

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ

DIPLOMOVÁ PRÁCE

Praha 2016

Bc. Antonín Štěpán

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
Katedra měření



Elektronická přístrojová deska

DIPLOMOVÁ PRÁCE

Studijní program: Kybernetika a robotika

Obor: Senzory a přístrojová technika

Vedoucí práce: doc. Ing. Jiří Novák, Ph.D.

Bc. Antonín Štěpán

Praha 2016

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 10.1.2016



Podpis



ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Antonín Štěpán**

Studijní program: **Kybernetika a robotika**
Obor: **Senzory a přístrojová technika**

Název tématu česky: **Elektronická přístrojová deska**

Název tématu anglicky: **Electronic Dashboard**

Pokyny pro vypracování:

Navrhněte a realizujte modul elektronické přístrojové desky pro analýzu a vizualizaci dat přijatých po sběrnici CAN. Dle dodaného schématu navrhněte desku plošných spojů a realizujte funkční vzorek zařízení. Mimo grafického výstupu musí zařízení podporovat práci s analogovými (např. měření napětí akumulátoru, měření vnitřní teploty zařízení) a digitálními (např. měření frekvence a střidy PWM signálu) vstupy. Pro práci s barevným LCD vytvořte grafickou knihovnu a multiplatformní konfigurační grafický software určený k návrhu uživatelského menu z osobního počítače. Veškerou funkcionalitu demonstруйте vzorovým příkladem.

Seznam odborné literatury:

- [1] Etschberger, K.: Controller Area Network, IXXAT Press 2001, ISBN 3-00-007376-0
- [2] Záhlava, V.: Návrh a konstrukce desek plošných spojů, Česká technika - nakladatelství ČVUT, 2005, ISBN 80-01-03351-1
- [3] Herout, P.: Java - grafické uživatelské prostředí a čeština, Kopp 2007, ISBN 80-7232-328-8

Vedoucí diplomové práce: doc. Ing. Jiří Novák, Ph.D.

Datum zadání diplomové práce: 19. listopadu 2014

Platnost zadání do¹: 31. srpna 2016

Doc. Ing. Jan Holub, Ph.D.
vedoucí katedry



Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 11. 2014

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Anotace

Cílem této práce je návrh elektronické přístrojové desky pro použití v segmentu automotive. Návrh desky plošných spojů je proveden dle dodaného schématu, přičemž součástí je také osazení a oživení elektroniky. Vizualizace dat je realizována pomocí LCD, zařízení kromě toho podporuje také měření analogových a frekvenčních vstupů. Vytvořená grafická knihovna slouží k zobrazování uživatelského rozhraní navrženého v konfiguračním grafickém softwaru pro PC. Funkce zařízení je předvedena na vzorovém příkladu, který posloužil také při testování výkonu systému překreslování.

Annotation

The goal of this diploma thesis is to design an electronic dashboard for automotive industry. Design of a printed circuit board was done according to provided schematic sheet and includes component mounting and electrical testing as well. Data visualisation is carried out using LCD. Device firmware supports functions for analog and frequency inputs measurement. Graphic library displays user interface designed using a developed configuration PC software. All available functions of the device are demonstrated within an example, which is also used during a graphic performance evaluation.

Poděkování

Rád bych tímto poděkoval svému vedoucímu doc. Ing. Jiřímu Novákovi, Ph.D. za podnětné rady při práci na projektu i při psaní závěrečné práce a dále své rodině za podporu během studia. Děkuji také společnosti TREMONDI s.r.o. za poskytnutí zadání, prostředků nezbytných pro vývoj a také cenných rad a připomínek v průběhu vývoje.

Obsah

1	Úvod	1
2	Teoretický rozbor	2
2.1	Zákaznické přístrojové desky	2
2.2	Specializované přístrojové desky	3
3	Hardware přístrojové desky	7
3.1	Schéma elektroniky	7
3.2	Návrh desky plošných spojů	9
3.2.1	Spínané zdroje	9
3.2.2	Rychlé obvody	10
3.2.3	Výkonové prvky	11
3.2.4	Externí signály	13
3.3	Podoba desky	14
4	Řídicí firmware	16
4.1	Systémová nastavení	16
4.1.1	Paměti	16
4.1.2	LCD	18
4.1.3	Sběrnice CAN	21
4.2	Správa zdrojů	22
4.2.1	Grafické podklady	22
4.2.2	Proměnné a jejich zpracování	23
4.3	Princip funkce	25
4.3.1	Start programu	25
4.3.2	Hlavní smyčka	26
4.4	Měřicí funkce	27
4.4.1	Interní napětí	27
4.4.2	Interní teplotní snímače	28
4.4.3	Externí napěťové vstupy	28
4.4.4	Externí teplotní čidla	30
4.4.5	Frekvenční vstupy	33
5	Lokální uživatelské rozhraní a jeho návrh	36
5.1	Standardizace formátů	36
5.1.1	Geometrické útvary	36

5.1.2	Rastrová grafika	37
5.1.3	Fonty	37
5.2	Grafická knihovna pro LCD	40
5.3	Grafický PC software	41
5.3.1	Správa zdrojů	42
5.3.2	Editor znakových sad	45
5.3.3	Grafický editor	46
5.3.4	Stažení projektu do přístrojové desky	52
6	Demonstrační software	54
6.1	Systémová nastavení	54
6.2	Konfigurace signálů	55
6.3	Obrazovky uživatelského rozhraní	57
6.4	Ovládání	58
6.5	Testování	58
6.5.1	Režimy testování	59
6.5.2	Konfigurace systému	60
6.5.3	Výsledky	61
7	Shrnutí	65
8	Závěr	67
9	Obsah CD	68
A	Deska plošných spojů	69
B	Programová struktura grafického editoru	71
C	Programová struktura firmwaru	73
	Seznam použité literatury	75

1. Úvod

Přístrojová deska jakožto nedílná součást automobilů urazila od svého počátku dlouhou cestu. Jak se vozidla postupně transformovala z prvních motorizovaných kočárů na konci 19. století v dnešní moderní automobily, tak procházel vývojem vzhled a využití přístrojové desky. Počáteční ochrana posádky před kamením odletujícím od kol se postupně přeměnila v panel poskytující řidiči přehled o hodnotách důležitých jízdních parametrů jako je rychlost, otáčky motoru nebo teplota oleje. Nelze opomenout také význam vizuálního provedení desky. Jelikož je přední část kabiny vozidla orientována především na řidiče, stává se často přístrojová deska spolu s volantem prvním prvkem, kterého si potenciální zákazník v interiéru všimá.

Rozšiřování elektronických prvků ve vozidle s sebou přineslo i nové měřené údaje, o kterých lze řidiče informovat, jako je například okamžitá a dlouhodobá spotřeba pohonných hmot, různé režimy měření uražené vzdálenosti nebo digitální ukazatel rychlosti. Logickou evolucí bylo postupné zavádění větších či menších displejů, na kterých lze všechny tyto parametry sledovat spolu se stavem komfortních funkcí, například klimatizace nebo autorádia. Poslední vývojový stupeň - kompletně elektronická přístrojová deska zobrazující na displeji vše včetně budíků otáček motoru a rychlosti - je tak očekávaným vyústěním předcházející situace.

Vývoj elektronické přístrojové desky orientované na motorsport probíhal podle zadání sestaveného dle dohody se společností TREMONDI s.r.o. [4]. Velkou motivací byla práce na zařízení určeném pro nasazení v reálných podmínkách automobilového průmyslu, a to v celém rozsahu vývoje.

Obsahem této diplomové práce je návrh desky plošných spojů, její osazení a elektrické oživení. Druhou částí je tvorba řídicího firmware pro zpracování dat přijatých ze sběrnice CAN, jejich převedení do numerické podoby a vizualizace na LCD. Součástí firmware je také grafická knihovna, jež obsahuje všechny grafické funkce nezbytné k vizualizaci dat, a dále funkce pro práci s teplotními, napěťovými a frekvenčními vstupy. Dalším bodem je návrh PC software sloužícího k rychlému a intuitivnímu návrhu uživatelského rozhraní, který respektuje standard definující jednotlivé signály a vytvořený taktéž pro účely této práce. Závěr je věnován sestavení uživatelského rozhraní demonstrujícího grafické funkce, činnost systému zpracování dat ze sběrnice CAN a testování výkonu celého systému.

2. Teoretický rozbor

Segment elektronických přístrojových desek na trhu lze dle univerzálnosti, rozšíření a fyzického provedení rozdělit do dvou kategorií: zákaznická zařízení určená pro osobní, nákladní či zemědělská vozidla, a specializovaná, nacházející využití například v motorsportu, armádních vozidlech či letectví.

2.1 Zákaznické přístrojové desky

Tato zařízení dosahují již z podstaty svého původu mnohem vyšší uživatelské intuitivnosti a grafické líbivosti. Každá taková přístrojová deska, ať už v civilním, nákladním nebo například i v zemědělském stroji, je vždy vytvářena na míru danému vozidlu. Hlavní výhoda spočívá v přístupu k širokému okruhu informací a konfigurovatelných parametrů, které jsou za běžných okolností nedosažitelné, neboť výrobci neposkytují sady pro dekodování komunikace na interních sběrnících ve volně přístupných databázích, ale pouze v uzavřených datových kontejnerech pro účely servisní diagnostiky nebo výroby kompatibilních OEM zařízení. Nastavení klimatizace nebo rádia je tak samozřejmostí, nepřekvapí ani integrovaná navigace nebo široká databáze informací o uražené trase, spotřebě, teplotách. Zároveň je v segmentu výkonnějších vozidel již poměrně běžná i konfigurace některých systémů, jako je rychlost řazení automatické převodovky, tuhost odpružení, odezva na plynový pedál nebo vypínání podpůrných elektronických systémů. Celek, popsaný předešlými řádky, lze z pohledu uživatele shrnout jako vyladěný systém, ve kterém se nachází vše podstatné na správných místech, který neobsahuje žádné funkční chyby vyvolávané běžným provozem a zároveň má i pěkný vzhled. Příkladem mohou být dvě přístrojové desky z moderních hybridních vozidel, jak lze vidět na obrázku 2.1.

Technické řešení je vzhledem k zástavbě do vozidla neveřejné, čas od času se nicméně určité detaily objeví v oficiálních prohlášeních. Takto byla na počátku roku 2014 [5] oznámena spolupráce automobilky Audi a společnosti NVIDIA při montáži grafických čipů Tegra K1 do přístrojových desek nových modelů vozidel. Tento procesor, využívaný například také automobilkami BMW, Tesla Motors, Honda či vícero dalšími



(a) Audi e-tron

(b) BMW i8

Obr. 2.1: Přístrojové desky moderních automobilů

členy koncernu Volkswagen, má 4 jádra pracující na frekvenci 2,3 GHz, 192 CUDA jader a podporuje práci s paměťmi typu DDR3L (*DDR3-Low Voltage*) a rozlišení až 4096×2160 . Využití nachází kromě automobilového průmyslu také v noteboocích, tabletech, mobilních telefonech či příručních herních konzolách. Nasazení podobného čipu klade samozřejmě vyšší nároky na návrh elektroniky i úvodní konfiguraci, jedná se nicméně o řešení, s nímž není žádný moderní grafický prvek nemožný, čímž dává vývojářům mnohem větší prostor pro práci a přizpůsobení výsledného vzhledu. Nevýhodou systému z pohledu zákazníka je tak pouze právě výše zmíněná odkázanost na rozložení funkčních prvků výrobcem, které nemusí vždy vyhovovat.

2.2 Specializované přístrojové desky

Segment specializovaných přístrojových desek se kategorii popsané výše nemůže rovnat ani v rovině designu, ani obsahem funkcí. Svůj účel ovšem splňuje v místech, kde je upřednostňován obsah a maximální konfigurovatelnost před vzhledem a kde má uživatel k dispozici dostatek informací o datech pohybujících se po datových sběrnicích vozidla.

Typickým odvětvím využívajícím tato zařízení je právě motorsport, kde je v rámci úspor hmotnosti odstraněno z vozu vše, co není nezbytné pro funkci vozidla a činnost posádky a není vyžadováno provozními předpisy. Zbytečné jsou zde i původní ukazatele rychlosti a otáček motoru, neboť k indikaci otáček slouží zeleno-oranžovo-červený pruh LED, dobře známý například z vozů Formule 1. Rychlost je oproti civilnímu prostředí údajem, který je důležitý pouze při přesunech po veřejných komunikacích v běžném provozu. Do popředí se naopak dostávají parametry jako zařazený rychlostní stupeň nebo teploty různých systémů. Vždy je však nezbytné zařídit dobrou čitelnost za všech světelných podmínek či dobrý kontrast barev pozadí a textu. V neposlední řadě využívají závodní vozidla speciální řídicí jednotky a nikoliv sériové, takže data odesílaná na sběrnici nejsou neveřejná, nýbrž definovaná koncovým zákazníkem, a tedy snadno konfigurovatelná do přístrojové desky.

Trh nabízí v současné době poměrně širokou nabídku specializovaných elektronických přístrojových desek pro motorsport, a to od zavedených výrobců s širokou oblastí vývoje a výroby i od menších firem specializujících se jen na malé okruhy výrobků.

Společnost **Magneti Marelli** [6], založená roku 1919 v Itálii, patří ke špičce v oblasti elektronických systému pro motorsport. Mimo jiné má ve svém portfoliu systémy pro řízení motoru (řídící jednotky, vstřikování, zapalovací svíčky), napájení vozidla (power boxy, alternátory), zpracování dat (datalogery, telemetrie, PC analýza) a samozřejmě také systém KERS (*Kinetic energy recovery system*), využívaný ve vozech Formule 1 k uchování kinetické energie vozidla při brzdění, která by byla jinak ztracena na brzdách ve formě tepla. Mezi výrobky se nachází také podpůrné systémy pro řidiče jako přístrojové desky a volanty.

Přístrojová deska **DDU 310** [7] v sobě kombinuje zobrazovací, záznamovou a měřicí funkci. Informace o hodnotách jednotlivých parametrů jsou uživateli předávány prostřednictvím 6,2" LCD TFT, 6 LED indikujících otáčky motoru a 2 varovných LED přiřaditelných k alarmům. Vzhled desky je následující:



Obr. 2.2: Přístrojová deska DDU 310 firmy Magneti Marelli

Zařízení nabízí uživateli širokou paletu funkcí - kromě zobrazování dat přijatých ze dvou sběrnic CAN dokáže ukládat do interní Flash paměti o velikosti 128 MByte hodnoty přijaté z až 128 kanálů. Měřicí funkce zahrnují externí a interní čidla NTC a PT1000 a Hallové sondy. Koncový zákazník si může v dodaném softwaru vytvořit vlastní obrazovky rozhraní, jež se přepínají tlačítky na krabičce, přičemž má k dispozici kromě výpisu hodnot jednotlivých signálů všechny standardní prvky, jako jsou bargrafy, diagnostické ikony určené k rychlé bitové analýze diagnostických signálů a až 48

konfigurovatelných alarmů s výstupem na displej a LED.

Nabídka společnosti **MoTeC** [8] je podobně jako u Magneti Marelli orientována na široký okruh systémů a doplňků pro motorsport a podobně jako její italský konkurent nabízí systémy řízení motoru, měření, logování a analýzu dat, elektrické rozvody, tlačítkové panely a samozřejmě také přístrojové desky.

Deska **MoTeC C187** [9] je distribuována se 7" displejem o rozlišení 800×480 s podporou anti-aliasingu. Měřicí funkce u tohoto zařízení zahrnují analogové napěťové a teplotní vstupy, digitální vstupy a také spínače typu Low side. Vnitřní datalogger má paměť 250 MByte a umí zpracovat až 1000 vzorků za sekundu. Vzhled desky se nachází na následujícím obrázku:



Obr. 2.3: Přístrojová deska C187 firmy MoTeC

Návrh uživatelského rozhraní se u společnosti MoTeC děje v odlišném duchu. Zákazník má po zakoupení k dispozici sadu obrazovek vytvořených výrobcem, konfigurovat může nicméně pouze zpracování zpráv ze sběrnice CAN. Za zpřístupnění možnosti úprav obrazovek si musí připlatit. Samotný konfigurační software ovšem nabízí velice široké možnosti editace jednotlivých prvků, velkou škálu dodatečných grafických efektů včetně vyhlazování a kruhových bargrafů a budíků pro zobrazení otáček motoru nebo rychlosti. Dá se tedy předpokládat, že obvodová část přístrojové desky obsahuje některý z grafických čipů spíše než řešení all-in-one v rámci jednoho mikrokontroléru.

Další zástupce na trhu, britská firma **General Engine Management Systems Ltd** [10] (GEMS), nemá ve svém portfoliu tolik výrobků jako výše zmínění výrobci. Nabízí řídicí jednotky, power boxy, řídicí systémy převodovek, zařízení pro sběr dat a přístrojové desky.

Deska **LDS4** [11] je určena k měření externích signálů, zobrazování dat přijatých ze sběrnice CAN a k záznamu příchozích dat. Je nabízena s 4,3" AMOLED displejem o rozlišení 480×272 a v plné barevné hloubce RGB888. Měřicí část zařízení počítá s 4 analogovými a 2 frekvenčními vstupy a jedním vstupem typu trigger pro spínání, například času na kolo. Vzhled desky se nachází na následujícím obrázku:



Obr. 2.4: Přístrojová deska LDS4 firmy GEMS

Software pro návrh uživatelského rozhraní dovoluje uživateli sestavit až 100 obrazovek. Mezi dostupnými funkcemi se nachází výpis hodnot signálů, dále vícero typů bargrafů a kruhových ukazatelů, diagnostické ikony, alarmy a grafy historie. Geometrické útvary (obdélník, elipsa) lze přidat jen s omezenými možnostmi konfigurace, nechybí však vkládání bitmap. Upload uživatelské konfigurace do zařízení je realizován pomocí USB, k dispozici je samozřejmě také funkce přeprogramování firmwaru zařízení na novější verzi.

3. Hardware přístrojové desky

Specifikace zařízení z hlediska funkčních prvků byla vytvořena zadavatelem. Mezi hlavní body patří využití mikrokontroléru s vestavěným řadičem pro TFT LCD, použití externí SDRAM pro potřeby LCD a FLASH paměti jako úložiště uživatelských dat, LED indikátory otáček motoru, výkonové spínače typu *High Side* a *Low Side*, vstupy pro analogové signály a měření frekvence na Hallových senzorech, měření všech interních napájecích napětí a teploty desky plošných spojů.

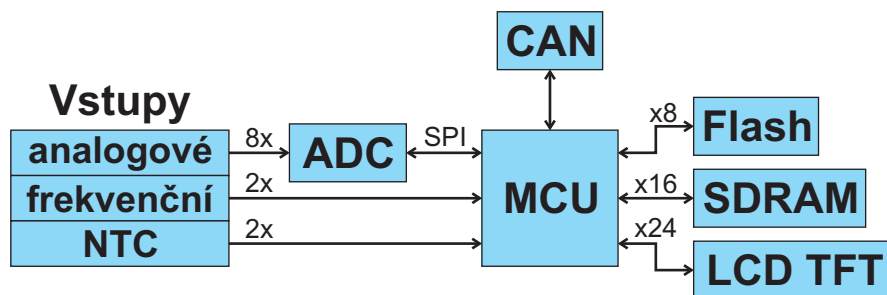
Práce na hardwaru byla rozdělena - návrh schémat provedl zadavatel, jejich překreslení do návrhového prostředí Altium Designer [12] a návrh DPS byl proveden v rámci diplomové práce. Následující podkapitola popisuje nejdůležitější integrované obvody použité ve schématu, nenachází se zde však schémata samotná, neboť nejsou náplní této práce.

3.1 Schéma elektroniky

Řízení přístrojové desky obstarává 32bit mikrokontrolér **ST32F429II** [13] od firmy STMicroelectronics. Tento čip v pouzdře LQFP176 je postaven na architektuře Cortex-M4, přičemž mezi jeho hlavní vlastnosti patří pracovní frekvence až 180 MHz, interní Flash paměť o velikosti 2 MB a 256 KB SRAM, dále interní řadič pro TFT LCD podporující rozlišení až 1024×768 a 24bit barevnou škálu, řadič pro SDRAM a různé typy Flash paměti, 2 CAN periferie, interní řadič USB 2.0 full-speed a 24 analogových vstupů.

Schéma zapojení mikrokontroléru obsahuje pouze nejnnutnější prvky pro rozvedení jednotlivých linek do zbylých schématických výkresů. Všechny napájecí piny jsou blokovány 100nF kondenzátory, reference pro ADC je externí o napětí 2,5 V. Oscilátor je taktěž externí. Blokové schéma elektroniky se nachází na obrázku 3.1.

SDRAM **IS42S16400J** [14] v pouzdře TSOP-54 poskytuje 16bit datové rozhraní při frekvencích mezi 133 a 200 MHz. Má velikost 64 MBit a slouží primárně jako buffer obrazových dat pro LCD. Zapojení čítá 16 datových a 12 adresních linek a dále systémové linky jako hodinový signál, Chip Select, Bank Select, atd, přičemž jsou všechny připojeny na odpovídající piny mikrokontroléru. Konkrétní detaily interní funkce a konfigurace jsou



Obr. 3.1: Blokové schéma elektroniky

uvedeny v podkapitole 4.1.1.

Externí paměť **MT29F1G08** [15] typu NAND Flash slouží k uskladnění uživatelských dat (konfigurace, obrazovky, fonty, grafika). Na desce se nachází v pouzdře TSOP-48, ovšem pouze s 8bit adresní/datovou sběrnicí. 16bit verze jsou prodávány pouze v pouzdře VFPGA, které není možné pájet běžnou cestou ručně při prototypové výrobě. Také pro tento typ paměti má mikrokontrolér připraveny odpovídající piny.

LCD modul **COG-T430V2100-L2** [16] typu WQVGA o rozlišení 480×272 podporuje až 24bit barevnou hloubku a obsahuje bílé LED podsvícení. K mikrokontroléru je zapojen v plně barevné šířce, řízení pak probíhá pomocí standardních linek DotCLK, VSYNC, HSYNC a DE (Data Enable). Podsvícení je řešeno zdrojem konstantního proudu, řízeným DA převodníkem z mikrokontroléru a opět navrženým zadavatelem.

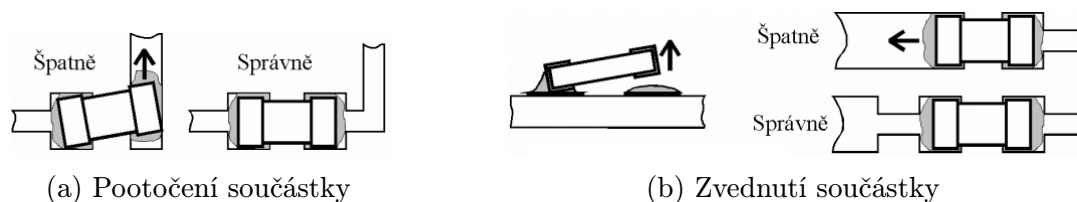
Zbývající funkcionality shrnuje následující tabulka:

Funkce		Komentář
Komunikace	CAN	CAN 2.0B kompatibilní, až 1MBit/sec
Uživatelské rozhraní	RPM LED	Indikace otáček motoru
	Alarm LED	Uživatelsky konfigurovatelné
	Spínače	Externí, uživatelsky konfigurovatelné
Vstupy	Analogové	12bit rozlišení, vstupní DP filtr
	Frekvenční	Hallovy senzory
	Teplotní	NTC senzory, rozsah -55 až $+150^{\circ}\text{C}$
	Napětí akumulátoru	vstupní DP filtr
Výstupy	Napěťová reference	5 V, 100 mA
	High Side	až 5,4 A, včetně zpětné vazby
	Low Side	až 3,5 A, včetně zpětné vazby
Konektor	Deutsch Autosport	

Tab. 3.1: Ostatní funkcionality přístrojové desky

3.2 Návrh desky plošných spojů

Výchozí požadavky na DPS byly stanoveny tvarem krabičky, umístěním konstrukčních prvků (šrouby, tvar konektoru) a také výrobními možnostmi. Deska je zhotovena jako čtyřvrstvá v konstrukční třídě 6. Vzhledem k různorodé povaze obvodů, kde je v některých místech nezbytné kombinovat výkonové obvody a spínané zdroje spolu se zpracováním analogových signálů, respektive velmi rychlé (SDRAM, LCD) a analogové signály, bylo nutné před započítáním práce zvážit i vzájemné rušení. V neposlední řadě se jedná také o dodržení pravidel elektromagnetické kompatibility (EMC) a ochrany proti elektrostatickým výbojům (ESD). Posledním požadavkem bylo dodržení návrhových pravidel pro strojové osazení a pájení součástek, což na jednu stranu umožňuje použití menších pouzder, neboť s ručním osazením se počítá pouze u prototypu a nevádí tak vyšší náročnost pájení, na druhou stranu je vyžadován specifitější přístup k jejich rozmisťování a propojování. Roli zde hraje odvod tepla při pájení přetavením a následném chladnutí. Je-li množství odváděného tepla na stranách součástky asymetrické (vodiče mají příliš rozdílnou šířku), může dojít buď k jejímu pootočení („plavání“) nebo zvednutí („Manhattan effect“), viz [2], str. 33. Pro lepší představu, jak se obě návrhové chyby projevují, slouží následující obrázky:



Obr. 3.2: Projevy špatného návrhu DPS (převzato z [2], str. 33)

Následující podkapitoly popisují návrh zapojení, která mají své specifické požadavky a z kterých vyplývají omezení pro zbytek desky.

