

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra aplikované elektroniky a telekomunikací

DIPLOMOVÁ PRÁCE

Jednotka displeje pro elektromotokáru

**Bc. Martin Sova
2016**

Anotace

Tato diplomová práce se zabývá návrhem jednotky displeje pro elektromotokáru. Jednotka displeje obsahuje ovládací tlačítka a displej pro zobrazování dat. Dalšími popisovanými funkcemi jednotky v této práci je měření provozních veličin, ukládání provozních dat, připojení na CAN sběrnici a bezdrátový přenos dat. V závěrečné části práce je oživení jednotky. Schémata zapojení jednotky displeje a řídicí software jsou součástí přílohy diplomové práce.

Klíčová slova

elektromotokára, motokára, automotive, přístrojový panel, mikrokontrolér, STM32F7, displej, provozní data, ukládání dat, CAN sběrnice, A/D převodník

Abstract

This diploma thesis deals with the design of the dashboard for electric kart. The dashboard includes control buttons and display for displaying data. Other of the described functions in the thesis is measurement of operating values, operational data storage, connection to the CAN bus a wireless data transmission. The final part is commissioning of the unit. The schematics diagram and control software are included in thesis as attachments.

Key words

Electric kart, Kart, Automotive, Dashboard, Microcontroller, STM32F7, Display, Operating data, Data storage, CAN bus, A/D converter

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

V Plzni dne 13.5.16

Martin Sova

.....

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Ing. Ludřkovi Elisovi za cenné profesionální rady, připomínky a metodické vedení práce.

Obsah

ÚVOD.....	8
1 POŽADAVKY NA PŘÍSTROJOVÉ PANELE A ELEKTRONIKU VOZIDEL.....	9
1.1 TYPY PŘÍSTROJOVÝCH PANEŮ PRO VOZIDLA.....	9
Ručičkové ukazatele.....	10
Ručičkové ukazatele s displejem.....	10
Displejový ukazatel.....	10
1.2 OCHRANA ELEKTRONIKY A ZDROJE RUŠENÍ V AUTOMOBILOVÝCH APLIKACÍCH.....	11
Návrh obvodu s ohledem na EMI a ESD.....	12
Návrh desky plošných spojů proti EMI a ESD.....	14
2 HARDWAROVÝ NÁVRH JEDNOTKY DISPLEJE.....	16
2.1 BLOKOVÁ SCHÉMATA.....	16
2.2 POPIS JEDNOTLIVÝCH FUNKČNÍCH BLOKŮ.....	19
Zdrojová část.....	19
Mikrokontrolér a displej.....	21
I/U převodníky.....	25
Komunikace po sběrnici CAN a bezdrátová komunikace.....	26
Ostatní periférie.....	27
2.3 NÁVRH DESKY PLOŠNÝCH SPOJŮ A ROZVRŽENÍ FUNKČNÍCH BLOKŮ.....	28
3 ŘÍDÍCÍ SOFTWARE JEDNOTKY DISPLEJE.....	30
3.1 PLATFORMA MIKROKONTROLÉRŮ STM32F7.....	30
3.2 VÝVOJOVÝ DIAGRAM PROGRAMU.....	32
Základní konfigurace při startu programu.....	33
Inicializace displeje a SDRAM paměti.....	34
Inicializace A/D převodníku.....	37
Inicializace CAN sběrnice.....	38
Inicializace UART, I2C, QSPI a USB.....	41
Nekonečná smyčka programu a obsluha přerušení.....	42
3.3 OŠETŘENÍ CHYBOVÝCH STAVŮ.....	44
4 TESTOVÁNÍ A OVĚŘENÍ FUNKČNOSTI.....	45
ZÁVĚR.....	48
SEZNAM SYMBOLŮ A ZKRATEK.....	50
POUŽITÁ LITERATURA A ZDROJE.....	51
PŘÍLOHY.....	55

Úvod

Předkládaná práce je zaměřena na realizaci jednotky displeje pro elektromotokáru. Jednotka displeje je určena k zabudování do elektrické motokáry vyvíjené na Fakultě elektrotechnické Západočeské univerzity v Plzni. Motokára je postavena na závodním šasi od společnosti MSKART. Jedná se o kompletní podvozek, řízení a brzdy. Kromě vyvíjené jednotky displeje motokára obsahuje řídicí jednotku motoru. Motokára je napájena čtyřmi akumulátory LiFeYPO₄. Každý akumulátor má vlastní monitorovací jednotku komunikující po RS485 s jednotkou motoru.

Jednotka displeje obsahuje kromě grafického displeje i ovládací tlačítka a stavové kontrolky. Jednotka má za úkol zobrazovat provozní veličiny na grafický displej, ukládat provozní data, signalizovat poruchy a stavová hlášení. Dále umožňuje snímání polohy brzdového a plynového pedálu. Mezi další doplňkové funkce patří určování polohy pomocí GPS, měření zrychlení a okolní teploty. Modul je navržen pro možnost bezdrátového přenosu dat během jízdy elektromotokáry. Pro řízení celé jednotky je použit výkonný mikrokontrolér s jádrem ARM® Cortex®-M7, který je určen pro použití v embedded aplikacích. Konkrétně tato platforma je od společnosti STMicroelectronics a nazývá se STM32F7.

V úvodní teoretické kapitole jsou popsány požadavky a typy přístrojových panelů. V druhé části této kapitoly jsou popsány typy rušení v automobilových aplikacích. S tím souvisí také druhy ochran automobilové elektroniky. Posledním bodem je vysvětlení navrhování plošných spojů s ohledem na elektromagnetické rušení.

Druhá kapitola se zabývá návrhem hardwaru jednotky displeje. Jsou zde popsána bloková schémata a propojení jednotlivých funkčních bloků. Následuje popis těchto bloků a schéma zapojení. V závěru kapitoly je popsán návrh desky plošných spojů jednotky displeje a fyzické umístění jednotlivých bloků na DPS.

Třetí kapitola je ryze softwarovou záležitostí. Je zde uveden typ platformy mikrokontroléru a popis použitého procesoru. Následuje vysvětlení inicializace a běhu řídicího programu. Popis programu je rozdělen na inicializaci jednotlivých vnitřních a vnějších periférií mikrokontroléru, hlavní programovou smyčku a obsluhu příchozích přerušování programu. V závěru této kapitoly je uvedeno ošetření chyb při běhu programu.

Poslední kapitola se zabývá oživením a odzkoušením jednotky displeje. Je zde ověřeno, zda navržený hardware a software splňuje zadané požadavky, případně jsou navržena opatření na zlepšení.

1 Požadavky na přístrojové panely a elektroniku vozidel

Přístrojový panel zobrazuje řidiči aktuální provozní údaje a všechny důležité informace o stavu vozidla. Nejčastěji se s nimi setkáváme ve vozidlech určených pro silniční provoz. Řidič musí vždy bezpečně vědět v jakém stavu se vozidlo nachází, jde o zajištění ochrany majetku a zdraví. Pokud není řidič informován např. o poruše na brzdách, může dojít k vážné nehodě. Požadavky na přístrojové panely vozidel silničního provozu jsou proto ošetřeny legislativou [3] a normami [2]. Tyto předpisy nám říkají, jak mají vypadat tlačítka a kontrolky jednotlivých zařízení, které vozidlo obsahuje. Je zde uveden jejich tvar, obrys a barva podsvícení.

Mezi běžně zobrazované funkce u motorových vozidel patří:

- aktuální rychlost vozidla
- otáčky motoru, teplota chladicí kapaliny
- zásoba paliva, ujetá vzdálenost
- kontrolní světla (signalizace určité funkce, závada na zařízení)

Zcela jiná situace je u vozidel, která nejsou určena do běžného silničního provozu. U závodních vozidel určených pro motoristické sporty, jsou požadavky dány závodními předpisy mezinárodní automobilové federace FIA dle druhu soutěže. Tyto předpisy jsou většinou velmi specifické a od silničních předpisů se hodně liší. Většinou upravují jiné parametry vozidla a přístrojovými panely se příliš nezabývají[4]. Všechna homologovaná závodní vozidla musí splňovat tyto předpisy. Většinou je k dispozici pouze ukazatel aktuální zařazené rychlosti a několik varovných kontrolků, které závodní jezdec nezbytně potřebuje.

1.1 Typy přístrojových panelů pro vozidla

Typ přístrojového panelu se odráží v aktuální potřebě zákazníka či uživatele, vědeckotechnickém pokroku, který se váže k době vývoje vozidla, a především pak ceně vozidla. Cena, rozměry a výbava určují do jaké obchodní třídy vozidlo spadá. Je rozdíl pokud se jedná o vozidlo segmentu A (malý automobil) nebo o vozidlo třídy F (luxusní automobil) [5]. Luxusní automobily mají většinou k dispozici ty nejmodernější dostupné technologie, které se až postupem času dostávají i do nižších segmentů. Komunikace jednotky přístrojového panelu probíhá nejčastěji po sběrnici CAN.

1.1.1 Ručičkové ukazatele

Mezi nejjednodušší a nejstarší přístrojové panely patří ručičkové ukazatele. Jednotlivé ručičky jsou poháněny aktuátory. U novějších typů je pohánějí krokové motory, které řídí mikrokontrolér. U starých automobilů jsou ovládány mechanicky přes náhon tachometru. Zobrazují pouze základní hodnoty (aktuální rychlost, otáčky motoru, zásoba paliva, teplota chladicí kapaliny, varovné kontrolky) a ujetou vzdálenost.

1.1.2 Ručičkové ukazatele s displejem

Kombinace ručičkového ukazatele s displejem se začala objevovat v automobilech v devadesátých letech. Kromě základních věcí mohou zobrazovat navíc spotřebu paliva, venkovní teplotu, dojezd vozidla, průměrnou rychlost, čas atd. Displej může být segmentový nebo grafický, záleží na parametrech, které chceme zobrazovat a na typu vozidla. Řídící jednotka musí umět tento displej ovládat, to znamená, že je třeba použít vhodný řadič displeje[6].

1.1.3 Displejový ukazatel

Tato varianta umožňuje zobrazit široké spektrum informací. Dovoluje nám to také změnu pohledu a rekonfiguraci ukazatelů. Většinou jsou zobrazovány analogové ukazatele jako u klasického přístrojového panelu, k tomu jsou navíc zobrazovány mapové podklady pro navigaci vozidla. Příklad displejového ukazatele je uveden na obrázku 1.



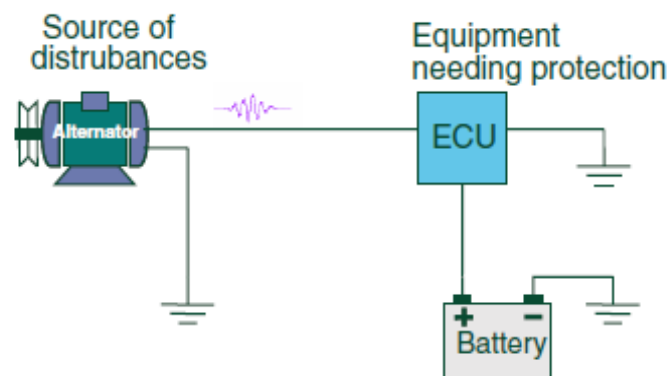
Obrázek 1: Přístrojový panel Volkswagen Passat 2015.

Zdroj: http://www.caricos.com/cars/v/vw/2015_volkswagen_passat/1920x1080/69.html

Další nadstavbou je průhledový displej, v anglické literatuře označovaný jako Head-Up displej. Toto řešení zobrazuje vybrané provozní údaje na čelní sklo do zorného pole řidiče nebo na průhledné sklo v zorném poli. Toto řešení bylo převzato z vojenského letectví, kde se všechny údaje zobrazují na sklo v pilotově helmě.

1.2 Ochrana elektroniky a zdroje rušení v automobilových aplikacích

Dnešní moderní automobily obsahují velké množství elektroniky, která obsluhuje široké spektrum funkcí nezbytných pro provoz vozidla a také komfortní funkce. Tato elektronika musí být spolehlivá a odolná proti elektromagnetickému rušení či výbojům. Během provozu vozidla nesmí nastat porucha, která by mohla ohrozit funkčnost a ovladatelnost vozidla a tím ohrozit bezpečnost posádky. Pokud už nějaká porucha nastane, musí být ošetřena tak, aby došlo k bezpečnému zastavení vozidla. Proto je důležité navrhovat elektronická zařízení v automobilu s ohledem na tyto požadavky. Elektronické zařízení musí být nejen odolné proti rušení, ale svojí činností nesmí ovlivňovat ostatní zařízení, pozemní vysílání (radio, televize) či mobilní síť.

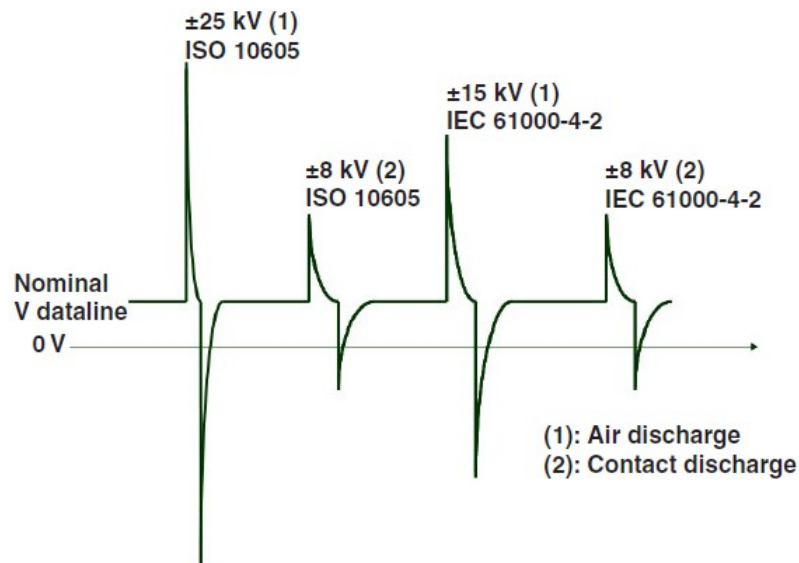


Obrázek 2: Příklad rušení po napájecích vodičích. Převzato z [8]

Zdroje rušení můžeme rozdělit podle toho, jakým způsobem se k zařízení dostávají. Prvním případem je rušení přenášené po napájecích vodičích od zařízení s induktivní zátěží, příkladem může být zapalovací cívka, alternátor, čerpadlo, aktuátory atd. Palubní elektronika musí být odolná i vůči zvlnění napájecího napětí, které vytváří alternátor. Zvlnění napětí je závislé na otáčkách motoru a pohybuje se v jednotkách kHz. Palubní elektronika (příkladem může být řídicí jednotka motoru) musí fungovat i během velkého kolísání napájecího napětí, typicky při startu vozidla. Napětí na baterii při startu vozidla může klesnout i na hodnotu 6 V, přesto musí elektronika motoru fungovat spolehlivě. Příklad rušení po vodičích je uveden na obrázku 2.

Dalším případem je rušení ve formě elektromagnetického záření, které generují zařízení jimiž prochází velký proud (řádově desítky ampér). Jako příklad můžeme uvést IGBT tranzistory, relé, zapalovací svíčky atd. Toto rušení se indukuje do všech vodičů, včetně datových vedení a sběrnic.

Posouzení odolnosti elektronického zařízení vůči EMI a EMC musí být ve shodě s mezinárodními normami. Mezi ně patří například ISO 7 637, ISO 11 451, ISO 11 452, ISO 10 605. Příklad posouzení odolnosti na datových vodičích je uveden na obrázku 3.



Obrázek 3: Přepětové pulzy na datových vodičích dle norem. Převzato z [8]

Normalizaci v automotive zastřešuje kromě ISO (International Organization for Standardization) několik dalších organizací, mezi ně patří SAE (Society of Automobile Engineers), AEC (Automotive Electronic Council) [9]. Automobiloví výrobci si také stanovují vlastní podnikové normy, kterými se musí řídit jejich dodavatelé (VWxxx.xx, FORD WSS-xxx, atd.).

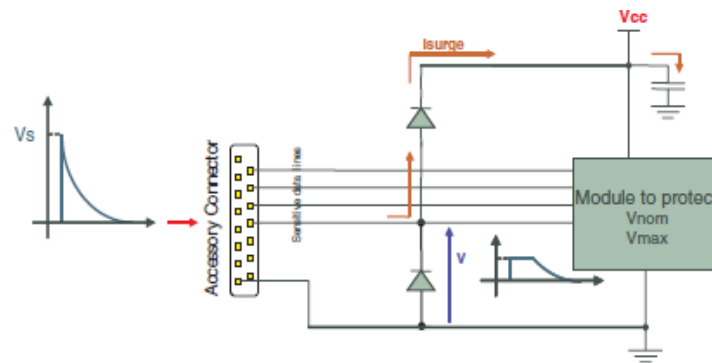
1.2.1 Návrh obvodu s ohledem na EMI a ESD

Pro návrh vhodné ochrany musíme znát parametry daného zařízení a provozního prostředí, mezi ty hlavní patří:

- Jmenovité napětí (V_{nom})
- Rozsah vstupního napájecího napětí, při kterém je zařízení schopno pracovat (V_{min} , V_{max})
- Teplota okolního prostředí (T_{amb})
- Maximální energie přepětového pulzu, na které je dimenzována ochrana

Jedna z častých ochranných opatření proti přepětí je ochrana pomocí diod připojených na napájení. Maximální velikost přepětí je dána velikostí napájecího napětí a úbytku napětí na diodě. Nejlépe to ilustruje obrázek 4.

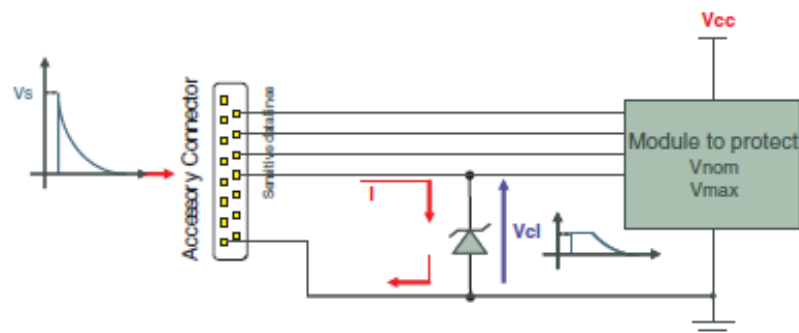
$$V_{MAX} = V_{CC} + V_{fdiody} \quad (1)$$



Obrázek 4: Ochrana sběrnice pomocí diod. Převzato z [8]

Nevýhodou je, že V_{MAX} závisí na velikosti napájecího napětí. Další nevýhodou je kolísání V_{CC} , proto se k této ochraně přidává kondenzátor (typicky 100nF), který zachytí energii přepěťového pulzu. Z principu se tato aplikace nedá použít na napájecí vedení.

Další možností je ochrana pomocí supresorové diody (transilu). Velkou výhodou je, že V_{MAX} nezávisí na napájecím napětí. Při výběru součástky se zaměřujeme na velikost V_{BR} (breakdown voltage), při kterém se transil otevře a na velikost V_{CL} . Velikost zbytkového napětí (V_{CL}) na transilu je dáno velikostí a strmostí nárůstu špičkového proudu I_{PP} . Další výhodou je použití jednoho transilu (oproti dvěma diodám) pro kladná i záporná přepětí. Při záporném přepětí se transil otevírá jako běžná dioda (napětí nižší než V_{BR}), a proto existují antisériová zapojení dvou transilů. Aplikace transilu je uvedena na obrázku 5.

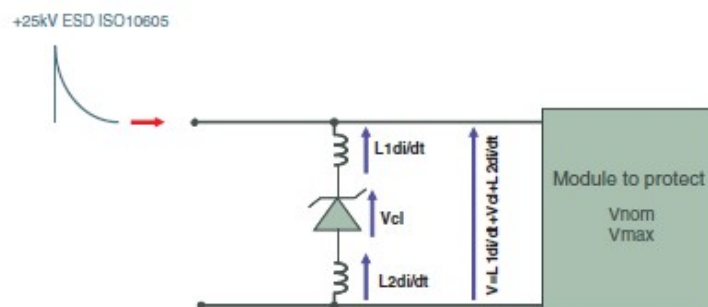


Obrázek 5: Ochrana sběrnice pomocí transilu. Převzato z [8]

Transil lze použít i na napájecí vedení. Pokud používáme transil na ochranu sběrnice, musíme brát v potaz jeho parazitní kapacitu, abychom negativně neovlivnili parametry vedení na sběrnici.

1.2.2 Návrh desky plošných spojů proti EMI a ESD

Při návrhu je vhodné umístit součástky pro ochranu co nejbližně chráněnému zařízení. Je to z důvodu parazitní kapacity a indukčnosti cest na DPS. Při špatném návrhu se nám díky parazitní indukčnosti cest dostane k chráněnému zařízení vyšší přepětí než na které je navrženo, protože transil zareaguje příliš pozdě (viz obrázek 6 a příklad dole).



