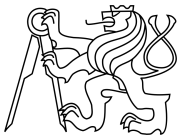
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Model přístrojové desky automobilu
Student:	Petr Socha
Vedoucí:	Ing. Miroslav Skrbek, Ph.D.
Studijní program:	Informatika
Studijní obor:	Počítačové inženýrství
Katedra:	Katedra číslicového návrhu
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Navrhněte a realizujte model přístrojové desky automobilu s ručičkovými a LED indikátory a displejem palubního počítače, který bude ovládán přes sběrnici CAN. Ručičkové indikátory realizujte modelářskými servy, včetně potřebných převodů. Jako řídící modul použijte vývojový kit KEA128 StarterTRAK s rozhraním sběrnice CAN a procesorem ARM. Implementujte potřebné programové vybavení pro interpretaci zpráv ze sběrnice CAN. Propracovanost modelu přístrojové desky a rozsah implementačních prací upřesněte po dohodě s vedoucím práce.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

doc. Ing. Hana Kubátová, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 16. prosince 2015

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra číslicového návrhu

Bakalářská práce

Model přístrojové desky automobilu

Petr Socha

Vedoucí práce: Ing. Miroslav Skrbek, Ph.D.

17. května 2016

Poděkování

Rád bych na tomto místě poděkoval panu Miroslavu Skrbkovi, vedoucímu mé práce, za pomoc s výběrem tématu, zapůjčení hardware a mnoho cenných nápadů a rad. Dále děkuji Miru Hrončokovi z 3Dprint LAB za pomoc s tiskem mechanických dílů. Díky za spolupráci patří také Pavlu Zajícovi, autorovi související práce. Na závěr děkuji také všem svým přátelům a rodině za podporu během celého mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 17. května 2016 .....

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2016 Petr Socha. Všechna práva vyhrazena.
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Socha, Petr. *Model přístrojové desky automobilu*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Cílem této práce je vytvořit fyzický model automobilové přístrojové desky, zobrazující pomocí ručičkových i světelných ukazatelů informace o stavu vozidla. Pro komunikaci s ostatními komponentami využívá v automobilech hojně rozšířené sběrnice CAN. Práce by měla sloužit jako výuková pomůcka na Fakultě informačních technologií na ČVUT v Praze.

Klíčová slova automobil, vozidlo, přístrojová deska, ukazatele, sběrnice CAN

Abstract

The primary goal of this work is to create a car dashboard, which uses gauges and lights to indicate various vehicle conditions. Communication with other components is realized using CAN bus, which is widely used in the automotive industry. The work should become educational material at the Faculty of information technology at CTU in Prague.

Keywords car, vehicle, dashboard, indicators, gauges, CAN bus

Obsah

Úvod **1**

1 Automobilové řídící systémy **3**
1.1 Přístrojová deska automobilu 4
1.2 Současné automobilové systémy 4
1.3 Sběrnice Controller Area Network 6

2 Návrh přístrojové desky **9**
2.1 Koncept modelu přístrojové desky 10
2.2 Mechanické prvky 11
2.3 Použitá elektronika a hardware 12
2.4 Použité softwarové prostředky 16
2.5 Komunikační protokol 17
2.6 Návrh programového vybavení 19

3 Realizace **23**
3.1 Plastové mechanické díly 23
3.2 Plošný spoj 25
3.3 Zapojení další elektroniky 28
3.4 Programové vybavení KEA128 30
3.5 Demoaplikace na STM32F429ZI 39

4 Testování **43**
4.1 Plošný spoj 43
4.2 Servomotory a převody 44
4.3 LED diody 44
4.4 Kinetis KEA128 44
4.5 STM32F429ZI Discovery Kit 45

Závěr **47**

Literatura **49**

A Seznam použitých zkratek **53**

B Obsah přiloženého CD **55**

C Seznam součástek a schéma plošného spoje **57**

D Maska plošného spoje **59**

Seznam obrázků

1.1 Příklad realizace automobilové sběrnice 7

2.1 Návrh vzhledu přístrojové desky 11
2.2 Blokové schéma hardwaru přístrojové desky 12
2.3 Blokové schéma programového vybavení pro Kinetis KEA128 . . . 20

3.1 Realizované úchyty a převody servomotorů 24
3.2 Realizovaný plošný spoj 25
3.3 Schéma spínaného zdroje 26
3.4 Schéma budiče LED diod 27
3.5 Zapojené vývody Kinetis KEA128 29
3.6 Výsledná přístrojová deska 41

C.1 Schéma plošného spoje 58

D.1 Maska plošného spoje ze strany součástek 59
D.2 Maska plošného spoje ze strany spojů 60

Úvod

Cílem práce je realizace modelu přístrojové desky automobilu, informující uživatele/řidiče o aktuální rychlosti vozidla, otáčkách motoru, stavu nádrže, světel a dalších důležitých informacích. Tyto informace deska získává z připojené automobilové sběrnice CAN.

Přístrojová deska by jako samostatný modul měla být součástí obsáhlejšího modelu elektroniky vozidla. Ten bude sloužit jako výuková pomůcka v předmětu Pokročilé vestavné systémy na FIT ČVUT.

Základním stavebním kamenem modelu je vývojová deska řady Kinetis společnosti Freescale, obsahující procesor ARM s integrovanými řadiči automobilových sběrnic CAN a LIN. Ručičkové ukazatele jsou realizovány servomotory, světelné pak pomocí LED diod a zařízení doplňuje dotykový display palubního počítače. Realizace programového vybavení palubního počítače však není náplní této práce.

V následujícím textu se nejprve zaměřím na současné automobilové systémy, způsob jakým je v nich realizována komunikace a směr, kterým se jejich vývoj ubírá. Dále se zabývám návrhem vlastního modelu přístrojové desky, výběrem vhodných komponent a jejich možnostmi. Ve třetí kapitole popisuji konkrétní realizaci mechanických dílů, zapojení elektronických součástek a programové vybavení mikrokontroléru. Na závěr práce popisuji jednoduché metody, použité pro ověření funkčnosti.

Kapitola 1

Automobilové řídící systémy

Od devatenáctého století, kdy se poprvé objevil koncept moderního vozidla tak, jak jej známe dnes, uplynula dlouhá doba a automobily se v různých podobách, od osobních po nákladní, staly nenahraditelnou součástí našich každodenních životů. Za dobu své existence prošly mohutným vývojem, nejen v oblasti strojírenství.

Celosvětově vysoká poptávka vede automobilový trh k neustálému zvyšování výkonu vozidel a rozšiřování funkcionality, ať už co se týče řízení nebo dnes například multimediálních zábavních a asistenčních systémů. Zároveň je trhem i vládními autoritami vyvíjen na výrobce tlak v oblasti snižování spotřeby a množství produkovaných emisí. Vzhledem k celkově vysoké konkurenci je důležitým faktorem také cena a tak je vysoký důraz kladen na zkrácení doby vývoje a snížení výrobních nákladů na minimum [1].

V šedesátých a sedmdesátých letech minulého století se na trhu objevila první analogová zařízení pro vstřikování paliva, řízení převodovky a karburátoru. Palivová krize v USA v sedmdesátých letech položila základ snaze vyvíjet vozidla s efektivnějším spalováním a v roce 1976 se v automobilech objevují první mikrokontroléry řízené systémy vstříkování paliva [2].

Spolu s rozvojem elektrotechniky a polovodičů se tak automobily stávají komplexními elektronicky řízenými systémy. Zatímco cena elektroniky v průměrném automobilu v roce 1977 byla $110 (amerických dolarů), v roce 2004 to bylo již $1341. Řídící elektronické systémy podle odhadů mohou tvořit až 25% ceny vozidla. Moderní vůz dnes obsahuje až 100 mikrokontrolérů, zajišťujících různé funkce [2].

Kromě méně kritických systémů se v dnešní době stále častěji objevují také systémy typu Drive-by-wire, přenášející všechny potřebné informace pro řízení vozidla, jako jsou poloha volantu, plynového a brzdovéhu pedálu, v digitální podobě [1][2]. Zvyšují se tedy také nároky na spolehlivost a bezpečnost takového systému.

1.1 Přístrojová deska automobilu

Důležitým úkolem automobilového systému je také komunikace s řidičem (anglicky Human-Machine Interface). Typickými vstupy jsou například volant s pedály, různá tlačítka a přepínače. Jednou z nejdůležitějších výstupních komunikačních jednotek je pak přístrojová deska.

Nejčastěji na přístrojové desce zobrazovanými údaji jsou rychlost vozidla a otáčky motoru, stav palivové nádrže, teplota chladící kapaliny nebo tlak oleje. Tyto údaje byly v minulosti přenášeny často analogovou cestou a jejich nejjednodušší přirozenou reprezentací byly ručičkové ukazatele. V moderních automobilech jsou data již obvykle získávána digitálně, z důvodu oblíbenosti a snadného čtení si však ručičkové ukazatele stále drží svou pozici na trhu, ačkoli v některých nových vozidlech jsou již plně nahrazeny displeji.

Pomocí světelných kontrolek nebo displejů jsou dále zobrazovány údaje o stavu dalších komponenent vozidla, jako světel, převodovky, stěračů a podobně. Významnou část kontrolek také tvoří výstrahy, například od brzdného systému, posilovače řízení a podobně.

V neposlední řadě zprostředkovává moderní přístrojová deska rozhraní mezi uživatelem a palubním počítačem, jehož funkce mohou být rozmanité. Vedle interaktivní konfigurace komponent vozidla, jako třeba tempomatu nebo parkovacího asistenta, se v současné době rozšiřují multimediální systémy, umožňující například párování vozidla s mobilním telefonem pomocí Bluetooth nebo využívající možností satelitní navigace. S tím souvisí také nové možnosti využití v oblasti bezpečnosti řidiče nebo zabezpečení vozidla proti krádeži [1].

1.2 Současné automobilové systémy

Zvyšující se množství vzájemně propojených komponent a předávaných informací vedlo k potřebě optimalizace komunikace. Namísto mechanických prvků byly zpočátku využívány dedikované P2P spoje, jejich množství však začalo brzy překračovat únosnou mez.

Namísto nich tedy byly zavedeny sdílené multiplexované sítě. Dnešní hardwarová architektura automobilu je typicky tvořena broadcastovou sběrnicí, k níž jsou připojeny jednotlivé samostatně funkční jednotky [1][2]. Ty se obvykle sestávají z vlastního mikrokontroléru, paměti a příslušných vstupně výstupních obvodů.

1.2.1 Souhrn komponent moderního vozidla

Při návrhu funkcionality přístrojové desky a komunikačního protokolu je třeba vzít v úvahu všechny komponenty, které se v moderním automobilu vyskytují a vzájemně komunikují.

Právě od nich přístrojová deska získává aktuální informace o stavu vozidla, které zobrazuje operátorovi/řidiči. Mezi nejvýznamnější řídící jednotky vyskytující se v automobilech patří: [1]

- řízení vstřikování paliva,
- regulace emisí,
- brzdový systém,
- automatická převodovka,
- tempomat, a podobně ...

Kromě těchto jednotek je ve vozidle široká škála senzorů, měřících: [1]

- otáčky motoru,
- rychlost vozidla,
- proudění vzduchu,
- pozici řadící páky,
- úhel natočení volantu,
- pozici plynového a brzdového pedálu, a podobně ...

Vzhledem k velkému množství různých komponent s různými požadavky je vhodné jednotlivé řídící jednotky kategorizovat. Dělení na základě funkcionality, od kterého je do jisté míry odvozeno též hierarchické členění sítě v automobilu, je následující: [1]

1. **řízení hnací soustavy** (anglicky powertrain), kde najdeme především systémy řídící motor, zapalování, převodovku, a podobně (o motor a převodovku se často stará jediná řídící jednotka),
2. **řízení vozidla** (anglicky vehicle control), jenž zahrnuje kontrolu tlumičů, volantu, brzd, tempomatu nebo třeba kontrolu trakce (tuto oblast v posledních letech provází bouřlivý rozvoj, patří sem například systémy pro automatické paralelní parkování),
3. a **řízení palubních systémů** (anglicky body control), kam patří například automatická klimatizace, zabezpečovací systémy nebo ovládání oken. Smyslem těchto systémů je obvykle zvýšení jízdního komfortu. *Do této kategorie patří také přístrojová deska spolu s dalšími indikátory.*

Podle míry kritičnosti můžeme také první dvě kategorie, řízení hnací soustavy a řízení vozidla, charakterizovat jako **systémovou elektroniku** a řízení palubních systémů jako **palubní elektroniku** [2]. Časové požadavky systémové elektroniky se pohybují mezi *jednotkami až desítkami milisekund*, zatímco časové požadavky palubní elektroniky jsou určeny reakčním časem řidiče, což je mezi *100 až 200 milisekundami* [2].

1.2.2 Komunikační síť moderního vozidla

Úkolem sítě uvnitř vozidla je co nejefektivněji zprostředkovat vzájemnou výměnu informací mezi jednotlivými komponentami a zároveň dodržet časové požadavky. Z toho důvodu je správný výběr a návrh komunikační sítě kritickou záležitostí.

Pro jednotky, jejichž komunikace je méně kritická, jako ovládání dveří, oken, sedadel nebo zrcátek se používají levnější a pomalejší řešení, jako je sběrnice Local Interconnect Network (LIN) navržená společností Motorola. Na sběrnici LIN je vždy právě jedna jednotka, iniciující komunikaci, a nejvýše 15 podřízených jednotek. Z toho důvodu se hodí spíše pro stavbu menších sítí, které mohou být vzájemně propojeny pomocí některé z dále zmíněných sběrnic [1].

Přenos multimediálních dat, pro realizaci nejen zábavního systému, ale třeba také handsfree a dalších funkcí, je často zajištěn sběrnicí Media Oriented Systems Transport (MOST), uzpůsobenou pro použití v automobilech.

Jsou-li přenášené informace kritické pro bezpečnost a výkon vozidla, obzvláště u systémů drive-by-wire, kde je bezpodmínečně nutné zajistit dodržení časových limitů, využívají se pro jejich přenos vysokorychlostní sběrnice Controller Area Network (CAN) nebo nověji také FlexRay [1]. Sběrnicí CAN se podrobněji zabývám v následující kapitole.

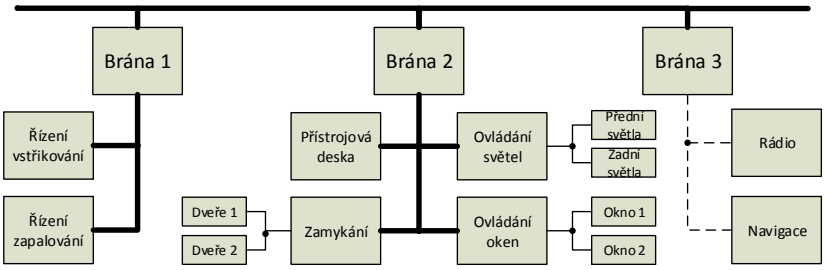
S ohledem na různé časové požadavky a kritičnost připojených komponent je ve vozidle obvykle realizováno několik sběrnic (tedy několik různých kolizních domén) vzájemně propojených pomocí bran (gateways) [1][2].

