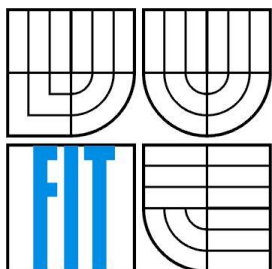


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# PŘEVODNÍK ETHERNET NA RS-232

ETHERNET TO RS-232 CONVERTER

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JIŘÍ DREISEITEL

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. JOSEF HÁJEK

BRNO 2012

## Zadání bakalářské práce

Řešitel: **Dreiseitel Jiří**  
Obor: Informační technologie  
Téma: **Převodník Ethernet na RS-232**  
**Ethernet to RS-232 Converter**

Kategorie: Vestavěné systémy

Pokyny:

1. Prostudujte problematiku vestavěných systémů a zaměřte se na možnosti síťové komunikace s využitím rozhraní Ethernet. Seznamte se standardem RS-232 a k čemu se využívá.
2. Seznamte se s hardwarovým vývojovým kitem (zapůjčí vedoucí práce), na kterém budete aplikaci vyvíjet.
3. Navrhněte elektrické schéma zapojení převodníku, který bude obousměrně převádět data mezi rozhraními Ethernet a RS-232.
4. Na základě bodu č. 3 navrhněte v systému Eagle nebo KiCAD desku plošných spojů a následně ji implementujte.
5. Implementujte firmware tak, že převodník bude schopný přijímat data přes protokol TCP/IP a vysílat je na sériovou linku RS-232. Převodník musí podporovat obousměrnou komunikaci.
6. Proveďte praktickou demonstraci na zařízení podporující RS-232 (např. PC se sériovým portem).

Literatura:

- HW Virtual Serial Port (<http://www.hw-group.com>)

Při obhajobě semestrální části projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hájek Josef, Ing.**, UITS FIT VUT  
Datum zadání: 1. listopadu 2011  
Datum odevzdání: 16. května 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav inteligentních systémů  
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

## **Abstrakt**

Práce je věnována problematice konstrukce převodníku Ethernet na RS-232 za pomoci jednočipového mikrokontroléru. Cílem je seznámit čtenáře se sítíovou technologií Ethernet a technologií pro sériový přenos založený na protokolu RS-232 a zároveň s technologií vestavěných systémů pro konstrukci zařízení.

Součástí práce je kompletní návrh převodníku Ethernet na RS-232 včetně návrhu a implementace firmware v jazyce C za využití LwIP TCP/IP stacku. Převodník je postaven na základě vývojového kitu STM3210C-eval, osazeného mikrokontrolerem rodiny STM32. V závěru práce je popsáno testování převodníku v modelových situacích a doporučené používání.

## **Abstract**

The thesis deals with construction of a convertor between Ethernet a RS-232 serial interface using a single-chip microcontroller. The aim is to make the reader acquainted with the technology of an Ethernet network and serial transfer based on RS-232 protocol. The embedded systems for the convertor construction are also explained.

The thesis contains a complete design of Ethernet to RS-232 convertor including firmware analysis and implementation in C language using LwIP TCP/IP stack. The convertor has been built on STM3210C-eval development kit, equipped with STM32 family microcontroller. The conclusion describes testing in model situations and recommended usage.

## **Klíčová slova**

Ethernet, RS-232, vestavěné systémy, STM32

## **Keywords**

Ethernet, RS-232, embedded systems, STM32

## **Citace**

Dreiseitel Jiří: Převodník Ethernet na RS-232, bakalářská práce, Brno, FIT VUT v Brně, 2012

# Převodník Ethernet na RS-232

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Josefa Hájka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jiří Dreiseitel

16.5.2012

## Poděkování

Děkuji za vedení a odbornou pomoc Ing. Hájkovi, při návrhu a konstrukci převodníku Ethernet na RS-232.

© Jiří Dreiseitel, 2012

*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

Obsah.....	1
1 Úvod.....	3
1.1 Seznámení s problematikou.....	3
1.2 Cíl práce.....	3
2 Rozbor zadání .....	4
2.1 Technologie Ethernet.....	4
2.1.1 Historie Ethernetu.....	4
2.1.2 Vznik standardu Ethernet.....	4
2.1.3 Princip činnosti .....	5
2.1.4 Ethernet z pohledu hardware .....	7
2.1.5 Přenosová média a technologie sítí.....	10
2.1.6 Shrnutí.....	14
2.2 Protokol RS-232 .....	14
2.2.1 Sériová komunikace.....	14
2.2.2 Rozhraní RS-232 .....	15
2.3 Síťové architektury .....	18
2.3.1 Referenční model OSI .....	19
2.3.2 Architektura TCP/IP .....	20
3 Návrh řešení .....	23
3.1 Mikrokontrolér STM32 .....	23
3.1.1 MCU STM32F107VCT6.....	23
3.1.2 Ethernet 10/100 MAC vrstva.....	24
3.2 TCP/IP Stack .....	25
3.2.1 LwIP stack .....	25
3.3 Vývojové prostředí .....	26
4 Návrh a konstrukce převodníku .....	27
4.1 Popis elektronického schéma zapojení .....	27
4.1.1 Napájecí zdroj.....	27
4.1.2 Napájení MCU.....	28
4.1.3 Resetovací tlačítko a bootování MCU .....	29
4.1.4 Oscilátory.....	30
4.1.5 Připojení PHY a konektoru RJ-45 .....	31
4.1.6 Uživatelské tlačítko a LED diody .....	31
4.1.7 Programovací JTAG konektor .....	32

4.1.8	RS-232 rozhraní.....	33
4.1.9	CAN rozhraní.....	33
4.2	Návrh desky plošných spojů.....	34
4.2.1	Deska plošných spojů .....	34
4.2.2	Rozmístění součástek na DPS.....	34
4.2.3	Nastavení zkratovacích propojek (jumper).....	35
5	Implementace firmware převodníku .....	36
5.1	Návrh firmware.....	36
5.1.1	Princip činnosti převodníku .....	37
5.2	Implementace návrhu firmware .....	37
5.2.1	Nastavení LwIP stacku pro firmware .....	37
5.2.2	Konfigurace MCU .....	38
5.2.3	Inicializace činnosti převodníku .....	38
5.2.4	Přerušovací systém .....	39
5.2.5	Hlavní program.....	39
5.2.6	Provádění programu.....	40
6	Převodník a jeho vlastnosti .....	42
6.1	Parametry pro datový přenos .....	42
6.2	Zprovoznění převodníku.....	42
6.3	Demonstrace činnosti.....	43
6.4	Elektrické vlastnosti převodníku .....	46
6.5	Návrh na další rozšíření funkčnosti .....	46
7	Závěr .....	47
	Seznam příloh .....	50

# 1 Úvod

## 1.1 Seznámení s problematikou

V dnešní době jsme obklopeni doslova nepřehledným množstvím standardů, norem, technologií pro přenos dat. S většinou se setkáváme v běžném každodenním životě, s jinými se shledáme už pouze v průmyslové výrobě. K těmto příkladům můžeme přímo zahrnout technologie vystavěné nad standardem *Ethernet*, nebo standard pro sériový přenos *RS-232*, který se pro svou jednoduchost, univerzálnost a zpětnou kompatibilitu stále často využívá v průmyslové výrobě.

Rozsáhlé rozšíření těchto dvou technologií přímo vybízí k tomu, aby vzniklo zařízení, které bude schopné tyto dvě technologie spojit a využít jejich předností. Takovéto zařízení může mít velký potenciál k využití, což nasvědčují příklady z praxe, ve kterých se převodník využívá pro mnoho aplikací, pro představu některé uvádím v následujícím seznamu:

- Vzdálená komunikace mnoha snímačů (například elektronických teploměrů) a PC prostřednictvím LAN.
- Dálkové řízení průmyslových strojů.
- Bezpečnostní systémy.
- Monitorovací a lékařské systémy ve zdravotnictví.
- Připojení PDA, tiskárny nebo mobilních telefonů do LAN.

Z uvedených příkladů lze vyvodit, že takovéto převodníku zasahuje skutečně do všech sektorů od soukromého až po průmyslový.

## 1.2 Cíl práce

Cílem této práce je čtenáře seznámit s technologií Ethernet pro budování LAN sítí a dále standardem pro sériový přenos RS-232. V textu se napřed seznámíme s problematikou datových přenosů a v dalších částech přejdeme přímo k návrhu a konstrukci převodníku Ethernet na RS-232, tak abychom splnili všechny body zadání.

Výsledkem práce by měl, být funkční převodník Ethernet na RS-232 včetně implementovaného firmware.

## 2 Rozbor zadání

V následujícím textu, bych vás rád seznámil s teorií týkající se sítě Ethernet a sériové linky RS-232. Pro pochopení problematiky je nutné dozvědět se i něco o současné technologii, ale také i historii a na závěr kapitoly se ještě seznámíme se síťovými architekturami OSI a TCP/IP.

### 2.1 Technologie Ethernet

Ethernet je v dnešní době nejpoužívanější síťovou technologií pro konstrukci domácích, ale i průmyslových počítačových sítí. Velký podíl na jeho rozšíření mají především tyto faktory jako nízké náklady, široké uplatnění, spolehlivost a jednoduché nástroje pro správu. V následující kapitole se budeme věnovat historii vzniku Ethernetu, principu jeho činnosti, vlastnostmi a nakonec jeho významem v referenčním modelu ISO/OSI. Podkladem pro souhrnný rozbor byla kniha Charlese Spurgeon vydána v roce 2000 [1].

#### 2.1.1 Historie Ethernetu

Historie vzniku Ethernetu se začala psát v roce 1972 ve výzkumném centru *PARC* v Kalifornii, za formálního otce Ethernetu můžeme považovat amerického inženýra Boba Metcalfa. Ten pracoval na projektu *ALOHA*<sup>1</sup>, který využíval velmi jednoduchý mechanismus pro datovou komunikaci. Vysílající stanice poslala zprávu a čekala na doručení potvrzení, že zpráva byla úspěšně přijata, pokud se tak nestalo do určité doby, došlo k opětovnému vyslání zprávy. Tento přístup zdaleka není schopen, využít maximální šířku kanálu, kvůli velkému růstu kolizí již při vytížení 18% kapacity linky jak uvádí Spurgeon ve své knize [1].

Metcalfe přišel na to, že tento systém lze vylepšit jednoduchým způsobem a to tak, že dojde k řízení komunikace na sdíleném kanálu, protokol, který představil, se nazývá **CSMA/CD**. Použití CSMA/CD v kombinaci s použitím zpětných sofistikovaných algoritmů navyšuje využití Ethernet sítě na 100% její kapacity. Podrobněji popsán v kapitole 2.1.3.2.

#### 2.1.2 Vznik standardu Ethernet

K první veřejné publikaci oficiálního standardu pro 10Mbps Ethernet došlo v roce 1980. Tento standard přešel ve známost jako DIX Ethernet a obsahoval specifikaci pro provoz sítě Ethernet postavený nad technologií 10Base5. V roce 1983 vydala společnost IEEE<sup>2</sup> vlastní verzi standardu pro Ethernet, která rozšiřovala původní verzi, pod názvem *802.3 CSMA/CD*. Jak vychází z názvu, tak

<sup>1</sup> Experimentální síť pro komunikaci mezi havajskými ostrovy pomocí rádiového přenosu.

<sup>2</sup> Institute of Electrical and Electronics Engineers, mezinárodní nezisková profesní organizace se sídlem v USA.



je zde kladen důraz právě na přístupovou technologii CSMA/CD a přestože se v názvu standardu nevyskytuje slovo Ethernet, tak se neoficiálně od vzniku až do dnešní doby tyto standardy řady IEEE 802.3 označují právě tímto názvem. Základní verze byla díky vzniku nových technologií, postupem času doplňována a dále rozšiřována až do dnešní doby.

### 2.1.3 Princip činnosti

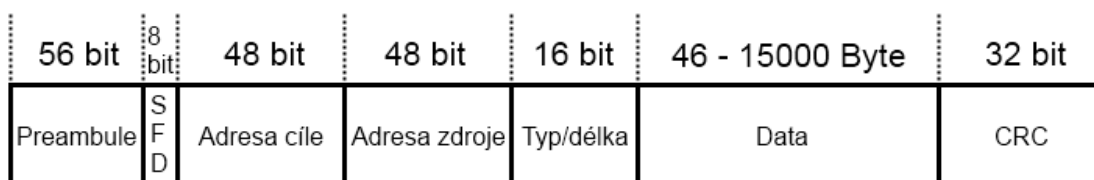
Ke správné činnosti LAN<sup>1</sup> sítí, které jsou postavené nad technologií Ethernet, potřebujeme, aby hardware a software spolupracovali, tak bude zajištěn korektní přesun digitálních dat mezi dvěma zařízeními připojených do této sítě.

Sítě Ethernet jsou vystavěny nad těmito čtyřmi základními prvky:

- **Datový rámeček** – předpis, který určuje podobu dat při přenosu po síti.
- **Protokol řídicí přístup k médium** – soubor pravidel implementovaných v rozhraní sítě Ethernet, zajišťující spravedlivý přístup na sdílenou linku.
- **Signalizační prvky** – standardizovaná elektronická zařízení po příjem a odesílání dat.
- **Přenosové fyzické médium** – fyzické přenosové médium jako kabel a další prvky zajišťující přenos digitálního signálu napříč sítí.

#### 2.1.3.1 Ethernetový datový rámeček

Datový rámeček Etherentu představuje jádro systému, jedná se jednoduché datové pole s proměnnou délkou, viz následující obrázek 2.1.



Obrázek 2.1: Ethernetový rámeček podle IEEE 802.3.

#### Význam jednotlivých polí rámečku

- **Preamble** – skládá se ze 7 oktétů, dochází ke střídání binárních 0 a 1, využívá se k synchronizaci hodin příjemce dat.
- **SFD<sup>2</sup>** – speciální oktét určující začátek rámečku, jeho podoba je 10101011.
- **Adresa cíle** – fyzická MAC adresa cílového zařízení.
- **Adresa zdroje** – fyzická MAC adresa zařízení identifikující vysílací zařízení.
- **Typ / délka** – pro IEEE802.3 udává délku pole dat, u standardu Ethernet II udával typ vyššího protokolu.

<sup>1</sup> LAN – Local Area Network, v překladu *místní síť* označuje počítačovou síť zahrnující malé geografické území.

<sup>2</sup> SFD – Start of Frame delimiter.

- **Data** – pole o proměnné délce minimálně 46 až 15000 Bytů.
- **CRC** – pole pro kontrolní 32bitový kontrolní součet, používá se ke kontrole správnosti dat, jindy označován jako FCS (Frame Check Sequence) pole.

### 2.1.3.2 CSMA/CD protokol

O tomto protokolu jsme se již zmínili v kapitole 2.1.1, právě tento protokol stál za úspěšností a rozšíření standardu Ethernet v jeho počátcích. Nyní si vysvětlíme princip jeho funkčnosti.

Základní vlastnosti protokolu vychází z jeho anglického názvu *Carrier Sense Multiple Access with Collision Detect*.

- **Carrier Sense** – znamená, že před začátkem každého vysílání musí zařízení vyčkat, doku sdílená linka není uvolněná.
- **Sense Multiple Access** – znamená, že všechna zařízení mají při přístupu ke sdílené lince kdykoliv potřebují a zároveň mají stejnou prioritu, ve chvíli kdy chtějí začít vysílat.
- **Collision Detect** – vzhledem k tomu, že všechna zařízení mají při přístupu ke sdílené lince stejnou prioritu, tak může nastat situace, kdy dojde ke kolizi dvou zařízení. Jakmile nastane taková situace tak do režie zasáhne zařízení, které kolizi rozpoznalo, rozešle všem rozhraním připojeným to Etherentu signál k tomu, aby zastavili své vysílání. Následně bude každé zařízení v náhodném čase vyžádáno, aby přeposlalo své data, tato operace se označuje anglickým termínem *backoff*.

Protokol CSMA/CD je tedy navržen tak, aby ke sdílenému kanálu existoval spravedlivý přístup pro všechny stanice, a zároveň zabraňuje uvážnutí jakékoliv stanice z důvodu zabránění linky jinou stanicí. Podrobnější informace o protokolu lze získat v již zmíněné knize [1], přesně na stranách 28 až 30.

### 2.1.3.3 Řízení přístupu ke sdílenému médiu

Pro řízení přístupu se v první verzi Ethernetu používá protokol s anglickým názvem *MAC protocol* (*media access control protocol*), který je právě vystavěn na základě CSMA/CD. Komunikace pod tímto protokolem probíhala v *half-duplex*<sup>1</sup> režimu a umožňovala spravedlivý přístup všech stanic ke sdílené lince. Pravidla protokolu určují chování stanic v síti Ethernet a včetně toho přesně stanovují, kdy může dojít ke komunikaci po síti a co dělat v případě, kdy nastane kolize. Centrální řízení komunikace není třeba díky dodržování *MAC protokolu* všemi zařízeními v síti.

---

<sup>1</sup> *Half-duplex* režim je označován komunikace, kdy se v jeden čas data přenáší jen v jednom směru.

## Zjednodušený princip činnost

1. Pokud není linka obsazena (není na ní žádný přenos) a uběhl již dostatečný čas od posledního vysílání na lince, tak může zařízení zahájit přenos jednoho datového rámce. Jestliže chce stanice přenášet více rámců za sebou, tak musí mezi každým přenosem jistou dobu čekat.<sup>1</sup>
2. Pokud je linka obsazena, tak zařízení vyčkává, dokud se linka nedostane do stavu, kdy není obsazená. Stanice začne vysílat, jakmile to bude možné, pro vysílání platí stejná pravidla jak v bodě 1.
3. Pokud je během začátku přenosu rozpoznána kolize, tak zařízení dovysílá posledních 32 bitů. Jestliže ke kolizi nastane v brzkém čase od začátku přenosu, tak dojde k odvysílání kompletní preambule a následujících 32 bitů.
  - a. Po odeslání posledních bitů zařízení čeká po náhodně zvolenou dobu, poté se snaží přenos restartovat podle bodu 1. Tím, že čekací časy jsou voleny náhodně, zařízení docílí toho, že je méně pravděpodobné, že dojde opět ke kolizi s jiným signálem. Tento postup nazýváme anglickým termínem *backoff*.
  - b. Pokud dojde při novém začátku vysílání opět ke kolizi, provede se opět *backoff* proces s tím rozdílem, že náhodně zvolená doba bude vyšší než v předchozím kroku, takto se snižuje pravděpodobnost další kolize.
4. Po odeslání 512bitů dat bez kolize, do kterých se nezapočítává preambule, stanice oznámí, že kanál je obsazen. Při správné funkčnosti Ethernetu, by po tomto oznámení nemělo docházet k dalším kolizím.

Výše uvedený princip, byl používán ve starších verzích Ethernetu, v dnešní době se v praxi používá technologie založena na *full-duplex* režimu, to znamená, že zařízení mohou komunikovat obousměrně ve stejný čas. Podrobnější rozbor činnosti a časování naleznete na stranách 50 – 60 v již zmíněné literatuře [1].

## 2.1.4 Ethernet z pohledu hardware

Ve výše uvedených kapitolách jsme se zabývali „softwarovou“ částí standardu Ethernet, nyní si zjednodušeně představíme standard Ethernet z hardwarového pohledu. V kapitole 2.1.3 byly zmíněny čtyři základní pilíře, na kterých je Ethernet založen, načež s hardwarem si můžeme spojit signalizační zařízení a přenosové médium. Použití různých elektronických komponent nebo přenosového média přímo souvisí s rychlostí, jako se budou data po síti přenášet.

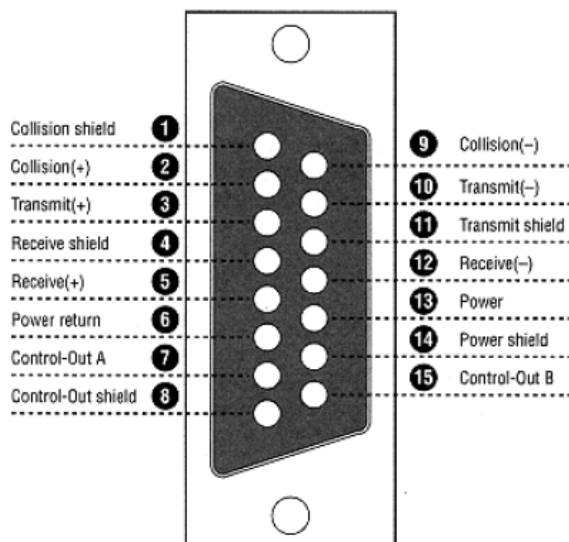
### 2.1.4.1 Attachment Unit Interface

Jedná se o rozhraní, které je na výstupu síťové karty a na vstupu externího vysílače<sup>1</sup>. Pro propojení externího vysílače a síťové karty se využívá AUI kabel. AUI je fyzicky realizován jako 15pinový

---

<sup>1</sup> Tato krátká časová prodleva mezi jednotlivými přenosy po lince se nazývá *interframe gap* (IFG.)

konektor typu CANON. Popis jednotlivých pinů je na obrázku 2.2. Při použití AUI rozhraní a externího vysílače můžeme docílit toho, že zařízení může být absolutně nezávislé na přenosovém médiu, které je použito pro budování sítě. Kompatibilitu právě zajišťuje vysílač.



Obrázek 2.2: AUI konektor.

### AUI kabel

Propojuje 10Mbit Ethernet zařízení a externí transiever, zajišťuje přenos těchto tří signálů:

- Vysílání dat z Ethernet zařízení směrem k externímu vysílači.
- Příjem dat od externího vysílače směrem k Ethernet zařízení.
- Signál signalizující kolizi, vysílaný z externího vysílače směrem k Ethernet zařízení.

Podle standardu IEEE má tloušťku cca 1cm a jeho maximální délka může být 50m, je tvořen dílčí symetrickou dvojicí vodičů. Obvykle je kabel opatřen na jedné straně AUI konektorem typu samec a na druhé typu samice.

### AUI Externí vysílač<sup>2</sup>

Jedná se o specializovanou jednotku, která zajišťuje spojovací vrstvu mezi Ethernet zařízením a systémem nad kterým je postavena Ethernet síť. Bývá konstruován jako malá krabička, jejíž hlavní funkcí je převádět signály, které mají různé elektronické vlastnosti. Na vstupu má AUI konektor, přičemž na výstupu může být jakékoliv 10Mbit přenosové médium, koaxiální kabel, křížená dvojlinka nebo optické vlákno.

<sup>1</sup> Anglickým termínem je tento vysílač označován jako *transiever*.

