

Univerzita Pardubice

Dopravní fakulta Jana Pernera

Převodník signálů pro motorovou zkušebnu

Tomáš Čepička

Diplomová práce

2017

Univerzita Pardubice
Dopravní fakulta Jana Pernera
Akademický rok: 2016/2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Čepička**
Osobní číslo: **D14545**
Studijní program: **N3708 Dopravní inženýrství a spoje**
Studijní obor: **Elektrotechnické a elektronické systémy v dopravě**
Název tématu: **Převodník signálů pro motorovou zkušebnu**
Zadávací katedra: **Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě**

Z á s a d y p r o v y p r a c o v á n í :

Vytvořte zařízení (HW + FW) pro převod přijatých dat ze sběrnice CAN osobního vozidla na elektrické signály (napětí, frekvence) za účelem jejich zavedení do měřicího systému motorové zkušebny. Vytvořte aplikaci na PC pro ovládání převodníku.

Pomocí této aplikace bude možné převodník parametrovat, vizualizovat měřené hodnoty a logovat je do souboru.

Ověřte funkci vytvořeného zařízení a SW v praxi.

Základní funkce převodníku:

- Vyčítané veličiny z CAN: otáčky motoru, tlak v sání, součinitel lambda
- Výstupy: frekvenční, analogové, digitální
- Podporované ECU: Ignitech, Pectel, DTA
- Komunikace s PC po USB.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

HEROUT, Pavel. Učebnice jazyka C. 4., přeprac. vyd. České Budějovice: Kopp, 2004, 271, viii s. ISBN 8072322206.

VIRIUS, Miroslav. C# 2010: hotová řešení. 1. vyd. Brno: Computer Press, 2012, 424 s. K okamžitému použití (Computer Press). ISBN 9788025137307.

PLÍVA, Zdeněk. EAGLE prakticky: řešení problémů při běžné práci. 2. vyd. Praha: BEN - technická literatura, 2010, 184 s. ISBN 9788073002527.

VOSS, Wilfried. A comprehensible guide to controller area network. Greenfield: Cooperhill Technologies Corporation, 2005, 150 s. ISBN 0976511606.

Datasheety výrobců komponent.

Vedoucí diplomové práce:

Ing. Zdeněk Mašek, Ph.D.

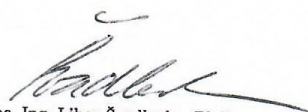
Katedra elektrotechniky, elektroniky a zabezpečovací
techniky v dopravě

Datum zadání diplomové práce:


24. listopadu 2016

Termín odevzdání diplomové práce:

26. května 2017


doc. Ing. Libor Švadlenka, Ph.D.
děkan

L.S.


Ing. Dušan Čermák, Ph.D.
vedoucí katedry

V Pardubicích dne 20. února 2017

Prohlášení autora

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury. Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích, dne 25. 5. 2017

Tomáš Čepička

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu panu Ing. Zdeňkovi Maškovi, Ph.D. za možnost pracovat s vlastním tématem práce, za jeho ochotu, rychlé reakce, připomínky a rady.

Dále bych chtěl poděkovat panu Ing. Radkovi Vancovi, za poskytnutí tématu, ochotu poradit mi v dost často složitých situacích a za předání cenných zkušeností v této velice zajímavé oblasti.

Největší poděkování však míří k mé rodině, bez jejíž podpory, ať finanční tak především duševní, bych takto velký krok nikdy nedokázal.

ANOTACE

Tato diplomová práce se zabývá návrhem a konstrukcí HW a SW a snahou vytvořit převodník, který po sběrnici CAN vyčítá data, která mu jsou poskytnuta prostřednictvím řídicích jednotek závodních vozů. Tato data zpracovává a dle požadavků uživatele data generuje v podobě fyzikálních veličin – frekvence a napětí na svých výstupech. Vytvořený převodník je postaven na mikroprocesoru Atmel ATMEGA16M1, který řídí veškeré vstupy i výstupy zařízení. Převodník je určen do motorové zkušebny ať už pro vývoj nebo přesné ladění motorů. Hlavní přínos převodníku je úspora času a minimalizace kabeláže při instalaci motoru na zkušebnu, díky sběrnici CAN. Po této sběrnici jsou data z jednotlivých senzorů zasílány. Sensory jsou myšleny snímače teploty, tlaku, kyslíkové sondy lambda čidlo otáček motoru. Jejich hodnoty jsou reprezentovány na jednom frekvenčním výstupu (pouze otáčky motoru) a na čtyřech analogových výstupech. Analogové výstupy je možno parametrizovat dle uživatelova nastavení. Tedy jinými slovy, lze volit, na jakém analogovém výstupu bude jaká veličina ze snímače reprezentována a jaký bude její rozsah.

Na analogových výstupech jsou tedy ve formě elektrického napětí v rozsahu 0÷5V generovány hodnoty ze snímačů, které lze poté měřit osciloskopem a získat tak přesné informace o stavu snímače.

Pro možnost nastavení parametrů byla navržena aplikace v jazyce C# v prostředí Visual Studio 2013. Uživatel pomocí této aplikace nastavuje požadované parametry s požadovaným rozsahem na požadovaném analogovém výstupu. Aplikace rovněž umí vyčítaná data přímo zobrazovat a logovat do souboru TXT.

KLÍČOVÁ SLOVA

Převodník, sběrnice CAN, frekvence, napětí

ABSTRACT

This diploma thesis concerns to design and to construct HW and SW in order to develop a converter that reads data via CAN bus which are received from racing cars ECU's. The data is processed by the converter and the data are generated as frequency and voltage on the converter's frequency and analog outputs by the user set. The converter is based on Atmel ATMEGA16M1 MCU whose purpose is to control the inputs and outputs. The converter's purpose is for install it into the dynamo test cell either for development or the precise tuning of the engines. The main benefit of the converter is saving time and the cables minimize during installing the engine to the dynamo test cell thanks to CAN bus. The data from the individual sensors is sent. The sensors means temperature sensors, air pressure sensors, oxygen lambda sensors and engine speed sensor.

The sensor's values are represented on the frequency output (only engine RPM) and on the four analog outputs. The analog outputs could be set by user desire. In other words, the user can choose, on which analog output will be which parameter and what range of parameter will be.

So in form of voltage in range 0 to 5V are generated the sensor's values on the analog outputs, which are measurable by oscilloscope to gain the exact information of the sensors.

It was developed a windows application for the possibility the parameters settings. The application is written in C# language in Visual Studio 2013 environment. Using this application gives the an ability to set the required parameters with the desired range to the desired analog output. The application also logs the data to the TXT file.

KEYWORDS

Converter, CAN bus, frequency, voltage

OBSAH

1	Úvod.....	- 11 -
2	Funkce a využití zařízení	- 12 -
2.1	Parametry vytvořeného převodníku CAN-data	- 12 -
3	Použité sběrnice	- 16 -
3.1	Sběrnice CAN	- 16 -
3.1.1	Struktura uzlu na sběrnici CAN	- 17 -
3.1.2	Síťový model CAN.....	- 18 -
3.1.3	Fyzická vrstva.....	- 18 -
3.1.4	Linková vrstva	- 19 -
3.1.5	Aplikační vrstva.....	- 20 -
3.1.6	Rychlosti na sběrnici	- 20 -
3.1.7	Přístup na sběrnici	- 21 -
3.1.8	Datová zpráva	- 21 -
3.1.9	Přenosové médium a konektory	- 23 -
3.1.10	Charakteristiky sběrnice CAN – shrnutí.....	- 24 -
3.2	Sběrnice SPI.....	- 24 -
3.3	Sběrnice USB.....	- 26 -
4	Vybraný hardware	- 27 -
4.1	Blokové schéma CAN-data převodníku	- 27 -
4.2	Zdroj stabilizovaného ss napětí – blok regulace napětí	- 27 -
4.2.1	Zapojení stabilizátoru napětí 7805CG.....	- 28 -
4.3	Mikroprocesor ATMEGA16M1	- 28 -
4.3.1	Parametry ATMEGA16M1	- 29 -
4.3.2	Pouzdro mikroprocesoru ATMEGA16M1	- 30 -
4.3.3	Použité vstupy a výstupy ATMEGA16M1	- 31 -
4.4	D/A převodník MCP4922	- 32 -

4.4.1	Zápis dat na D/A převodník	- 33 -
4.4.2	Parametry D/A převodníku MCP4922	- 33 -
4.4.3	Zapojení D/A převodníků v obvodu.....	- 34 -
4.5	Frekvenční výstup (spínání výstupu)	- 35 -
4.5.1	Parametry IRF7103[3].....	- 36 -
4.5.2	Zapojení frekvenčního výstupu v obvodu	- 36 -
4.6	Budič CAN MCP2551	- 37 -
4.6.1	Parametry budiče MCP2551.....	- 37 -
4.6.2	Obvodové zapojení budiče MCP2551	- 38 -
4.7	Převodník USB/UART – FT230X.....	- 38 -
4.7.1	Parametry převodníku USB/UART – FT230X	- 39 -
4.7.2	Zapojení USB/UART převodníku.....	- 39 -
4.8	Indikační LED diody a jejich funkce	- 40 -
4.8.1	Obvodové zapojení indikačních LED diod	- 42 -
5	Konstrukce zařízení.....	- 43 -
5.1	Návrh desky plošného spoje	- 43 -
5.1.1	Rozměry DPS	- 44 -
5.1.2	Konektor na DPS	- 44 -
5.1.3	Tažení vodivých cest na DPS	- 45 -
5.1.4	Osazení DPS	- 46 -
5.1.5	Oživení zařízení.....	- 47 -
5.2	Krabička.....	- 48 -
5.2.1	Umístění DPS do krabičky	- 49 -
5.2.2	Výroba víček krabičky	- 49 -
6	Programové vybavení.....	- 51 -
6.1	Nastavení alternativního programátoru USBasp	- 51 -
6.2	Firmware	- 53 -

6.2.1	Funkce <i>Main()</i>	- 54 -
6.2.2	Smyčka <i>While</i>	- 55 -
6.2.3	Funkce <i>IgnitechReading</i>	- 55 -
6.2.4	Funkce <i>DTAReading</i>	- 56 -
6.2.5	Funkce <i>setOutputs</i>	- 56 -
6.2.6	Funkce <i>UARTIncoming</i>	- 56 -
6.2.7	Funkce <i>CanBusOff</i>	- 59 -
6.2.8	Zásobník zpráv sběrnice CAN	- 59 -
6.2.9	Zachytávání zpráv ze sběrnice CAN	- 60 -
6.2.10	Filtrování zpráv na sběrnici CAN	- 60 -
6.2.11	Analogové výstupy	- 61 -
6.2.12	Frekvenční výstupy	- 61 -
6.2.13	Sériová linka UART	- 62 -
6.2.1	Ověření funkce frekvenčního a analogových výstupů	- 62 -
6.3	Software CAN-DATA Converter	- 64 -
6.3.1	Funkce programu CAN-data Converter	- 65 -
7	ZÁVĚR	- 70 -
8	SEZNAM POUŽITÉ LITERATURY	- 71 -
	PŘÍLOHA A – NÁVRH DPS	- 75 -
	PŘÍLOHA B – FOTOGRAFIE ZAŘÍZENÍ	- 76 -
	PŘÍLOHA C – vývojové diagramy	- 77 -
	PŘÍLOHA D – schémata zapojení	- 79 -

1 ÚVOD

Cílem práce je vytvořit HW a FW zařízení – převodník, pro převod přijatých dat ze sběrnice CAN osobního vozidla na elektrické signály (napětí, frekvence) za účelem jejich zavedení do měřicího systému motorové zkušebny. Práce vychází z první verze tohoto převodníku, jež byl vytvořen v rámci ročníkového projektu, ale měl ještě několik nedostatků. Mezi nedostatky patřily pevně nastavené analogové výstupy pro parametr, pevně dané rozsahy, pevně daná řídicí jednotka, z níž je po sběrnici CAN vyčítáno. Snaha je tedy vylepšit toto zařízení a zajistit možnost jeho parametrizace dle uživatelského nastavení. Nutností je přepracovat stávající hardware i firmware a vytvořit zcela nový software v podobě aplikace, která slouží k parametrování, vizualizaci a logování dat na tomto převodníku.

Práce je členěna do 6 kapitol. Kapitola 2 je věnována obecnějšímu popisu celého převodníku jako jeho funkce a využití, jeho parametry, podporované jednotky a rovněž je zde zmínka o každém parametru, který je v motorové zkušebně třeba snímat za účelem ladění motoru.

Další kapitola 3 pojednává o použitých sběrnicích na převodníku a jejich vlastnostech. Jmenovitě jsou zde postupně popsány sběrnice CAN, sběrnice SPI a sběrnice USB.

Po seznámení s použitými sběrnicemi navazuje další kapitola 4, která pojednává o vybraných hardwarových prvcích umístěných na plošném spoji převodníku. Je kladen důraz na vlastnosti a popis těchto jednotlivých prvků, popis funkce a jejich schematické zapojení v obvodu.

Pátá kapitola se již zaměřena na praktickou část práce, tedy od návrhu převodníku, jeho zhotovení, které bylo následováno otestováním a jeho oživením a umístěním do krabičky.

Poslední šestá kapitola cílí na programovou část tvorby. První část této kapitoly se zabývá tvorbou firmware, jeho funkcemi a vlastnostmi, které jsou zde popsány. Druhá část této kapitoly pojednává o uživatelské aplikaci pro parametrování převodníku a popis jeho funkce.

2 FUNKCE A VYUŽITÍ ZAŘÍZENÍ

Funkce zařízení spočívá ve vyčítání hodnot ze sběrnice CAN z řídicích jednotek závodních vozů, které následně převádí na elektrické fyzikální veličiny. Tyto signály se dále zpracovávají v měřicí ústředně motorové zkušebny. Hlavním cílem je minimalizovat čas nutný pro instalaci motoru na zkušebnu. Čas je ušetřen díky minimalizaci kabeláže ke snímačům, které poskytují sledované hodnoty. Jde o čtení dat na sběrnici CAN, jako jsou otáčky motoru, hodnota koeficientu lambda, hodnota tlaku v sání a také teploty vzduchu a teplota vody. Všechny hodnoty jsou převáděny na elektrické napětí, kromě otáček motoru. Otáčky motoru jsou generovány v podobě frekvence, jejíž hodnota je na nich přímo závislá, tento signál je poté generován na frekvenčním výstupu.

2.1 Parametry vytvořeného převodníku CAN-data

Analogové výstupy:

počet:	4
výstupní rozsah:	0 ÷ 5V
výstupní parametry:	teplota vody, teplota v sání, hodnota lambda, tlak v sání, poloha škrtkové klapky

Frekvenční výstupy:

počet:	1
výstupní rozsah:	3,3 ÷ 333Hz
výstupní parametr:	otáčky motoru

Podporované ECU: Ignitech Ignijet, DTAfast S-Series

Rozhraní: CAN, USB

Napájení: 9÷15V DC

Rozměry: Š:58 x D:54 x V:25 mm

Celkem byly zadány tři řídicí jednotky k vyčítání – Ignitech Ignijet, DTAfast S Series a PECTEL. Poslední jmenovaná PECTEL se nakonec musela vyškrtnout ze seznamu. Výrobce si svá data chrání a nebyla poskytnuta, tudíž nebylo možné určit, na jakých identifikátorech jednotka posílá hodnoty připojených snímačů.

V následujících tabulkách jsou informace poskytnuté výrobcem jednotek Ignitech a DTA, obsahují název parametru, ID parametru, pozici ve zprávě a poznámku.

Ignitech Ignijet			
Obecné parametry řídicí jednotky:			
<ul style="list-style-type: none"> ▪ Bit Rate: 500k ▪ Identifikátor: Standard – 11b ▪ Perioda zasilání zpráv: 10 Hz (100ms) ▪ Délka zprávy: 8 bajtů ▪ Byte order: LSB first 			
Parametr	Identifikátor (ID)	Data na bajtu č.:	Pozn.
Teplota vody	806	4	1b = 1°C Offset: +100°C
Teplota v sání		5	1b = 1°C Offset: +100°C
Lambda hodnota	807	0	1b = 0,1
Otáčky motoru		2, 3	1b = 1RPM
Tlak v sání		4, 5	1b = 0,1kPa (10 mbar)
Poloha škrt. klapky	809	2, 3	1b = 0,1%

Tabulka 1 – Parametry jednotky Ignitech

DTA fast S-Series			
Obecné parametry řídicí jednotky:			
<ul style="list-style-type: none"> ▪ Bit Rate: 1M ▪ Identifikátor: Extended – 29b ▪ Perioda zasilání zpráv: 10–50 Hz (20–100ms) ▪ Délka zprávy: 8 bajtů ▪ Byte order: LSB first 			
Parametr	Identifikátor (ID)	Data na bajtu č.	Pozn.
Otáčky motoru	0x2000	0, 1	1b = 1RPM
Poloha škrt. klapky		2, 3	1b = 0.1%
Teplota vody		4, 5	1b = 1°C
Teplota v sání		6, 7	1b = 1°C
Tlak v sání	0x2001	0, 1	1b = 1kPa
Lambda hodnota		2, 3	1b = 0,001

Tabulka 2 – Parametry jednotky DTA

Otáčky motoru

Frekvence tedy reprezentuje otáčky motoru, rozsah snímání otáček je dle zadání $0 \div 20000 \text{ ot./min}$. reálně ovšem nelze v této úloze spodní hranice rozsahu dosáhnout. Je to dáno nastavením pracovní frekvence mikroprocesoru a blíže je tento jev vysvětlen v kapitole 4.5 – Frekvenční výstup. Reálné hodnoty snímaných otáček jsou tedy v rozsahu $200 \div 20000 \text{ ot./min}$. Takto zvolená spodní hranice je více než dostačující, pokud bereme v úvahu, že běžné otáčky motoru na volnoběhu bývají kolem $800 \div 900 \text{ ot./min}$, záleží na konkrétním motoru. Horní hranice snímaných otáček je takto zvolena záměrně pro případ, že by se v motorové zkušebně ladily motocykly, jejichž otáčky motoru běžně dosahují vyšších hodnot, než maximální otáčky automobilových motorů. Očekávané frekvence na frekvenčním výstupu budou v rozsahu 3,33Hz pro 200 ot./min a 333Hz pro 20000 ot./min .

Koeficient lambda (λ)

Hodnotu koeficientu lambda poskytuje senzor, označovaný jako lambda sonda. Tato sonda je umístěna ve výfukovém potrubí u automobilů se zážehovými motory vybavených katalyzátorem výfukových plynů. Většinou jsou tyto sondy dvě, první je regulační a je umístěna před katalyzátorem, druhá sonda je kontrolní a je umístěna za katalyzátorem. Úkolem lambda sondy je porovnávání okolního vzduchu a výfukových plynů a na základě těchto hodnot je řídicí jednotkou vozu určen směšovací poměr paliva a vzduchu. Pokud je směšovací poměr $\lambda = 1$, jedná se o tzv. stechiometrický poměr a z hlediska emisí je to ideální stav. Znamená to, že ke spálení 1 kg benzínu je zapotřebí kolem 14,7 kg vzduchu, jinak také udávaný jako poměr vzduchu a paliva 14,7:1 (AFR – Air-Fuel Ratio). Dále může lambda sonda vyhodnotit další dva stavy. Pokud je poměr vzduchu ve směsi nižší, než u stechiometrické směsi, je směs tzv. bohatá ($\lambda < 1$). Naopak pokud je poměr vzduchu ve směsi vyšší, než poměr paliva, jedná se o tzv. chudou směs ($\lambda > 1$).

Převodník vyhodnocuje poměr vzduchu a paliva v rozsahu, který si zvolí uživatel obslužné aplikace. Zvolený rozsah minimální a maximální hodnoty je vždy přepočítán a odpovídá tak i hraničním hodnotám napětového výstupu na D/A převodníku – analogovém výstupu zařízení.

Tlak v sání

Tlak v sání patří na vozidle do skupiny nízkých tlaků dosahujících hodnot do cca 500kPa, což v jednotkách ze zadání odpovídá hodnotě 5 bar. Měření tlaku se provádí přímo, prostřednictvím deformování membrány, nebo použitím snímače síly. Snímače tlaku jsou buď dynamické, které měří pouze změny tlaku nebo statické snímače, měřící i neměnný tlak. Statické snímače mají při měření tlaků na vozidlech převažující postavení. Nejpoužívanější metodou snímání tlaku je pomocí membránových snímačů, kde tlak působí na membránu a prohýbá ji.

Na výstupu převodníku se bude generovat napětí 0V pro tlak odpovídající minimální hodnotě a napětí 5V pro tlak odpovídající maximální hodnotě zvolené uživatelem.

Teplota v sání a teplota vody

Dalším vyčítaným parametrem je teplota, kterou lze u kapalných a plyných medií měřit ve všech prostorových bodech a v případě pevných těles pouze na jejich povrchu. Jak pro případ měření teploty v sání, tak teplota vody je snímána v prostoru. Stejně jako u předchozích parametrů, tak i u obou teplot si uživatel volí v obslužné aplikaci jejich minimální a maximální hodnoty, na jejichž základě jsou poté buzeny odpovídající analogové výstupy napětím v rozsahu 0 až 5V.

Poloha škrticí klapky

Škrticí klapka je na vozidlech se zážehovými motory nezbytnou součástí pro řízení množství nasávaného vzduchu a následný správný chod motoru. Proto musí řídicí jednotka vozu dostávat informaci o její aktuální poloze. Její poloha je zpravidla snímána potenciometrem, umístěným přímo na těle klapky, na jehož výstupu je napětí, odpovídající její skutečné poloze. Tato poloha je rovněž reprezentována procentuálně v rozsahu 0 ÷ 100%. Logicky hodnota 0% odpovídá zavřené klapce a plně otevřené klapce pak odpovídá hodnota 100%.

Vypočtené a následně vygenerované hodnoty napětí na analogových výstupech tedy reprezentují vždy přijatou aktuální hodnotu lambda sondy, snímače tlaku v sání, senzorů teplot a polohu škrticí klapky ze sběrnice CAN. Aktuální hodnota otáček motoru je generována ve formě frekvence na frekvenčním výstupu.

3 POUŽITÉ SBĚRNICE

Na zařízení budou využity tři typy sběrnic, a proto je vhodné zmínit se o každé z nich. Primární použitá sběrnice, sloužící k vyčítání hodnot z řídicí jednotky vozidla je dvou vodičová sběrnice CAN. Tato sběrnice vlastně vytváří rozhraní mezi převodníkem a snímaným objektem – řídicí jednotkou vozidla. Druhá sběrnice, již se využívá ke komunikaci mezi obvody na plošném spoji, je sběrnice SPI. Sběrnice SPI je ve vytvořeném převodníku využívána k jednostranné komunikaci konkrétně k odesílání dat z mikroprocesoru do D/A převodníku, která jsou jím následně převedena na fyzikální veličiny na výstupní svorky konektoru. Třetí použitou sběrnici, která je troufám si říct nejrozšířenější sériová sběrnice, je sběrnice USB. Tato sběrnice vytváří komunikační kanál mezi převodníkem a osobním počítačem – uživatelskou aplikací. Úloha spojení s osobním počítačem spočívá v možnosti parametrizovat zařízení: volbou řídicí jednotky, volbou rychlosti CAN sběrnice, přiřazení parametru a jeho rozsahu k danému analogovému výstupu.

