



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV MIKROELEKTRONIKY

DEPARTMENT OF MICROELECTRONICS

## ŘÍDICÍ SYSTÉM PRO ZÁVODNÍ PNEUMATICKÝ MONOPOST

CONTROL SYSTEM FOR RACING PNEUMATIC MONOPOST

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Jakub Kaderka

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vladimír Levek

BRNO 2019

# Diplomová práce

magisterský navazující studijní obor **Mikroelektronika**  
Ústav mikroelektroniky

**Student:** Bc. Jakub Kaderka

**ID:** 173672

**Ročník:** 2

**Akademický rok:** 2018/19

**NÁZEV TÉMATU:**

## Řídicí systém pro závodní pneumatický monopost

### POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a zrealizujte řídicí systém pro závodní pneumatický monopost. Systém bude řídit procesy všech elektronických periférií v monopostu, tedy: ovládání pneumatických ventilů, snímání polohy pístu, měření tlaku, snímání ovládacích prvků. Dále bude systém komunikovat s palubním dotykovým displejem. Součástí bude rovněž telemetrická komunikace přenášející provozní a diagnostická data do počítače prostřednictvím bezdrátové sítě. Tato data budou zároveň zálohována prostřednictvím SD karty.

V rámci diplomové práce navrhněte a realizujte celý řídicí systém včetně jeho periférií. Dále popište dotčené komunikační technologie a klíčové technologie použité v návrhu. Všechny dílčí etapy návrhu popište a technicky vyjádřete. Závěrem proveďte testování a ověření funkčnosti celého systému i s ohledem na bezpečnost provozu.

### DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

**Termín zadání:** 4.2.2019

**Termín odevzdání:** 21.5.2019

**Vedoucí práce:** Ing. Vladimír Levek

**Konzultant:**

**doc. Ing. Lukáš Fucík, Ph.D.**  
*předseda oborové rady*

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Práce se zabývá vývojem řídicího systému pro studentské pneumaticky poháněné závodní vozidlo Falcon.

Navrhovaný systém je pro zvýšení spolehlivosti a udržitelnosti rozdělen do 4 samostatných bloků - jednotka řízení pohonu, napájecí zdroj, modul pro interakci s uživatelem a rychloměr. Tyto moduly jsou propojeny sběrnici CAN, pro kterou byl vyvinut vlastní komunikační protokol optimalizovaný pro jednoduchost implementace a rychlost přenosu dat. Mimo samotného řízení pneumatického okruhu umožňuje elektronika měření provozních údajů vozidla (tlaky v různých místech okruhu, rychlost, ujetá vzdálenost, atd.). Tyto hodnoty jsou ukládány na SD kartu a volitelně odesílány bezdrátově do počítače.

Druhá část práce se zaměřuje na popis řídicího programu a jeho funkčních bloků. Pro účely ladění byl vyvinut systém zpracování a odesílání systémových výpisů, všechny zprávy z jednotlivých modulů jsou ukládány na SD kartu pro pozdější analýzu. Pro základní programové bloky byly napsány jednotkové testy pro automatické testování.

Vozidlo Falcon s touto elektronikou se v květnu 2019 zúčastnilo závodů pořádaných firmou Aventics, kde se umístilo v závodu na dojezd na 8. místě a v závodu na akceleraci na 7. místě.

## KLÍČOVÁ SLOVA

Pneumatický monopost, STM32, ChibiOS, pneumobil

## ABSTRACT

The thesis is focused on developing a control system for a students project of racing car Falcon with pneumatic propulsion system.

The electronic design was splitted into 4 standalone modules to increase reliability and maintainability of the system. The modules are engine control unit, power supply, human machine interface and speedometer. A CAN bus with customly designed communication protocol is used to interconnect these modules. Except the pneumatic circuit control the electronics takes care also about measuring the race related data (pressures in the pneumatic circuit, car speed, etc.). These values are stored on SD card and optionally sent over a wireless link to PC.

In the second part of the thesis design of the firmware is described. For the debugging purposes a logging system was developed. The log messages from all modules are sent over CAN and stored on the SD card for post-mortem analysis. Unit tests were written for all elementary software blocks.

The Falcon vehicle with this control system participated in the competition organised by the Hungarian company Aventics during May 2019. The Falcon ended on 8th place in long distance race and on 7th place in arcade race.

## KEYWORDS

Pneumatický monopost, STM32, ChibiOS, pneumobil

KADERKA, Jakub. *Řídicí systém pro závodní pneumatický monopost*. Brno, 2019, 70 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce: Ing. Vladimír Levek

## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Řídicí systém pro závodní pneumatický monopost“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora

## PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Vladimíru Levkovi za odborné vedení, Jakubovi Lešinskému a Ing. Janu Hrbáčkovi, Ph.D. za odborné diskuze, Ing. Filipovi Jenišovi za vtažení do tohoto projektu, Ing. Petrovi Krejčíříkovi za zajištění financování od ústavu konstruování, Ing. Matuši Ranušovi, Ph.D. za neutuchající nadšení při organizaci vývoje pneumobilů a v neposlední řadě své přítelkyni za nekonečnou trpělivost.

Brno .....

.....

podpis autora

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře  
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072  
**Centrum senzorických, informačních a komunikačních systémů (SIX)**  
operačního programu Výzkum a vývoj pro inovace.

# Obsah

<b>Úvod</b>	<b>10</b>
<b>1 Teoretický rozbor</b>	<b>12</b>
1.1 Funkční princip pneumatického monopostu . . . . .	12
1.2 Možnosti meziprocesorové komunikace . . . . .	13
1.2.1 Rozhraní UART . . . . .	13
1.2.2 Sběrnice I2C . . . . .	13
1.2.3 Průmyslová sběrnice RS-485 . . . . .	13
1.2.4 Automobilová sběrnice LIN . . . . .	14
1.2.5 Automobilová sběrnice CAN . . . . .	14
1.3 Možnosti bezdrátové komunikace . . . . .	15
1.3.1 Moduly BLE . . . . .	16
1.3.2 Moduly Nordic NRF24 . . . . .	16
1.3.3 Wi-Fi modul ESP8266 . . . . .	16
1.3.4 Moduly řady ZigBee . . . . .	16
1.3.5 Modul HopeRF RFM69 . . . . .	17
1.4 Mikroprocesory a knihovny pro ovládání periferií . . . . .	17
1.4.1 Knihovna HAL . . . . .	18
1.4.2 Knihovna Low Level Library . . . . .	18
1.4.3 Projekt Libopencm3 . . . . .	18
1.4.4 Knihovna CMSIS . . . . .	19
1.4.5 Operační systém ChibiOS . . . . .	19
<b>2 Návrh řídicího systému</b>	<b>20</b>
2.1 Požadavky na elektroniku vozidla . . . . .	20
2.1.1 Mechanické provedení . . . . .	21
2.2 Blokový návrh . . . . .	22
2.2.1 Distribuce napájení . . . . .	22
2.2.2 Univerzální komunikační a napájecí konektor . . . . .	23
2.3 Jednotka řízení motoru (ECU) . . . . .	23
2.3.1 Digitální vstupy . . . . .	24
2.3.2 Měření rychlosti . . . . .	24
2.3.3 Měření tlaku . . . . .	24
2.3.4 Měření teploty . . . . .	25
2.3.5 Výkonové výstupy . . . . .	26
2.3.6 Proudová smyčka 4–20 mA . . . . .	26
2.3.7 Napájení . . . . .	27
2.4 Jednotka telemetrie a interakce s uživatelem (HMI) . . . . .	27
2.4.1 Napájení . . . . .	28
2.5 Jednotka zobrazovače rychlosti vozu (SDU) . . . . .	29

2.6	Napájecí zdroj (PSU)	30
2.6.1	Spínané zdroje	30
2.6.2	Měření proudů a napětí	32
2.6.3	Proudová ochrana a ochrana akumulátoru	32
2.7	Bezdrátový komunikační modul pro PC	33
<b>3</b>	<b>Komunikační protokol</b>	<b>34</b>
3.1	Nízkoúrovňový protokol pro sběrnici CAN	34
3.1.1	Identifikátor rámce	35
3.1.2	Odesílání dlouhých zpráv	35
3.2	Nízkoúrovňový protokol pro sériové rozhraní	36
3.3	Vysokoúrovňový komunikační protokol	36
3.4	Bezdrátová komunikace	37
<b>4</b>	<b>Programové vybavení</b>	<b>38</b>
4.1	Funkční bloky	38
4.1.1	Ovladače (Drivers)	38
4.1.2	Systémový záznam (Log)	40
4.1.3	Konfigurace (Config)	40
4.1.4	Komunikace (Comm)	41
4.1.5	Modul ECU	41
4.1.6	Modul HMI	42
4.1.7	Modul PSU	43
4.1.8	Modul SDU	44
4.1.9	Převodník RFM69 na USB	44
4.2	Řízení pneumatického okruhu	44
4.2.1	Měření polohy pístu ve válci	45
4.2.2	Závodní režimy	45
4.3	Testování	45
4.4	Dokumentace	46
4.5	Programy pro počítač	47
4.5.1	Generování konfiguračních parametrů	47
4.5.2	Generování komunikačních příkazů	47
4.5.3	Telemetrie	48
<b>5</b>	<b>Konstrukce a oživení elektroniky</b>	<b>49</b>
5.1	Desky plošných spojů	49
5.2	Oživování elektroniky	51
5.3	Umístění elektroniky ve vozidle	52
5.4	Bezpečnost	53
<b>6</b>	<b>Závěr</b>	<b>54</b>



<b>Literatura</b>	<b>56</b>
<b>Seznam symbolů, veličin a zkratk</b>	<b>57</b>
<b>Seznam příloh</b>	<b>60</b>
<b>A Schémata zapojení</b>	<b>62</b>
A.1 Modul ECU . . . . .	62
A.1.1 Hlavní list . . . . .	62
A.1.2 Vstupy . . . . .	63
A.1.3 Výstupy . . . . .	64
A.1.4 Řízení ventilů . . . . .	65
A.2 Modul SDU . . . . .	66
A.3 Modul HMI . . . . .	67
A.3.1 Hlavní list . . . . .	67
A.3.2 Napájecí zdroj . . . . .	68
A.3.3 SD karta . . . . .	69
A.3.4 Tlačítka . . . . .	70
A.4 Modul PSU . . . . .	71

# Úvod

V roce 2017 mi byla nabídnuta účast na vývoji pneumaticky poháněného závodního vozu Falcon (Obr. 1), jehož stavba probíhá pod záštitou Ústavu konstruování strojní fakulty VUT.



Obr. 1: Závodní monopost Falcon (převzato z [1])

Od roku 2007 pořádá firma Aventics, dodavatel pneumatických prvků, závody studentských pneumaticky poháněných vozidel v maďarském Egeru. V roce 2012 se poprvé zapojili do závodů studenti z Fakulty strojního inženýrství na VUT a zahájili tak již sedmiletou tradici účasti na tomto závodním klání.

Účastníci spolu závodí ve třech hlavních disciplínách - v závodě na čas (arcade race), závodě na dojezd (long distance race) a závodě na akceleraci (acceleration race).

Závod na čas probíhá ve třech kolech, kdy do dalšího postupují pouze ti nejlepší z kola předchozího. Na trať je vypuštěno současně několik vozů a je měřena doba, za kterou ujedou dvě kola. Při závodě na akceleraci proti sobě startují dva vozy a snaží se dojet do cíle v nejkratším možném čase. Je vyhodnocována i maximální rychlost vozu, která je měřena v cílové rovině. Při závodech na dojezd každý tým dostává malou tlakovou láhev s přesně daným množstvím vzduchu a cílem je urazit co největší vzdálenost. Mimo těchto hlavních disciplín je také hodnoceno zpracování projektové dokumentace, vzhled vozidla, měření a zpracování provozních veličin během závodu (telemetrie), atd.

Během posledních let se brněnskému týmu podařilo probojovat až na první příčky závodu a zařadit stavbu pneumaticky poháněných vozidel do projektové výuky Ústavu konstruování.

Jelikož vozidlo vzniká pod patronátem strojní fakulty VUT, bylo provedení elektronického řízení pneumatických prvků dlouho zanedbáváno a omezeno pouze na základní funkcionalitu nutnou k provozu.

V loňském návrhu jsem opustil dříve používanou platformu Arduino a řídicí systém začal navrhovat od začátku. Bohužel vzhledem k nedostatečnému množství volného času nebyl návrh rozvinut do očekávaného rozsahu. Dále se během závodu projevily mnohé nedostatky systému, selhání jednoho z konektorů nakonec zavinilo zastavení vozidla v druhém kole závodu na čas a tedy i jeho nepostoupení do kola finálního.

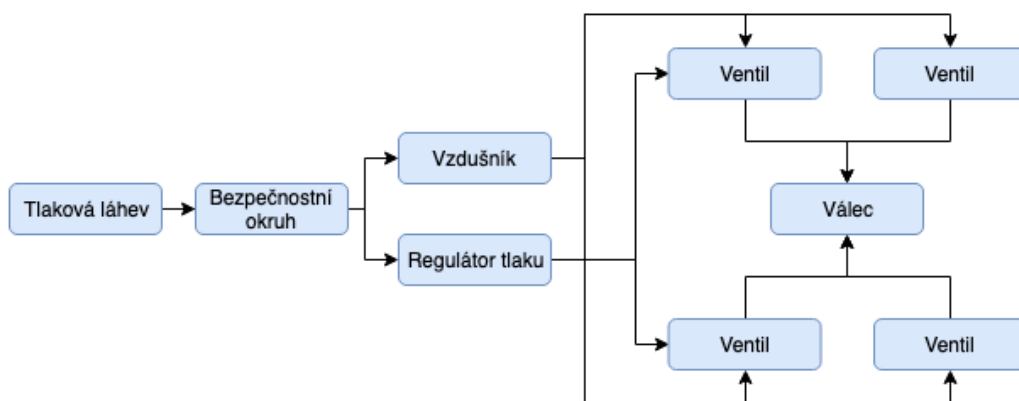
Cílem této práce je navrhnout nový řídicí systém, který by řešil nedostatky systému předchozího, výrazně rozvinul jeho možnosti a byl použitelný i pro vývojově starší vůz Javelin. Na jaře roku 2019 se pak oba vozy zúčastní mezinárodních závodů v maďarském Egeru.

# 1 Teoretický rozbor

## 1.1 Funkční princip pneumatického monopostu

Základním úkolem konstrukčního týmu je navrhnout vozidlo poháněné stlačeným vzduchem z tlakové nádoby (výchozí tlak 200 barů) za využití pneumatických prvků dodávaných firmou Aventics.

Blokové schéma pneumatického okruhu vozidla Falcon je na obrázku 1.1. Vzduch z tlakové nádoby (lahve) je přes redukční ventil přiveden na bezpečnostní okruh. Ten slouží jako tlaková pojistka, pokud je tlak na výstupu vyšší jak na vstupu, ventil uvnitř se otevře a vzduch z okruhu vypustí.



Obr. 1.1: Blokové schéma pneumatického okruhu

Následuje rozdělení do dvou větví. V jedné je zapojen elektronický redukční ventil, který umožňuje řízení tlaku v okruhu při závodech na dojezd, kdy změnou tlaku v závislosti na aktuální rychlosti lze dosáhnout snížení spotřeby vzduchu.

Druhá paralelní větev obsahuje vzdušník (tlakovou nádobu o větším objemu) na vyrovnávání tlaků v okruhu při velkém odběru vzduchu, kdy redukční ventil na láhvi omezuje maximální průtok. Velký průtok vzduchu způsobuje ochlazování ventilu, v extrémních případech (závod na akceleraci) je pokles teploty natolik velký, že ventil zamrzá. Bohužel pravidla neumožňují záměnu ventilu za vhodnější typ s větším maximálním průtokem.

Během závodu je aktivní pouze jedna z větví, u závodů na čas a na akceleraci je do obvodu vřazen vzdušník a redukční ventil je odpojen (tlak v okruhu je dán nastavením ventilu na lahvi - maximálně 10 barů). Při závodech na dojezd je vzdušník odpojen a redukční ventil v druhé větvi snižuje tlak na hodnotu určenou řídicím algoritmem, typicky na méně jak 5 barů.

Tyto dvě větve jsou připojeny na 4 elektronicky řízené ventily, které přepouští tlak z okruhu do pneumatického válce. Na každé straně válce se nacházejí dva ventily z důvodu zvýšení maximálního průtoku vzduchu. Při závodu na dojezd je využita pouze jedna dvojice ventilů.

Píst ve válci je spojen s ozubnicí, jejíž pohyb je převeden dvojicí antiparalelně zapojených volnoběžek na dvojstupňovou převodovku bez synchronizace (při řazení se píst nepohybuje a celé převodové ústrojí je v klidu). Převodovka pak přes řetěz roztáčí zadní kolo [2].

## 1.2 Možnosti meziprocesorové komunikace

Řídicí jednotka bude koncipována modulárně, z čehož plyne nutnost vybrat vhodné rozhraní, které by umožnilo obousměrnou komunikaci mezi moduly. V této podkapitole záměrně nebudou zmíněna bezdrátová řešení, která nejsou pro daný účel vhodná z hlediska vyšší spotřeby a rizika zarušení. Tabulka 1.1 shrnuje vlastnosti níže popisovaných rozhraní.

Tab. 1.1: Srovnání vlastností komunikačních rozhraní

Rozhraní	Uzlů	Multi-master	Rychlost [Bd/s]	Dosah	EMI odolnost
UART	2	-	$\geq 115200$	Jednotky m	Nevyhovující
I2C	128	Ano	400 k	Desítky cm	Nevyhovující
RS-485	32	Ne	10 M	Stovky m	Velká
LIN	16	Ne	20 k	Desítky m	Malá
CAN	$\geq 128$	Ano	1 M	Stovky m	Velká

### 1.2.1 Rozhraní UART

UART je jednou z nejběžnějších komunikačních periférií mikroprocesorů s komunikační rychlostí běžně dosahující 115200 Baud. Je určen pro plně duplexní komunikaci mezi dvěma účastníky pomocí dvou vodičů (označovaných RxD a TxD). Pro propojení většího množství účastníků není příliš vhodný.

### 1.2.2 Sběrnice I2C

I2C [3] je multi-master sběrnice vyvinutá pro komunikaci mezi integrovanými obvody. Jedná o velmi rozšířenou sběrnici, jejíž implementace se vyskytuje prakticky v každém mikroprocesoru. Náklady na realizaci jsou proto minimální. Zařízení jsou ke sběrnici připojena výstupy s otevřeným kolektorem. Sběrnice je velmi náchylná na elektromagnetické rušení a nehodí se pro komunikaci na vzdálenosti větší, než několik desítek centimetrů v rámci jedné desky.

### 1.2.3 Průmyslová sběrnice RS-485

RS-485 je standard sériové komunikace pro průmyslové prostředí, pro komunikaci využívá přenos diferenciálního signálu dvojicí vodičů, je tedy poloduplexní, ale oproti UARTu

umožňuje připojení až 32 zařízení na jednu sběrnici. Komunikaci lze realizovat s minimálními náklady využitím periferie pro UART a převodníku úrovní.

Sběrnice může být jak typu master-slave, tak i multi-master, standard ale nedefinuje řešení kolizí a její implementaci nechává na vyšší komunikační vrstvě.

#### 1.2.4 Automobilová sběrnice LIN

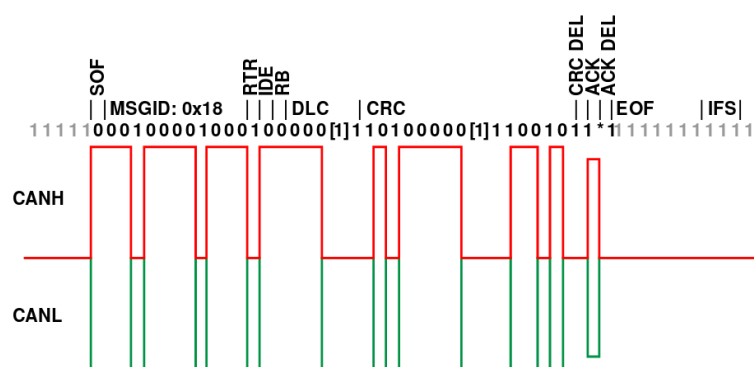
LIN (Local Interconnect Network) [4] je sběrnice určená pro propojování lokálních sítí v dopravních prostředcích. Principem vychází z UARTu, ale využívá pouze jednoho komunikačního vodiče. Na sběrnici může být pouze jeden master a až 15 slave zařízení. Master inicializuje přístup na sběrnici, ke kolizím tedy nedochází. Nevýhodou jednovodičového řešení je nižší odolnost proti elektromagnetickému rušení a tedy i nižší podporované komunikační rychlosti (běžně do 19200 Baud/s).

Zařízení jsou na sběrnici připojena pomocí výstupu s otevřeným kolektorem a pull-up rezistoru. Každé zařízení je schopno během vysílání zároveň přijímat odeslaná data. Fyzickou vrstvu je možné realizovat pomocí UART periferie a jednoduchého převodníku, nicméně některé mikroprocesory mají přímo specializovanou LIN periferii.

Pomocí schopnosti přijímat data vyslaná na sběrnici by bylo možné detekovat kolize v komunikaci a realizovat i multi-master komunikaci za cenu vyšší složitosti při implementaci programu.

#### 1.2.5 Automobilová sběrnice CAN

CAN (Controller Area Network) [5] je multi-master sběrnice určená pro automobilový průmysl. Díky diferenciálnímu páru vodičů je velmi odolná proti rušení a umožňuje komunikaci rychlostí až 1 Mbit/s. Výhodou oproti předchozím řešením je definice řešení kolizí na lince již v rámci protokolu samotného. Příklad přenosu jednoho CAN rámce po diferenciálním páru je na obrázku 1.2.