<sup>2</sup> Anglickým termínem označován jako Medium Attachment Unit.

### 2.1.4.2 Medium-Independent Interface (MII)

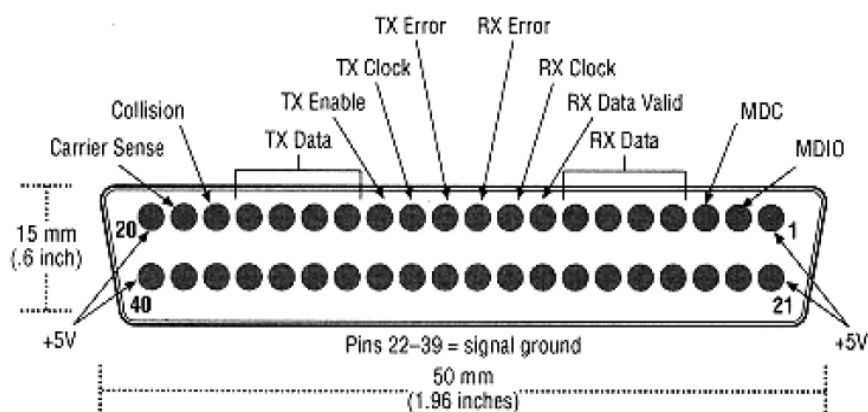
Jelikož AUI bylo navrženo pouze pro zařízení s rychlostí do 10Mb/s, proto bylo vyvinuto nové rozhraní, které je schopno pracovat na obou rychlostech 100Mb/s i 10Mb/s. Jedním z hlavních rozdílů oproti předchozí AUI bylo to, že místo externího vysílače je zde *physical layer device* (PHY), ve skutečnosti MII rozšiřuje původní AUI.

MII může být přímo vestavěné společně i s PHY na zařízení Ethernet, s takovým přístupem se v dnešní době setkáváme nejčastěji, ale také může být, stejně jak u AUI PHY použito jako externí transiever, toto řešení vyžaduje MII konektor, na která se externí PHY připojuje.

#### MII konektor a jeho signály

Tento 40pinový konektor typu CANON viz obrázek 2.3 z knihy [1], zajišťuje propojení mezi MII rozhraním a externím PHY vysílačem, ve většině případů je externí PHY přímo přizpůsobeno k připojení na MII, aniž by bylo potřeba použít MII kabel.

- *Management data I/O (MDIO)* – signál pro přenos kontrolních a řídicích informací mezi zařízeními.
- *Management data clock* – signál pro přenos času, používajícího se jako reference pro sériové přenosy na MDIO.
- *TX Data* – signály pro přenos dat ve směru od Ethernet zařízení k PHY.
- *TX Clock* – hodinový čas pro vysílání na TX, pro 100Mb/s režim 25MHz a pro 10Mb/s režim 2,5MHz. Slouží k synchronizaci při přenosu dat.
- *TX Error* – signál může využívat opakovač, pro šíření chyby, lze ho použít jen za určitých podmínek, stanice Ethernet ho nikdy nevyužívá.
- *TX Enable* – přenáší signál od Ethernet rozhraní k PHY, signalizuje povolení přenosu.



Obrázek 2.3: Konektor MII.

- *RX Data* – signály pro přenos dat ve směru od PHY k rozhraní Ethernet.
- *RX Clock* – hodinový čas pro vysílání na RX, pro 100Mb/s režim 25MHz a pro 10Mb/s režim 2,5MHz. Slouží k synchronizaci při přenosu dat.

- *RX Error* – signál vyslaný vysílačem, jako detekce přijaté chyby.
- *RX Data Valid* – signál vyslaný vysílačem, při přijetí validního rámce dat.
- *Collision* – signál pro detekci kolize na síti, pro full-duplex režim není přesně definován.
- *Carrier Sense* – indikuje síťovou aktivitu mezi Ethernet rozhraním a PHY.

#### 2.1.4.3 Gigabit Medium-Independent Interface (GMII)

Jak lze vyčíst z názvu, jedná se o nadstavbu MII. Hlavní rozdíl je taktéž vidět přímo v názvu, GMII totiž umožňuje přenos dat o rychlosti 1Gb/s. Další velký rozdíl je v tom, že při takto vysoké rychlosti již není možné ke GMII externě připojovat transiever. Jako nejčastější výstupní port se používá 8pinový RJ-45 port a kroucená dvojlinka. Pro další informace o GMII doporučuji pročíst literaturu, například [1].

### 2.1.5 Přenosová média a technologie sítí

Druh přenosového média, které je použito pro výstavbu Ethernet sítě, přímo souvisí s vlastnostmi této sítě, ty jsou vypsány v následujícím seznamu:

- Přenosová rychlost dat.
- Maximální vzdálenost mezi jednotlivými zařízeními.
- Počet zařízení připojených na jednu linku.
- Elektronické vlastnosti sítě.

#### 2.1.5.1 Koaxiální kabel

Koaxiální kabel se používal v počátcích sítě Ethernet, byly na něm postaveny sítě označované jako 10Base5 a 10Base2, topologie sítě byla doslova ušita na míru koaxiálnímu kabelu.

##### Ethernet 10Base5

Koaxiální kabel v této síti byl často nazýván *tlustý koaxiální kabel*, takovéto přívlastko vychází ze specifikace kabelu, protože jeho průměr je přibližně 10mm, impedance kabelu je podle normy 50Ω.

Tlustý koaxiální kabel má velmi dobré vlastnosti proti rušení a to díky čtyřnásobnému opletení, tento vliv se projevuje na rychlosti přenosu signálu, ale také na tuhosti kabelu. Minimální vzdálenost mezi transievery je 2,5m, ty lze připojit speciálním konektorem, kde jeden kontakt se zařeže až do středového vodiče a druhý kontakt do opletení.

Síť 10Base5 má sběrnicovou technologii, proto taky přenos dat probíhá v režimu half-duplex. Maximální délka kabelu je 500m a musí být na obou koncích ukončen terminálním rezistorem o stejné impedanci 50Ω, toto řešení zabraňuje odrazům signálu na konci kabelu, takovéto odrazy by

jinak rušili díky interferenci další signály. Takto konstruovaná sběrnice se nazývá segment a lze do ní připojit až 100 stanic.

Pro propojování jednotlivých segmentů slouží aktivní prvek *opakovač* (angl. *repeater*), takto propojené segmenty tvoří jednu sběrnici a signál vyslaný z jedné stanice se šíří po všech propojených segmentech. Segmenty ovšem nelze propojovat do nekonečna, existují zde jistá omezení, a to že stanice mohou mezi sebou opakovat maximálně přes čtyři opakovače a zároveň jen tři segmenty mohou být obsazené stanicemi. Pokud dodržíme všechny normy pro konstrukci sítě, tak mohou stanice komunikovat maximálně na vzdálenost 2800m.

## **Ethernet 10Base2**

Tato technologie je postavena na koaxiálním kabelu, který je o polovinu tenčí než kabel použitý v 10Base5. Jeho tloušťka je přibližně 5mm a impedance je taktéž 50Ω. Výhoda je v tom, že je levnější a má lepší mechanické vlastnosti, zhoršené elektronické vlastnosti neznamenají zase takový problém.

Síť 10Base2 je také založena na sběrnice technologii avšak vlastnosti má jiné. Maximální délka segmentu může být 185m a lze na něj připojit 30 stanic, kde minimální rozestupy musí být 0,5m. Platí zde stejné pravidlo pro maximální počet opakovačů mezi stanicemi jako u 10Base5 sítě, s tím že maximální vzdálenost mezi stanicemi může být 1555m.

Pro připojení jednotlivých stanic lze využít konektor BNC-T, ten má nevýhodu v tom, že je třeba vést kabel ke každé stanici a může nastat situace, že celý segment přestane fungovat při náhodném rozpojení nebo chybě jiného druhu. Tuto nevýhodu řeší použití EAD kabelů a EAD zásuvek, ovšem takovéto řešení je více nákladné. Nejnákladnějším řešením je použít stejnou metodu jak u 10Base5.

### **2.1.5.2 Kroucená dvojlinka**

Technologie kroucené dvojlinky se začala požívat po roce 1990, ve chvíli kdy byl zaveden Ethernet 10Base-T. O konstrukci tohoto vedení nám již napovídá název, kroucená dvojlinka je tvořena párem dvou vodičů a právě tyto vodiče jsou po celé délce do sebe zakrouceny. Způsob tohoto vedení nazýváme *symetrické vedení* a signál takto přenášený je vyjádřen rozdílem potenciálů obou vodičů. Takovéto provedení má výhodu v tom, že dochází ochraně proti rušení, to se minimalizuje tím, že jsou vodiče do sebe zapleteny a pravidelně se zakrucují.

Kabely této konstrukce jsou rozděleny do různých kategorií. První verze vycházela z telefonního vedení a spadala do kategorie 3, na té bylo postaveno vedení sítě 10Base-T. V dnešní době se nejčastěji vyskytuje kategorie 5 nebo 5e, které se používají pro stavbu sítí 100Mb/s v případě 5e 1Gb/s. Postupem času se začínají rozšiřovat i kategorie 6 a 7, které jsou navrženy pro konstrukci sítí 10Gb/s nebo 100Gb/s. V následující tabulce 2.1 je přehled standardů, tabulka je převzata z [2] strana 12.

název kabelu	standard	označení	rychlost přenosu	konektor	šířka pásma
<b>Kroucená dvojlinka</b>	100Base-T	Kategorie 5	100 Mb/s	RJ – 45	100 MHz
<b>Kroucená dvojlinka</b>	1000Base-T	Kategorie 5e	1000 Mb/s	RJ – 45	125 MHz
<b>Kroucená dvojlinka</b>	1000Base-TX	Kategorie 6	1000 Mb/s	RJ – 45	250 MHz
<b>Kroucená dvojlinka</b>	1000Base-TX2	Kategorie 7	1000 Mb/s	GC45, TERA	600 MHz

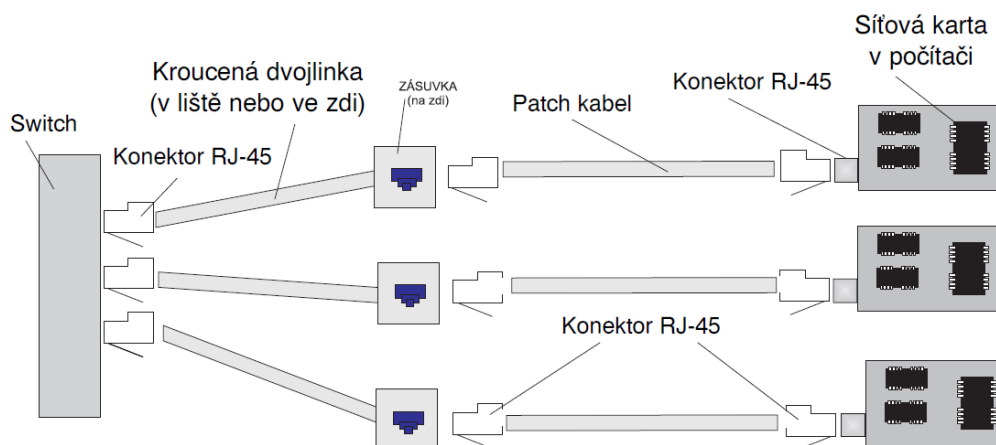
Tabulka 2.1: Vlastnosti křížené dvojlinky.

## UTP a STP

Kroucené dvojlinky se vyrábí ve dvou typech, stíněná (STP) a nestíněná (UTP) dvojlinka. U stíněných kabelů dochází k lepší ochraně proti rušivým elementům. Lze stínit přímo vodiče nebo celý svazek vodičů. Větší složitost STP kabelů se samozřejmě projevuje na ceně, která je vyšší než u UTP.

## Praktické zapojení kroucené dvojlinky s využitím RJ-45

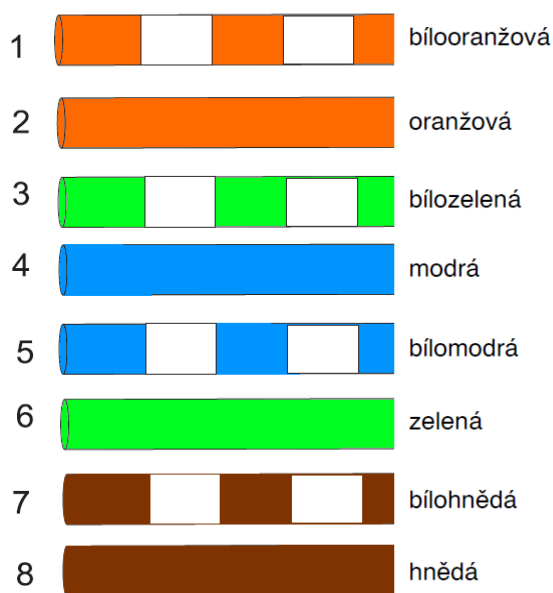
Na obrázku 2.4 převzatého z [2] lze vidět zjednodušený princip kabeláže při použití kroucené dvojlinky. Vidíme, že koncepce se od použití koaxiálního kabelu liší, jelikož nelze kabely fyzicky spojovat, je třeba použít aktivní prvek anglicky označován jako *switch*, česky *přepínač*. Switch provádí programové přepínání mezi jednotlivými linkami a zajišťuje, že komunikace může probíhat mezi kterýmkoliv stanicemi připojených do sítě, dalo by se říct, že nahrazuje sběrnici, která je použita u technologií s koaxiálním kabelem.



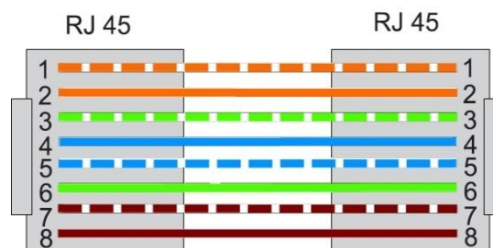
Obrázek 2.4: Příklad kabeláže při použití kroucené dvojlinky.

Na kroucenou dvojlinku se připojuje standardizovaný konektor RJ-45 s osmi piny, připojení jednotlivých vodičů se řídí normou. Příklady zapojení jsou uvedeny na obrázku 2.6. Jako *patch kabel* nazýváme kroucenou dvojlinku zakončenou na obou koncích konektorem RJ-45. Jak již bylo zmíněno, kroucená dvojlinka se skládá z osmi vodičů a pokud budujeme síť s rychlostí 10Mb/s nebo 100Mb/s tak nám postačí pouze dva páry vodičů, zbylé dva páry jsou nevyužity, nebo se často

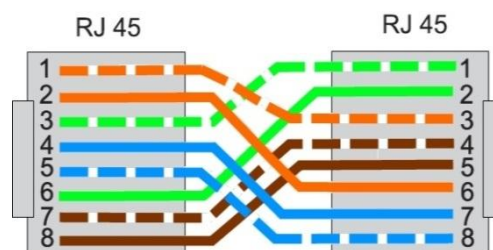




Obrázek 2.6: Barevné zapojení podle TIA/EIA 568-B.



Obrázek 2.5: Přímý kabel.



Obrázek 2.7: Křížený kabel.

používají pro napájení síťového prvku<sup>1</sup>, u varianty 1Gb/s dochází k využití všech čtyř párů vodičů. Pro propojování stanice se switchem se využívá přímého kabelu, pokud propojujeme dvě stanice mezi sebou (například dva počítače), tak je potřeba použít křížený kabel, oba příklady jsou na obrázku 2.7 a 2.7: Křížený kabel.2.7.

### Ethernet 100Base-T

V dnešní době nejrozšířenější normou, vychází z normy IEEE 802.3 a stejně jak její předchůdci je založena na pravidlech Ethernetu včetně přístupu CSMA/CD. Jedná se o standard pro *Fast Ethernet*, tedy počítá s přenosovou rychlostí 100Mb/s.

Pro konstrukci se využívá především kroucené dvojlinky, kde zapojení je uvedené výše a na rozdíl od technologií s koaxiálním kabelem, využívá hvězdicovou topologii. Fast Ethernet je definován v těchto třech variantách:

- *100BASE-TX* – jako kabeláž je použita nestíněná kroucená dvojlinka kategorie 5, vyžívá dvou párů vodičů. Maximální délka segmentu může být 100m, pro kvalitnější vedení je možné použít kabel typu STP.
- *100BASE-FX* – varianta pro optické kabely, délka segmentu dosahuje až 412 metrů pro vícevidové kabely s *half-duplexem*, nebo až 10000 m pro jednovidový kabel a *full-duplex* mód.
- *100Base-T4* – nejstarší normou, která se v dnešní době už téměř nepoužívá, využívala kabelů kategorie 3 a 4. Maximální délka segmentu je stejně jak u 100Base-TX 100 m, rozdíl je v tom, že využívá všech čtyř párů vodičů.

<sup>1</sup> PoE (Power over Ethernet) napájení po datovém síťovém kabelu, bez nutnosti přivést napájecí napětí k přístroji dalším samostatným kabelem.

### 2.1.5.3 Optické vlákno

Optické vlákno je zatím poslední přenosové médium, které je použito v sítích Ethernet. Jeho výhodou je schopnost přenášet daleko větší množství dat na větší vzdálenost a vzhledem k tomu, že se nepřenáší elektronický signál tak na přenos nepůsobí žádné rušení.

Uplatňuje se především při propojování páteřních sítí. Pro výstavbu lokálních sítí není úplně vhodné kvůli své vysoké ceně a nákladům na příslušenství spojených s jeho používáním. Dělí se na dvě skupiny:

- *mnohovidové* – paprsek je během přenosu rozložen na více částí, přenos je způsoben odrazy paprsků od pláště vlákna. Má horší optické vlastnosti, ale je levnější a tedy i používanější.
- *jednovidový* – přenášen jen jeden paprsek, má lepší optické vlastnosti a také větší propustnost na delší vzdálenost.

Pro převod světelného signálu na elektronický je na každém konci umístěn *převodník*, ten bývá často zabudován až ve switchi. Dále existuje *konvertor*, který je schopen propojit kroucenou dvojlinku a optické vlákno a převádět elektrický signál na světelný a naopak.

## 2.1.6 Shrnutí

V předchozích kapitolách jsme se seznámili se základy, ale i se současnými technologiemi Ethernet, byl vysvětlen princip činnosti původního návrhu sítě a dále zařízení, která jsou nezbytně nutná pro jeho funkčnost. Při návrhu převodníku Ethernet na RS-232 můžeme mnohé znalosti využít.

## 2.2 Protokol RS-232

Protokol RS-232 patří mezi jedny nejrozšířenějších technologií pro sériový přenos dat. V následující kapitole se s tímto protokolem seznámíme podrobněji.

### 2.2.1 Sériová komunikace

V počítačovém světě se setkávám s dvěmi základními přístupy pro přenos dat, tím je přenos paralelní a přenos sériový. Vzhledem k tomu, že předmětem práce je komunikace po sériovém rozhraní RS-232, tak se zaměříme pouze na sériovou komunikaci.

Základní principy vychází již přímo z názvu komunikace, potřebná data se přenášejí po jediném vodiči v čase za sebou, zatímco u paralelního přenosu, jsou data v jeden čas vysílána po více vodičích.

## Synchronní a asynchronní komunikace

Pokud jsou data přenášena sériově tak to lze dvěma způsoby: *synchronně* a *asynchronně*. Pokud se data (jednotlivé bity) přenáší po jedné lince, je potřeba jednoznačně určit, v kterém okamžiku je na datovém vodiči hodnota bitu a jeho význam.

**Synchronní sériový přenos** – společně s daty je přenášen i hodinový signál, dochází k přenosu celých bloků dat, kde začátek bloku je identifikován jedním nebo více synchronizačními znaky (nástupné a sestupné hrany signálu), tím zajistíme potřebnou synchronizaci odesílatele a příjemce dat. Datový blok je opět ukončen synchronizačními znaky. Tento přístup je rychlejší než u asynchronní komunikace (není zde zbytečná režie start a stop bitů), ale technické a programové vybavení je náročnější.

**Asynchronní sériový přenos** – společně s daty nedochází k přenosu hodinového signálu, přijímač si ho musí vygenerovat sám. To provádí pomocí vhodného příznaku, kterým je *start-bit* umožňující příjemci vhodně nastavit svůj generátor času, tak aby byl synchronizován s přijímanými daty. Datový blok bývá obvykle ukončen *paritním* bitem a *stop-bitem*.

## 2.2.2 Rozhraní RS-232

Norma RS-232-C byla navržena z důvodu kompatibility zařízení od různých výrobců a byla publikována v roce 1969 společností Electronic Industries Association. Určuje přesně elektronické vlastnosti obvodů mezi dvěma zařízeními. Definuje čísla a názvy vodičů potřebných pro jejich propojení.

### DTE a DCE

Zařízení jsou rozdělena na dva typy a to z důvodu, že na jednom vodiči zařízení vysílá a na druhém přijímá. Podle standardu RS-232-C mají zařízení DTE konektory s piny a naopak DCE zdířky, ve skutečnosti výrobci toto pravidlo ne vždy dodržují.

DTE zařízení bývá označováno jako terminál, pro vysílání používá vodič (pin) č.2 a pro příjem dat používá vodič č.3, jak lze vidět na obrázku 2.8.

DCE zařízení bývá označováno jako modem, pro vysílání používá vodič č.3 a pro příjem dat používá vodič č.2.

Pokud vím, zda se jedná o zařízení typu DTE nebo DCE také je můžeme jednoduše propojit vhodným typem kabelu. Pro propojení DTE s DCE se používá přímého kabelu, v jiných případech je potřeba použít kabel křížený.

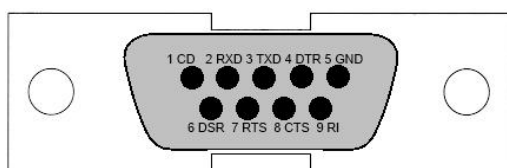
### 2.2.2.1 Konektory

Pro sériové rozhraní existuje větší množství konektorů, nejčastější je však použití konektorů typu D (svým tvarem připomíná písmeno D). Tento konektor má určitý počet pinů nebo zdířek, které jsou očíslovány. Podle Golfona v [3] patří mezi nejpoužívanější 25pinový a 9pinový konektor D.

