



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezipřoborových studií ■

# Zobrazovací jednotka pro experimentální elektromobil

## Diplomová práce

*Studijní program:* N2612 – Elektrotechnika a informatika

*Studijní obor:* 1802T007 – Automatické řízení a inženýrská informatika

*Autor práce:* **Bc. Jan Kovář**

*Vedoucí práce:* Ing. Pavel Jandura

---

Liberec 2015



TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky a mezioborových studií  
Akademický rok: 2014/2015

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan Kovář**  
Osobní číslo: **M13000210**  
Studijní program: **N2612 Elektrotechnika a informatika**  
Studijní obor: **Automatické řízení a inženýrská informatika**  
Název tématu: **Zobrazovací jednotka pro experimentální elektromobil**  
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

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

1. Provedte rešerši aktuálně používaných řešení zobrazovacích jednotek pro řidiče u moderních motorových vozidel, zejména se zaměřením na elektromobily.
2. Vyberte vhodnou mikrokontrolérovou platformu a display pro zobrazovací jednotku. Navrhněte schéma zapojení a jednotku realizujte.
3. Pro jednotku naprogramujte vhodný firmware (i grafické zobrazení), který bude zobrazovat aktuální jízdní informace, primárně získané po sběrnici CAN od systému BMS, nabíječů a měničů pohonu vozidla. Sekundárně pak pomocí digitálních a analogových vstupů.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **40–50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

- [1] **ĎAĎO, Stanislav. Senzory a měřicí obvody. 1. vyd. Praha: ČVUT, 1996, 315 s. ISBN 80-010-1500-9.**
- [2] **BANZI, Massimo. Getting started with Arduino. 2nd ed. Farnham: O'Reilly, 2011, 299 pages. ISBN 978-144-9309-879. Dostupné z: [http://phylab.fudan.edu.cn/lib/exe/fetch.php?media=yuandi:arduino:getting\\_started\\_with\\_arduino\\_v2.pdf](http://phylab.fudan.edu.cn/lib/exe/fetch.php?media=yuandi:arduino:getting_started_with_arduino_v2.pdf)**
- [3] **WEICKER, Phillip. A systems approach to lithium-ion battery management. Boston: Artech House, c2010, 299 pages. ISBN 16-080-7659-8.**

Vedoucí diplomové práce:

**Ing. Pavel Jandura**

Ústav mechatroniky a technické informatiky

Konzultant diplomové práce:

**Ing. Martin Kysela**

Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: **10. října 2014**

Termín odevzdání diplomové práce: **15. května 2015**

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Milan Kolář, CSc.  
vedoucí ústavu

V Liberci dne 10. října 2014

# Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

15. 5. 2015

Podpis:



# Poděkování

Děkuji vedoucímu mé práce Ing. Pavlu Jandurovi za cenné rady a připomínky, metodické vedení práce a veškerý jeho čas, který věnoval této diplomové práci.

Tato práce byla podpořena Studentskou grantovou soutěží Technické univerzity v Liberci.

# Anotace

Cílem práce je funkční zobrazovací jednotka experimentálního elektromobilu. Zobrazovací jednotka musí umět číst data přenášená po CAN sběrnici. Po této CAN sběrnici bude komunikovat frekvenční měnič, BMS, nabíječka baterie, volící páka DSG převodovky. Dalším problémem je zprovoznění modulu LCD displeje připojeného k systému Arduino včetně jeho dotykové vrstvy.

Po splnění těchto kroků se dostávám k vytvoření zobrazovacího zařízení informujícího o stavech vozidla. Je zde požadavek na řidičsky přívětivé rozvržení informujících prvků a jejich viditelnost.

V teoretické části diplomové práce se zabývám popisem CAN sběrnice, výběrem vhodného mikrokontroléru a informacemi o použitých komponentách sestavené zobrazovací jednotky.

V praktické části je cílem představit použité knihovny a jejich implementaci pro náš účel. Po těchto přípravách jsem představil palubní desku s uživatelsky přijatelným rozhraním.

Nezbytnou součástí práce je CD-ROM obsahující tuto zprávu, veškeré zdrojové kódy a záznamy měření CAN komunikace.

## Klíčová slova

CAN BUS, Arduino DUE, LCD, Open Source, BMS, elektromobil, zobrazovací jednotka

# Abstract

The aim of this thesis is to create a display unit for university experimental electric vehicle. Display unit must read data from CAN bus. Via CAN bus will communicate frequency inverter, BMS, battery charger and selector lever. Secondary issue is to put into operation 7" LCD touchpanel with selected development board Arduino DUE.

After these pre operations I created dashboard with driver friendly layout. It is necessary for all the elements to be clearly visible.

The theoretical part of the thesis deals with the description of the CAN bus, selection of an appropriate microcontroller and information about the components of the assembled display unit.

In the practical part my goal is to introduce the libraries and their implementation for our case. After these pre operations I created dashboard with driver friendly layout.

CD - ROM contains this report, source code and measurement records from CAN communication.

## Keywords

CAN BUS, Arduino DUE, LCD, Open Source, BMS, electric vehicle, display unit

# Obsah

Prohlášení.....	4
Poděkování.....	5
Anotace .....	6
Klíčová slova .....	6
Abstract.....	7
Keywords .....	7
Obsah .....	8
Seznam obrázků.....	11
1 Úvod .....	12
2 Zobrazovací jednotky u elektromobilů.....	13
2.1.1 Nissan Leaf .....	13
2.1.2 Mitsubishi i-MiEV .....	13
2.1.3 BMW i3 .....	14
2.1.4 Tesla Model S .....	14
3 Sběrnice CAN.....	17
3.1.1 Fyzická vrstva .....	18
3.1.2 Transportní vrstva .....	18
3.2 Definice CAN zpráv.....	19
3.2.1 Standardní zpráva .....	19
3.2.2 Rozšířená zpráva.....	21
3.2.3 Zabezpečení CAN sběrnice .....	21
3.3 Arduino UNO.....	22
3.3.1 Arduino DUE.....	23



3.3.2	Raspberry PI .....	23
4	Použité komponenty zobrazovací jednotky .....	25
4.1	Arduino DUE .....	25
4.1.1	Přehled základních vlastností.....	26
4.1.2	CANbus .....	26
4.1.3	Vývojové prostředí Arduino IDE .....	27
4.2	MD070SD .....	28
4.3	CAN Board SN65HVD230.....	30
4.4	Orion BMS .....	32
4.4.1	Sledování parametrů .....	32
4.4.2	Vyhodnocování parametrů.....	32
4.4.3	Komunikace .....	33
4.5	Nabíječka Elcon .....	36
4.6	Sevcon GEN4.....	36
4.7	Volící páka převodovky DSG .....	38
4.7.1	Měření ve Škoda Auto a.s.....	38
4.7.2	Nalezené stupně .....	40
5	Realizace programu pro Arduino DUE .....	42
5.1	Zapojení.....	42
5.2	Prvotní nastavení a použité knihovny .....	43
5.2.1	UTFT .....	44
5.2.2	UTouch .....	44
5.2.3	DueTimer .....	45
5.2.4	Due_can .....	45
5.3	Grafické rozhraní .....	46
5.3.1	Třída Pbar .....	47

5.3.2	Třída Kvalt.....	47
5.3.3	Metoda zapis7SEG .....	47
5.4	Zpracování CAN zpráv .....	48
6	Závěr.....	50
	Zdroje.....	51
	Obsah přiloženého CD-ROM .....	53

# Seznam obrázků

Obrázek 1: Přístrojový panel Nissan Leaf  foto: Nissan.....	13
Obrázek 2: Přístrojový panel Mitsubishi iMiEV  foto: Mitsubishi.....	13
Obrázek 3: Přístrojový panel BMW i3  foto: BMW.....	14
Obrázek 4: Tesla Model S pohled z pozice řidiče  foto: Steve Jurvetson.....	15
Obrázek 5: Tesla Model S přístrojový panel  foto: Tesla .....	15
Obrázek 6: Tesla Model S přístrojový panel  foto: Tesla .....	16
Obrázek 7: Fyzická vrstva CAN sběrnice[3].....	18
Obrázek 8: Realizace fyzické vrstvy protokolu CAN[3].....	18
Obrázek 9: Standardní formát CAN zprávy CAN 2.0B  CC licence.....	19
Obrázek 10: Arduino UNO[4].....	22
Obrázek 11: Raspberry PI[6].....	23
Obrázek 12: Arduino DUE[4].....	25
Obrázek 13: Přední strana zobrazovacího modulu[8].....	28
Obrázek 14: Zadní strana zobrazovacího modulu[8].....	29
Obrázek 15: CTE TFT LCD/SD card shield for Arduino DUE[9].....	29
Obrázek 16: CAN Board SN65HVD230 [10].....	30
Obrázek 17: Schéma zapojení WaveShare SN65HVD230 CAN Board[10].....	30
Obrázek 18: Togglebit Arduino DUE CANshield[11].....	31
Obrázek 19: Konfigurace CANbus Orion BMS.....	33
Obrázek 20: Editace zpráv Orion BMS.....	34
Obrázek 21: Konfigurace křivky stavu nabití.....	35
Obrázek 22: Konfigurace flag bytů Orion BMS.....	35
Obrázek 23: Nastavení CAN zpráv Sevcon GEN4.....	37
Obrázek 24: Nastavení CAN zprávy Sevcon GEN4.....	37

Obrázek 25: Volící páka DSG [14].....	38
Obrázek 26: CAN komunikace.....	41
Obrázek 27: Sestavená zobrazovací jednotka.....	42
Obrázek 28: Vývojové prostředí Arduino .....	43
Obrázek 29: Výstup zobrazovací jednotky .....	46

# 1 Úvod

Vývoj přístrojových desek v osobních vozech za dobu vývoje automobilismu značně pokročil. Neustále se objevuje snaha o použití levnějších, přesnějších a spolehlivějších řešení. Od klasických přístrojů s mechanickým převodem, přes přístroje s analogovým signálem až po práci s digitálními hodnotami vysílanými po sběrnici.

LCD panely za poslední roky udělali velký pokrok. Došlo ke zvýšení rozlišení, k dosažení větších zorných úhlů i viditelnosti na slunci. Zároveň jsou cenově dostupné i rozměrné modely. Díky tomu řada vozů již klasicky pojaté přístrojové desky nevyužívá a přechází se na toto moderní řešení.

Historie vývoje elektromobilů se na půdě Technické univerzity v Liberci datuje již od roku 2011. V současné době probíhá stavba v pořadí druhého modelu lehkého bateriového elektromobilu. Jedná se o dvoumístné dvoustopé vozidlo karoserie typu Roadster vybaveno samonosným rámem vozidla tvořeným z hliníkových profilů. Jeho technické vybavení zahrnuje tři rozdílná zařízení komunikující po CAN sběrnici.

Pro zobrazování jejich výstupů bylo třeba navrhnout a sestavit zobrazovací jednotku, která by ukazovala základní informace o vozidle.

## 2 Zobrazovací jednotky u elektromobilů

V následující části je ukázka zobrazovacích jednotek u čtyř aktuálně vyráběných elektromobilů. Z obrázků je patrné, že konvenční řešení s využitím pohyblivých ukazatelů se již stávají v oblasti elektromobilů minulostí

### 2.1.1 Nissan Leaf

Jeden z nejdostupnějších elektromobilů využívá několik segmentových displejů rozložených do dvou jednotek. Horní jednotka zahrnuje rychloměr, ekonoměr, hodiny a teploměr. Zbylé hodnoty zobrazuje spodní jednotka.



Obrázek 1: Přístrojový panel Nissan Leaf | foto: Nissan

### 2.1.2 Mitsubishi i-MiEV

Další z běžných elektromobilů je vybaven třemi jednoduchými segmentovými displeji. Ostatní stavy jsou zobrazeny standardními kontrolkami.