1.3 Sběrnice Controller Area Network

Controller Area Network (CAN) je sběrnice navržená společností Bosch pro použití v automobilovém průmyslu [3]. Standard z roku 1991 definuje objektovou a přenosovou vrstvu (anglicky object, transfer layer), které společně tvoří spojovou vrstvu (anglicky data-link layer) definovanou ISO/OSI modelem.

Fyzická vrstva, která zajišťuje přenos jednotlivých bitů, není původní specifikací firmy Bosch nijak upravena a může být tedy zvolena libovolně na základě konkrétních požadavků [3]. V automobilech často využívaná norma ISO 11898-2:2003 definuje pro vysokorychlostí sběrnici CAN fyzickou vrstvu jako dva symetrické vodiče [4].

Mezi nejvýznamnější vlastnosti sběrnice CAN patří [3]:



Obrázek 1.1: Příklad možné realizace sítě ve vozidle. Tučnou čarou je znázorněna sběrnice CAN nebo FlexRay, tenkou LIN a přerušovanou čarou MOST.

1. **priorita zpráv**, tedy zaručení toho, že zprávy s důležitějšími daty budou odeslány dříve, než zprávy méně důležité (s nižší prioritou),
2. **garantovaná latence zpráv**,
3. **multi-cast přenos zpráv**, tedy jednu odeslanou zprávu může přijmout více příjemců zároveň (specifikace firmy Bosch definuje také prostředky pro časovou synchronizaci),
4. **multi-master přenos**, tedy každý připojený uzel může vysílat data na sběrnici a zároveň přijímat data od ostatních uzlů,
5. **detekce chyb a vadných komunikačních uzlů**, které se při detekci poruchy samy odpojí od sběrnice,
6. **automatické znovu-odeslání nepotvrzených nebo chybně odeslaných dat** ihned, jakmile je sběrnice volná (nikdo nevysílá data) a za předpokladu, že se jiný uzel nepokusí přenést zprávu s vyšší prioritou.

Přenosová rychlost na sběrnici CAN může být různá, až do 1MBit/s, v každém jednotlivém systému je však pevná a neměnná [3].

1.3.1 Formát zpráv

Specifikace CAN verze 2 společnosti Bosch definuje dva druhy zpráv: se standardním identifikátorem délky 11 bitů a zprávu s rozšířeným identifikátorem o délce 29 bitů [3]: Každá zpráva na sběrnici začíná start sekvencí, tvořenou jedním dominantním bitem. Vysílač smí tento bit vyslat pouze tehdy, pokud je sběrnice neaktivní, tedy na ní neprobíhá žádný přenos.

Bezprostředně po něm následuje 11 bitů standardního identifikátoru nebo prvních 11 bitů rozšířeného identifikátoru.

Následující bit u standardní zprávy signalizuje RTR rámec (viz dále), u zprávy s rozšířeným identifikátorem je namísto něj substituční bit, který je vždy recesivní.

Další bit rozlišuje mezi typy identifikátorů: recesivní pro rozšířenou a dominantní pro standardní délku identifikátoru. Z toho plyne také vyšší priorita zpráv se standardní délkou identifikátoru (viz dále). V případě zprávy s rozšířeným identifikátorem nyní následuje posledních 18 bitů identifikátoru a RTR bit.

Dále zpráva obsahuje šestibitové řídící pole. Jeho první dva bity jsou rezervovány, další čtyři pak udávají délku dat v bajtech, která bezprostředně za ním následují. Nejvyšší možná velikost dat je osm bajtů, nejmenší nula. Po datech následuje kód CRC.

Po kontrolním součtu následuje potvrzovací (ACK) slot, v němž vysílač vysílá recesivní bit a naslouchá sběrnici. *Všechny přijímače, které zprávu přijmou, ji potvrdí vysláním dominantního bitu* na sběrnici právě v této chvíli. Pokud není zpráva potvrzena, považuje se za nedoručenou a je odeslána znovu, jakmile je sběrnice volná.

Detekují-li přijímače chybný rámec (neodpovídající formát nebo špatný kontrolní součet), vyšlou na sběrnici chybový rámec.

Obdobně v případě přetížení sběrnice vysílají uzly overload rámec.

Zvláštní typ zpráv tvoří tzv. Remote Transmittion Request (viz výše zmíněný RTR bit), česky Vzdálená žádost o vysílání. V případě speciálních aplikací může uzel pomocí této zprávy žádat jiný vzdálený uzel (například senzor) o zaslání dat. V případě kritických aplikací, kdy více jednotek současně a pravidelně vyžaduje stejný typ zprávy, je úspornější periodické zasílání dat senzorem, namísto zatěžování sběrnice žádostmi.

1.3.2 Arbitrace

CAN je sběrnice s otevřeným kolektorem: dominantní úroveň (tedy například logická nula) zvítězí nad recesivní úrovní (logickou jedničkou). Tohoto principu je využito při arbitraci.

Připojená jednotka může zahájit komunikaci pouze tehdy, pokud je sběrnice klidná - neprobíhá aktuálně žádný přenos. Během vysílání uzel zároveň monitoruje skutečný stav sběrnice.

Pokud začnou současně vysílat dvě jednotky různé zprávy ve stejný čas, dojde během vysílání identifikátoru ke kolizi: vysílač, přenášející zprávu s nižší prioritou (vyšším identifikátorem) zjistí při vysílání recesivního bitu na sběrnici dominantní úroveň, tedy opačnou hodnotu, než vysílá. Tento vysílač v tuto chvíli přestává vysílat a ustupuje vysílači s důležitější zprávou. Díky tomuto principu přidělování sběrnice nedochází ke zbytečné ztrátě času, ani informace [3].

Na sběrnici CAN je tedy identifikátor využit nikoliv k adresaci cílového uzlu, nýbrž k identifikaci obsahu zprávy a určení její priority.

Kapitola 2

Návrh přístrojové desky

Při návrhu práce byla jedním z kritérií podobnost s přístrojovými deskami současných automobilů. Vizuální podoba, alespoň v rámci konceptu umístění indikátorů, je u mnoha výrobců obdobná a dle mého názoru byla zpočátku určena prostorovými omezeními a dnes především historicky (zvyklost trhu). Funkční požadavky vyplývají z principu činnosti automobilu, který je také u většiny vozidel obdobný.

Přístrojová deska bude obsahovat čtyři ručičkové ukazatele:

1. **rychlost vozidla** v kilometrech za hodinu,
2. **otáčky motoru** v tisících otáček za minutu,
3. **stav nádrže** v procentech a
4. **teplota chladící kapaliny** ve stupních Celsia.

Doplňuje je čtrnáct kontrolek:

1. **potkávací světla**,
2. **dálková světla**,
3. **přední mlhová světla**,
4. **zadní mlhová světla**,
5. **levé směrové světlo**,
6. **pravé směrové světlo**,
7. **zapnuté stěrače**,
8. **nedostatek paliva**,
9. **výstraha akumulátoru**,

10. **výstraha chlazení motoru**,
11. **výstraha brzdového systému**,
12. **výstraha mazání motoru**,
13. **obecná výstraha motoru**,
14. **výstraha systému ABS**.

Instalovaná vývojová deska od společnosti ST, s barevným displayem a s dotykovou vrstvou, může v budoucnu funkcionalitu přístrojové desky dále rozšířit například o parkovacího asistenta, navigaci, audio přehrávač nebo podobně.

Dalším neméně významným kritériem zvažovaným při návrhu přístrojové desky bylo plánované využití práce jako výukové pomůcky. Souběžně s touto bakalářskou prací vzniká další práce zabývající se připojením USB herního volantu a pedálů ke sběrnici CAN. V budoucnu je plánován také vývoj modulu pro ovládání světel a využívajícího sběrnic CAN a LIN. Kromě těchto komponent by měl být na sběrnici ideálně připojen také simulátor fyzického modelu vozidla, zajišťující příjem a vysílání zpráv od jednotek a senzorů, které se ve výsledném modelu vyskytovat nebudou (jako motor, brzdy a podobně).

Tyto samostatně funkční části by měly být schopny vzájemné komunikace. Přístrojová deska a volant by měly z hlediska konstrukce tvořit jeden celek.

Všechny tyto skutečnosti bylo nutno zohlednit nejen při návrhu a realizaci mechanických dílů, ale také při návrhu komunikačního protokolu. Konkrétní realizace komunikace ve vozidle je obvykle obestřena obchodním tajemstvím výrobců automobilů a není tedy možné z těchto informací čerpat. Proto byla výsledná mechanická konstrukce i komunikační protokol navržen především s ohledem na jednoduchost.

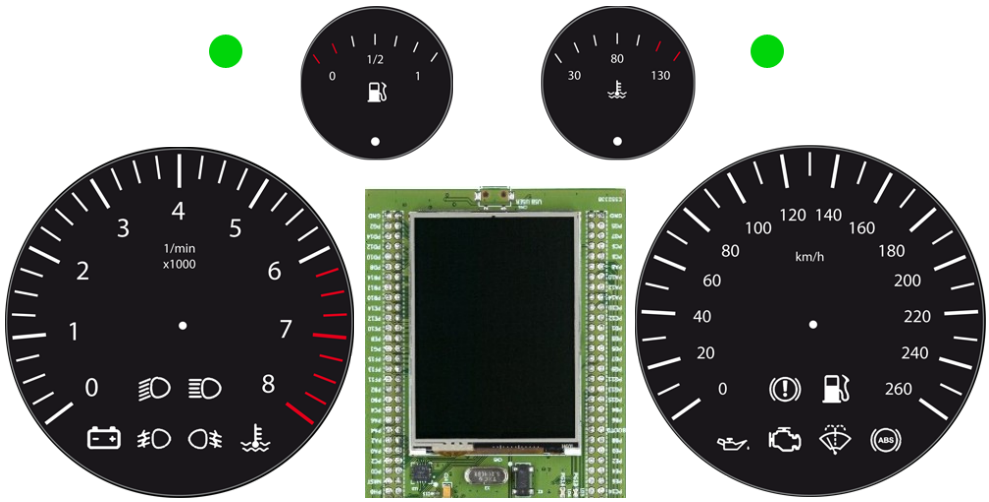
2.1 Koncept modelu přístrojové desky

Výsledná přístrojová deska a její rozložení vychází ze všeobecně zažitého konceptu, vizte obrázek 2.1.

Rychlost vozidla a rychlost motoru je zobrazována na dvou větších ukazatelích, o průměru deset centimetrů, umístěných symetricky podle vertikální osy desky. Osička obou ukazatelů prochází jejich středem a ručička se pohybuje v rozsahu 270 stupňů, kdy slepý úhel je umístěn ve spodu ukazatele.

Teplota chladící kapaliny a hladina paliva v nádrži je pak na dvou menších ukazatelích o průměru 4,2 centrimetru, umístěných také symetricky, nad většími ukazateli a blíže k ose. Osička ukazatelů zde není umístěna ve středu, nýbrž ve spodní části ukazatele a pohybuje se v rozsahu 70 stupňů.

Nalevo a napravo od menších ručičkových ukazatelů jsou umístěny LED indikátory levého a pravého směrového světla. Zbylých 12 světelných indikátorů je umístěno ve slepém úhlu, ve spodní části velkých ručičkových ukazatelů, šest v každém.



Obrázek 2.1: Výsledný návrh vizuální podoby realizované přístrojové desky.

2.2 Mechanické prvky

Pro instalaci ukazatelů je třeba zvolit vhodnou podpůrnou konstrukci. Na té by měl kromě indikátorů a podpůrné elektroniky související s touto bakalářskou prací být umístěn také výše zmíněný volant.

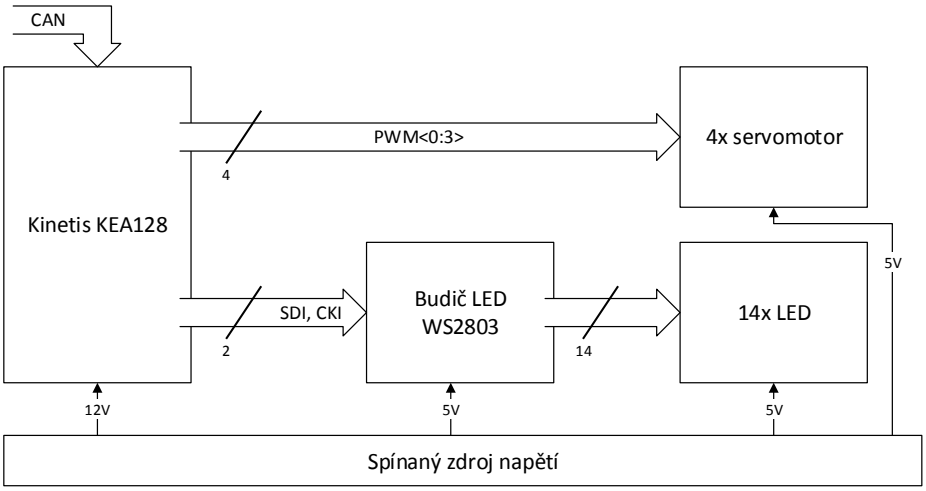
Jako základna celé konstrukce byla zvolena bílá nábytková deska z dřevotřísky. V její zadní části pak bude vztyčena deska z plexiskla, na níž budou realizovány všechny ručičkové a světelné ukazatele, přichycen modul s displayem palubního počítače a další podpůrné prvky a která tak bude sloužit jako hlavní konstrukční prvek přístrojové desky. Z toho vyplývá nutnost instalace opěrek, zajišťujících vhodný sklon.

Pro realizaci ručičkových ukazatelů byly zadáním práce určeny modelářské servomotory. Jejich přichycení k plexisklu je opět nutné řešit pomocí vhodných držáků. Problém nastává také v případě jejich rozsahu. Dva z ručičkových ukazatelů jsou v rozsahu 270 stupňů, zatímco dostupné servomotory jsou schopny se otočit pouze o 180 stupňů. Z toho vyplývá nutnost návrhu a realizace potřebných převodů. Je třeba také řešit přichycení osičky ručičkových ukazatelů k motorům.

Dalším mechanickým prvkem, který je nutno zajistit, jsou samotné ručičky ukazatelů.

Vzhledem k velkému množství součástek, které je nutno navrhnout a vytvořit, bylo zvoleno využít na zdejší fakultě dostupné laboratoře 3D tisku a navrhnout všechny potřebné součástky na míru jejich aplikaci. Na tomto místě považuji za vhodné znovu poděkovat Miru Hrončokovi za pomoc s tiskem navržených součástek.

V poslední řadě je třeba vyřešit montáž vývojové desky od společnosti ST (vizte kapitolu 2.3.6) a její připojení k budiči sběrnice CAN.



Obrázek 2.2: Blokové schéma navrhovaného hardwaru přístrojové desky.

2.3 Použitá elektronika a hardware

Realizace všech zmíněných ukazatelů vyžaduje kromě mechanické konstrukce také množství elektronických prvků. V této kapitole se pokusím shrnout dostupné technické vybavení a jeho možnosti.