## 9pinový konektor D pro RS-232-C

Na obrázku 2.8 je zapojení konektorů pro RS-232 podle normy, zde je popis jednotlivých signálů:

- *CD (Carrier Detect)* – detekce nosného signálu, protistrana (modem) signalizuje, že detekovala nosný kmitočet a je připravena na komunikaci.
- *RXD (Receive Data)* – přijímání data z modemu.
- *TXD (Transmit Data)* – odesílání data z terminálu.
- *DTR (Data Terminal Ready)* – signalizace protistraně (modemu), že jsme připraveni na komunikaci.
- *GND (Ground)* – Signálová zem, napětí signálů se určuje proti této zemi.
- *DSR (Data Set Ready)* – protistrana (modem) signalizuje, že je připravena komunikovat (můžeme odesílat data).
- *RTS (Request To Send)* – na tomto signálu terminál oznamuje, že komunikační cesta je volná.
- *CTS (Clear To Send)* – na tomto signálu modem oznamuje, že komunikační cesta je volná.
- *RI (Ring Indicator)* – indikátor zvonění, modem oznamuje detekci zvonění na lince (někdo požaduje datové spojení).



Obrázek 2.8: 9pinový D konektor pro RS232-C.

### 2.2.2.2 Elektrické signály

Norma RS-232-C povoluje pro datové vodiče pouze dva stavy:

- SPACE – odpovídá logické 0 a má kladné napětí.
- MARK – odpovídá logické 1 a má záporné napětí.

Hodnoty napětí jsou určeny takto:

- SPACE – pro výstup +5 až +15 volty a pro vstup mezi +3 a +15 volty.
- MARK – pro výstup -5 až -15 volty a pro vstup mezi -3 a -15 volty.

Pokud použijeme příliš dlouhé vedení, může se stát, že napěťové úrovně klesnou mimo povolené hranice, taktéž má na kvalitu signálu vliv kapacita kabelů (má vliv na ostrost přechodů mezi log. 0 a log. 1). Z výše uvedených informací lze vyvodit to, že norma není navržena pro přenos dat na větší vzdálenosti, jako maximální vzdálenost považujeme délku 15m při obvyklých rychlostech. Údaje jsou převzaty z knihy [3].

### 2.2.2.3 Handshaking

Česky řečeno *řízení toku dat*, představuje na hardwarové nebo softwarové úrovni rozhraní potvrzení přijatých dat nebo připravenost zařízení k zahájení přenosu dat. Oba dva přístupy využívají signály přicházející z přijímacího zařízení do zařízení vysílacího.

- *Hardwarový handshaking* – přijímací zařízení vysílá, kladné napětí ve chvíli kdy je připraveno přijímat data, přičemž využívá příslušný vodič pro řízení toku dat. Pokud vysílací počítač přijme záporné napětí, zastaví vysílání dat.
- *Softwarový handshaking* – signály pro řízení toku dat se skládají ze speciálních znaků a jsou vysílány pouze po datových vodičích, nevyužívá se tedy specializovaných vodičů a řízení toku dat je řešeno programem.

### 2.2.2.4 Asynchronní komunikace podrobněji

Asynchronní sériovou komunikaci jsme si představili už v kapitole 2.2.1, vzhledem k tomu, že RS-232-C je přímo postavenou na tomto přístupu, je dobré si k ní přidat podrobnější informace.

#### Start-bit

Vkládaný na začátek rámce, tím upozorní přijímací zařízení, že začíná přenos dat. Přijímač provede synchronizaci přijímacího mechanismu, tak aby mohl přijaté bity rozpoznávat. Start-bit má vždy hodnotu různou od klidového stavu. První informační bit následuje po start-bitu.

#### Parita

Tento speciální bit slouží pro testování správnosti přenosu. Hodnota paritního bitu se mění na základě datových bitů a kontroluje ji přijímací zařízení vzhledem k ostatním datovým bitům. Pokud se paritní bit neshoduje s předpokladem, tak muselo dojít k chybě během přenosu dat. Příklady parity:

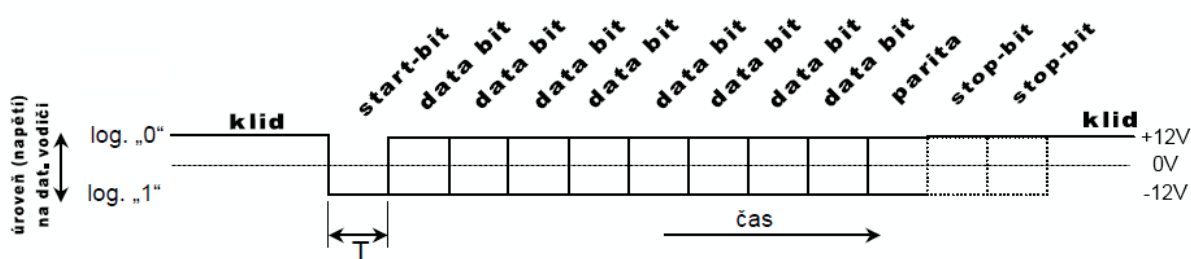
- *Sudá parita* – celkový počet datových bitů v log. 1 a hodnota paritního bitu tvoří sudé číslo.
- *Lichá parita* – celkový počet datových bitů v log. 1 a hodnota paritního bitu tvoří liché číslo.
- *Bez parity* – paritní bit se nevyužívá nebo se přijímacím zařízením ignoruje, souvisí s naprogramováním jednotlivých zařízení.

#### Stop-bit

Za posledním datovým bitem a případným paritním bitem je potřeba vyslat minimálně jeden stop-bit, jeho hodnota je vždy hodnota klidového stavu a jeho význam je v tom, že odděluje od sebe jednotlivé datové rámce. Pokud přijímací zařízení vyžaduje před zpracováním dalšího rámce určitou dobu, je vysíláno stop-bitů více.

## Rámeček

Za rámeček se považuje celá soustava tvořená start-bitem, datovými bity, paritou a zakončena stop-bitem. Takovýto rámeček je rozdělen na jednotlivé časové intervaly, kde jeden interval odpovídá jednomu bitu a také hodinovému taktu vysílače. Ukázka je na obrázku 2.9, převzatého z [4], na obrázku lze vidět jeden start-bit, osm datových bitů s paritou a zakončení stop-bitem. Pro správnou funkci sériové komunikace musí být na vysílači i přijímači nastaven generátor hodin se stejnou periodou  $T$ , která je taktéž vyznačena na obrázku 2.9.



Obrázek 2.9: Rámeček pro asynchronní sériový přenos.

### 2.2.2.5 Přenosová rychlost

Vyjadřuje počet diskretních signálů za vteřinu a je uváděna v bitech za sekundu, nebo podle francouzského badatele Baudota je zvykem psát rychlost v **baudech (Bd)** za sekundu. U výše zmíněného přenosu je důležité si uvědomit, že počet přenesených *datových bitů* bude nižší než uvedená rychlost, důvod tkví v tom, že rámeček obsahuje i několik bitů, které neobsahují žádnou informaci (start-bit, stop-bit, parita). Přenosovou rychlostí v bitech za sekundu získáme převrácením hodnoty periody hodinového signálu, tedy odpovídá frekvenci hodinového signálu.

Obvyklé rychlosti v bitech za sekundu jsou 110, 150, 300, 600, 1200, 2400, 4800, 9600 a 19200. Pro modemy se používají rychlosti 1200 a 2400, mezi terminálem a počítačem je standardní rychlost 9600 a pro přenos z počítače na tiskárnu bývá rychlost 9600, jak je uvedeno v [3].

### 2.2.2.6 Parametry asynchronní komunikace podle RS-232-C

Jak už bylo uvedeno výše, standard přesně definuje asynchronní komunikaci pro přenos dat, datové byty jsou vysílány od nejméně významného po nejvíce významný bit, kde počet datových bitů může být volitelný. Nejobvyklejší je přenášet data po 8 bitech tedy jednom Bytu. Pro přenos ASCII znaků může být postačující pouze 7 datových bitů, jako dobrý příklad komunikace může být obrázek 2.9.

## 2.3 Síťové architektury

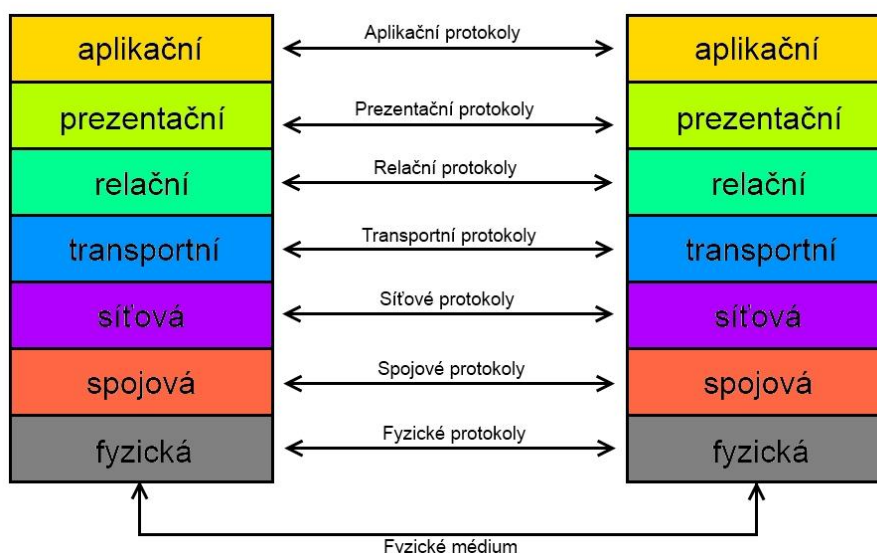
V následující kapitole se seznámíme se základem síťových architektur, ukážeme si základní principy činnosti a jejich význam.

Pod pojmem síťové architektury si musíme představit strukturu řízení komunikace v komunikačních systémech, to znamená souhrn pravidel pro výměnu dat mezi komunikujícími systémy. Celý tento systém je velmi složitý, protože se skládá z mnoha dílčích úkolů a problémů. To vedlo ke vzniku takzvaných **vrstev**, proto se síťové architektury nazývají vrstevové. Představíme si dva vrstevové modely, model OSI a model TCP/IP nad kterým je vybudován současný internet.

### 2.3.1 Referenční model OSI

Tento model byl přijat jako norma IS 7498 v roce 1984<sup>1</sup>, jeho hlavním úkolem je poskytnout referenční základnu pro koordinované vypracování norem, které na něj mohou přímo navazovat (takto vzniklo TCP/IP).

Referenční model OSI má sedm vrstev, každá tato vrstva má přesně definované funkce, potřebné pro komunikaci s jiným systémem. Model je zobrazen na obrázku 2.10. Každá vrstva



Obrázek 2.10: Referenční model OSI.

využívá služeb své sousední nižší vrstvy a sama poskytuje svoje vlastní služby pro sousední vrstvy, které jsou nad ní.

#### 2.3.1.1 Popis vrstev referenčního modelu OSI

Jak už bylo řečeno, model má sedm vrstev, kde nejnižší dvě vrstvy je možné implementovat hardwarově i softwarově, vyšší vrstvy se implementují pouze softwarově.

##### Vrstva 7 – aplikační vrstva

Vrstva poskytuje aplikačním procesům přístup ke komunikačním systémům, čímž umožňuje fyzickou spolupráci. Příkladem mohou být aplikace pro elektronickou poštu (protokol SMTP), vzdálený přístup (protokol TELNET), správa sítí (protokol SNMP) nebo přenosy souborů (protokol FTP).

<sup>1</sup> Jak uvádí Pužmanová v (8)

### **Vrstva 6 – prezentační vrstva**

Vrstva zajišťuje transparentní přenos zpráv mezi koncovými uživateli, zabývá se strukturou zpráv, ne jejím významem. Příkladem může být protokol Samba (SMB).

### **Vrstva 5 – relační vrstva**

Vrstva zajišťuje organizaci a synchronizaci dialogu mezi spolupracujícími prezentačními entitami a řídí výměnu dat mezi nimi. Do vrstvy můžeme zařadit například NetBios, AppleTalk a další podobné protokoly.

### **Vrstva 4 – transportní vrstva**

Vrstva poskytuje transparentní, spolehlivý přenos s požadovanou kvalitou, dále také optimalizuje nejrůznější síťové vrstvy. Zajišťuje komunikaci mezi jednotlivými systémy. Patří sem protokoly UDP a TCP.

### **Vrstva 3 – síťová vrstva**

Vrstva zajišťuje směrování v síti a síťové adresování, poskytuje spojení mezi systémy, které spolu nesousedí. Obsahuje nástroje pro překlenutí rozdílných technologií použitých pro přenosové sítě. Typickým protokolem pro síťovou vrstvu je protokol IP, dále ICMP a ARP.

### **Vrstva 2 – spojová vrstva**

Zajišťuje spojení mezi dvěma sousedícími komunikujícími systémy. Data z fyzické vrstvy uspořádává do datových rámců. Spojení se vytváří i zakončují dynamicky.

### **Vrstva 1 – fyzická vrstva**

Vrstva podporuje fyzickou komunikaci dat mezi systémy, aktivuje, udržuje a deaktivuje fyzická spojení pro přenos bitů, také definuje elektrické a fyzikální vlastnosti zařízení.

## **2.3.2 Architektura TCP/IP<sup>1</sup>**

Označuje celou síťovou architekturu, umožňuje oprostít se od závislosti na síťové infrastruktuře, to znamená, že je možné sítě libovolně propojovat, aniž by se uživatelé museli zajímat o detaily způsobu propojení. Protokol TCP/IP se chová ke všem sítím stejně, bez ohledu na fyzické charakteristiky.

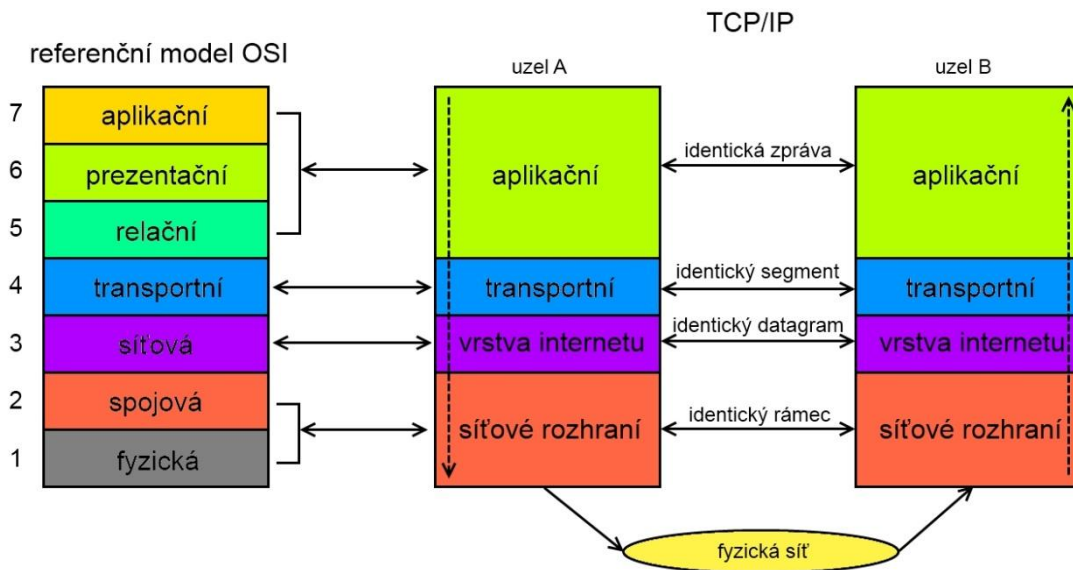
Architektura TCP/IP je opět vrstvová, na rozdíl od model OSI má jen čtyři vrstvy, srovnání lze vidět na obrázku 2.11. Vidíme, že *síťové rozhraní* odpovídá spojové a fyzické vrstvě. *Vrstva internetu* odpovídá svými funkcemi vrstvě síťové, velmi podobné je to i u *vrstvy transportní*, ta odpovídá se svými funkcemi transportní vrstvě definované v OSI. Vrstva na nejvyšší úrovni *aplikační* v sobě zahrnuje relační, prezentační a aplikační vrstvy definované v OSI.

---

<sup>1</sup> TCP/IP – Transmissio Control Protocol/Internet protokol.



Způsob komunikace je naznačen na obrázku 2.11, zpráva vygenerovaná uživatelem A postupně prochází vrstvy od nejvyšší po nejnižší a je následně poslána po fyzickém médiu uživateli B, kde je zpráva opět postupuje od nejnižší po nejvyšší, až se projeví v aplikaci u uživatele B.



Obrázek 2.11: Porovnání architektur OSI a TCP/IP a vrstvý princip komunikace u TCP/IP.

### Vrstva síťového rozhraní

Tato nejnižší vrstva umožňuje přístup k fyzickému přenosovému médiu, je specifická pro každou síť podle její implementace. Funkce na této vrstvě zajišťují mapování IP adres na fyzické adresy používané v dané síti. Jako příklady sítí můžeme uvést Ethernet, X.25, FDDI, Token Ring a další.

### Vrstva internetu

Mezisíťová vrstva, její funkce odpovídají síťové vrstvě referenčního modelu OSI a jak už bylo zmíněno, hlavním úkolem je síťová adresace, směrování a předávání datagramu přes komunikační podsíť včetně směrování. Je implementovaná ve všech prvcích sítě (směrovače, koncové zařízení). Zahrnuje protokoly IP (vysílání datagramu), ARP (mapování adres), ICMP (řízení hlášení) a směrovací protokoly jako OSPF a také protokol IGMP pro správu skupin.

### Transportní vrstva

Odpovídá transportní vrstvě referenčního modelu OSI, poskytuje mechanismus pro koncový přenos dat mezi dvěma stanicemi. Pro to využívá dvou protokolů:

- *User Datagram Protocol (UDP)* – jednoduchá transportní služba bez souvislého spojení, jedná se o takzvaný nespolehlivý transportní protokol, protože nekontroluje doručení dat.
- *Transmission Control Protocol (TCP)* – transportní služba se spojením, včetně zabezpečení datového toku, proto se jedná o spolehlivý transportní protokol.

## **Aplikační vrstva**

Nejvyšší vrstva síťové architektury Internetu a obsahuje všechny protokoly poskytující uživatelům konkrétní aplikace. Tyto protokoly podporují jak uživatelské aplikace, ale také administrativní aplikace, zde je výčet nejpoužívanějších protokolů:

- **TELNET** – používaný pro vzdálené připojení.
- **FTP** – používaný pro přenos souborů.
- **SMTP** – používaný pro přenos zpráv elektronické pošty.
- **NFS** – vzdálený přístup k souborovému systému.
- **DHCP** – vzdálená konfigurace počítačů připojených do sítě.
- **DNS** – používá se pro doménový překlad adres.
- **HTTP** – používaný pro výměnu hypertextových dokumentů.

## 3 Návrh řešení

V následujících kapitolách si představíme návrh mého řešení problematiky převodníku Ethernet na RS-232. Seznámíme se s mikroprocesory řady STM32, dále zde představíme TCP/IP stack, který bude sloužit jako základní framework pro firmware zařízení a v závěru si představíme programové nástroje použité při vývoji převodníku.

### 3.1 Mikrokontrolér STM32

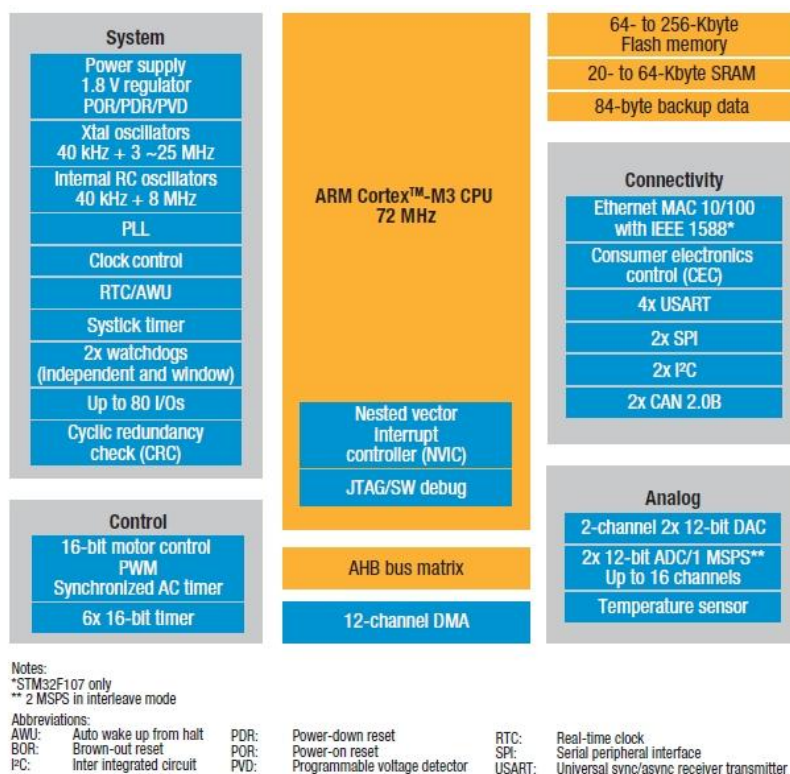
Pro převodník jsem zvolil 32-bitový procesor řady STM32, tento mikrokontrolér je založen na jádru ARM-Cortex M3, který na trh přinesla společnost ST-Microelectronics v roce 2009. Mikrokontroléry řady STM32 obsahují většinu standardních periferních zařízení jako různé časovače, A/D a D/A převodníky, sériové rozhraní USART. Jeho velkou výhodou je to, že dále obsahuje i rozhraní USB, CAN a některé verze mají v sobě zabudovanou Ethernet 10/100 MAC vrstvu, právě díky tomuto nadstandardnímu rozšíření je použití MCU řady STM32 ideální pro naši aplikaci.

#### 3.1.1 MCU STM32F107VCT6

Pro převodník budeme používat přímo obvod STM32F107VCT6, následuje výčet vlastností charakterizující zvolený obvod:

- 256kB programovatelná FLASH paměť.
- 64kB SRAM paměť.
- Pouzdro LQFP100.
- Ethernet 10/100 MAC vrstva.
- 7 časovačů.
- 3 rozhraní SPI(I<sup>2</sup>S).
- 1 rozhraní I<sup>2</sup>C.
- 5 rozhraní USART.
- 1 USB rozhraní.
- 2 CAN rozhraní.
- 80 vstupních výstupních pinů.
- Frekvence CPU 72MHz.
- Operační napětí od 2,0 do 3,6 V.