Obr. 1.2: Příklad přenosu jednoho rámce na sběrnici CAN (převzato z [5])

Arbitráž sběrnice funguje následujícím způsobem. První zařízení začne vysílat data, zároveň čte stav sběrnice a ověřuje, zda odeslaná data odpovídají stavu na sběrnici. V okamžiku, kdy do komunikace vstoupí další zařízení, záleží další dění na odeslaných datech.

Logická nula na sběrnici je dominantní, logická jednička recesivní. Vyšší prioritu má zařízení vysílající dominantní bit, který přebije bit recesivní. Pokud vysílač detekuje kolizi, opustí vysílání a stává se přijímačem. Specifikace CAN sběrnice obsahuje tedy nejen vrstvu fyzickou, definující elektronické propojení zařízení, ale i linkovou, která obsahuje definice formátu přenášených dat a způsob řízení přístupu na sběrnici.

Periferie CAN v mikroprocesorech obsahují obě tyto vrstvy a tak značně zjednodušují logiku řídicího programu. Linková vrstva v sobě zahrnuje i definici komunikačního rámce, který se skládá mimo jiné z 11 bitů dlouhého identifikátoru, 0 až 8 bajtů uživatelských dat a z kontrolního součtu. Mikroprocesor tedy sám zajišťuje nejen opakované odesílání dat v případě kolize na sběrnici, ale také kontrolu správnosti přijímaných informací. Další běžnou vlastností CAN periferii je možnost filtrovat přijímaná data na základě jejich identifikátoru.

Jistou nevýhodou tohoto řešení je snížená maximální propustnost sběrnice, kdy na 64 bitů užitečných dat připadá 44 bitů „neužitečných“. To je daň za vysokou robustnost a spolehlivost přenosu. Pro odeslání většího množství dat než 8 bajtů je nutné implementovat další komunikační vrstvu. Existující standardy jsou např. CANopen, MilCAN, nebo CANaerospace.

Tato sběrnice se jeví jako nejvhodnější a bude použita pro komunikaci mezi moduly. Její výhody převažují nevýhodu v podobě menšího výběru mikroprocesorů s integrovanou CAN periferií.

### 1.3 Možnosti bezdrátové komunikace

Součástí hodnocení závodních týmů je i odesílání a zpracování telemetrických zpráv o vozu. Maximální vzdálenost mezi pozemním týmem a vozidlem se pohybuje okolo 500 metrů, přičemž v cestě stojí často nejen rostliny (keře, stromy), ale i diváci, tribuny a další překážky. Výběr způsobu bezdrátové komunikace musí být tedy přizpůsoben schopnosti komunikovat s minimálními výpadky ve výše uvedených podmínkách. Některé dostupné technologie shrnuje tabulka 1.2. Uváděné hodnoty dosahů a rychlostí jsou maximální možné. Maximálního dosahu lze zpravidla dosáhnout jen s dobrou anténou a velmi nízkou přenosovou rychlostí.

Tab. 1.2: Shrnutí vlastností bezdrátových modulů

Modul	f [MHz]	Dosah [m]	Rychlost [Bd/s]	Rozhraní	Topologie
BLE	2400	~50	1 M	UART, SPI	Master-slave
NRF24	2400	~1500	2 M	SPI	Peer-to-peer
ESP8266	2400	≥ 1000	54 M	UART	Master-slave
ZigBee	868/2400	≥ 75	250 k	UART	Hvězda, strom
RFM69	433/868	~1000	300 k	SPI	Broadcast

### 1.3.1 Moduly BLE

BLE (Bluetooth Low Energy) je nejnovějším standardem Bluetooth. Na trhu je široká škála modulů řízených přes rozhraní UART, nebo SPI. Komunikace funguje na bázi master-slave, kdy master je nejčastěji mobilní telefon a slave samotné zařízení. Před začátkem první komunikace je nutné tato zařízení spárovat zadáním klíče, nebo potvrzením shodnosti klíčů dle zvolené párovací metody.

BLE využívá na pracovní frekvenci 2.4 GHz 40 kanálů, které zařízení během komunikace pravidelně mění. Přenosová rychlost dosahuje až 1 Mb/s. Nevýhodou je nízký dosah v řádu desítek metrů.

### 1.3.2 Moduly Nordic NRF24

Moduly NRF24 jsou mezi vývojáři velmi populární díky nízké ceně a velkému dosahu. Pracují na frekvenci 2,4 GHz, obsahují automatické ověřování validity přijatých dat a opakované odesílání požadavku v případě, že komunikace neproběhla korektně. Přenos dat probíhá obousměrně mezi dvěma body. S vhodnou anténou a maximálním vysílacím výkonem 18 dBm je v otevřeném prostoru možné komunikovat na vzdálenost více jak 1,5 km, avšak s rostoucím množstvím překážek díky vysoké pracovní frekvenci dosah rychle klesá.

Komunikace probíhá na předem zvoleném kanálu, v případě kolize kanálů s jiným závodním týmem tak může dojít ke ztrátě dat.

### 1.3.3 Wi-Fi modul ESP8266

Další z nesčetných možností bezdrátové komunikace je počítačová Wi-Fi síť. Zde se přímo nabízí populární modul ESP8266 se sériovým rozhraním. ESP není jen běžný bezdrátový modul, ale plnohodnotný mikroprocesor. Velkou část logiky programu je možné implementovat přímo v modulu samotném.

S vhodnou anténou lze ve volném prostoru komunikovat na vzdálenost několika kilometrů, v členitější zástavbě však dosah velmi rychle klesá.

Velkou nevýhodou komunikace na 2,4 GHz v pásmu Wi-Fi je velké množství stanic těchto frekvencí využívajících a tedy i velké rušení. Druhým problémem je nutnost autentizace zařízení k přístupovému bodu, což v případě rušení, nebo velké vzdálenosti může způsobit nestabilitu spojení (opakované odpojování a připojování). Výhodou je přítomnost tohoto rozhraní ve většině počítačů a mobilních telefonů a možnost připojit k jednomu přístupovému bodu několik zařízení současně.

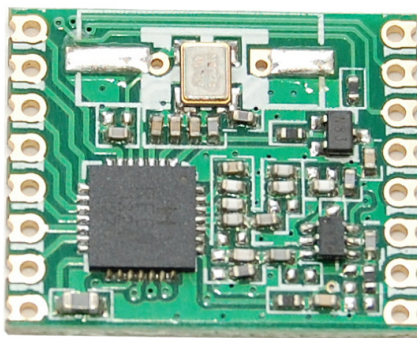
### 1.3.4 Moduly řady ZigBee

ZigBee je technologie pracující v industriálních pásmech 868 MHz a 2,4 GHz. Umožňuje tvorbu sítí složených z několika zařízení (s topologiemi hvězda, síť a strom), často se používá v automatizaci budov a průmyslové automatizaci. Nevýhodou je nízký dosah (okolo 100 metrů) a ve srovnání s dříve zmíněnými moduly i vyšší cena.



### 1.3.5 Modul HopeRF RFM69

RFM69 (obrázek 1.3) je rodina bezdrátových modulů komunikujících v bezlicenčních industriálních pásmech 433, 868 a 915 MHz (pásmo 915 MHz je určeno pro americký trh). Maximální vysílací výkon je 20 dBm, což při citlivosti přijímače -120 dBm (pro komunikační rychlost 1200 baud) umožňuje udržovat spojení i na vzdálenosti přes jeden kilometr. Nižší pracovní frekvence se z zástavbě šíří lépe než rozšířenější 2,4 GHz.



Obr. 1.3: Modul RFM69

Komunikace s řídicím mikroprocesorem probíhá po SPI a modul obsahuje širokou škálu nastavitelných parametrů komunikace, od výstupního výkonu, přes modulaci, šifrování (AES-128), po úpravy parametrů vstupního digitálního filtru.

Při vhodném nastavení je možné z jednoho vysílače posílat data několika přijímačům současně, což by umožnilo simultánní zpracování telemetrie např. na počítači a tabletu. Tento modul se z hlediska pracovní frekvence a vlastností jeví jako nejvhodnější řešení pro odesílání telemetrie vozidla.

## 1.4 Mikroprocesory a knihovny pro ovládání periférií

Každý z bloků řídicího systému bude obsahovat vlastní mikroprocesor (MCU) pro ovládání periférií a zpracování dat. Z hlediska sdílení části kódu, např. pro komunikační rozhraní, a možnosti používat stejné vývojové nástroje je vhodné uvažovat nad jednou rodinou mikroprocesorů pro všechny bloky. Jelikož budou vyrobeny pouze jednotky kusů, není cena kritickým parametrem. Mnohem důležitější jsou kvalitní systémové knihovny a dostupnost vývojových prostředků.

Řízení grafického dotykového displeje pro interakci s uživatelem bude vyžadovat poměrně velké množství výkonu a paměti, proto nejsou příliš vhodné osmibitové mikroprocesory (AVR, PIC). Z výběru 32 bitových rodin se jako nejvhodnější jeví procesory ARM, které jsou velmi levné (mnohdy levnější jak vyšší řady AVR) a obsahují ladicí rozhraní, které umožňuje zasahovat do běhu programu, vyčítat hodnoty proměnných, atd. . .

Po zkušenostech s různými výrobci (Infineon, Toshiba, TI) mikroprocesorů s ARM jádrem byla zvolena rodina STM32 firmy STMicroelectronics. Pro tyto MCU existuje několik různých knihoven s ovladači pro periferie. Jednotlivé knihovny se značně liší ve své komplexnosti, systémových nárocích a ideologickém přístupu. V následující části budou zmíněny možné knihovny, na kterých dále záleží výběr konkrétního MCU a přenositelnost kódu na jiné typy.

Velkou výhodou rodiny STM32 je pinová kompatibilita mezi některými řadami, pokud je během vývoje zjištěno, že vybraný mikroprocesor nedostačuje výkonově, nebo má nedostatek paměti, je možné ho vyměnit za jiný bez nutnosti zasahovat do návrhu desky, nebo přepisovat kód.

#### **1.4.1 Knihovna HAL**

HAL (Hardware Abstraction Layer) je univerzální knihovna pro ovládání periférií mikroprocesorů STM32. Je dodávána společně s nástrojem STM32CubeMx, který umožňuje grafické nastavení jednotlivých periférií a vygenerování konfigurace pro konkrétní zařízení.

HAL poskytuje vysokoúrovňový přístup k HW. Výsledný kód je přenositelný na celou rodinu STM32. Odvrácenou stranou je pak velká náročnost na programovou paměť. Druhým, dříve hojným, problémem je množství chyb v knihovně, zvláště u ovladačů méně často používaných periférií. Konečně vysoká míra abstrakce neumožňuje specifitější nastavení procesoru.

#### **1.4.2 Knihovna Low Level Library**

Low Level Library je nástupcem dřívější Standard Peripheral Library od ST. Oproti HAL je výrazně jednodušší, obsahuje pouze funkce pro základní přístup k perifériím, vyšší úroveň abstrakce je ponechána na uživateli. Výhodou je výrazně menší velikost přeloženého kódu než u knihovny HAL a menší množství chyb v knihovně. To celé ovšem za cenu složitějšího programování a nižší přenositelnosti na jiné rodiny STM32.

Tato knihovna je pro průmyslové prostřední výrazně vhodnější než HAL díky vyšší míře kontroly nad procesorem.

#### **1.4.3 Projekt Libopencm3**

Libopencm3 vychází z podobného přístupu jako Low Level Library, oproti dříve zmíněným knihovnám je vydána pod licencí LGPL a je aktivně vyvíjena opensource komunitou. Dle názvu je určena pro jádra Cortex M3, ale byla portována i na jádra M0, M0+ procesorů STM32. Postupně je přidávána podpora i pro rodiny Atmel SAM3, NXP LPC 11XX, ...

Výhodou je nízkourovňový přístup zajišťující velkou rychlost výsledného kódu a možnost velmi komplexní konfigurace jednotlivých periférií. Podpora řad STM32F0 a F1 je téměř kompletní. Řady F2 a F4 ještě nejsou plně dokončeny.

#### 1.4.4 Knihovna CMSIS

CMSIS je standardní knihovna pro přístup k funkcím samotného ARM jádra. Je vysoce přenositelná, minimalistická a rychlá, ale neobsahuje ovládání většiny periférií, proto jí dále nebude věnována pozornost.

#### 1.4.5 Operační systém ChibiOS

ChibiOS je primárně operační systém reálného času (RTOS) pro vestavěná zařízení, nicméně projekt obsahuje i vrstvu abstrakce nad perifériemi ChibiOS/HAL [6]. Kód abstrakční vrstvy je velmi čistý a rychlý, připravený pro použití s libovolným RTOS. V kombinaci se samotným ChibiOS se jedná o kompletní řešení pro programování mikroprocesorů STM32. Výsledný kód je s minimálními změnami přenositelný mezi jednotlivými rodinami (F0, F1, F4, ...). Pro nekomerční projekty je knihovna uvolněna pod licencí GNU GPL3, popř. pod Apache 2.0.

RTOS zjednodušuje vývoj programu, protože jednotlivé logické funkční bloky lze oddělit do samostatných vláken bez nutnosti jejich periodického volání z hlavní smyčky. Nevýhodou je složitější ladění, zvláště pokud se problém týká současného přístupu ke sdíleným zdrojům.

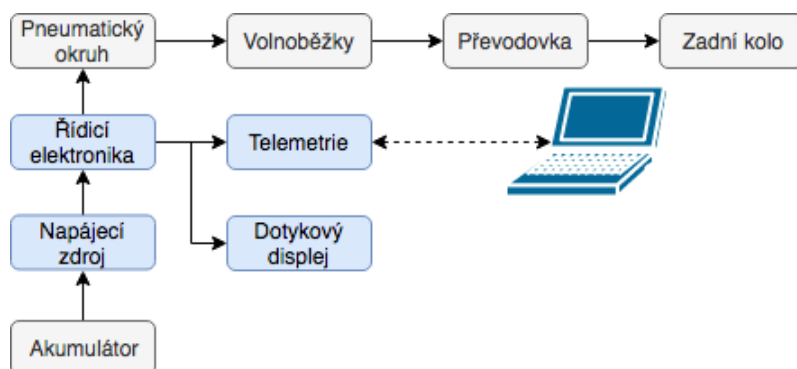
Rozšířenějším konkurentem na poli systémů reálného času je projekt FreeRTOS, který ale neobsahuje vrstvu abstrakce nad hardware, proto není v práci dále rozebírán.

Z výše uvedených důvodů bude pro vývoj použita knihovna ChibiOS/HAL, díky dobré přenositelnosti je možné při návrhu elektroniky zvolit nejvhodnější MCU z rodiny STM32 pro každý z bloků.

## 2 Návrh řídicího systému

Základním a kritickým úkolem navrhovaného systému je řízení pneumatického okruhu vozidla. K tomuto úkolu se přidávají další prvky, jako sbírání telemetrických údajů o vozidle a jejich bezdrátové odesílání do počítače, interakce s řidičem, atd. Výsledkem je poměrně komplexní systém, který bude podrobně popsán v této kapitole.

Celkové blokové schéma vozu Falcon je na obrázku 2.1. Modře zvýrazněné bloky jsou navrhovány v rámci této práce.



Obr. 2.1: Blokové funkční schéma pneumobilu

### 2.1 Požadavky na elektroniku vozidla

Dle pravidel [7] musí být každé vozidlo vybaveno brzdovým světlem a nouzovým vypínačem dostupným jak pro řidiče, tak pro okolo se nacházející osoby. Dále se doporučuje napájecí napětí 24 V, které je vyžadováno většinou pneumatických prvků.

Volitelně je možné soutěžit i v kategorii telemetrie, kdy je hodnoceno ukládání, odesílání a zpracování různých provozních veličin. Posledním požadavkem je krytí elektroniky před vodou minimálně dle normy IP54.

Na základě zkušeností z předchozích let a diskuze s konstrukčním týmem byly vytvořeny podrobnější požadavky. Požadované obecné vlastnosti systému jsou:

- Napájecí napětí 24 V.
- Možnost ukládat provozní data na SD kartu.
- Odesílání dat bezdrátově do počítače během závodu.
- Dotykový displej pro interakci s řidičem.
- Měření zbývajících kapacity akumulátoru a proudového odběru.
- Měření zrychlení vozidla.
- Mechanicky robustní konektory.

Řízení pohonu vyžaduje čtení stavu pedálů vozu a znalost několika různých veličin. Všechny požadované vstupy do jednotky jsou:

- Plynový pedál – indukční spínač, popř. mechanické tlačítko.
- Brzdový pedál – indukční spínač, popř. mechanické tlačítko.

- 2 magnetické senzory krajních poloh pístu.
- Inkrementální enkodér na snímání polohy pístu.
- 2–3 indukční senzory na převodovce (neutrál, první a druhý stupeň).
- Optická brána na měření rychlosti vozu.
- Tlačítka pro klakson a přepínání závodního režimu.
- 3 tlakoměry.
- Teploměry v různých místech pneumatického okruhu.

Poslední částí specifikací jsou výstupy jednotky, mimo samotných pneumatických ventilů je třeba řídit i další funkce:

- 4 elektromagnetické ventily.
- Proudová smyčka 4–20 mA pro elektronický redukční ventil.
- Brzdové světlo.
- Klakson.
- Displej zobrazující rychlost vozu.

### 2.1.1 Mechanické provedení

Závody se odehrávají ve venkovním prostředí, elektronika tedy musí být chráněna před prachem a vlhkostí (je to dáno i pravidly), potažmo vodou (v roce 2018 během části závodu pršelo). To bude zajištěno umístěním kritických částí do plastového montážního boxu s krytím dle normy IP54.

Ze zkušeností se jako problematický bod konstrukce jeví propojovací konektory. Velké množství vstupů a výstupů znamená obvykle i velké množství konektorů, kdy s jejich rostoucím počtem roste i šance selhání některého z nich. Selhání mechanicky nevyhovujícího konektoru bylo příčinou prohry závodu na rychlost v roce 2018, kdy vůz Falcon v prvním kole skončil na druhém místě, ale druhé kolo kvůli ztrátě kontaktu v konektoru jednoho z čidel krajních poloh pístu nedokončil.

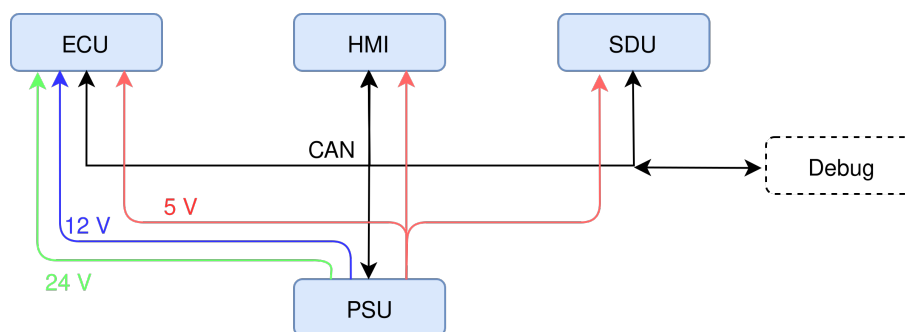
V nové konstrukci je vyžadováno použití kvalitních konektorů s mechanickou pojistkou proti vytržení. Zvolené konektory by neměly umožňovat vzájemnou záměnu, respektive v případě prohození konektorů při zapojování elektroniky by nemělo dojít k jejímu poškození.

Po analýze dostupných možností bylo rozhodnuto sdružit jednotlivé senzory a výstupy s podobnou funkcionalitou do společných konektorů. Byly zvoleny konektory DSUB (v provedení DE-9 a DA-15), které jsou často používány v průmyslovém prostředí a odolnější varianty se objevují i v zařízeních určených pro vesmírné sondy. Místo běžnějšího mechanického zámku je spoj proti rozpojení zajištěn dvojicí šroubů. Výkonové prvky budou připojeny přes průmyslové konektory PSG se šroubovacím zámkem.

Na deskách plošných spojů budou použity konektory MC-254 firmy Multicomp s mechanickým zámkem, konektory pro vyšší zátěž budou řešeny zasouvací svorkovnicí firmy Phoenix. Propojovací konektory budou krimpovací, zbytek konektorů bude pájecí.

## 2.2 Blokový návrh

Řídicí systém bude rozdělena do několika nezávislých bloků, které budou komunikovat po sběrnici CAN. Kriticky důležité řízení pneumatického kruhu tak bude odděleno od méně důležitého modulu pro interakci s uživatelem. Zároveň bude možné v případě modifikace systému řízení pneumatiky zachovat část elektroniky a navrhnout znovu pouze nezbytné části. Další výhodou je jednoduchá a rychlá výměna nefunkční desky během závodů. Blokové schéma řídicí elektroniky je na obrázku 2.2.



Obr. 2.2: Blokové schéma elektroniky

Jednotlivé navrhované bloky a jejich funkce jsou:

- **ECU** (Engine control unit) – řízení pneumatiky a sběr dat z tlakoměrů a ostatních senzorů.
- **HMI** (Human machine interface) – interakce s uživatelem, ukládání a odesílání telemetrie.
- **PSU** (Power supply unit) – napájecí zdroj.
- **SDU** (Speed display unit) – rychloměr v kapotě vozu.

Sběrnice CAN bude mimo samotné komunikace mezi moduly sloužit i k diagnostice a případně i programování připojených zařízení. S vhodnou aplikací pro počítač bude možné pomocí převodníku ze sběrnice USB na CAN konfigurovat jednotlivé moduly, vyčítat hodnoty interních proměnných a čist chybové hlásky.