2.3.1 LED diody

Pro realizaci kontrolek, zajišťujících předávání informací o vozidle a výstrah řidiči, byly zvoleny světelné indikátory. Historicky vzato jsou jedním z možných řešení například žárovky; vzhledem k úspoře energie, vyšší odolnosti proti poruše a snažší instalaci jsou však zvoleny LED diody.

Dioda je polovodič, dvojpól, sestávající se ze dvou materiálů a tvořící tzv. P-N přechod. Ten umožňuje tok proudu pouze jedním směrem. Existují různé druhy diod určené pro různé aplikace. Jedním z druhů, se kterým se běžný uživatel nejčastěji setká, je právě LED dioda, konstruovaná tak, že při průchodu proudu vyzařuje světlo v neviditelném nebo viditelném spektru [5].

Všechny světelné ukazatele jsou tvořeny nízkopříkonovými LED diodami červené, žluté, zelené a modré barvy. Menší diody je třeba napájet proudem 5mA, dvě větší diody použité pro směrová světla při uvedeném proudu svítí dostatečně, pro jejich úplné rozsvícení je však vhodné proud zvýšit alespoň na 10mA.

S rostoucím počtem ukazatelů také lineárně roste počet vodičů potřebných pro jejich řízení. Řešením tohoto problému se zabývám v kapitole 2.3.4.

2.3.2 Servomotory

Otáčení ručiček v ručičkových ukazatelích je úkolem servomotorů. Stejnosměrných servomotorů existuje celá řada, s výkonem od zlomků až po stovky koňských sil. Obvykle mají dvě oddělená vinutí, zvlášť na statoru a zvlášť na kotvě (objevují se i serva, která doplňují nebo nahrazují jedno z vinutí permanentním magnetem) a obsahují vlastní elektroniku, založenou na zpětné vazbě (obvykle realizované pomocí rezistoru – rotor se s natáčením chová jako potenciometr) a zajišťující řízení polohy [6].

Aktuální natočení osy servomotoru je určeno pulsně šířkovou modulací na jeho řídícím vstupu. Řídící elektronika servomotoru očekává každých 20 milisekund puls. Na základě délky jeho trvání je pak rotor natočen do odpovídající polohy [6]. Z výše uvedeného vyplývá nejkratší možná reakční doba ručičkových ukazatelů, která je dále ještě omezena rychlostí motoru.

Odběr proudu použitých servomotorů při pohybu z jednoho konce na druhý se pohybuje kolem 200 až 300mA.

2.3.3 Zdroj napětí

Při špičkovém zatížení přístrojové desky by nebyl zdroj na dostupné řídící desce (vizte kapitolu 2.3.5) dostačující [7]. Vzhledem k velkému množství světelných ukazatelů, odběru servomotorů a možné budoucí potřebě napájet další komponenty, jako je displej nebo celý palubní počítač, jsem se rozhodl zkonstruovat vlastní zdroj napětí.

Vstupní napětí bylo zvoleno 12V, především z důvodu snadné dostupnosti potřebných síťových napájecích adaptérů. Servomotory je však třeba napájet 5V, obdobně jako další součástky (vizte například následující kapitolu 2.3.4). Řešení pro snížení napětí na požadovanou úroveň se nabízí několik.

Prvním a nejjednodušším řešením je odporový dělič napětí. Výhodou je snadný návrh i realizace. Nevýhodou je pak vysoká tepelná ztráta a s ní související neúčinnost. Vzhledem k předpokládanému vysokému odběru ze zdroje byla tato varianta ihned zavržena.

Druhou variantu představovalo použití lineárního napěťového stabilizátoru. Jeho nasazení bylo též zavrženo z důvodu možných vysokých tepelných ztrát a nutnosti chlazení.

Jako nejvhodnější možnost vzhledem k účinnosti a možnému zatížení byl zvolen spínaný zdroj v konfiguraci step-down (nebo také buck), tedy snižující vstupní napětí. Jeho klíčovou komponentou je spínač, řízený zpětnou vazbou, produkující pulsně šířkovou modulaci s potřebnou střední hodnotou napětí. K němu připojená indukčnost akumuluje energii, když je spínač sepnutý, a uvolňuje energii do zátěže, když je rozepnutý. Dalšími důležitými prvky jsou vstupní a výstupní kondenzátory, zajišťující vyhlazení napětí, a rychlá schottkyho dioda [8].

Pro realizaci zdroje byl použit integrovaný obvod LM2676, od společnosti Texas Instruments, ve variantě s pevným pětivoltovým výstupem. Na jeho vstup lze připojit široký rozsah napětí, od 8V až po 40V, zatížit jej lze proudem až 3A a zajišťuje účinnost až 94% [8]. Vyžaduje připojení minimálního množství externích součástek, především výše zmíněných kondenzátorů, schottkyho diody a indukčnosti [8].

2.3.4 Sériově řízený budič LED

Na navrhované přístrojové desce se nachází 14 LED diod, sloužících jako indikátory, připojených celkem pomocí 28 vodičů.

Při jejich přímém připojení na mikrokontrolér (vizte kapitolu 2.3.5) je pro jejich řízení potřeba 14 pinů, tedy alespoň dva osmibitové porty. Vyvstává také konstrukční problém s připojením napájení a rezistorů potřebných k zajištění vhodného proudu diodou.

Z toho důvodu bylo pro LED diody zároveň rozhodnuto využít jiného řešení tak, aby řízení mikrokontrolérem probíhalo sériově a omezil se počet přímo připojených vodičů na minimum.

Prvním zvažovaným řešením bylo využití některého z dostupných I2C vstupně/výstupních expandérů. Počet vodičů připojených k mikrokontroléru za účelem řízení diod by se snížil na dva. Kromě samotného expandéru však zůstává potřeba připojit napájení a potřebné rezistory.

Jako vhodnější řešení byl zvolen obvod WS2803 od společnosti WorldSemi, sloužící jako zdroj konstantního proudu [9]. Poskytuje osmnáct kanálů, umožňujících připojit až osmnáct diod. Každý kanál má vlastní nastavení pulsně šířkové modulace a umožňuje tedy i stmívání připojených diod [9] – pro potřeby této aplikace je využíváno jen plně zhasnutého a plně rozsvíceného stavu. Nastavení jednotlivých kanálů probíhá pomocí rozhraní SPI. Obvod má jen datový a hodinový vstup a dále datový a hodinový výstup pro kaskádové zapojení dalších obvodů [9]. Mikrokontrolér tedy řídí LED diody pouze dvěma vodiči a zároveň odpadá nutnost připojení rezistorů.

Vzhledem k nutnosti napájet dvě z diod větším proudem a nevyužitým kanálům na budiči je připojení těchto LED zdvojeno.

2.3.5 Freescale Kinetis KEA128 StarterTRAK

K řízení výše zmíněných součástek a obvodů je zadáním určena vývojová deska Kinetis KEA128 StarterTRAK od společnosti Freescale, dnes již převzaté společností NXP [10].

Její hlavní komponentou je mikroprocesor Freescale Kinetis KEA128 postavený na bázi ARM Cortex-M0+ [11]. Ten v sobě integruje následující, pro tuto práci nejpodstatnější, komponenty:

- 32-bitové jádro ARM Cortex-M0+, implementující architekturu podle ARMv6, s jednotaktovou násobičkou, určené pro vestavné aplikace s nízkou spotřebou [11][12], spolu s řadičem přerušení NVIC [12],

- tři 16-bitové čítače/časovače FlexTimer s přímou podporou pro generování pulsně šířkové modulace [11],

- dvě osmibitová rozhraní SPI [11]

- a modul MSCAN, implementující CAN2.0A/B podle specifikace společnosti Bosch [11].

Na vývojové desce se nachází také budič sběrnice CAN [13] a deska je tak vhodná pro přímé připojení ke sběrnici bez dalších komponent (vyjma možných rezistorů sloužících k impedančnímu přizpůsobení).

Pro programování a debugging slouží na desce integrované USB rozhraní OpenSDA. To využívá sériového single-wire debug systému v procesoru ARM [13].

2.3.6 STM32F429ZI Discovery kit

Namísto samostatného displeje bylo po konzultaci s vedoucím práce rozhodnuto osadit práci vývojovou deskou od společnosti ST, obsahující mikroprocesor na bázi ARM Cortex-M4 a s integrovaným barevným LCD displayem s dotykovou vrstvou [14].

Procesor obsahuje vlastní řadič sběrnice CAN, moduly pro SPI a I2C komunikaci, DMA, LTDC řadič pro obsluhu displeje a velké množství dalších komponent. Narozdíl od Kinetisu (2.3.5) není na této vývojové desce integrován budič sběrnice CAN a je tak nutné využít externích součástek [14].

Z tohoto důvodu je třeba využít modul navržený Ing. Matějem Bartíkem pro výuku v předmětu MI-PVS, který obsahuje kromě budiče sběrnice SN65HVD233 také CAN řadič MCP2515 [15], komunikující s procesorem pomocí SPI [16].

Procesor včetně celého vývojového kitu je podporován projektem ARM mBed. Ten kromě online vývojového prostředí a kompilátoru poskytuje sadu ovladačů a funkcí pro práci s periferiemi a usnadňuje tak vývoj i přenositelnost kódu (vizte kapitolu 2.4.4).

Deska může v budoucnu sloužit jako palubní počítač pracující s multimédii, implementovat parkovacího asistenta nebo realizovat jakoukoli jinou funkci. Cílem této práce není implementace programového vybavení palubního počítače.

2.4 Použité softwarové prostředky

2.4.1 OpenSCAD

OpenSCAD je CAD software určený pro modelování 3D objektů a použitý při návrhu mechanických dílů. Na rozdíl od interaktivních modelovacích nástrojů, OpenSCAD zpracovává script psaný vlastním deskriptivním jazykem a umožňuje tak snadný parametrický návrh. Jeho výstupem je soubor STL, přímo použitelný ke slicování a následnému tisku na 3D tiskárně.

K dispozici je zdarma včetně zdrojových kódů, uvolněný pod licencí GPLv2 [17].

2.4.2 Eagle

EAGLE od společnosti CadSoft je nástroj sloužící k návrhu plošných spojů. Obsahuje tři moduly: editor elektrického schématu, editor plošného spoje (včetně kontroly návrhu) a autorouter (v této práci nepoužitý). Jeho výstupem jsou soubory ve formátu Gerber a Excellon, sloužící k výrobě spoje.

Ve verzi Freeware, obsahující omezení týkající se velikosti výsledného tištěného spoje, je pro nekomerční účely k dispozici zdarma [18].

2.4.3 Freescale CodeWarrior

CodeWarrior je integrované vývojové prostředí od společnosti Freescale, postavené na bázi Eclipse, podporující mj. používaný procesor řady Kinetis EA a debuggovací adaptér OpenSDA [19]. Pro projekt automaticky generuje makefile, flashovací scripty, kód pro počáteční inicializaci procesoru a poskytuje hlavičkové soubory implementující paměťové mapy podporovaných procesorů.

K dispozici je zkušební třicetidenní verze, po jejímž uplynují je ovšem základní funkcionalita prostředí (kompilace, sestavení, nahrání, debugging) zachována.

2.4.4 ARM mBed

mBed je online vývojové prostředí a platforma poskytující nástroje a rozhraní pro vývojáře pracující s podporovanými procesory ARM. Využitím knihoven implementujících ovladače hardware a vytvářejících hardwarovou abstrakci může být zjednodušen a urychlen proces vývoje, a zajištěna snadnější přenositelnost [20].

Nástoj je k dispozici zdarma po registraci na webových stránkách.

2.5 Komunikační protokol

Všechny jednotky, připojené ke stejné sběrnici CAN, musí komunikovat stejnou rychlostí. Ta byla po konzultaci s autorem související práce, Pavel Zajícem, stanovena na 125kbps.

Řízení přístrojové desky probíhá pomocí zpráv přijatých ze sběrnice CAN. Ty jsou zasílány příslušnými jednotkami, vizte kapitolu 1.2.2. Jejich formát je následující:

Pokud není určeno jinak a zpráva nese více datových bytů, první datový byte je nejvyšší byte, poslední datový byte je nejnižší byte přenášených dat. Pokud jsou popisovány jednotlivé bity, nejnižší bit je označován jako 0. bit.		
Název zprávy	**Identifikátor**	**Počet datových bytů**
Rychlost vozidla	0x40	2
Zpráva obsahuje rychlost vozidla jako celé číslo bez znaménka, v kilometrech za hodinu. Nejvyšší rychlost zobrazitelná přístrojovou deskou je 260 km/h.		
Otáčky motoru	0x48	2
Zpráva obsahuje otáčky motoru jako celé číslo bez znaménka, v otáčkách za minutu. Nejvyšší rychlost motoru zobrazitelná přístrojovou deskou je 8000 ot/m.		
Teplota chladící kapaliny	0x50	2
Zpráva obsahuje teplotu chladící kapaliny motoru jako celé číslo bez znaménka, ve stupních Celsia. Nejvyšší teplota zobrazitelná přístrojovou deskou je 130 stupňů.		
Stav nádrže	0x80	1
Zpráva obsahuje informaci o hladině paliva v nádrži jako celé číslo bez znaménka, v procentech. Přístrojová deska umí zobrazit hodnoty od 0 do 100. Hodnotu nižší než 15 navíc deska indikuje rozsvícením příslušného indikátoru.		

Stav světel	0x30	1
Zpráva nese informaci o aktuálním stavu světel. • 0. bit indikuje zapnutá potkávací světla • 1. bit indikuje zapnutá dálková světla • 2. bit indikuje zapnutá přední mlhová světla • 3. bit indikuje zapnutá zadní mlhová světla • 4. bit indikuje zapnuté levé směrové světlo • 5. bit indikuje zapnuté pravé směrové světlo • zbylé bity jsou rezervované Přístrojová deska indikuje zapnuté světlo rozsvícením příslušného indikátoru.		

Stav motoru	0x10	1
Zpráva nese informaci o aktuálním stavu motoru. • 0. bit indikuje obecnou výstrahu motoru • 1. bit indikuje výstrahu mazání motoru • 2. bit indikuje výstrahu chlazení motoru • zbylé bity jsou rezervované Přístrojová deska indikuje výstrahy rozsvícením příslušných indikátorů.		

Stav akumulátoru	0x28	1
Zpráva nese informaci o aktuálním stavu akumulátoru. • 0. bit indikuje závadu na akumulátoru • zbylé bity jsou rezervované Přístrojová deska indikuje závadu rozsvícením příslušného indikátoru.		

Stav brzdového systému	0x08	1
Zpráva nese informaci o aktuálním stavu brzd. • 0. bit indikuje výstrahu brzdového systému • zbylé bity jsou rezervované Přístrojová deska indikuje výstrahu rozsvícením příslušného indikátoru.		

Stav stěračů	0x18	1
Zpráva nese informaci o aktuálním stavu stěračů. • 0. bit indikuje zapnuté stěrače • zbylé bity jsou rezervované Přístrojová deska indikuje zapnutí stěračů rozsvícením příslušného indikátoru.		
Stav ABS	0x20	1
Zpráva nese informaci o aktuálním stavu ABS. • 0. bit indikuje závadu na ABS • zbylé bity jsou rezervované Přístrojová deska indikuje závadu rozsvícením příslušného indikátoru.		

