

Univerzita Pardubice
Dopravní fakulta Jana Pernera

Implementace dotykového displeje do řídicích systémů experimentálního
kolejového vozidla
Petr Žižlavský

Bakalářská práce
2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Žižlavský**
Osobní číslo: **D15202**
Studijní program: **B3709 Dopravní technologie a spoje**
Studijní obor: **Elektrotechnické a elektronické systémy v dopravě**
Název tématu: **Implementace dotykového displeje do řídicích systémů experimentálního kolejového vozidla**
Zadávající 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 :

Cílem bakalářské práce bude naprogramovat dotykový displej a zabudovat jej do hlavního vozidlového rozvaděče. Displej bude sloužit pro zobrazování informací o stavu baterie. Informace budou získávány ze systému bateriového managementu.

Úkolem bakaláře bude:

- 1) Seznámit se s dotykovým displejem eMT3070B
- 2) Zabudovat displej do hlavního rozvaděče vozidla
- 3) Propojit displej pomocí sběrnice CAN s řídicí jednotkou systému bateriového managementu
- 4) Naprogramovat uživatelské rozhraní displeje
- 5) Zpracovat a zobrazit data z komunikační linky

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

Firemní literatura společnosti WEINTEK

VOBECKÝ, Jan a Vít ZÁHLAVA. Elektronika: součástky a obvody, principy a příklady. 3., rozš. vyd. Praha: Grada, 2005, 220 s. ISBN 80-247-1241-5.

DOSTÁL, Jiří. Operační zesilovače. 1. vyd. Praha: BEN - technická literatura, 2005, xxix, 504 s. ISBN 80-7300-049-0.

Vedoucí bakalářské práce:

Ing. Tomáš Lelek, Ph.D.


Katedra elektrotechniky, elektroniky a zabezpečovací
techniky v dopravě

Datum zadání bakalářské práce:

5. ledna 2018

Termín odevzdání bakalářské práce:

18. května 2018


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

L.S.


Ing. Dušan Čermák, Ph.D.
vedoucí katedry

V Pardubicích dne 25. dubna 2018

Prohlášení autora

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.

Tato bakalářská práce byla realizována s využitím technologií Výukového a výzkumného centra v dopravě.

V Pardubicích dne 18. 5. 2018

Petr Žižlavský

Poděkování

Chtěl bych poděkovat svému vedoucímu práce panu Ing. Tomáši Lelkovi, Ph. D za ochotu a asistenci při řešení problémů spojené s touto prací. Rád bych poděkoval i panu Ing. Petru Sýkorovi za rady při práci na experimentálním kolejovém vozidle. Velký dík patří také mé rodině, která mě po celou dobu mého studia podporovala.

Anotace

Tato bakalářská práce se zabývá naprogramováním a zabudováním dotykového displeje WEINTEK eMT3070B do experimentálního kolejového vozidla. V práci je nejprve krátce popsáno kolejové vozidlo, zejména akumulátor a řídicí jednotka bateriového systému. Řídicí jednotka bateriového systému komunikuje s dotykovým displejem po sběrnici CAN, tudíž další část se věnuje způsobu přenosu dat. Druhá polovina práce popisuje tvorbu programu, způsoby jeho testování a následně implementaci celého zařízení včetně nezbytných komponent do experimentálního vozidla.

Klíčová slova

HMI, dotykový displej, experimentální kolejové vozidlo, CAN bus

Title

Implementation of touch screen to control systems of experimental railway vehicle

Annotation

This bachelor thesis deals with programming and implementation the touch screen WEINTEK eMT3070B into the experimental railway vehicle. At first, thesis shortly describes railway vehicle, especially battery and battery management system. Battery management system communicates with the touch screen by CAN bus, therefore the next part is dedicated to way of data transfer. Second half of thesis describes developing a programme, ways of testing it and then implementation whole device with all components into the experimental vehicle.

Keywords

HMI, touch screen, experimental railway vehicle, CAN bus

Obsah

Úvod.....	9
1 Experimentální kolejové vozidlo.....	10
1.1 Battery Management System	10
2 HMI	12
2.1 Dotykový displej	12
3 Popis komunikace.....	15
3.1 Referenční model ISO/OSI	15
3.1.1 Fyzická vrstva	16
3.1.2 Linková vrstva	16
3.1.3 Síťová vrstva.....	16
3.1.4 Transportní vrstva	16
3.1.5 Relační vrstva	17
3.1.6 Prezentační vrstva	17
3.1.7 Aplikační vrstva.....	17
3.2 CAN bus.....	17
3.2.1 Struktura sběrnice	17
3.2.2 Priorita zpráv.....	19
3.2.3 Struktura zprávy.....	19
3.2.4 Aplikační protokoly	20
4 Software.....	22
4.1 Vývojové prostředí Easybuilder Pro	22
4.2 Tvorba programu.....	22
4.2.1 Základní ovládací prvky	22
4.2.2 Data sampling	24
4.2.3 Alarmy	25
4.2.4 Tagy a paměť	25

4.2.5	Makra	25
4.2.6	Systemové nastavení	28
4.2.7	Obrazovky	29
4.3	Nahrání programu	31
5	Testování programu	33
5.1	Testování pomocí tlačítek v programu	33
5.2	Testování pomocí CAN analyzátoru	34
6	Implementace displeje do experimentálního vozidla	36
6.1	Blokové schéma	36
6.2	BMS	36
6.3	Rozvaděč	37
6.4	Svorkovnice u displeje	38
6.5	Konektory displeje	39
7	Testování displeje na vozidle	40
	Závěr	42
	Použitá literatura	43
	Seznam obrázků	45
	Seznam tabulek	46
	Seznam příloh	47
	Přílohy	48

Úvod

Dopravní fakulta Jana Pernera vlastní experimentální kolejové vozidlo, které bylo postaveno pro výzkumné účely. Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě ve spolupráci se společností Škoda Electric, a. s. a s Výzkumným ústavem kolejových vozidel provedla na tomto vozidle výzkum řízení trakčních pohonů s nezávisle otočnými koly.

Vozidlo se stává i předmětem studentských prací a to znamená, že jako takové tedy stále prochází vývojem.

Cílem této práce je implementovat dotykový displej WEINTEK eMT3070b do tohoto vozidla a zobrazovat požadovaná data. Tím dojde k usnadnění a urychlení práce při testování vozidla, případně při budoucím výzkumu. Pomocí displeje se uživatel dozví nejen údaje o některých základních veličinách, ale i aktuální pracovní stav, případně typ poruchy. Displej je připojen na Battery Management System (BMS). Přenos dat probíhá po sběrnici CAN.

Zadání práce počítá s displejem pouze jako s výstupním zařízením, ovšem v budoucnosti je možné ho použít jako vstupní zařízení pro zadávání různých požadovaných hodnot. V tom případě by ale musel proběhnout hlubší zásah do systémů vozidla a přizpůsobit je na tento druh ovládání.

Displej je fyzicky zabudován v krabici, ke které vede kabel z konektoru v hlavním rozvaděči kolejového vozidla. Je tedy možné zařízení přesouvat po celém vozidle, a tím je dostupné i pro členy obsluhy, kteří se nepohybují v bezprostřední blízkosti rozvaděče. V případě, že displej není nutný, je možné ho jednoduše vypojit z konektoru. Pro realizaci bylo také potřeba najít vhodné umístění přenosových vodičů uvnitř vozidla.

1 Experimentální kolejové vozidlo

Jak již bylo uvedeno, vozidlo významně napomáhá při výzkumu na katedře KEEZ. Jedná se o sofistikované zařízení. Je konstruováno tak, aby sloužilo k simulaci tramvaje. Jde především o konstrukci podvozku a instalované motory. Vzhledem k nutnosti maximální nízkopodlažnosti tramvají totiž nelze použít klasické dvojkolí, a proto jsou ve výzkumném kolejovém vozidle použity motory ke každému kolu zvlášť. Jedná se o synchronní trakční motory s permanentními magnety. Výhodou těchto motorů je, že není nutné použití převodovky. Díky tomu dojde k ušetření místa a tramvaj může být plně nízkopodlažní. Nevýhodou jednoho motoru u každého kola je však vysoká složitost řízení jízdy. Vozidlo je realizováno pro rozchod 600 mm, jako testovací okruh se naskytlo využití mladějovské úzkorozchodné dráhy. [1]

Následně bylo potřeba zajistit zdroj energie. Trakční vedení nepřichází v úvahu, a proto jsou v podlaze vozidla zabudovány akumulátory. Nejprve se jednalo o olověné akumulátory, nyní už jsou nainstalovány moderní lithiové články, které jsou řízeny a regulovány vlastním systémem. Dále vozidlo dokáže rekuperovat energii, a tím šetřit přebytečnou energii. Rekuperace se používá při brzdění. Je možné použít i vřetenovou ruční brzdu, tato brzda se ovšem používá pouze při parkování nebo v nouzových situacích. [2]

Aby byla možnost provádět měření jak s nezávislými koly, tak i v konstrukci s pevným dvojkolím, je do vozidla zabudována pružná spojka. Tato spojka umožňuje spojení a rozpojení levého a pravého kola. Měření při obou způsobech lze tedy přímo porovnávat. Vozidlo dále obsahuje jednoduché vypružení, primární i sekundární. [3]

Optimalizace pohonu jednotlivých kol obstarává řídicí jednotka pohonu, která ovládá podřazené regulátory. Řídicí jednotku zastává kontrolér CompactRIO cRIO 9022 od firmy National Instruments. Jádrem tohoto systému je RT procesor 533 MHz. Dále obsahuje interní paměť 2 GB a operační paměť 256 MB DRAM. Zařízení je naprogramováno pomocí grafického vývojového prostředí LABView. Pro vstupy a výstupy je použito pět I/O modulů. Komunikace s okolím je možná buď přes dva konektory Ethernet nebo pomocí rozhraní RS-232. [3]

Pro tuto práci bude podstatný Battery Management System a umístění vodičů a konektoru v rozvaděči vozidla.

1.1 Battery Management System

Jak již bylo zmíněno, dříve byly ve vozidle olověné články, ovšem kvůli výzkumu byly odstraněny a nahrazeny novými modernějšími lithiovými články, které nabízejí mnohem vyšší

hustotu uložené energie, a tím lze dosáhnout i nižší hmotnosti. Stručné srovnání parametrů obou akumulátorů je možné vidět v následující tabulce.

Parametr	Varianta s Pb aku	Nová varianta s Li aku
Typ článku	EXIDE FT 12 150	Sinopoly SP-LFP100AHA
Jmenovité napětí [V]	12	3,2
Počet kusů v sérii [ks]	8	30
Celková energie [kWh]	14,4	9,6
Hmotnost jednoho článku [kg]	53	3,3
Celková hmotnost článků [kg]	424	99
Celková hmotnost včetně příslušenství [kg]	424	150
Hmotnostní hustota energie [Wh/kg]	34	64

Tabulka 1 Porovnání oloveného a lithiového akumulátoru [4]

Při bližším pohledu na tabulku je patrný markantní rozdíl hmotností, který svědčí ve prospěch lithiových článků. Na druhou stranu je celková energie, kterou je možné uložit v oloveném akumulátoru, mnohem vyšší než s články lithiovými. Nicméně hustota energie je stále mnohem příznivější u lithiových článků, a to přibližně dvojnásobně.

Uvnitř akumulátorového packu je nyní uloženo 30 lithiových článků typu LFP. Dohromady tyto články poskytují napětí 96 V a jsou schopny pojmout 9,6 kWh energie. Ve vozidle jsou také nainstalovány články NMC. Oba typy akumulátorových packů se tak mohou vzájemně střídat.