### 2.2.1 Distribuce napájení

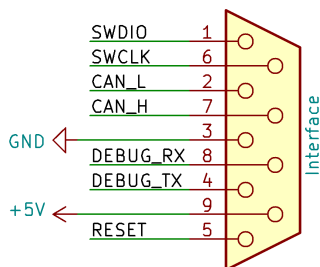
Napájecí napětí doporučené firmou Aventics je 24 V. Tímto napětím jsou napájeny pneumatické ventily a tlakoměry. Většina ostatních senzorů vyžaduje napětí 12 V (indukční senzory, inkrementální enkodér,...). MCU a přidružené obvody fungují při napětí 5 V, případně 3,3 V.

Jak ukazuje blokové schéma na obrázku 2.2, napětí 12 a 24 V využívá pouze modul ECU, který sdružuje veškeré vysokonapěťové prvky. 5 V větev sdílejí všechny moduly.

Napájecí zdroj bude umístěn v montážní krabici společně s ECU. Modul SDU bude umístěn v karoserii vozu a HMI na palubní desce vozidla. Napájení a datová sběrnice bude přivedena k HMI a SDU kabelovým svazkem.

### 2.2.2 Univerzální komunikační a napájecí konektor

Jednotlivé moduly budou propojeny pomocí devítipinových konektorů DSUB DE-9. Na konektor je vyvedeno napájecí napětí (zem a 5 V) a komunikační sběrnice CAN. Pro usnadnění ladění programu jsou na zbývajících pinech signály ladicího rozhraní SWD (SWDIO, SWCLK) a sériové rozhraní UART (pokud jím modul disponuje). Přesné zapojení je na obrázku 2.3.

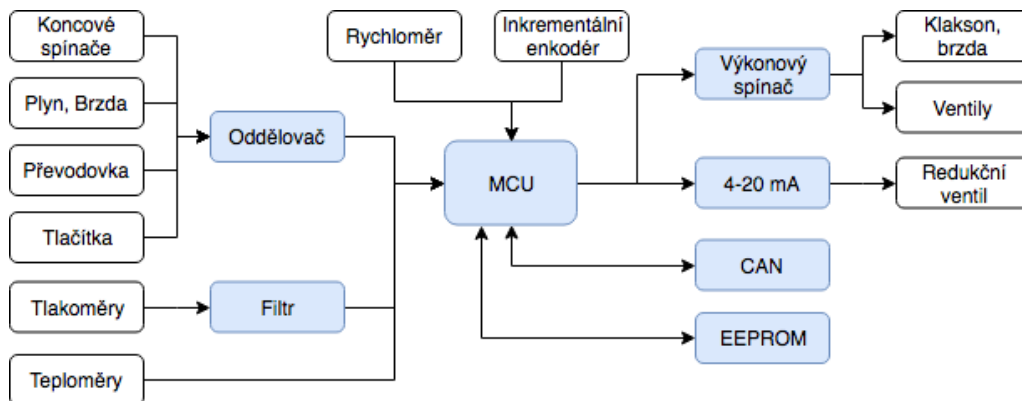


Obr. 2.3: Zapojení univerzálního komunikačního konektoru

Při propojování modulů budou použity pouze 4 vodiče nezbytné pro funkci (napájení a CAN), rozhraní SWD není vhodné z důvodu rizika přivedení rušení na MCU připojovat dlouhým vodičem.

### 2.3 Jednotka řízení motoru (ECU)

Jednotka ECU (Engine Control Unit) zajišťuje řízení pneumatiky a sběr dat ze senzorů. Blokové schéma je na obrázku 2.4.



Obr. 2.4: Blokové schéma modulu ECU

Srdcem zapojení je mikroprocesor STM32F072RB s jádrem Cortex M0 o pracovní frekvenci až 48 MHz. Pro ukládání konfigurace je k procesoru připojena I2C EEPROM paměť M24C64 o kapacitě 8 kB. Komunikaci s okolím zajišťuje CAN transceiver MCP2562 a rozhraní UART (určené k ladění programu). Pro případ budoucí potřeby byla do zapojení přidána i sběrnice USB (např. pro přehrávání firmware).

### 2.3.1 Digitální vstupy

Magnetické senzory krajních poloh pístu pracují při napětí 12 V, stejně tak indukční senzory využívané na snímání zařazeného rychlostního stupně, případně i na detekci sešlápnutého plynového a brzdového pedálu. Aby toto napětí nepoškodilo MCU s pracovním napětím 3,3 V, je každý vstup přiveden na procesor přes obvod 74HC4050, což je šestinásobný buffer s tolerancí vstupního napětí až do 15 V. Na jeho výstupu je pak maximálně napájecí napětí 3,3 V.

Podle vzoru průmyslových zařízení je ke každému vstupu připojena LED dioda signalizující jeho aktuální stav. Pro ochranu MCU při chybné konfiguraci pinu je obvod 74HC4050 od procesoru oddělen rezistorem o hodnotě 10 k $\Omega$ , který omezí maximální proud na 330  $\mu$ A.

Frekvence spínání se bude pohybovat maximálně v jednotkách Hz, odstranění případných zámkitů bude provedeno programově. Pro případné budoucí rozšiřování funkcionality obsahuje návrh 2 nevyužité vstupy navíc.

Relativní poloha pístu je snímána inkrementálním enkodérem, který disponuje výstupy s otevřeným kolektorem. Pro odstranění případného zakmitávání výstupů při přechodu mezi jednotlivými stavy byl výstup enkodéru doplněn volitelným RC filtrem. Hodnota součástek v tomto filtru bude v případě osazení určena dle reálně změřené frekvence zakmitávání a maximální rychlost pístu.

### 2.3.2 Měření rychlosti

Rychlost vozu je měřena optickou bránou, která snímá pohyb děrované šablony pevně spojené se zadním kolem. Rychlost je přímo úměrná počtu pulzů za časový interval a průměru kola a nepřímo úměrná počtu otvorů v šabloně. Přesnou hodnotu lze spočítat jako:

$$v = \frac{2\pi r}{ot} p \quad (2.1)$$

Kde  $v$  je rychlost,  $r$  poloměr kola,  $o$  počet otvorů v šabloně a  $p$  počet pulzů za interval  $t$ .

### 2.3.3 Měření tlaku

Na pneumobilu jsou umístěny tři tlakoměry R412010773 firmy Aventics napájené napětím 24 V. Výstupní napětí 0–10 V tlakoměru je přímo úměrné tlaku 0–10 Bar. Toto napětí je snižováno na hodnotu 3,3 V pro MCU odporovým děličem, který zároveň slouží jako část RC antialiasingového filtru. Zapojení na obrázku 2.5 má přenos:

$$K = \frac{U_{out}}{U_{in}} = \frac{\frac{\frac{R_2}{pC}}{R_2 + \frac{1}{pC}}}{R_1 + \frac{\frac{R_2}{pC}}{R_2 + \frac{1}{pC}}} = \frac{R_2}{pR_1R_2C + R_1 + R_2} \quad (2.2)$$

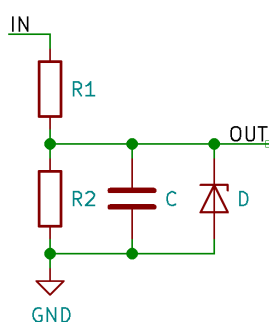


Pro mezní frekvenci pak platí:

$$0,5 \frac{R_2}{R_1 + R_2} = \frac{R_2}{pR_1R_2C + R_1 + R_2} \quad (2.3)$$

$$f_m = \frac{R_1 + R_2}{2\pi R_1R_2C} \quad (2.4)$$

Byly zvoleny rezistory o hodnotách 6,8 kΩ a 3,3 kΩ, při kterých bude příčný proud děličem maximálně 1 mA. Rychlost změn tlaku se pohybuje v desítkách Hz, s rezervou byla zvolena mezní frekvence filtru 100 Hz, čemuž odpovídá kondenzátor 680 nF. Podle Nyquistova teorému je nutné vzorkovat signál s minimálně dvojnásobnou frekvencí, než je maximální frekvence v signálu se vyskytující. Jako vhodné se jeví vzorkování na frekvenci 1 kHz, při které je útlum filtru 10 dB.



Obr. 2.5: Vstupní filtr tlakoměru

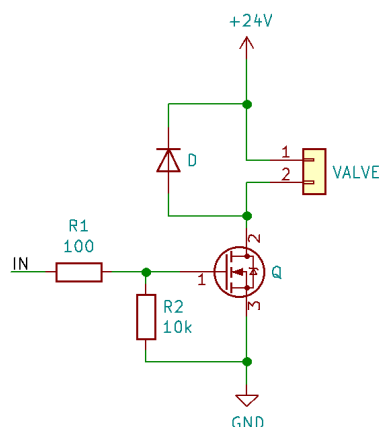
Zenerova dioda 3,3 V na výstupu slouží jako ochrana mikroprocesoru v případě, že na vstup filtru bude připojeno vyšší napětí než 10 V (např. při zkratu v konektoru). MCU má na každém pinu diody, které omezí vstupní napětí na hodnotu napětí napájecího plus úbytek na diodě. Bohužel proudová zatížitelnost této diody je poměrně nízká a není přesněji specifikována. Při zkratu na 24 V by diodou tekla proud téměř 2 mA, proto je vhodnější použít externí diodu s vyšší proudovou zatížitelností, která se otevře dříve než dioda v MCU.

Pro možnost budoucího rozšiřování obsahuje modul ECU celkem 6 analogových vstupů.

### 2.3.4 Měření teploty

Pro podrobnější analýzu chování pneumatického okruhu při různé zátěži je mimo aktuálního tlaku nutné znát i teplotu na kritických místech okruhu (válec, vzdušník). Pro tento účel byl zvolen teploměr DS18B20, který je dodáván i v průmyslovém pouzdru vhodném k přímé instalaci do čel pneumatického válce.

Teploměr komunikuje po jednodrátové sběrnici Dallas 1-Wire. Zvolený MCU touto periferií nedisponuje, proto bude tento protokol nutné implementovat programově.



Obr. 2.6: Spínání ventilů

### 2.3.5 Výkonové výstupy

Ventily jsou ovládány tranzistory MOSFET, klidový příkon ventilu dle dokumentace je 2,1 W, což při napájecím napětí 24 V odpovídá proudu přibližně 90 mA. V zapojení dle obrázku 2.6 rezistor R2 definuje napětí na hradle v případě, kdy je daný pin mikroprocesoru nastaven jako vstup. Rezistor R1 omezuje proud do kapacity hradla a chrání tak MCU před poškozením nadproudem.

Ventil pracuje na elektromagnetickém principu, uvnitř se nachází cívka, která při průchodu proudu přitáhne kotvu připojenou na mechanický ventil. Tranzistor tedy spíná indukční zátěž. Dioda D omezuje napěťové špičky vznikající na cívce při rozepnutí tranzistoru.

Zvolený tranzistor IPD30N06S4L23 má v sepnutém stavu při napětí 3,3 V odpor kanálu 30 mΩ, celková výkonová ztráta v ustáleném stavu je méně jak 1 mW, tedy naprosto zanedbatelná. Pro rychlou diagnostiku je každý z výstupů doplněn LED diodou indikující jeho aktuální stav.

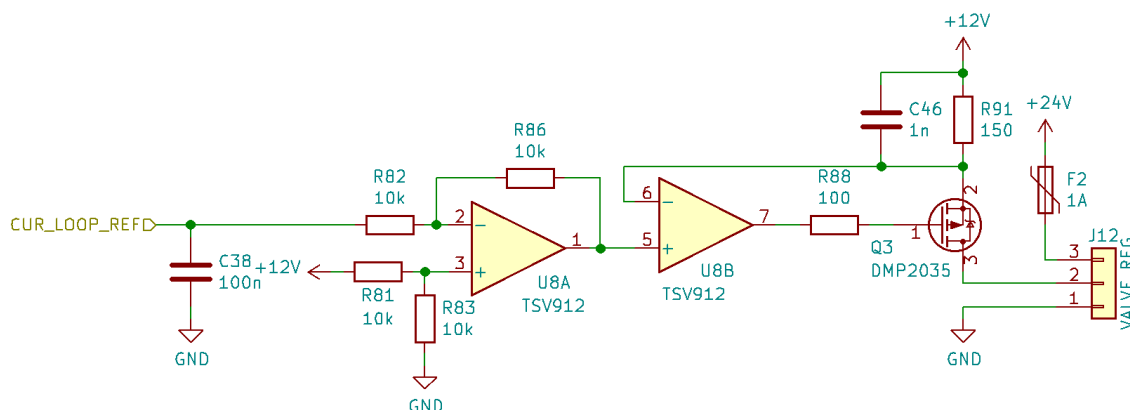
Pro snížení variability použitých součástek stejné zapojení realizuje i spínání brzdového světla a klaksonu. Pro případ budoucího rozšiřování poskytuje ECU další 2 výkonové výstupy, nyní nevyužité.

### 2.3.6 Proudová smyčka 4–20 mA

Elektronický redukční ventil je řízen proudovou smyčkou 4–20 mA, což je standardní průmyslové řešení.

Zapojení na obrázku 2.7 využívá operačního zesilovače U8B jako zpětnovazebního regulátoru, který udržuje na rezistoru R91 v proudové smyčce konstantní úbytek napětí a tedy i konstantní proud proudovou smyčkou. Napětí na neinvertujícím vstupu určuje přesnou hodnotu proudu. Použitý redukční ventil má záporný výstup proudové smyčky spojen s napájecí zemí, proto je použito zapojení s tranzistorem PMOS (zdroj typu source).

Nevýhodou tohoto řešení je vztažení referenčního napětí zesilovače U8B k napětí 12 V, nikoliv k zemi.



Obr. 2.7: Zapojení proudové smyčky

Operační zesilovač U8A v rozdílovém zapojení odečítá napětí generované DA převodníkem MCU od napětí 12 V, jinými slovy převádí vstupní napětí vztažené k zemi na napětí vztažené k hodnotě 12 V. Změnou napětí na výstupu MCU tak lze řídit proud smyčkou v celém požadovaném rozsahu.

### 2.3.7 Napájení

Modul ECU vyžaduje napájecí napětí 24, 12, 5 a 3,3 V. Mimo 3,3 V jsou všechna napětí poskytována modulem PSU. Napětí 3,3 V je tvořeno přímo v modulu ECU z 5 V větve lineárním stabilizátorem (LDO) TS1117BCW33. Maximální možný odběr 200 mA (MCU, ostatní integrované obvody, indikační LED) vytvoří na LDO výkonovou ztrátu 340 mW.

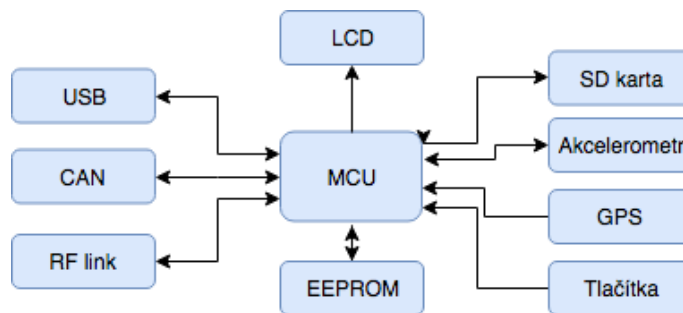
Napájení 5 V je možné připojit nejen z PSU, ale i z ladicího rozhraní nebo z USB portu. Aby v případě současného připojení několika zdrojů nedošlo k jejich poškození je každý vstup oddělen od obvodu Schottkyho diodou.

Maximální proudový odběr jednotlivých větví je:

- 3,3 V – 200 mA (MCU, indikační LED, ostatní integrované obvody)
- 5 V – 220 mA (3,3 V větev, indikační LED)
- 12 V – 600 mA (6 tlakoměrů, rotační enkodér, brzdové světlo, proudová smyčka, indukční snímače)
- 24 V – 600 mA (4 pneumatické ventily, redukční ventil)

## 2.4 Jednotka telemetrie a interakce s uživatelem (HMI)

Jednotka HMI (Human Machine Interface) slouží primárně k zobrazování diagnostických údajů a nastavování provozních parametrů. Sekundárním úkolem je ukládání dat na SD kartu a odesílání telemetrie do počítače. Blokové schéma modulu je na obrázku 2.8.



Obr. 2.8: Blokové schéma modulu HMI

Desku řídí mikroprocesor STM32F412VE s jádrem Cortex M4 o pracovní frekvenci až 100 MHz. Hlavním kritériem při výběru byla přítomnost periferie FSMC (Flexible static memory controller), která umožňuje namapování displeje do paměťového prostoru mikroprocesoru a tím i vysokou rychlost vykreslování. Druhým kritériem byl dostatek programové paměti k uložení potřebných obrazových dat (bitmap).

Obdobně jako u modulu ECU je i HMI vybaven rozhraním CAN a pamětí EEPROM. SD karta je připojena rychlým rozhraním SDIO. Pro přenos dat z karty na USB flash disk je osazen i USB konektor.

Interakci s uživatelem zajišťuje grafický dotykový displej o úhlopříčce 5 palců s řadičem displeje SSD1963 a řadičem rezistivní dotykové plochy XPT2046. Možnosti ovládání doplňuje šestice tlačítek umístěná pod displejem. Podrobnější analýzu výkonů vozidla umožňuje akcelerometr LSM303C a GPS přijímač NEO-6M (umístěný mimo desku).

Bezdrátové spojení s počítačem je realizováno modulem HopeRF RFM69 pracujícím na frekvenci 868 MHz. Pro zajištění maximálního dosahu je anténní výstup přiveden na konektor SMA, na který bude připojena externí anténa.

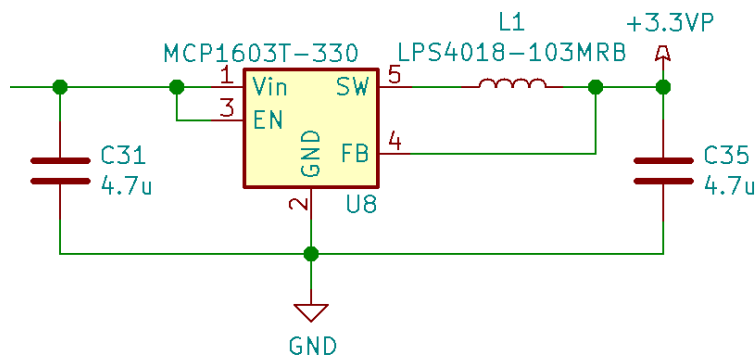
### 2.4.1 Napájení

Celkový maximální odběr všech modulů napájených z 3,3 V větve přesahuje 500 mA, což při napájecím napětí 5 V odpovídá výkonové ztrátě na regulátoru 850 mW. Lineární regulátor v pouzdru SOT-223 by se již výrazněji zahříval a bylo by nutné připojit externí chladič, respektive jako chladič využít zemnicí plochu na DPS.

Výhodnější je použít spínaný snižující měnič, na kterém bude výkonová ztráta minimální, ovšem za cenu vyšší složitosti konstrukce a možného zvýšení vyzařování elektromagnetického rušení do okolí. Měnič bude realizován obvodem MCP1603T, který pracuje na frekvenci 2 MHz (obrázek 2.9). Vysoká spínací frekvence umožňuje použít cívku o malé indukčnosti, tedy i menším ohmickém odporu. Díky tomu je celé zapojení regulátoru prostorově nenáročné a dosahuje v oblasti uvažovaných proudů vysoké účinnosti přes 85 %.

S induktorem 10  $\mu\text{H}$  bude zvolnění výstupního proudu dle [8] rovno:

$$\Delta I_L = \frac{V_{\text{out}}}{f \cdot L} \left( 1 - \frac{V_{\text{out}}}{V_{\text{in}}} \right) = \frac{3,3}{2 \cdot 10^6 \cdot 10^{-5}} \left( 1 - \frac{3,3}{5} \right) = 56 \text{ mA} \quad (2.5)$$



Obr. 2.9: Zapojení spínaného zdroje s obvodem MCP1603T

To je méně než doporučených maximálně 20 %. S kondenzátorem 10  $\mu\text{F}$  na výstupu bude maximální zvlnění výstupního napětí při filtračním keramickém kondenzátoru s ESR 200 m $\Omega$ :

$$\Delta U_{\text{out}} = \Delta I_L \left( ESR + \frac{1}{8fC} \right) = 0,056 \left( 0,2 + \frac{1}{8 \cdot 2 \cdot 10^6 \cdot 4,7 \cdot 10^{-6}} \right) = 12 \text{ mV} \quad (2.6)$$

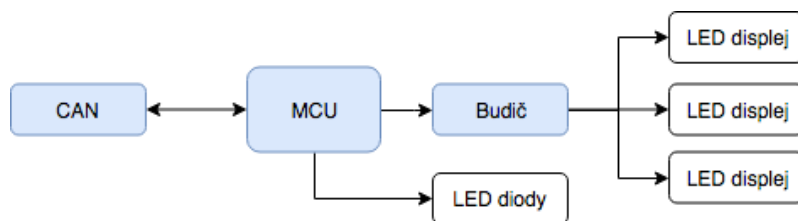
Celková maximální spotřeba modulu je:

- 3,3 V – 550 mA (MCU, SD karta, LCD logika, RFM69)
- 5 V – 1700 mA (3,3 V větev, LCD, USB zařízení)

Nutno podotknout, že tento proudový odběr je maximální možný, velkou část tvoří podsvícení displeje a maximální napájecí proud pro případné připojené USB zařízení, průměrný odběr bude přibližně poloviční.