3.2.1 Spínané zdroje

Přístrojová deska je dle specifikace dimenzována na vstupní napětí 8-18 V z akumulátoru vozidla. Převod napájení na napětí využívaná jednotlivými obvody je realizován řetězcem dvou spínaných zdrojů, prvním na 5 V a z něj pak druhým na 3,3 V. Třetím zdrojem na desce je zdroj konstantního proudu pro podsvícení displeje. Napěťové zdroje jsou typu Step-Down (*Buck*), tedy s výstupním napětím nižším než je vstupní. Proudový zdroj je pak typu SEPIC, jenž umožňuje vstupní napětí vyšší, stejné i nižší než je vstupní. Konstrukčně se jedná o řetězec dvou zdrojů, z nichž první je typu Step-Up a druhý Step-Down.

Návrh DPS probíhal u všech tří obvodů v souladu s návrhovými pravidly, jež lze podrobněji nalézt například v [17] a z nichž nejdůležitější jsou ve zkratce následující:

1. **Zpětná vazba:** Vodič ZV musí být veden co nejdále od induktoru a zdrojů rušení obecně. Optimálním řešením je umístit jej na opačnou stranu desky plošných spojů než je induktor.
2. **Blokovací kondenzátory:** Jako u všech ostatních integrovaných obvodů, i zde je nezbytné umístit je co nejbližší napájecích pinů.
3. **Vodiče:** Vodiče mezi jednotlivými součástkami ve zpětné vazbě a výkonových okruzích musí být co možná nejkratší, v opačném případě může kvůli vysokým spínacím frekvencím vznikat nežádoucí elektromagnetické rušení vedoucí k problémům na testech elektromagnetické kompatibility (EMC). Zároveň by měly být vstupní kondenzátory uzemněny ve stejném bodě jako výstupní kondenzátory a dioda.
4. **Zemnicí plochy:** Rozmístění ploch na obou stranách DPS je doporučováno pro redukci elektromagnetického vyzařování, zároveň mohou být tyto plochy využity k odvodu tepla z integrovaného obvodu.

3.2.2 Rychlé obvody

Přítomnost externích pamětí a LCD bez vlastního řadiče ve schématu vnesla do plánování rozvržení desky plošných spojů další aspekt, jímž je umísťování cest s rychlými signály. Nejedná se samozřejmě o obvody s pracovní frekvencí v řádech GHz, bylo nicméně nutné začít na dané zapojení pohlížet s určitou opatrností, neboť nevhodně provedený návrh by mohl vyústit v náhodné chyby a tudíž i nutnost uměle snižovat výkon daných periférií.

Oba typy pamětí i LCD přenášejí data běžnými paralelními sběrnicemi. Externí SDRAM na rozdíl od novějších pamětí typu DDR neklade důraz na impedanční přizpůsobení cest, jejich minimální a maximální délku ve smyslu absolutních hodnot a nezavádí diferenční páry. Jediným parametrem, jenž musí být respektován, se tak stává rychlost šíření signálu plošným spojením.

Výpočet tzv. *propagation delay* (zpoždění signálu na přenosové lince) lze pojmut s různou složitostí podle toho, jak moc je tento parametr v návrhu PCB důležitý. Vzorce výpočtu pro různé typy rozložení vodičů a vodivých ploch na desce plošných spojů je možno najít i v [2], kapitola 4, nicméně pro naše účely si vystačíme se známou konstantou zpoždění přenosu na DPS **15 cm / ns**.

Pracovní frekvence jednotlivých obvodů jsou následující:

- **SDRAM:** hodinový signál $f_{SDRAM} = 84 \text{ MHz}$
- **LCD TFT:** hodinový signál $f_{LCD} = 10 \text{ MHz}$
- **NAND Flash:** bez hodinového signálu, časovací konstanty v desítkách ns

Délka jedné logické úrovně je rovna polovině délky periody, u externí RAM, která je se svou nejvyšší pracovní frekvencí potenciálně nejrizikovějším elementem, pak dopravní zpoždění nesmí přesáhnout

$$\frac{1}{2 \cdot 84\,000\,000} \cdot 1\,000\,000\,000 \approx 5,952 \text{ ns}$$

Vezmeme-li tedy v potaz hodnotu přibližného dopravního zpoždění na jednotku vzdálenosti uvedenou výše, pak vidíme, že maximální délka vedení mezi mikrokontrolérem a SDRAM by neměla přesáhnout

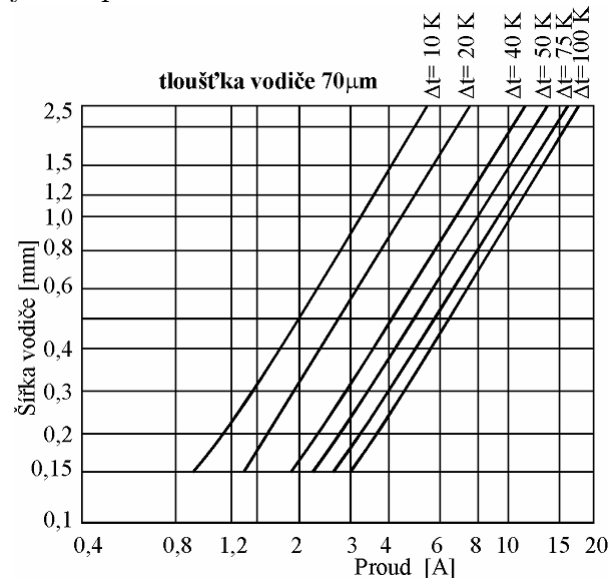
$$5,952 \cdot 15 \approx 89,3 \text{ cm}$$

což při umístění paměti na opačné straně plošného spoje přímo pod mikrokontrolérem nebylo problémem dodržet. Praxe tuto domněnku potvrdila - maximální délka vodiče mezi SDRAM a mikrokontrolérem je 45 mm, u LCD je to pak 81 mm. Je samozřejmě nezbytné uvažovat použitou konstantu dopravního zpoždění 15 cm / ns s určitou rezervou, neboť síť cest pod mikrokontrolérem je poměrně komplikovaná (nacházejí se zde prokovy a ve vnitřních vrstvách jsou rozlité plochy), nicméně polovina takto vypočítaného maxima je stále dostatečným prostorem pro další případná zpoždění. I následná práce s pamětí ukázala, že k žádnému překročení limitu vzhledem k nulové chybovosti datových přenosů pravděpodobně nedošlo.

3.2.3 Výkonové prvky

Design přístrojové desky obsahuje kromě běžných digitálních a analogových obvodů a rychlých obvodů také výkonové prvky sloužící k spínání vyšších zátěží. Na desce se nachází vždy dva spínače typu High Side (spínání v kladné větvi) a dva spínače typu Low Side („tvrdé“ uzemnění). Jak je uvedeno v tabulce 3.1, spínače High Side mohou dohromady poskytnout až 11 A a Low Side pojmout až 7 A společně. Takové proudové zatížení je pro plošný spoj přirozeně potenciálně rizikové, neboť nevhodným návrhem by mohlo dojít k velmi rychlému poškození desky vlivem přehřátí. Rozmístění jednotlivých výkonových spínačů vůči konektoru, výběr pinů na konektoru i rozložení ploch ve vnitřních vrstvách bylo plně podřízeno požadované proudové zatížitelnosti a dobrému odvodu tepla.

Deska plošných spojů je tvořena 4 vrstvami - vnější jsou z mědi o tloušťce 0,018 mm, vnitřní vrstvy pak 0,036 mm. Jedna z vnějších vrstev je sice v prostoru hlavního konektoru vyhrazena pro ochrannou bariéru proti rušení vnikajícímu do elektroniky z konektoru, bylo nicméně možné umístit na ni alespoň rozlitou plochu pro odvod tepla. Souhrnně tak bylo pro přenos proudu k dispozici 90 μm mědi a dále rozlitá plocha o tloušťce 18 μm . Minimální šířka cest byla čerpána z obrázku 3.3.



Obr. 3.3: Proudová zatížitelnost vodičů (převzato z [2], str. 43)

Tabulka počítá s tloušťkou mědi 70 μm , což je méně, než bylo při návrhu desky k dispozici, musel však být brán v potaz nedokonalý odvod tepla z vnitřních vrstev a také samozřejmě fakt, že se jedná o více měděných plátů se souhrnnou tloušťkou a ne o jeden celistvý. Přihlédneme-li k faktu, že krabička přístrojové desky je do jisté míry utěsněná a prostor automobilu nebude pravděpodobně příliš chlazen, neměla by se teplota plošného spoje při plném proudovém zatížení zvýšit o více než 40 - 50 $^{\circ}\text{C}$, aby nebyl zbytečně tepelně namáhán spolu s osazenými spínači. Z tabulky podle této úvahy plyne, že šířka výkonových cest by neměla být menší než 1,5 - 2 mm u High Side a 1 - 1,4 mm u Low Side snímačů.

Finální verze desky plošných spojů má šířky výkonových cest navrženy spíše nad horní hranici intervalů. Také faktické proudové zatížení je v praxi nižší, než je uvedeno ve specifikaci, o to lépe tak bude deska zátěž snášet.

3.2.4 Externí signály

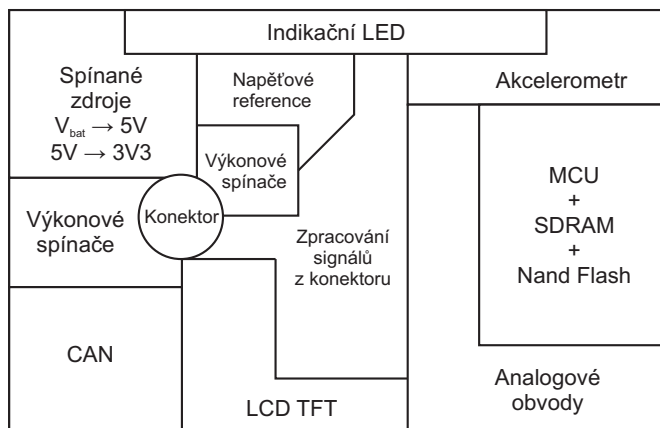
Propojení přístrojové desky s vozidlem je dle verze možné realizovat dvěma způsoby. Prvním je 6pin konektor obsahující pouze nejzákladnější vodiče jako je napájení, zem, datové vodiče sběrnice CAN a servisní vodič. Rozšířená verze přístrojové desky využívá konektor firmy Deutsch Autosport [18], díky čemuž je k dispozici i více funkcionalit. Jak je uvedeno v tabulce 3.1, přítomny jsou analogové a frekvenční linky, vodiče vyhrazené pro připojení NTC senzorů či výkonové vstupy a výstupy.

Všechny linky vedoucí z konektoru do plošného spoje jsou potenciálním vstupem vnějšího elektromagnetického rušení, kterém je elektronika vystavena při běžném provozu. Každá linka je proto nějakým způsobem chráněna proti ESD, a to ať už sériovým odporem, specializovaným integrovaným obvodem nebo TVS diodami, nebo se ochrana nachází v rámci vstupu do obvodu pro zpracování signálu. Všechny linky mají také vlastní nízkokapacitní keramický kondenzátor určený k odstranění vstupního rušení.

Signály z konektoru jsou za ESD ochranou přivedeny na tvarovače signálu, přičemž jejich povaze odpovídá také způsob úpravy - analogové signály procházejí pouze základním DP filtrem, signály z NTC korigují zapojení s operačním zesilovačem a externí analogové signály dále vedou na externí analogově - digitální převodník a prostřednictvím rozhraní SPI do mikrokontroléru. Měření interních napětí a NTC čidel provádí přímo mikrokontrolér prostřednictvím interního A/D převodníku.

3.3 Podoba desky

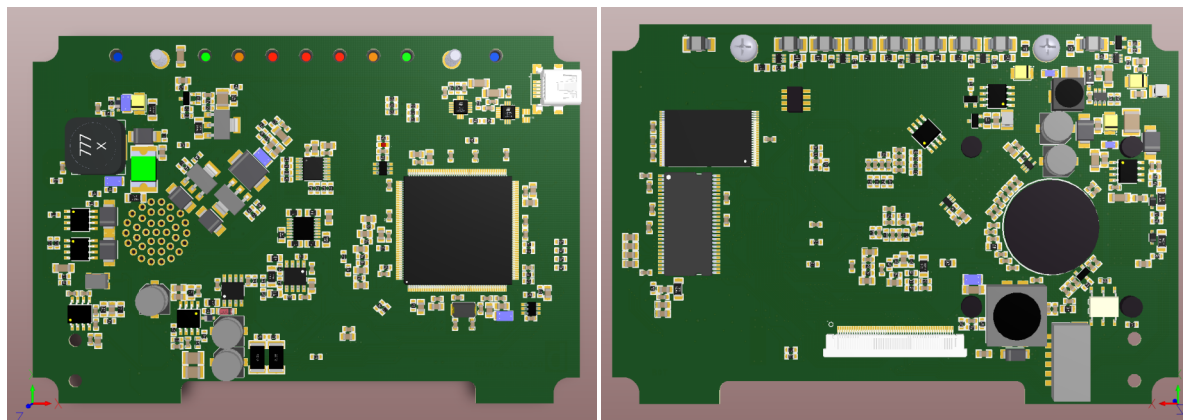
Návrh desky plošných spojů byl průběžně konzultován se zadavatelem. Na základě dohody je možné v této diplomové práci zobrazit pouze náhledy jednotlivých vrstev desky plošných spojů, nikoliv však použitá schémata. Pro lepší představu o rozmístění jednotlivých funkčních regionů poslouží znázornění na obrázku 3.4.



Obr. 3.4: Rozmístění funkčních bloků v elektronice

Rozmístění „regionů“ na desce odpovídá požadavkům jednotlivých součástí elektroniky a také výrobním možnostem. Pozice indikačních LED byla zadána pevná a bez možnosti změny, neboť s nimi na daných místech počítá krabice. Také umístění hlavního konektoru nebylo libovolné kvůli montáži do vozidla. Po usazení těchto 2 součástí následovaly výkonové spínače se svým požadavkem na co nejkratší vodiče, které se tedy musí zákonitě nacházet co nejbližší konektoru. Jelikož jsou spínané zdroje převádějící napětí akumulátoru rizikové z hlediska rušení, musí ležet co možná nejdále od analogových oblastí. Napěťové reference pro analogově-digitální převodník a externí použití na konektoru by neměly být moc vzdáleny jednak od jejich zdroje, jednak od spotřebiče - zde A/D převodníku a operačních zesilovačů. Analogová oblast okolo mikrokontroléru přebírá již zpracované signály a pouze je vyhlazuje koncovými filtry. Mikrokontrolér samotný je tak oddělen od všech rušivých elementů dostatečným prostorem. Potenciálně slabým místem je jeho vzdálenost od spínaných zdrojů z hlediska pokrytí neočekávaných výkyvů v proudovém odběru, tento problém se nicméně díky šířce rozlité plochy v napájecí vrstvě a blokaci napájecího napětí kondenzátory s dostatečnou kapacitou neprojevil.

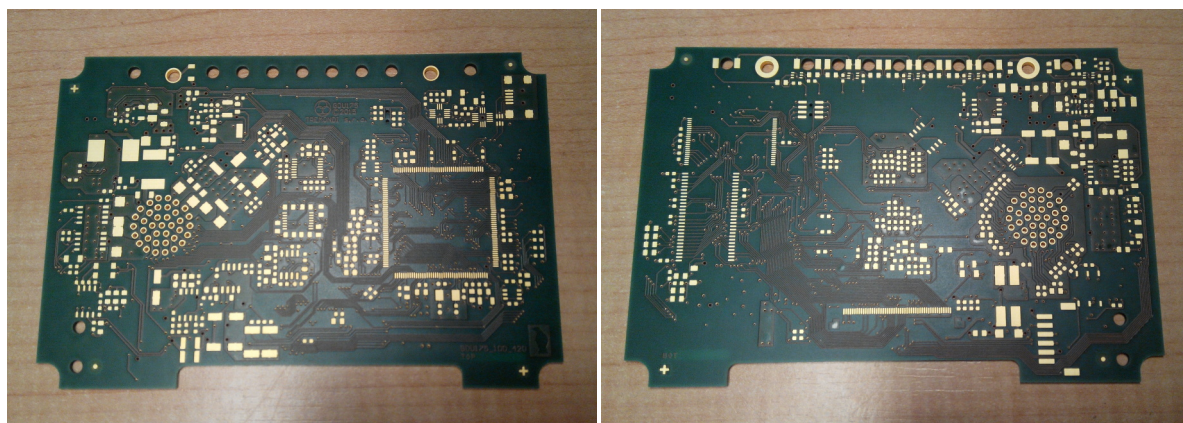
Výsledný vzhled elektroniky byl zdokumentován ve 3 fázích - 3D rendery v Altium Designeru a vyrobené neosazené a osazené desky plošných spojů. Všechny fáze jsou zachyceny na následujících obrázcích 3.5, 3.6 a 3.7.



(a) Strana Top

(b) Strana Bottom

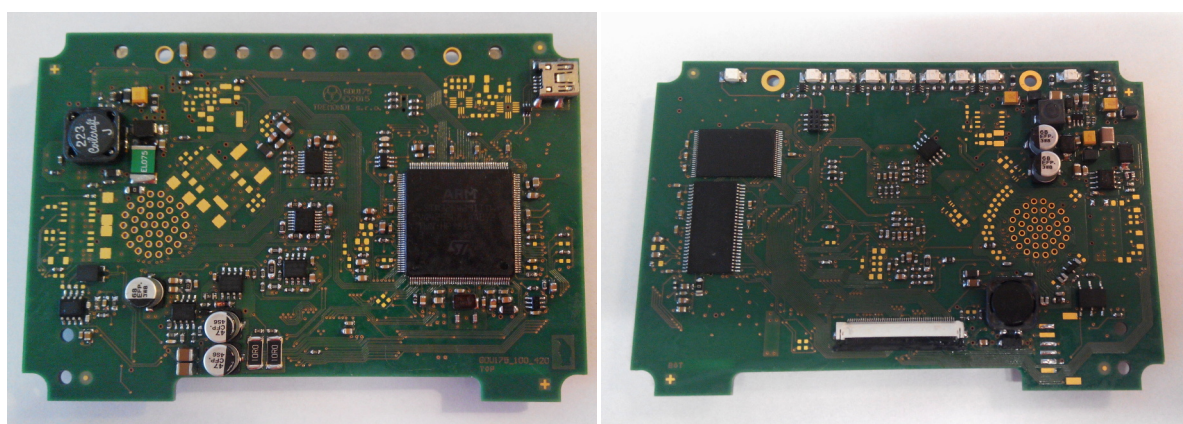
Obr. 3.5: 3D rendery desky



(a) Strana Top

(b) Strana Bottom

Obr. 3.6: Neosazená deska plošných spojů



(a) Strana Top

(b) Strana Bottom

Obr. 3.7: Osazená deska plošných spojů

4. Řídicí firmware

Základní funkce řídicího softwaru přístrojové desky může být shrnuta jako nekončený cyklus příjmu dat ze sběrnice CAN, jejich zpracování a převodu na výstupní data a zobrazení na LCD. Pro snazší rozšiřování funkcí přístrojové desky bylo žádoucí navrhnout celý systém dostatečně univerzálně, aby vyhověl všem možným požadavkům uživatele bez nutnosti externích zásahů do zdrojového kódu. Firmware má proto pevně definovány jen určité konstanty, jako jsou například adresy sektorů Flash paměti, do kterých jsou ukládány údaje o grafických podkladech. Ostatní uživatelská data v podobě počtu obrazovek a jejich zdrojových dat, obrázků, fontů, nastavení filtrace příchozích zpráv na sběrnici CAN, atp. se doplňují až následně dle požadavků uživatele.

4.1 Systémová nastavení

Základní konfigurace periférií mikrokontroléru je orientována na maximální výkon a tudíž minimální čas nutný například ke zpracování příchozích dat. S ohledem na plánované rozšíření funkce zařízení o přenosy po USB byly systémové hodiny mikrokontroléru nastaveny na 168 MHz. Této frekvenci bylo dále přizpůsobeno časování paměti a obnovovací frekvence LCD.

4.1.1 Paměti

Součástí přístrojové desky jsou dvě externí paměti - SDRAM jakožto obrazový buffer pro LCD a NAND Flash paměť pro ukládání uživatelských zdrojů. Obě paměti jsou k mikrokontroléru připojeny pomocí pinů vyhrazených interní periférii *Flexible memory controller* (FMC). Ta kromě obou použitých pamětí podporuje také paměti typu NOR Flash, PSRAM a PC CARD.

SDRAM

Integrovaný obvod **IS42S16400J** o velikosti 64 MBit, popsany již v podkapitole 3.1, je k FMC připojen 16 bitovou datovou a 12 bitovou adresní sběrnici. Řídicí frekvence

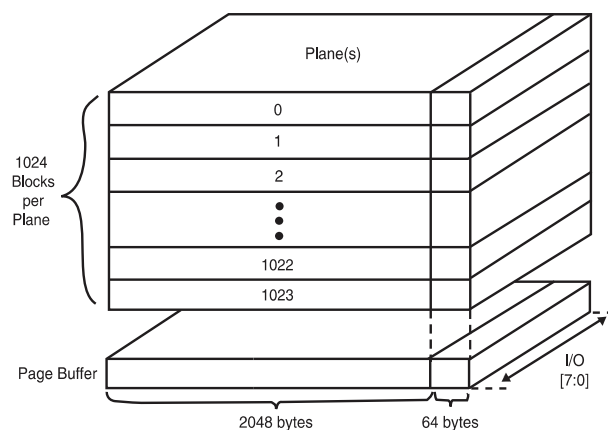
byla nastavena na polovinu systémové frekvence mikrokontroléru, tedy 84 Mhz, neboť paměť samotná podporuje maximálně 143 MHz. Další nastavení, jako jsou prodlevy pro načítání adresy a pro obnovu (refresh) paměťových buněk a především startovací sekvence nastavující všechny automatizované procesy, byly vzhledem k použití kompatibilní paměti a mikrokontroléru převzaty z oficiálních zdrojových kódů a upraveny pro nižší frekvenci.

Práce s pamětí po dokončení inicializace odpovídá práci s interní SRAM v mikrokontroléru. Lze tedy využít všechny dostupné paměťové funkce od čtení a zápisu libovolného objemu dat na jakoukoliv adresu až po datové přesuny pomocí DMA (*Direct memory access*), čímž se velmi zjednodušuje další správa grafických zdrojů.