Obrázek 2: Přístrojový panel Mitsubishi iMiEV | foto: Mitsubishi

### 2.1.3 BMW i3

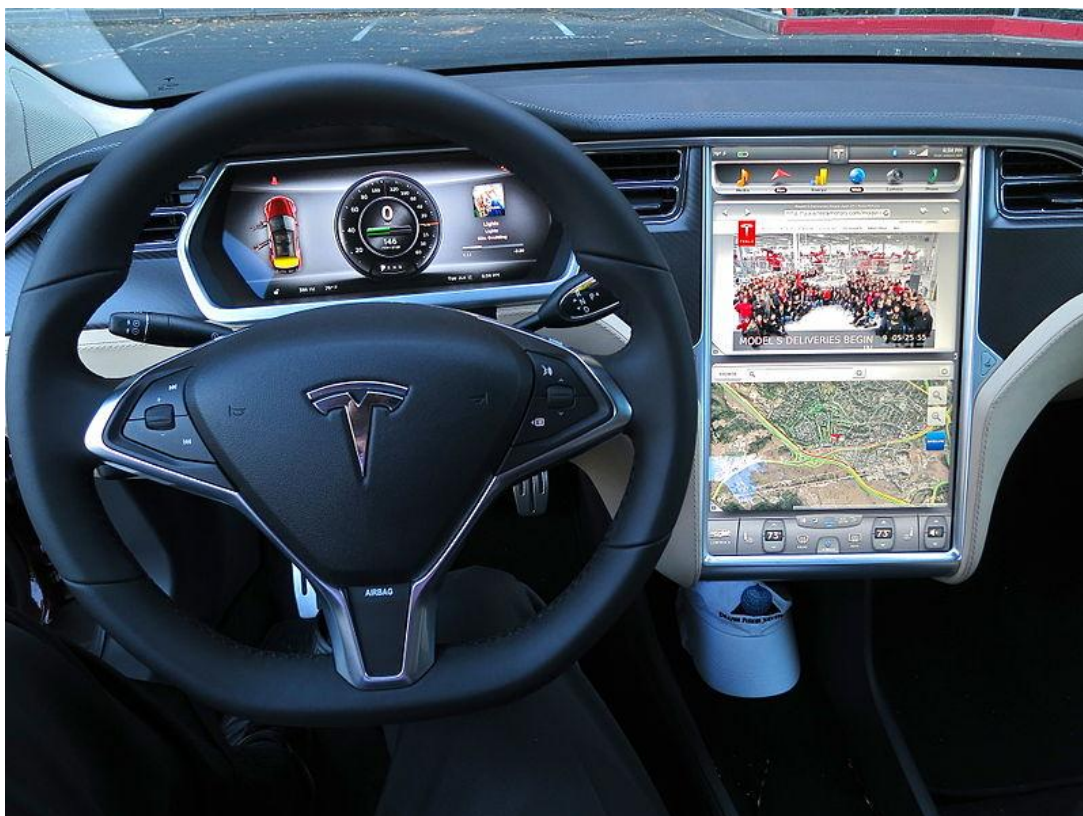
BMW i3 je malý městský elektromobil vybavený LCD přístrojovým panelem. Jednoduchá grafika je efektní a přitom ergonomická pro řidiče..



Obrázek 3: Přístrojový panel BMW i3| foto: BMW

### 2.1.4 Tesla Model S

Tesla Model S patří mezi nejlepší sériově vyráběné vozy současnosti. Toto hodnocení si nese i díky přístrojovému panelu vybavenému rozměrným barevným LCD displejem zobrazujícím různé stavy vozu. Dále zaujme 17“ dotykový barevný LCD vyplňující středový panel.



Obrázek 4: Tesla Model S pohled z pozice řidiče | foto: Steve Jurvetson



Obrázek 5: Tesla Model S přístrojový panel | foto: Tesla



Přístrojový panel zobrazuje stavy podle požadavku řidiče. Kromě informace rychlosti vozu, zvolené polohy volící páky automatické převodovky nebo dennímu nájezdu, lze zobrazit i dvě uživatelská „okna“. V těchto částech obrazovky je například možné zobrazit dojezd, graf závislosti Wh/míle nebo obrazovku navigace či autorádia.



Obrázek 6: Tesla Model S přístrojový panel| foto: Tesla

## 3 Sběrnice CAN

Sběrnice CAN (controller area network) je sériová sběrnice vytvořená za účelem zpřístupnění komunikace mikrokontrolerů a dalších zařízení mezi sebou bez hostujícího počítače.

CAN sběrnice je původně navržena pro automobilové využití, nicméně v současné době se používá také v letectví, námořní dopravě, v průmyslové výrobě a má i využití v medicínských zařízeních.

Vývoj CAN sběrnice začal v roce 1983 v Robert Bosch GmbH. Protokol byl oficiálně představen v roce 1986 na kongresu Society of Automotive Engineers (SAE) v Detroitu. S prvním CAN kontrolérem přišel Philips a Intel v roce 1987.

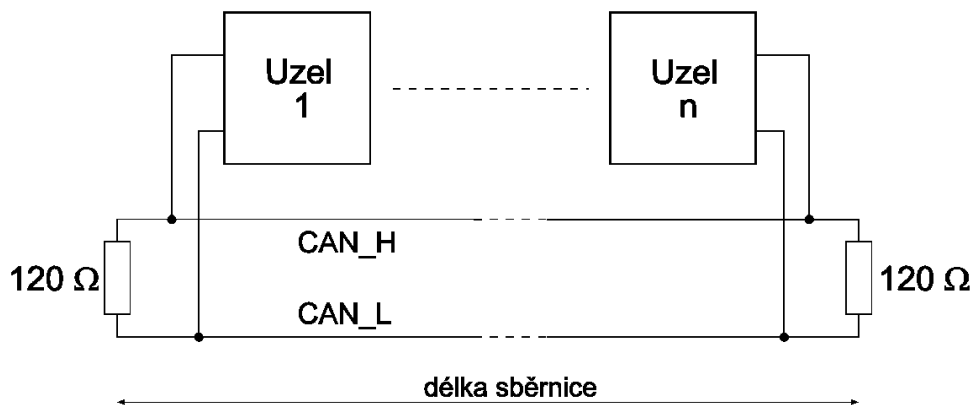
Bosch vydal několik verzí CAN specifikace a poslední CAN 2.0 byla vydána v roce 1991. Specifikace zahrnuje dvě části. Část A je standardní formát s 11-bitovým identifikátorem. Část B představuje rozšířený formát s 29-bitovým identifikátorem. CAN zařízení využívající 11-bitový identifikátor se běžně nazývá CAN 2.0A a zařízení využívající 29-bitový identifikátor používá označení CAN 2.0B. Tyto standardy jsou volně k dispozici od firmy Bosch spolu s dalšími specifikacemi.[1]

V roce 1993 Mezinárodní organizace pro standardizaci vydala CAN standard ISO 11898, který později přestavila do dvou částí. ISO 11898-1 zahrnuje datovou vrstvu, a ISO 11898-2 zahrnuje CAN fyzickou vrstvu pro high-speed CAN. Později vydaný fault-tolerant CAN podle standardu ISO 11898-3 a zahrnuje fyzickou vrstvu pro malou rychlost. Fyzické vrstvy použité v ISO 11898-2 a ISO 11898-3 nejsou součástí specifikace Bosch CAN 2.0.[2]

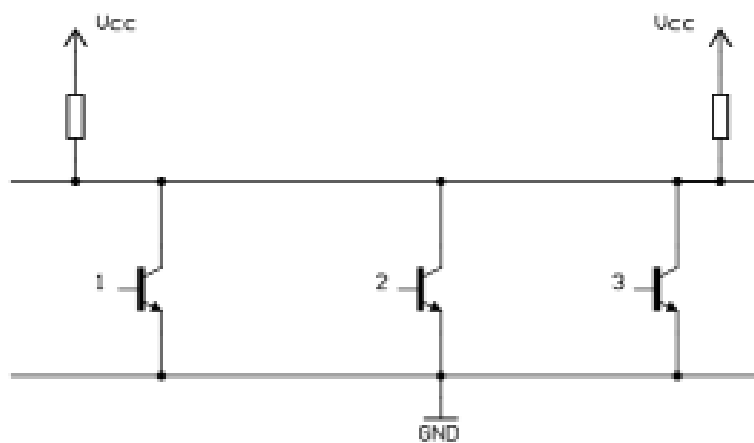
Sběrnice CAN je jedna z pěti protokolů využívající v palubní diagnostice OBD-II. Zde je definován pod normou ISO 15765. V rámci OBD II může datová komunikace po sběrnici probíhat v rychlostech 250 kbit/s a 500 kbit/s. CAN<sub>H</sub> využívá pin 6, CAN<sub>L</sub> pin 14 konektoru OBD II.

### 3.1.1 Fyzická vrstva

Je tvořena dvěma vodiči označenými CAN\_H a CAN\_L. Základním požadavkem je realizace funkce logického součinu. Jedná se o symetrický spoj z nutnosti zvýšení odolnosti proti rušení a rychlosti. Rozlišujeme dvě napěťové úrovně dominantní a recesivní.[1]



Obrázek 7: Fyzická vrstva CAN sběrnice[3]



Obrázek 8: Realizace fyzické vrstvy protokolu CAN[3]

### 3.1.2 Transportní vrstva

Fyzické uspořádání sítě CAN dle ISO 11898

- MAC (Medium Access Control)
- LLC (Logical Link Control)

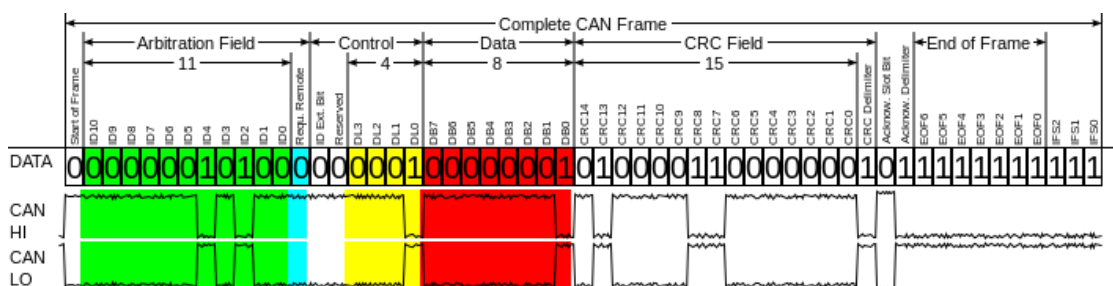
Protokol CAN umožňuje distribuované řízení systémů v reálném čase se vysokou odolností vůči chybám. Představuje protokol typu multi-master, jakýkoliv uzel sběrnice může pracovat jako master a řídit jiné uzly.

## 3.2 Definice CAN zpráv

- **datová zpráva (Data Frame)** – identifikátor ve zprávě odpovídá vysílacímu uzlu. Zpráva obsahuje 0-8 datových bajtů.[3]
- **žádost o data (Remote Frame)** – uzel žádající ostatní uzly na sběrnici o zaslání dat. Zpráva neobsahuje dat. bajty, identifikátor shodný s číslem uzlu, po kterém zaslání dat požadujeme.[3]
- **chybová zpráva (Error Frame)** – libovolný uzel detekující na sběrnici chybu vygeneruje chybovou zprávu.[3]
- **zpráva o přetížení (Overload Frame)** – podobné chybové zprávě, může být vyslána až po konci zprávy.[3]

### 3.2.1 Standardní zpráva

Sběrnici CAN můžeme konfigurovat pro práci s dvěma různými formáty zprávy. Jedná se o standardní a rozšířený rámeček. Na následujícím obrázku můžeme vidět zprávu se standardním 11 bitovým identifikátorem.