## 2.5 Jednotka zobrazovače rychlosti vozu (SDU)

SDU (Speed Display Unit) je jednotka zobrazující aktuální, popř. průměrnou rychlost vozu. Blokové schéma je na obrázku 2.10. Jádrem je mikroprocesor STM32F042, který mimo komunikace po sběrnici CAN zajišťuje již jen multiplexové řízení tří sedmisegmentových LED displejů. Jednotlivé segmenty displeje se společnou anodou jsou buzeny obvodem 74LS47, což je převodník z BCD na sedmisegmentový displej s výkonovými výstupy s otevřeným kolektorem.



Obr. 2.10: Blokové schéma modulu SDU

Mimo aktuální rychlosti modul zobrazuje také aktuálně zařazení převodový stupeň pomocí trojice LED (neutrál, 1, 2). Obdobně jako u modulu ECU je napětí pro MCU

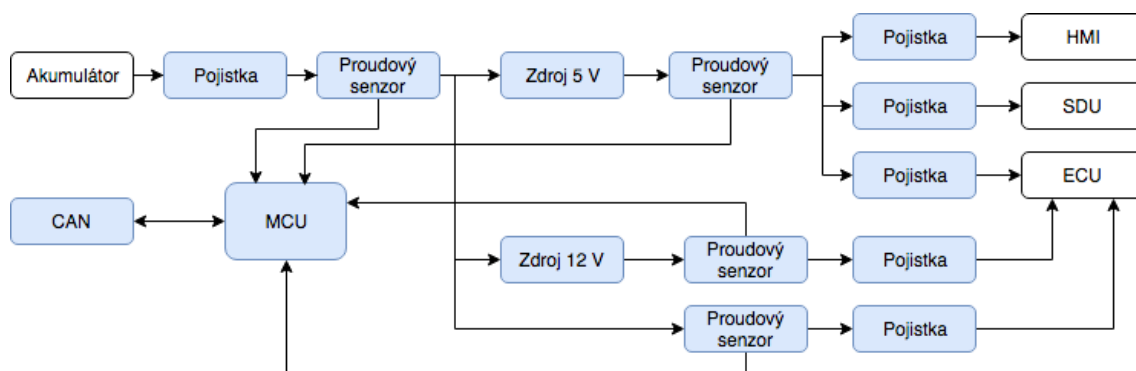
tvořeno lineárním stabilizátorem TS1117BCW33. Modul samotný je napájen napětím 5 V. Maximální proudový odběr jednotlivých větví je:

- 3,3 V – 60 mA (MCU, indikační LED)
- 5 V – 140 mA (3,3 V větev, LED displeje)

Tento modul bude osazen pouze na voze Falcon, který má v karoserii výřez odpovídajících rozměrů. V případě instalace stejné elektroniky i do vozu Javelin bude rychlost zobrazována na dotykovém displeji jednotky HMI, který bude umístěn na volantu.

## 2.6 Napájecí zdroj (PSU)

Modul PSU (Power Supply Unit) zajišťuje napájení všech ostatních částí. Blokové schéma je na obrázku 2.11. Procesor STM32F042 slouží k měření proudů a napětí jednotlivých větví. Naměřená data jsou odesílána po sběrnici CAN do ostatních modulů pro případnou analýzu.



Obr. 2.11: Blokové schéma modulu PSU

Vůz je napájen z dvojice modelářských tříčlánkových lithium polymerových akumulátorů s nominálním napětím 11,1 V (v plně nabitém stavu 12,6 V). V sériovém zapojení poskytují akumulátory nominální napětí 22,2 V. Ventily vyžadují napětí 24 V, nicméně i při nižším napětí pracují naprosto spolehlivě, není tedy nutné napětí akumulátorů zvyšovat.

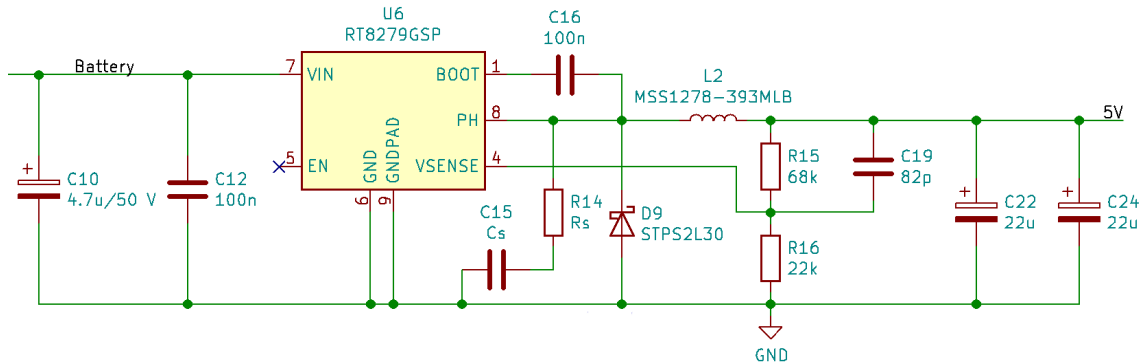
Jednotlivé napájecí větve mají celkový maximální odběr (daný součtem odběrů ostatních modulů):

- 5 V – 2000 mA
- 12 V – 600 mA
- 24 V – 600 mA

### 2.6.1 Spínané zdroje

Maximální odběr z akumulátoru je 3200 mA. Při těchto proudech by se výkonová ztráta lineárních stabilizátorech pohybovala v desítkách wattů. Tak vysoký ztrátový výkon by

vyžadoval masivní chladič a využití energie akumulátoru by bylo značně neefektivní. Z tohoto důvodu jsou napětí 12 a 5 V generována spínanými snižujícími měniči RT8279, které pracují na frekvenci 500 kHz. Schéma zapojení 5 V větve je na obrázku 2.12.



Obr. 2.12: Schéma 5 V větve napájecího zdroje

Výstupní napětí spínaného zdroje je určeno dělicím poměrem odporového děliče ve zpětné vazbě zdroje. Referenční napětí obvodu je 1,222 V. Pro 5 V větev byly zvoleny rezistory 68 a 22 kΩ, což dává výstupní napětí přesně 5 V. U 12 V větve nelze z řady E12 sestavit odporový dělič s přesným dělicím poměrem, nejbližší hodnoty 33 a 3,9 kΩ odpovídají výstupnímu napětí 11,6 V. Použití řady E24 by znamenalo zvětšení součástkové základny a v případě poškození rezistoru komplikovanější shánění nového během závodů. Vzhledem k povaze napájených spotřebičů a širokému rozmezí napájecích napětí je tato odchylka akceptovatelná.

Minimální indukčnost pro zvlnění do 20 % maximálního výstupního proudu lze vyjádřit dle [8] jako:

$$L_{\min} = \frac{V_{\text{out}}}{f \Delta I_{L_{\max}}} \left( 1 - \frac{V_{\text{out}}}{V_{\text{IN}_{\max}}} \right) = \frac{5}{500 \cdot 10^3 \cdot 0,2 \cdot 2} \left( 1 - \frac{5}{25,2} \right) = 20 \mu\text{H} \quad (2.7)$$

Zvolený induktor MSS1278-393MLB ve stíněném provedení má indukčnost 39 µH, výsledné proudové zvlnění s tímto induktorem bude 206 mA. Napětové zvlnění je dáno kapacitou na výstupu a ESR použitého kondenzátoru. Pro snížení zvlnění jsou na výstup připojeny paralelně dva kondenzátory o ESR 200 mΩ. Výsledné ESR je tedy poloviční, což dává pro dva kondenzátory s kapacitou 22 µF zvlnění:

$$\Delta V_{\text{out}} = \Delta I_L \left( ESR + \frac{1}{8fC_{\text{out}}} \right) = 0,206 \left( 0,1 + \frac{1}{8 \cdot 500 \cdot 10^3 \cdot 2 \cdot 22 \cdot 10^{-6}} \right) = 21 \text{ mV} \quad (2.8)$$

Jak je patrné z výše uvedeného vzorce, minimální hodnota zvlnění je dána primárně ESR použitých kondenzátorů, pro její snížení by bylo nutné použít výrazně dražší kondenzátory s nižším ESR, nebo zapojit více běžných paralelně. Hodnota 21 mV je nicméně přijatelná.

Maximální výkonová ztráta na cívce při proudu 2 A bude (odpor cívky 80 mΩ) 320 mW. Ztráty na spínaném měniči jsou dány odporem interního spínače v sepnutém

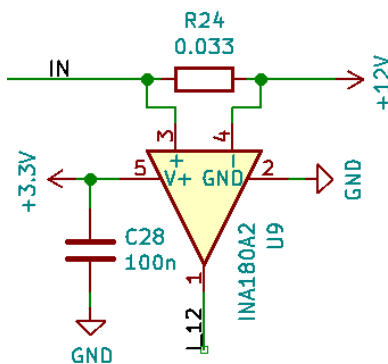
stavu, střídou spínání a spínacími ztrátami. Spínací ztráty nelze z údajů v katalogovém listu počítat. Ztráty na interním spínači s odporem v sepnutém stavu  $110\text{ m}\Omega$  při 100 % střídě jsou  $440\text{ mW}$ . Maximální ztrátový výkon s minimální velikostí zemní plochy pod obvodem je  $1,3\text{ W}$ , výkonová rezerva je tedy dostačující. Pro snížení teploty měniče bude při návrhu plošného spoje maximalizována zemní plocha pod obvodem. Výkonová ztráta na Schottkyho diodě s úbytkem  $0,375\text{ V}$  při proudu  $2\text{ A}$  je  $750\text{ mW}$ . Obdobně jako u spínacího prvku bude reální ztráta nižší podle aktuální střídavy spínání. Uvedené spínací ztráty platí pro maximální možný proud  $2\text{ A}$ , který v praxi téměř nemůže nastat, průměrný odběr se bude pohybovat zhruba na poloviční hodnotě.

Obdobně pro  $12\text{ V}$  větev lze určit minimální hodnotu induktoru  $100\text{ }\mu\text{H}$ , zvolený induktor SRU1048-101Y ve stíněném má tutéž hodnotu a stejnosměrný odpor  $160\text{ m}\Omega$ . Při použití stejných výstupných kondenzátorů vychází maximální zvlnění napětí na výstupu  $13\text{ mV}$ . Na induktoru bude výkonová ztráta  $58\text{ mW}$ , na měniči  $40\text{ mW}$  a na diodě  $230\text{ mW}$ .

### 2.6.2 Měření proudů a napětí

Schéma zapojení pro měření proudu v jednotlivých větvích je na obrázku 2.13. Úbytek na snímacím odporu je zesílen 50 krát přesným diferenciálním zesilovačem INA180. Zesílené napětí je přivedeno na AD převodník MCU.

Na rezistoru o hodnotě  $3\text{ m}\Omega$  vzniká při maximálním proudu  $2\text{ A}$  úbytek napětí  $66\text{ mV}$ . Při padesátinásobném zesílení bude na výstupu zesilovače napětí  $3,3\text{ V}$ . Výstup zesilovače je od MCU oddělen ochranným rezistorem  $10\text{ k}\Omega$ . Maximální výkonová ztráta na rezistoru je  $130\text{ mW}$ , hluboko pod maximálním zatížením  $500\text{ mW}$ .



Obr. 2.13: Zapojení pro měření proudu

Napětí jednotlivých větví je měřeno stejným způsobem jako napětí na výstupu tlakoměrů.

### 2.6.3 Proudová ochrana a ochrana akumulátoru

Na všechny napájecí výstupy jsou zapojeny vratné polymerové pojistky, které při přetížení omezí proud do problematického modulu na minimální hodnotu. Nevýhodou těchto pojistek je pomalá reakce na nadproudové stavy – v desítkách až stovkách milisekund.



Rychlou zkratovou ochranu zajišťuje na napětích 5 a 12 V samotný snižující měnič, který při proudech nad 6 A vypíná.

Ochrana akumulátoru proti nadměrnému vybití není řešena záměrně. Závod je nutné dokončit i za cenu trvalého poškození akumulátorů. Akumulátor se dá koupit nový, závod opakovat nelze. Částečně lze zajistit šetření energií při vybitém stavu snížením jasu displeje v modulu HMI, v nouzovém případě softwarovým vypnutím všech nekritických modulů (HMI, SDU).

## **2.7 Bezdrátový komunikační modul pro PC**

Přenos telemetrických dat do počítače zajišťuje převodník mezi bezdrátovým modulem HopeRF RFM69 a sběrnici USB. Převodník je napájen z počítače přes rozhraní USB, které poskytuje napětí 5 V. Toto napětí je sníženo na pracovní hodnotu MCU 3,3 V lineárním stabilizátorem TS1117BCW33.

Jádrem převodníku je mikroprocesor STM32F042 s integrovaným rozhraním USB. Výhodou je možnost synchronizace interních hodin MCU s USB bez nutnosti použít přesný externí krystal. Komunikace s bezdrátovým modulem probíhá přes rozhraní SPI. Obvod je doplněn od dvě LED diody k indikaci příchozí a odchozí bezdrátové komunikace.

### 3 Komunikační protokol

Pro komunikaci mezi jednotlivými moduly, případně mezi moduly a počítačem, je nutné zvolit vhodný komunikační protokol. Ten musí umožňovat jak pravidelné zasílání zpráv o stavu vozidla, tak nepravidelné odesílání příkazů a ladicích informací. Výhodou je i možnost určovat prioritu přenášených zpráv, ať už pro přenos po komunikační sběrnici, nebo pro zpracování v přijímači.

Při návrhu byl protokol rozdělen na dvě části: nízkourovňový, který řeší přenos dat po sběrnici bez znalosti jejich obsahu a vysokoúrovňový, který obsahuje informaci o typu zprávy, parametrech, atd.

#### 3.1 Nízkourovňový protokol pro sběrnici CAN

Jednotlivé moduly jsou propojené automobilovou sběrnici CAN. Nabízí se tedy velké množství existujících CAN protokolů navržených pro komunikaci v automobilech, letadlech a dalších kritických systémech. Např. CANopen, CANaerospace, MilCAN, atd. Nevýhodou těchto řešení je vysoká komplexnost a snížená propustnost díky většímu množství přenášených metadat.

Z těchto důvodů byl navržen vlastní komunikační protokol. Při návrhu byl kladen důraz na maximalizaci množství dat přenášených v jednom datovém rámci CAN a na jednoduchost implementace.

Komunikace po sběrnici CAN je realizována pomocí datových rámců, jejichž strukturu zobrazuje tabulka 3.1. Formát rámců je pevně dán a jeho implementace je zajištěna přímo v periférii MCU, nelze tedy měnit.

Tab. 3.1: Struktura datového rámce sběrnice CAN [5]

Název	Délka [bitů]	Popis
Start-of-frame	1	Označení začátku rámce
Identifikátor (ID)	11	Identifikátor rámce, určuje též jeho prioritu
RTR	1	Logická 0 určuje datový rámec, 1 žádost o zprávu
IDE	1	Logická 0 určuje standardní délku ID
rezervováno	1	Musí být logická 0
Délka dat (DLC)	4	Počet datových bytů
Data	0–64	0–8 bytů dat
CRC	15	Kontrolní součet
Oddělovač	1	Musí být logická 1
ACK	1	Logická 0 znamená potvrzení přijetí zprávy
Oddělovač	1	Musí být logická 1
End-of-frame	1	Konec zprávy

### 3.1.1 Identifikátor rámce

ID přednášené zprávy lze s výhodou využít k přenosu adresy zdrojového a cílového modulu. V případě kolize při odesílání rámce více moduly současně dostává přednost ten, který dříve odešle dominantní bit (0). Identifikátor tedy určuje i priority zprávy. Vhodným návrhem lze i na přetížené sběrnici dosáhnout spolehlivého a rychlého přenosu důležitých zpráv.

Zvolený formát identifikátoru popisuje tabulka 3.2. První pole určuje priority zprávy (příkazy, stav vozidla, ladicí výpis) nezávisle na zdrojovém a cílovém modulu. Následuje identifikátor zdrojového zařízení, čím nižší číslo, tím vyšší prioritu zařízení má. Nejvyšší priorita je rezervována pro počítač (ladění, programování), následuje PSU (zprávy o výpadku napájení, vybitém akumulátoru, příliš vysokém proudu,...), HMI (příkazy pro ostatní moduly), ECU (stav vozidla) a SDU, v tomto pořadí.

Jako poslední je odeslána identifikace cílového zařízení. Zvláštním případem cílového zařízení je adresa 0x07, tzv. broadcast, tedy vysílání pro všechna zařízení. Poslední bit určuje, zda je rámec samostatný, nebo zda se jedná o část delší zprávy rozdělené do více rámců.

Tab. 3.2: Formát identifikátoru zprávy

Název	Délka [bitů]	Popis
Priorita	2	Priorita zprávy od 0 (kritická zpráva) po 3 (ladicí výpis)
Zdroj	4	Identifikátor odesílatele
Cíl	4	Identifikátor cílového zařízení
Multipart	1	Identifikace zprávy složené z více částí

### 3.1.2 Odesílání dlouhých zpráv

Je-li odesílaná zpráva dlouhá nejvýše 8 bytů, je možné celý její obsah přenést v jediném rámci. Informaci o délce a kontrolní součet není nutné přenášet, jsou již obsaženy v samotném CAN rámci.

Problém nastává u zpráv delších než 8 bytů. Tyto je nutné skládat z více rámců. Jednotlivé bloky zprávy mají nastavený bit *multipart* do logické 1. První byte rámce obsahuje pořadové číslo, první rámec má toto číslo nulové, s každým odeslaným rámcem se pořadové číslo zvětšuje o jedničku.

První rámec obsahuje mimo samotného pořadového bytu také jeden byte s celkovou délkou odesílaných dat (mimo pořadová čísla a tento byte). Poslední rámec končí kontrolním součtem pro ověření, že zpráva došla kompletní a bez poškození. Odesílání dlouhé zprávy je ilustrováno v tabulce 3.3.

Tab. 3.3: Příklad odesílání dlouhé zprávy, jednotlivé rámce jsou odlišeny barevně

Bit	0	1	2-7	8	9-15	16	17-19	20
Obsah	0x00	0x10 (Délka)	Data	0x01	Data	0x02	Data	CRC

## 3.2 Nízkoúrovňový protokol pro sériové rozhraní

Pro ladění je vhodné využít sériové rozhraní UART, které nevyžaduje specializovaný převodník mezi sběrnici USB a CAN. Převodník z USB na UART je výrazně běžnější a levnější.

Začátek přenosu je označen „magickým“ číslem 0xff, poté následuje délka přenášených dat, samotná data a nakonec kontrolní součet pro ověření správnosti přenesených dat. Názorně je formát ilustrován tabulkou 3.4.

Tab. 3.4: Formát zprávy na sériovém rozhraní

Byte	0	1	2 – (n - 1)	n
Obsah	0xff	Délka	Data	CRC

## 3.3 Vysokoúrovňový komunikační protokol

Data přijatá některým z dříve uvedených způsobů mají stejný formát, nezávislý na rozhraní. První byte obsahuje příkaz, který se má provést. Zároveň jsou v tomto bytu obsaženy informace, zda je zpráva příkazem, nebo odpovědí na příkaz a zda došlo při zpracování k chybě. Formát příkazu detailně popisuje tabulka 3.5. Zbývající byty obsahují data podle konkrétního příkazu.

Tab. 3.5: Formát příkazu

Bit	7	6	5-0
Obsah	Odpověď na příkaz	Chyba	Příkaz

Formát konkrétního příkazu ilustruje příklad v tabulce 3.6. Příjemce na takovou zprávu odpoví odesláním dvou bajtů. První byte obsahuje stejný příkaz a nastavený bit 7 (odpověď na příkaz) do 1. Pokud došlo ke zpracování k chybě, je nastaven i bit číslo 6. V druhém byte je pak uložena nulová hodnota v případě korektního zpracování, nebo číslo chyby.

Pokud je vyžadována složitější odpověď, příjemce odešle místo dvou bytů větší množství dat a chybovou hodnotu v případě korektního zpracování neodesílá. V případě, že dojde k chybě, nastaví odesílatel chybový příznak a místo dat odešle chybovou hodnotu jako v předchozím případě.

Tab. 3.6: Demonstrační příklad nastavení parametrů závodního módu

Byte	0	1	2-3	4-5
Obsah	Příkaz (tab. 3.5)	Závodní režim	Procento plnění	Čas v úvrati

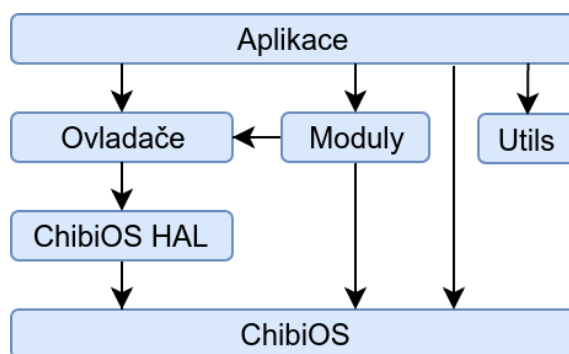
### 3.4 Bezdrátová komunikace

Data mezi moduly RFM69 jsou přenášena jako packety o maximální délce 64 bytů. Na základě konfigurace modulů lze navíc packety filtrovat podle identifikátoru sítě a identifikátoru cílového zařízení. V aktuálně navrhovaném systému budou bezdrátově přenášena pouze telemetrická data a ladicí hlášky z jednotlivých modulů. Formát přenášených dat vychází z dříve popisovaného vysokoúrovňového protokolu, rozšířením je přidání bytu s identifikátorem zdrojového zařízení na začátek zprávy. Komunikace je pouze jednosměrná a zprávy nejsou nijak potvrzovány.