NAND Flash

Paměť **MT29F1G08** je s mikrokontrolérem, potažmo FMC, propojena 8 bitovou sběrnicí sloužící pro přenos jak dat, tak i adresy. Už z konstrukční podstaty paměti vyplývá, že inicializace i přístup do paměťového prostoru se budou od SDRAM výrazně odlišovat. Zatímco v paměti typu SDRAM musí být obsah paměťových buněk periodicky obnovován, Flash paměť si svůj obsah zachovává i po odpojení napájení. Daní za tuto vlastnost je vyšší komplikovanost zápisu nových dat, neboť není možné zápisem nastavit paměťové buňce \log_2 (dodat elektrický náboj), pouze jej odstranit. Zápis dat je tak ve většině případů uvozen mazáním odpovídající stránky paměti.

Organizace použité 1Gbit varianty je 1024 bloků po 64 stránkách, každá stránka má velikosti 2048 B + 64 bajtů tzv. ECC paměti (*Error-correcting code memory*), která, jak už název napovídá, po aktivaci v mikrokontroléru zabezpečuje opravu případných chyb zápisu do paměti. Lepší představu o organizaci paměti poskytuje obrázek 4.1.



Obr. 4.1: Organizace Flash paměti (převzato z [15], str. 16)

Inicializace
FMC se omezuje na nastavení několika časových konstant (např. zpoždění mezi nastavením a validitou adresy, případně příkazu).

Data odesílaná na adresně - datovou sběrnicí jsou rozlišována pomocí logických úrovní signálů *Command latch enable* (CLE) a *Address latch enable* (ALE), nastavovaných vždy spolu s odesílanými daty. Každý z příkazů podporovaný pamětí je definován 1 až 4 speciálními hodnotami, které jsou odesílány ve specifických fázích datového přenosu.

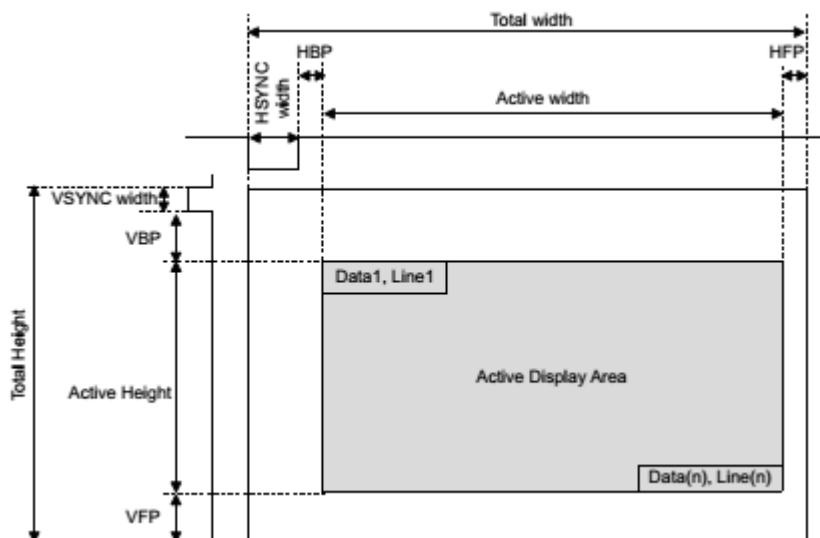
Zároveň je nezbytné nastavit adresu, a to opět podle povahy požadované činnosti - mazání bloku vyžaduje pouze číslo daného bloku, zatímco čtení nebo zápis dat vedle čísla stránky dovoluje nastavit i offset v jejím rámci, čímž lze přistupovat až k jednotlivým bajtům.

Knihovna pro práci s Flash pamětí byla sestavena částečně za užití oficiálních zdrojových kódů pro vývojovou desku STM3210E-EVAL [19]. Převzata byla proto i většina definic a maker, doplněn byl offset zápisu a čtení a také funkce pro načítání dat z Flash pomocí DMA.

4.1.2 LCD

Úvodní nastavení displeje **COG-T430V2100-L2** se dělí na několik fází. Nejdříve je nakonfigurován samotný hardware, tedy piny pro plnou barevnou škálu RGB888 a řídicí signály DotCLK, VSYNC, HSYNC a DE, a dále konfigurace polarit těchto signálů a rozměrů obrazu (rozlišení, šířka vertikální a horizontální synchronizace, vertikální a horizontální zatemňovací impulsy). V druhém kroku jsou nastaveny vlastnosti vrstev a ve třetím DMA pro přenos velkých objemů dat.

Inicializace hardwaru vždy začíná časováním. Všechny parametry zde použité lze nalézt již u standardu VGA, rozdílem je nicméně hodinový signál DotCLK namísto pevného časování pomocí časových intervalů. Všechny důležité parametry shrnuje následující obrázek:



Obr. 4.2: Konfigurace časování LCD (převzato z [16], str. 478)

Vertikální a horizontální synchronizační pulzy VSYNC a HSYNC dávají displeji čas přepnout ukazatel z konce jednoho řádku (sloupce) na následující. Standard VGA tuto prodlevu využíval k přesunu zobrazovacího paprsku, LCD pouze přepíná adresní ukazatel.

Z toho důvodu jsou oba pulzy dlouhé pouze v řádu jednotek hodinových pulzů.

Parametry *Front porch* (FP) a *Back porch* (BP) se v konfiguraci nachází opět spíše z historických důvodů. Slouží jako zpoždění mezi odesláním validních dat a synchronizačního pulzu (FP), respektive mezi synchronizačním pulzem a novými daty (BP).

Rozměry *Active height* a *Active width* se odkazují již na konkrétní rozlišení displeje, které činí v případě přístrojové desky 480×272 obrazových bodů.

Konkrétní konfiguraci parametrů displeje shrnuje následující tabulka:

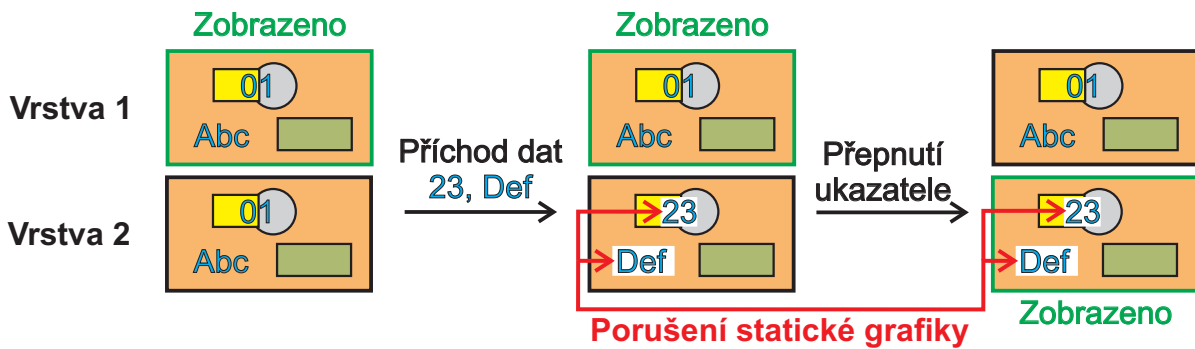
Parametr		Min.	Typ.	Max.	Jednotka
Frekvence DCLK		5	9,18	12	MHz
HSYNC	Kompletní perioda	490	531	605	DCLK
	Perioda zobrazení		480		DCLK
	Back Porch	8	43		DCLK
	Front Porch	2	8		DCLK
	Šířka pulzu	1			DCLK
VSYNC	Kompletní perioda	275	288	335	Horizontální řádek
	Perioda zobrazení		272		
	Back Porch	2	12		
	Front Porch	1	4		
	Šířka pulzu	1	10		

Tab. 4.1: Doporučené časování LCD TFT (převzato z [16], str. 15)

Výše uvedené údaje ukazují, že většina parametrů může být nastavena na velmi nízké hodnoty a displej bude stále fungovat, což bylo v rámci testování ověřeno i prakticky. Jedinou podmínkou správné funkčnosti je přizpůsobení hodnoty kompletní vertikální a horizontální periody zbývajícím parametrům.

Řízení zobrazování dat na displeji obsluhuje interní periferie mikrokontroléru LTDC, jejímž nakonfigurováním se celý proces redukuje prakticky pouze na editaci zobrazovaných dat. Mezi její nejdůležitější vlastnosti patří 24bit paralelní sběrnice pro přenos barev z palety RGB888, maximální rozlišení displeje 1024×768 a podpora práce s vrstvami. Poslední zmíněný bod v sobě konkrétně zahrnuje 2 vrstvy definované odkazem do paměti a jednou jednobarevnou vrstvou sloužící jako pozadí, přičemž je možné libovolně měnit průhlednost jednotlivých vrstev a vytvářet tak různé kombinace.

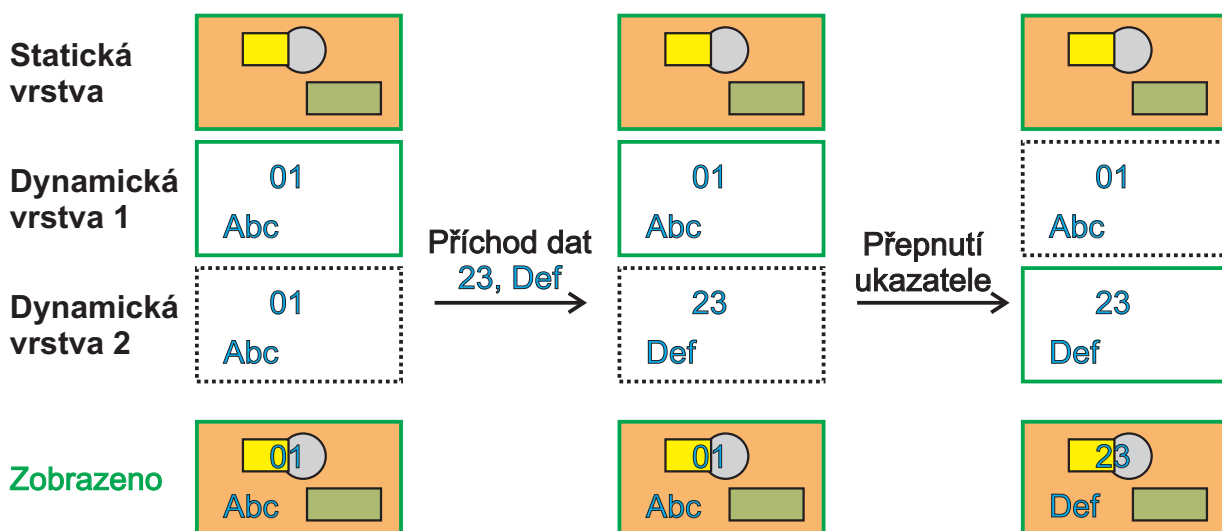
První variantou, která byla použita v počátcích projektu, je oddělení vrstev nastavením jejich průhlednosti tak, aby byla vždy jedna zobrazována (žádná průhlednost) a do druhé (maximální průhlednost) byla zapisována nová data. Výsledný systém má na první pohled největší výhodu v minimální systémové zátěži, neboť tento zobrazuje data z paměti bez dalšího zpracování a musí řídit přístup maximálně do dvou míst externí paměti. V průběhu vývoje firmwaru a testovaných grafických podkladů se nicméně ukázala slabina, kterou nejlépe vystihuje obrázek 4.3.



Obr. 4.3: Přepis dat na oddělených vrstvách

Uživatelská grafika se sestává jednak ze statických prvků (obdélníky, čáry, obrázky), jednak z prvků dynamických, kterými jsou hodnoty přijaté ze sběrnice CAN. Je-li přijatá hodnota odlišná od předcházející, je samozřejmě nezbytné starou z obrazu vymazat a nahradit ji novou, což by při statické grafice, která je komplexnější než jednodílná barva, znamenalo znovu ji rekonstruovat čtením z paměti Flash, tedy narušovat datový tok na sběrnici periferie FMC přepínáním paměti. Systém se neosvědčil právě v otázce univerzality, a proto byl nahrazen následujícím.

Druhá varianta sice vyžaduje větší zapojení interních funkcí mikrokontroléru a je také paměťově náročnější, přesto se osvědčila snadnou modifikovatelností dynamických dat a stále vysokou rychlostí práce. Shrnuje ji následující obrázek:



Obr. 4.4: Přepis dat s klíčováním vrstev

Periferie LTDC je nakonfigurována s oběma vrstvami v nulové průhlednosti, avšak se zapnutým klíčováním barev (*Color keying*), které je velmi dobře známé z filmové produkce jako „zelené plátno“. V případě přístrojové desky jsou tedy využity vždy 2 vrstvy - v jedné se nachází statická grafika, ve druhé, vyplněné klíčovací barvou, jsou rozmisťovány

dynamické prvky, přičemž mikrokontrolér pak obě vrstvy automaticky spojí do jedné a odešle na displej. Změna dat v dynamické vrstvě znamená v tomto případě smazání starých klíčovací barvou a náhradu novými hodnotami. Zároveň byl modifikován i celkový náhled na vrstvy, protože periferie samotná pracuje pouze se dvěma ukazateli do paměti. Není nicméně problém je měnit dle okamžité potřeby. Z toho důvodu byly v paměti zachovány 2 dynamické vrstvy a dále úložiště vrstev statických (pozadí).

Celý systém je ve výsledku charakterizován následujícími body:

- **Paměťová náročnost:** 2 dynamické a až 16 statických vrstev
- **Princip funkce:** barevné klíčování vrstvy s dynamickými prvky na vrstvu se statickou grafikou
- **Editace dynamických dat:** změny prováděny ve druhé (právě neviditelné) dynamické vrstvě, po dokončení je přepsán ukazatel klíčované vrstvy na oblast s novými daty
- **Přepínání obrazovek:** přepsáním ukazatele do oblasti příslušející zvolené obrazovce (okamžitá změna)

4.1.3 Sběrnice CAN

Sběrnice *Controller Area Network* byla poprvé představena firmou Robert Bosch GmbH v roce 1986, v roce 1993 pak vznikl její ISO standard [20]. Jedná se o multi-master sběrnici, ve které jsou zprávy vysílány principem *broadcast*, tedy do všech připojených zařízení najednou. Standard ISO definuje u CAN dvě nejnižší vrstvy - fyzickou a spojovou (linkovou). Mezi základní vlastnosti fyzické vrstvy sběrnice patří maximální přenosová rychlost až 1 Mbit/s a dvouvodičové diferenciální zapojení (CANH a CANL) se 120Ω terminací na obou koncích. Ve spojové vrstvě je obstaráváno kanálové kódování, detekce chyb a zabezpečení dat a prioritizace vysílání.

Pro výše uvedené vlastnosti je sběrnice masivně využívána v automobilovém průmyslu. V současných aplikacích jsou její parametry stále dostačující, není však vhodná pro nasazení v systémech typu *Drive by wire*, neboť jednotlivé uzly nemají garantován přístup na sběrnici (na rozdíl od sběrnice FlexRay). V úvodu každého přenosu totiž probíhá arbitráž, během které je díky drátovému součinu upřednostněn vždy uzel s nižší hodnotou identifikátoru rámce, a tedy vyšší prioritou. V samotném rámci se vedle systémových bitů (SOF, RTR, atd.) nachází až 8 datových bajtů. Důležitou vlastností je, že vzhledem k přenosu typu broadcast nemusí identifikátor rámce vyjadřovat adresu uzlu, nýbrž může sloužit k identifikaci odesílaných dat a je buď 11, nebo 29 bitů dlouhý.

Mikrokontrolér použitý v přístrojové desce podporuje CAN ve verzi 2.0A i 2.0B

aktivně, je tedy schopen jak přijímat, tak i odesílat rámce s rozšířeným 29bit identifikátorem. Umí také pracovat až na maximální rychlosti 1 MBit/s a poskytuje uživateli 3 schránky pro odesílaná data, 2 zásobníky pro příjem dat a až 28 filtrů pro interní filtraci, než jsou data předána k dalšímu zpracování. Komunikační rychlost byla nastavena na maximum, časování bylo přizpůsobeno hodinovému signálu o frekvenci 42 MHz, který je na periférii připojen. Filtraci příchozích zpráv lze libovolně nakombinovat z maskování a přímého srovnávání identifikátorů. Propojení s fyzickou vrstvou obstarává obvod **MCP2551** [21].

4.2 Správa zdrojů

Přijetí dat ze sběrnice CAN a jejich úspěšný průchod filtrací jsou prvním z několika kroků, které na konci vedou k zobrazení nové hodnoty na displeji. Veškeré zdroje využitě v tomto procesu jsou pro mikrokontrolér předem neznámé, a tudíž systém musí při jejich ukládání respektovat stanovený řád. Podle typu zdroje je buď na předem definovaném místě v paměti umístěna hlavičková tabulka s ukazateli dále do paměti, nebo se zdroje nacházejí přímo na předdefinovaném místě.

4.2.1 Grafické podklady

Znakové sady a obrázky jsou mezi paměťmi rozděleny tak, aby přístup k nim minimálně narušoval činnost displeje. Zatímco kompletní data jsou uložena v externí SDRAM (více o důvodech je uvedeno v podkapitole 4.3.1), jejich hlavičky se pro rychlé prohledávání nacházejí přímo v interní SRAM mikrokontroléru. Každý ze seznamů obsahuje ve své struktuře počet prvků spolu s kontrolním součtem tohoto počtu jakožto jisticím prvkem, že je seznam správný. Následují již samotné hlavičky, z nichž každá nese adresu podkladu v SDRAM a dále jeho kontrolní součet. Nejedná se však o CRC v klasickém tvaru, nýbrž o kontrolní součet označovaný jako Adler-32 [22]. Důvodem volby je především snadný výpočet součtu již během zpracování dat v PC softwaru bez nutnosti zavádění dalších procesů jako například u klasického CRC.

Výpočet součtu Adler-32 vyžaduje 16bit proměnné A (výchozí hodnota 1) a B (výchozí hodnota 0) a pole datových bajtů D_1 až D_n . Průběh výpočtu je následující:

$$\begin{aligned}
 A &= 1 + D_1 + D_2 + \dots + D_n \pmod{65521} \\
 B &= (1 + D_1) + (1 + D_1 + D_2) + \dots + (1 + D_1 + D_2 + \dots + D_n) \pmod{65521} = \\
 &= n \cdot D_1 + (n - 1) \cdot D_2 + (n - 2) \cdot D_3 + \dots + D_n + 1 \pmod{65521}
 \end{aligned}$$

Z rovnice plyne, že při každé iteraci je k proměnné A přičtena hodnota právě zpracovávaného bajtu a k B pak současná hodnota A. Po každé iteraci jsou obě proměnné vyděleny největším prvočíslem, které lze do 16bit proměnné zapsat. Po dokončení výpočtu na všech bytech už zbývá jen vytvořit samotný součet:

$$\text{Adler-32} = 65536 \cdot B + A$$

Výsledek splňuje díky povaze výpočtu jedno z důležitých kritérií kontrolních součtů - při malé změně na vstupu dojde k velké změně na výstupu. Pro různé obrázky, případně znakové sady je tak v této podobě dostatečně robustní.

Výše uvedený kontrolní součet slouží pro identifikaci grafických prvků v rámci celého systému. Před každým jejich použitím jsou nejdříve dohledávány podle součtu a až při nalezení shody dále využity. Systém je tak odolný vůči chybám načítání z externí paměti, a tedy i nečekaným pádům a resetům z důvodu přístupu do nesprávných míst paměti.

4.2.2 Proměnné a jejich zpracování

Paket přijatý ze sběrnice CAN musí v první řadě obdržet kontext, tedy jaké veličiny se v něm nacházejí, kolik bajtů každá z nich zabírá, s jakou endianitou a o jaké vstupní typy se jedná. Následuje přepočítání na reálnou hodnotu a nakonec převedení na displej. Z výčtu vyplývá, že každý signál je definován nemalým množstvím doplňujících informací, které musí být v zájmu zachování určité jednoty sepsány stejným systémem. V praxi je používán standard CanDB, v diplomové práci byl od zadavatele požadován vlastní formát, rozšiřující obecný standard o další možnosti a nastavení.

Tvorba struktury proměnných v paměti byla podřízena jednak jednotlivým krokům zpracování příchozích dat, jednak požadavkům zadavatele na podporované funkce. Část parametrů není dostupná pro uživatelské konfigurace, neboť musí být sladěna s nastavením řídicí jednotky přítomné v autě. Editace je tak možná pouze v oblasti grafické interpretace výstupu.

První část obsahuje vše nezbytné pro zařazení přijaté hodnoty do systému, tedy číslo stránky, na které se proměnná nachází, CAN identifikátor spolu s offsetem v rámci 8B datového paketu, endianita, kde je podporován jak Little Endian standardizovaný v zařízeních Intel, tak i Big Endian od Motoroly, dále bajtová délka datového paketu a vstupní typ. Již v tomto místě se definiční tabulka začíná rozdělovat, neboť různé typy zpráv vyžadují různý způsob zpracování nebo dodatečné údaje.

Podporované vstupní typy jsou následující:

- **Signed a Unsigned:** Jedná se o běžné hodnoty, které jsou před odesláním na displej matematicky zpracovány. Mohou také sloužit jako zdroj pro všechny výstupní textové a grafické typy.
- **Float:** Data, zapsaná dle standardu IEEE 754 a přenášená vždy v 4B dávkách, jsou před zpracováním převedena na numerickou hodnotu. Podobně jako předchozí kategorie jsou dále zpracovávána.
- **RAW:** Speciální typ nesoucí už dle svého názvu data v „surovém“ tvaru. Jelikož nepodléhá žádné matematické úpravě, je sice možné jej využít pro přenos celočíselných hodnot, přesto slouží především pro diagnostické účely, kde může každý z bitů sloužit jako logický indikátor např. poruchových stavů.
- **ASCII:** Některé konfigurace řídicích jednotek odesílají namísto poruchových nebo jiných informačních zpráv ve formě kódů hlášení přímo v textovém tvaru. Tento vstupní typ proto neobsahuje žádné dodatečné zpracování a také nemůže sloužit jako zdroj pro další grafické prvky.

Druhá část struktury proměnné se odlišuje pro jednotlivé typy tak, jak jsou popsány výše. Je-li podporováno matematické zpracování, pak jsou k dispozici hodnoty pro násobení a posun. Dále je pro všechny vstupní typy kromě textového k dispozici volba výstupního tvaru a také počtu zobrazených desetinných míst. Výstupní typy jsou následující:

- **Decimal:** Standardní výstup v dekadickém tvaru.
Příklad: 123.4
- **Hexadecimal:** Výpis v šestnáctkové soustavě uvozený standardním prefixem *0x*.
Odstraňuje desetinnou část čísla.
Příklad: 0x1234
- **Binary:** Výstup v dvojkové soustavě, může sloužit pro diagnostické účely.
Odstraňuje desetinnou část čísla.
Příklad: 0b0001001000110100
- **Float:** Výpis s užitím vědecké notace.
Příklad: 1.23e4

Mezi další parametry patří část opět společná všem signálům, a to uživatelem zvolený font písma, pozice prvku na obrazovce a barva textu. Posledním parametrem hlavní části je prvek rozhodující o tom, bude-li výstup ponechán ve tvaru určeném dosavadními parametry, tj. běžný text, nebo má-li se jednat o některý ze speciálních grafických výstupů.

V rámci této práce je podporován jediný speciální výstup.

Specifická část definuje výstup jako výčtový typ (enumerace). Hodnota, získaná z libovolného vstupního typu kromě textového, je porovnávána s intervaly, z nichž každý má přiřazen vlastní výstupní text. Zobrazena proto není přímo hodnota, ale pouze řetězec znaků zastřešující odpovídající interval. V rámci definice je pak uvedeno, kolik intervalů výčtový typ obsahuje, jaké jsou limitní hodnoty a jaký text má být zobrazen. K dispozici je také prostor pro jeden řetězec, který je zobrazen, nevyhovuje-li vstupní hodnotě žádný z intervalů.