Lze shrnout, že lithiové články mají tedy lepší parametry, ovšem oproti oloveným jsou náročnější na řídicí elektroniku. K tomuto řízení a kontrole celého akumulátoru proto slouží Battery Management System (dále také „BMS“). BMS se tak stává podstatnou částí celého zdrojového systému. Strukturu BMS tvoří hlavní jednotka, kterou můžeme nazvat MASTER, a balancovací obvody pro jednotlivé články. U každého bateriového článku tedy musí být jeden balancér, který články měří a případně vyrovnává, tedy balancuje, jak je patrné z jeho názvu, napětí mezi jednotlivými články. Údaje z balancérů se následně shromažďují v hlavní jednotce, která dává pokyny, jakým způsobem se balancéry mají chovat například při nabíjení. [4]

2 HMI

Dotykový displej, který bude implementován do experimentálního vozítka, spadá do zařízení HMI, v tomto případě stejně jako notebook, ze kterého je vozidlo ovládáno a řízeno. Tato zkratka znamená Human-Machine-Interface a lze ji přeložit jako rozhraní mezi člověkem a strojem. Jedná se tedy o zařízení, které zprostředkovává komunikaci mezi obsluhující osobou a řídicí jednotkou ovládaného systému. Řídicí jednotka tak přes toto rozhraní dostává povely, případně přebírá data, a naopak relevantní informace posílá zpět uživateli. Typickým příkladem grafického zařízení je právě dotykový displej, nicméně to může být i jiná vstupně/výstupní periférie.

Grafické HMI dotykové panely se používají především v oblasti průmyslu a automatizace, kde mají za úkol zajistit co nejsnazší, nejefektivnější a uživatelsky přívětivou komunikaci stroje s operátorem. Obsluha může rychle a jednoduše zadávat povely a požadované parametry, a současně sledovat stav stroje.

2.1 Dotykový displej

Použitý dotykový displej WEINTEK eMT3070b vyrábí firma WEINTEK Labs, která se specializuje na grafické operátorské panely a průmyslové počítače. Řada eMT představuje moderní panely nové generace. Aby nedošlo k poškození při provozu v průmyslových podmínkách, je zařízení uloženo v celokovovém pouzdře. Cena těchto displejů se pohybuje v řádech desetitisíců korun, v závislosti na konkrétním modelu. [5]

Model eMT3070b má sedmipalcový, LED technologií podsvícený, dotykový displej LCD typu TFT (Thin-Film Transistors). Rozlišení je pro nenáročné průmyslové použití dostatečné, a to 800x480 pixelů. V popisu produktu dále výrobce uvádí galvanické oddělení portů. Rozsah tolerovaných provozních teplot je pak od -20 °C po +50 °C, což splňuje standardní průmyslové podmínky. Jádrem zařízení je procesor ARTEX A8 RISC. [5]

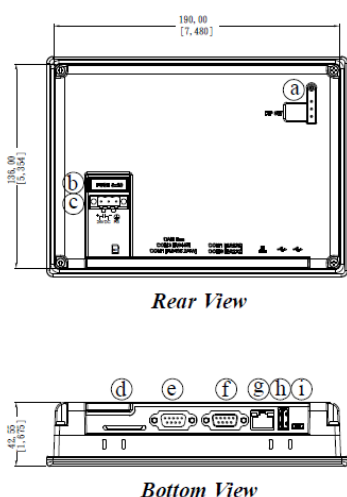
Displej podporuje připojení několika různých typů sběrnic. Na zadní straně najdeme mimo jiné konektor s devíti piny, který bude sloužit k propojení s bateriovým systémem ve výzkumném vozidle. Je to konektor (označen písmenem „e“, samec) pro připojení sériové linky RS-485 a sběrnice CAN. Dále se musí k připojení použít správné piny tohoto konektoru. Přehledné rozmístění pinů zobrazuje tabulka na následující straně čerpající z datasheetu přístroje.

CAN Bus (konektor „e“)	
Číslo pinu	Význam
1	
2	
3	
4	
5	GND
6	
7	CAN_L
8	CAN_H
9	

Tabulka 2 Zapojení pinů v konektorech displeje [6]

Jak vyplývá z tabulky 2, podstatnými piny konektoru pro připojení CAN sběrnice jsou sedm a osm. Samozřejmě je nutné připojit i GND (zemní vodič), a to na pin pět.

Na další port displeje můžeme připojit sběrnici Ethernet. Lze také připojit USB, a to jak klasické USB 2.0, tak i Micro-USB. Na displeji je možné ještě nalézt slot pro paměťovou kartu SD/SDHC. Dále je zabudována i pojistka, kterou lze v případě potřeby jednoduše vyměnit bez většího zásahu do zařízení. Všechny porty displeje jsou zobrazeny na obrázku 1.



a	DIP SW	f	COM1 RS-232, COM3 RS-232
b	Fuse	g	Ethernet Port
c	Power Connector	h	USB Host Port
d	SD Card Slot	i	USB Client Port
e	COM1 RS-485, COM3 RS-485, CAN Bus		

Obrázek 1 Rozmístění konektorů na displeji [6]

Zařízení je napájeno stejnosměrným napětím 24 V. Třípinový konektor pro napájení je umístěn pod pouzdem pojistky na zadní straně displeje. Výrobce připouští odchylku 20 % od této hodnoty. Při jmenovitém napětí by měl displej odebírat maximálně 400 mA.

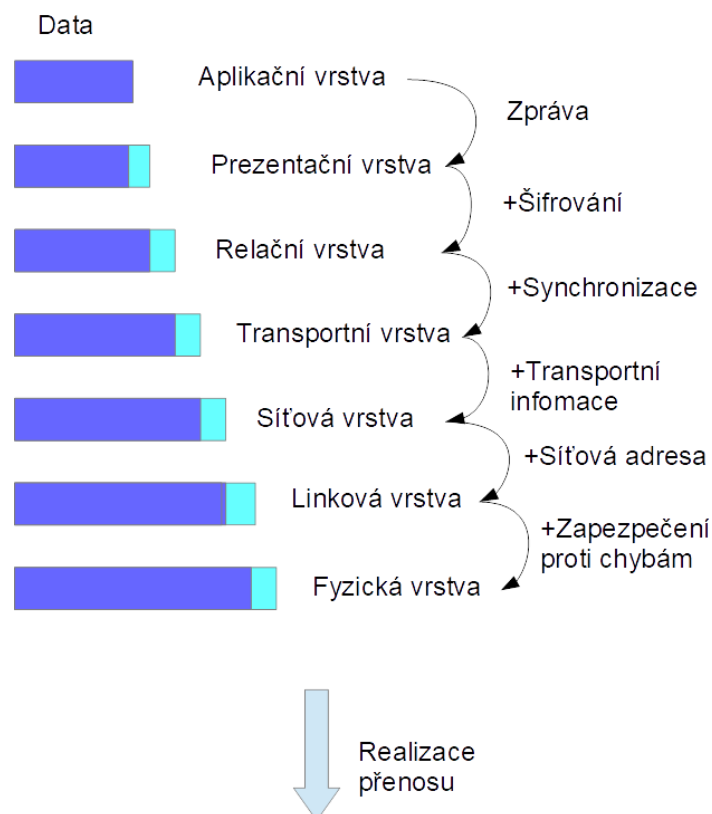
Váha činí necelý kilogram. Rozměry, které je nutné vzít v potaz při montování do přenosné krabice, jsou 200,3x146,3x42,5 mm. Hloubka krabice musí být taková, aby se do ní displej vešel a zároveň zbylo ještě dostatek místa pro svorkovnici.

3 Popis komunikace

Jak už bylo uvedeno výše, displej bude komunikovat po sběrnici CAN. Elektrické parametry sběrnice jsou standardizovány svou vlastní normou ISO 11898. Norma definuje, jakým způsobem probíhá přenos informací, a také zajišťují, aby přenos dat byl sjednocen dle daných pravidel. Částečně pro tuto sběrnici můžeme uplatnit referenční model ISO/OSI, který je popsán níže.

3.1 Referenční model ISO/OSI

Jedná se o bazální doporučení pro komunikaci a přenos dat mezi dvěma entitami. [7] Model byl vypracován Mezinárodní organizací pro normalizaci, anglický název International Standards Organization (ISO). Na zmíněném modelu je založeno mnoho protokolů a standardů. Zkratka OSI (Open Systems Interconnection) pak označuje, že jde o realizaci propojení dvou, případně i více, otevřených systémů. Celý přenos dat lze dle tohoto referenčního modelu rozdělit na několik vrstev (zpravidla se udává sedm), přičemž každá vrstva má jiný účel. Protokoly jednotlivých vrstev nejsou pevně definovány, jelikož by bylo velice obtížné, vzhledem k jejich velkému množství, zahrnout je pouze do jedné normy. [8] Posloupnost vrstev zobrazuje obrázek 2.



Obrázek 2 Princip referenčního modelu ISO/OSI

Jednotlivé vrstvy modelu si mezi sebou předávají informace a každá vrstva k nim připojí svá vlastní důležitá data pro celkový přenos. Ve výsledku tak bude fyzická vrstva pracovat s nejvíce daty. Všechna musí nejprve odeslat na přenosový kanál a přijímač následně zpracuje data od fyzické vrstvy až po aplikační.

3.1.1 Fyzická vrstva

Fyzickou vrstvou se rozumí nejnižší přenosová vrstva. Realizuje fyzický přenos dat. Určuje typ propojení, zda se jedná o optický kabel, typ metalického vedení, případně bezdrátové spojení atd. Dále fyzická vrstva definuje napěťové úrovně a počty využívaných vodičů, které musí být zapojeny. Kromě toho určuje i modulaci signálu a strukturu sítě. [7] Fyzická vrstva spouští a ukončuje fyzické spojení. Kvalitu fyzické vrstvy může značit například chybovost. [8]

3.1.2 Linková vrstva

Linková vrstva poskytuje předávání dat mezi dvěma body. Můžeme ji rozdělit na dvě podvrstvy a to MAC (Medium Access Control) a LLC (Logical Link Control). [7] Linková vrstva informaci rozdělí do rámců (frames). Do těchto rámců pak kromě dat přidává ještě další parametry jako záhlaví a adresaci, eventuálně zabezpečení proti chybám.

3.1.3 Síťová vrstva

Hlavním úkolem síťové vrstvy je správné nasměrování dat. Tomuto pomáhá síťová adresace, se kterou vrstva pracuje. [8] Tato vrstva se uplatňuje především v případě, kdy máme do síťové struktury připojena více než dvě zařízení. V takových sítích totiž vznikají uzly, ze kterých je nutné data správně rozesílat.