Do budoucna je možné po vyřešení zabezpečení přenosu bez zásahu do protokolu přidat možnost oboustranné komunikace a bezdrátové spojení umístit na stejnou úroveň jako sběrnici CAN a sériové rozhraní. V tomto případě první byte odeslaný z počítače do řídicí elektroniky určuje modul, pro který je zpráva určena.

## 4 Programové vybavení

Díky volbě rodiny MCU STM32 a operačního systému reálného času ChibiOS bylo možné sjednotit velkou část zdrojových kódů pro jednotlivé desky. Vývoj probíhal v textovém editoru Vim. K testování bylo použito vývojové prostředí Atollic TrueSTUDIO, které umožňuje pokročilé ladění za běhu programu, vyčítání hodnot registrů, atd. Program byl přeložen svobodným kompilátorem GCC ve variantě pro arm (arm-none-eabi-gdb). Automatizaci překladu a některých operací (nahrávání programu do MCU) realizuje program Make. Většina kódu je napsána v jazyce C, část pomocných skriptů pro počítač vznikla v jazyce Python.



Obr. 4.1: Základní struktura zdrojového kódu

Základní organizace zdrojového kódu je na obrázku 4.1. Kód aplikace specifický pouze pro dané zařízení je od zbytku jasně oddělen (vrstva aplikace). Ovladače zprostředkovávají vysokoúrovňový přístup k nízkoúrovňové abstrakci nad HW, kterou poskytuje knihovna ChibiOS HAL. Většinu společného kódu pak tvoří jednotlivé moduly, které obsahují komunikaci, konfiguraci systému, ladění, atd. Různé pomocné funkce, nezávislé na HW, jsou obsaženy v bloku *utils*.

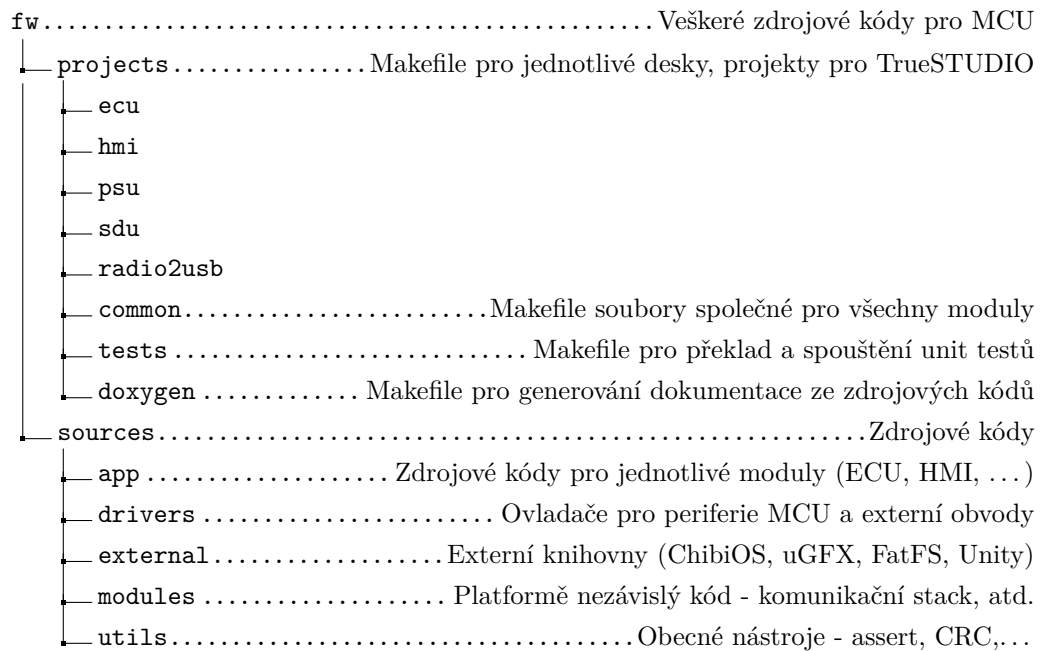
Zdrojové kódy jsou organizovány v adresářové struktuře na obrázku 4.2. Všechny projekty mají podobnou adresářovou strukturu, která je z velké části dána požadovanou konfigurací pro ChibiOS. Na obrázku 4.3 je uveden příklad takové struktury pro modul ECU.

### 4.1 Funkční bloky

V této podkapitole budou popsány významné funkční bloky programového vybavení. Nejedná se o komplexní popis programu, to bez delších ukázek zdrojového kódu nelze dost dobře provést, ale o ideové principy, na kterých jsou tyto bloky založeny.

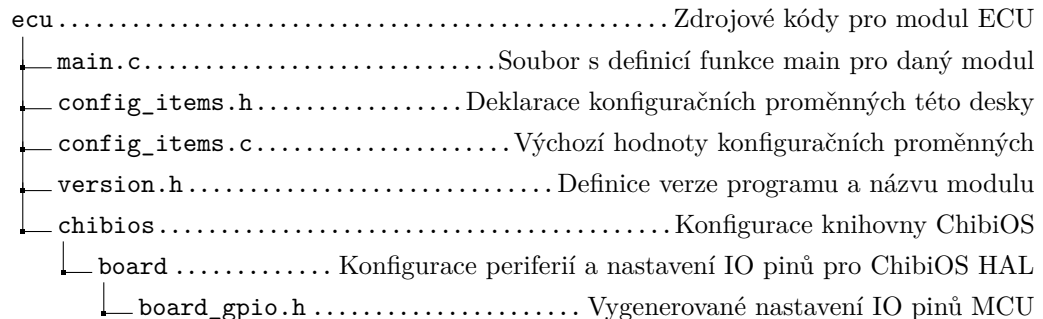
#### 4.1.1 Ovladače (Drivers)

Nízkoúrovňový přístup k perifériím zajišťuje knihovna ChibiOS HAL. Pro každou desku byla vytvořena specifická konfigurace této knihovny s definicí povolených periférií, velikostí paměti pro zásobník, atd. Nad touto vrstvou byla napsána vysokoúrovňová abstrakce,



Obr. 4.2: Adresářová struktura zdrojových kódů

která při startu programu zajišťuje konfiguraci jednotlivých periférií dle požadovaného nastavení, zpracovává události perifériemi generované a poskytuje jednoduché rozhraní pro další vrstvy.



Obr. 4.3: Adresářová struktura projektu ECU (fw/sources/app)

V této části kódu jsou také obsaženy ovladače pro externí zařízení (eeprom, RFM69) a propojovací kód pro externí knihovny (ovladače displeje a dotykové plochy pro grafickou knihovnu uGFX, propojení mezi knihovnou FatFS a SDIO ovladačem z knihovny ChibiOS).

ChibiOS HAL využívá konfiguraci v hlavičkovém souboru *board.h* k inicializaci IO pinů MCU. Tento způsob je ale značně nepohodlný a nepřehledný (nastavují se přímo hodnoty jednotlivých registrů). Naproti tomu pro knihovnu ST HAL existuje elegantní grafický nástroj STM32CubeMx, který ale generuje přímo zdrojové kódy. Pro převod projektů z tohoto nástroje do formátu pro ChibiOS slouží skript *mx2board.py* z komunitního repozitáře ChibiOS-Contrib. Tímto skriptem je během překladu generován soubor *board\_gpio.h*,

který obsahuje kompletní nastavení všech pinů MCU.

Během vývoje byla objevena řada nedostatků tohoto skriptu, která si vyžádala několik zásahů do zdrojového kódu. Tyto změny byly po otestování začleněny zpět do repozitáře a zveřejněny pro využití komunitou.

#### 4.1.2 Systémový záznam (Log)

Pro ladicí, informační a chybové výpisy byla navržena knihovna Log, která přijímá všechny zprávy generované v jednotlivých blocích kódu a předává je registrovaným odběratelům (odesílání po sběrnici CAN, ukládání na SD kartu) dle nastavených priorit.

První část je tvořena funkcemi pro odeslání zpráv (Log\_Error, Log\_Warning,...), které fungují obdobně jako standardní knihovní funkce printf, tedy přijímají formátovací řetězec a volitelné parametry. Navíc je povinně předáván zdroj zprávy (modul, ovladač, aplikace,...) k identifikaci a pozdějšímu filtrování.

Po naformátování předaného řetězce je textová zpráva umístěna do fonty (datová struktura), odkud pak logovací proces jednotlivé zprávy vybírá a odesílá je registrovaným odběratelům dle nastavených filtrů.

Odběratelé se k modulu registrují pomocí tzv. zpětných volání (callback). Každý odběratel je charakterizován adresou funkce, které se má zpráva předat a seznamem zdrojů a priorit (od ladicích výpisů po chybové hlášky), které odběratele zajímají. Tímto způsobem lze snadno určit, že po sběrnici CAN mají být posílány pouze zprávy s prioritou chyba a varování, na SD kartu se ukládá vše, atd. . .

Každá zpráva obsahuje navíc informaci, z kterého modulu pochází (ECU, HMI,...). Díky tomu lze zprávy přijaté po sběrnici CAN zpracovat v tomto modulu a teprve jednotliví odběratelé se rozhodnou, zda chtějí přijímat všechny zprávy, nebo pouze zprávy vznikající v zařízení, na kterém program běží. Např. na sběrnici CAN jsou odesílány pouze zprávy vznikající v aktuálním zařízení, ale SD karta zapisuje vše.

#### 4.1.3 Konfigurace (Config)

Konfigurační knihovna sdružuje přístup k nastavitelným parametrům, jejichž hodnota musí být zachována i po restartu zařízení, nebo ztrátě napájení.

Data jsou uložena v externí EEPROM a jsou načítána po zapnutí napájení. Validita dat je ověřena kontrolním součtem. Pokud součet neodpovídá (prázdná EEPROM, poškozená data), jsou použity výchozí hodnoty. K hodnotám pak lze přistupovat pomocí Get a Set funkcí pro jednotlivé typy proměnných (float, int, bool).

Pro omezení množství zápisů do EEPROM byl implementován časovač, který je restartován s každou změnou hodnoty libovolného parametru. Nedojde-li k restartu časovače do 5 vteřin, jsou data zapsána.

Přístup ke konfiguračním parametrům a k úložišti je jasně oddělen, v budoucnu tak bude možné s minimálními zásahy do kódu ukládat data i na jiné médium, např. na SD kartu. Formát uložených dat je plně ve správě úložiště. Aktuální verze kódu ukládá



jednotlivé typy proměnných do samostatných bloků s vlastním CRC. V budoucnu by bylo vhodné realizovat ukládání konfigurace v textovém souboru na SD kartu.

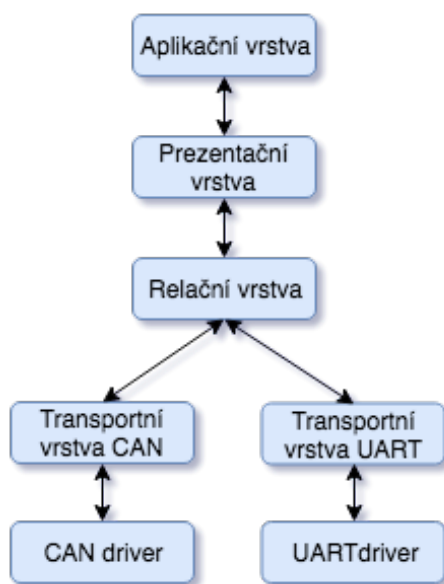
#### 4.1.4 Komunikace (Comm)

Návrh komunikace s ostatními moduly byl inspirován referenčním modelem ISO/OSI [9]. Jednotlivé vrstvy navrženého komunikačního modelu a jejich propojení ukazuje obrázek 4.4.

Nejnižší transportní vrstva přijímá data z ovladačů periférií CAN a UART, uspořádává je do rámců, kontroluje správnost pořadí přijímaných bloků a jejich validitu. Při odesílání naopak data z rámců dělí na menší bloky (pro CAN), generuje kontrolní součet a zajišťuje jejich postupné odesílání.

Relační vrstva (session) navazuje spojení mezi jednotlivými moduly a přiřazuje přijaté odpovědi k odeslaným příkazům. Každé spojení je charakterizováno příkazem a cílovým zařízením, po odeslání zprávy vrstva očekává odpověď se stejným příkazem od adresovaného zařízení.

Prezentační vrstva rozděluje přijatá data podle typu příkazu, kontroluje správnou délku parametrů příkazu a zpracované příkazy předává konkrétní funkci do aplikační vrstvy. Aplikační vrstva poté tvoří jednotné rozhraní mezi komunikačním blokem a zbytkem systému.



Obr. 4.4: Blokové schéma komunikačního modulu

#### 4.1.5 Modul ECU

Primárním úkolem modulu ECU je řízení pneumatického okruhu, princip bude podrobněji rozebrán v podkapitole 4.2. V této části budou rozebrány zbývající funkční principy.

Jednotka ECU obsahuje množství vstupů a výstupů. Pro řízení je nezbytně nutné znát stavy koncových senzorů, polohu pístu a stav plynového pedálu. Poloha pístu je snímána rotačním enkodérem, který je připojen k internímu časovači s podporou generování hodinového signálu na základě signálů z enkodéru. Tento způsob značně šetří strojový čas, protože zpracování jednotlivých pulzů je prováděno přímo na úrovni hardwaru. Při sepnutí koncového senzoru je stav čítače polohy pístu vynulován. Všechny ostatní logické vstupy do jednotky jsou zpracovány blokem na odstranění zákmitů.

Měření rychlosti je realizováno optickou bránou s děrovanou předlohou na zadním kole. Obdélníkový signál je přiveden na interní časovač jako externí hodinový signál. Softwarový časovač v ChibiOS volá s periodou 250 ms funkci na zpracování naměřené hodnoty. Ze znalosti počtu pulzů na otáčku, průměru kola, počtu naměřených pulzů a periody zpracování je počítána aktuální rychlost vozu. Hodnota načítaná v časovači také slouží k přesnému výpočtu ujeté vzdálenosti.

Tlaky v okruhu jsou měřeny pomocí analogových tlakoměrů. Modul ADC je nakonfigurován tak, aby naměřené hodnoty ukládal pomocí DMA (Direct Memory Access) přímo do paměti RAM, odkud jsou programem periodicky vyčítány.

Modul všechny naměřené provozní údaje (tlaky, polohu pístu, rychlost, atd.) odesílá po sběrnici CAN s periodou 10 ms.

#### 4.1.6 Modul HMI

Jednotka HMI má několik úkolů, ukládá veškeré provozní údaje a systémové zprávy na SD kartu, odesílá telemetrii do počítače a slouží jako rozhraní mezi uživatelem a řídicím systémem.

Veškeré údaje přijímané po sběrnici CAN jsou ukládány do interní datové struktury pro využití jednotlivými částmi kódu.

Zápis na SD kartu zajišťuje populární knihovna *FatFS*. Jedná se o knihovnu pro přístup k souborům na souborovém systému FAT32. K propojení této knihovny a SD karty slouží knihovna ChibiOS HAL. Po vložení karty do jednotky je vytvořena složka pro záznamy z jednotlivých závodů. Při přepnutí závodního módu je automaticky v této složce vytvořen nový soubor formátu *csv*, do kterého je s periodou 10 ms ukládán záznam o aktuálním stavu vozidla (rychlost, tlaky, ujetá vzdálenost, poloha pístu, atd.). Na kartě je také vytvořen textový soubor, kam jsou ukládány všechny zprávy z modulu Log mimo ladicích. Takový záznam je velmi užitečný pro následnou analýzu v případě, že během závodu došlo k nestandardnímu chování.

Vybraná část údajů o stavu vozidla je periodicky odesílána přes bezdrátový modul do počítače. Perioda odesílání je 100 ms. Mimo tato data jsou odesílány i vybrané chybové hlášky, které může tým v depu interpretovat a přes vysílačku instruovat řidiče, jak problém vyřešit (restartovat vozidlo, změnit závodní mód, ...). Veškerá bezdrátová komunikace je z důvodu bezpečnosti pouze jednosměrná.

Rozhraní pro interakci s uživatelem je realizováno pomocí dotykového grafického displeje. Pro zjednodušení vývoje byla zvolena knihovna *uGFX*, která umožňuje snadnou

tvorbu grafického dotykového rozhraní. Komunikace s displejem je realizována po 16 bitové paralelní lince, značné zvýšení rychlosti zápisu umožňuje periferie FSMC (Flexible Memory Controller), která automaticky generuje při zápisu na konkrétní adresu v paměti procesoru potřebné signály pro zápis do displeje (obsah datové sběrnice, zda jde o instrukci, nebo data, atd.).

Uživatelské rozhraní se skládá z několika obrazovek. Na úvodní obrazovce jsou zobrazeny informace podstatné během závodu - aktuální rychlost, ujetá vzdálenost, čas závodu, zařazený rychlostní stupeň, atd. Ve spodní části obrazovky jsou pak tlačítka na změnu závodního režimu, restart závodu (vytvoří nový soubor na SD kartě, vynuluje časovač) a ovládání klaksonu (viz obrázek 4.5).



Obr. 4.5: Závodní obrazovka

Další z oken obsahuje kompletní data o vozidle, stav všech vstupů, napětí v jednotlivých napájecích větvích, atd. Následující obrazovka umožňuje nastavení aktuálního data a času, parametrů plnění a času v úvratí pro jednotlivé závodní režimy.

Pro účely ladění byla vytvořena obrazovka, která dovoluje ovládat výstupy jednotky ECU manuálně (ovládání ventilů, klaksonu, brzdového světla, ...). Na poslední obrazovku je vypisován systémový záznam, jednotlivé závažnosti chybových zpráv jsou odlišeny barevně.

Během testování byla objevena chyba v kódu knihovny uGFX pro odřádkování konzolového výpisu. Díky přístupu ke zdrojovým kódům knihovny bylo možné tuto chybu rychle lokalizovat a opravit. Oprava byla poskytnuta zpět komunitě.

#### 4.1.7 Modul PSU

Kód pro modul PSU obsahuje poměrně jednoduchou logiku. Ve smyčce jsou čteny proudy a napětí v jednotlivých napájecích větvích a na akumulátorech. Pokud některá z hodnot překročí definovanou mez, je rozsvícena červená LED dioda indikující problém.

Dále je z napětí akumulátoru odhadována zbývající kapacita a tato je pravidelně odesílána po sběrnici CAN ostatním modulům.

#### 4.1.8 Modul SDU

Tento modul přijímá po sběrnici CAN z jednotky ECU informace o aktuální rychlosti vozu a ujeté vzdálenosti. Tato data poté zobrazuje na sedmisegmentovém displeji. Řízení displeje je multiplexní s pracovní frekvencí 1 kHz. Na displeji je možné zobrazit pouze číslice, použitý budič s integrovaným BCD dekodérem jiné znaky nepodporuje.

Trojice LED diod měla původně sloužit k zobrazení aktuálně zařazeného rychlostního stupně, ale díky absenci senzorů na převodovce vozu Falcon nyní zobrazuje aktuální závodní režim.

#### 4.1.9 Převodník RFM69 na USB

Modul využívá ke komunikaci s počítačem po USB virtuální sériový port, jehož implementace je součástí knihovny ChibiOS. Zařízení je pasivní, jeho jedinou funkcí je přeposlání přijaté telemetrie do počítače. Komunikace je pouze jednosměrná.

Zpráva přijatá z modulu RFM69 je do počítače přenášena pomocí stejného protokolu jako je popsán v kapitole 3.2. Blok dat je doplněn identifikátorem začátku zprávy, její délkou a kontrolním součtem pro ověření správnosti přenesených dat. Ověřování správnosti přenášovaných dat na úrovni bezdrátového spojení zajišťuje samotný modul RFM69.

Přenášená data se skládají z jednotlivých proměnných uložených sekvenčně za sebou, formát zprávy je přesně daný.

### 4.2 Řízení pneumatického okruhu

Základními parametry pro řízení pneumatického okruhu jsou procento plnění a čas v úvrati. Procento plnění udává po jakou část pohybu pístu má být dodáván do válce tlakový vzduch. Čas v úvrati určuje dobu, po kterou jsou ventily mezi jednotlivými zdvihy uzavřené.

Princip ilustruje následující příklad: chceme-li dosáhnout maximální rychlosti a velkého zrychlení, bude procento plnění nastavené na maximální hodnotu (100 %) a čas v úvrati na nulu. Píst se tedy pohybuje od dorazu k dorazu maximální možnou rychlostí a mezi jednotlivými zdvihy je prodleva daná pouze mechanickým zpožděním ventilů. Pokud kládeme důraz na dojezd, bude procento plnění nastavené na nízkou hodnotu tak, aby po uzavření přívodu vzduchu dojel píst setrvačností poblíž koncové polohy. Čas v úvrati bude v řádu jednotek sekund.

Při návrhu bylo počítáno s možností selhání koncových senzorů, nebo rotačního enkodéru. V prvním případě se selhání na funkci vozu po inicializaci již nijak neprojeví, v druhém zůstane zachována schopnost jízdy, ovšem bez možnosti řídit procento plnění (nastaveno fixně na 100 %)

### 4.2.1 Měření polohy pístu ve válci

Základním údajem pro řízení pneumatiky je poloha pístu ve válci. Jelikož pohyb pístu je snímán pouze relativně rotačním enkodérem, je nutné nějakým způsobem zjistit počet pulzů enkodéru na přejezd z jedné krajní polohy do druhé a určit počáteční bod. V předchozích letech byla nutná kooperace s řidičem, který vyřadil a nechal elektroniku provést dva zdvihy. Tento způsob ale plýtvá vzduchem a vyžaduje určité množství času, kterého se nemusí dostávat (např. při restartu elektroniky během závodu).

