle následujícího vztahu:

$$t_z = Nt_h + Mt_{pr} + t_{km} \quad (4)$$

Kde t_h je zpoždění způsobené hradly (řádově v ps), N je počet hradel, t_{pr} je zpoždění způsobené posuvným registrem (řádově v ns), M je počet posuvných registrů a t_{km} je zpoždění způsobené komunikačním mikrokontrolérem, které závisí na jeho taktovací frekvenci (od desítek MHz, po GHz).

U této varianty by tedy pravděpodobně nebyl problém, aby vyrobený reflektor mohl pracovat se standardem 100BASE-T.

Dalším kritériem byla časová a finanční náročnost daného řešení. V případě realizace pomocí hradel a posuvných registrů není ani tak problém se samotnými registry a hradly, jako s komunikačním mikrokontrolérem, který je schopen komunikovat s hradlovou vyhodnocovací jednotkou. Takovéto mikrokontroléry se cenově pohybují okolo 500 Kč plus dopravné. V případě jednoho takového mikrokontroléru bychom se tedy dostali na hodnotu okolo 800 Kč, za jeden takový kontrolér s dobou dodání okolo jednoho měsíce.

Posledním kritériem byla celková náročnost realizace. Varianta hradel a registrů je velmi náročná ani ne tak na správné vyčítání a zapisování registrů, ale na samotnou komunikaci na fyzické a spojové vrstvě. Tato varianta závisí na výstupu z komunikačního mikrokontroléru, zda by se jednalo o čistý ethernetový výstup tzn., že výstupem by byl nepozměněný ethernetový rámec, nebo zda by výstupem byla data v konkrétní podobě specifikované výrobcem.

Varianta realizace pomocí hradel a posuvných registrů by tedy byla relativně obtížná na realizaci, finančně by celková hodnota výrobku vycházela okolo 2500 Kč, a byla by použitelná pro standard ethernetu 100BASE-T.

3.1.2 Softwarové zpracování dat v mikrokontroléru

U této varianty budou hlavními použitými zařízeními komunikační mikrokontrolér a vyhodnocovací mikročip. Tato dvě zařízení ovšem musí mezi sebou komunikovat, tudíž je zřejmá nutnost komunikační sběrnice. Dále pak celková doba zpoždění bude záviset na množství zpracovávaných dat mikročipem. Doba zpoždění se tedy vypočítá dle (5).

$$t_z = t_{km} + t_{ks} + N \cdot t_m \quad (5)$$

Kde t_{km} je zpoždění taktovací frekvencí komunikačního mikrokontroléru (stovky ps až desítky ns), t_{ks} je zpoždění komunikační sběrnice (desítky ns), t_m je zpoždění způsobené zpracováváním dat v mikrokontroléru (stovky ps až desítky ns), a N je počet jednocyklových instrukcí potřebných ke zpracování dat. U této varianty je nejpomalejším prvkem komunikační sběrnice, tudíž se dá očekávat, že tato varianta bude podporovat standard 10BASE-T.

Co se týče finanční náročnosti této varianty, cena mikrokontroléru zde bude nižší, (okolo 300 Kč i s dopravou). Cena vyhodnocovacího mikročipu závisí na použitém mikropočítači. Pro naši potřebu by měl stačit mikročip okolo 100 Kč. Celková cena této varianty by neměla přesáhnout 1500 Kč.

Posledním kritériem je obtížnost realizace. Softwarové řešení Ethernetového zrcadla se zdá být lehčím než řešení prostřednictvím hradel a posuvných registrů. Obrovské ulehčení je v samotném konceptu tohoto řešení. Jednalo by se o přijetí ethernetového rámce komunikačním mikrokontrolérem, jeho odeslání vyhodnocovacímu mikropočítači, kde by došlo k softwarovému zpracování dat. Následně by pak byl rámec odeslán zpět komunikačnímu mikrokontroléru.

Varianta softwarového zpracování dat v mikrokontroléru je tedy jednodušejší realizovatelná a finančně méně náročná. Její nevýhodou je použití pravděpodobně na linkách 10 Mb/s.

3.2 Napájení

Dalším tématem k zamyšlení bylo vyřešit napájení celého zařízení. I v případě napájení je několik kritérií, dle kterých by mělo být rozhodováno:

- a) náročnost,
- b) cena,
- c) pohodlnost.

Jako možnosti, mezi kterými by mělo být rozhodováno, bylo navrženo napájení externím zdrojem (ze sítě 230V), USB konektorem z PC a pomocí technologie Power over Ethernet (IEEE 802.3at).

3.2.1 Napájení technologií Power over Ethernet

Princip technologie Power over Ethernet spočívá ve využití nepoužívaných vodičů UTP kabelu k napájení vzdáleného přístroje. Tabulka 1 zobrazuje maximální použitelná napětí a proudy v jednotlivých třídách.

Tabulka 1: Maximální přípustné proudy a napětí technologie PoE¹⁶

Parametr	Napětí [V]	Minimum [mA]	Maximum [mA]
Proud třídy 0	14.5 – 20.5	0	4
Proud třídy 1	14.5 – 20.5	9	12
Proud třídy 2	14.5 – 20.5	17	20
Proud třídy 3	14.5 – 20.5	26	30
Proud třídy 4	14.5 – 20.5	36	44

Z této tabulky je zřejmé, že se technologie PoE hodí k napájení obvodů s vysokou vstupní impedancí, jelikož počítá s vysokým budicím napětím a nízkými přenášenými proudy.

Z hlediska náročnosti realizace je v poměru ke zbylým možnostem technologie Power over Ethernet jednoznačně nejtěžší. Samotnému přivedení napětí předchází vyjednávání o tom, která z uvedených tříd bude pro napájení zařízení použita. Toto vyjednávání se odehrává na úrovni spojové vrstvy, ale vyjednávání přebírá zařízení ještě před PHY rozhraním komunikačního mikrokontroléru.

Z předchozí věty vyplývá hned i cena, respektive její změna při použití této technologie. PoE kontroléry stojí okolo 100Kč (s dopravou 300Kč), takže cena zařízení by se navýšila o tuto hodnotu.

Ohledně pohodlnosti napájení je jasné, že tato varianta bude jednoznačně pro případné zákazníky nejpohodlnější. Nemusí mít sebou externí zdroj, ani PC s USB konektorem.

¹⁶ [3] section 2, str. 660, table 33–16

3.2.2 Napájení pomocí USB konektoru

Tato možnost připadá v úvahu, jelikož se počítá s použitím zařízení v telekomunikační technice. Napájení pomocí USB konektoru má výhodu například v proudové ochraně a přesně definovaném napětí. Jedná se napětí 5V a maximální možný odebíraný proud na jeden port je 500mA.

Z hlediska ceny by se jednalo o nejlevnější variantu (v případě, že obsluha bude s sebou mít jakékoli zařízení s USB konektorem a to zařízení bude zapnuto).

Také jednoduchostí předčí ostatní varianty, jelikož není třeba dalších kontrolérů ani jiných elektronických součástek.

Z hlediska pohodlnosti jsme již vedeni k hlubšímu zamyšlení. Je otázkou kolik lidí s sebou nosí notebook (příp. netbook), které bude schopno poskytnout dostatečný výkon našemu zařízení. Nehledě na to, že by toto zařízení muselo být k našemu výrobku trvale připojeno.

3.2.3 Napájení pomocí externího zdroje

Princip napájení Ethernetového reflektoru pomocí externího zdroje je z principu nejjednodušší ze všech tří mnou uvažovaných variant. Externím zdrojem je myšleno připojení k elektrické síti (230V) prostřednictvím další „krabičky“.

Z hlediska pohodlnosti pro zákazníka je tedy tato varianta nejméně pohodlná, jelikož musí s sebou nosit další zařízení (jak si ovšem popíšeme níže, toto by teoreticky nebylo nutné) a mimo připojení k Ethernetové síti musí být k dispozici i volná elektrická zásuvka.

U externích napájecích zdrojů obecně, máme dvě možnosti řešení těchto zařízení. Jedná se o:

- a) Zdroj klasické koncepce
- b) Spínaný zdroj

Princip a vlastnosti možných typů externích zdrojů si popíšeme jenom zhruba z hlediska základních parametrů.

3.2.3.1 *Zdroj klasické koncepce*

Zdroje klasické koncepce se vyznačují svou jednoduchostí, nízkou výrobní cenou, vysokým poměrem hmotnost/výkon a s touto vlastností související i velkým objemem. V této větě byly shrnuty základní vlastnosti zdrojů klasické koncepce, které se nám projeví i v konečném řešení výrobku.

Při napájení zdrojem klasické koncepce by tedy určitě došlo k přidání dalšího zařízení. Jak jsem napsal výše, byla by zajímavá myšlenka mít výkonovou část jako přímou součást výrobku. Pokud tuto myšlenku budeme aplikovat na zdroj klasické koncepce, zjistíme, že tato varianta není v tomto provedení úplně ideální. Pokud začneme u vlastnosti poměr hmotnost/výkon, je jasné, že výrobek by nabyl na váze (i objemu). Pokud bychom chtěli zachovat velikost výrobku (tj. pohodlně přenositelný), pak by došlo k zahřátí výrobku a tím by mohlo dojít k ovlivnění některých součástek citlivějších na teplotu.

Zůstaneme-li ovšem u myšlenky zdroje jako externího zařízení, pak má tato varianta velké výhody především v jednoduchosti realizace a celkové ceně (která by se mohla pohybovat okolo 200Kč).

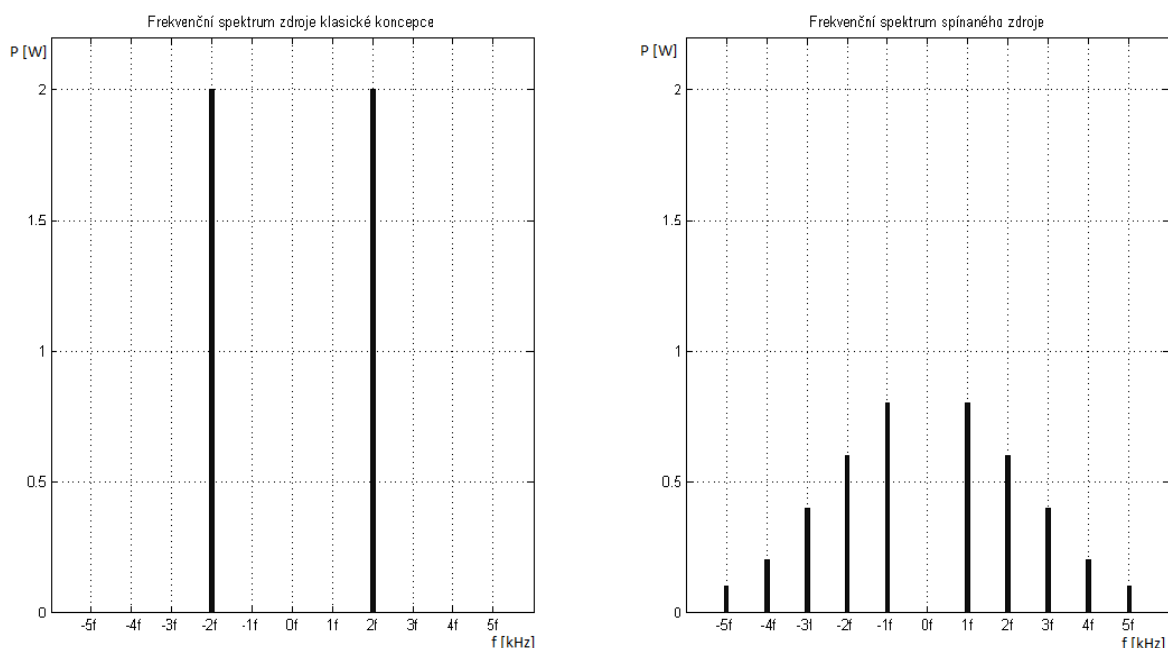
3.2.3.2 Spínaný zdroj

Ve zkratce můžeme napsat, že spínané zdroje jsou opakem zdrojů klasické koncepce. Jsou tedy složitější na realizaci, dražší, výkonnější s nižším stupněm přehřívání se, a tudíž mohou být menší.