V souvislosti se síťovou vrstvou rozlišujeme dva typy přepojování:

- Přepojování okruhů (circuit switching) – zde dochází k přímému spojování dvou uzlů. Jednotlivé uzly tedy postupně vytvoří komunikační kanál. Komunikace mezi jinými účastníky není možná. [7]
- Přepojování paketů (packet switching) – data jsou posílána do uzlů ve formě paketů. Každý paket obsahuje cílovou adresu a na základě toho probíhá rozřazení uvnitř uzlu. [7]

3.1.4 Transportní vrstva

Další vrstvou v pořadí je transportní. Tato vrstva potvrzuje zdařilý příjem dat, detekuje eventuální chyby a určuje, zda je přenosový kanál volný k dalšímu přenosu. Mezi její úkoly

patří i správné sestavení dat. Je totiž časté, že data v podobě paketů nepřijdou ve stejné sekvenci jako při odesílání, a to vlivem rozdělení v uzlech, které vždy vybírají nejlepší variantu přenosové dráhy vzhledem k zatížení sítě. Transportní vrstva tvoří hranici mezi vyššími vrstvami, které se pojí s aplikací, a nižšími vrstvami, které se zabývají přenosem bitů. [8]

3.1.5 Relační vrstva

Relační vrstva obecně udržuje kontakt mezi dvěma entitami. Vytváří, řídí a ukončuje relace mezi uzly, synchronizuje relační spojení. Dále může například pozdržet přenos zpráv. [8]

3.1.6 Prezentační vrstva

Prezentační vrstva pracuje především se strukturou zpráv. Zajišťuje převod datového formátu, například překódování textů. [7]

3.1.7 Aplikační vrstva

Aplikační vrstva je nejvyšší vrstvou referenčního modelu a jejím úkolem je umožnit komunikaci mezi dvěma aplikacemi na různých zařízeních. Na úrovni aplikační vrstvy je realizováno mnoho různých typů protokolů (například FTP v architektuře TCP/IP). [8]

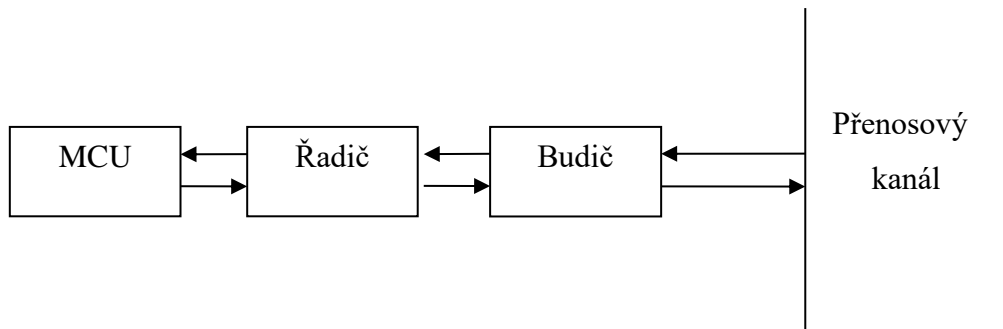
3.2 CAN bus

Sběrnice CAN byla vytvořena za účelem jednoduché komunikace uvnitř dopravních prostředků, především osobních automobilů. Výhoda této sběrnice spočívá v úspoře vodičů, efektivnosti a v poměrně slušném zabezpečení proti vzniku chyb. Úspora zejména proto, že dříve se pro přenos dat používalo přímé propojení mezi komunikujícími entitami. To znamenalo mnoho fyzických propojení. CAN bus spojuje každý objekt pouze jednou cestou, ale současně žádným způsobem neomezuje spojení dvou libovolných bodů. Další výhodou je tzv. priorita (přednost) zpráv, která řeší případnou kolizi více účastníků na sběrnici v jeden moment. Postupem času tak byla zaváděna do silničních vozidel a nyní tvoří jejich důležitou součást. [9]

3.2.1 Struktura sběrnice

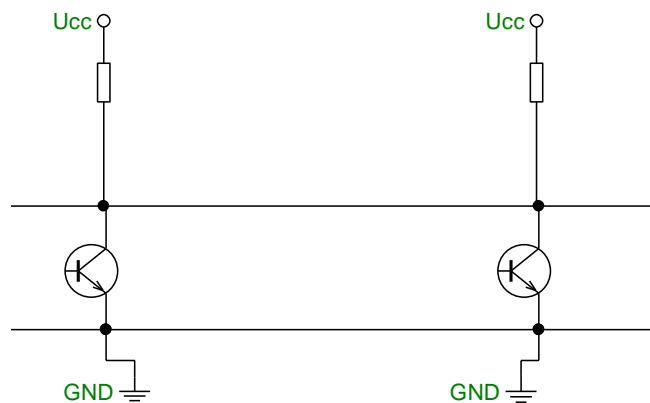
U CAN sběrnice nenalezneme všechny vrstvy popsané výše v referenčním modelu ISO/OSI. Podstatnými jsou fyzická, linková a aplikační vrstva, která určuje různé aplikační protokoly. [7] První částí při přenosu je přijímač. Většinou se jedná o libovolnou řídicí jednotku, která obsahuje mikroprocesor. Ten rozhoduje, jaká data se budou posílat.

Za mikroprocesorem nalezneme řadič, který zpracovává obdržená data. Hlavním úkolem řadiče je upravit tato data a poslat je dále do budiče, jehož funkcí je správná aktivace elektrického signálu na sběrnici tak, aby přenos proběhl v pořádku a dle stanovených pravidel. Po sběrnici následně proběhnou data, která jsou na straně přijímače vyhodnocována opět nejprve budičem, který zpracuje elektrický signál, dále řadičem, až se postupně data doručí ve správném formátu k příslušné jednotce, respektive mikroprocesoru.



Obrázek 3 Blokové schéma CAN komunikace [9]

Přenos signálu na sběrnici ovládá budič. Způsob aktivování sběrnice je následující. Každý připojený uzel na CAN je zastoupen tranzistorem.



Obrázek 4 Realizace sběrnice CAN [9]

Pokud dojde k sepnutí tranzistoru, tak bude napájení U_{cc} propojeno přes rezistor na zem. Na sběrnici se objeví logická nula, která reprezentuje dominantní stav. Z obrázku 4 vyplývá, že k tomuto postačuje sepnutí pouze jednoho tranzistoru, proto mluvíme o tzv. dominantním stavu. Naopak pokud není sepnutý ani jeden tranzistor, tak je na sběrnici pasivní stav, který představuje logickou jedničku. Logickou úroveň na sběrnici určujeme z difference napětí mezi dvěma vodiči. Ty se nazývají CAN_L a CAN_H. Různé normy mohou mít odlišně definované přesné napěťové úrovně, nicméně platí, že při aktivním stavu musí být rozdíl napětí mezi vodiči v rozmezí 2–5 V a při pasivním stavu musí být tento rozdíl nulový s minimálními odchylkami. [9] Dále musí být na každém konci vedení zakončovací rezistor. Tento odpor je nutný, abychom

odstranili zpětné odrazy na konci vedení, které by způsobily nežádoucí rušení právě vysílaného signálu. Zmíněný odpor, pro rychlou sběrnici CAN, obvykle činí 120 Ω.

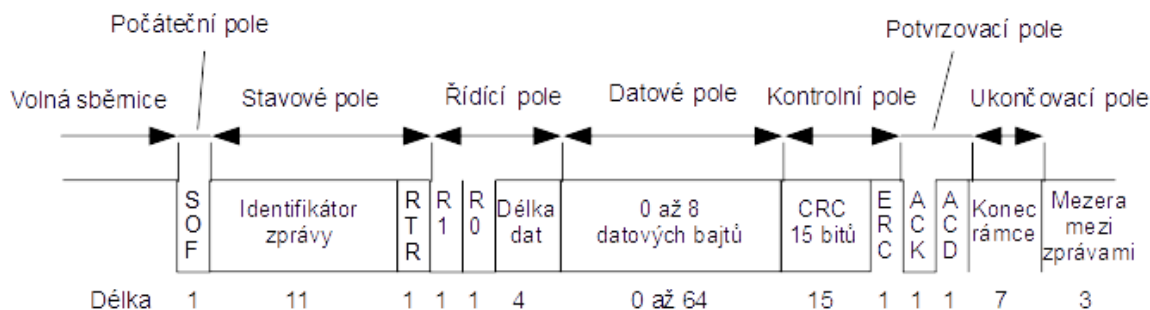
3.2.2 Priorita zpráv

Sběrnice CAN je sériová, a tedy nelze vysílat více dat v jednom okamžiku. Existuje pouze jeden přenosový kanál. Abychom zajistili, že důležité informace budou přenášeny před těmi méně důležitými, je nutné zavést prioritu zpráv. Každá zpráva je určena identifikátorem. Priorita zpráv je stanovena dle konkrétního identifikátoru. Čím nižší je hodnota identifikátoru, tím vyšší je priorita přenášené informace. To je způsobeno tím, že stav logické nuly je dominantní stav. Pokud tedy vysílají dva uzly současně, každý s jiným aktuálním bitem identifikátoru, tak se na sběrnici objeví dominantní stav. Vysílač, který v daný okamžik na sběrnici posílal identifikátor s pasivním stavem, zjistí, že dochází k vysílání dat s vyšší prioritou a svou snahu o přenos zprávy ukončí, případně pouze pozastaví. [7]

Pokud nedojde k tomuto koliznímu stavu a o přenos dat se snaží pouze jeden uzel a pokud proběhne přenos identifikátoru správně, dojde i k přenosu zprávy, aniž by ostatní uzly, přestože mají vyšší prioritu, mohly poté do přenosu zasahovat. Vždy se tedy přenesou celá zpráva, a až poté je sběrnice opět volná.

3.2.3 Struktura zprávy

Zpráva jako celek obsahuje různé části, které, jak již bylo zmíněno, přidávají vyšší přenosové vrstvy. Zprávu dělíme na rámce. Rámce pak dále dělíme na pole. Nejdůležitější část rámce je úsek datového pole, který obsahuje užitečné informace pro příjemce. Za účelem přenosu těchto dat je ovšem nutné přenést i další data, která můžou například identifikovat příjemce nebo detekovat chybu v přenosu. Rámec se tedy skládá z více polí. Pole a jejich obsah jsou zobrazeny na obrázku 5.



Obrázek 5 Přenosový rámec CAN 2.0A [9]

První bit rámce SOF (Start of Frame) je dominantní a značí začátek přenosu. Dále dochází k přenosu stavového pole. Patří sem identifikátor a RTR (Remote Request) [7]. Identifikátor je ve standardu CAN 2.0A 11bitový a ve standardu CAN 2.0B 29bitový. Z toho plyne, že na sběrnici CAN 2.0B je větší možnost priority, než u CAN 2.0A. RTR informace se skládá pouze z jediného bitu. Tento bit udává, zda se jedná o zprávu či o žádost o zprávu.

Následuje řídicí pole. To obsahuje rezervní bity a informaci o délce dat. Po řídicím poli jsou přenášeny vlastní datové bajty. Velikost dat je od nuly do osmi bajtů.

Jak bylo zmíněno, kód přenášený v rámci dokáže detekovat chybu v datových bajtech. Tato chyba může být způsobena vnějším okolím, např. rušením. Detekující kód neumí odhalit chyby způsobené záměrně hackerem. Ve většině případů se ovšem CAN sběrnice používá v uzavřených systémech, a proto tedy ochrana nebývá nutná. Kód, který zjišťuje chybu, nalezneme ihned za datovými bajty. Jde o CRC (Cyclic Redundancy Check) [9]. Tento kód je vygenerován na straně odesílatele pomocí matematických vzorců, přičemž jako parametry do těchto výpočtů vstupují přenášená data. Příjímač použije pro kontrolu stejné matematické vzorce, a pokud výpočet neodpovídá, tak je odhalena chyba v přenosu.

Na konci zprávy se následně už jen potvrdí správné přijetí zprávy (ACK) a dojde k jejímu ukončení. Pokud zpráva není potvrzena, vznikne chybová zpráva. Nakonec by za každou zprávou měla následovat mezera tří pasivních bitů.