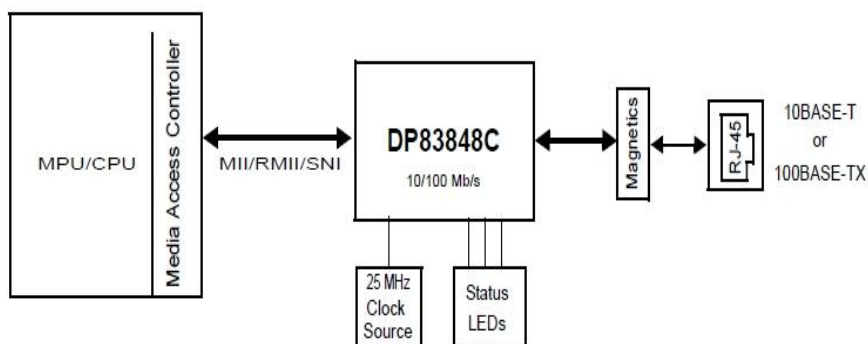
Blokové schéma obvodů řady STM32 je na obrázku 3.1<sup>1</sup>. Technické informace jsou z dokumentace k MCU [5].



Obrázek 3.1: Blokové schéma MCU řady STM32.

### 3.1.2 Ethernet 10/100 MAC vrstva

Jak jsem se již zmínil, MCU má v sobě zabudovaný Ethernet rozhraní. Ethernet 10/100 MAC vrstva odpovídá v referenčním modelu OSI spojové vrstvě viz kapitola 2.3.1, z toho důvodu je ještě potřeba použít zařízení, které nám bude zastupovat vrstvu fyzickou, tak aby mohly být data vysílána do sítě. Takovéto zařízení se anglicky nazývá Ethernet physical transceiver (PHY). Tento obvod obsahuje



Obrázek 3.2: Typické zapojení PHY obvodu.

<sup>1</sup> Obrázek převzat od ze stránek výrobce MCU (www.st.com).

standardní komunikační rozhraní MII<sup>1</sup> respektive RMII<sup>2</sup> a dále výstupní budiče, které se připojují na transformátor, který je dále napojen na Ethernet konektor, standardně RJ-45.

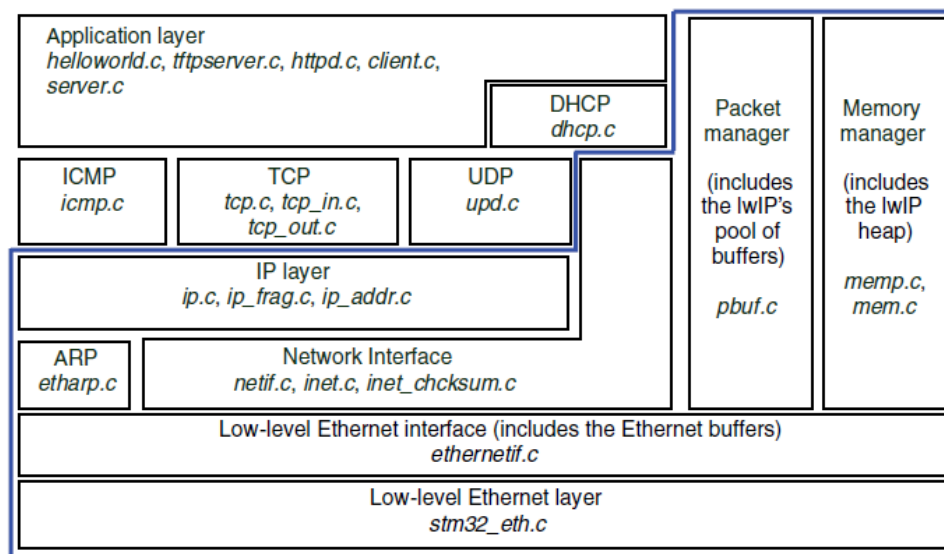
Jako PHY obvod bude v převodníku použit integrovaný obvod DP83848CVV, bližší informace o obvodu naleznete v technické dokumentaci od výrobce [6]. Typické použití obvodu PHY lze vidět na obrázku 3.2<sup>3</sup>.

## 3.2 TCP/IP Stack

Zadáním práce je zkonstruovat převodník Ethernet na RS-232, který přijímá data přes TCP/IP protokol, proto je klíčové mít k dispozici TCP/IP stack, který programátorovi usnadní práci při implementaci firmware, protože nebude muset programovat celou komunikaci přes Ethernet rozhraní. Já jsem si pro svou práci vybral LwIP stack volně šiřitelný pod BSD licenci<sup>4</sup>.

### 3.2.1 LwIP stack

LwIP stack je velmi propracované řešení TCP/IP stacku. Velkou výhodou tohoto stacku jsou demonstrační programy na základě, kterých lze jednoduše postavit vlastní aplikaci. Na obrázku 3.3<sup>5</sup> lze vidět uspořádání LwIP stacku a jeho knihoven. Lze vidět, že stack implementuje všechny vrstvy



Obrázek 3.3: Uspořádání LwIP stacku.

síťového modelu TCP/IP od síťové až po transportní vrstvu, aplikační vrstva už závisí na implementaci uživatele.

<sup>1</sup>Viz kapitola 2.1.4.2.

<sup>2</sup>Jedná se o omezené MII, anglicky Reduce MII.

<sup>3</sup>Převzatý z [6].

<sup>4</sup>Umožňuje volné šíření licencovaného obsahu, přičemž vyžaduje pouze uvedení autora a informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

<sup>5</sup>Převzatý z [15].

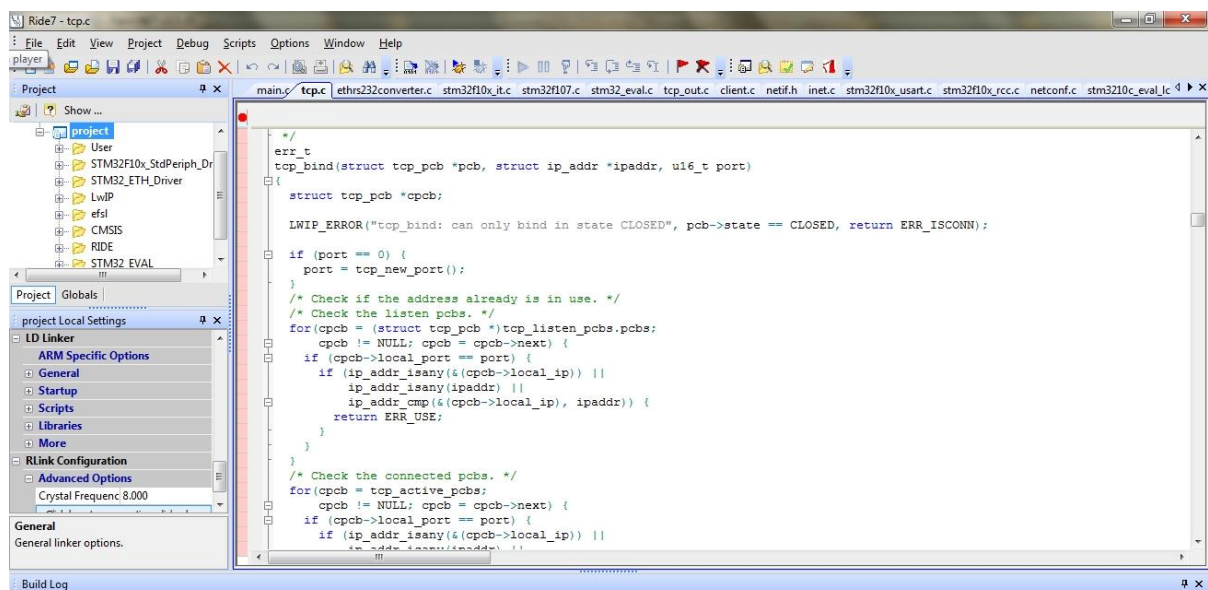
### 3.3 Vývojové prostředí

Při vývoji firmware pro převodník bylo třeba zvolit vhodné prostředí a nástroje pro debugování aplikace. Vzhledem k tomu, že jsem se rozhodl využít pro svoji aplikaci demonstrační příklady LwIP tak se nabízí použití softwaru RIDE7 od společnosti Raisonance.

Tento software obsahuje vývojové prostředí pro mikrokontroléry postavené na jádře ARM-Cortex M3, dále obsahuje nástroje pro optimalizaci kódu, projektového manažera, nástroje pro debugování ve vyšším programovacím jazyce (jazyce C), a editor kódu se syntaktickou kontrolou. Aplikační okno je na obrázku 3.4.

Firmware byl vyvíjen na vývojovém kitu SMT3210C-EVAL, který umožňuje využít všechny možné periférie, připojené k MCU. Právě pro tento kit jsou taky určeny demonstrační aplikace LwIP stacku, proto i výsledný převodník kvůli kompatibilitě vychází z návrhu vývojového kitu.

Jako programovací a debugovací nástroj byl použit programátor RLink od společnosti Raisonance pro mikrokontroléry společnosti ST Microcontrollers.



Obrázek 3.4: Aplikační okno software RIDE7.

## 4 Návrh a konstrukce převodníku

Cílem této kapitoly je seznámit čtenáře s celkovým návrhem zařízení a konstrukce výsledné DPS. Vzhledem k rozsahu návrhu, je třeba přistupovat k jednotlivým blokům postupně, kde všechny schémata jsou uvedena v příloze 1 až 7.

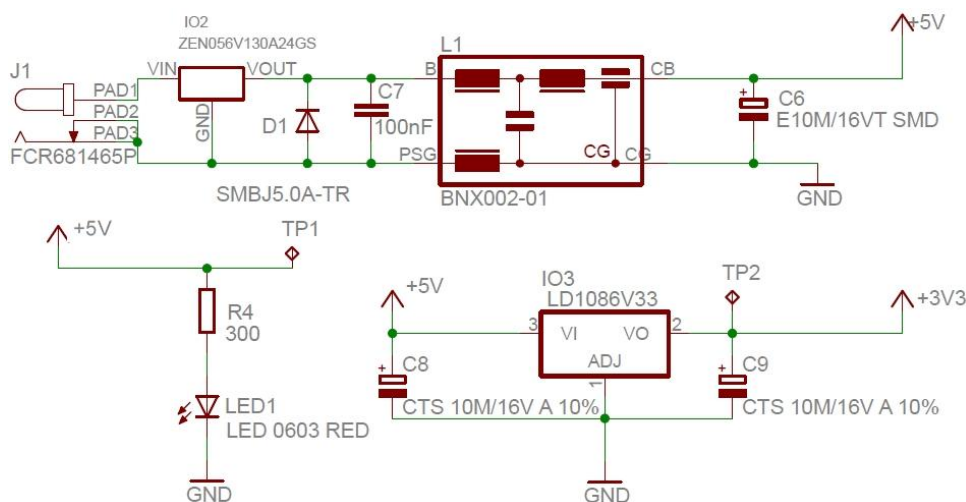
Při návrhu zařízení a DPS bylo použito softwaru Eagle od společnosti Cadsoft, jehož výhodou je v tom, že poskytuje velké množství knihoven s většinou standardních součástek a ve free verzi umožňuje návrh dvouvrstvé DPS.

### 4.1 Popis elektronického schéma zapojení

#### 4.1.1 Napájecí zdroj

Převodník je navržen tak, aby mohl být napájen stejnosměrným napětím o hodnotě +5V, na vstupu je chráněn PolyZenerovou diodou ZEN056V130A24GS (IO2), která zařízení chrání proti přepólování (chybné připojení napájecího napětí), anebo chrání proti příliš velkému napětí na vstupu. Dále je zde použita další zenerova dioda pro ochranu proti přechodovému napětí (D1) a frekvenční EMI filtr BNX002-01 (L1).

Kvalita zařízení se bezpochyby odvíjí od kvality elektronického napájení, které je pro zařízení použito. Kompletní schéma, ve kterém jsou popsány i výše uvedené součástky, je na obrázku 4.1. Pro stabilizaci napětí je použit nenastavitelný lineární stabilizátor LD1086V33 (IO3) od společnosti ST-Microelectronics. Jeho výhodou je to, že se jedná o low-dropout stabilizátor, tedy stabilizátor nízkého odstavu, to má význam v tom, že mezi vstupní a výstupní svorkou dochází k velmi malému úbytku napětí, v tomto případě 1,3V při maximálním výstupním proudu 1,5A. Abychom na výstupu dosáhli požadovaných 3,3V je potřeba mít minimální napájecí napětí 4,6V.

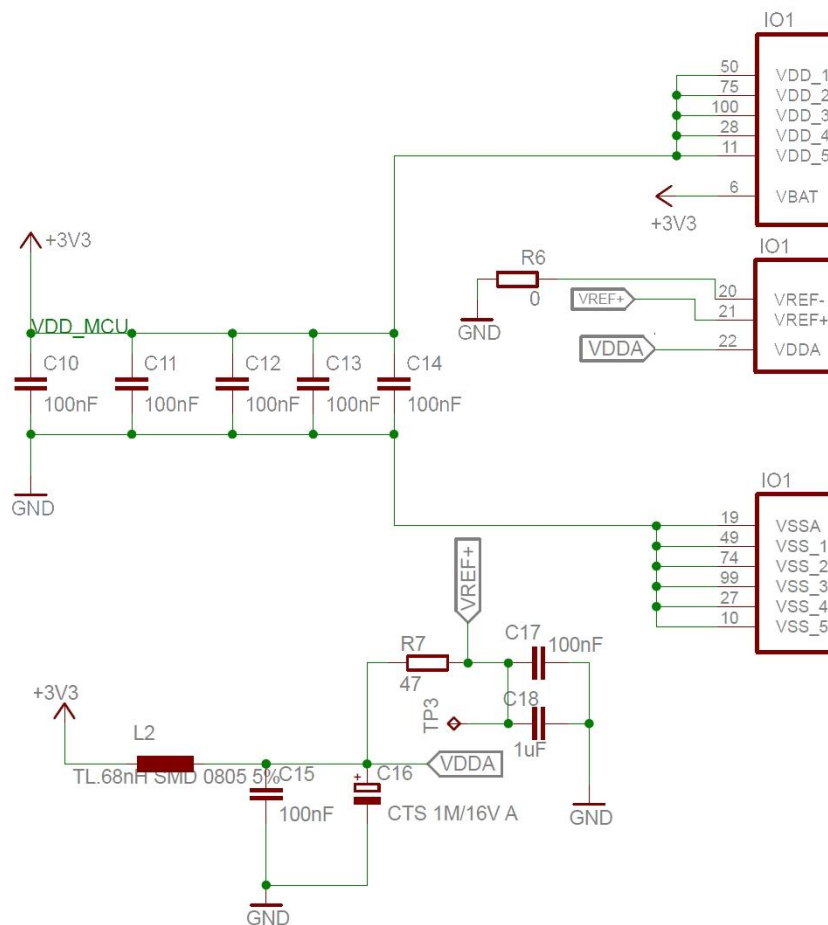


Obrázek 4.1: Schéma napájecího zdroje pro převodník.

Zapojení stabilizátoru vychází přesně s katalogového zapojení, včetně hodnot použitých součástek tantalových kondenzátorů C8 a C9 o hodnotě 10 $\mu$ F. Zdroj je opatřen dvěma testovacími body TP1 pro +5V a TP2 pro +3,3V, ty lze využít pro kontrolu při potížích s napájením. Také je zde červená LED dioda (LED1), která slouží pro rychlou kontrolu funkčnosti napájení, je zapojena v sérii s rezistorem R4 o hodnotě 300 $\Omega$ , tak aby diodou procházel proud o velikosti 10mA.

## 4.1.2 Napájení MCU

Pro napájení MCU je využito 15 pinů, jak lze vidět na obrázku 4.2. Jsou zde blokovací kondenzátory



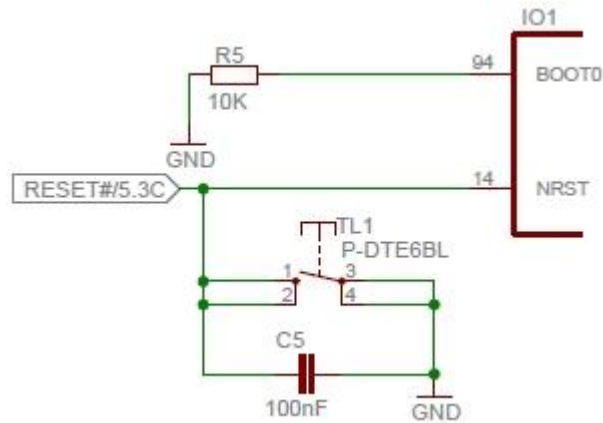
Obrázek 4.2: Schéma zapojení napájení MCU.

C10 až C14 o hodnotě 100nF, které jsou připojeny vždy mezi napájecí piny (VDD\_X, VSSA, VSS\_X). Další důležitou částí je napájení analogové sekce MCU, ta obsahuje A/D a D/A převodníky, které pro svoji činnost potřebují dobře odfiltrované napětí na pinu 21 (VREF+) a pinu 22 (VDDA). Toho jsme dosáhli zapojením tlumivky L2 do cesty napájení, za kterou následuje dvojice kondenzátorů, blokovací keramický C15 o hodnotě 100nF a tantalový o hodnotě 1 $\mu$ F z důvodu větších požadavků na odběr napětí. Takto ošetřené napětí je dále připojeno přes rezistor R7 přímo na vstup VREF+. Je zde opět kontrolní ploška TP3 a další blokovací kondenzátor C17 mezi zemí a VREF+.



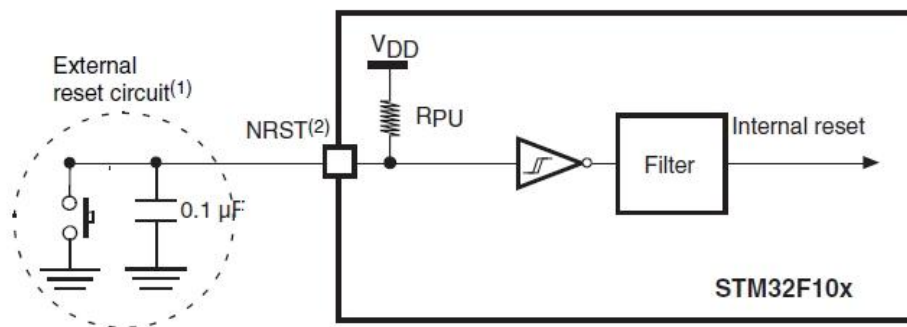
### 4.1.3 Resetovací tlačítko a bootování MCU

Součástí převodníku je taktéž resetovací tlačítko, po jehož stisknutí dojde k restartu MCU a prováděný program se začne vykonávat od začátku, zapojení tlačítka je na obrázku 4.3.



Obrázek 4.3: Schéma zapojení resetovacího tlačítka a BOOT0.

Po stisknutí tlačítka TL1 dojde k propojení NRST pinu se zemí, k tlačítku TL1 je paralelně připojený kondenzátor C5 o hodnotě 100nF, toto zapojení vychází z doporučení výrobce, který jej uvádí v dokumentaci k MCU [5], doporučené schéma zapojení je na obrázku 4.3, z kterého také vychází má konstrukce.



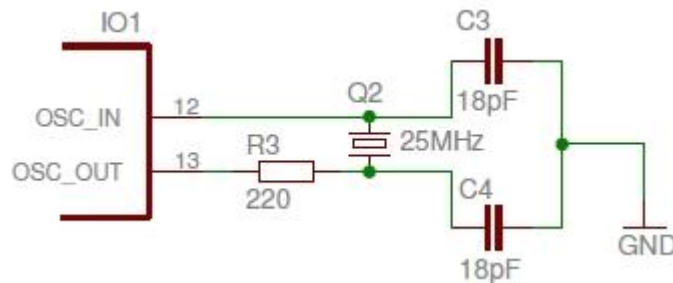
Obrázek 4.4: Schéma pro doporučené zapojení resetovacího mechanismu MCU.

Zvolené MCU má různé režimy bootování, pro řešení naší aplikace si vystačíme pouze s jediným způsobem a to bootování programu z vnitřní paměti FLASH. To je zajištěno tím, že pin BOOT0 (pin 94) je připojen přes rezistor R5 o hodnotě 10kΩ na zem. Tím zajistíme, že na vstupu BOOT0 bude neustále logická 0, která nám zajistí, že procesor bude po restartu nebo při spouštění načítat program z vnitřní paměti FLASH.

## 4.1.4 Oscilátory

Součástí převodníku je několik generátorů hodin neboli oscilátorů. MCU řady STM32F107XX mají v sobě zabudovaný interní 8 MHz oscilátor, který je po restartu používán jako defaultní, ale také může být použit externí oscilátor v rozsahu 3 až 25 MHz. Použitím externího oscilátoru si zajistíme, že časování zařízení bude přesnější než při použití interního oscilátoru.

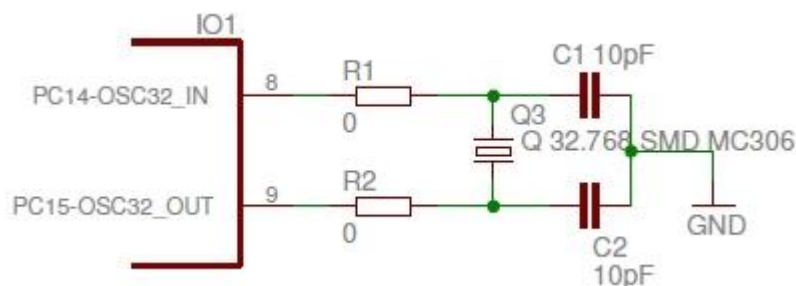
Externí oscilátor se připojuje na piny označované jako OSC\_IN (pin 12) a OSC\_OUT (pin 13), jak lze vidět na obrázku 4.5, tak je na zmíněné piny připojen 25MHz oscilátor Q2, u OSC\_IN je připojen přímo, u OSC\_OUT je připojen ještě přes rezistor R3 o hodnotě 220Ω. Dále jsou na nožky oscilátoru připojeny startovací kondenzátory C3 a C4 o hodnotě 18pF, hodnota byla vybrána na základě doporučení výrobce z rozsahu 5 až 25pF. Celé zapojení vychází z doporučení výrobce v dokumentaci k MCU [6]. Frekvence 25Mhz byla vybrána s ohledem na činnost zařízení, které bude



Obrázek 4.5: Schéma zapojení externího oscilátoru.

komunikovat po MII sběrnici.