I u této varianty napájecího zdroje je zajímavé zabývat se myšlenkou umístit tento zdroj přímo do výrobku. Jak bylo již napsáno výše, u této varianty by to teoreticky šlo. Spínaný zdroj může při stejné velikosti jako zdroj klasické koncepce, dodávat vyšší výkon. Neovlivnil by tedy tepelně ani velikostně zařízení natolik, aby to mělo vliv na mikročipy umístěné ve výrobku, či na manipulační vlastnosti zařízení.

Spínané zdroje pracují na jednoduchém principu přenosu výkonu přenosu středofrekvenčního širokopásmového přenosu. Vzhledem k tomu, že práce se přímo nezabývá problematikou zdrojů, popíšu základní princip spínaného zdroje velmi stručně.

Po přivedení střídavého napětí 230V/50Hz na vstup zdroje dojde k jeho usměrnění diodovým můstkem, následně dojde k jeho rozstředování a středofrekvenčním transformátorem pak dojde k jeho přenesení na nižší/vyšší napět'ové úrovně a opět k jeho usměrnění.



Obr. 3.2 Frekvenční spektra spínaného a klasického zdroje¹⁷

Nejdůležitější fází je zde právě ono rozstředování, kde dojde k předělání stejnosměrného signálu na signál obdélníkový s frekvencí první harmonické okolo desítek kHz. Graf na Obr. 3.2 nám zobrazuje základní rozdíly mezi spínaným a klasickým zdrojem z hlediska přenosu energie. Zatímco u klasického zdroje se energie přenáší na jednom

¹⁷ Grafy jsou pouze ilustrativní

kmitočtu, z čehož vyplývá namáhání jenom úzkého pásma a tím i extrémní zátěž, tak spínané zdroje využívají k přenosu stejného množství energie pásma mnohem širšího.

Z hlediska finanční náročnosti spínaný zdroj jednoznačně převyšuje zdroj klasický. Celková cena realizace spínaného zdroje by se totiž pohybovala okolo 700 Kč.

4 Realizace

V této kapitole se budu zabývat vybranými možnostmi představené v kapitole 3, zdůvodněním proč právě tyto možnosti byly vybrány a konkrétními prvky, pomocí nichž bude daná možnost realizována.

4.1 Vybraný způsob realizace a napájení

Po zvážení rozhodujících faktorů, které byly popsány a představeny v předchozí kapitole, byly vybrány tyto varianty:

- a) Softwarové řešení
- b) Napájení externím zdrojem klasické koncepce

Softwarové řešení bylo vybráno na úkor řešení pomocí hradel a posuvných registrů především na základě náročnosti realizace a finanční dostupnosti komunikačního mikrokontroléru. Jelikož mnou zjištěný hardware potřebný pro variantu posuvných registrů byl těžko k dostání za rozumnou cenu (do 700 Kč) a v rozumné kvantitě (protože plno mikrokontrolérů má minimální objednávkové množství >500, což je pro jediný výrobek nepřijatelné).

Z hlediska náročnosti realizace, se zdá být softwarová realizace mnohem jednodušší než realizace pomocí hradel a posuvných registrů. U softwarové realizace komunikační mikrokontroléry již obsahují registry a buffery, které například ethernetový rámec umí rozdělit na jednotlivé části a vyčíst pouze to, co je potřeba.

U napájení bylo rozhodování poněkud obtížnější i vzhledem k více možnostem a jejich relativní podobnosti. Napájení pomocí technologie PoE bylo zamítnuto, ačkoli se jedná o zajímavou možnost vyřešit pohodlí zákazníka, bylo zjištěno, že problematika technologie PoE byla příliš rozsáhlá na to, aby se dala v daném časovém horizontu realizovat.

Další možností byla možnost napájet přístroj pomocí USB, které poskytuje permanentně 5V, 500mA, ale tato možnost byla zamítnuta, ačkoli se jedná o nejjednodušší a nejlevnější variantu napájení, a to z důvodu pohodlnosti při přenosu zařízení, které by potřebovalo ke svému chodu USB konektoru např. na PC.

Poslední možností, která tedy zbývala, bylo napájení pomocí externího zdroje. Jak již bylo napsáno, spínaný zdroj má mnoho výhod, ale nakonec bylo rozhodnuto pro externí napájecí zdroj klasické koncepce a to z důvodu jednoduchosti realizace a ceny. Zdroj klasické koncepce jsme bez větších problémů schopni realizovat ve výrobních podmínkách sami a v tuto chvíli hraje klíčovou roli faktor jednoduchosti výroby.

Ethernet reflektor tedy bude realizován pomocí komunikačního mikrokontroléru, který bude zprostředkovávat komunikaci na spojové a fyzické vrstvě, řídicího mikrokontroléru, který bude softwarově řešit zadanou problematiku zpětné smyčky a napájecího adaptéru.

4.2 Komunikační mikrokontrolér ENC28J60

K zajištění komunikace na spojové a fyzické vrstvě ISO/OSI modelu byl vybrán komunikační mikrokontrolér ENC28J60 a to z důvodu jednoduché práce s tímto mikrokontrolérem a cenovou a kvantitativní dostupností.

4.2.1 Základní vlastnosti ENC28J60

Mikrokontrolér ENC28J60 je komunikační kontrolér kompatibilní s technologií Ethernet s integrovaným MAC a 10BASE-T PHY rozhraním. Podporuje jeden 10-BASET port s automatickou detekcí a korekcí polarity stejně tak jako polo-, či plně-duplexní komunikaci. Komunikace s řídicím mikropočítačem je pak řízena po SPI sběrnici s taktem do 20 MHz.

Tento mikrokontrolér má 8kB buffer typu SRAM pro přijímání a vysílání paketů, přičemž tato je konfigurovatelná. Buffer je typu FIFO, což znamená, že první co do bufferu přišlo, první i odejde. Výhodou je, že ukazatel na poslední položku v bufferu má automatickou inkrementaci a kontrolér je vybaven DMA pro rychlý přístup k datům v bufferu.

Mikrokontrolér dále podporuje Unicast, Multicastové a Broadcastové pakety. Dále pak má programovatelný paketový filtr, a wake-up a to s následujícími možnostmi filtrace:

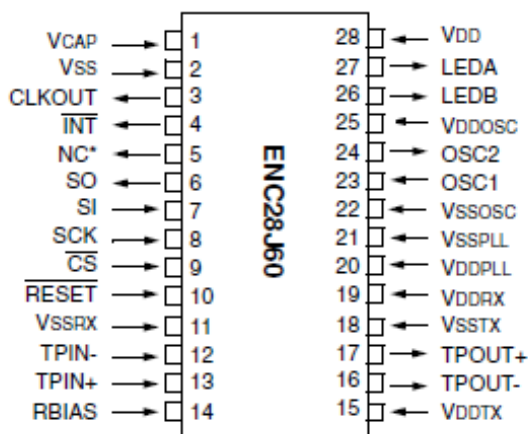
- Unicast cílová adresa,
- Multicast adresa,
- Broadcast adresa.

Ve výčtu možností jsou uvedeny pouze ty nejdůležitější pro naši práci, dále pak je zde možnost filtrovat na základě tabulky skupiny adres a spojení s uživatelem definovaným offsetem. Samotná filtrace je pak prováděna pomocí logického součtu a součinu.

Mikrokontrolér je dále vybaven výstupy pro programovatelné linkové TX a RX LED. Dále pak mikrokontrolér využívá šesti vstupních a jednoho výstupního pinu pro přerušení. Mikrokontrolér má taktovací frekvenci 25 MHz programovatelným výstupním hodinovým pinem. Napájecí napětí mikrokontroléru ENC28J60 je od 3,1 do 3,6V (typicky pak 3,3V). Na určitých pinech má mikrokontrolér toleranci do 5V. Odebíraný pracovní proud je potom typicky 160mA, maximálně do 180 mA¹⁸. Teplotní rozsah se pohybuje od -40°C do 85°C.¹⁹

¹⁸ [4] – str. 80

¹⁹ [4] – str. 1



Obr. 4.1 Rozložení pinů ENC28J60, pouzdro SOIC²⁰

4.2.2 Přehled pinů ENC28J60

Na Obr. 4.1 můžeme vidět rozložení pinů mikrokontroléru ENC28J60 pro námi použité pouzdro 28-pin SOIC. Piny mikrokontroléru můžeme rozdělit do čtyř kategorií:

1. Napájecí,
2. Řídící,
3. Komunikační,
4. Indikační.

4.2.2.1 Napájecí piny

Na mikrokontroléru ENC28J60 je celkem 11 napájecích pinů. Mimo hlavních napájecích pinů +3,3V a 0V jsou zde ještě další 4 referenční piny a stejné množství napájecích pinů pro komunikaci. Popis napájecích pinů je uveden v tabulce 2.

Tabulka 2: Napájecí piny mikrokontroléru ENC28J60²¹

Název pinu	Číslo pinu	Popis
V _{CAP}	1	2,5V výstup z vnitřního stabilizátoru
V _{SS}	2	Referenční hodnota 0V
V _{SSRX}	11	Referenční hodnota 0V pro PHY RX
V _{DDTX}	15	Napájení 3,3V pro PHY TX
V _{SSTX}	18	Referenční hodnota 0V pro PHY TX
V _{DDRX}	19	Napájení 3,3V pro PHY RX
V _{DDPL}	20	Napájení 3,3V pro PHY PLL
V _{SSPL}	21	Referenční hodnota pro PHY PLL
V _{SSOSC}	22	Referenční hodnota 0V pro oscilátor
V _{DDOSC}	25	Napájení 3,3V pro oscilátor
V _{DD}	28	Napájení mikrokontroléru 3,3V

²⁰ [4] – str. 1

²¹ [4] – str. 4

Funkčnost referenčních a napájecí pinů je celkem jasná, jediný zajímavý pin je zde V_{CAP} , který slouží jako ekvivalentní sériový rezistor, měl by k němu být připojen kapacitor o minimální velikosti $1\mu F$ (typicky pak $10\mu F$)²².

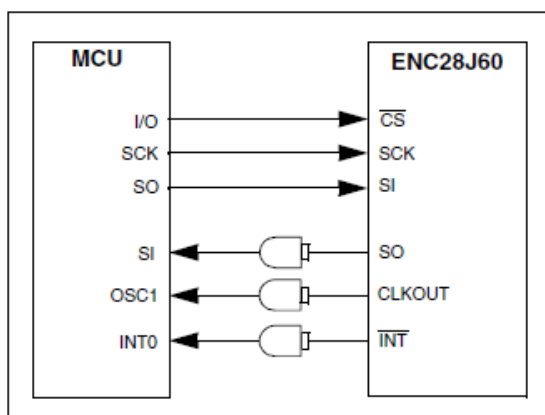
4.2.2.2 Řídicí piny

Jak již název této kategorie napovídá, jedná se o piny, které slouží k řízení kontroléru ENC28J60 a komunikaci mezi komunikačním mikrokontrolérem a řídicím mikrokontrolérem. V tabulce 3 Tabulka 3 jsou uvedeny řídicí piny mikrokontroléru ENC28J60 se stručným popisem jejich funkce.

Tabulka 3: Řídicí piny mikrokontroléru ENC28J60²³

Název Pinu	Číslo pinu	Popis
CLKOUT	3	Programovatelný hodinový výstup
\overline{INT}	4	Přerušení (výstupní pin)
SO	6	Výstupní pin toku dat sběrnice SPI
SI	7	Vstupní pin toku dat sběrnice SPI
SCK	8	Hodinový signál sběrnice SPI
\overline{CS}	9	Chip select
\overline{RESET}	10	Hardwarový reset
OSC1	23	Vstup krystalu
OSC2	24	Výstup krystalu

Jak již bylo napsáno, architektura kontroléru ENC28J60 počítá s napájením 3,3V. I přes tuto skutečnost však řídicí piny podporují 5V logiku na svém vstupu (tj. piny SI, SCK, \overline{CS} a \overline{RESET}). Výstupní pin (SO) je ale v logice 3,3V. Pro potřeby komunikace s řídicím kontrolérem, který vyžaduje logiku 5V, je pak možné použít doporučeného schéma na Obr. 4.2.