Chybová zpráva nemusí být vygenerována pouze na základě CRC kódu a potvrzení o přijetí zprávy. Existují i další způsoby, jak zabezpečit přenos. Například tzv. monitoring, který porovnává hodnotu vysílanou s aktuální hodnotou na sběrnici. Dále to může být vkládání bitů, které se uplatní pouze při přenosu několika stejných po sobě jdoucích bitů. Do této sekvence je uměle vložen jeden bit opačného významu. [9]

Z výše uvedeného vyplývá, že existuje datová zpráva (dominantní RTR), žádost o zprávu (pasivní RTR) nebo chybová zpráva, která vznikne například po odhalení chyby CRC kódem. Jednotlivé uzly ovšem mohou být zaneprázdněné, a proto se zavádí ještě zpráva o přetížení. Chybová zpráva a zpráva o přetížení mají mnohem jednodušší strukturu a méně přenášených bitů než u datové zprávy. Obsahují příznaky o chybách, popřípadě o přetíženích. [9]

3.2.4 Aplikační protokoly

Na nejvyšší vrstvě existuje mnoho různých protokolů, které definují přenos dat mezi samotnými aplikacemi. Aplikační protokoly se nezabývají nižšími vrstvami, a je tak možné přenášet data s jedním protokolem na více fyzických vedeních. Mezi aplikační protokoly patří

například CANopen, CMS (CAN Message Specification), NMT (Network Management) nebo DBT (Distributor). [7]

4 Software

4.1 Vývojové prostředí Easybuilder Pro

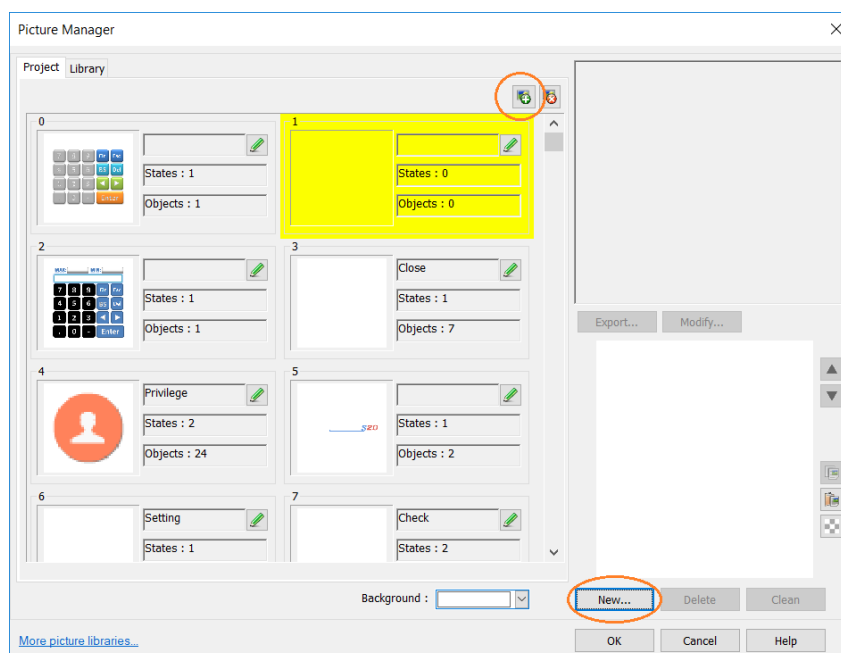
Společnost Weintek pro svá zařízení poskytuje i vývojové prostředí. Tento software je volně ke stažení, dostupný ať už na oficiálních stránkách výrobce nebo na stránkách českého distributora. K dispozici je také kompletní manuál, který popisuje veškeré funkce tohoto vývojového prostředí. Displej byl naprogramován ve verzi Easybuilder Pro V5.

4.2 Tvorba programu

4.2.1 Základní ovládací prvky

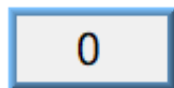
Mezi nejjednodušší prvky použité v programu patří Function Key. Tento objekt umožňuje uživateli přepínat mezi obrazovkami, umí vyvolat dialogové okno, či spustit makro. Je možné nastavit vlastnosti objektu, tedy vybrat obrázek objektu z knihovny. Tlačítko má dva stavy, a to stisknuto a nestisknuto. Vývojové prostředí dovoluje nastavit reakci na stisknutí i na uvolnění tlačítka. V programu je toto tlačítko využito mnohokrát, například na první obrazovce Start se nachází tlačítko, které není vidět, a které přepne uživatele na další obrazovku. Pomocí prvku Function Key je řešeno veškeré přepínání mezi obrazovkami.

Do programu lze vložit obrázek. Ten musí být ovšem uložen v knihovně projektu, kam je nutné jej tedy nejprve nahrát. Postup nahrávání obrázku znázorňuje obrázek 6. Dále je možné do programu vložit textový objekt, různé tvary obrazců, či tabulku.



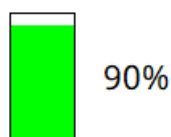
Obrázek 6 Přidání obrázku do projektu

Numerický objekt je určen k zobrazování číselných hodnot. Také je v programu často využit. Prvek dovoluje zobrazování hodnoty z předem zadané destinace, buď v místní paměti displeje, nebo ze vzdáleného zařízení. Je možné hodnotu i zadávat a zapsat ji na jinou adresu, než ze které jsou data čtena. Opět lze upravovat vzhled objektu a formát zobrazeného čísla je možné nastavit dle potřeby. Pomocí ASCII objektu, který je velmi podobný tomuto, lze číslo převést na písmeno odpovídající ASCII kódu. Numerický objekt dokáže posunovat desetinnou čárku doleva a prakticky tak umí vydělit číslo násobky deseti.



Obrázek 7 Numerický objekt

Bar Graph slouží ke grafickému zobrazení hodnot. Je na vývojáři, zda bude zobrazení dat v kruhovém nebo sloupcovém grafu. K parametrům, jako je nastavení zdroje dat pro objekt a hranice zobrazených hodnot, se přidává i možnost reagovat na hodnoty například změnou barvy. To se využívá při zobrazování procentuálního nabití baterie. Pokud je hodnota nabití akumulátoru 20 % a méně, bude graf červený, pokud 90 % a výše, bude graf zelený. Mezi těmito hodnotami je graf modrý.



Obrázek 8 Sloupcový graf s numerickým objektem v programu

Posledním elementárním objektem použitým v aplikaci je Word Lamp. Tento objekt může nabývat více stavů. Maximální počet stavů závisí na zvoleném módu pro vyhodnocování stavu. V prvním módu Value je možné porovnávat hodnotu v paměti s nastavenou hodnotou offsetu a na základě toho vyhodnotit a aktivovat daný stav kontrolky. Pokud tedy například zadáme offset pět a bude číslo pět nalezeno také ve zdrojovém registru, kontrolka se přestaví do stavu nula. Pro hodnotu šest v registru nastane stav jedna. Nevýhodou tohoto módu je, že pro další stav kontrolky musí být hodnota v registru vždy přesně o jedna větší. Lampa tedy nemůže být ve stavu nula při hodnotě například osm v registru a zároveň ve stavu jedna při hodnotě tři v registru. Další mód umožňuje reakci na LSB (Last Significant Bit), následující mód mění stavy v nastavitelném časovém intervalu. Poslední mód, použitý v programu, je reakce kontrolky na kombinaci přesně určených bitů. Adresa bitů musí být stanovena, a pro více stavů je samozřejmě nutné více kombinací, a tedy více čtených bitů.

Dále je možné použít i jednodušší objekt (Bit Lamp), který ovšem reaguje pouze na jeden určený bit, a má tedy maximálně pouze dva stavy.

4.2.2 Data sampling

Data sampling neboli vzorkování dat je nutné pro to, abychom mohli vytvořit graf. Data, která se v zadaném registru mění v čase, jsou zaznamenávána, případně trvale uložena, a lze je následně zobrazit. Aplikace obsahuje jeden Data Sampling s názvem SampTep. Nejprve je nutné vytvořit vzorkování a v dialogovém okně nastavit parametry. V sekci Sampling Mode lze určit, kdy se budou data vzorkovat. První volba (Time-based) znamená, že se budou data vzorkovat na základě časového intervalu, který je možné zvolit, druhá volba (Trigger-based) nabízí možnost reagovat na změnu zadaného bitu v paměti. Tato volba dokáže realizovat reakci na náběžnou, sestupnou hranu, ale i pro obě hrany dohromady. Dále je nutné doplnit adresu počátečního registru, od kterého budou vzorkovaná data vyčítána.

V úseku Data Record můžeme navolit maximální množství navzorkovaných dat. Pokud není zaškrtnuto políčko Auto. stop, začnou se po dosažení maximálního množství navzorkovaných dat nejstarší data přemazávat. V nastavení Data Format určujeme, jaký budou mít data formát a kolik proměnných se bude od počáteční adresy směrem k paměti s vyšší adresou vyčítat. Dále lze povolit vymazání navzorkovaných dat na základě změny stavu určeného bitu. Další definovaný bit pak dokáže pozastavit vzorkování. Jako poslední výběr v dialogovém okně je možné nastavit, zda a kam se budou data ukládat. K dispozici je výběr ze tří možností, a to uložení do paměti displeje nebo na externí paměti, buď SD kartu nebo USB disk.

The screenshot shows the 'Data Sampling Object' configuration window. It includes a 'Comment' field with 'SampTep', a 'Sampling mode' section with 'Time-based' selected and a '1 second(s)' interval, a 'Read address' section with 'Local HMI' PLC and 'LW 4' address, a 'Data Record' section with '10000' records and 'Auto. stop' unchecked, and several checkboxes for 'Clear real-time data address', 'Hold address', and 'History files' (Save to HMI memory, Save to USB disk, Save to SD card). 'OK' and 'Cancel' buttons are at the bottom right.

Obrázek 9 Nastavení vzorkování použité v programu

4.2.3 Alarmy

Pro zobrazení chybových hlášení jsou v programu použity alarmy. Nejprve je nutné vytvořit alarmy jako Event (Alarm) Log. Při nastavování alarmu se zadává adresa pro čtení hodnoty a podmínka, se kterou se hodnota porovnává a na základě toho vyhodnocuje aktivaci alarmu. V programu jsou použity podmínky na rovnost s kódem chyby přijatým z BMS. Jako reakci na splnění podmínky je pak pro příslušnou chybu přidělena textová zpráva, která uživateli stručně popisuje, co není v pořádku. Program obsahuje celkem osm chybových hlášení a jedno hlášení signalizující bezchybnou funkci systému. Příznak chybových hlášení a jejich stručný popis vidíme v příloze B.

4.2.4 Tagy a paměť

Pro jednodušší a rychlejší orientaci mezi uloženými daty jsou vytvořeny tagy. Jedná se o textové pojmenování registru, případně bitu. Všechny tagy nalezneme v Adress Tag Library v programu. Vytvořené tagy obsahuje příloha A.

Pro zpracování dat v displeji jsou použity registry LW (Local Word), které se adresují pouze výběrem možnosti LW a číslem příslušného registru. Využito je celkem patnáct registrů LW. U tagů, které ukazují na data přenesená z BMS se jejich adresa liší na první pohled v délce. Například tag pro bit Čtení určuje adresa 14A1010161, kde prvních osm znaků tvoří identifikátor pro CAN protokol zapsaný v šestnáctkové soustavě. Zbylé dvě číslice uvádí číslo registru (6) a pozici bitu v tomto registru (1). U tagu nebitové proměnné je číslo identifikátoru na stejném místě, další číslice opět znamená číslo registru, ovšem poslední dvojčíslí udává délku čtené proměnné. Díky tagům tak není nutné při každém vyčítání dat opisovat celou adresu a zároveň umožňují lepší přehlednost.

4.2.5 Makra

Makro je sled po sobě jdoucích instrukcí, které se provedou na základě jeho zavolání. To například umožňuje přidat do programu výpočty nutné pro splnění funkce displeje. Makra mohou být volána periodicky s daným časovým intervalem, který je možné nastavit. Každá obrazovka (okno) může mít přidělené makro, které se bude provádět v případě, že je obrazovka aktivní (zobrazená uživateli). Dále lze makro spustit v attributech každého okna, buď pouze jednorázově při otevření, nebo provádět opakovaně stále, nebo pouze při ukončení obrazovky. Makro lze také spustit jednou při startu displeje. Jako poslední možnost zavolání makra je reakce na aktuální stav určeného bitu. Podle toho se makro opakovaně provádí nebo ne.