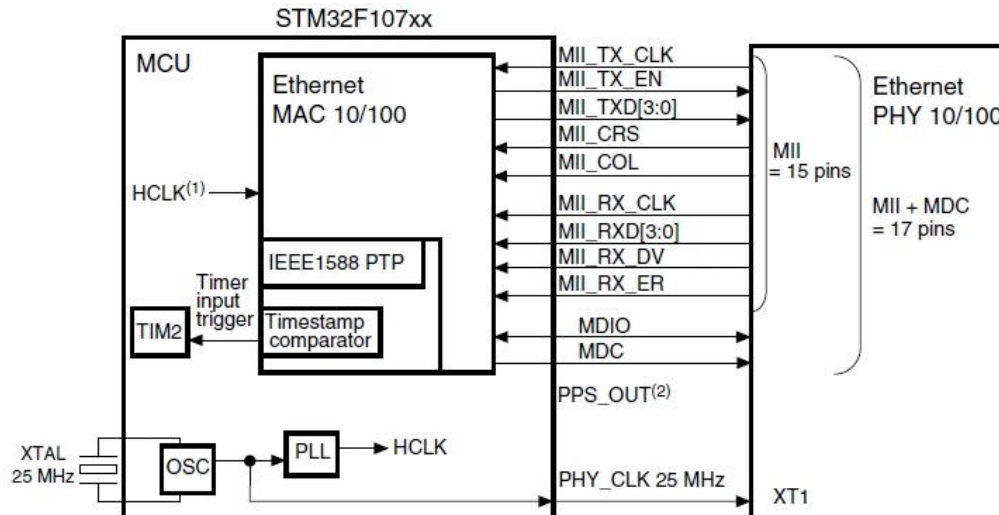
K MCU je připojen ještě jeden externí oscilátor Q3 o hodnotě 32,768 kHz. Hodnota tohoto pomalého oscilátoru souvisí s tím, že je jednoduše dělitelná mocninou s dvojkovým základem. Tento oscilátor je napojen na piny OSC32\_IN (pin 8) a OSC32\_OUT (pin 9), které jsou určeny vnitřním RTC hodinám, ty jsou použity pro generování reálného času, například denního času. Schéma zapojení je uvedeno na obrázku 4.6 a lze vidět, že je opět použito opět startovacích kondenzátorů C1 a C2, tentokrát o hodnotě 10pF, jedná se o střední hodnotu z rozsahu 5 až 15pF, které doporučuje výrobce. Zapojení opět vychází z doporučení výrobce uvedeného v dokumentaci k MCU [5].



Obrázek 4.6: Schéma zapojení real-time oscilátoru.

## 4.1.5 Připojení PHY a konektoru RJ-45

V kapitole 3.1 jsem již vysvětlil, že samotné MCU nemá fyzickou vrstvu pro vysílání dat, proto je potřeba použít externího obvodu PHY. Blokové schéma zapojení PHY a MCU je na obrázku 4.7<sup>1</sup>, při návrhu jsem se držel doporučeného zapojení, které lze právě vidět obrázku 4.7.



Obrázek 4.7: Blokové schéma zapojení MCU a PHY bez použití externího oscilátoru.

Kompletní schéma zapojení fyzické vrstvy Ethernet je v příloze 6. Na schématu dominují hlavně dvě součástky IO6 (DP83848CVV) a J2 (RJMG163218101NR) což je konektor RJ-45 s přímo zabudovaným transformátorem pro vysílání signálů. PHY nemá vlastní resetovací tlačítko, ale jak lze vidět tak pin pro reset obvodu (pin 29) je připojen na společný tlačítko TL1 sloužící i pro restartování MCU. Zapojení veškerých součástek a jejich hodnot vychází z doporučení výrobce viz dokumentace [6].

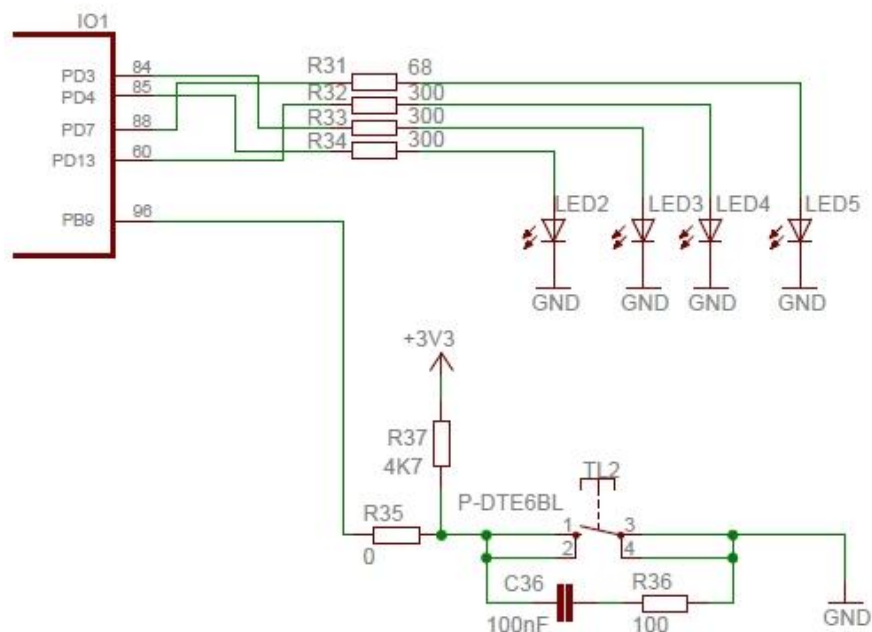
## 4.1.6 Uživatelské tlačítko a LED diody

Součástí převodníku je i tlačítko TL2, nepočítám, že by tlačítko mělo mít v původní implementaci nějakou funkci, ale v budoucnu při rozšiřování firmware může být pro uživatele a programátora nápomocné. Zapojení tlačítka je na obrázku 4.8, po stisknutí tlačítka by měl být vyslán signál přibližně o délce 47ms, tato časová konstanta je dána trojicí součástek R37 o hodnotě 4,7k $\Omega$ , C36 o hodnotě 100nF a R36 o hodnotě 100 $\Omega$ .

V návrhu jsem nezapomněl opatřit převodník o čtveřici LED diod<sup>2</sup>, které mohou být nápomocné jak při vývoji firmware, tak dále při signalizaci činnosti převodníku. Zapojení LED diod je taktéž na obrázku 4.8. Hodnoty rezistorů byly voleny tak, aby diodami procházel proud odpovídající katalogové hodnotě, při předpokládaném úbytku napětí na diodě.

<sup>1</sup> Převzatý z [6].

<sup>2</sup> LED2 žlutozelená, LED3 žlutá, LED4 červená a LED5 modrá.

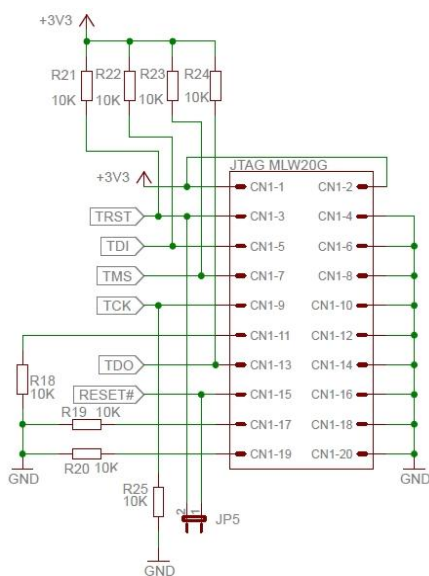


Obrázek 4.8: Schéma zapojení uživatelského tlačítka a LED diod.

#### 4.1.7 Programovací JTAG konektor

Pro nahrávání programu do paměti FLASH, má MCU rozhraní JTAG nebo SWD. Obě tyto rozhraní jsou vyvedeny na 20pinový konektor CN1. Zapojení je na obrázku 4.9, konektor má dvě řady po deseti pinech se standardním rozstupem 2,54mm.

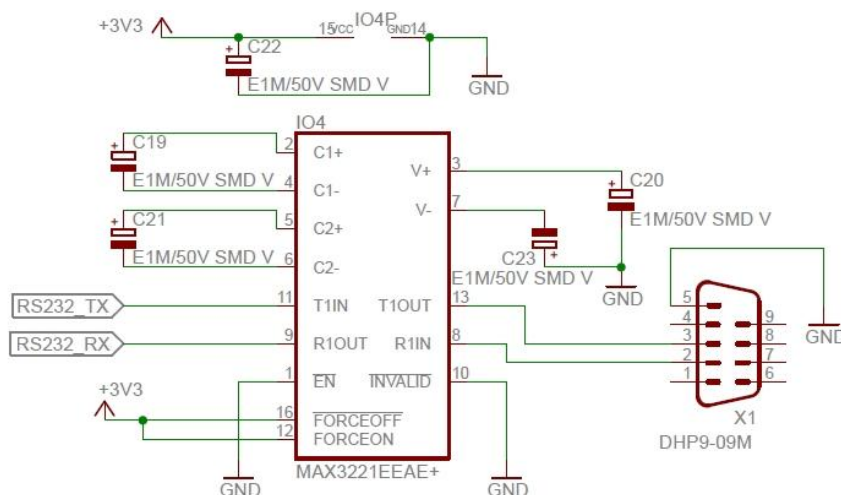
Zapojení konektoru odpovídá standardnímu rozmístění vývodů pro většinu ARM JTAG adaptérů, tímto způsobem propojujeme počítač a MCU. Zapojení odpovídá zapojení schématu pro vývojový kit STM3210C-eval.



Obrázek 4.9: Schéma zapojení JTAG konektoru.

## 4.1.8 RS-232 rozhraní

Pro vysílač a přijímač rozhraní RS-232 jsem vybral integrovaný obvod od společnosti Maxim Integrated Products MAX3221EEAE+, který je ve schématu označen jako IO4. Tento obvod má výhodu v tom, že je chráněn proti nežádoucí statické elektřině, která by ho mohla za normálních okolností poškodit. Maximální přenosová rychlost je 250kb/s a je ho možno napájet napětím v rozmezí od 3 do 5,5V. Zapojení, které lze vidět na obrázku 4.10, vychází přímo z doporučení



Obrázek 4.10: Schéma zapojení RS-232 transieveru.

výrobce a je uvedené v technické dokumentaci [7]. Integrovaný obvod má jenom jeden vysílač a jeden přijímač, což jde vyčíst se schématu zapojení, více pro činnost rozhraní RS-232 nepotřebujeme, protože zmíněný *handshaking* v kapitole 2.2.2.3 bude řešen softwarově nebo vůbec.

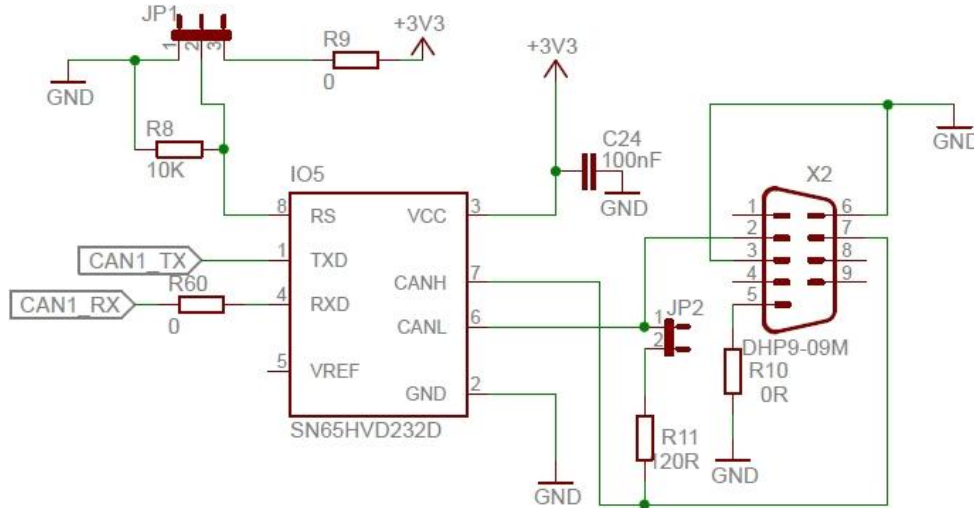
Obvody od firmy MAXIM založeny na této technologii patřily k průlomovým objevům. Obvod je totiž schopen sám generovat dostatečné napětí odpovídající standardu pro RS-232-C, přičemž k napájení stačí jedno kladné napětí. Potřebné napětí se vytváří pomocí elektrolytických kondenzátorů, ty lze vidět ve schématu pod označením C19 až C23. Na transiever je připojen standardní 9pinový D-SUB konektor, kde připojení jednotlivých pinů odpovídá normě, ve schématu je označen jako X1.

## 4.1.9 CAN rozhraní

Vzhledem k tomu, že použitím mikroprocesoru řady STM32 pouze pro převod z Etherentu na sériové rozhraní RS-232, by bylo plýtvání potenciálem tohoto MCU, tak jsem se rozhodl k převodníku ještě přidat CAN rozhraní.

V budoucnu pak bude možné na MCU nahrát firmware, který bude řešit převod z Ethernet rozhraní na RS-232 nebo na rozhraní CAN.

Zapojení CAN rozhraní je ještě jednodušší jak u rozhraní RS-232, máme zde opět integrovaný obvod IO5, který zajišťuje vysílání a příjem dat přes 9pinový D-SUB konektor X2. Jako CAN transiever jsem zvolil integrovaný obvod SN65HVD232D a návrh zapojení provedl podle doporučení výrobce v dokumentaci [8]. Výsledné schéma zapojení je na následujícím obrázku 4.11.



Obrázek 4.11: Schéma zapojení rozhraní CAN.

## 4.2 Návrh desky plošných spojů

Návrh desky plošných spojů přímo vychází ze zapojení u vedeného v kapitole 4.1.

### 4.2.1 Deska plošných spojů

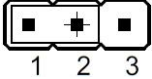
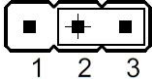
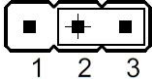
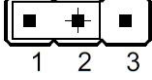
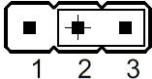
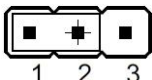
Výsledkem návrhu je dvouvrstvá DPS o rozměrech 100mm na výšku a 160mm na šířku. Celkový pohled na DPS z hora lze vidět v příloze 8.

### 4.2.2 Rozmístění součástek na DPS

Přesné rozmístění součástek lze vidět v příloze 9. Konektory pro CAN (X2) a RS-232 (X1) jsou umístěny v horní části, zatímco konektor RJ-45 (J2) je umístěn na spodní straně DPS. V levé části DPS se nachází zdrojová část převodníku, je zde 2,5mm konektor pro připojení napájení ze sítě, dále filtr a stabilizátor napětí. Z výstupu stabilizátoru (IO3) je kladné napětí 3,3V rozvedeno po celé desce. V levé části se také nachází programovací 20pinový JTAG konektor. MCU je umístěno přibližně uprostřed desky, tak aby k němu byl dobrý přístup a nebylo náročné z něj vyvést potřebná propojení. PHY obvod je umístěn v pravé části DPS poblíž konektoru RJ-45. Uživatelské tlačítko (TL2) je umístěno jako první zleva, tlačítko pro reset (TL1) je druhé zleva.

### 4.2.3 Nastavení zkratovacích propojek (jumper)

Převodník je navržený tak, že některá jeho činnost, lze hardwarově nastavit zkratovacíma propojkami, v následující tabulce 4.1 je seznam jumperů a popis jejich zapojení.

Jumper	Popis	Konfigurace
JP1	CAN transiever pracuje v high-speed režimu, pokud je takto nastaven: <b>(výchozí nastavení)</b>	
	CAN transiever je nastaven do standby režimu:	
	CAN transiever úracuje ve slope control režimu když je JP1 otevřen	
JP2	CAN terminalni rezistor R11 je zapojen, když je JP2 propojen. <b>(ve výchozím nastavení je otevřený)</b>	
JP3	JP3 vybírá MII nebo RMII režim Ethernet rozhraní, MII je zapnuto, když je JP3 otevřený, RMII je zapnuto, když je JP3 propojen. <b>(ve výchozím nastavení je otevřený)</b>	
JP4	JP4 by měl být takto zapojen, pokud je vybrán MII režim. <b>(výchozí nastavení)</b>	
	JP4 by měl být takto zapojen, pokud je vybrán RMII režim.	
JP5	JP5 povoluj reset převodníku od vestavěného JTAG TAP řadiče, pokaždé když systém reset vyžaduje. Propojuje TRST signál z JTAG konektoru se systémový signálem RESET#. <b>(ve výchozím nastavení je otevřený)</b>	
JP6	25MHz hodiny pro MII nebo 50MHz hodiny pro RMII jsou zajištěny MCO na pinu PA8, pokud je JP6 zapojen takto: <b>(výchozí nastavení)</b>	
	25MH hodiny jsou zajišťovány externím oscilátorem Q1 (funguje pouze v MII režimu).	

Tabulka 4.1: Nastavení zkratovacích propojek.

## 5 Implementace firmware převodníku

V následující kapitole, bude proveden návrh činnosti převodníku a ukázán následný postup při implementaci tohoto návrhu v podobě firmware pro převodník Ethernet na RS-232. Pro návrh a implementaci bylo použito zdrojových kódů demonstračního příkladu pro LwIP a to *HelloWord* aplikace, který je níže popsán.

### HelloWord aplikace

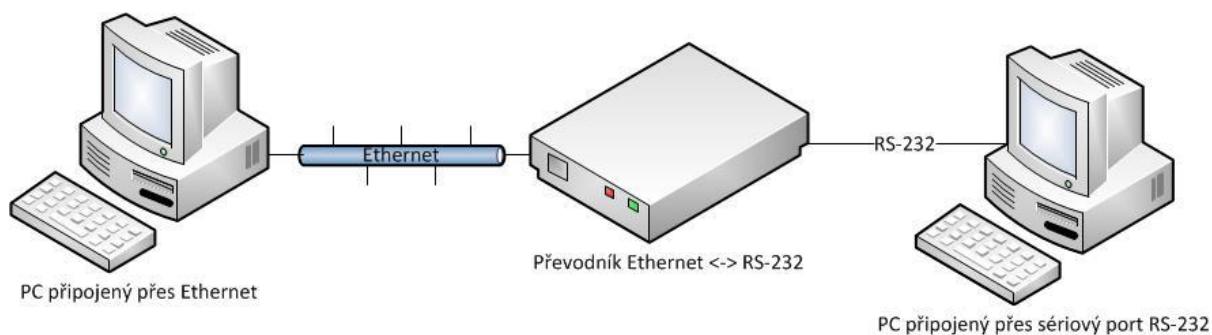
Pro spuštění aplikace je potřeba propojit vývojový kit s počítačem pomocí kroucené dvojlinky a navázat TELNET spojení. Připojení pod operačním systémem Windows provede pomocí programu Telnet, v příkazovém řádku tak učiníme následujícím příkazem:

```
>telnet „adresa vývojového kitu“
```

Po navázání spojení klient spuštěný na vývojovém kitu vypíše informační zprávu a očekává napsání nějakého textu, který bude zakončen ukončovujícími znaky OS Windows. Pokud obdrží požadovanou zprávu, tak ji opět pře pošle na telnet klienta.

### 5.1 Návrh firmware

Tak abych splnil bod zadání, požadující aby převodník přijímal data přes protokol TCP/IP, rozhodl jsem se, že komunikace mezi převodníkem a počítačem bude probíhat nad protokolem TELNET, tento protokol odpovídající aplikační vrstvě v architektuře TCP/IP a existují jednoduché aplikace, které umí nad tímto protokolem komunikovat, jako příklad uvedu aplikaci *Telnet* dostupnou v operačním systému MS Windows nebo aplikace *Putty*. Ukázka možného použití převodníku je na obrázku 5.1. Na počítači, který bude připojen přes sériový port, je možné použít aplikaci *Putty*, která je schopna naslouchat na sériovém portu počítače a také umožňuje posílání dat přes tento sériový port.

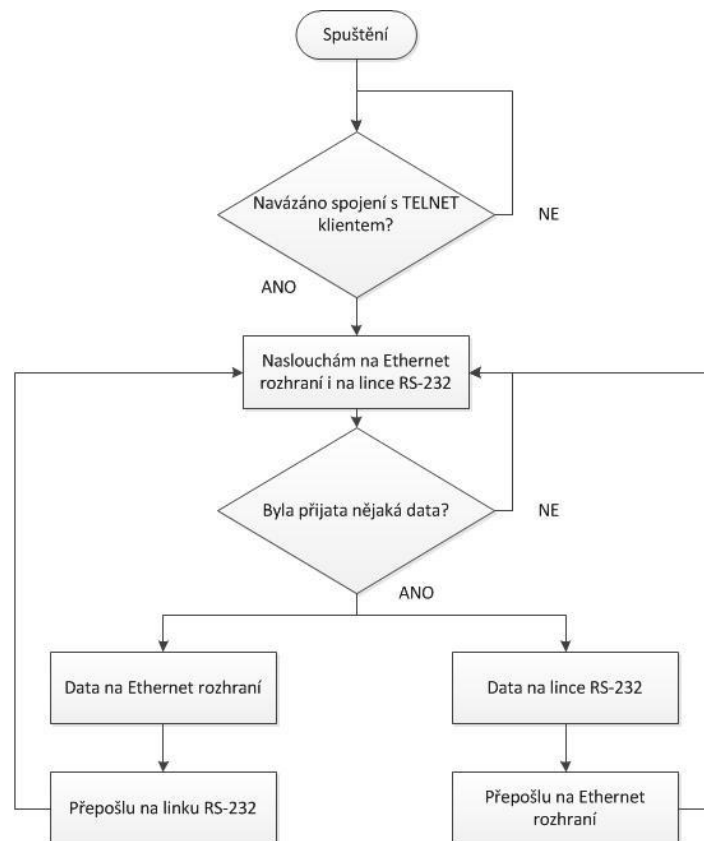


Obrázek 5.1: Možné použití převodníku.



### 5.1.1 Princip činnosti převodníku

Návrh činnosti převodníku lze shrnout v několika větách. Převodník po spuštění a inicializaci vstoupí do stavu, kdy očekává navázání spojení přes aplikační protokol telnet. Jakmile dojde k navázání spojení, tak naslouchá jak na straně Ethernet rozhraní tak i na sériové lince. Pokud převodník přijme data na jedné nebo druhé straně, tak je neprodleně přepoše na odpovídající rozhraní. Tímto přístupem bude převodník splňovat základní požadavky zadání. To co jsem napsal slovně, lze vyčíst z diagramu na obrázku 2.1.



Obrázek 5.2: Princip činnosti převodníku.

## 5.2 Implementace návrhu firmware

Výše uvedený návrh v kapitole 5.1 je potřeba přenést do softwarové podoby. Vzhledem k tomu, že se jedná především o demonstrační firmware, tak jsem využil zdrojový kódů pro demonstrování činnosti LwIP stacku na vývojovém kitu STM3210C-eval.