V loňském roce byla snaha tento proces zkrátit a zjednodušit. Navržený postup již nevyžadoval interakci se řidičem. Po stisku plynu dojel píst do jedné krajní polohy, čítač impulzů byl vynulován a po dojezdu do druhé krajní polohy byla uložena délka pístu v počtu pulzů. To ale způsobilo stoprocentní plnění pro první cyklus.

Tento způsob byl v aktuální verzi optimalizován tak, aby nedocházelo k mrhání vzduchem při prvním cyklu. Při prvním zapnutí jednotky proběhne stejný proces jako v předchozím odstavci. Rozdíl je ten, že změřená hodnota délky pístu je uložena do nevolatilní paměti, při dalším spuštění je použita již dříve naměřená hodnota. Pokud není píst po zapnutí a zrychlené inicializaci v krajní poloze, nelze jeho přesnou polohu zjistit. Známe ovšem délku chodu pístu a tak je možné provést relativní pohyb o požadované procento plnění. Při pohybu buď dojde k dojezdu na koncový spínač, nebo píst zastaví někde mezi spínači. I při nízkém procentu plnění dojde nejvýše po jednotkách zdvihů k dojezdu na některý koncový spínač a tedy k určení přesné polohy.

### 4.2.2 Závodní režimy

Řídicí logika obsahuje pro každý režim závodu (arkáda, akcelpace, závod na dojezd) vlastní nastavení procenta plnění a času v úvrati. Bohužel vzhledem k probíhajícím úpravám na vozidle nebylo možné provést dostatečně důkladné testování a sběr dat, na základě kterého by bylo možné jednotlivé režimy optimalizovat.

Programová logika řízení pneumatiky je od nastavení procenta plnění a času v úvrati striktně oddělena. Dynamické řízení parametrů pro jednotlivé zdvihy je tedy možné implementovat bez zásahů do logiky samotné. Do dalších let je po důkladném testování vlivu jednotlivých parametrů a jízdních režimů na spotřebu vzduchu a maximální rychlost plánována úprava algoritmu řízení pro získání co nejlepších jízdních vlastností.

Původně zamýšlený elektronický redukční ventil nebyl na vozidlo na základě rozhodnutí konstrukčního týmu z FSI osazen.

## 4.3 Testování

Většina funkčních bloků byla testována ještě před sestavením finálního programu na úrovni tzv. jednotkových (unit) testů [10]. Jedná se o metodu využívající automatizovaného testování software na úrovni jednotek, tedy jednotlivých funkcí a funkčních celků, za pomoci testovacích knihoven.

Ke každé testované jednotce je vytvořena sada testů, které ověřují její fungování jak pro validní, tak pro nevalidní vstupy. Pomocí různých dalších nástrojů je poté možné ověřit pokrytí kódu testy. A to včetně všech možných kombinací průchodů přes větvení testovaného kódu.

Testy je možné pouštět nad každou revizí kódu a ověřovat tak jeho správnost. V případě nevědomého zavlečení chyby do testovaného kódu je tato chyba odhalena ihned po spuštění testů. Další výhodou je urychlení vývoje díky včasné detekci chyb. S výhodou lze unit testování využít například při psaní komunikačního rozhraní, kdy většina funkcionality může být nejprve popsána unit testy a teprve poté naprogramována. Tímto způsobem jde program testovat průběžně během jeho implementace a odhalovat chyby prakticky okamžitě po jejich vytvoření.

Tento projekt využívá knihovnu *Unity*, která je velmi minimalistická a nenáročná na paměť. Část testů je proto možné spouštět i na fyzickém zařízení a ověřit tak například komunikaci s externími periferiemi. Níže je uveden příklad jednoduchého testu knihovny na výpočet kontrolního součtu:

```
TEST(CRC, CRC8)
{
    uint8_t buf[10] =
        {0x0a, 0x13, 0x15, 0xff, 0x11,
         0xa8, 0xfa, 0xac, 0xcd, 0x00};

    TEST_ASSERT_EQUAL(0xcf, CRC8(buf, sizeof(buf)));
}
```

Části, které nebylo možno ověřit automaticky byly pečlivě testovány manuálně. Testování celku probíhalo na vozidle Falcon krátce před závody, dřívější testování nebylo možné díky probíhajícím úpravám na řízení a podvozku.

## 4.4 Dokumentace

Zdrojové kódy jsou průběžně dokumentovány pomocí nástroje *Doxygen*, který generuje dokumentaci v různých formátech (PDF, HTML, ...) z komentářů ve zdrojových kódech. Příklad takového komentáře je uveden níže:

```
/**
 * Get name of the given log source in string
 *
 * @param [in] src      Log source to get name for
 * @return Pointer to constant string
 */
extern const char *Log_GetSourceStr(log_src_t src);
```

## 4.5 Programy pro počítač

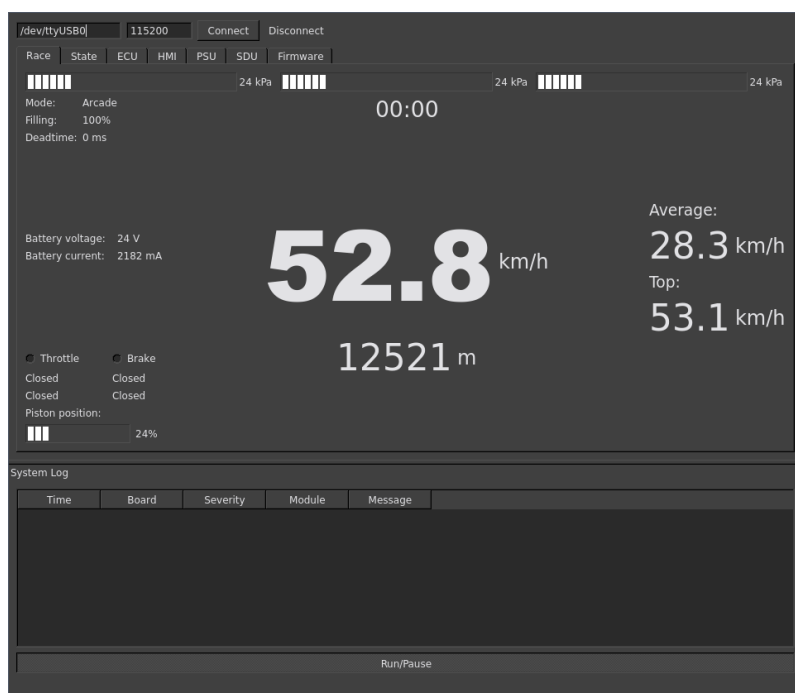
Během vývoje vyvstala nutnost vytvořit několik nástrojů pro usnadnění a zpřehlednění programování. K jejich implementaci byl použit jazyk Python 3 [11].

### 4.5.1 Generování konfiguračních parametrů

Modul konfigurace (config) vyžaduje soubory *config\_items.h* a *config\_items.c* s definicí existujících konfiguračních voleb a jejich výchozích hodnot. Manuální psaní a udržování těchto souborů je pracné a nepřehledné.

V první fázi vývoje byly jednotlivé parametry společně s datovým typem, výchozí hodnotou a komentářem sepsány v tabulkovém procesoru. Aby tato data nebylo nutné manuálně přepisovat do formátu pro programovací jazyk C, byl napsán jednoduchý skript, který načítá data z tabulkového procesoru a na jejich základě generuje příslušné soubory pro konfigurační modul.

Výhodou tohoto řešení je mimo zjednodušení práce i umožnění generování ekvivalentních struktur pro počítačový program v pytonu, který by umožnil s adekvátním převodníkem z USB na CAN vyčítání a zapisování konfiguračních voleb do jednotlivých zařízení.



Obr. 4.6: Ukázka programu na příjem telemetrie

### 4.5.2 Generování komunikačních příkazů

Obdobně jako v předchozím případě byly při návrhu komunikačního rozhraní sepsány možné příkazy a jejich formát do tabulkového procesoru. Skript tato data převádí do datové struktury jazyka C pro komunikační modul a do slovníku jazyka python pro ladicí

účely. Zajišťuje tak shodnost číselného identifikátoru příkazu mezi externím zařízením (HMI, ECU, . . .) a aplikací v počítači.

### **4.5.3 Telemetrie**

Data přijatá převodníkem ze sběrnice USB na modul RFM69 jsou zpracována v grafické aplikaci vytvořené s použitím knihovny QT. Uživatel po spuštění aplikace zvolí virtuální sériový port, ke kterému je převodník připojen a aplikace následně zajišťuje vykreslování aktuálních provozních informací vozidla (obrázek 4.6).

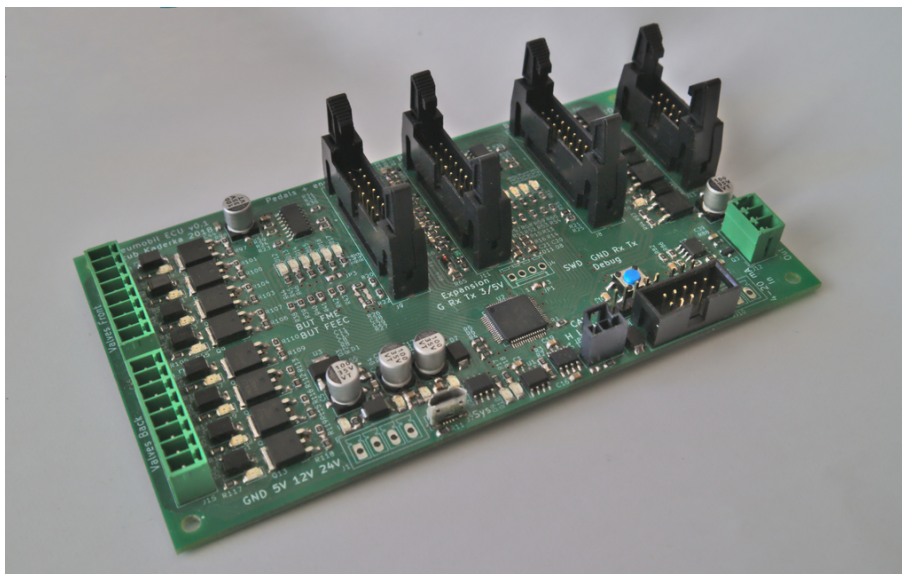
Ve spodní části obrazovky jsou navíc vypisovány systémové zprávy z jednotlivých modulů. Výchozí myšlenkou bylo tuto aplikaci rozvinout do podoby univerzálního ladicího rozhraní, které by umožnilo přístup k jednotlivým modulům, čtení jejich stavu, řízení výstupů, přehrávání firmware, atd. Nicméně díky nedostatku času nebyla tato funkcionality do programu plně implementována.



## 5 Konstrukce a oživení elektroniky

### 5.1 Desky plošných spojů

Návrh schémat a plošných spojů probíhal v programu KiCad, což je návrhový nástroj s otevřenými zdrojovými kódy. Velkou výhodou tohoto programu je sdílená knihovna součástek a oddělení schématického symbolu od fyzického pouzdra. Během návrhu bylo nutné vytvořit knihovní prvky pro několik chybějících součástek, které byly posléze poskytnuty komunitě a po schvalovacím procesu začleněny do oficiálních repozitářů.



Obr. 5.1: Jednotka ECU

Všechny desky byly navrženy jako dvojvrstvé, horní vrstva slouží k vedení signálových spojů a napájecích napětí, spodní vrstva je z velké části vyplněna zemnicí plochou pro snížení zemní impedance. Výrobu desek plošných spojů zajistila firma Gatema v rámci jejich služby „Pool Service“.

V návrhu jsou ve velké míře použity součástky pro povrchovou montáž, pouze konektory jsou ve vývodovém provedení pro zvýšení mechanické odolnosti. Součástky jsou osazeny převážně z jedné strany s výjimkou modulu převodníku z USB na RFM69, kde pro zmenšení rozměrů desky je bezdrátový modul osazen ze spodní strany a modulu SDU, kde přední stranu obsadily sedmissegmentové displeje a veškeré zbývající součástky jsou osazeny na straně zadní.

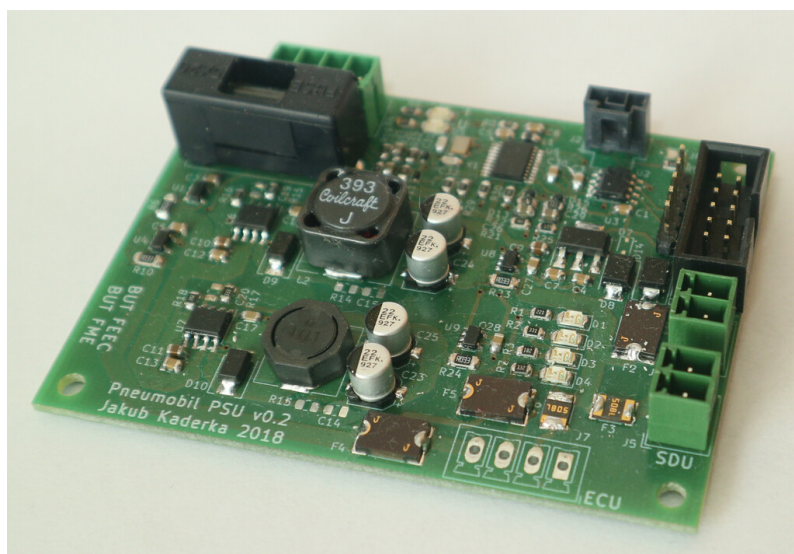
Návrh modulu ecu ECU (obrázek 5.1) je díky velkému množství signálů nejsložitější. LED diody indukující aktuální stav vstupů a výstupů jsou umístěny v těsné blízkosti příslušných konektorů.

Deska modulu HMI (obrázek 5.2) odpovídá svými rozměry zvolenému dotykovému displeji. Pod displejem je umístěna šestice tlačítek a USB konektor pro flash disk. Na zadní straně se nachází konektor sdružující napájení, programovací rozhraní a komunikační



Obr. 5.2: Jednotka HMI (bez osazeného LCD)

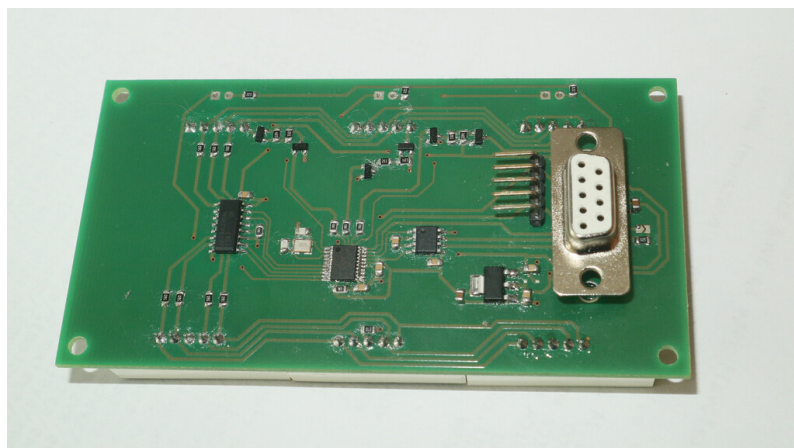
sběrnice. Na této straně je též umístěn koaxiální konektor pro připojení externí antény k bezdrátovému modulu.



Obr. 5.3: Jednotka PSU

Napájecí zdroj (obrázek 5.3) se skládá ze dvou shodných bloků pro 12 a 5 V spínané zdroje, v horní části se pak nachází MCU a měřicí obvody.

Konstrukčně jednodušší modul rychloměru (obrázek 5.4) byl navržen tak, aby svými rozměry odpovídal rychloměru z předchozího roku, aby nebylo nutné upravovat výřez



Obr. 5.4: Jednotka SDU

v kapotě vozidla. Obdobně jako modul HMI i tento blok obsahuje konektor pro připojení napájení a datové sběrnice.



Obr. 5.5: Přijímač telemetrie

Modul převodníku z RFM69 na USB (obrázek 5.5) byl navržen s minimálními možnými rozměry. Pro účely ladění obsahuje tato varianta piny s programovacím rozhraním a sériovým portem.

## 5.2 Oživování elektroniky

Během testování elektroniky došlo k zničení integrovaného obvodu RT8279 z nezjištěných příčin (pravděpodobně zkrat kovovou nečistotou na desce). Z důvodu aktuální nedostupnosti u běžných dodavatelů byl tento obvod nahrazen regulátorem TPS5430 se stejným rozložením vývodů a shodnými parametry.

Komunikace po sběrnici CAN s využitím interních oscilátorů MCU se ukázala jako nestabilní a vysoce chybová i při rychlostech do 100 kBaud/s a to i přes skutečnost, že periferie CAN obsahuje algoritmus pro korekci vzorkování sběrnice na základě přenášených



synchronizačních bitů. Naštěstí byl tento problém odhalen v okamžiku, kdy ještě nebyly všechny desky vyrobeny a do finální série tak mohl být doplněn externí krystal ke každému MCU. Na již vyrobených deskách bylo pouzdro krystalu přilepeno k desce plošného spoje a k MCU připojeno tenkým vodičem.

Toto řešení samozřejmě není vhodné pro použití ve vozidle díky nižší mechanické odolnosti, tyto desky proto slouží pro testování a jako záložní pro případ poškození finálních desek během závodu.

### 5.3 Umístění elektroniky ve vozidle

Moduly PSU a ECU jsou umístěny v plastové instalační krabici s třídou ochrany IP54 po pravici řidiče (obrázek 5.6). K jednotlivým prvkům se pak od krabice rozbíhají jednotlivé kabely vyvázané do svazků vinutou bužírkou. Takto vytvořené svazky tvoří kompaktní celek, který je z vozidla v případě větších úprav (např. zásahy do rámu) možno vyjmout a bez větších komplikací zase vrátit.



Obr. 5.6: Desky PSU a ECU umístěné v instalační krabici

Modul SDU je umístěn přímo v kapotě vozu, při sundávání kapoty je bohužel nutné nejprve odpojit poměrně špatně přístupný konektor. Pro zjednodušení této činnosti je přívodní vodič delší, než je nutné.

Jednotka HMI je umístěna spolu s hlavním vypínačem a několika tlačítky po řidičově pravici. Tato tlačítka (změna závodního režimu, klakson) jsou připojena přímo k modulu ECU, v případě poruchy na modulu HMI je vozidlo dále plně pojízdné a ovladatelné. Pohled do kabiny vozu je na obrázku 5.7.

Akumulátory jsou umístěny v samostatné instalační krabici s IP54. Záporný pól akumulátoru je připojen na kostru vozidla, kladný pól je přes pojistku vyveden na konektor



Obr. 5.7: Pohled do kabiny vozu

na krabici.

## 5.4 Bezpečnost

Jelikož se jedná o vozidlo, ve kterém budou jezdit lidé, je nutné uvažovat i situace, které by mohly vést ke zranění řidiče, nebo osob v okolí. Nejzákladnějším bezpečnostním opatřením, které vyžadují i pravidla, je nouzový vypínač dostupný jak řidiči, tak okolo stojícím lidem. Vypínač je umístěn mezi akumulátorem a elektronikou, jeho stisknutí tedy způsobí kompletní vypnutí elektroniky a ventily se bez napájení samovolně uzavřou.

Dalším prvkem je samotný způsob řízení pneumatického okruhu. Pokud dojde k selhání (zastavení) řídicího programu, stav ventilů se dále nemění a po ukončení jednoho zdvihu se píst zastaví.

Kritickou částí je spínač plynového pedálu, v případě jeho selhání ve stavu sepnuto by vozidlo dále akcelerovalo nezávisle na vůli řidiče, který by musel zasáhnout a použít nouzový vypínač. Zkratování spínače je ale krajně nepravděpodobné, v případě selhání konektoru je spínač považován za rozpojený a vozidlo nejede.

Posledním bezpečnostním prvkem je spínač na brzdovém pedálu. Při jeho aktivaci dojde k uzavření ventilů.

Z elektrického hlediska je kritickým bodem akumulátor, Li-Poly akumulátor se při zkratu rychle zahřívá a hrozí jeho zahoření. První linií obrany je tavná pojistka přímo u akumulátoru. Sekundární ochranu zajišťují polymerové pojistky v napájecí cestě k jednotlivým modulům.

## 6 Závěr

V úvodu práce byl rozebrán princip závodního pneumatického monopostu Falcon, který je stavěn pod patronátem strojí fakulty VUT. Dále proběhla analýza vhodných komunikačních sběrnic a bezdrátových modulů. Na základě této analýzy byla pro komunikaci mezi elektronickými celky zvolena automobilová sběrnice CAN. Bezdrátové spojení zajišťuje modul RFM69.

Řídicí elektronika byla navržena jako modulární, skládá se ze 4 částí. Modul ECU se stará o řízení pneumatického pohonu a měření důležitých veličin (rychlost vozu, tlaky v okruhu, ...). Interakci s řidičem přes grafický dotykový displej provádí jednotka HMI. Ta také zajišťuje odesílání telemetrie do počítače a ukládání záznamu o závodu a chybových zpráv na SD kartu. Jednotka SDU umístěná v kapotě vozu slouží k zobrazování rychlosti vozu a aktuálně aktivního závodního režimu. Napájení celého systému zajišťuje jednotka PSU, která z napětí akumulátorů generuje potřebná nižší napětí. PSU také měří napětí a proudy jednotlivých napájecích větví.

Tyto moduly doplňuje ještě přijímač telemetrie pro počítač s rozhraním USB. Srdcem všech modulů jsou mikroprocesory řady STM32, Od nejjednodušších STM32F0xx po výkonné STM32F4xx v HMI.

