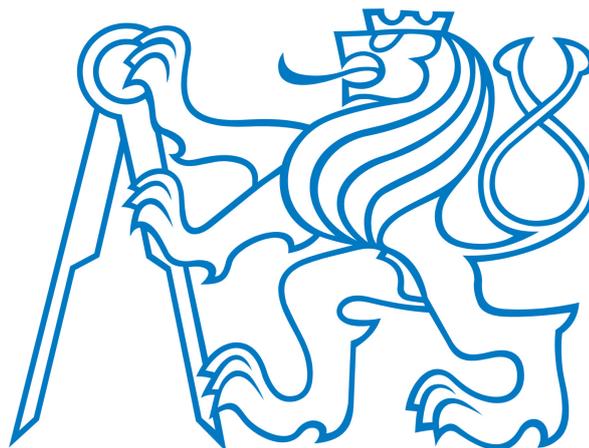


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

---

FAKULTA ELEKTROTECHNICKÁ, Katedra řídicí techniky



# Návrh a realizace řídicí jednotky malého proudového motoru

BAKALÁŘSKÁ PRÁCE

2008

Vypracoval: Michal Vosecký

---

Vedoucí práce: Ing. Ondřej Špinka

České vysoké učení technické v Praze  
Fakulta elektrotechnická

Katedra řídicí techniky

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Michal Vosecký**

Studijní program: Elektrotechnika a informatika (bakalářský), strukturovaný  
Obor: Kybernetika a měření

Název tématu: **Návrh a realizace řídicí jednotky malého proudového motoru**

Pokyny pro vypracování:

1. Prostudujte existující průmyslové řídicí jednotky, které jsou k dispozici v naší laboratoři.
2. Navrhněte a realizujte HW řídicí jednotky nebo použijte hardware již existující.
3. Navrhněte a implementujte základní softwarové vybavení.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. Ondřej Špínka

Platnost zadání: do konce zimního semestru 2008/2009

prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry



doc. Ing. Boris Šimák, CSc.  
děkan

V Praze dne 25. 2. 2008

## **Abstrakt**

Tato práce se zabývá základním návrhem řídicí jednotky (systému FADEC) pro malý proudový motor. Práce je navrhována tak, aby bylo možné finální celek propojit s uživatelským SW navrhovaným Miroslavem Hájkem.

První část obsahuje stručné seznámení s problematikou principu funkce a principu řízení modelářské turbíny. Dále je proveden návrh HW řídicí jednotky. Jako řídicí jednotka je použita deska SpejblARM s 32b mikrokontrolérem LPC2119. Na tuto řídicí jednotku byly implementovány základní nízkoúrovňové funkce. V další části je proveden návrh a realizace podpůrné I/O desky. V poslední části je provedeno shrnutí dosažených výsledků.

## **Abstract**

This thesis deals with conceptual design and construction of a digital control unit for modeler jet engine (FADEC). The unit is designed to be compatible with visualization software provided by a fellow student Miroslav Hájek.

A brief description of basic functional principles and properties of a small turbine is given in the first part. The control unit, built around the Philips LPC 2119 microcontroller is described in the following parts, along with some preliminary results acquired during lab testing.

## **Prohlášení**

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

V Praze dne .....

.....

(podpis autora)

## Poděkování

Chtěl bych poděkovat všem lidem, bez nichž by má bakalářská práce nemohla vzniknout. Děkuji především vedoucímu bakalářské práce Ing. Ondřeji Špínkovi za jeho vedení, připomínky, trpělivost a rady při zpracování této práce. V neposlední řadě bych chtěl poděkovat rodině a všem přátelům za podporu. Děkuji!

---

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
<b>2</b>	<b>Modelářská turbína</b>	<b>10</b>
2.1	Princip funkce turbíny . . . . .	10
2.2	Provozní režimy . . . . .	11
2.3	Vstupy a výstupy . . . . .	11
2.4	Start turbíny . . . . .	12
2.5	Běh turbíny . . . . .	13
<b>3</b>	<b>Princip funkce řídicí jednotky</b>	<b>14</b>
3.1	Popis vstupů a výstupů navrhované řídicí jednotky FADEC . . . . .	14
3.2	Princip řízení modelářské turbíny . . . . .	15
<b>4</b>	<b>Návrh HW řídicí jednotky</b>	<b>17</b>
4.1	Dostupné jednotky v laboratoři CAK . . . . .	17
4.1.1	LPC2119 . . . . .	18
<b>5</b>	<b>Návrh desky vstupů/výstupů</b>	<b>19</b>
5.1	Napájecí obvod . . . . .	19
5.2	Senzor otáček . . . . .	19
5.2.1	Měřicí obvod senzoru otáček . . . . .	20
5.3	Senzor teploty . . . . .	21
5.3.1	Měřicí obvod senzoru teploty . . . . .	21
5.4	Galvanické oddělení řídicích signálů . . . . .	22
5.4.1	Zapojení galvanického oddělení . . . . .	23
<b>6</b>	<b>SW vybavení řídicí jednotky</b>	<b>24</b>
6.1	Inicializační funkce . . . . .	24
6.2	Nízkoúrovňové funkce . . . . .	26
6.3	Komunikace řídicí jednotky s uživatelským počítačem . . . . .	27