Obr. 4.2 Schéma připojení ENC28J60 na 5V logiku MCU²⁴

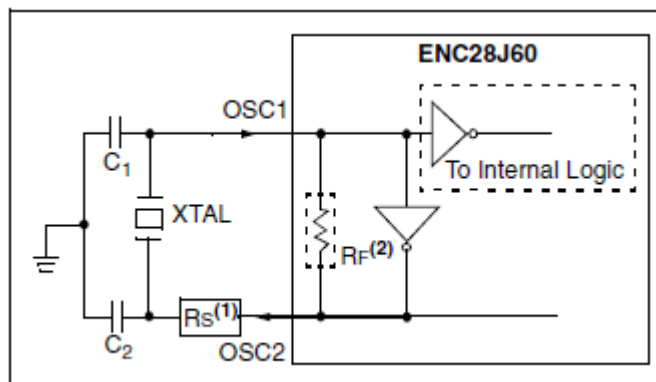
²² [4]– str. 4

²³ [4] – str. 4

²⁴ [4]– str. 8, figure 2-5

Mezi řídicí piny byly zařazeny i piny OSC1 a OSC2, které slouží k připojení krystalu určujícího pracovní kmitočet komunikačního mikrokontroléru. Krystal by měl být k mikrokontroléru připojen dle schéma na Obr. 4.3.

Na schéma je pak zobrazen i vnitřek mikrokontroléru včetně zpětnovazebního rezistoru R_F (typická hodnota od 2 do 10 M Ω) a volitelného rezistoru R_S , který by mohl být vyžadován pro krystal typu AT strip-cut. Velikost kondenzátorů C1 a C2 je ovlivněna kmitočtem použitého krystalu (typická kapacita u krystalů je 18 – 27pF).



Obr. 4.3 Schéma připojení krystalu k ENC28J60²⁵

4.2.2.3 Komunikační piny

Komunikační piny slouží k přenosu dat mezi fyzickými vrstvami ISO/OSI modelu. Jsou tedy na jedné straně připojeny k PHY rozhraní mikrokontroléru ENC28J60 a na druhé straně k přizpůsobovacím obvodům a konektoru RJ-45. Tyto piny jsou stručně popsány v tabulce 4.

Tabulka 4: Komunikační piny mikrokontroléru ENC28J60²⁶

Název pinu	Číslo pinu	Popis
TPIN-	12	Signálový vstup
TPIN+	13	Signálový vstup (stínění)
TPOUT-	16	Signálový výstup (stínění)
TPOUT+	17	Signálový výstup
RBIAS	14	Ochranný pin PHY rozhraní

Funkce komunikačních pinů je zřejmá, jedinou zvláštností je pin RBIAS. Tento pin slouží k ochraně PHY rozhraní mikrokontroléru. Podle katalogu má mít tento rezistor hodnotu 2,32 k Ω s přesností 1%. A má být na desce plošných spojů umístěn co nejbližší komunikačního mikrokontroléru a měl by být v provedení pro povrchovou montáž. Tento rezistor ovlivňuje amplitudu signálu na pinech TPOUT+/-²⁷.

²⁵ [4] – str. 5, figure 2-1

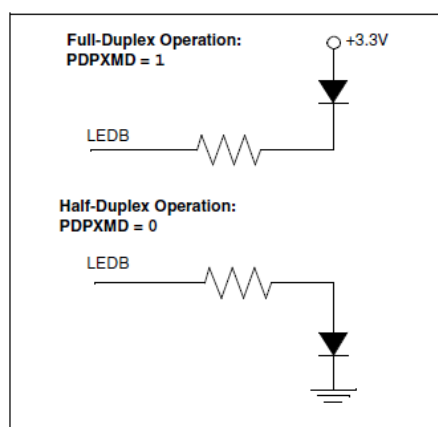
²⁶ [4] – str. 4

²⁷ [4] – str. 7

4.2.2.4 Indikační piny

Indikační piny jsou na mikrokontroléru pouze dva a to LEDA a LEDB. Oba slouží k indikaci komunikace (LEDA indikuje přijímání a LEDB indikuje vysílání). K těmto pinům jsou připojeny LED diody a proud, který teče piny LEDA a LEDB je omezen na 12mA.

Pin LEDB má ještě jednu unikátní vlastnost a to že pomocí něj je inicializován registr PHCON a bit PDPXMD. Jak vyplývá ze schéma zapojení pinu LEDB na Obr. 4.4, tak pokud proud do pinu LEDB vtéká, pak se bit PDPXMD nastaví na log. „1“ a kontrolér bude podporovat plně-duplexní komunikaci a pokud proud z pinu LEDB vytéká, pak bit PDPXMD je nastaven na log. „0“ a mikrokontrolér podporuje polo-duplexní komunikaci. V případě nepřipojení LED diody k pinu LEDB dojde k chybné inicializaci a bit PDPXMD bude nastaven na neurčitou hodnotu.



Obr. 4.4 Možné připojení LED diody k pinu LEDB

4.2.3 Organizace paměti ENC28J60

Mikrokontrolér ENC28J60 disponuje celkem 8352B SRAM paměti, která, jak je vidět z Obr. 4.5 Organizace paměti ENC28J60 se dá rozdělit do tří základních skupin:

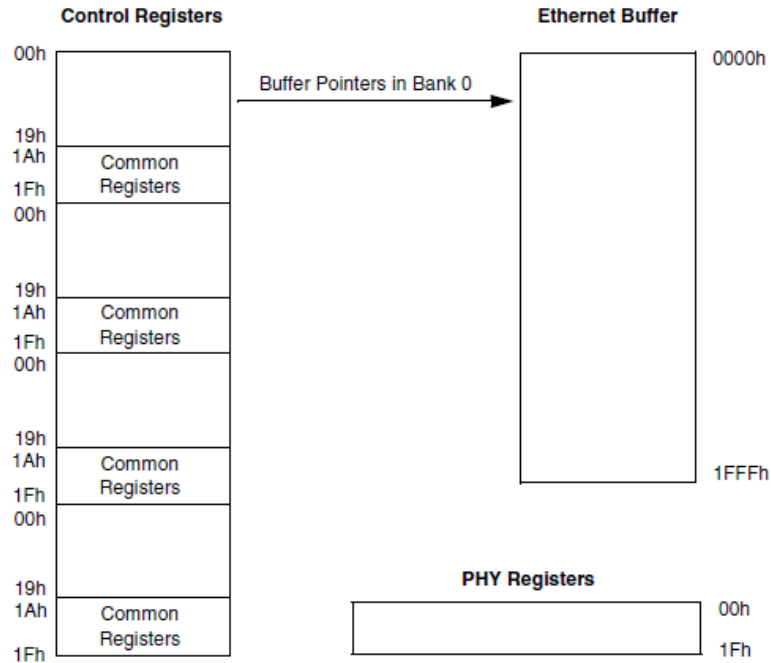
- 1 Kontrolní registry,
- 2 Ethernet buffer,
- 3 PHY registry.

Kontrolní registry jsou používány pro řízení, nastavování a kontrolování stavu komunikačního kontroléru ENC28J60. Tyto registry jsou zapisovány a čteny přímo pomocí SPI rozhraní. Tyto registry zabírají celkem 128B paměti.

Ethernet buffer je zásobník pro přijímání a odesílání Ethernet rámců a je mu vyhrazena samostatná paměť. Buffer má k dispozici přes 8192B.

Registry PHY jsou používány pro řízení, nastavování a kontrolování stavu modulu fyzického rozhraní. Tyto registry jsou přístupné pouze pomocí Media Interface Management (MIM), které jsou implementovány v MAC.²⁸

²⁸ [4] str. 13



Obr. 4.5 Organizace paměti ENC28J60²⁹

4.2.3.1 Kontrolní registry ENC28J60

Nyní se budu podrobněji zabývat kontrolními registry komunikačního mikrokontroléru ENC28J60. Bude zde rozebrána jejich paměťová organizace, přístup k jednotlivým registrům a význam dvou nejdůležitějších z nich.

Bank 0 Address	Name	Bank 1 Address	Name	Bank 2 Address	Name	Bank 3 Address	Name
00h	ERDPTL	00h	EHT0	00h	MACON1	00h	MAADR5
01h							
↑							
↓							
10h							
1Ah	Reserved	1Ah	Reserved	1Ah	Reserved	1Ah	Reserved
1Bh	EIE	1Bh	EIE	1Bh	EIE	1Bh	EIE
1Ch	EIR	1Ch	EIR	1Ch	EIR	1Ch	EIR
1Dh	ESTAT	1Dh	ESTAT	1Dh	ESTAT	1Dh	ESTAT
1Eh	ECON2	1Eh	ECON2	1Eh	ECON2	1Eh	ECON2
1Fh	ECON1	1Fh	ECON1	1Fh	ECON1	1Fh	ECON1

Obr. 4.6 Organizace kontrolních registrů ENC28J60³⁰

Z Obr. 4.6 vyplývá, že ke kontrolní registry jsou rozděleny do čtyř banků. V každém banku je 32 registrů, z toho posledních 6 (adresy 1Bh – 1Fh) je stejných a to z důvodu univerzálnosti přístupu, kdy nezáleží na vybraném banku ale pouze na adrese. Jsou to hlavní kontrolní registry. Na adresách 00h – 10h jsou registry kontrolující například přístup do bufferu (Bank0), kontrolu CRC (Bank1), přijímání a odesílání rámců na MAC podvrstvě (Bank2), Nastavení MAC vrstvy (Bank 3).

²⁹ [4] str. 11, Figure 3-1

³⁰ Upraveno z [4] str. 12, Table 3-1

Mikrokontrolér ENC28J60 dělí kontrolní registry do tří skupin:

1. ETH,
2. MAC,
3. MII.

Rozdíl v těchto registrech je v jejich čtení pomocí SPI rozhraní.

Jedním z nejdůležitějších registrů je registr ECON1. Tento registr, sídlící na adrese 1Fh, je potřebný pro výběr banků a nastavuje příznaky komunikace. Na Obr. 4.7 je vidět význam jednotlivých bitů v registru ECON1.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXRST	RXRST	DMAST	CSUMEN	TXRTS	RXEN	BSEL1	BSEL0
bit 7							bit 0

Obr. 4.7 Registr ECON1³¹

bit 7 TXRST: Transmit Logic Reset Bit

- 1 = Odesílání je v reset stavu
- 0 = Běžný chod kontroléru

bit 6 RXRST: Receive Logic Reset Bit

- 1 = Přijímání je v reset stavu
- 0 = Běžný chod kontroléru

bit 5 DMAST: DMA Start and Busy Status Bit

- 1 = DMA pracuje
- 0 = DMA nepracuje

bit 4 CSUMEN: DMA Checksum Enable bit

- 1 = DMA hardware počítá kontrolní součet
- 0 = DMA hardware kopíruje buffer

bit 3 TXRTS: Transmin Request to Send bit

- 1 = Odesílací logika se pokouší o odeslání paketu
- 0 = Odesílací logika nepracuje

bit 2 RXEN: Receive Enable bit

- 1 = Paket, který prošel nakonfigurovaným filtrem, bude zapsán do přijímacího bufferu
- 0 = Všechny pakety jsou ignorovány

bit 1-0 BSEL1.BSEL0

- 11 = SPI přístup k registrům banku 3
- 10 = SPI přístup k registrům banku 2
- 01 = SPI přístup k registrům banku 1
- 00 = SPI přístup k registrům banku 0

³¹ [4] str 15, register 3-1

Druhým důležitým registrem je ECON2, který se nachází na adrese 1Eh. Tento registr je používán pro nastavování hlavních funkcí mikrokontroléru ENC28J60.