Při návrhu softwarové architektury byl kladen důraz na přehlednou strukturu kódu a maximální znovupoužitelnost jednotlivých bloků. Pečlivé oddělení funkčních bloků do celků s jasně definovaným rozhraním zpřehledňuje výsledný program tak, aby část vývoje mohli v dalších letech převzít studenti strojí fakulty.

Návrh značně ulehčil operační systém reálného času ChibiOS a jeho abstrakce nad periferiemi ChibiOS/HAL. Po zkušenostech z vývoje komerčních zařízení byla pro účely ladění vytvořena knihovna pro zpracování ladicích a informačních výpisů. Tato knihovna umožňuje zpracování zpráv z různých částí kódu a jejich filtraci na základě jejich původu a zvolených priorit.

Pro sběrnici CAN, která zajišťuje datové spojení mezi moduly, byl navržen vlastní protokol s důrazem na snadnou implementaci a vysokou komunikační rychlost. Při testování se ukázal jako spolehlivý a bezproblémový. Pro účely ladění byly vyšší vrstvy komunikace implementovány genericky a zprávy je možné přijímat jak po sběrnici CAN, tak po sériovém rozhraní, v budoucnu pak případně i přes bezdrátové spojení.

Vybrané programové bloky byly testovány na úrovni jednotkových testů. Kód je průběžně dokumentován komentáři a přehledná dokumentace je z těchto komentářů generována programem Doxygen. Zdrojové soubory mají dohromady přes 13 tisíc řádků kódu.

Při instalaci elektroniky do vozidla byl kladen důraz výběr na kvalitních konektorech, aby se neopakovala situace z roku 2018, kdy vinou selhání konektoru vůz Falcon jednu závodní disciplínu nedokončil.

Z důvodu úprav na závodních vozidlech, které probíhaly až do doby těsně před závody mi nebylo umožněno dostatečně dlouhé testování, na základě kterého by bylo možné provést analýzu a optimalizaci řídicího algoritmu. První jízda proběhla ve čtvrtek 2. května,

poté se auto opětovně rozebralo pro další úpravy a finální testování proběhlo teprve v úterý 7. května, ve středu ráno se již odjíždělo na závody.

Nicméně i přes tuto skutečnost se vozu Falcon s touto elektronikou podařilo umístit na 8. místě v závodě na dojezd s dojezdem téměř 7 km. V prvním kole závodu na čas jsme dojeli druzí se ztrátou pouhých 0,5 s na vítěze. Druhé kolo jsme bohužel kvůli závadě na mechanice vozu nedokončili. V závodě na akceleraci jsme ztratili drahocenný čas při rozjezdu kvůli závadě na převodovce, ale i přesto jsme dosáhli nejvyšší rychlosti ze všech vozů (50,3 km/h) a umístit se na celkovém 7. místě.

Během závodů došlo vinou chyby v programu k zaseknutí jednotky HMI, nicméně díky modulárnímu návrhu tato chyba nijak neohrozila provoz vozu. Jediným důsledkem byla ztráta záznamu měřených veličin z části závodu. Oproti ostatním vozům zaujal porotu vůz Falcon pečlivě vyvázanou a přehlednou kabeláží a profesionálně vypadajícími deskami plošných spojů. Ostatní týmy většinou spoléhaly na průmyslové programovatelné logické automaty, nebo Arduino a kabeláž mnohdy připomínala vrabčí hnízdo.

Po analýze výsledků aktuálních závodů a zhodnocení řešení ostatních závodníků bude do příštího roku vůz doplněn o automatické řazení a dvoupolohový plynový pedál. Díky dostatečnému množství nevyužitých vstupů a výstupů nebude tato změna vyžadovat mimo úpravy kódu zásah do návrhu desek plošných spojů.



Obr. 6.1: Vozidlo Falcon během kvalifikace do závodu na čas

## Literatura

- [1] *Pneumobil Racing Team Brno* [online]. [cit. 2018-11-06]. Dostupné z: <http://pneumobilteam.cz/>
- [2] KROUPA, Jiří, Igor HAŠLÍK, Jan HAJDA a Jiří KOČAŘ. *Realizace pneumatického a elektronického systému pneumobilu Falcon*. Brno, 2018. Technická zpráva. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav konstruování. Vedoucí práce Ing. Matúš Ranuša.
- [3] *I2C-bus specification and user manual: Rev. 6* [online]. 2014 [cit. 2019-05-12]. Dostupné z: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf>
- [4] *LIN package specification: Revision 2.2A* [online]. 31.12. 2010, , 194 [cit. 2018-11-06]. Dostupné z: [https://www.cs-group.de/wp-content/uploads/2016/11/LIN\\_Specification\\_Package\\_2.2A.pdf](https://www.cs-group.de/wp-content/uploads/2016/11/LIN_Specification_Package_2.2A.pdf)
- [5] WATTERSON, Conal. *Controller Area Network (CAN) Implementation Guide: APPLICATION NOTE AN-1123* [online]. Analog Devices, , 14 [cit. 2018-11-06]. Dostupné z: <https://www.analog.com/media/en/technical-documentation/application-notes/AN-1123.pdf>
- [6] DI SIRIO, Giovanni. *ChibiOS/RT 3.0: The Ultimate Guide* [online]. 2015 [cit. 2019-04-26]. Dostupné z: <http://www.chibios.org/dokuwiki/doku.php?id=chibios:book:start>
- [7] *Announcement and rules: XII. International Aventics Pneumobile Competition 2019 – powered by Emerson*[online]. Aventics [cit. 2018-11-21]. Dostupné z: [http://en.pneumobil.hu/content/download/11463/44860/file/Announcement\\_and\\_rules\\_2019\\_final\\_0927.pdf](http://en.pneumobil.hu/content/download/11463/44860/file/Announcement_and_rules_2019_final_0927.pdf)
- [8] *RT8279: 5A, 36V, 500kHz Step-Down Converter* [online]. 2011 [cit. 2018-11-22]. Dostupné z: [https://www.richtek.com/assets/product\\_file/RT8279/DS8279-01.pdf](https://www.richtek.com/assets/product_file/RT8279/DS8279-01.pdf)
- [9] *Open systemms interconnection - Model and notation: Basic reference model X.200*. ITU, 1994.
- [10] HEROUT, Pavel. *Testování pro programátory*. České Budějovice: Kopp, 2016. ISBN 978-80-7232-481-1.
- [11] PILGRIM, Mark. *Ponořme se do Python(u) 3: Dive into Python 3*. Praha: CZ.NIC, c2010. CZ.NIC. ISBN 978-80-904248-2-1.



## Seznam symbolů, veličin a zkratk

<b>AD(C)</b>	Analog to Digital (Converter)
<b>BLE</b>	Bluetooth Low Energy
<b>CAN</b>	Controller Area Network
<b>CRC</b>	Cyklický redundantní součet – Cyclic Redundancy Check
<b>DAC</b>	Digital to Analog Converter
<b>DMA</b>	Direct Memory Access
<b>DPS</b>	Deska plošných spojů
<b>ECU</b>	Jednotka řízení motoru – Engine Control Unit
<b>EMI</b>	Elektromagnetické rušení – Electromagnetic interference
<b>ESR</b>	Ekvivalentní sériový odpor – Equivalent Series Resistance
<b>FAT32</b>	Souborový systém File Allocation Table
<b>FSMC</b>	Flexible Static Memory Controller
<b>GCC</b>	GNU Compiler Collection
<b>GPS</b>	Lokalizační systém Global Positioning System
<b>HAL</b>	Vrstva abstrakce nad HW – Hardware Abstraction Layer
<b>HMI</b>	Uživatelské rozhraní – Human Machine Interface
<b>HW</b>	Hardware
<b>IO</b>	Vstup-výstup – Input-Output
<b>I2C</b>	Komunikační sběrnice Inter-Integrated Circuit
<b>LCD</b>	Liquid-Crystal display
<b>LDO</b>	Lineární regulátor – Low Dropout Regulator
<b>LED</b>	Svítivá dioda – Light Emitting diode
<b>LGPL</b>	GNU Lesser General Public License
<b>LIN</b>	Sběrnice Local Interconnect Network
<b>MCU</b>	Mikrokontrolér
<b>PSU</b>	Jednotka napájení – Power Supply Unit
<b>RTOS</b>	Operační systém reálného času – Real Time Operating System
<b>SD</b>	Secure Digital paměťová karta
<b>SDIO</b>	Komunikační rozhraní SD karet – Secure Digital Input Output
<b>SDU</b>	Jednotka zobrazovače rychlosti – Speed Display Unit
<b>SPI</b>	Serial Peripheral Interface
<b>SW</b>	Software
<b>SWD</b>	Ladicí rozhraní Serial Wire Debug
<b>UART</b>	Sériové rozhraní – Universal Asynchronous Receiver-Transmitter
<b>USB</b>	Rozhraní Universal Serial Bus
$f_m$	Mezní frekvence
$\Delta I_L$	Zvlnění proudu spínaného regulátoru
$\Delta U_{out}$	Zvlnění výstupního napětí spínaného regulátoru

## Seznam obrázků

1	Závodní monopost Falcon (převzato z [1]) . . . . .	10
1.1	Blokové schéma pneumatického okruhu . . . . .	12
1.2	Příklad přenosu jednoho rámce na sběrnici CAN (převzato z [5]) . . . . .	14
1.3	Modul RFM69 . . . . .	17
2.1	Blokové funkční schéma pneumobilu . . . . .	20
2.2	Blokové schéma elektroniky . . . . .	22
2.3	Zapojení univerzálního komunikačního konektoru . . . . .	23
2.4	Blokové schéma modulu ECU . . . . .	23
2.5	Vstupní filtr tlakoměru . . . . .	25
2.6	Spínání ventilů . . . . .	26
2.7	Zapojení proudové smyčky . . . . .	27
2.8	Blokové schéma modulu HMI . . . . .	28
2.9	Zapojení spínaného zdroje s obvodem MCP1603T . . . . .	29
2.10	Blokové schéma modulu SDU . . . . .	29
2.11	Blokové schéma modulu PSU . . . . .	30
2.12	Schéma 5 V větve napájecího zdroje . . . . .	31
2.13	Zapojení pro měření proudu . . . . .	32
4.1	Základní struktura zdrojového kódu . . . . .	38
4.2	Adresářová struktura zdrojových kódů . . . . .	39
4.3	Adresářová struktura projektu ECU (fw/sources/app) . . . . .	39
4.4	Blokové schéma komunikačního modulu . . . . .	41
4.5	Závodní obrazovka . . . . .	43
4.6	Ukázka programu na příjem telemetrie . . . . .	47
5.1	Jednotka ECU . . . . .	49
5.2	Jednotka HMI (bez osazeného LCD) . . . . .	50
5.3	Jednotka PSU . . . . .	50
5.4	Jednotka SDU . . . . .	51
5.5	Přijímač telemetrie . . . . .	51
5.6	Desky PSU a ECU umístěné v instalační krabici . . . . .	52
5.7	Pohled do kabiny vozu . . . . .	53
6.1	Vozidlo Falcon během kvalifikace do závodu na čas . . . . .	55

## Seznam tabulek

1.1	Srovnání vlastností komunikačních rozhraní . . . . .	13
1.2	Shrnutí vlastností bezdrátových modulů . . . . .	15
3.1	Struktura datového rámce sběrnice CAN [5] . . . . .	34
3.2	Formát identifikátoru zprávy . . . . .	35
3.3	Příklad odesílání dlouhé zprávy, jednotlivé rámce jsou odlišeny barevně . .	36
3.4	Formát zprávy na sériovém rozhraní . . . . .	36
3.5	Formát příkazu . . . . .	36
3.6	Demonstrační příklad nastavení parametrů závodního módu . . . . .	37

# Seznam příloh

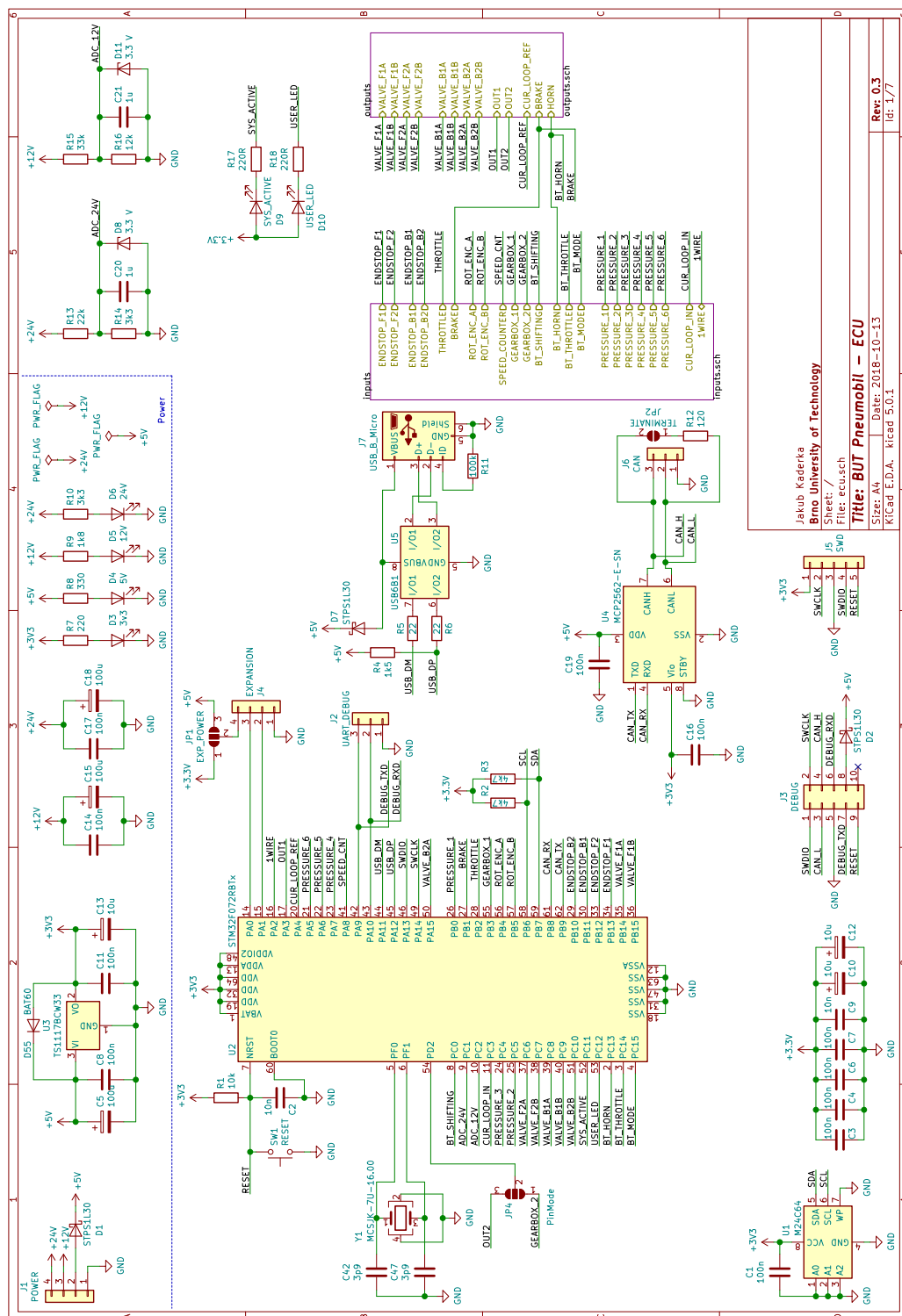
<b>A</b>	<b>Schémata zapojení</b>	<b>62</b>
A.1	Modul ECU . . . . .	62
A.1.1	Hlavní list . . . . .	62
A.1.2	Vstupy . . . . .	63
A.1.3	Výstupy . . . . .	64
A.1.4	Řízení ventilů . . . . .	65
A.2	Modul SDU . . . . .	66
A.3	Modul HMI . . . . .	67
A.3.1	Hlavní list . . . . .	67
A.3.2	Napájecí zdroj . . . . .	68
A.3.3	SD karta . . . . .	69
A.3.4	Tlačítka . . . . .	70
A.4	Modul PSU . . . . .	71



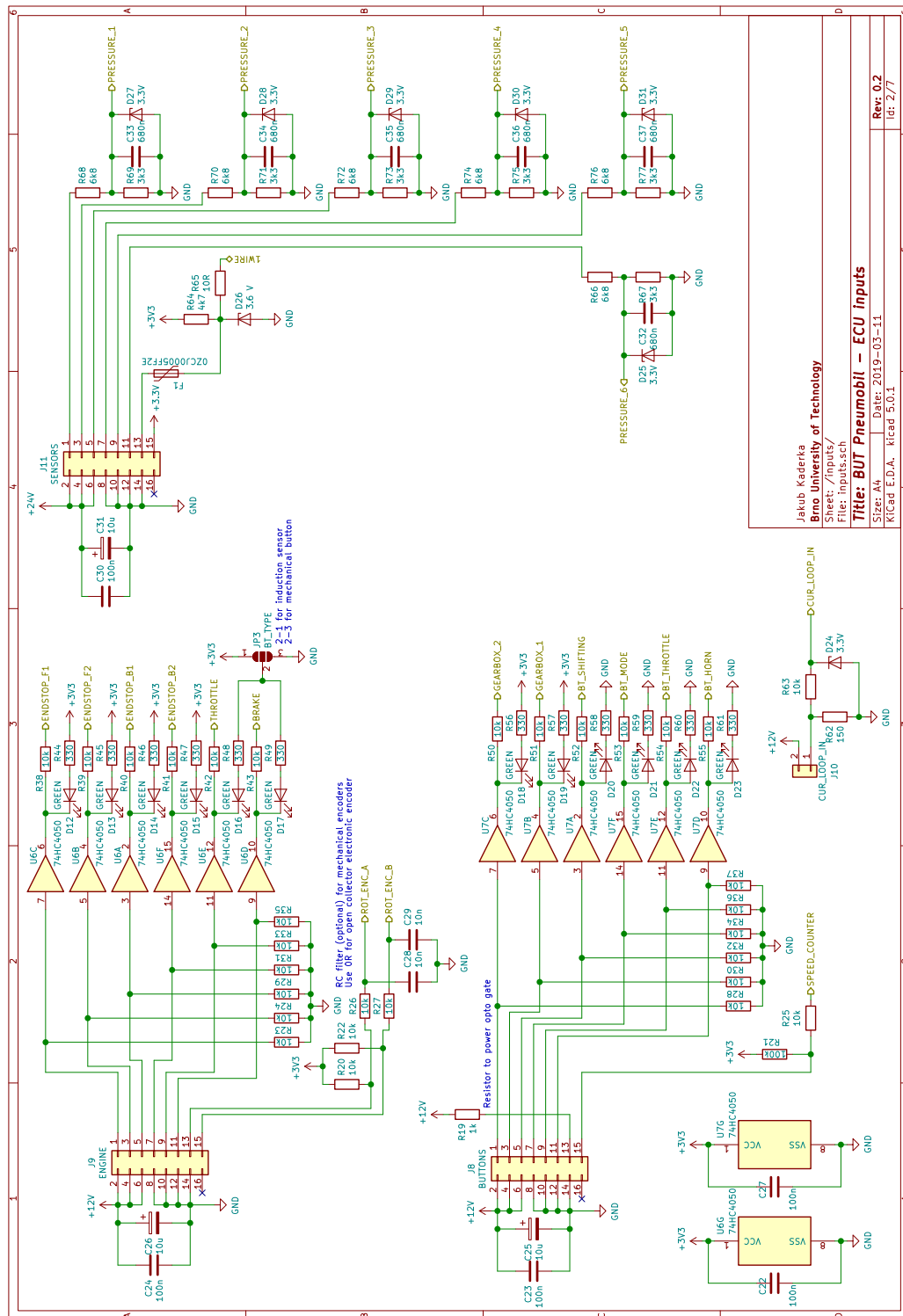
# A Schémata zapojení

## A.1 Modul ECU

### A.1.1 Hlavní list



## A.1.2 Vstupy



Jakub Kaderka  
Brno University of Technology  
Sheet: //inputs//  
File: inputs.sch

