

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SNIFFER SBĚRNICE CAN

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB SKALICKÝ

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SNIFFER SBĚRNICE CAN

CAN SNIFFER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB SKALICKÝ

VEDOUcí PRÁCE
SUPERVISOR

Ing. FRANTIŠEK BURIAN

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Jakub Skalický

Ročník: 3

ID: 125628

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Sniffer sběrnice CAN

POKYNY PRO VYPRACOVÁNÍ:

Rozšiřte OpenSource modul realizující funkci odposlechu CAN sběrnice. GIT repozitář projektu se nachází na adrese <https://github.com/uamt-brno/canshark>, tento projekt převezměte a rozšiřte o nové vlastnosti.

Modul musí splňovat následující parametry:

- 1) Modul musí být schopen příjmu všech zpráv na dvou linkách sběrnice CAN bez výpadku při plné rychlosti 1Mbps.
- 2) Pro přenos dat do počítače a konfiguraci je použito ethernetového rozhraní na modulu.
- 3) Modul musí být umístěn do vhodné krabičky, s vhodnými konektory pro jednoduché připojení sběrnice CAN v terénu.
- 4) Program v počítači (C#) umožňuje základní spojení s Wiresharkem přes pipe i přes soubory. Vylepšete GUI tohoto programu.

DOPORUČENÁ LITERATURA:

ARM architecture reference manual. Harlow: Addison-Wesley, 2001. 2. ed. ISBN 978-020-1737-196.
BROWN, Geoffrey. Discovering the STM32 Microcontroller. [online]. 2013, s. 244 [cit. 2013-05-21].
Dostupné z: <http://www.cs.indiana.edu/~geobrown/book.pdf>

Termín zadání: 10.2.2014

Termín odevzdání: 26.5.2014

Vedoucí práce: Ing. František Burian

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

ABSTRAKT

Tato bakalářská práce se zabývá návrhem snifferu sběrnice CAN odesílajícím data přes rozhraní Ethernet. Navržená deska plošných spojů je realizována za použití mikrokontroléru STM32F407VGT6 firmy STMicroelectronics a Ethernet PHY KSZ8051MLL. V teoretické části je uveden krátký přehled ARM a použitého mikroprocesoru. Dále je kapitola s informacemi o protokolu Ethernet a UDP. Poté je popsán princip sběrnice CAN. Praktická část se zabývá návrhem hardware včetně jeho oživení a změření jeho parametrů. Poslední kapitola popisuje úpravy provedené v software. Nejdříve jsou popsány úpravy firmware pro desku CanShark, poté je popsána nová funkcionality přidaná do PC aplikace canshark-gui.

KLÍČOVÁ SLOVA

ARM, STM32, STM32F4, KSZ8051MLL, Ethernet, UDP, CAN, Sniffer, CanShark, Wireshark

ABSTRACT

This bachelor thesis deals with creating CAN Bus sniffer with sending data to PC by Ethernet. The main parts of developed PCB are microcontroller STM32F407VGT6 by STMicroelectronics and Ethernet PHY KSZ8051MLL. In theoretical part you can find ARM and STM32F407VGT6 information followed by Ethernet and UDP protocol introduction. The end of theoretical part is about CAN bus.

Practical part is about design of board, developing it and measuring parameters of board. The end part of this thesis describes changes of software. Firstly there are described changes of firmware for add support of CanShark board. Followed is described adding of new functionality of canshark-gui PC application.

KEYWORDS

ARM, STM32, STM32F4, KSZ8051MLL, Ethernet, UDP, CAN, Sniffer, CanShark, Wireshark

SKALICKÝ, Jakub *Sniffer sběrnice CAN*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2014. 66 s. Vedoucí práce byl Ing. František Burian,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Sniffer sběrnice CAN“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Františku Burianovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

1	Úvod	11
2	Architektura ARM a mikroprocesor STM32F407VG	12
2.1	CORTEX M4	12
2.2	STM32F407VG	13
2.2.1	Ethernet periferie	13
2.2.2	CAN periferie	13
2.3	Vývojová prostředí	13
3	Ethernet	17
3.1	Ethernetová komunikace z pohledu ISO/OSI	17
3.2	Ethernetový rámec	18
3.3	UDP protokol	19
3.4	ARM a Ethernet	19
3.4.1	MII-Media Independent Interface	20
3.4.2	RMII-Reduced MII	20
3.4.3	MIIM - MII Management Interface	21
4	Sběrnice CAN	22
4.1	Fyzická vrstva CAN	22
4.1.1	Časování sběrnice CAN	22
4.2	Linková vrstva sběrnice CAN	23
4.2.1	Řízení přístupu k médium (MAC)	23
4.2.2	Formáty datových rámců na sběrnici CAN	23
4.3	Ostatní vrstvy sběrnice CAN	25
4.4	Zabezpečení dat na sběrnici CAN	25
5	Návrh hardware	26
5.1	MCU	26
5.1.1	Nahrávání a ladění programu	27
5.1.2	Napájení STM32F407	29
5.2	CAN1 a CAN2	29
5.2.1	Volba módu CAN budiče	29
5.2.2	Volba vhodných konektorů	30
5.3	Ethernet	30
5.3.1	Napájení KSZ8051MLL	31
5.3.2	Zdroj hodinového signálu	31
5.3.3	MII Interface	32

5.3.4	Připojení Ethernet konektoru	32
5.3.5	Ostatní signály	32
5.4	Analogová část	33
5.5	PWR	34
5.5.1	Zdroj 3.3V	34
5.5.2	Zdroj pro analogovou část	36
5.6	Návrh desky plošných spojů	37
6	Výroba a oživení DPS	38
6.1	Testování zapojení	38
6.1.1	Testování s pomocí NAND Tree	39
7	Meření parametrů desky	40
7.1	Měření napájecích zdrojů	40
7.2	Měření analogové části	41
7.3	Zkouška příjmu a odesílání zpráv	42
8	Návrh software	44
8.1	Firmware pro CanShark board	44
8.2	Software pro PC	44
8.2.1	Aktuální verze	44
8.2.2	Provedené úpravy	44
8.2.3	Návrh komponenty Histogram	46
8.3	Vývojové a testovací utility	47
8.3.1	CANBoardUDSimulator	47
8.3.2	Board Tester	48
8.3.3	Board Analog Tester	48
9	Závěr	49
	Literatura	51
	Seznam symbolů, veličin a zkratk	53
	Seznam příloh	54
A	Schéma zapojení navržené desky	55
B	Vrstva TOP DPS a její osazovací plán	60
C	Vrstva Bottom DPS a její osazovací plán	62

D Fotky navržené desky CanShark	64
E Obsah přiloženého CD	66

SEZNAM OBRÁZKŮ

2.1	Blokové schéma jádra Cortex-M4 [8]	12
2.2	Blokové schéma procesoru STM32F407VGT6 [9]	14
2.3	Blokové schéma periferie Ethernet procesoru STM32F407 [4]	15
2.4	Blokové schéma periferie CAN procesoru STM32F407 [4]	16
3.1	Ethernet frame	18
3.2	Hlavička UDP protokolu	19
5.1	Blokové schéma desky CanShark	26
5.2	Configurační program STM32CubeMX	27
5.3	Přehled napájení pro KSZ8051MLL [2]	31
5.4	Propojení LED ethernet konektoru s PHY [2]	33
7.1	Změřené zvlnění napájecího napětí 3,3V	40
7.2	Změřené zvlnění napájecího napětí 10V	41
7.3	Vstupní pilovité napětí a výstup triggeru analogové části	42
7.4	Zapojení při zkoušce přenosu zpráv maximální rychlostí	43
8.1	Okno programu canshark-gui před provedením úprav	45
8.2	Okno programu canshark-gui po provedených úpravách	46
8.3	Program CanShark simulator	47

SEZNAM TABULEK

3.1	Signály rozhraní MII	20
5.1	Zapojení SWD konektoru	28
5.2	Piny použitelné pro interní bootloader	28
7.1	Naměřené a vypočtené hodnoty pro napájecí zdroje	40
7.2	Naměřené a vypočtené hodnoty pro napájecí zdroje	41
8.1	Parametry metody ChangeDeleteMode	45
8.2	Metody třídy CanBusHistogram poskytované pro komunikaci s okolím	46

1 ÚVOD

Cílem této práce je rozšířit OpenSource modul realizující odposlech sběrnice CAN. Výchozí projekt, který tato práce rozšiřuje, je umístěn na adrese <https://github.com/uamt-brno/canshark>. Cílem tohoto projektu je vytvořit kompletní (jak HW tak SW) vybavení pro odposlech a analýzu v současné době velice rozšířené sběrnice CAN.

Cíl této bakalářské práce lze rozdělit na dvě základní části. První částí je návrh desky plošných spojů, která bude kompatibilní s výše uvedeným modulem. Musí umožnit odposlouchávat dvě sběrnice CAN vysílající na rychlosti až 1Mbps a odesílat přijatá data UDP pakety přes síť ethernet do PC. V druhé části je cílem upravit existující firmware pro správnou funkčnost s navrženou deskou a rozšířit aplikaci canshark-gui, která je určená pro analýzu provozu na síti.

Na začátku této práce je popsán mikroprocesor STM32F407VGT6 a základní informace pro seznámení se s problematikou rozhraní Ethernet a CAN, použitých v této práci.

Hlavním úkolem této práce je návrh hardware, který umožní odposlech sběrnice CAN. Popis návrhu desky plošných spojů obsahuje postup zapojení hlavních součástek (mikroprocesor, ethernet PHY, budiče CAN, napájecí zdroje) a volbu součástek nutných pro jejich správnou funkci.

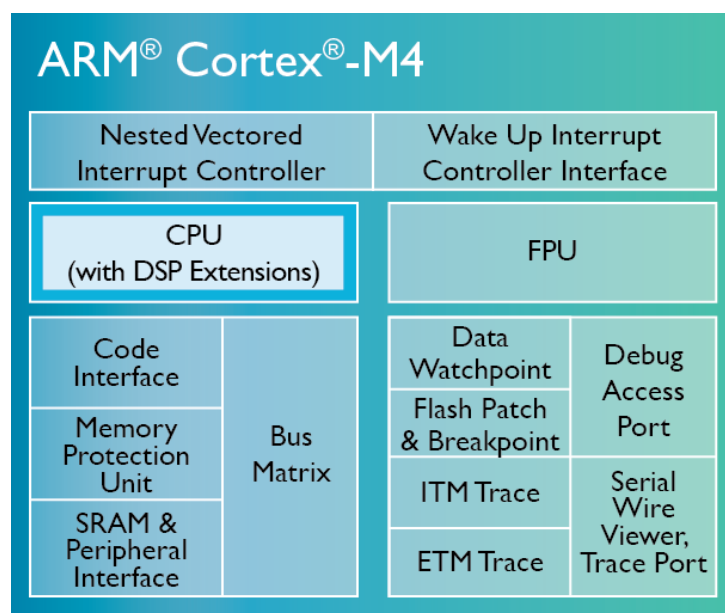
Druhou stěžejní částí je úprava v modulu již obsaženého firmware a rozšíření software pro PC o nové funkce. Úpravy firmware musí zohlednit navržené zapojení desky plošných spojů tak, aby výsledný program umožňoval správnou funkčnost i na nové desce, pro kterou nebyl původně navržen. Úpravy software pro PC mají za úkol rozšířit jeho funkčnost analýzy a logování sběrnice CAN o nové možnosti. Podrobněji o daném tématu pojednává závěrečná kapitola.

2 ARCHITEKTURA ARM A MIKROPROCESOR STM32F407VG

Architektura ARM (Advanced RISC Machine, dříve Acorn RISC Machine) vyvíjená firmou ARM Holding je jednou z nejrozšířenějších procesorových architektur. Jedná se o 32-bitovou RISC architekturu vyznačující se nízkou spotřebou v poměru k vysokému výpočetnímu výkonu. To spolu s možností výrobců licencovat si od ARM Holding návrh jádra a implementovat ho s vlastními periferiemi do SOC dělá z procesorů s архитектурou ARM jednu ze světově nejrozšířenějších architektur. Je možné ji najít jak ve spotřební elektronice (mobilní telefony, routery, tablety atd.) tak v průmyslové oblasti řízení strojů, sběru a zpracování dat atd [8].

2.1 CORTEX M4

Cortex M4 je v současné době nejnovější jádro navržené společností ARM pro embedded procesory.



Obr. 2.1: Blokové schéma jádra Cortex-M4 [8]

Základními parametry jádra Cortex-M4 jsou:

- Podporované instukční sady Thumb/Thumb-2
- DSP rozšíření s podporou jednociklových 16,32-bit MAC, jednociklových MAC s 2x16-bitů operandy. Dále s podporou 8 a 16 bitové SIMD aritmetiky a Harwarovou děličkou.

- S volitelnou podporou FPU
- Volitelnou MPU (Memory Protection Unit)
- od 1 do 240 fyzických přerušení + Nemaskovatelné přerušení
- Debug přes rozhraní JTAG a SWD s možností 8-mi hardwerových breakpointů a 4 watchpointů.

2.2 STM32F407VG

STM32F407VG je mikroprocesor řady F4 výrobce ST Microelectronics. Řada F4 byla představena v roce 2011. Jedná se mikroprocesor založený na 32-bitovém RISC jádře ARM Cortex-M4 taktovaném na frekvenci až 168MHz. Obsahuje jednotku FPU a instrukční sada obsahující DSP instrukce ho předučuje k použití v řídicích aplikacích a zpracování signálů. Dle označení se jedná o verzi s rozšířenou podporou komunikačních rozhraní v pouzdře LQFP100. Procesor obsahuje 1Mbyte interní Flash paměti pro program a 192+4Kbyte interní SRAM. Blokové schéma procesorů STM32F407 je zobrazeno na obrázku 2.2. Mikroprocesor SMT32F407VGT6 obsahuje velké množství periférií, z nichž pro tuto bakalářskou práci jsou nejdůležitější periférie CAN1 a CAN2 a periférie MAC pro připojení k ethernetu.

2.2.1 Ethernet periférie

Tato periférie umožňuje přijímat a vysílat data přes ethernet dle normy IEEE 802.3-2002. Její blokové schéma je zobrazeno na obrázku 2.3.