4.3 Princip funkce

4.3.1 Start programu

Úspěšnost úvodní inicializace řídicí desky je kromě správného nastavení hardwaru závislá také na dostupné uživatelské konfiguraci. V první fázi jsou z externí Flash paměti načteny sektory obsahující hlavičkové tabulky jednotlivých prvků uživatelského rozhraní, tj. pro fonty, obrázky, stránky (obrazovky), filtry CAN a proměnné. Každý takto načtený sektor je postupně rozdělen na jednotlivé prvky a uložen do interní SRAM mikrokontroléru. Je-li v průběhu načítání dané kategorie dat detekována nesrovnalost, jako například cílová adresa v paměti mimo povolené meze, nevhodné hodnoty určitých parametrů nebo neodpovídající kontrolní součet počtu prvků v kategorii, je kategorie považována za špatně zapsanou a její počet prvků nastaven na 0. Zařízení je pak i nadále schopno pracovat, ovšem s chybějícími zdroji z dané kategorie.

Fonty a obrázky mají v interní SRAM uloženy svůj kontrolní součet, zdrojovou adresu ve Flash a cílovou adresu do SDRAM a datový objem. Na rozdíl od původního plánu načítat grafické zdroje v běhu programu přímo z Flash jsou tyto ihned po startu přeprogramovány do SDRAM, odkud jsou později načítány. Důvodem je zde limitace hardwarem, zmíněná i v eratech [23] použitého mikrokontroléru, která efektivně zabraňuje používat na jedné datové sběrnici dvě externí paměti, aniž by se navzájem ovlivňovaly. Také internetové diskuze výrobce zmiňují stejný případ čtení externí Flash způsobující artefakty na LCD. Tato chyba by měla být opravena v poslední revizi čipu, který se však vzhledem k přetrvávajícím problémům s obrazem v přístrojové desce nenachází.

Obrazovky jsou pro zvýšení celkové rychlosti běhu firmwaru staženy ihned po startu do SDRAM, ve které se pak pouze přepínají ukazatele na ně. V této fázi se také mažou prostory pro dynamické vrstvy.

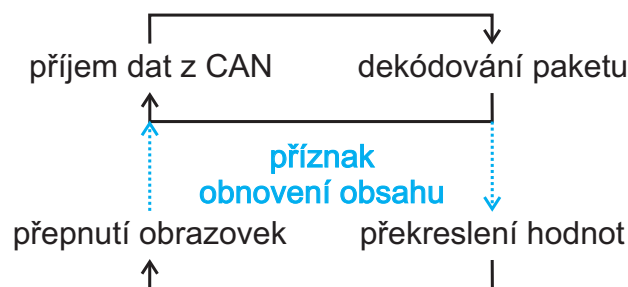
CAN filtry jsou ve Flash zapsány jako soubor údajů, sestávající se z definice typu filtru, srovnávaném identifikátoru a jeho případné masce, či přiřazení k některému

CAN zásobníku pro příjem dat. Data se nikam nezapisují, nýbrž se rovnou používají k inicializaci struktur CAN periferie.

Proměnné jsou z paměti načteny jako celek do jednoho velkého prostoru v interní SRAM. Kromě uvedeného prostoru je v SRAM založeno ještě jedno pole, ve kterém jsou uschovány pouze nejnnutnější údaje pro zpracování dat ze sběrnice CAN a proměnné pro uchování nových a původních hodnot. Tato redundance je v současné verzi nepotřebná, slouží nicméně jako základ pro verzi budoucí, ve které bude již vyřešený problém s paralelním provozem obou externích pamětí.

4.3.2 Hlavní smyčka

Dokončení inicializace hardwaru a grafických zdrojů je následováno přesunem do hlavní části, která již realizuje funkční řetězec ztvárněný na následujícím obrázku:



Obr. 4.5: Vizualizace hlavní programové smyčky

Pakety přijaté ze sběrnice CAN jsou ihned po přijetí zpracovány dle definiční tabulky v interní SRAM. Dojde tak k přidělení výše zmiňovaného kontextu, tedy k rozdělení 8B datového prostoru na jednotlivé proměnné, úprava pořadí bajtů, je-li pro danou proměnnou definována endianita typu MSB (Motorola), a dále porovnání nově přijaté a předešlé hodnoty této proměnné. Jsou-li data odlišná, nastaví se ve struktuře proměnné příznak značící změnu, a tedy povolení k dalšímu zpracování při grafickém přepisu.

Změna hodnoty některé z proměnných není pro mikrokontrolér jedinou podmínkou, aby provedl obnovu dat na displeji. Souběžně s hlavní smyčkou běží v mikrokontroléru interní časovač, který pravidelně každých 50 ms nastaví příznak pro kontrolu aktuálnosti dat. V dalším průchodu smyčkou pak algoritmus projde pole hlaviček proměnných a začne přepisovat grafická data, kde se uplatní zbytek parametrů definujících jednotlivé proměnné. Jako první je „surová“ hodnota získaná z CAN paketu převedena na hodnotu odpovídající parametru vstupního typu. U běžných signálů buď obdrží tvar se znaménkem (Signed), nebo je ponechána v původním tvaru (Unsigned, RAW), případně převedena funkcí *memcpy* z 4B pole na tvar s plovoucí desetinnou čárkou (Float). Dalším krokem

je násobení a posun hodnoty předem definovanými konstantami, je-li tato operace přípustná pro daný vstupní typ. Výsledek je nakonec zpracován do podoby řetězce znaků formátovaného dle předdefinovaného výstupního typu - základem je tvar DEC pro dekadický výstup, dále diagnostické tvary (HEX, BIN) ořezávající z vypočteného čísla desetinnou část a na závěr vědecký tvar.

Zapsání nové hodnoty na displej je uvozeno smazáním hodnoty dosavadní. Za tímto účelem je proto v hlavičce každé proměnné prostor jak pro novou, tak stávající hodnotu. V této fázi je také využit zbytek parametrů jako je pozice počátku textu na obrazovce, barva a font.

4.4 Měřicí funkce

Zapojení přístrojové desky do systému může být uživatelem vedeno dvěma směry: buď jako pasivní prvek zobrazující příchozí data, nebo jako aktivní prvek, který kromě vizualizace příchozích dat dokáže také měřit vnější veličiny a odesílat je po sběrnici CAN dalším zařízením. Ať už je volba uživatele jakákoliv, přístrojová deska by vždy měla mít přehled minimálně o vlastních řídicích veličinách. Z toho důvodu nesmí programu chybět zpětná vazba v podobě měření napětí a teplot na kritických místech systému. Specifikace zařízení obsahuje kromě těchto kontrolních měřicích bodů také linky na konektoru s externími signály frekvence, teploty a napětí. Ač zatím není k dispozici, v dalších etapách vývoje se počítá s odesíláním hodnot všech vstupních signálů na sběrnici CAN. K dispozici tak budou hodnoty externích napětí, teplot a frekvencí, výstupní napětí výkonových spínačů a velikost napětí na vnější referenci. Naopak, odesílány nebudou interní měřené veličiny, napájecí napětí interních logických systémů, napětí a proud podsvícením displeje. Jedinou výjimkou bude teplota DPS, s jejímž odesíláním je počítáno také.

4.4.1 Interní napětí

Snímání napětí na interních referencích je zatím v systému přichystáno pro pozdější využití, kdy budou údaje sloužit pro autodiagnostiku zařízení. Jiná situace panuje u měření napětí různých periférií, jako jsou například výkonové vstupy a výstupy, napětí na externí referenci či proud tekoucí podsvícením LCD. Zde již má měření vliv na funkci zařízení, neboť mikrokontrolér všechny tyto periferie přímo ovládá a má tedy možnost je buď úplně odpojit (spínače), nebo alespoň omezit na úplné minimum (podsvícení displeje).

Zapojení na desce plošných spojů je realizováno děličem doplněným o kondenzátor umístěný co možná nejbližší vstupu mikrokontroléru kvůli potlačení případného vnikajícího rušení, neboť některé z těchto linek vedou po značné délce desky přímo od

zdrojů napětí, potažmo referencí.

Interní 12bit analogově-digitální převodník využívá externí referenční napětí 2,5 V. Spouštění převodu je vzhledem k informativnímu charakteru měřených veličin prováděno jednorázovými odměry vyvolávanými v rámci hlavní programové smyčky. Do budoucna lze uvažovat o průběžné kontrole všech kritických interních napětí a teplot s využitím kontinuálního sběru dat, neboť A/D převodník tuto funkci nabízí spolu s možností automatického přepínání kanálů a zápisu do paměti pomocí DMA.

4.4.2 Interní teplotní snímače

Měření teploty v různých oblastech desky slouží k omezování maximálního proudu podsvícením LCD, které je tepelně závislé. Na desce plošných spojů se nacházejí celkem 2 NTC termistory **NCP18XW222J03RB** [24] v SMD pouzdru, přičemž první je umístěn u mikrokontroléru a druhý v prostoru napájecích zdrojů. Vyhodnocení hodnoty odporu je na rozdíl od externích čidel popsanych dále řešeno děličem napětí, v němž je k čidlu připojen do série rezistor o odporu 1,3 kΩ. Napájení děliče je shodné s referencí A/D převodníku v mikrokontroléru, získaná hodnota X_{ADC} pro přímo vyjadřuje poměr velikostí odporu rezistoru a NTC čidla, jehož odpor je následující:

$$\frac{R_{NTC}}{R_{NTC} + 1,3 \text{ k}\Omega} = \frac{X_{ADC}}{2^{12}} \Rightarrow R_{NTC} = \frac{X_{ADC} \cdot 1,3 \text{ k}\Omega}{2^{12} - X_{ADC}}$$

Výpočet teploty termistoru z hodnoty jeho odporu je možný více způsoby, záleží tak především na podkladech, které jsou k dispozici. Katalogový list uvádí v případě použitého termistoru nominální odpor $R_{t=25^{\circ}\text{C}} = 2200 \Omega$ a parametr β platný v rozsahu teplot 25 - 85 °C o hodnotě 3980. Přepočet je definován vztahem

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} \ln \left(\frac{R_{NTC}}{R_0} \right),$$

kde R_0 je nominální odpor při 25°C, T_0 nominální termodynamická teplota 298,15 K a R_{NTC} odpor vypočítaný ze změřeného napětí.

4.4.3 Externí napěťové vstupy

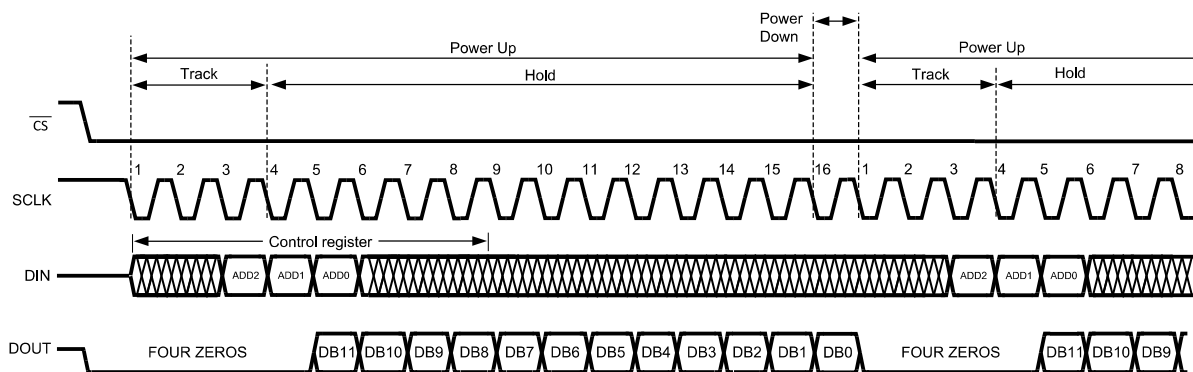
Konektor Deutsch rozšiřuje počet signálových linek přiváděných z vozidla do přístrojové desky z 5 na takřka 40. Jedním z prvků, jež se u zařízení tohoto typu často vyskytují, jsou vstupy pro měření napětí, v případě popisované přístrojové desky se jedná o rovných 8 analogových linek. Měření napětí je realizováno externím 12 bitovým analogově-digitálním převodníkem **ADC128S022** [25] od firmy Texas Instruments.

Důvodů pro uvedené řešení je několik: dodatečná ochranná bariéra před přepětím přivedeným na vstupy, vysoká rychlost a snadná konfigurovatelnost průběžného měření a v neposlední řadě také absence dostatku analogových vstupů na mikrokontroléru, z nichž mnohé se nacházejí na stejných pinech jako vstupy časovačů či unikátnější signálové linky, například pro řízení externích paměťových modulů. Referenční vstup použitého A/D převodníku je připojen na 5 V, což je stejná hodnota jako na výstupu externí reference. Komunikace s mikrokontrolérem probíhá za využití napětí 3,3 V.

Nastavení mikrokontroléru pro práci s A/D převodníkem zahrnuje následující parametry:

- **Frekvence SCK:** použita předdělička 64, pak $f_{SCK} = \frac{f_{HCLK}}{64} = 2,625 \text{ MHz}$
- **Mód funkce:** plně duplexní
- **Délka datového slova:** 16 bitů
- **Polarita SCK při neaktivitě:** logická 1
- **Aktivní hrana:** sudá, tj. vzestupná hrana

Po inicializaci SPI periferie v mikrokontroléru a přerušení je komunikace samotná již značně přímočará, neboť převodník akceptuje jediný příkaz a to na výběr měření jednoho z 8 dostupných vstupů. Pro lepší představu slouží následující diagram:



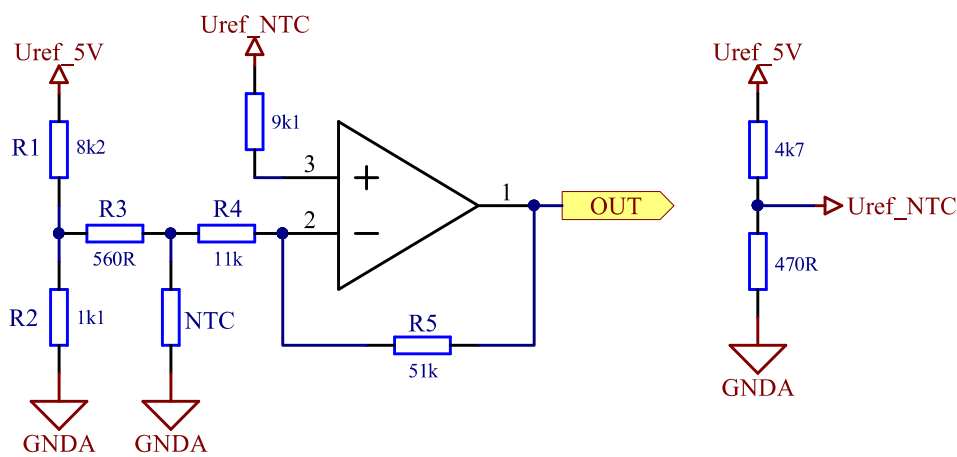
Obr. 4.6: Diagram komunikace s externím A/D převodníkem (převzato z [25], str.7)

Měření není v současné verzi prováděno, systém je nicméně připraven pro budoucí periodické snímání s využitím přerušení, kde dokončením měření je hodnota daného kanálu uložena do paměti a rovnou spuštěno měření napětí na dalším vstupu, přičemž další zpracování a přepočítání hodnot na napětí provede mikrokontrolér, až bude potřeba.

4.4.4 Externí teplotní čidla

Dvě z linek dostupných na rozšiřujícím konektoru jsou vyhrazeny pro externí NTC termistory. Zapojení měřicího obvodu je oproti interním teplotním čidlům odlišné, neboť se o vnější vstup, který musí být odolný vůči nesprávnému zapojení a chybám. Měření je také nezbytné provádět přesně, nikoliv pouze informativně jako u interních snímačů na DPS.

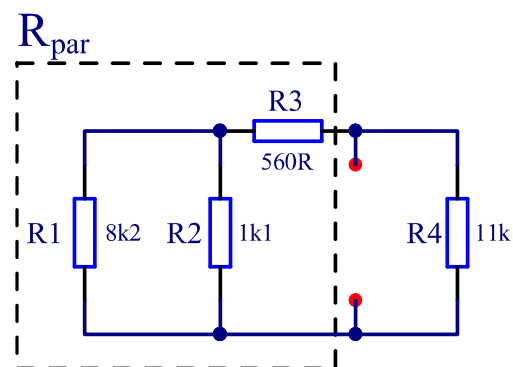
Obvod měřicí odpor externího NTC čidla využívá běžné zapojení invertujícího zesilovače napětí s posunutou nulou podle schématu na obrázku 4.7.



Obr. 4.7: Zapojení externích NTC čidel

Měřený signál OUT lze spočítat z napětí na čidle a známého zesílení a offsetu zesilovače. Komplikovanější je však výpočet napětí na NTC čidle, protože na něj mají vliv pomyslné Pull-up rezistory R_1 a R_4 . Za použití Théveninovy věty bylo proto vypočítáno napětí U_x a vnitřní odpor R_x náhradního zdroje.

Jako první se vypočítá vnitřní odpor. V tomto okamžiku se z obvodu odstraní zátěž a zdroje jsou nahrazeny jejich vnitřními odpory (místo napěťových zkrat a místo proudových rozpojení).



Obr. 4.8: Schéma zapojení pro výpočet vnitřního odporu

Odpor je pak počítán mezi červeně zvýrazněnými body na obrázku 4.8 a je definován následovně:

$$R_{par} = \left(\frac{1}{R_1} + \frac{1}{R_2} \right)^{-1} + R_3$$

$$R_x = \left(\frac{1}{R_{par}} + \frac{1}{R_4} \right)^{-1} \approx 1343,09 \Omega$$

Svorkové napětí náhradního zdroje se vypočítá tentokrát již s oběma zdroji zapojenými a je definováno vztahem

$$U_x = \frac{\frac{R_2}{R_1+R_2} \cdot U_{ref} - U_{ref(NTC)}}{R_{par} + R_4} \cdot R_4 + U_{ref(NTC)} \approx 0,5747 V$$

Stanovením U_x a R_x bylo zjištěno vše nezbytné, finální náhradní schéma se nachází na obrázku 4.9. Napětí na termistoru je pak následující:

$$U_{NTC} = \frac{NTC}{R_x + NTC} \cdot U_x.$$

Zbývá už pouze určit závislost výstupního napětí na napěťovém úbytku na termistoru:

$$U_{OUT} = U_{ref(NTC)} - \frac{R_5}{R_4} \cdot (U_{NTC} - U_{ref(NTC)}).$$

Výstup zesilovače je připojen přes DP filtr (RC článek) na analogový vstup mikrokontroléru. Vzhledem k externí referenci A/D převodníku je tak měření výstupního napětí zesilovače omezeno na pásmo 0-2,5 V, respektive minimální a maximální hodnotu odporu NTC čidla

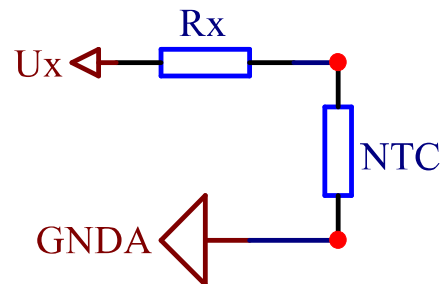
$$R_{NTC}(U_{OUT} = 0 V) = 33577 \Omega$$

$$R_{NTC}(U_{OUT} = 2,5 V) = 31,99 \Omega.$$

Zesilovač je primárně navržen pro NTC termistor [26] od firmy Magneti Marelli s nominálním odporem

$$R_{(25^\circ C)} = 2252 \Omega,$$

není však problém zapojit libovolné jiné NTC s nominálním odporem mezi 2 k Ω a 3 k Ω . Jeho katalogový list obsahuje pouze převodní tabulku hodnot $T \rightarrow R$, pro účely výpočtu



Obr. 4.9: Výsledné náhradní zapojení zesilovače

bylo proto nezbytné křivku aproximovat.

Popis křivky byl proveden **Steinhart-Hartovou rovnicí** [27] definovanou následujícím vztahem:

$$\frac{1}{T} = A + B \cdot \ln(R_T) + C \cdot (\ln(R_T))^3.$$

Tabulka hodnot použitá k aproximaci je následující:

t [°C]	R_t	t [°C]	R_t	t [°C]	R_t
10	4481,48	50	811,17	100	152,96
20	2812,75	60	560,30	110	115,03
25	2252,00	70	394,33	120	87,69
30	1814,66	80	282,85	130	67,70
40	1199,42	90	206,37	140	52,88

Tab. 4.2: Tabulka hodnot NTC termistoru

Výpočet koeficientů A, B, C pro výše zmíněnou rovnici je veden jako řešení soustavy 3 rovnic o 3 neznámých v následujícím tvaru:

$$\begin{bmatrix} 1 & \ln(R_1) & \ln^3(R_1) \\ 1 & \ln(R_2) & \ln^3(R_2) \\ 1 & \ln(R_3) & \ln^3(R_3) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \frac{1}{T_1} \\ \frac{1}{T_2} \\ \frac{1}{T_3} \end{bmatrix}$$

Aproximace byla provedena v bodech 10, 60 a 110 °C a jim náležejících hodnotách odporu. Konstanty byly výpočtem stanoveny následovně:

$$A = 1,471025 \cdot 10^{-3}$$

$$B = 2,376417 \cdot 10^{-4}$$

$$C = 1,054102 \cdot 10^{-7}$$

Vypočtené konstanty byly poté dosazeny do Steinhart-Hartovy rovnice. Pro srovnání aproximace s výchozími tabulkovými hodnotami byla použita inverzní rovnice (k nalezení v [28])

$$R = e^{\sqrt[3]{(-m+\sqrt{m^2+n^3})} + \sqrt[3]{(-m-\sqrt{m^2+n^3})}}$$

kde

$$m = \frac{A - \frac{1}{T}}{C}$$

$$n = \frac{B}{3C}.$$

Srovnání aproximace s hodnotami dodanými výrobcem bylo provedeno dosazením tabulkových termodynamických teplot do předcházející rovnice. Výsledné srovnání zobrazuje následující tabulka 4.3.

t [°C]	Rt [Ω] (výrobce)	Rt [Ω] (aproximace)
10	4481,480	4481,480
20	2812,748	2813,443
25	2252,000	2252,062
30	1814,662	1814,344
40	1199,415	1199,361
50	811,170	811,148
60	560,298	560,300
70	394,325	394,667
80	282,851	283,083
90	206,373	206,493
110	115,032	115,030
130	67,695	67,663

Tab. 4.3: Srovnání tabulkových hodnot a aproximace

Praktické testování měření nebylo bohužel provedeno, neboť se nepodařilo získat zamýšlený NTC snímač. Opět ovšem platí, že algoritmus je ve firmware připraven a může být kdykoliv zařazen do funkčního cyklu.

4.4.5 Frekvenční vstupy

Další funkce dostupná při použití rozšiřujícího konektoru jsou dva vstupy pro snímání impulzů z Hallových sond, přestože okruh využití této funkce není natolik široký jako například u měření externích analogových signálů. Nabízí se snímání otáček motoru a kol vozidla, které je sice vcelku běžně zpracováváno a rozesíláno po sběrnici CAN řídicí jednotkou motoru, nejedná se však o pravidlo.

Ve spolupráci se zadavatelem byl již ve fázi obvodového návrhu sestaven algoritmus pro měření vstupní frekvence v rozsahu 1 Hz - 10 kHz. Základem snímání každého ze signálů jsou 2 periferie mikrokontroléru připojené na výstup tvarovače signálu z Hallové sondy - jeden čítač, jež zaznamenává počet vzestupných hran signálu, a jeden časovač pro měření celkové doby uplynulé od první měřené vzestupné hrany signálu.