R/W-1	R/W-0 ⁽¹⁾	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
AUTOINC	PKTDEC	PWRSV	r	VRPS	—	—	—
bit 7							bit 0

Obr. 4.8 Registr ECON2³²

bit 7 AUTONIC: Automatic Buffer Pointer Increment Enable Bit

1 = Automatická inkrementace ERDPT nebo EWRPT při čtení, nebo zápisu do EDATA

0 = Nemění se ERDPT a EWRPT po přistoupení do bufferu

bit 6 PKTDEC: Packet Decrement Bit

1 = Dekrementace EPKTCNT registru o jednu

0 = EPKTCNT zůstává beze změny

bit 5 PWRSV: Power Save Enable Bit

1 = MAC, PHY a kontrolní logika jsou přepnuty do Low-Power módu

0 = Běžný chod

bit 4 Reservován (obvykle „0“)

bit 3 VRPS: Voltage Regulator Power Save Enable Bit

Pokud je PWRSV=1:

1 = Vnitřní regulátor je v módu nízkého proudu

0 = Vnitřní regulátor je v běžném chodu

Pokud je PWRSV=0:

Tento bit je ignorován

bit 2–0: Neimplementováno, Čteno jako „0“

Pro nás jsou z tohoto registru nejdůležitější 7. a 6. bit, které nastavují čtení z bufferu a posun paměťového pointeru.

4.2.3.2 Organizace Ethernetového Bufferu

Jak již bylo naspáno výše, Ethernetový buffer má velikost 8kB, která je rozdělena na dvě části:

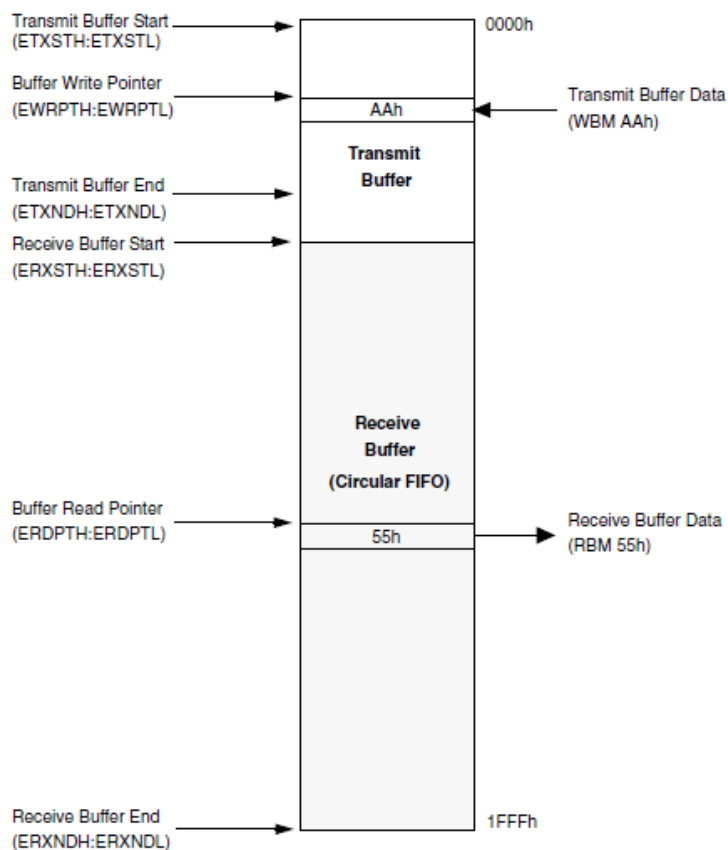
1. Přijímací buffer,
2. Odesílací buffer.

Velikost a adresace obou paměťových částí je plně nastavitelná pomocí rozhraní SPI.

³² [4] str. 16, register 3-2

Jak pro přijímání, tak pro odesílání jsou potřeba komunikační registry, které se nachází v banku 0. Podrobným popisem těchto registrů se nebudu zabývat, jelikož se jedná většinou o posuvné registry a pointery.

Daleko zajímavější je pro nás výchozího rozdělení Ethernetového bufferu, které je vidět na Obr. 4.9 Organizace Ethernetového Bufferu ENC28J60.



Obr. 4.9 Organizace Ethernetového Bufferu ENC28J60³³

Následující podkapitola je věnována základním vlastnostem a signálům SPI rozhraní. Do této kapitoly je vnořena z důvodu lepší návaznosti na následující problematiku komunikace řídicího mikro počítače s komunikačním mikrokontrolérem.

4.2.4 SPI rozhraní

Rozhraní SPI je sériovým komunikačním rozhraní typu master (pán), slave (otrok). To znamená, že jedno zařízení (master) říká druhému (slave), co má v daný moment dělat. Komunikační rozhraní SPI používá čtyři druhy signálů:

1. SCK³⁴,
2. SS³⁵,
3. MOSI³⁶,
4. MISO³⁷.

³³ [4] str 18, figure 3-2

³⁴ Signal clock

³⁵ Slave select

³⁶ Master out slave in

Řazení těchto signálů jsem zvolil záměrně, protože zhruba odpovídají tomu, v jakém pořadí musí být na sběrnici přítomny. Signál SCK musí být přítomen, jelikož ten dává celé sériové komunikaci hodinový takt. Tento signál je většinou přítomen, i pokud zrovna zařízení nekomunikují, jelikož bývá pro větší přesnost odvozen externím krystalem.

Signál SS je spouštěcím signálem celé komunikace. Pokud se na sběrnici objeví tento signál, slave je připraven přijmout řídicí příkaz od mastera. Ve většině zařízení je vstup tohoto signálu negovaný, takže je spínáný sestupnou hranou. Využití tohoto signálu je především v aplikacích s jedním masterem a více slavy, kde tito jsou voleny právě pomocí SS signálu a zabráňuje se tudíž kolizím.

Signál MOSI následuje po signálu SS. Jedná se o příkazy, nebo data mastera posílané slavu. Příkazy mastera jsou definovány instrukční sadou slava.

Signál MISO je naopak odpověď slavu na příkaz mastera. Jedná se především o posílání obsahu paměti, nebo řídicích registrů.

Tento výčet základních vlastností rozhraní SPI, stačí pro pochopení další práce. Bližší informace jsou k dostání např. na webových stránkách [8].

4.2.5 SPI rozhraní v ENC28J60

Jak bylo napsáno výše, master je povinován dodržovat správnou instrukční sadu slavu. Instrukční SPI sada komunikačního mikrokontroléru ENC28J60 je uvedena v tabulce 5.

Tabulka 5: Instrukční sada SPI rozhraní kontroléru ENC28J60³⁸

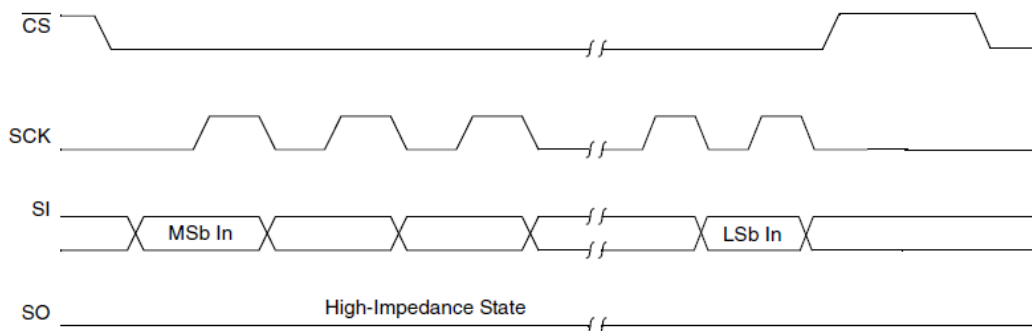
Název instrukce	Byte 0		Byte 1 (příp. další byty)
	Opcode	Argument	Data
Čtení kontrolního registru (RCR)	0 0 0	a a a a a	N/A
Čtení z bufferu (RBM)	0 0 1	1 1 0 1 0	N/A
Zápis kontrolního registru (WCR)	0 1 0	a a a a a	d d d d d d d d
Zápis do bufferu (WBM)	0 1 1	1 1 0 1 0	d d d d d d d d
Nastavení bitového pole (BFS)	1 0 0	a a a a a	d d d d d d d d
Nulování bitového pole (BFC)	1 0 1	a a a a a	d d d d d d d d
Příkaz systémového restu (SRC)	1 1 1	1 1 1 1 1	N/A

Důležitým aspektem jakékoli sériové komunikace je časování (hodinový signál) a pořadí, v jakém jsou bity vysílány. U ENC28J60 toto popisují Obr. 4.10 a Obr. 4.11.

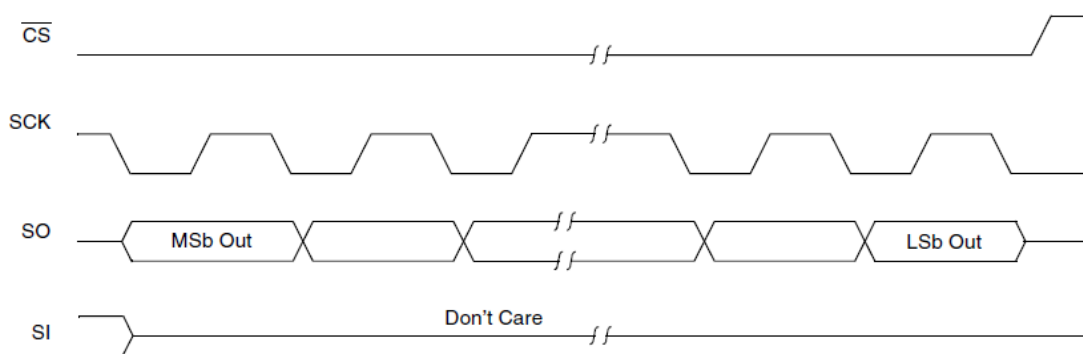
³⁷ Master in slave out

³⁸ [4] table 4-1, str. 28

Nejdůležitější pro nás je posloupnost vstupních a výstupních bytů, které začínají MSb³⁹ a to, že čtení bytů probíhá na nástupné hraně hodinového signálu.

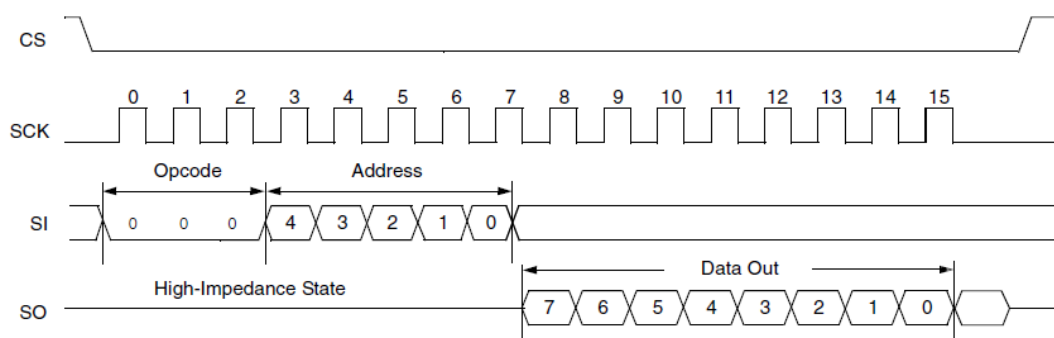


Obr. 4.10 Vstupní časování ENC28J60⁴⁰



Obr. 4.11 Výstupní časování ENC28J60⁴¹

Čtení kontrolních registrů ENC28J60 je různé, podle toho o jakou skupinu se jedná. U registrů skupiny ETH následuje hned po přijetí adresy odeslání obsahu registru. U skupin registrů MAC a MII se mezi adresu a obsah vkládá tzv. dummy bajt, který je tvořen samými log. „1“.