Za hlavní parametry lze považovat:

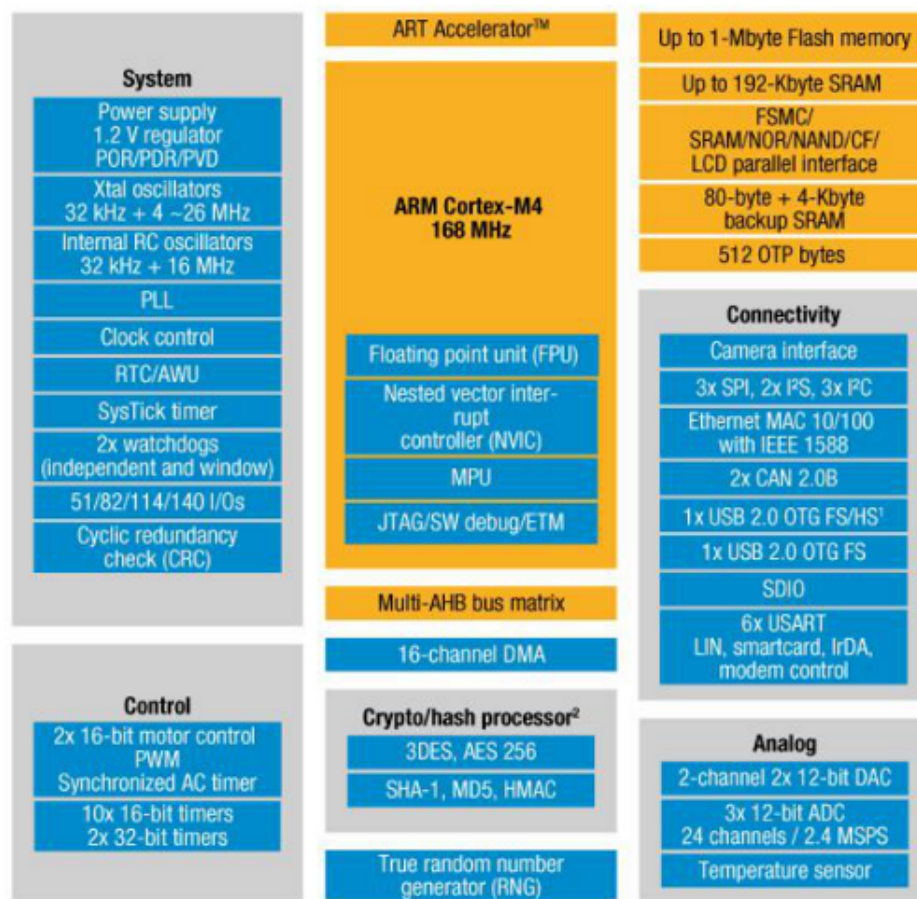
- Podporu standardních rozhraní MII a RMII pro připojení externího PHY.
- Integrované DMA + 2KB Transmit FIFO a 2KB Receive FIFO
- Podpora full-duplex i half-duplex

2.2.2 CAN periférie

Integrovaná periférie CAN podporuje obě verze protokolu (CAN2.0A i CAN2.0B) s podporovanými rychlostmi až 1Mbps. Obsahuje tři vysílací mailboxy s programovatelnou prioritou odesílání. Dále 28 přijímacích filtrů sdílených pro CAN1 a CAN2 a dvě přijímací FIFO. Blokové schéma periférie CAN je na obrázku 2.4

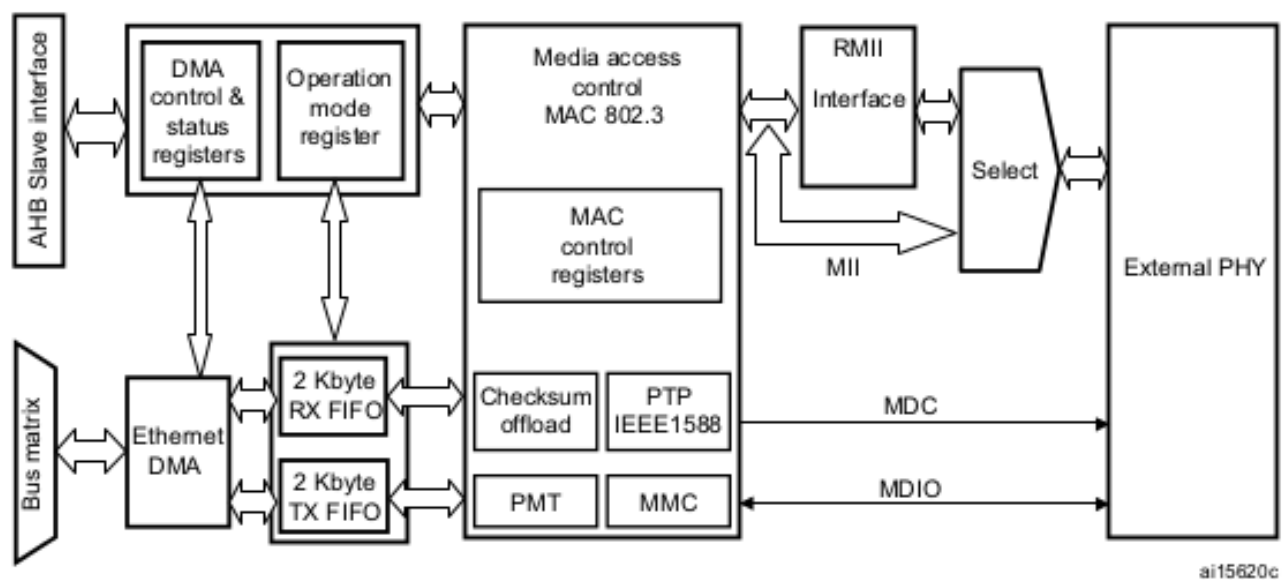
2.3 Vývojová prostředí

Pro vývoj pro procesory architektury ARM existuje několik kompilátorů, jak proprietární tak s otevřeným zdrojovým kódem. Tyto kompilátory pak mohou být inte-

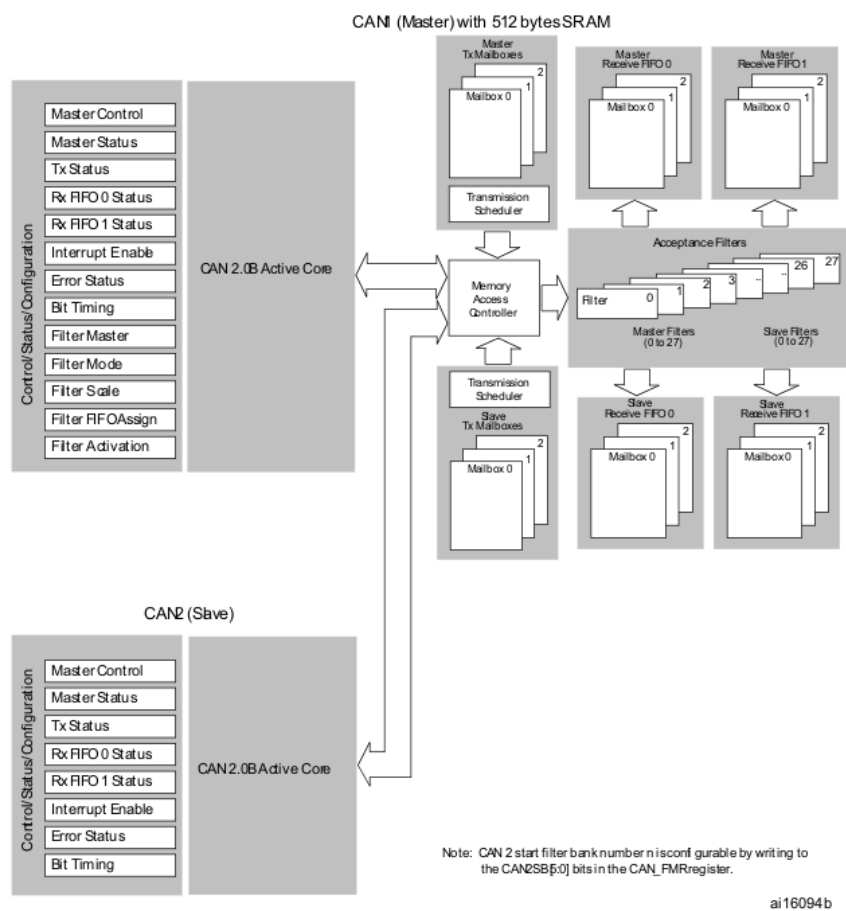


Obr. 2.2: Blokové schéma procesoru STM32F407VGT6 [9]

grovány buď do IDE (Integrated Development Environment) spolu s DBG serverem a dalšími utilitami. Pro úpravy firmware v této bakalářské práci jsem použil vývojové prostředí EM::Blocks. Jedná se all-in-one řešení pro vývoj pro vestavná zařízení na platformě Windows. IDE je odvozeno od opensource projektu Code::Blocks a využívá v instalačním balíčku zahrnutý překladač ARM GCC Compiler. Výhodou tohoto projektu je plná podpora procesorů rodiny STM32F4 a integrovaná podpora pro debugování a programování za použití STLinkV2.



Obr. 2.3: Blokové schéma periferie Ethernet procesoru STM32F407 [4]



Obr. 2.4: Blokové schéma periferie CAN procesoru STM32F407 [4]

3 ETHERNET

Ethernet je základní protokolem rozšířeným především v lokálních sítích LAN.

3.1 Ethernetová komunikace z pohledu ISO/OSI

Následující přehled kategorizuje Ethernetovou komunikaci do jednotlivých vrstev protokolu ISO/OSI. Přestože Ethernet jako takový je definován pouze na nejnižších dvou vrstvách, jsou v přehledu uvedeny i protokoly z vyšších vrstev, které k přenosu používají ethernet a jsou použité v této práci. Přehled zároveň uvádí, kterou částí HW či SW je v projektu CanShark realizována.

- Fyzická vrstva - standard IEEE 802.3 podporuje více verzí fyzické vrstvy. Mezi nejstarší lze zařadit dnes již minimálně používané 10Base-5 a 10Base-2, které jako médium používaly tlustý respektive tenký koaxiální kabel. Tyto verze měly schodně přenosovou rychlost 10Mbit/s a měly sběrníkovou topologii s maximální vzdáleností mezi segmenty 500m respektive 180m. Novější verze 10Base-T (10Mbit/s), 100Base-TX (100Mbit/s) a 1000Base-T (1Gbit/s) používají jako médium kroucený pár a jsou v současné době nejrozšířenější variantou fyzické vrstvy v sítích LAN. Na velké vzdálenosti či pro větší rychlosti je možné použít i verze, které jako fyzické médium používají optický kabel (100Base-FX, 1000Base-FX a další). Fyzická vrstva je pak v praxi realizována PHY. KSZ8051MLL podporuje fyzické vrstvy verze 10Base-T a 100Base-TX.
- Spojová vrstva - dělí se na dvě podvrstvy MAC (Media Access Control) a LLC (Logical Link Control). MAC zajišťuje fyzické adresování a řídí přístup k médiu. K tomu používá metodu CSMA/CD (Carrier Sense Multiple Access/Collision Detection). LLC pak zajišťuje předávání dat z vyšších vrstev na MAC vrstvu a naopak. Realizuje řízení toku a kontroluje případné chyby v datech. Tato vrstva je plně implementována v periférii MAC v mikroprocesoru.
- Síťová vrstva - Nejčastěji používanou metodou v síťové vrstvě je IP (Internet Protokol). Ten umožňuje spojení mezi různými částmi sítě. Tato vrstva bývá již plně realizována softwarem. V současné době jsou nejpoužívanější verze IPv4 a IPv6. Rozdíl mezi nimi je v délce adresy a tím i množství adres.
- Transportní vrstva - využívá protokolů UDP a TCP. TCP je základní transportní protokol s potvrzováním přijetí. Při ztrátě dat jsou tak vyslána data znovu. UDP je na rozdíl od TCP bez povrzení a proto nezaručuje, že se některá data neztratí. Díky tomu je však UDP jednodušší na implementaci, protože nepotřebuje v paměti uchovávat odeslané pakety pro jejich případné znovuo-deslání.

- Další vrstvy - v závislosti na použití jsou v dalších vrstvách jako nadstavba nad TCP a UDP použity další protokoly (např. HTTP, DNS atd.) Ty však nejsou pro tuto aplikaci podstatné.

3.2 Ethernetový rámeček

Jedná se o základní rámeček zabalující všechna data posílaná po ethernetu. Jeho rozdělení je vidět na obr. ???. Rámeček se skládá z těchto částí

- Preamble - obsahuje 7 Byte kde se střídají '1' a '0'. Byte č.8, který značen SFD (Start of Frame delimiter) pak obsahuje hodnotu '10101011' a značí konec preamble a start vysílání. Celá preamble slouží k synchronizaci hodin přijímající části.
- MAC adresa cíle - je 48 bitů dlouhá a může být všeobecná (broadcast), individuální (unicast) nebo skupinová (multicast).
- MAC adresa zdroje je adresa odesílatele rámečku.
- Pole Typ určuje typ vyššího protokolu. Například pro IPv4 protokol nabývá hodnoty 0x0800.
- Data - užitečná data rámečku mají minimální délku 46 Byte a maximální délku 1500 Byte.
- CRC - kontrolní součet. Spočítáním kontrolního součtu celého rámečku a porovnáním výsledku s hodnotou uloženou v tomto poli lze detekovat případné změny dat, které nastaly při přenosu. Rámeček se špatným CRC je zahozen jako chybný.

8 Byte	6 Byte	6 Byte	2 Byte	46 -1500 Byte	4 Byte
Preamble	Cílová adresa	Zdrojová adresa	Typ	Data	Kontrolní součet

Obr. 3.1: Ethernet frame

3.3 UDP protokol

UDP (User Datagram Protokol) je součástí protokolů združených v TCP/IP. Na rozdíl od protokolu TCP nezaručuje, že byl rámec doručen. Případné ztráty dat tak musí být řešeny vyšší vrstvou. Udp se často používá pro komunikaci stylu otázka - odpověď, která z vlastního principu nepotřebuje potvrzování. Výhodou UDP je pak jeho snazší implementovatelnost a menší nároky na výkon a paměť procesoru. Je to dáno tím, že procesor nemusí uchovávat již odeslané rámce pro případné znovuo-deslání. Hlavička protokolu UDP 3.2 obsahuje pouze zdrojový port, cílový port, délku přenášených dat, kontrolní součet celého rámce a vlastní data.

+	bity 0-15	bity 16-31
0	Zdrojový port	Cílový port
32	Délka	Kontrolní součet
64 ⋮ ⋮ ⋮	DATA ⋮ ⋮ ⋮	

Obr. 3.2: Hlavička UDP protokolu

3.4 ARM a Ethernet

Pro připojení ethernetu obecně ke všem typům jendochipových mikroprocesorů se používají tři základní principy, které se liší mírou integrace funkcí popsaných modelem ISO/OSI do cílového procesoru.

1. Připojení k ethernetu pomocí modulu - Toto řešení přenáší veškerou podporu ethernetu na připojený modul či IO.Ten pak implementuje veškeré vrstvy ISO/OSI včetně kompletního TCP/IP stacku. K cílovému procesoru je pak připojen nejčastěji některou ze sériových sběrnic (USART, SPI, atd.). Výhodou tohoto řešení je možnost připojit téměř jakýkoliv procesor. Pokud jsou

některé funkce výrobce obvodu podporovány (např. webserver), lze je často zprovoznit pouhým nastavením parametrů. Velkou nevýhodou těchto obvodů však je rychlost, která se nejčastěji pohybuje pouze v řádu jednotek Mb/s.

2. Připojení k ethernetu přes interní MAC. Jedná se o plnohodnotné připojení kdy procesor obsahuje hardwarem realizovanou MAC vrstvu. K té se za pomoci některého ze standardizovaných rozhraní (MII či RMII) připojí externí PHY. Výhodou je možnost využití maximální rychlosti ethernetu a plná kontrola nad ním. Nevýhodou je nutnost většího množství RAM paměti a výkonu, které jsou potřeba pro obsluhu ethernetu. Proto tuto možnost nejčastěji nenajdeme v nejméně výkonném segmentu mikroprocesorů.
3. Připojení přes interní MAC a PHY. Jedná se o rozšíření předchozí možnosti o v mikroprocesoru integrovanou vrstvu PHY. Výhodou tohoto způsobu připojení je zjednodušení návrhu zařízení. Nevýhodou naopak omezení na konkrétní fyzické vrstvy podporované integrovaným PHY.

3.4.1 MII-Media Independent Interface

Jedná se o standardizované rozhraní (IEEE 802.3u) pro propojení v procesoru integrovaného MAC modulu a externího PHY. Je navrženo pro plnou podporu rychlostí 100Mb/s.

Jméno signálu	I/O popis u MAC	Popis
TXC	Input	Hodiny pro synchronizaci při odesílání
TXEN	Output	Potvrzení dat vystavených na datových vodičích
TXD[3:0]	Output	Vysílaná data
RXC	Input	Hodiny pro synchronizaci příjmu
RXDV	Input	Potvrzení dat vystavených na datových vodičích
RXD[3:0]	Input	Přijímaná data
RXER	Input	Signalizuje error při příjmu
CRS	Input	Detekce začátku vysílání
COL	Input	Detekce kolize

Tab. 3.1: Signály rozhraní MII

3.4.2 RMII-Reduced MII

Reduced MII redukuje počet pinů nutných pro připojení PHY. Narozdíl od MII tak využívá pouze dva datové vodiče RXD[1:0] a TXD[1:0]. Hodinové signály RXC

a TXC jsou sloučeny do jednoho REF_CLK o dvojnásobné frekvenci. Do jednoho vodiče jsou také sloučeny signály RXDV a CRS. Signál COL byl pak plně vynechán. To vede ke celkové redukci potřebného počtu pinů z 15 pro MII na 8 pro RMII.