### 5.2.1 Nastavení LwIP stacku pro firmware

LwIP stack umožňuje mnohá nastavení, které nám mohou ulehčit práci při implementaci vlastní aplikace. LwIP stack se nastavuje v hlavičkovém souboru `lwipopts.h`. Zde uvádím některá nastavení, které mají vliv na naši aplikaci:

- `MEM_ALIGNMENT 4` – zarovnání paměti na 4 Byty.
- `MEM_SIZE (20*1024)` – velikost alokované paměti pro LwIP.
- `MEMP_NUM_TCP_PCB 1` – povolíme pouze jedno aktivní TCP spojení, umožňuje pouze připojení jednoho klienta.
- `CHECKSUM_BY_HARDWARE` – kontrolní součet bude vypočítáván pomocí hardware.

Aby mohl LwIP stack správně fungovat je potřeba nastavit IP adresu a MAC adresu PHY obvodu, to se provádí v souboru `netconf.c`, který obsahuje funkci `LwIP_init()`, která provádí výše uvedené nastavení. Nastavení jsem nechal na výchozích hodnotách, které jsou pro náš převodník dostačující.

## 5.2.2 Konfigurace MCU

Inicializaci MCU zajišťují funkce uvedené v souboru `stm32f107.c`, výpis těchto funkcí je v následujícím seznamu:

- `System_Setup()` – hlavní funkce pro konfiguraci MCU, jsou zde postupně volány funkce uvedené níže. Po zavolání této funkce dojde k systémové inicializaci (nastavení časovačů a paměti), dále nastavení systému přerušování, inicializaci LCD (projeví se jen při použití `STM3210C_EVAL`), nastavení vstupních výstupních pinů a mnoho dalších potřebných nastavení. Do této funkce bylo potřeba doplnit inicializaci, spuštění a povolení přerušování od `USART2` rozhraní, které bude použito pro komunikaci po sériové lince.
- `Ethernet_Configuration()` – funkce zajišťující konfiguraci MII respektive RMII pinů. Pokud potřebujeme použít MII rozhraní, musíme na začátku souboru definovat `#DEFINE MII_MODE`, pro rozhraní RMII předchozí definici jednoduše přepíšeme na `#DEFINE RMII_MODE`. V této funkci nebylo třeba dělat žádné úpravy.
- `GPIO_Configuration()` – důležitá funkce zajišťující správné namapování vstupních, výstupních portů. V této funkci nebylo třeba dělat žádné úpravy.
- `ADC_Configuration()` – funkce pro nastavení A/D a D/A převodníků, také nebylo třeba upravovat.
- `NVIC_Configuration()` – funkce pro nastavení vektorů přerušování systémů, původní funkci bylo potřeba rozšířit tak, aby bylo zachytáváno přerušování od `USART2` periférie.

## 5.2.3 Inicializace činnosti převodníku

Jádrem firmware, které zajišťuje funkci převodníku, je funkce `EthRs232Converter_init()`, která je společně s dalšími potřebnými funkcemi umístěna v souboru `ethrs232converter.c`.

Po zvolání této funkce, dojde k naslouchání na portu 23<sup>1</sup> a k nastavení call-back funkce, která je volána při navázání spojení s telnet klientem. Následuje výčet funkcí v souboru `ethrs232converter.c`:

- `TELNET_accept()` – funkce volaná po navázání spojení s telnet klientem, provede naalokování paměti pro TCP komunikaci a nastaví call-back funkci při přijetí paketu a call-back funkci volanou při chybě spojení. Nakonec funkce provede informační výpis na obrazovku telnet klienta.
- `TELNET_data_recv()` – funkce volaná při příjmu paketu na aktivním spojení. Při obdržení dat provede jejich načtení do pomocného bufferu a dále je okamžitě přepošle na sériovou RS-232 linku.
- `Connection_err()` – funkce volaná ve chvíli kdy nastane chyba spojení, uvolní paměť.
- `sendDataOverTCP()` – funkce, která umožňuje posílat data na TCP spojení, její výhodou je v tom, že ji lze volat z jakékoliv části programu, jelikož data posílá vždy na aktivní spojení.

## 5.2.4 Přerušovací systém

Přerušovací systém v podstatě zajišťuje činnost převodníku. Funkce, které jsou volány při přerušení, jsou uloženy v souboru `stm32f10x_it.c`. Nás zajímají především tyto dvě funkce:

- `USART2_IRQHandler()` – funkce volaná při vzniku přerušení od USART2 rozhraní. Nás zajímá především přerušení signalizující, že do datového registru byla přijata data. Tyto data si uložíme do pomocného bufferu a pomocí funkce `sendDataOverTCP()` přepošleme na telnet klienta.
- `ETH_IRQHandler()` – funkce volaná při zachycení přerušení od Ethernet rozhraní, které bývá vyvoláno zachycením paketu. Funkce zajistí, že režie je předána LwIP stacku, který na aktuální situaci reaguje voláním potřebných call-back funkcí, dokud nedojde ke zpracování všech přijatých dat.

## 5.2.5 Hlavní program

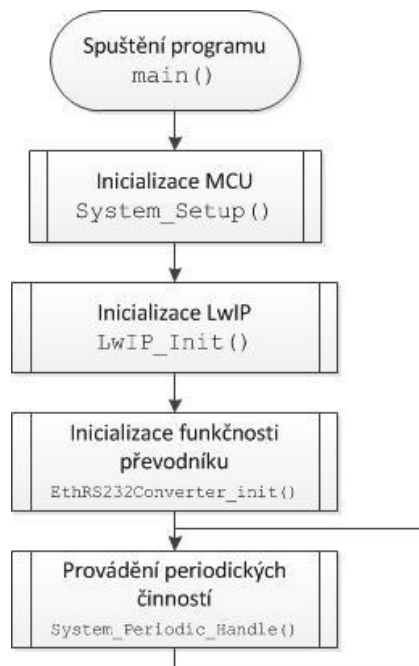
Nyní, když jsme nastavili všechny potřebné funkce, které bude využívat hlavní program, tak se přesuneme k jeho popisu. Hlavní program firmware je umístěn v souboru `main.c`. Vývojový diagram je na obrázku 5.3.

---

<sup>1</sup> Port 23 je používán pro aplikační protokol TELNET.

## Slovní popis k hlavnímu programu

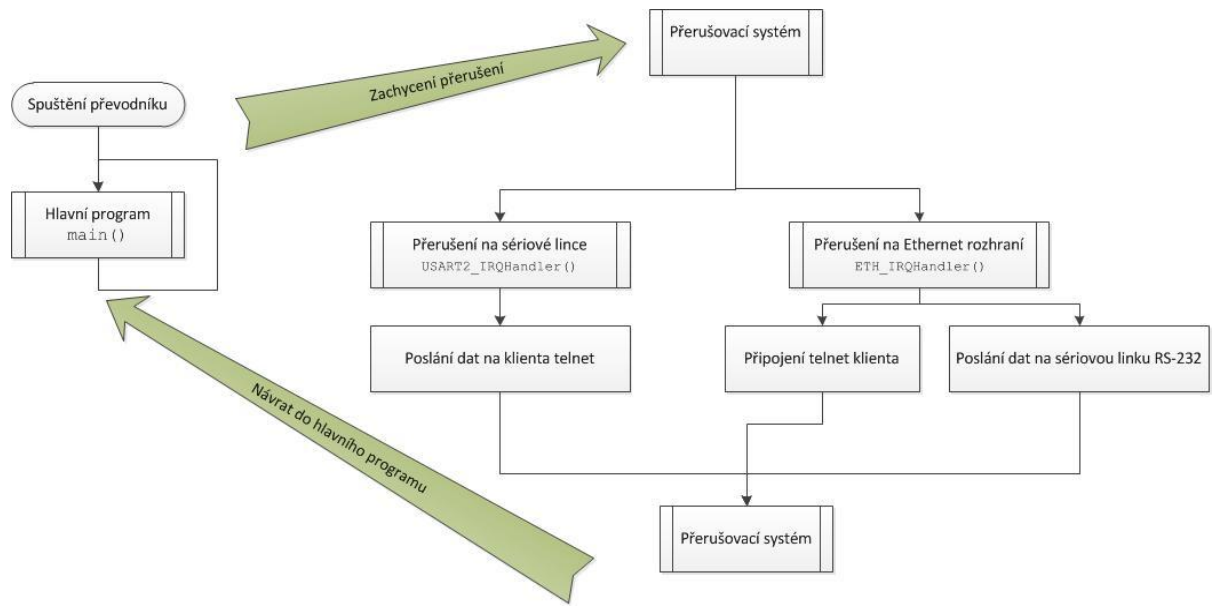
Hlavní program převodníku není příliš složitý, jeho hlavním úkolem je postupná inicializace všech potřebných periférií a jejich nastavení, jak lze vidět na obrázku 5.3. Jakmile dojde k veškerým potřebným nastavením, tak dojde k zacyklení programu, ve kterém se pouze volá funkce zajišťující provádění periodický úkolů v čase, jako aktualizace LED diod, popřípadě displeje a také provádění periodických úloh vyžadovaných pro správnou činnost stacku.



Obrázek 5.3: Vývojový diagram hlavní smyčky programu.

## 5.2.6 Provádění programu

Protože, jsem již představil všechny důležité části firmware, nyní už je jen potřeba vysvětlit, jak to všechno funguje. Činnost aplikace lze pochopit z obrázku 5.4. Vidíme, že po spuštění provádění firmware dojde ke všem potřebným inicializacím a následně program čeká na vzniknutí přerušení, které odchytil přerušovací systém. Pokud nastalo přerušení od Ethernet rozhraní, tak dojde buď k navázání spojení s klientem a nebo pokud je spojení již aktivní, tak došlo k zachycení paketu a přijatá data jsou přeposlaná na sériovou linku. Pokud bylo zachyceno přerušení od USART2 rozhraní, tak byla přijata data na sériové lince a ta jsou přeposlána aktivní spojení s klientem. Tímto jsme dosáhli činnosti, kterou jsme navrhli v kapitole 5.1 a je naznačena na obrázku 5.2.



Obrázek 5.4: Provádění programu.

## 6 Převodník a jeho vlastnosti

V předposlední kapitole si ukážeme použití převodníku v modelových situacích, a co potřebujeme za softwarové a hardwarové nástroje, aby komunikace probíhala správně.

### 6.1 Parametry pro datový přenos

Pro správnou činnost a používání převodníku, je třeba znát jeho parametry pro datový přenos a to především IP adresu převodníku, a dále parametry pro sériový přenos na lince RS-232.

- IP adresa převodníku: statická IPv4 adresa 192.168.0.8.
- Rychlost sériového přenosu: 9600bps.
- Počet datových bitů: 8.
- Počet stop bitů: 1.
- Parita: není.
- Flow control: není.

### 6.2 Zprovoznění převodníku

#### **Telnet klient**

Převodník byl testován a doporučuje se používat s programem Telnet, který je normálně zabudován v OS Windows. Na přiloženém CD je umístěn dávkovací soubor `run.bat`, který provede potřebné spuštění aplikace pod systémem Windows.

#### **RS-232 Terminál**

Jako terminál pro odesílání a přijímání dat ze sériové linky doporučuji použití aplikace Putty a nebo HyperTerminal. Činnost převodníku byla ověřena na obou zmíněných programech, pro správnou funkci je nutné dodržet parametry pro datový přenos uvedené v kapitole 6.1.

#### **Fyzické médium pro Ethernet rozhraní**

Převodník je možné propojovat pouze pomocí UTP kabelu, který je zakončen konektorem RJ-45 a patří do odpovídající kategorie viz kapitola 2.1.5.2.

#### **Fyzické médium pro RS-232 rozhraní**

Vzhledem k tomu, že konektor pro rozhraní RS-232 má na výstupu kolíky, je třeba použít křížený kabel, který má na jedné straně zdířky, druhá strana musí odpovídat zařízení, ke kterému převodník připojujeme. Správná funkčnost bude zajištěna, pokud bude kabel odpovídat normě viz kapitola 2.2.

## Napájení převodníku

Převodník je konstruován pro napájení stabilizovaným zdrojem stejnosměrného napětí o velikosti 5V.

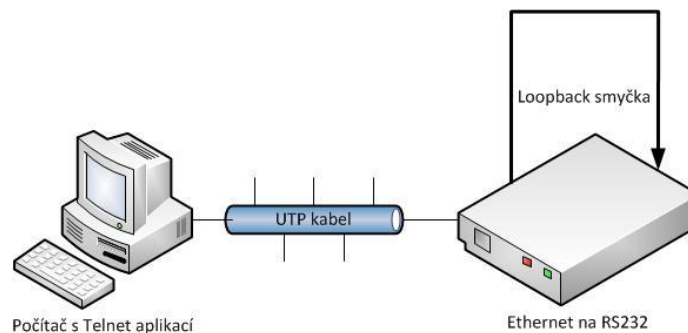
## 6.3 Demonstrace činnosti

Převodník byl otestován v následujících třech situacích.

### Zapojení počítač <-> převodník <-> loopback smyčka

Na obrázku 6.1 lze vidět blokové schéma zapojení, předpoklady pro správnou funkci jsou tyto:

- Počítač s Telnet programem a Ethernet rozhraním.
- UTP kabel pro propojení počítače a převodníku.
- Převodník Ethernet na RS-232.
- Loopback smyčka připojená na rozhraní RS-232 převodníku.



Obrázek 6.1: Zapojení počítač <-> převodník <-> loopback smyčka.

Předpokládané chování je, že jakýkoliv znak vyslán z počítače, bude ihned přeposlán zpět, to znamená, že se bude zdát jako bychom napsali znak v aplikaci telnet napsali dvakrát.

Výstup testu je na obrázku 6.2.

The screenshot shows a Telnet window titled 'Telnet 192.168.0.8'. The output text is as follows:

```
Demonstrace cinnosti prevodniku Ethernet na RS-232
Autor programu: Jiri Dreiseitel xdreis00@stud.fit.vutbr.cz

-----
Program a cely prevodnik vznikl v ramci skolni bakalarske prace
Cinnost: Po navazani spojeni program preposila postupne
vsechny znaky, ktere prijme. Prijata data na Ethernet
rozhrani okamzite preposila na seriovou linku RS-232.
Prijata data na seriove lince okamzite preposila na
navazane spojeni (s programem TELNET).
-----

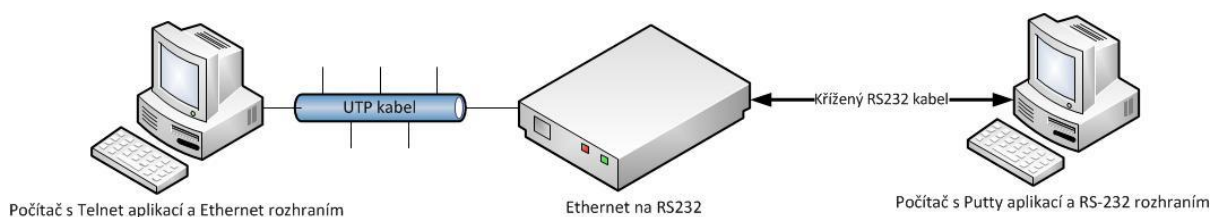
JJsseemm pprriippoojjeenn nnaa lloooppbaacckk ssmmyyckkuu..
TTeesstt ddooppaaddll ddoobbrree..
```

Obrázek 6.2: Výstup testu zapojení podle obrázku 6.1.

## Zapojení počítač <-> převodník <-> počítač

Na obrázku 6.3 lze vidět blokové schéma zapojení, předpoklady pro správnou funkci jsou tyto:

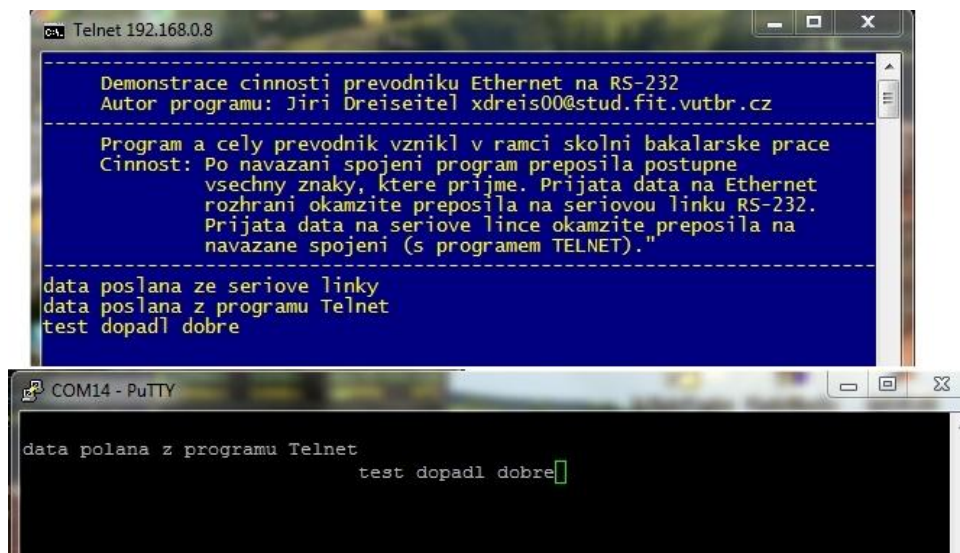
- Počítač s Telnet programem a Ethernet rozhraním.
- UTP kabel pro propojení počítače a převodníku.
- Převodník Ethernet na RS-232.
- Křížený kabel pro propojení převodníku s počítačem.
- Počítač s aplikací Putty a sériovým portem.



Obrázek 6.3: Zapojení počítač <-> převodník <-> počítač

Předpokládané chování je, že bude docházet k transparentnímu přenosu znaků mezi aplikacemi Telnet a Putty. Výstup testu je na obrázku 6.4.

## Zapojení počítač <-> převodník <-> převodník <-> počítač



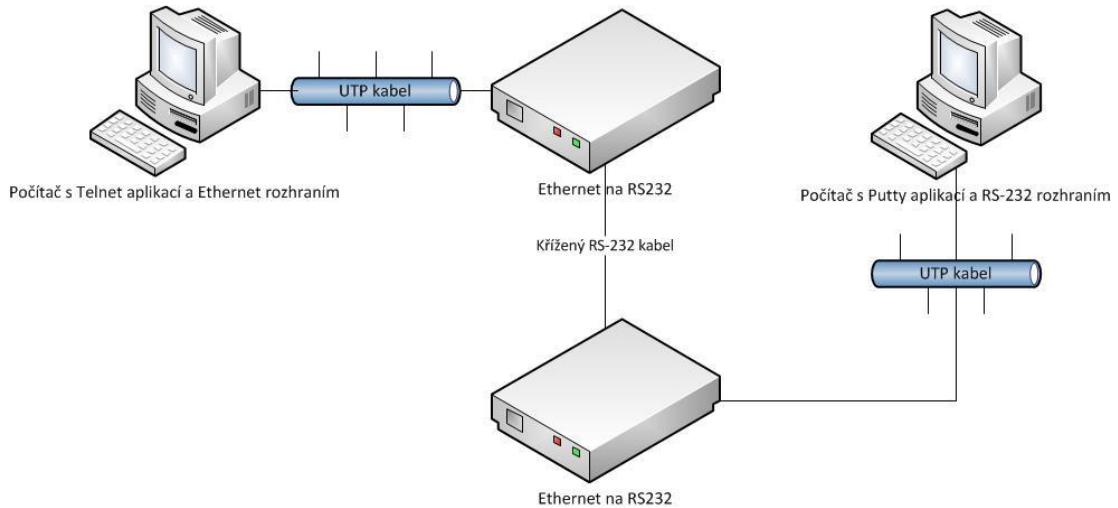
Obrázek 6.4: Výstup testu zapojení podle obrázku 6.3.

Na obrázku 6.5 lze vidět blokové schéma zapojení, předpoklady pro správnou funkci jsou tyto:

- První počítač s Telnet programem a Ethernet rozhraním.
- UTP kabel pro propojení počítače a převodníku.
- První převodník Ethernet na RS-232.

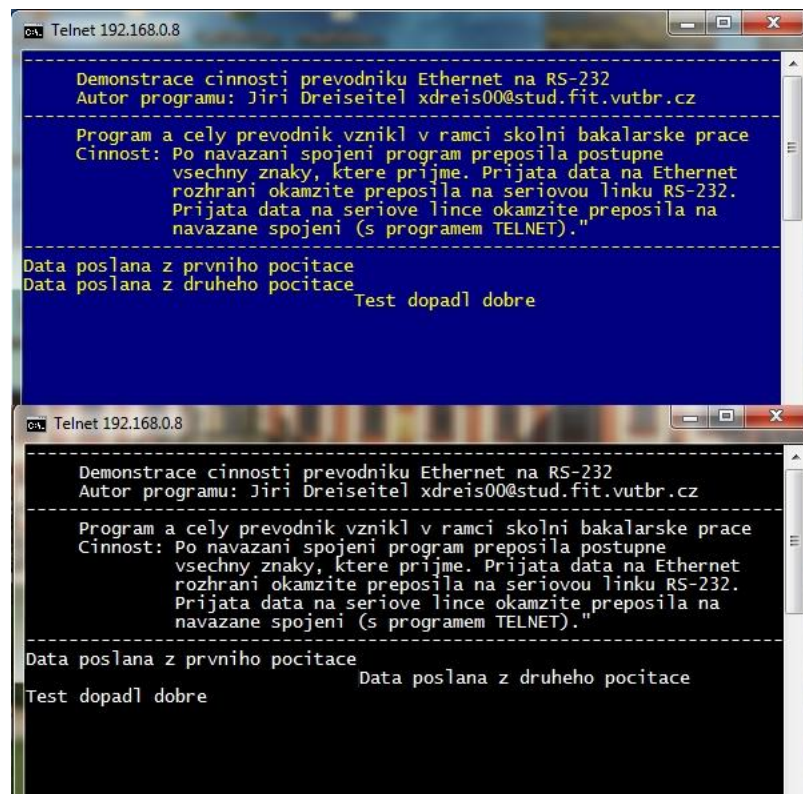


- Křížený kabel pro propojení převodníku s druhým převodníkem.
- Druhý převodník Ethernet na RS-232.
- UTP kabel pro propojení počítače a převodníku.
- Druhý počítač s Telnet programem a Ethernet rozhraním.



Obrázek 6.5: Zapojení počítač <-> převodník <-> převodník <-> počítač

Předpokládané chování je, že bude docházet k transparentnímu přenosu znaků mezi aplikacemi Telnet a Telnet na obou počítačích. Výstup testu je na obrázku 6.6.



Obrázek 6.6: Výstup testu zapojení podle obrázku 6.5.