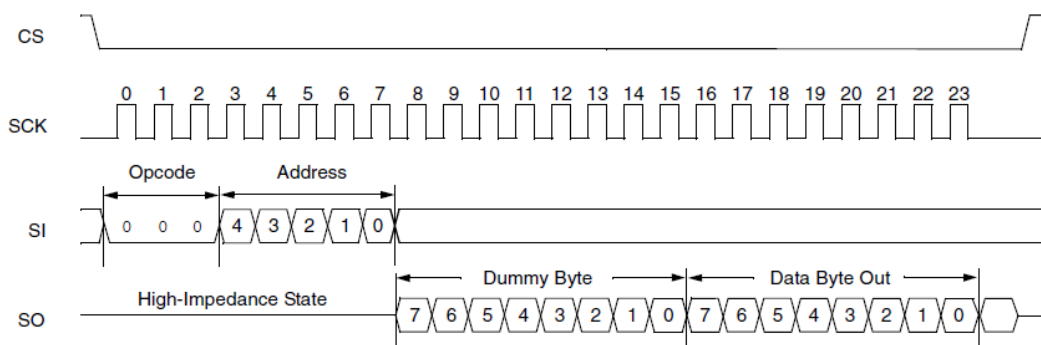


Obr. 4.12 Čtení kontrolního registru skupiny ETH⁴²

³⁹ Most significant bit (7. bit bajtu)

⁴⁰ [4] – str. 25; figure 4-1

⁴¹ [4] – str. 25; figure 4-2



Obr. 4.13 Čtení kontrolního registru skupiny MAC a MII⁴³

4.3 Řídicí mikrokontrolér ATmega16

Jako řídicí mikrokontrolér, který bude zprostředkovávat řízení komunikačního mikrokontroléru ENC28J60, byl vybrán mikroprocesor ATmega16 z rodiny mikroprocesorů AVR od firmy Atmel. Mikročip je vystaven na architektuře RISC. Zvolen byl na základě jeho dostupnosti (cenové i kvantitativní) i vzhledu k velikosti paměti (16kB), která se zdá být pro naše potřeby dostatečná.

4.3.1 Základní vlastnosti ATmega16

Mikroprocesor ATmega16 disponuje 131 instrukcemi, převážně vykonávanými v jednom cyklu a může využívat až 32 osmi-bitových pracovních registrů. Pracovní kmitočet mikroprocesoru ATmega16 je doporučen do 16MHz.

Z hlediska paměti pak mikroprocesor využívá 16kB flash paměti pro uložení programu (možné přepsat až 10 000), 512 bytů paměti EEPROM (možné přepsat až 100 000) a vnitřní paměti SRAM o velikosti 1kB.

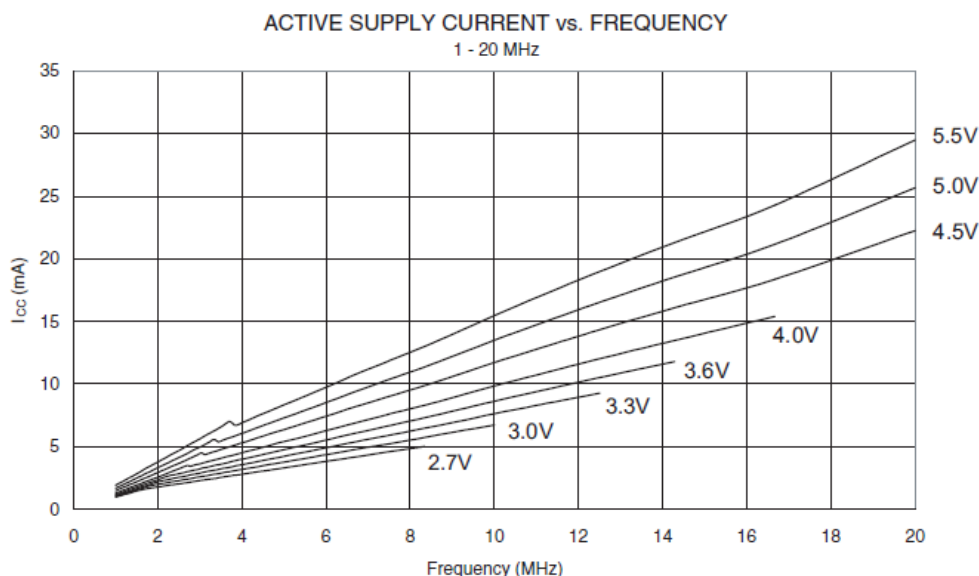
Pro nás je nejdůležitější, že mikroprocesor disponuje dostatečným množstvím pinů, které budou používány pro odladění programu, kdy přečtený registr z komunikačního mikrokontroléru může být zobrazen na řadě LED diod připojených k I/O portům. Další důležitou vlastností je, že mikroprocesor ATmega16 plně podporuje SPI rozhraní a to nejen na úrovni slave, ale i na úrovni master⁴⁴.

Podle Obr. 4.14 je potřeba pro co nejvyšší pracovní kmitočet (tj. 20MHz) zajistit napájení od 4,5V do 5,5V (spíše víc, než méně). Z tohoto obrázku se dá odečíst i předpokládaný proudový odběr mikrokontroléru, který bude činit přibližně 26mA.

⁴² [4] – str. 27; figure 4-3

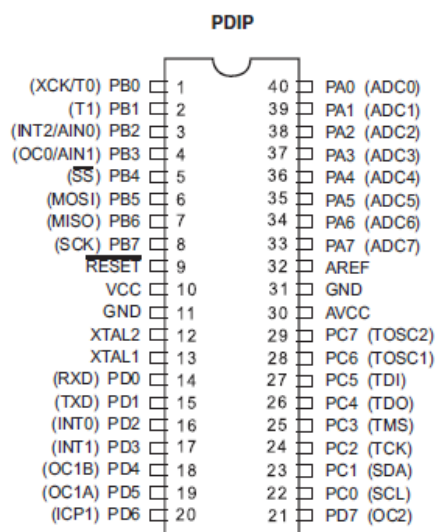
⁴³ [4] – str. 27; figure 4-4

⁴⁴ [5] – str. 1



Obr. 4.14 Graf závislosti odebraného proudu na taktovacím kmitočtu a napětí⁴⁵

Mikropočítač ATmega16 má pro potřeby programátora a uživatele 32 I/O pinů, 5 piny napájecích nebo referenčních, dva piny pro připojení krystalu a jeden pin resetovací. Rozložení jednotlivých pinů pro pouzdro PDIP je zobrazeno na Obr. 4.15.



Obr. 4.15 Rozložení pinů ATmega16, pouzdro PDIP⁴⁶

Jak je z Obr. 4.15 Rozložení pinů ATmega16, pouzdro PDIP vidět, mají všechny piny více konfigurovatelných vlastností. V rámci této práce se budeme zabývat pouze těmi vlastnostmi daných pinů, které jsou nezbytné pro správný chod výrobku a odladění programu. Tyto vybrané vlastnosti jsou stručně popsány v tabulce 6⁴⁷.

⁴⁵ [5] – str. 300, figure 150

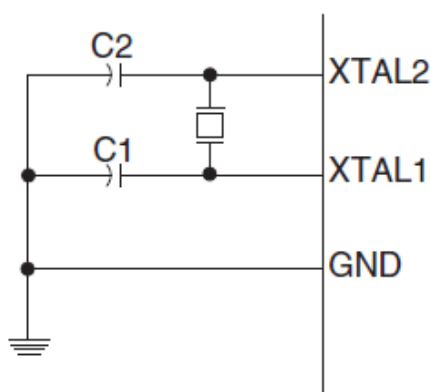
⁴⁶ [5] – str. 2

⁴⁷ Ve sloupci Název pinu je uveden pouze název pro vybranou vlastnost

Tabulka 6: Popis pinů ATmega16⁴⁸

Název pinu	Číslo pinu	Popis
PB0	1	I/O pin, hardwarový reset kontroléru ENC28J60
PB1	2	I/O pin, indikační LED spuštění
INT2	3	Přerušení přijímané od kontroléru ENC28J60
PB3	4	I/O pin
\overline{SS}	5	Slave select
MOSI	6	Výstupní datový tok z řídicího kontroléru sběrnice SPI
MISO	7	Vstupní datový tok do řídicího kontroléru sběrnice SPI
SCK	8	Hodinový signál sběrnice SPI
\overline{RESET}	9	Hardwarový reset mikropočítače
VCC	10	Napájení mikrokontroléru 5V
GND	11, 31	Referenční hodnota napájení 0V
XTAL1	12	Vstup krystalu
XTAL2	13	Výstup krystalu
PD0-PD7	14-21	I/O port
PC0-PC7	22-29	I/O port
AVCC	30	Podpurné napětí pro vnitřní A/D převodník
AREF	32	Analogová reference pro vnitřní A/D převodník
PA0-PA7	33-40	I/O port

Stejně jako kontrolér ENC28J60 má i ATmega16 doporučeno připojení některých pinů. U mikropočítače ATmega16 se jedná především o připojení externího krystalu, který je stejně jako v případě kontroléru ENC28J60 připojen k zemi přes dva paralelní kondenzátory, které slouží jako filtr.



Obr. 4.16 Doporučené připojení krystalu k ATmega16⁴⁹

⁴⁸ [5] – str. 4-5

⁴⁹ [5] – str. 26, figure 12

Tabulka 7: Ceny součástek použitých pro výrobu vývojového kitu s ENC28J60⁵²

Použitá součástka	Množství [Ks]	Cena
Krystal 25MHz	1	10,01 Kč
ENC28J60	1	90,50 Kč
Tlumivka LQH	1	16,13 Kč
Kondenzátor 470nF	2	6,16 Kč
Přízpusobovací transformátor HXN1188NC	1	52,80 Kč
Rezistor 2,32 k Ω	1	0,85 Kč
Rezistor 49,9 Ω	4	1,05 Kč
Kondenzátor 100nF	2	0,50 Kč
Kondenzátor 10 μ F	1	2,50 Kč
Rezistor 220 Ω	2	0,96 Kč
Kondenzátor 1nF	1	2,10 Kč
Rezistor 75 Ω	4	0,96 Kč
Stabilizátor LF33CDT	1	17,00 Kč
Konektor RJ-45	1	22,62 Kč
	Celkem	237,79 Kč

Jak je z Tabulka 7 patrné, výrobní cena by byla asi 3x nižší, než samotný nákup. Přesto bylo rozhodnuto vývojový kit s mikrokontrolérem ENC28J60 nevyrábět a to z časové a hardwarové (technologick) náročnosti.

Další potřebnou vývojovou sadou byl programátor společně s něčím, na čem se bude ladit program. V úvahu přicházela opět možnost koupit kit již hotový, nebo navrhnout a vyrobit si vlastní. Ceny vývojových kitů obsahující některý z mikropočítačů z rodiny ATmega se pohybují od 1200 Kč do 2 000 Kč⁵³. Tyto sady již obsahují programátor, který by, v případě vlastního návrhu, bylo nutné dokoupit. V tabulce 8 jsou uvedeny ceny součástek pro výrobu vývojového kitu s ATmega16.

Tabulka 8: Ceny součástek použité pro vývojový kit s ATmega16⁵⁴

Použitá součástka	Množství [Ks]	Cena
ATmega16	1	179,00 Kč
Zelené LED	8	1,68 Kč
Žluté LED	8	16,13 Kč
Červená LED	5	3,70 Kč
Vypínač	1	2,91 Kč
Konektor pro napájení	1	8,51 Kč
Patice DIL40	1	28,34 Kč
	Celkem	379,74 Kč

⁵² Ceny jsou uvedeny k datu 30.12.2014 a jsou převzaty ze serveru www.gme.cz

⁵³ <http://www.gme.cz/vyvojove-prostredky>

⁵⁴ Ceny jsou uvedeny k datu 30.12.2014 a jsou převzaty ze serveru www.gme.cz

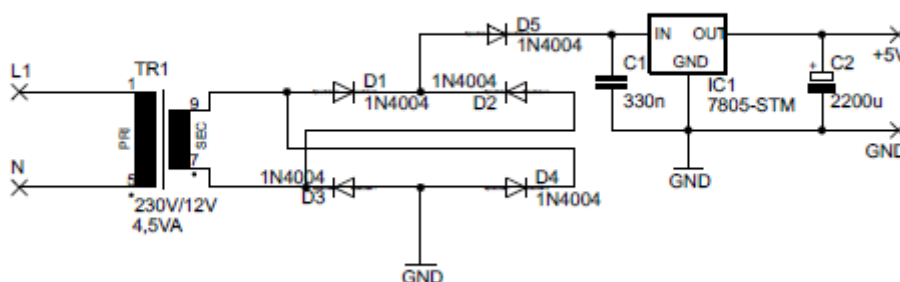
Cena je u mého návrhu takřka 4x nižší, než u kupovaného výrobku. Důvodem je především použití externího programátoru (ASIX PRESTO [7]), který je k dispozici. Tento kit je pouze k odladění programu a tudíž se jím dále nebudu zabývat.