Program obsahuje celkem pět maker, které se spouští různými způsoby. Každé makro má své ID číslo. Jejich funkce bude postupně rozebrána. První makro začíná s číslem nula. Syntaxe programovacího jazyka připomíná jazyk C, ovšem úplně stejná není. Nabídka obsahuje již vytvořené funkce, které lze v makru použít. V následujících makrech jsou aplikovány funkce GetData a SetData, které vyčítají nebo nastavují příslušné hodnoty do nebo ze zadaných cílových adres. Podporované datové typy jsou vypsány v tabulce 3.

Zápis datového typu	Obsazení paměti	Rozsah
bool	1 bit	0, 1
char	8 bitů	+127 až -128
short	16 bitů	+32767 až -32768
int	32 bitů	+2 147 483 647 až -2 147 483 648
float	32 bitů	
unsigned char	8 bitů	0 až 255
unsigned short	16 bitů	0 až 65535
unsigned int	32 bitů	0 až 4 294 976 295

Tabulka 3 Popis datových typů [10]

První makro se nazývá Baterie_1. Jeho funkcí je vyčtení proudu a celkového napětí z BMS a převod těchto hodnot do zobrazitelných údajů. Vzhledem k tomu, že BMS dělí napětí do dvou bajtů (vyššího a nižšího), je nutné tyto bajty správně spojit do jednoho slova, které se následně uloží do paměti displeje a je k dispozici pro další práci. Není nutné napětí dělit v makru, protože numerický objekt umí posunout desetinnou čárku. Například hodnota 100 uložená v registru se po správném nastavení numerického objektu vyčte a zobrazí se číslo 1.00. Další funkce makra je převod proudu. Proud je vyčten jako osmibitové znaménkové číslo, které je ovšem nutné přetypovat do šestnáctibitového znaménkového čísla, protože paměť displeje nedokáže zpracovat osmibitový formát. Před přetypováním je ještě potřeba korigovat chybu displeje, který posílá data v nesprávném formátu, jak bylo zjištěno při testování zařízení na vozidle. Jsou zavedeny tři podmínky, které proud přepočítávají na správnou hodnotu. Tento přepočít byl vytvořen na základě empirického testování. Před uložením proudu na správnou adresu se musí ještě hodnota vynásobit zadaným násobičem proudu. Celé makro Baterie_1 vidíme na následující straně.

```

macro_command main()           //Makro Baterie_1

unsigned char data[2]
unsigned short vysl
char proud
short prev_proud
char n

//Prevod napeti
GetData(data[0], "BMS", "CelkNap-H", 1)           //Vycteni vyssiho bajtu
GetData(data[1], "BMS", "CelkNap-L", 1)           //Vycteni nizsiho bajtu
vysl = (data[0] << 8) + data[1]                   //Slozeni dvou bajtu do jednoho slova
SetData(vysl, "Local HMI", LW, 0, 1)              //Ulozeni do prislusneho registru

//Prevod proudu
GetData(proud, "BMS", "Proud", 1)                 //Vycteni proudu
GetData(n, "Local HMI", "Nasobic", 1)            //Vycteni nasobice
if proud > -128 and proud < 0 then                //Osetreni nespravneho formatu dat
prev_proud = (proud+128) * n                       //prijateho z BMS
end if
if proud > 0 then
prev_proud = (proud-128) * n
end if
if proud == -128 then
prev_proud = 0
end if

SetData(prev_proud, "Local HMI", LW, 3, 1)        //Ulozeni prepocteneho proudu do registru

end macro_command

```

Druhé makro Teplota slouží k přetypování maximální a minimální teploty. V makru se provede vyčtení hodnot a jejich přetypování do šestnáctibitového znaménkového čísla. Data jsou uložena do svých registrů v paměti displeje.

Třetí makro Napeti je nejobsáhlejší. Jeho funkcí je převod maximálního a minimálního napětí článků a převod adresy těchto článků tak, aby bylo možné články indikovat pomocí objektu „Word Lamp“. Nejprve se převádí minimální a maximální napětí stejným způsobem jako v prvním makru. Obě hodnoty je nutné správně složit ze dvou příslušných bajtů přijatých z BMS.

Další část kódu se zaměřuje na převod dat z BMS na bitovou kombinaci. Každý bit ve dvou šestnáctibitových registrech představuje jeden článek. Maximálně tak tento systém může pracovat s 32 články. V případě vyššího počtu nebudou příznaky pro články s adresou 32 a vyšší zaznamenány v registrech. Kód pracuje tak, že v závislosti na čísle nastaví příslušný bit. Číslování začíná od nuly. Pro číslo článku nula tedy nastaví bit na nejnižší pozici v nižším registru do stavu jedna. Při minimálním napětí na článku 31 se nastaví jednička na nejvyšší pozici ve vyšším registru. Aby nedošlo k nastavení bitů v obou registrech zároveň, je nutné vždy vynulovat ten registr, do kterého v daný moment není zapisováno. Pro příznak článku s maximálním napětím jsou vyhrazeny další dva registry a způsob nastavování bitů je stejný. Na základě bitů v těchto registrech se poté rozsvěčují kontrolky, které informují uživatele

o člancích s minimálním a maximálním napětím. Část kódu makra sloužící pro převod adresy článku s maximálním napětím lze vidět níže.

```
unsigned char data[2]
unsigned short vysl
char null = 0

// Prevod adresy clanku s maximalnim napetim
GetData(data[0], "BMS", "MaxNapAdr", 1) // Vyceteni dat
if data[0] < 16 then // Test, do ktereho registru bude adresa ukladana
vysl = 1 << (data[0]) // Nastaveni bitu na pozici odpovidajici adrese clanku
SetData(vysl, "Local HMI", LW, 9, 1) // Zapis do registru
SetData(null, "Local HMI", LW, 10, 1) // Vynulovani dat z druhého registru
else
vysl = 1 << (data[0] - 16) // Nastaveni bitu na pozici odpovidajici adrese clanku
SetData(vysl, "Local HMI", LW, 10, 1) // Zapis do registru
SetData(null, "Local HMI", LW, 9, 1) // Vynulovani dat z prvního registru
end if

end macro_command
```

Čtvrté makro Poruchy má za úkol pouze uložení chybového kódu přijímaného z BMS do paměti displeje.

Páté makro Graf_nast slouží k jednorázovému nastavení os grafu při startu programu. Je zadán časový interval na ose x šedesát sekund, minimální hodnota na ose y nula a maximální hodnota na ose y čtyřicet. V případě požadavku na jiné počáteční krajní hodnoty grafu jde jednoduše pozměnit program a nastavit je. Další funkce tohoto makra spočívá v nastavení výchozího násobiče proudu. Je zadána hodnota násobiče dva a nikoliv pět, jak je uváděno v přenosovém protokolu BMS.

```
macro_command main() //Makro Graf_nast

char n = 2 //Nasobic proudu
char max = 40 //Maximalni teplota v grafu
char min = 0 //Minimalni teplota v grafu
char cas = 60 //Delka zobrazovaneho casoveho useku

SetData(max, "Local HMI", LW, 13, 1)
SetData(min, "Local HMI", LW, 12, 1)
SetData(cas, "Local HMI", LW, 14, 1)
SetData(n, "Local HMI", LW, 16, 1)

end macro_command
```

4.2.6 Systémové nastavení

Dialogové okno System Parameter Settings umožňuje nastavovat obecné parametry jako například typ modelu, pro který je program vytvořen, nebo připojení k programu libovolné dostupné písmo. Dále se zde může nastavit přídatná paměť.

Nejdůležitější část je ovšem na kartě Device, kde lze nakonfigurovat zařízení připojená k displeji. Konfigurace vlastního displeje vývojář musí udělat už při vytváření projektu. Další

zařízení je potřeba nastavit později. Nyní program obsahuje jedno zařízení pod názvem BMS. Toto zařízení, již dle názvu, představuje Battery Management Systém zabudovaný na experimentálním vozidle. V nastavení je vybráno, že BMS se připojuje přímo do displeje pomocí sběrnice CAN bus. Pro komunikaci slouží vybraný protokol CAN bus 2.0B, který se liší délkou identifikátoru, a rychlost sběrnice, která odpovídá rychlosti nastavené v BMS, a to 250 kb/s.

4.2.7 Obrazovky

Program se sestává z šesti obrazovek. Navigace mezi obrazovkami je řešena pomocí lišty tlačítek umístěné na každé obrazovce vždy u dolního okraje. Přesun na požadovanou obrazovku je rychlý, není nutné přeskakovat jiné obrazovky.

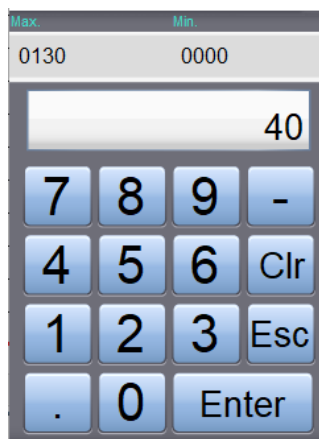
První obrazovka Start je úvodní, zobrazí se pouze při startu a po postupu uživatele dále se na ni již nelze zpět dostat.

Druhá obrazovka Baterie_1 zobrazuje základní informace o akumulátoru. Patří sem aktuální stav BMS, procenta nabití akumulátoru a údaje o odebíraném/dodávaném proudu a celkovém napětí na akumulátoru. Pro indikaci stavu jsou použity různobarevné kontrolky. Vzhledem k tomu, že údaj o proudu je nutné vynásobit násobičem, který se dá měnit v nastavení BMS, lze i zde změnit násobič, který je po startu displeje ve výchozím nastavení, a to dva. Grafický ukazatel nabití může měnit barvu v závislosti na procentuálním nabití. Pokud je aktivní tato obrazovka, je spuštěno makro Baterie_1, které zajišťuje přepočty zobrazovaných údajů.

Třetí obrazovka Baterie_2 ukazuje uživateli aktuální informace o maximální a minimální teplotě článku akumulátoru. Následně tyto hodnoty vykresluje do grafu závislosti teploty na čase. V grafu jsou dvě křivky, modrá značí nejnižší teplotu článku, červená nejvyšší teplotu článku. Aby bylo zajištěno, že data v grafu budou vykreslována správně, je nutné, aby makro pro přepočty teploty bylo spuštěno neustále, nezávisle na aktuálně uživatelem zvolené obrazovce. Makro se proto volá periodicky s časovým intervalem jedna sekunda. Se stejným časovým intervalem probíhá i vzorkování dat.

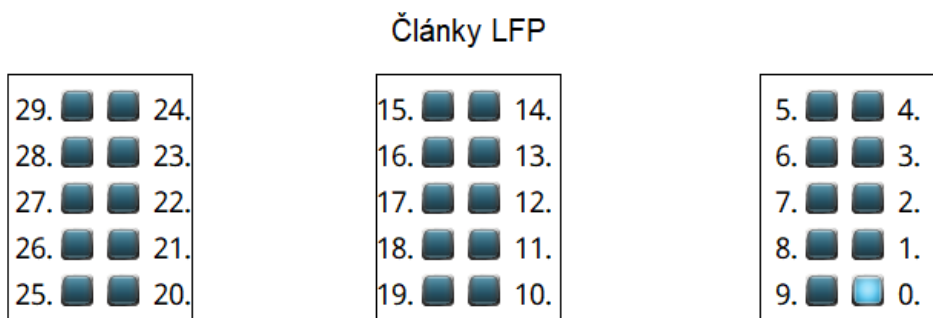
Na třetí obrazovce se ještě nachází vstupní numerické objekty, po jejichž aktivaci se otevře dialogové okno se zadáváním čísel. Tyto objekty slouží pro nastavování hodnot na osách grafu, přičemž jsou ošetřeny proti zadání nesmyslných hodnot. Maximální hranice je omezena v rozsahu od 0 do 130, minimální zobrazenou hodnotu lze pak nastavit v intervalu od -130 do 0. Nejdelší možný časový interval na ose x může být 9 999 sekund. Delší interval nemá smysl

vzhledem k tomu, že po dosažení 10 000 vzorků se tyto vzorky začnou přemazávat od nejstarších zaznamenaných.