Obrázek 9: Standardní formát CAN zprávy CAN 2.0B| CC licence

Každý rámeček začíná startovacím bytem SOF, který zaujímá dominantní úroveň.

### Arbitrační pole

- **Identifikátor zprávy** – 11 b, identifikace významu zprávy a její priority
- **Remote transmission request** – 1b, pro datové rámce je dominantní, pro žádost o data recesivní

### Řídící pole

- **ID Ext. Bit** – 1b, rozlišuje formát zprávy, dominantní pro standardní, recesivní pro rozšířený
- **Rezervovaný bit** – má vždy dominantní úroveň
- **Kód délky dat (DLC)** – 4 b, informuje o počtu bytů ve zprávě (0 – 8 bytů)

### Datové pole

- **Data** – 0- 64 b, přenášená data, jejich velikost je definována v DLC

### CRC pole

- **CRC** – 15 b, cyklický redundantní součet
- **CRC oddělovač** – 1 b, musí být recesivní

### Potvrzení

- **ACK** – 1 b, bit vyhrazený pro potvrzování správnosti přijatých zpráv přijímacími uzly
- **ACD** – 1 b, slouží k oddělení potvrzení

### Zakončení

- **Konec rámce** – 7 b recesivní
- **Mezera mezi zprávami** – 7b recesivní[1]

### 3.2.2 Rozšířená zpráva

Tato zpráva využívá 28-bitový identifikátor zprávy. Identifikátor je rozdělen na dvě části o délkách 11 a 18 bitů.

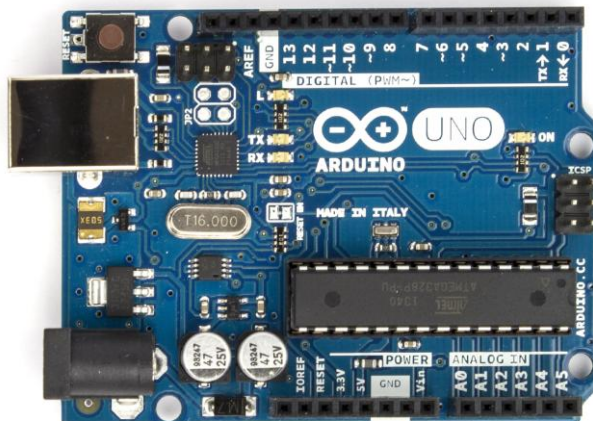
### 3.2.3 Zabezpečení CAN sběrnice

K zabezpečení CAN sběrnice se využívá několika metod, které jsem níže krátce představil.

- **Monitoring** - porovnávání vysílané hodnoty bitu s úrovní na sběrnici.[1]
- **CRC kód** – Cyklický redundantní součet s délkou 15 bitů zaujímá poslední pole vysílané zprávy. Pokud jakýkoliv uzel na sběrnici detekuje chybu, tak dojde k vygenerování chyby CRC.[1]
- **Vkládání bitu** – Pokud bylo po sběrnici vysláno za sebou pět shodných bitů, dojde k vyslání bitu opačné úrovně.[1]
- **Kontrola přijetí zprávy** – Proběhne kontrola zprávy dle specifikace, pokud je na určité pozici bitu zprávy detekována chyba, je vygenerována chyba rámce.[1]
- **Potvrzení přijetí zprávy** – Všechna zařízení připojená ke sběrnici musí korektně přijatou zprávu potvrdit. Proběhne to změnou bitu v poli ACK, ze stavu recessive, jež vysílá vysílač, na stav dominant. Toto probíhá i u zařízení, u kterých zpráva neprojde filtrem.[1]

V následující části uvádím rešerši dostupných zařízení umožňující práci s CAN sběrnici a 7“ displejem.

### 3.3 Arduino UNO



Obrázek 10: Arduino UNO[4]

Jedná se o vývojovou desku vybavenou mikrokontrolérem ATmega328P architektury RISC o frekvenci 16 MHz. Paměť flash pro uložený kód má kapacitu 32 KB. Arduino UNO poskytuje 14 nastavitelných digitálních vstupů / výstupů a 6 analogových vstupů. Programování probíhá pomocí vývojového prostředí Arduino IDE využívající jazyk Wiring podobný jazyku C++. Pro napojení na CAN sběrnici je možné použít rozšiřující shield (desku) osazenou čipy MCP2515 a MCP2551 od výrobce SeedStudio. U verzí s patičkou lze jednoduše vyndat mikrokontrolér ATmega328P a vložit do patice s vlastním PCB. Na trhu se objevuje řada kopií tohoto zařízení, které se prodávají za cenu již od \$3.04 (zdroj: [www.aliexpress.com](http://www.aliexpress.com)). Originál je na trhu dostupný za cenu kolem \$25.[4]

Tabulka 1: Výhody/nevýhody Arduino UNO

Výhody	Nevýhody
Minimální cena	Nižší výkon
Množství hardwarových rozšíření	Nemá nativní CANbus
Mnoho knihoven od nezávislých tvůrců	
Možnost naprogramování vyjmutelného ATmega328P	

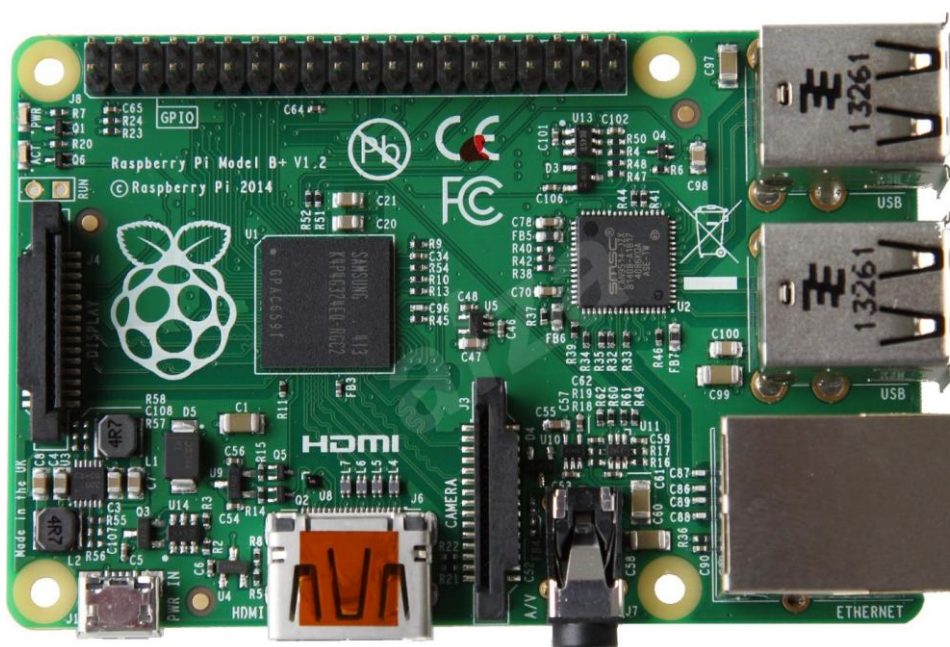
### 3.3.1 Arduino DUE

Vyšší řada Arduina vybavena čipem Atmel SAM3X8E ARM CORTEX-M3 pracujícím na frekvenci 84 MHz. Kapacita paměti flash dosahuje 512 KB. Tato výkonnější verze Arduina zaujme i velkým počtem digitálních vstupů/výstupů a nativním rozhraním sběrnice CAN.[5] K připojení do CAN sběrnice je třeba dodat pouze budící člen.

Tabulka 2: Výhody/nevýhody Arduino DUE

Výhody	Nevýhody
Slušný výkon	Nižší počet dostupných knihoven a shieldů než na Arduino UNO
Mnoho vestavěných rozhraní	Napětí I/O 3.3 V
Má nativní podporu CAN bus	

### 3.3.2 Raspberry Pi



Obrázek 11: Raspberry Pi[6]



Raspberry PI je nízkonákladový počítač o velikosti kreditní karty vybavený ARM procesorem. Vyvíjen je britskou nadací Raspberry Pi Foundation. Základem počítače v první generaci je SoC Broadcom BCM2835, obsahující CPU ARM1176JZF-S o frekvenci 700 MHz, grafický procesor VideoCore IV, paměť RAM o kapacitě 256 MB nebo 512 MB, podle modelu. Pro ukládání dat a zavádění systému se využívá slot na SD kartu. [6]

**Tabulka 3: Výhody/nevýhody Raspberry PI**

<b>Výhody</b>	<b>Nevýhody</b>
Velmi vysoký výkon	Nížší počet dostupných knihoven a rozšiřujících desek než na Arduino UNO
Vestavěné HDMI rozhraní a USB rozhraní	Složitější programování
Více možností	Vyšší cena

## 4 Použité komponenty zobrazovací jednotky

Pro sestavení zobrazovací jednotky bylo využito následujících komponent:

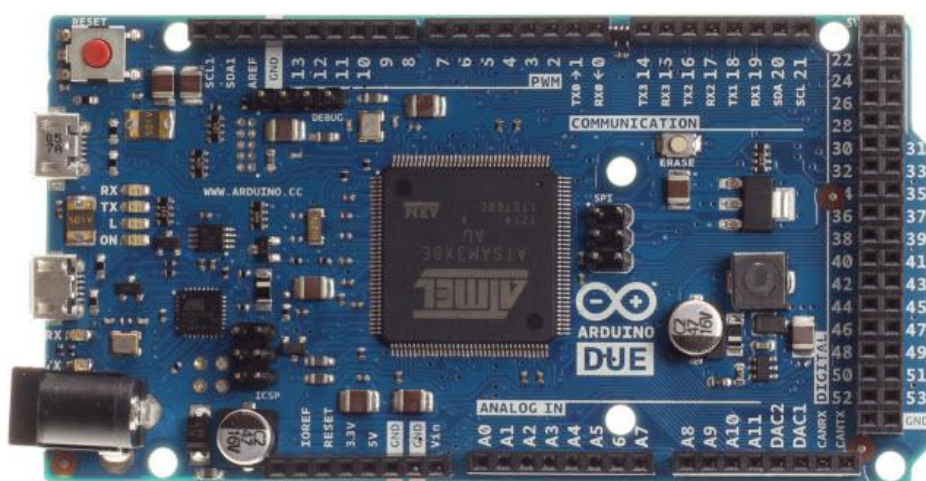
Arduino Due, MD070SD, CTE TFT LCD / SD card shield for Arduino DUE,  
SN65HVD230 CAN Board

### 4.1 Arduino DUE

Arduino DUE bylo zvoleno na základě vysoké rychlosti mikrokontroléru a přítomnosti nativních CAN portů. Díky těmto náležitostem je možné rychle zpracovávat obrazová data pro dotykový displej s rozlišením 800x480 pixelů.

Mikrokontrolér Arduino je open source zařízení zkonstruované za účelem jednoduchého rozhraní pro práci s mnoha různými senzory, jako uživatelskými vstupy. V závislosti na jejich chování vytváří výstup pro externí komponenty jako jsou LED, motory a reproduktory.