3.4.3 MIIM - MII Management Interface

Někdy je také označována SMI (Serial management interface). Jedná se o IEEE 802.3 standardizovanou sběrnici pro vyčítání a nastavování vnitřních registrů PHY. Dává tak MAC vrstvě procesoru plnou kontrolu nad aktuálním statusem a všemi nastaveními PHY. Sběrnice se skládá ze dvou signálů

- MDIO - Management data input/output je datová sběrnice pro komunikaci s PHY
- MDC - hodinový signál jako reference pro komunikaci na MDIO

4 SBĚRNICE CAN

CAN (Controller Area Network) je sériový komunikační protokol vyvinutý v 80. letech firmou Robert Bosch GmbH pro použití v automobilovém průmyslu s cílem snížit množství kabeláže potřebné k propojení elektronických systémů automobilů. Jedná se o robustní síť typu multimaster (každý z uzlů může nezávisle na ostatních zahájit vysílání) s vysokou mírou odolnosti proti chybovým stavům a s maximální přenosovou rychlostí až 1Mbit/s. Fyzická vrstva a linková vrstva sběrnice CAN jsou definovány normou ISO 11898. Vzhledem k podpoře CAN v obvodech většiny výrobců, nízké ceně a jednoduchosti implementace se tato sběrnice rozšířila z automobilového průmyslu i do jiných průmyslových odvětví.

4.1 Fyzická vrstva CAN

Norma ISO 11898 definuje požadavky na fyzickou vrstvu sběrnice CAN. Fyzická vrstva sběrnice CAN musí přenášet dva stavy: dominantní a recesivní. Pokud jakékoliv zařízení vysílá na sběrnici dominantní stav, je sběrnice v dominantním stavu. V recesivním stavu se sběrnice nachází pouze tehdy, když všechna zařízení připojená ke sběrnici vysílají recesivní úroveň. V normě ISO 11898 je definována diferenciální sběrnice, která se pro realizaci fyzického přenosového média používá nejčastěji. Tvoří ji dva vodiče, které jsou označovány jako CAN_H a CAN_L a jednotlivé stavy sběrnice (dominant a recesive) jsou definovány rozdílovým napětím těchto vodičů. Za recesivní je považován stav s rozdílovým napětím $V_{diff} = 0V$ a stav dominantní pak pro $V_{diff} > 2V$.

4.1.1 Časování sběrnice CAN

Sběrnice CAN používá kódování NRZ (Non Return to Zero - Bez návratu k nule) což snižuje nároky na šířku pásma vodiče, zároveň to ale přináší potřebu řešit synchronizaci přijímače. Rozlišujeme dva druhy synchronizace na sběrnici:

- Počáteční synchronizace - přijímač se synchronizuje na základě synchronizačního pulzu na začátku zprávy.
- Resynchronizace - přijímač si v průběhu příjmu zprávy opravuje synchronizaci podle změn jednotlivých bitů na sběrnici.

Aby mohla být správně provedena resynchronizace přijímače, musí Linková vrstva zajistit, že sběrnice nebude v jednom stavu déle než dobu potřebnou pro přenos pěti bitů. Toho se docílí metodou Bit stuffing. Linková vrstva musí za každých po sobě jdoucích 5bitů stejné hodnoty vložit bit opačné logické úrovně. Na tomto bitu pak může přijímač provést resynchronizaci.

Každý bit na sběrnici si přijímač vnitřně dělí do čtyř částí:

- SYNC_SEG - v tomto časovém úseku se očekává přítomnost hrany signálu.
- PROP_SEG - tento časový úsek je určen pro kompenzaci prodlevy nutné k šíření signálu mezi nejvzdálenějšími uzly na sběrnici.
- PHASE_SEG1 - ohraničuje vzorkovací bod zleva. V případě, že je hrana signálu detekována až za SYNC_SEG, je PHASE_SEG1 prodloužen tak, aby byl posunut bod vzorkování.
- PHASE_SEG2 - ohraničuje bod vzorkování zprava. V případě, že je hrana signálu detekována až za SYNC_SEG, je tento segment zkrácen tak, aby byl správně navzorkován následující bit.

4.2 Linková vrstva sběrnice CAN

Linkovou vrstvu sběrnice CAN lze rozdělit na:

- Podvrstva řízení přístupu k médiu (MAC - Medium Access Control). Tato podvrstva zajišťuje kódování dat, vkládání doplňkových bitů (Bit stuffing). Dále řídí přístup k médiu, řeší kolize na sběrnici a potvrzuje přijetí zpráv.
- Podvrstva řízení datového spoje (LLC - Logical Link Control). Tato podvrstva zajišťuje filtrování přijatých zpráv a nastavování přerušování.

4.2.1 Řízení přístupu k médiu (MAC)

Každý uzel může začít na sběrnici vysílat v libovolný okamžik, pokud se sběrnice nachází v klidovém stavu (bus free). Vzhledem k možnosti, že začne v jeden okamžik na sběrnici vysílat více uzlů, obsahuje návrh sběrnice CAN systém řešení kolizí. Každý vysílající uzel porovnává stav sběrnice s aktuálně vysílaným bitem. Pokud na sběrnici není stejná úroveň, jakou uzel vysílá, okamžitě přeruší další vysílání a pokračuje v příjmu dané zprávy. Ke kolizi tak dojde až v případě, kdy vysílané rámce obsahují na daném bitu jinou úroveň. Tím je zajištěno, že na sběrnici zůstanou data s vyšší prioritou.

4.2.2 Formáty datových rámců na sběrnici CAN

Sběrnice CAN obsahuje 4 druhy datových rámců[4]:

Datová zpráva (Data Frame)

Rozlišujeme dva typy datových zpráv. Standartní, která je definována specifikací 2.0A a Extended, která je definována specifikací 2.0B. Rozdílem v těchto zprávách

je délka identifikátoru zprávy, která je u Extended frame 29bitů, zatímco u Standard frame je to pouze 11bitů. Standardní formát zprávy je zobrazen na obrázku a obsahuje následující části:

- SOF - 1bit: Začátek zprávy. Slouží k synchronizaci přijímačů.
- Arbitration Field - 11 + 1 bitů: Určuje prioritu zprávy na sběrnici. Obsahuje 11bitů dlouhý identifikátor zprávy a 1 bit (RTR) pro rozlišení datové zprávy od zprávy žádost o data. V datové zprávě je bit RTR dominant. Tím je určena vyšší priorita datové zprávy před zprávou žádost o data.
- Control Field - 2 + 4 bity: Obsahuje dva rezervní bity (R0 a R1) ve stavu dominant a 4 bity určující počet datových bajtů ve zprávě.
- Data Field - 0 až 64 bitů: Obsahuje 0 až 8 datových bajtů.
- CRC Field - 16 bitů: Obsahuje 15 bitů dlouhý kontrolní CRC kód umožňující kontrolu nepoškozenosti zprávy a 1 dominant bit oddělující CRC od ACK bitu.
- ACK Field - 2 bity: Pole potvrzení obsahuje 1 recesivní bit, který jakýkoliv přijímač na sběrnici přeruší bitem dominantním a tím vysílači potvrdí příjem. Druhý bit je recesivní a odděluje potvrzení.
- End Of Frame - 7 bitů: Označuje konec zprávy vysláním recessive bitů.

Extended formát zprávy využívá rezervní bit R1 na rozlišení typu rámce. Dle CAN 2.0B je nazýván IDE (Identifier Extended) a je dominant pro standardní formát a recessive pro rozšířený formát. Bit RTR je nahrazen recessive bitem SRR. Při kolizi Standard a Extended formátu se stejným 11-ti bitovým identifikátorem zajišťuje bit SRR přednost pro formát standardní. Za bitem IDE Extended formátu následuje 18 bitů rozšiřující identifikátor zprávy. Další struktura zprávy je stejná se standardním formátem.

Žádost o data (Remote Frame)

Žádost o data má podobnou strukturu jako datová zpráva. Žádost o data neobsahuje datovou oblast (Data Field) a bit RTR je nastaven na recessive. Tím je zajištěno, že pokud se jeden uzel vysílá žádost o data a druhý posílá data se stejným identifikátorem ve stejnou dobu, tak datová zpráva bude mít na sběrnici přednost.

Chybová zpráva (Error Frame)

Chybovou zprávu vysílá každý uzel, který detekuje na sběrnici chybu. Dle svého vnitřního stavu vysílá buď aktivní nebo pasivní chybovou zprávu. Ta se skládá ze šesti po sobě jdoucích dominantních bitů pro aktivní chybovou zprávu nebo ze šesti recesivních bitů pro pasivní chybovou zprávu. Při vysílání aktivní chybové zprávy je přenášená zpráva poškozena (nedodržení pravidla vkládání bitů) což detekují i ostatní uzly na sběrnici a začnou také vysílat chybové zprávy. Délka úseku ERROR

FLAG pak je minimálně 6 bitů a maximálně 12 bitů dlouhá. Za polem ERROR FLAG následuje 7 reccesive bitů (ERROR DELIMITER) indikující konec chybové zprávy. Každý uzel po odvysílání ERROR FLAG kontroluje stav sběrnice a čeká na přijmutí celého pole ERROR DELIMITER.

Zpráva o přetížení (Overload Frame)

Zprávu o přetížení vysílá uzel, který potřebuje oddálit vyslání dalších zpráv na sběrnici. Struktura zprávy je podobná chybové zprávě, narozdíl od ní je však vysílána až po konci zprávy.

4.3 Ostatní vrstvy sběrnice CAN

Vyšší vrstvy ISO/OSI modelu jsou na sběrnici CAN reprezentovány mnohými, často vzájemně nekompatibilními standardy, jako CANOpen, DeviceNet a další standardy založené na SAE J1939 ??.

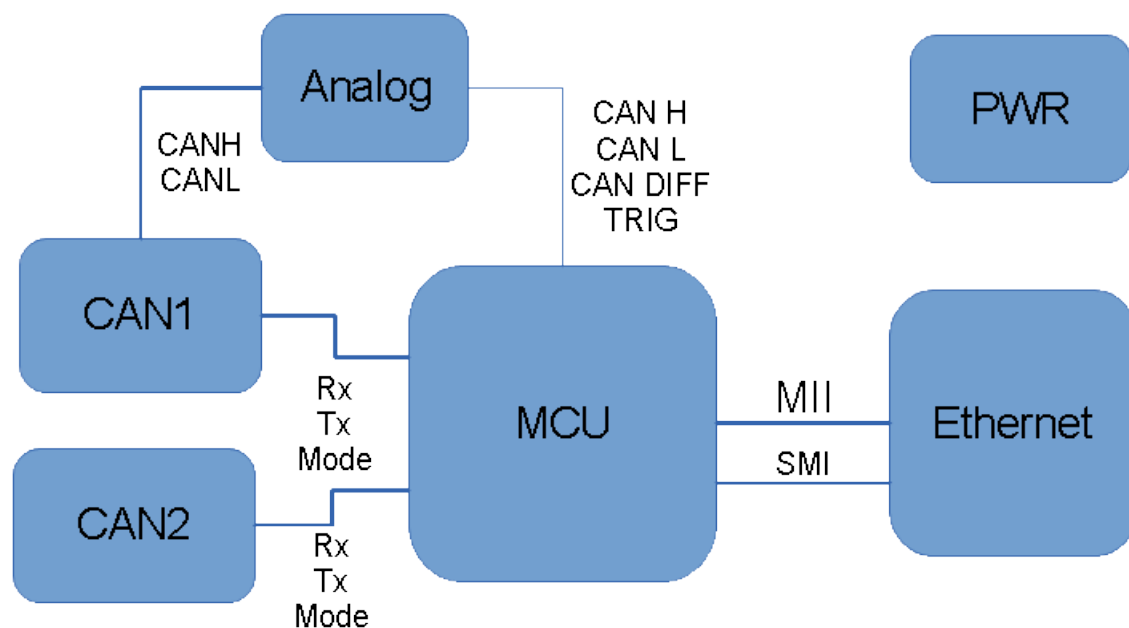
4.4 Zabezpečení dat na sběrnici CAN

Pro zajištění spolehlivosti přenosu dat na sběrnici CAN definuje tato sběrnice několik způsobů zabezpečení. Mezi nejdůležitější patří:

- Kontrola přenášených zpráv za pomoci CRC, které je vysíláno na konci každé zprávy.
- Potvrzení přijetí ACK.
- Bit stuffing (vkládání bitu) po pěti po sobě následujících bitů stejné úrovně.
- Kontrola formátu zprávy
- Kontrola sběrnice během odesílání, kdy zařízení detekuje jiný stav sběrnice než zařízením vysílaná hodnota.

5 NÁVRH HARDWARE

Navrhovanou desku lze rozdělit na několik základních bloků, ty jsou rozbrazeny na blokovém schématu 5.1. Jedná se o bloky MCU, Ethernet, CAN1, CAN2, PWR a ADC.



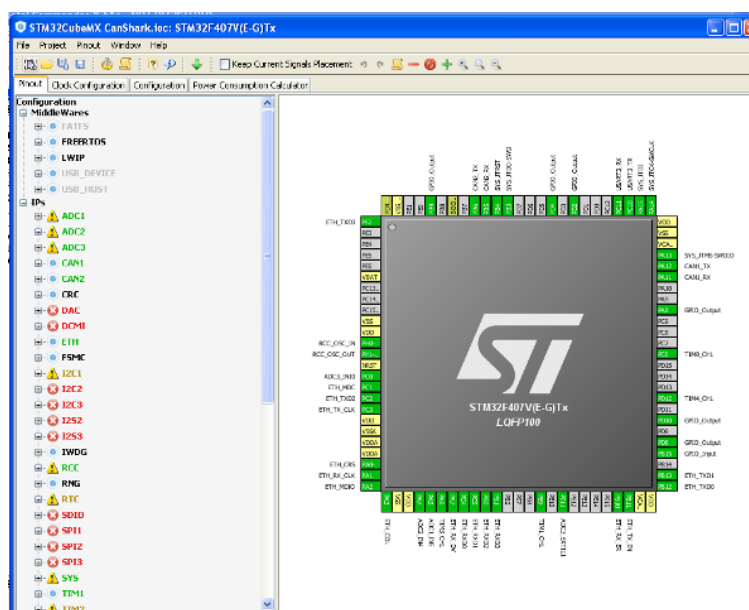
Obr. 5.1: Blokové schéma desky CanShark

Popis návrhu jednotlivých částí je popsán v následujících podkapitolách.

5.1 MCU

Před návrhem jednotlivých částí HW bylo potřeba ověřit, zda vedoucím práce doporučená mikroprocesor STM32F407VG splňuje všechny požadavky. K tomu lze využít utilitu STM32CubeMX. Jedná se o program dodávaný firmou STMicroelectronics pro návrh využití periférií mikroprocesorů. Program automaticky ověří možnost přiřadit danou periférii k příslušnému pinu bez nutnosti studovat datasheet. Okno

programu je vidět na obrázku 5.2 a kompletní konfigurační soubor je umístěn na příloženém CD.



Obr. 5.2: Konfigurační program STM32CubeMX

5.1.1 Nahrávání a ladění programu

Pro nahrání programu a jeho ladění v mikroprocesoru podporuje STM32F407VGT6 rozhraní JTAG a SWD. Zároveň lze k nahrání použít interní bootloader mikrokontroléru. Pro navrhovanou desku jsem se rozhodl použít obě možnosti nahrávání.