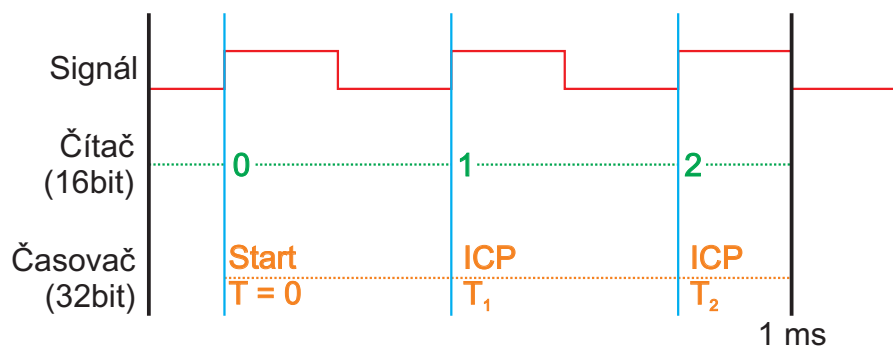
Princip funkce je následující:

1. **Začátek:** čítač impulzů je resetován, časovač vynulován a zastaven
2. **První vzestupná hrana:** čítač impulzů zvyšuje svoji hodnotu, časovač si připojuje interní hodinový signál o frekvenci 84 MHz

3. **Další vzestupné hrany:** čítač impulzů zvyšuje svoji hodnotu, časovač zaznamenává funkcí *Input Capture* (ICP) do proměnné svou hodnotu (celkový čas od první hrany)
4. **Uplynutí 1 ms:** je-li hodnota čítače impulzů vyšší než 0, pak je z počtu příchozích impulzů a poslední zapsané hodnoty časovače vypočítána frekvence a poté je spuštěno nové měření (návrat na 1. bod algoritmu), v opačném případě program pokračuje následujícím bodem
5. **Vyčkávání na vzestupnou hranu nebo uplynutí 1 s:** podle typu události, která nastane dříve, se liší také reakce algoritmu:
 - **Vzestupná hrana:** software dále nevyčkává na další vzestupné hrany a rovnou vypočítá frekvenci signálu z hodnoty časovače pro jeden impulz
 - **Uplynutí 1 s:** na vstup buď není přiveden žádný signál, nebo je jeho frekvence nižší než 1 Hz, vyhodnocena je proto frekvence 0 Hz

Ať už nastane první nebo druhá situace, program se vrací zpět na bod 1).

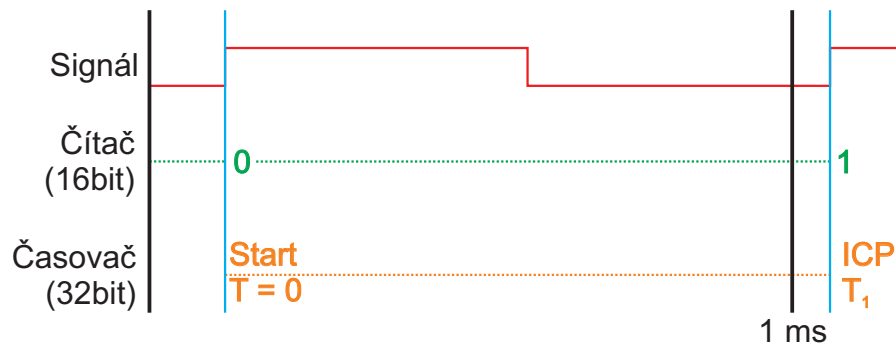
Pro lepší názornost poslouží časové průběhy na obrázcích 4.10 a 4.11.



Obr. 4.10: Měření frekvence $f \geq 1 \text{ kHz}$

Obrázek 4.10 ilustruje rozdíl mezi interním zapojením časovače a čítače. Zatímco čítač je vyjma doby výpočtu frekvence aktivní trvale, časovač se aktivuje až s první vzestupnou hranou nového měření. Při frekvencích vyšších než 1 kHz jsou pak oběma periferiím odpojeny zdroje hodinového signálu po uplynutí 1 ms, neboť tou dobou je již hodnota čítače vyšší než 0 a je tedy z čeho frekvenci počítat.

Obrázek 4.11 se soustředí na situaci, ve které uplynutí doby 1 ms nehraje roli, protože vstupní signál od startovací vzestupné hrany ještě nedokončil celou periodu. Jakmile však tato dorazí, čítači i časovači je odpojen hodinový signál, vypočítá se frekvence a cyklus začíná nanovo.

Obr. 4.11: Měření frekvence $f < 1\text{ kHz}$

Vizualizace situace se signálem o frekvenci nižší než 1 Hz již není třeba, v takovém případě je po uplynutí 1 s bez příchodu druhé vzestupné hrany odpojen hodinový signál časovače, který je zároveň vynulován. Mikrokontrolér se dále vrací na začátek smyčky, kde opět vyčkává na přichozí vzestupnou hranu.

Výhoda výše popsaného způsobu měření tkví především v použití minimální doby 1 ms, během níž algoritmus vyčkává na přichozí vzestupné hrany. Jednodušší způsob založený na měření délky jedné periody funguje také, v projektu jsou pomocí něj měřeny signály o frekvenci nižší než 1 kHz, potenciálně problematické je však měření vyšších frekvencí. Signál přivedený do mikrokontroléru je sice již předzpracován tvarovačem, ovšem vzhledem k univerzálnosti frekvenčního vstupu nelze vždy předpokládat, že použitá Hallova sonda pracuje s konzistentním výstupem. Je-li tedy u signálů o frekvenci vyšší než 1 kHz změřena doba více period, dochází vlivem průměrování ke snižování chyby měření.

Pokud by byl rozdíl mezi vzorkovací (84 MHz) a předpokládanou maximální frekvencí (10 kHz) o několik řádů menší, docházelo by u více frekvencí k přiřazení jedné hodnoty časovače, čímž by se samozřejmě výrazně zvýšila chyba měření. Pokud je však algoritmus nastaven na minimální dobu měření, detekuje se během ní více period, z nichž až poslední je zatížena chybou, zatímco ostatní jsou měřeny přesně.

5. Lokální uživatelské rozhraní a jeho návrh

Vizualizace dat přijatých po sběrnici CAN, respektive veličin naměřených a zpracovaných přímo mikrokontrolérem, je nejdůležitějším výstupním prvkem zařízení tohoto typu. O to vyšší nároky jsou kladeny nejen na hardwarovou oblast řešení jako je kontrast LCD a jeho dobrá čitelnost za všech světelných podmínek, ale ve velké míře také na vzhled samotného uživatelského prostředí. Důležitým faktorem se tak stává i rozčlenění zobrazovaných veličin do jednotlivých obrazovek a jejich celkové rozmístění nebo nároky na hardware pro co nejnižší vytížení použitého mikrokontroléru.

Na internetu lze najít nepřeberné množství knihoven pro různé architektury mikrokontrolérů a typy displejů, požadavkem zadavatele byl nicméně návrh vlastní grafické knihovny a z ní vycházejícího konfiguračního softwaru. Bylo proto nezbytné definovat jednotlivé typy objektů a podkladů tak, aby byly mezi knihovnou a softwarem přenositelné.

5.1 Standardizace formátů

Systém návrhu a vizualizace grafických prvků obsahuje 3 kategorie objektů, které bylo nezbytné pevně definovat:

- geometrické útvary
- rastrová grafika (obrázky)
- fonty

5.1.1 Geometrické útvary

Vzhledem k zaměření cílového zařízení na rychlost a účelnost jsou podporovány pouze základní útvary - bod, čára, obdélník a kružnice. Všechny jsou definovány výchozím a koncovým bodem (nebo poloměrem), dále tloušťkou čáry a barvou. Důležitým faktem

je, že oba body, respektive střed a poloměr u kružnice, definují vnější rozměry útvaru, tj. zvýšením tloušťky čáry dojde ke zmenšení vnitřního prostoru objektu. Uvedená funkce má především zjednodušit uživateli korekce velikosti při návrhu jednotlivých obrazovek a rastrů.

5.1.2 Rastrová grafika

Obrázek může být definován buď v monochromatickém, nebo standardním RGB tvaru. Systém tyto tvary a další vlastnosti rozlišuje v následující hlavičce, kterou je každý soubor uvozen:

- **Byte 0+1** - šířka
- **Byte 2+3** - výška
- **Byte 4** - barevná hloubka
- **Byte 5+6+7** - objem obrazových dat

Barevná hloubka podává informaci o množství dat nezbytných pro zobrazení 1 pixelu. Podporován je jak monochromatický formát dat, tak i RGB565 a RGB888.

Monochromatické obrázky slouží především jako ikony - typickým příkladem mohou být různé kontrolky automobilu, blinkry. Obrazová data v tomto případě nenesou údaj o barvě bodů, nýbrž pouze o stavu (log.1 - pixel přítomen, jinak log.0). Toto kódování umožňuje výrazně zmenšit objem ukládaných dat, neboť každý bod si nárokuje pouze 1 b, přičemž tyto jsou organizovány do vertikálních bajtů v pořadí „vyšší souřadnice Y pixelu = vyšší index bitu“. S obrázkem nakládá jak knihovna, tak i software jako s průhledným, tj. kde není bod definován, tam je zachováno původní pozadí. Zároveň je také možné libovolně měnit barvu obrázku, což je výhodné právě u kontrolkek ve stavu aktivní / neaktivní.

Barevné obrázky slouží v systému jako statické ukazatele nebo dekorace (loga). Zobrazování je závislé na parametru barevné hloubky, neboť tento rozhoduje o počtu bajtů, které budou z paměti přečteny a dále zakresleny. Zároveň parametr slouží jako indikátor, je-li nezbytné barvy přepočítávat na hloubku nastavenou v systému, nejsou-li tyto dvě shodné.

5.1.3 Fonty

Princip kódování fontů byl převzat z knihovny *u8glib* [29]. Uvedený zdroj obsahuje mimo jiné databázi již vytvořených fontů, čímž se stal vhodným výchozím bodem pro práci se stávajícími a pro návrh nových. Společným prvkem všech znakových sad je shodně

uspořádaná globální hlavička a standardizovaná hlavička a zápis jednotlivých znaků.

Hlavička fontu má konstantní délku 17B a obsahuje jednak údaj pro dekodování jednotlivých znaků, jednak globální informace o obsažených znacích, maximech přesahů v obou vertikálních směrech atp. Pro potřeby přepisování znaků v grafice se zde také nachází informace o „obálce znaku“, tj. obecném prostoru, do kterého lze všechny tyto jednotlivě umístit, aniž by přesahovaly ven. Konkrétní údaje se nacházejí v následující tabulce:

Byte #	Název parametru	Komentář
0	font format	Formát znakové hlavičky (== 0 - plný; == 1 - redukovaný)
1	BOUNDINGBOX width	„obálka znaku“ - šířka
2	BOUNDINGBOX height	„obálka znaku“ - výška
3	BOUNDINGBOX x-offset	„obálka znaku“ - horizontální posun
4	BOUNDINGBOX y-offset	„obálka znaku“ - vertikální posun
5	capital A height	výška znaku 'A'
6	start 'A'	pozice znaku 'A' v datovém poli (MSB)
7		pozice znaku 'A' v datovém poli (LSB)
8	start 'a'	pozice znaku 'a' v datovém poli (MSB)
9		pozice znaku 'a' v datovém poli (LSB)
10	encoding start	první dostupný znak
11	encoding end	poslední dostupný znak
12	descent 'g'	přesah písmene 'g' pod úroveň řádku
13	font max ascent	maximální přesah nad úroveň řádku
14	font min decent	maximální přesah pod úroveň řádku
15	font xascent	výchozí přesah nad úroveň řádku
16	font xdescent	výchozí přesah pod úroveň řádku

Tab. 5.1: Hlavička znakové sady

Data dále pokračují jednotlivými znaky seřazenými vzestupně podle ASCII hodnoty, přičemž jejich hlavička je dekodována právě na základě parametru *font format*. Rozlišují se následovně:

Byte #	Parametr
0	výška
1	šířka
2	datový objem $((š+7)/8) \times v$
3	šířka včetně mezer
4	horizontální posun dx
5	vertikální posun dy

Tab. 5.2: Formát 0 - rozšířený

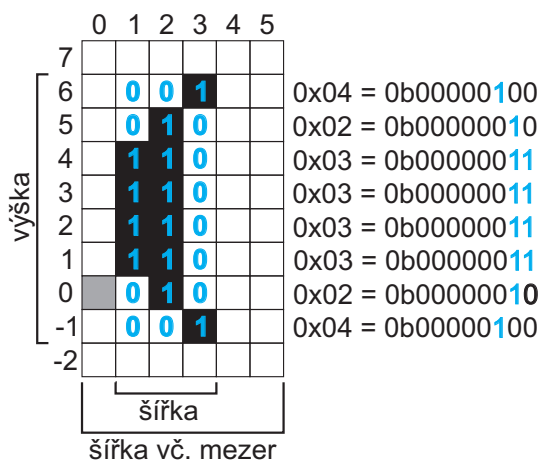
Byte #	Parametr
0, vyšší nibble	horizontální posun dx
0, nižší nibble	vertikální posun dy
1, vyšší nibble	šířka
1, nižší nibble	výška
2, vyšší nibble	šířka včetně mezer
2, nižší nibble	datový objem $((š+7)/8) \times v$

Tab. 5.3: Formát 1 - redukovaný

Hlavička znaku je následována daty o počtu odpovídajícím parametru *datový objem*, přičemž organizovány jsou do horizontální bajtů. Parametry *výška* a *šířka* nicméně informují pouze o nejmenší možné „obálce“, do které lze znak umístit, reálná pozice vůči

řádku je určována až ve spolupráci s vertikálním a horizontálním posunem.

Pro vytvoření lepší představy o významu jednotlivých parametrů slouží obrázek 5.1.



Obr. 5.1: Princip kódování znaku

Font je v tomto případě definován s následujícími základními parametry:

- Formát fontu = 1
- Maximální vertikální přesah nad úroveň řádku = 8
- Maximální vertikální přesah pod úroveň řádku = 2

Znak '4' má pak v hlavičce zapsány následující informace:

- výška = 8
- šířka = 3
- šířka včetně mezer = 6
- datový objem = 8
- dx = 1
- dy = -1

Po dekódování lze vidět, že znak je vůči počátku znázorněnému šedým bodem posunut o 1 px doprava a 1 px dolů. Šířka včetně mezer nabývá významu při zapisování textů, neboť prostým přičítáním těchto hodnot k výchozí horizontální pozici X lze snadno umisťovat další znaky tak, aby respektovaly tzv. *kerning* neboli úpravu mezer mezi specifickými znaky pro zvýšení přívětivosti čtení.

5.2 Grafická knihovna pro LCD

Vývoj grafické knihovny podléhal požadavkům na následující funkcionality:

- kreslení základních geometrických útvarů
- práce s obrázky
- podpora práce s textem s využitím znakových sad
- optimalizace na rychlost
- univerzálnost pro různé typy LCD
- kompatibilita s navrženým grafickým softwarem

Struktura knihovny se dělí mezi práci s hardwarem a samotné grafické funkce.

První část konfiguruje systémové parametry a provádí obsluhu interního LCD řadiče mikrokontroléru, jak je popsán v kapitole 4.1.2. Je možné zde nastavit jednak rozměry displeje, jednak požadovanou barevnou hloubku. Knihovna pak na základě těchto parametrů stanoví rozsahy pro čtení a zápis do SDRAM. Uživatel má dále k dispozici nástroje pro práci s dynamickými vrstvami, konkrétně volbu jedné ze 2 dostupných - zjištění aktuálně aktivní a přepnutí vrstvy na opačnou. Tato funkce pak slouží při zakreslování grafických podkladů, které vždy zapisují do právě neaktivní vrstvy, aby mohly být zobrazeny najednou a nedocházelo k vytváření artefaktů na displeji vlivem okamžitého přepisování zdrojových dat. Nachází se zde také jediná hardwarově orientovaná funkce vyžadovaná grafickými funkcionalitami - nastavení pixelu na požadovanou barvu.

Podporované grafické prvky se vzhledem k orientaci na rychlost omezují na základní geometrické útvary, práci s textem a vkládání obrázků. Uživatel má k dispozici bod, čáru, obdélník a kružnici, vše s možností nastavit barvu a tloušťku čar nebo výplň. Obrázky jsou podporovány ve tvaru monochromatickém s volitelnou barvou nebo barevném, oba dle standardu definovaného v podkapitole 5.1.2.

5.3 Grafický PC software

Software pro kompletní návrh obrazovek uživatelského rozhraní včetně obrazových a textových podkladů byl vyvinut s ohledem na následující požadavky:

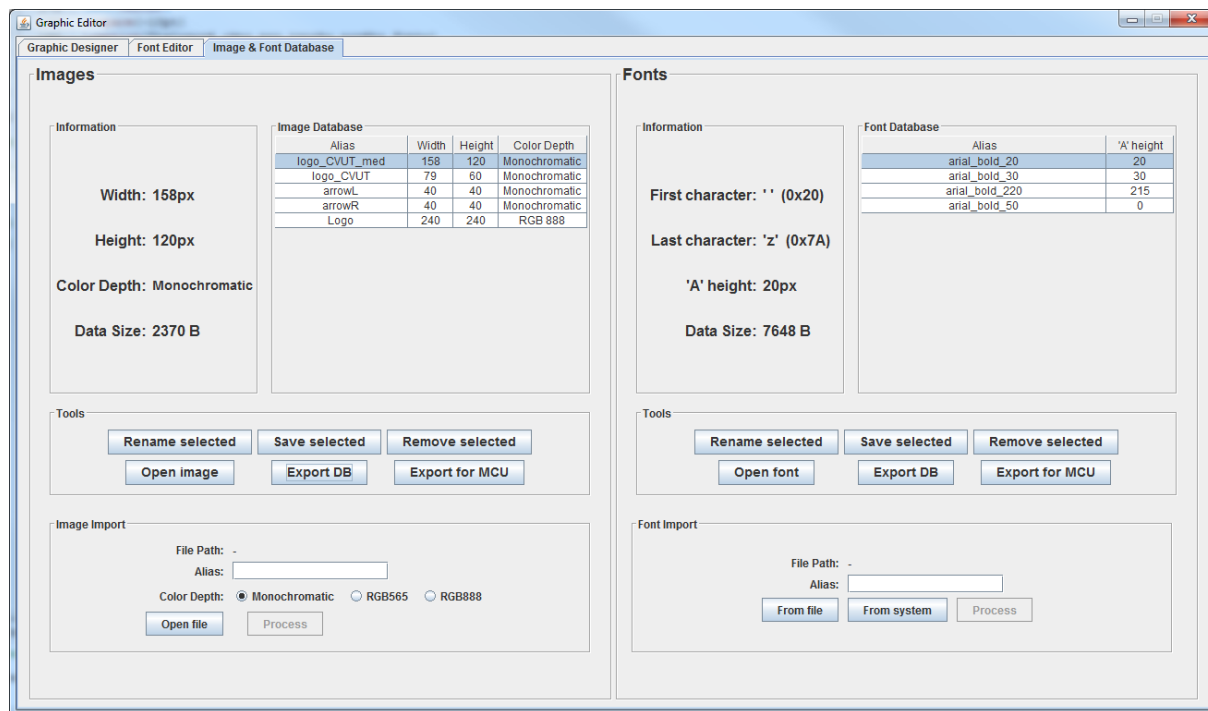
- podpora všech geometrických útvarů z grafické knihovny, respektování standardů definujících obrázky a znakové sady
- „objektový“ návrh obrazovek uživatelského menu, editor rastrových podkladů
- editor obsahu znakových sad včetně návrhu nových znaků a celých fontů
- projektová správa zdrojů - sdružení obrázků a fontů do seznamů společných pro celý projekt
- export obrazovek, obrázků a fontů pro mikrokontrolér

Hlavní ideou editoru grafických podkladů a fontů je práce s materiály již zpracovanými do určitých pseudodatabází sdružujících vždy podklady dané kategorie. Tento přístup byl zvolen jakožto imitace podmínek v mikrokontroléru, kde se také pro jeden identifikátor nachází např. pouze jeden font, a tudíž změna jeho obsahu se projeví všude tam, kde je použit. Z pohledu uživatele to znamená omezení importu podkladů z textových souborů (fonty), potažmo bitmap (obrázky) pouze na záložku softwaru zodpovědnou za správu těchto zdrojů. Editace podkladů je tímto prováděna přímo na zdrojích z databáze a projevuje se tak v celém projektu. Uživateli proto odpadá starost o aktuálnost podkladů v uživatelských obrazovkách, neboť se obnovují automaticky.

Software byl vytvořen v jazyce Java ve vývojovém prostředí NetBeans [30]. Následující podkapitoly popisují jednotlivé záložky softwaru a odpovídající funkcionality.

5.3.1 Správa zdrojů

Záložka Image & Font Database poskytuje uživateli funkce pro import, export a správu obsahu obou databází - grafické a textové. Vzhled obrazovky se nachází na obrázku 5.2.



Obr. 5.2: Vzhled okna správy zdrojů

Jak obrázky, tak fonty mají svůj vlastní seznam se základními údaji jako je název, šířka, výška a barevná hloubka u obrázků, respektive název a výška písmene *A* u fontů. Zvolením libovolné položky se nejen zobrazí dodatečné informace o ní, ale otevře se i možnost správy. Tlačítka „Rename selected” a „Remove selected” provedou přesně tyto úkony, „Save selected” pak uloží danou položku v odpovídajícím formátu do datového souboru. Je zde také možnost stiskem „Open image”, respektive „Open font”, otevřít soubor, kdy je podle typu souboru načtena a přidána do databáze jedna položka nebo rovnou celá sada obrázků nebo fontů. Tlačítko „Export DB” převezme kompletní obsah odpovídajícího seznamu a uloží jej do jednoho souboru, což uživateli šetří čas s otevíráním většího množství jednotlivých souborů ve větších projektech. Nakonec „Export for MCU” převede zvolenou položku podle stanoveného algoritmu na data zpracovatelná mikrokontrolérem. Obrázek je uveden hlavičkou (viz 5.1.2) a dále zapsán v odpovídající barevné hloubce, fontu je zpracována a zapsána hlavička podle tabulky 5.1 následovaná daty jednotlivých znaků. Je-li některý znak ve fontu nepřítomen, nahrazuje jej algoritmus znakem s ASCII hodnotou *0xFF*, čímž dá následně mikrokontroléru najevo, že má pokračovat ve vyhledávání následujícího znaku na další pozici.

Import obrázků

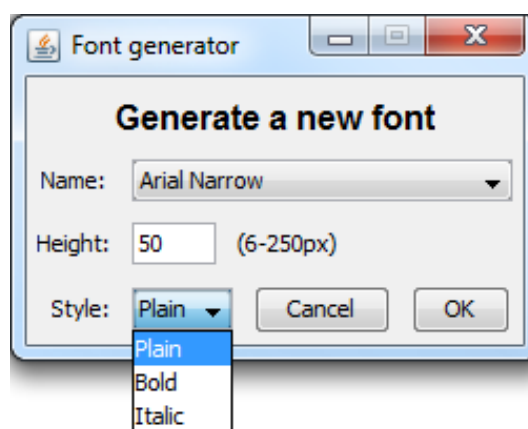
Zařazení nových grafických prvků do databáze je možné v zásadě dvěma způsoby, a to buď ručním vytvořením v editoru, nebo právě importem ze souboru. Software akceptuje soubory s příponou **.bmp*, **.jpg* a **.png*. v libovolné barevné hloubce, které jsou dále zpracovány interní třídou *BufferedImage* a přečteny pixel po pixelu. Již v této fázi dochází k úpravě barevné hloubky podle přepínače: při *RGB* je zkoumaná barva ponechána na původní hodnotě, u *Monochromatic* převádí software barvy na datový typ Boolean systémem

- bílá barva (0xFFFFFFFF) → log.0 (false)
- jiná barva → log.1 (true),

což při návrhu obrazovek uživatelského menu umožňuje libovolně měnit barvu takových obrázků včetně jejich průhlednosti tam, kde se barva nenachází. Uživatel si v tomto kroku musí zvolit ještě název, pod kterým bude grafický podklad vystupovat v rámci projektu, a dále už jen vše potvrdit stiskem tlačítka „Process”.

Import fontů

Přidat nový font do databáze lze dvěma způsoby. První možností je přes textový soubor načítaný tlačítkem „From file”, v němž se nachází font ve formátu čitelném mikrokontrolérem, tedy s hlavičkou a jednotlivými znaky. Software soubor načte, zkontroluje konzistenci dat, uloží si hlavičku a dále seznam objektů tvořený jednotlivými znaky, z nichž každý je dále definován vlastní hlavičkou a samotnými daty. Druhá varianta je přímé vytvoření nového fontu tlačítkem „From system”. Software v tomto případě využívá jako zdroj fontů nabídku operačního systému. Vzhled dialogového okna pro návrh nové znakové sady se nachází na obrázku 5.3.