Arduino Due představuje vývojovou desku postavenou na SAM3X8E ARM Cortex M3. Zahrnuje 54 uživatelsky určených digitálních výstupů / vstupů, z nichž se 16 dá použít jako PWM výstup s nastavitelným rozlišením. Analogových vstupů obsahuje celkem 12 a to s 12-bitovým rozlišením. Dále obsahuje 4 UART a 2 DAC výstupy.[5]



Obrázek 12: Arduino DUE[4]

### 4.1.1 Přehled základních vlastností

Základní vlastnosti se dají shrnout v následujícím přehledu[4]

- Mikrokontrolér AT91SAM3X8E
- Pracovní napětí 3.3V
- Vstupní napětí (doporučené) 7-12V
- Vstupní napětí (limity) 6-20V
- Digitální I/O piny 54 (z nichž 16 poskytuje PWM výstup)
- Analogové vstupní piny 12
- Analogové výstupní piny 2 (DAC)
- Celkový DC výstupní proud na všech I/O řádcích 130 mA
- DC proud pro 3.3V pin 800 mA
- DC proud pro 5V pin teoreticky 1A, reálně 800mA
- Flash paměť 512 KB všechny k dispozici pro uživatelské aplikace
- SRAM 96 KB (64+32KB)
- DataFlash 2 Mbit (250 KB)
- Rychlost hodin 84 MHz

### 4.1.2 CANbus

Vestavěný mikrořadič AT91SAM3X8E obsahuje dva CANbus kanály. Bohužel se jedná jen o ovládací logiku. Proto je nutné dodat ještě budiče umožňující přenos CAN zpráv po sběrnici. Tyto budiče nejsou vestavěny v Arduino DUE vývojové desce ani nejsou vestavěné příkazy pro obsluhu CAN sběrnice v Arduino IDE.

Pro zprovoznění komunikace pro CAN sběrnici je nutné dodat následující

- Hardwarový budiče a fyzický konektor pro připojení systému do CAN sběrnice
- Knihovnu umožňující využívat CAN příkazy

**Tabulka 4: Piny CAN rozhraní**

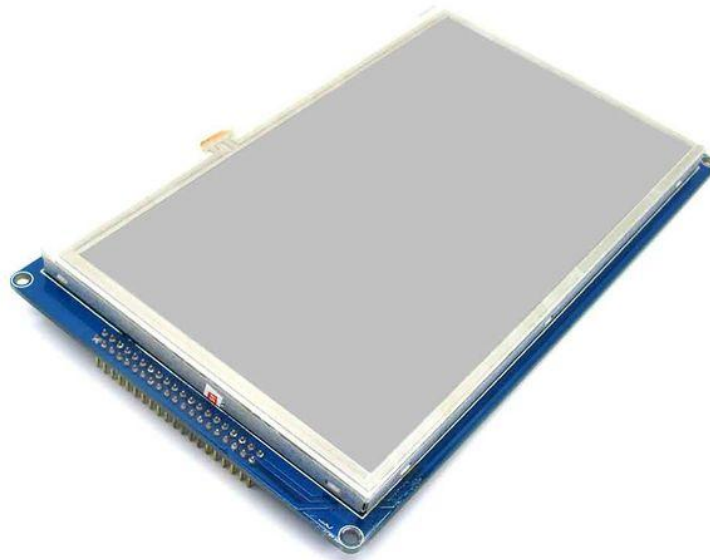
<b>Rozhraní</b>	<b>CAN Tx</b>	<b>CAN Rx</b>
CAN0	CANTX	CANRX
CAN1	53	DAC1

### **4.1.3 Vývojové prostředí Arduino IDE**

Vývojové prostředí Arduino IDE umožňuje rychlé psaní vlastních programů. Kompilátor i IDE je napsané v jazyce Java, díky čemuž je možné spustit na platformách MS Windows, GNU/Linux, Mac OS X. Vývojové prostředí obsahuje kompilátory pro kompletní řadu vývojových desek Arduino.[7]

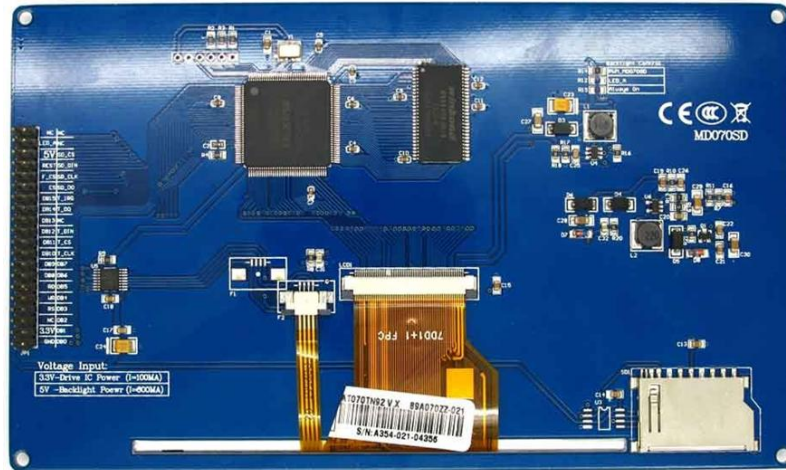
## 4.2 MD070SD

Jedná se o displejový modul se 7“ LCD o rozlišení 800x480 pixelů vybaveným dotykovou vrstvou.



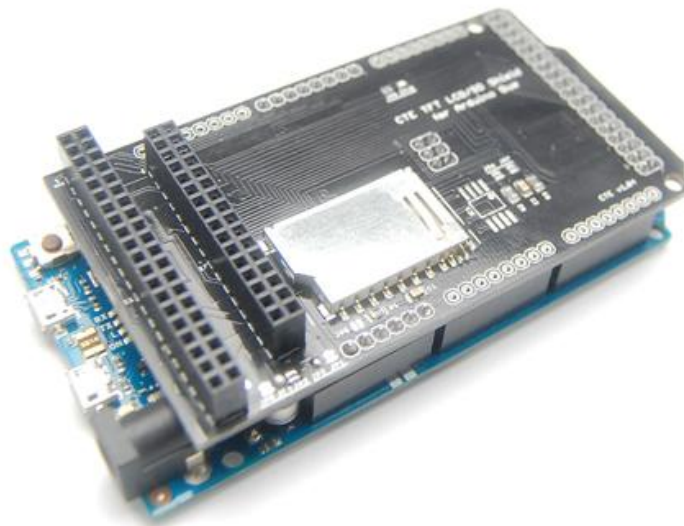
**Obrázek 13: Přední strana zobrazovacího modulu[8]**

MD070 využívá 8080 časovací sekvenci s 16 bitovým paralelním rozhraním, rozlišením 800x480, 16 milionů barev a integrovaný 8 stránkový video buffer. Pro obsluhu využívá CPLD čip a SDRAM paměť. Dotyková vrstva využívá kontrolér XPT2046. Jedná se o 4 vodičový dotykový kontrolér pracující na 12 bit 125 kHz vzorkování.[8]



**Obrázek 14: Zadní strana zobrazovacího modulu[8]**

Pro připojení k displeji je využít CTE TFT LCD/SD card shield for Arduino DUE, dále jen CTE shield. Tento shield umožňuje okamžité připojení displeje. Nevýhodou je, že tato redukce omezuje přístup k nevyužitým pinům a znesnadňuje další využití napájecích či zemíciho pinu, který nemají žádné další rozšíření. Tento nedostatek jsem pocítil při připojování CAN boardu, který je uveden v následující části.



**Obrázek 15: CTE TFT LCD/SD card shield for Arduino DUE[9]**

Problém s připojením jsem vyřešil použitím IDE 40 pin rozšiřujícího kabelu, kterým jsem propojil CTE shield a modul displeje.

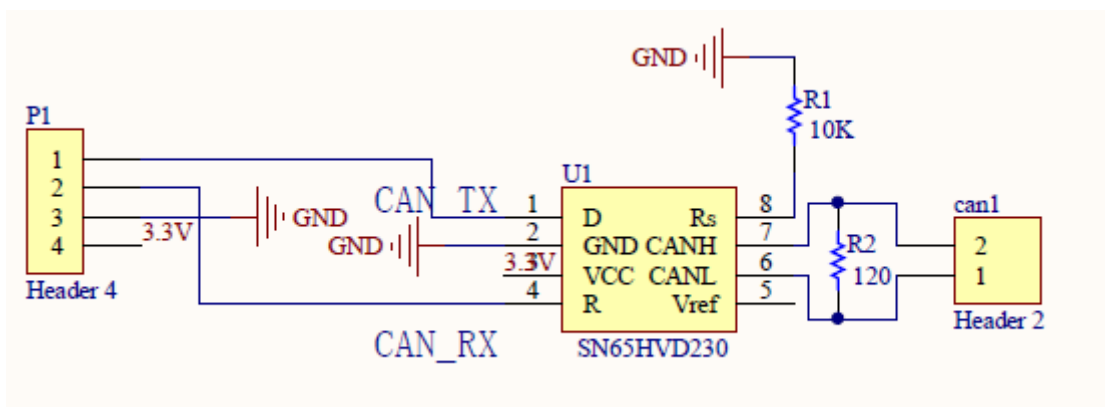
### 4.3 CAN Board SN65HVD230



Obrázek 16: CAN Board SN65HVD230 [10]

Ač Arduino Due má ve svém mikrokontroléru integrovaný kontrolér pro CAN sběrnici, tak pro komunikaci je ještě nutné mít budič sběrnice. Mezi tyto transceivery patří čipy MCP2550, MCP2552, MCP 2550, MCP2561, MCP2562 od výrobce Microchip. Případně je možné opatřit čipy SN65HVD231 nebo SN65HVD230 výrobce Texas Instrument. Já využil osazenou desku WaveShare SN65HVD230 CAN Board, využívající právě obvod TI SN65HVD230. [10]

Deska WaveShare obsahuje TI SN65HVD230, dva 120 k $\Omega$  rezistory, konektor pro připojení do 4-pinový konektor a svorkovnici pro připojení CAN hladin Low a High. Na následujícím obrázku je jednoduché schéma zapojení.



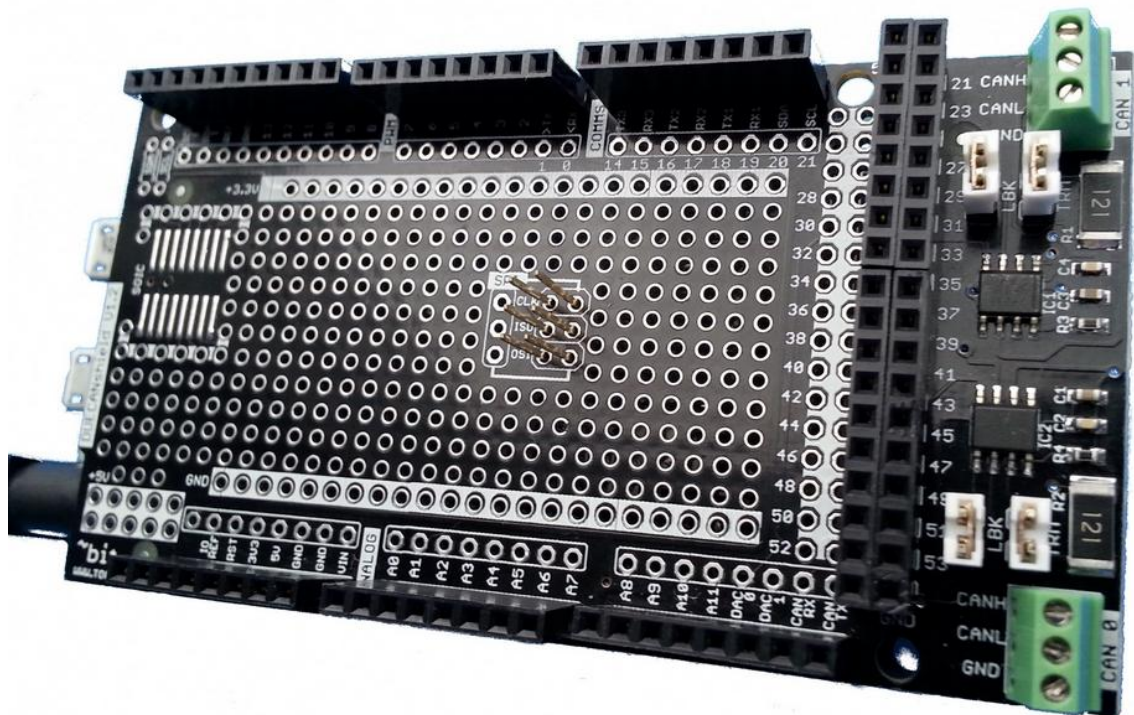
Obrázek 17: Schéma zapojení WaveShare SN65HVD230 CAN Board[10]

### Vlastnosti obvodu SN65HVD230

- 3.3 V napájení
- ESD ochrana překračující 16 kV HBM
- Vysoká vstupní impedance dovolující 120 připojených zařízení ke sběrnici
- Nízký proud v pohotovostním režimu typicky 370  $\mu\text{A}$
- Navrženo pro signály dosahující až 1 Mb/s
- Ochrana vypnutí při přehřátí
- Ochrana proti přepólování
- Ochrana pro připojení za chodu

### Alternativa

K vývojové desce Arduino DUE je dále dostupná rozšiřující deska Togglebit Arduino DUE CANshield. Tato deska je připravena pro připojení dvou nezávislých CAN bus rozhraní a je velmi dobře připravena pro rozšíření o další součásti. Nevýhodou je delší dodací doba daná pozicí distributora a vyšší cena \$75.[11]



Obrázek 18: Togglebit Arduino DUE CANshield[11]



## 4.4 Orion BMS

Orion BMS (Battery Management System) od firmy Ewert Energy Systems splňuje náročné požadavky na ochranu a řízení bateriových bloků pro elektrické vozy EV, plug-in hybridní vozy (PHEV) a hybridní vozy (HEV) v automotive kvalitě.