Nahrávání pomocí programátoru

Pro nahrávání pomocí programátoru jsem implementoval pouze podporu pro SWD. Narozdíl od rozhraní JTAG tak došlo k ušetření GPIO pinů procesoru. Dalším důvodem pro toto rozhodnutí bylo plánované použití Discovery kytu pro nahrávání navržené desky. Ten však podporuje pouze rozhraní SWD.

Jelikož není žádný standardizovaný konektor pro rozhraní SWD, byl na navržené desce použit šestipinový header, jehož zapojení je kompatibilní se zapojením SWD konektoru na DISCOVERY kitech od firmy STMicroelectronics, které budou nejčastěji používány k programování. Popis jednotlivých pinů včetně popisu jejich zapojení obsahuje tabulka 5.1.

Číslo pinu	Název	Popis
1	UCC	Tento pin slouží programátoru k měření napájecího napětí procesoru. Na navržené desce je tento pin připojen na napájecí napětí přes propojku R42. Tu jsem ponechal neosazenou, neboť mnou používaný programátor informací o napájecím napětí desky nevyžaduje. Pokud by byla propojka osazena, zvyšuje se riziko zkratu napájecího napětí při manipulaci s programovacími vodiči.
2	SWCLK	Hodinový signál pro přenášená sériová data. Vyžaduje pull-down rezistor $R37 = 100k\Omega$.
3	GND	
4	SWDIO	Datový vodič pro sériovou komunikaci. Připojeno pull-up rezistorem $R38 = 100k\Omega$ na napájecí napětí 3,3V
5	NRST	
6	SWO	Vodič pro posílání sériových debugovacích zpráv při použití interní ITM (Instrumentation trace macrocell)

Tab. 5.1: Zapojení SWD konektoru

Interní bootloader

Interní bootloader je uložen v systémové části paměti a umožňuje nahrávání kódu přes rozhraní uvedená v tabulce 5.2. Interní bootloader je možné zapnout buď pro-

Rozhraní	Použitelné piny
USART1	PA9 a PA10
USART3	PB10 a PB11 nebo PC10 a PC11
CAN2	PB5 a PB6
USB OTG FS in Device mode	PA11 a PA12

Tab. 5.2: Piny použitelné pro interní bootloader

gramově skokem na paměťové místo, kde se nachází kód tohoto bootladeru, a nebo pomocí pinu BOOT0. Hodnoty na boot pinech jsou zaznamenány při čtvrté náběžné hraně SYSCLK po resetu. Pokud je na BOOT0 zaznamenána log 0, je načten programový kód z interní flash paměti. V opačném případě se na základě hodnoty BOOT1 (pin PB2) spustí buď interní bootloader, nebo program ve vestavěné paměti SRAM ???. Aby mohl být na navrhované DPS využit interní bootloader, použil jsem JP1 pro přepínání konfigurace BOOT0. Přepnutím jednoho jumperu je tak

možné přivést na BOOT0 napětí 3,3V, nebo ho připojit k GND. Pin PB2 (BOOT1) je pak možné připojit propojkou R1 k napájecímu napětí pro nahrávání kódu z SDRAM či propojkou R2 k zemi pro spouštění interního bootloaderu. Pro možnost použití interního bootloaderu nejen s rozhraním CAN ale i přes USART, který je díky dostupnosti převodníků USB/USART vhodnější pro update firmware bez programátoru, jsou vyvedeny piny PC10 a PC11 (USART3) na pinovou lištu. Tohoto rozhraní lze kromě využití pro update firmware využít i pro zasílání ladících zpráv z desky CanShark do PC.

5.1.2 Napájení STM32F407

Mikroprocesor STM32F407 pracuje při napájecích napětích 1,8V až 3,6V. Obsahuje interní lineární regulátor s napětím 1,2V pro napájení vnitřních částí. Pro jeho správnou funkčnost je zapotřebí připojit na piny V_{CAP_1} a $V_{CA/_2}$ blokovací kondenzátory $2,2\mu F$. Každý napájecí pin Vdd pak je přemostěn 100nF kondenzátorem pro zamezení zákmitů napájecího napětí. Vstup pro bateriové zálohování MCU, které není v navrhované desce použito je přes blokovací kondenzátor připojen na napájecí napětí Vdd.

Zvláštní kategorii pak tvoří vstupy pro napájení analogové části. Ty umožňují využít pro napájení ADC přesnější napájecí zdroj a tím snížit rušení od digitálních částí a zvýšit přesnost měření ADC. Jelikož přesnost ADC není v této aplikaci kritická, jsem pro napájení použil stejného 3,3V spínaného zdroje jako pro digitální část. Zmíněné vstupy jsou spolu se vstupy V_{REF+} a V_{REF-} , které určují rozsah měření ADC, přemostěny dle doporučení v ??, tj. každý kondenzátory o hodnotách 100nF a $1\mu F$.

5.2 CAN1 a CAN2

Aby nemusela deska obsahovat další napájecí zdroj pro budiče sběrnice CAN. Rozhodl jsem se vybrat budiče s napájecím napětím 3,3V a to konkrétně SN65HVD od výrobce Texas Instruments. Jedná se o budiče s podporou pro sběrnice o rychlosti až 1Mbps.

5.2.1 Volba módu CAN budiče

Budič SN65HVD230 umožňuje pinem Rs určit jeden z následujících módů funkčnosti [11]:

- Standby mode - funkčnost vysílače je potlačena a driver funguje pouze v režimu příjmu. Tento mód je aplikován pokud je napětí na vstupu $V_{Rs} > 0,75 \cdot V_{cc}$

- Slope control mode - v tomto módu je omezena rychlost změny výstupního signálu (a tím i délky náběžných hran) na cca $2V/\mu s$ až cca $15V/\mu s$. Tento mód lze nastavit připojením pinu Rs na GND přes rezistor, jehož velikostí se určuje maximální proud z výstupního pinu. Sklonu $2V/\mu s$ dosáhneme s rezistorem $100k\Omega$, $15V/\mu s$ pak s rezistorem $10k\Omega$.
- Hight speed (no slope control mode) - vysílač budí sběrnici s co nejkratšími náběžnými hranami, které jsou limitovány pouze parametry připojeného vedení.

Tento mód je aktivován pokud napětí na vstupu $V_{Rs} < 1V$

Navržené schéma umožňuje zvolit jakýkoliv z těchto módů. Jednou z možností je propojit přes rezistor R24 pin Rs na výstupní GPIO pin mikroprocesoru. Jestliže $R24 = 0\Omega$, pak aplikace přepíná mezi Stanby a Hight speed mode. Pokud bude hodnota R24 v rozsahu $10k\Omega$ až $100k\Omega$, bude mikroprocesor přepínat mezi módy Stanby a Slope control. Má-li být mód nastaven nezávisle na stavu procesoru, je možné osadit buď pull-up rezistor R25 pro trvalý silent mode nebo pull-down rezistor R26, jímž podle jeho velikosti zajistíme trvalý Hight speed/Slope control mode. Toto na procesoru nezávislé nastavení se může hodit, jestliže například chceme zjistit stav sběrnice bez zásahu do její komunikace a přitom chceme předejít chybné konfiguraci sw obsluhou. Dalším potřebným použitím je přehrávání firmware snifferu za pomoci interního bootloderu přes rozhraní CAN. V době spuštění interního bootloderu je totiž pin ovládající výběr módu budiče v nedefinovaném stavu. Tento pin tak musí být přes pull-down rezistor připojen trvale k zemi a určovat tak mód, pokud není definován mikroprocesorem.

5.2.2 Volba vhodných konektorů

Jelikož neexistuje standardizovaný konektor pro připojení sběrnice CAN, je možné se u různých výrobců zařízení setkat s různými typy konektorů. Aby tedy bylo možné co nejjednodušeji připojit k navrhované desce libovolné zařízení, rozhodl jsem se použít konektory se svorkovnicí. Svorkovnice umožňuje jednoduše připojit jakékoliv vodiče a umožňuje připojovat zařízení stejně rychle jako konektor.

U obou konektorů pro připojení sběrnice CAN je zakončovací odpor sběrnice (120Ω), který je možné připojit pomocí jumperů JP3 a JP5.

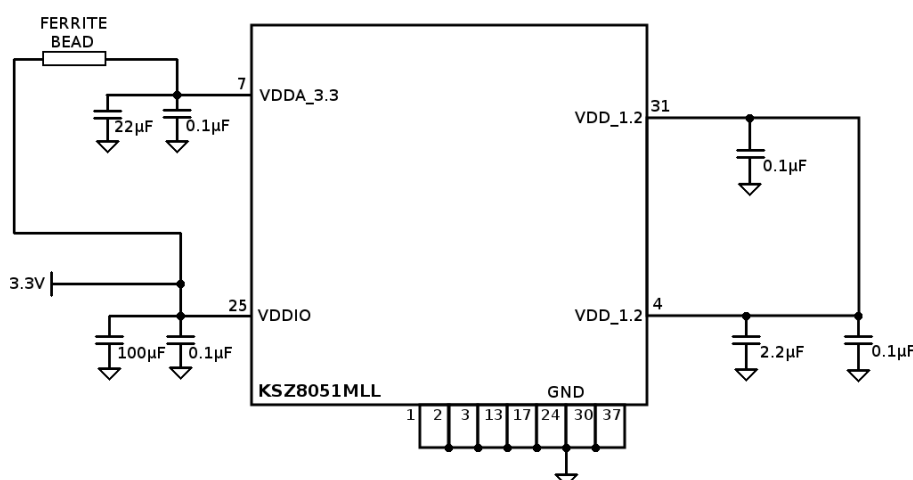
5.3 Ethernet

Pro připojení k síti ethernet byl zvolen pin KSZ8051MLL. Jedná se o integrovaný obvod vyžadující jedno napájecí napětí ($3,3V$) s vnitřním $1,2V$ zdrojem pro napájení jádra. Obsahuje integrované zakončovací rezistory pro oba páry diferenčních vodičů a je navržený pro co nejjednodušší návrh pcb. Jednotlivé piny KSZ8051MLL zle rozdělit

do 4 skupin a to Napájení, Zdroj hodinového signálu, MII interface, výstupy Ethernet a ostatní signály.

5.3.1 Napájení KSZ8051MLL

Schéma napájení je znázorněno na obr 5.3 a ukazuje požadavky na napájení. KSZ8051MLL potřebuje jedno napájecí napětí 3.3V přivést na piny VDDA_3.3 a VDDIO. Oba piny jsou blokovány předepsanými kondenzátory. Dále je potřeba pouze spojit piny VDD_1.2 (pin č.4 a č.31) a doplnit příslušnými blokačními kondenzátory.



Obr. 5.3: Přehled napájení pro KSZ8051MLL [2]

5.3.2 Zdroj hodinového signálu

KSZ8051MLL potřebuje ke své funkci hodinový signál o frekvenci $25\text{MHz} \pm$ s přesností minimálně 50ppm. K tomuto účelu jsou použity piny XI a XO. XO je výstupní pin, pro uzavření smyčky při použití krystalu nebo oscilátoru. XI je pak vstup hodinového signálu. Pokud je nadřazená jednotka (procesor) schopen dodávat hodinový signál potřebné kvality, pak je možné použít právě a pouze vstup XI. Použitý mikroprocesor STM32F407VGT6 má možnost generovat na výstupu svého pinu hodinový signál potřebných parametrů, ale pouze za předpokladu, taktován 25MHz krystalem. Mikroprocesor pak na daný pin pouze přeposílá vstupní hodinový singnál. K tomu lze u STM32F407VGT6 použít pouze pin PB5. Na ten je však připojen signál CAN2_RX. Při použití obou periferií CAN a Ethernetu tak musí mít PHY vlastní

krystal. Deska proto byla navržena pro taktování s použitím 25MHz krystalu se zatěžovací kapacitou 18pF. Podle vzorce

$$C_{20} = C_{21} = 2 * CL - (CP + CI) \quad (5.1)$$

Kde $C_{20}C_{21}$ jsou hodnoty zemních kondenzátorů, CL je zatěžovací kapacita krystalu, CP je parazitní kapacita DPS a CI je vstupní kapacita PHY. Kapacitu desky a vstupů jsem odhadl na 14pF ze vzorce 5.1 pak vyšlo $C_{20} = C_{21} = 22pF$

5.3.3 MII Interface

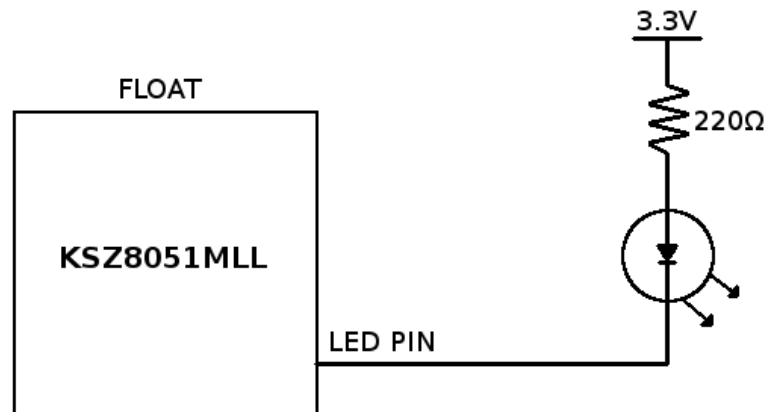
Piny MII nepotřebují žádné speciální zapojení. V referenčním zapojení je však doporučeno připojit piny, které jsou vůči KSZ8051MLL jako výstupy přes rezistor 33R z důvodů potlačení odrazů.

5.3.4 Připojení Ethernet konektoru

Jako konektor pro připojení UTP kabelu byl vybrán konektor Rofa a to z důvodu jeho dostupnosti na UAMT. Konektor má integrovaný oddělovací transformátor a dvě signalizační LED. Konektor je s PHY spojen dvěma diferenciálními páry vodičů, které jsou routovány na desce tak, aby měli co nejsouhlasnější délku. Cesty vysílačího a přijímacího páru by měli být routovány s co nejschodnější délkou a vzdáleny od ostatních signálů nejméně 18mil. Zapojení zakončovacích rezistorů není potřeba, protože ty jsou integrovány uvnitř PHY. Jelikož konektor rofa má všechny vodiče nepoužité pro komunikaci přímo vyvedeny, je potřeba je zapojit. Střed transformátoru na straně PHY tak se uzemní kondenzátory C_{14}, C_{15} o hodnotě 100nF. Střed transformátorů na straně pro připojení UTP a nepoužité vývody konektoru byli spojeny přes rezistory 75R. Tento střed pak byl vysokonapětovým kondenzátorem uzeměn.

5.3.5 Ostatní signály

- piny LED0 a LED1 slouží pro připojení LED integrovaných v ethernet konektoru. Ty pak signalizují připojný UTP kabel, nastavenou rychlost připojení či aktivitu na lince. Diody jsou zapojeny jako plovoucí viz obrázek 5.4
- RST# pin je zapojen k PD8 GPIO pinu mikroprocesoru. Uzemněním toho pinu dojde k resetu zařízení. Proti náhodnému resetu je pin chráněn blokovacím kondenzátorem $C_{11} = 100nF$. Jeho nabití po připojení napájení zajišťuje pull-up rezistor $R_{15} = 10k\Omega$.
- INTRP pin slouží pro možnost generování přerušení pro nadřazený systém. Jelikož tento pin není součástí MII, nemá ani MAC na použitém mikroprocesoru



Obr. 5.4: Propojení LED ethernet konektoru s PHY [2]

přesně definovaný pin pro použití jako přerušení. Jako pin pro příjem přerušení tak byl použit jeden z volných GPIO pinů mikroprocesoru a to pin PB15. INTRP pin má na KSZ8051MLL také funkci zapnutí NAND Tree Mode. Ten je zapnut pokud je INTRP pin uzemněn právě při uvolnění. Aby byl při běžném zapnutí mód NAND Tree vypnut, je připojen pin INTRPT přes pull-up rezistor $R_{14} = 4k7$ k napájecímu napětí 3,3V. RST# signálu. Aby tak nedocházelo k svévolnému zapnutí NAND Tree Mode, je INTRP pin připojen přes pull-up rezistor R_{13} .