Tabulka 2.1: Řízení přístrojové desky

Kromě výše uvedených zpráv, navržených v této bakalářské práce a významných pro řízení přístrojové desky, jsou dále definovány zprávy přenášející pozici volantu, pozici plynového pedálu, pozici brzdového pedálu a informaci o stisku tlačítka na volantu.

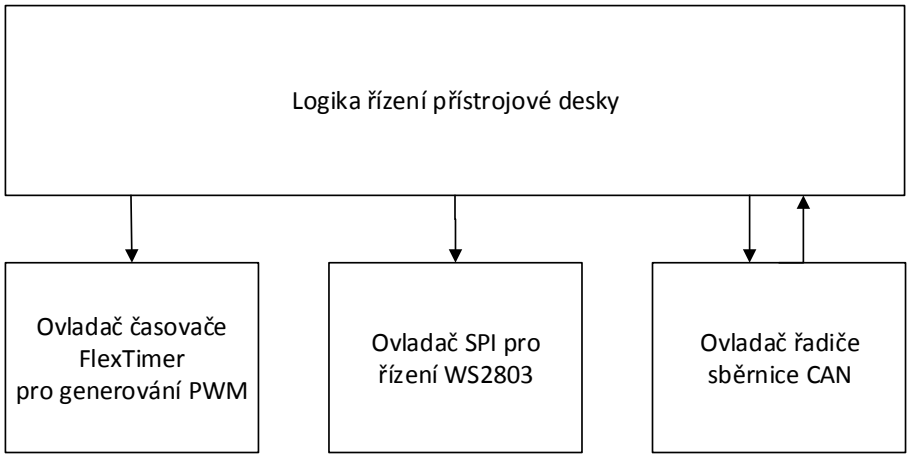
Identifikátory zpráv byly navrženy s ohledem na to, jak často se dané zprávy na sběrnici vysílají. Informace, které jsou vysílány nejčastěji a pravidelně, jako rychlost vozidla nebo otáčky motoru, mají nižší prioritu, než zprávy, které jsou zasílány sporadicky nebo zcela asynchronně (například stisknutí tlačítka nebo následná zpráva informující o novém stavu světel). Předchází se tím hladovění komunikujících jednotek.

2.6 Návrh programového vybavení

2.6.1 Freescale Kinetis KEA128

Řízení komponent přístrojové desky je úkolem mikroprocesoru Kinetis KEA128 (vizte kapitolu 2.3.5). Pro tento procesor je nutné navrhnout potřebné programové vybavení v jazyce C. Jeho hlavní úkoly jsou:

1. řízení servomotorů, tedy generování pulsně šířkové modulace (vizte kapitolu 2.3.2),
2. řízení světelných indikátorů pomocí sériové komunikace (vizte kapitolu 2.3.4),
3. komunikace na připojené sběrnici CAN (vizte kapitolu 1.3)
4. a implementace logiky přístrojové desky (reakce na přijané zprávy).



Obrázek 2.3: Blokové schéma programového vybavení Kinetis KEA128.

2.6.1.1 Komunikace na sběrnici CAN

Pro ovládání modulu MSCAN byl zvolen ovladač poskytnutý zdarma společností Freescale. Ten zajišťuje mj. [21]:

- obsluhu hardwarového modulu MSCAN, včetně obsluhy přerušení,
- práci až s 255 identifikátory nad až 32 buffery,
- správu priority zpráv určených k odeslání, včetně odebrání zprávy z výstupního bufferu a jejího nahrazení v případě zpracovávání důležitější zprávy (tato vlastnost je nezbytná pro zajištění časových požadavků definovaných standardem popsaným v kapitole 1.3),
- automatickou odpověď na zprávy typu RTR, ...

V případě přijetí zprávy ovladačem je nastaven flag u příslušného vstupního bufferu. Tyto buffery jsou neustále kontrolovány v hlavní smyčce programu a v případě přijetí nové zprávy jsou vykonány nezbytné akce, popsané v následujících kapitolách.

2.6.1.2 Pulsně šířková modulace

Generování pulsně šířkové modulace, která bude řídit servomotory, má na starosti modul čítače FlexTimer. Po jeho počáteční konfiguraci jsou výstupní signály generovány již bez součinnosti procesoru a ten se tak může zabývat jinou činností.

Změna střídy generovaného signálu probíhá po přijetí příslušné zprávy z CAN sběrnice a je realizována krátkým výpočtem a zápisem do konfiguračního registru čítače.

2.6.1.3 Sériová komunikace

Sériovou komunikaci pro řízení LED diod obstarává modul rozhraní SPI. Po inicializaci modulu probíhá odesílání dat samočinně a při jeho dokončení je generováno přerušení, které je obslouženo zapsáním dalších dat, čekajících k odeslání, do příslušného bufferu.

Při přijetí zprávy ze sběrnice CAN může být upraven stav indikátorů v místní paměti. Aktualizovat stav kontrolek je nutné jednou za 100 až 200ms, vzhledem k reakční době řidiče [2] (vizte kapitolu 1.2.1). Přenos dat je tedy spouštěn vždy na základě periodického přerušení od časovače.

2.6.1.4 Struktura programu

Program přímo používá 7 komponent:

- NVIC – řadič přerušení
- SIM – modul zajišťující multiplexování signálů na pinech
- GPIO – modul obstarávající přímý programový přístup k pinům
- MSCAN – řadič sběrnice CAN
- FlexTimer – časovač pro generování PWM
- SPI – modul pro komunikaci
- PIT – modul pro generování periodických přerušení

Obsluhu zmíněného hardwaru zajišťuje několik softwarových knihoven:

- ovladač MSCAN od společnosti Freescale,
- vlastní ovladač pro generování PWM, obsluhující čítač FlexTimer,
- vlastní ovladač pro WS2803, obsluhující modul SPI
- a kód implementující komunikační protokol a funkční logiku.

Poslední zmíněná knihovna pomocí výše zmíněných ovladačů čte příchozí zprávy ze sběrnice CAN a na jejich základě provádí příslušné akce: rozsvícení nebo zhasnutí LED diod a přenastavení střídy PWM.

2.6.2 STM32F429ZI Discovery Kit

Palubní počítač realizovaný deskou STM32F429ZI Discovery Kit (vizte kapitolu 2.3.6), podobně Kinetis, bude komunikovat samostatně přes připojenou sběrnici CAN. Cílem této práce není návrh a implementace komplexního programového vybavení palubního počítače, spíše jeho mechanické zakomponování do celku přístrojové desky a připojení ke sběrnici.

Výstupem této části, po dohodě s vedoucím práce, bude tedy na přístrojové desce umístěný a namontovaný palubní počítač spolu s minimální demo aplikací, demonstrující funkčnost zapojení.

2.6.2.1 Komunikace po sběrnici CAN

Pro komunikaci po sběrnici CAN lze využít dvou řešení:

1. v procesoru vestavěného CAN řadiče, jehož výstup je dále nutné připojit na budič sběrnice, a pro který jsou k dispozici ovladače přímo od společnosti ST [14]
2. řadiče MCP2515, který je na připojené podpůrné desce spolu s budičem [15], a který komunikuje s procesorem pomocí SPI [16].

Vzhledem k mnoha neúspěšným pokusům s interním řadičem je pro demoaplikaci zvolen externí řadič MCP2515. Pro ten je mezi knihovnami projektu mBed také dostupný jednoduchý ovladač.

2.6.2.2 Displej a dotyková vrstva

Barevný displej, který se na desce vyskytuje, využívá řadič ili9341 a dotyková vrstva pak řadič stmpe811. Pro obě komponenty jsou v mBed dostupné ovladače, od nízkoúrovňových až po jednoduše použitelné API [22].

2.6.2.3 Struktura programu

Program využívá možností prostředí mBed, ve kterém je napsán, kdykoli je to možné. Pro ovládání displeje a dotykové vrstvy, stejně jako připojeného CAN kontroléru, se používají veřejně dostupné knihovny.

V hlavní smyčce program kontroluje příchozí zprávy ze sběrnice CAN a jejich obsah pak vypisuje na obrazovku. Ve stejné smyčce lze detekovat dotyk a realizovat pomocí něj například softwarová tlačítka.

Kapitola 3

Realizace

3.1 Plastové mechanické díly

Jak bylo zmíněno již v kapitole 2.2, pro realizaci je nezbytné využití mnoha pomocných mechanických prvků. Ty jsou tvořeny plastovými díly, navrženými pomocí CAD nástroje (vizte 2.4.1) a následně vytištěnými na 3D tiskárně v laboratoři na FIT ČVUT.

Pro tisk byl použitý materiál ABS v přírodní barvě (tedy bílé až světle šedivé). Ten se vyznačuje vysokou odolností a pevností, stejně jako vcelku vysokou odolností proti teplu (tedy je například možné do něho vrtat).

3.1.1 Opěrky

Prvním nezbytným dílem jsou opěrky, podepírající plexisklo, které tvoří přístrojovou desku. Ty musí být dostatečně vysoké a široké pro zajištění stability.

Přístrojová deska by měla být sklopena tak, aby umožňovala operátorovi pohodlné odčítání informací (také vzhledem k volantu umístěnému před ní). Experimentálně bylo jako vhodný úhel zvoleno 70 stupňů.

Výsledná opěrka je pět centimetrů široká a tři centimetry vysoká. K dřevotřískové desce, tvořící základ konstrukce, je uchycena dvěma šrouby M4 a k plexisklové desce jedním šroubem M4.

Na modelu jsou použity celkem čtyři rovnoměrně umístěné opěrky.

3.1.2 Úchyty pro servomotory

Dalšími důležitými díly jsou úchyty pro servomotory. Situace je jednodušší v případě malých ukazatelů, kde je rozsah servomotorů dostačující a není třeba realizovat převod.

K uchycení serva k plexisklu slouží malá obdelníková konstrukce s otvorem pro motor. Ten je na ní připevněn pomocí dvou šroubů M2 a samotná konstrukce je následně pomocí čtyř sloupků a šroubů M3 přichycena na plexisklo.

Obrázek 3.1: Realizované úchyty a převody pro ručičkové ukazatele.

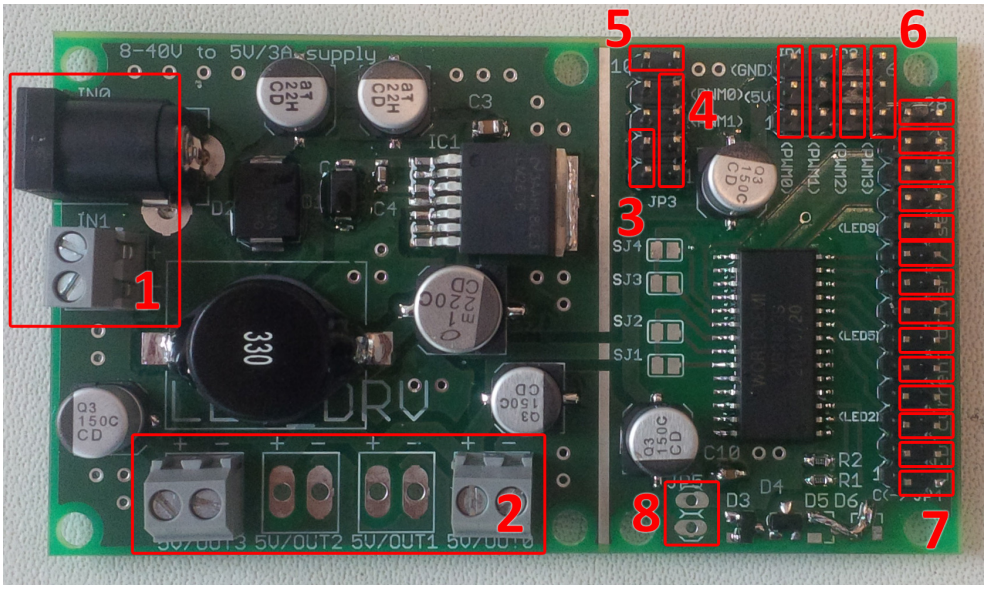
K servomotoru je dále nutné uchytit osičku, otáčející ručičkou v ukazateli. Vzhledem k plánované rozebiratelnosti je spoj realizován opět dvěma plastovými díly. Větší prstenec je přichycen pomocí dvou M2 šroubů k rotoru servomotoru. Osička je vteřinovým lepidlem přichycena k dalšímu dílu, který je do prstence vložen a k němu přišroubován opět dvěma šrouby M2.

3.1.3 Převody pro servomotory

Složitější situace nastává v případě velkých ručičkových ukazatelů. Zde je kromě uchycení servomotoru nutné dále realizovat převod, který za 180 stupňů, o které se je schopen otočit servomotor, otočí ručičku ukazatele o 270 stupňů.

Uchycení servomotoru je opět řešeno pomocí obdelníkové konstrukce, ke které je přichycen dvěma šrouby M2. Oproti malým úchytům, zmíněným o kapitolu výše, má tato konstrukce ještě druhý, centimetr a půl vzdálený protikus, přichycený pomocí čtyř kovových sloupků.

Mezi těmito dvěma díly je realizován zmíněný převod, v poměru 2:3. Větší kolo měří 21 milimetrů a je přichyceno k servomotoru. Menší kolo měří 14 milimetrů a je vteřinovým lepidlem uchyceno k osičce ukazatele. Obě kola jsou svázána vlascem, protaženým otvory v kolech k tomu určenými.



Obrázek 3.2: Plošný spoj: 1. Vstupní napájení, 2. Výstupy 5V, 3. Vstupy pro programování LED driveru, 4. Vstupy PWM, 5. Země, 6. Čtyři výstupy pro servomotory, 7. Čtrnáct výstupů pro LED diody, 8. Napájení 5V

3.1.4 Ručičky

Posledním tisknutým dílem jsou samotné ručičky ukazatelů. Velká ručička, použitá ve dvou větších ukazatelích, měří 46 milimetrů od osy otáčení. Menší ručička pak měří 28 milimetrů.

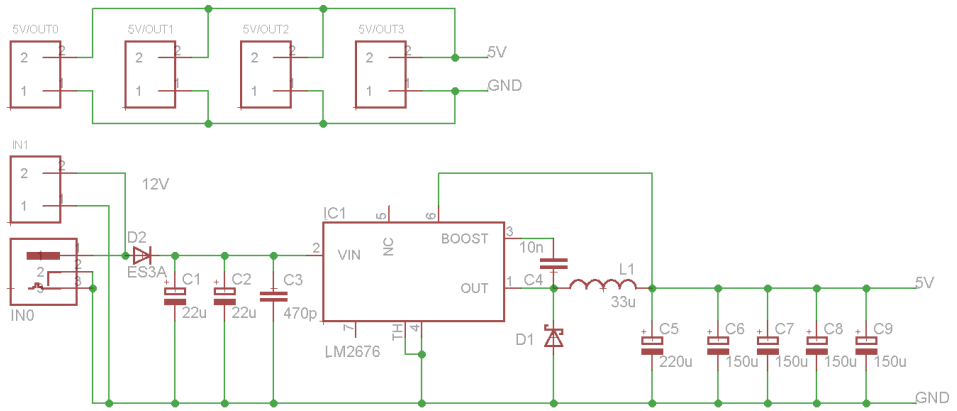
Jejich uchycení k osičce, opět vzhledem k nutnosti rozebiratelnosti spoje, je realizováno pomocí gumy, určené ke gumování tužky, nařezané na malé kousky, uprostřed kterých byla vytvořena díra. Ty jsou vloženy do otvorů v ručičkách a jsou silou natlačeny na osičky.