Obrázek 10 Okno pro zadávání horní hranice na ose y v grafu teploty

Na čtvrté obrazovce Baterie_3 nalezneme tabulku maximálního a minimálního napětí na článku spolu s adresou tohoto článku, dále obrázek uspořádání článků akumulátoru typu LFP. Články jsou reprezentovány jednotlivými kontrolkami, adresy článků začínají od nuly. Této obrazovce je přiřazeno makro Napeti. Díky tomu se kontrolky u akumulátoru LFP rozsvěčují dle minimálního nebo maximálního napětí. Kontrolky jsou třístavové, při maximálním napětí daného článku se rozsvítí modře, při minimálním napětí červeně. Situace, kdy jsou aktivní oba stavy, reálně nemůže nastat. Pokud by k ní přesto z nějakého důvodu došlo, má modrá kontrolka přednost před červenou, a tedy bude aktivní signalizace maximálního napětí na daném článku.



Obrázek 11 Grafické znázornění článku LFP

Pátá obrazovka Baterie_4 se velmi podobá předchozí. Tabulka i přiřazení makra jsou stejné, ovšem obrázek s kontrolkami je tentokrát určený pro články akumulátoru NMC. BMS nerozezná, který akumulátor je připojen. Nepozná to tedy ani displej a je na uživateli, aby použil správnou obrazovku, dle aktuálního využívaného typu akumulátoru, pro zobrazování údajů o napětí článků a jejich adresách.

Články NMC

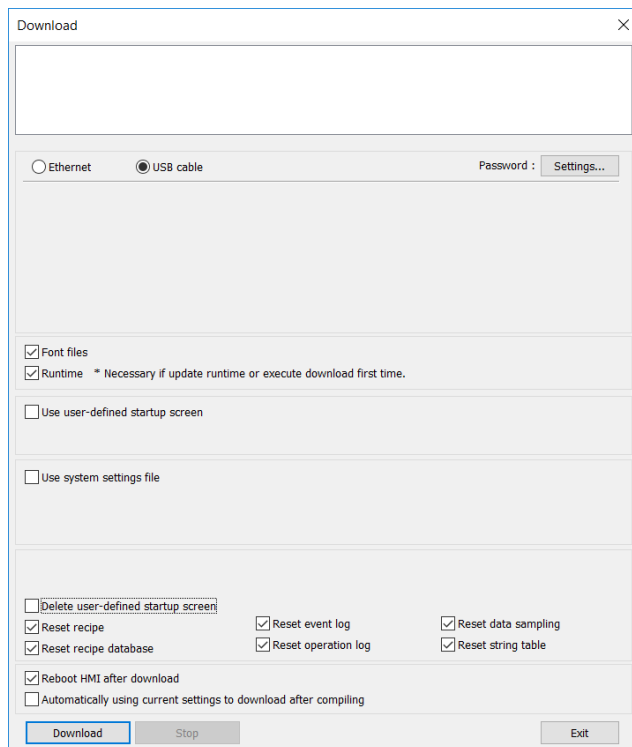


Obrázek 12 Grafické znázornění článků NMC

Poslední obrazovka Poruchy zobrazuje chybová hlášení. Je možné identifikovat, který článek je v poruše a o jakou chybu se jedná. Aktuální chybová hlášení zobrazuje objekt Alarm Bar. V tabulce, umístěné v dolní části obrazovky, vidíme historii chyb a čas, kdy byly zaznamenány. Na této obrazovce probíhá makro ukládající kódy chyb do paměti displeje, a to makro s názvem Poruchy.

4.3 Nahrání programu

Nahrát program do displeje je jednoduché. Stačí propojit displej s počítačem buď přes ethernet nebo USB kabelem. Displej podporuje i programování po micro-USB, na zadní straně nalezneme k tomu určený konektor. Pro nahrání programu je nutné vyvolat dialogové okno Download. Displej musí být připojený a zapnutý. Před otevřením dialogového okna dojde ke kompilaci programu. Přes jaké rozhraní bude displej komunikovat s počítačem se následně vybere v tomto okně. Dále lze zadat heslo, ve výchozím nastavení by ale mělo být heslo shodné, a to šest jedniček, a není třeba se o něj starat. Po odkliknutí se do displeje nahraje program a na konci celého procesu se restartuje. Programování po USB je snazší, v případě ethernetu je ještě nutné zadat IP adresu.



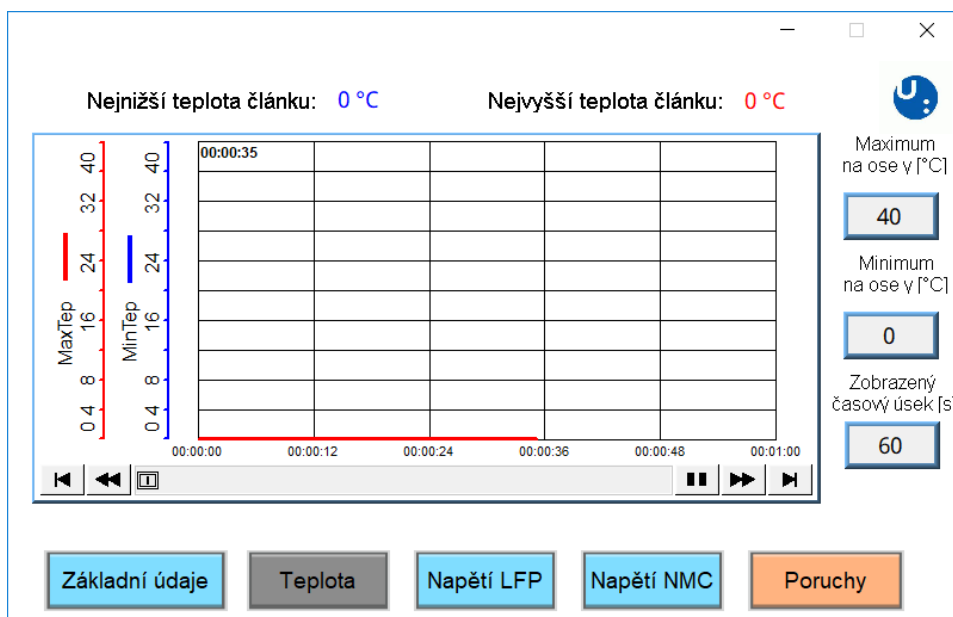
Obrázek 13 Nahrání programu do displeje

Vývojové prostředí dále nabízí nahrát program na USB disk či SD kartu. Při programování displeje pak není nutné mít počítač přímo u sebe. Zařízení má svůj systém, který dokáže vyčíst a zapsat program do své paměti z externího zdroje. V případě potřeby je též možnost jednoduchého vyčtení programu z displeje a uložení do počítače.

5 Testování programu

Při vývoji programu bylo nutné provádět průběžné testy, které ověřily správnou funkčnost dle požadavků. Testování proběhlo více způsoby tak, aby byla zajištěna maximální spolehlivost.

Základní ověření funkčnosti programu jistě umožňuje simulátor integrovaný přímo ve vývojovém prostředí. Je to jednoduchá a rychlá možnost testování. Při zapnutí simulace se vytvoří okno, které se chová jako displej.



Obrázek 14 Simulace programu

5.1 Testování pomocí tlačítek v programu

V tomto případě je využito, že uživatel může nastavovat jakékoliv hodnoty pomocí přídavných tlačítek v programu, které sloužily pouze pro tento účel. Například pro ověření správné funkčnosti makra byl tento prostředek vyhovující. Vytvoříme objekt Set Word, který po stisknutí při běhu programu nastaví na adresu BMS, odpovídající hornímu bajtu napětí akumulátoru, hodnotu dvacet. K získání skutečného napětí je nutné vyšší bajt vynásobit 256. Pokud tedy makro pracuje správně, mělo by být zobrazené napětí 512.0 V.

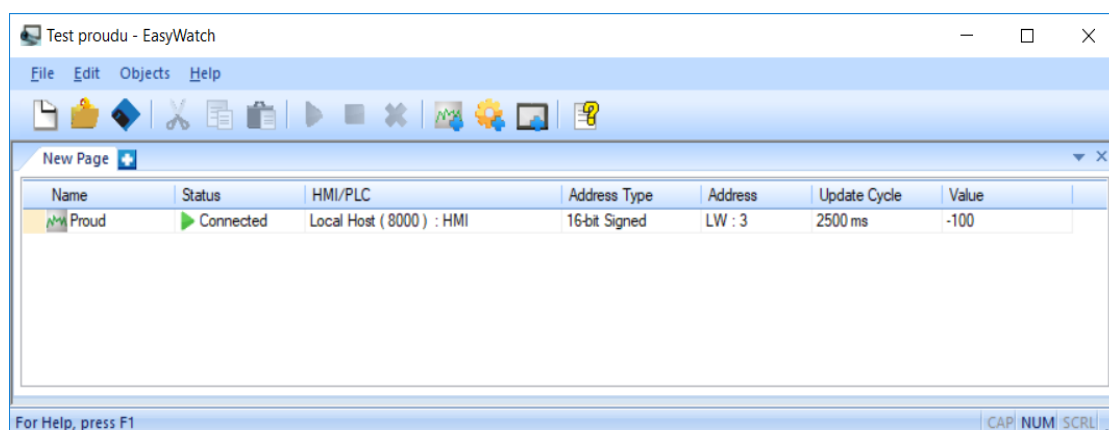
Nastav 20

Celkové napětí: 512.0 V

Obrázek 15 Test pomocí přidaného tlačítka

Pro testování teploty lze využít vlastnosti, že zmíněný objekt umí generovat i proměnné hodnoty, a tedy můžeme otestovat, jak se mění graf v čase. Je třeba ovšem umístit tento objekt na Common Window. Tím zajistíme, že proměnné hodnoty se budou generovat nezávisle na právě aktivní obrazovce, což je při testování grafu teploty podstatné.

Pokročilejší možnost testu je využití přídatného softwaru k vývojovému prostředí. Jedná se o program EasyWatch. Tento program dovoluje za chodu simulace sledovat a zapisovat hodnoty uložené v registrech. Po spuštění se musí nejdříve nakonfigurovat displej v HMI Manager. Po přidání nového zařízení je nutné zaškrtnout políčko Use Local HMI. Poté už můžeme přidat sledovaný registr. Lze nastavit, jak často se bude hodnota z displeje odečítat a v jakém formátu má být hodnota zobrazena. Test proudu by tak mohl vypadat jako na obrázku 16.



Obrázek 16 Test hodnoty proudu v EasyWatch

5.2 Testování pomocí CAN analyzátoru

Po otestování, že jsou všechny výpočty správné a že se údaje přepočítávají a zobrazují dle požadavků, byla provedena zkouška programu pomocí CAN analyzátoru. Princip byl takový, že do displeje byla posílána data simulující řídicí jednotku BMS. K testování se použil analyzátor USB-CAN Adapter, TRIPLE drivers od firmy IMFsoft, s.r.o. Vzhledem k tomu, že na straně displeje jsou v konektoru vyvedeny jiné piny pro CAN linku než na straně analyzátoru, bylo nutné vytvořit redukci. Porovnání pinů ukazuje následující tabulka.