- pin REXT , pomocí něhož se nastavuje výstupní proud PHY. Dle [2] musí tento pin být spojen přes přesný 1% rezistor $6,49k\Omega$ se zemí.

5.4 Analogová část

Pro rozšíření možnosti desky jsem se rozhodl navrhnout analogovou část, která by umožnila získávat parametry reálného průběhu jednotlivých signálů na sběrnici CAN. Těch by bylo možné využít pro autodetekci přenosové rychlosti, chybného nastavení přenosové rychlosti jednotlivých uzlů a jiných chybových stavů na sběrnici CAN. Jelikož STM32F407 disponuje třemi ADC, rozhodl jsem se je využít a navrhnout analogovou část tak, aby mohla měřit úroveň vodičů CAN_H a CAN_L a rozdíl mezi nimi. Protože využití ADC jsou schopny vzorkovat signál s frekvencí 1MHz, což je schodné s maximální rychlostí sběrnice CAN, bylo upuštěno od časové rekonstrukce signálů naměřených ADC. Pro možnost volby odstartování měření byl

přidán triger. Navrženou analogovou část je možné rozdělit na čtyři bloky.

- Impedanční oddělení
- Rozdílový zesilovač
- Trigger
- Zdroj komparačního napětí pro triger

Jako impedanční oddělení od sběrnice CAN jsou určeny operační zesilovače OZ3 a OZ4, které jsou zapojeny jako napěťový sledovač. Operační zesilovač OZ4 je zapojen jako diferenciální zesilovač, umožňující přímo měřit rozdílové napětí na sběrnici CAN. Operační zesilovač OZ1 je zapojen jako komparátor mezi diferenciální napětí a referenční napětí. Jako zdroje referenčního napětí je použit operační zesilovač OZ5 zapojený jako dolní propust.

5.5 PWR

Deska CanShark je navržena pro vstupní napájecí napětí 12V. Po výměně vstupních kondenzátorů zdrojů (C29 a C38) za typy pro vyšší napětí lze použít napájecí napětí až 45V. Pokud jsou osazeny vysokonapěťové verze obvodů spínaných zdrojů (LM2594HV-ADJ), které jsou pinové kompatibilní s použitými LM2594M-ADJ, je možné použít napájecí napětí až 60V.

Pro přivedení napájecího napětí je určen souosý napájecí konektor 5,5 x 2,1mm. Napájecí napětí je zároveň vyvedeno na pin 4 konektorů pro připojení sběrnice CAN. Zařízení tak lze použít ve dvou módech v závislosti na velikosti připojované sběrnice a dostupnosti napájecích napětí.

- Připojení se k existující síti. V tomto případě je možné napájet CanShark z napájecího napětí ostatních CAN zařízení. Není tak zapotřebí externí zdroj.
- Testování jednotlivých CAN zařízení. V tomto případě je možné napájet desku CanShark z externího adaptéru a k napájení testovaných zařízení využít vývod napájecího napětí na konektorech CAN.

5.5.1 Zdroj 3.3V

Protože všechny obvody (kromě části ADC) pracují s napájecím napětím 3,3V, byl navržen následující zdroj. Používá obvodu TI LM2594M-ADJ, což je spínaný regulátor pracující na frekvenci 150kHz jehož výstupní napětí je nastavitelné za pomoci rezistorového děliče. Maximální výstupní proud tohoto regulátoru je pak 0,5A. Samotný obvod pro svou správnou funkci potřebuje pouze několik externích součástek. Postup jejich volby je dopodrobna popsán v datasheetu. Postup volby součástek pro můj případ je popsán v následujícím textu. Ze zdroje budou napájeny obvody MCU STM32F407VG, Ethernet PHY KSZ8051MLL, oba budiče sběrnice

CAN SN65HVD a signalizační diody. Předpokládaný odběr jednotlivých čipů obsahuje následující tabulka. Jedná se o maximální hodnoty uvedené v datasheetech k jednotlivým součástkám

Součástka	Maximální odběr [mA]
STM32F407VG	150
KSZ8051MLL	50
2x SN65HVD	2x 48
LED	50mA
celkem	346

Celkový odběr součástek z 3,3V napájecí větve by neměl překročit 346mA. Zroj byl navržen na oběry 400mA, disponuje tak dostatečnou proudovou rezervou. Správná funkčnost zdroje je signalizována zelenou diodou LED2.

Volba výstupního napětí: Výstupní napětí je nastaveno poměrem dvojice rezistorů ve zpětné vazbě. Hodnota výstupního napětí je dána vztahem

$$V_{OUT} = V_{ref} \cdot \left(1 + \frac{R_2}{R_1}\right) \quad (5.2)$$

Kde $V_{ref} = 1,23V$. Hodnotu R_1 volíme v rozsahu 240Ω až $1,5k\Omega$ tak, aby nám vyšla dostupná velikost R_2 . Pokud zvolíme $R_1 = 1000\Omega$ pak

$$R_2 = R_1 \cdot \left(\frac{V_{OUT}}{V_{ref}} - 1\right) = 1000 \cdot \left(\frac{3,3}{1,23} - 1\right) = 1,68k\Omega \quad (5.3)$$

.

Výběr indukčnosti: Výhodnou výstupní cívku vybíráme dle grafu č.7 v [?] na základě odebíraného výstupního proudu a E.T konstanty.

$$E \cdot T = (V_{IN} - V_{OUT} - V_{SAT}) \cdot \frac{V_{OUT} + V_D}{V_{IN} - V_{SAT} + V_D} \cdot \frac{1000}{150} \quad (5.4)$$

Kde $V_{SAT} = 0,9V$ a $V_D = 0,5V$. Pro můj případ tedy

$$E \cdot T = (12 - 3,3 - 0,9) \cdot \frac{3,3 + 0,5}{12 - 0,9 + 0,5} \cdot \frac{1000}{150} = 17,034V \cdot \mu s \quad (5.5)$$

V grafu č.7 v [?] pak vyčteme že na základě $E \cdot T = 17,034V \cdot \mu s$ a očekávaného proudu do 400mA je nevhodnější výstupní cívka L20 s indukčností $100\mu H$ dimenzovaná na dostatečnou proudovou zátěž.

Výběr výstupního kondenzátoru: Výstupní kondenzátor může nabývat hodnot v rozsahu $82\mu F$ až $220\mu F$. Měl by být dimenzován na napětí nejméně 1,5 krát větší než výstupní napětí a měl by mít co nejmenší ESR. Pro jednodušší volbu je k dispozici tabulka č.3 v V té lze dle hodnoty výstupního napětí nalézt doporučené hodnoty C_{VYST} . Nejbližší hodnota pro napětí 3.3V je $100\mu F$. Použit byl tedy tantalový kondenzátor $100\mu F/16V$.

Kondenzátor ve zpětné vazbě: Pro zvýšení kvality regulace výstupního napětí má být pro výstupní napětí vyšší než 10V ve zpětné vazbě umístěn kondenzátor o hodnotách 50pF až 10nF. Jelikož je výstupní napětí 3.3V nebyl tento kondenzátor do návrhu přidán.

Dioda: Výběr vhodné diody je dán jejími parametry. Musí se jednat o rychlou Schottkyho diodu dimenzovanou na 1,3krát větší proud, než je maximální proud zátěže. Závěrné napětí pak musí být minimálně 1,25-ti násobkem napětí vstupního. Toto vše splňuje dioda SK26A.

Vstupní kondenzátor: Jako vstupní kondenzátor lze dle datasheetu [1] použít elektrolytický či tantalový kondenzátor s co nejnižším ESR. Měl by zvládat RMS proud do výše 1/2 výstupního proudu a napětově by měl být koncipován na 1,5 násobek maximálního vstupního napětí. Jako vstupní kondenzátor byl zvolen tantalový kondenzátor $100\mu F/20V$.

5.5.2 Zdroj pro analogovou část

Pro výše popsanou analogovou část je zapotřebí napájecí napětí 10V. Navržená deska tak obsahuje druhý spínaný zdroj s integrovaným obvodem TI LM2594M-ADJ. Postup návrhu je stejný jako u 3,3V zdroje. Předpokládaná proudová zátěž zdroje je do 100mA.

Volba výstupního napětí: Pokud zvolíme $R_1 = 1,1k\Omega$ pak

$$R_2 = R_1 \cdot \left(\frac{V_{OUT}}{V_{ref} - 1} \right) = 1000 \cdot \left(\frac{10}{1,23} - 1 \right) = 7843\Omega \quad (5.6)$$

. V řadě přesných rezistorů E24 jsou této hodnotě nejbližší rezistory $7,5k\Omega$ a $8,2k\Omega$. Osazením rezistoru $8,2k\Omega$ dojde dle 5.2 ke zvýšení výstupního napětí na 10,4V. To je vhodnější varianta, neboť zdroj tak bude kompenzovat útlum operačních zesilovačů.

Výběr indukčnosti: Výhodnou výstupní cívku vybíráme dle grafu č.7 v [1] na základě odebíraného výstupního proudu a E.T konstanty. Dosazením do 5.5

$$E \cdot T = (12 - 10 - 0,9) \cdot \frac{10 + 0,5}{12 - 0,9 + 0,5} \cdot \frac{1000}{150} = 6,638V \cdot \mu s \quad (5.7)$$

V grafu č.7 v [1] lze pro $E \cdot T = 6,638V \cdot \mu s$ a předpokládaný proud do 100mA je nejvhodnější cívka L4 s indukčností $68\mu H$ a proudovou zatížitelností 0,32A.

Výběr výstupního kondenzátoru: Jako výstupní kondenzátor je v souladu s tabulkou č.3 v [1] použit tantalový kondenzátor $100\mu F/16V$.

Kondenzátor ve zpětné vazbě: Pro zvýšení kvality regulace výstupního napětí má být pro výstupní napětí vyšší než 10V ve zpětné vazbě umístěn kondenzátor o hodnotách 50pF až 10nF. Tabulka č.3 v udává doporučené hodnoty pro jednotlivá výstupní napětí. Nejblíže navrhovanému napětí 10V je v tabulce uvedených 9V, pro které je doporučen kondenzátor o kapacitě 3,3nF.

Dioda: Podmínky pro výběr výstupní shottkyho diody splňuje i pro výstupní napětí 10V dioda SK26A.

Vstupní kondenzátor: Vstupní kondenzátor byl z důvodu použití co nejmenšího množství různých typů součástek zvolen stejný jako u 3,3V zdroje tj. tantalový kondenzátor v pouzdře C s hodnotou $100\mu F/20V$.

5.6 Návrh desky plošných spojů

Desku plošných spojů jsem navrhl jako dvouvrstvou s rozměry 97mm x 80mm. Jako vhodnou krabičku jsem zvolil plastovou krabičku KM35. Jedná se o plastovou krabičku z materiálu ABS s vyjímatelnými čely. Do těch jsou vyříznuty otvory pro připojení konektorů. Ve víčku krabičky jsou vyvrtány tři díry pro signalizační LED. Výkres s rozměry krabičky je ke stažení z [7]. Přesné rozměry navržené DPS jsou umístěny v příloze této práce.

6 VÝROBA A OŽIVENÍ DPS

Navrženou desku plošných spojů jsem nechal vyrobit u firmy PRINTED s.r.o. Následně jsem si ji svépomocí osadil součástkami. Během osazování jsem zjistil následující problémy s navrženou deskou:

- V návrhu chybí kondenzátor ve zpětné vazbě zdroje 10V. Jelikož je pro 10V tento zpětnovazební kondenzátor výrobcem doporučován, byl na navržené desce provizorně připájen dohromady s rezistorem R30 na jedno pájecí místo.
- Navrhovaný IO operačních zesilovačů MCP602 nepodporuje napájecí napětí 10V. Na DPS proto byl osazen pinově kompatibilní OZ LM2904QDR. Ten sice nedosahuje takových parametrů jako navrhovaný MCP602 (bandwidth pouze 700kHz a maximální změna výstupního napětí pouze $0,3V/\mu s$). Tyto parametry jsou však pro navržený generátor referenčního napětí pro komparátor dostačující.
- Navržené vstupní kondenzátory $100\mu F/20V$ se mi pro velikost pouzdra C nepodařilo sehnat. Provizorně jsem proto použil kondenzátory $100\mu F/16V$, které jsou plně funkční, avšak jsou náchylnější k poškození zvýšením napájecího napětí na vstupu desy CanShark.
- Byla navržena špatná orientace napájecího konektoru (Top vs Bottom). Protože však deska nevyužívá třetí rozpínací pin, lze tento konektor otočit.

6.1 Testování zapojení

Ihned po osazení jsem provedl kontrolu správného zapájení za pomoci multimetru. Kontroloval jsem také případný zkrat mezi piny KSZ8051MLL a dalších IO. Protože bylo vše bez problémů, přistoupil jsem k testování pod napětím. Po připojení napájecího napětí jsem zkontroloval napětí jednotlivých zdrojů. Změřil jsem také napětí 1,2V generovaného interními zdroji KSZ8051MLL na pinech VDD_1.2 a STM32F407VGT6 na pinech VCAP_1 a VCAP_2.

Jednou z vlastností KSZ8051MLL, kterou je možnost využít k testování funkce je generování hodin na pinech MII_TXC a MII_RXC. Na těchto pinech je při správné funkci KSZ8051MLL ihned po resetu generován signál 25MHz, který spolehlivě značí správné ožítí PHY po resetu.

Poslední fází provedeného testu je využití v PHY integrované funkce NAND_TREE pro testování správného spojení pinů mikroprocesoru a PHY.

6.1.1 Testování s pomocí NAND Tree

Jak již název funkce napovídá, funguje na principu hradel NAND zapojených do řetězu. Každé hradlo tak má jako jeden vstup daný pin a jako druhý vstup výstup předchozího hradla. Uvedeme-li při startu všechny piny do logické '1', pak pokud při změně prvního pinu z '1' do '0' musí dojít ke změně na výstupu z '0' na '1'. Při uzemnění dalšího pinu se stav výstupu opět změní zpět. Pokud při změně každého vstupního pinu dojde ke změně na výstupu, je možné prohlásit všechny piny za správně připojené. Pořadí pinů, které je možné tímto způsobem testovat je uvedeno v [2].

Pro možnost testování za pomoci NAND Tree jsem vytvořil jednoduchý program. Ten nastaví pin INTRP do logické '0' a provede reset PHY. Tím se spustí NAND Tree mode. Poté program nastaví všechny piny, které je možné testovat do logické '1'. Následné přepínání jednotlivých pinů do logické '0' je potřeba krokovat debuggerem. Program zároveň provádí vyhodnocení změny vstupního pinu oproti předchozímu stavu.

7 MEŘENÍ PARAMETRŮ DESKY

Pro ověření parametrů navržené desky jsem provedl následující měření popsaná v následujících podkapitolách.