Obrázek 6: Parazitní indukčnosti na DPS. Převzato z [8]

Př: $I = 12,5A$, $L_1 = 5nH$, $L_2 = 3nH$

$$\frac{di}{dt} = \frac{12,5}{10^{-9}} = 12,5 \cdot 10^{-9} A \cdot s^{-1} \quad (2)$$

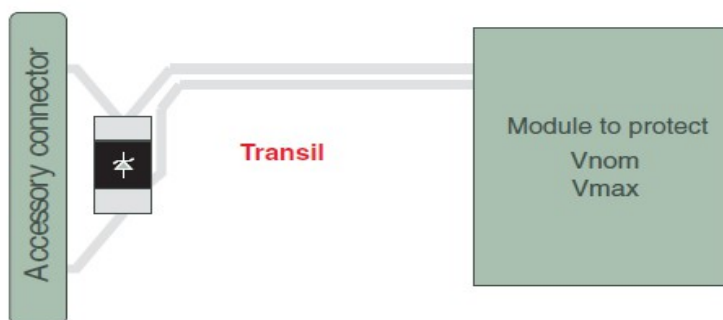
$$L1 \frac{di}{dt} \approx 62,5V \quad (3)$$

$$L2 \frac{di}{dt} \approx 37,5V \quad (4)$$

$$V = 62,5 + 37,5 = 100V \quad (5)$$

Celkové přepětí na zátěži je 100V!

Jak je patrné z předchozího příkladu, i při správných součástkách nám nemusí ochrana fungovat správně. Abychom zamezili tomuto problému, připojíme přívodní cesty přímo k transilu a od něj pak vyvedeme cesty k chráněnému zařízení. Snažíme se také o co nejmenší velikost smyčky vedení z důvodu indukování rušivého napětí. Nejlépe to ilustruje následující obrázek 7.



Obrázek 7: Doporučený design cest na DPS pro nejlepší ochranu zařízení. Převzato z [8]

Ochrana automobilové elektroniky proti EMI a ESD je vždy důležitou součástí návrhu. Elektronika musí spolehlivě fungovat ve velmi nepříznivém prostředí, kde hrají svoji roli i další faktory (teplota, vlhkost, vibrace). Automobilová elektronika prochází v této oblasti náročnými testy, jsou zde kladeny vyšší požadavky na odolnost než u běžné např. spotřební elektroniky. Při návrhu je dobré držet se návrhových pravidel výrobce automobilu, pokud k nim nemáme přístup, držíme se mezinárodních norem. Důležité je vybrat vhodný typ ochrany, umístěný co nejbližně chráněnému zařízení a s co nejmenší plochou smyček vodičů. Pak už jen zbývá otestovat navržený obvod, zdali splňuje všechny nároky v oblasti odolnosti vůči EMI a ESD.

2 Hardwarový návrh jednotky displeje

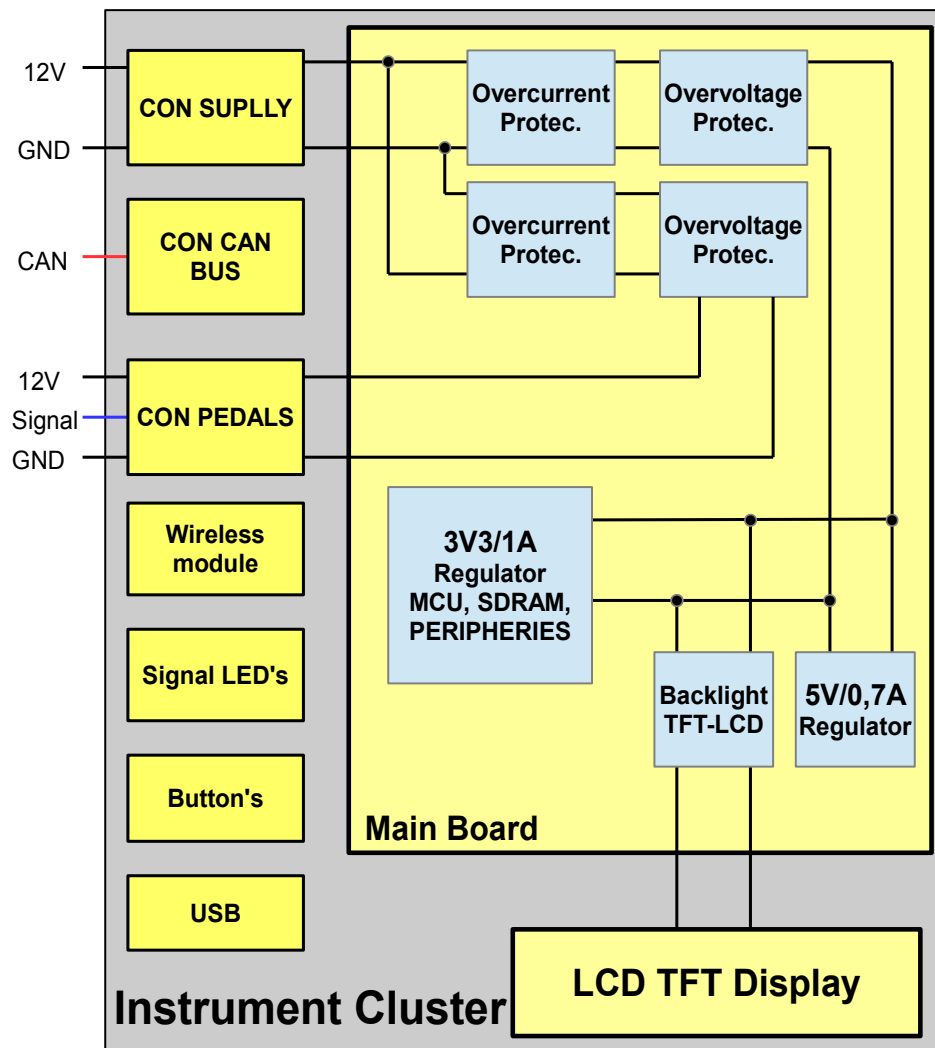
Jednotka displeje bude pracovat v podmínkách, kde jsou kladeny specifické provozní nároky především na odolnost proti elektromagnetickému rušení a odolnost vůči ESD na datových a napájecích vodičích. Na napájecích vodičích je požadována také odolnost proti kolísání napětí, přepětí a proti přepólování. Dále se zde vyskytují velké teplotní výkyvy během provozu zařízení. Z tohoto důvodu je nutné použít součástky, které jsou určeny pro automotive aplikace. Součástky se dělí do čtyř skupin (Grade 0 až 3), podle teplotního rozpětí prostředí, kde budou provozovány. Rozmezí teplot je od $-40\text{ }^{\circ}\text{C}$ až do $+150\text{ }^{\circ}\text{C}$ [7].

Požadavky a funkčnost pro jednotku displeje vychází z projektu elektromotokáry realizovaného Katedrou elektromechaniky a výkonové elektroniky a Katedrou aplikované elektroniky a telekomunikací. Jedním z hlavních požadavků je zobrazování provozních veličin na sedmi palcovém grafickém displeji, které jsou přijímány přes sběrnici CAN od řídicí jednotky motoru. Mezi tyto veličiny patří otáčky motoru, rychlost motokáry a stav nabití akumulátoru. Volby v grafickém rozhraní se provádí tlačítky umístěnými pod displejem. Dalším požadavkem je ukládání těchto veličin do vnitřní paměti a jejich bezdrátový přenos. Poslední důležitou funkcí je snímání polohy brzdového a plynového pedálu pomocí A/D převodníků. Informace o poloze pedálu je odesílána po CAN sběrnici řídicí jednotce motoru, která podle těchto dat upraví požadavek na akceleraci nebo brzdění. Mezi doplňkové periferie patří snímač okolní teploty, MEMS akcelerometr, GPS přijímač, USB rozhraní a výstup pro stavové kontrolky.

2.1 Bloková schémata

Návrh dílčích obvodů vychází z blokových schémat. Návrh je rozdělen na dvě dílčí části a každá z nich reprezentuje jedno blokové schéma. Prvním je blokové napájecí schéma s konektory, kde je zobrazeno napájení a ochrany jednotlivých subsystémů jednotky displeje a také připojení dalších externích periférií a sběrnic. Toto ilustruje obrázek 8.

Jednotka displeje obsahuje jeden společný napájecí konektor. Za ním následuje rozdělení napájení pro jednotku a pro pedály. Ochrana proti nadproudu a proti přepětí je také rozdělena. Brzdový a plynový pedál se připojuje každý na jeden konektor. Na jednotce je napájecí napětí rozděleno na tři větve. První je $3,3\text{ V}$ větev pro napájení mikrokontroléru, SDRAM paměti, převodníků I/U a dalších integrovaných obvodů.



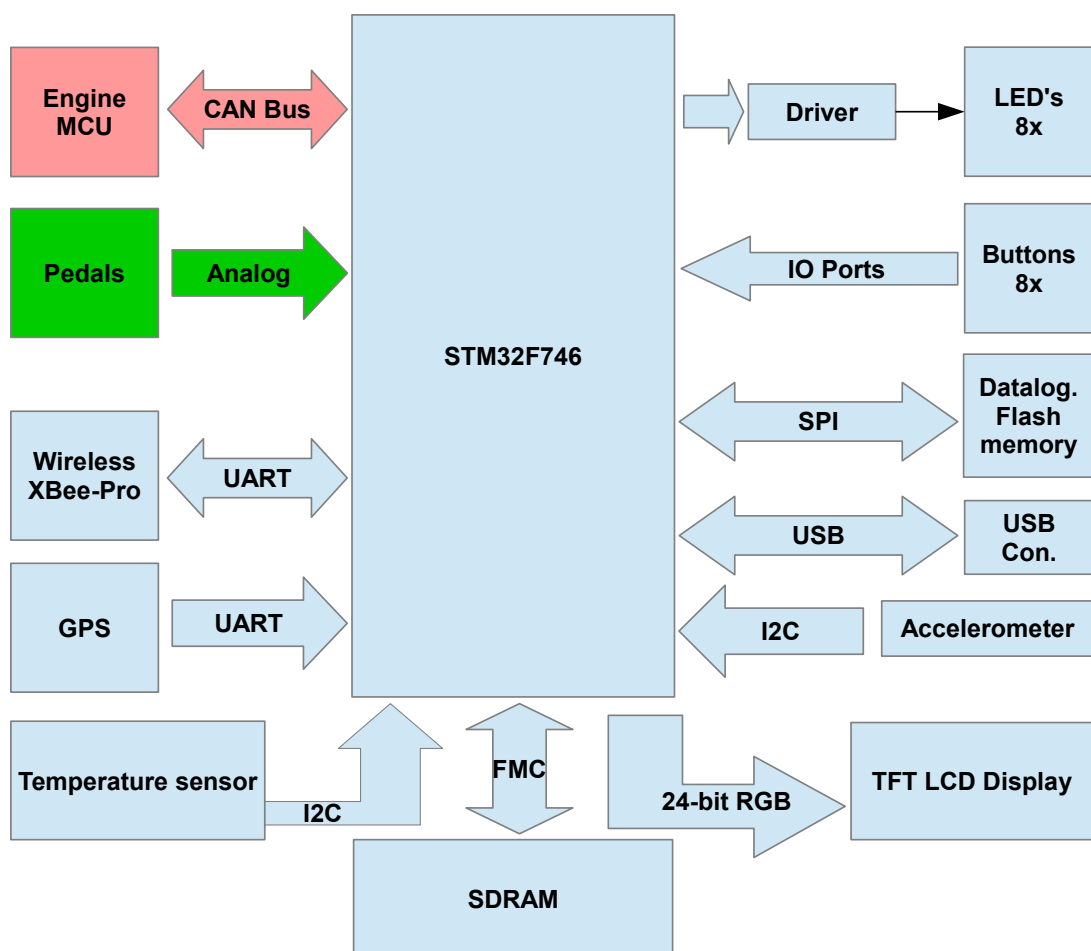
Obrázek 8: Napájecí blokové schéma a konektory přístrojového panelu

Druhá větev 5 V je určena pro napájení stavových kontrolky a USB rozhraní. Třetí větev je zdroj konstantního proudu pro napájení podsvětlení displeje.

Dále jsou zde:

- konektory pro připojení osmi stavových kontrolky
- konektor sběrnice CAN pro komunikaci s ostatními jednotkami
- externí tlačítka pro rozšiřující funkce, pokud by nestačila stávající tlačítka na jednotce,
- bezdrátový modul a USB konektor.
- Konektor na LDC-TFT display, který je připojen pomocí páskového vodiče, je sdruženo napájení a datové vodiče.

Druhé blokové schéma popisuje připojení periférií k mikrokontroléru a typy sběrnic kterými jsou připojeny případně typ řadiče pro obsluhu. Pro řízení celé jednotky je použit mikrokontrolér STM32F746IGT. Více ilustruje obrázek 9. Jednotka displeje komunikuje s řídicí jednotkou motoru po sběrnici CAN. Motorová jednotka poskytuje údaje o aktuálních otáčkách motoru, aktuální výkon a stav nabití akumulátoru. Poloha plynového a brzdového pedálu je převedena pomocí A/D převodníku do digitální podoby a v pravidelných intervalech je odesílána řídicí jednotce motoru. Pro ukládání provozní dat slouží externí Flash paměť, která komunikuje po sběrnici SPI. Dále jsou také provozní data odesílána rádiově pomocí bezdrátového modulu XBee-Pro®, který je připojen sběrnici UART. Zobrazování provozních dat na sedmi-palcový displej LCD-TFT probíhá pomocí paralelního 24 bitového rozhraní. Rozšiřitelnost vnitřní SRAM paměti je možná díky externí SDRAM paměti připojené přes řadič FMC (Flexible Memory Controller).



Obrázek 9: Blokové schéma periférií připojených k mikrokontroléru

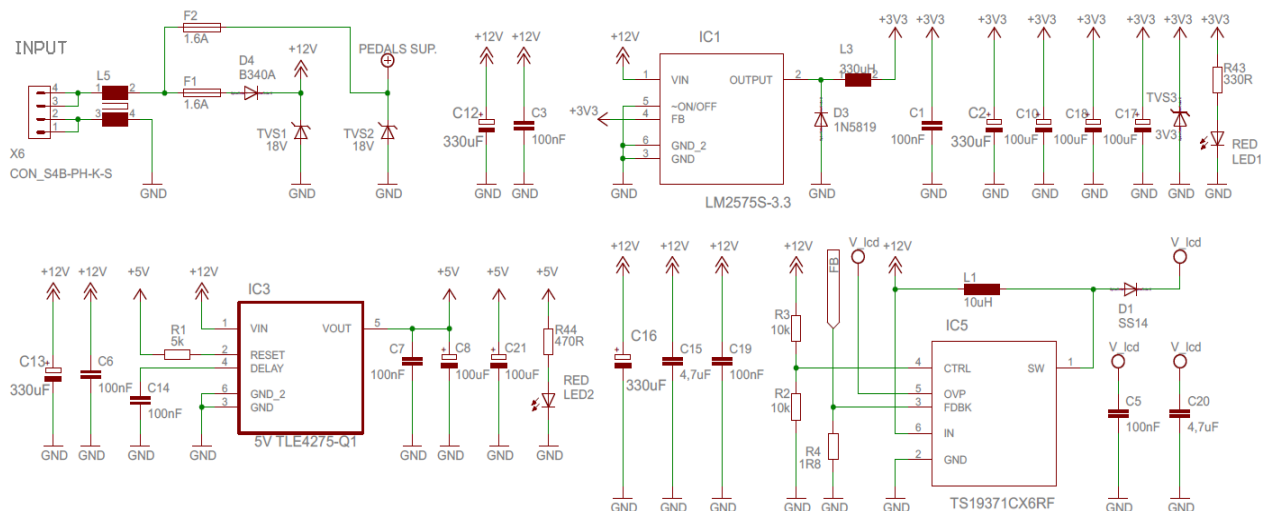
Pro stažení provozních dat nebo přeprogramování mikrokontroléru (viz datový list [12]) je k dispozici USB rozhraní. Stavové kontrolky, reprezentované LED diodami, jsou ovládány jednotlivými výstupními porty přes budič pro větší proudovou zatížitelnost. Na vstupní porty jsou připojena tlačítka pro ovládání a nastavení funkcí jednotky displeje. Jednotka GPS odesílá souřadnice o aktuální poloze po sběrnici UART. Teplotní senzor je na společné I2C sběrnici s akcelerometrem. Obě tyto periférie mají rozdílné adresy na sběrnici, aby s nimi bylo možné komunikovat. Detailní popis jednotlivých periférií a funkčních bloků je uveden v následující podkapitole.

2.2 Popis jednotlivých funkčních bloků

Tato část se zabývá podrobnějším popisem jednotlivých funkčních bloků. V podkapitolách jsou popsány principy zapojení, které vycházejí z katalogových listů výrobců součástek nebo z typických zapojení. Dále je zde uvedeno podrobnější propojení periférií. Popis vychází ze schématu zapojení jednotky displeje, které je uvedeno v příloze 1.

2.2.1 Zdrojová část

Zdrojová část je rozdělena na tři napěťové úrovně a navíc se zde nachází obvody ochrany. Za vstupním napájecím konektorem X6 je umístěna tlumivka pro filtraci proudových špiček. Za tlumivkou následuje rozdělení napájení pro jednotku displeje a pro pedály. Ochranu proti přepětí zajišťují supresorové diody TVS1 a TVS2. Ochranu proti nadproudu jsou pojistky F1 a F2. Ochrana proti přepólování napájecího napětí je realizována Shottkyho diodou (D4) se závěrným napětím 40 V. Pro lepší pochopení slouží obrázek 10.



Obrázek 10: Schéma zapojení zdrojové části

Napájecí větev 5 V je určena pro napájení stavových kontrolky a USB rozhraní. Realizována je pomocí lineárního regulátoru TLE4 275-Q1 s maximálním proudovým

výstupem 0,7 A. Regulátor spadá do kategorie LDO (regulátor s nízkým úbytkem) a je určen pro automotive aplikace. Dokáže správně fungovat již od 5,5 V napájecího napětí a rozsah provozních teplot je $-40\text{ }^{\circ}\text{C}$ až $+150\text{ }^{\circ}\text{C}$. Obvod je zapojen podle typického zapojení uvedeného v datovém listu součástky [10]. Za zmínku stojí možnost nastavení zpoždění náběhu výstupního napětí po připojení napájecího napětí pomocí pinu DELAY na obr 10. Dále je zde možnost vypnutí regulátoru pomocí mikrokontroléru pro snížení spotřeby (pin RESET). Ovládání pomocí MCU zde není použito. Samozřejmostí je ochrana proti přetížení, přehřátí a přepólování. Signalizace správné funkce je zajištěna pomocí LED diody (LED2).

Druhou napájecí větví je 3,3 V spínaný regulátor LM2 575S-3.3 s maximální výstupním proudem 1 A. Tato větev pokrývá napájení většiny obvodů na jednotce, a proto není použit lineární regulátor z důvodu vysoké výkonové ztráty. Pokud bychom uvažovali výstupní proud 1 A a napájecí napětí 12 V výkonová ztráta by byla podle rovnice 6 8,7 W. Spínaná varianta dosahuje při stejné zátěži účinnosti 75 %, což je výkonová ztráta 2,5 W. Hodnoty součástek, především velikost tlumivky L3 a typ Shottkyho diody, jsou převzaty z datového listu výrobce společně s typickým zapojením [11].

$$P_{ztrátový} = (U_{CC} - U_{OUT}) \cdot I_{OUT} = (12 - 3,3) \cdot 1 = 8,7 \text{ W} \quad (6)$$

Maximální proudové zatížení 3,3 V větve je součtem maximálních proudů jednotlivých obvodů. Velikost tohoto proudu po sečtení dílčích proudů podle datových listů je 794,25 mA.

$$I_{(\Sigma \text{ MAX})} = I_{MCU} + I_{SDRAM} + I_{FLASH} + I_{(I/U)} + I_{GPS} + I_{XBEE} + I_{(CAN \text{ DRIVER})} + I_{TEMP} + I_{MEMS} \quad (7)$$

$$I_{(\Sigma \text{ MAX})} = 320 + 100 + 20 + 8 + 36 + 270 + 20 + 20 + 0,25 = 794,25 [\text{mA}] \quad (8)$$

Výpočet maximálního proudového zatížení popisují rovnice 7 a 8. Proudová spotřeba jednotlivých obvodů je převzata z datových listů od výrobce [12][17][13][19][23][22][20][25][24]. Regulátor funguje jako snižující měnič v režimu bez přerušovaného proudu. Kondenzátory C2 a C10 slouží k filtraci výstupního napětí regulátoru. Signalizace funkčnosti této napájecí větve je pomocí červené LED diody (LED1). Obvod dále obsahuje ochranu proti zkratu a přehřátí.

Posledním obvodem ve zdrojové části je obvod pro napájení podsvětlení LCD-TFT displeje. Obvod TS19371 funguje jako zvyšující měnič. Je určen pro řízení LED diod pomocí zdroje konstantního proudu. Je schopen řídit až 9 bílých LED diod v sérii a maximální výstupní proud je 650 mA. Velikost procházejícího proudu je určena velikostí odporu R4. Napětí na tomto odporu je porovnáváno s vnitřní napětíovou referencí obvodu o velikosti

95 mV. Výpočet odporu R1 je uveden v rovnici 9. Velikost proudu procházejícího diodami (I_{LED}) je odvozena z datového listu výrobce displeje [15]. Při nominálním proudu $I_F = 60 \text{ mA}$ je napětí na LED diodách 19,2 V. Pro ochranu diod proti přepětí je použit transil TVS5.

$$RI = \frac{U_{REF}}{I_{LED}} = \frac{0,95}{0,53} = 1,8 [V, A; \Omega] \quad (9)$$

Velikost tlumivky a Shottkyho dioda je zvolena podle doporučení výrobce [14]. Obvod dále obsahuje ochranu proti přepětí a možnost řízení intenzity podsvětlení pomocí PWM modulace, ve schématu tato funkce není použita.