3.1 Sběrnice CAN

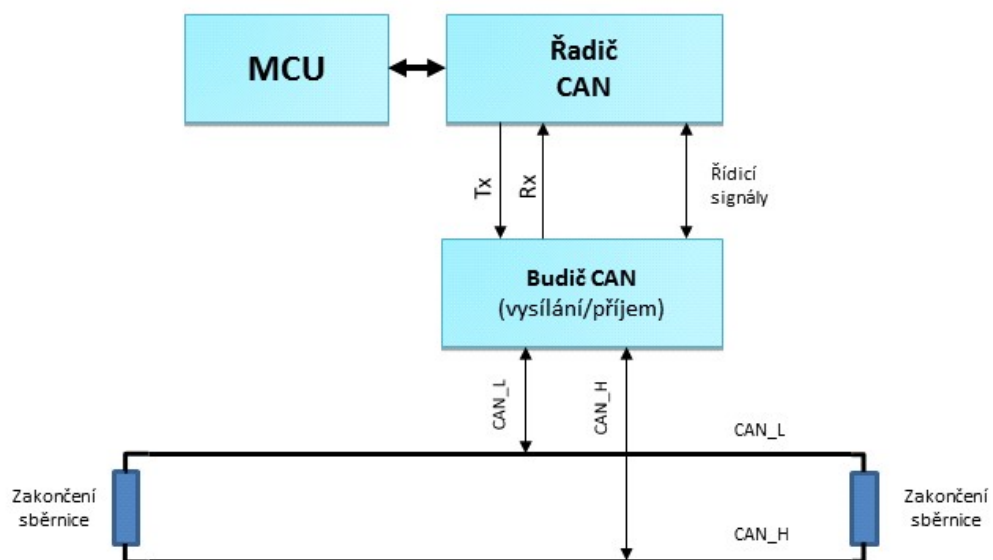
Historie této sběrnice sahá do 80. let minulého století. Vývojáři firmy BOSCH měli tehdy za úkol vytvořit sběrnici, která by byla využitelná v neustále se rozvíjejícím automobilovém průmyslu. Pod pojmem sběrnice je myšlena určitá soustava vodičů, která má určitý počet vodičů. Zařízení, která jsou na této sběrnici připojena, jsou díky tomu schopna mezi sebou komunikovat. Vývoj podobné sběrnice byl naprosto přirozeným krokem z nutnosti snímání více a více údajů ze senzorů v automobilu. Sběrnice CAN je v automobilovém průmyslu využívána pro meziprocesorovou komunikaci mezi jednotlivými řídicími jednotkami v automobilu a u modernějších vozidel i ke komunikaci mezi jednotlivými senzory a jejich jednotkami. Díky svým výhodným vlastnostem se dnes využívá i v dalších oblastech např. v průmyslové automatizaci, senzorové technice, automatizaci budov, vlakové dopravě, automobilové nákladní dopravě aj.

CAN je sériový komunikační protokol. CAN (*z angl. Controller Area Network*) umožňuje efektivně provádět řízení systémů v reálném čase s přenosovou rychlostí do 1Mbit/s při vysokém stupni zabezpečení přenosu proti chybám. Jedná se o protokol typu multi-master, kde každý uzel sběrnice může být master a řídit tak chování jiných uzlů. Není tedy nutné řídit celou síť z jednoho nadřazeného uzlu.

Přináší to zjednodušení řízení a zvyšuje spolehlivost a nezávislost. Nezávislost, že při poruše jednoho uzlu může zbytek sítě pracovat bezchybně dál.[8]

3.1.1 Struktura uzlu na sběrnici CAN

Každý uzel komunikující na sběrnici je jeden celek, který se skládá ze třech základních prvků – mikroprocesoru, řadiče, který realizuje linkovou vrstvu a budiče, který realizuje fyzickou vrstvu komunikace. Je také nutno podotknout, že sběrnice CAN má na konci svých signálových vodičů (*CAN_L* a *CAN_H*) zakončovací rezistory s hodnotou 120Ω . Tyto zakončovací rezistory slouží k zamezení odrazů na vedení, které je potom impedančně přizpůsobeno.

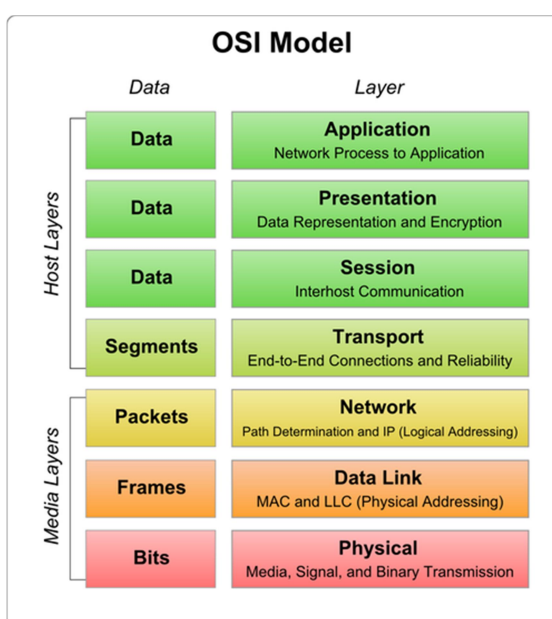


Obrázek 1 - Struktura uzlu na sběrnici

3.1.2 Síťový model CAN

Pojmem sériový komunikační protokol je myšlen způsob komunikace (přenos dat) mezi dvěma uzly na sběrnici. Protokoly jsou rozděleny do vrstev, které jsou definovány OSI modelem na obrázku. Celkově je těchto vrstev sedm a jedná se o vrstvy: aplikační, prezentační, relační, transportní, síťovou, linkovou a fyzickou. Každá z vrstev má danou úlohu, přijímá úkoly od nadřazenější vrstvy a dává úkoly vrstvě podřazené.

Standard ISO 11898 definuje pouze **fyzickou** a **linkovou** vrstvu modelu. Vzájemnou funkčnost obou vrstev zajišťuje řadič (linková vrstva) a budič (fyzická vrstva). Dnes je běžné, že řadič je již implementován v mikroprocesoru. [8]



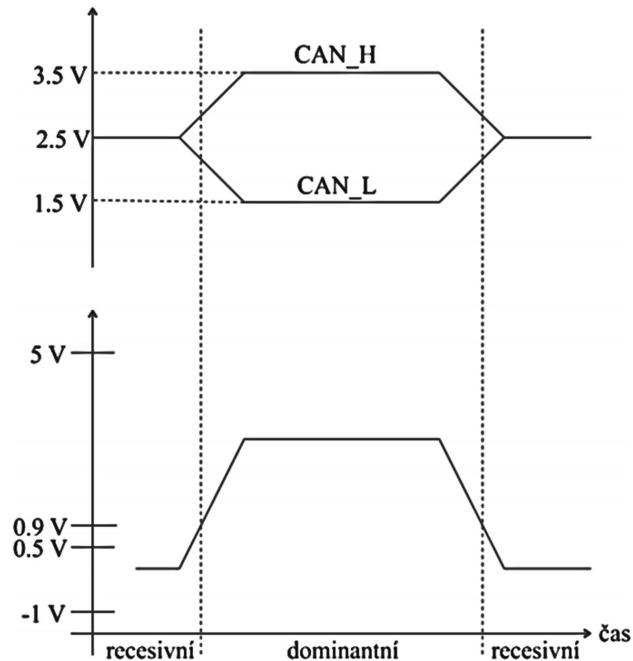
Obrázek 2 - OSI model [8]

3.1.3 Fyzická vrstva

Fyzická vrstva představuje použitý hardware, tedy fyzické spojení mezi uzly na sběrnici. Jejím úkolem je přenos bitů mezi uzly s respektováním všech elektrických vlastností. Tato vrstva musí být v rámci jedné sítě stejná pro všechny uzly. Definuje napěťové úrovně signálů (dle budiče), reprezentuje jednotlivé bity (dominantní, recesivní), definuje typ přenosového média (metalické vedení, optika, rádiový přenos).

Jako přenosové médium na sběrnici slouží stíněná či nestíněná kroucená dvoulinka pracující na napěťové diferenci mezi těmito dvěma vodiči, označované jako CAN_H a CAN_L. Při přenosu vznikají dva stavy – dominantní (aktivní stav

reprezentován log.“0“) a recesivní (neaktivní stav pomocí log.“1“). Stav na sběrnici je poté dán napětovým rozdílem na vodičích *CAN_H* a *CAN_L*. Pokud je rozdílové napětí mezi vodiči 0V, je na sběrnici recesivní stav. Pro dominantní stav je odpovídající rozdílové napětí okolo 2V – Obrázek 3.[7]



Obrázek 3 - Rozdíl diferenčního napětí [8]

Co se týče hardwaru, který je nutný ke komunikaci po sběrnici CAN, je třeba využít řadič CAN a budič CAN. V dnešní době již mají vybrané AVR mikroprocesory CAN řadič implementován, což významně usnadňuje další práci. Budiče pro sběrnici CAN se využívají externí. Budič vytváří spojení mezi fyzickou vrstvou sběrnice a linkovou vrstvou – řadičem CAN. Od něj dostává data, které se mají převést na vhodnou úroveň pro vysílání po sběrnici. Zároveň data na sběrnici přijímá a předává je ve správné formě řadiči CAN.

3.1.4 Linková vrstva

Linková vrstva klade důraz na to, aby správně rozpoznala počátek a konec vyslané zprávy a uspořádala je do celků tzv. rámců a kontrolovala neporušenost (integritu) přijatého rámce. Jejím úkolem tedy je vložení a vyjmutí dat do i z komunikačního rámce. Dále se stará o filtrování zpráv, rozhoduje o tom, které zprávy přijaté na fyzické vrstvě mají být dále zpracovány. Detekuje a signalizuje chyby na sběrnici. Řídí přístup uzlů na přenosové médium, takže rozhoduje o tom, zdali je sběrnice připravena pro odesílání, či příjem dat.

Zprávy vysílané po sběrnici protokolem CAN jsou tedy ve formě datových rámců, které se skládají z určitého počtu log. „0“ a log. „1“. Samotné zprávy přenášené po sběrnici neobsahují žádnou informaci o cílovém uzlu (zařízení), kterému jsou určeny, a jsou přijímány všemi ostatními uzly připojenými ke sběrnici. Každá zpráva obsahuje identifikátor, který určuje význam zprávy a její prioritu. Protokol CAN zajišťuje, aby zpráva s vyšší prioritou byla v případě kolize dvou zpráv doručena přednostně. Každý uzel připojený na sběrnici přijímá zprávy a na základě jejího identifikátoru rozhodne, zda na zprávu bude reagovat či nikoliv.[8]

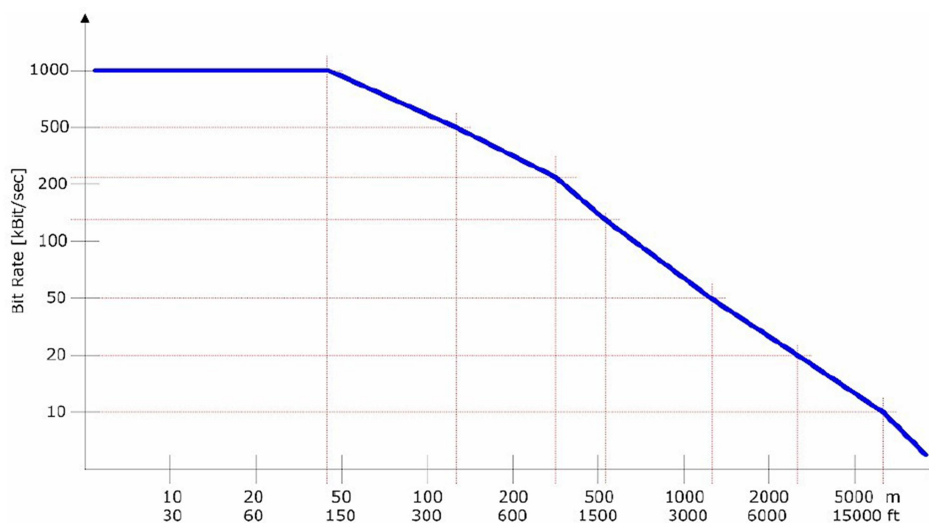
3.1.5 Aplikační vrstva

Poskytuje rozhraní pro aplikaci, která komunikuje po síti (předává data aplikací a přebírá je od aplikace). Řada výrobců čipů a prodejců zařízení komunikujících po sběrnici CAN poskytují knihovny funkcí (API – application interface) pro svá zařízení. Tyto funkce zajišťují propojení mezi aplikační a linkovou vrstvou (jsou to drivery, pomocí kterých se přistupuje na HW). [8]

Tato vrstva je dána tabulkami 1 a 2 v kapitole 2.1.

3.1.6 Rychlosti na sběrnici

Standardní přenosové rychlosti na sběrnici jsou 10, 20, 50, 100, 125, 250, 500 a 1000 v jednotkách kbit/s. Maximální rychlost 1Mbit/s je garantována při délce vedení do 40m a s rostoucí délkou vedení rychlost klesá přibližně dle následujícího obrázku. [8]



Obrázek 4 - Závislost rychlosti na délce vedení [8]

3.1.7 Přístup na sběrnici

Pro přístup na sběrnici existují dva základní způsoby – centralizovaný a decentralizovaný. U centralizovaného způsobu se jedná o tzv. master-slave architekturu, kdy je na sběrnici pouze jedna řídicí (master) jednotka, která řídí ostatní podřízené (slave) jednotky. Výhodou tohoto způsobu je jeho jednoduchost a nižší náklady, ale v případě poruchy hlavní řídicí (master) jednotky dochází k zastavení veškeré komunikace.

Decentralizovaný způsob přístupu tzv. master-master je charakteristický v tom, že každé zařízení na sběrnici je typu master, tedy je řídicí a schopno komunikovat na sběrnici nezávisle. Pokud dojde k chybě, či poškození na kterémkoliv uzlu, komunikace pokračuje dál bez vadného uzlu.

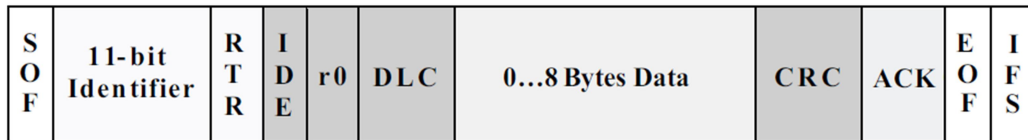
Typickým představitelem master-master architektury je právě sběrnice CAN, která využívá metody náhodného přístupu na sběrnici. U této metody nejprve jednotka, která chce vysílat, zkontroluje, zda je sběrnice volná. Pokud je sběrnice volná, vyšle jednotka zprávu. Kvůli náhodnému přístupu na sběrnici se stává, že začnou vysílat dvě jednotky najednou, pak musí rozhodnout priorita zprávy – zpráva s vyšší prioritou je odvyšována jako první a poté následuje zpráva s nižší prioritou, jde o tzv. CSMA (Carrier Sense Multiple Access). Priorita je daná tzv. identifikátorem, který obsahuje každá zpráva. Čím je identifikátor nižší, tím je priorita zprávy vyšší, což může vést k určité nevýhodě ohledně doručování zpráv s vyššími čísly identifikátorů, pokud je například kapacita sběrnice obsazena velkým počtem zpráv s nízkými identifikátory. [8]

3.1.8 Datová zpráva

Protokol CAN využívá dva typy datových zpráv – CAN2.0A a CAN2.0B. Jediný rozdíl mezi těmito dvěma typy zpráv je v délce jejich identifikátoru.

3.1.8.1 CAN2.0A – Standard frame

Formát zprávy dle specifikace CAN2.0A, který se také nazývá Standard frame čili Standardní formát, který má identifikátor o délce 11 bitů. Po jednoduchém výpočtu pak vyplývá, že při použití 11 bit identifikátoru jsme schopni rozlišit 2048 zpráv.



Obrázek 5 - Struktura datové zprávy – Standardní formát (11 bit ID)

Struktura datové zprávy se skládá z následujících polí:

SOF (1bit) – značí počátek zprávy → stav *dominant*

Identifikátor (11bit) – identifikátor zprávy, který určuje i její prioritu

RTR (1bit) – RTR *dominant* → datová zpráva

RTR *recessive* → žádost o data

R0 (1bit) – rezervovaný bit vždy *dominant*

DLC (4bit) – délka dat, udává počet přenášených bajtů a slouží pro kontrolu, zdali příjemci došla celá zpráva.

Data (0-8B) – datové bajty zprávy, její délka může být až 8 bajtů tedy 64 bitů.

CRC (16bit) – kontrolní součet zprávy (15bit) + delimiter (1bit) – oddělovač (vždy *recessive*) vytvoření časové prodlevy k výpočtu CRC, kontrolní součet CRC je vždy počítán od počátku zprávy (bit SOF) až po poslední datový bajt zprávy.

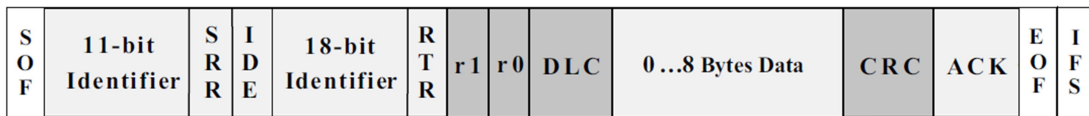
ACK (2bit) – potvrzení vysílači, že zpráva byla příjemcem přijata bez chyb. Každý uzel znovu vypočte CRC, pokud se shoduje s CRC přijatým od vysílače, je zpráva v pořádku.

EOF (7bit) – ukončovací pole, vždy po sobě jdoucích 7 bitů ve stavu *recessive* značí konec datového rámce

IFS (3bit) – vždy po sobě jdoucí 3 bity ve stavu *recessive* jako minimální vzdálenost mezi zprávami jakéhokoliv typu, během této doby nemůže žádný uzel začít vysílat datovou zprávu

3.1.8.2 CAN2.0B – Extended frame

Druhým formátem je dle specifikace CAN2.0B, známý také jako Extended frame čili rozšířený formát, který má identifikátor o délce 29 bitů. S tímto 29 bit identifikátorem jsme schopni rozpoznat téměř 537 miliónů zpráv.



Obrázek 6 - Struktura datové zprávy – Rozšířený formát (29 bit ID)

Struktura datové zprávy dle specifikace CAN2.0B, tedy Extended je v podstatě stejná jako Standard, rozdíl je pouze v délce identifikátoru a v polích, jenž s identifikátorem souvisí.

SRR (1bit) – jako náhrada za bit **RTR** a je vždy *recessive*

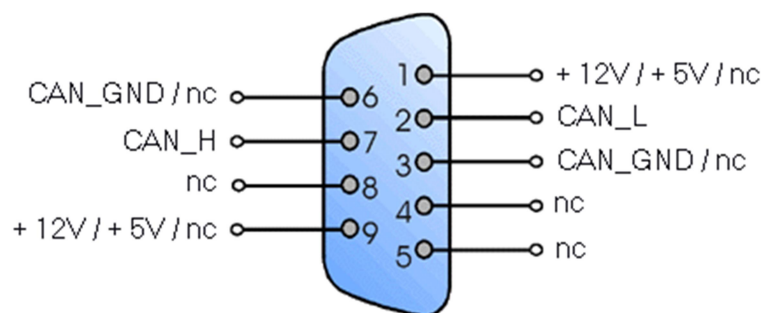
IDE (1bit) – tento bit v úrovni *recessive* značí, že jde o rozšířený (extended) identifikátor, takže do pole **18-bit Identifier** následují další bity rozšířeného identifikátoru

R1 (1bit) – rezerva, vždy *dominant*

3.1.9 Přenosové médium a konektory

Data jsou nejčastěji přenášena přes metalické vedení kroucenou dvoulinkou, která může být nestíněná, nebo pokud to vyžadují provozní podmínky, kde by mohlo docházet k elektromagnetickému rušení, použije se kroucená dvoulinka stíněná.

Pro připojení se používá 9 pinový konektor známý také pod označením CANON9. Používá se jak typ samec (CANON 9M), tak samice (CANON 9F). Význam jednotlivých pinů konektoru je v tabulce. [8]



Obrázek 7 - Pinout konektoru CANON9

Pin	Signál	Popis
1	CAN_V+	Volitelné napájení
2	CAN_L	Signál CAN_L
3	CAN_GND	Uzemnění
4	---	Rezerva
5	CAN_SHLD	Stínění (volitelné)
6	CAN_GND	Uzemnění (volitelné)
7	CAN_H	Signál CAN_H
8	---	Rezerva
9	CAN_V+	Volitelné napájení

Tabulka 3 – Význam pinů konektoru CANON9

3.1.10 Charakteristiky sběrnice CAN – shrnutí

- Multi-master – libovolný uzel může zahájit a řídit komunikaci
- Sdílená sběrnice s prioritním rozhodováním podle identifikátoru zprávy
- Náhodný přístup na sběrnici (CSMA)
- Přenosová rychlost 1Mbit/s do vzdálenosti až 40m
- Přenos zpráv typu multi-cast – zprávy jsou přijímány všemi uzly, ale reagují na zprávu pouze některé uzly (filtrace zpráv)
- Dobrá detekce a signalizace chyb
- Automatické odpojení uzlů generující mnoho chyb
- Komunikace probíhá v rámci s volitelným počtem Bytů (max. 8B/rámeček)
- HW implementace protokolu CAN – jednoduchý návrh SW
- Dvě varianty: CAN2.0A a CAN2.0B – liší se délkou identifikátoru

3.2 Sběrnice SPI

Rozhraní SPI, patří k velice známým, jednoduchým a dost často používaným typům sériových sběrnic. Využívá se k propojení dvou a více komunikujících uzlů na DPS. Oproti zmíněné sběrnici CAN je zde vždy pouze jeden uzel typu Master a ostatní připojené uzly jsou typu Slave, které jsou jím řízeny. Přenosové médium je složeno ze čtyř vodičů – MOSI, MISO, SCK a CS. Díky odděleným vodičům pro odesílání a příjem dat pracuje sběrnice v režimu full-duplex, tudíž se značně

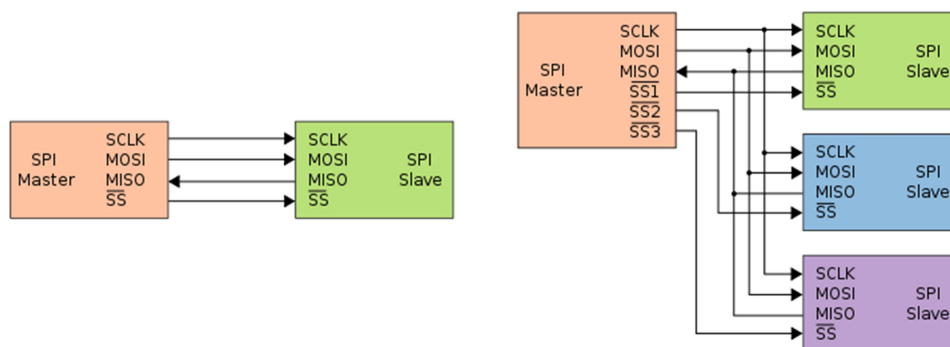
zjednodušuje komunikace, jelikož odpadá nutnost přepínání mezi odesíláním a příjmem dat.

Obousměrný přenos dat jsou jednotlivé uzly propojeny vodiči MOSI (Master Out, Slave In) a MISO (Master In, Slave Out) z nichž každý je určen pro jeden směr. Přenos dat je synchronní díky generátoru hodinového signálu SCK uzlu Master. Tento hodinový signál je připojen na každý uzel typu Slave. Vodičem CS (Chip Select), aktivujeme vybraný uzel Slave na sběrnici.

Komunikace probíhá tak, že uzel Master nastaví log.“0“ na pinu CS uzlu, se kterým chce komunikovat. Poté začne generovat hodinový signál na SCK a v té chvíli vyšlou obě zařízení svoje data, přičemž MOSI je vždy Master výstup, Slave vstup a MISO je Master Vstup, Slave výstup. Jakmile jsou data vyslána, může komunikace dále pokračovat a hodnota CS se nemění. Pokud má být komunikace ukončena, je nastavena hodnota CS uzlu Slave do log.“1“.

Níže jsou dvě možné struktury zapojení na sběrnici SPI. První případ ukazuje zapojení, kdy je na sběrnici jeden uzel typu Master a jeden uzel Slave. V druhém případě je uzlů Slave více. K zahájení komunikace s jednotlivými uzly Slave slouží samostatné vodiče CS (SS). Vodiče MOSI, MISO a SCK (SCLK) jsou vždy společné pro všechny uzly.

Pravá část obrázku 8 představuje v podstatě schéma v případě vytvořeného převodníku. Jako uzel „SPI Master“ je na převodníku mikroprocesor ATMEGA16M1 a jako „SPI Slave“ uzly jsou dva použité D/A převodníky (zelený a modrý blok schématu).



Obrázek 8 - Možnosti zapojení na sběrnici SPI [6]

3.3 Sběrnice USB

USB je univerzální sériová sběrnice (z *angl. Universal Serial Bus*), která přišla jako nástupce do té doby běžně používaných sběrnic (sériový/paralelní port, PS/2, aj.) pro připojení periférií k osobnímu počítači. V dnešní době je touto sběrnici vybaveno snad téměř každé zařízení připojitelné k počítači počínaje od myši, klávesnic, tiskáren, modemů přes digitální fotoaparáty, hudební přenosné přehrávače, videokamery, čtečky paměťových karet, flash disky, externí disky až po nejrůznější druhy zařízení vybavené právě touto sběrnici.