## 6.4 Elektrické vlastnosti převodníku

Na závěr musíme uvést také některé elektrické charakteristiky. Jak už bylo jednou uvedeno, převodník je konstruován tak, aby mohl být napájen stejnosměrným napětím o velikosti 5V. Při tomto napětí, byl při běžné činnosti zaznamenán odběr elektrického proudu 200mA, což odpovídá spotřebě 1W. Důvodem vyššího odběru je především energetická náročnost PHY obvodu společně s konektorem RJ-45 a dále taky nemalá spotřeba LED diod, které jsou při běžné činnosti rozsvícené.

Při návrhu byla snaha, aby byly součástky měly ochranu proti statické elektřině, ale přesto doporučuji zacházet se zařízením velmi opatrně, jedná se o jednu elektroniku, která může být náchylná na poškození.

## 6.5 Návrh na další rozšíření funkčnosti

Při návrhu převodníku jsem myslel na možnost budoucí rozšíření jeho funkčnosti, proto je také na desce přítomen ještě CAN transiever s konektorem. Převodník by tedy mohl být schopen obsluhovat jak sériové rozhraní RS232, tak rozhraní CAN.

Další možností rozšíření je implementování daleko složitějšího firmware, protože mnou implementovaný firmware má spíše demonstrační a výukové využití, proto by přišla vhod implementace dalších protokolů. Při návrhu složitější funkčnosti by se však nesmělo zapomínat na omezené možnosti MCU a kapacity sériové linky, která je o poznání pomalejší než Ethernet.

## 7 Závěr

V této práci je řešena problematika návrhu a konstrukce převodníku Ethernet na RS-232. Výsledkem práce je funkční návrh i konstrukce převodníku, ve kterém je implementovaný firmware zajišťující základní funkci, kterou od převodníku očekáváme. Výhoda implementovaného firmware je, že byl napsán v přenositelném jazyce C, proto není náročné ho použít na jiné platformě, popřípadě jednoduše rozšířit.

Zadání bylo splněno ve všech bodech, v úvodu práce se čtenář seznámí s problematikou Ethernet, RS-232 a TCP/IP. Poté následuje návrh řešení problematiky a podrobný rozbor konstrukce převodníku s popisem použitých součástí a ukázkou důležitých schématických zapojení. Funkčnost navrženého firmware byla ověřena ve třech modelových situacích, kde výstup těchto testů je v práci uveden.

Pro práci, byl velkým přínosem vývojový kit STM3210C-eval, z něhož jsem vycházel při návrhu převodníku. Dalším důležitým krokem k řešení problematiky bylo použití LwIP stacku společně s demonstračními příklady, díky kterým se výrazně zjednodušila softwarová implementace převodníku.

Výsledný převodník využívá pouze menší procento svého potenciálu, jak jsem uvedl v kapitole 6.5, může se převodník stát podkladem pro mnohem náročnější práci. Proto považuji za vhodné, aby se převodník s implementovaným firmware použil jako podklad pro další bakalářské popřípadě diplomové práce.

# Literatura

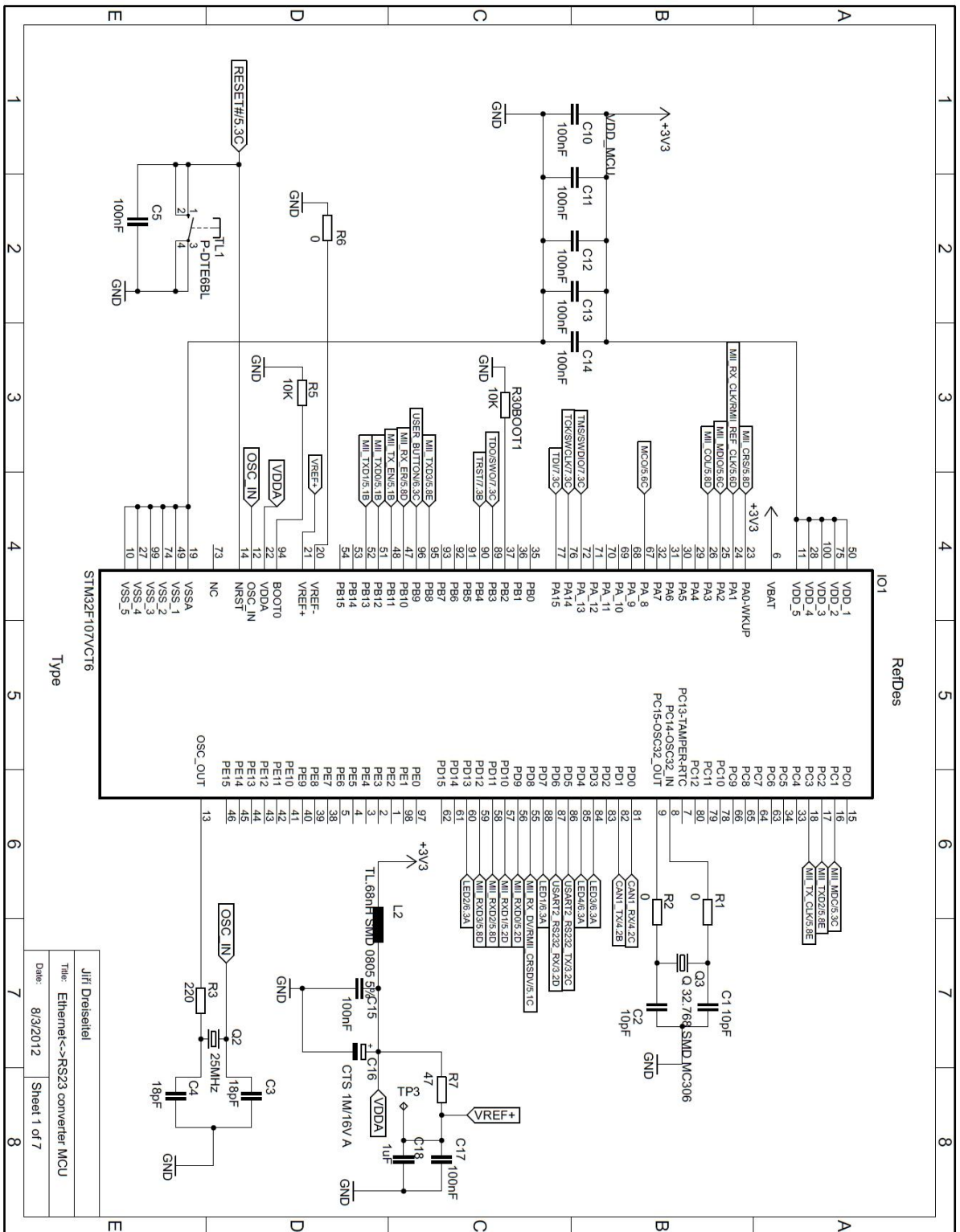
- [1] SPURGEON, C. E. *Ethernet: The Definitive Guide*. 1. vydání. Sebastopol: O'Reilly & Associates, 2000. ISBN 1-56592-660-9.
- [2] HORÁK, J. a M. KERŠLÁGER. *Počítačové sítě pro začínající správce: 3. aktualizované vydání*. Brno: Vydavatelství a nakladatelství Computer Press, a.s., 2006. ISBN 80-251-0892-9.
- [3] GOFTON, P. W. *Sériová komunikace*. Praha: Grada, 1995. ISBN 80-7169-131-3.
- [4] SCHWARZ, J., R. RŮŽIČKA a J. STRNADEL. Mikroprocesorové a vestavěné IMP: Studijní opora. In: [online]. Brno: Leden. 2005 [cit. 2012-Květen-10]. Dostupné z: [https://www.fit.vutbr.cz/study/courses/IMP/private/VYUKA/CVICENI\\_A\\_LABORATORE/imp\\_studijni\\_opora\\_tisk.pdf](https://www.fit.vutbr.cz/study/courses/IMP/private/VYUKA/CVICENI_A_LABORATORE/imp_studijni_opora_tisk.pdf)
- [5] NATIONAL SEMICONDUCTOR. Datasheet DP83848C PHYTER. In: [online]. Květen. 2008 [cit. 2012-Květen-10]. Dostupné z: <http://www.farnell.com/datasheets/90639.pdf>
- [6] ST MICROELECTRONICS. Datasheet STM32F105XX, STM32F107XX. In. *Doc ID 15274 Rev. 6* [online]. © 2011 [cit. 2012-Květen]. Dostupné z: [http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/DATASHEET/CD00220364.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00220364.pdf)
- [7] MAXIM INTEGRATED PRODUCTS. Datasheet MAX3221E/MAX3223E/MAX3243E. In: [online]. Zář. 2005 [cit. 2012-Květen-10]. Dostupné z: <http://www.farnell.com/datasheets/58519.pdf>
- [8] TEXAS INSTRUMENTS INCORPORATED. Datasheet SN65HVD230, SN65HVD231, SN65HVD232. In. *Datasheet SN65HVD230, SN65HVD231, SN65HVD232* [online]. Březen. 2011 [cit. 2012-Květen-10]. Dostupné z: <http://www.farnell.com/datasheets/1446397.pdf>
- [9] PUŽMANOVÁ, R. *TCP/IP v kostce*. 1. České Budějovice: KOPP, 2004. ISBN 80-7232-236-2.
- [10] ABEL, M. *Pločné spoje se SMD, návrh a konstrukce*. 1. Pardubice: Nakladatelství Platan, 2000. ISBN 80-902733-2-7.
- [11] BAŘTIPÁN, A. *Moduly s rozhraním ethernet pro systém sběru dat*. Praha: České vysoké učení technické, Fakulta elektrotechnická, Katedra měření, 2011
- [12] FK TECHNICS. *Mikrokontroléry STM32-Cortex M3* [online]. 9. Listopad. 2009. Dostupné z: [http://www.fktechnics.cz/cz/clanky/soucastky/art\\_132/mikrokontrolery-stm32-cortex-m3.aspx](http://www.fktechnics.cz/cz/clanky/soucastky/art_132/mikrokontrolery-stm32-cortex-m3.aspx)
- [13] OLMR, V. *HW server představuje - Sériová linka RS-232* [online]. 12. Prosinec. 2005 [cit. 2012-Květen-10]. Dostupné z: <http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html>
- [14] PETERKA, J. *eArchiv.cz archiv článků a přednášek Jiřího Peterky: Základní formy přenosů* [online]. © 2011 [cit. 2012-Květen-10]. Dostupné z: <http://www.earchiv.cz/a91/a140c110.php3>

- [15] ST MICROELECTRONICS. AN3102 Application note lwIP TCP/IP stack demonstration for STM32F107xx connectivity line microcontrollers. In: [online]. Listopad. 2009 [cit. 2012-Květen-10]. Dostupné z: [http://www.st.com/internet/com/TECHNICAL\\_RESOURCES/TECHNICAL\\_LITERATURE/APPLICATION\\_NOTE/CD00255062.pdf](http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/CD00255062.pdf)

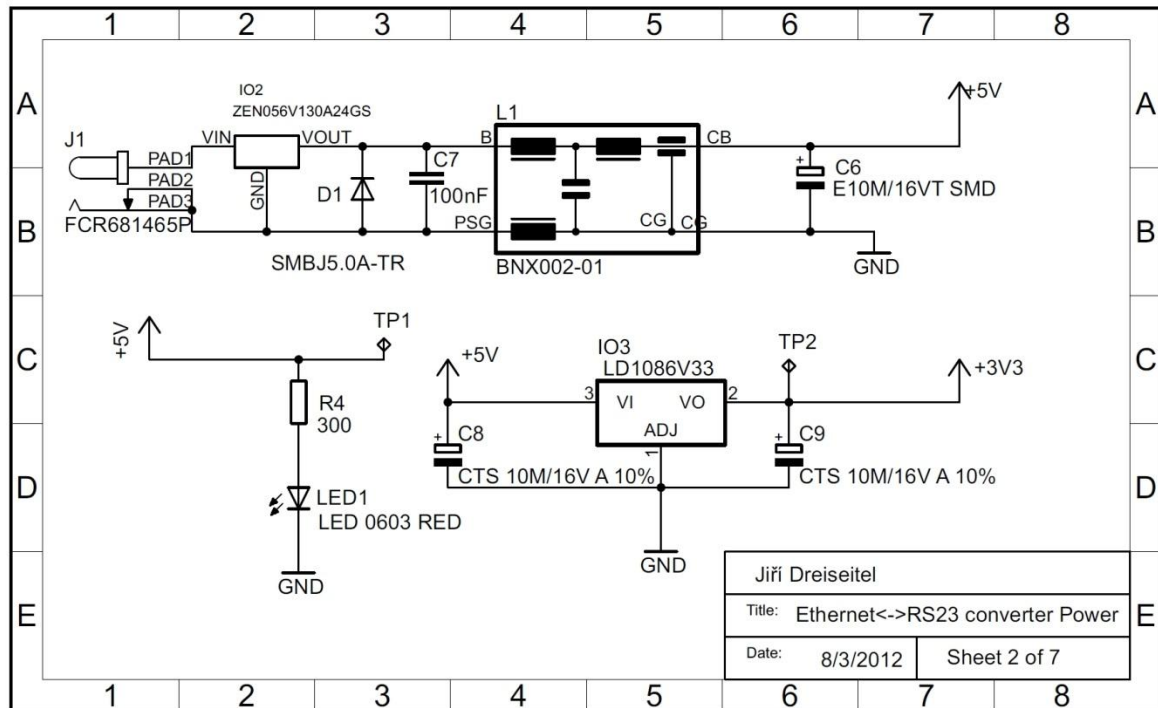
# Seznam příloh

- Příloha 1. Schéma zapojení MCU.
- Příloha 2. Schéma zapojení napájecího zdroje.
- Příloha 3. Schéma zapojení RS-232 rozhraní.
- Příloha 4. Schéma zapojení CAN rozhraní.
- Příloha 5. Schéma zapojení LED diod a uživatelského tlačítka.
- Příloha 6. Schéma zapojení PHY.
- Příloha 7. Schéma zapojení JTAG konektoru.
- Příloha 8. Návrh desky plošného spoje.
- Příloha 9: Rozmístění součástek na DPS
- Příloha 10. Seznam součástek.
- Příloha 11. CD se zdrojovými kódy, obrázky a návrhem DPS v elektronické podobě.