2.2.2 Mikrokontrolér a displej

Řízení celé jednotky displeje obstarává 32-bitový mikrokontrolér STM32F746IGT od společnosti ST Microelectronics. Jádrem mikrokontroléru je vysoce výkonný RISC procesor ARM® Cortex®-M7 harvardské architektury, který je určen pro embedded aplikace. Vyznačuje se vysokým výpočetním výkonem, dobrou energetickou účinností a je vhodný pro průmyslové a automotive aplikace [16]. Maximální provozní frekvence procesoru je 216 Mhz. Jádro také obsahuje FPU pro aritmetickologické operace s proměnnými s plovoucí desetinnou čárkou. Velikost programové paměti 1 MB a celková velikost SRAM paměti 320 kB. Z vnitřních periférií jsou využívány integrovaný řadič LCD-TFT s paralelním výstupem, USB rozhraní, CAN rozhraní, audio vstup atd. Všechny ostatní periférie jsou popsány v datovém listu mikrokontroléru [12]. Typ pouzdra je LQFP176. Toto pouzdro je použito z důvodu velkého množství signálů vedoucích do mikrokontroléru a s ohledem na ruční pájení součástky na DPS.

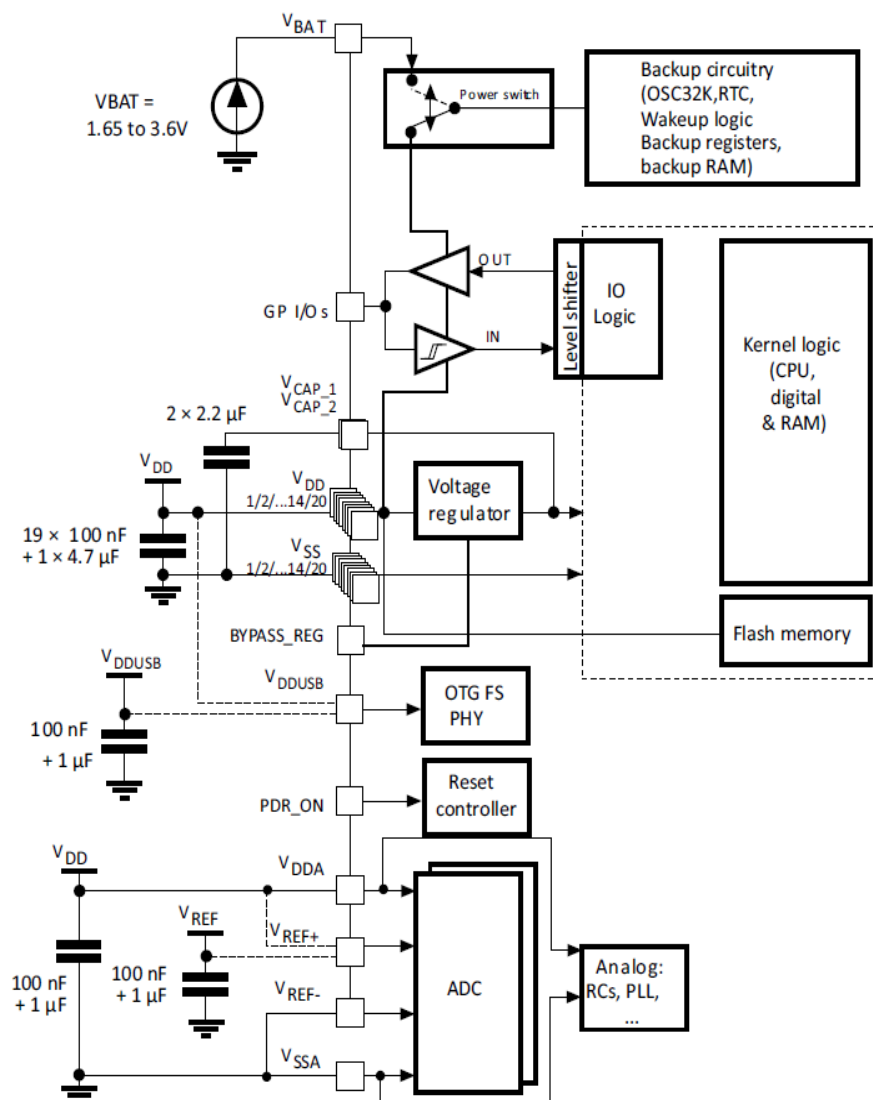
Displej

Samotný sedmipalcový displej MIDAS je připojen k jednotce skrze konektor na plochý kabel X3. Typ použitého displeje má široký rozsah provozní teploty od -20°C do $+70^\circ\text{C}$, a proto je vhodný pro použití v jednotce displeje. Je zde připojeno jak napájení elektroniky a podsvětlení, tak datové vodiče. Vodiče pro přenos obrazu jsou rozděleny na přenos tří základních barev (červená, zelená, modrá) v barevné hloubce osm bitů, celkem tedy 24 vodičů. Ostatní čtyři vodiče slouží pro synchronizaci obrazu a přenos hodinového signálu. Celkem je to 28 datových vodičů. Schéma zapojení konektoru pro připojení displeje je uvedeno v datovém listu výrobce displeje [15]. Konektivita s displejem ze strany mikrokontroléru je řešena podle tabulky 10 v odkazu [12]. Každý pin z pouzdra, kromě napájení, je připojen na vstupně výstupní port, který má definovanou alternativní funkci kam je připojena vnitřní periférie. Většinou je možno zvolit si vyvedení jednoho signálu z několika

pinů součástky. Je to z toho důvodu, že na jednom portu je více alternativních funkcí a aktivací jedné vnitřní periférie bychom si zablokovali použití jiné. Toto je potřeba při návrhu schématu zohlednit, aby se daly připojit všechny potřebné signály pro funkčnost jednotky podle zadání. Provozní teplota je -20°C až $+70^{\circ}\text{C}$. Pinout mikrokontroléru je v příloze 2.

Mikrokontrolér

Pro správnou funkci mikrokontroléru je potřeba dostatečně dimenzovat napájení. Ke každému napájecímu pinu je potřeba umístit co nejbližší blokovací kondenzátor. Napájení je rozdělené na analogovou a digitální část, tyto dvě větve rozděluje filtr tvořený tlumivkou L2 (viz schéma v příloze 1). Je zde kvůli potlačení proudových špiček v digitální části, které by mohli rušit analogové obvody. Napájecí schéma mikrokontroléru popisuje obrázek 11. Jsou zde vyvedeny dva piny V_{CAP_1} a V_{CAP_2} , určené pro správnou funkci vnitřního napěťového regulátoru. K těmto pinům jsou připojeny kondenzátory $2 \times 2,2 \mu\text{F}$. Pin V_{BAT} slouží k připojení zálohovací baterie, ze které se napájí záložní registry pokud není k dispozici hlavní napájení.



Obrázek 11: Napájecí schéma STM32F746IGT. Převzato z [16]

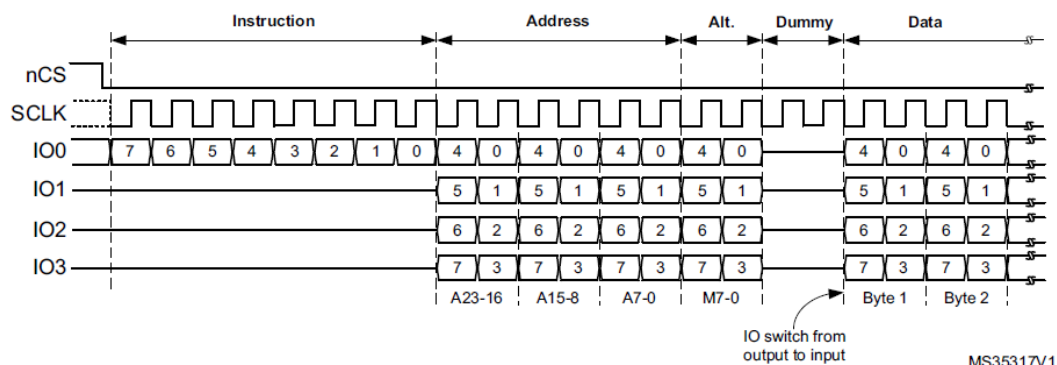
Zdrojem hodinového signálu jsou dva krystalové oscilátory Q1 a Q2. První z nich o kmitočtu 8 MHz je zdrojem hodin pro celé pro jádro a všechny periferie mikrokontroléru. Krystal Q2 (32,768 kHz) je zdroj hodin pro RTC (Real-time Clock) periférii, která obsahuje registry pro zpracování času, data a režimy snížené spotřeby. Vybrané registry mohou pracovat i bez napájení pouze na zálohovací baterii CR2032.

Pro rozšíření vnitřní SRAM paměti je k dispozici 64 Mbitová SDRAM v samostatném obvodu. Pro grafické aplikace není vnitřní paměť (320 kB) dostačující a aktuální obraz na displeji musí být uložen v externí paměti. Při použití displeje o rozlišení 800x480 pixelů a barevné hloubce 16 bitů potřebujeme paměťový prostor o velikosti 768 kB. Výpočet velikosti paměti je uveden v rovnici 10. Při barevné hloubce 24 bitů se již dostáváme na hodnotu 1 152 kB. Paměť IS42S16400J je spojena s MCU přes 12 adresových vodičů, 8 řídicích signálů a 2 signály pro výběr paměťové banky. Počet bank v paměti jsou čtyři. Šířka paměťové sběrnice je 16 bitů, což je stejný počet vodičů. Celkem potřebujeme k propojení 38 datových vodičů. Stejně jako u mikrokontroléru musí být u každého napájecího pinu blokovací kondenzátor 100 nF pro správnou funkčnost paměti [17].

$$\text{Obsazená paměť při 16b RGB} \approx 800 \cdot 480 \cdot 16 = 6\,144\,000 \text{ b} = 768 \text{ kB} \quad (10)$$

Mikrokontrolér ovládá SDRAM paměť skrze softwarově nastavitelný flexibilní paměťový řadič (FMC). Tento řadič je vnitřní periférií a kromě SDRAM paměti dokáže ovládat NOR/PSRAM a NAND flash paměti.

Pro ukládání provozních dat slouží NOR flash paměť Micron (N25Q128A) o velikosti 128 Mb. Připojena je pomocí sběrnice SPI, není to ovšem standardní přenosový protokol. Používá se tzv. Quad-SPI protokol, kde nejsou pouze dva datové vodiče pro vstupní a výstupní data ze strany masteru, ale čtyři datové vodiče. Data jsou přijímána nebo vysílána současně skrze všechny datové vodiče. Rámec zprávy jako příklad je uveden na obrázku 12. Tomuto protokolu se také v některé literatuře říká Serial Quad I/O Protocol.



Obrázek 12: Rámec zprávy Quad-SPI protokolu. Převzato z [18]

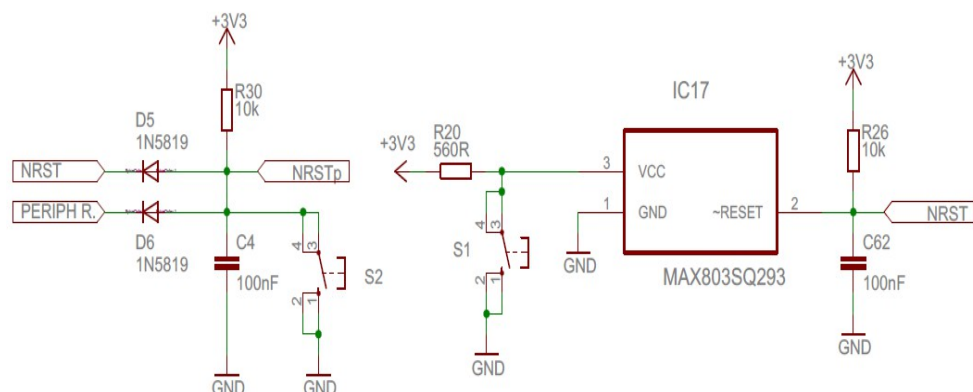
MS35317V1

Jako první je odeslán typ instrukce, kterou má paměť provést. Většina instrukcí se odesílá jednobitově pouze prvním datovým vodičem. Lze ale v registrech nastavit odesílání čtyř bitů najednou, jako je tomu u následujícího pole adresy. Po adrese registru následuje pole alternativních bitů, zpravidla řídicí signály pro ovládání paměti. Pak je zde mezera, kdy není nic vysíláno. Je nastavitelná v rozmezí 1 až 31 hodinových pulzů a slouží pro přepnutí výstupních dat na vstupní data. Dále také flash paměť potřebuje určitý čas pro výpis dat na sběrnici. Nakonec jsou přijímána data. Během jednoho hodinového cyklu jsou přijaty čtyři bity. Díky tomuto protokolu pro komunikaci s flash pamětí je přenosová rychlost čtyřnásobná.

Konektivita s mikrokontrolérem je zajištěna USB (konektor X1) rozhraním a SWD/JTAG (konektor X2). První zmíněné rozhraní je určeno především pro komunikaci s Flash pamětí. Pro úpravě zapojení, které zahrnuje přepojení pinu BOOT0 na +3,3 V, je možno přehrát vnitřní programovou paměť kontroléru. Ladění programu přes USB není možné, proto je tu druhé rozhraní SWD/JTAG, které umožňuje nahrávání programu i ladění.

Reset mikrokontroléru po připojení napájecího napětí se provede automaticky vnitřním resetovacím obvodem. Pro hlídání správné velikosti napájecího napětí a tím i správné funkce kontroléru je vnitřní resetovací obvod navíc doplněn obvodem MAX803SQ293. Rozhodovací úroveň napětí pro spuštění resetu je v rozmezí 2,89 V až 2,96 V [21]. Resetovací obvod mikrokontroléru je také spojen s resetovacími obvody GPS modulu a XBee-Pro® modulu. Tyto dvě periférie se mohou resetovat samostatně pomocí tlačítka S2. Reset celé jednotky je možný tlačítkem S1. Periférie lze také vyresetovat softwarově pomocí signálu PERIPH_R na portu PI1. Resetovací obvod je na obrázku 13.

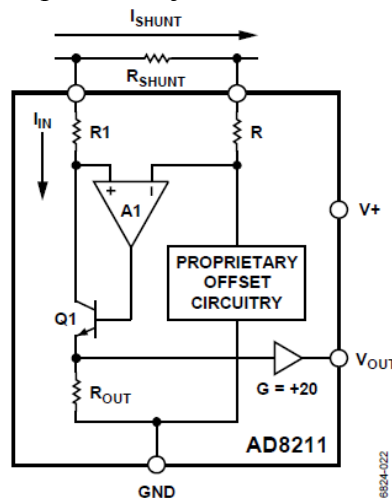
LED stavové kontrolky (LED1 až LED8) jsou ovládány výstupními porty skrze tranzistorové pole ULN2803 s maximálním výstupním proudem 500 mA pro každý kanál. Tlačítka pro ovládání jednotky (BUTTON1 až BUTTON8) jsou spojena se vstupními porty mikrokontroléru přes Schmittův klopný obvod, který upravuje výstupní signál, aby nedošlo k falešnému stisku tlačítka, které by mohlo vyvolat nežádoucí rušení.



Obrázek 13: Resetovací obvod

2.2.3 I/U převodníky

Senzory sešlápnutí brzdového a plynového pedálu pracují s proudovým výstupem o velikosti 4 mA až 20 mA. Pro vyšší bezpečnost je snímání a vyhodnocování obou pedálů zdvojnásobeno. Pedály jsou připojeny dvěma 6-ti pinovými konektory, které zajišťují i jejich napájení. Celkem jsou to 4 převodníky I/U (IC6 až IC9 v příloze 1) Analog Devices AD8211. Tento obvod měří úbytek napětí na externím rezistoru (R6, R8, R10, R12) pomocí vnitřního rozdílového zesilovače. Velikost napětí na výstupu je určeno procházejícím proudem I_{SHUNT} , odporem R_{SHUNT} a ziskem výstupního zesilovače, který je roven dvaceti [19]. Lépe to popisuje obrázek 14 a výpočet výstupního napětí ukazuje rovnice 11 a 12.

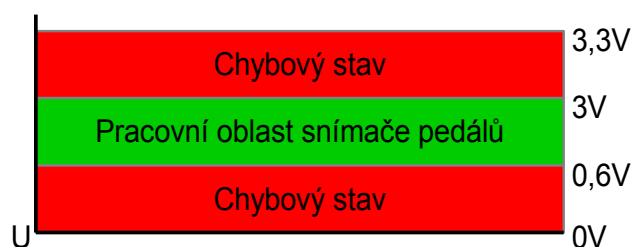


Obrázek 14: Vnitřní schéma obvodu AD8211. Převzato z [19]

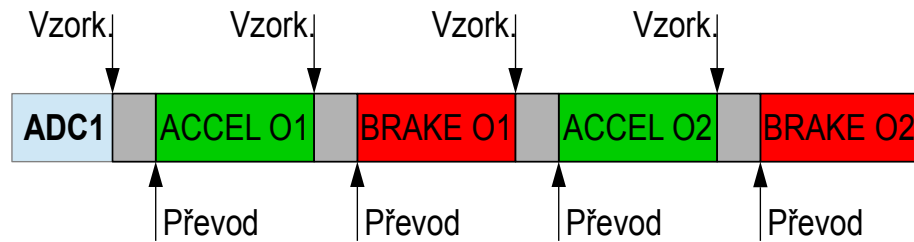
$$V_{(OUT\ MIN)} = (I_{SHUNT} \cdot R_{SHUNT}) \cdot 20 = (0,004 \cdot 7,5) \cdot 20 = 0,6 [A, \Omega ; V] \quad (11)$$

$$V_{(OUT\ MAX)} = (I_{SHUNT} \cdot R_{SHUNT}) \cdot 20 = (0,02 \cdot 7,5) \cdot 20 = 3 [A, \Omega ; V] \quad (12)$$

Z předchozích výpočtů je vidět minimální a maximální hodnota pro A/D převodník mikrokontroléru. Měřicí rozsah A/D převodníku je 0 V až 3,3 V. Rezerva mezi $V_{OUT\ MIN}$ a 0 V respektive $V_{OUT\ MAX}$ a 3,3 V je z důvodu rozpoznání chybového stavu snímače pedálu. Pokud je měřené napětí mimo rozsah $V_{OUT\ MIN}$ a $V_{OUT\ MAX}$ musí řídicí software mikrokontroléru přejít do bezpečného stavu. Ilustruje to obrázek 15.



Obrázek 15: Rozsah provozního napětí snímače pedálů



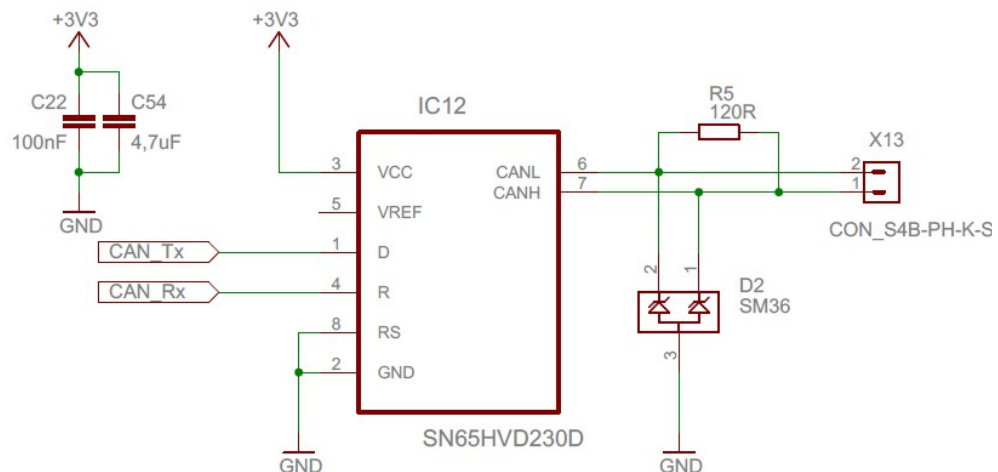
Obrázek 16: Princip vzorkování pedálů

Maximální rozlišení převodníku je 12 bitů, lze nastavit i nižší rozlišení 10 a 8 bitů. A/D převod každého z pedálů probíhá postupně jediným A/D převodníkem. Nejdříve se navzorkuje vstupní signál plynového pedálu ACCEL O1, potom Brake O1. Následuje ACCEL O2 a Brake O2. Lépe to popisuje obrázek 16. Vzorkovací frekvence a její nastavení je popsáno v kapitole 3.2.3 .

2.2.4 Komunikace po sběrnici CAN a bezdrátová komunikace

Nejdůležitější sběrnici pro komunikaci je CAN. Díky ní komunikuje jednotka displeje s řídicí jednotkou motoru, které posílá zprávy o poloze brzdového a plynového pedálu. Řídicí jednotka motoru pak podle toho řídí výkon motoru. Zároveň jednotka displeje přijímá zprávy o otáčkách motoru, jeho výkonu, stavu baterie a případná chybová hlášení. Mikrokontrolér obsahuje dva řadiče sběrnice CAN. Oba mají k dispozici jeden vysílací zásobník FIFO a dva přijímací zásobníky FIFO. V každém zásobníku mohou být uloženy až tři kompletní zprávy. Tyto zásobníky jsou řízeny kompletně hardwarově [18].

Fyzickou vrstvu pro připojení na sběrnici zajišťuje vysílač SN65HVD230D. Ochrana proti ESD na sběrnici je pomocí transilu D2 (obrázek 17). Terminace sběrnice je pomocí odporu R 5 o velikosti 120 Ω . Přenosová rychlost sběrnice CAN je nastavena na 500 kbit/s.