V následující části jsou popsány základní funkce BMS.[12]

### 4.4.1 Sledování parametrů

BMS monitoruje stavy mnoha položek které představují:

- **Napětí** – celkové napětí, napětí na jednotlivých člancích, minimální a maximální napětí článků
- **Teplota** – průměrná teplota, teplota na jednotlivých termistorech
- **Stav nabití (SOC) nebo hloubka vybití (DOD)** - indikující úroveň nabití baterie
- **Stav životnosti (SOH)** – různě definovaní měření celkové životnosti baterie
- **Proud** – vstupní a výstupní proud baterie

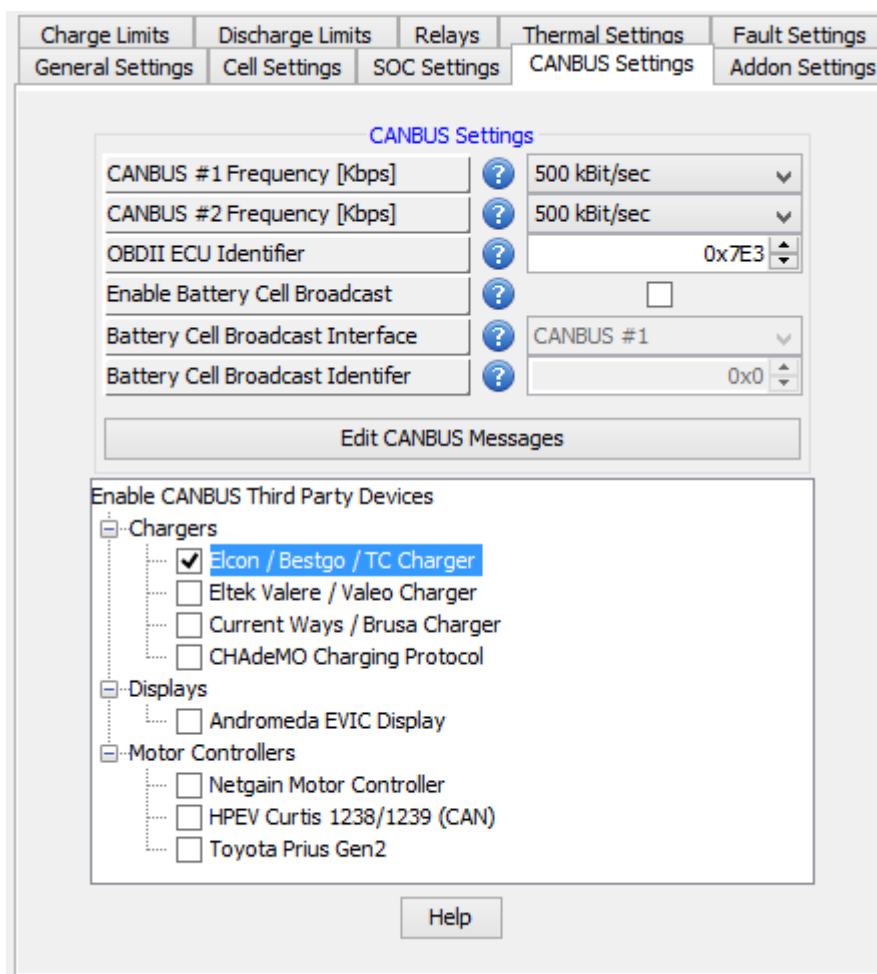
### 4.4.2 Vyhodnocování parametrů

Následně provede vyhodnocení těchto položek

- **Limit nabíjecího proudu (CCL)** – maximum nabíjecího proudu
- **Limit vybíjecího proudu (DCL)** – maximum vybíjecího proudu
- **Dodaná energie [kWh] během posledního nabíjení nebo nabíjecího cyklu**
- **Interní impedance článku**
- **Náboj [Ah] dodaný nebo uložený**
- **Celková dodaná energie od prvního použití**
- **Celkový počet pracovních hodin od prvního použití**
- **Celkový počet nabíjecích cyklů**

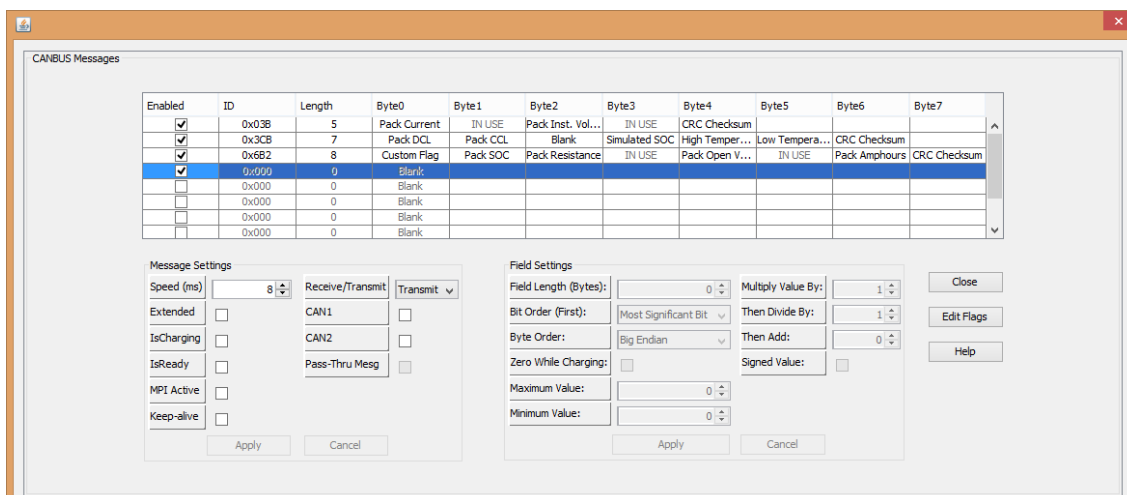
### 4.4.3 Komunikace

Orion BMS obsahuje dvě CANbus rozhraní, která lze nezávisle na sobě nakonfigurovat. Lze pro každé rozhraní nastavit rozdílnou rychlost přenosu, zároveň lze u každé zprávy nastavit periodu odesílání nebo čtení. Je možné nastavit i přeposílání zpráv z jednoho rozhraní na druhé – funkce Pass-Thru Mesg. Pomocí Enable Battery Cell broadcast lze jednou zprávou periodicky odesílat informaci o napětí všech bateriových článků, kde se napětí a identifikátory jednotlivých galvanických článků periodicky střídají.



Obrázek 19: Konfigurace CANbus Orion BMS

U Orion BMS je možné parametrizovat jednotlivé zprávy po CAN sběrnici. Celkem lze nastavit 10 různých zpráv až po 8 bytech.



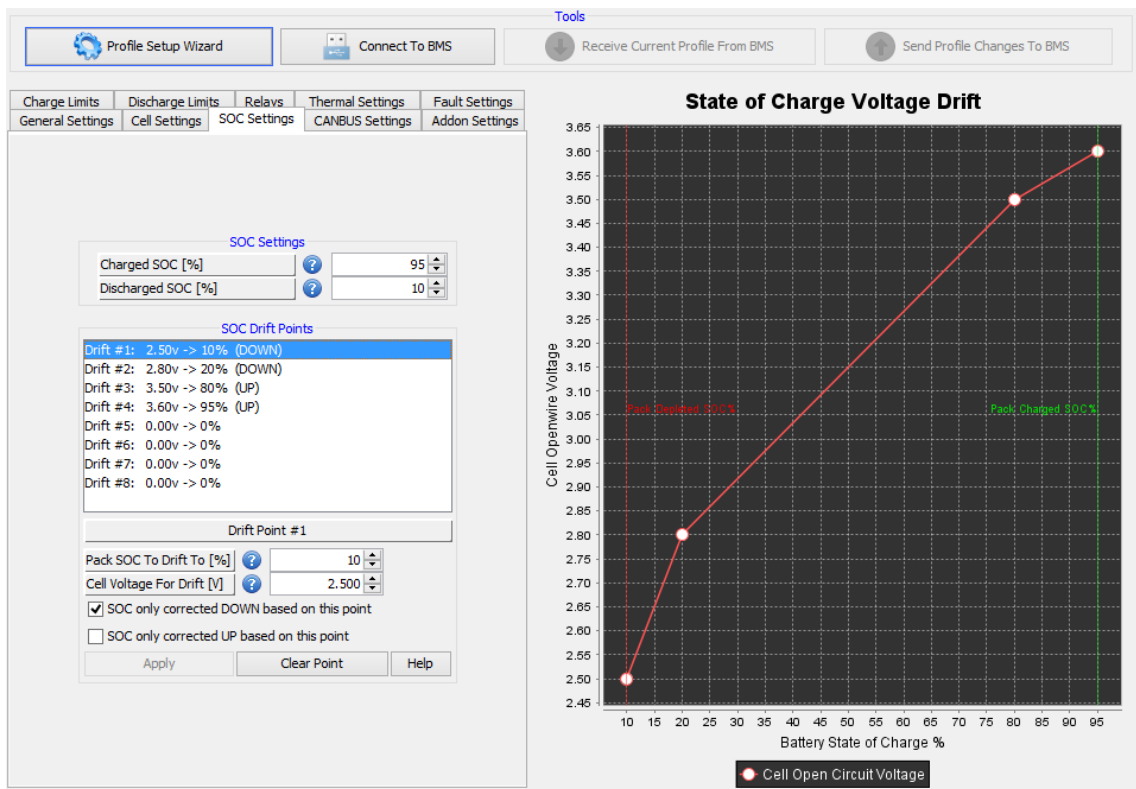
Obrázek 20: Editace zpráv Orion BMS

Níže uvádím kompletní přehled informací, které lze vložit do bytů odesílaných zpráv. Tento přehled jsem přidal z důvodu absence soupisu v nápovědě k Orion BMS.