Číslo pinu	CAN analyzátor	Číslo pinu	Displej eMT3070B
2	CAN_L	5	GND
5	GND	7	CAN_L
7	CAN_H	8	CAN_H

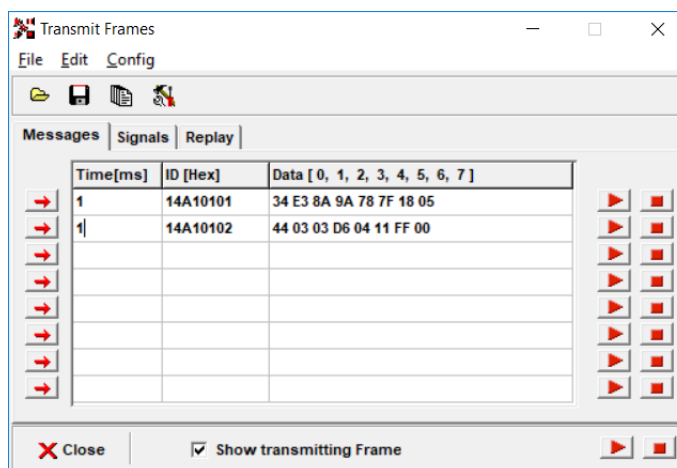
Tabulka 4 Rozmístění pinů [6] [11]

Mezi vodiče CAN_L a CAN_H je nutné připájet zakončovací odpor, bez kterého by redukce nemohla fungovat. Testovací zapojení CAN analyzátoru s redukcí je vidět na obrázku 17.



Obrázek 17 Zapojení CAN analyzátoru s redukcí při testování displeje

Software pro analyzátor vyžaduje nainstalovat ovladače. Po spuštění aplikace se musí v nastavení zvolit správný protokol (2.0B) a nastavit shodnou rychlost s nastavením v displeji (250 kb/s). Poté je možné začít vysílat data. Je potřeba zadat identifikátor a v šestnáctkové soustavě data, která budou přenášena. Například dle obrázku 18.



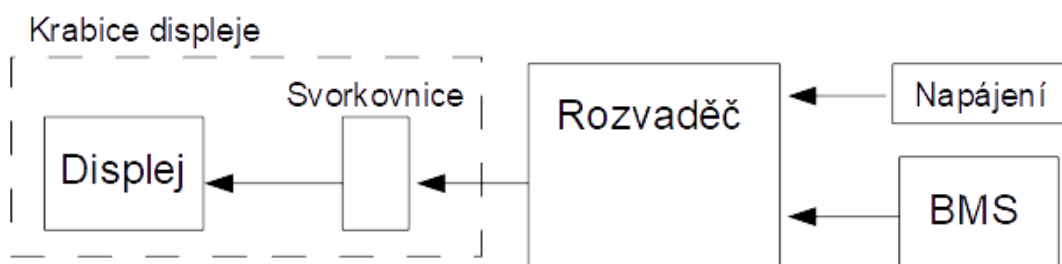
Obrázek 18 Vysílání dat z CAN analyzátoru

Spojení počítače s displejem přes CAN analyzátor a redukcí proběhlo úspěšně, data byla přijata a zobrazena dle očekávání.

6 Implementace displeje do experimentálního vozidla

6.1 Blokové schéma

Propojení dotykového displeje s experimentálním vozidlem je zobrazeno na blokovém schématu zapojení. Z BMS vede kabel se čtyřmi vodiči do rozvaděče. V rozvaděči se k těmto vodičům přidá ještě napájení. Tento svazek je vyveden na konektor ve dveřích rozvaděče. Z rozvaděče pak vede šestnáctižilový kabel do krabice displeje a uvnitř této krabice je každý vodič vyveden na svorkovnici. Ne všechny vodiče tohoto kabelu jsou nyní využity. Ze svorkovnice pak vodiče jdou přes příslušný konektor do displeje.



Obrázek 19 Blokové schéma zapojení

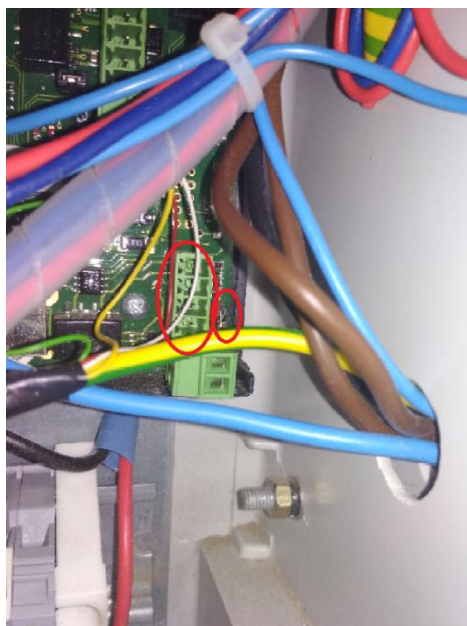
6.2 BMS

Jak již bylo zmíněno, z BMS jsou vyvedené čtyři vodiče. Tyto jsou připojeny přes čtyřpinovou šroubovou svorkovnici na desku plošných spojů hlavní řídicí jednotky BMS, do konektoru L. Na svorkovnici nalezneme vodiče CAN_L, CAN_H, +5 V a GND, jejich rozmístění ukazuje následující tabulka:

BMS	Číslo pinu svorkovnice	Číslo vodiče v kabelu
+5 V output	1	1
CAN_L	2	2
GND	3	3
CAN_H	4	4

Tabulka 5 Rozmístění vodičů v BMS

Čísla jednotlivých zapojených žil kabelu jsou v tomto případě shodné s číslem na svorkovnici. Stínění kabelu není připojeno. Z důvodu nutnosti zakončovacího rezistoru je dále potřeba propojit zkratosvorku U2 poblíž konektoru pro CAN komunikaci na desce řídicí jednotky.



Obrázek 20 Vyznačený konektor pro CAN linku a zkratosvorka v BMS

6.3 Rozvaděč

Kabely jsou uvnitř rozvaděče nataženy v kabelových žlabech. Signálové vodiče z BMS jsou protaženy do rozvaděče jeho spodním otvorem, napájecí vodiče jsou vedeny v rozvaděči přes ochranné pojistky. Ve dveřích rozvaděče je umístěn 24-pinový konektor, do kterého jsou přivedené vodiče připájeny. Čísla žil v kabelu přivedeného od BMS se neshodují s čísly pinů konektoru. Rozmístění vodičů v konektoru zobrazuje tabulka 6.

Číslo vodiče v přívodním kabelu	Číslo pinu v konektoru	Význam
x (červený)	1	+24 V
x (modrý)	2	0 V
1	3	+5 V output
2	4	CAN_L
3	5	GND (BMS)
4	6	CAN_H
x (žlutozelený)	16	GND (vozidlo)

Tabulka 6 Rozmístění vodičů v konektoru na dveřích rozvaděče

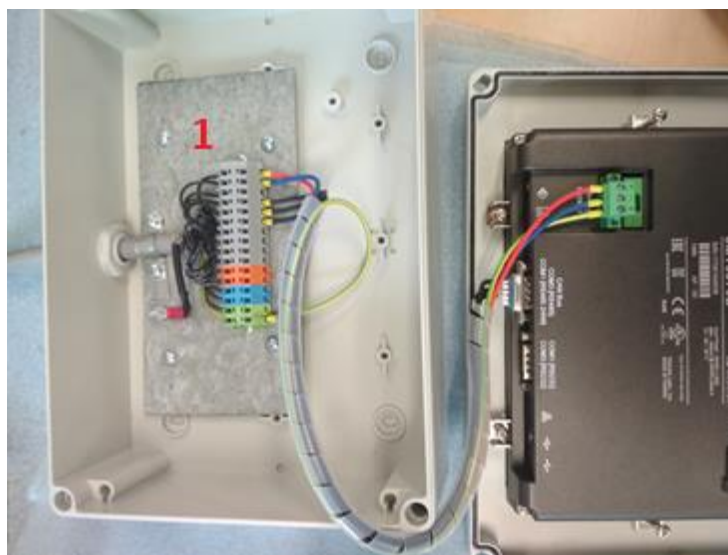
Na pin 16 je připojeno stínění a nulový potenciál z kostry vozidla. Čísla vodičů v kabelu, vycházejícího z rozvaděče dále do displeje, jsou shodná s čísly pinů konektoru. Například na pinu sedm konektoru je připojen vodič číslo sedm z šestnáctižilového kabelu, který dále pokračuje do krabice displeje. Vodič číslo šestnáct je žlutozelený.



Obrázek 21 Připájený konektor ke kabelu

6.4 Svorkovnice u displeje

Z rozvaděče vede kabel do krabice, ve které se nachází displej a svorkovnice. Uvnitř krabice je připevněn plech, ke kterému je přišroubovaná svorkovnice. Pro zajištění kabelu před vytrhnutím slouží přídatná objímka. Kabel vstupuje do krabice z horní části při pohledu na displej. Celkovou realizaci vnitřní části krabice zobrazuje obrázek 22. Vodiče jsou zapojeny do svorek postupně od vodiče s nejnižším číslem jedna až po vodič s číslem šestnáct. Na obrázku je první vodič vyznačen červenou jedničkou.



Obrázek 22 Vnitřní část krabice

Svorky vyrábí firma WAGO. Jsou pružinové, což je výhodné zejména díky vyšší odolnosti proti otřesům, které bývají u vozidla a při manipulování s displejem časté.

6.5 Konektory displeje

Z výstupní strany svorkovnice vychází vodiče už přímo do konektorů displeje. Do displeje jsou nyní připojeny dva konektory, a to napájecí a pro sběrnici CAN. Do napájecího konektoru vedou tři vodiče. Napájení + 24 V (červený), 0 V (modrý) a zemnicí žlutozelený vodič GND. Devítipinový konektor pak zajišťuje připojení CAN linky, je zapojen do zásuvky „e“ dotykového displeje. Jak již bylo zmíněno v kapitole o displeji, je nutné, aby byly připojeny tři vodiče. GND je na pinu číslo pět, CAN_L na pinu sedm a CAN_H na pinu osm. Vzhledem k tomu, že se jedná o topologicky koncové zařízení, je nutné připojit mezi piny CAN_H a CAN_L zakončovací rezistor o odporu sto ohmů.

7 Testování displeje na vozidle

Testování displeje na vozidle mělo za úkol ověřit přijímání a zobrazování dat přes instalované kabelové vedení ve vozidle. Po zapojení všech konektorů na správná místa bylo pomocí digitálního multimetru ověřeno propojení příslušných vodičů od BMS do displeje. Při testování byl využit i CAN analyzátor, díky němuž bylo možné rovnou porovnávat zobrazované údaje na obrazovce a skutečné hodnoty poslané řídicí jednotkou bateriového systému.

Pokud je zapnutá řídicí jednotka baterií a napájení displeje, začnou se okamžitě posílat a zobrazovat příslušná data na displeji. Při testování displej správně zobrazoval data. Jediná chyba byla nalezena u zobrazení hodnoty odebíraného nebo dodávaného proudu. V programu byl řešen přepočítání proudu přetypováním a vynásobením aktuálním násobičem. Tato varianta ovšem nefungovala, navzdory tomu, že posílaná data, která oznamují teplotu ve stejném formátu jako proud, jsou řešena obdobně přetypováním a na displeji se zobrazovala správně. Bylo proto nutné změnit přepočítání proudu na správné hodnoty. Při nulovém proudu byla přijímána hodnota -128. Na základě testování byly vytvořeny podmínky, které upravují hodnotu proudu tak, aby zobrazovaný údaj odpovídal realitě. Při následném testování se porovnával proud zobrazovaný displejem a proud z proudových kleští, které byly připojeny na jeden pól akumulátoru. Přepočítání bylo otestováno jak pro kladný, tak pro záporný proud. Pomocí tohoto měření se také nastavila výchozí hodnota násobiče proudu, protože velikost násobiče udávaná v komunikačním protokolu neodpovídala správným hodnotám.