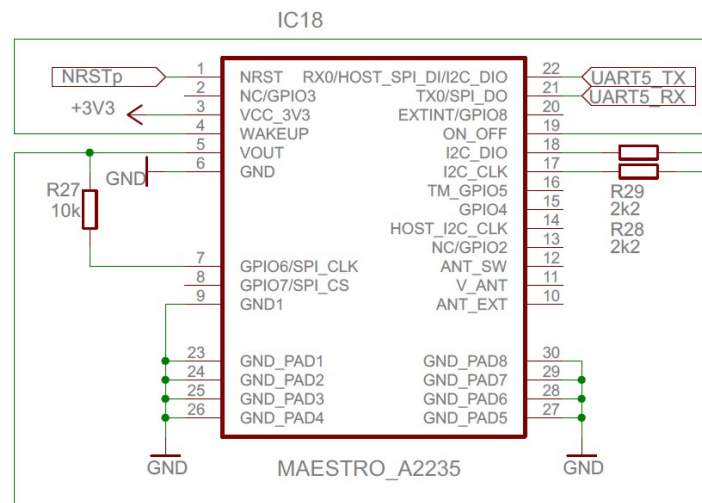


Obrázek 17: Schéma zapojení budiče CAN

Bezdrátovou komunikaci elektromotokáry se základnou zajišťuje bezdrátový modul XBee Pro® osazený na jednotce displeje. Komunikace probíhá na kmitočtu 2,4 GHz a dosah ve volném prostoru je 3200 metrů, v uzavřeném prostoru 90 metrů [22]. Komunikace s kontrolérem je po rozhraní UART. Ostatní periférie tohoto modulu nejsou použity. XBee Pro® je zde pouze ve funkci bezdrátového vysílače/přijímače.

2.2.5 Ostatní periférie

Pro určení polohy motokáry slouží modul GPS Maestro A2235. Tento přijímací modul má v sobě zabudován navigační čip SiRF Star IV a integrována je i anténa pro příjem. Komunikace s mikrokontrolérem je po rozhraní UART. Modul posílá informace o poloze ve formě NMEA zpráv. Schéma zapojení vychází z uživatelského manuálu výrobce. Je vytvořeno tak, aby se modul sám zapnul a začal přijímat GPS signál bez dalšího nastavování [23]. Toto schéma je na obrázku 18 dole.



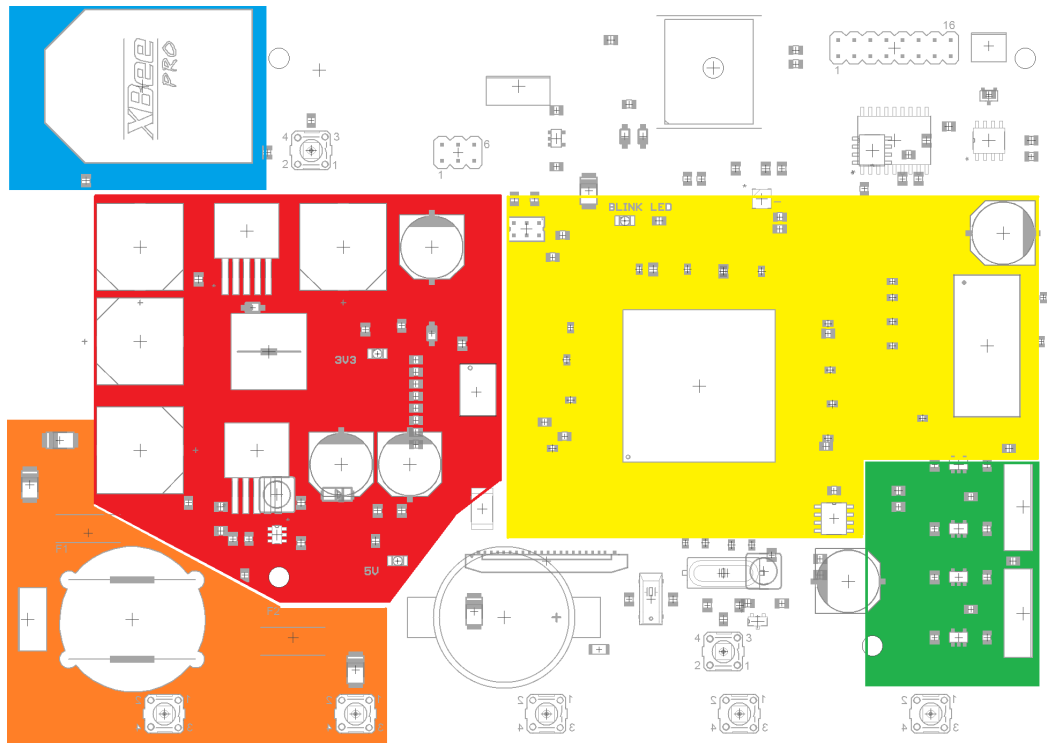
Obrázek 18: Schéma zapojení GPS modulu

Zrychlení motokáry je snímáno tří-osým MEMS akcelerometrem LIS331HH. Nastavení a čtení hodnot z akcelerometru probíhá po I2C sběrnici. Měřicí rozsah je na nastavitelný ve třech úrovních ($\pm 6g$, $\pm 12g$, $\pm 24g$). Tento obvod je velmi kompaktní a obsahuje vlastní diagnostický obvod. Nevyžaduje žádné další externí součástky kromě blokovacích kondenzátorů na napájecích pinech a rezistoru pro nastavení adresy. Adresa se nastavuje pomocí pinu SDO/SA0 a jde jím nastavit poslední nejvyšší bit adresy [24]. Adresa akcelerometru je 0011000. Celková velikost adresy je 7 bitů.

Posledním integrovaným obvodem je digitální teplotní senzor LM75BIM. Připojen je na I2C sběrnici společně s akcelerometrem. Adresa se nastavuje hardwarově pomocí adresovacích pinů A0, A1, A2. Kombinací těchto pinů, připojených na napájecí napětí nebo na zem, dosáhneme změny posledních tří nejvyšších bitů adresy [25]. Adresa je 1001101.

2.3 Návrh desky plošných spojů a rozvržení funkčních bloků

Návrh desky plošný spojů (DPS) se drží návrhových pravidel a doporučení jednotlivých výrobců integrovaných obvodů. Nejdůležitější je správné rozmístění jednotlivých komponent, aby spoje byly co nejkratší a nezvyšovaly se tím parazitní kapacity a indukčnosti. Obrázek 19 ilustruje rozmístění jednotlivých funkčních bloků na DPS. Rozměry plošného spoje jsou 166x117,8 mm. Plošný spoj má celkem čtyři vrstvy, které se nazývají TOP, IN2, IN3 a BOT.



Obrázek 19: Rozmístění jednotlivých bloků na DPS

Oranžová barva ohraničuje blok vstupních ochran na napájení. Je zde filtrační tlumivka na společném jádře (1,2 mH/3A), SMD pojistky o hodnotě 1,6 A a transily 18 V. Všechny součástky z tohoto bloku jsou umístěny ve vrstvě TOP. Následuje červená zdrojová část, která je osazena z obou stran. Na straně BOT je osazen zdroj proudu pro podsvětlení displeje. Spojení zdroje s konektorem displeje je vedeno ve stejné vrstvě. Zbylé dva zdroje (+3,3V, +5V) jsou na straně TOP. Rozvod jejich napájecího napětí po plošném spoji je přes vnitřní vrstvu IN2. Rozvod napájení +3,3 V je rozdělen v této vrstvě na dvě větve. Není vhodné je spojovat, protože by jsme tak vytvořili smyčku na které by vznikalo rušení (viz příloha 4). První větev napájí polovinu mikrokontroléru, SDRAM paměť, budič CAN, teploměr, akcelerometr a GPS modul. Druhá větev potom zbytek kontroléru, displej, Flash paměť a analogovou část obvodu.

Vysokofrekvenční číslicové obvody (mikrokontrolér, SDRAM, displej, Flash paměť) jsou označeny žlutou barvou. Komunikace mezi nimi probíhá v řádu desítek megahertzů, a proto se musí dbát na vhodné vedení cesty daného signálu. To znamená, že vodiče musí mít vzhledem ke shodnému zpoždění signálu na všech cestách přibližně stejnou délku. Také by měly vést co nejvíce přímo bez zbytečných smyček, pokud je to možné. Proto je umístění periférií vzhledem ke kontroléru na straně, ve které signálové cesty vychází nejpříznivěji s ohledem na ostatní součástky a cesty na plošném spoji. Propojení této části je vedeno ve všech čtyřech vrstvách, součástky jsou na straně TOP kromě konektoru displeje.

Zelená barva označuje analogovou část s převodníky I/U. Převodníky jsou umístěny vpravo dole na straně TOP. V žádné vrstvě pod nimi nevede signálová cesta z jiných obvodů. Je to z důvodu co nejmenšího rušení ze strany číslicových obvodů. Stejně tak signály od převodníků vedou co nejkratší cestou na příslušné piny A/D převodníku mikrokontroléru. Křížení s číslicovými signály u kontroléru mezi jednotlivými vrstvami je vedeno co nejvíce v kolmém směru, Je to z důvodu potlačení kapacitní vazby.

Poslední částí je bezdrátový vysílač XBee Pro® označený modrou barvou. Ten je umístěn vlevo nahoře. V jeho blízkém okolí nejsou žádné další obvody, aby jeho vysílací anténa nezpůsobovala jejich rušení. Vysílač je v patici, z důvodu případné snadné výměny. Pokud bychom chtěli změnit například vysílací kmitočet nebo výkon vysílače, lze to provést jednoduchou výměnou celého modulu.

3 Řídící software jednotky displeje

Řídící software jednotky displeje uložený v programové paměti mikrokontroléru musí být schopen zobrazovat provozní data, řídit všechny externí periferie na jednotce a zároveň komunikovat s řídicí jednotkou motoru. Vzhledem k tomuto množství hardwaru je jasné, že rozsáhlý bude i software. Musí se definovat priorita jednotlivých úloh pro správnou funkci a ošetřit události při, kterých vznikne chyba. Při vzniku chyby se musí přejít do bezpečného stavu a informovat o jejím vzniku. Při startu programu je také potřeba definovat postupnou inicializaci jednotlivých komponent, které se budou zapínat postupně za sebou. Například musíme nejdříve inicializovat řadič displeje, alokovat potřebný paměťový prostor a poté můžeme začít zobrazovat text a proměnné na displeji. Software je vytvořen ve vývojovém prostředí μ Vision5 od společnosti ARM.

3.1 Platforma mikrokontrolérů STM32F7

Řada mikrokontrolérů STM32F7 patří do skupiny velmi výkonných kontrolérů pro vestavěné (embedded) aplikace. V kontroléru je 32-bitový procesor ARM® Cortex®-M7. Jeho základní popis je v kapitole 2.2.2. Zde se budeme zabývat vnitřní systémovou architekturou.

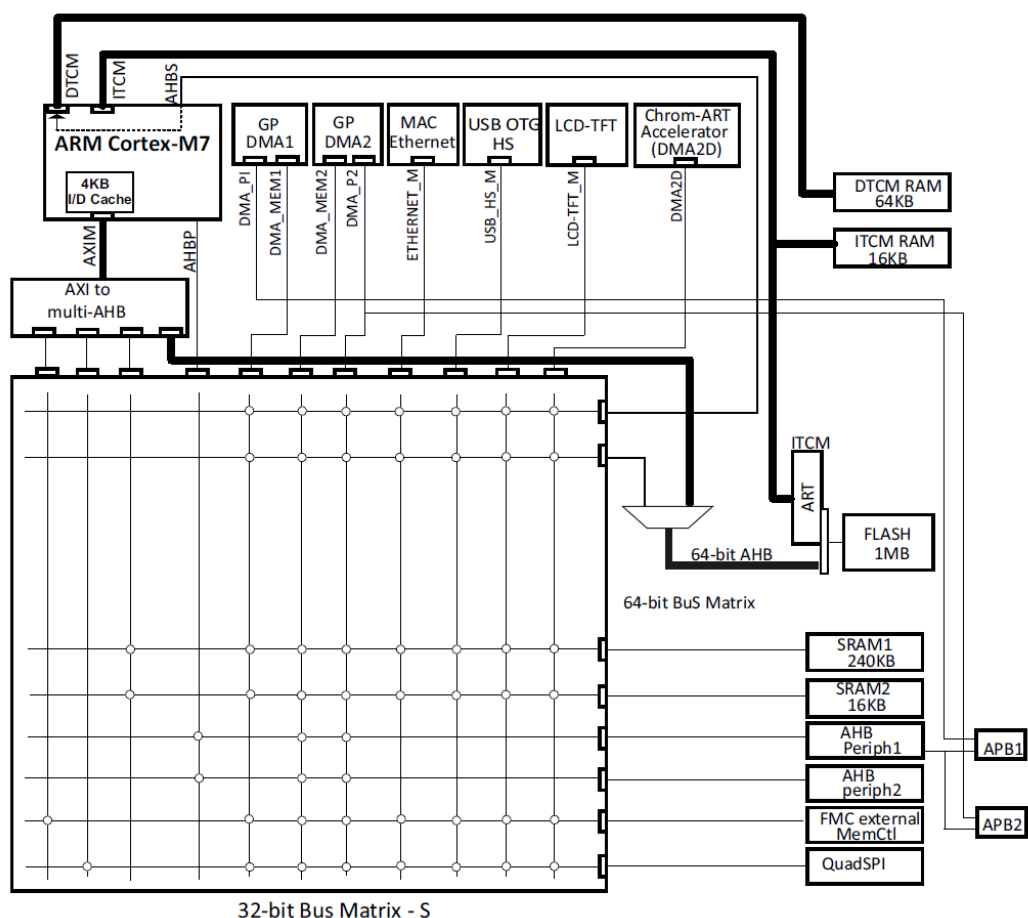
Procesor Cortex®-M7 umožňuje velmi rychlou odezvu na příchod přerušování při běhu programu, vysoký výkon a zpětnou kompatibilitu s ostatními procesory Cortex®-M. Skládá se ze šestistupňového superskaláru. To znamená, že většina instrukcí probíhá současně, včetně instrukcí sloužících pro načítání a ukládání do paměti. Je to umožněno díky vícenásobnému paměťovému rozhraní. Mezi klíčové vlastnosti patří predikce skoku při běhu programu, podpora instrukční sady Thumb®, hardwarové dělení, jednotka pro práci s plovoucí desetinnou čárkou atd [26].

Systémová architektura mikrokontroléru STM32F7 je rozdělena na dva podsystémy:

- TCM rozhraní (DTCM a ITCM) pro komunikaci s RAM pamětí a Flash pamětí
- AHBS rozhraní pro komunikaci DMA řadiče s RAM pamětí přes TCM rozhraní
- AHBP rozhraní pro komunikaci s perifériemi mikrokontroléru
- Převodník sběrnice AXI na multi-AHB sběrnici, který převádí AXI4 protokol na AHB-Lite protokol a obsahuje:
 - 1x 64-bitové AHB rozhraní pro připojení vestavěné flash paměti
 - 3x 32-bitové AHB rozhraní pro připojení na maticovou sběrnici

- Multi-AHB maticová sběrnice pro propojení jednotlivých zařízení obsahuje:
 - 32-bitovou multi-AHB maticovou sběrnici
 - 64-bitovou mutli-AHB maticovou sběrnici pro propojení DMA řadiče a flash paměti

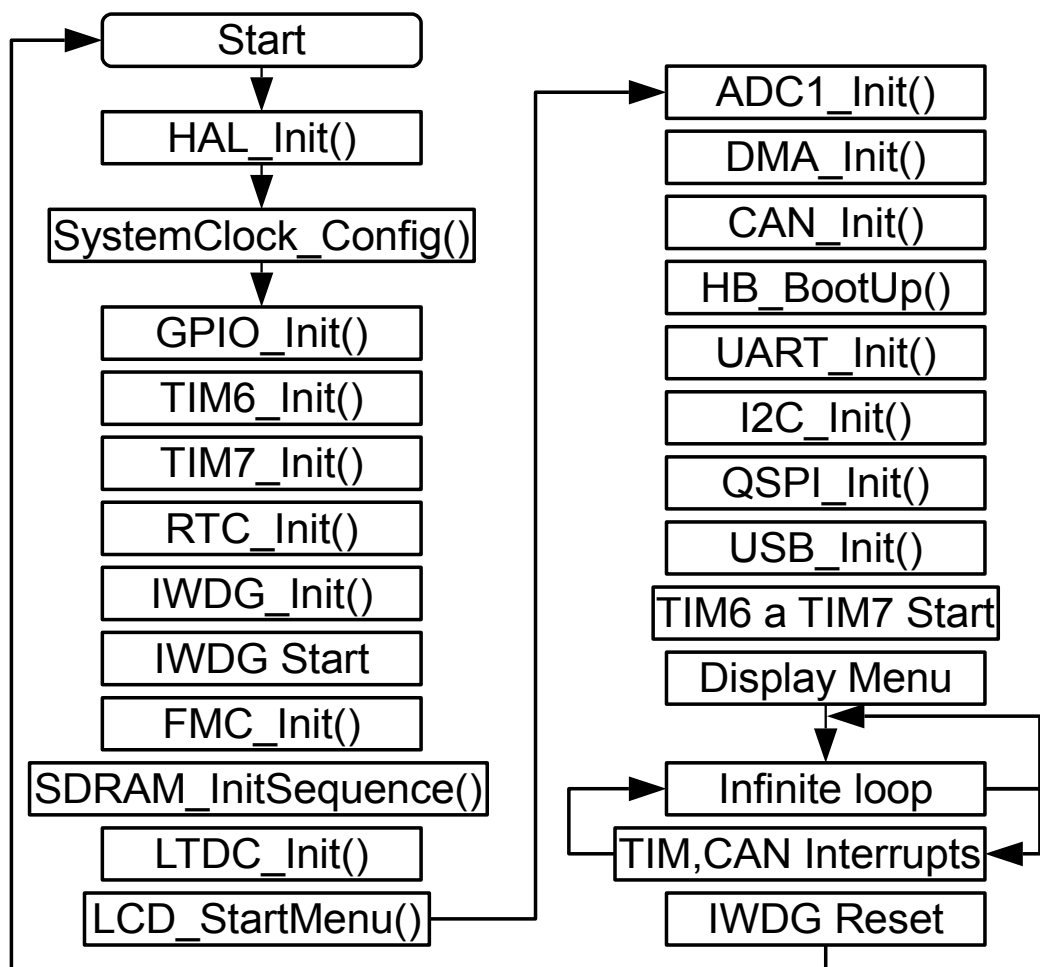
Systémovou architekturu nejlépe popisuje obrázek 20 dole. Jak je na obrázku vidět, přístup k Flash paměti je možný dvěma cestami. To umožňuje zpracovávat instrukce programu rychleji díky přenosu po instrukční sběrnici ITCM o šířce 64 bitů. Využívá se k tomu ART akcelerátor. Ten je k dispozici pouze na této sběrnici. Při běžné práci s pamětí musí procesor čekat určitý počet hodinových pulzů (tzv. Wait stavy), než získá přístup k datům. Je to způsobeno rozdílnými pracovními frekvencemi procesoru a paměti. ART Akcelerátor obsahuje vyrovnávací paměť pro instrukce a skoky v programu a tím zvyšuje rychlost běhu programu. Díky tomu je přístupová doba rovna 0 wait stavům při běhu programu až do frekvence 216 Mhz. Výpočetní výkon při tomto kmitočtu je 462 DMIPS, to odpovídá výpočetnímu výkonu 2,14 DMIPS/1 Mhz [12][18].



Obrázek 20: Systémová architektura STM32F74xxx. Převzato z [18]

3.2 Vývojový diagram programu

Program pro řídicí jednotku displeje obsahuje hlavní funkci Main, kde se nejdříve inicializují jednotlivé dílčí funkce programu. Jako například inicializace pinů, nastavení hodin, nastavení časovačů, nastavení vnitřních a externích periférií atd. Poté následuje nekonečná smyčka programu, aby se vykonávání programu nezastavilo, protože zde není zaveden žádný operační systém. V této smyčce se zjišťuje, zdali nebylo stisknuto tlačítko a provádí se zde další potřebné operace. Instrukce co se provádějí pravidelně, například odesílání zpráv po sběrnici CAN nebo čtení A/D převodníku, jsou vykonávány pomocí přerušení od časovačů. Zpracování příchozích zpráv ze sběrnice CAN je obsluhováno také přerušením, které se spustí při příchodu nové zprávy. Vývojový diagram hlavní funkce main je na obrázku 21 dole. Pro nastavení periférií se používají HAL (Hardware Abstraction Layer) knihovny od výrobce MCU. Je to rozhraní mezi aplikační vrstvou a nižšími vrstvami.



Obrázek 21: Vývojový diagram hlavní funkce Main

3.2.1 Základní konfigurace při startu programu

Po připojení napájecího napětí nebo resetu mikrokontroléru následuje krok Start, což je inicializace procesoru pomocí startupu. To zahrnuje definování velikosti zásobníku instrukcí (Stack Memory), nastavení programového čítače na Reset Handler (definuje kde má začít běh programu), nastavení tabulky vektorů přerušení (závisí na typu mikrokontroléru) a skok do hlavní programové funkce Main.