Typů USB je několik, historicky nejstarší USB 1.0 následovala verze 1.1, 2.0 a vývoj pokračoval až do dnešní doby k USB 3.1. Rozdíl byl vždy v maximální přenosové rychlosti sběrnice, avšak ten největší rozdíl nastal u verze USB 3.0, kdy se zvýšil i počet vodičů, ale zpětná kompatibilita byla zachována. Novými typy USB se tato práce zabývat nebude. Na zařízení je využito dostačující USB 2.0. Jeho zapojení je provedeno pomocí čtyř vodičů, kde jsou dva vodiče datové a jsou kroucené (DATA+ a DATA-) a další dva vodiče jsou napájecí nekroucené (+5V a GND).

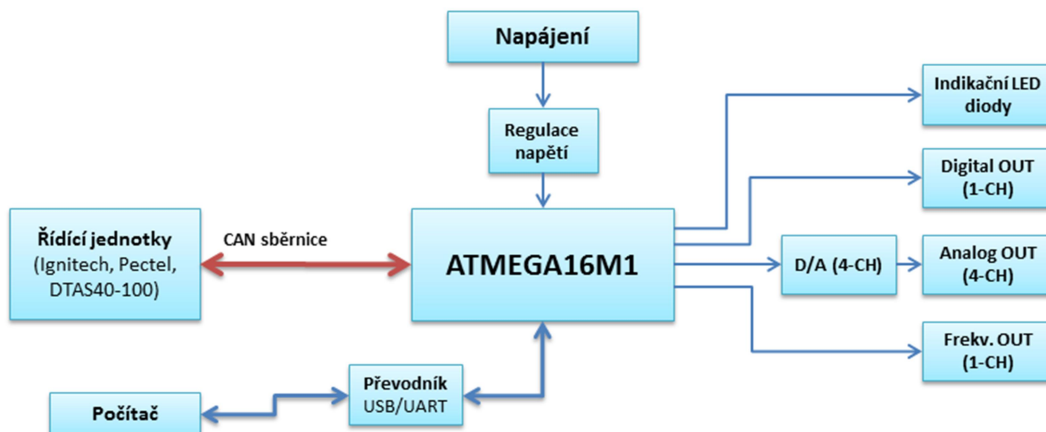
Ohromnou výhodou je možnost připojení zařízení za chodu počítače, bez nutnosti vypínání či restartování počítače, protože po připojení zařízení dojde pouze k restartu daného portu na sběrnici. V mnoha případech také není nutná ruční instalace ovladačů. Tyto vlastnosti nesou označení Plug & Play.

V případě vytvořeného převodníku je spojením uživatelské aplikace přes rozhraní USB realizována jeho parametrizace. Jsou přenášena data s aktuální konfigurací uloženou v EEPROM převodníku, pak data nového uživatelského nastavení, která jsou přenášena do převodníku. A také periodické zasílání skutečných hodnot vyčtených ze sběrnice CAN pro vizualizaci v uživatelské aplikaci.

4 VYBRANÝ HARDWARE

4.1 Blokové schéma CAN-data převodníku

Pro přehlednost je níže vyobrazeno blokové schéma celého převodníku. Jeho jednotlivé prvky – bloky budou v následujících kapitolách popsány podrobněji.

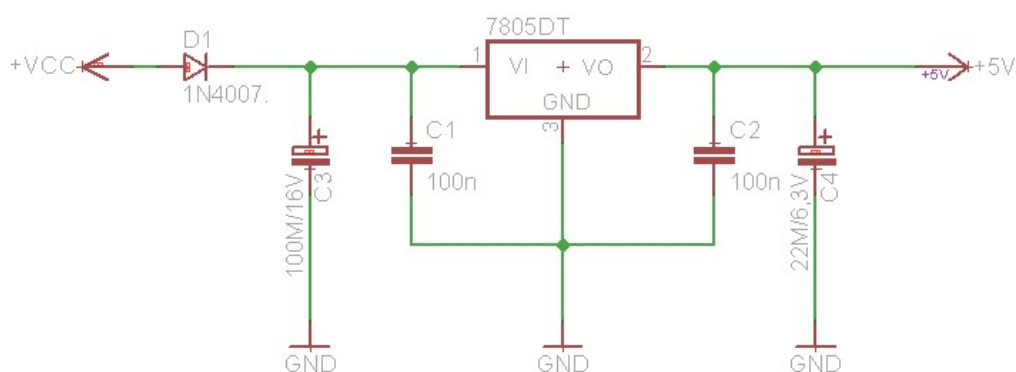


Obrázek 9 - Blokové schéma zařízení

4.2 Zdroj stabilizovaného ss napětí – blok regulace napětí

Celý obvod je napájen zdrojem stejnosměrného napětí +5V. Napájecí napětí zařízení je stabilizované integrovaným obvodem 7805, jenž na svém výstupu generuje žádané stabilizované napětí +5V. Konkrétní IO je 7805CG od firmy SEMICONDUCTOR, který dodává maximální výstupní proud 500 mA, uložen je v SMD pouzdru typu DPAK. Na vstupní části je připojen vyhlazovací kondenzátor s maximálním napětím 16V, dle datasheetu však může být na zvolený stabilizátor připojeno stejnosměrné napájecí napětí o hodnotě až 35V. Tedy pokud by byla vyžadována podpora 24V systémů, postačí tento vstupní kondenzátor nahradit kondenzátorem s vyšším maximálním napětím. Nutnou součástí tohoto obvodu jsou filtrační kondenzátory na vstupních i výstupních svorkách. Dále je na filtrační vstupní části umístěna křemíková usměrňovací dioda 1N4007 zapojená v propustném směru, která zde slouží jako ochrana zařízení proti přepólování napájecích svorek.

4.2.1 Zapojení stabilizátoru napětí 7805CG



Obrázek 10 - Obvodové zapojení stabilizátoru napětí 7805CG

4.3 Mikroprocesor ATMEGA16M1

Hlavní částí, dnes každého náročnějšího elektronického zařízení, je mikroprocesor. Mozkem zařízení je mikroprocesor AVR od firmy ATMEL. Tyto mikroprocesory se řadí mezi jedny z nejpoužívanějších na trhu. Jedná se o 8bit CMOS mikroprocesor s velice nízkým příkonem, založený na architektuře RISC. Jedna instrukce procesoru je vždy provedena v jednom taktu procesoru a díky tomu dosahuje procesor 1 MIPS (mega instrukce za sekundu) při 1 MHz taktu.[1]

Jádro AVR nabízí 32 pracovních registrů, které jsou přímo napojeny na aritmeticko-logickou jednotku (ALU), což dovoluje nezávislý přístup do dvou různých registrů během jednoho tiku hodin. Jinak se tomuto jevu také říká zřetězení instrukcí. Tato výhoda vytváří architekturu, která má až 10x větší výpočetní výkon, než architektura CISC.[1]

Hlavním důvodem volby tohoto mikroprocesoru je integrovaný řadič rozhraní CAN, tedy odpadá nutnost volby dalšího hardwaru, jenž se stará o komunikaci mezi mikroprocesorem a budičem CAN. Cenově je tento typ příznivý a také jeho nabídka v malých pouzdrech pro povrchové pájení, což je dobrý krok pro splnění co nejmenších celkových rozměrů výsledného převodníku.

Mikroprocesory ATMEGA s implementovaným řadičem CAN se vyrábí v pěti variantách a liší se podle paměti Flash, EEPROM a RAM. Konkrétní hodnoty kapacit pamětí jsou uvedeny v Tabulce 4.

MCU	Flash	EEPROM	RAM
ATMEGA32C1	32 kB	1 kB	2 kB
ATMEGA64C1	64 kB	2 kB	4 kB
ATMEGA16M1	16kB	512 B	1 kB
ATMEGA32M1	32 kB	1 kB	2 kB
ATMEGA64M1	64 kB	2 kB	4kB

Tabulka 4 - Rozdělení MCU dle paměti

Mikroprocesor ATMEGA16M1 je uložen v pouzdře buď typu TQFP32 nebo QFN32, pro převodník bylo zvoleno a použito pouzdro TQFP. Obě pouzdra mají 32 pinů a jsou určena pro povrchové pájení, tedy SMD.

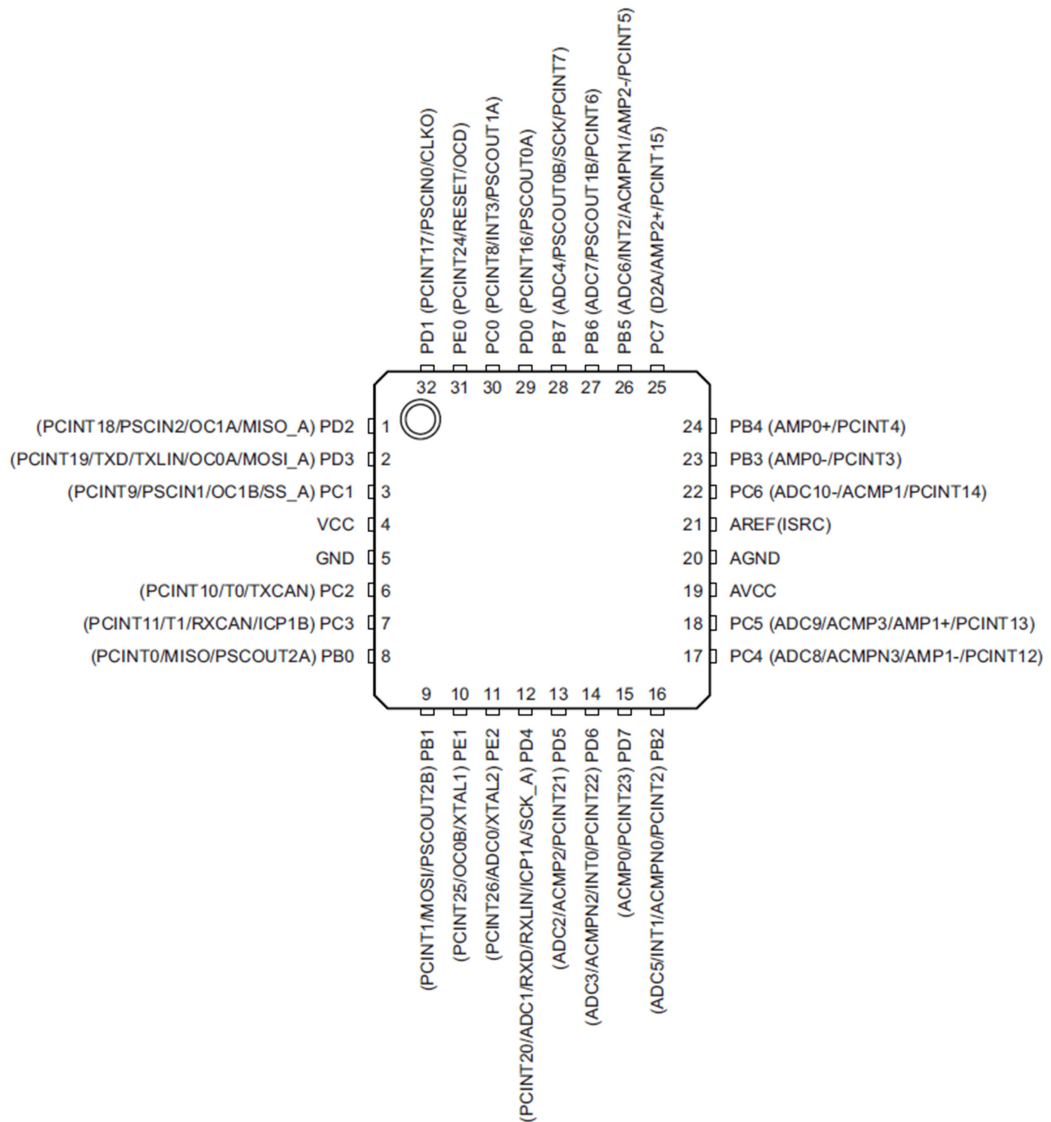
4.3.1 Parametry ATMEGA16M1

- Pracovní napětí: 2,7 ÷ 5,5 V
- Pracovní frekvence: 8MHz už při 2,7 V a 16 MHz už při 4,5 V
- 27 programovatelných vstupně/výstupních pinů
- 1 x 8 bit synchronní časovač/čítač (Timer/Counter0)
- 1 x 8 bit asynchronní časovač/čítač (Timer/Counter2)
- 2 x 16 bit synchronní časovač/čítač (Timer/Counter1, Timer/Counter3)
- 8 - kanálový 10 bit A/D převodník
- 1 x 10 bit D/A převodník
- Rozhraní: SPI, USART, CAN
- CAN2.0A a CAN2.0B

Díky kombinaci nízké spotřeby energie, poměrně vysokého výpočetního výkonu, nízké ceně a velké podpoře široké škály vývojových prostředí se tyto mikroprocesory jeví jako velice vhodné pro řešení domácích, ale i průmyslových aplikací.

4.3.2 Pouzdro mikroprocesoru ATMEGA16M1

Použitý mikroprocesor má celkem 32 pinů a je uložen v pouzdru TQFP, které je určeno pro povrchové pájení. Na obrázku je pinové rozložení mikroprocesoru. Význam a funkce všech pinů je k nalezení v datasheetu součástky od výrobce.[1]



Obrázek 11 - ATMEGA16M1 pinout (pouzdro TQFP) [1]

4.3.3 Použité vstupy a výstupy ATMEGA16M1

Mikroprocesor ATMEGA16M1 je napájen na pinu Vcc (pin 4) napětím +5V a je uzemněn na pinu GND (pin 5). Napájení (nevyužitého) integrovaného A/D převodníku i jeho napěťové reference je též připojeno na +5V a GND. Na všechny napájecí piny jsou připojeny filtrační kondenzátory.

Mikroprocesor ATMEGA16M1 disponuje celkem čtyřmi porty – PORTB, PORTC, PORTD a PORTE na které jsou připojeny jednotlivé periferie zařízení.

4.3.3.1 PORTB

Prvním využívaným portem je PORTB, na který jsou připojeny oba D/A převodníky – tedy signálový vodič SDI přenášející vstupní data pro převodník, hodinový signál CLK a dva vodiče Chip Select pro volbu konkrétního D/A převodníku.

4.3.3.2 PORTC

Tento port je hardwarově svázán s budičem MCP2551 pro rozhraní CAN. Dále PORTC obsluhuje výstup pro generování frekvence, jež reprezentuje otáčky motoru. O generování frekvence se stará 16-bit časovač *TimerCounter1*.

4.3.3.3 PORTD

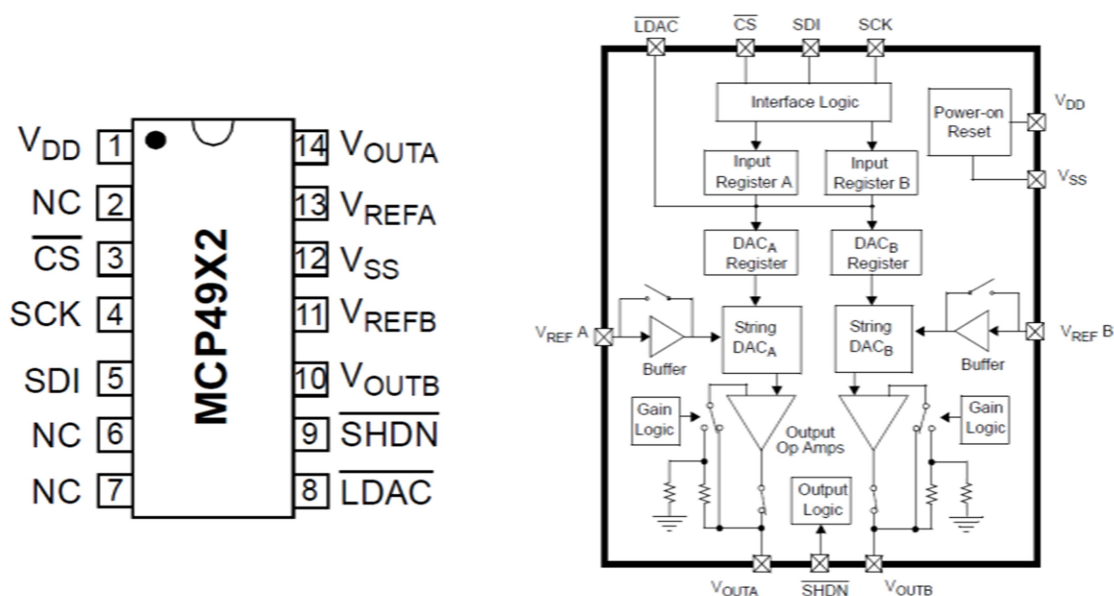
Další využitý port je PORTD sloužící pro programovací rozhraní ISP a dále sdílí vodiče od sériové linky, konkrétně vodiče vedoucí k převodníku FTDI sloužící jako převodník UART a rozhraní USB připojitelné do počítače uživatele.

4.3.3.4 PORTE

Posledním portem na mikroprocesoru je PORTE, má pouze tři piny. Dva piny jsou využity ke spojení s externím krystalem, který je naladěn na přesnou frekvenci 16MHz. Na třetím pinu je RESET.

4.4 D/A převodník MCP4922

Pro vygenerování analogových (spojitých) signálů na výstupu z přivedených digitálních (číslicových) signálů na vstupu, je nutné použít D/A převodník. Mikroprocesor ATMEGA16M1 sice má integrovaný D/A převodník, ale jeho rozlišovací schopnost, která činí 10 bitů, není pro konkrétní aplikaci dostačující. Z tohoto důvodu jsme využili převodník externí. Aby bylo dosaženo dobrého rozlišení, byl zvolen dvoukanálový 12 bitový D/A převodník MCP4922 od firmy MICROCHIP. Tyto převodníky mohou dle datasheetu výrobce pracovat až do teploty +125°C. Disponují vysokou přesností, nízkou hodnotou šumu pro běžné i průmyslové využití. Převodníky MCP4902/4912/4922 jsou dostupné v pouzdech PDIP, SOIC a TSSOP, načež pro naše zařízení bylo zvoleno pouzdro SOIC pro povrchové pájení. Použité pouzdro a vnitřní struktura D/A převodníku je na obrázku níže.



Obrázek 12 - Pouzdro a vnitřní struktura MCP4922 [2]

Dvoukanálový 12bit D/A převodník MCP4922 tedy provádí převod digitálního signálu, který je z mikroprocesoru přiveden po sběrnici SPI na pin SDI (Serial Data Input) a dále je přiveden generátor hodinového signálu na pin SCK (Serial Clock). Napájení převodníku je na pinech V_{DD} (+5V) a V_{SS} (GND). Výstup z jednotlivých kanálů (V_{OUTA} či V_{OUTB}) je volen pinem CS (Chip Select). Max. hodnota výstupního napětí je dána podle napěťové reference V_{REFA} či V_{REFB}, která může být navíc zesílena činitelem 1, nebo 2. To tedy znamená, že výstupní napětí může

nabývat maximální hodnoty $2 * V_{REFA}$ (či V_{REFB}) při použití dvojnásobného zesilovacího činitele.

Uživatelsky lze činnost obou kanálů D/A převodníku vypnout a to pinem SHDN (Shut down). V ten okamžik se dostává převodník do úsporného režimu a většina vnitřních obvodů je vypnuta právě za účelem úspory energie. Impedance zesilovače na výstupu je v tomto stavu nakonfigurována typicky na hodnotu 500kΩ. Tohoto úsporného režimu, kdy není potřeba využívat funkci D/A převodníku se využije například při chodu zařízení na baterii pro maximální úsporu energie. V případě vytvořeného zařízení je předpoklad, že zařízení bude vždy zapojeno na dostatečně silný zdroj energie, proto nebylo třeba brát v úvahu žádné úsporné režimy. Proto je pin SHDN trvale přiveden na napájení a je ve stavu log.1.

Pin LDAC slouží k synchronizaci obou výstupních kanálů, v této aplikaci je pin LDAC připojen na zem a není jeho funkce využita. [2]

4.4.1 Zápis dat na D/A převodník

Data se zapisují ve formě 16 bitového čísla, rozděleného po 2 Bajtech. Zápis dat na D/A převodník probíhá ve dvou fázích. Probíhá nejprve od nejvýznamnějšího bitu (MSB) k nejméně významnému bitu (LSB). První 4 MSB bity ($15 \div 12$) jsou bity konfigurační či ovládací a nastavují výstupní kanál, zdali bude použit buffer, zdali bude použito zesílení a zároveň se těmito ovládacími bity buď zapínají, nebo vypínají jednotky v převodníku, pokud není vyžadován D/A převod. Od 5. MSB bitu ($11 \div 0$) už jsou datové bity o 12 bitové délce, což poskytuje už solidní rozlišení pro dané použití. Registr pro zápis a rozložení bitů je lépe pochopitelný z obrázku.

REGISTER 5-1: WRITE COMMAND REGISTER FOR MCP4922 (12-BIT DAC)

W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
$\overline{A/B}$	BUF	\overline{GA}	\overline{SHDN}	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
bit 15								bit 0							

Obrázek 13 - Bitové rozložení zapisovaných dat [2]

4.4.2 Parametry D/A převodníku MCP4922

- Pracovní napájecí napětí: $2,7 \div 5,5V$
- Rozlišení: 12bit \rightarrow 4096 napěťových úrovní
- Doba ustálení: $4,5\mu s$
- Frekvence rozhraní SPI: 20MHz

- 2 kanálový převod – kanál A, kanál B
- Teplotní rozsah: $-40^{\circ}\text{C} \div +125^{\circ}\text{C}$

D/A převodník generuje na svém výstupu napětí v rozsahu $0 \div 5\text{V}$ v závislosti na přivedeném vstupním digitálním datovém signálu. Jeho referenční napětí V_{REF} je 5V . Výstupní analogová hodnota na výstupu, bude po přepočtení odpovídat aktuální vyčtené hodnotě ze senzorů lambda sondy, tlaku v sání a teplotě v sání, teplotě vody a poloze škrticí klapky.

Pro výpočet výstupní analogové hodnoty se vychází z následujícího vztahu, pro který platí, že výstupní napětí D/A převodníku je přímo úměrné součinu referenčního napětí V_{REF} , vstupního digitálního signálu D_N a hodnotě zesilovacího činitele G a nepřímo úměrný rozlišení převodníku 2^N , kde písmeno N značí bitové rozlišení D/A převodníku.

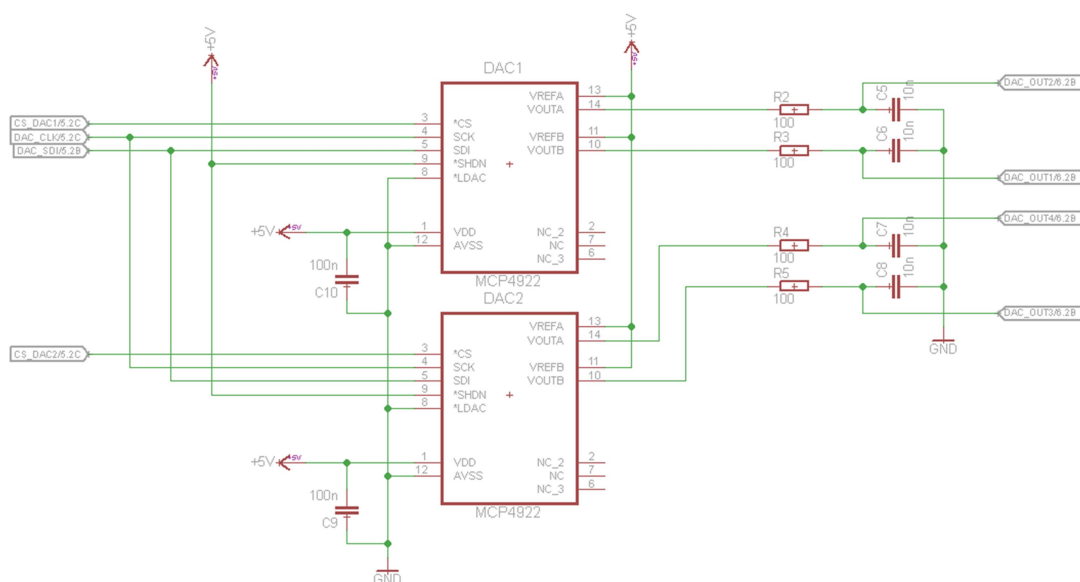
$$V_{\text{OUT}} = \frac{V_{\text{REF}} \cdot D_N}{2^N} \cdot G$$

V_{REF} napěťová reference

D_N vstupní digitální signál

G zesílení

4.4.3 Zapojení D/A převodníků v obvodu



Obrázek 14 - Obvodové zapojení D/A převodníků MCP4922

4.5 Frekvenční výstup (spínání výstupu)

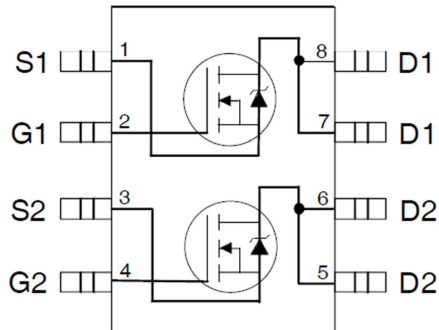
Po vygenerování analogových hodnot D/A převodníkem je třeba generovat další fyzikální veličinu – frekvenci. Frekvence slouží jako vhodný ekvivalent vyjádření otáček motoru.