### Přehled parametrů:

Blank, Constant Value, CRC Checksum, Custom Flag, Relay State, HEM Mode, Max Cell Number, Populated Cells, Pack DCL, Pack Current, Pack Inst. Voltage, Pack Open Voltage, Pack SOC, Pack Amphours, Pack Resistance, Pack DOD, Pack Health, Pack Summed Voltage, Pak ABS. Current (Unsigned), Maximum Pack Voltage, Minimum Pack Voltage, Total Pack Cycles, Current Limits Status, Pak CCL KW, Pack DCL KW, Maximum Pack DL, Maximum Pack CCL, Simulated SOC, Simulated Mode, Simulated Req. Mode, TP ADC, High Temperature, Low Thermistor ID, Low Cell Voltage, High Cell Voltage, Avg. Cell Voltage, Low Opencell Voltage, High OpenCell Voltage, Avg. Opencell Voltage, Low Cell Resistance, High Cell Voltage ID, Low Cell Voltage ID, High Opencell ID, Low Opencell ID, High Intres ID, Low Intres ID, 12v Supply, Current ADC1, Current ADC2, Fan Voltage, Isolation ADC, Shortest Wave, Isolation Clipping, IsolationThreshold

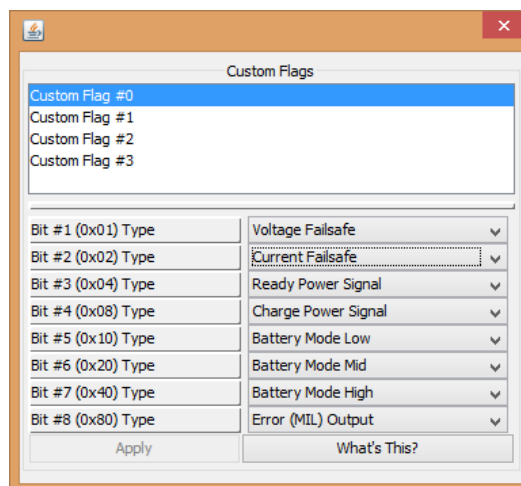
Z těchto informací je pro nás podstatný byte Pack Summed Voltage, udávající celkové napětí baterie. Dále je důležitý Pack SOC informující o stavu kapacity baterie v procentech



Obrázek 21: Konfigurace křivky stavu nabití

Stav nabití baterie se určuje v závislosti na napětí baterie. Vzhledem k nelineární charakteristice si lze závislost vytvořit v záložce SOC Settings.

Podstatný byte představuje Custom Flag. Je možné si vybrat jeden ze čtyř předvolených flag polí. Jedná se byte složený ze sekvence booleovských hodnot.



Obrázek 22: Konfigurace flag bytů Orion BMS

Následně uvádím kompletní přehled možných bytů pro flag zprávy.

### **Přehled flag bytů:**

Blank, Constant 1, Ready Power Signal, Charge Power Signal, Error (MIL) Output, Depleted, Mode Denied, Discharge relay, Charge Relay, Charge Safety, Charge Interlock, Battery Mode High, Battery Mode Mid, Battery Mode Low, Voltage Failsafe, Current Failsafe, 12v Supply Failsafe, 12v supply Failsafe, Multi-Purpose Input, Balancing Active, Multi-Purpose Output, Relay Failsafe, DTC: Internal Communication Fault, DTC: Internal Conversion Fault, DTC: Weak Cell Fault, DTC: Low Cell Voltage Fault, DTC: Open Cell Voltage Fault, DTC: Current Sensor Fault, DTC: Pack Voltage Sensor Fault, DTC: Voltage Redundancy Fault, DTC: Weak Pack Fault, DTC: Fan Monitor Fault, DTC: Thermistor Fault, DTC: Communication Fault, DTC: Always-On Supply Fault, DTC: High Voltage Isolation Fault, DTC: Charge-Enable Relay Fault, DTC: Discharge-Enable Relay Fault, DTC: Charger Safety Relay Fault, DTC: Internal Memory Fault, DTC: Internal Thermistor Fault, DTC: Internal Logic Fault

## **4.5 Nabíječka Elcon**

S touto nabíječkou jsem přímo nepřišel do styku, protože veškeré informace o stavech nabíjení je schopno předat Orion BMS přímo ze svých CAN zpráv. CAN komunikace probíhá v rychlosti 250 kb/s bez možnosti jiného nastavení.

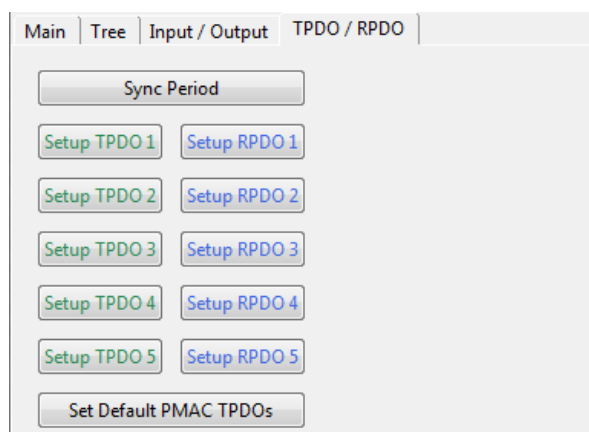
## **4.6 Sevcon GEN4**

Jedná se o frekvenční měnič kompaktních rozměrů pro asynchronní elektromotor. Jednotka nabízí komplexní funkce pro využití v elektromobilech.

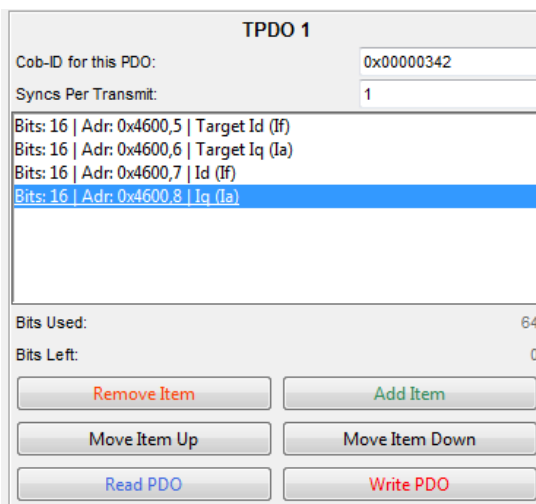
### **Základní vlastnosti jednotky[13]:**

- Pokročilé vektorové řízení asynchronního elektromotoru
- Automatická systémová diagnostika
- 13 konfigurovatelných digitálních vstupů
- 5 konfigurovatelných analogových výstupů
- 3 konfigurovatelné reléové výstupy
- Možnost rozšíření různými moduly

Pro konfiguraci frekvenčního měniče *GEN4* prostřednictvím PC je USB CAN převodník IXXAT USB to CAN Adapter a software od distributora měniče Sevcon DVT. Software Sevcon DVT pomocí CANopen protokolu umožňuje nastavit komplexní parametry daného zařízení. V GUI prostředí DVT Helper lze jednoduše nakonfigurovat až 5 odesílaných a 5 přijímaných CAN zpráv. Každá ze zpráv může využít až 64 bitů.



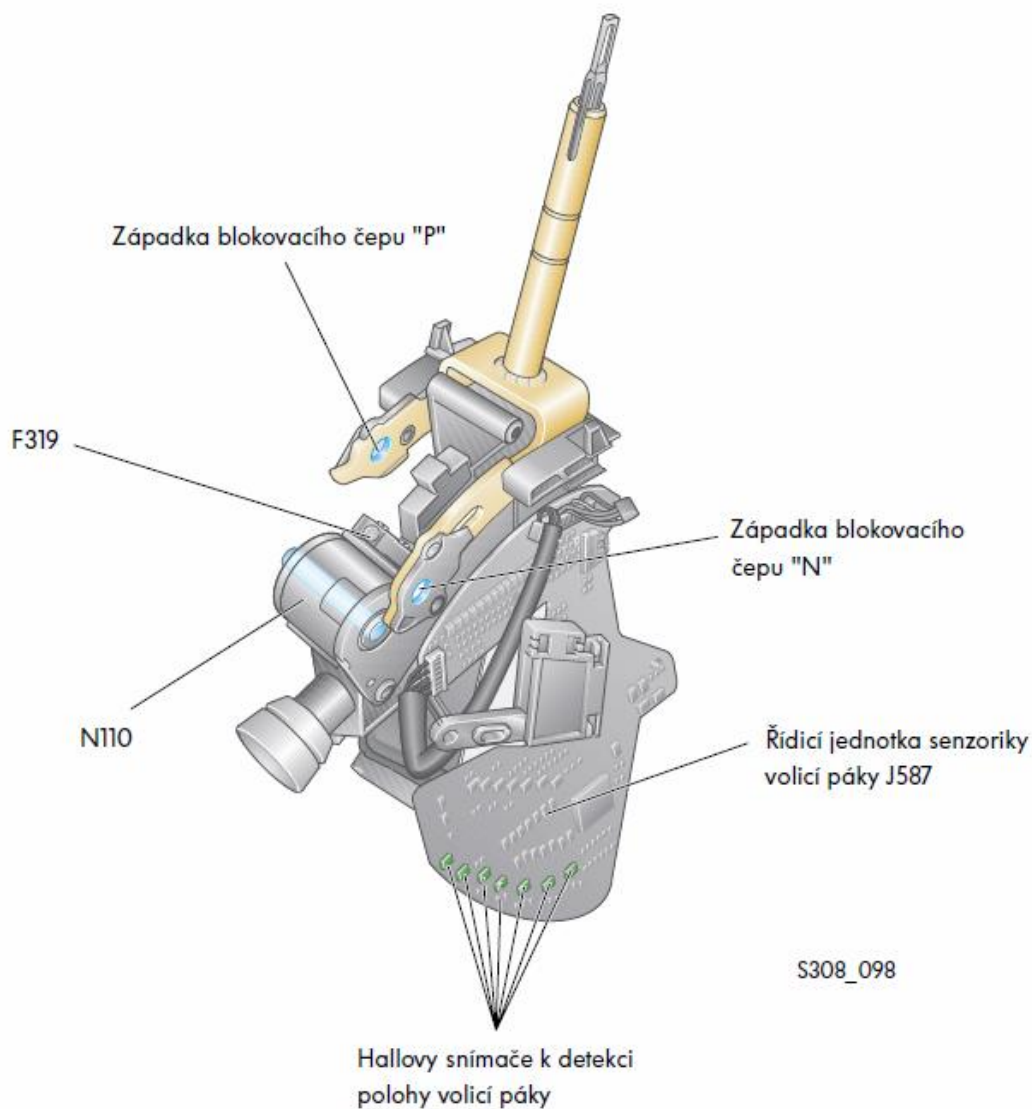
**Obrázek 23: Nastavení CAN zpráv Sevcon GEN4**



**Obrázek 24: Nastavení CAN zprávy Sevcon GEN4**

## 4.7 Volící páka převodovky DSG

Jedná se o volící páku určenou k převodovkám DSG koncernu VW. Řídící jednotka této volící páky komunikuje s převodovkou DSG po CAN sběrnici rychlostí 500 kb/s.



Obrázek 25: Volící páka DSG [14]

### 4.7.1 Měření ve Škoda Auto a.s.

V rámci diplomové práce jsme zavítali do Servisního tréninkového centra Škoda Auto a.s. Během návštěvy jsem sledoval datový tok CAN sběrnice vozu.