Title: BUT Pneumobil – ECU Inputs

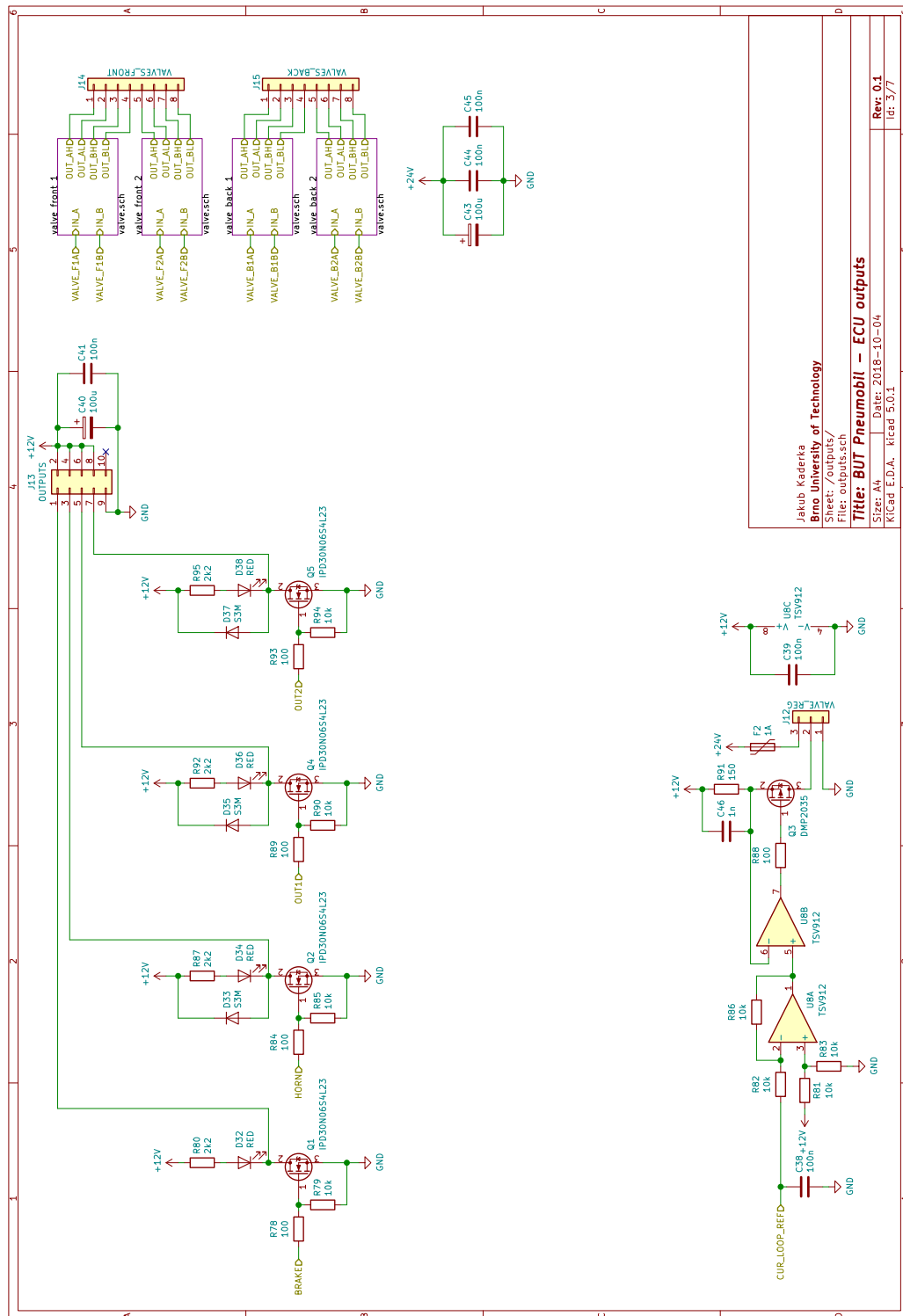
Size: A4 Date: 2019-03-11

KiCad E.D.A. Kicad 5.0.1

Rev: 0.2

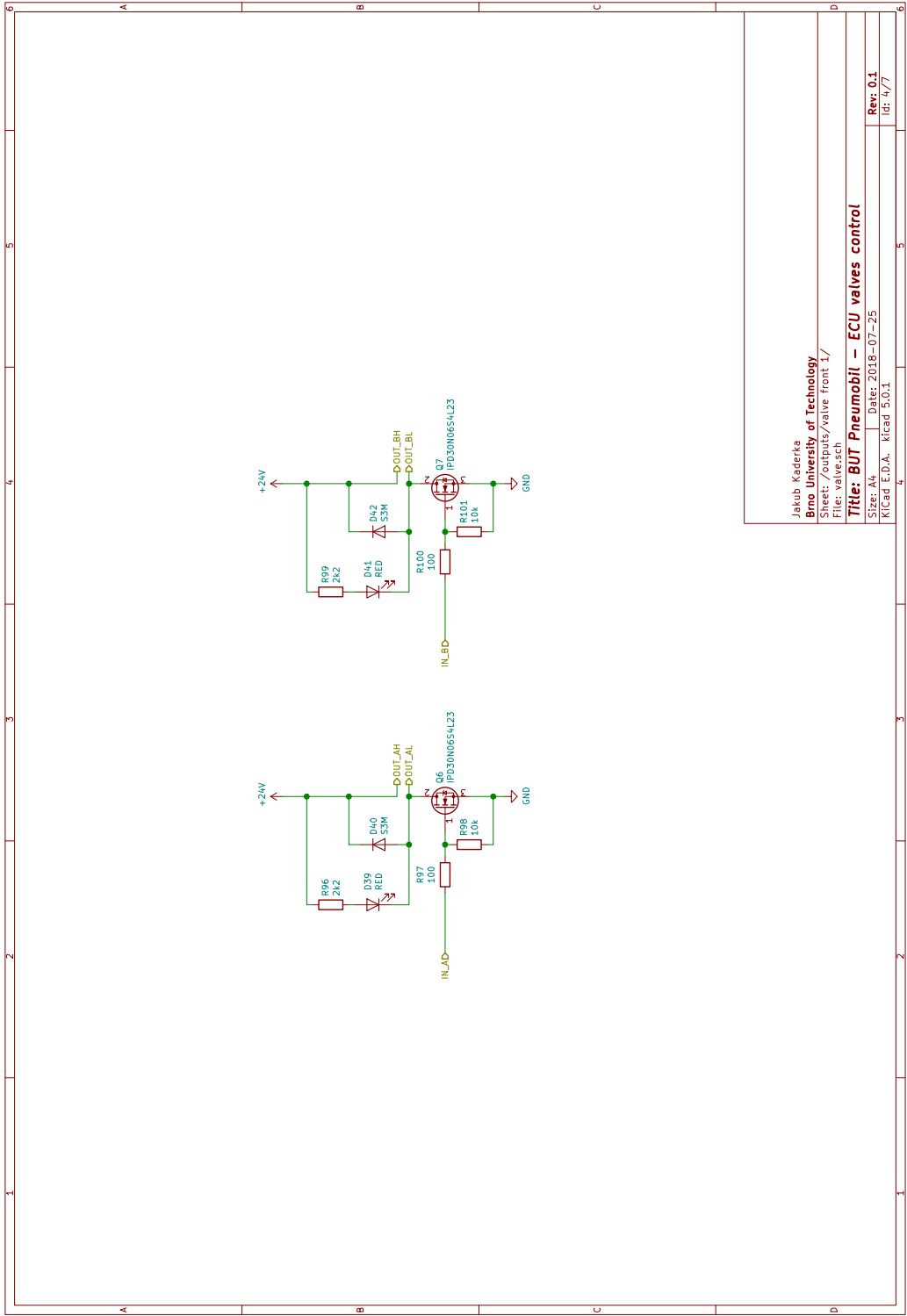
Id: 2/7

## A.1.3 Výstupy

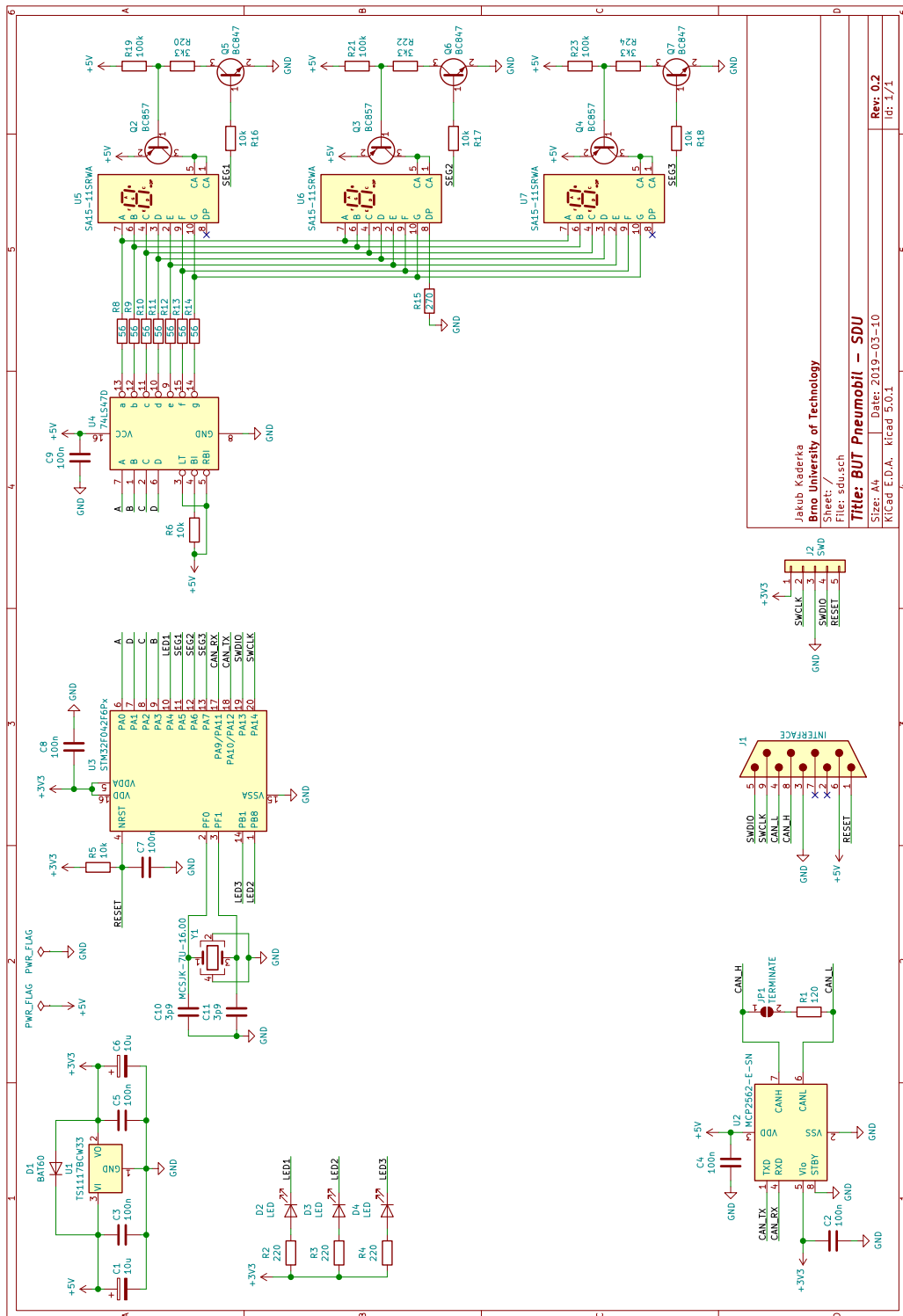




# A.1.4 Řízení ventilů

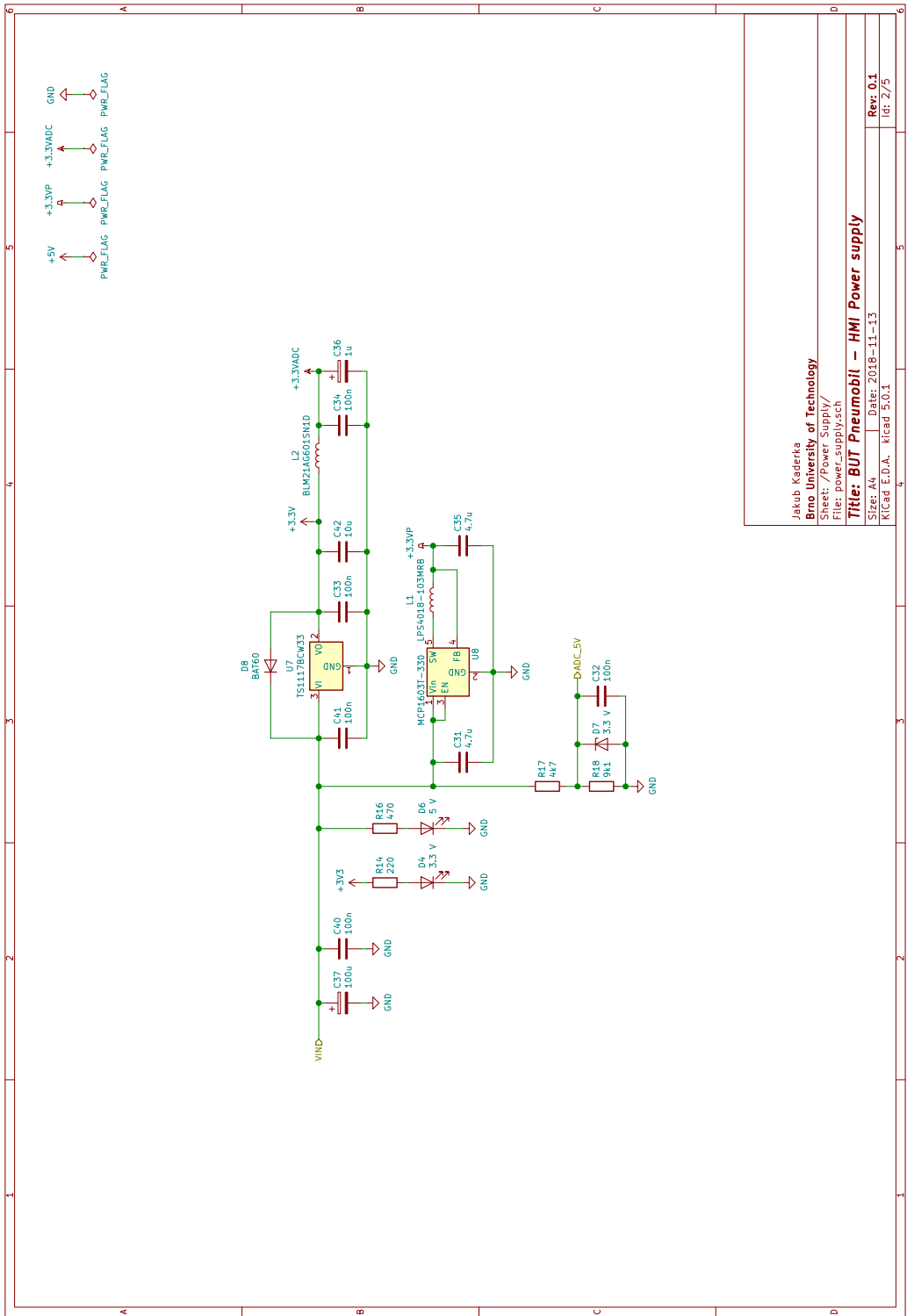


## A.2 Modul SDU

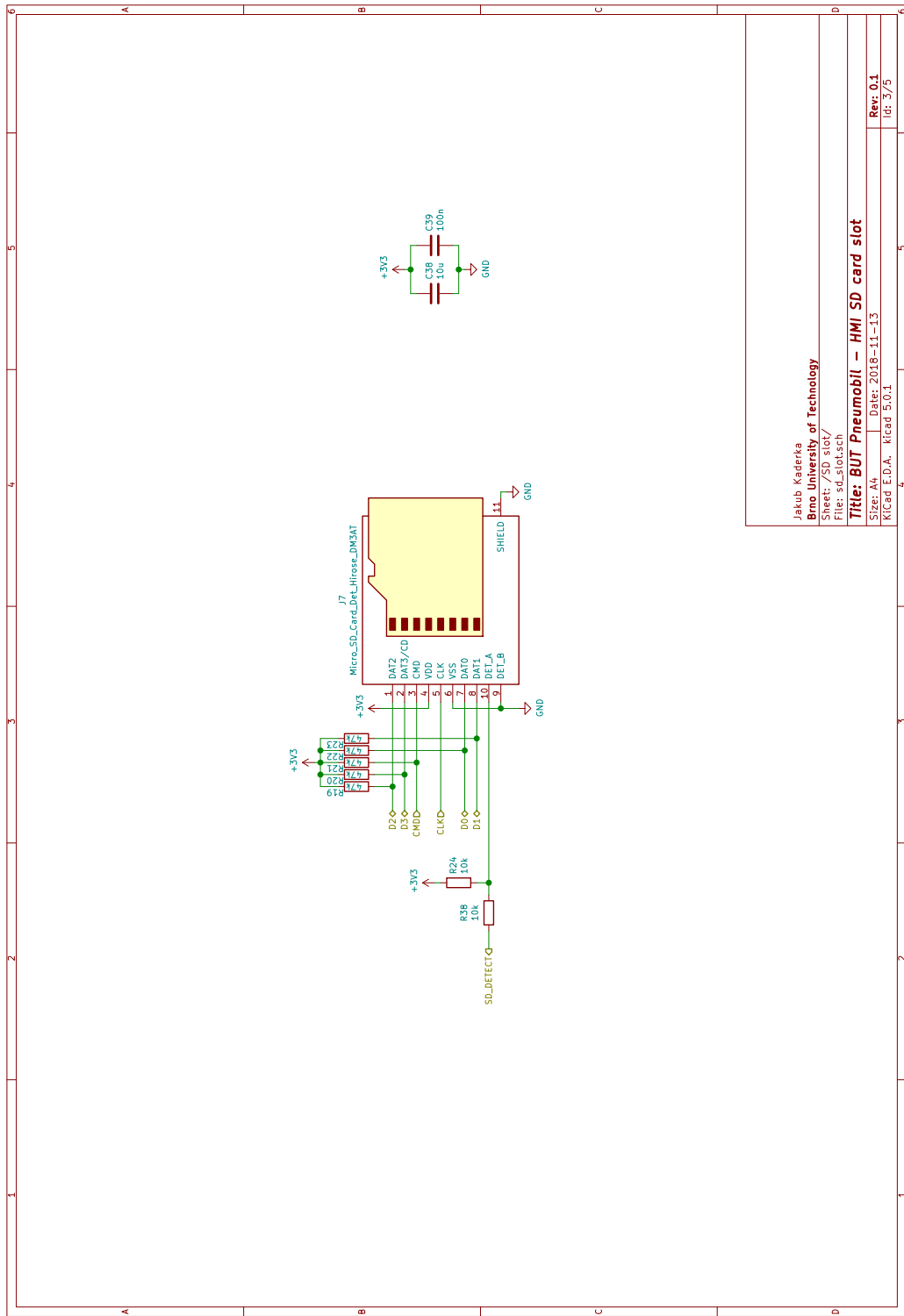




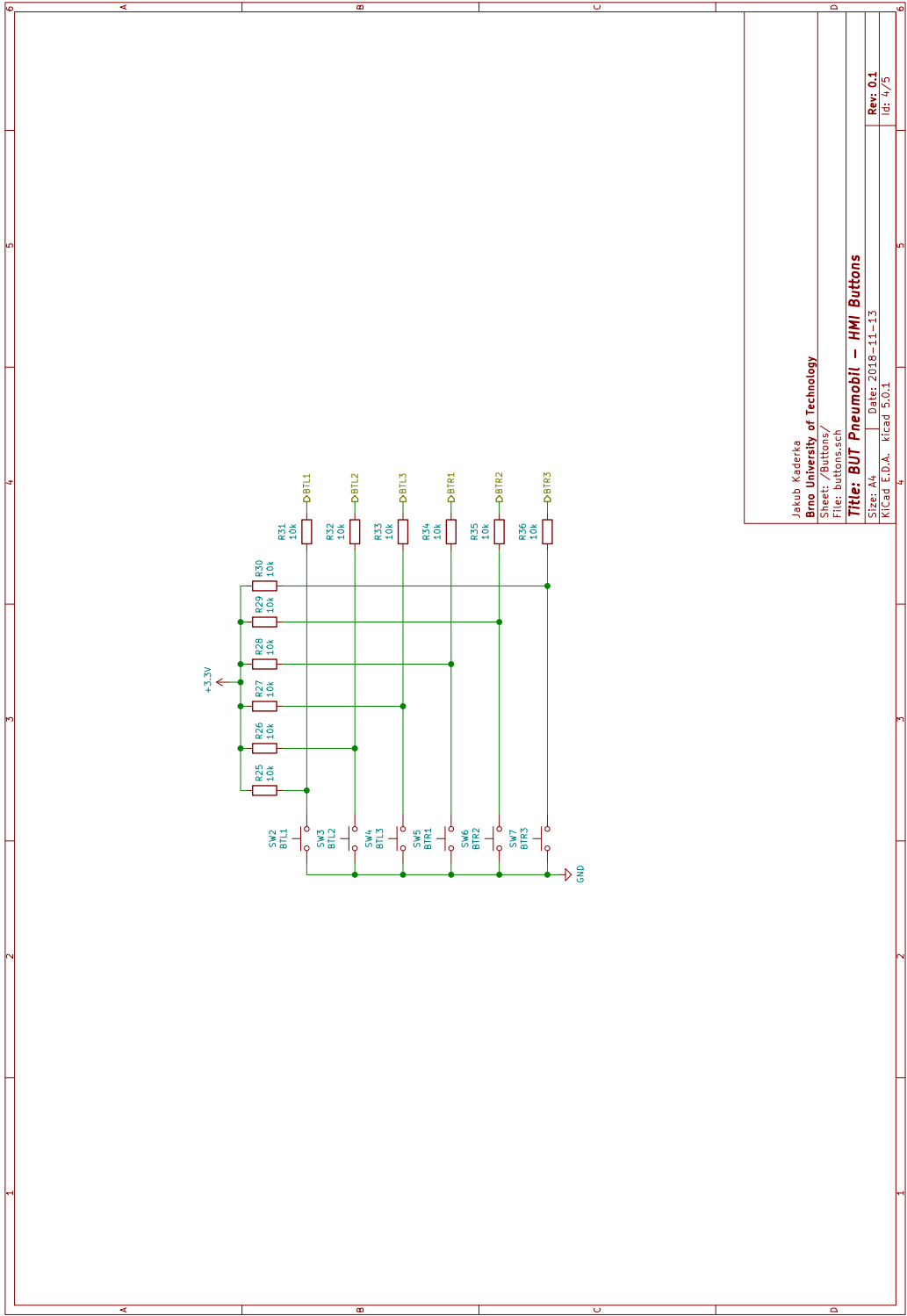
A.3.2 Napájecí zdroj



### A.3.3 SD karta



A.3.4 Tlačítka



## A.4 Modul PSU