V úvodu bylo zmíněno, že původní spodní hranice 0 [ot./min] byla změněna a to na hodnotu 200 [ot./min]. Spodní hranice byla takto zvolena na základě teoretických výpočtů a také praktickým ověřením. Vysvětleno v následujícím odstavci.

Frekvence mikroprocesoru je 16 MHz, nastavená předdělička má hodnotu 64, což zajišťuje frekvenci časovače 250 kHz. Při této frekvenci potom vychází doba trvání jednoho tiků časovače $4\mu s$. Vzhledem k tomu, že zvolený časovač je 16 bitový, je jeho maximální hodnota $2^{16} - 1 = 65\,535$ tiků. Po vynásobení celkového počtu tiků časovače a doby trvání jednoho tiků dostáváme hodnotu přibližně $T_{OVF} = 0,262$ s, což je doba, za kterou časovač dosáhne své maximální hodnoty a při dalším tiků dojde k jeho přetečení, jinými slovy je to perioda přetečení. Tato doba je zároveň maximální možná doba trvání jednoho logického stavu na výstupu a označíme ji jako půlperiodu trvání log. stavu. Jelikož na výstupu potřebujeme generovat změnu logického stavu s každým přetečením časovače je potom celková perioda výstupu složená ze dvou půlperiod logických stavů, tedy je dvojnásobná ($2 \cdot 0,262$), z čehož dostáváme frekvenci přibližně $\frac{1}{2 \cdot 0,262\,s} = 1,9$ Hz. To je minimální frekvence, kterou je schopen časovač generovat a odpovídá hodnotě 114 [ot./min.]. Zvolená hodnota 200 [ot./min.] vychází právě z tohoto výpočtu, jen je zaokrouhlena na celé stovky nahoru. Tato spodní hranice poté odpovídá frekvenci 3,33 Hz. Horní hranice frekvenčního výstupu je omezena hodnotou 333,33 Hz, což odpovídá hodnotě 20000 [ot./min.]. Tato hodnota byla zvolena jako rozumná mez, která je bohatě dostačující i pro otáčky motocyklových motorů.

Spínání frekvenčního výstupu zajišťuje jeden z dvojice tranzistorů typu MOSFET uložený v SMD pouzdru typu SO-8. Druhý tranzistor je nevyužitý, ale dal by se využít například jako watchdog, tedy jako výstup od watchdogu, například při hlídání některého stavu v programu, ke kterému pokud dojde, je na tomto výstupu vyvolán požadovaný stav. Oba tranzistory jsou v zapojení OC (Open Collector), takže výstup je spínán proti zemi.

Díky velmi malým rozměrům je IRF7103 vhodný všude tam, kde je požadavek na co možná největší úsporu místa na desce plošného spoje oproti pájení dvou MOSFET tranzistorů ve vlastních pouzdrech.



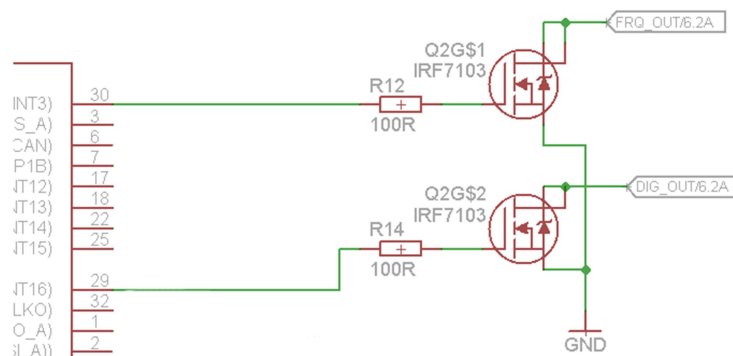
Obrázek 15 - Pouzdro a vnitřní struktura IRF7103 [3]

4.5.1 Parametry IRF7103[3]

- Max. napětí $V_{DS} = 50V$
- Max. proud $I_D = 3A$
- Odpor $R_{DS} = 0.13\Omega$
- Dvojice MOSFET tranzistorů s N-kanálem
- Velice rychlé spínání výstupu
- SMD provedení – malé rozměry

4.5.2 Zapojení frekvenčního výstupu v obvodu

Obvodové zapojení je jednoduché, spínací tranzistory jsou v zapojení typu otevřeného kolektoru (OC) jež spínají připojený výstup proti zemi. Frekvence spínání tranzistoru vychází přímo z otáček motoru vozidla přijatých od řídicí jednotky.



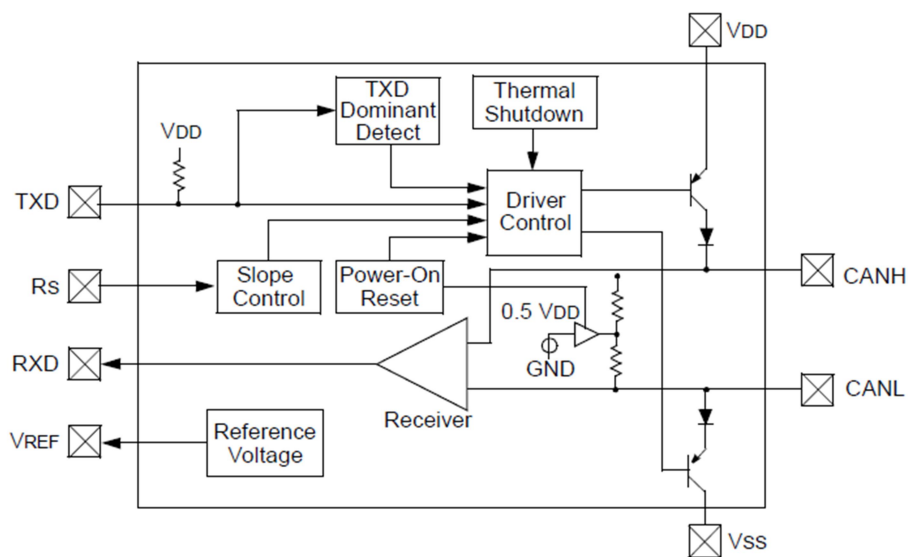
Obrázek 16 - Obvodové zapojení tranzistorů IRF7103

4.6 Budič CAN MCP2551

Obecně musí mít každý uzel komunikující po sběrnici CAN takové zařízení, které vytvoří rozhraní mezi digitálními signály generované CAN řadičem se signály, které se dají přenášet po přenosovém médiu, tedy po dvou vodičovém kabelovém diferenciálním vedení. Zařízení, která se o tento převod starají, jsou budiče. Integrovaný obvod MCP2551 je vysokorychlostní CAN budič, který slouží jako propojovací rozhraní mezi linkovou a fyzickou vrstvou v komunikaci na sběrnici CAN.

Jedná se o obvod vyrobený firmou MICROCHIP a je kompatibilní s normou ISO 11898. Jeho nejčastější využití je obecně tam, kde se nachází sběrnice CAN, nejčastěji v automobilovém průmyslu.[9] Použití sběrnice CAN však není jen automobilovou záležitostí, toto rozhraní je dnes využito i v kolejové dopravě, konkrétně v řídicích jednotkách modernějších lokomotiv.

Na obrázku je vyobrazena vnitřní bloková struktura budiče MCP2551.



Obrázek 17 - Vnitřní struktura MCP2551 [3]

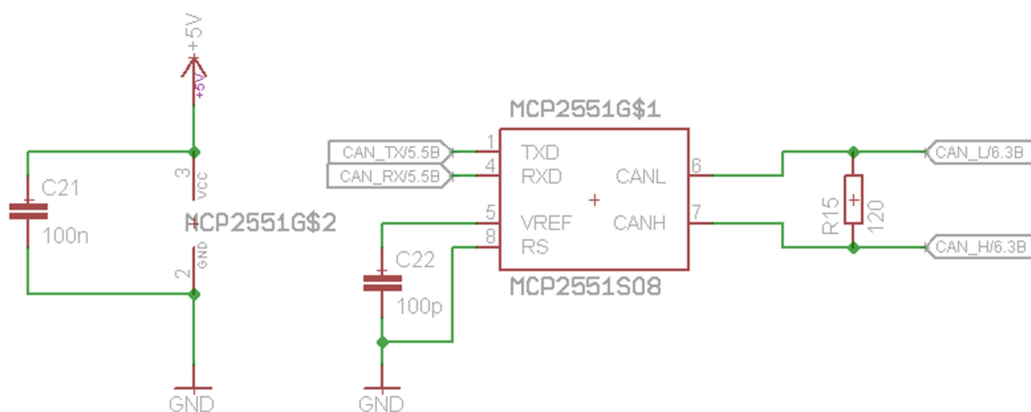
4.6.1 Parametry budiče MCP2551

- Rychlost až 1Mb/s
- Vhodný pro 12V i 24V systémy
- Detekce permanentně dominantního stavu na vstupu TXD
- Nízký odběr proudu ve StandBy režimu

- Ochrana proti přepólování napájení
- Automatická tepelná pojistka
- Podporuje připojení až 112 uzlů
- Vysoká odolnost vůči rušení

4.6.2 Obvodové zapojení budiče MCP2551

Napájení budiče je obstaráno skrze stabilizované napětí +5V a zem GND. Fyzická vrstva budiče je tvořena piny CANH a CANL. Impedanční přizpůsobení proti odrazům na vedení zajišťuje paralelně připojený rezistor s odporem 120Ω mezi svorky CANH a CANL. Piny TXD a RXD jsou připojeny na řadič CAN implementovaný v mikroprocesoru ATMEGA16M1. Piny RS a VREF jsou uzemněny.



Obrázek 18 - Obvodové zapojení budiče MCP2551

4.7 Převodník USB/UART – FT230X

Velice důležitou součástí pro komunikaci mezi uživatelským PC se SW a mikroprocesorem ATMEGA16M1 je právě převodník USB-UART. Tento převodník se stará o veškerou komunikaci mezi uživatelem a CAN-data převodníkem.

Pro tuto funkci byl zvolen převodník FT230X od firmy FTDI. Jedná se o převodník běžně používaný v zařízeních, jež mají rozhraní USB. Tedy USB-RS232/RS422/RS485 převodníky, mp3 přehrávače, digitální fotoaparáty a kamery, čtečky karet, čtečky čárových kódů, set-top-boxy s USB rozhraním a mnoho dalších.

Obrovskou výhodou těchto převodníků je jejich multi-platformová podpora napříč různými operačními systémy bez nutnosti instalace složitějších driverů. Podporu zahrnují systémy Microsoft Windows od řady XP až po nejnovější

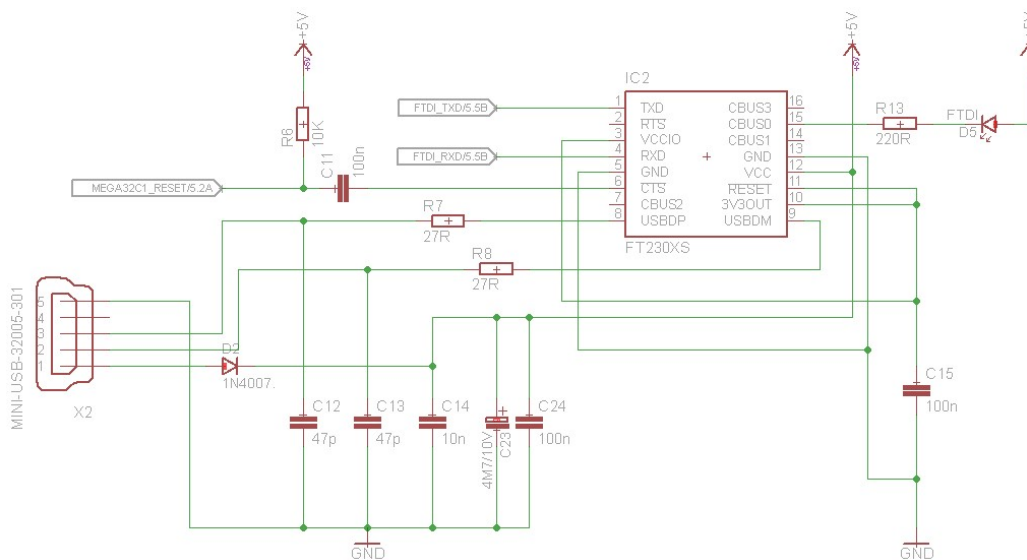
Windows 10. Dále převodník podporuje systémy Linux verze 3.2 a vyšší, podporu má také systém Mac OS-X a v neposlední řadě též systém spíše přenosných zařízení Android. V našem případě si vystačíme s podporou pro systém Windows 10.

Po připojení převodníku FT230X k počítači, se nahlásí operačnímu systému jako další připojený COM port. Výhodou tohoto převodníku je zjednodušení komunikace na straně mikroprocesoru. Ten nemusí mít implementován vlastní USB řadič, vystačí si pouze s vlastním interním UART rozhraním. Díky tomu, že rozhraní UART má snadnou obsluhu, je snadné skrze něj realizovat komunikaci mikroprocesoru a převodníku FT230X.

4.7.1 Parametry převodníku USB/UART – FT230X

- Nejrozšířenější USB/UART převodník
- Nízká cena, pouzdra SMD
- Plná kompatibilita s rozhraním USB2.0
- Napájení 3,3 ÷ 5,25V
- Maximální proudový odběr 8mA
- Konfigurovatelné I/O piny CBUS0 ÷ CBUS3
- FIFO buffer RX, TX – 512 Bajtů
- Podpora standardním i nestandardních rychlostí Baud Rate
- Baud Rate v rozsahu 183 Baud ÷ 3 Mbaud

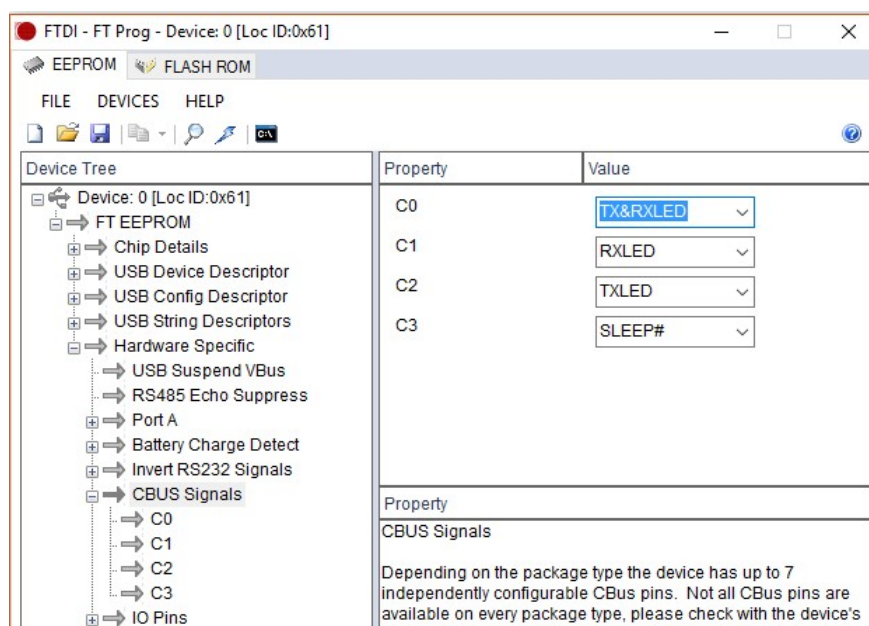
4.7.2 Zapojení USB/UART převodníku



Obrázek 19 - Obvodové zapojení FTDI převodníku FT230X

Indikační LED diodu značící provoz na sériové sběrnici bylo nutné nejprve nastavit. Vzhledem k úspoře místa byla zvolena pouze jedna LED, tedy bylo nutné, aby indikovala jak příjem, tak odesílání na sběrnici. K tomuto účelu mají převodníky FT230X takzvané CBUS I/O piny. Tyto CBUS piny jsou celkem 4 a mohou být nastaveny jako příznaky pro nejrůznější účely, čímž jsem se podrobněji nezabýval. Pro můj účel mi pouze stačilo nastavit si pin CBUS0 jako výstupní ovládací pin modré LED diody.

Pro tato nastavení je nutné použít aplikaci *FT Prog* poskytovaný od výrobce obvodu FT230X – obrázek níže. Po spuštění programu stačí jen načíst obvod FT230X, který je již připojen k USB. Po načtení obvodu již stačí rozkliknutím *FT EEPROM – Hardware Specific – CBUS Signals* nastavit na zvoleném CBUS pinu 0 volbu *TX&RXLED*. Toto nastavení tedy zajistí indikaci při příjmu i odesílání značené jednou LED diodou.



Obrázek 20 – Nastavení CBUS pinů v FT Prog











4.8 Indikační LED diody a jejich funkce

Pro sledování základních stavů zařízení, jsou v obvodu připojeny tři indikační LED diody – bipolární zeleno/červená, žlutá a modrá. Provoz na sběrnici CAN je indikován bipolární zeleno/červenou a žlutou diodou. Obecně nikdy nedojde ke

stavu, že by svítila červená i zelená LED zároveň, je to dáno jejím elektrickým zapojením. Stejně jako všechny ostatní LED má pouze dva vodiče, z nichž při jedné polaritě připojeného napětí svítí právě v zelené barvě a při opačné polaritě svítí červenou barvou.

Jestliže na sběrnici probíhá komunikace, začne žlutá LED dioda měnit střídavě stav – blikat, dle provozu na lince. Stav žluté LED diody se mění s každou bezchybně přijatou zprávou a rovněž tak s každou přijatou zprávou bez chyby je rozsvícena zelená LED. Pokud jsou data chybná, nebo neprobíhá žádná komunikace po sběrnici, značí to rozsvícení červené LED a programově je zajištěno, že rovněž žlutá LED nesvítí. Pokud není správně nastavená rychlost na sběrnici CAN, červená LED dioda rovněž svítí.

Obvodové zapojení LED diod je vyobrazeno na výřezu ze schématu na Obrázku 21. Modrá LED v této části schématu není, je zapojena samostatně ve schématu na obrázku 19. Popis funkcí LED diod uvedený výše je graficky pro větší přehlednost zobrazen v následující tabulce 5.

Po připojení k PC – před spojením na CAN		
Barva	Stav	Význam
	Rychlé zablikání	Uživatelské nastavení bylo úspěšně uloženo do EEPROM.
	Rychlé zablikání	Uživatelská nastavení byla úspěšně vyčtena z EEPROM.
	Bliká	Provoz na USB sběrnici – příjem nebo odeslání dat.
	Bliká	Není spojení na sběrnici CAN.
Při spojení na CAN sběrnici		
	Bliká	Neprobíhá komunikace po CAN sběrnici. Nebo jsou zprávy na sběrnici, ale není nastavena správná rychlost převodníku na CAN sběrnici.
	Vypnutá	
	Vypnutá	Zpráva CAN přijata bez chyby.
	Svítí	
	Bliká	Pokud je ID zprávy na seznamu CAN filtru a je přijata bez chyby, byla uložena do bufferu zpráv CAN.
	Bliká	Provoz na USB sběrnici – příjem nebo odeslání dat.

Tabulka 5 – Barevné označení LED a jejich význam

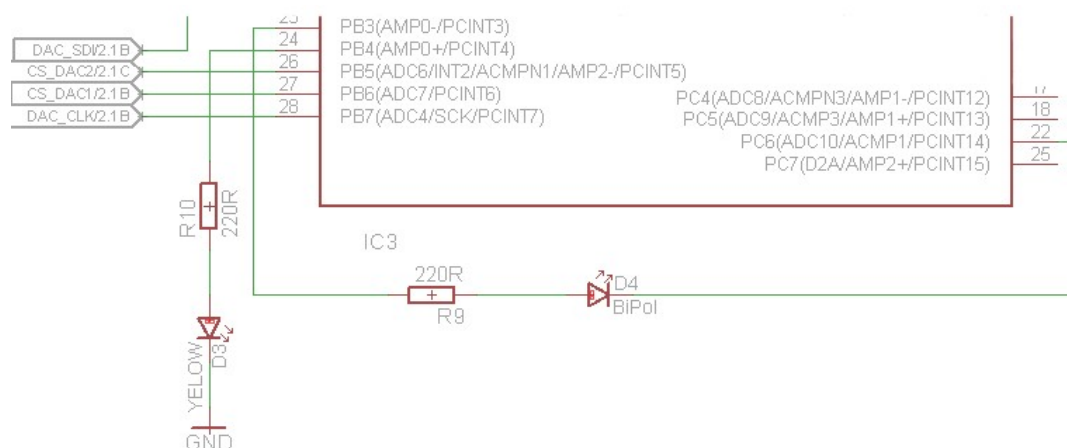
4.8.1 Obvodové zapojení indikačních LED diod

V obvodu jsou diody zapojeny každá jinak. Žlutá dioda je jediná připojena klasicky a to tak, že je rozsvěcena přivedením kladného signálu přes předřadný odpor 220Ω na její anodu z pinů mikroprocesoru a svou katodu má napevno připojenou k zemi.

Bipolární červeno/zelená dioda má oba své piny spojeny s piny mikroprocesoru, ten poté řídí polaritu na těchto pinech. Je-li na jeden z nich přiveden kladný signál v podobě stavu log. „1“ a na druhý stav log. „0“ je potom rozsvícena jedna ze dvou barev. Při přepólování je analogicky rozsvěcena barva druhá. Při zapojování není tedy třeba brát ohled na zapojení do DPS, nastavení barev je poté čistě programátorskou záležitostí. Rovněž má předřadný odpor o velikosti 220Ω .

Poslední je indikační LED dioda modré barvy, jež značí provoz na sériové lince. Tato dioda je svou katodou připojena přes předřadný odpor 220Ω na pin CBUS0 převodníku FT230X. Anoda diody je pevně připojena na větev +5V a její řízení je prováděno právě stavu log. „0“ přivedenými pinem CBUS0 na její katodu. Zapojení v obvodu je na Obrázku 21 – Obvodové zapojení FTDI převodníku FT230X.

Zapojení předchozích dvou LED jsou na obrázku níže.



Obrázek 21 – Obvodové zapojení indikačních LED diod

5 KONSTRUKCE ZAŘÍZENÍ

Tato část práce nastiňuje postup při návrhu a konstrukci vytvořeného převodníku. Zabývá se popisem tvorby plošného spoje zařízení, postup při jeho ožívování a jeho prvního spuštění. Větší část je věnována programové části zařízení, tedy jeho firmware a software v podobě aplikace pro Windows

5.1 Návrh desky plošného spoje

Pro návrh desky plošného spoje – dále jen DPS, byl využit program EAGLE. Program pro návrhy desek plošných spojů EAGLE ve verzi 7.2.0 je designově trochu odlišný od starších verzí a tak chvíli trvalo, než jsem se zorientoval. I přesto, že rozložení jednotlivých nástrojů je v této verzi stejné, jako v těch starších, tak grafický vzhled nástrojů je v odlišném stylu a z počátku může dělat problémy, než si návrhář tyto nově vypadající prvky osvojí.

Před návrhem DPS je v návrhovém prostředí EAGLE nejprve nutno vytvořit schéma, na základě kterého se přejde na návrh DPS.

V této verzi je možnost tzv. sheetů, česky by se dalo říct listů. Tuto funkci se vyplatí použít zejména tehdy, kdy se tvoří rozsáhlejší návrh zařízení, které se rozdělí do menších dílčích bloků a tyto jednotlivé bloky jsou potom vždy na vlastním sheetu. Tím se získá daleko větší přehlednost v návrhu a celková přehlednost v projektu. Vzájemné propojování vodičů mezi jednotlivými sheety se pak jednoduše provádí za pomoci tzv. labelů. Propojeny jsou mezi sebou pak vždy ty vodiče, které mají shodný popis na svém labelu.