3.2 Plošný spoj

Nutnost realizace a zapojení dalších komponent vedla k návrhu a výrobě plošného spoje. Ten je rozdělen na dvě části, schopné samostatné funkce, pro možnost využití v dalších projektech.

Levá část plošného spoje realizuje spínaný zdroj. Na jeho vstup lze připojit stejnosměrné napětí v rozsahu 8 až 40V a na výstupu je 5V. Zatížit lze proudem až 3A.

Na pravé straně je pak sériově řízený 18-kanálový zdroj konstantního proudu pro připojení 14 LED diod, kde dvě z nich mohou mít až trojnásobný proudový odběr, než ostatní. Dále je zde realizováno připojení pulsně šířkové modulace, spolu s napájením, na vstup servomotorů.



Obrázek 3.3: Schéma levé části plošného spoje, realizující spínaný zdroj napětí.

Kromě napájecího konektoru, svorek a pinových lišt jsou všechny použité součástky určené pro povrchovou montáž (SMD).

Kompletní seznam všech použitých součástek lze nalézt v příloze.

3.2.1 Zdroj napětí

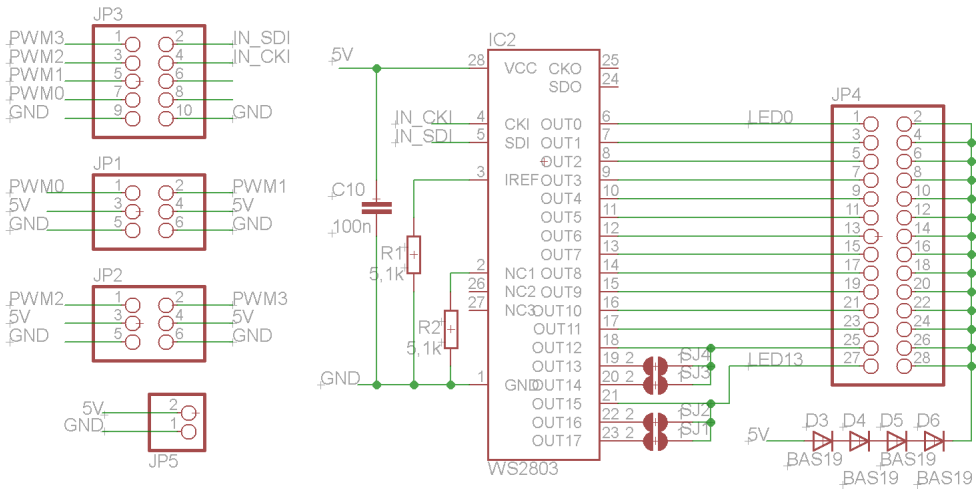
Základem zdroje napětí (vizte obrázek 3.3) realizovaného na levé straně plošného spoje je integrovaný obvod LM2676-5.0. Jeho úkolem je spínání vstupního napětí, podle zpětné vazby z výstupu, za účelem stabilizace výstupního napětí [8].

Vstupní napětí 8-40V je připojeno na konektor IN0 nebo svorku IN1. Sériově zapojená dioda D2 slouží jako ochrana proti přepólování a je dimenzována na patřičný proud (3A).

Rychlé změny v toku proudu na vstupu, způsobené spínáním, představují vysokou zátěž. Elektrolytické kondenzátory C1 a C2 proto zajišťují dostatek proudu a vyhlazení vstupního napětí. Pro redukci rušení je u vstupu integrovaného obvodu ještě blokovací keramický kondenzátor C3. Další keramický kondenzátor C4 slouží k vybuzení spínače uvnitř integrovaného obvodu [8].

Velmi podstatnými součástkami jsou Schottkyho dioda D1, indukčnost L1 a elektrolytický kondenzátor C5. Když integrovaný obvod sepne spínač, začne protékat cívkou L1 proud do zátěže. Zároveň se na indukčnosti začne vytvářet napětí. Cívka se po rozepnutí spínače snaží zachovat velikost a směr protékajícího proudu (Lenzův zákon), v důsledku čehož změní polaritu napětí. Proud zátěží začne klesat a uzavírá se přes diodu D1. Kondenzátor C5 slouží k vyhlazení výstupního napětí, jehož střední hodnota je závislá na střídě spínaného vstupu, regulované výše zmíněným integrovaným obvodem [8].

Kondenzátory C6, C7, C8 a C9 jsou rozmístěny po celém plošném spoji a slouží k udržení napětí při nárazovém zatížení a jeho vyhlazení.



Obrázek 3.4: Schéma pravé části plošného spoje se zdrojem konstantního proudu pro LED diody.

Výstupní napětí 5V je vyvedeno do pravé části plošného spoje a na svorky OUT0 až OUT3. Maximální zatížení zdroje je 3A, což je více než dostačující pro napájení motorů, LED diod i použitých mikroprocesorů

3.2.2 Budič LED

Nejvýznamnějším prvkem pravé části plošného spoje (vizte obrázek 3.4) je integrovaný obvod WS2803. Ten na pinech OUT0 až OUT17 omezuje protékající proud na nastavenou úroveň a zároveň jej spíná pomocí pulsně šířkové modulace. Díky tomu také umožňuje nastavení jasu připojených LED diod [9].

Nastavení střídy na jednotlivých kanálech probíhá pomocí sériové komunikace na vstupních pinech CKI a SDI. Po alespoň 500us klidové úrovně (tj. logická nula) na hodinovém signálu očekává obvod celkem 144 náběžných hodinových hran, při kterých vzorkuje data na datovém signálu do posuvného registru. Celkem tedy načte 18 bajtů, kde každý bajt určuje střídu PWM (tedy jas) diody na příslušném kanálu. Proud kanálem je určen odporem R1 [9].

Obvod na sobě, vzhledem k přijatelným tepelným ztrátám, předpokládá úbytek napětí přibližně 1V. Z toho důvodu jsou před LED diodami, které jsou připojeny na pinovou lištu J4, zapojeny ještě čtyři usměrňovací diody. Jejich úkolem je snížit napětí ze vstupních 5V na přijatelnou úroveň. V reálné aplikaci je třeba zapojit diody jen dvě, zbylé dvě je třeba nahradit vodičem.

Poslední dvě LED diody, LED12 a LED13, jsou vždy připojeny ke třem kanálům budiče, ke dvěma z nich přes pájecí spoj. Jejich propájením lze zvýšit proud LED diodou až na trojnásobek. **Tuto skutečnost je třeba uvážit také při programování budiče.**

Pinové lišty JP1 a JP2 slouží k připojení servomotorů. Je na ně přivedeno napájení a pulsně šířková modulace.

Pro připojení k řídícímu mikrokontroleru slouží pinová lišta JP3 (vizte obrázek 3.4):

- na pinech 2 a 4 jsou vyvedeny programovací vstupy budiče LED diod,
- piny 1, 3, 5 a 7 slouží jako vstup pulsně šířkové modulace pro řízení připojených servomotorů
- na pinech 9 a 10 je pak přivedena zem.

3.3 Zapojení další elektroniky

Kromě servomotorů a realizovaného plošného spoje je na přístrojové desce nutné umístit další elektroniku.

3.3.1 LED diody

Světelné ukazatele jsou realizovány LED diodami. Pro ty jsou v plexiskle vyvrtány díry, ve kterých jsou diody umístěny. Na jejich vývodech jsou napájeny kabely, z druhé strany zakončené konektory na pinovou lištu.

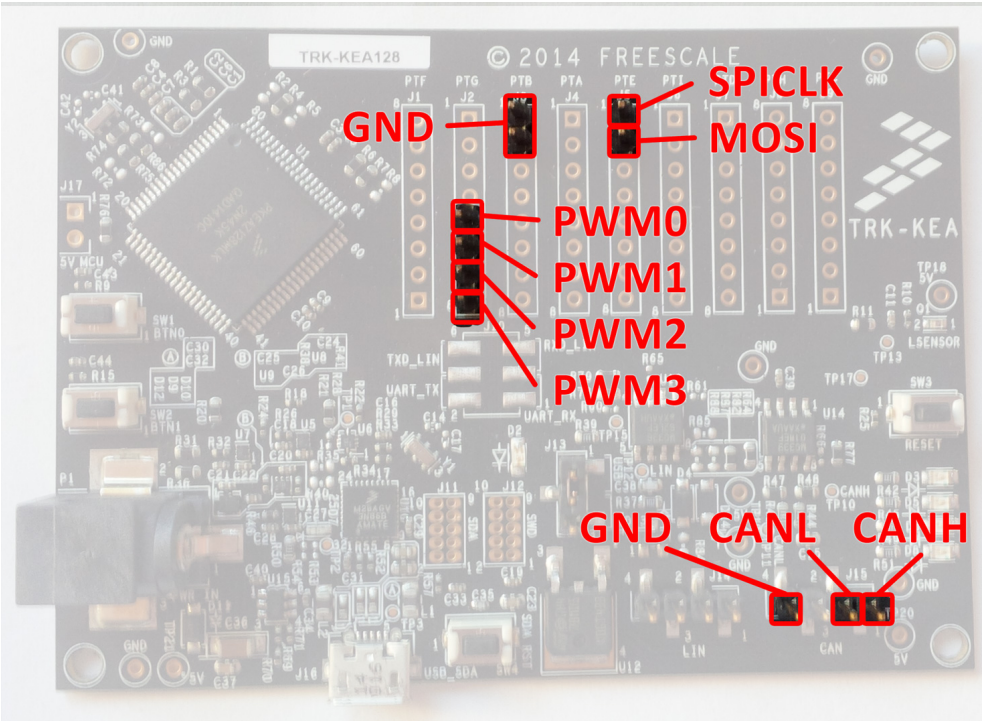
Ty jsou příslušně zapojeny na výše zmíněném plošném spoji, čímž jsou jednoznačně identifikovány. Spojení čísla diody s její funkcí je realizováno v řídicím softwaru.

3.3.2 Řídicí deska TRK-KEA128

Z desky KEA128 vede na plošný spoj celkem osm vodičů:

- dva zemnící vodiče, připojené na jumperu J3, pozicích 1 a 2,
- čtyři vodiče s pulsně šířkovou modulací pro řízení servomotorů:
  1. PWM0 na jumperu J2, pozici 5, připojené na pin PTG4,
  2. PWM1 na jumperu J2, pozici 6, připojené na pin PTG5,
  3. PWM2 na jumperu J2, pozici 7, připojené na pin PTG6,
  4. PWM3 na jumperu J2, pozici 8, připojené na pin PTG7,
- dva vodiče se sériovou komunikací pro budič LED:
  1. hodinový signál na jumperu J5, pozici 1, připojené na pin PTE0,
  2. datový signál na jumperu J5, pozici 2, připojené na pin PTE1.

Vzhledem k zaměření práce jako učební pomůcky je vývojová deska umístěna na přístrojové desce tak, aby byl snadno dostupný microUSB port debuggeru a uživatelská tlačítka, především tlačítko pro reset.



Obrázek 3.5: Připojené vodiče na Kinetis KEA128.

3.3.3 Palubní počítač

Palubní počítač je připojen ke sběrnici CAN a tvoří samostatnou komponentu. Využitá vývojová deska obsahuje množství příhodně vyvedených pinů, kterých je s úspěchem využito při jejím uchycení – jsou zasunuty do pinových konektorů umístěných na univerzálním plošném spoji, který je přišroubován k plexisklu. Na univerzálním pološném spoji se kromě zmíněného palubního počítače nachází ještě modul s řadičem a budičem sběrnice CAN. Připojení tohoto modulu k desce STM32F429ZI je následující:

- MOSI na pin PE6,
- MISO na pin PE5,
- CLK na pin PE2,
- SS# na pin PE3,
- RESET# na pin PF9,
- a INT# na pin PF7.

Modul s řadičem CAN má vyvedeny kromě výše zmíněných, zapojených pinů, dále ještě piny RX a TX, pro přímé připojení k budiči v případě využití jiného řadiče. Vzhledem k neúspěšnému pokusu o zprovoznění integrovaného CAN řadiče v procesoru STM24F429ZI nejsou tyto vývody k palubnímu počítači prozatím připojeny.

3.4 Programové vybavení KEA128

Program pro Kinetis KEA128 je psaný v jazyce C a pro jeho vývoj jsem použil prostředí CodeWarrior (vizte kapitolu 2.4.3). To poskytuje kromě kompileru, programátoru a debuggeru také hlavičkové soubory s popsanou paměťovou mapou použitého procesoru.

Je třeba implementovat ovladače pro časovač, který generuje pulsně šířkovou modulaci pro ovládání servomotorů. Důležitý je také ovladač SPI, který zajišťuje programování budiče LED. V neposlední řadě je potřeba komunikovat po sběrnici CAN – obsluhu řadiče MSCAN zajišťuje dostupný ovladač od společnosti Freescale.

Kromě zmíněných hardwarových modulů je nutné pracovat s některými dalšími komponentami, podporujícími jejich funkci.

3.4.1 Řadič přerušení NVIC

Všechny externí signály přerušení jsou připojeny k řadiči NVIC, přítomnému v procesoru Cortex-M0+. K jeho řízení slouží čtyři registry [12]. Zápis jedničky na konkrétní bit způsobí změnu nastavení připojeného přerušení. Do těchto registrů je zapisováno v inicializačních funkcích dále popisovaných ovladačů.

- NVIC_ISER – Interrupt Set-Enable Register, sloužící k povolení přerušení.
- NVIC_ICER – Interrupt Clear-Enable Register, sloužící k zakázání přerušení.
- NVIC_ISPR – Interrupt Set-Pending Register, sloužící k vyvolání přerušení.
- NVIC_ICPR – Interrupt Clear-Pending Register, sloužící ke zrušení čekajících přerušení.

Kromě těchto čtyř registrů obsahuje NVIC ještě osm registrů NVIC_IPR0-NVIC_IPR7, umožňujících nastavení priority jednotlivým přerušením.

Významná přerušení jsou popsána v kapitolách, věnujících se implementaci příslušných ovladačů. Zapojení všech přerušení je uvedeno v referenčním manuálu KEA128 [11].

3.4.2 System Integration Module

Úkolem modulu SIM je řízení toku signálů uvnitř procesoru. Pro tuto aplikaci jsou významné především tři registry. Ty jsou opět nastavovány v inicializačních funkcích ovladačů, spolu s inicializací dalších komponent.

- SIM_PINSEL0 a SIM_PINSEL1 – Pin Selection Register, sloužící k výběru pinu, připojeného k určité funkci. Každá funkce (jako třeba kanál časovače nebo výstup SPI) může být nastavena na několik pinů procesoru, pro zajištění určité míry flexibility. Je-li komponenta, která je namapována na daný pin, neaktivní, má pin funkci GPIO.
- SIM_SCGC – System Clock Gating Control Register, povolující hodinový vstup jednotlivým komponentám procesoru.