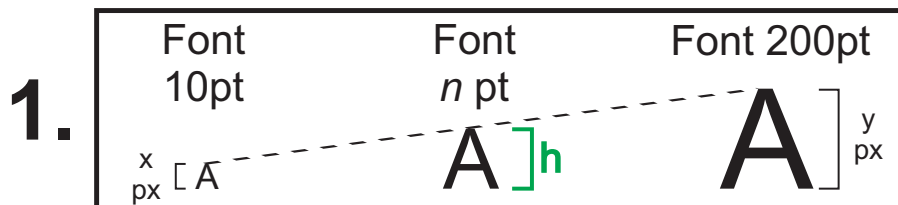


Obr. 5.3: Dialogové okno pro tvorbu nového fontu

Potvrzením volby některého systémového fontu a jeho výsledné výšky v pixelech (jako reference se považuje písmeno 'A') provede software následující kroky:

1. Určení systémové velikosti fontu dle zvolené výšky v px

Jazyk Java umožňuje prostřednictvím interních tříd *BufferedImage* a *Graphics2D* vkládat do virtuální kreslicí plochy kromě jiného také znaky a dále je analyzovat. Jako parametr pro velikost písma nicméně využívá systémovou velikost používanou běžně v textových editorech. To se ovšem pro tvorbu nových znakových sad v aplikaci nehodí, neboť tato systémová velikost není nijak vázána na reálnou velikost v pixelech, a tudíž by uživatel předem nevěděl, jakou zvolit, aby mu text do jeho grafického prostředí zapadl. Namísto toho software převezme požadovanou velikost v pixelech, interně zjistí výšku znaků fontu ve 2 různých systémových velikostech a lineární interpolací dopočítá nejbližší celou systémovou velikost odpovídající zadané výšce v pixelech. Postup dokumentuje následující obrázek:



Obr. 5.4: Výpočet velikosti fontu odpovídající požadované výšce

2. Identifikace základních parametrů každého znaku

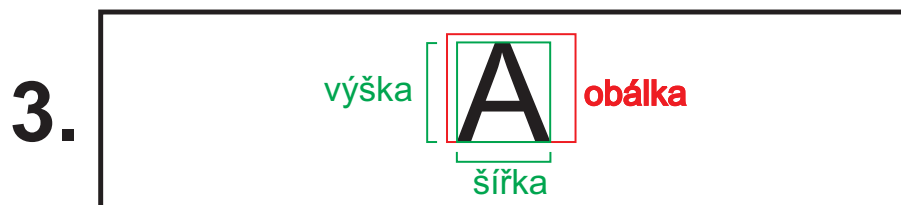
Další z nástrojů jazyka je stanovení hlavních pozičních parametrů zadaného textu. Je-li vložen pouze znak, pak logicky lze o každém zjistit jeho výšku, šířku a vertikální a horizontální posun vůči počátku. Z testování algoritmu však vyplynulo, že všechny výše zmíněné parametry nelze po vyčtení považovat s naprostou jistotou jako pravdivé, zvláště je-li zvolen font v kurzívě. Využívají se tak pouze hodnoty velikosti vertikální a horizontální obálky znaku, ve které je znak dohledán. Proces identifikace dokumentuje následující obrázek:



Obr. 5.5: Hledání důležitých parametrů znaku

3. Získání obrazových dat

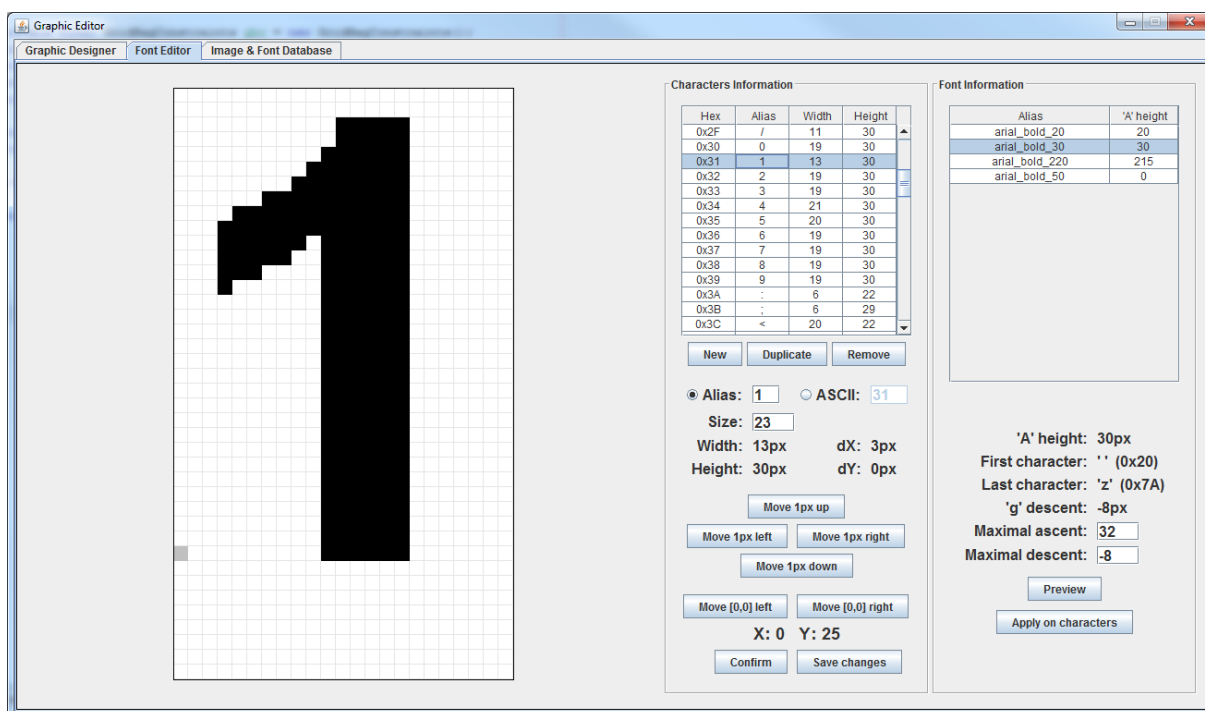
Poslední krok vytvoření nového znaku je převedení virtuálního znaku na data srozumitelná softwaru. Algoritmus nejdříve zjistí vertikální posun, poté výšku, dále horizontální posun a na závěr šířku znaku. Pro lepší představu poslouží následující obrázek:



Obr. 5.6: Generovaná obrazová data

5.3.2 Editor znakových sad

Druhá záložka softwaru slouží jednak pro správu obsahu znakových sad (fontů), jednak k editaci jednotlivých znaků. Pro lepší ilustraci je zde nejdříve podoba záložky nacházející se na následujícím obrázku:



Obr. 5.7: Vzhled editoru fontů a znaků

Seznam v pravé části obsahuje všechny fonty načtené do databáze. Volbou libovolného z nich dojde k zobrazení dodatečných informací jako je první a poslední znak v sadě, popřípadě výška znaku *A* (respektive nejvyššího znaku, není-li písmeno *A* v sadě

přítomno). Uživatel má zároveň možnost změnit maxima přesahů v obou vertikálních směrech, ilustrovaných prostorem nad a pod úrovní šedé tečky (neboli řádku) na obrázku, má-li v úmyslu provádět editaci znaků na grafické úrovni. Tyto 2 hodnoty jsou z hlediska fontu globální, nevyužitý prostor nad a pod znakem objem uložených dat nezvyšuje. Jedná se pouze o pomocné parametry, které mikrokontrolér může využívat například pro výpočet globální obálky textu při jeho mazání.

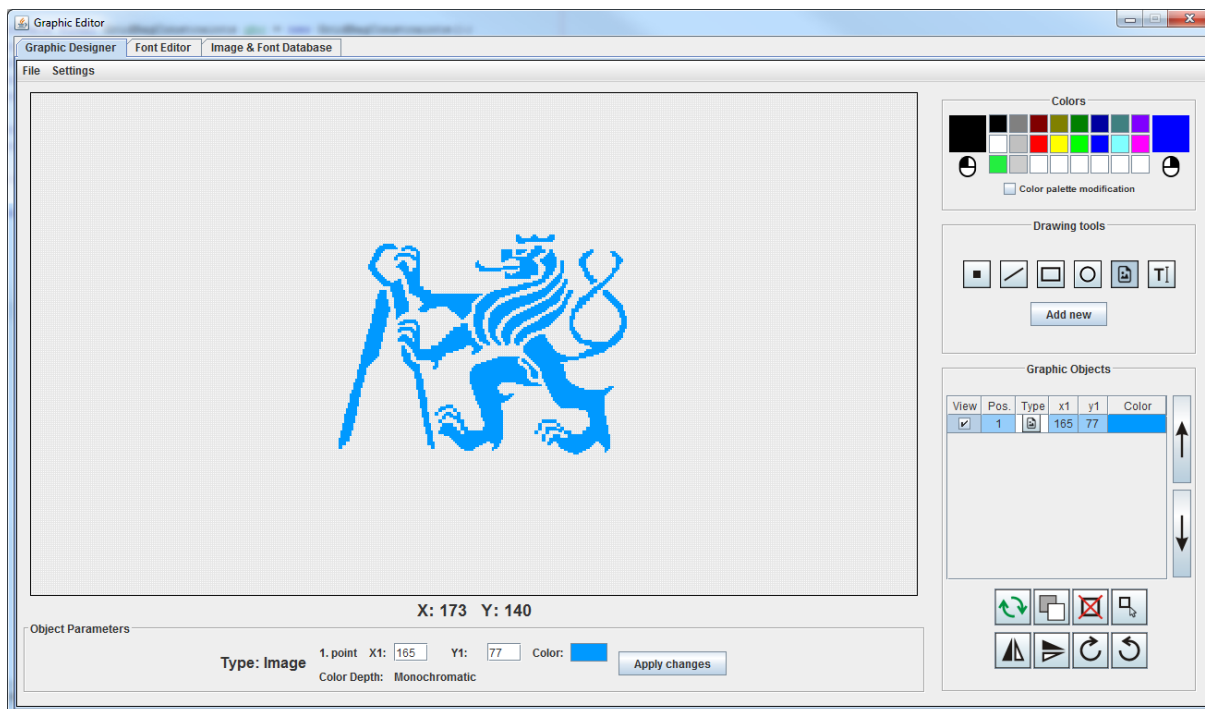
Tabulka uprostřed zobrazuje všechny v sadě dostupné znaky, a to včetně základních údajů - výšky a šířky bez mezer, aliasu a pozice znaku v ASCII tabulce. K tabulce náleží také 3 tlačítka: stiskem „New” je do seznamu přidán nový prázdný znak bez aliasu a výchozí šířkou 1 px., „Duplicate” znak zkopíruje, což značně zjednodušuje tvorbu fontů s diakritikou, a „Remove” zvolený znak odstraní. Níže pod tlačítka se nachází prostředky pro editaci konkrétních znaků. Je možné změnit jak alias, tak i ASCII hodnotu znaku, je-li vyžadováno umístění například některého speciálního znaku do úseku mezi hodnotami 0x80 a 0xFF, a dále šířku kreslicího prostoru, který kromě znaku zahrnuje i mezery před a za ním. Dále jsou k dispozici tlačítka pro posun znaku ve všech směrech v rámci daného prostoru a nakonec posun počátku doleva a doprava. Tato funkce se uplatní v případech, kdy má znak být pro lepší čitelnost umístěn před počátek, jinými slovy blíže předešlému písmenu. Všechny úpravy, které se přímo dotýkají rozměrů kreslicí plochy, lze buď zobrazit jako náhled stiskem tlačítka „Confirm”, respektive potvrdit a uložit všechny úpravy tlačítkem „Save changes”. Rozdílem ve funkci je, že náhled slouží pro utvoření lepší představy o tom, jak bude kreslicí plocha například široká. Pokud by tedy uživatel prostor zúžil tak, že by se některé pixely již nezobrazovaly, nedojde ke ztrátě dat a lze v úpravě šířky nadále pokračovat. Je-li však editace potvrzena a uložena, generují se obrazová data znovu ze zobrazeného prostoru, a tudíž může dojít ke ztrátě části dat.

Editovaný znak se nachází v levé části záložky. Stiskem levého tlačítka je pixelu nastavena černá barva, pravým tlačítkem pak bílá. Každou editaci pixelů je nezbytné potvrdit stiskem tlačítka „Save changes”, v opačném případě budou úpravy ztraceny.

5.3.3 Grafický editor

Nejdůležitější část softwaru - editor grafických podkladů pro jednotlivé stránky uživatelského rozhraní přístrojové desky - čerpá ze všech součástí zmíněných v podkapitolách výše. Klíčovým funkčním požadavkem pro návrh uživatelského rozhraní softwaru byla plná kompatibilita s grafickou knihovnou popsanou v podkapitole 5.2. Mezi další požadavky patřilo ukládání a načítání grafických podkladů v různých formátech, objektově orientovaný editor obrazovek a rastrový editor podpůrných grafických podkladů a dále standardní funkce pro lepší ergonomii ovládání jako jsou klávesové zkratky či modifikace prvků tahem myši, podobně jako v jiných vektorových programech.

Vzhled celého okna včetně všech prvků se nachází následujícím obrázku 5.8.

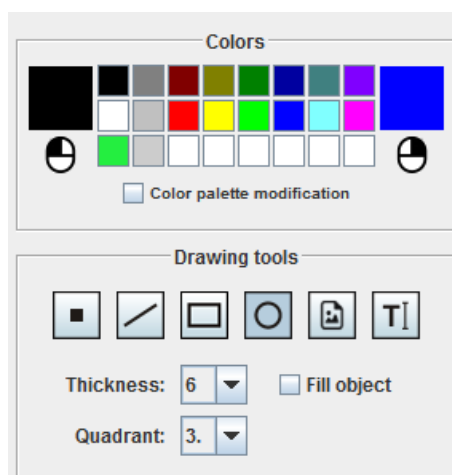


Obr. 5.8: Vzhled hlavního editoru

Jednotlivé panely a jejich funkce jsou popsány v následujících podkapitolách.

5.3.3.1 Panel grafických prvků

Tato součást editačních prostředků je společná pro práci v objektivém i rastrovém editoru. Dostupné prvky shrnuje následující obrázek:



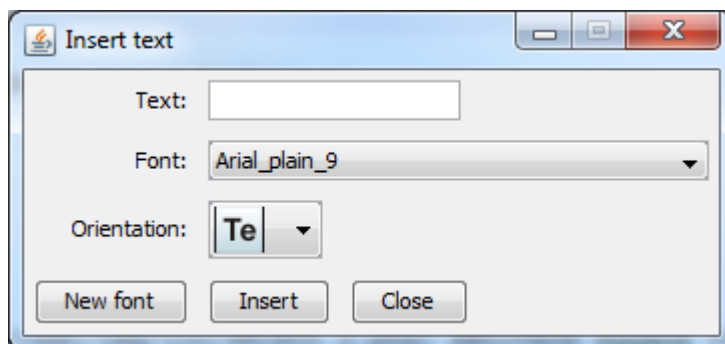
Obr. 5.9: Společný panel nástrojů

Barevná paleta umožňuje uživateli nastavit barvu odděleně pro levé a pravé tlačítko myši. Stiskem zaškrtačovacího políčka a kliknutím na některé z políček ve spodní řadě pak lze paletu doplnit o vlastní barvy. Tato uživatelská paleta barev je při ukládání obrazových dat vždy přítomna, aby nebylo při příští práci nutné znovu všechny barvy definovat.

Paleta grafických prvků pracuje ve shodě s vytvořenou grafickou knihovnou. Již před vložením každého z nich je možné nastavit mu všechny nezbytné vzhledové vlastnosti, tj. podle povahy geometrického útvaru barvu, tloušťku čáry, výplň. Kružnice má navíc ještě parametr pro zobrazení pouze určitého kvadrantu (viz obr. 5.9). Prvky nastavující uvedené parametry se podle typu geometrického útvaru zobrazují přímo pod přepínači.

Vložení obrázku do kreslicí plochy předchází jeho výběr z nabídky bitmap přítomných v obrazové databázi. Je-li obrázek v plné barevné škále RGB, je vložen bez další možnosti nastavení, monochromatické mají nicméně k dispozici ještě volbu barvy ploch reprezentovaných na obrázku černou barvou. Bílé plochy se u těchto obrázků stanou průhlednými.

Přidání textu na kreslicí plochu má oproti bitmapám širší možnosti nastavení. Okno pro editaci a vložení textu je následující:

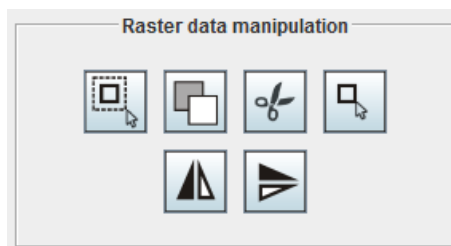


Obr. 5.10: Dialogové okno pro přidání nového textu

Kromě definice samotného textového obsahu a seznamu dosud vytvořených fontů má uživatel vždy k dispozici tlačítko pro vytvoření nových fontů. Může také zvolit orientaci textu.







5.3.3.2 Panel nástrojů - rastrový editor

Editace nebo tvorba nové bitmapy se v mnohém podobá běžným softwarům, např. jako je MS Paint. Jednou zakreslená grafika se stává permanentní součástí celku, dodatečná manipulace tak není na rozdíl od objektového editoru možná. Všechny dostupné funkce se nacházejí na následujícím obrázku:



Obr. 5.11: Panel nástrojů rastrového editoru

Funkce jednotlivých tlačítek jsou následující:

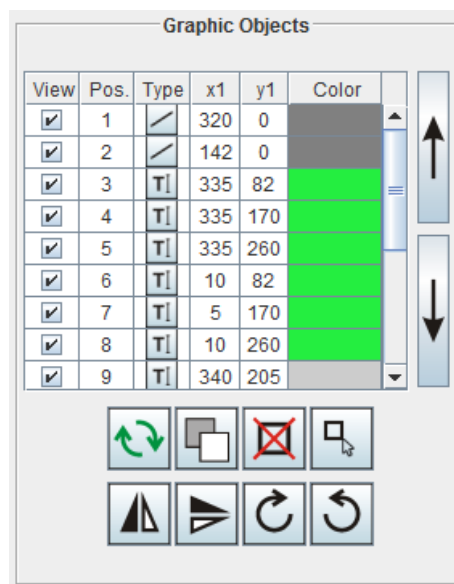
-  Aktivace funkce výběru. Lze označit libovolnou oblast kreslicí plochy.
-  Vytvoření kopie zvolené plochy. Označená oblast je zachována.
-  Výřez zvolené plochy. Označená oblast je smazána.
-  Aktivace funkce vložení zkopírované nebo vyříznuté plochy.
-   Převrácení zvolené oblasti nebo grafiky podle vertikální/horizontální osy.

Jednotlivým datovým funkcím jsou přiřazeny i klasické klávesové zkratky, tedy kopie (Ctrl+C), výřez (Ctrl+X) a vložení (Ctrl+V).

Vytvořené grafické podklady je možné zpracovat do více výstupních tvarů. První možností je uložení do obrazové databáze, kde bude následně připraven pro použití v rámci obrazovek rozhraní. Druhou variantou je uložení do souboru v podobě datového kontejneru, který obsahuje vše nezbytné pro načtení do jiného projektu bez nutnosti importování dat a definování jejich barevné hloubky. Samozřejmostí je export do textového souboru, odkud lze data přímo zkopírovat do zdrojového kódu pro mikrokontrolér. Ať už je volba uživatele jakákoliv, vždy má na výběr, bude-li výstup v barevné paletě RGB, nebo pouze monochromatický.

5.3.3.3 Panel nástrojů - objektový editor







Návrh obrazovek uživatelského rozhraní se od tvorby běžné rastrové grafiky liší právě celkovou modifikovatelností v průběhu práce. V tomto případě je každý vložený prvek definován jako objekt obsahující všechny nezbytné parametry nejen pro vykreslení v rámci PC softwaru, ale i pro případný export. Uživatel má zde totiž možnost buď opět převést vše na rastr, nebo mu software poskytne soubor příkazů včetně přepínání barev a fontů. Po vložení do zdrojového kódu pak bude vše vykresleno tak, jak si navrhl předem na PC. Vzhled panelu nástrojů je na následujícím obrázku:



Obr. 5.12: Panel nástrojů objektového editoru

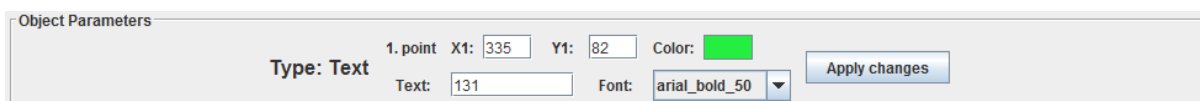
Vyobrazená tabulka slouží jako seznam všech objektů přítomných na dané obrazovce, přičemž uživatelé informuje o pozici prvního bodu, barvě, typu objektu a dále prioritě, se kterou bude objekt vykreslen. Zde platí, že čím nižší číslo v kolonce priority, tím „výše“ se objekt bude nacházet. Změna je pak možná šipkami po pravé straně tabulky.

Tlačítka níže pod tabulkou mají následující funkce:

-  Překreslení všech objektů. Používá se například při změně priorit prvků.
-  Duplikace zvolených objektů a vložení na stejnou pozici jako jejich předlohy.
-  Odstranění zvolených objektů následované překreslením obrazovky.
-  Připnutí zvolených objektů ke kurzoru. Slouží k manipulaci.
-  Převrácení objektu podle vertikální/horizontální osy.
-  Rotace objektu po/proti směru hodinových ručiček.

5.3.3.4 Panel editace objektů

Návrh obrazovek uživatelského rozhraní se neobejde bez dodatečných modifikací rozmístění jednotlivých prvků, jejich velikosti, barvy či dalších parametrů. Každý z objektů nacházející se v tabulce na obrázku 5.12 tak může projít kompletní změnou v rámci parametrů daného objektu. Za tímto účelem se pod kreslicí plochou nachází panel zobrazující buď všechny dostupné parametry objektu, je-li zvolen pouze jeden, nebo všechny parametry společně zvolené skupině objektů. Konkrétní podoba panelu je následující:



Obr. 5.13: Panel editace objektů

V tomto případě se jedná o editaci jednoho z textových objektů a jak z obrázku plyne, uživatel má k dispozici všechny parametry jak vzhledové, tak i poziční.

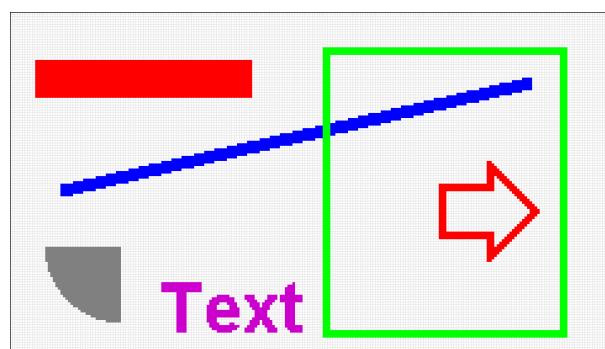
5.3.3.5 Export grafických podkladů

Ukládání obrazovek a bitmap závisí jednak na typu vytvářené grafiky, jednak na dalších záměrech uživatele. Jako první se nabízí speciální datové soubory obsahující všechny informace nezbytné k rekonstrukci obrázku při příštím spuštění programu. Vždy jsou zpracovány rozměry a barevná hloubka, u bitmap je pak uložena kreslicí plocha a u uživatelských obrazovek všechny jim náležející grafické objekty. Tyto soubory jsou pro uživatele nečitelné a zjednodušují program

načítání již jednou vytvořených obrazových dat.

Druhou variantou je export dat ve tvaru vhodném pro vývojáře. Výstupem je tentokrát textový soubor, který obsahuje buď obrazová data statické grafiky

odpovídající definici uvedené v podkapitole 5.1.2, nebo příkazy grafické knihovny, jedná-li se obrazovku uživatelského rozhraní.