V rámci návrhu převodníku bylo využito celkem 6 sheetů. Na prvním sheetu je celá napájecí část zařízení, na druhém je celý blok digitálně-analogové části, tedy D/A převodníky. Na třetím je komunikační část pro počítač, tedy převodník rozhraní USB-UART FT230X. Na dalším sheetu je umístěn blok další komunikační části – rozhraní CAN, konkrétně CAN budič neboli rozhraní mezi linkovou a fyzickou vrstvou CAN sběrnice. Na pátém sheetu je umístěn hlavní blok celého zařízení, tedy mikroprocesor ATMEGA16M1 společně s blokem frekvenčního výstupu a konektorem programovacího rozhraní ISP. Na posledním, tedy šestém sheetu je 16 pinový konektor, kterým je zařízení spojeno s okolím.

Po návrhu a kontrole schématu je teprve možné přejít k návrhu DPS. Postup při návrhu je popsán do jednotlivých bodů níže.

5.1.1 Rozměry DPS

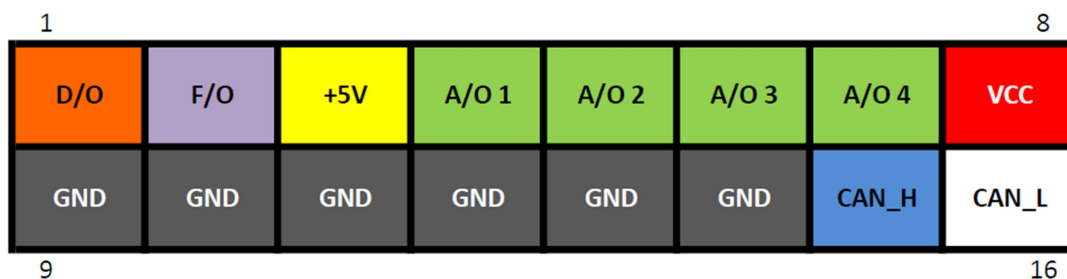
První co je třeba vždy rozhodnout, jsou celkové rozměry desky. Je to důležitý parametr, protože dodatečné zmenšování rozměrů DPS se později nemusí dělat úplně snadno, pokud jsou již natažené vodivé cesty mezi jednotlivými součástkami. Požadavek na rozměry DPS byl ideálně co nejmenší. Finální rozměr desky byl nakonec rozhodnut na 50 x 50 mm. Důvod takto zvolených rozměrů vychází z budoucí montáže zařízení do hliníkové profilové krabičky, která se vyrábí v omezené nabídce rozměrů. Pro zajištění co nejmenších rozměrů DPS je nutnost použití všech součástek typu SMD, výjimku tvoří krystal a indikační LED diody.

5.1.2 Konektor na DPS

Celá DPS je připojena na hlavní 16-pinový konektor s přesným označením N42GK-16, dle výrobce. Jsou na něm připojeny všechny vstupní i výstupní signály a napájení. Tato myšlenka vznikla právě z důvodu co největší úspory na celkových rozměrech převodníku. Jelikož pro komunikaci po sběrnici CAN, napájení zařízení, připojení výstupních signálových vodičů, by byly třeba různé konektory, které by na DPS zabíraly dost místa.

Jako náhrada za různé konektory byl zvolen jeden 16-pinový konektor molex, který je dostatečný na počet připojených vodičů a dále je i splněn požadavek na dostatečnou robustnost a rozměry. Z tohoto konektoru se až poté dle konkrétních požadavků uživatele vytvoří jednotlivé redukce pro připojení periférií jako je sběrnice CAN, analogové výstupy, frekvenční výstup atd. Redukce bude tedy odpojitelná a bude mimo zařízení.

5.1.2.1 Zapojení DPS na N42GK-16



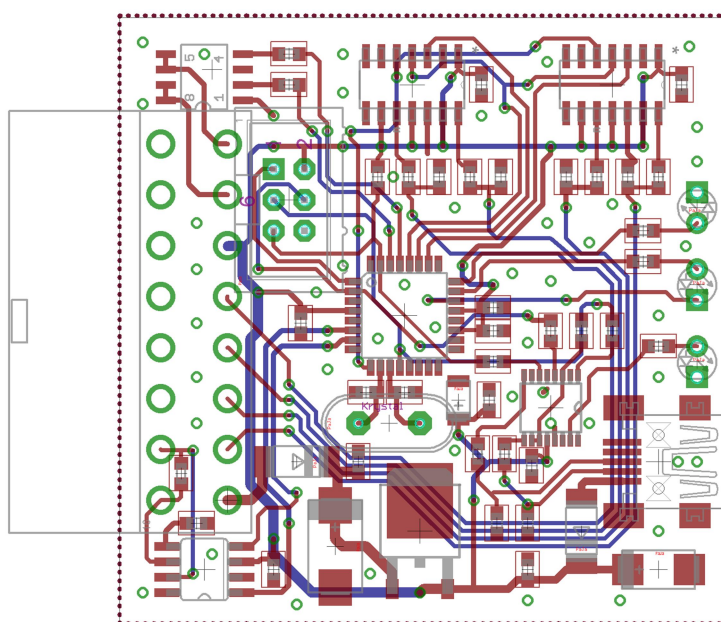
Obrázek 22 - Zapojení 16-pin konektoru N42GK

5.1.3 Tažení vodivých cest na DPS

Před samotným propojováním vodivými cestami je třeba zvolit vhodné rozmístění součástek na desce. Rozložení musí primárně splňovat funkčnost a také to, aby následné tažení vodivých cest bylo co možná nejefektivnější a nedocházelo například k tažení přes celou DPS. Jako sekundární požadavek je dobré, když je rozmístění součástek i estetické.

Po rozmístění součástek na desku přichází vzájemné propojení vodivými cestami. Výsledná DPS bude dvouvrstvá, což značně ulehčuje práci, pokud dochází ke křížení jednotlivých vodivých cest. Na takovéto křížení se využije prokovů, díky kterým se vodivá cesta dostane na druhou vrstvu plošného spoje a zde může pokračovat dál k dalšímu prokovu či součástce.

Pro vodivé cesty, které slouží jako napájecí, je vždy kladen důraz na to, aby jejich šířka, lépe řečeno průřez byl vzhledem k rozměrům desky a prostorovým možnostem co možná největší. U signálových vodičů již není požadavek na průřez tak přísný, jen by měl být zachován v stejné v rámci signálů, které spolu přímo souvisí. Dále je nutné dodržovat pravidlo, že při tažení jakýchkoliv sběrnových vodičů je nutné zachovat stejnou délku tažených cest pro každý z vodičů, tedy ideální je, aby byly stále sdruženě tažené od bodu A do bodu B. Na místech, kde je rozlita měď, jsou obě vrstvy vzájemně spojeny prokovy a přímo spojeny se zemí GND, pro snížení impedance. Výsledná DPS s rozmístěnými součástkami a nataženými cestami je v následujícím obrázku.



Obrázek 23 - Celková DPS

5.1.4 Osazení DPS

Po navržení DPS a její kontrole může deska přejít ke zhotovení do výroby. Na domácí testování se dají jednovrstvé (ale i dvouvrstvé) desky udělat svépomocí, nejčastěji fotocestou. Zjednodušeně se to dělá tak, že se na průhlednou fólii vytiskne vzor a přes tento vzor se poté plná deska osvítí UV světlem. Díky chemickým procesům se vzor otiskne na DPS a po vyleptání například v chloridu železitém zůstává tento vzor na desce v podobě nevyleptané mědi.

V tomto případě to je ovšem složitější, deska je dvouvrstvá a obsahuje prokovy, proto byl návrh DPS odeslán a vyhotoven firmou PRAGOBOARD sídlem v Praze.

Vzhledem k použitým SMD součástkám jsou třeba předchozí zkušenosti s osazováním součástkami pro povrchové pájení. Možnosti jak osadit jednotlivé součástky jsou dvě – ručně nebo strojově. V tomto případě je jasnou volbou ruční osazení, jelikož se jedná o prototyp a bylo by neefektivní a ekonomicky nevýhodné programovat osazovací stroj. Na pájecí plošky DPS je nejprve injekční stříkačkou nanesena pájecí pasta obsahující směs pájky a tavidla, poté se umístí všechny SMD součástky na svou pozici a samotné pájení se provede v pájecí peci. V tomto případě se pájení provedlo infračervené bezolovnaté pájecí peci značky LTC. Tyto pece mají malé rozměry, tedy i plošné spoje v nich pájené nebývají veliké a jsou určeny pro malosériovou výrobu.

V případě, více kusové výroby, je ideálním řešením nechat si vyrobit planžetu, tj. tenký nerezový plech, na kterém jsou laserem vyřezaná místa, na pozicích pájených ploch DPS. Planžeta se poté na tzv. natahovacím stroji položí přesně na desku a stěrkou se provede nanesení pasty. Výhodou je přesnost nanesení, množství pasty a především rychlost. Výrobu takové planžety též poskytuje firma PRAGOBOARD a dle počtu zářezů laserem vychází od 1000 do cca 3000,-Kč.

Pokud je použita kvalitní pájecí pasta, není třeba se bát, že by pájení skončilo neúspěšně. Díky povrchovému napětí si roztavená pájka většinou nedokonale položené součástky srovná sama, ale vždy je třeba osazenou desku zkontrolovat a případné nedokonalosti odhalit a opravit například mikropájkou.

Častým problémem bývá například zvednutí součástek – SMD odporů, SMD kondenzátorů, SMD diod, SMD tranzistorů a všech dalších SMD součástek, které mají tři a méně nožiček – zde je právě největší riziko zvednutí. Tento jev je velice často způsobem nevhodně zvoleným pájecím profilem, kdy je teplotní nárůst buď

příliš rychlý, nebo naopak pomalý. Dále je třeba dbát, aby v prostoru uvnitř pájecí pece byla v každém bodu teplota co možná nejvíce stejná. Potom dochází k prohřátí a následnému tavení pájky dříve na jedné straně součástky než na druhé. V tento okamžik se povrchové napětí bohužel stává nežádaným jevem.

V mém případě naštěstí chyby zvednutím součástek nenastaly a i přes to, že by nastaly, deska není osazena tolika součástkami, aby se nedaly odhalit vizuální kontrolou. Pokud se takovéto chyby vyskytnou, je to nepříjemné, ale je nutno je řešit. Řešením bývá testování jiných pájecích profilů či volba pájecí pasty od jiného výrobce. Je to sice občas velká alchymie, ale nutnost při sériové více kusové výrobě. Následné opravy stovek či tisícovek kusů a zjišťování jejich chyb v pájení je velice neefektivní a výrobky poté mohou mít větší procento chybovosti.

5.1.5 Oživení zařízení

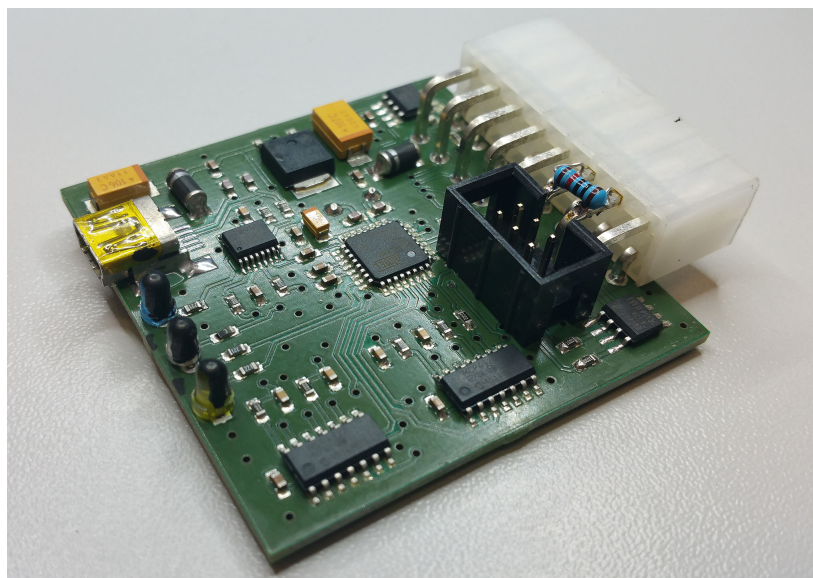
Po dokončení pájecího programu je vždy nutné každý takový výrobek otestovat, než ho bude možné připojit na napájení a případně nahrát firmware, pokud se jedná o výrobek, jenž je osazen mikroprocesorem, CPLD či FPGA čipem.

První fáze oživení spočívá v otestování, zdali nejsou někde na napájecích větvích zkratky, které mohou mít různou příčinu. Nejčastěji to bývá způsobeno špatným nanesením pájecí pasty, protože bylo použito zbytečně velké množství. Toto riziko je nejčastější, jsou-li DPS osazovány SMD součástkami ve velice malých pouzdrech či BGA pouzdrech, kdy rozteč pinů bývá i kolem 0,1 mm a při špatném nanesení je slítí pájky pravděpodobné.

Po proměření napájecí větve ohmmetrem a testu na dostatečně velký odpor na napájecích větvích a vizuální kontrole DPS zpravidla nic nebrání tomu, připojit desku na zdroj napájení. Následujícím krokem je změření napájecích větví na desce. Na jednotce převodníku je pouze jedna napájecí větev +5V jako výstup ze stabilizátoru, což velice zjednoduší a urychlí testování. Měření je vhodné udělat na všech součástkách, které jsou danou větví napájeny. Dají se tak odhalit případné studené spoje a předejde se tak problémům do budoucna, kdy například D/A převodník nebude na svém výstupu nic generovat a my budeme zbytečně hledat chybu v programu, kde chyba vůbec není.

Při takových testech je vždy vhodné používat laboratorní zdroje s proudovým omezením, velice snadno se tak dá odhalit případný další problém v podobě podezřele vysokého proudového odběru. Díky tomu, že je proud omezen zdrojem,

tak nedojde k nenávratnému poškození zařízení. Na obrázku 24 je vyfocen převodník po úspěšném osazení součástek a jejich pájení v infračervené pájecí peci.



Obrázek 24 – Osazená DPS

5.2 Krabíčka

Při volbě uložení zařízení bylo bráno v potaz, aby byly rozměry co nejbližší výrobku, tedy aby DPS zapadla svými rozměry co nejlépe. Po krátkém hledání na internetu jsem na zahraničním nákupním portálu Aliexpress našel hliníkovou krabíčku, černě eloxovanou, včetně černě eloxovaných víček. Cena za krabíčku zhruba \$4, a další \$4 za poštovné mi přišla celkem příznivá. Bohužel jsem zapomněl ji vyfotit v původním stavu před zásahem vrtáků a pilníku.



Obrázek 25 – profil hliníkové krabíčky



Obrázek 26 – ručně pilovaná krycí víčka pro krabičku

5.2.1 Umístění DPS do krabičky

Když krabička konečně dorazila, zkusil jsem do ní umístit DPS, ale bohužel jeho rozměry byly menší než drážky krabičky. Jako řešení mě napadlo, že jsem do drážek na obou stranách vlepil po celé délce drátky s izolací o průměru cca 1,5 mm, které společně s 50 mm rozměrem desky tvoří celkový rozměr 53 mm, což je i rozměr mezi oběma drážkami, poté už DPS do drážek perfektně zapadla. Délka krabičky je přesných 50 mm, což na délku DPS sedí akorát.

Dalším krokem byla úprava obou víček na krabičku. První bylo třeba připravit pro miniUSB konektor a indikační LED diody a druhé pro hlavní 16 pin molex konektor.

5.2.2 Výroba víček krabičky

Prvotní myšlenka byla, nechat si víčka vyfrézovat na fréze. Nakonec však byla vyrobena ručně v domácích podmínkách. Nejprve jsem vyvrtal otvory pro hlavní molex konektor, které jsem následně zapiloval pilníkem a poté stejným způsobem následovalo víčko pro miniUSB konektor. Nakonec jsem vyvrtal tři otvory pro indikační LED diody, které jsem vyplnil tavnou pistolí a vznikly tak celkem pěkné světlovody.



Obrázek 27 – Výsledná krabička, přední část



Obrázek 28 – Výsledná krabička, zadní část

Výsledek zpracování sice není vzhledově úplně dokonalý a podle představ, ale pro potřeby uložení a testování tohoto prototypu to je, myslím si, dostačující.

6 PROGRAMOVÉ VYBAVENÍ

Po osazení desky bylo na řadě její oživení a ověření funkčnosti. Ke každému funkčnímu zařízení je nutné použít dobře napsaný firmware. V první verzi zařízení, což byla náplň mého Ročníkového projektu 1, byl firmware napsán ve vývojovém prostředí CodeVision. K tomuto rozhodnutí vedlo především absolvování předmětu Aplikace mikroprocesorů, ve kterém se toto prostředí využívalo. V další verzi, byl firmware přepsán ve vývojovém prostředí Atmel Studio verze 7.0. S využitím některých výpočetních funkcí a některých dalších částí, se však musel firmware od základu přepsat. Prostředí CodeVision a Atmel Studio mezi sebou bohužel nejsou příliš kompatibilní a využívají jiné vlastní funkce a jiné zápisy, ale díky přenositelnosti jazyka C to nebyl zase takový problém.

Nejprve bylo nutné nastavit nový mikroprocesor v prostředí Atmel Studia, který byl programován přes ISP rozhraní pomocí zapůjčeného programátoru AVR Dragon. Po odzkoušení nejzákladnějšího programu – blikání LED diody a změření periody blikání na osciloskopu bylo zřejmé, že je nastaven správně včetně zdroje hodinového signálu.

6.1 Nastavení alternativního programátoru USBasp

Zapůjčený programátor AVR Dragon bylo nutno vrátit majiteli a tak přišlo na řadu rozhodnutí jakou zvolit alternativu. Vzhledem k poměrně vysoké ceně programátoru AVR Dragon, který je k dostání kolem 2000,- Kč, padla volba na alternativu, konkrétně programátor USBasp. Jelikož se jedná o neoficiální alternativní zařízení pro AVR mikroprocesory, vede k pohodlnému nahrávání firmwaru složitější cesta než s oficiálním programátorem, který jen stačí připojit k počítači a téměř jedním kliknutím je možno mikroprocesor ihned naprogramovat.

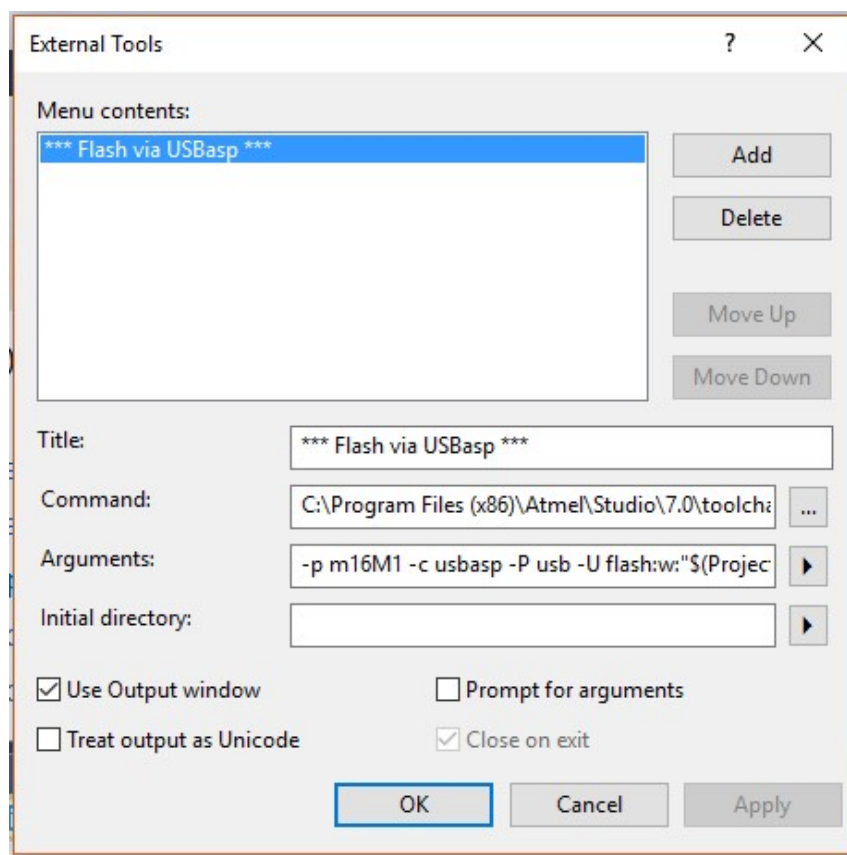
Po připojení k počítači je nutné nainstalovat ovladače programátoru USBasp. Poté je třeba použít externí program *avrdude.exe*, ke kterému je nutné mít konfigurační soubor **.conf* a v něm mít naimportovaný vybraný mikroprocesor. V případě ATMEGA16M1 musel být konfigurační soubor upraven právě pro tento procesor, vzhledem k jeho nepříliš častému použití ho konfigurační soubor neobsahoval. Na internetu jsou tyto soubory poměrně dobře vyhledatelné a tak byl problém vyřešen a konfigurační soubor upraven a ihned otestován. Otestování proběhlo jednoduše, v GUI verzi programu *avrdudeGUI.exe* stačí vybrat typ

mikroprocesoru a správný soubor *.HEX vygenerovaný ve složce s projektem. Pokud se programátor nenastaví sám, je nutno jej nastavit a vybrat na jakém je připojen portu. Pak už by po stisku tlačítka *Program* v oddělení FLASH paměti, mělo dojít k nahrání mikroprocesoru.

Pro vyšší komfort a možnost nahrávání přímo z prostředí Atmel Studia je nutné přímo v prostředí tento externí programátor nakonfigurovat v sekci *External Tools* a poté jej spouštět s parametrem, který obsahuje údaje o programovaném mikroprocesoru, typu programátoru a rozhraní přes které je programátor připojen viz. Obrázek 29 – Náhled nastavení externího programátoru. V tomto případě jsem použil parametr:

```
-p m16M1 -c usbasp -P usb -U flash:w:"$(ProjectDir)Debug\$(TargetName).hex":i
```

Při nastavení parametru je také nutné odkázat na složku, ve které se nachází program *avrdude.exe*. Konfigurační soubor *.conf musí být rovněž ve stejné složce. Díky tomuto nastavení a spouštění se zadaným parametrem lze pak provádět nahrání mikroprocesoru jednoduše jedním tlačítkem přímo z programovacího prostředí Atmel Studia jako s jiným oficiálním programátorem.



Obrázek 29 – Náhled nastavení externího programátoru

Po tomto nastavení se nahrávání provede stisknutím nově přidané položky na panel hlavní nabídky Atmel Studia. Jediné co je oproti nahrávání oficiálním programátorem jiné, je nutnost projekt nejprve přeložit klepnutím na *Build* → *Build solution* a až poté nahrát klepnutím na příslušné tlačítko. V mém případě tlačítko *****Flash via USBasp*****.