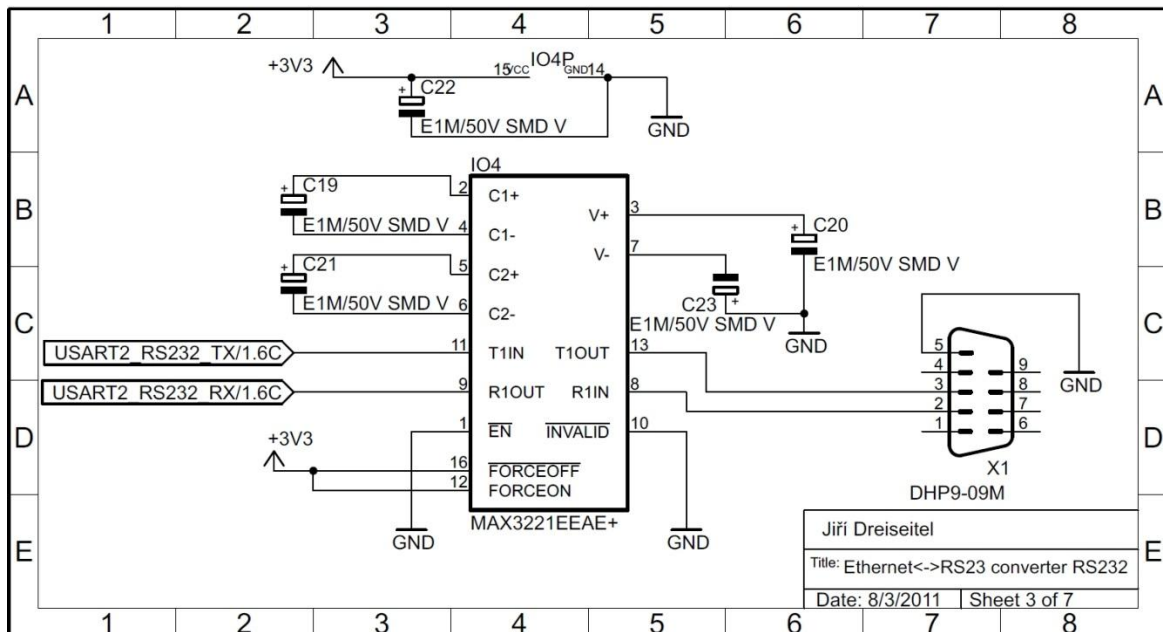
# Príloha 1. Schéma zapojení MCU



## Příloha 2. Schéma zapojení napájecího zdroje

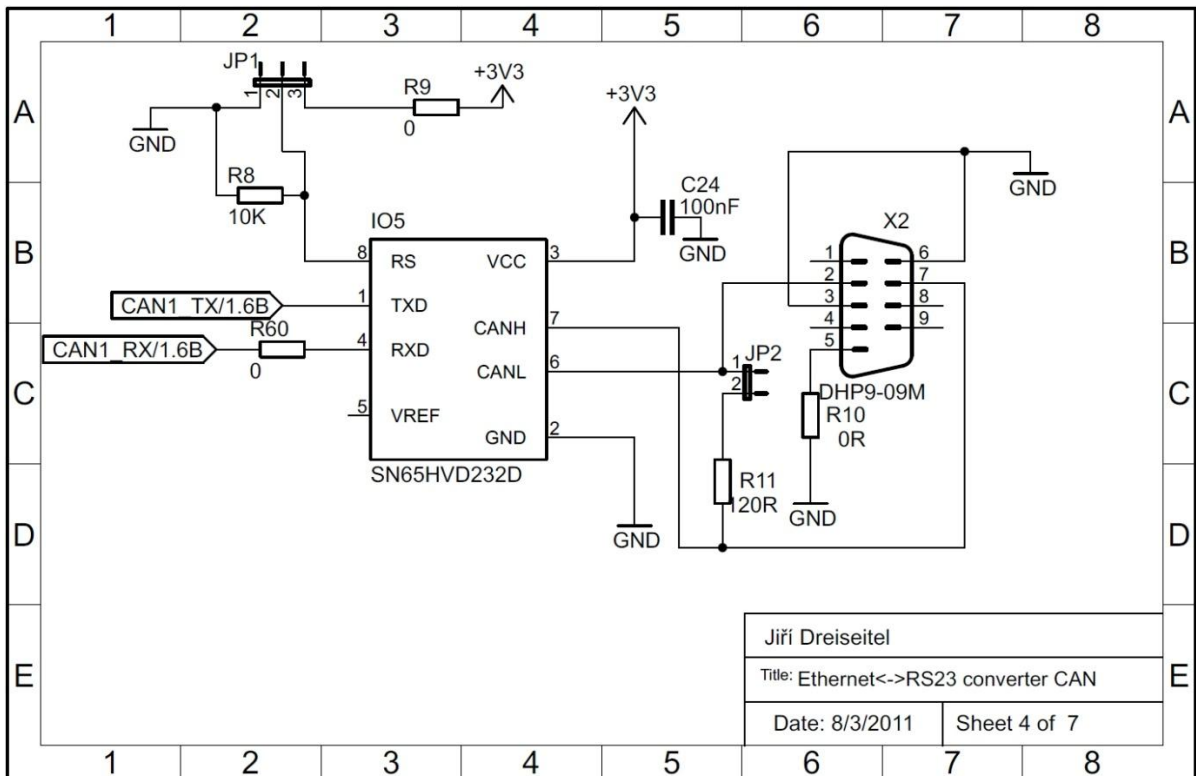


## Příloha 3. Schéma zapojení RS-232 rozhraní

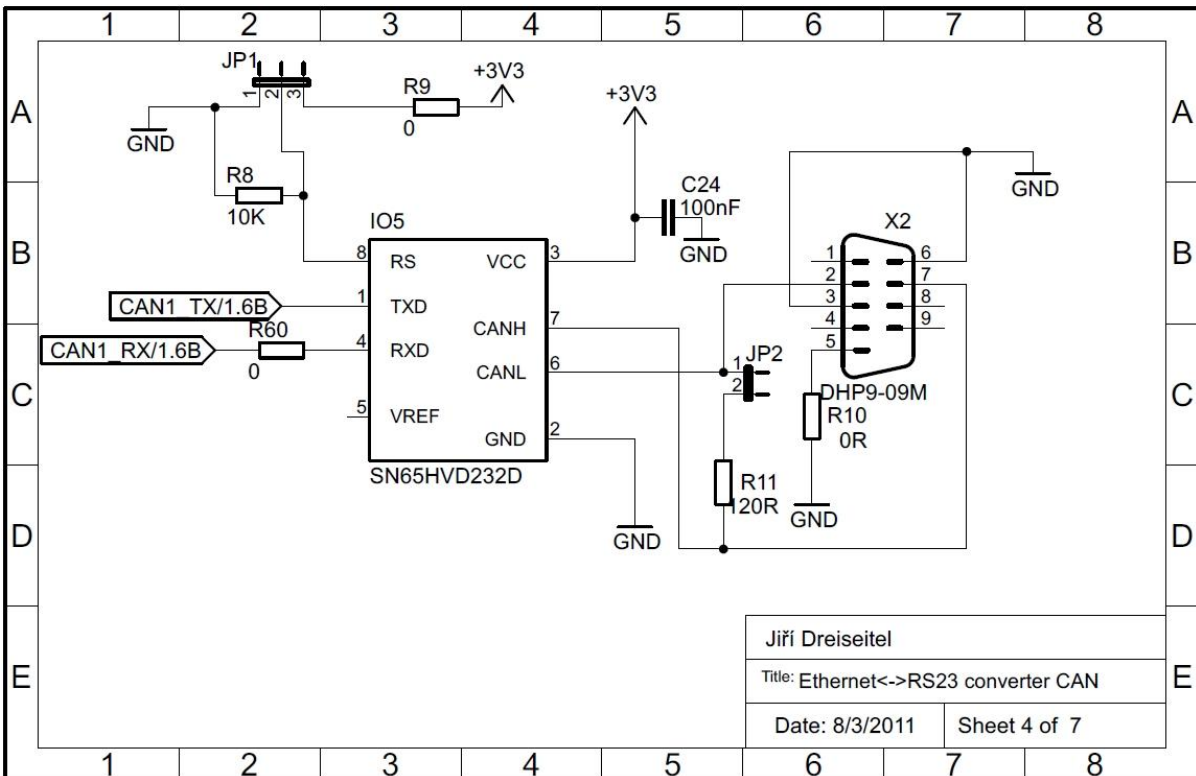




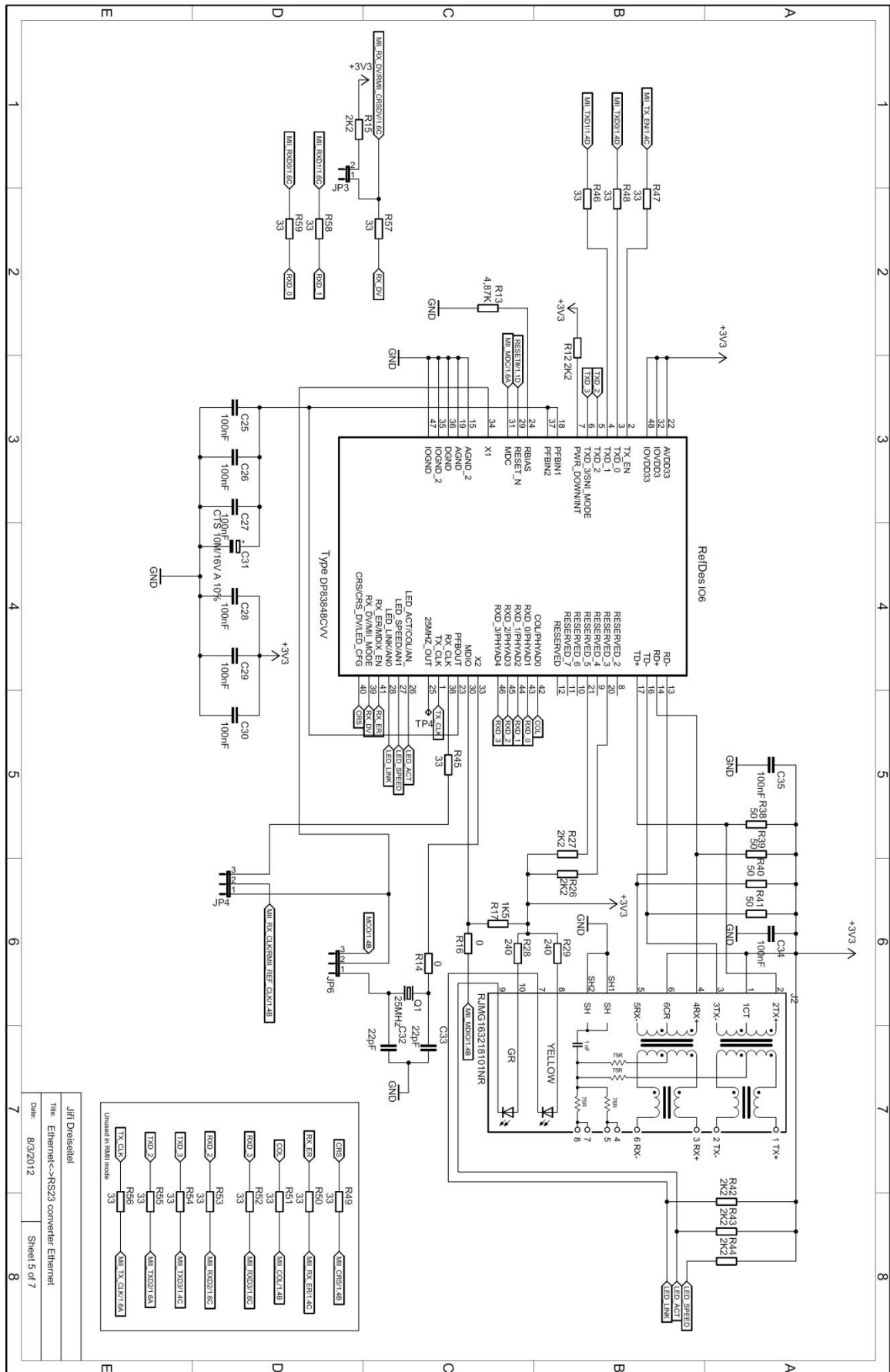
## Příloha 4. Schéma zapojení CAN rozhraní



## Příloha 5. Schéma zapojení LED diod a uživatelského tlačítka

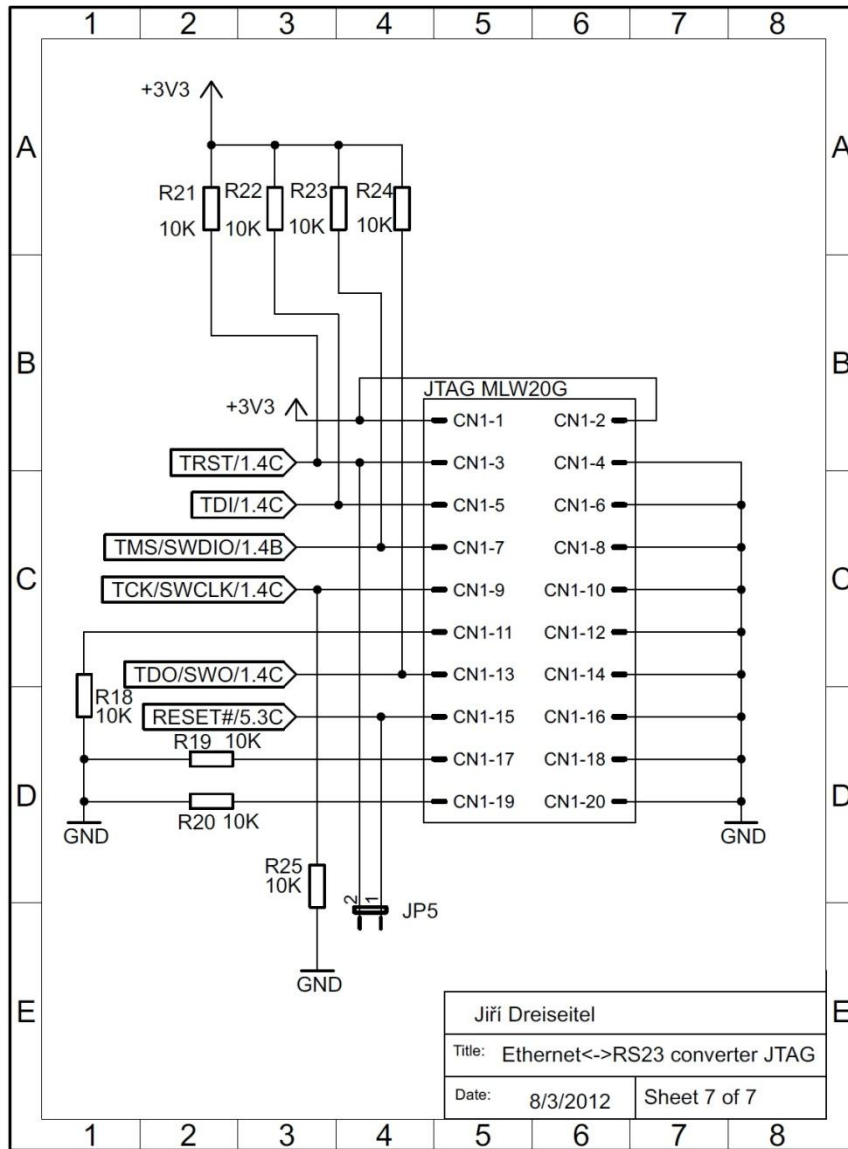


# Příloha 6. Schéma zapojení PHY

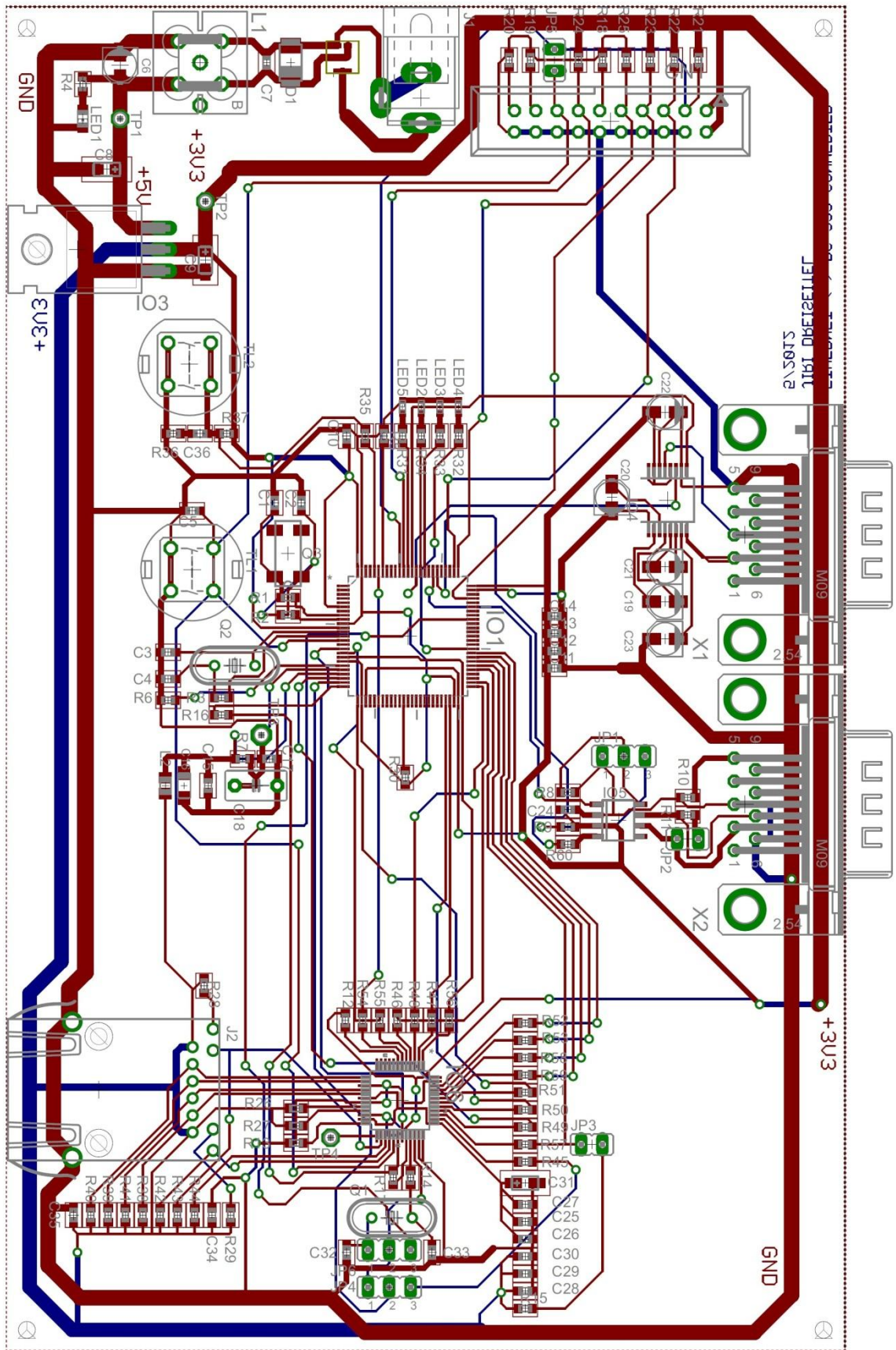


Jiří Dvořák  
 Title: Ethernet<->RS232 converter Ethernet  
 Date: 8/3/2012  
 Sheet 5 of 7

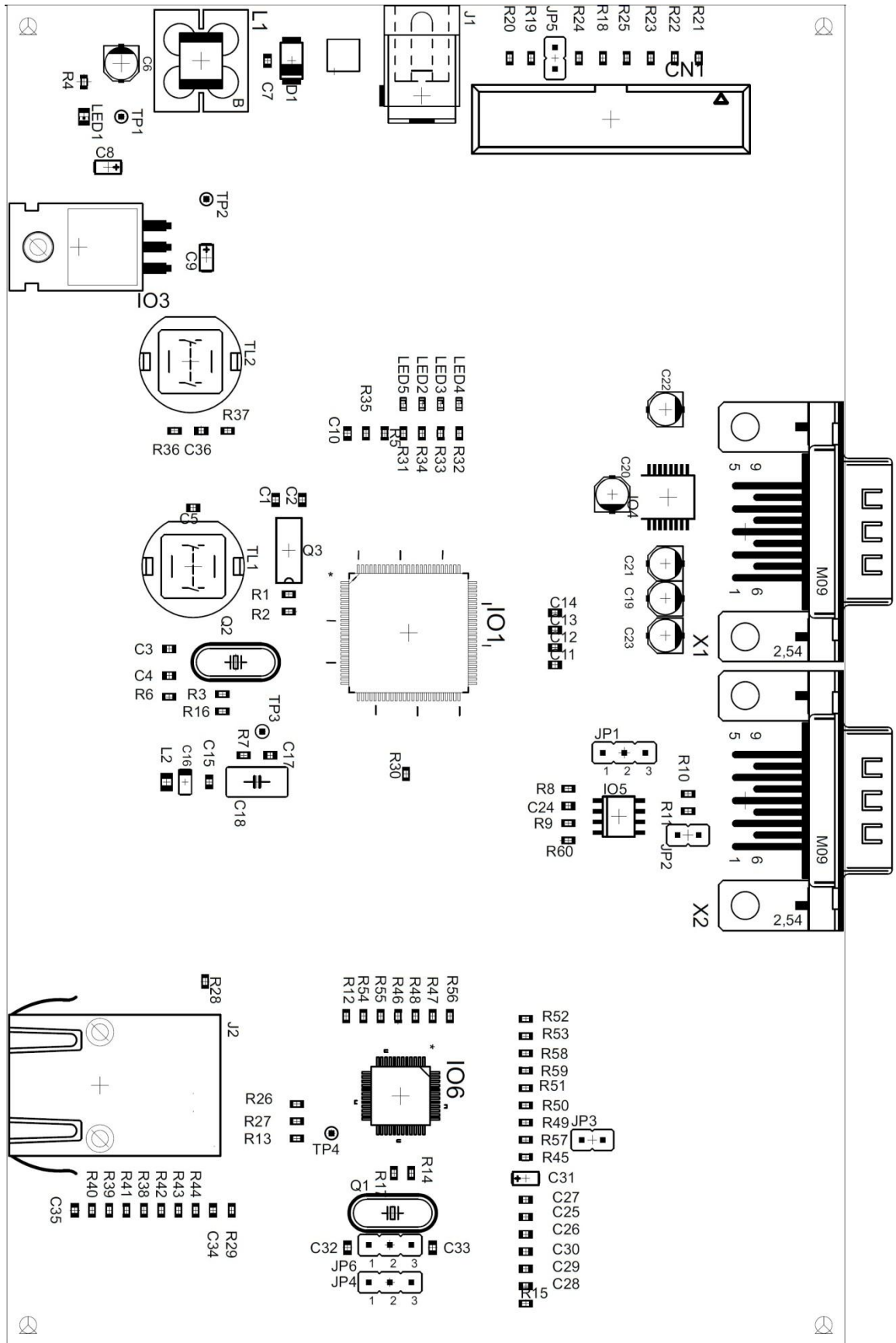
# Příloha 7. Schéma zapojení JTAG konektoru



# Příloha 8. Návrh desky plošného spoje.



# Příloha 9. Seznam součástek



## Příloha 10. Seznam součástek

Označení	Hodnota	Název součástky
C1	10pF	CK0603 10P/50V NPO
C2	10pF	CK0603 10P/50V NPO
C3	18pF	CK0603 18P/50V NPO
C4	18pF	CK0603 18P/50V NPO
C5	100nF	CK0603 100N/50V X7R
C6	10uF	E10M/16VT SMD
C7	100nF	CK0603 100N/50V X7R
C8	10uF	CTS 10M/16V A 10%
C9	10uF	CTS 10M/16V A 10%
C10	100nF	CK0603 100N/50V X7R
C11	100nF	CK0603 100N/50V X7R
C12	100nF	CK0603 100N/50V X7R
C13	100nF	CK0603 100N/50V X7R
C14	100nF	CK0603 100N/50V X7R
C15	100nF	CK0603 100N/50V X7R
C16	1uF	CTS 1M/16V A 20%
C17	100nF	CK0603 100N/50V X7R
C18	1uF	CK 1M/50V
C19	1uF	E1M/50V SMD V
C20	1uF	E1M/50V SMD V
C21	1uF	E1M/50V SMD V
C22	1uF	E1M/50V SMD V
C23	1uF	E1M/50V SMD V
C24	100nF	CK0603 100N/50V X7R
C25	100nF	CK0603 100N/50V X7R
C26	100nF	CK0603 100N/50V X7R
C27	100nF	CK0603 100N/50V X7R
C28	100nF	CK0603 100N/50V X7R
C29	100nF	CK0603 100N/50V X7R
C30	100nF	CK0603 100N/50V X7R
C31	10uF	CTS 10M/16V A 10%
C32	22pF	CK0603 22P/50V NPO
C33	22pF	CK0603 22P/50V NPO
C34	100nF	CK0603 100N/50V X7R
C35	100nF	CK0603 100N/50V X7R
C36	100nF	CK0603 100N/50V X7R
CN1	Konektor pro ploché kabely do DPS přímý, rozteč 2,54mm, 2x10	Konektor MLW20G
D1	DIODE, TVS, 5V, 600W	SMBJ5.0A-TR - DIODE
IO1	MCU, 32B, CORTEXM3, 256K FLASH, 100LQFP	STM32F107VCT6
IO2	DIODE, ZENER, 5.6V, 0.7W, 1.3A, SMD	ZEN056V130A24GS
IO3	Stabilizátor napětí: LDO, nenastavitelný, 1,5A; 3,3V; TO220	LD1086V33
IO4	MAX3221EEAE+ - TRANSCEIVE	MAX3221EEAE+
IO5	CAN TRANSCEIVER, 65HVD232, SOIC8	SN65HVD232D
IO6	10/100 ETHERNET PHY, SMD, LQFP48	DP83848CVV
J1	FCR681465P - DC SKT, DUAL, 2,1/2.5MM, PCB, DC10L	FCR681465P
J2	Pulse Jack RJ45	AMPHENOL - RJMG163218101NR - JACK, 10/100 BASE T, ROHS5
JP1	Oboustranný kolík S1G20 2,54mm	S1G20 2,54mm
JP2	Oboustranný kolík S1G20 2,54mm	S1G20 2,54mm
JP3	Oboustranný kolík S1G20 2,54mm	S1G20 2,54mm
JP4	Oboustranný kolík S1G20 2,54mm	S1G20 2,54mm
JP5	Oboustranný kolík S1G20 2,54mm	S1G20 2,54mm
JP6	Oboustranný kolík S1G20 2,54mm	S1G20 2,54mm
L1	Odrušovací filtr	MURATA BNX002-01
L2	SMD Tlumivka 68nH	TL 68nH SMD 0805 5%
LED1	LED SMD 0603 červená; 130°, 6 mcd; 2 V; 10 mA	LED 0603 RED
LED2	LED SMD 0603 žlutozelená; 130°, 6 mcd; 2,1 V; 10 mA	LED 0603 GREEN
LED3	LED SMD 0603 žlutá; 130°, 60 mcd; 2 V; 10 mA	LED 0603 YELLOW 60/130°
LED4	LED SMD 0603 červená; 130°, 6 mcd; 2 V; 10 mA	LED 0603 RED
LED5	LED SMD 0603 modrá; 110°, 4,5 mcd; 3,8 V; 20 mA	LED 0603 BLUE
Q1	25MHz oscilátor	QM 25.000MHz
Q2	25MHz oscilátor	QM 25.000MHz
Q3	32,768KHz krystal	Q 32.768 SMD MC306
R1	0R	R0603 0R0 5%
R2	0R	R0603 0R0 5%
R3	220R	R0603 220R 1%
R4	300R	R0603 300R 5%
R5	10K	R0603 10K 1%
R6	0R	R0603 0R0 5%
R7	47R	R0603 47R 1%
R8	10K	R0603 10K 1%
R9	0R	R0603 0R0 5%
R10	0R	R0603 0R0 5%
R11	120R	R0603 120R 1%
R12	2K2	R0603 2K2 1%
R13	4K87	WELWYN - PCF0603R-4K87BT1
R14	0R	R0603 0R0 5%
R15	2K2	R0603 2K2 1%
R16	0R	R0603 0R0 5%
R17	1K5	R0603 1K5 1%
R18	10K	R0603 10K 1%
R19	10K	R0603 10K 1%
R20	10K	R0603 10K 1%

R21	10K	R0603 10K 1%
R22	10K	R0603 10K 1%
R23	10K	R0603 10K 1%
R24	10K	R0603 10K 1%
R25	10K	R0603 10K 1%
R26	2K2	R0603 2K2 1%
R27	2K2	R0603 2K2 1%
R28	240R	R0603 240R 1%
R29	240R	R0603 240R 1%
R30	10K	R0603 10K 1%
R31	Rezistor SMD 0603 68R Ohm	R0603 68R 1%
R32	Rezistor SMD 0603 300R Ohm	R0603 300R 5%
R33	Rezistor SMD 0603 300R Ohm	R0603 300R 5%
R34	Rezistor SMD 0603 300R Ohm	R0603 300R 5%
R35	0R	R0603 0R0 5%
R36	100R	R0603 100R 1%
R37	4K7	R0603 4K7 1%
R38	50R	PAT0603E50R5BST5
R39	50R	PAT0603E50R5BST5
R40	50R	PAT0603E50R5BST5
R41	50R	PAT0603E50R5BST5
R42	2K2	R0603 2K2 1%
R43	2K2	R0603 2K2 1%
R44	2K2	R0603 2K2 1%
R45	33R	R0603 33R 1%
R46	33R	R0603 33R 1%
R47	33R	R0603 33R 1%
R48	33R	R0603 33R 1%
R49	33R	R0603 33R 1%
R50	33R	R0603 33R 1%
R51	33R	R0603 33R 1%
R52	33R	R0603 33R 1%
R53	33R	R0603 33R 1%
R54	33R	R0603 33R 1%
R55	33R	R0603 33R 1%
R56	33R	R0603 33R 1%
R57	33R	R0603 33R 1%
R58	33R	R0603 33R 1%
R59	33R	R0603 33R 1%
R60	0R	R0603 0R0 5%
TL1	Tlačítkový spínač bez aretace do DPS 1-pólový spínací ON-OFF.	P-DTE6BL
TL2	Tlačítkový spínač bez aretace do DPS 1-pólový spínací ON-OFF.	P-DTE6BL
X1	Konektor: D-Sub; vidlice, úhlový,	TE CONNECTIVITY / AMP - 3-1634580-2 - PLUG, 9WAY
X2	Konektor: D-Sub; vidlice; úhlový,	TE CONNECTIVITY / AMP - 3-1634580-2 - PLUG, 9WAY