Tabulka 5: Část zachycené CAN komunikace ve voze Škoda Rapid [hex]

ID	Lenght	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
544	8	00	00	00	02	00	00	00	56
540	8	F0	00	FF	00	FF	00	12	0F
548	3	84	99	99	00	00	00	00	00
050	4	00	A0	94	34	00	00	00	00
280	8	01	00	00	00	00	00	2D	00
380	8	04	60	00	00	00	00	98	14
488	8	62	00	00	7C	FE	FF	FF	E0
5A0	8	7F	00	00	40	00	00	00	AB
3A0	8	0F	04	00	04	00	00	00	0F
4A0	8	00	00	00	00	00	00	00	00
448	5	83	00	00	19	70	00	00	00
2A0	8	00	00	00	A4	00	00	80	24
1A0	8	00	00	00	00	00	FE	00	18
0D0	6	97	60	05	0C	4E	B0	00	00
3D0	6	00	40	08	00	00	64	00	00
440	8	00	80	00	FE	81	00	00	00
0C2	8	0E	81	00	00	3E	61	00	0F
1AC	8	6E	04	00	00	69	01	00	02
67A	2	18	42	00	00	00	00	00	00
0D0	6	A6	50	04	0C	4E	B0	00	00
320	8	00	00	16	01	00	0B	00	80



#### 4.7.2 Nalezené stupně

Informace získané z CAN sběrnice posloužily pro nalezení ID zprávy a konkrétního bytu informujícího o zvoleném stavu volicí páky, která je použita u školního experimentálního elektromobilu.

Dále jsme na základě pozorování zjistili stavy ruční brzdy a brzdového pedálu.

**Tabulka 6: Zjištěné odesílané hodnoty [hex]**

Druh	ID	Stav	Num. Byte	Byte
Brzd. pedál	728	sešlápnutá	3	AC
		vyšlápnutá		2C
Ruční brzda	320	p. horní	0	02
		p. dolní		00
Volící páka	448	inicializace	0	08
		Inicializace1		8B
		P		83
		P s b. ped.		82
		R		72
		N		62
		N s b. ped.		63
		D		52
		S		C2
		+ -		E2
		+		A2
		-		B2

Logování zpráv probíhalo přes software Orion BMS. Ač je tento program určen primárně pro konfiguraci a diagnostiku BMS značky Orion, tak se díky záložce *Live CANBUS Traffic*, dá sledovat zprávy na sběrnici CAN a ukládat je souboru.

ID	Length	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Count	Timestamp
58C	8	90	E0	00	00	1A	00	00	02	4D	96
58F	8	00	F0	FF	7F	00	00	00	C2	45	482
5A0	8	7F	00	00	50	00	00	10	AB	80	20
5B7	8	00	C0	00	00	00	00	00	00	AF	96
5D0	8	80	32	59	AF	3F	5E	00	00	6F	94
5D2	8	00	00	00	00	00	54	4D	42	2E	192
5D6	5	00	00	00	00	00				4A	96
5DC	8	77	00	F0	01	10	18	00	48	6A	96
5E0	8	60	00	1A	00	00	00	84	00	70	96
5F3	8	00	00	7C	01	00	00	00	00	7A	482
60E	2	08	00							3E	966
676	2	12	82							3A	482
67A	2	18	42							82	482
6C8	2	13	02							AE	49
6DA	5	32	8E	0A	00	03				1	965
723	7	04	03	01	00	00	00	00		A8	192
728	7	04	01	01	2C	00	00	00		A	192
729	7	04	03	01	00	00	00	00		F8	193
7D0	8	01	07	00	30	F0	43	D5	04	3B	965

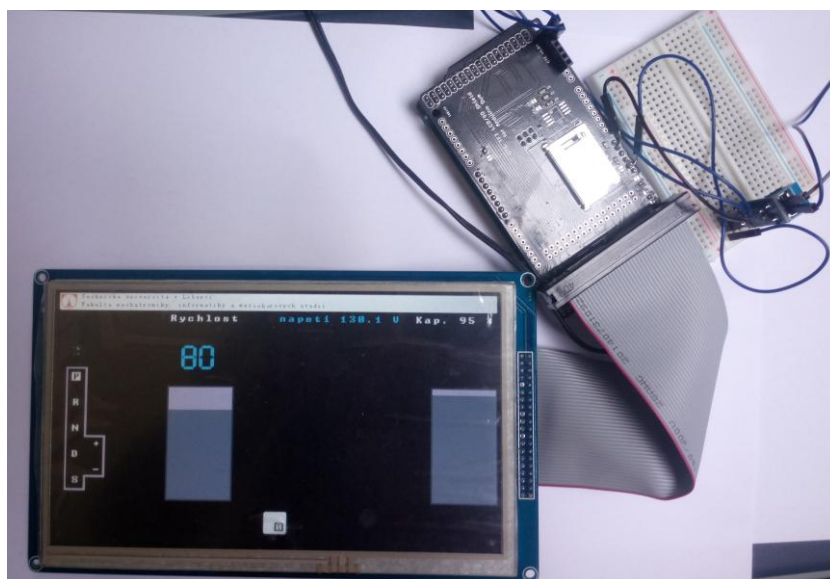
Obrázek 26: CAN komunikace

V okně je zobrazena délka zprávy, hodnoty jednotlivých bytů, počet přečtených zpráv s tímto ID a jejich časový odstup.

## 5 Realizace programu pro Arduino DUE

### 5.1 Zapojení

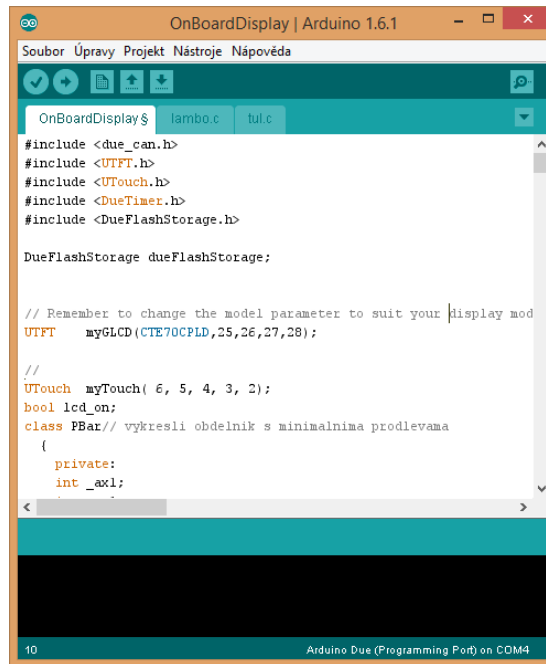
Při realizaci experimentálního zapojení jsem nejprve propojil vývojovou desku Arduino DUE s rozšiřující deskou CTE shield umožňující rychlé připojení displeje. K umožnění přístupu k dalším pinům vývojové desky, jsem displejový modul s CTE shield propojil 40 pinovým rozšiřujícím ATA kabelem. Dále jsem připojil konektor s prodlouženými a zapojil desku s CANbus budičem, kde stačilo propojit 3.3 V, GND, CANRX a CANTX s ekvivalentními piny na Arduino DUE. Vzhledem k vysokému proudovému odběru displeje, bylo nutné kromě napájení USB portem připojit zdroj napětí o výkonu 10 W.



Obrázek 27: Sestavená zobrazovací jednotka

## 5.2 Prvotní nastavení a použité knihovny

Po nainstalování vývojového prostředí Arduino IDE je třeba zvolit správný virtuální COM port a vývojovou desku. Následně se můžeme pustit do vytváření nového projektu.



Obrázek 28: Vývojové prostředí Arduino

Prvním krokem pro realizaci programu je přidání knihoven, které obsahují funkce, jež usnadní programování. K Arduino jsou knihovny dostupné v rámci rozsáhlé uživatelské komunity.

Implementace knihoven se provádí pomocí voleb v menu vývojového prostředí Arduina v následujícím sledu Projekt-Import knihovny- přidat knihovnu. Následně je možné zvolit adresář nebo zip soubor s danou knihovnou. Soubor knihovny se skládá ze souborů s příponami .c .cpp a .h. Soubory c a cpp obsahují vlastní kód knihovny a soubory h pak předpisy jednotlivých tříd či funkcí. Využití knihovny usnadňují dodávané příklady programů využívajících tyto knihovny nalezitelné v Soubor -> příklady. Použití knihovny se následně provádí pomocí přidáním tohoto řádku:

```
#include <jmeno_knihovny.h>
```

Pro tento projekt byly použity knihovny UTFT, UTouch, due\_can, DueTimer.

### 5.2.1 UTFT

Knihovna slouží pro práci s barevnými LCD displeji. Jednoduše umožní nakonfigurovat připojení k použitému displeji a vytvořit grafiku s možností vkládání fontů, obrázků, obdelníků, atd. Dále u některých modelů umožňuje vypínání a zapínání displeje. Podporovány jsou následující typy vývojových desek. Ke knihovně je kvalitní dokumentace a množství ukázkových příkladů.

Podporovány jsou následující typy vývojových desek

Arduino Duemilanove w/ATmega328, Arduino Uno, Arduino Leonardo, Arduino Mega, Arduino Mega2560, Arduino Due, Bobuino, Teensy 3.x.

Rozsáhlý seznam podporovaných displejů je dodán spolu s knihovnou. Pro zprovoznění displeje je třeba zapsat přesnou konfiguraci daného modulu. V našem případě se provede inicializace přidáním následujícího řádku.

```
UTFT myGLCD(CTE70CPLD,25,26,27,28);
```

Dalším krokem je aktivace CTE shieldu. Pro tuto aktivaci je třeba odkomentovat (odstranit po sobě jdoucí dvě lomítka) řádek obsahující

```
#define CTE_DUE_SHIELD 1
```

v souboru uloženém v adresáři knihovny pod cestou

hardware/arm/HW\_ARM\_defines.h.

Ke knihovně je k dispozici rozsáhlá příručka v souboru PDF. V této příručce jsou detailně rozepsány všechny vyskytující se metody umožňující rychlou práci s displejem.

### 5.2.2 UTouch

UTouch slouží pro zpracování dotykové vrstvy. Tuto knihovnu stejně jako předěšlou vytvořil Henning Karlsen. Jedná se o nadstavbu nad UTFT, avšak nevyžaduje její přítomnost. Knihovna nabízí opět podrobnou příručku. Utouch umožňuje snadno lokalizovat souřadnice X, Y na dotykové vrstvě displeje s různými hodnotami citlivostí. Samotné inicializace dotykového rozhraní, které je součástí našeho displejového modulu se provádí připsáním následujícího řádku `UTouch myTouch( 6, 5, 4, 3, 2);`

Součástí knihovny je kalibrační program *UTouch\_Calibration*. Po kompilaci daného programu se zobrazí výzva k postupné kalibraci na vyznačených místech. Pro přesnou kalibraci je vhodné použít stylus. Následně je třeba zobrazené hodnoty zapsat do souboru, který je uložen v adresáři knihovny *UtouchCD.h*. Pro náš případ se jedná o následující řádky.

```
#define CAL_X 0x03F38027UL
#define CAL_Y 0x003CCF35UL
#define CAL_S 0x8031F1DFUL
```

### 5.2.3 DueTimer

Knihovna pro vytvoření časovačů v prostředí Arduino Due. Vytvořený časovač omezuje počet překreslení za vteřinu. Nedochází tak ke zbytečnému zatěžování displejového modulu i mikrořadiče vývojové desky.

### 5.2.4 Due\_can

*Due\_can* obsluhuje odesílání/přijímání CAN zpráv prostřednictvím určených pinů Arduina DUE. Jiné knihovny umožňují použití externího CAN kontroléru připojeného k rozhraní SPI. Pomocí *due\_can* lze nakonfigurovat připojení do dvou nezávislých CAN sběrnic o různých rychlostech.

Pro nastavení čtení z CAN sběrnice je nejprve nutné provést inicializaci rozhraní, dále nastavit filtr čtených zpráv, tento kód se zapíše do inicializační metody *setup()*.