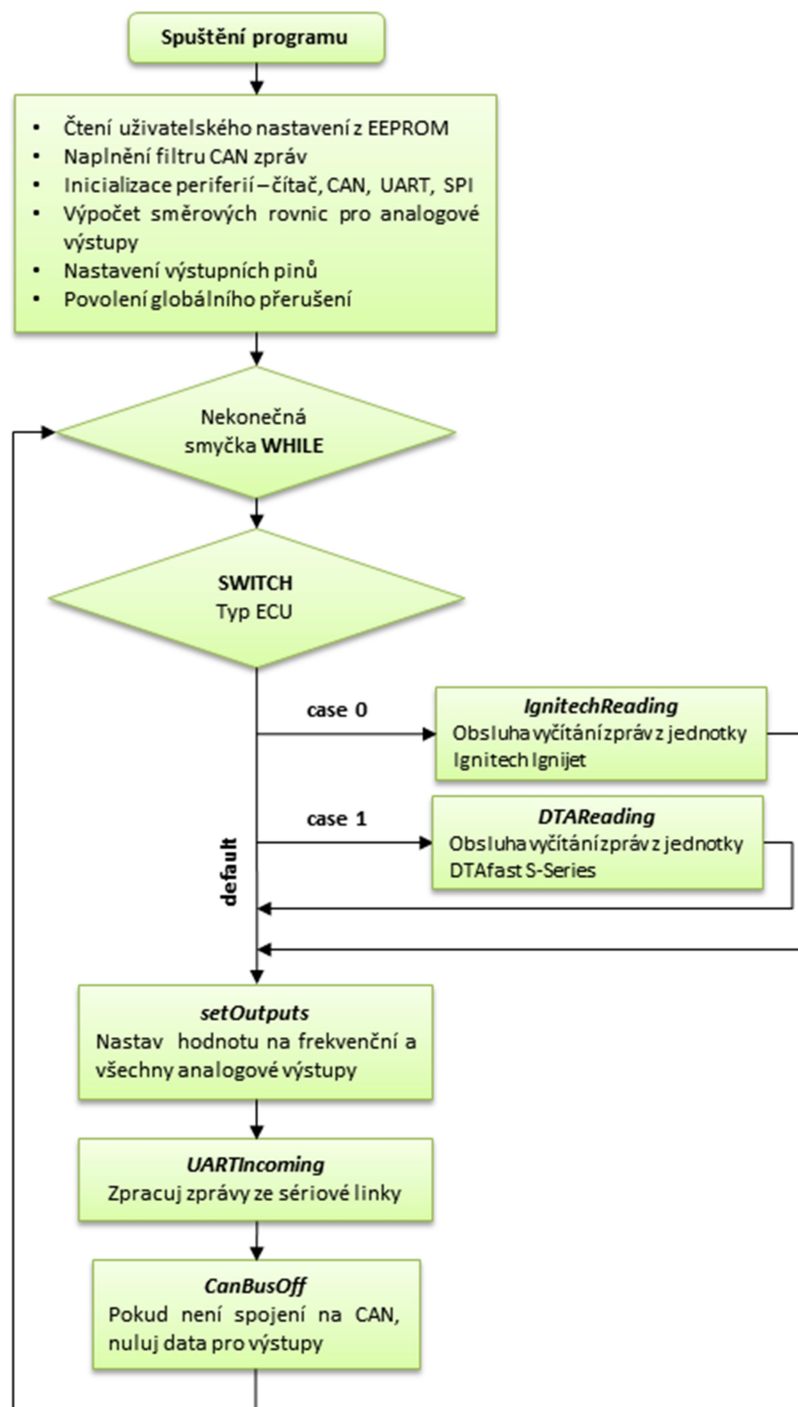
Obrázek 30 – Programovací tlačítko alternativního programátoru

6.2 Firmware

Filosofií celého firmwaru je to, aby byl kód psán co možná nejvíce přehledně pro případné úpravy a proto je rozdělen do jednotlivých C a hlavičkových H souborů s vlastním názvem daných periférií pro jasnou orientaci programátora. Díky tomuto rozdělení bylo i ožívování a prvotní testování jednotlivých bloků jednodušší.

Pro přehlednost v programu je na obrázku 31 přiložen vývojový diagram znázorňující průběh od spuštění zařízení až po skok do nekonečné smyčky *while*. Popis zmíněného diagramu je k dohledání v kapitole 6.2.1 a 6.2.2.

V Příloze C je pro kapitolu 6.2.3 rovněž přiložen vývojový diagram, ve kterém znázorněn průběh funkce *IgnitechReading*, která je provedena za předpokladu, že je zvolena uživatelem řídicí jednotka IGNITECH. Tato funkce se stará o vyčítání hodnot ze zásobníku CAN zpráv a předání vyčtených hodnot do struktury *UNIT_SELECT*. Funkce *DTAReading* pracuje naprosto stejně jako *IgnitechReading*, liší se pouze v testování jiných ID zpráv a prováděna je tehdy, je-li uživatelem zvolena řídicí jednotka DTA, proto bylo zbytečné k této funkci vytvářet totožný diagram.



Obrázek 31 – Vývojový diagram funkce Main()

6.2.1 Funkce *Main()*

O obsluhu hlavní části programu se stará *CAN_data_v2.c*. Ve funkci *main()* je nejprve inicializováno rozhraní SPI, dále časovač TimerCounter1. Poté probíhá

inicializace součástí pro rozhraní CAN – filtr zpráv CAN a samotné rozhraní CAN. Následuje inicializace rozhraní UART. Dalším krokem je vyčtení uživatelského nastavení z paměti EEPROM. Po vyčtení z paměti EEPROM a předání hodnot do struktury UNIT_SELECT jsou na základě hodnot ve struktuře vypočteny rovnice směrnice přímky pro každý kanál D/A převodníku. Poté je provedeno nastavení směrových registrů LED diod a frekvenčního výstupu, je povoleno globální přerušení. Jako indikace úspěšného spuštění převodníku jsou postupně rozsvíceny zelená, žlutá a červená LED, která zůstává blikat, dokud není spojení na sběrnici CAN.

6.2.2 Smyčka *While*

Po inicializaci následuje skok do hlavní smyčky programu, do smyčky *while*. Vše se vykonává maximální rychlostí bez časování či zpoždovacích prvků. Ve smyčce *while* dochází k periodickému dekódování zvolené řídicí jednotky. Testuje se hodnota ze struktury UNIT_SELECT, kde pak na základě výsledku program volá funkci dle zvolené řídicí jednotky – Ignitech či DTA. Po vykonání tohoto kroku program odskakuje do funkce *setOutputs*, jež obsluhuje frekvenční výstup a analogové výstupy vyčtením hodnot ze struktury UNIT_SELECT a následným předáním těchto hodnot na zvolené analogové výstupy resp. frekvenční výstup. Následuje skok do funkce *UARTIncoming*, která testuje příchozí zprávy na sériové lince z obslužné aplikace. Dle hlavičky příchozí zprávy se pak provádí další funkce, které jsou detailněji popsány v kapitole 6.2.6. Program se vrací zpět do smyčky *while*, kde následuje skok do funkce *CanBusOff*. Tato funkce testuje, provoz na sběrnici CAN. Pokud uběhne cca 2000 ms od poslední přijaté zprávy a další zpráva na sběrnici CAN není přijata, funkce rozsvítí červenou LED diodu, jako indikátor, že neprobíhá žádný provoz.

6.2.3 Funkce *IgnitechReading*

Tato funkce je prováděna vždy, když je stejnojmenná řídicí jednotka zvolena uživatelem. Nejprve se testuje, zda je v zásobníku CAN zpráv uložena hodnota ID, shodná se zvoleným parametrem, pokud ano, testuje se, o který konkrétní parametr na tomto identifikátoru se jedná. Zpravidla jednotka IGNITECH zasílá 4 parametry se stejným identifikátorem. Pokud je splněna podmínka požadovaného parametru, je

ze zásobníku zpráv CAN vyčten a předán do prvku `channel[n].result.data` ($n = 0, 1, 2, 3$) struktury `UNIT_SELECT` na n -tou pozici, která odpovídá příslušnému kanálu D/A převodníku.

Před vyskočením z funkce je z přijaté zprávy vždy vyčten nejdůležitější parametr – otáčky motoru, který je předán do prvku `engine_rpm` struktury `UNIT_SELECT`. Celá tato funkce je také popsána vývojovým diagramem na konci dokumentu v Příloze C.

6.2.4 Funkce `DTAReading`

Průběh této funkce je zcela totožný jako průběh funkce `IgnitechReading`. Jedinou odlišností je, že řídicí jednotka DTA zasílá zprávy s jinými ID, což je třeba při testování zajistit.

6.2.5 Funkce `setOutputs`

V hlavní smyčce `while` je periodicky volaná tato funkce, jejímž úkolem je nastavování výstupních hodnot na frekvenčním výstupu a na všech čtyřech analogových výstupech D/A převodníků dle hodnot uložených v prvcích `engine_rpm` a `channel[n].result.data` ($n = 0, 1, 2, 3$) ze struktury `UNIT_SELECT`.

6.2.6 Funkce `UARTIncoming`

Pokud je převodník spojen s počítačem po sériové lince, tedy přes rozhraní USB, přichází zprávy s určitou hlavičkou, bajtovou délkou a ukončovacím znakem. Úlohou funkce `UARTIncoming` je tyto zprávy zpracovat a provést další postup v programu. Pro tento účel komunikace byly vytvořeny dva komunikační kanály, které se starají o přenos dat mezi vytvořeným převodníkem a uživatelskou aplikací.

Momentálně jsou zpracovávány celkem tři typy příchozích zpráv z uživatelské aplikace. Zpráva „s – set“ obsahuje nastavení parametrů převodníku a je příkazem k uložení do EEPROM paměti převodníku. Zpráva „l – load“ obsahuje uložené nastavení z převodníku. Zpráva „r – real data“ zasílá reálná data z řídicí jednotky do uživatelské aplikace, kde je zobrazí obsluze. Levá strana tabulky říká, na jakém bajtu hodnota proměnné začíná a na levé straně je velikost v bajtech dané proměnné.

Set			Load			Real data		
0	's'	1	0	'l'	1	0	'r'	1
1	ecu_unit	1	1	ecu_unit	1	1	engine_rpm	2
2	can_speed	1	2	can_speed	1	3	AO_0	4
3	par_type[0]	1	3	par_type[0]	1	7	AO_1	4
4	par_min[0]	4	4	par_min[0]	4	11	AO_2	4
8	par_max[0]	4	8	par_max[0]	4	15	AO_3	4
12	par_type[1]	1	12	par_type[1]	1	19	'x'	1
13	par_min[1]	4	13	par_min[1]	4			
17	par_max[1]	4	17	par_max[1]	4			
21	par_type[2]	1	21	par_type[2]	1			
22	par_min[2]	4	22	par_min[2]	4			
26	par_max[2]	4	26	par_max[2]	4			
30	par_type[3]	1	30	par_type[3]	1			
31	par_min[3]	4	31	par_min[3]	4			
35	par_max[3]	4	35	par_max[3]	4			
39	'x'	1	39	'x'	1			

Tabulka 6 – Složení zpráv sériové linky

- **S** (set/store)

Za touto hlavičkou, se skrývá zpráva o celkové délce 40 bajtů, které obsahují nastavení zvolené uživatelem aplikace, skládá se:

- hlavička zprávy – „s“ [1B]
- typ zvolené řídicí jednotky [1B]
- rychlost sběrnice CAN [1B]
- parametr pro kanál 0 [1B], min. hodnota [4B], max. hodnota [4B]
- parametr pro kanál 1 [1B], min. hodnota [4B], max. hodnota [4B]
- parametr pro kanál 2 [1B], min. hodnota [4B], max. hodnota [4B]
- parametr pro kanál 3 [1B], min. hodnota [4B], max. hodnota [4B]
- ukončovací znak – „x“ [1B]

Pokud je přijata takováto zpráva, provede se složení všech 4 - bajtových hodnot pomocí bitových posunů a přetypování do proměnné správného typu. Protože může uživatel v aplikaci volit minimální a maximální hodnoty parametrů na přesnost setin, jsou vždy tyto hodnoty před uložením do struktury poděleny 100, protože tímto koeficientem jsou na straně aplikace před odesláním vynásobeny a odeslány jako celé číslo. Všechny hodnoty jsou pak přiřazovány do daných prvků struktury UNIT_SELECT ve stejném pořadí jako jsou vyčítány ze zásobníku, do kterého byla celá 40 bajtová zpráva uložena. Celá struktura s nastavením je následně uložena do paměti EEPROM mikroprocesoru.

Nakonec je nutné provést aktualizaci nastavení rychlosti sběrnice CAN a aktualizaci obsahu filtru zpráv CAN. Nastavení rychlosti na sběrnici CAN se provede funkcí *setCanSpeed* jejíž vstupní parametr je hodnota prvku *ecu.can_speed* ze struktury UNIT_SELECT.

Zařízení je tímto nastaveno a je připraveno na řídicí jednotku se zvolenou rychlostí CAN a pro příjem těch parametrů, které jsou zvoleny uživatelem na jednotlivých analogových kanálech.

- **L (load)**

Zpráva s touto hlavičkou je vlastně opak zprávy s hlavičkou „s“, nyní se zasílá uživatelské nastavení uložené v EEPROM paměti mikroprocesoru směrem k uživateli do aplikace, pro zjištění aktuální parametrizace.

Struktura zprávy je totožná, obě zprávy liší se pouze ve svých hlavičkách, ukončovací znak je rovněž znak „x“.

Příjde-li požadavek na zaslání aktuální parametrizace, jsou tyto hodnoty vyčteny ze struktury UNIT_SELECT, následně rozděleny po bajtech. Před rozdělením jsou opět 4 - bajtové hodnoty násobeny koeficientem 100. Pak je všech 40 bajtů jeden po druhém odeslány po sériové lince.

- **R (real)**

Po připojení převodníku k počítači a spojením na příslušný COM port je tato zpráva je uživatelskou aplikací zasílána periodicky každých 500ms. Aplikace si touto zprávou žádá o reálná data, která byla vyčtena z řídicí jednotky. Skutečné hodnoty

jsou poté vizualizovány v uživatelské aplikaci. Zpráva ve směru z aplikace má pouze znak „r“, není ověřována délka, ani ukončovací znak.

Je-li zpráva odesílána ve směru do aplikace, je její délka 20 Bajtů. Obsahuje hlavičkový soubor „r“, datové bajty skutečných hodnot a ukončovací znak „x“:

- hlavička zprávy – „r“ [1B]
- hodnota otáček motoru [2B]
- hodnota parametru na kanálu 0 [4B]
- hodnota parametru na kanálu 1 [4B]
- hodnota parametru na kanálu 2 [4B]
- hodnota parametru na kanálu 3 [4B]
- ukončovací znak – „x“ [1B]

6.2.7 Funkce CanBusOff

Z hlavní smyčky programu je periodicky spouštěna tato funkce. Její smysl spočívá v detekci odpojené sběrnice CAN. Obsahuje čítač, jehož hodnota *CanCnt* je inkrementována vždy o +1, pokud není splněna podmínka $CanCnt > CAN_CNT_MAX$. Pokud je podmínka splněna, začne blikat červená LED dioda jako příznak odpojené sběrnice CAN. Navíc jsou hodnoty frekvenčního výstupu i všech 4 analogových výstupů nastaveny na 0.

S každou přijatou bezchybně přijatou zprávou CAN je však hodnota čítače nulována a je tak zajištěno, že ke správné detekci dojde skutečně tehdy, je-li sběrnice odpojena.

6.2.8 Zásobník zpráv sběrnice CAN

Fronta je paměť typu FIFO – First In, First Out. Slouží zde jako vyrovnávací paměť k zamezení ztráty dat při příjmu více zpráv najednou na sběrnici CAN. Fronta je kompletně obsluhována z vlastního souboru *FIFO.c*. Obsahuje dvě funkce: *FIFOWrite* – zápis do fronty a *FIFORead* – čtení z fronty.

Tato fronta využívá ukazatel head značící pozici pro zápis do fronty a ukazatel tail s pozicí pro čtení z fronty. S každým zápisem se hodnota ukazatele head inkrementuje o 1 a rovněž s každým čtením se hodnota ukazatele tail inkrementuje o 1. Záleží na velikosti paměti, ale pokud se ukazatel head dostane na maximální hodnotu paměti a ukazatel tail pro čtení bude stále v počáteční výchozí pozici, tedy z paměti nebude vyčítáno, dojde k jejímu zaplnění. Při vyčtení a posunutí ukazatele

tail se předchozí pozice uvolní k zápisu a ukazatel head může po dosažení konce paměti opět na její začátek – díky této vlastnosti se tato paměť označuje jako kruhová – kruhový zásobník.

Funkce pro zápis *FIFOWrite* si nejprve vypočte další pozici k zápisu – head. Otestuje si, zdali se tato pozice neshoduje s pozicí se čtením, což by znamenalo plnou paměť. Je-li podmínka splněna, jsou data uložena a zrovna tak uložena nová pozice pro další zápis a testuje se, jestli není nastaven příznak plné paměti, pokud ano, je vynulován. V případě, že došlo k zaplnění paměti, je nastaven příznak zaplnění. Funkce pro čtení *FIFORead* nejprve testuje, zdali je co z fronty číst, pokud ne, je nastaven příznak prázdné fronty a funkce končí bez návratové hodnoty. Pokud jsou ve frontě data ke čtení, jsou vyčtena a následně je uložena nová pozice pro čtení – tail. Pokud byl nastavený příznak o prázdné paměti, vynuluje se.

6.2.9 Zachytávání zpráv ze sběrnice CAN

V první verzi nebyl výše zmíněný kruhový zásobník použit, což způsobovalo, že při příjmu více zpráv najednou docházelo ke ztrátě dat z důvodu dlouhého zpracování a výpočtů na mikroprocesoru. Tento problém musel být prioritně vyřešen. Řešením bylo využití přerušování na rozhraní CAN a ukládání přijatých zpráv do fronty – zásobníku, ze kterého jsou obsahy zpráv periodicky vyčtena a zpracována ve funkci *IgnitechReading* volanou v hlavní smyčce programu.

Příjem zpráv a ukládání je obsluhován ve vlastním souboru *CAN.c*. Při skoku do přerušování, které je vyvoláno příjmem zprávy na sběrnici, je nejprve registr CANPAGE nastaven tak, aby byl využíván pouze MOB0 a dále je nastavena auto inkrementace, díky které je vyčten celý obsah registru CANMSG. Projde-li zpráva ověřením, že byla přijata bez chyby, je dekodován její identifikátor, který je následně uložen do vlastní struktury CAN_MESSAGE a zrovna tak jsou do této struktury uložena vlastní data zprávy. Následuje ověření, zdali se ID zprávy nachází ve filtru zpráv viz. 6.2.10 – Filtrování zpráv na sběrnici CAN. Pokud ano, je tato struktura uložena do kruhového zásobníku FIFO_BUFFER pro pozdější vyčtení a zpracování.

6.2.10 Filtrování zpráv na sběrnici CAN

Proti zbytečnému přeplňování zásobníku fronty na sběrnici CAN, byl vytvořen softwarový filtr těchto zpráv. Jádro tohoto filtru se nachází v *CAN.c* a je spouštěna již při startu mikroprocesoru. Filtr se vždy naplní takovými ID, které odpovídají

požadovaným ID potažmo parametrům zvolené jednotky. To je při plnění filtru zpráv ošetřeno a k těmto duplicitám nedochází. Velikost filtru byla zvolena jako pole o 5 prvcích. Tato velikost je akorát dostačující, protože první prvek pole ve filtru je rezervován pro parametr otáček motoru, který je zpracován vždy. Zbývající čtyři prvky zaberou parametry určené ke zpracování na čtyřech analogových výstupech

6.2.11 Analogové výstupy

Generování hodnot na analogových výstupech má za úkol funkce *SendData*, která je volaná funkcí *setOutputs* z hlavní smyčky a je umístěná v souboru *AnalogOut.c*. Má dva vstupní parametry, prvním je napětí, které má D/A převodník generovat a kanál, na kterém D/A převodníku se má toto napětí generovat.

Díky možnosti parametrizace analogových výstupů, si uživatel volí v obslužné aplikaci ke každému výstupu jeho parametr a také jeho rozsah minimálních a maximálních hodnot, ve kterých se bude vzhledem k napěťovému rozsahu každého analogového výstupu generovat.

Toto nastavení, je zasláno a následně uloženo v EEPROM paměti vytvořeného převodníku. S každým spuštěním je nastavení z EEPROM paměti opět vyčteno a není tak třeba neustále připojovat počítač a nastavovat převodník znovu, tedy pokud uživatel nevyžaduje změny v nastavení.

6.2.12 Frekvenční výstupy

Jedná-li se o zprávu, nesoucí informaci o otáčkách je pro jejich reprezentaci využit frekvenční výstup. K tomuto účelu slouží *FreqOut.c*. Spínání frekvenčního výstupu je zajištěno 16 bit časovačem Timer/Counter1, který je nejprve inicializován ve funkci *initTimer*. Tato funkce nastaví parametry jako je zdroj frekvence hodinového signálu, předdělička a mód čítání a dále se také inicializuje přerušeni při přetečení časovače.

Ke změně stavu na frekvenčním výstupu dochází s každým přetečením časovače přepnutím pinu odpovídajícímu tomuto výstupu. Ve funkci *RPMTime*, jejíž vstupním parametrem je hodnota otáček, dochází k výpočtu předvolby pro časovač. Vstupní otáčky jsou nejprve testovány, zdali jsou v rozsahu 200 – 20000 [ot./min.], pokud ne, musí být do následujícího výpočtu limitovány buď spodní či horní hranicí rozsahu. Tím je zajištěno, že ve výpočtu předvolby nedojde ke zpracování otáček mimo rozsah. Perioda přetékaní je tedy dána z přepočtu vstupní hodnoty otáček na počet

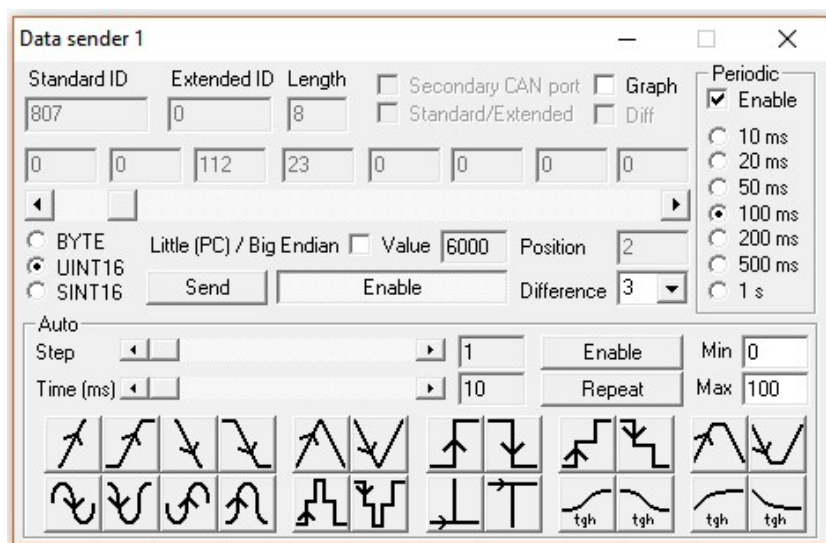
tíků – hodnoty odkud časovač začne čítat – předvolby. A tato předvolba je v přerušovací rutině časovače vždy aktualizována a zapisována do registrů *TCNTIH* a *TCNTIL*.

6.2.13 Sériová linka UART

Při startu mikroprocesoru dojde k inicializaci rozhraní UART ve funkci *UART_init*. V této funkci dochází nejprve k deklaraci vysílacího a přijímacího bufferu o velikosti 128 bajtů jak pro vysílací tak pro odesílací buffer. Poté se inicializují vstupní a výstupní piny RX a TX, nastaví se ovládací registr *LINCR*, dále časovací registr *LINBTR*, rychlost baudrate registrem *LINBRR* a nakonec se nastaví přerušovací registr *LINENIR*, aby bylo přerušování vyvoláno vždy s bezchybně příchozím (*LENRXOK*) i odchozím (*LENTXOK*) bajtem na lince.

6.2.1 Ověření funkce frekvenčního a analogových výstupů

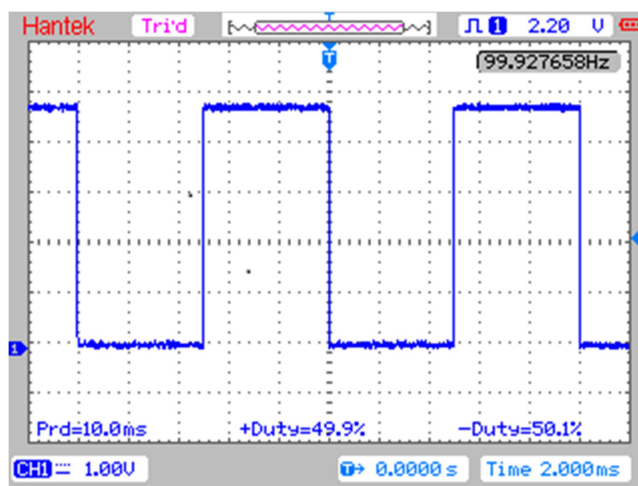
Po nahrání firmware do převodníku bylo třeba ověřit jeho výstupní část, zdali skutečně generuje žádané hodnoty na základě přijatých dat ze sběrnice CAN. Jako vysílač zpráv po CAN lince posloužil CAN analyzátor USB2CAN od firmy CANLAB. Dodávaný software PP2CAN umožňuje nasimulovat jakoukoliv zprávu se zvoleným identifikátorem a hodnotou, jež zpráva obsahuje. Na obrázku 32 je vidět zpráva s periodou 100ms, ID 807 a zaplněnými datovými bajty na pozici 2 a 3, což odpovídá parametru otáček. Hodnota zaslaných otáček je 6000, tato zpráva poslouží pro první test frekvenčního výstupu.