Obr. 5.14: Testovací obrazovka pro demonstraci možností výstupu

Software v takovém případě zapisuje vše včetně příkazů pro přepnutí barvy a fontu u textů.

Pro lepší představu byla z dostupných grafických prvků sestavena obrazovka, viz obrázek 5.14. Dále byl zvolen export do souboru ve formě příkazů z grafické knihovny. Celá obrazovka je tak definována následujícím kódem:

```
1 gColDisp_DrawLine(20, 70, 206, 28, 0x1F, 5);
2 gColDisp_DrawFrame(124, 14, 220, 128, 0x7E0, 3);
3 gColDisp_DrawBox(10, 19, 95, 33, 0xF800);
4 gColDisp_DrawDisc(43, 93, 29, 4, 0x8410);
5 gColDisp_SetMonoColor(0xF800);
6 gColDisp_DrawImage(0x19E4AD54, 170, 59, 0, 0);
7 gColDisp_SetFont(0x4652638D);
8 gColDisp_SetForeground(0xC819);
9 gColDisp_PutString(59, 126, "Text", 0, 0);
```

Všechny příkazy pro vkládání grafických útvarů vždy začínají pozicí prvního bodu (levý horní u geometrických a obrázků, levý spodní u textu), následují parametry specifické danému útvaru jako druhý bod u obdélníku, případně tloušťka čáry, výplň, barva. Příkaz pro nastavení fontu operuje již s kontrolním součtem Adler-32, všechna jemu následující vložení textu jej pak využívají. Funkce pro zakreslení obrázku také využívá kontrolní součet Adler-32.

Export obrazovek uživatelského rozhraní do příkazového tvaru není v projektu primárně využíván. Je totiž mnohem rychlejší nahrát jednorázově při startu programu celou obrazovku z externí Flash do SDRAM, než ji při každém přepnutí obrazovky zapisovat pomocí příkazů, neboť první zmíněný způsob lze z velké části realizovat pomocí DMA, zatímco příkazy musí zpracovat přímo mikrokontrolér. Jedná se nicméně o cestu, jak grafiku vytvořit v mikrokontrolérech, které pracují s menším displejem a především nemají k dispozici dostatečný prostor pro ukládání kompletních grafických podkladů.

5.3.4 Stažení projektu do přístrojové desky

Verze zařízení popisovaná v této diplomové práci nedisponuje prostředky pro přímé stažení projektu vytvořeného v PC softwaru do elektroniky. Zavedení nového uživatelského rozhraní je proto prováděno přes firmware zařízení, do kterého jsou jednotlivá zdrojová data nahrána ručně a poté zavoláním jediné funkce přenesena do externí Flash paměti.

Zdrojový kód popsáný v příloze C má v souboru *Dashboard.var.c* předpřipraveny funkce pro vložení všech nezbytných definičních a obrazových dat. Založena jsou zde pole pro hlavičky fontů, obrázků a uživatelských obrazovek, do nichž se nejdříve zapíše počet použitých objektů daného typu spolu s kontrolním součtem tohoto počtu, jak je zmíněno v podkapitole 4.3.1. Kontrolní součet je následován hlavičkami jednotlivých objektů daného typu sestavenými dle předdefinovaného systému. Kompletní grafické podklady pro fonty se ukládají do souboru *UI/data_fonts.h*, obrazovky a obrázky pak

do *UI/data_graphics.h*. Kompletní znění proměnných je taktéž uloženo ve vyhrazeném poli, přičemž jejich definování je na základě několika vstupních parametrů obstaráno tomu přizpůsobenou funkcí *initSet()*. Podobně byla vytvořena funkce s názvem *initGear()* vyhrazená pro vstupní typ Enum. Tímto úkonem jsou hlavičkové tabulky a proměnné hotovy, zbývá doplnit volání zápisu grafických podkladů na vyhrazené místo v paměti shodující se s umístěními zvolenými při vytváření hlavičkových tabulek. Po jednorázovém spuštění kódu v mikrokontroléru se všechna data zapíše na odpovídající místa v externí Flash paměti, je tedy možné volání inicializačních funkcí ve firmwaru opět zablokovat.

6. Demonstrační software

Komplexní testování celého systému, vytvořeného v rámci diplomové práce, proběhlo na datech sestavených specificky za tímto účelem. Vzhledem k množství variant interpretace dat přijatých ze sběrnice CAN a jejich převodu na výstup a zobrazování nejsou některé signály nastaveny tak, jak by v praxi za normálních okolností měly být. V těchto případech byl upřednostněn demonstrační požadavek na ukázkou různých kombinací nastavení.

6.1 Systémová nastavení

Konfigurace všech důležitých systémových parametrů byla použita ve stejné podobě, jako byla popsána v předchozích kapitolách. Pro shrnutí se jedná o následující body:

- **Mikrokontrolér:** pracovní frekvence 168 MHz
- **LCD TFT:** rozlišení 480×272 bodů při frekvenci $f_{DotCLK} = 10$ MHz (obnovovací frekvence displeje ≈ 68 Hz) a 16bit barevné hloubce (RGB 565)
- **SDRAM:** řídicí frekvence 84 MHz, vyhrazený prostor pro 16 uživatelských a 2 dynamické obrazovky zobrazující hodnoty signálů; zbylý prostor přiřazen grafickým zdrojům (fonty, obrázky)
- **NAND Flash:** využit systém hlavičkových tabulek s adresami všech grafických podkladů a konfigurací signálů
- **CAN periferie:** fixní rychlost 1 MBit/s

Další parametry jako nastavení filtrace paketů přijatých z CAN či počet obrazovek a rozmístění signálů na nich jsou již součástí uživatelské konfigurace.

6.2 Konfigurace signálů

Demonstrační software pro přístrojovou desku se snaží pokrýt maximum dostupných parametrů nastavení signálů, jak byly popsány v podkapitole 4.2.2. Pro zopakování se jedná o následující:

- **CAN identifikátor:** základní (11bit), rozšířený (29bit)
- **Endianita:** standard Intel (LSB), standard Motorola (MSB)
- **Vstupní typ:** Signed, Unsigned, Float, RAW, ASCII, Enum
- **Vstupní délka**
 - 1, 2 nebo 4 B (Signed, Unsigned, RAW, Enum)
 - 4 B (Float)
 - 1-8 B (ASCII)
- **Výstupní typ:** Decimal, Hexadecimal, Binary, Float

Seznam základního nastavení signálů sloužícího k demonstraci všech funkcí se nachází v následující tabulce:

Signál	CAN ID/Offset	Délka [B]	Endianita	Vstupní typ	Výstupní typ
Speed	0x0120/0	2	Intel (LSB)	Unsigned	Decimal
RPM	0x0120/2	2	Intel (LSB)	Unsigned	Decimal
T Water	0x5500/0	2	Motorola (MSB)	Signed	Decimal
P Oil	0x5500/2	2	Motorola (MSB)	Unsigned	Decimal
T Oil	0x5500/4	2	Motorola (MSB)	Signed	Decimal
P Boost	0x5500/6	2	Motorola (MSB)	Unsigned	Decimal
T Air	0x6000/0	2	Intel (LSB)	Signed	Decimal
P Fuel	0x6000/2	2	Intel (LSB)	Unsigned	Decimal
Diag	0x6000/4	2	Intel (LSB)	RAW	Hexadecimal
Message	0x6020/0	8	Intel (LSB)	ASCII	-
V Bat	0x6040/0	4	Intel (LSB)	Float	Float
Gear	0x6040/4	1	Intel (LSB)	Enum	-

Tab. 6.1: Základní nastavení demonstračních signálů

Doplňující informace k jednotlivým signálům shrnuje následující tabulka:

Signál	Zesílení	Posun	Desetinná místa
Speed	0,01	0	0
RPM	0,1	0	0
T Water	0,1	0	1
P Oil	0,01	0	1
T Oil	0,1	0	1
P Boost	0,001	0	2
T Air	0,1	0	1
P Fuel	0,01	0	2
Diag	-	-	-
Message	-	-	-
V Bat	1	0	1
Gear	1	0	0

Tab. 6.2: Rozšiřující nastavení demonstračních signálů

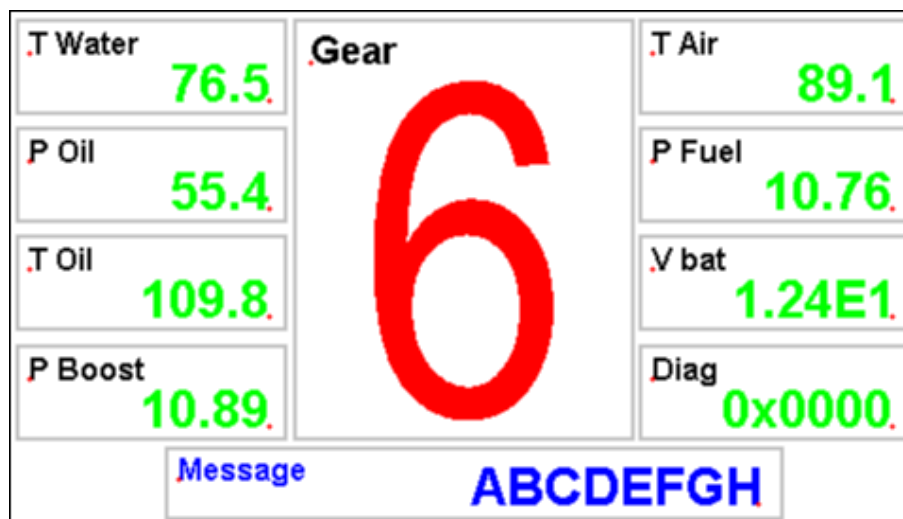
Jediný signál z obou tabulek, jenž nebyl ještě řádně zdefinován, náleží poloze řadicí páky. Zde se v praxi běžně využívá právě vstupní typ Enum, který převádí předem stanovené intervaly na textový výstup. Druhým poznatkem z praxe je, že číslice a písmena reprezentující jednotlivé stavy mohou být odesílány v ASCII podobě nebo číselnou hodnotou. Uživatel také obvykle požaduje pokrytí všech hodnot mimo stanovené intervaly předem definovaným znakem nebo řetězcem znaků, což vstupní typ Enum podporuje již ve svém základu, jak již bylo popsáno v podkapitole 4.2.2. Následující tabulka ukazuje konfiguraci signálu *Gear*:

Začátek intervalu (vč.)	Konec intervalu (vč.)	Výstup	Komentář
49	49	'1'	
50	50	'2'	
51	51	'3'	
52	52	'4'	
53	53	'5'	
54	54	'6'	
69	69	'E'	chyba
78	78	'N'	neutrální poloha
82	82	'R'	zpětný chod
	ostatní	'*'	

Tab. 6.3: Nastavení signálu typu Enum

6.3 Obrazovky uživatelského rozhraní

Databáze signálů, navržená v předchozí podkapitole, byla využita k sestavení dvou demonstračních obrazovek obsahujících všechny zadané signály. Jejich vzhled je následující:



Obr. 6.1: Navržené uživatelské obrazovky

Výchozí hodnoty signálů byly vyplněny ručně. Červené body u textů slouží k jejich zarovnání, kde statické texty jsou zarovnány zleva, tedy tak, jak je text běžně generován. Standardní signály se zarovnávají zprava a výčtový typ Enum uprostřed, neboť se ve velké míře využívá u jednoznakových výstupů, kde centrování přispívá k lepšímu vzhledu uživatelského prostředí.

6.4 Ovládání

Přepínání uživatelských obrazovek na přístrojových deskách je velmi často realizováno externími prostředky, například tlačítkovými panely nebo spínači umístěnými na volantu, viz následující obrázky:



(a) Panel spínačů

(b) Nástavba na volant

Obr. 6.2: Externí ovládací prvky

Vzhledem k absenci těchto přepínačů byl za účelem testování vložen přímo do funkce pro zpracování dat ze sběrnice CAN kus kódu vyhledávající paket se specifickým identifikátorem, jenž obsahuje číslo požadované strany. Obdrží-li tak mikrokontrolér paket s daným ID a validním datovým obsahem, přenastaví se na zvolenou obrazovku, smaže obě dynamické vrstvy a vynutí překreslení všech signálů nacházejících se na nově zvolené obrazovce.

6.5 Testování

Optimalizace jednotlivých funkčních modulů, jako je zpracování dat ze sběrnice CAN, vyhledávání změn hodnot či překreslování, byla prováděna průběžně v rámci vývoje. Kromě zmíněných funkcí bylo hledáno také optimální sestavení jednotlivých typů signálů, fontů a grafických podkladů s cílem prohledávat jednotlivé seznamy a zapisovat data co nejrychleji při minimálním zdvojení dat v paměti.

Hlavní parametr definující výkon přístrojové desky je počet překreslení obrazovky za sekundu. Výrobce displeje udává odezvu 40 ms jakožto součet časů přepnutí z černé barvy na bílou a zpět. Teoreticky by se tak displej měl bez větších potíží překreslovat i více než 30× za sekundu. Algoritmus překreslování je vnitřně omezen na 20 snímků/s, pro potřeby testování bylo však toto omezení odstraněno, protože důležitý je především teoretický výkon systému, nikoliv fyzické možnosti displeje.

6.5.1 Režimy testování

Měření výkonu bylo prováděno v několika konfiguracích pro srovnání podmínek ideálních, praktických a tzv. „Worst-case scenario”. Nejdůležitější výsledek samozřejmě odpovídá zpracování dat z vytížené sběrnice CAN při dostatečném množství překreslovaných dat na obrazovce. Provedeny však byly také testy zjišťující maximální možný výkon.

Testování rychlosti přepisu dat proběhlo v následujících obrazových konfiguracích:

1. **1 signál, drobné písmo:** Praktický význam testu tkví ve zjištění rychlosti, s jakou je algoritmus schopen přenést příchozí data s CAN paketu na obrazovku.
2. **1 signál, střední písmo:** Ověření rychlosti přepisu signálu s fontem běžně využívaným v praxi.
3. **1 signál, mohutné písmo:** Uživatel bude mít při návrhu rozhraní tendenci využívat pro nejdůležitější parametry velké fonty zlepšující čitelnost. Tento test odhaluje optimalizaci algoritmu vyhledávání a zápisu písmen na displej.
4. **Standardní obrazovka, kombinace velikostí písem:** Test v podmínkách očekávaných v praxi.

Zdrojová data pro testování popisují následující body:

- A) **Vynucené překreslování:** Zobrazovaná data nepocházejí z CAN paketů, jedná se o výchozí hodnoty překreslované v nekonečné smyčce.
- B) **CAN se zapnutou filtrací:** Stav, ve kterém by systém měl být provozován, tj. identifikátory paketů byly zvoleny s ohledem na optimální využití konfigurovatelných filtrů v mikrokontroléru.
- C) **CAN bez filtrace:** Testování algoritmu v situaci, kdy je objem použitých CAN identifikátorů natolik veliký, že nelze najít optimální nastavení filtrace. Tento scénář by nastal až při použití řádově 100 vzájemně odlišných CAN identifikátorů v použitých signálech a několikanásobku podobně sestavených identifikátorů pohybujících se po sběrnici, kdy by velké množství paketů vyhovělo filtrům na bázi masek a muselo být zamítnuto až samotným algoritmem.

6.5.2 Konfigurace systému

Měření výkonu proběhlo podle bodů A až C z předchozí podkapitoly na čtyřech oddělených obrazovkách, rozdílné však bylo využití sad testovacích dat. Obrazovky s jedním signálem (body 1 až 3) byly měřeny pouze v režimu vynuceného překreslování, neboť hlavní roli zde hraje zjištění maximálního výkonu. Kompletní uživatelská obrazovka (bod 4) byla testována na všech 3 sadách dat pro srovnání maximálního výkonu, výkonu v optimálních podmínkách (zapnutá filtrace) a v silně zatíženém systému (vypnutá filtrace).

První obrazovka obsahovala signál *P Fuel* s očekávanou hodnotou v řádu desítek a dvěma desetinnými místy, graficky reprezentovaný fontem o velikosti 9 bodů.

Druhá, vycházející z první, využívala stejné sestavení, ovšem s 20b fontem, který se v dosavadním praktickém nasazení zařízení projevil jako optimální volba, zajišťující dobrý poměr mezi čitelností a maximálním počtem signálů na jedné obrazovce.

Třetí obrazovka byla sestavena stejně jako první, pouze font byl zvolen o velikosti 80 b. Nejedná se samozřejmě o maximum - například signál polohy řadicí páky má v demonstračním projektu text vysoký 180 b, praxe nicméně ukázala, že řidič vyžaduje zobrazení více signálů byť menším, ale stejným písmem než jednu hodnotu písmem velkým a zbytek drobným. Z toho důvodu byla pro testování zvolena velikost, která stále dovoluje umístění více signálů na jednu obrazovku.

Čtvrtá testovací obrazovka s objemem signálů podobným praktickému nasazení je shodná s druhou demonstrační obrazovkou, tj. 9 signálů 20b fontem a jeden signál s písmem vysokým 180 b.

Filtrace CAN paketů byla vzhledem ke složení identifikátorů v projektu nastavena přímo na tato ID, propouštěny tak byly pouze pakety, jež bylo možné dále zpracovat. Měření času bylo realizováno interním 32bit časovačem, který byl spuštěn vždy pouze po dobu překreslování předem definovaného počtu obrazovek.

Zdrojová data pro body B a C byla generována převodníkem USB ↔ CAN **PCAN-USB** [31] od firmy PEAK-System Technik, který v kombinaci se softwarem PCAN-View dokáže generovat CAN pakety, každý s frekvencí odesílání až 1000 Hz.

Výsledky testů výkonu ve všech výše uvedených situacích byly zaznamenány do tabulek, v nichž figurují následující údaje:

- **Timer:** Hodnota časovače po skončení měření (hodinový signál 84 MHz).
- **Překreslení:** Počet překreslení (přepnutí) obrazovky v průběhu testu.
- **t (čas/obrazovka):** Doba nutná k přepsání obsahu obrazovky.
- **FPS:** Zkratka pro *Frames per second*, tj. počet snímků zobrazených za sekundu

6.5.3 Výsledky

Měření každé konfigurace bylo provedeno opakovaně, neboť zvláště u obrazovky s více signály měl na výsledek nezanedbatelný vliv souběh příchozích dat ze sběrnice CAN. Software pro převodník USB ↔ CAN totiž umožňuje vysílání pouze manuálně nastartovat a ukončit, nikoliv nastavit prodlevy mezi startem vysílání jednotlivých kanálů. Může tak dojít k souběžnému vysílání více různých hodnot na stejném kanálu, z nichž pak mikrokontrolér nemusí zpracovat úplně všechny.

Jako první byly provedeny testy s **vynuceným překreslováním**, jež shrnuje následující tabulka:

Font 9 b				
	Timer	Překreslení	t [ms]	FPS
1)	48173062	1000	0,5735	1743,71
2)	48194110	1000	0,5737	1742,95
3)	48187954	1000	0,5737	1743,17
4)	24108828	500	0,5740	1742,10

Font 20 b				
	Timer	Překreslení	t [ms]	FPS
1)	225722328	500	5,374	186,07
2)	225729314	500	5,375	186,06
3)	225671570	500	5,373	186,11
4)	225681482	500	5,373	186,10

Font 80 b				
	Timer	Překreslení	t [ms]	FPS
1)	153126106	50	36,459	27,43
2)	153122556	50	36,458	27,43
3)	306265688	100	36,460	27,43
4)	306293452	100	36,464	27,42

Tab. 6.4: Test výkonu při vynuceném překreslování

Hodnoty uvedené v tabulkách vyjadřují maximální výkon, kterého je grafické jádro přístrojové desky schopno dosáhnout. Zkouška s nejmenším fontem poskytuje určitou představu o době, během které se mikrokontrolér věnuje režii nově příchozích hodnot, tedy hledání příznaků změny hodnoty v seznamu signálů, přepočtu hodnoty z CAN paketu na výstupní a jeho formátování, hledání správného fontu a jednotlivých znaků v něm. Následuje mazání původního textu a zápis nového, kde dochází k určitým zpožděním vlivem univerzálnosti zápisové funkce (zapnutí/vypnutí přepisování pozadí, rotace textu) a kvůli kontrolám pozice zápisu, není-li mimo oblast displeje.

Zápisová rychlost fontu o velikosti 20 bodů se v praktických testech ukázala jako vyhovující, neboť mnoho signálů odesílaných řídicí jednotkou lze ve srovnání s obnovovací frekvencí uživatelského rozhraní 20 Hz považovat takřka za statické, například některé

údaje o teplotách, tlacích, napětí či zařazeném rychlostním stupni. Obvykle je tak v cyklu přepisován jen omezený počet signálů a je tedy možné udržet dostatečnou frekvenci překreslování.

Měření největšího fontu o velikosti 80 b by mohlo evokovat nízký výkon, jsou-li běžně využívány fonty i 2× větší. Praxe tuto domněnku nicméně nepotvrdila, protože takto velké fonty se zatím využívaly takřka výhradně pro signalizaci zařazeného rychlostního stupně, který svou hodnotu mění oproti jiným signálům jen velmi výjimečně a jehož zpracování vzhledem ke vstupnímu formátu typu Enum trvá mnohem kratší dobu, než přepočítání a formátování standardních signálů. Testovací podklady, vytvořené pro popisované měření, tak poskytují dobrý ukazatel pro další optimalizace zobrazovacího algoritmu.

Druhá část testovacích měření byla prováděna na druhé demonstrační obrazovce, a to opět při konfiguraci vynuceného překreslování a dále s daty přijímanými ze sběrnice CAN. Zprávy určené pro zpracování mikrokontrolérem byly odesílány podle praktické zvyklosti s frekvencí 100 Hz. Zprávy simulující neužitečná data, určená k rozlišení testu se zapnutou a vypnutou filtrací CAN zpráv, byly vysílány s frekvencí 1000 Hz. Vytvořeny byly celkem 4 dvojice zpráv využitelných k tomu, aby se hodnoty jednotlivých signálů měnily a algoritmus byl přinucen displej překreslit, a 5 dvojic zpráv sloužících jako datový šum.

Výsledky jednotlivých měření shrnují následující tabulky:

CAN s filtrací zpráv				
	Timer	Překreslení	t [ms]	FPS
1)	604786562	100	71,998	13,89
2)	620983164	100	73,927	13,53
3)	615994756	100	73,333	13,64
4)	616251350	100	73,363	13,63

CAN bez filtrace zpráv				
	Timer	Překreslení	t [ms]	FPS
1)	626241938	100	74,553	13,41
2)	622930584	100	74,158	13,48
3)	629902508	100	74,988	13,34
4)	631386500	100	75,165	13,30

Vynucené překreslování				
	Timer	Překreslení	t [ms]	FPS
1)	276783444	50	65,901	15,17
2)	276764974	50	65,896	15,18
3)	553526502	100	65,896	15,18
4)	553562164	100	65,900	15,17

Tab. 6.5: Test výkonu v reálných podmínkách

Měření s vynuceným překreslováním vykázalo dle očekávání lepší výsledky než při zpracovávání dat ze sběrnice CAN. Ztrátu necelých 2FPS oproti ideálnímu stavu lze přičíst právě zpracování příchozích dat, jež probíhá i během úkonů prováděných s novými hodnotami signálů, a tudíž tuto činnost zpomaluje. Rozdíl mezi filtrovanou a nefiltrovanou CAN komunikací je spíše zanedbatelný. Příčinou tohoto jevu je nicméně sám mikrokontrolér, který při standardním nastavení periferie CAN přijímaná data zařazuje do třímístného posuvného registru, vypnutí filtrace tak zapříčiní ztrátu validních dat, nikoliv zahlcení systému, jak by se dalo čekat. Minimální rozdíl v FPS však indikuje, že algoritmus zpracování příchozích dat se při běžném zatížení pohybuje na hranici svých možností. Jedná se tedy o jedno z potenciálně slabých míst, na které by se měla orientovat další optimalizace.