4.5 Elektrická schémata

Od tohoto místa dále již začíná aplikování jednotlivých poznatků získaných v předchozích kapitolách. Návrhy elektrických schémat a návrhy desek plošných spojů byly vytvářeny v návrhovém prostředí CadSoft EAGLE 7.0.1 se studentskou licencí a v mnoha případech vychází z doporučených schémat z katalogových listů výrobců jednotlivých součástek.

4.5.1 Elektrické schéma napájecího zdroje

Jedním z důvodů, proč chci začít tuto kapitolu schématem zdroje, je jednoduchost návrhu a zároveň zde ukážu základní výpočty, na které se budu odkazovat v dalších podkapitolách.



Obr. 4.18 Elektrické schéma napájecího zdroje

Schéma zdroje se skládá z transformátoru TR1, diodového dvoucestného usměrňovacího můstku (D1-D4), diody D5, která zabráňuje přepólování kondenzátoru C1 a stabilizátoru napětí IC1 a vyhlazovacího kondenzátoru C2.

4.5.1.1 Určení parametrů stabilizátoru IC1

Parametry stabilizátoru byly určeny podle výstupních parametrů, tj. podle výstupního napětí a výstupního odebíraného proudu a podle nabízených parametrů výrobců a ceny zařízení.

Výstupní napětí je jasně dáno napájením řídicího mikro počítače, tj. 5V. Z katalogových parametrů jsou nejdůležitější úbytek napětí na prvku (U_{drop}), maximální možný dodávaný proud (I_f), nominální výstupní napětí (U_{out}) a maximální vstupní napětí ($U_{in,max}$). Vždy je otázkou kompromisu, které parametry zvolíme jako hlavní, a které jako vedlejší. Vzhledem k tomu, že se v našem případě nejedná o žádnou specifickou aplikaci, mohl být zvolen stabilizátor s vyšším úbytkem napětí s důsledkem nižší ceny. Výstupní proud stabilizátoru je potom volen podle odebíraného proudu obvodu napájeného zařízení. Výstupní proud je tedy vypočítán dle vztahu (6).

$$I_{\max} = I_{AT} + I_{ENC} + I_r \quad (6)$$

Kde:

I_{AT} je proud odebíraný mikropočítačem ATmega16 (cca 26 mA při 5V a 20MHz⁵⁵)

I_{ENC} je proud odebíraný mikrokontrolérem ENC28J60 (max. 180 mA⁵⁶)

I_r je proud rezervní (počítáno 30%⁵⁷).

Jako referenci rezervního proudu jsem volil odběry řídicího mikropočítače a komunikačního mikrokontroléru, jelikož tato dvě zařízení mají většinový odběr. Proud byl tedy zvolen jako 30% ze součtu odběrů ATMegy16 a mikrokontroléru ENC28J60. Z tohoto tedy vyplývá:

$$I_{\max} = 180 \cdot 10^{-3} + 26 \cdot 10^{-3} + 0,25(80 \cdot 10^{-3} + 26 \cdot 10^{-3}) \approx 257 \text{ mA}$$

Maximální možný proud dodávaný stabilizátorem tedy musí být alespoň 260 mA. Parametry zvoleného stabilizátoru IC1 jsou uvedeny v tabulce Tabulka 9.

Tabulka 9: Parametry zvoleného stabilizátoru IC1⁵⁸

U_{drop} [V]	U_{out} [V]	$U_{in\max}$ [V]	I_{\max} [V]
2	5	35	750 mA

4.5.1.2 Výpočet parametrů transformátoru TRI

První bylo potřeba navrhnout výkon transformátoru a jeho výstupní napětí. Výstupní napětí bylo zvoleno vzhledem k nutnosti +5V napájecího napětí na výstupu zdroje. Mnou zvolený stabilizátor IC1 má na sobě úbytek napětí cca 1,7V při 25°C a odebíraných cca 250mA⁵⁹. Další napěťovou ztrátou jsou diody v usměrňovacím můstku plus dioda D5, která chrání stabilizátor IC1 proti přepólování. Můžeme počítat, že každá z těchto diod má napěťovou ztrátu cca 0,7V. Celkový úbytek napětí pak tedy spočteme dle vztahu (7).

$$U_{ztr} = U_{IC1} + 3U_D \quad (7)$$

$$U_{ztr} = 1,7 + 3 \cdot 0,7 = 3,8 \text{ V}$$

Napětí na výstupu transformátoru pak zjistíme tak, že výstupní napětí 5V přičteme ke ztrátovému napětí.

⁵⁵ [5] str. 300, Figure 150; Kapitola 4.3.1

⁵⁶ [4] str. 80, 16.1; Kapitola 4.2.1

⁵⁷ 30% z $I_{AT} + I_{ENC}$. Rezerva je počítána pro LED, budiče atd.

⁵⁸ 53[6]

⁵⁹ [5] str. 15, Figure 4

$$U_{outTrans} = U_{ztr} + U_{vyst} = 3,8 + 5 = \underline{8,8V}$$

Nejlépe by tedy vyhovoval transformátor s výstupním napětím minimálně 10V. Bohužel se sériově vyrábějí pouze transformátory s napětím na sekundárním vinutí 9V a 12V. Vzhledem k rezervě byl tedy zvolen transformátor s výstupním napětím +12V.

Dalším úkolem bylo zvolit výkon transformátoru. Tento výkon je volen vzhledem k předpokládanému maximálnímu odběru proudu, který je vypočítán dle vztahu (6) a který činí 260 mA.

Zdánlivý výkon transformátoru byl pak spočítán na základě maximálního odebíraného proudu a napětí na sekundárním vinutí dle vztahu (8) $S_{trans} = U_{sek} \cdot I_{max}$ (8).

$$S_{trans} = U_{sek} \cdot I_{max} \quad (8)$$

Výsledná hodnota zdánlivého výkonu je $S_{transf} = 12 \cdot 257 \cdot 10^{-3} = \underline{3,084VA}$

Vzhledem k vypočtenému výkonu a možnosti nákupu sériového transformátoru byla zvolena nejbližší vyšší hodnota výkonu, tj. 4,5VA.

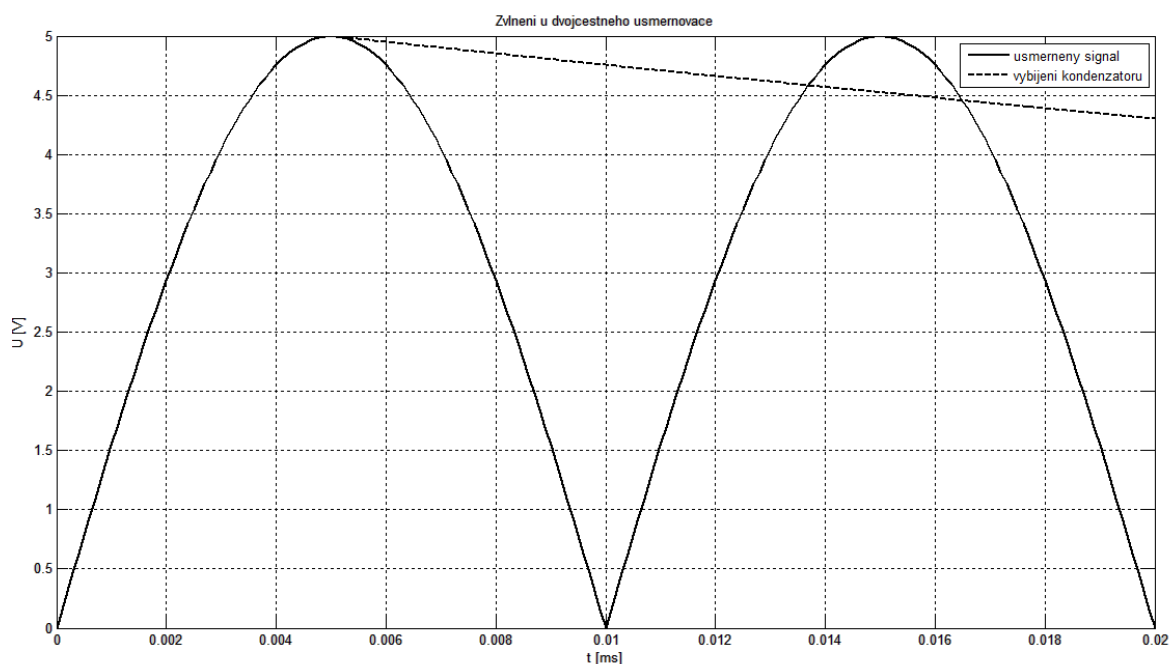
4.5.1.3 Výpočet kapacit kondenzátorů C1, C2

Jak již bylo výše řečeno, kondenzátor C2 slouží ke snížení zvlnění u dvojcestného usměrňovače. Pro výpočet kapacity kondenzátoru je třeba nejprve zjistit, jaké hodnoty v čase budeme potřebovat.

Jasně je, že na vstupu kondenzátoru bude maximální napětí $U_c = 5V$, které získáváme ze stabilizátoru IC1. Za minimální hodnotu budeme uvažovat doporučenou minimální hodnotu mikroprocesoru pro jeho stabilní výkon⁶⁰, což jest přibližně 4,5V. Z uvedeného vyplývá, že maximální přípustné napětí, o které se kondenzátor může vybít je 0,5V.

Dále je pak nutno zjistit, jak dlouho se bude tento kondenzátor vybíjet, aby po dobu vybíjení zajistil příslušné napětí. Z Obr. 4.19, vyplývá, že kondenzátor se bude vybíjet z maximální hodnoty až do následující půlperiody.

⁶⁰ Obr. 4.14 Graf závislosti odebíraného proudu na taktovacím kmitočtu a napětí



Obr. 4.19 Model dvojecestně usměrněného napětí

Matematicky lze zapsat, že pro dvojecestně usměrněný signál platí vztah (9). Z tohoto vztahu, jsme schopni určit, kdy dosáhne vybíjení kondenzátoru hraničních 4,5V. Tuto dosáhneme odvozením (12) z (11).

$$u(t) = U_0 |\sin(\omega t)| \quad (9)$$

$$t = \frac{1}{2\pi f} \arcsin\left(\frac{u(t)}{U_0}\right) + \frac{T}{2} \quad (10)$$

Jelikož se pohybujeme ve druhé půlperiodě, a v absolutní hodnotě, je tudíž nutné připočítat polovinu periody abychom se časově dostali do druhé půlperiody. Z (10) a vstupních hodnot zjistíme, že hodnota 4,5V nastane cca v čase $t \cong 13,5ms$. Z uvedené doby vyplývá, že kondenzátor se bude vybíjet $t_{vyb} = 8,5ms$. Tuto hodnotu jsme dostali z (11).

$$t_{vyb} = t - \frac{T}{4} \quad (11)$$

Z rovnice vybíjení kondenzátoru (12) a z doby vybíjení můžeme spočítat velikost příslušného stabilizačního kondenzátoru.

$$u_c(t) = U_0 e^{-\frac{t}{\tau}} \quad (12)$$

kde:

U_0 maximální napětí na kondenzátoru

t je doba vybíjení

τ je vybíjecí konstanta, která je rovna $\tau = RC$, kde R je odpor do kterého se kondenzátor vybíjí a C je kapacita kondenzátoru.

Z výše uvedeného vztahu je zřejmé, že jediné, co neznáme je vybíjecí odpor, který je počítán podle napětí a odebíraného proudu dle vztahu (13), kde ΔU je rozdíl mezi maximálním a minimálním přípustným napětím.

$$R = \frac{\Delta U}{I} \quad (13)$$

Dosadíme-li do (13) naše hodnoty, tj. za $\Delta U = 0,5V$ a $I \cong 260mA$, získáme hodnotu vstupního odporu obvodu, který je roven cca hodnotě $R = 1,923\Omega$.