Funkce *HAL_Init* resetuje všechny periférie mikrokontroléru, inicializuje flash paměť pro použití ART akcelerátoru, inicializuje systémový čítač s periodou 1 ms (SysTick). Také nastavuje prioritu přerušení a inicializaci jednotlivých portů nebo pinů, které jsou používány perifériemi mikrokontroléru. Následuje nastavení hodinového signálu (*SystemClock_Config*) pro procesor a všechny používané periférie [29]. Zdrojem hodinového signálu pro mikrokontrolér jsou dva externí krystalové rezonátory (viz předchozí podkapitola 2.2.2). Obvod pro nastavení hodinového kmitočtu (Reset and clock control) obsahuje celkem tři fázové závěsy pro nastavení hodin všech periférií, a také povolení hodin pro tyto periférie. Takzvaný strom hodin je na obrázku 12 v odkazu [18]. Pro snadnější nastavení registrů hodinového kmitočtu je vhodné použít software STMCube™, kde je k dispozici přehledné grafické nastavení hodin. Tento software je ke stažení na webových stránkách výrobce mikrokontroléru. Nastavený kmitočet jednotlivých sběrnice a vybraných periférií je uveden v následující tabulce 1.

Název sběrnice nebo periférie	Nastavený kmitočet [MHz]
AHB sběrnice	180
APB1 sběrnice	45
APB2 sběrnice	90
Procesor	180
Řadič LCD-TFT displeje	24
Řadič SDRAM	90
Řadič USB	48

Tabulka 1: Kmitočet periférií a sběrnice nastavený funkcí *SystemClock_Config*

Funkce *GPIO_Init* nastavuje výstupní piny pro tlačítka a LED diody. V této funkci se nenastavují piny pro výstup periférií, ty jsou již inicializovány ve funkci *HAL_Init*. Poté následuje nastavení čítačů (TIM6 a TIM7) o periodě 500 ms respektive 200 ms. Tyto čítače slouží pro vyvolání přerušení při přetečení.

Nastavení čítače pro vyvolání přerušení při přetečení je uvedeno níže. Zapnutí čítačů je provedeno po inicializaci všech periférií. Pro nastavení času a data slouží funkce *RTC_Init*.

```

void TIM6_Init(void)
{
    /* 500ms*/
    TIM_MasterConfigTypeDef sMasterConfig;
    htim6.Instance = TIM6; // Cislo Timeru
    htim6.Init.Prescaler = 9000; // Nastaveni predelicky 90MHz/9000
    htim6.Init.CounterMode = TIM_COUNTERMODE_UP; // Citani nahoru
    htim6.Init.Period = 5000; // Perioda pretečení
    HAL_TIM_Base_Init(&htim6); // Nastaveni registru dle vyse uvedených hodnot
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET; // synchronizace s dalsim TIM nepouzivano
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE; // nepouzivano
    HAL_TIMEx_MasterConfigSynchronization(&htim6, &sMasterConfig); // Nastaveni registru
    TIM6->DIER = TIM_DIER_UIE; // Povoleni preruseni od Timeru6
}

```

Software pro řízení jednotky displeje je velmi rozsáhlý a složitý. Pokud se stane, že se běh programu zastaví nebo zacyklí, je zde funkce *IWDG_Init*. Tato funkce nastavuje nezávislý obvod pro hlídání běhu programu tzv. Independent Watchdog. Tento obvod je zjednodušeně nastavitelný čítač s čítáním směrem dolů s vnitřním zdrojem hodin o frekvenci 32 kHz. Pokud během běhu programu opakovaně nastavujeme čítač Watchdogu, nikdy nenastane situace, že by čítač doběhl do nuly. Tato situace nastane pouze v případě, že program neběží správně. Čítač poté doběhne do nuly a vyvolá reset mikrokontroléru. Perioda přetečení obvodu jsou dvě sekundy. Obvod Watchdogu se aktivuje funkcí *IWDG_Start*.

3.2.2 Inicializace displeje a SDRAM paměti

K zobrazování provozních veličin a nastavení všech dalších hodnot na displeji slouží několik periférií včetně externí SDRAM paměti. Pro správnou funkci displeje je nejdříve potřeba alokovat potřebný paměťový prostor, kde bude uložen aktuální obraz na displeji. Velikost potřebné paměti je přibližně 768 kB (viz předchozí podkapitola 2.2.2). Pro nastavení správného typu paměti slouží funkce *FMC_Init*. Tato funkce nastavuje řadič externích pamětí, který je nastaven na SDRAM paměť. Řadič má k dispozici celkem pět paměťových bank, každá o pevné velikosti 256 MB. Pro SDRAM paměti jsou k dispozici dvě banky, každá pro jednu paměť. Ve funkci *FMC_Init* se používá SDRAM Bank 1. Alokovaná velikost je 64 MB. Adresa paměti je pevně daná a je to adresa 0xC000 0000. Od této adresy ukládáme veškerá data, která chceme zobrazit na displeji. Tato adresa je uvedena v souboru SDRAM.h (*#define SDRAM_BANK_ADDR ((uint32_t)0xC0000000)*) v příloze na CD. Celkový adresový rozsah 64 MB paměti je 0xC000 0000 až 0xC400 000. Na tuto adresu je také nastaven řadič LCD-TFT displeje, který zde čte data zobrazovaná na displeji. Dole je uvedena inicializace FMC řadiče.

```

void FMC_Init(void)
{
    hsdr1.Instance = FMC_SDRAM_DEVICE; // Definice typu paměti
    /* Inicializace SDRAM*/
    hsdr1.Init.SDBank = FMC_SDRAM_BANK1; // Číslo banky
    hsdr1.Init.ColumnBitsNumber = FMC_SDRAM_COLUMN_BITS_NUM_8; //počet bitů v řádku
}

```

```

hsdram1.Init.RowBitsNumber = FMC_SDRAM_ROW_BITS_NUM_12; //počet bitů ve sloupci
hsdram1.Init.MemoryDataWidth = FMC_SDRAM_MEM_BUS_WIDTH_16; //šířka datové sběrnice
hsdram1.Init.InternalBankNumber = FMC_SDRAM_INTERN_BANKS_NUM_4; //počet bank v paměti
hsdram1.Init.CASLatency = FMC_SDRAM_CAS_LATENCY_2; // CAS latence
hsdram1.Init.WriteProtection = FMC_SDRAM_WRITE_PROTECTION_DISABLE; // ochrana přepsání
hsdram1.Init.SDClockPeriod = FMC_SDRAM_CLOCK_PERIOD_2; // Perioda hodin
hsdram1.Init.ReadBurst = FMC_SDRAM_RBURST_DISABLE;
hsdram1.Init.ReadPipeDelay = FMC_SDRAM_RPIPE_DELAY_2; // Zpoždění
/* SDRAM časování podle datového listu výrobce */
SdramTiming.LoadToActiveDelay = 2;
SdramTiming.ExitSelfRefreshDelay = 7;
SdramTiming.SelfRefreshTime = 4;
SdramTiming.RowCycleDelay = 7;
SdramTiming.WriteRecoveryTime = 3;
SdramTiming.RPDelay = 2;
SdramTiming.RCDDelay = 2;
if(HAL_SDRAM_Init(& hsdram1, &SdramTiming) != HAL_OK) // inicializace a kontrola nastavení
{
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // rozsvícení LED diody
    err = 20; // číslo chyby
}

```

Hodnoty jednotlivých proměnných ve struktuře jsou nastaveny podle datového listu výrobce [17]. Po nastavení typu paměti a jejího časování je potřeba nastavit inicializační sekvenci pro samotnou SDRAM paměť, konkrétně nastavení SDRAM Bank 1. Do nastavení této banky patří povolení hodin pro paměť, nastavení přednabití paměťových buněk a periodické obnovování informací v nich. Dále pak nastavení registrů pro čtení a zápis, a také nastavení čítače pro automatickou obnovu informací v paměti. Část inicializace SDRAM paměti je uvedena níže.

```

void SDRAM_InitSequence(void)
{
    __IO uint32_t tmpmrd = 0;
    /*Krok1: nastavení hodin */
    Command.CommandMode= FMC_SDRAM_CMD_CLK_ENABLE;
    Command.CommandTarget= FMC_SDRAM_CMD_TARGET_BANK1;
    Command.AutoRefreshNumber= 1;
    Command.ModeRegisterDefinition = 0;
    if(HAL_SDRAM_SendCommand(&hsdram1, &Command, SDRAM_TIMEOUT) != HAL_OK) // Nahrání
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // Chyba nastavení SDRAM
        err = 21; // Číslo chyby
    }
    HAL_Delay(1); // Vložení prodlevy 1ms
    /* Krok 2: nastavení příkazu přednabití*/

```

```

Command.CommandMode= FMC_SDRAM_CMD_PALL;
Command.CommandTarget= FMC_SDRAM_CMD_TARGET_BANK1;
Command.AutoRefreshNumber = 1;
Command.ModeRegisterDefinition = 0;
if(HAL_SDRAM_SendCommand(&hsdram1, &Command, SDRAM_TIMEOUT) != HAL_OK) //Nahrání
{
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // Chyba nastaveni SDRAM
    err = 22;//Číslo chyby
}
    HAL_Delay(1); // Vložení prodlevy 1ms
/* Step 3: nastavení aut. obnovy pamťových buněk*/
Command.CommandMode= FMC_SDRAM_CMD_AUTOREFRESH_MODE;
Command.CommandTarget= FMC_SDRAM_CMD_TARGET_BANK1;
Command.AutoRefreshNumber= 4;
Command.ModeRegisterDefinition = 0;
if(HAL_SDRAM_SendCommand(&hsdram1, &Command, SDRAM_TIMEOUT)!= HAL_OK)
{
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // Chyba nastaveni SDRAM
    err = 23;//Číslo chyby
}

```

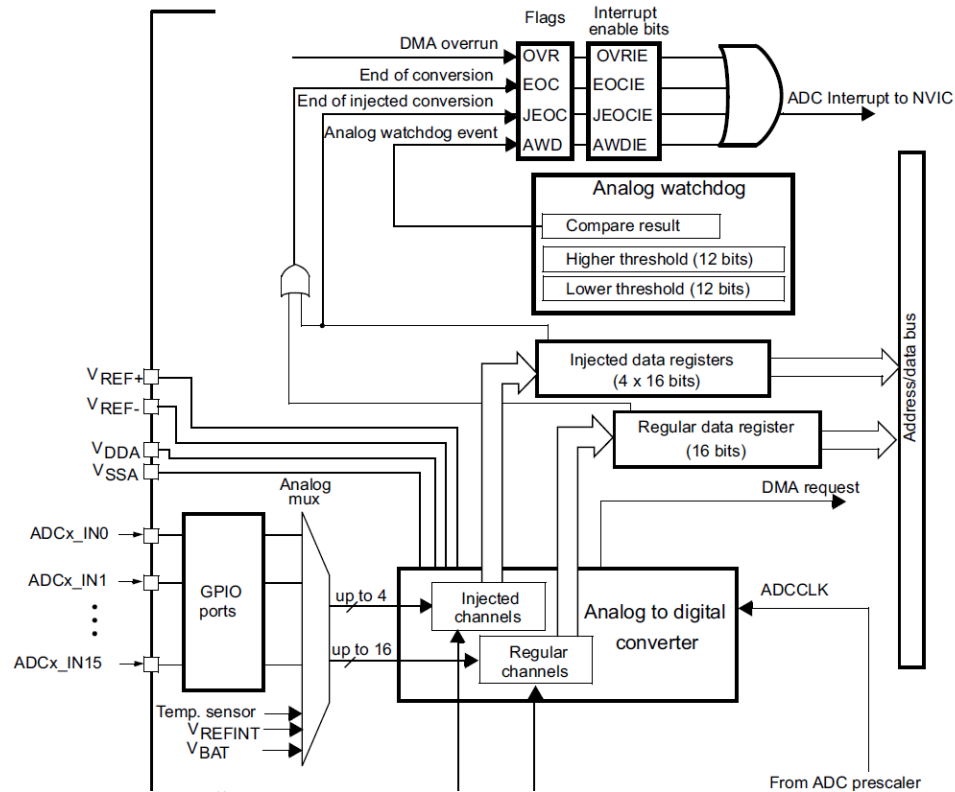
Po inicializaci SDRAM paměti následuje inicializace LCD-TFT displeje (funkce *LTDC_Init*). Nyní již má alokovaný potřebný paměťový prostor pro svoji funkci. Inicializace závisí na typu použitého displeje. Zde je použit sedmi palcový displej o rozlišení 800x480 pixelů. Nejdříve je nastaveno rozlišení, časování a synchronizace displeje. Poté následuje nastavení dvou vrstev, které zahrnuje následující parametry [18]:

- Velikost zobrazovaného okna (menší nebo rovno rozlišení displeje)
- Pozice zobrazovaného okna (pokud je menší než rozlišení displeje)
- Barevné pozadí mimo zobrazované okno (pokud je menší než rozlišení displeje)
- Formát pixelů (nastaven formát RGB565, což je barevná hloubka 16 bitů)
- Průhlednost vrstvy (vrstva číslo 2 je vždy v popředí) na vrstvu pod ní
- Počáteční adresa na zobrazovaná data (což je 0xC000 0000 na SDRAM paměti)

Po inicializaci LDC-TFT displeje se zobrazí úvodní obrazovka, kde je napsáno, že jednotka displeje se inicializuje. Po zobrazení úvodní obrazovky totiž následují inicializace dalších systému, což ještě nějaký čas trvá (přibližně 1 až 2 sekundy). Proto je vhodné uživatele informovat o tom, že jednotka již pracuje, ale ještě nejsou dokončena všechna potřebná nastavení jednotky při startu.

3.2.3 Inicializace A/D převodníku

Po zobrazení úvodní obrazovky následuje inicializace A/D převodníku. Blokový diagram A/D převodníku je v odkazu [18] na obrázku 69. Výřez z tohoto blokového diagramu je na obrázku 22 dole.



Obrázek 22: Část blokového schématu A/D převodníku. Převzato z [18]

Nejdříve se nastaví před dělička hodin ze sběrnice APB2, na kterou je A/D převodník připojen. Z této frekvence se nastaví perioda hodin, která je vypočtena v rovnici 13.

$$t_{ADCCLK} = 1 / (APB2_{CLK} / 8) = 1 / (90\text{MHz} / 8) = 1 / 11,25\text{MHz} = 88,8\text{ns} \quad (13)$$

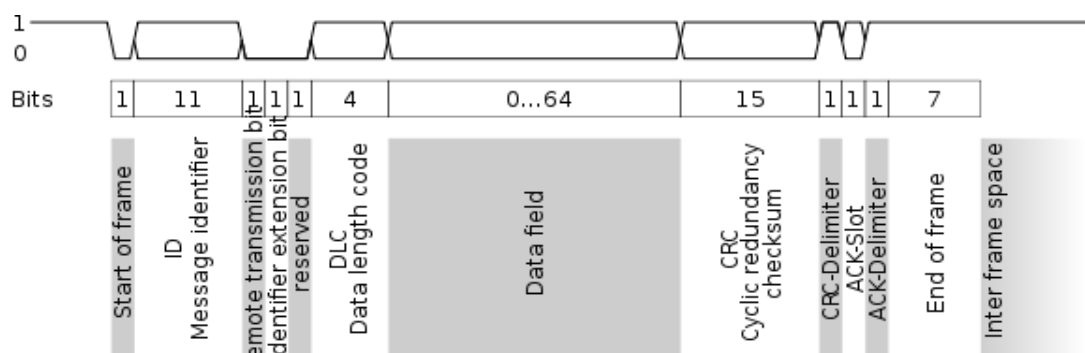
$$t_{vzork} = t_{(ADCCLK)} \cdot 480 = 42,7\ \mu\text{s} \quad (14)$$

Doba vzorkování jednoho kanálu (jednotka má celkem 4 kanály) je nastavitelná od 3 hodinových pulzů t_{vzork} až do 480 hodinových pulzů. V našem případě je vzorkovací doba nastavena na 480 pulzů. Vzorkování tedy podle rovnice 14 trvá 42,7 μs . Dále se nastavuje zarovnání dat v registru, která jsou v tomto případě zarovnána vpravo. To znamená, že na prvním bitu v datovém registru A/D převodníku je nejnižší bit převedené hodnoty. Rozlišení A/D převodníku je 8 bitů (256 hodnot). Při referenčním napětí 3,3 V to znamená rozlišení 13 mV na jeden bit. Na pracovním rozsahu A/D převodníku (0,6V až 3V) tedy rozlišujeme 185 hodnot sešlápnutí plynového pedálu.

Typ používaných kanálů A/D převodníku je ze skupiny Regular channels, kde jsou pro převod hodnot z pedálů používány kanály ADC1_IN0 až ADC1_IN3. Při dokončení převodu každého z kanálů je vyvolán požadavek na přenos hodnoty pomocí DMA kanálu. Inicializace DMA kanálu je vytvořena stejnojmennou funkcí (*DMA_Init*). Po přenosu dat pomocí DMA je smazán příznakový bit o dokončení převodu v registru A/D převodníku a převodník pokračuje s převodem na následujícím kanálu. Po převedení všech čtyř kanálů je převod ukončen. Jeho pravidelné spouštění se děje v rámci obsluhy přerušení od čítače 7 (každých 200 ms). Pro hlídání velikosti vstupního napětí z pedálů, které musí být v rozmezí 0,6 V až 3 V, slouží jako redundance analogový watchdog. Pokud A/D převodník selže a nezjistí poruchu na pedálech (hodnoty mimo rozsah), analogový watchdog může generovat přerušení nebo flag jako informaci o nesprávné funkci pedálů. Hodnoty pro reakci watchdogu jsou nastaveny na velikost napětí 0,559 V respektive 3,055 V.

3.2.4 Inicializace CAN sběrnice

CAN rozhraní je určeno pro komunikaci s řídicí jednotkou motoru. Hlavní úlohou je příjem dat o rychlosti, otáčkách a stavu nabití baterie. Další důležitou funkcí CAN rozhraní je vysílání polohy sešlápnutí brzdového a plynového pedálu. Posledním úkolem je pravidelné vysílání zpráv o funkčnosti jednotky displeje (tzv. HeartBeat zpráva), chybové zprávy jednotky, případně příjem chybových zpráv od řídicí jednotky motoru. Obecný rámec CAN zprávy ilustruje obrázek 23. Za bitem označujícím začátek rámce zprávy se nachází identifikátor zprávy, tedy označení od které jednotky zpráva pochází. Následuje typ zprávy-zda má zpráva rozšířený identifikátor-počet bajtů dat a samotná data. Na konci vysílaných dat je pak kontrolní součet, potvrzení přijetí a konec rámce.



Obrázek 23: Rámec CAN zprávy Převzato z: https://de.wikipedia.org/wiki/Controller_Area_Network

Jelikož se zde vyskytuje několik typů zpráv, je potřeba jejich vzájemné rozlišení, aby jednotka displeje nebo motorová jednotka hned rozlišila o jaký typ zprávy se jedná. Proto je zde využita implementace aplikační vrstvy MicroCANopen. Jedná se o zjednodušenou verzi protokolového standardu CANopen. Z této aplikační vrstvy jsou převzata označení,

identifikátory a významy jednotlivých vysílaných nebo přijímaných zpráv [30]. Při pravidelném vysílání HeartBeat zprávy se k identifikátoru jednotky (ID) přidává číslo v hexadecimálním tvaru 0x700. O pravidelné vysílání této zprávy každých 500 ms se stará obsluha přerušení od čítače 6. Přenášená data mají velikost jeden bajt a hodnota dat je 0x05. To znamená, že jednotka pracuje (mód Operational). Se stejnou adresou se vysílá tzv. Boot-Up zpráva, která je vysílána pouze jednou a to při zapnutí jednotky displeje. Liší se pouze hodnota dat, která jsou 0x00 (mód Boot-Up).

Zpráva o poloze brzdového a plynového pedálu je označena zkratkou TPDO (Transmit Process Data Message), která je určena pro vysílání dat. Celkový rozsah datových zpráv je TPDO1 až TPDO4 a k ID jednotky se přičítá číslo 0x180 až 0x480. Záleží na čísle zprávy TPDO. Pro vysílání polohy pedálů je určena zpráva TPDO1 (0x180+ID) a její velikost jsou tři bajty. Třetí bajt je pro softwarový čítač, který inkrementuje odchozí zprávy a je pojistkou, že software funguje a hodnoty pedálů jsou relevantní. Zpráva se vysílá každých 200 ms během obsluhy přerušení časovače 7. Pokud by bylo zapotřebí vysílat jednotce motoru další data jako například zrychlení nebo teplotu, můžeme použít zbylé TPDO zprávy.

Chybové zprávy jednotky (EMCY) jsou vysílány s přičtením hexadecimálního čísla 0x80 k identifikátoru jednotky. Rozsah vysílaných dat jsou čtyři bajty přičemž první tři bajty jsou nulové. Číslo chyby obsahuje čtvrtý bajt, zbylé bajty (5 bajt až 8 bajt) se nepoužívají [31]. Pro lepší představu slouží tabulka 2.

CAN identifikátor	Označení zprávy	Počet bajtů	Druh vysílaných dat
0x80 + ID jednotky	EMCY	4	Kód chyby
0x180 + ID jednotky	TPDO1	2	Poloha pedálů
0x280,0x380,0x480+ ID	TPDO2-TPDO4	Rezerva	Rezerva
0x700 + ID jednotky	BootUp, HeartBeat	1	0x00 nebo 0x05

Tabulka 2: Typy vysílaných CAN zpráv

Inicializace řadiče CAN v mikrokontroléru a nastavení jednotlivých druhů zpráv je v souboru CAN.c na přiloženém CD. Celkem jsou k dispozici dva CAN řadiče. Zde se používá řadič CAN1, který řídí i vyrovnávací paměť pro vysílání a příjem zpráv. Dále je potřeba nastavit přenosovou rychlost na 500 kbit/s, časové zpoždění na sběrnici (synchronizační segment, kdy se nepředpokládá změna bitu) a synchronizaci pomocí bitových segmentů 1 a 2 (viz kapitola 36.7.7 v [18]).