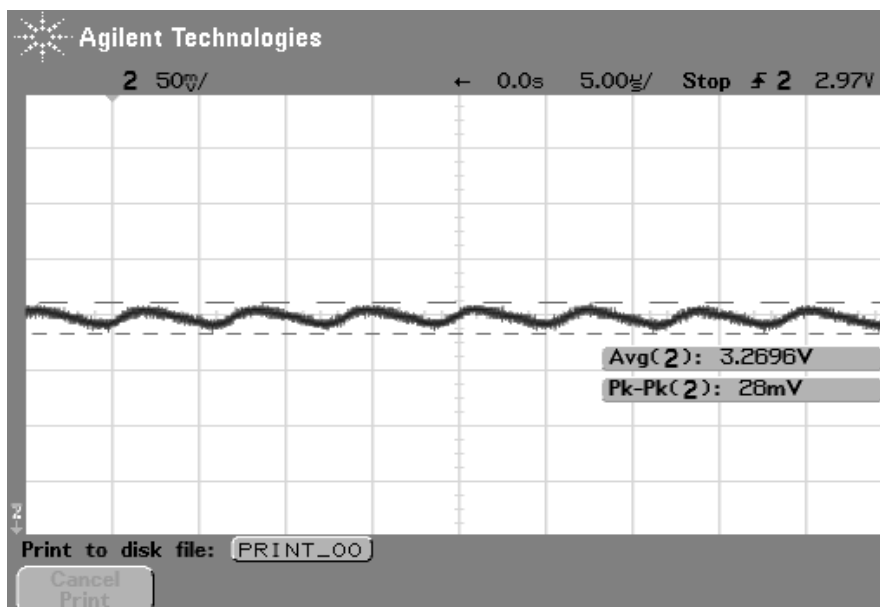
Všechna data byla naměřena osciloskopem Agilent serial number: MY42000909.

7.1 Měření napájecích zdrojů

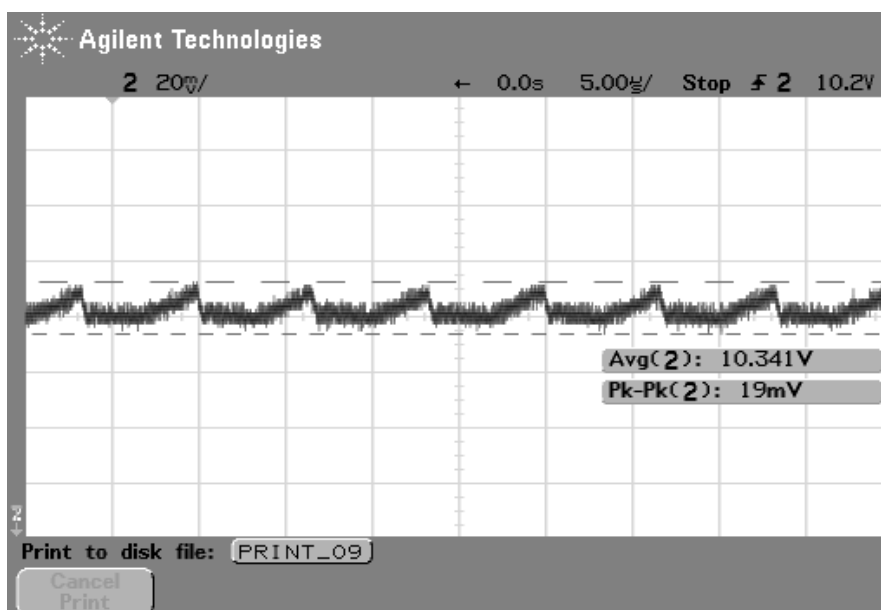
Měření výstupního napětí napájecích zdrojů bylo provedeno při zátěži těchto zdrojů způsobené běžným provozem MCU, které odesílalo po Ethernetu data přijatá na sběrnici CAN. Naměřený průběh výstupního napětí je zobrazen na obrázku 7.1 pro zdroj 3.3V a na obrázku 7.2 pro zdroj 10V. Na základě hodnot v tabulce 7.1 lze navržené zdroje považovat za dostatečně robustní pro danou aplikaci.

Navržené napětí dle vzorce 5.2	3,2964V	10,3990V
Změřené napětí	3,2696V	10,341V
Odchylka	-0,81%	-0,56%
Zvlnění ΔU	28mV	19mV
Činitel zvlnění	0,86%	0,18%

Tab. 7.1: Naměřené a vypočtené hodnoty pro napájecí zdroje



Obr. 7.1: Změřené zvlnění napájecího napětí 3,3V



Obr. 7.2: Změřené zvlnění napájecího napětí 10V

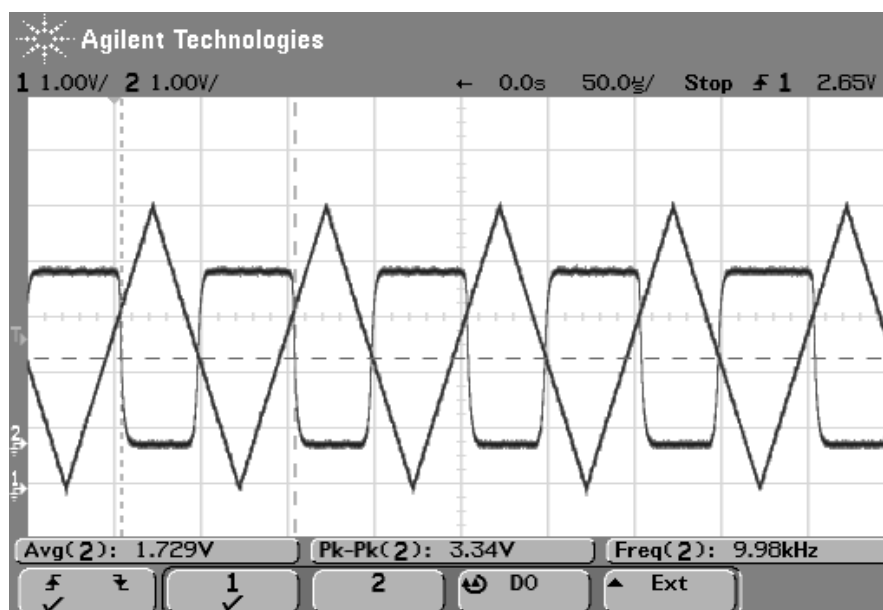
7.2 Měření analogové části

Měření analogové části lze rozdělit do dvou částí. V první části byl měřen blok, generující na základě střídavého signálu generovaného procesorem napětí, definující komparační úroveň trigovacího komparátoru. Tabulka 7.2 zobrazuje nastavené a naměřené hodnoty.

Střída signálu	Naměřené napětí	Zvlnění
10%	1,0146V	10,341V
20%	1,9640V	30mV
50%	4,8880V	41mV%
80%	7,7940V	34mV
100%	9,1887V	23mV

Tab. 7.2: Naměřené a vypočtené hodnoty pro napájecí zdroje

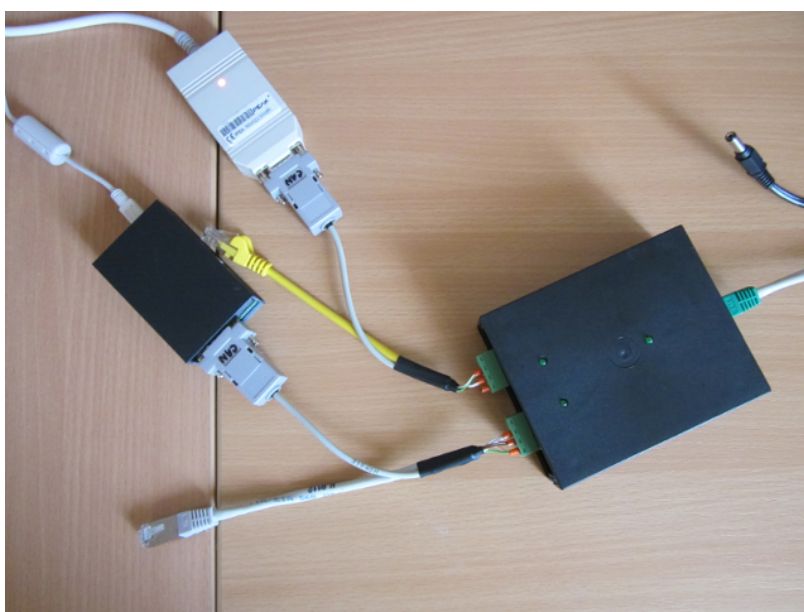
Druhou fází pak bylo vlastní měření komparátoru. Na vstupní vodiče byl přiveden signál pilovitého průběhu s amplitudou 5V. Komparátor byl nastaven na napětí 2,5V. Průběh sepnutí komparátoru je zobrazen na obrázku



Obr. 7.3: Vstupní pilovité napětí a výstup triggeru analogové části

7.3 Zkouška příjmu a odesílání zpráv

Ověření správné funkčnosti celé desky s finálním firmwarem jsem provedl při zapojení dle obrázku 7.4. Data na CAN vstupy byla generována připojenými převodníky PCan USB dongle od firmy PEAK System a převodníkem CAN2Usart. Rychlost sběrnice byla nastavena na 1Mbps a byla na ní odeslána dávka jednoho tisíce zpráv v maximální možné rychlosti. Na PC připojeném přes ethernet s deskou CANShark jsem pomocí aplikace canshark-console ověřil přijmutí všech zpráv.



Obr. 7.4: Zapojení při zkoušce přenosu zpráv maximální rychlostí

8 NÁVRH SOFTWARE

8.1 Firmware pro CanShark board

Modul CanShark, který v této práci rozšiřuji, obsahuje funkční verzi firmware. Pro funkčnost této verze na navrhované desce bylo zapotřebí upravit pouze následující části kódu:

- Předefinování GPIO pinů MCU využitých pro připojení k Ethernet PHY.
- Předefinování GPIO pinů použitých pro CAN
- Předefinování pinů a funkčnosti LED

Všechny piny byli předefinovány dle rozvržení navrženého za pomoci STM32CubeMX. U LED signalizující průbeh hlavní smyčkou byla přidána proměnná sloužící jako předdělička. Ta snižuje frekvenci blikání tak, aby byla postřehnutelná.

8.2 Software pro PC

8.2.1 Aktuální verze

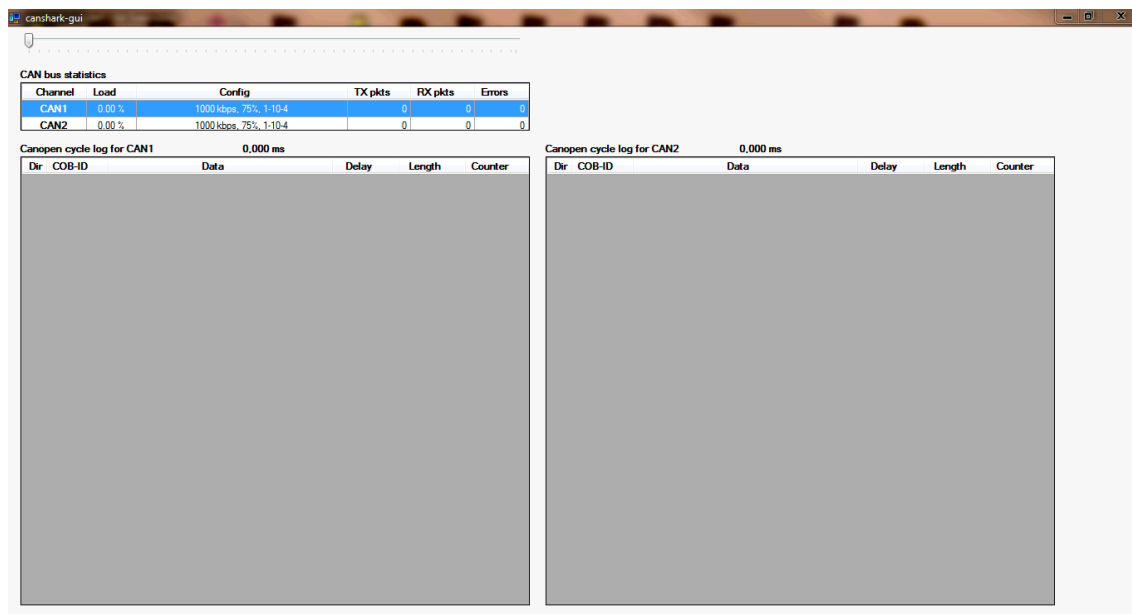
Aktuální verze CanShark-GUI programu je zobrazena na obrázku 8.1. Obsahuje tabulku zobrazující počty přijatých a odeslaných zpráv na jednotlivých sběrnicih. Pod ní jsou pak umístěny dvě tabulky, ve kterých jsou zobrazeny zprávy přijaté na dané sběrnici během jednoho CANOpen cyklu. Jeho délka je zobrazena nad každou z tabulek.

8.2.2 Provedené úpravy

Po konzultaci s vedoucím práce jsem se rozhodl do programu přidat grafické zobrazení provozu na sběrnici. To zobrazuje přijímané zprávy na základě četnosti výskytu jednotlivých identifikátorů zpráv za daný časový úsek. Tím je umožněno se rychle zorientovat ve stavu proudících dat na sběrnici, a tím případně určit i vyšší komunikační vrstvu na sběrnici použitou. Návrh je rozdělen do dvou částí. První část tvoří třídu pro měření četnosti výskytu jednotlivých identifikátorů mezi přijatými zprávami, druhá část pak zajišťuje jejich zobrazení.

Návrh třídy CanBusHistogram

Tato třída slouží k oddělení části zpracování dat o části vizualizace. Jejím úkolem je měřit četnost výskytu jednotlivých zpráv a to se rozdělením na zprávy standartní a rozšířené viz. CAN. Třída umožňuje nastavení práce ve dvou módech:



Obr. 8.1: Okno programu canshark-gui před provedením úprav

- Kumulační mód - tento mód slouží k zobrazení veškerého dění na sběrnici od startu programu po jeho ukončení či vymazání naměřených dat tlačítkem reset.
- Plovoucí mód - slouží k zobrazení aktuálního stavu na sběrnici. Data jsou v tomto módu zobrazována pouze po přesně nastavenou dobu.

Mezi těmito módy lze přepínat voláním metody `ChangeDeleteMode`. Její parametry popisuje tabulka 8.1.

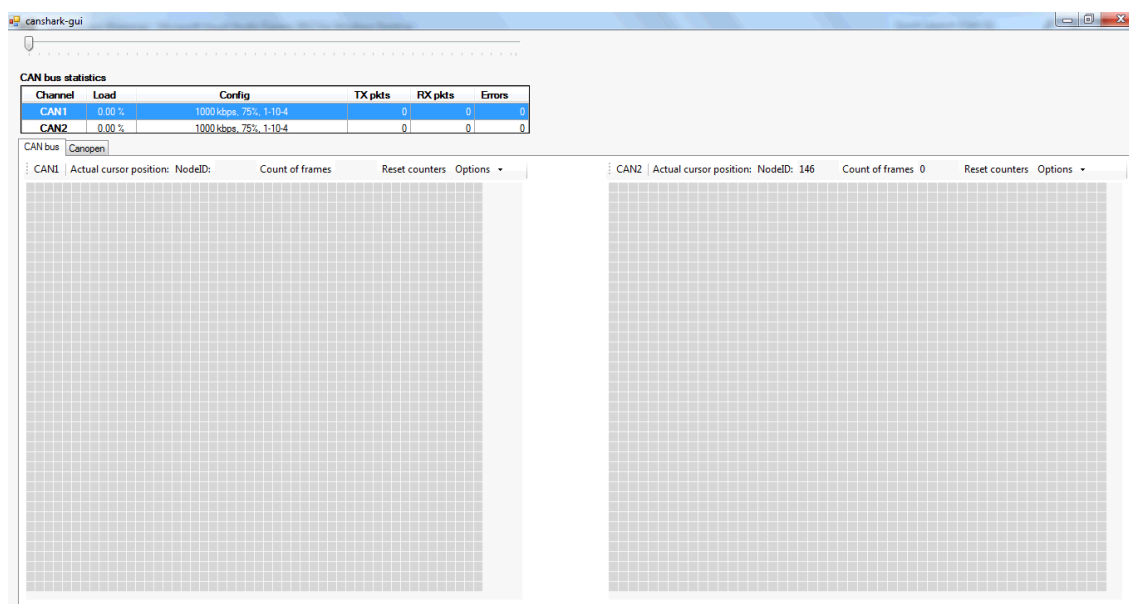
Parametr	datový typ	Popis funkce
<code>new_state</code>	<code>bool</code>	Tato hodnota určuje, zda bude zapnut Plovoucí mód. Nabývá li hodnoty <code>True</code> , pak bude zapnut.
<code>Delete_time</code>	<code>uint</code>	Určuje délku zobrazovaného okna v ms. Data starší jak nastavený čas jsou mazána.
<code>StepTime</code>	<code>uint</code>	Určuje časový interval v ms, s jakým se okno posouvá nad daty přijatými ze sběrnice.