Hodnotu vstupního odporu obvodu dosadíme do (14), kterou jsme si odvodily z (13), a vyjádřili jsme si kapacitu kondenzátoru $C2$. Po dosazení hodnot nám vyjde hodnota kondenzátoru cca $C = 443\mu F$.

$$C = \frac{-t}{R} \frac{1}{\ln\left(\frac{u_c(t)}{U_0}\right)} \quad (14)$$

Nejbližší vyšší hodnotou vyráběných kondenzátorů je $820\mu F$, ale s tolerancí 20%. V případě maximální špatné hodnoty by kapacita kondenzátoru byla $656\mu F$. Kondenzátor o velikosti $820\mu F$ je tudíž dostačující.

Kapacita $C1$ je pak volen s ohledem na katalogový list stabilizátoru⁶¹. V doporučení se jedná o hodnotu $330nF$. Tento kondenzátor slouží k odfiltrování špičkových napětí na výstupu transformátoru a tudíž k ochraně stabilizátoru IC1.

4.5.2 Elektrické schéma hlavní desky

Vzhledem k použití vývojového kitu s mikrokontrolérem ENC28J60, se samotné elektrické schéma hlavní desky skládá pouze z mikroprocesoru ATmega16 a konektoru pro připojení k vývojovému kitu. Elektrické schéma hlavní desky je uvedeno na Obr. 4.20.

Nejvýraznějším prvkem hlavní desky je řídicí mikroprocesor procesor ATmega16 (IC1). Přes konektor SV1 komunikuje s vývojovým kitem, na kterém je komunikační mikrokontrolér ENC28J60.

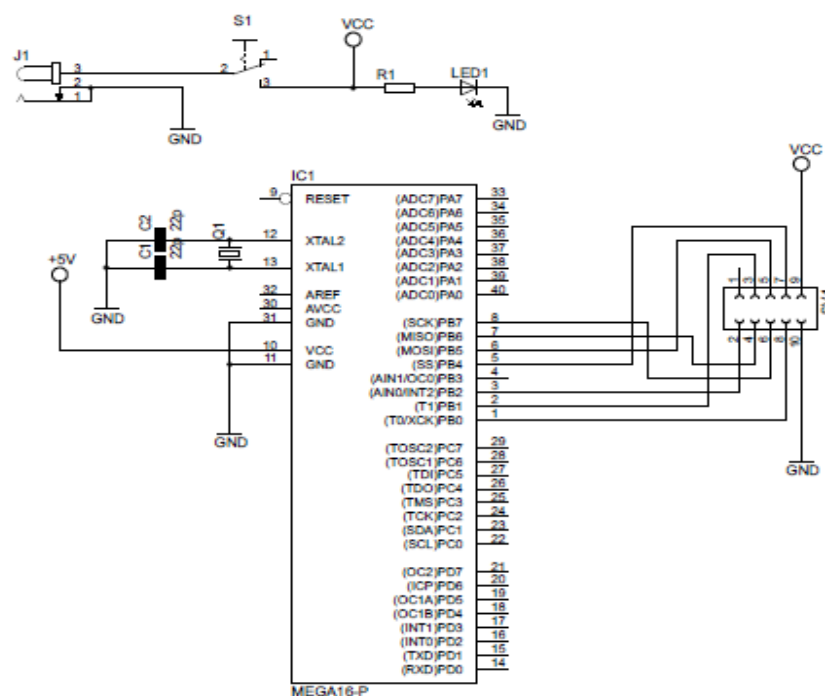
Z důvodu stability taktovacího signálu je použit externí krystal Q1. Tento je zapojen podle doporučení výrobce⁶². Hodnoty kondenzátorů $C1$ a $C2$ jsou specifikovány ve stejném doporučení⁶³.

⁶¹ [5] figure 4, str. 4

⁶² [5] figure 12, str. 26

⁶³ [5] table 4, str. 26

Přepínač S1 slouží pouze k přivedení napájecího napětí ze zdroje, a dioda LED1 pak indikuje přítomnost napájecího napětí na desce.



Obr. 4.20 Elektrické schéma výrobku

4.5.3 Elektrické schéma vývojového komunikačního kitu

Jak již bylo napsáno, rozhodl jsem se zakoupit vývojový komunikační kit s mikrokontrolérem ENC28J60, který bude připojen pomocí konektoru k hlavní desce. Konkrétní schéma tohoto kitu není od výrobce k dispozici, což nevadí, protože je takřka totožné s doporučeným schéma výrobce mikrokontrolér ENC28J60. Toto schéma je na Obr. 4.17.

Po seznámení se s vývojovým kitem, se tento liší pouze v tom, že používá již konektor RJ-45, který obsahuje přizpůsobovací transformátory a LED diody indikující vysílání a přijímání dat.

Mezi výstupními piny SPI rozhraní komunikačního mikrokontroléru ENC28J60 a vstupními piny SPI rozhraní řídicího mikroprocesoru AtMega16 je zařazen budič sběrnice. Tento budič má za úkol přizpůsobit 3,3V logiku vývojového komunikačního kitu s 5V logikou řídicí desky. Pro ilustraci je na Obr. 4.21 uvedeno částečné elektrické schéma vývojového komunikačního kitu.

5 Program

Program byl psán v jazyce BASCOM, ve vývojovém prostředí BASCOM-AVR. Jedná se o jazyk speciálně vyvinutý pro programování procesorů společnosti Atmel z rodiny AVR. Jazyk samotný je podobný jazyku C, ale obsahuje i možnost doplnění jazykem assembler.

Rozebírat zde v této kapitole celý program, by bylo na dlouho, tudíž zde rozeberu ty části programu, které považuji za nejdůležitější. Bude se jednat o nastavení SPI komunikace, výběr banku a odeslání paketu.

5.1 Nastavení SPI komunikace

```
Config Spi = Hard , Interrupt = Off , Data Order = Msb , Master = Yes ,  
Polarity = Low , Phase = 0 , Clockrate = 4 , Noss = 0
```

Klíčové slovo config označuje, co budeme v mikrokontroléru nastavovat. Bráno položky zleva, nastavujeme zde: Hardwarové rozložení pinů (SPI), komunikaci nebude indikovat přerušování (interrupt), pořadí bajtů je první MSb (Data Order)⁶⁴, mikrokontrolér je v komunikaci masterem (Master), hodinový signál je log. „0“, pokud komunikace po SPI sběrnici neprobíhá (Polarity), čtení signálu probíhá okamžitě po náběžné hraně bez zpoždění (Phase), s hodinovým signálem 1 MHz a mikroprocesor nebude automaticky generovat signál SS.

5.2 Select bank

```
Sub Enc28j60selectbank(bank As Byte)  
  A(1) = &B000_11111  
  Enc28j60_cs = 0  
  Spiout A(1) , 1  
  Spiin A(1) , 2  
  Enc28j60_cs = 1  
  A(2) = A(2) And &B1111_1100  
  A(2) = A(2) Or Bank  
  A(1) = &B010_11111  
  Enc28j60_cs = 0  
  Spiout A(1) , 2  
  Enc28j60_cs = 1  
End Sub
```

Rutina Enc28j60selectbank slouží k výběru banku pro čtení a zápis kontrolních registrů. Vstupním parametrem je proměnná bank, která je definována v pomocném souboru jako binární číslo velikosti bajtu, které se zapisuje do kontrolního registru ECON1⁶⁵.

Než budeme přepisovat registr, je třeba zajistit, aby byla přepsána pouze správná část. Proto si nejprve přečteme obsah registru ECON1 a to tak, že do pole A() vložíme jeden bajt, který obsahuje op kód⁶⁶ pro čtení kontrolního registru a adresu registru ECON1.

⁶⁴ Viz kapitola 4.2.5

⁶⁵ Viz kapitola 4.2.3.1

⁶⁶ Tabulka 5, kapitola 4.2.5

Toto pole odešleme pomocí funkce `Spiout`. Pro přijímání dat slouží funkce `Spiin`, kterou musíme opatřit argumentem očekávané velikosti dat. Podle kapitoly 4.2.5 budeme očekávat vždy příjem dvou bytů.

Z praktických zkušeností mám ověřeno, že druhý bajt je bajt vyčtených hodnot registrů. Nejprve si vynulujeme daný registr `ECON1` na pozicích, kde se nachází `BSEL` bity. To je provedeno logickým součinem, takže na pozicích násobených log. „1“ zůstanou původní log. hodnoty a na pozicích násobených log. „0“ budou log. „0“. Následně tentýž bajt logicky sečteme s bajtem `bank`, který změní hodnoty pouze na pozicích `BSEL` bitů.

V posledním kroku umístíme na první pozici pole `op` kód zápisu kontrolního registru a adresy registru `ECON1` a odešleme pole `A()` po sběrnici `SPI`.

Všimněme si, že signál \overline{SS} (zde interpretován aliasem⁶⁷ `Enc28j60_cs`) vybírá `ENC28J60` pouze v případě aktivní komunikace `SPI` sběrnice.

5.3 Odeslání paketu

```
Sub Enc28j60packetSend(pcktlén As Word)
    Enc28j60_cs = 0

    Spdr = Enc28j60_write_buf_mem

    Do
    Loop Until Spsr.spif = 1

    Spdr = &B000_1110

    Do
    Loop Until Spsr.spif = 1

    For X = 1 To Pcktlén
        Spdr = Buffer(x)
        Do
        Loop Until Spsr.spif = 1
    Next X

    Enc28j60_cs = 1

    Value = 0
    Value.econ1_txrst = 1
    Call Enc28j60bitfield_set(econ1 , Value)
    Call Enc28j60bitfield_clear(econ1 , Value)
    Value = Low(txstart_init)
    Call Enc28j60writecontrolregbyte(ewrptl , Value)
    Value = High(txstart_init)
    Call Enc28j60writecontrolregbyte(ewrpth , Value)
    Value = Low(txstart_init)
    Value = Value + Low(pcktlén)
    Call Enc28j60writecontrolregbyte(etxndl , Value)
    Value = High(txstart_init)
    Value = Value + High(pcktlén)
    Call Enc28j60writecontrolregbyte(etxndh , Value)
    Value = 0
    Value.econ1_txrts = 1
```

⁶⁷ alias – v programování mikropočítačů se používá pro přejmenování určitého pinu (např. `PB1`) na název lidsky použitelný a více vypovídající


```
Call Enc28j60bitfield_set(econ1 , Value)
End Sub
```

Pro odesílání paketů slouží rutina `Enc28j60packetsend`, se vstupním parametrem `pcktlen`, který má velikost `Word` (16 bitů `unsigned`⁶⁸).

Nejprve si do registru `spdr` uložíme byte, kterým odešleme příkaz pro ENC28J60 že následuje zápis do bufferu. Tento příkaz je definován v konstantě `Enc28j60_write_buf_mem`, která je definována v pomocném souboru `enc28j60.inc`.

Komunikace Mikroprocesoru AtMega16 může fungovat i tak, že dokud je SPDR registr plný, odesílá data po SPI sběrnici. Následně pak v prázdné smyčce `DO – LOOP` čeká, dokud v registru `SPSR` nebude flag `SPIF` log „1“. Jakmile se toto stane, komunikace byla dokončena a mikroprocesor může přejít k další činnosti.

Dalším krokem je odeslání bajtu s hodnotou `0001 110`, což je přidání paritního bytu ke každému paketu[13]. Pak mikroprocesor opět čeká na odeslání tohoto bytu.

Následuje smyčka `for x - next x`, která vysílá postupně jeden bajt za druhým. Bajty jsou vysílány tak, že s každým novým cyklem je do registru `SPDR` načten nový bajt pomocí pole `buffer(x)`. Poté mikroprocesor opět čeká na odeslání bajtu.

Formule `next x` slouží k inkrementaci `x` a tudíž načtení dalšího bajtu do pole `buffer`. Toto vše probíhá, dokud se `x` nerovná proměnné `pcktlen`, která určuje délku odesílaného paketu.

Následující řádky slouží pouze k resetování stavu odesílací logiky mikrokontroléru ENC28J60.

⁶⁸ `unsigned` – značí datový typ, který má velikost počítanou od 0 do max. velikosti (v případě `Word` je to 0 - 65 535)

6 Oživení

Jak bylo napsáno výše, pro oživení byl použit vývojový kit vlastního návrhu (jeho schéma je uvedeno v příloze). Na tomto kitu byl odladován program a kontrolovány obsahy kontrolních, řídicích a komunikačních registrů mikrokontroléru ENC28J60.