Významné bity těchto registrů jsou popsány v následujících kapitolách. Popis všech bitů lze opět nalézt v referenčním manuálu KEA128 [11].

3.4.3 PWM

Knihovna PWM, tvořená soubory pwm.c a pwm.h, slouží k obsluze časovače FlexTimer za účelem generování PWM. Pro svou funkci vyžaduje pouze platnou paměťovou mapu, popisující umístění registrů a jejich jednotlivé bity.

Využívá modulu čítače/časovače FTM2 a výstupní signál generuje na čtyřech jeho kanálech, 2 až 5.

3.4.3.1 Konfigurace knihovny

Pro nastavení ovladače slouží konstanty definované v hlavičkovém souboru pwm.h:

- PWM_PERIOD slouží k nastavení periody pulsně šířkové modulace. Definovaná hodnota je nastavena jako maximální hodnota 16-bitového čítače před přetečením. Ten se inkrementuje každých 6,4 mikrosekundy, vizte implementaci ovladače dále.
- PWM_DEFAULT_PULSE_WIDTH slouží k nastavení výchozí hodnoty střídy generovaného signálu. Ta je nastavena při inicializaci časovače.

3.4.3.2 Dostupné rozhraní

- PWM_Init() slouží k inicializaci modulu časovače FTM2 na režim generování stranově zarovnané PWM, a nastavení SIM.
- PWM_Start() slouží ke spuštění časovače.

- PWM_Stop() slouží k zastavení časovače.
- PWM_Set_Pulse_Width(UINT8 ch, UINT8 width) nastaví šířku pulsu, tedy střídu, pulsně-šířkové modulace na příslušném kanálu. Pro parametr ch lze využít předdefinovaných konstant PWM0 až PWM3. Parametr width by měl být menší, než PWM_PERIOD.

3.4.3.3 Implementace [11]

PWM_Init Úkolem inicializační funkce je nastavení všech registrů nutných ke správné funkci časovače tak, aby generoval pulsně šířkovou modulaci.

1. Nejprve je třeba zapnout hodiny pro FTM2. To se provede zápisem jedničky na 6. bit registru SIM_SCGC.
2. Dále je třeba nastavit výstupy kanálů na příslušné piny. Kanály 2 a 3 se nastaví na piny PTG4 a PTG5 zápisem '10' na bity 4, 5 a 6, 7 registru SIM_PINSEL1. Kanály 4 a 5 se na PTG6 a PTG7 nastaví zápisem jedničky na 8. a 9. bit téhož registru.

   Následuje konfigurace samotného modulu FTM2.
3. Nastavení předděličky pro vstup čítače probíhá pomocí registru FTM2_SC. Zapsáním '111' na nejnižší pozice proběhně výběr předděličky 128. Vstupní takt předděličky je 20MHz, k inkrementaci čítače tedy dojde každých 6,4 mikrosekund.
4. Počáteční hodnota čítače je pomocí registru FTM2_CNTIN nastavena na nulu, a poslední hodnota čítače, než dojde k nastavení příznaku přetečení, je nastavena pomocí registru FTM2_MOD na hodnotu 3124. Výsledná perioda generovaného signálu se bude rovnat $(3124 + 1) * 6.4us = 20ms$.
5. Pomocí registru FTM2_CnSC (kde n je číslo kanálu) se provádí nastavení speciálních funkcí časovače. Vynulováním bitů DECAPEN, COMBINE a CPWMS a nastavením bitu MSB dojde k výběru režimu generování hranově zarovnané PWM.

   Nastavením bitu ELSB na jedničku a bitu ELSA na nulu se nastaví konfigurace PWM taková, kdy výstup kanálu je vynulován při shodě porovnávacího registru s hodnotou čítače.
6. Tato hodnota pro porovnání se nastaví zápisem do registru FTM2_CnV.
7. Na závěr inicializace je nutné zapsat do registru FTM2_CNT, tj. registru obsahujícího aktuální hodnotu čítače, libovolnou hodnotu. Tím dojde k resetování čítače a načtení nových konfiguračních hodnot.

PWM_Start Spuštění čítače se provede výběrem zdroje hodin. K tomu slouží bity 4-3 registru FTM2_SC. Zápisem '01' dojde k výběru systémových hodin.

PWM_Stop Zastavení čítače probíhá opět zápisem na 4-3 bit registru FTM2_SC. Zápisem '00' dojde k vypnutí čítače.

PWM_Set_Pulse_Width Šířka pulsu je určena hodnotou v registru FTM2_CnV, která je porovnávána s aktuálním stavem čítače. Pokud je předané číslo kanálu 2, 3, 4 nebo 5, dojde k zápisu předané šířky pulsu do příslušného registru.

3.4.4 LEDSPI

Knihovna LEDSPI, tvořená soubory ledspi.c a ledspi.h, slouží jako ovladač pro připojený obvod WS2803.

Využívá modulu SPI0 a časovače FTM1 pro zajištění časového zpoždění vysílání (pro zajištění časování požadového obvodem WS2803 i za předpokladu opakované žádosti o vysílání dat nadřazenou knihovnou).

3.4.4.1 Dostupné rozhraní

- LEDSPI_Init() slouží k nastavení modulů SPI0, FTM1, SIM a NVIC.
- LEDSPI_LoadData(UINT8 data[18]) slouží k nahrání dat pro odeslání do paměti ovladače. Vzhledem k tomu, že obvod WS2803 očekává během jednoho komunikačního cyklu 18 bajtů dat, je velikost dat 18 bajtů.
- LEDSPI_StartTX() spustí odesílání dat uložených v paměti ovladače, se zpožděním cca 600 mikrosekund od volání funkce. Další volání funkce dříve, než je dokončen předchozí přenos, skončí neúspěchem.

3.4.4.2 Implementace [11]

LEDSPI_Init Úkolem této funkce je správná inicializace modulu SPI0 a čítače FTM1.

1. Zapnutí hodin pro SPI0 se provede zápisem jedničky na 18. bit registru SIM_SCGC.
2. Nastavení výstupů SCK na PTE0 a MOSI na PTE1 se provede zápisem jedničky na 6. bit registru SIM_PINSEL0.
3. Dále je třeba nastavit konfigurační registr SPI0_C1. Nastavením jedničky na 4. bit se modul nastaví do režimu Master. Všechny ostatní bity jsou vhodně nastavené již po resetu: polarita hodin (neaktivní v nule), náběžná hrana uprostřed datového cyklu.

4. Rychlost komunikace je upravena pomocí dvou předděliček, jejichž vstupem jsou systémové hodiny, a nastavených v registru SPI0_BR. Zápisem '100' na bity 4-6 dojde k výběru předděličky /5, zápisem '100' do spodních čtyř bitů k výběru předděličky /32. Výsledná komunikační frekvence je $(20MHz/5)/32 = 125kHz$.

5. Nastavení časovače FTM1 probíhá analogicky, jako v případě knihovny PWM. Využívá se 0. kanál pro generování přerušení po uplynutí nezbytné doby. Jediným podstatným rozdílem je nastavení registru nastavujícího funkci časovače FTM1_C0SC: nastavením bitu MSA dojde k výběru funkce Output Compare.

6. Na závěr inicializace je nutné povolit přerušení v řadiči NVIC. Nejprve jsou zrušena všechna čekající přerušení zápisem do registru ICPR a následně jsou přerušení povolena pomocí registru ISER. Číslo vektoru přerušení SPI0 je 26, FTM1 pak 34. Číslo přerušení, a také odpovídajícího bitu v registrech NVIC, je o 16 menší.

LEDSPI_StartTX Funkce spustí přenos dat uložených v lokálním poli knihovny.

1. Nejprve je nastavena proměnná indikující, že zbývá 18 bajtů k odeslání. Následně je vynulován časovač, povoleno přerušení na kanále 0 a časovač je spuštěn. Díky tomuto přístupu není také nutno řešit vstup do kritické sekce: i pokud je funkce spuštěna vícekrát, pokračování přenosu proběhne až po uplynutí doby inicializačního cyklu, definovaného budičem, který musí trvat alespoň 500us.

2. V obsluze přerušení časovače je povoleno přerušení modulu SPI0 zápisem do registru SPI0_C1. To je zpravidla ihned vyvoláno z důvodu prázdného bufferu, určeného k odeslání dat.

3. Obsluha přerušení SPI0 zkontroluje, zda je prázdný výstupní buffer. Pokud ano, zapíše do něj data určená k odeslání a dekrementuje řídící proměnnou. V případě, že dochází k odesílání posledního bajtu dat, je zároveň další přerušení zakázáno.

LEDSPI_LoadData Tato funkce zkopíruje 18 bajtů z předaného ukazatele do svého lokálního pole. Následně je možné data odeslat pomocí funkce LEDSPI_StartTX.

3.4.5 MSCAN

Knihovna MSCAN, tvořená soubory mscan.c a mscan.h, slouží k inicializaci ovladače modulu MSCAN. Pro svou funkci vyžaduje platnou paměťovou mapu procesoru a zmíněný ovladač MSCAN od společnosti Freescale.

3.4.5.1 Konfigurace ovladače od Freescale [21]

Konfiguraci ovladače je třeba provést před překladem. K tomu slouží soubory msCANcfg.h a msCANID.h, které obsahují definice mnoha maker:

- no_of_ID_CANx, počet Message Object, se kterými ovladač dále pracuje. Jeden Message Object reprezentuje zprávu s právě jedním identifikátorem. Všechny ovladačem přijímané identifikátory musí být definovány v souboru msCANID.h.
- no_of_mb_CANx, počet alokovaných bufferů. Buffer ovladače může být již za běhu asociován s konkrétním Message Objectem a nastaven jako přijímací nebo odesílací.

  Výchozí hodnota obou zmíněných maker, 16, je pro potřeby přístrojové desky dostačující.
- K nastavení komunikační rychlosti slouží následující makra:

  1. PRESCALER_CANx, určující předděličku, jejímž vstupem jsou systémové hodiny a výstupem signál o periodě jedné časové jednotky (time quantum),
  2. TIME_SEG1_CANx, určující počet časových jednotek v první fázi vzorkovací periody,
  3. TIME_SEG2_CANx, určující počet časových jednotek v druhé fázi vzorkovací periody,
  4. RJW_CANx, určující maximální možný počet časových jednotek, o který mohou být jednotlivé fáze prodlouženy nebo zkráceny za účelem resynchronizace se sběrnicí.

  Výsledná doba trvání jednoho bitu na sběrnici CAN je

  $$(1 + TIME_SEG1_CANx + TIME_SEG2_CANx) * timequantum.$$

  V případě této konkrétní aplikace je rychlost nastavena na 125kbps:

  $$timequantum = 1/(20MHz/8) = 400ns,$$
  $$(1 + 13 + 6) * 400ns = 8us = 1/125kHz.$$
- SAMPLEX3_CANx, nastavený na TRUE, čímž je zajištěno trojí vzorkování bitu namísto jednoho.

Soubory obsahují množství dalších nastavení, jejichž kompletní popis lze nalézt v manuálu k ovladači [21].

Ovladač umožňuje práci s více CAN řadiči. Procesor integrovaný na použité desce TRK-KEA128 obsahuje pouze jediný řadič, CAN0.

3.4.5.2 Rozhraní knihovny

- MSCAN_Init() nastaví moduly SIM a NVIC.
- MSCAN_Start() spustí inicializační funkci použitého ovladače a vyčká, než dojde k synchronizace řadiče se sběrnicí CAN.

3.4.5.3 Rozhraní použitého ovladače od Freescale [21]

Ovladač poskytuje velké množství funkcí pro komunikaci po sběrnici CAN, včetně funkcí pro obsluhu přerušení od modulu MSCAN:

- Init_CAN, zajišťující inicializaci řadiče a počátek synchronizace.
- Check_CAN_Status, kontrolující aktuální stav řadiče.
- Config_CAN_MB, asociující konkrétní buffer v módu odesílání/přijímání pro konkrétní Message Object.
- Check_CAN_MB_Status, vracející stav konkrétního bufferu. Pro přijímací buffer jsou nejvýznamnější stavy NEWDATA a OVERRUN, signalizující příjem nových dat.
- Load_CAN_MB, která nakopíruje data z aplikace do bufferu ovladače.
- Transmit_CAN_MB, která spustí odesílání bufferu.
- Read_CAN_MB_Data, která nakopíruje data z bufferu ovladače do aplikace.
- CAN_Receive pro obsluhu přerušení přijímače CAN. V případě přijetí zprávy funkce zkontroluje všechny definované buffery a v případě shody identifikátorů do něj zprávu nakopíruje a změní jeho status.
- CAN_Transmit, CAN_ERR a CAN_Wakeup pro obsluhu přerušení vysílače CAN

Ovladač poskytuje velké množství dalších funkcí pro práci s řadičem. Jejich popis, včetně kompletních prototypů, lze nalézt v manuálu k ovladači [21].

3.4.5.4 Implementace knihovny MSCAN [11][21]

Rozhraní ovladače je voláno přímo z knihovny Dashboard, s výjimkou inicializace. Ta je implementována zvlášť v knihovně MSCAN:

MSCAN_Init Úkolem této funkce je nastavení modulů SIM a NVIC.

1. Nejprve je vynulován 16. bit registru SIM_PINSEL1, čímž je zajištěn výstup řadiče na pinech, připojených k budiči integrovanému na desce TRK-KEA128.

2. Dále je pomocí registru NVIC_IPR7 nastavena priorita obou MSCAN přerušení a přerušení jsou povolena pomocí registru NVIC_ISER.

MSCAN_Start

1. Funkce nejprve zapíše jedničku na výstupní pin PTB16, který je připojen na STB vstup budiče CAN. Bez této operace je budič neaktivní [13][11].

2. Dále zavolá funkci Init_CAN použitého ovladače, čte aktuální stav řadiče pomocí Check_CAN_Status a vyčkává, než dojde k synchronizaci se sběrnicí.

Po návratu z této funkce je ovladač i řadič CAN připraven pro použití.

3.4.6 Dashboard

Úkolem knihovny Dashboard, tvořené soubory dashboard.c a dashboard.h, je implementace navrženého komunikačního protokolu a obsluha přijatých zpráv s využitím výše popsaných ovladačů.

Využívá zmíněné knihovny PWM, LEDSPI a MSCAN. Dále používá modul PIT (Periodic Interrupt Timer) pro generování periodických přerušení.

3.4.6.1 Konfigurace přístrojové desky

Pro aplikaci přístrojové desky je nutné nastavit mnoho parametrů. V souboru dashboard.h se nacházejí následující makra:

- Dashboard_LightsCount = 14, toto makro je vhodné měnit pouze v případě změny hardwarového zapojení LED diod.

- Dashboard_MessageBufferCount = 12, toto makro je vhodné měnit pouze v případě změny počtu přijímaných zpráv. Spolu s jeho změnou je nutné změnit konfiguraci ovladače a obslužných callbacků, vizte dále.