Tab. 8.1: Parametry metody `ChangeDeleteMode`

Přehled dalších metod, které poskytuje třída `CanBusHistogram` pro komunikaci s okolím zobrazuje tabulka 8.2.

ReceivedMessage	Je volána z nadřezené aplikace. Jako parametr dostává přijatou zprávu, kterou zpracovává.
GetExtIDChanges	Vrací slovník obsahující dvojice extended NodeID a jeho četnost pro zobrazení.
GetStdIDChanges	Vrací slovník obsahující dvojice standard NodeID a jeho četnost pro zobrazení.
ResetCounters	Nuluje všechna uložená data.

Tab. 8.2: Metody třídy CanBusHistogram poskytované pro komunikaci s okolím



Obr. 8.2: Okno programu canshark-gui po provedených úpravách

8.2.3 Návrh komponenty Histogram

Jedná se o univerzální komponentu určenou k maticovému zobrazování dat, kde barva jednotlivých bodů určuje jejich četnost. V okně properties vývojového prostředí je možnost nastavit pro tuto komponentu počet zobrazovaných řádků a sloupců, text zobrazený v levém horním rohu komponenty a hexadecimální formát zobrazovaných dat.

Komponenta Histogram je navržena tak, že všechna data vykresluje za použití třídy Graphics do bitmapového obrázku. Velikost jednotlivých bodů v pixelech je vypočítávána na základě aktuální velikosti okna a nastaveného počtu řádků a sloupců. Tato velikost jednotlivých bodů se mění s každou změnou velikosti okna.

Barevné spektrum a jednotlivé rozhodovací úrovně jsou nastaveny v metodách

ReinitPenList a ReinitCompareValues. V současné verzi programu je nastaveno osm rozhodovacích úrovní a jim odpovídající barvy. Tyto jsou zobrazeny na obrázku

Ovládání zobrazení

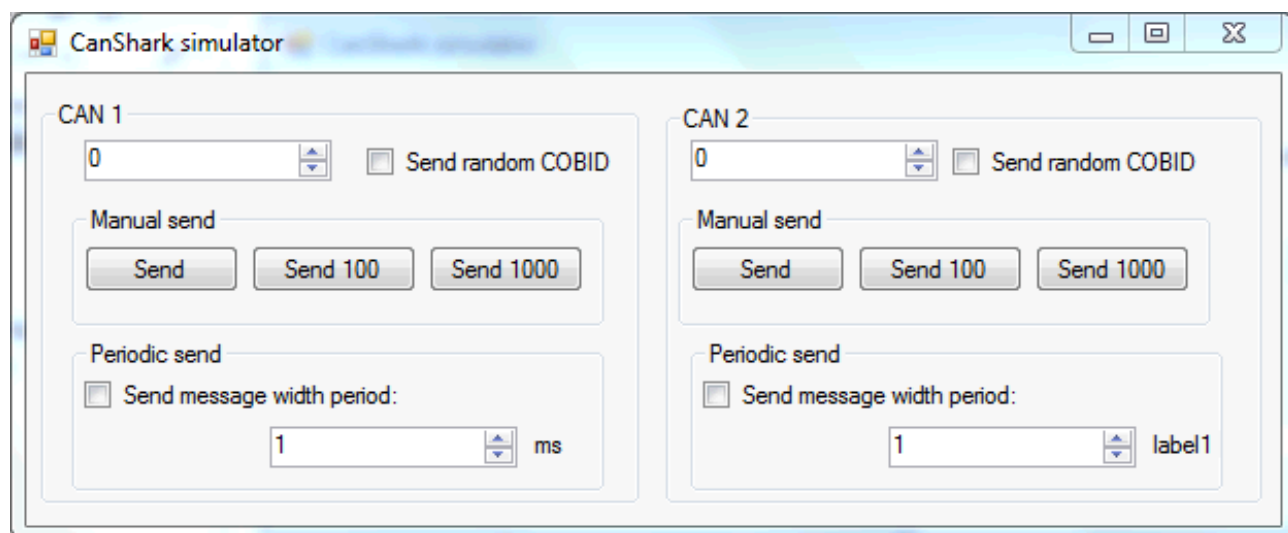
K veškeré interakci s uživatelem je použito komponenty ToolStrip umístěné nad zobrazovnými daty. Jedná se o pole pro zobrazení dat o aktuální pozici kurzoru myši. Zde se zobrazuje identifikátor zprávy a počet výskytů zprávy, kterou reprezentuje oblast grafu pod kurzorem myši umístěným nad grafem. Dále je zde pak tlačítko reset pro vynulování naměřených dat a rozbalovací menu Options. V tom je možnost přepínat mezi oběma módy práce. V případě, že je nastaven Plovoucí mód, dojde k povolení nastavování délky zobrazovaného okna.

8.3 Vývojové a testovací utility

Pro zjednodušení vývoje GUI aplikace a otestování správnosti návrhu a osazení desky jsem vytvořil následující aplikace.

8.3.1 CANBoardUDSimulator

Jedná se o jednoduchou aplikaci, umožňující simulovat UDP pakety, které posílá deska CanShark. Umožňuje posílat jednotlivé zprávy i dávky zpráv s pevně daným či náhodným identifikátorem. Lze také nastavit automatické odesílání zpráv s nastavenou periodou. Okno této aplikace je zobrazeno na obrázku ??



Obr. 8.3: Program CanShark simulator

8.3.2 Board Tester

Tento firmware je upraveným demoprogramem od firmy STMicroelectronics dodávaným k čipům STM32F407. Jeho úkolem je inicializovat a provést test NandTree, ověřit komunikaci SMI s PHY, zkontrolovat provedení Autonegotiation a vyčíst stav jednotlivých registrů PHY.

8.3.3 Board Analog Tester

Tento firmware slouží k ověření návrhu a parametrů analogové části desky CanShark. Generuje PWM signál na vstup analogové části, čímž určuje trigrovací napětí.

9 ZÁVĚR

Cílem této bakalářské práce bylo rozšířit OpenSource modul určený k odposlechu sběrnice CAN. Rozšíření spočívalo v návrhu vhodné desky plošných spojů, která je schopná přijímat data ze dvou sběrnic CAN bez výpadku při rychlosti 1Mbps. Přijatá data posílá po rozhraní ethernet do připojeného PC pro další analýzu.

V prvních kapitolách práce se věnuji teoretickým znalostem nutným ke zpracování rozšíření daného modulu. Čtenář je v nich seznámen se základními informacemi o architektuře procesorů ARM. Dále je zde základní popis mikrokontroléru STM32F407VGT6, který je použit jako řídicí část navržené desky. V další kapitole je uveden popis rozhraní Ethernet včetně popisu protokolu UDP, který je použit pro komunikaci mezi CanShark snifferem a připojeným PC. Nachází se zde také podkapitola pojednávající o nejčastějších možnostech připojení mikroprocesorů k síti ethernet. Čtvrtá kapitola obsahuje teoretický rozbor sběrnice CAN. Popis je pojat jako porovnání s univerzálním modelem ISO/OSI. Je zde popsána fyzická vrstva sběrnice, její časování a řešení přístupu k médiu. Tato kapitola též obsahuje přehled datových rámců, které se mohou na sběrnici vyskytnout a informace o zabezpečení dat proti náhodným chybám.

Kapitola 5 popisuje vlastní návrh desky plošných spojů modulu CanShark. Deska je navržena o rozměrech 97x80mm tak, aby pasovala do krabičky KM35, která byla vybrána jako nejvhodnější kandidát pro umístění desky CanShark. Navržená deska využívá mikroprocesoru STM32F407VGT6 ve spolupráci s ethernet PHY KSZ8051MLL, budiči CAN SN65HVD230 jejichž společné zapojení včetně volby součástek je v této kapitole dopodrobna popsáno. Kapitulu uzavírá popis návrhu spínaného zdroje za použití IO LM2594, který slouží jako zdroj napájecího napětí pro všechny použité součástky.

Nad rámec zadání se v kapitole 5 zabývám také návrhem analogové části. Cílem bylo rozšířit požadované schopnosti desky o možnost přímého měření průběhu signálu na vodičích sběrnice CAN. To tvoří navrženou DPS více univerzální a umožňuje to v budoucnu rozšířit analyzační SW o další funkce, jako například auto baudrate detection, detekci prohození vodičů CAN_L a CAN_H či kontrolu napěťových úrovní sběrnice CAN.

Kapitoly 6 a 7 výrobu a měření prototypu navržené desky. V kapitole 6 jsou popsány chyby návrhu, které byly objeveny během testování osazené desky. Žádná z objevených chyb však neměla vliv na celkovou funkčnost desky.

Kapitola 8 popisuje úpravy software a firmware CanShark modulu. Jelikož navržená deska se liší jen minimálně od desky, pro kterou byla vyvíjena původní verze firmware. Veškeré nutné úpravy firmware spočívaly pouze v přemapování použitých GPIO pinů pro připojení ethernet PHY, budičů CAN a signalizačních LED na

správné místo. V závěrečné kapitole jsou také popsány úpravy provedené v aplikaci canshark-gui. V rámci těchto úprav byla do dané aplikace přidána možnost sledovat průběh zpráv na sběrnici CAN. Data jsou interpretována jako histogram měnící barvu podle četnosti výskytu zprávy z daným identifikátorem za daný čas. Tento čas může být neomezený a data jsou v tomto případě archivována až do stisku tlačítka reset. Druhou možností je nastavit vlastní čas, kdy zprávou starší než daný časový interval jsou z histogramu odmazávány.

Zkušebním logováním sběrnice CAN popsaným v kapitole 7 jsem ověřil, že navržená deska splňuje zadáním na ní kladené požadavky.

Podařilo se mi navrhnout, vyrobit a oživit desku, která splňuje dané požadavky, což bylo ověřeno za pomoci zkušebního logování sběrnice CAN. Nad rámec zadání jsem doplnil navrženou desku o blok analogové části, který nabízí možnost implementace nových funkcí měření sběrnice CAN. Program canshark-gui byl úspěšně rozšířen o novou funkci, která zobrazuje data na sběrnici jako grafický histogram.

LITERATURA

- [1] TEXAS INSTRUMENTS. *LM2594/LM2594HV SIMPLE SWITCHER® Power Converter 150 kHz 0.5A Step-Down Voltage Regulator: PRODUCTION DATA information*. SNVS118C –DECEMBER 1999–REVISED APRIL 2013. [online]. [cit. 2014-05-20]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm2594.pdf>
- [2] MICREL. *KSZ8051MLL: Data Sheet*. [online]. [cit. 2014-05-20]. Dostupné z: http://www.micrel.com/_PDF/Ethernet/datasheets/ksz8051mll.pdf
- [3] STMICROELECTRONICS. *STM32F405xx STM32F407xx: Datasheet - production data*. DocID022152 Rev 4. 185 s. [online]. [cit. 2014-05-20]. Dostupné z: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf>
- [4] STMICROELECTRONICS. RM0090 Reference manual: STM32F40x, STM32F41x, STM32F42x, STM32F43x advanced ARM-based 32-bit MCUs. DocID018909 Rev 7. [online]. [cit. 2014-05-20]. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/reference_manual/DM00031020.pdf
- [5] CAN in Automation. [online]. [cit. 2014-05-20]. Dostupné z: <http://www.can-cia.org/>
- [6] Cortex-M4 Processor. [online]. [cit. 2014-05-20]. Dostupné z: <http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php>
- [7] KRABICKY.CZ. *Výkres krabičky KM35*. [online]. [cit. 2014-05-26]. Dostupné z: http://www.krabicky.cz/media/catalog/product/cache/1/image/cf117fd24af9920a43e8b800437aaf7a/k/m/km_35.jpeg
- [8] ARM.com. [online]. [cit. 2014-05-26]. Dostupné z: <http://arm.com>
- [9] STMICROELECTRONICS. *STM32 F4 series: product presentation*. Dostupné z: http://www.st.com/st-web-ui/static/active/en/resource/sales_and_marketing/presentation/product_presentation/stm32f4_marketing_pres.pdf
- [10] ZEŽULKA, František; HYNČICA, Ondřej. Průmyslový Ethernet I : Historický úvod. AUTOMA. 2007, 1, s. 41-43.

- [11] TEXAS INSTRUMENTS. *3.3-V CAN TRANSCEIVERS: SN65HVD230 SN65HVD231 SN65HVD232* [online]. SLOS346K MARCH 2001-REVISED FEBRUARY 2011. [cit. 20-5-2014]. Dostupné z: <http://www.ti.com/lit/ds/symlink/sn65hvd232.pdf>
- [12] ZEZULKA, František; HYNČICA, Ondřej. Průmyslový Ethernet II : Referenční model ISO/OSI. AUTOMA. 2007, 3, s. 86-90.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

UDP User Datagram Protokol - Prtokol z rodiny TCP/IP

PCB Printed circuit board - Deska pločných spojů

DPS Deska pločných spojů

MII Media Independent Interface

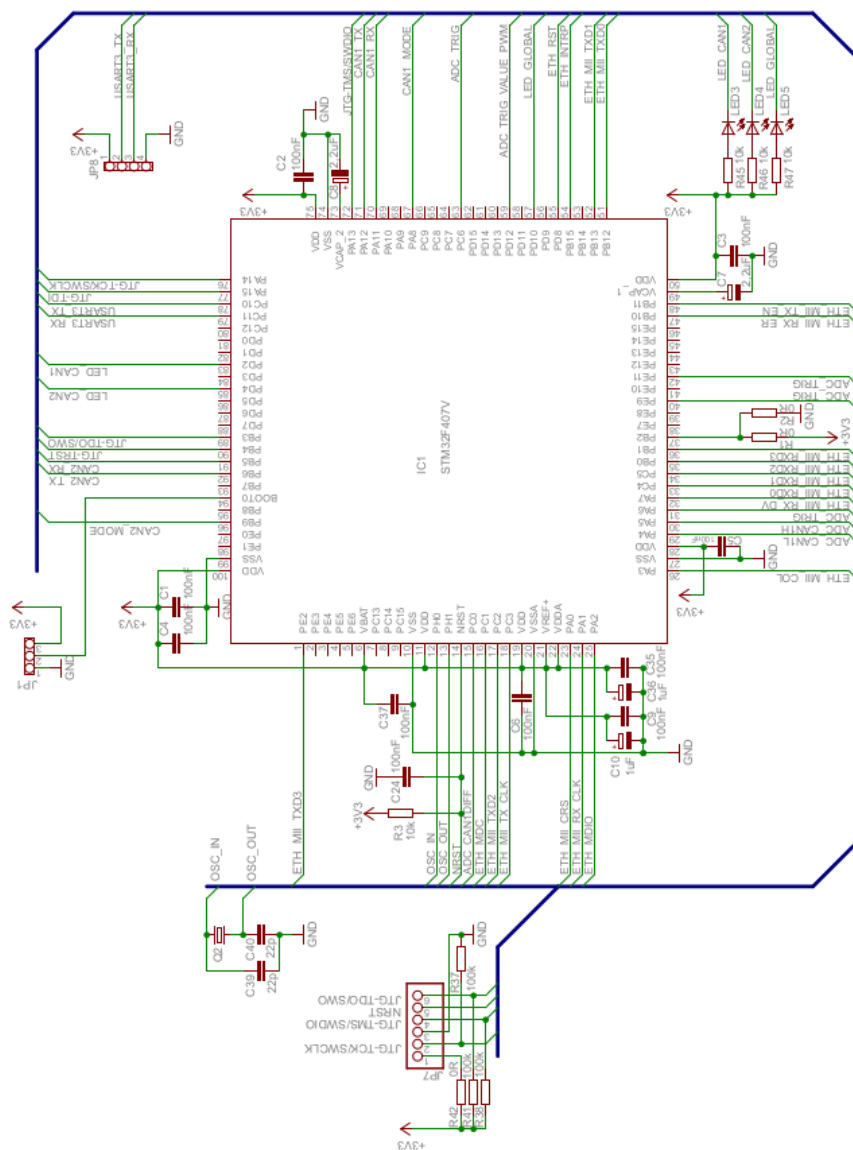
RMII Reduced Media Independent Interface