Pro kontrolu bude použit program wireshark, příkazový řádek OS windows a terminál OS Linux Debian, který běžel pomocí emulátoru OracleVM virtual box.

Nejprve je potřeba nastavit na síťovém adaptéru, kde je připojeno zařízení, IPv4 adresu na hodnotu 192.168.1.x (kde x je libovolná adresa vyjma 60, což je adresa našeho zařízení) a podsíťovou masku 255.255.255.0. Výchozí bránu nastavovat není potřeba, protože zařízení bude pracovat v rámci stejné sítě.

Po odeslání zkušebního ICMP paketu na adresu zařízení, která je v programu nastavena na 192.168.1.60, byl v příkazovém řádku windows 7 zaznamenán tento řádek:

```
Odpoověď od 192.168.1.1: Cílový hostitel není dostupný.
```

Z tohoto řádku jasně vyplývá, že zařízení neodpovídá na ICMP dotaz. Je tedy třeba zjistit, proč. K tomuto účelu jsem použil program wireshark, který slouží k zachytávání paketů. Dle tohoto programu bylo zjištěno, že zařízení neodpovídá na ARP dotaz. Bohužel se mi z časových důvodů nepodařilo zjistit, kde v programu je chyba.

V rámci zkoušení jsem proto zavedl do ARP tabulky systému windows záznam odpovídající našemu zařízení pomocí tohoto řádku:

```
arp -s 192.168.1.60 43-c6-b3-f3-ab-9d
```

Ani po této úprav systémové tabulky jsem bohužel nedosáhl odpovědi na ICMP dotaz. Z časových důvodů bohužel nebylo možno oživit zařízení k plné funkčnosti. Nicméně, zařízení nějakým způsobem komunikuje (resp. Windows si nače, že se jedná o neznámou síť)

K další analýze a hledání důvodů proč zařízení nekomunikuje, by bylo zapotřebí najít komunikační prostředek, který bude mikrokontrolér ENC28J60 analyzovat za chodu (čtení aktuálního stavu paměti, aktuálního nastavení kontr. registrů atd.). Mnou navržený vývojový kit bohužel nespĺňuje možnost analýzy za chodu.

7 Závěr

Úkolem této práce bylo seznámení se s činností na samostatném projektu od konkrétního zadání až po technické řešení. Pro určení cesty, po které se projekt bude ubírat, byly brány v potaz tři hlediska, a to: technická náročnost, finanční náročnost a náročnost v případě potenciálních uživatelů.

Hlavním kritériem byla vždy technická náročnost, kterou jsem se snažil udržet co nejnižší, byť v případě napájecího zdroje se jednalo i o ekonomicky méně výhodnou variantu.

To na úvod k závěru a nyní bych překročil k samotnému hodnocení práce. Problematika komunikace pomocí ethernetového rozhraní je poměrně náročná a v určeném časovém horizontu se mi bohužel nepodařilo dosáhnout pozitivního výsledku. Nicméně práce vytyčila směr, kterým by se dalo kýženého výsledku (tj. zrcadla na rozhraní IP/ethernet dosáhnout). Respektive nedosáhnout, jelikož bych nyní pravděpodobně zkusil volbu čistě hardwarového řešení pomocí posuvných registrů.

Budu-li hodnotit projekt z finančního hlediska, pak si myslím, že volba byla ekonomicky výhodná a předpokládané cenové relace byly dodrženy. Pro mnou zvolenou realizaci byla napočítána na cca 1300 Kč, oproti předpokládané relaci cca 2200 Kč. Bohužel do této částky by bylo třeba započítat ještě případný prostředek pro monitorování chování komunikačního mikrokontroléru ENC28J60 (pravděpodobně by se jednalo o nějaký RS232/ USB převodník pro komunikaci s mikropočítačem AtMega16 pomocí terminálu).

Práce byla poměrně obsáhlá z hlediska množství zahrnutých informací, která jsem byl nucen načerpat. To pravděpodobně vedlo také k tomu, že se mi práci nepodařilo dotáhnout do úspěšného konce v časovém horizontu odpovídajícím bakalářské práci.

8 Seznam zkratek a symbolů

°C	Stupeň Celsia
A	Ampér
b	bit
B	byte
b/s	bit za sekundu
CRC	Cycle Redundancy Check
CSMA/CD	Sense Multiple Access with Collision Detection
DMA	Direct Access Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
F	Farad
FIFO	First In First Out
Hz	Hertz
IP	Internet Protocol
ISO/OSI	International Organization for Standardation/ Open Systems Interconnection
LED	Light Emission Diode
LLC	Logical Link Control
MAC	Media Access Control
MCU	Master Control Unit
PC	Personal Computer
PDIP	Plastic Dual Inline Package
PHY	Rozhraní fyzické vrstvy
PLL	Phase Locked Loop
PoE	Power over Ethernet
RX	Receive
s	sekunda
SOIC	Small Outline Integrated Circuit
SPI	Seriál Pheripheral Interface
SRAM	Static Random Access Memory
TX	Transmit
USB	Univerzal Serial Bus
V	Volt
Ω	Ohm

9 Použitá literatura

- [1] BOHÁČ, Leoš. *LAN-CSMA*. Praha, 2006. Dostupné z: www.comtel.cz/files/download.php?id=541
- [2] BOHÁČ, Leoš a Pavel BEZPALEC. *Datové sítě: přednášky*. 1. vyd. V Praze: České vysoké učení technické, 2011, 204 s. ISBN 978-80-01-04694-4.
- [3] IEEE 802.3. *IEEE Standard for Ethernet*. New York: IEEE Standards Association, 2012.
- [4] MICROCHIP. *ENC28J60 Datasheet: Stand-Alone Ethernet Controller with SPI Interface*. 2006. Dostupné z: http://pdf.datasheetcatalog.com/datasheets2/14/149054_1.pdf
- [5] ATMEL. *ATmega16: Microcontroller with 16K Bytes In-System Programmable Flash*. 2009. Dostupné z: <http://pdf.datasheetcatalog.com/datasheet/atmel/2466S.pdf>
- [6] ST. *Positive Voltage Regulators*. 2004. Dostupné z: http://pdf.datasheetcatalog.com/datasheet_pdf/sgs-thomson-microelectronics/L7805CD2T_to_L7885CV.pdf
- [7] ASIX. PRESTO USB In-system programmer. 23.9.2014. Dostupné z: http://www.asix.cz/download/programmers/presto_cz.pdf
- [8] SPI Bus interface. EE Herald [online]. 2006 [cit. 2014-11-22]. Dostupné z: <http://www.eeherald.com/section/design-guide/esmod12.html>
- [9] INTERNET PROTOCOL. 1981. Dostupné z: <https://www.rfc-editor.org/rfc/rfc791.txt>
- [10] INTERNET CONTROL MESSAGE PROTOCOL. 1981. Dostupné z: <https://tools.ietf.org/html/rfc792>
- [11] An Ethernet Address Resolution Protocol. 1982. Dostupné z: <http://tools.ietf.org/html/rfc826>
- [12] BASCOM-AVR. BASCOM-AVR [online]. Version 2.0.7.7 document build 52. 2014 [cit. 2014-12-04]. Dostupné z: <http://avrhelp.mcselec.com/index.html>
- [13] FONSECA, Rodrigo. CSCI-1680: Link Layer Reliability. 2014. Dostupné z: <http://cs.brown.edu/courses/csci1680/s12/lectures/04-link.pdf>

10 Seznam Obrázků

Obr. 1.1 Referenční modle ISO/OSI.....	8
Obr. 2.1 Stavový diagram CSMA/CD.....	12
Obr. 2.2 Struktura Ethernetového rámce	13
Obr. 2.3 Průběh linkového kódu Manchester	13
Obr. 2.4 Formát adresy	14
Obr. 3.1 Stavový diagram Ethernet reflektoru.....	17
Obr. 3.2 Frekvenční spektra spínaného a klasického zdroje	21
Obr. 4.1 Rozložení pinů ENC28J60, pouzdro SOIC	25
Obr. 4.2 Schéma připojení ENC28J60 na 5V logiku MCU	26
Obr. 4.3 Schéma připojení krystalu k ENC28J60.....	27
Obr. 4.4 Možné připojení LED diody k pinu LEDB	28
Obr. 4.5 Organizace paměti ENC28J60	29
Obr. 4.6 Organizace kontrolních registrů ENC28J60.....	29
Obr. 4.7 Registr ECON1	30
Obr. 4.8 Registr ECON2.....	31
Obr. 4.9 Organizace Ethernetového Bufferu ENC28J60.....	32
Obr. 4.10 Vstupní časování ENC28J60	34
Obr. 4.11 Výstupní časování ENC28J60.....	34
Obr. 4.12 Čtení kontrolního registru skupiny ETH	34
Obr. 4.13 Čtení kontrolního registru skupiny MAC a MII.....	35
Obr. 4.14 Graf závislosti odebíraného proudu na taktovacím kmitočtu a napětí	36
Obr. 4.15 Rozložení pinů ATmega16, pouzdro PDIP	36
Obr. 4.16 Doporučené připojení krystalu k ATmega16	37
Obr. 4.17 Doporučené schéma zapojení ENC28J60	38
Obr. 4.18 Elektrické schéma napájecího zdroje	40
Obr. 4.19 Model dvoucestně usměrněného napětí.....	43
Obr. 4.20 Elektrické schéma výrobku	45
Obr. 4.21 Zjednodušené schéma vývojového komunikačního kitu.....	46

11 Seznam tabulek

Tabulka 1: Maximální přípustné proudy a napětí technologie PoE	19
Tabulka 2: Napájecí piny mikrokontroléru ENC28J60	25
Tabulka 3: Řídící piny mikrokontroléru ENC28J60	26
Tabulka 4: Komunikační piny mikrokontroléru ENC28J60	27
Tabulka 5: Instrukční sada SPI rozhraní kontroléru ENC28J60	33
Tabulka 6: Popis pinů ATmega16	37
Tabulka 7: Ceny součástek použitých pro výrobu vývojového kitu s ENC28J60 ..	39
Tabulka 8: Ceny součástek použité pro vývojový kit s ATmega16	39
Tabulka 9: Parametry zvoleného stabilizátoru IC1	41

12 Seznam rovnic

$$PadSize = [0, minFrameSize - (clientsDataSize + 2AddressSize + 48)] \quad [bit] \quad (1)15$$

$$\frac{D(x) \cdot x^{31}}{G(x)} = R(x) + r(x) \quad (2) \dots\dots\dots 16$$

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \quad (3) \dots\dots\dots 16$$

$$t_z = Nt_h + Mt_{pr} + t_{km} \quad (4) \dots\dots\dots 17$$

$$t_z = t_{km} + t_{ks} + N \cdot t_m \quad (5) \dots\dots\dots 18$$

$$I_{max} = I_{AT} + I_{ENC} + I_r \quad (6) \dots\dots\dots 41$$

$$U_{ztr} = U_{IC1} + 3U_D \quad (7) \dots\dots\dots 41$$

$$S_{trans} = U_{sek} \cdot I_{max} \quad (8) \dots\dots\dots 42$$

$$u(t) = U_0 |\sin(\omega t)| \quad (9) \dots\dots\dots 43$$

$$t = \frac{1}{2\pi f} \arccos\left(\frac{u(t)}{U_0}\right) + \frac{T}{2} \quad (10) \dots\dots\dots 43$$

$$t_{vyb} = t - \frac{T}{4} \quad (11) \dots\dots\dots 43$$

$$u_c(t) = U_0 e^{-\frac{t}{\tau}} \quad (12) \dots\dots\dots 43$$

$$R = \frac{\Delta U}{I} \quad (13) \dots\dots\dots 44$$

$$C = \frac{-t}{R} \frac{1}{Ln\left(\frac{u_c(t)}{U_0}\right)} \quad (14) \dots\dots\dots 44$$

13 Seznam příloh

- 1 Katalogové list vybraných použitých elektronických součástek
- 2 Prorgam (soubor .bas, .hex)
- 3 Schéma použitého vývojového kitu
- 4 Návrhy desek plošných spojů
- 5 Nákupní rozpiska součástek