Dalším krokem je nastavení vysílaných zpráv. Níže je uveden typ HeartBeat zprávy. Nejdříve se zjistí, zda je řadič CAN sběrnice připraven pro přenos zprávy. Dále pak nastavíme identifikátor zprávy, typ zprávy, typ identifikátoru, délku dat a samotná data. Na konci této

funkce se provede kontrola, zda byla data odeslána v pořádku.

```
void HeartBeatSend(void)
{
    if(HAL_CAN_GetState(&CanHandle) != HAL_CAN_STATE_READY) // Kontrola připravenosti přenosu CAN zprávy
    {
        err = 53; // Číslo chyby
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // Rozsvícení LED
    }
    CanHandle.pTxMsg->StdId = (0x700 + Node_ID2); // Identifikátor zprávy
    CanHandle.pTxMsg->RTR = CAN_RTR_DATA; // Typ zprávy: vysíláme data
    CanHandle.pTxMsg->IDE = CAN_ID_STD; // Standardní ID
    CanHandle.pTxMsg->DLC = 1; // Délka dat 1 byte
    CanHandle.pTxMsg->Data[0] = 0x05; // Data
    if(HAL_CAN_Transmit(&CanHandle, 10) != HAL_OK) // Kontrola správného vyslání zprávy
    {
        err = 52; // Číslo chyby
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // Rozsvícení LED
    }
}
```

Posledním krokem je nastavení filtrů přijímaných zpráv. I když bude jednotka displeje na CAN sběrnici pouze s motorovou jednotkou, je potřeba, aby přijímala zprávy, které jsou pro ni určené. Je to z toho důvodu, kdyby se později na sběrnici přidala další jednotka. Přijímací filtry na použitém mikrokontroléru mohou pracovat ve dvou režimech. Prvním je režimem je maskování. To znamená, že určíme, které bity v identifikátoru musí souhlasit a na kterých nezáleží. Druhým režimem je nastavení filtrů na pevný identifikátor, takže jsou přijaty zprávy, kde identifikátor přesně souhlasí. Celkem je k dispozici 28 filtrů zpráv. Příklad konfigurace filtru pro přijímání zprávy TPDO1 v souboru CAN.c je uveden níže.

```
sFilterConfig.FilterNumber = 1; //Číslo filtru
sFilterConfig.FilterMode = CAN_FILTERMODE_IDLIST; // Typ filtru nastaven na pevný identifikátor
sFilterConfig.FilterScale = CAN_FILTERSCALE_16BIT; // Rozsah filtru 16 bitů
sFilterConfig.FilterIdHigh = (0x180 + Node_ID2) << 5; // Identifikátor
sFilterConfig.FilterIdLow = 0x0000;
sFilterConfig.FilterFIFOAssignment = 1; // Výběr přijímací FIFO paměti
sFilterConfig.FilterActivation = ENABLE; //Povolení filtru
sFilterConfig.BankNumber = 13; //Číslo banky
if(HAL_CAN_ConfigFilter(&CanHandle, &sFilterConfig) != HAL_OK) // Kontrola nastavení filtru
{
    err = 51; //Číslo chyby
    HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET); // LED on
}
```

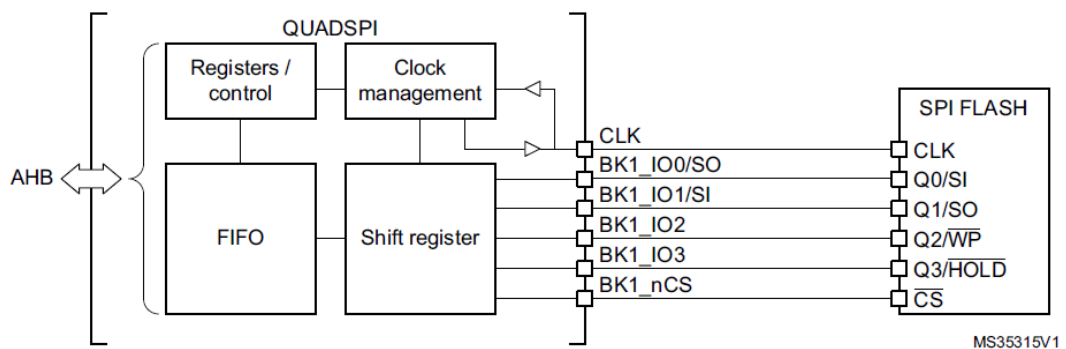
Po nastavení řadiče CAN vyšle jednotka displeje zprávu HB_BootUp. Tato zpráva informuje motorovou jednotku o svém zapnutí a inicializaci. Po inicializaci ostatních periférií již následuje pravidelné vysílání HeartBeat zpráv po 500 ms.

3.2.5 Inicializace UART, I2C, QSPI a USB

Rozhraní UART slouží ke komunikaci s bezdrátovým modulem Xbee-Pro® a GPS modulem. Inicializuje se funkcí *UART_Init*. GPS modul ve zvolené hardwarové konfiguraci uvedené v podkapitole 2.2.5 komunikuje přenosovou rychlostí 4800 baudů za sekundu a délka dat v rámci je osm bitů. Typ přijímané zprávy je ve formátu NMEA [23]. Bezdrátový komunikační modul Xbee-Pro® má stejnou délku datového rámce, pouze přenosová rychlost je 9600 baudů za sekundu.

Po I2C sběrnici (funkce *I2C_Init*) komunikuje mikrokontrolér s MEMS akcelerometrem a teplotním čidlem. Přenosová rychlost je 100 kbit/s (Standart mode) a velikost adresy je 7 bitů. Adresa akcelerometru na sběrnici je 00110000 a adresa teplotního čidla je 1001101. Sběrnice funguje na principu řízení Master-Slave, kde řídicím zařízením je mikrokontrolér. Nejdříve je vyslána mikrokontrolérem sedmibitová adresa pro danou periférii, poté následuje osmý bit pro čtení nebo zápis do registru ve Slavu. Podle osmého bitu v adrese, který nastavil čtení/zápis se vyšlou/zapíše data z/do registru Slavu.

Funkce *QSPI_Init* slouží k inicializaci externí Flash paměti o velikosti 128 Mb. Jak již bylo zmíněno v předchozí kapitole, paměť slouží pro ukládání provozních hodnot elektromotokáry. Řadič je připojen k rychlé AHB sběrnici, která propojuje procesor a všechna ostatní paměťová rozhraní (vnitřní SRAM, externí SDRAM, vnitřní Flash). Pro nastavení kmitočtu slouží předdělička, nastavená na dělení dvěma. To znamená, že QSPI sběrnice pracuje na frekvenci 90 Mhz. Dále se nastaví počet adresových bitů pro adresaci celé paměti, což je 22 bitů. Paměť je rozdělena na 16 777 216 bajtů, tyto bajty jsou rozděleny do 256 sektorů (každý sektor má 64 kB) a 4096 subsektorů (4 kB každý subsektor). Smazání paměti je možné ve třech režimech. To je smazání celé paměti (Full-chip erase), smazání 64 kB sektorů a smazání subsektorů (4 kB). Posledním krokem je nastavení prodlevy hodinového signálu mezi dvěma příkazy, polarity hodin a nastavení identifikátoru paměti. Blokové schéma QSPI je na obrázku 24.



Obrázek 24: Blokové schéma QSPI. Převzato z [18]

Poslední periférií je nastavení USB rozhraní. Toto rozhraní je zde jako doplňkové pro pozdější využití. Funkce pro inicializaci je prozatím prázdná. Zde je možnost nastavení jednotky, aby se zobrazovala jako paměťové zařízení (při připojení do PC) nebo by bylo možné po připojení flash disku nahrát data do něj. Může se například použít pro stažení provozních dat nebo konfiguraci jednotky displeje.

3.2.6 Nekonečná smyčka programu a obsluha přerušení

Po nastavení všech periférií jednotky displeje se na displeji zobrazí hlavní obrazovka s hlavní nabídkou (*Display Menu*). V tomto menu si můžeme navolit pomocí tlačítek zobrazované veličiny nebo nastavení jednotky. Můžeme si také zobrazit chybová hlášení. V tomto okamžiku již běží nekonečná smyčka programu a obsluha jednotlivých přerušení. V nekonečné smyčce se kontroluje zdali bylo stisknuto nějaké tlačítko. Pokud ano, zobrazí se zvolená obrazovka. V hlavní nabídce menu jsou tři možnosti. Jsou na to použita tlačítka Button 1, Button 3 a Button 5. První tlačítko je zobrazení telemetrie. To zahrnuje zobrazení zrychlení, teplotu, hodnoty A/D převodníku a komunikace po sběrnici CAN. Druhé tlačítko je na zobrazení aktuální rychlosti, stavu nabití baterie a dalších veličin od řídicí jednotky motoru. Poslední třetí tlačítko je na zobrazení aktuálního nastavení jednotky displeje. Je zde zobrazen kmitočet mikrokontroléru, přenosová rychlost CAN sběrnice a verze softwaru. Dále je možnost zobrazit nastavení i dalších periférií. Menu obrazovky lze v případě potřeby jednoduše rozšířit o další dvě položky menu. Jejich volby se přidávají k nepoužívaným tlačítkům (Button 2 a 4) v hlavním menu. V nekonečné smyčce se také kontroluje, zda nenastal nějaký chybový stav. Pokud ano, číslo chyby se zobrazí na displeji a také se odešle zpráva motorové jednotce. Chybové stavy jsou popsány v následující podkapitole.

Při běhu nekonečné smyčky programu mikrokontrolér reaguje na požadavky přerušení od periférií a procesoru. Přerušení je pravidelně vyvoláváno dvěma čítači (Timer6, Timer7) a nepravidelně při příchodu nové zprávy po sběrnici CAN. Výpis používaných přerušení je uveden v následující tabulce 3.

Druh přerušení	Periodicita	Vykonávané příkazy
Nemaskovatelné přerušení (NMI)	NE	Rozsvícení LED (Blink LED)
Systémový čítač (SysTick)	1 ms	Inkrement systémového čítače
Příjem CAN zprávy	NE	Přečtení CAN zprávy a zobraz. na displeji
Přetečení čítače 6 (Timer6)	500 ms	Vyslání HeartBeat zprávy
Přetečení čítače 7 (Timer7)	200 ms	Vzorkování pedálů a vyslání TPDO1 zprávy

Tabulka 3: Používaná přerušení v řídicím programu jednotky displeje

Nemaskovatelné přerušení je spuštěno, pokud procesor mikrokontroléru zjistí chybu při

běhu programu. Takováto chyba je velmi vážná a není možné je vypsát na displej, protože funkce sloužící k zápisu na displej nemusí pracovat. Proto je rozsvícena pouze LED dioda. Taková chyba by ovšem neměla nikdy nastat a vyskytuje se většinou u softwaru s chybou na kterou nepřišel překladač vývojového prostředí. Dalším přerušením je systémový čítač, který běží s inkrementem 1 ms. To nám umožňuje vytvořit na jeho základě jednoduchou funkci, která nám realizuje například přesné zpoždění při běhu programu.

Příchod nové CAN zprávy můžeme kontrolovat v hlavní smyčce programu. To je ale nevýhodné, protože nám to zabírá výpočetní výkon. Proto je vhodné použít přerušení, které se spouští při příchodu nové CAN zprávy. Po jeho spuštění je nová zpráva přečtena z přijímacího registru CAN řadiče. Poté následuje její zobrazení na displej popřípadě uložení do paměti.

Obsluha přerušení při přetečení čítače 6 má na starosti každých 500 ms vyslat zprávu HeartBeat. Podobnou úlohu má přerušení od čítače 7. Spouští každých 200 ms vzorkování pedálů pomocí A/D převodníku a vysílání jejich hodnoty (velikost sešlápnutí) po CAN sběrnici motorové jednotce.

Pro všechna používaná přerušení jsou definovány jejich priority. To znamená, že pokud se provádí obsluha přerušení a přijde přerušení s vyšší prioritou, vykoná se to s vyšší prioritou. Po dokončení jeho obsluhy se program vrátí k předchozímu přerušení s nižší prioritou a poté do hlavní programové smyčky. Priorita přerušení je v tabulce 4. Vyšší priorita znamená menší číslo. Z tabulky je patrné, že nejvyšší prioritu mají přerušení od procesoru mikrokontroléru. Následují přerušení od čítačů, které mají druhou nejvyšší prioritu. Je to z důvodu, že zprávy musí být vysílány v pravidelných intervalech. Poslední je čtení příchozích zpráv po sběrnici CAN. Ty mají nejnižší prioritu a nevádí, když se data zpracují až po obsluze předchozích přerušení. Například údaj o rychlosti můžeme zobrazit o několik okamžiků později a řidič nezaregistruje prodlevu. Pokud by byla priorita čítačů a příjmu zpráv otočená, nejdříve bychom obsluhovali přerušení od příchozích CAN zpráv a zobrazování dat na displeji. Tyto příkazy zaberou více času než vzorkování a odesílání polohy pedálů. Zde by již mohly nastat problémy s prodlevou na sešlápnutí pedálů.

Druh přerušení	Priorita
Nemaskovatelné přerušení (NMI)	0
Systémový čítač (SysTick)	0
Příjem CAN zprávy	2
Přetečení čítače 6 (Timer6)	1
Přetečení čítače 7 (Timer7)	1

Tabulka 4: Priorita přerušení v řídicím programu

3.3 Ošetření chybových stavů

Jednotka displeje má zabudovanou v řídicím softwaru vlastní diagnostiku. Při inicializaci se kontroluje, zda byla daná periférie nastavena správně. Pokud ne, zaznamená se číslo chyby. Dále se kontroluje správné odesílání zpráv na CAN sběrnici a spouštění A/D převodníku. Pokud nastane chyba, v hlavní programové smyčce je splněna podmínka, která kontroluje výskyt chyby. Číslo chyby je poté zobrazeno na displeji a také je odesláno motorové jednotce. Dále může být chyba uložena do paměti pro pozdější analýzu problému. Čísla jednotlivých chyb a jejich význam je uveden v tabulce 5 dole.

Druh chyby	Číslo chyby
SDRAM chyba inicializace	20
SDRAM chyba nastavení hodin	21
SDRAM chyba před nabití paměť. buněk	22
SDRAM chyba aut. obnovování paměť. buněk	23
SDRAM chyba nastavení řídicích registrů	24
A/D chyba inicializace	40
A/D chyba nastavení kanálu 0	41
A/D chyba nastavení kanálu 1	42
A/D chyba nastavení kanálu 2	43
A/D chyba nastavení kanálu 3	44
A/D chyba při spuštění převodu	45
A/D chyba analogového watchdogu	46
CAN chyba inicializace	50
CAN chyba nastavení filtru zpráv	51
CAN chyba při odesílání zprávy	52
CAN řadič není připraven na vyslání zprávy	53
LCD řadič chyba inicializace	10
LCD chyba nastavení vrstvy displeje	11

Tabulka 5: Čísla chybových stavů v řídicím softwaru

4 Testování a ověření funkčnosti

Tato kapitola se zabývá ověřením správné funkčnosti navrženého hardwaru a softwaru. Je zde shrnuto, zda výsledný hardware a software splňuje zadanou specifikaci. Případně jsou zde popsány návrhy na zlepšení funkčnosti.

Po vyrobení desky plošných spojů přišlo jako první na řadu osazení zdrojové části. Poté byla vyzkoušena její funkčnost. Oba dva stabilizátory napětí (3,3 V a 5 V) fungovaly bez problémů. Stejně tomu bylo i u zdroje konstantního proudu pro podsvětlení displeje. Následovalo osazení zbylých součástek (číslicové a analogové obvody). Jejich oživení se neobešlo bez drobných problémů. Prvním problémem byl resetovací obvod MAX803SQ293, který držel mikrokontrolér v neustálém resetu i při správné úrovni napájecího napětí (3,3 V). Bylo to způsobeno špatným typem obvodu, který byl na vyšší resetovací napětí (požadovaná úroveň resetovacího napětí je 2,93 V). Obvod byl prozatím odstraněn a mikrokontrolér již nebyl v neustálém resetu. Mikrokontrolér má již v sobě zabudovaný obvod pro kontrolu napájecího napětí, takže jsme o tuto funkci nepřišli. Dalším drobným problémem byla záměna vstupních a výstupních pinů na Schmittově klopném obvodu, který je připojen k tlačítkům. V tomto případě pomohlo otočení obvodu a tlačítka již fungovala. V novém návrhu je tato chyba opravena. Muselo se také opravit napájení I/U převodníků. Byly totiž připojeny na napájení 3.3 V, ale pracují až od napětí 4,5V. Oprava byla provedena jednoduchou úpravou - přerušením původní napájecí cesty 3,3 V a pomocí drátové propojky připojení na 5 V. Po změně napájení již fungovaly podle předpokladů. Poslední změnou hardwaru byla úprava držáku zálohovací baterie, který měl otočenou polaritu (špatné pouzdro držáku baterie) a přepájení pouzdra akcelerometru z důvodu nálezu tzv. studeného spoje.

Jedním z návrhů na zlepšení je zapracování pinu na reset mikrokontroléru k programovacímu rozhraní. V původním návrhu je pouze JTAG reset, nikoliv klasický reset. Je to z důvodu ladění softwaru, kdy je mikrokontrolér držen v resetu programátorem. Jako provizorní řešení byl připájen vodič k resetovacímu tlačítku. Samotný LDC-TFT displej, byl k desce plošných spojů přišroubován pomocí distančních sloupků. Spojení DPS a displeje je na fotografii v příloze 4. Hardware byl kompletně osazen, oživen a připraven na nahrání a ladění řídicího softwaru.

U řídicího softwaru byly nejprve namapovány jednotlivé piny na vnitřní periférie mikrokontroléru. Dalším krokem bylo zprovoznění stavových kontrol (LED diody) a tlačítek. Následně byl oživen řadič LCD-TFT displeje. K němu je vázáno naprogramování externí SDRAM paměti. Po menších problémech byl odzkoušen zápis do externí paměti a

poté nahrání a zobrazení dat z této paměti na displej. Programování zobrazovaných dat je řešeno pomocí knihoven pro ovládání displeje. Uživatel programuje zobrazované veličiny tak, že si jednoduše nastaví velikost písma, barvu pozadí, barvu písma a zobrazovaný text. Je to díky vestavěným funkcím na vykreslování jednotlivých znaků a barev pozadí. Velikosti písma jsou řešeny pomocí fontů v souboru fonts.c, kde si můžeme případně doplnit další potřebné velikosti písma. Další doplňkovou funkcí, která se zatím nepoužívá, je vykreslování obrazců jako například čáry nebo kružnice.

CAN sběrnice pro komunikaci s motorovou jednotkou byla nejprve odzkoušena v módu pro ladění CAN sběrnice (tzv. Loop-Back mode). V tomto módu se zprávy vysílají a zároveň probíhá jejich příjem. Nečeká se na potvrzení zprávy a můžeme tím zjistit správnost odesílaných zpráv bez nutnosti další jednotky na sběrnici. Po odzkoušení na rychlostech 125 kbit/s a 500kbit/s se připojilo USB/CAN rozhraní do PC pro příjem a záznam zpráv. CAN řadič se přepnul do standardního pracovního módu (Normal mode). V této konfiguraci bylo odzkoušeno pravidelné odesílání HeartBeat zprávy a TPDO1 zprávy. Během tohoto měření se zachytilo celkem 307 zpráv včetně úvodní BootUp zprávy. Průměrné doby mezi odesílanými zprávami jsou uvedeny v tabulce 6. Dále bylo otestováno zda data v jednotlivých bajtech zprávy odpovídají vysílané zprávě. To znamená, že v HeartBeat zprávě je hodnota prvního bajtu 0x05. Ve zprávě TPDO1 jsou první dva bajty údaj o sešlápnutí pedálů. Třetí bajt je hodnota softwarového čítače, který se inkrementuje s každou odeslanou zprávou. Tento čítač fungoval také podle předpokladů. Filtr příchozích zpráv byl otestován v Loop-Back módu, kdy se přijímaly vysílané zprávy. Po změně adresy pro přijímané zprávy (jiná než jednotky displeje) již žádné zprávy nebyly přijímány. V normálním módu je filtr zpráv nastaven na příjem zpráv od motorové jednotky.

Typ zprávy	Počet přijatých zpráv	Nastavená četnost odesílání zpráv	Průměrná četnost přijímaných zpráv
HeartBeat	89	500 ms	500,1 ms
TPDO1	218	200 ms	200,1 ms

Tabulka 6: Počet a průměrná doba mezi odesílanými CAN zprávami

Funkčnost A/D převodníku byla ověřena simulací pedálu pomocí potenciometru. Na vstupu jednoho z pedálů se měnila velikost procházejícího proudu (4 mA až 20 mA). Tento proud převáděl I/U převodník na ekvivalentní hodnotu napětí (0,6 V až 3 V). Toto napětí snímal A/D převodník v mikrokontroléru a velikost tohoto napětí byla zobrazována na displeji.

Bezdrátová komunikace pomocí Xbee-Pro® modulu byla otestována vysláním zprávy po UART rozhraní. Je zde nastavena základní přenosová rychlost 9600 baudů. V této konfiguraci by měla přijímací stanice přijímat zprávy bez dalšího nastavování. Bohužel nebyla k dispozici žádná druhá přijímací stanice pro ověření přijaté zprávy. Do budoucna je nutné vytvořit jednoduchý přijímací modul. Ten je možné vytvořit s použitím jednoduchého obvodu FTDI.