- Dashboard_MaxVehicleSpeed, Dashboard_MaxEngineSpeed a Dashboard_MaxEngineTemperature, určující maximální rozsah ručičkových ukazatelů.

- Dashboard_LargeGaugeMinPWM a Dashboard_LargeGaugeMaxPWM, určující délku impulzu pro krajní polohy servomotorů ve velkých ukazatelích.

- Dashboard_SmallGaugeMinPWM a Dashboard_SmallGaugeMaxPWM, určující délku impulzu pro krajní polohy servomotorů v malých ukazatelích.
- Makra asociující jednotlivé ručičkové ukazatele s číslem kanálu čítače.
- Makra asociující světelný ukazatel s číslem kanálu na budiči LED.

Soubor dashboard.c dále obsahuje definici pole s callback funkcemi Dashboard_MessageCallbacks, spuštěnými v případě přijetí zprávy s odpovídajícím ID. Velikost tohoto pole je určena hodnotou Dashboard_MessageBufferCount.

3.4.6.2 Dostupné rozhraní

Knihovna je ovládána pomocí následujících funkcí:

- Dashboard_Init, zajišťující inicializaci pomocných ovladačů a nastavení všech ukazatelů na výchozí hodnoty.
- Dashboard_Start, spouštějící generování PWM, časovač PIT pro periodické obnovování LED diod za sériově řízeným budičem a ovladač CAN. Dále inicializuje buffery a spustí smyčku zajišťující příjem zpráv.

3.4.6.3 Implementace [11]

Dashboard_Init Funkce spustí inicializaci knihoven PWM, LEDSPI a MSCAN. Dále nastaví na nulu všechny ručičkové ukazatele a rozsvítí všechna světla.

Dashboard_Start Funkce spustí generování PWM, zinicializuje PIT, CAN a spustí přijímací smyčku.

Dashboard_InitPIT Úkolem této funkce je inicializace časovače pro periodická přerušení PIT.

1. Nejprve povolí hodiny pro modul PIT zapsáním jedničky na 1. bit registru SIM_SCGC.
2. Dále zapne modul čítače zapsáním nuly do registru PIT_MCR.
3. Čítač je s každým taktem dekrementován a při podtečení způsobí přerušení. Zároveň je do něj nahrána jeho počáteční hodnota, nastavená v registru PIT_LDVAL0. Pro periodické přerušení každých 150ms je nutné nastavit čítač na hodnotu 3000000.

4. Zapsáním jedničky na 1. bit registru PIT_TCTRL0 se povolí přerušení v časovači pro nultý kanál a zapsáním jedničky na 0. bit je spuštěn běh časovače.

5. Zbývá povolit přerušení v řadiči NVIC, analogicky jako v předchozích případech.

Funkce Dashboard_FlushLightsCache, která provede kopii lokálního pole, držícího stav LED diod, do ovladače LEDSPI a spustí sériový přenos, je volána v obsluze přerušení od PIT.

Dashboard_InitCANBuffers Funkce nastaví buffery ovladače v rozsahu 0 až Dashboard_MessageBufferCount jako přijímací buffery pro Message Objecty se stejným ID, jako je ID nastavovaného bufferu.

Tedy Message Object ID 0 je přijímán do bufferu č. 0. Po přijetí zprávy je spuštěn callback č. 0.

Dashboard_ReceiveLoop Smyčka definovaná v této funkci kontroluje buffery, zinicializované funkcí Dashboard_InitCANBuffers. V případě přijetí nových dat spustí callback funkci definovanou pro daný buffer v poli Dashboard_MessageCallbacks.

Dashboard_SetVehicleSpeed Tato funkce, obdobně jako SetEngineSpeed, SetEngineTemperature, SetEngineTemperature a SetGasPercentage, provede přepočet přijatých dat, podle nastavených maker, na šířku pulsu pro servomotor a nastaví ji voláním rozhraní knihovny PWM.

Dashboard_SetLight Funkce nastaví stav LED diody, určené argumentem, na 0xFF (tedy plně rozsvíceno). Analogicky ClearLight nastaví stav LED diody na 0x00 (plně zhasnuto). Vzhledem k použitému budiči LED je možné diody rozsvítit s menším jasem, tato možnost však není současnou aplikací využívána. Poslední dvě LED, 12 a 13, mají ztrojeno zapojení na budič, proto ve funkci probíhá zápis do všech tří kanálů zároveň.

Dashboard_Receive... 12 callback funkcí, spuštěných při přijetí příslušné zprávy ze sběrnice CAN. Uvnitř funkce je definována akce: nastavení analogového ukazatele nebo rozsvícení/zhasnutí kontrolky.

3.5 Demoaplikace na STM32F429ZI

K otestování funkčnosti zapojení vývojové desky STM32F429ZI Discovery Kit byla realizována jednoduchá demoaplikace. Ta čte zprávy ze sběrnice CAN a jejich obsah vypisuje na obrazovku. Pro její vývoj bylo použito online prostředí mBed [20] a v něm dostupných knihoven.

3.5.1 LCD displej

Jako vzor demoaplikace sloužila ukázka kódu, dostupná v mBed pod názvem DISCO-F429ZI_LCDTS_demo [22].

Aplikace využívá knihovny BSP_DISCO_F429ZI, LCD_DISCO_F429ZI a TS_DISCO_F429ZI, implementující obsluhu na desce dostupného hardwaru, jako řadiče barevného LCD displeje a dotykové vrstvy.

Prostřednictvím rozhraní těchto knihoven lze obě zmíněné komponenty snadno ovládat vysokoúrovňovými příkazy.

3.5.2 Komunikace po sběrnici CAN

Vzhledem k neúspěšným pokusům s interním řadičem byl pro komunikaci využit externí řadič MCP2515. Ten s procesorem komunikuje prostřednictvím rozhraní SPI [16].

Pro tento obvod je v prostředí mBed k dispozici knihovna s názvem MCP2515, implementující základní operace s řadičem. Jejím vstupem je objekt typu SPI, využitý pro komunikaci, a pin použitý jako Slave Select.

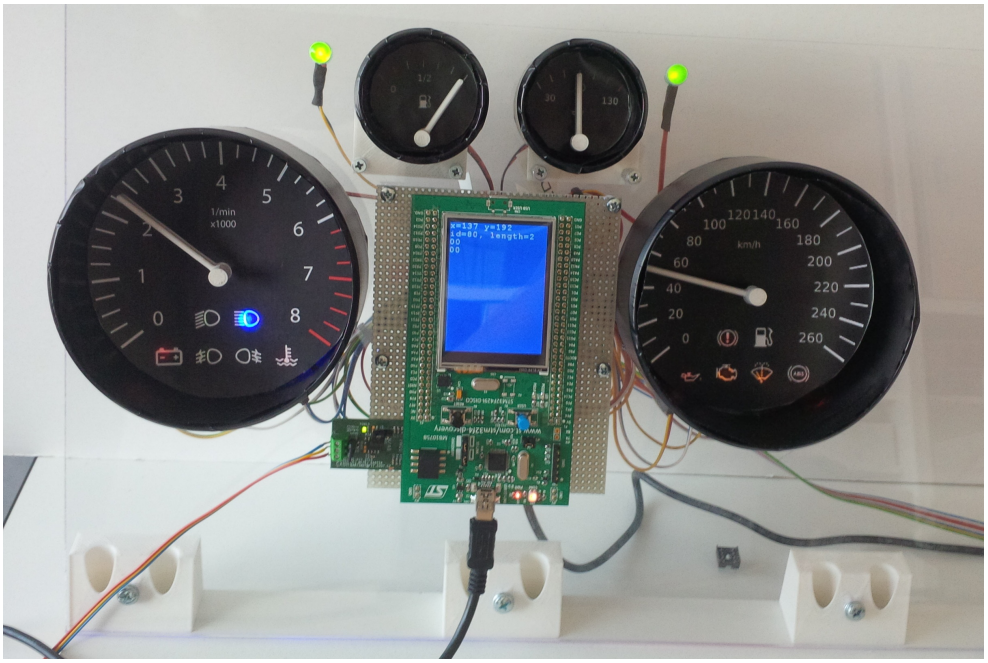
Nejvýznamnější funkce této knihovny jsou:

- readStatus, která z řadiče vyčte stavové slovo, obsahující například informaci o nově přijaté zprávě.
- readDATA_ff_0, která vyčte data z přijímacího bufferu 0. Analogicky pro buffer 1.
- load_ff_0, která zapíše data do vysílacího bufferu 0. Analogicky pro buffer 1.
- send_0, která odešle data zapsaná do bufferu 0.

3.5.3 Implementace

Samotná aplikace je, kromě výše zmíněných knihoven, tvořena jediným souborem, main.cpp. V něm jednoduchá smyčka kontroluje stav dotykové vrstvy a řadiče CAN.

V případě dotyku vyšle aplikace jeho souřadnice po sběrnici CAN. V případě přijaté zprávy jsou informace o ní vypsány na displej.



Obrázek 3.6: Fotografie výsledného modelu přístrojové desky.

Kapitola 4

Testování

Součástí realizace bylo také průběžné testování všech komponent.

4.1 Plošný spoj

Výroba plošného spoje byla zadána společnosti PragoBoard s.r.o., kde proběhlo také jeho elektrické testování. To by mělo odhalit všechna přerušení a zkraty, vzniklé během výrobního procesu.

Přesto jsem před osazováním spoje provedl alespoň jeho základní kontrolu, zaměřenou především na zkraty mezi napájecími vstupy a výstupy.

4.1.1 Napájecí zdroj

Následně jsem jako první osadil levou část plošného spoje, na které se nachází spínaný zdroj. Po připojení vstupního napětí 12V jsem zkontroloval výstupní napětí, které bylo dle očekávání 5V. Další test jsem provedl po zatížení výstupních svorek zdroje odporem 4,7kΩ, čímž jsem zajistil odběr přibližně 1A, což se na velikosti výstupního napětí nijak neprojevilo.

Po měření výstupu osciloskopem je nutné konstatovat, že napětí vykazuje nemalé zvlnění. Výsledky měření je však nutno brát s rezervou, neboť vstupní napětí, poskytované 220/12V zdrojem, vykazuje zvlnění srovnatelné. Na funkci žádné z napájených komponent se dále tato vlastnost negativně neprojevila.

4.1.2 Budič LED

Následovalo osazení a test pravé části plošného spoje, obsahující především sériově řízený budič LED diod. Po zapojení diod byla zasíláním testovacích dat po SPI ověřena jeho správná funkce.

4.2 Servomotory a převody

Nezbytné bylo také otestování použitých servomotorů. Pro implementaci programového vybavení je nutné znát délky impulsu v jejich krajních polohách. Toto měření proběhlo experimentálně, čímž byla zároveň ověřena správná funkčnost motorů.

Druhé testování proběhlo po osazení servomotorů do připravených převodů. Jeho úkolem bylo především ověřit funkčnost převodů, jejich dostatečnou pevnost a zároveň snadné otáčení.

4.3 LED diody

Provést měření bylo nutné také v případě použitých LED. Jeho výstupem byly voltampérové charakteristiky diod, které sloužily jako podklad pro volbu odporu, nastavujícího protékající proud kanálem budiče WS2803.

Test jejich funkčnosti po zapojení proběhl současně s testem budiče.

4.4 Kinetis KEA128

4.4.1 Knihovny PWM a LEDSPI

První byla ověřena správná funkčnost ovladačů PWM a LEDSPI. Testovacím programem byla měněna střída pulsně-šířkové modulace a zároveň odesílána data přes SPI rozhraní. Správný průběh signálů na výstupních pinech byl kontrolován pomocí osciloskopu.

4.4.2 Ovladač řadiče CAN

Pro testování komunikace po sběrnici CAN byl využit přípravek USBTin, zprostředkující rozhraní mezi počítačem připojeným pomocí USB kabelu a připojenou sběrnicí CAN [23].

Správný příjem a odesílání zpráv z Kinetis KEA128 byl ověřen jednoduchou demoaplikací. Ta očekávala zprávu, inkrementovala přijatá data a s novým identifikátorem je odeslala zpět.

4.4.3 Test funkční logiky

Poslední test proběhl po zapojení všech komponent přístrojové desky a implementaci řídící logiky. Pomocí zpráv zasílaných po sběrnici CAN byla vyvolána všechna možná větvení kódu a kontrolována správná reakce přístrojové desky. Během tohoto testu bylo odhaleno a opraveno několik funkčních nedostatků.

4.5 STM32F429ZI Discovery Kit

Otestováno bylo také připojení desky palubního počítače k řadiči MCP2515 na realizované desce univerzálního plošného spoje. Komunikace po sběrnici CAN byla ověřena prostřednictvím jednoduché demoaplikace navržené v rámci této práce. K tomu byl využit opět přípravek USBTin.

Závěr

Cílem zadání bylo vytvořit model přístrojové desky, vhodný k použití jako pomůcka při výuce automobilových sběrnic na zdejší fakultě.

Na začátku práce jsem vybral potřebné indikátory a navrhl jejich vizuální podobu a vhodné rozmístění na přístrojové desce, s ohledem na uživatelskou použitelnost.

Navrhl a vymodeloval jsem potřebné podpůrné prvky, které byly vytištěny z plastu na 3D tiskárně. Mezi ně patřily opěrky pro přichycení přístrojové desky, ručičky ukazatelů, držáky pro instalaci servomotorů a potřebné převody. Ty jsem namontoval a jejich funkčnost v rámci několika testů ověřil.

Pro zapojení zdroje napětí a budiče LED jsem vybral vhodné součástky a navrhl jsem plošný spoj, který řeší také připojení servomotorů a zmíněných diod. Tištěný spoj jsem následně osadil, oživil a otestoval.

Navrhl jsem programové vybavení, pro desku TRK-KEA128 s procesorem ARM, zajišťující čtení zpráv ze sběrnice CAN, ovládání servomotorů pomocí pulzně-šířkové modulace a budiče LED diod pomocí sériového rozhraní. Správnou funkci programu jsem opět ověřil jednoduchými testy.

Kromě toho jsem na přístrojovou desku instaloval modul palubního počítače včetně hardwarového vybavení potřebného pro komunikaci přes sběrnici CAN a jednoduchou demoaplikací ověřil jeho funkci.

Sestavil jsem přístrojovou desku jako funkční celek a ověřil jsem funkčnost jejího ovládání přes sběrnici CAN.

Literatura

[1] Ulsoy, A. G.; Peng, H.; Cakmakci, M.: *Automotive Control Systems*. Cambridge University Press, 2012, ISBN 978-1-107-01011-6, [cit. 2016-04-30].

[2] Bouyssounouse, B.; Sifakis, J.: *Embedded Systems Design*. Springer Berlin Heidelberg New York, 2005, ISBN 3-540-25107-3, [cit. 2016-04-30].

[3] Robert Bosch GmbH: *CAN Specification 2.0 [online]*. 1991, [cit. 2016-04-30]. Dostupné z: `http://www.kvaser.com/software/7330130980914/V1/can2spec.pdf`

[4] International Organization for Standardization: *ISO 11898-2:2003 Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit*. 2003.