Obrázek 32 – Zpráva CAN s ID807 (otáčky motoru)

6.2.1.1 Test funkce frekvenčního výstupu

Při ověřování funkčnosti vytvořeného převodníku byl nejprve ověřen frekvenční výstup. Nejprve byla změřena frekvence při otáčkách 6000 [ot./min.] s očekávanou hodnotou 100Hz. Ze snímku osciloskopu – obrázek 33, je vidět, že s minimální odchylkou je předpoklad splněn, což předpovídá správnou funkci výstupu frekvence. Tento výstup byl dále měřen v rozsahu frekvence převodníku a výsledky měření byly zapsány do tabulky 7.



Obrázek 33 – Snímek změřené frekvence při 6000 ot./min

Otáčky byly voleny náhodně v tomto rozsahu. Nejprve se vyslaly nulové otáčky, aby se ověřilo, že je správně funguje spodní limit 200 ot./min. a dále v blízkosti této hodnoty, zda došlo ke zvýšení frekvence při zvýšení otáček nad tento spodní limit. Z výsledků měření je vidět, že až na malé zanedbatelné odchylky frekvenční výstup funguje správně. V tabulce 7 jsou vidět výsledky několika bodů měření.

Zaslaná hodnota [ot./min.]	Očekávaná hodnota [Hz]	Skutečná hodnota [Hz]
0	3,333	3,333
200	3,333	3,333
250	4,166	4,166
2000	33,333	33,327
5800	96,666	96,608
8000	133,333	133,274
14500	241,666	241,329
19800	330,000	329,842

Tabulka 7 – Výsledek měření frekvenčního výstupu (otáčky motoru)

6.2.1.2 Test funkce analogových výstupů

Po úspěšném ověření frekvenčního výstupu následoval test napětí na analogových výstupech. Test byl proveden na parametru teploty vody a hodnoty koeficientu lambda. Zasiílané hodnoty byly voleny náhodně.

Z naměřených dat je patrné, že až na drobné odchylky je funkce analogových výstupů správná. Odchylky mohou být způsobené nepřesnou referencí D/A převodníků, či volbou rozsahu měřicího přístroje.

Analogový výstup 1 – parametrovaný jako TEPLOTA VODY Rozsah nastaven na 0 – 150°C → Výstup: 33,33mV/1°C		
Zaslaná hodnota [°C]	Očekávaná výstupní hodnota [mV]	Skutečná výstupní Hodnota [mV]
0	0	1
10	333,33	336
75	2500	2525
100	3333,33	3364
145	4832	4872

Tabulka 8 – Výsledky měření na analogovém výstupu 1

Analogový výstup 2 – parametrovaný jako LAMBDA koef. Rozsah nastaven na 10 – 18 AFR → Výstup: 62,5mV/ 0,1AFR		
Zaslaná hodnota [AFR]	Očekávaná výstupní hodnota [mV]	Skutečná výstupní Hodnota [mV]
10	0	1,3
10,5	312,5	311,7
12,8	1750	1761
15	3125	3153
17,8	4875	4914

Tabulka 9 – Výsledky měření na analogovém výstupu 2

Závěrem testování lze říct, že jak test frekvenčního výstupu, tak test na analogových výstupech dopadl úspěšně. Veškeré odchylky očekávaných a naměřených dat jsou spíše zanedbatelné a není nutné se tímto dále zabývat.

V kapitole 6.3.1.3 také bude podobným způsobem ověřeno, zdali uživatelská aplikace zobrazuje správná data zaslaná z USB2CAN analyzátoru.

6.3 Software CAN-DATA Converter

Pro možnost plného využití zařízení je jeho ovládání řešeno skrze uživatelský PC s uživatelským SW a linkou USB na počítači a rozhraním USB/UART na převodníku. Uživatelský SW je psaný v jazyce C# v programovém prostředí Visual Studio Express 2013.

Hlavním přínosem tohoto SW je možnost parametrizace převodníku, tedy dát uživateli možnost zvolit si typ jednotky, ze které bude zařízení zpracovávat zprávy po sběrnici CAN. Dále si uživatel volí, které parametry, o nichž jednotka zasílá zprávy, se budou vyhodnocovat. Při volbě parametrů se zároveň zvolí, na kterém kanálu analogového výstupu bude jejich hodnota vygenerována. Parametr otáčky motoru je zpracováván vždy. Je to dáno tím, že je hardwarově svázán s frekvenčním výstupem převodníku, tedy nelze nijak parametrizovat. Parametr otáček je neustále zpracováván, protože jsou na něm ostatní parametry závislé při ladění motoru.

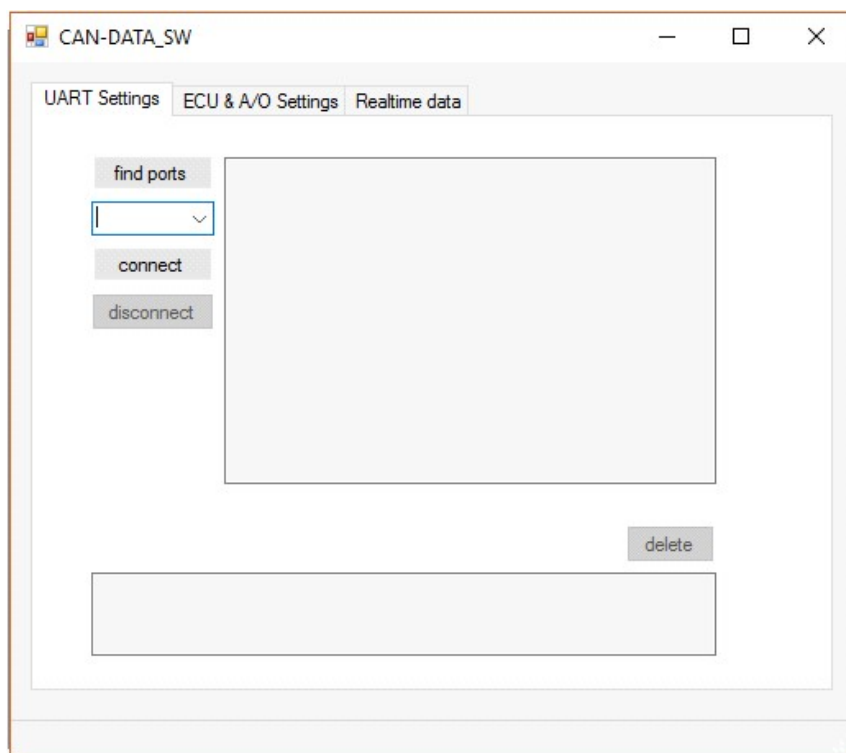
6.3.1 Funkce programu CAN-data Converter

V této části je vlastní popis funkcí uživatelského software. Ze zadání plyne, že by se software měl pomocí sériové linky spojit se zařízením a po této lince mezi sebou i komunikovat. Po připojení je již možno zařízení uživatelsky parametrizovat. Pokud existuje spojení s řídicí jednotkou na sběrnici CAN, je možná vizualizace přijímaných zpráv a následně hodnoty těchto zpráv zjednodušeně logovat do souboru TXT.

6.3.1.1 Nastavení a připojení na sériovou linku

Na obrázku níže je pro lepší představu náhled z aplikace. Připojení k sériové lince je prováděno na kartě *UART Settings*. Nejprve se tlačítkem „*Find Ports*“ vyhledají dostupné porty připojených zařízení, nalezená zařízení se přidají do rozbalovacího seznamu. Po výběru zařízení na připojeném na daném COM portu z rozbalovacího seznamu dojde po klepnutí na tlačítko „*Connect*“ k vytvoření spojení. Spojení je navázáno s následujícími parametry: **rychlost** – 19 200 Baudů, **parita** – NE, **datové bity** – 8, **Stop bity** – 1.

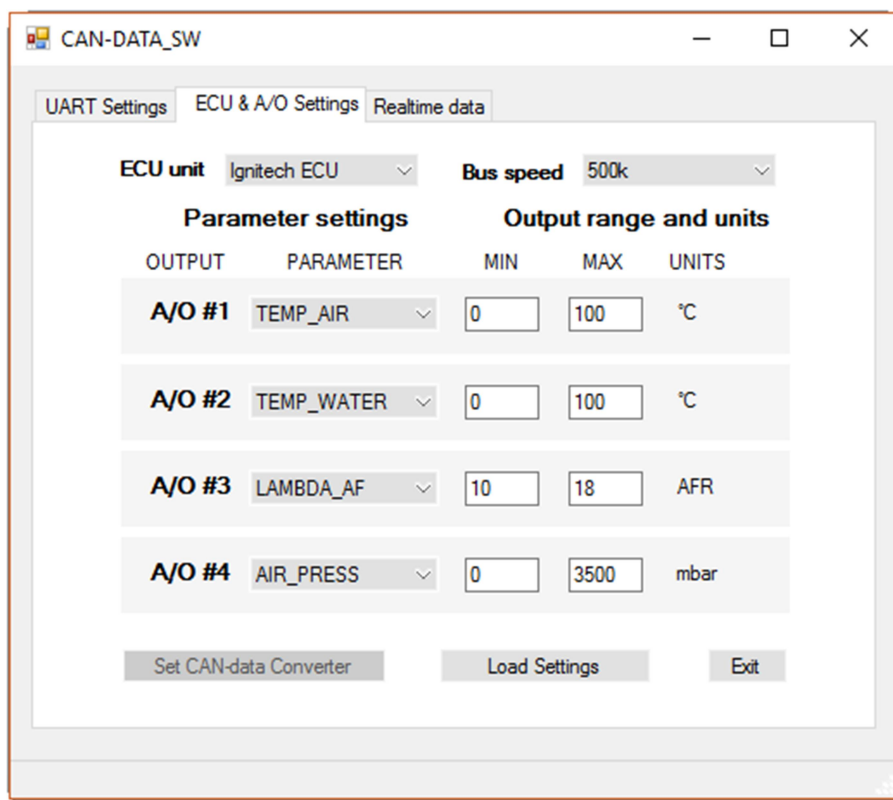
Klepnutím na tlačítko „*Disconnect*“ dojde k ukončení spojení a uvolnění sériové linky.



Obrázek 34 – Karta nastavení UART

6.3.1.1 Parametrizace analogových výstupů

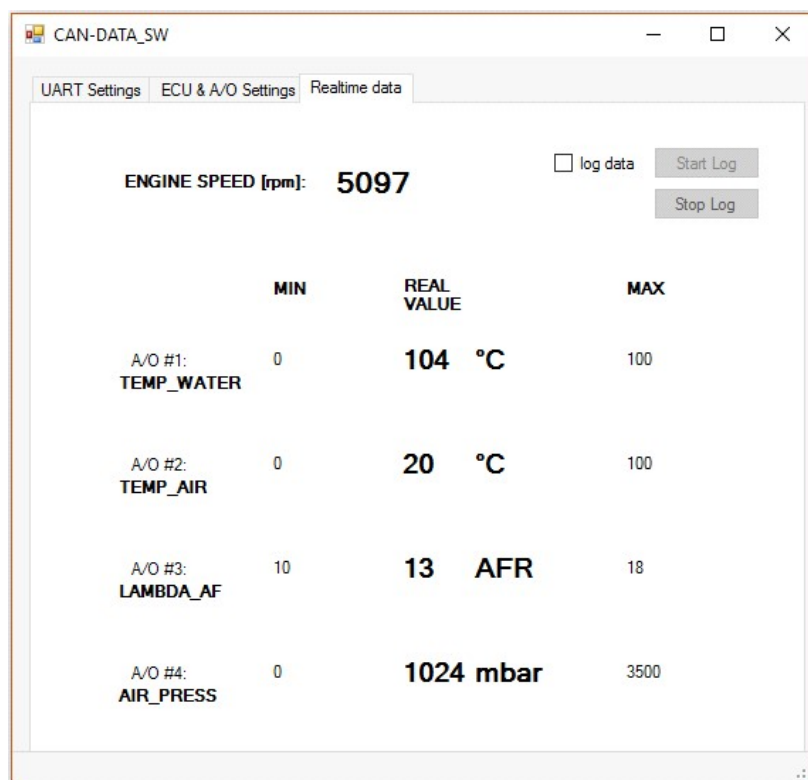
Na kartě *ECU & A/O Settings* je celkem 6 rozbalovacích seznamů. V prvním a druhém z nich si uživatel volí typ řídicí jednotky a rychlost, kterou je jednotka připojena na sběrnici CAN. Následující 4 rozbalovací seznamy patří 4 analogovým výstupům, z každého z nich si uživatel zvolí, které parametry na daném analogovém výstupu budou vyčítány. Ke každému parametru si uživatel zvolí jeho minimální a maximální hodnotu. V tomto rozsahu hodnot je potom daný parametr vizualizován a také zaslán na analogový výstup, jehož výstup je vždy v rozsahu 0 ÷ 5V. Po nastavení požadovaných parametrů stačí kliknout na tlačítko „*Set CAN-data Converter*“ a toto nastavení se pošle a následně uloží do převodníku. Pro zpětné vyčtení nastavení ze zařízení stačí poklepat na tlačítko „*Load Settings*“.



Obrázek 35 – Karta parametrizace zařízení

6.3.1.2 Vizualizace vyčítaných dat ze sběrnice CAN

Vizualizace přijatých dat je realizována prostřednictvím karty „*Realtime data*“ viz. obrázek 34. Aplikace periodicky vysílá dotaz na reálná data s periodou 500 ms a s touto samou periodou jsou i data na kartě aktualizována. Na následujícím obrázku je zachycena vizualizace dat po simulování z log souboru, který mi byl poskytnut a je přímo ze skutečného záznamu z jednotky Ignitech.

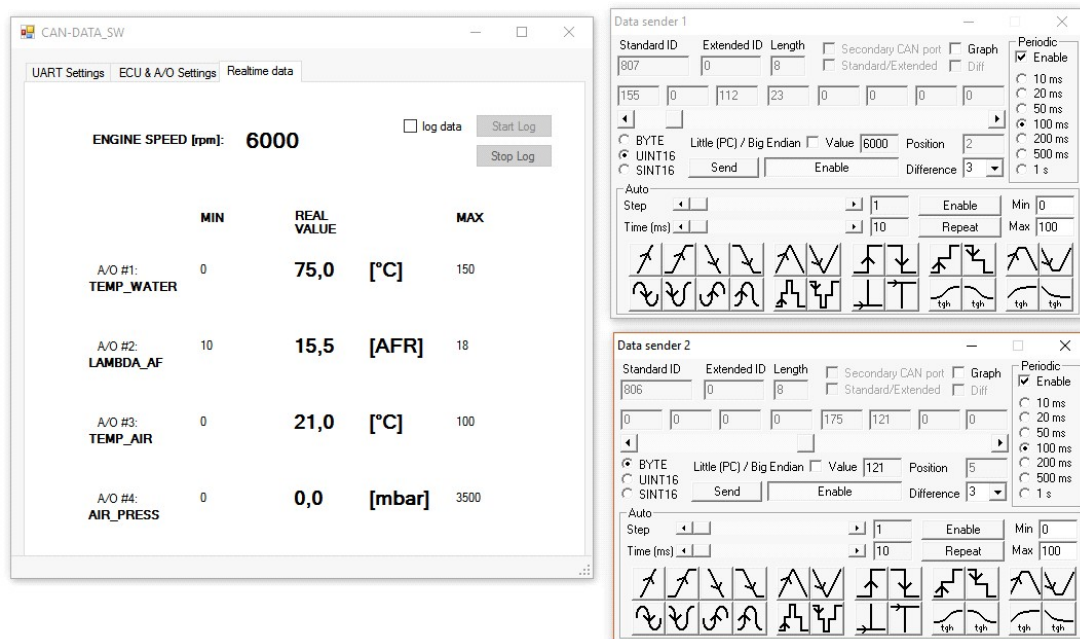


Obrázek 36 – Vizualizace naměřených dat

6.3.1.3 Ověření funkce vizualizace přijatých dat

Test správné vizualizace proběhl jednoduše a rychle. Otestována byla vizualizace otáček motoru, což obsahuje zpráva s ID807 a její hodnotou 6000 v bajtech na pozici 2 a 3. Se stejným ID byla též odeslána data pro hodnotu lambda koeficientu, která byla zvolena náhodně na 15,5 AFR. Hodnoty obou těchto parametrů jsou zasílány v okně Data Sender 1 na obrázku v pravé horní části.

V okně Data Sender 2 (Obrázek 37, vpravo dole) byly nasimulovány a zaslány hodnoty obou teplotních čidel od teploty vody a teploty vzduchu. Obě tyto hodnoty jsou uloženy v jednom bajtu ve zprávě CAN, konkrétně na pozici 4 pro teplotu vody a na pozici 5 pro teplotu vzduchu.



Obrázek 37 – Test funkčnosti vizualizace reálných dat

6.3.1.4 Logování dat

Logování dat je realizováno po stisku checkboxu „log data“, po odkliknutí tohoto checkboxu dojde k ukončení logování. Vytvořený log soubor je v textovém formátu uložen. Textový soubor obsahuje v každém řádku jako referenční bod vždy pořadí vzorku následovaný hodnotou otáček a poté hodnotami jednotlivých analogových výstupů v chronologickém pořadí. Každá hodnota je oddělena středníkem pro možnost importu a dalšího zpracování tabulkovým procesorem a následným vynesemím do grafů. Log soubor je ukládán do složky „Dokumenty“ v uživatelském účtu počítače. Název log souboru je dle data jeho vytvoření.

Funkce logování je v tomto stavu pouze dočasná a bude třeba ji zdokonalit a odladit veškeré chyby týkající se její správné funkčnosti.

ZÁVĚR

Cílem této práce byl návrh a realizace převodníku, pro převod přijatých dat ze sběrnice CAN na elektrické signály (napětí, frekvence) za účelem jeho použití v měřicím systému motorové zkušebny.

Myšlenka práce vycházela ze zařízení vytvořeného v rámci Ročníkového projektu, ale musela být od základu přepracována od návrhu nového HW, kompletního přepsání FW a vytvoření uživatelské aplikace pro parametrizaci.

Při přepracování vznikala nemalá řada problémů, která byla dost časově náročná, a za účelem jakékoliv změny v programu bylo při testování postupovat velice opatrně. Dohledávání změn při občasně nefunkčnosti určitých periférií nadělaly občas dost problémů.

Původní převodník měl po otestování problém s příjmem většího množství CAN zpráv, protože zprávy nepřijímal v přerušení. Tento problém byl vyřešen implementací zásobníku, jehož plnění probíhá právě v přerušení od rozhraní CAN. Dalším zdokonalením byla možnost parametrizace převodníku dle uživatelského nastavení. K tomuto účelu byla vytvořena jednoduchá aplikace, s jejíž pomocí si může uživatel zvolit parametr pro každý analogový výstup a jeho rozsah, ve kterém je poté reprezentován na svém výstupu.

Rovněž jsou hodnoty těchto výstupů přímo vizualizovány v uživatelské aplikaci a není tak třeba připojovat měřicí zařízení – osciloskop.

Ověření funkčnosti čímž se zabývá kapitola 6, proběhlo pomocí nalogovaných dat z jednotky Ignitech, pro kterou je převodník primárně určen. Po otestování příjmu nalogovaného souboru se zdá být funkce bezproblémová a hodnoty na frekvenčních i analogových výstupech odpovídají předpokladům.

Tyto hodnoty na výstupech převodníku poté slouží mechanikovi závodního vozidla jako směrodatné informace pro ladění vozu k dosažení těch nejlepších výsledků ať už primárně těch výkonnostních, či dosažení ideálního spalování při minimální spotřebě, což se už netýká sféry závodního sportu.

Celkově tedy návrh, realizace a testování vyšlo podle předpokladů, avšak programová část ještě potřebuje doladit za chodu po ostřejším provozu v motorové zkušebně.

SEZNAM POUŽITÉ LITERATURY

- [1] Datasheet Atmel Corporation - Microcontrollers, 32-bit, and touch solutions [online]. Copyright © [cit. 24.05.2017]. Dostupné z: http://www.atmel.com/Images/Atmel-8209-8-bit%20AVR%20ATmega16M1-32M1-64M1_Summary.pdf
- [2] Datasheet MCP4922: [online]. Copyright © [cit. 24.05.2017]. Dostupné z: <http://ww1.microchip.com/downloads/jp/DeviceDoc/jp547858.pdf>
- [3] Datasheet IRF7103PBF: Semiconductor & System Solutions - Infineon Technologies [online]. Copyright © [cit. 24.05.2017]. Dostupné z: <http://www.infineon.com/dgdl/irf7103.pdf?fileId=5546d462533600a4015355f0e95a1ac4>
- [4] Datasheet MCP2551: [online]. Copyright © [cit. 24.05.2017]. Dostupné z: <http://users.ece.utexas.edu/~valvano/Datasheets/MCP2551.pdf>
- [5] Externí sériové sběrnice SPI a I²C - Root.cz. Root.cz - informace nejen ze světa Linuxu [online]. Copyright © 1998 [cit. 24.05.2017]. Dostupné z: <https://www.root.cz/clanky/externi-seriove-sberrnice-spi-a-i2c/>
- [6] Serial Peripheral Interface Bus - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus
- [7] Sběrnice CAN: Elektorevue [online]. Dostupné z: <http://www.elektorevue.cz/clanky/03021/index.html>
- [8] MAŠEK, Zdeněk. Přednášky z předmětů AMSTP a PAMPP

SEZNAM OBRÁZKŮ

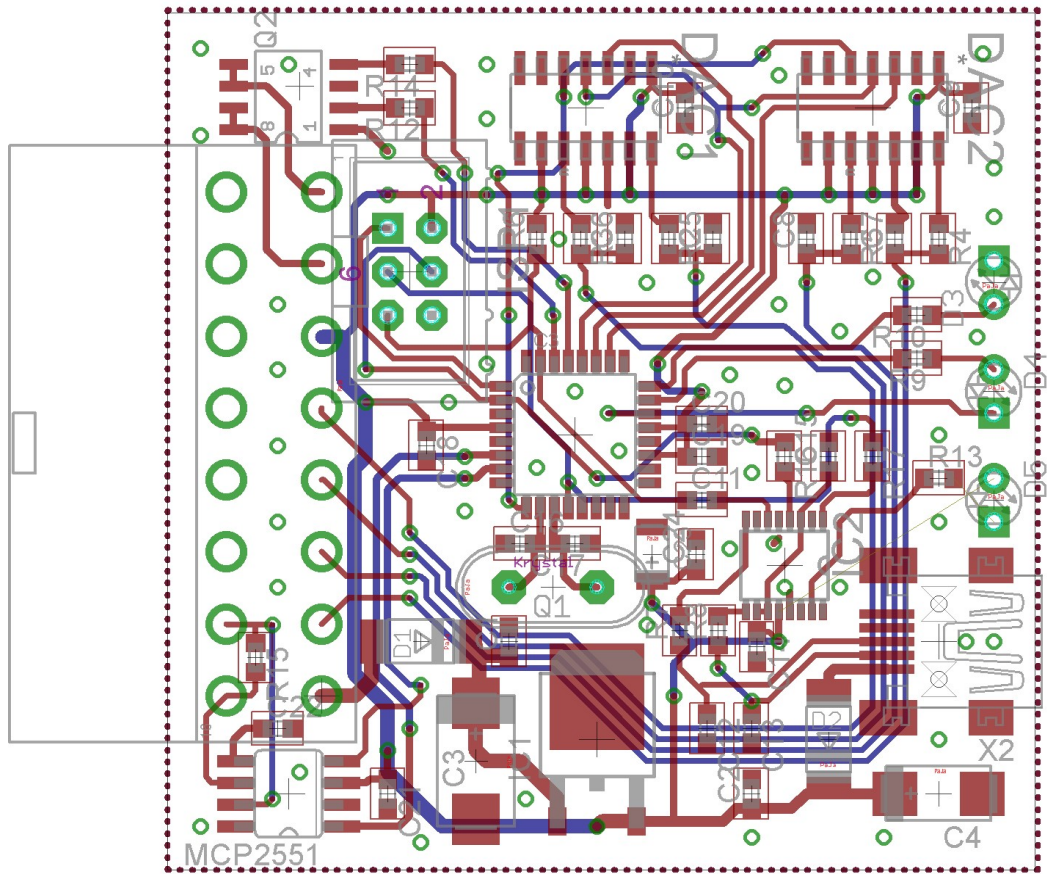
Obrázek 1 – Struktura uzlu na sběrnici	17 -
Obrázek 2 – OSI model [8]	18 -
Obrázek 3 – Rozdíl diferenčního napětí.....	19 -
Obrázek 4 – Závislost rychlosti na délce vedení [8]	20 -
Obrázek 5 – Struktura datové zprávy – Standardní formát (11 bit ID).....	22 -
Obrázek 6 – Struktura datové zprávy – Rozšířený formát (29 bit ID).....	23 -
Obrázek 7 – Pinout konektoru CANON9	23 -
Obrázek 8 – Možnosti zapojení na sběrnici SPI [6].....	25 -
Obrázek 9 – Blokové schéma zařízení	27 -
Obrázek 10 – Obvodové zapojení stabilizátoru napětí 7805CG	28 -
Obrázek 11 – ATMEGA16M1 pinout (pouzdro TQFP) [1]	30 -
Obrázek 12 – Pouzdro a vnitřní struktura MCP4922	32 -
Obrázek 13 – Bitové rozložení zapisovaných dat [2]	33 -
Obrázek 14 – Obvodové zapojení D/A převodníků MCP4922.....	34 -
Obrázek 15 – Pouzdro a vnitřní struktura IRF7103 [3]	36 -
Obrázek 16 – Obvodové zapojení tranzistorů IRF7103.....	36 -
Obrázek 17 – Vnitřní struktura MCP2551 [3]	37 -
Obrázek 18 – Obvodové zapojení budiče MCP2551	38 -
Obrázek 19 – Obvodové zapojení FTDI převodníku FT230X	39 -
Obrázek 20 – Nastavení CBUS pinů v FT Prog.....	40 -
Obrázek 21 – Obvodové zapojení indikačních LED diod.....	42 -
Obrázek 22 – Zapojení 16-pin konektoru N42GK.....	44 -
Obrázek 23 – Celková DPS.....	45 -
Obrázek 24 – Osazená DPS	48 -
Obrázek 25 – profil hliníkové krabičky	48 -
Obrázek 26 – ručně pilovaná krycí víčka pro krabičku	49 -
Obrázek 27 – Výsledná krabička, přední část	50 -
Obrázek 28 – Výsledná krabička, zadní část.....	50 -
Obrázek 29 – Náhled nastavení externího programátoru.....	52 -
Obrázek 30 – Programovací tlačítko alternativního programátoru.....	53 -
Obrázek 31 – Vývojový diagram funkce Main()	54 -
Obrázek 32 – Zpráva CAN s ID807 (otáčky motoru).....	62 -
Obrázek 33 – Snímek změřené frekvence při 6000 ot./min.....	63 -