GPS modul je k mikrokontroléru připojen také pomocí UART. Při testování byl ověřen přenos dat a GPS modul by měl tedy pracovat podle předpokladů, kdy vysílá NMEA zprávy o poloze.

Teplotní senzor komunikující po I2C sběrnici byl odzkoušen, zda odpovídá na požadavek čtení teploty od mikrokontroléru. Tato komunikace probíhala spolehlivě a pomocí jednoduchých zpráv byly přečteny vybrané registry obvodu.

Větší problémy nastaly při komunikaci s MEMS akcelerometrem. V testu byla s obvodem navázána komunikace a senzor odpovídal na vysílanou adresu. Základ komunikace je v programu připraven, ale je nutné dodělat inicializační a další funkce pracující se senzorem.

Flash paměť na sběrnici Quad-SPI je potřeba nejdříve naformátovat a naprogramovat propojení počítače a mikrokontroléru pro čtení nebo zápis dat do paměti. Práce s tímto typem paměti není triviální. Z časových důvodů tak nebyla otestována její funkčnost. Při jejím ožívování by se neměly vyskytovat žádné problémy, jelikož zapojení vychází z ověřeného modulu od výrobce mikrokontroléru. Tento modul slouží k seznámení se s vlastnosti a funkcemi mikrokontroléru z produktové řady STM32F7. Konkrétně se jedná o modul s názvem STM32F746G-Disco. Je zde použit stejný typ paměti, a proto je při jejím programování výhodné vycházet ze zdrojového kódu výrobce modulu.

Závěr

V této diplomové práci bylo hlavním cílem vytvořit modul elektronické jednotky, která bude sloužit jako přístrojový panel. Tato jednotka obsahuje displej pro zobrazování provozních dat a ovládací tlačítka. Dalšími funkcemi jsou měření provozních veličin, snímání poloh ovládacích pedálů, připojení ke sběrnici CAN, bezdrátový přenos dat a ukládání provozních dat. Posledním krokem bylo oživení modulu jednotky displeje a ověření funkčnosti.

Hardwarový návrh jednotky displeje popisují blokové napájecí schéma a blokové datové schéma. V blokovém napájecím schématu je popsána ochrana navržených napájecích obvodů proti EMI a ESD a rozvod napájení pro jednotlivé subsystémů jednotky displeje. Blokové datové schéma popisuje připojení periférií k mikrokontroléru. Podrobný popis hardwaru je rozdělený na zdrojovou část, mikrokontrolér a displej, I/U převodníky, komunikace po CAN sběrnici, bezdrátové vysílání dat a ostatní periférie. Vše je rozmístěno na čtyřvrstvé desce plošných spojů o velikosti 166x117,8 mm. Rozmístění součástek na desce je vyřešeno tak, aby respektovalo rozdělení do jednotlivých bloků. Především to je rozdělení na zdrojovou část, analogovou část a digitální část.

Modul jednotky displeje řídí 32-bitový mikrokontrolér STM32F746IGT od společnosti ST Microelectronics. Jedná se o nejnovější a nejvýkonnější platformu z řady mikrokontrolérů určených pro embedded aplikace. Jádrem mikrokontroléru je procesor ARM® Cortex®-M7. Takto výkonný mikrokontrolér je zde z důvodu použití velkého sedmipalcového displeje o rozlišení 800x480 pixelů. Dalším důvodem je obsluha velkého množství vnějších periférií po různých typech sběrnic, které jsou k mikrokontroléru připojeny. Je to relativně nový mikrokontrolér, na který prozatím existuje velmi málo aplikačních příkladů, ve kterých by se mohl uživatel podrobně seznámit se všemi jeho funkcemi. Výhodou jeho použití je moderní platforma, která má dostatečnou rezervou pro případnou rozšiřitelnost funkcí jednotky displeje.

Po oživení a nahrání řídicího softwaru jednotka displeje splňuje požadavky, které na ni jsou kladeny. Po zapnutí následuje inicializace jednotky a zobrazení úvodní obrazovky, poté je načteno menu pro výběr zobrazované veličiny. Všechna zobrazovaná data jsou uložena na externí SDRAM paměti. Jednotka displeje, již od úvodní obrazovky komunikuje po sběrnici CAN s motorovou jednotkou, kam odesílá zprávy o poloze sešlápnutí pedálů a zprávy o tom, že jednotka je zapnuta (HeartBeat). Přesné odesílání zpráv je dosaženo pomocí přerušení od čítačů. Jednotka také komunikuje s ostatními perifériemi během hlavní

programové smyčky. K této komunikaci patří například bezdrátová komunikace nebo čtení aktuální teploty prostředí.

Jednotka navíc obsahuje vlastní diagnostická hlášení při výskytu chyby inicializace vnitřní periférie a při chybě přenosu po CAN sběrnici. Čísla chyb jsou zobrazována na displeji a zároveň odesílána po sběrnici CAN.

Zadáním diplomové práce bylo vybrat pro jednotku displeje vhodný mikrokontrolér a displej. Implementovat měření provozních veličin, ukládání provozních dat, připojení ke sběrnici CAN a možnost bezdrátové komunikace. Posledním bodem zadání bylo oživit modul jednotky displeje. Všechny body zadání diplomové práce byly splněny. Nad rámec zadání je zde navíc snímání polohy brzdového a plynového pedálu.

Seznam symbolů a zkratek

ART	Adaptive real-time	Akcelerátor pro čtení Flash paměti
CAN	Controller Area Network	Automobilová a prům. sběrnice
DMA	Direct Memory Access	Přímý paměťový přístup
DMIPS	Dhrystone Million instructions per sec.	Jednotka výpočetního výkonu proc.
ESD	Electrostatic Discharge	Elektrostatický výboj
EMI	Electromagnetic interference	Elektromagnetická interference
FIA	Fédération Internationale de l'Automobile	Mezinárodní automobilová federace
FPU	Floating-point unit	Výp. jednotka pro plovoucí des. čárku
FMC	Flexible Memory Controller	Flexibilní paměťový řadič
GPS	Global Positioning System	Globální navigační systém
HDMI	High-Definition Multimedia Interface	Multimediální rozhraní
I2C	Inter-Integrated Circuit	Sériová sběrnice
JTAG	Joint Test Action Group	Standardní rozhraní pro testování
LCD-TFT	Thin Film Transistor-Liquid Crystal Display	Typ LCD displeje s tenkovrstvými tranzistory
LDO	Low Dropout	Stabilizátory s nízkým úbytkem
MEMS	Micro Electro Mechanical System	Mikro elektro-mechanický systém
MCU	Microcontroller	Mikrokontrolér
NMEA	National Marine Electronics Assoc.	Formát GPS zpráv
RISC	Reduced Instruction Set Computing	Redukovaná instrukční sada
SPI	Serial Peripheral Interface	Typ sériové sběrnice
SRAM	Static Random Access Memory	Statická RAM paměť
SDRAM	Synchronous Dynamic Random Access Memory	Dynamická RAM paměť
SWD	Serial Wire Debug	Rozhraní od ARMu podobné JTAG
TPDO	Transmit Process Data Message	Vysílací procesní objekt
USB	Universal Serial Bus	Univerzální sériová sběrnice
UART	Universal Asynchronous Receiver and Transmitter	Univerzální asynchronní sériové rozhraní

Použitá literatura a zdroje

- [1] REIF, Konrad a Karl-Heinz DIETSCHE. *Automotive handbook: Bosch - invented for life. 8th ed., revised and extended.* Chichester: John Wiley, 2011. ISBN 978-1-119-97556-4.
- [2] ČSN 30 5100: *Kontrolní a měřicí přístroje automobilů. Všeobecné technické požadavky. Metody zkoušení. 1/1983.* Praha: ČMI, 1983.
- [3] RADA EVROPSKÉ UNIE, *Evropský parlament. SMĚRNICE EVROPSKÉHO PARLAMENTU A RADY 2007/46/ES.* In: . Brusel, 2007, ročník 2007, číslo 263.
- [4] *Fédération Internationale de l'Automobile. FIA Homologations [online]. [cit. 2016-02-23]. Dostupné z: <http://www.fia.com/homologations>*
- [5] *Euro Car Segment. World Public Library [online]. [cit. 2016-02-24]. Dostupné z: http://www.worldlibrary.org/articles/euro_car_segment*
- [6] *Designing a Vehicle Instrument Panel Cluster—A Case Study. Embedded.com [online]. [cit. 2016-03-05]. Dostupné z: <http://www.embedded.com/design/prototyping-and-development/4017937/Designing-a-Vehicle-Instrument-Panel-Cluster-151-A-Case-Study>*
- [7] *AEC - Q100: FAILURE MECHANISM BASED STRESS TEST QUALIFICATION FOR INTEGRATED CIRCUITS.* Rev-H. AEC, 2014. Dostupné z: http://www.aecouncil.com/Documents/AEC_Q100_Rev_H_Base_Document.pdf
- [8] *Protection of automotive electronics from electrical hazards, guidelines for design and component selection. Application note. [online]. 10.04.2016 [cit. 2016-04-10]. Dostupné z: http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/application_note/CD181783.pdf*
- [9] *International Organization for Standardization. ISO. [online]. 10.04.2016 [cit. 2016-04-10]. Dostupné z: <http://www.iso.org/iso/home.htm>*
- [10] *TLE4275-Q1 5-V Low-Dropout Voltage Regulator. Texas Instruments [online]. [cit.2016-04-15]. Dostupné z: <http://www.ti.com/lit/ds/symlink/tle4275-q1.pdf>*
- [11] *LM1575/LM2575/LM2575HV SIMPLE SWITCHER ® 1A Step-Down Voltage Regulator. Texas Instruments [online]. [cit. 2016-04-15]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm1575.pdf>*
- [12] *STM32F745xx , STM32F746xx ARM ® -based Cortex ® -M7 32b MCU+FPU. ST Microelectronics [online]. [cit. 2016-04-15]. Dostupné z: <http://www2.st.com/content/ccc/resource/technical/document/datasheet/96/ed/61/9b/e0/6c/45/0b/DM00166116.pdf/files/DM00166116.pdf/jcr:content/translations/en.DM00166116.pdf>*
- [13] *Micron Serial NOR Flash Memory N25Q128A: Datasheet. Micron [online]. [cit. 2016-04-15]. Dostupné z: <https://www.micron.com/resource-details/4c4af943-8a6b-4a29-b124-08b345151576>*
- [14] *BOOST (STEP-UP) WLED DRIVER WITH OVP: Datasheet TS19371. Taiwan Semiconductor [online]. [cit. 2016-04-15]. Dostupné z: <http://www.taiwansemi.com/en/products/details/TS19371CX6>*
- [15] *Display LCD-TFT 7": Datasheet MCT070M6W800480LML. [online].*

- [cit. 2016-04-15]. Dostupné z: <http://www.midasdisplays.com/products/4-10-14.page>
- [16] Cortex-M7 Processor - ARM. *ARM* [online]. [cit. 2016-04-16]. Dostupné z: <http://www.arm.com/products/processors/cortex-m/cortex-m7-processor.php>
- [17] ISSI SYNCHRONOUS DYNAMIC RAM 64-MBIT: Datasheet IS42S16400J. *Integrated Silicon Solutions Inc.* [online]. [cit. 2016-04-17]. Dostupné z: <http://www.issi.com/WW/pdf/42-45S16400J.pdf>
- [18] STM32F75xxx and STM32F74xxx advanced ARM®-based 32-bit MCUs: Reference manual. *ST Microelectronics* [online]. [cit. 2016-04-17]. Dostupné z: http://www2.st.com/resource/en/reference_manual/dm00124865.pdf
- [19] High Voltage Current Shunt Monitor: Datasheet AD8211. *Analog Devices* [online]. [cit. 2016-04-17]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD8211.pdf>
- [20] SN65HVD23x 3.3-V CAN Bus Transceivers: Datasheet. *Texas Instruments* [online]. [cit. 2016-04-22]. Dostupné z: <http://www.ti.com/lit/ds/symlink/sn65hvd230.pdf>
- [21] Very Low Supply Current 3-Pin Microprocessor Reset Monitor: Datasheet. *ON Semiconductor* [online]. [cit. 2016-04-22]. Dostupné z: http://www.onsemi.com/pub_link/Collateral/MAX803-D.PDF
- [22] XBee and XBee-PRO ZigBee RF Modules. *Digi* [online]. [cit. 2016-04-22]. Dostupné z: <http://www.digi.com/products/xbee-rf-solutions/rf-modules/xbee-zigbee#partnumbers-xbeeprosmt>
- [23] GPS Receiver A2235-H: User's Manual. *Maestro-wireless* [online]. [cit. 2016-04-22]. Dostupné z: <http://www.maestro-wireless.com/a2235-h-technical-specifications/>
- [24] MEMS digital output motion sensor ultra low-power high full-scale 3-axes "nano" accelerometer: Datasheet. *ST Microelectronics* [online]. [cit. 2016-04-22]. Dostupné z: <http://www2.st.com/content/ccc/resource/technical/document/datasheet/58/d2/32/92/1c/e3/43/fd/CD00250937.pdf/files/CD00250937.pdf/jcr:content/translations/en.CD00250937.pdf>
- [25] LM75x Digital Temperature Sensor and Thermal Watchdog With Two-Wire Interface: Datasheet. *Texas Instruments* [online]. [cit. 2016-04-22]. Dostupné z: <http://www.ti.com/lit/ds/symlink/lm75b.pdf>
- [26] ARM® Cortex® -M7 Processor: Technical Reference Manual. *ARM* [online]. [cit. 2016-04-22]. Dostupné z: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0489d/DDI0489D_cortex_m7_trm.pdf
- [27] VLK, František. *Elektronické systémy motorových vozidel*. 1. vyd. Brno: František Vlk, 2002. ISBN 80-238-7282-6.
- [28] BARRETT, Stephen. Cortex-M7 Launches: Embedded, IoT and Wearables. In: *Anandtech* [online]. 2014 [cit. 2016-04-26]. Dostupné z: <http://www.anandtech.com/show/8542/cortexm7-launches-embedded-iot-and-wearables/2>
- [29] Description of STM32F7xx HAL drivers: User Manual. *ST Microelectronics* [online]. [cit. 2016-05-02]. Dostupné z: http://www.st.com/st-web-ui/static/active/cn/resource/technical/document/user_manual/DM00189702.pdf
- [30] CANopen – The standardized embedded network. *CiA* [online]. [cit. 2016-05-02]. Dostupné z: <http://www.can-cia.org/can-knowledge/canopen/canopen/>

- [31] CANopen COMMUNICATION MANUAL. *ADVANCED Motion Controls* [online]. [cit. 2016-05-02]. Dostupné z: http://www.a-m-c.com/download/sw/dw300_3-0-3/CAN_Manual300_3-0-3.pdf
- [32] *XBee*® /*XBee-PRO*® *RF Modules: Product Manual* [online]. [cit. 2016-05-02]. Dostupné z: <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf>