7. Shrnutí

Vývoj elektronické přístrojové desky proběhl v několika fázích. V rámci práce na návrhu desky plošných spojů bylo do prostředí Altium Designeru převedeno schéma vytvořené zadavatelem, čímž bylo možné přejít k samotnému návrhu desky plošných spojů. Zde byla průběžně aplikována řada poznatků a pravidel pro návrh zapojení rychlých, výkonových i analogových obvodů. Důraz byl kladen také na odolnost vůči elektrostatickým výbojům a na nízké elektromagnetické vyzařování kvůli budoucím testům na elektromagnetickou kompatibilitu.

Firmware zařízení byl sestaven po částech. Na úvod byly naprogramovány knihovny pro řízení periferií jako je SDRAM, externí Flash, CAN periferie či LCD a jeho podsvícení. Spuštěny byly také všechny automatizované procesy, například periodické obnovování obsahu SDRAM nebo DMA pro přenos obrazových dat z paměti na displej. Následoval řetězec příjmů, zpracování a zobrazení dat, tedy konfigurace filtrace CAN zpráv a konverze přijatých dat na hodnoty určené k přepočtu, výstupní formátování a zápis na LCD. Posledním bodem byly funkce měření napětí na interních referencích a teplot v různých místech desky plošných spojů. Nechybí také měření externích analogových (napěťových a teplotních) signálů a frekvenčních vstupů.

Systém přepisu obrazových dat byl postupně otestován ve dvou verzích. První, založená na střídání 2 vrstev obsahujících statickou grafiku i proměnné, byla kvůli hardwarovým limitacím přepínání přístupu do SDRAM a externí Flash paměti nahrazena druhou variantou, v níž je využito klíčování vrstvy s dynamickými prvky na statickou vrstvu s pozadím. Grafická knihovna byla navržena a optimalizována specificky pro tyto účely, obsahuje však také funkce pro zápis základních geometrických útvarů a bitmap. Zápis textů je v knihovně podporován s rozšířenými konfiguračními možnostmi. Vzhledem k funkční povaze zařízení a rozmanitým požadavkům uživatele je možné textům přiřadit nejen barvu a natočení, ale i libovolné písmo vytvořené v konfiguračním softwaru, díky čemuž se výrazně zvyšuje rozmanitost jednotlivých uživatelských rozhraní.

Definice a správa automobilových signálů byla sestavena s ohledem na rychlost při zachování dostatečné univerzálnosti pro zavádění nových vstupních a výstupních typů. Každý signál obsahuje všechny údaje nezbytné k separaci dat z jemu příslušejícího

CAN paketu a jejich přepočítání na reálnou hodnotu, formátování na výstupní tvar a v neposlední řadě také parametry definující grafickou reprezentaci výstupu, například pozice na obrazovce, barva nebo písmo.

Grafický PC software vytvořený v jazyce Java slouží ke kompletnímu návrhu uživatelského rozhraní. Databázová práce s obrázky a písmi nabízí řadu možností, jak tato zdrojová data do rozhraní vkládat nebo je ukládat. Import bitmap je možný buď z běžných obrázků v několika formátech ze souboru zpracovaného samotným softwarem, nebo vytvořením v editoru. Primární způsob vkládání nových fontů je ze seznamu systémových písem, jenž software podle volby uživatele zpracuje do tvaru čitelného jak softwarem, tak i firmwarem přístrojové desky. Nabízí se samozřejmě také import již zpracovaných fontů ze souboru nebo z webu knihovny, jejíž systém komprimace písem byl použit i v této práci. Další funkcí PC softwaru je editor znakových sad, v němž je možné provést vzhledové korekce jednotlivých znaků, přidávat je či odebírat a také vytvořit font úplně nový. Všechny zmíněné grafické podklady využívá editor podporující stejné funkce jako grafická knihovna, tedy základní geometrické útvary, obrázky a texty. Výstupem editoru jsou primárně statické grafické podklady sloužící jako pozadí pro zapisované hodnoty signálů, dostupný je však i výstup v podobě příkazů, které lze vložit do zdrojového kódu a zobrazovat rozhraní pomocí jednotlivých grafických útvarů.

Závěr práce je věnován návrhu demonstračního uživatelského rozhraní určeného k ověření různých možností konfigurace zobrazovaných signálů a grafických podkladů. Je v něm zahrnuto také měření maximálního výkonu systému v ideálních podmínkách a výkonu v konfiguraci očekávané při reálném nasazení.

8. Závěr

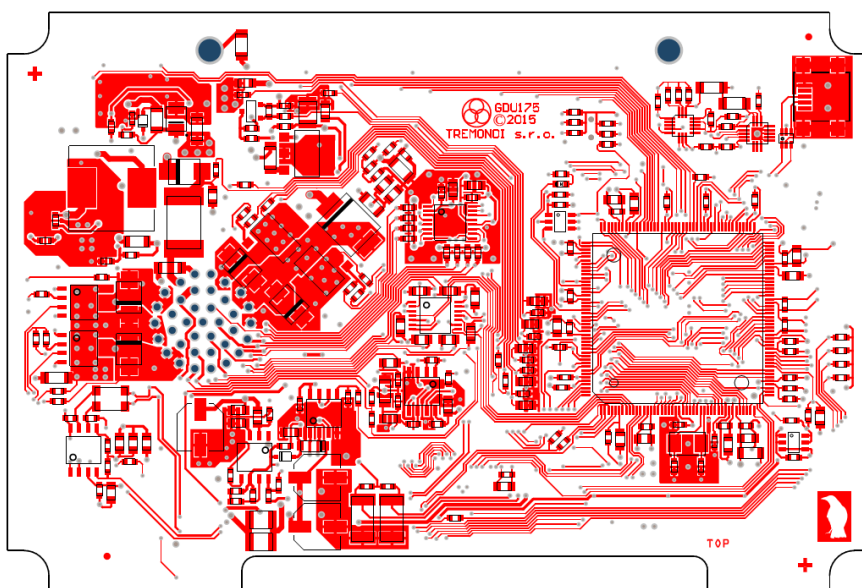
V rámci této diplomové práce byl vyvinut kompletní systém správy a návrhu uživatelského rozhraní a zobrazování dat pohybujících se po sběrnici CAN. Navržená deska plošných spojů funguje podle požadavků, dosavadní práce neodhalila žádné závažné funkční nedostatky. Ladění grafické knihovny a procesu zapisování dat na displej proběhlo s ohledem na optimální využití prostředků dostupných v mikrokontroléru a samozřejmě také na výslednou rychlost. Mezi podporované prvky patří základní geometrické útvary a obrázky v monochromatickém a barevném formátu. Nechybí také funkce pro vkládání textů s využitím externě sestavených znakových sad. Zobrazování je realizováno pomocí metody klíčování vrstvy s hodnotami signálů na vrstvu se statickými prvky. Správa grafických podkladů a uživatelské konfigurace zobrazovaných veličin byla navržena s důrazem na univerzálnost a snadnou rozšiřitelnost o nové prvky, to vše při zachování rychlosti zpracování dat a optimálního využití externí Flash paměti. PC software vytvořený v jazyce Java slouží ke správě grafických podkladů a znakových sad s širokou paletou možností v oblasti jejich editace, načítání a ukládání. Součástí je také editor obrazovek uživatelského rozhraní, s nabízející všechny funkce podporované grafickou knihovnou.

Testování výkonu ukázalo, že rychlost zpracování surových dat na výstupní hodnotu a vyhledávání odpovídajících znaků ve fontu lze navzdory značné univerzálnosti vstupních a výstupních typů označit za vyhovující, - proběhnout může každých cca 570 μ s. Testování zápisu textů standardním fontem odhalilo, že běžné signály lze i při jejich vyšším počtu zapisovat stále dostatečně rychle (186 překreslení/s), zatímco velké fonty musí být užívány s rozvahou, má-li být zachována dobrá reakční doba systému (27 překreslení/s). Měření na reálných podkladech s daty přijímanými ze sběrnice CAN proběhlo s výsledkem 15 překreslení/s, což je pro zobrazování numerických a textových výstupů stále dostatečná hodnota. V průběhu testování byla také identifikována potenciálně slabá místa vytvořeného zobrazovacího systému, do budoucna bude proto nezbytné zaměřit se na další optimalizace, například v oblasti zápisu znaků na displej nebo filtrace a zpracování CAN zpráv.

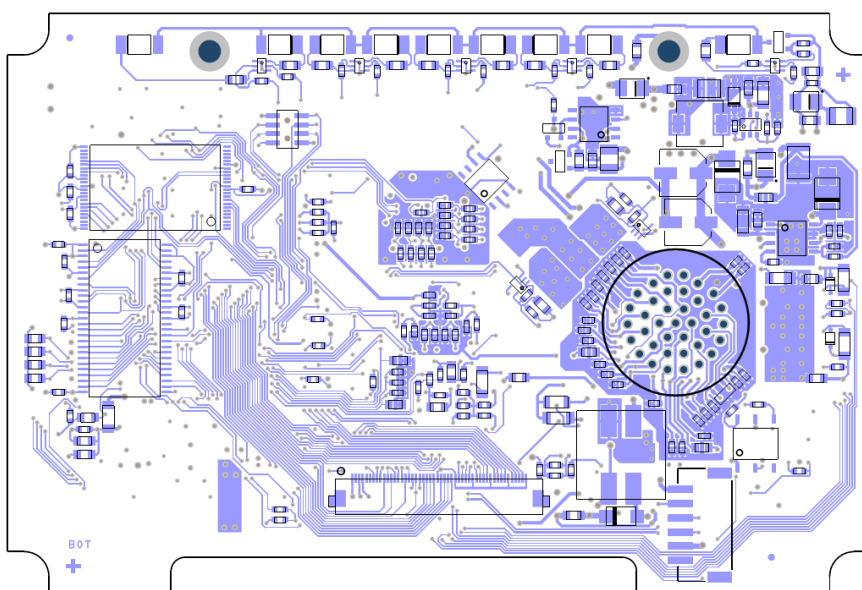
9. Obsah CD

Příloha	Název adresáře	Obsah
1	DP	diplovová práce ve formátu PDF
2	PDF	datasheety použitých komponent

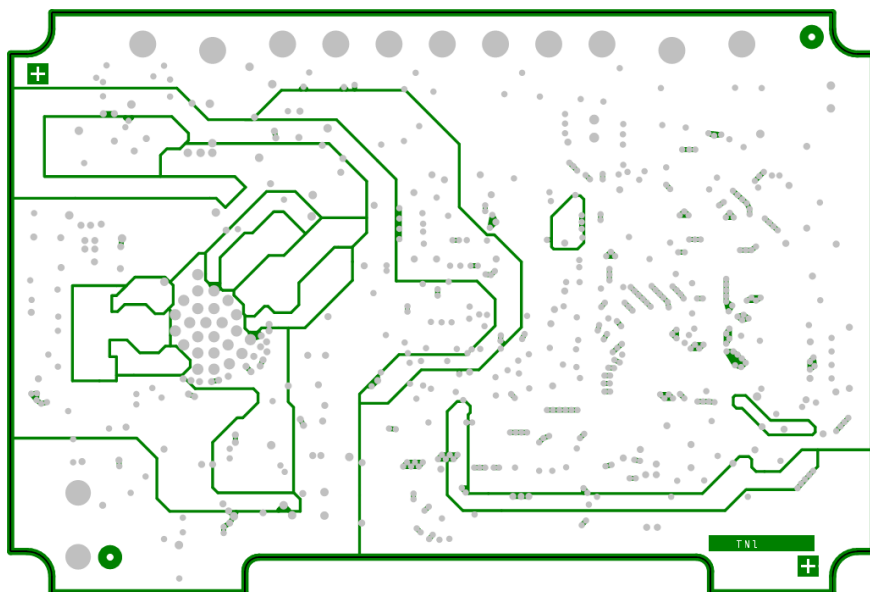
A. Deska plošných spojů



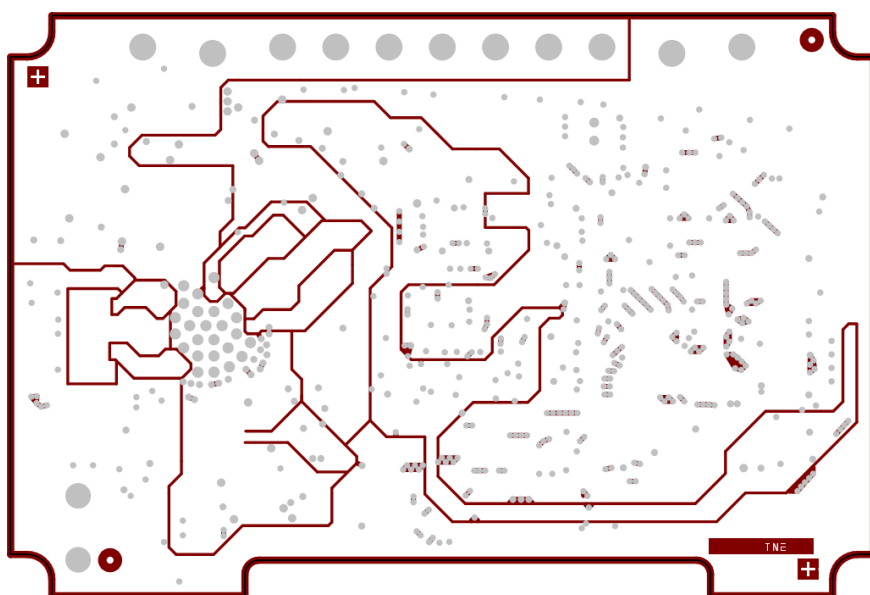
Obr. A.1: Vrstva Top



Obr. A.2: Vrstva Bottom (zrcadleno)



Obr. A.3: Vrstva napájení (invertováno)











Obr. A.4: Vrstva zemnění (invertováno)

B. Programová struktura grafického editoru

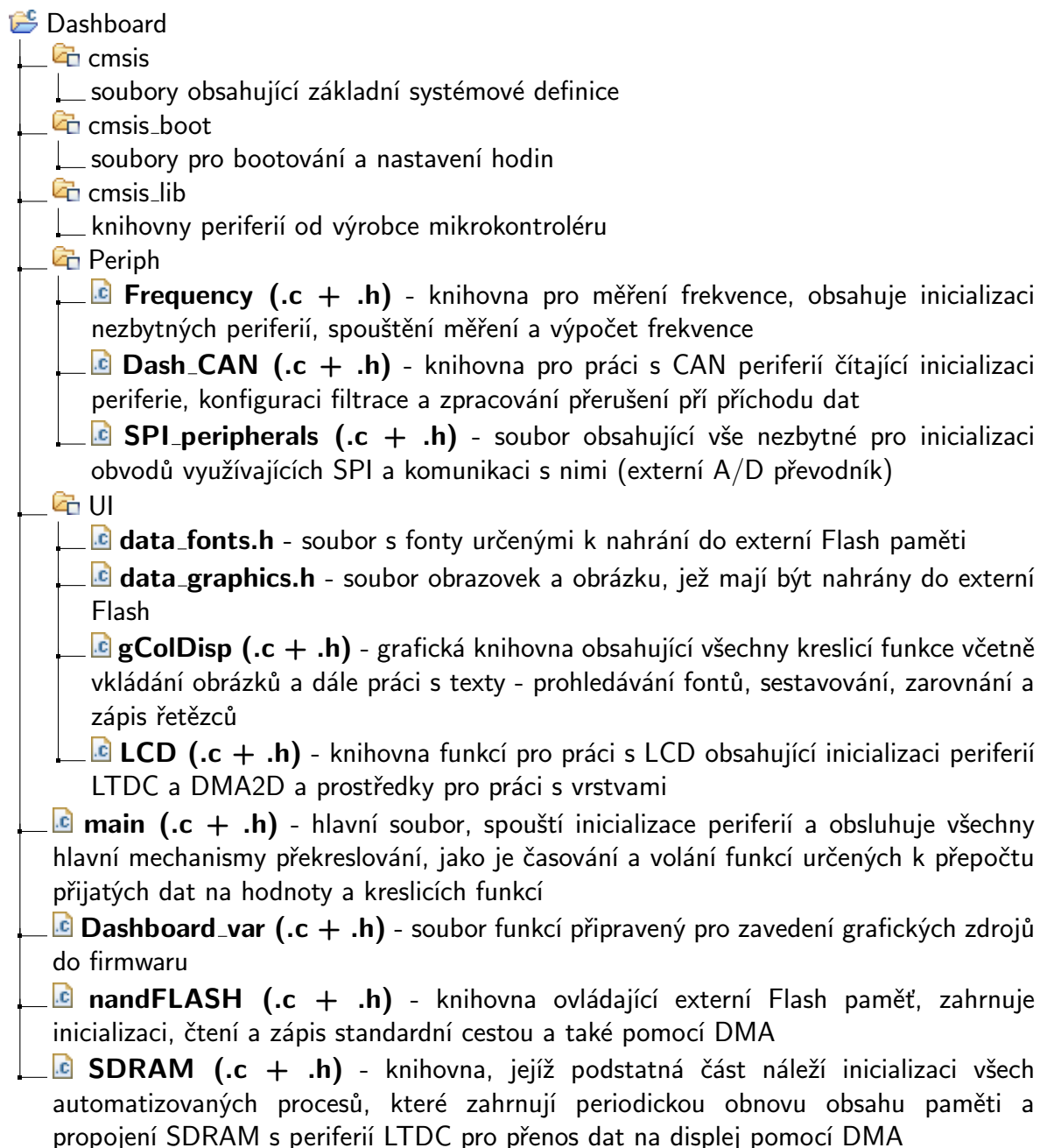


Graphic Editor (pokračování)




GUI.support

-  **DBpanel_fonts.java** - panel je umístěn na záložku s databázemi, slouží ke správě fontů z hlediska souborů (načítání, ukládání, tvorba nových)
-  **DBpanel_images.java** - panel správy obrázků, je vložen na záložku s databázemi
-  **Frame_ImageSize.java** - samostatné okno umožňující změnit rozměry kreslicí plochy v grafickém editoru
-  **Frame_NewImage.java** - okno, na němž si uživatel zvolí parametry nově vytvářené grafiky (rozměry, objektové/rastrové kreslení)
-  **Frame_SelectFont.java** - okno zobrazující se při vkládání a editaci textových objektů
-  **Frame_SelectImage.java** - okno se seznamem obrázků dostupných v databázi, zvolení jednoho je vložen na kreslicí plochu
-  **TablePaint.java** - panel fungující jako kreslicí plocha, který odpovídá za překreslování obrazu a správné nastavení měřítka jednotlivých obrazových bodů podle rozměrů kreslicí plochy
-  **ToolPanel_objectInfo.java** - panel umístěný vespod obrazovky, jenž umožňuje uživateli dodatečnou editaci všech vložených grafických objektů

C. Programová struktura firmwaru



Dashboard (pokračování)

-  **stm32f429_it (.c + .h)** - soubor, v němž jsou zapsány funkce pro zpracování kritických systémových přerušení spolu s ostatními přerušeními od časovačů, SPI, DMA2D, atd.
-  **stm32f4xx_conf.h** - soubor s odkazy na všechny používané systémové knihovny
-  **variableCore (.c + .h)** - knihovna odpovědná za načítání grafických zdrojů z paměti při startu programu a za zpracování datových paketů CAN zpráv na výstupní tvar dle konfigurace sestavené uživatelem

Seznam použité literatury

- [1] **Etschberger, K.:** *Controller Area Network*
IXXAT Press, 2001, ISBN 3-00-007376-0
- [2] **Záhlava, V.:** *Návrh a konstrukce desek plošných spojů*
Česká technika - nakladatelství ČVUT, 2005, ISBN: 80-01-03351-1
- [3] **Herout, P.:** *Java - grafické uživatelské prostředí a čeština*
Kopp, 2007, ISBN: 80-7232-328-9
- [4] Web společnosti TREMONDI s.r.o. [online] 4.12.2015
<http://www.tremondi.cz/cs>
- [5] Tisková zpráva o spolupráci značek Audi a NVIDIA [online] 5.12.2015
<http://nvidianews.nvidia.com/news/audi-and-nvidia-expand-visual-computing-in-the-car>
- [6] Web společnosti Magneti Marelli [online] 6.12.2015
<http://www.magnetimarelli.com>
- [7] Přístrojová deska Magneti Marelli DDU 310 [online] 6.12.2015
http://www.magnetimarelli.com/business_areas/motorsport/data-acquisition,-display-&-lap-trigger/data-display#tab--4
- [8] Web společnosti MoTeC [online] 6.12.2015
<http://www.motec.com/home>
- [9] Přístrojová deska MoTeC D187 [online] 6.12.2015
<http://www.motec.com/c187/c187overview/>
- [10] Web společnosti General Engine Management Systems Ltd [online] 6.12.2015
<http://www.gems.co.uk>
- [11] Přístrojová deska GEMS LDS4 [online] 6.12.2015
<http://www.gems.co.uk/?content=pages&id=lds4-driver-display>
- [12] Domovská stránka softwaru Altium Designer [online] 6.12.2015
<http://www.altium.com/altium-designer/overview>
- [13] Datasheet: mikrokontrolér STM32F429 [online] 6.12.2015
<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00071990.pdf>

- [14] Datasheet: SDRAM IS42S16400J [online] 6.12.2015
<http://www.issi.com/WW/pdf/42-45S16400J.pdf>
- [15] Datasheet: NAND Flash MT29F1G08 [online] 6.12.2015
http://elcodis.com/parts/5219850/MT29F1G08ABADAH4D_p16.html#datasheet
- [16] Datasheet: LCD TFT COG-T430V2100-L2 [online] 12.11.2015
http://www.varitronix.com/prod_pdf/COG-T430V2100-L1.pdf
- [17] Pravidla pro návrh spínaných zdrojů [online] 15.11.2015
<http://www.ti.com/lit/an/snva021c/snva021c.pdf>
- [18] Konektory Deutsch Autosport [online] 12.11.2015
<http://www.deutsch.cz/zlacze/as-autosport/?lang=en>
- [19] Knihovna pro práci s externí Flash pamětí [online] 6.12.2015
http://stf12.googlecode.com/svn/tags/STM32F10x_StdPeriph_Lib_v3.4.0/STM32F10x_StdPeriph_Lib/Source/Utilities/STM32_EVAL/STM3210E_EVAL/stm3210e_eval_fsmc_nand.c
- [20] Sběrnice CAN [online] 18.10.2015
<http://www.eecs.umich.edu/eecs/courses/eecs373/Lec/W12Student/373CANpreso.pdf>
- [21] Datasheet: CAN transceiver MCP2551 [online] 18.10.2015
<http://ww1.microchip.com/downloads/en/DeviceDoc/21667f.pdf>
- [22] Pojednání o kontrolním součtu Adler [online] 25.10.2015
http://www.zlib.net/maxino06_fletcher-adler.pdf
- [23] Datasheet: errata pro mikrokontrolér STM32F429 [online] 6.12.2015
http://www.st.com/web/en/resource/technical/document/errata_sheet/DM00068628.pdf
- [24] Datasheet: NTC termistor NCP18XW222J03RB [online] 6.12.2015
<http://www.mouser.com/ds/2/281/r44e-522712.pdf>
- [25] Datasheet: AD převodník ADC128S022 [online] 6.12.2015
<http://www.ti.com/lit/ds/symlink/adc128s022.pdf>
- [26] Datasheet: senzory Magneti Marelli [online] 6.12.2015
http://www.pim-engineering.com/tiedostot/Magneti_Marelli_sensors.pdf
- [27] Steinhart-Hartova rovnice [online] 15.12.2015
<http://www.thinksrs.com/downloads/PDFs/ApplicationNotes/LDC%20Note%204%20NTC%20Calculator.pdf>
- [28] Inverzní Steinhart-Hartova rovnice [online] 15.12.2015
<http://thermistor.sourceforge.net>
- [29] Grafická knihovna u8glib [online] 11.2.2015
<https://code.google.com/p/u8glib>

- [30] Vývojové prostředí NetBeans [online] 6.12.2015
<https://netbeans.org>
- [31] Převodník USB ↔ CAN [online] 15.12.2015
<http://www.peak-system.com/PCAN-USB.199.0.html?L=1>