[5] Noergaard, T.: *Embedded Systems Architecture*. Elsevier, 2005, ISBN 0-7506-7792-9, [cit. 2016-05-02].

[6] Firoozian, R.: *Servo Motors and Industrial Control Theory*. Springer Cham Heidelberd New York Dordrecht London, 2014, ISBN 978-3-319-07275-3, [cit. 2016-05-07].

[7] Freescale Semiconductor, Inc.: *KEA128 Sub-Family Data Sheet [online]*. 2014, [cit. 2016-04-30]. Dostupné z: `http://cache.nxp.com/files/microcontrollers/doc/data_sheet/S9KEA128P80M48SF0.pdf`

[8] Texas Instruments Incorporated: *LM2676 SIMPLE SWITCHER® High Efficiency 3A Step-Down Voltage Regulator [online]*. 2013, [cit. 2016-04-30]. Dostupné z: `http://www.ti.com/lit/ds/symlink/lm2676.pdf`

[9] Worldsemi: *Preliminary WS2803 18-Channel Constant Current LED Driver With Programmable PWM Outputs [online]*. 2008, [cit. 2016-04-30]. Dostupné z: `http://www.world-semi.com/uploads/soft/120502/1-120502110159.rar`

[10] NXP Semiconductors Netherlands B.V.: *NXP and Freescale Announce Completion of Merger [online]*. 2015, [cit. 2016-04-30]. Dostupné z: http://investors.nxp.com/phoenix.zhtml?c=209114&p=irol-newsArticle&ID=2120581

[11] Freescale Semiconductor, Inc.: *KEA128 Sub-Family Reference Manual [online]*. 2014, [cit. 2016-04-30]. Dostupné z: http://cache.nxp.com/files/microcontrollers/doc/ref_manual/KEA128RM.pdf

[12] ARM: *Cortex M0+ Technical Reference Manual [online]*. 2012, [cit. 2016-04-30]. Dostupné z: http://infocenter.arm.com/help/topic/com.arm.doc.ddi0484c/DDI0484C_cortex_m0p_r0p1_trm.pdf

[13] Freescale Semiconductor, Inc.: *TRK-KEA128 Board Schematics [online]*. 2013, [cit. 2016-04-30]. Dostupné z: http://cache.nxp.com/files/soft_dev_tools/hardware_tools/schematics/TRK-KEA128_SCH.pdf

[14] STMicroelectronics: *Discovery kit with STM32F429ZI MCU [online]*. 2016, [cit. 2016-04-30]. Dostupné z: http://www2.st.com/content/ccc/resource/technical/document/user_manual/6b/25/05/23/a9/45/4d/6a/DM00093903.pdf/files/DM00093903.pdf/jcr:content/translations/en.DM00093903.pdf

[15] Ing. Matěj Bartík: *MI-PVS ZYBO CAN PMOD INTERFACE BOARD (BARTIMAT@FIT.CVUT.CZ)*. ČVUT FIT, 2015.

[16] Microchip Technology Inc: *MCP2515 Stand-Alone CAN Controller With SPI Interface [online]*. 2007, [cit. 2016-05-12]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/21801e.pdf

[17] OpenSCAD: *About OpenSCAD [online]*. [cit. 2016-04-30]. Dostupné z: http://www.openscad.org/about.html

[18] CadSoft: *What is EAGLE [online]*. 2011, [cit. 2016-04-30]. Dostupné z: http://www.cadsoftusa.com/eagle-pcb-design-software/about-eagle/

[19] Freescale Semiconductor, Inc.: *Quick Reference Guide For CodeWarrior Suites [online]*. 2015, [cit. 2016-04-30]. Dostupné z: http://cache.nxp.com/files/soft_dev_tools/doc/fact_sheet/950-00437.pdf

[20] ARM: *The mbed OS User Guide [online]*. [cit. 2016-04-30]. Dostupné z: https://docs.mbed.com/docs/getting-started-mbed-os/en/latest/

[21] Freescale Semiconductor, Inc.: *MSCANDriver For KEA User Manual [online].* 2014, [cit. 2016-04-30]. Dostupné z: `http://cache.nxp.com/files/microcontrollers/software/device_drivers/MSCAN_KEA_DEVD.zip`

[22] ARM mBed: *DISCO-F429ZI [online].* Dostupné z: `https://developer.mbed.org/platforms/ST-Discovery-F429ZI/`

[23] Thomas Fischl: USBtin - USB to CAN interface. 2016. Dostupné z: `http://www.fischl.de/usbtin/`

Příloha A

Seznam použitých zkratek

API Application Programming Interface

ARM Acorn RISC Machine

CAD Computer Aided Design

CAN Controller Area Network

ČVUT České vysoké učení technické

FIT Fakulta informačních technologií

GPIO General Purpose Input/Output

I2C Inter-Integrated Circuit

LED Light-emitting Diode

LIN Local Interconnect Network

NVIC Nested Vector Interrupt Controller

P2P Peer-to-Peer

PIT Periodic Interrupt Timer

PWM Pulse-Width Modulation

RISC Reduced Instruction Set Computing

RTR Remote Transmission Request

SIM System Integration Module

SMD Surface-Mount Device

SPI Serial Peripheral Interface

Příloha B

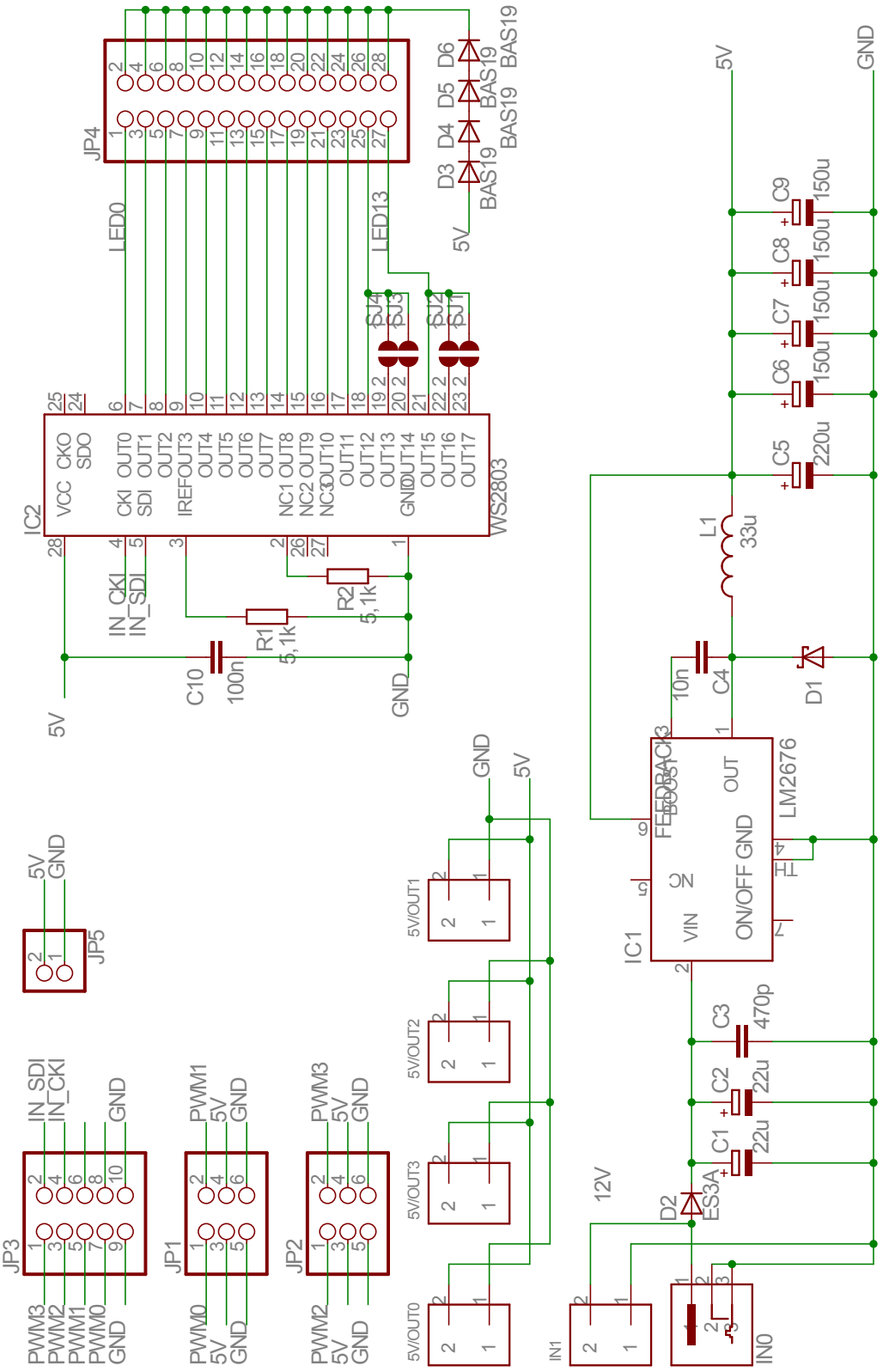
Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `flash`
  - `kea128` ..... Přeložený kód pro KEA128
  - `stm32f4demo` ..... Přeložený kód pro STM32F4
- `src`
  - `eagle` ..... projekt Eagle s plošným spojem
  - `kea128` ..... zdrojové kódy aplikace pro Kinetis KEA128
  - `stm32f4demo` ..... zdrojové kódy demoaplikace pro ST32F4
  - `thesis` ..... zdrojový text práce ve formátu LaTeX
- `text` ..... text práce
  - `BP_Socha_Petr_2016.pdf` ..... text práce ve formátu PDF

Příloha C

Seznam součástek a schéma plošného spoje

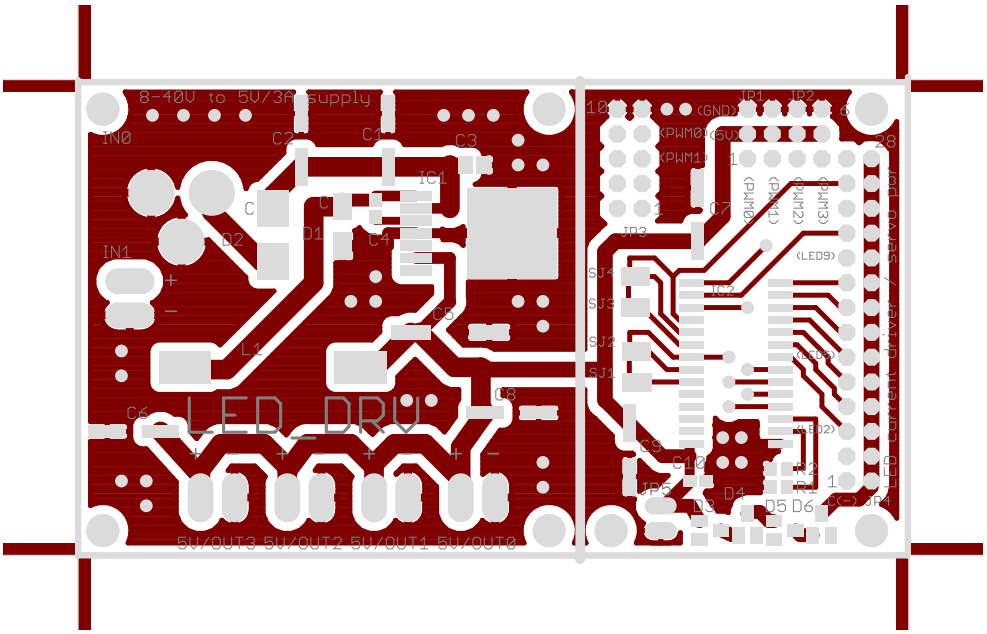
Jméno	Hodnota	Pouzdro	Výrobce a katalogový název
C1, C2	22uF	6,3mm	NICHICON UCD1H220MCL1GS
C3	470nF	0805	AVX 08055C474KAT2A
C4	10nF	0603	AVX 06033C103KAT2A
C5	220uF	8mm	NICHICON UCD1C221MCL6GS
C6-C9	150uF	6,3mm	NICHICON UCD1C151MCL1GS
C10	100nF	0603	AVX 0603YC104JAT2A
D1		DO214 AC	DIODES INCORPORATED B350A-13-F
D2		DO214 AB	DIOTEC SEMICONDUCTOR ES3A
D3, D4		SOT23	NXP BAS19.215
D5, D6			Neosazeny, namísto nich vodič
IC1		D2PAK-7	TEXAS INSTRUMENTS LM2676S-5.0/NOPB
IC2		SOP28	WORLDSEMI WS2803
IN0			CLIFF FC681465P (DC-10LP)
IN1			DEGSON ELECTRONICS DG381-3.5-02P-11-00A(H)
L1	33uH		FERROCORE DL50-33
OUT1-OUT3			DEGSON ELECTRONICS DG381-3.5-02P-11-00A(H)
R1, R2	5,1k	0603	VISHAY CRCW06035K10FKTABC



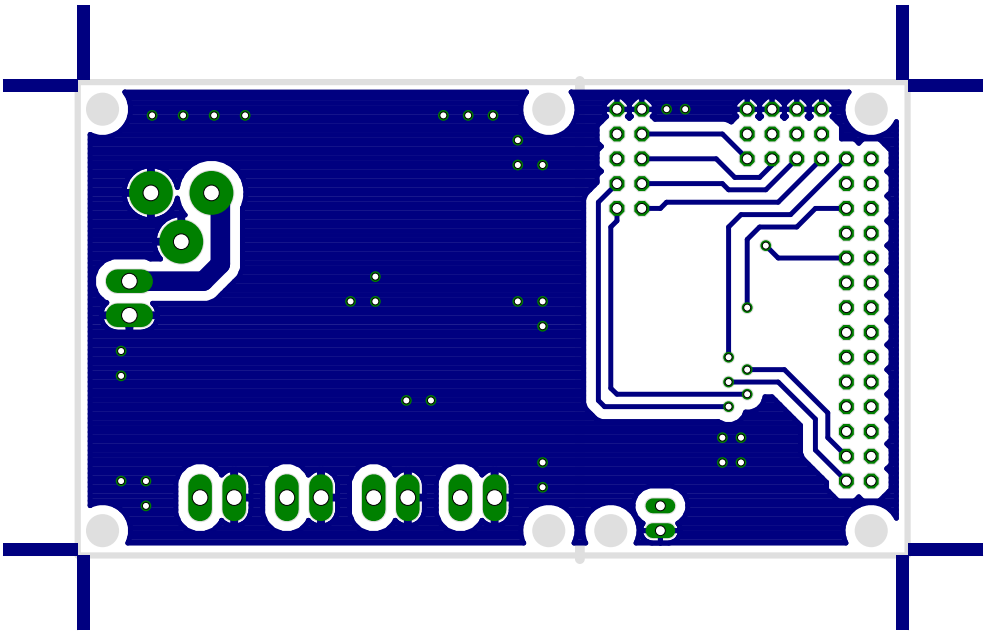
Obrázek C.1: Schéma plošného spoje

Příloha D

Maska plošného spoje



Obrázek D.1: Maska plošného spoje, pohled ze strany součástek



Obrázek D.2: Maska plošného spoje, pohled ze strany spojů