Přílohy

Příloha 1 : Schéma zapojení

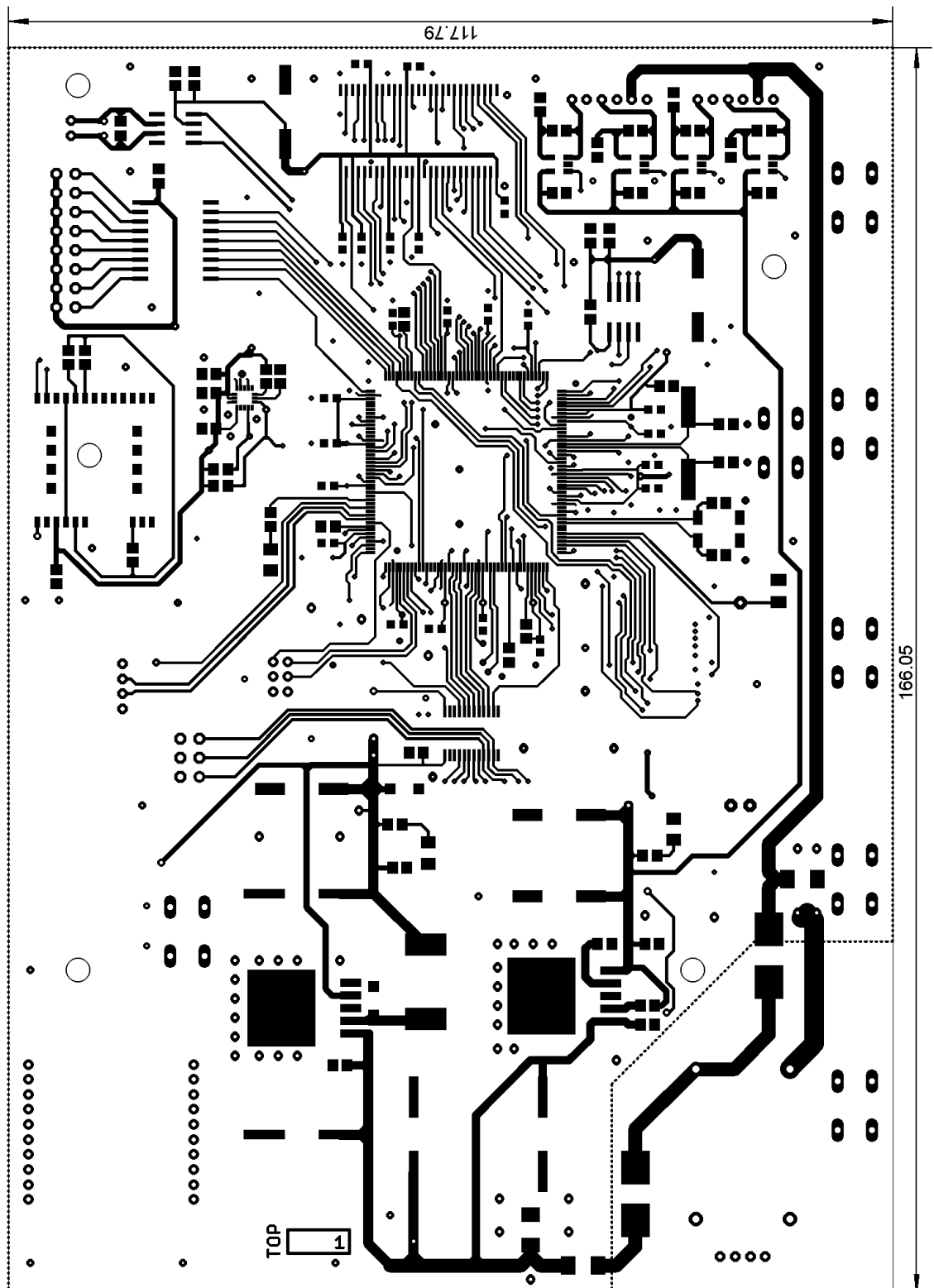
Příloha 2: Pinout STM32F746IGT

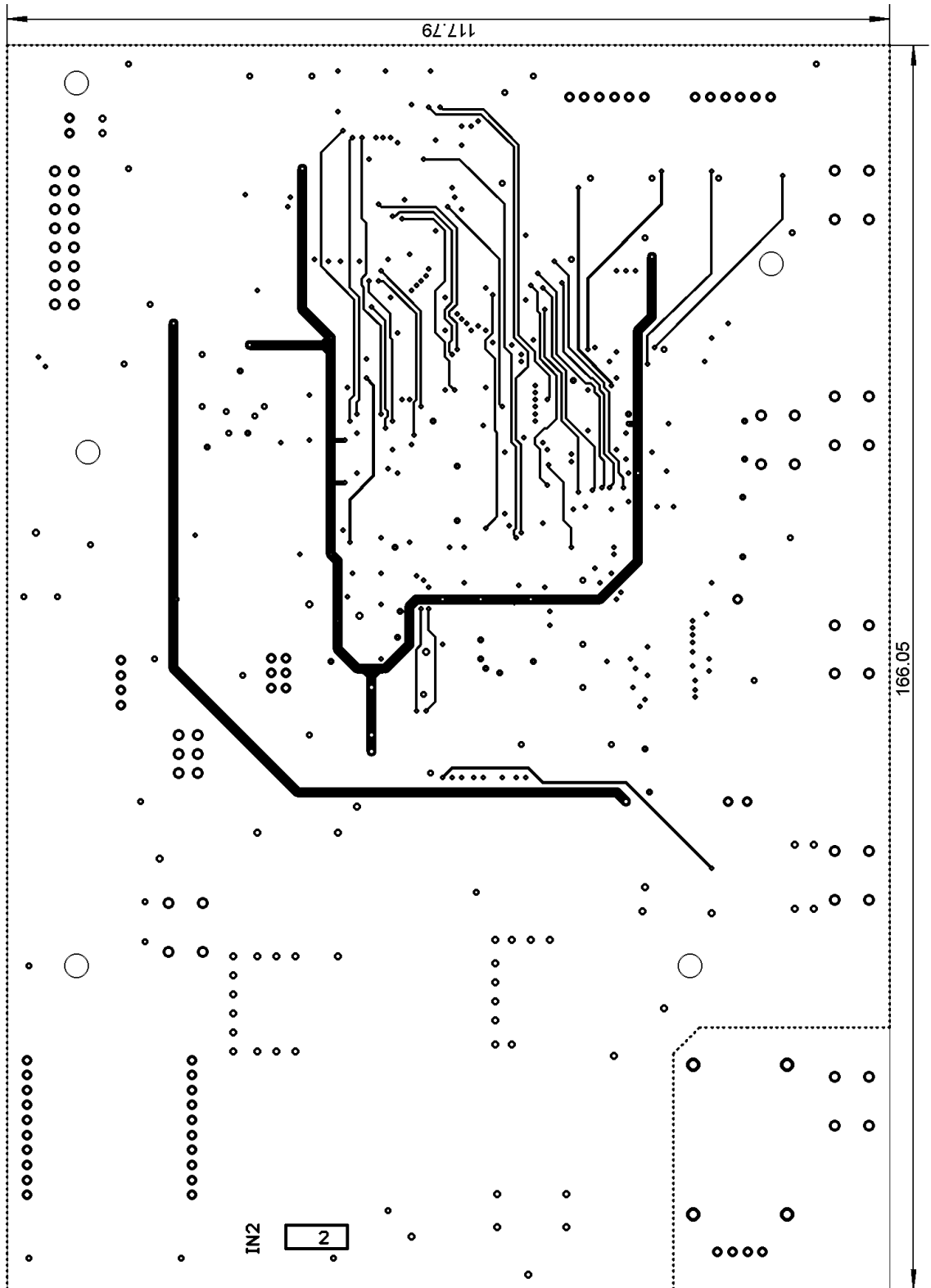
Pin	Name	Type	Signal	Label
1	PE2	I/O	QUADSPI_BK1_IO2	FLASH MEMORY
2	PE3	I/O		
3	PE4	I/O	LTDC_B0	LCD
4	PE5	I/O	LTDC_G0	LCD
5	PE6	I/O	LTDC_G1	LCD
6	VBAT	Power		
7	PI8	I/O		
8	PC13	I/O		
9	PC14/OSC32_IN	I/O	RCC_OSC32_IN	XTAL 32kHz
10	PC15/OSC32_OUT	I/O	RCC_OSC32_OUT	XTAL 32kHz
11	PI9	I/O	CAN1_RX	CAN Rx
12	PI10	I/O	LTDC_HSYNC	LCD
13	PI11	I/O		
14	VSS	Power		
15	VDD	Power		
16	PF0	I/O	FMC_A0	SDRAM
17	PF1	I/O	FMC_A1	SDRAM
18	PF2	I/O	FMC_A2	SDRAM
19	PF3	I/O	FMC_A3	SDRAM
20	PF4	I/O	FMC_A4	SDRAM
21	PF5	I/O	FMC_A5	SDRAM
22	VSS	Power		
23	VDD	Power		
24	PF6	I/O	QUADSPI_BK1_IO3	FLASH MEMORY
25	PF7	Output	GPIO_Output	LED DIODE1
26	PF8	I/O	QUADSPI_BK1_IO0	FLASH MEMORY
27	PF9	I/O	QUADSPI_BK1_IO1	FLASH MEMORY
28	PF10	I/O	LTDC_DE	LCD
29	PH0/OSC_IN	I/O	RCC_OSC_IN	XTAL 8MHz
30	PH1/OSC_OUT	I/O	RCC_OSC_OUT	XTAL 8MHz
31	NRST	Reset		
32	PC0	I/O	LTDC_R5	LCD
33	PC1	I/O		
34	PC2	I/O	FMC_SDNE0	SDRAM CS
35	PC3	I/O	FMC_SDCKE0	SDRAM CKE
36	VDD	Power		
37	VSSA	Power		
38	VREF+	Power		
39	VDDA	Power		
40	PA0/WKUP	I/O	ADC1_IN0 ADC2_IN0	ADC_ACCEL
41	PA1	I/O	ADC1_IN1 ADC2_IN1	ACD_ACCEL2
42	PA2	I/O	ADC1_IN2 ADC2_IN2	ADC_BRAKE
43	PH2	I/O	LTDC_R0	LCD
44	PH3	I/O	LTDC_R1	LCD
45	PH4	Output	GPIO_Output	LED DIODE2
46	PH5	I/O	FMC_SDNWE	SDRAM
47	PA3	I/O	ADC1_IN3 ADC2_IN3	ADC_BRAKE2
48	BYPASS_REG	Reset		
49	VDD	Power		

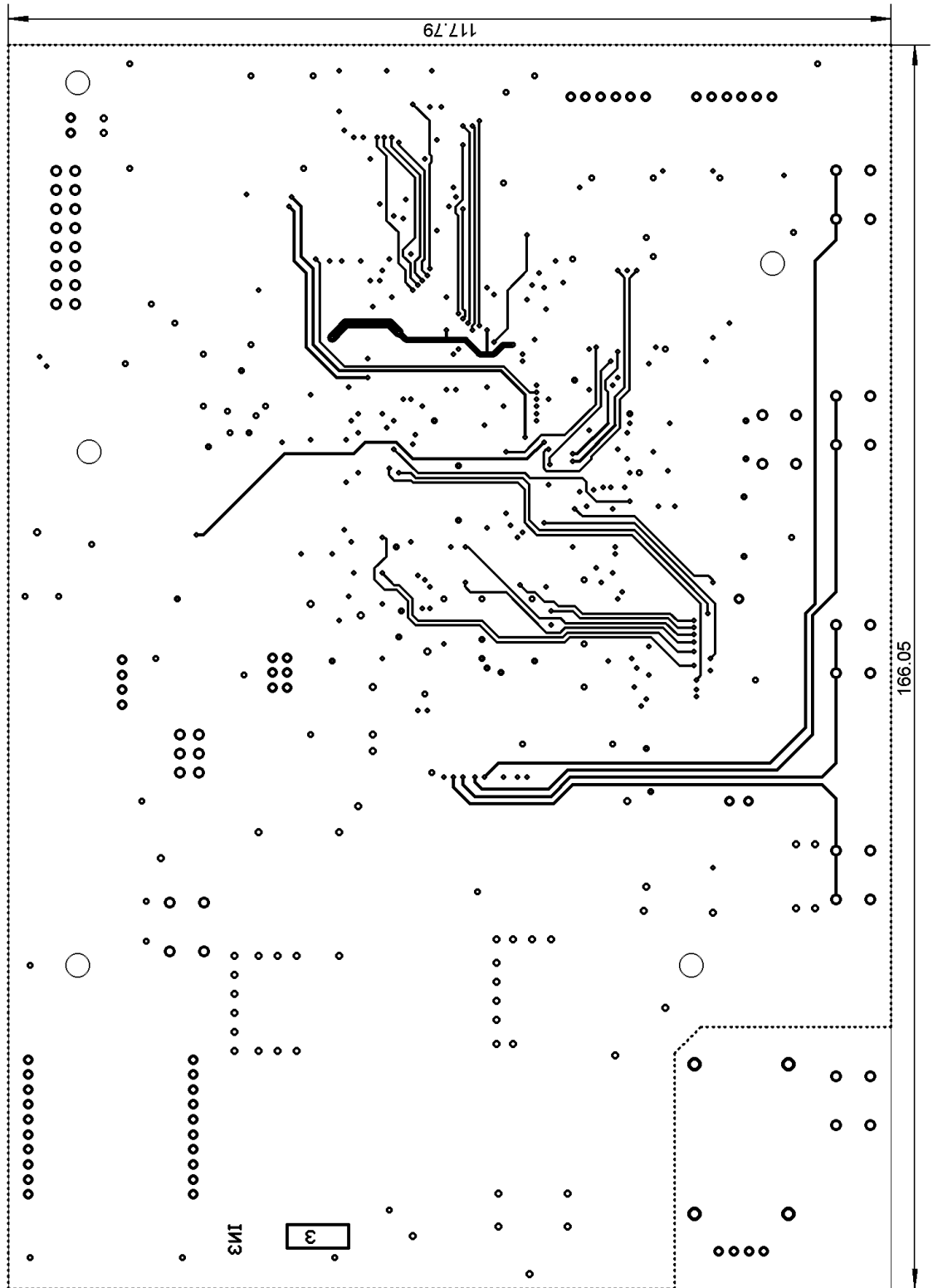
50	PA4	I/O	LTDC_VSYNC	LCD
51	PA5	I/O	LTDC_R4	LCD
52	PA6	I/O	LTDC_G2	LCD
53	PA7	I/O		
54	PC4	I/O		
55	PC5	I/O		
56	PB0	I/O	LTDC_R3	LCD
57	PB1	I/O	LTDC_R6	LCD
58	PB2	I/O	QUADSPI_CLK	FLASH MEMORY
59	PF11	I/O	FMC_SDNRAS	SDRAM
60	PF12	I/O	FMC_A6	SDRAM
61	VSS	Power		
62	VDD	Power		
63	PF13	I/O	FMC_A7	SDRAM
64	PF14	I/O	FMC_A8	SDRAM
65	PF15	I/O	FMC_A9	SDRAM
66	PG0	I/O	FMC_A10	SDRAM
67	PG1	I/O	FMC_A11	SDRAM
68	PE7	I/O	FMC_D4	SDRAM
69	PE8	I/O	FMC_D5	SDRAM
70	PE9	I/O	FMC_D6	SDRAM
71	VSS	Power		
72	VDD	Power		
73	PE10	I/O	FMC_D7	SDRAM
74	PE11	I/O	FMC_D8	SDRAM
75	PE12	I/O	FMC_D9	SDRAM
76	PE13	I/O	FMC_D10	SDRAM
77	PE14	I/O	FMC_D11	SDRAM
78	PE15	I/O	FMC_D12	SDRAM
79	PB10	I/O	LTDC_G4	LCD
80	PB11	I/O	LTDC_G5	LCD
81	VCAP_1	Power		
82	VDD	Power		
83	PH6	Output	GPIO_Output	LED DIODE3
84	PH7	Output	GPIO_Output	LED DIODE4
85	PH8	I/O	LTDC_R2	LCD
86	PH9	Output	GPIO_Output	LED DIODE5
87	PH10	Output	GPIO_Output	LED DIODE6
88	PH11	Output	GPIO_Output	LED DIODE7
89	PH12	Output	GPIO_Output	LED DIODE8
90	VSS	Power		
91	VDD	Power		
92	PB12	I/O		
93	PB13	I/O		
94	PB14	I/O		
95	PB15	I/O		
96	PD8	I/O	FMC_D13	SDRAM
97	PD9	I/O	FMC_D14	SDRAM
98	PD10	I/O	FMC_D15	SDRAM
99	PD11	I/O		
100	PD12	I/O		
101	PD13	I/O		
102	VSS	Power		
103	VDD	Power		
104	PD14	I/O	FMC_D0	SDRAM
105	PD15	I/O	FMC_D1	SDRAM
106	PG2	I/O		
107	PG3	Output	GPIO_Output	LCD_ON
108	PG4	I/O	FMC_BA0	SDRAM
109	PG5	I/O	FMC_BA1	SDRAM

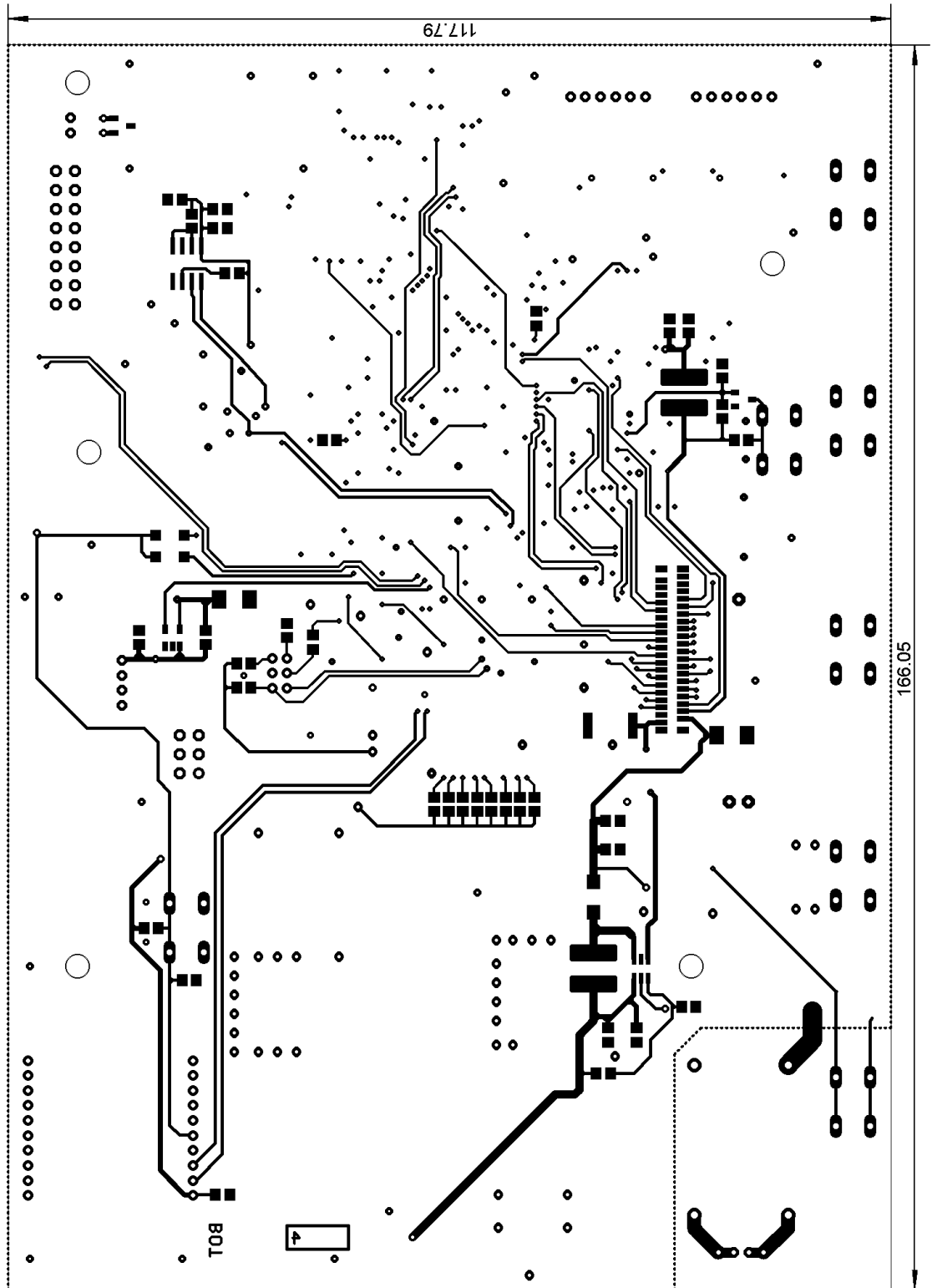
110	PG6	I/O	LTDC_R7	LCD
111	PG7	I/O	LTDC_CLK	LCD
112	PG8	I/O	FMC_SDCLK	SDRAM
113	VSS	Power		
114	VDDUSB	Power		
115	PC6	I/O		
116	PC7	I/O	LTDC_G6	LCD
117	PC8	I/O		
118	PC9	Output	GPIO_Output	BLINKING LED
119	PA8	I/O		
120	PA9	I/O	USB_OTG_FS_VBUS	USB
121	PA10	I/O	USB_OTG_FS_ID	USB
122	PA11	I/O	USB_OTG_FS_DM	USB
123	PA12	I/O	USB_OTG_FS_DP	USB
124	PA13	I/O	SYS_JTMS-SWDIO	JTAG
125	VCAP_2	Power		
126	VSS	Power		
127	VDD	Power		
128	PH13	I/O	CAN1_TX	CAN Tx
129	PH14	I/O	LTDC_G3	LCD
130	PH15	Input	GPIO_Input	BUTTON1
131	PI0	I/O		
132	PI1	Output	GPIO_Output	PERIPH RESET
133	PI2	I/O	LTDC_G7	LCD
134	PI3	Input	GPIO_Input	BUTTON2
135	VSS	Power		
136	VDD	Power		
137	PA14	I/O	SYS_JTCK-SWCLK	JTAG
138	PA15	I/O	SYS_JTDI	JTAG
139	PC10	I/O	UART4_TX	XBEE Din
140	PC11	I/O	UART4_RX	XBEE Dout
141	PC12	I/O	UART5_TX	GPS RX0
142	PD0	I/O	FMC_D2	SDRAM
143	PD1	I/O	FMC_D3	SDRAM
144	PD2	I/O	UART5_RX	GPS TX0
145	PD3	I/O		
146	PD4	Output	GPIO_Output	XBee /SLEEP
147	PD5	Output	GPIO_Output	USB_ON
148	VSS	Power		
149	VDD	Power		
150	PD6	I/O	LTDC_B2	LCD
151	PD7	Input	GPIO_Input	BUTTON3
152	PG9	Input	GPIO_Input	BUTTON4
153	PG10	Input	GPIO_Input	BUTTON5
154	PG11	I/O	LTDC_B3	LCD
155	PG12	I/O	LTDC_B1	LCD
156	PG13	Input	GPIO_Input	BUTTON6
157	PG14	Input	GPIO_Input	BUTTON7
158	VSS	Power		
159	VDD	Power		
160	PG15	I/O	FMC_SDNCAS	SDRAM
161	PB3	I/O	SYS_JTDO-SWO	JTAG
162	PB4	I/O	SYS_JTRST	JTAG
163	PB5	I/O		
164	PB6	I/O	QUADSPI_BK1_NCS	FLASH MEMORY
165	PB7	I/O	I2C1_SDA	I2C
166	BOOT0	Boot		
167	PB8	I/O	I2C1_SCL	I2C
168	PB9	I/O	LTDC_B7	LCD
169	PE0	I/O	FMC_NBL0	SDRAM
170	PE1	I/O	FMC_NBL1	SDRAM
171	PDR_ON	Reset		
172	VDD	Power		
173	PI4	I/O	LTDC_B4	LCD
174	PI5	I/O	LTDC_B5	LCD
175	PI6	I/O	LTDC_B6	LCD
176	PI7	Input	GPIO_Input	BUTTON8

Příloha 3: Layout desky plošných spojů

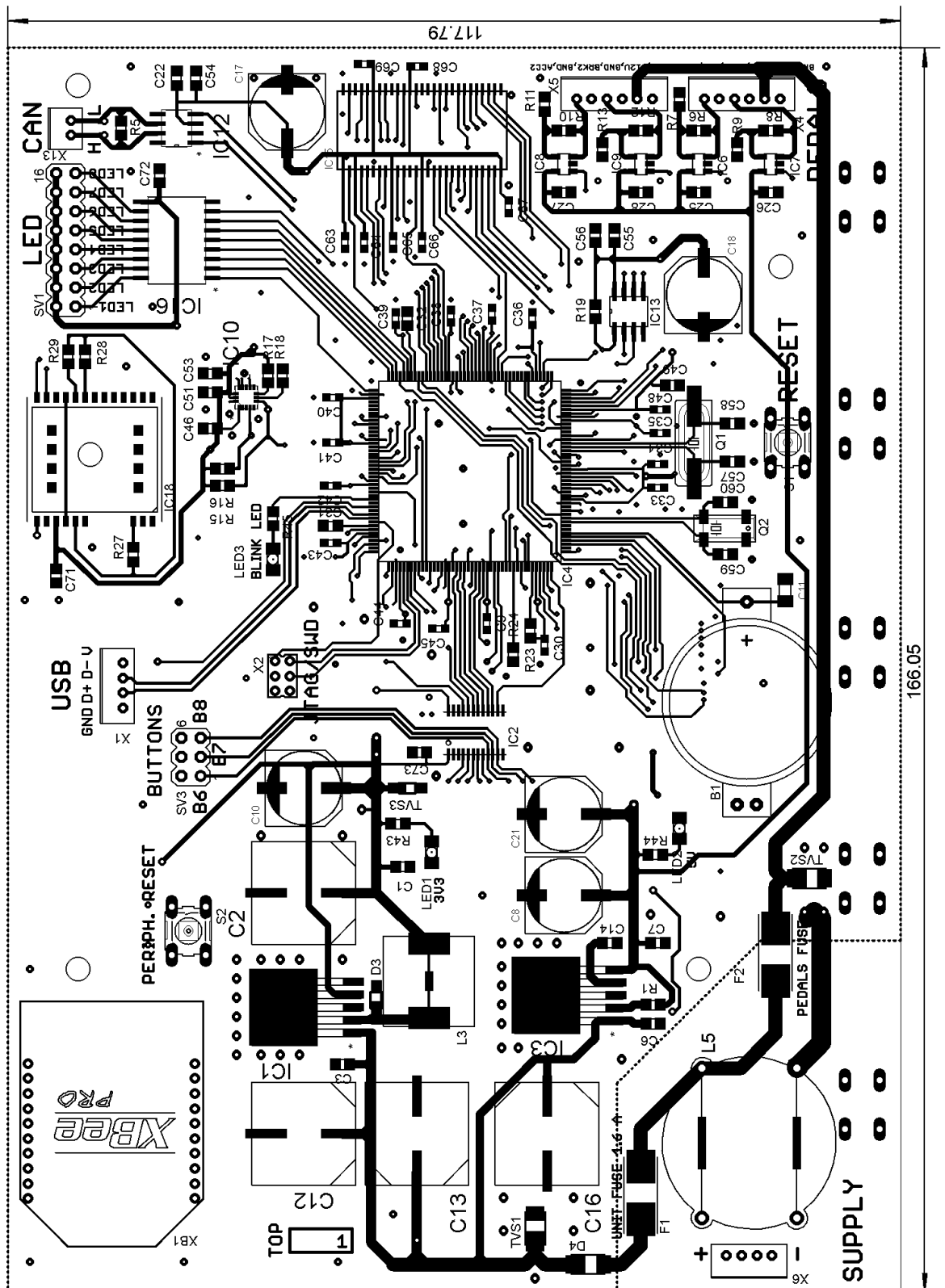








Příloha 4 : Osazovací schéma DPS



Příloha 5 : Fotografie jednotky displeje