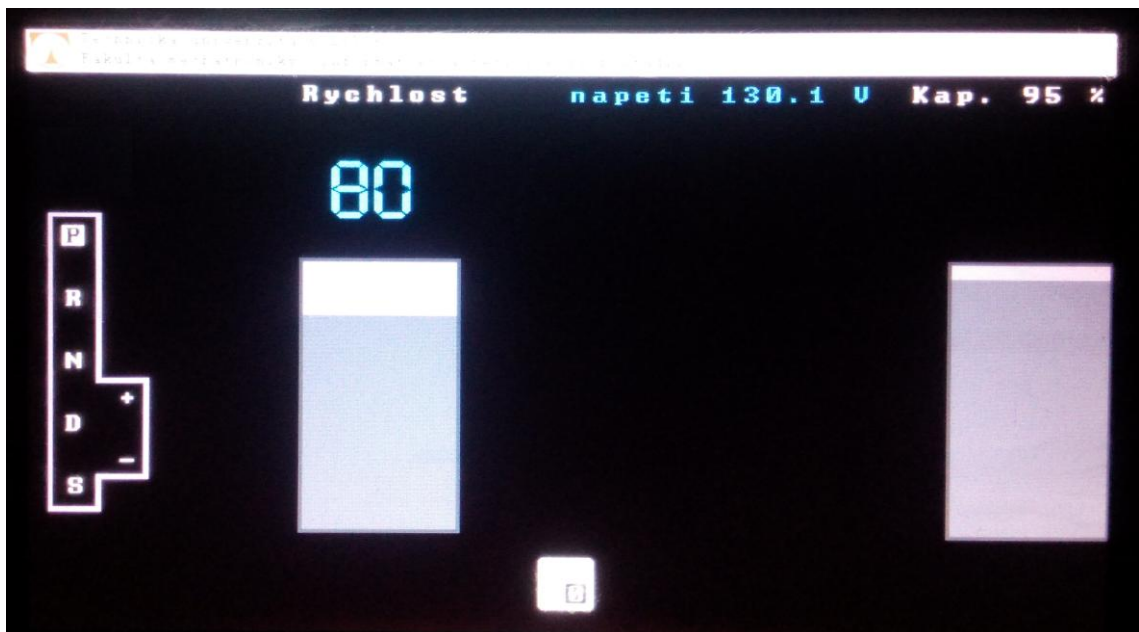
```
Can0.begin(CAN_BPS_500K);
int filter;
for (filter = 0; filter < 3; filter++) { //rozšířený formát
Can0.setRXFilter(filter, 0, 0, true);}
for (int filter = 3; filter < 7; filter++) { //standardní formát
Can0.setRXFilter(filter, 0, 0, false);}
```

Dalším krokem je samotné čtení vysílaných zpráv v metodě *loop()*. Nejprve proběhne inicializace příchozího rámce s datovým typem CAN\_FRAME. Následně za předpokladu, že je Can\_0 dostupný, dojde k předání příchozí zprávy vytvořené metodě.

```
CAN_FRAME incoming;
if (Can0.available() > 0) {
    Can0.read(incoming);
    zpracujFrame(incoming);
}
```

### 5.3 Grafické rozhraní

Nejprve bylo třeba vytvořit grafické rozhraní. Na obrázku je vidět ukazatel zvoleného stupně, dosažená rychlost, celkové napětí baterie a úroveň nabití baterie. Rychlost a úroveň nabití je navíc zobrazena ve formě sloupcového grafu. Dolní čtverec slouží jako virtuální tlačítko sloužící pro vypínání/zapínání LCD. V budoucnu může virtuální tlačítko fungovat např. pro nulování denních kilometrů.



Obrázek 29: Výstup zobrazovací jednotky

Při prvním spuštění se v inializační metodě *setup()* spustí metoda *vykrR()*. Tato metoda obsahuje metody pro vykreslení statických prvků. Zahrnuje vykreslení loga TUL v levé horní části, vypsání všech textů, vykreslení rámečků sloupových grafů a grafického znázornění stupňů volící páky.

Následně v metodě *loop()* nepřetržitě dochází k vyvolávání metod pro vykreslování a čtení rámců z CAN sběrnice. Vytvořené metody a třídy jsou popsány v následujících podkapitolách.

### 5.3.1 Třída Pbar

Tato třída slouží k vyvolání metod vykreslujících obdelníkový graf. V inicializačních parametrech lze nastavit počáteční souřadnice levého horního rohu, pravého dolního rohu, barvu grafu, barvu rámečku grafu a pozadí rámečku.

```
PBar( int ax1, int ay1, int ax2, int ay2, int cR, int cG, int cB, int rR, int rG, int rB,int pR,int pG,int pB);
```

Následně pomocí metody *vykreslit(int procenta)* zadáváme potřebné vyplnění grafu v procentech. Vykreslují se vždy pixely u kterých nastane změna, díky tomu nedochází ke zbytečnému překreslování a šetří se výpočetní výkon.

### 5.3.2 Třída Kvalt

Zvláštní důraz byl kladen na ukazatel zvoleného stupně volící páky. Pro tento účel byla vytvořena třída *Kvalt*. Tato třída obsahuje metody pro vykreslení zvoleného stavu volící páky bez zbytečného překreslování prvků, u kterých nedošlo ke změně.

### 5.3.3 Metoda zapis7SEG

Tato metoda vypisuje zadaná čísla s využitím písma sedmisegmentového displeje. První parameter určuje číslo, druhý a třetí souřadnice. Do čtvrtého parametru je nutné vložit maximální počet číslic. Pokud má číslo nižší počet míst než je maximální počet číslic, dojde k nahrazení číslic vykreslením prázdného obdelníku. Poslední tři parametry představují RGB složky barev písma.

```
zapis7SEG (int h,int p_x, int p_y, int pocet_z, int bR,int bG,int bB)
```



## 5.4 Zpracování CAN zpráv

CAN Komunikace probíhá rychlostí 500 kb/s. Během každého cyklu vykonávaných instrukcí dochází ke čtení příchozího rámce. Po uložení obsahu rámce je vyvolána metoda `zpracujFrame(CAN_FRAME &frame)`. Část kódu této metody uvádím na následující straně.

```
void zpracujFrame(CAN_FRAME &frame) {
    if(frame.id== 1096)//volicka
    {
        switch (frame.data.bytes[0]) {
            case 131: //P
                prevod = 1;
                break;
        }
    }
}
```

Z instrukcí je patrné, že nejprve se hodnota ID přijatého rámce porovná s ID určeného rámce volící páky. Poté, co se ID rámců shoduje, dojde k vyhodnocení žádaného bajtu.

Celkové napětí bateriového bloku je vyjádřeno jako zápis dvou po sobě jdoucích bajtů. Je tedy nutné oba bajty spojit a převést je na decimální číslo

U vyjádření SOC nebylo třeba dalšího převodu. Celý bajt je po převedení do decimálního tvaru možné přímo vypsát na displej.

Veškeré zobrazované stavy jsou vypsány v tabulce na následující straně. Pro lepší přehlednost jsou hodnoty zadány v decimálním tvaru. Samotná knihovna pracuje s decimálními hodnotami, je tedy nutný převod.

**Tabulka 7: Zpracované CAN bajty [hex]**

<b>Stav</b>	<b>ID</b>	<b>Num Byte.</b>	<b>Byte</b>
BMS celk. napětí	30	0+1	/
BMS SOC	6B2	1	/
VP P	448	0	83
VP R			72
VP N			62
VP D			52; E2
VP S			C2
VP +			A2
VP -			B2

## 6 Závěr

V rámci práce bylo úspěšně zprovozněna zobrazovací jednotka, které čerpá data ze čtyř rozdílných zařízení připojených k CAN sběrnici. Zobrazuje zvolenou polohu volící páky, aktuální stav nabití akumulátoru, napětí akumulátoru. Je připravena i pro čtení rychlosti, kde bude lepší provést odladění s konečnou konfigurací CAN zpráv frekvenčního měniče.

Setkal jsem se s mnoha problémy, souvisejícími zejména s dlouhou dobou dodání použitých komponent a nedostupnou dokumentací. Nicméně se zobrazovací jednotku podařilo v určeném čase zrealizovat a vyzkoušet na dostupných CAN zařízeních.

Vytvořený program obsahuje třídu pro generování sloupcového grafu vhodného pro zobrazení procentuálních stavů. Dále obsahuje třídu pro generování grafiky aktuálně zařazeného stupně.

Sepsané poznatky mohou přispět k budoucímu vytvoření desky plošných spojů a snadné integraci zobrazovací jednotky do systémů elektrického vozidla.

## Zdroje

- [1] BOSCH: CAN Specification Version 2.0, Stuttgart, 1991, 72 s.
- [2] CAN history. *CAN in Automation (CiA)* [online]. [2008] [cit. 2015-05-15]. Dostupné z: <http://www.can-cia.de/index.php?id=161>
- [3] Sběrnice CAN. 2003. *Elektrorevue* [online]. 2003 [cit. 2015-05-14]. Dostupné z: <http://www.elektrorevue.cz/clanky/03021/index.html>
- [4] Arduino Uno. *Arduino* [online]. [2013]. [cit. 2015-05-15]. Dostupné z: <http://www.arduino.cc/en/Main/ArduinoBoardUno>
- [5] Arduino Due. *Arduino* [online]. [2014] [cit. 2015-05-15]. Dostupné z: <http://www.arduino.cc/en/Main/arduinoBoardDue>
- [6] CO JE RASPBERRY PI?. *RASPI.cz* :: *Raspberry Pi* [online]. 2011 [cit. 2015-05-15]. Dostupné z: <http://www.raspi.cz/2011/12/co-je-raspberry-pi/>
- [7] Arduino IDE. *Arduino* [online]. [2015] [cit. 2015-05-15]. Dostupné z: <http://www.arduino.cc/en/Main/Software>
- [8] MD070SD 7" 16-BIT PARALLEL BUS TFT MODULE WITH TOUCH SCREEN. *IMall* [online]. [2015] [cit. 2015-05-22]. Dostupné z: <http://imall.iteadstudio.com/md070sd-7-16-bit-parallel-bus-tft-module.html>
- [9] CTE TFT LCD / SD card shield for Arduino DUE. *Ebay* [online]. 2015 [cit. 2015-05-15]. Dostupné z: <http://www.ebay.com/itm/TFT-SD-Shield-for-Arduino-DUE-TFT-LCD-Module-SD-Card-Adapter-2-8-3-2-inch-Mega-/121057447668>
- [10] SN65HVD230 CAN Board. *Waveshare Electronics* [online]. 2014 [cit. 2015-05-15]. Dostupné z: <http://www.wvshare.com/product/SN65HVD230-CAN-Board.htm>
- [11] Arduino DUE CANShield. *Togglebit* [online]. [cit. 2015-05-14]. [2014] Dostupné z: <http://togglebit.net/product/arduino-due-can-shield/>

- [12] Standard Features. *Orion BMS* [online]. [2014] [cit. 2015-05-15]. Dostupné z: <http://www.orionbms.com/features/basic-featuresshield/>
- [13] Gen4™. *Sevcon* [online]. 2012 [cit. 2015-05-15]. Dostupné z: <http://www.sevcon.com/ac-controllers/gen4™.aspx>
- [14] Samostudijní program 308 Převodovka s přímým řazením rychlostních stupňů 02E [PDF]. 2003 [cit. 2015-05-20]. Dostupné z: Samostudijní materiály Škoda Auto a.s.

# Obsah přiloženého CD-ROM

Na přiloženém CD-ROMu je uložen text této práce, naměřená data a veškeré zdrojové kódy včetně použitých knihoven.

Následuje seznam adresářové struktury CD-ROMu. Uvedeny jsou jen adresáře do první úrovně vnoření.

*„data/“*

Naměřená data na CAN sběrnici

*„dokumentace/“*

Diplomová práce v elektronické podobě.

*„knihovny/“*

Zdrojové kódy použitých knihoven

*„zdrojovy\_kod/“*

Zdrojový kód programu