Obrázek 23 Dotykový displej v provozu

V průběhu testování byly také prováděny změny v nastavení řídicí jednotky BMS. Byla provedena kontrola nastavení protokolu CAN 2.0B a přenosové rychlosti 250 kb/s. Dále byl CAN analyzátozem otestován parametr CAN posun adresy, který, pokud je nastavena nesprávná hodnota, posunuje datové bajty a displej kvůli tomu zobrazuje nesmyslné údaje.

Ve výchozím stavu posílá BMS na pozici čísla chyby číslo osm. Toto číslo nemá dle datasheetu BMS přidělený význam. Po kontrole ukazatele chyby v řídicí jednotce nebyla zjištěna žádná závada a tomuto chybovému číslu tedy bylo přiděleno hlášení o správné funkci bateriového systému.

Závěr

Cílem této práce bylo naprogramovat a připojit displej do experimentálního kolejového vozidla. Práce vyžadovala seznámení se s experimentálním vozidlem, především s jeho částmi podstatnými pro tuto práci, a také s přenosovou sběrnici CAN.

Způsob programování displeje je specifický, daný vlastním vývojovým prostředím od firmy WEINTEK.

Samotná realizace zabudování displeje vyžadovala navržení umístění přenosových vodičů a konektoru uvnitř rozvaděče. Zabudování displeje do samostatné krabice představuje výhodu v možnosti libovolného přemístování zařízení v rámci vozidla. Pro každého člena obsluhy se tak stává snadno dostupným a čitelným. Je také možné displej jednoduše odpojit v případě, že pro provoz není potřebný.

Zařízení bylo otestováno několika způsoby. Nejprve byla ověřována funkčnost programu a poté i celkového zapojení uvnitř experimentálního kolejového vozidla, přičemž v průběhu testování byl program upravován a laděn.

Displej bude nyní sloužit pro prezentační účely, nicméně toto rozhraní pro komunikaci s řídicími jednotkami experimentálního kolejového vozidla má velký potenciál. Panel pouze zobrazuje data, avšak v budoucnosti může být použit pro ovládání vozidla nebo pro jeho servis.

Použitá literatura

[1] Průmyslové muzeum Mladějov a Mladějovská průmyslová dráha. *Nejmodernější vozidlo na rozchodu 600 mm v Mladějově* [online]. [cit. 2018-04-03] Dostupné z: <http://www.mladejov.cz/aktuality-experimentalni-vozdlo-upce>

[2] ČERNÝ, Ondřej, LENOCH, Václav, NOVÁK, Jaroslav, SÝKORA, Petr. *Výzkum řízení pohonnů kolejového vozidla s nezávisle otáčivými koly* [online], Scientific Papers, University of Pardubice, Jan Perner Transport Fakulty, 18/2012, [cit. 2018-04-10] Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/54109/CernyO_VyzkumRizeni_2013.pdf?sequence=1&isAllowed=y

[3] LENOCH, Václav. *Elektronická optimalizace jízdy kolejového vozidla s nezávisle otáčivými koly* [online]. Pardubice, 2015. Disertační práce. Univerzita Pardubice, Dopavní fakulta Jana Pernera, Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě, [cit. 2018-04-16] Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/63951/LenochV_ElektronickaOptimalizace_JN_2015.pdf?sequence=1&isAllowed=y

[4] SADÍLEK, Ondřej. *Návrh a optimalizace bateriového systému pro kolejové vozidlo* [online]. Pardubice, 2017. Disertační práce. Univerzita Pardubice, Dopavní fakulta Jana Pernera, Katedra elektrotechniky, elektroniky a zabezpečovací techniky v dopravě, [cit. 2018-04-16] Dostupné z: http://dspace.upce.cz/bitstream/handle/10195/67888/SadilekO_NavrhaOptimalizace_SG_2017.pdf?sequence=1&isAllowed=y

[5] TECON, s.r.o. *Grafický panel WEINTEK eMT3070B* [online]. [cit. 2018-04-17] Dostupné z: https://www.weintek.cz/eMT_3070B.php

[6] WEINTEK LABS., INC. *Datasheet eMT3070B* [online]. [cit. 2018-04-16] Dostupné z: <https://www.weintek.cz/pdf/eMT3070B.pdf>

[7] KOCOUREK, Petr a Jiří NOVÁK. *Přenos informace*. Praha: Vydavatelství ČVUT, 2004. ISBN 80-01-02892-5.

[8] PUŽMANOVÁ, Rita. *Moderní komunikační sítě od A do Z: [technologie pro datovou, hlasovou i multimediální komunikaci]*. 2., aktualiz. vyd. Brno: Computer Press, 2006. ISBN 80-251-1278-0.

[9] GREGORA, Stanislav a Zdeněk MAŠEK. *Elektronické a mechatronické systémy v konstrukci silničních vozidel*. Pardubice: Univerzita Pardubice, 2008. ISBN 978-80-7395-082-8.

[10] WEINTEK LABS., INC. *Easybuilder Pro Ver. 5.05.02* [online]. 2013. [cit. 2018-04-29] Dostupné z: https://www.weintek.cz/pdf/EBPro_manual.pdf

[11] IMFsoft, s.r.o. *USB-CAN Adapter* [online]. 2006. [cit. 2018-04-24] Dostupné z: https://imfsoft.com/files/usb-can-adapter/documentation/usb-can-adapter_CZ.pdf

Seznam obrázků

Obrázek 1 Rozmístění konektorů na displeji	13
Obrázek 2 Princip referenčního modelu ISO/OSI	15
Obrázek 3 Blokové schéma CAN komunikace	18
Obrázek 4 Realizace sběrnice CAN	18
Obrázek 5 Přenosový rámec CAN 2.0A	19
Obrázek 6 Přidání obrázku do projektu	22
Obrázek 7 Numerický objekt.....	23
Obrázek 8 Sloupcový graf s numerickým objektem v programu	23
Obrázek 9 Nastavení vzorkování použité v programu.....	24
Obrázek 10 Okno pro zadávání horní hranice na ose y v grafu teploty.....	30
Obrázek 11 Grafické znázornění článku LFP	30
Obrázek 12 Grafické znázornění článků NMC.....	31
Obrázek 13 Nahrání programu do displeje	32
Obrázek 14 Simulace programu	33
Obrázek 15 Test pomocí přidaného tlačítka	33
Obrázek 16 Test hodnoty proudu v EasyWatch	34
Obrázek 17 Zapojení CAN analyzátoru s redukcí při testování displeje.....	35
Obrázek 18 Vysílání dat z CAN analyzátoru.....	35
Obrázek 19 Blokové schéma zapojení.....	36
Obrázek 20 Vyznačený konektor pro CAN linku a zkratosvorka v BMS	37
Obrázek 21 Připájený konektor ke kabelu	38
Obrázek 22 Vnitřní část krabice	38
Obrázek 23 Dotykový displej v provozu	40

Seznam tabulek

Tabulka 1 Porovnání olověného a lithiového akumulátoru	11
Tabulka 2 Zapojení pinů v konektorech displeje.....	13
Tabulka 3 Popis datových typů	26
Tabulka 4 Rozmístění pinů	34
Tabulka 5 Rozmístění vodičů v BMS.....	36
Tabulka 6 Rozmístění vodičů v konektoru na dveřích rozvaděče	37

Seznam příloh

Příloha A – Souhrn všech vytvořených a použitých tagů	48
Příloha B – Výčet všech alarmů	49
Příloha C – Protokol přenosu dat po CAN lince z BMS	50
Příloha D – CD s aplikací	

Přílohy

Příloha A – Souhrn všech vytvořených a použitých tagů

Název tagu	PLC	Typ	Adresa
Cteni	BMS	DATA_Bit	14A1010161
Cekani	BMS	DATA_Bit	14A1010162
Vyvozovani	BMS	DATA_Bit	14A1010163
Beh	BMS	DATA_Bit	14A1010164
Nizke napeti	BMS	DATA_Bit	14A1010165
Porucha	BMS	DATA_Bit	14A1010166
Vazna porucha	BMS	DATA_Bit	14A1010167
CelkNap	Local HMI	LW	0
MinNap	Local HMI	LW	1
MaxNap	Local HMI	LW	2
PrevedenyProud	Local HMI	LW	3
PrevMinTep	Local HMI	LW	4
PrevMaxTep	Local HMI	LW	5
Alarm	Local HMI	LW	6
MinNapObr_1	Local HMI	LW	7
MinNapObr_2	Local HMI	LW	8
MaxNapObr_1	Local HMI	LW	9
MaxNapObr_2	Local HMI	LW	10
Rezerva	Local HMI	LW	11
GrafMin	Local HMI	LW	12
GrafMax	Local HMI	LW	13
GrafCas	Local HMI	LW	14
Nasobic	Local HMI	LW	16
Nabiti	BMS	DATA	14A101011108
Proud	BMS	DATA	14A101012108
CelkNap-L	BMS	DATA	14A101013108
CelkNap-H	BMS	DATA	14A101014108
MaxTep	BMS	DATA	14A101015108
Chyba	BMS	DATA	14A101017108
KodChyby	BMS	DATA	14A101018108
MinNap-L	BMS	DATA	14A101021108
MinNap-H	BMS	DATA	14A101022108
MinNapAdr	BMS	DATA	14A101023108
MaxNap-L	BMS	DATA	14A101024108
MaxNap-H	BMS	DATA	14A101025108
MaxNapAdr	BMS	DATA	14A101026108
MinTep	BMS	DATA	14A101027108

Příloha B – Výčet všech alarmů

Příznak alarmu	Chybové hlášení
0	Chyba komunikace
1	Poškození EEPROM s kalibračními daty
2	Vadný balancovací FET – proud neteče – balancer nebalancuje
3	Vadný balancovací FET – proud teče trvale, balancer vybíjí článek
4	Vadný měnič napětí balanceru
5	Přehřátí balanceru > 130 °C
6	Přerušené teplotní čidlo
7	Přerušené teplotní čidlo baterie
8	Žádná chyba

Příloha C – Protokol přenosu dat po CAN lince z BMS

Adresa 0x14A10101	
Datový bajt	Popis dat
D0	Battery pack total capacity (0 .. 100 %)
D1	Current (-127 ... +126, value * 5 = current [A])
D2	Total voltage (low byte)
D3	Total voltage (high byte) - range 0 ... 65535, value / 10 = voltage [V]
D4	Highest temperature on the cells (-127 ... +126 deg. C.)
D5	Bit 0 - reading, Bit 1 - waiting, Bit 2 - balancing , Bit 3 - running, Bit 4 - low voltage, Bit 5 - error (not serious), Bit 6 - error (serious)
D6	The address of the CBU causing last error
D7	Internal error code (only for service purposes)

Adresa 0x14A10102	
Datový bajt	Popis dat
D0	Lowest voltage of the cell (low byte)
D1	Lowest voltage of the cell (high byte) - range 0-65535, value / 100 = voltage [V]
D2	The address of the cell with the lowest voltage
D3	Highest voltage of the cell (low byte)
D4	Highest voltage of the cell (high byte) - range 0-65535, value / 100 = voltage [V]
D5	The address of the cell with the highest voltage
D6	Lowest temperature on the cells (-127 ... +126 deg. C.)
D7	Reserved