Obrázek 34 – Karta nastavení UART	- 66 -
Obrázek 35 – Karta parametrizace zařízení	- 67 -
Obrázek 36 – Vizualizace naměřených dat	- 68 -
Obrázek 37 – Test funkčnosti vizualizace reálných dat	- 69 -

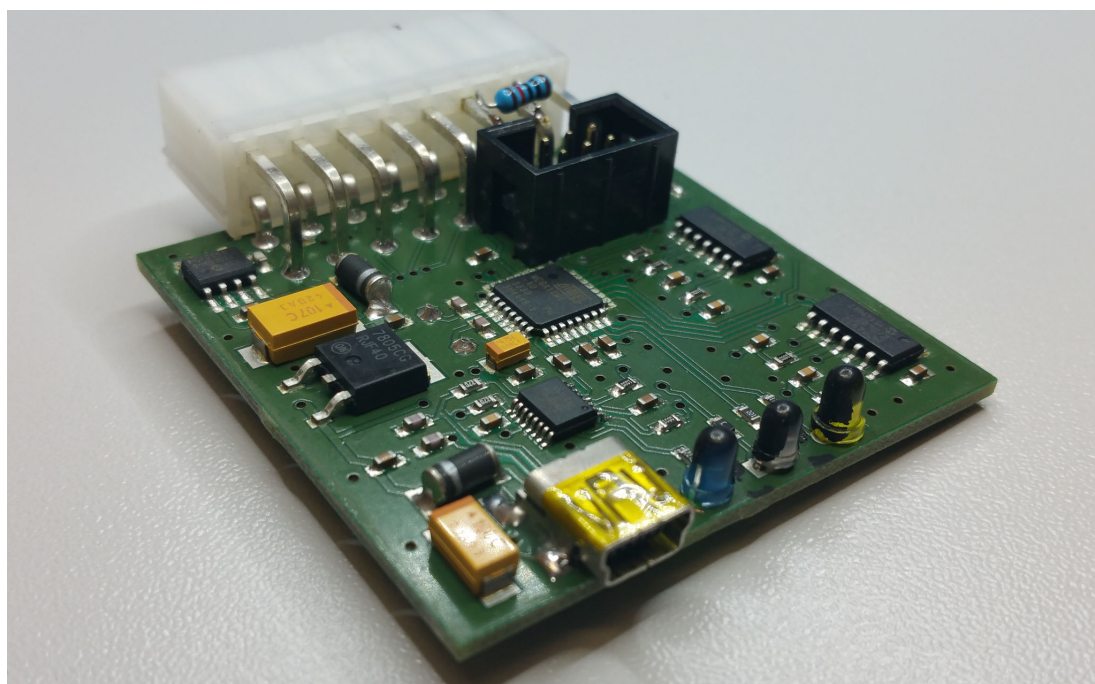
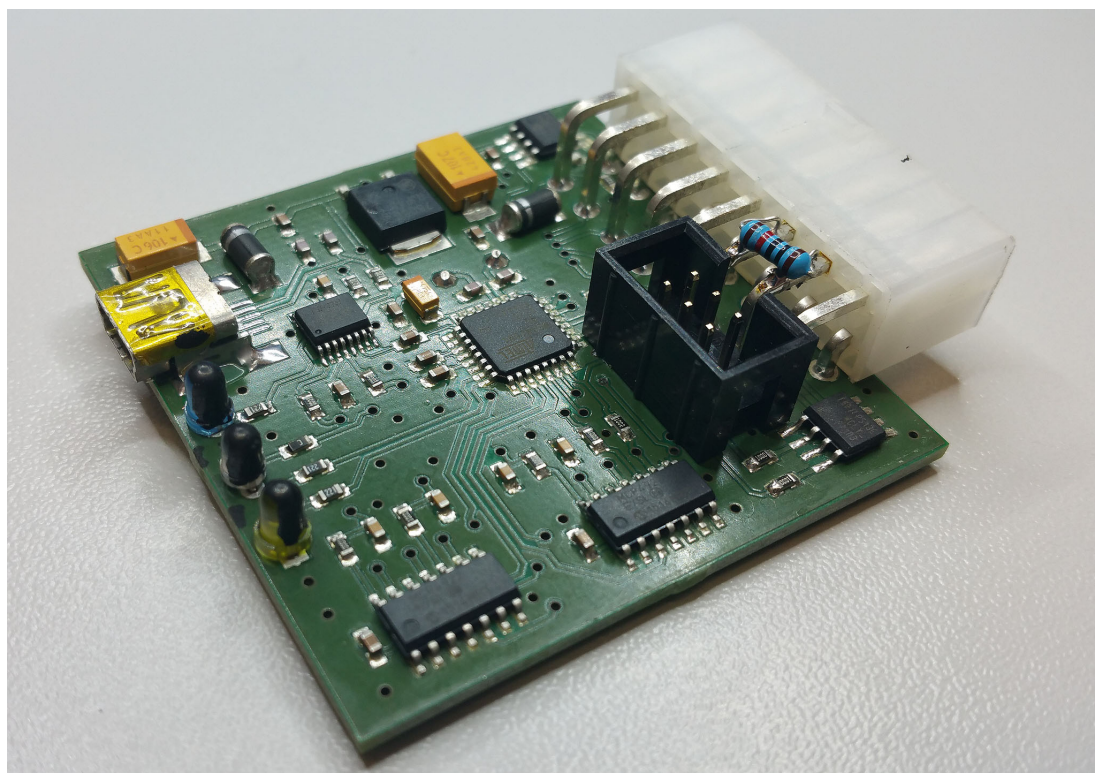
SEZNAM TABULEK

Tabulka 1 – Parametry jednotky Ignitech	- 13 -
Tabulka 2 – Parametry jednotky DTA	- 13 -
Tabulka 3 – Význam pinů konektoru CANON9.....	- 24 -
Tabulka 4 - Rozdělení MCU dle paměti	- 29 -
Tabulka 5 – Barevné označení LED a jejich význam	- 41 -
Tabulka 6 – Složení zpráv sériové linky	- 57 -
Tabulka 7 – Výsledek měření frekvenčního výstupu (otáčky motoru).....	- 63 -
Tabulka 8 – Výsledky měření na analogovém výstupu 1	- 64 -
Tabulka 9 – Výsledky měření na analogovém výstupu 2	- 64 -

PŘÍLOHA A – NÁVRH DPS

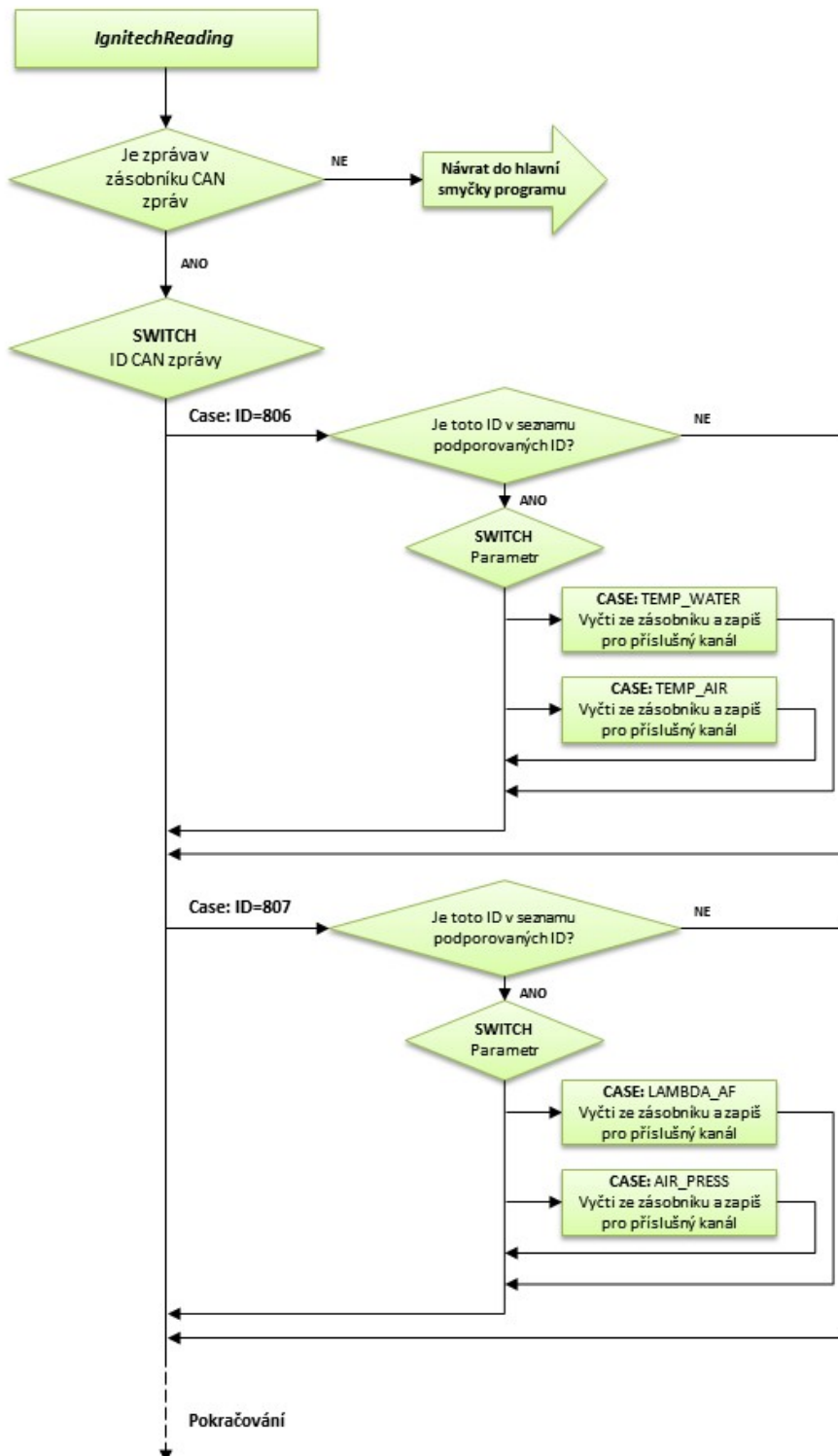


PŘÍLOHA B – FOTOGRAFIE ZAŘÍZENÍ

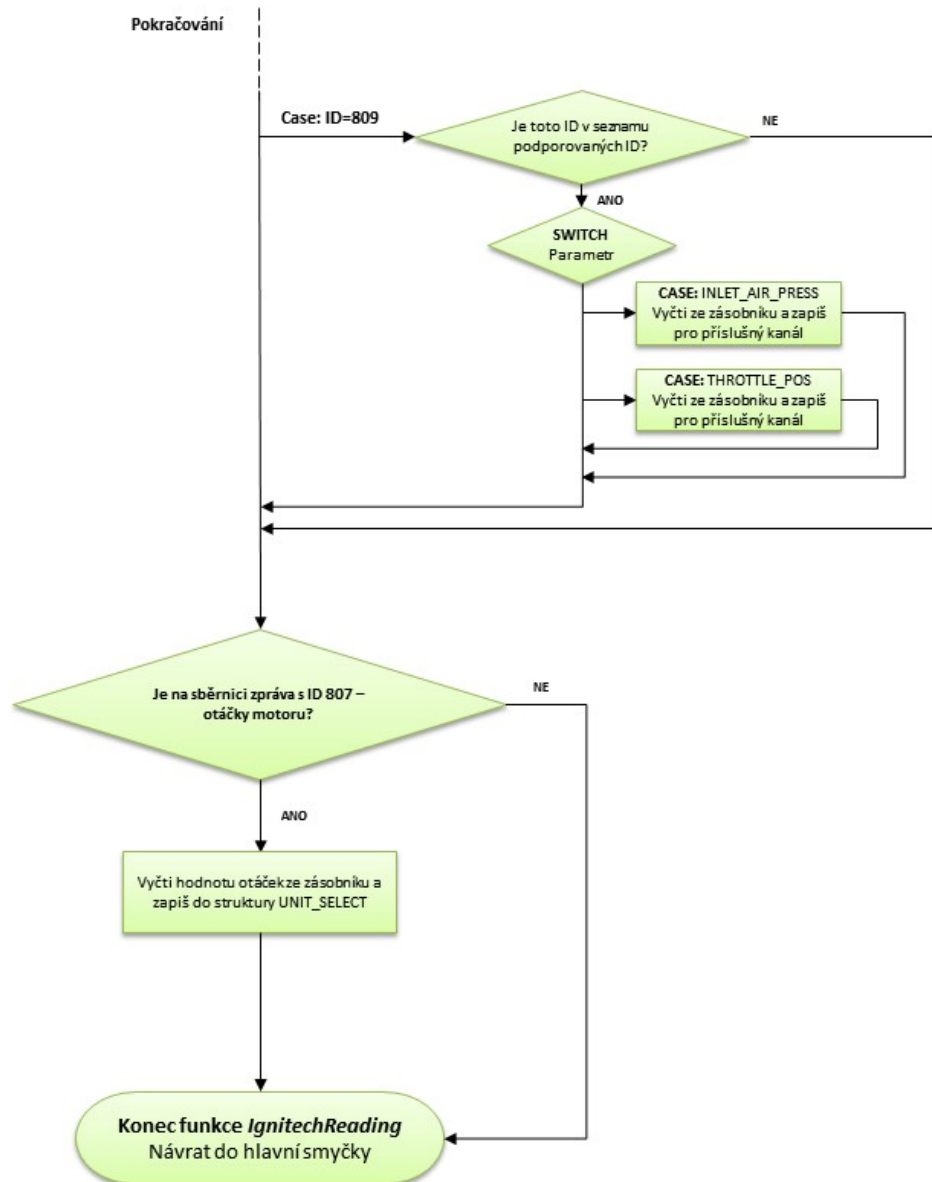


PŘÍLOHA C – VÝVOJOVÉ DIAGRAMY

Funkce *IgnitechReading()* – část A

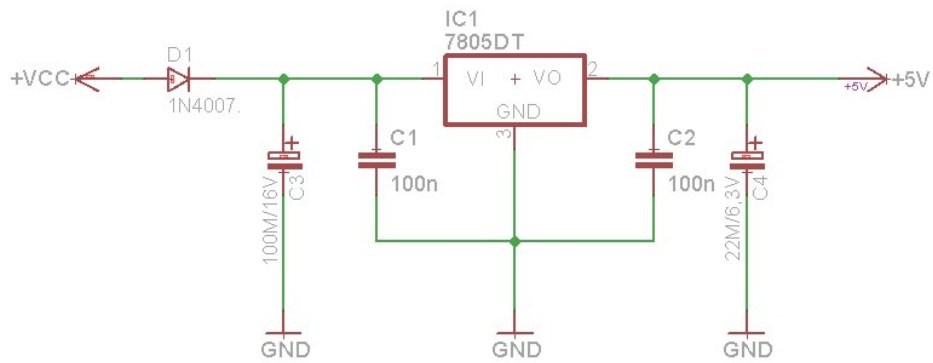


Funkce *IgnitechReading()* – část B

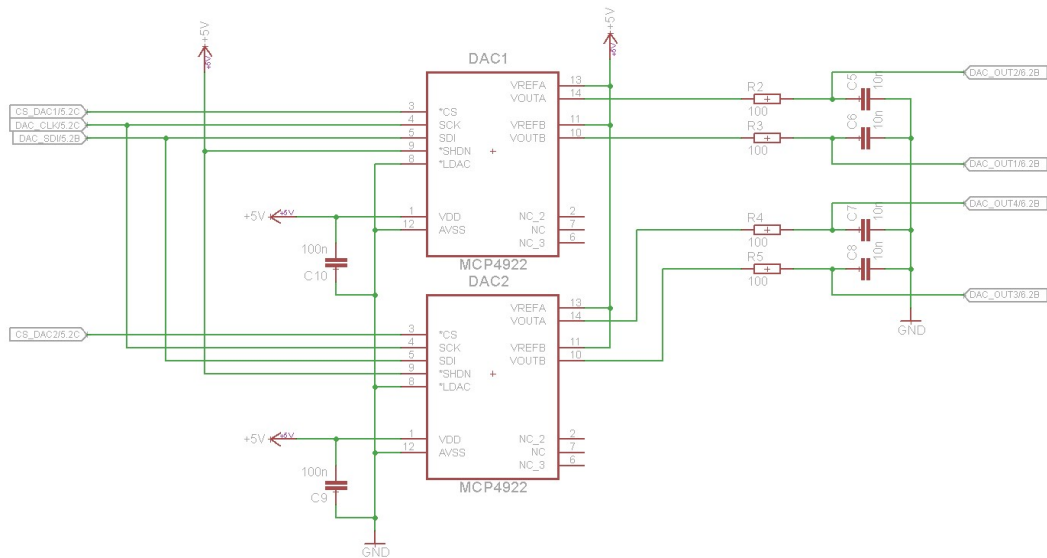


PŘÍLOHA D – SCHÉMATA ZAPOJENÍ

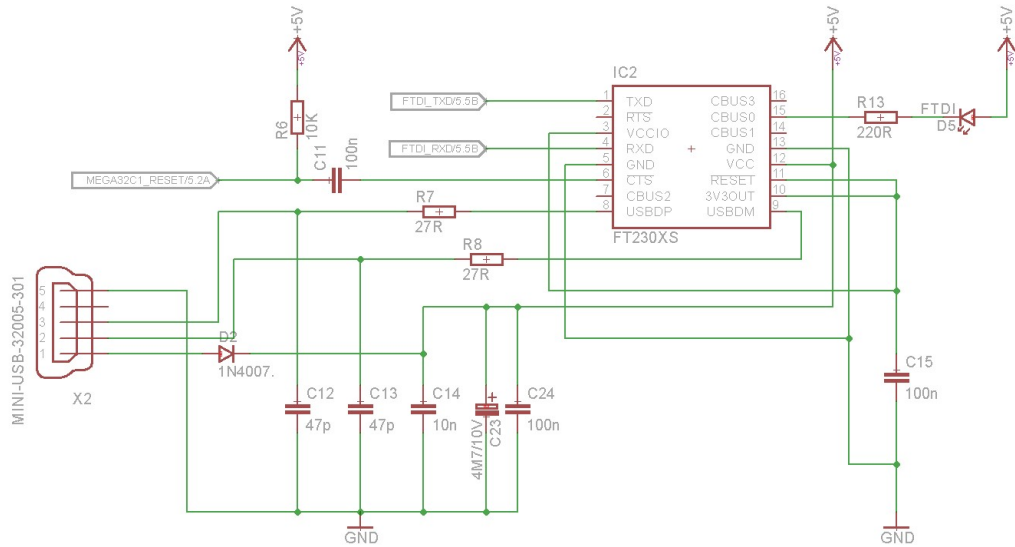
Napájecí zdroj



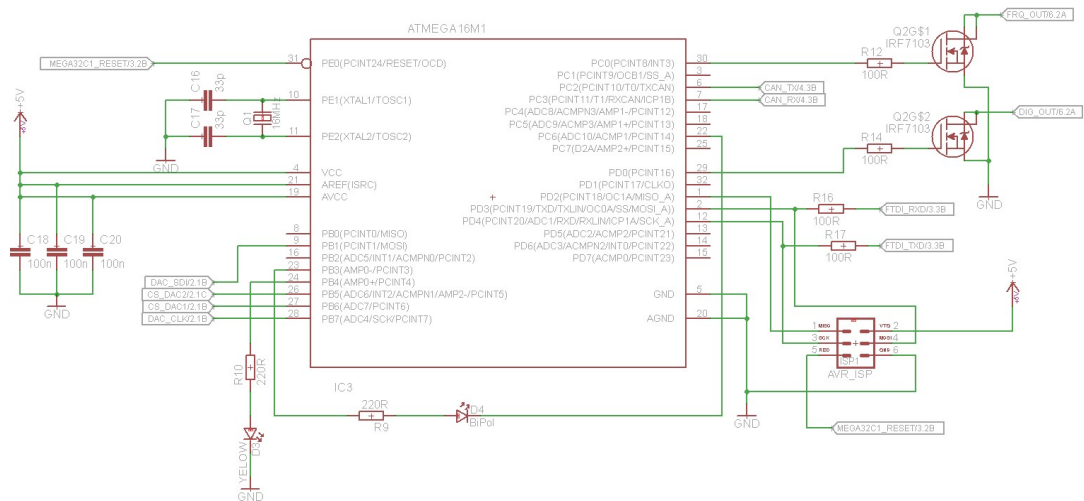
D/A převodníky



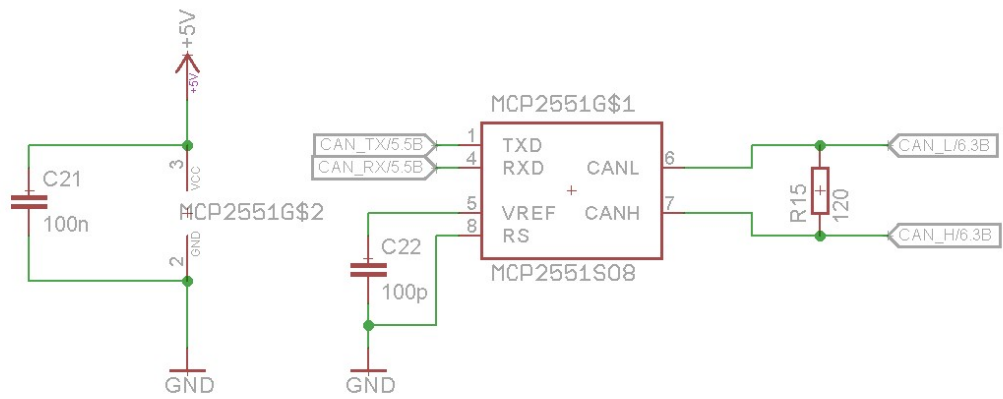
FTDI-FT230X



ATMEGA16M1



CAN budič



Konektor