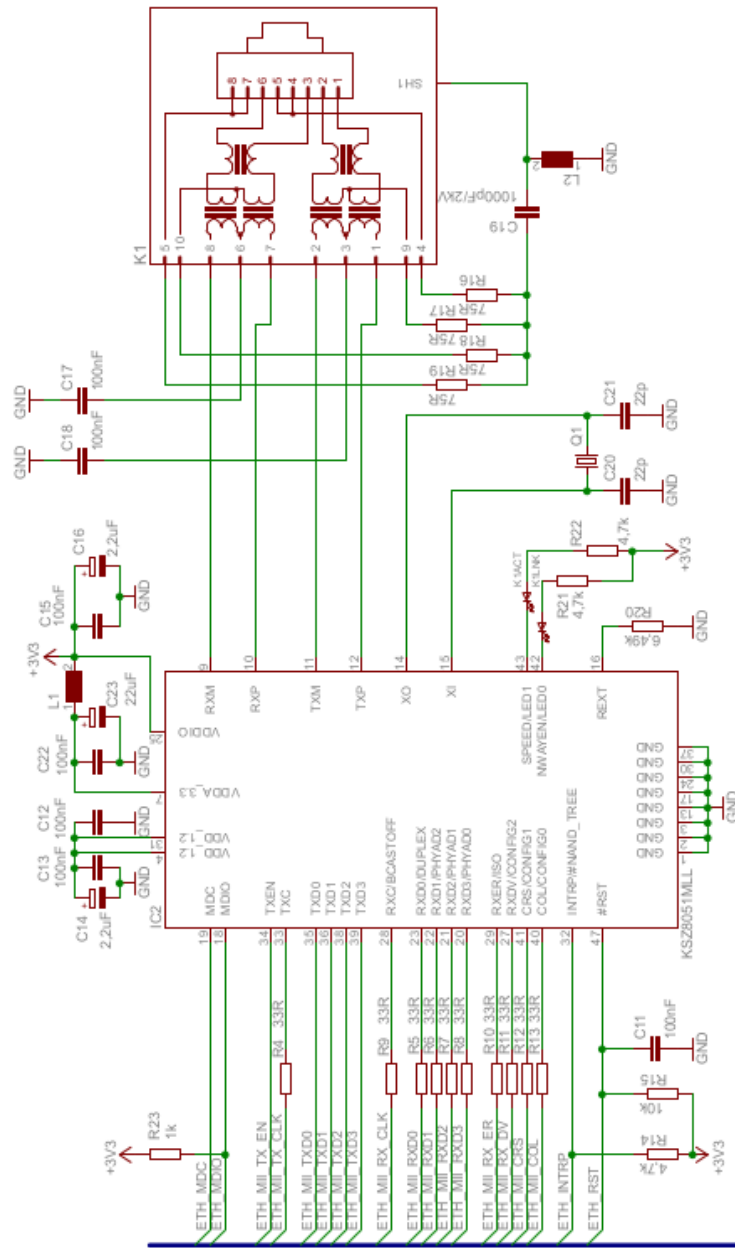
SOC System on chip

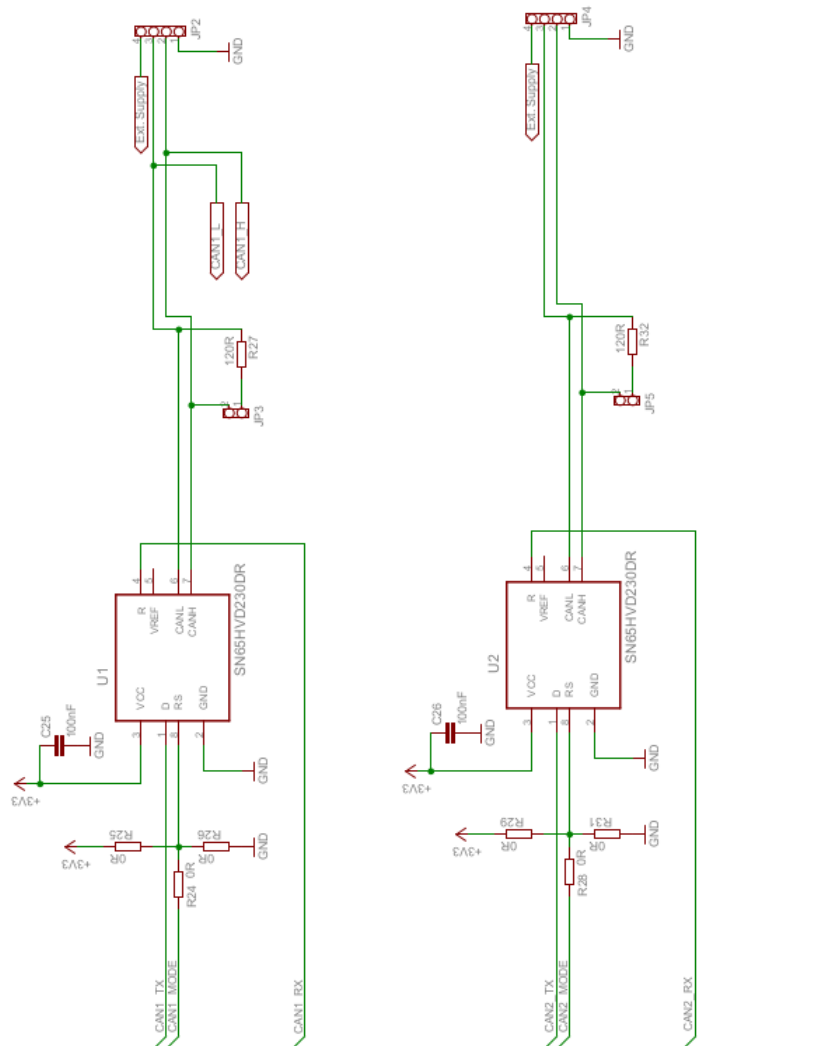
SEZNAM PŘÍLOH

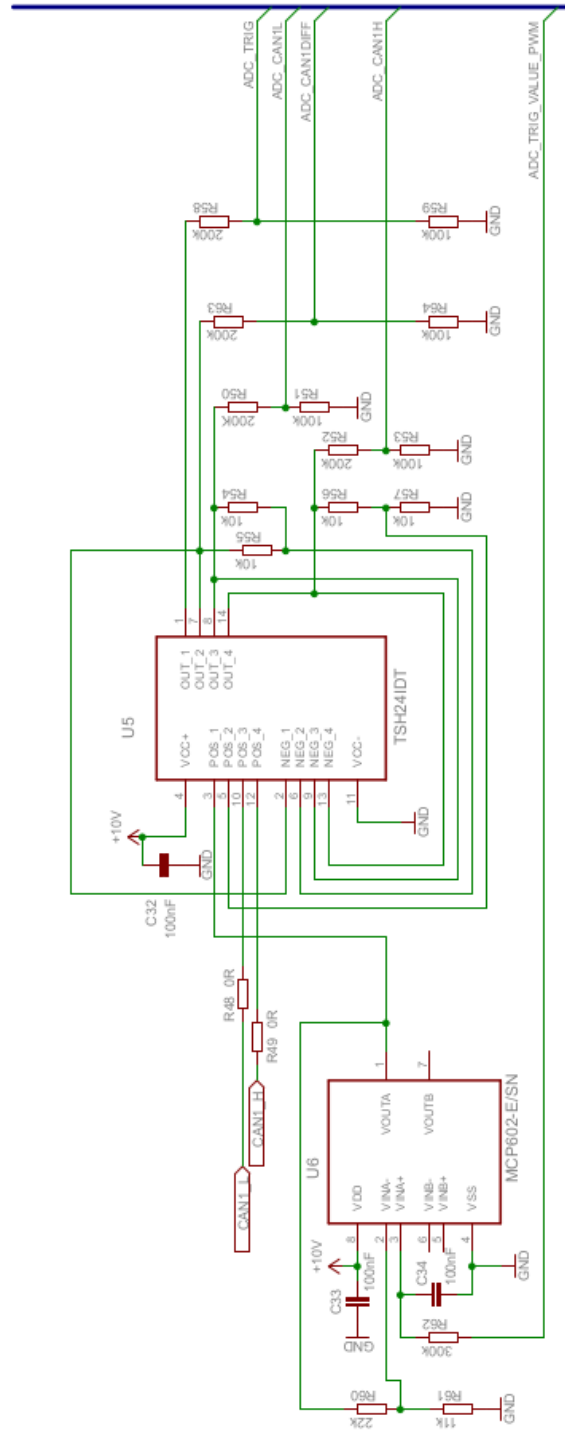
A Schéma zapojení navržené desky	55
B Vrstva TOP DPS a její osazovací plán	60
C Vrstva Bottom DPS a její osazovací plán	62
D Fotky navržené desky CanShark	64
E Obsah přiloženého CD	66

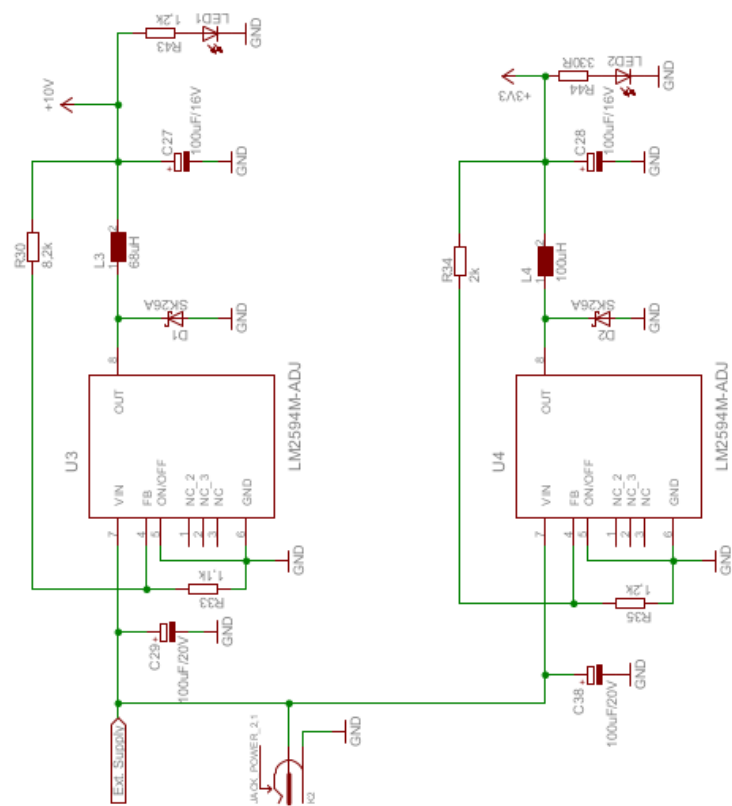
A SCHÉMA ZAPOJENÍ NAVRŽENÉ DESKY



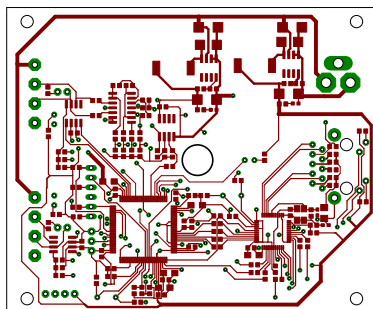


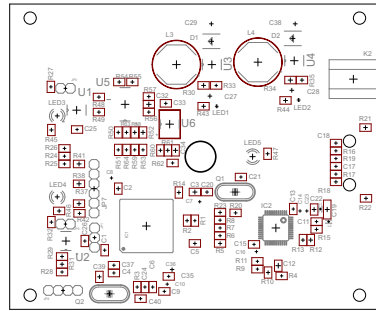




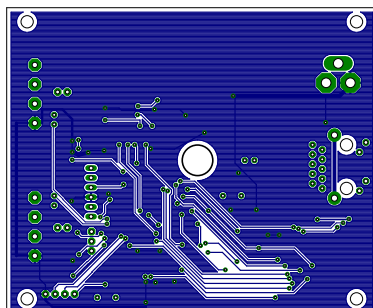


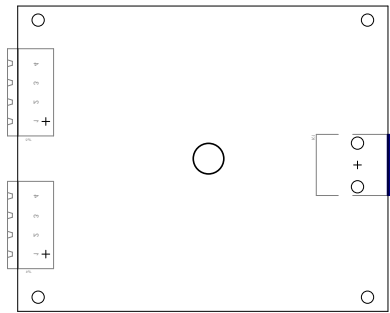
B VRSTVA TOP DPS A JEJÍ OSAZOVACÍ PLÁN



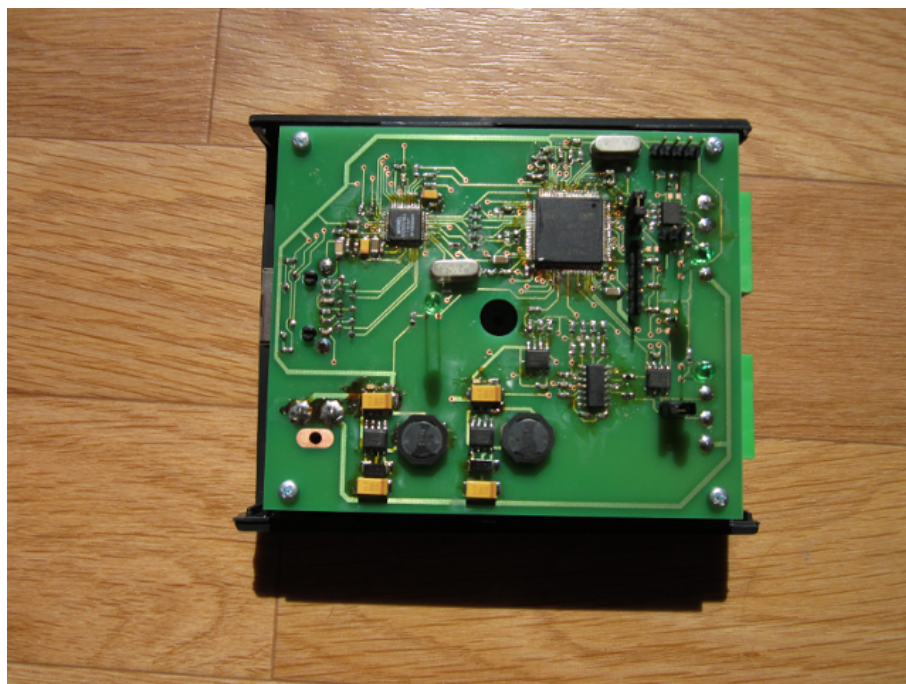


C VRSTVA BOTTOM DPS A JEJÍ OSAZOVACÍ PLÁN





D FOTKY NAVRŽENÉ DESKY CANSHARK





E OBSAH PŘILOŽENÉHO CD

- Elektronická verze práce
- GIT repository stažená z <https://github.com/xskali12/canshark>
- Návrh DPS v programu EAGLE
- Fotodokumentace