OBSAH	7
6.3.1 Návrh stavového automatu komunikace . . . . .	27
<b>7 Realizace a výsledky</b>	<b>29</b>
<b>8 Závěr</b>	<b>31</b>
<b>Literatura</b>	<b>32</b>
<b>A Schémata</b>	<b>I</b>
<b>B Komunikační protokol</b>	<b>VI</b>
B.1 Protokol . . . . .	VI
<b>C Obsah CD</b>	<b>XIV</b>

# 1 Úvod

Modelářské proudové motory se v dnešní době těší velké oblibě jak z řad modelářů, tak z řad diváků. Modelářský proudový motor (dále jen turbína) se principem funkce příliš neliší od skutečného proudového motoru, z čehož vyplývá i nutnost použití kvalitní řídicí jednotky, která zaručí bezproblémový a bezpečný chod motoru.

V dnešní době se komerční výrobou modelářských motorů a jejich řídicích jednotek zabývá německá modelářská firma JetCat [3], která poskytuje velice široký sortiment zboží právě tohoto druhu. Mezi další konstruktéry řídicích jednotek patří Gaspar Espiell [5], jehož jednotky jsou ve velké oblibě, vyznačují se velikou spolehlivostí a životností. Dále je to americká firma Wren [4] a v neposlední řadě byly na toto téma zpracovány dvě bakalářské práce na katedře měření ČVUT-FEL (Fibich, Juříček).

Práce vychází a zároveň navazuje na bakalářskou práci zpracovanou Miroslavem Hájkem [1], která byla zaměřena na identifikaci turbíny, přičemž v této práci byla použita pro zpracování telemetrických dat řídicí jednotka na bázi Motoroly HC12.<sup>1</sup>

Cílem této práce je přeprogramovat stávající řídicí jednotku s mikrokontrolérem HC12, který je v dnešní době již zastaralý a málo dostupný na modernější platformu mikrokontroléru s jádrem ARM7. Dále navrhnout a vylepšit stávající zapojení desky vstupů/výstupů (dále jen I/O deska).

Hlavní výhody navrhované řídicí jednotky jsou:

- větší univerzálnost z hlediska použití, jednoduchost
- realizace na moderní platformě mikrokontroléru s jádrem ARM
- malé rozměry

Druhá kapitola se zabývá popisem turbíny, charakterizuje jednotlivé pracovní režimy, vstupní/výstupní signály, provozní režimy a popisuje start turbíny. Ve třetí kapitole je popsán princip funkce řídicí jednotky a princip řízení turbíny. Čtvrtá kapitola se zabývá výběrem vhodné řídicí jednotky z dostupných jednotek laboratoře CAK. Pátá kapitola

---

<sup>1</sup>16 bit mikrokontrolér Motorola HC12

---

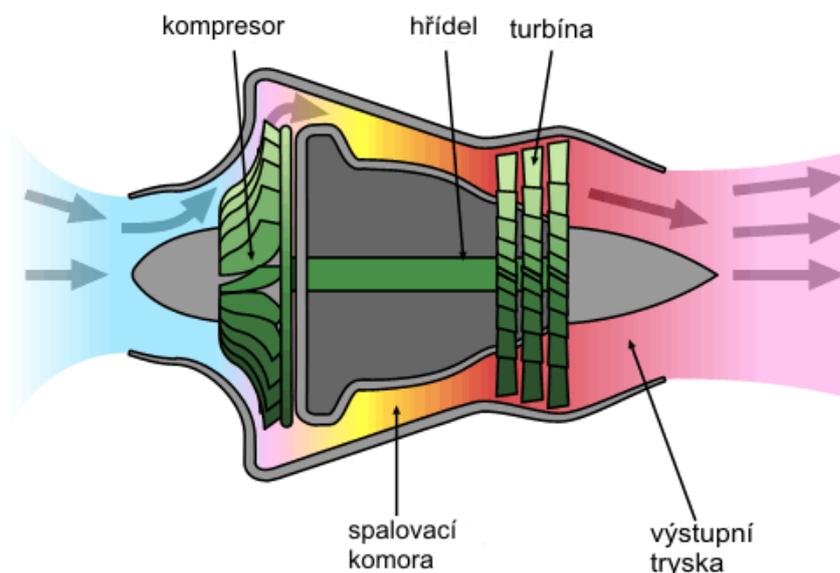
se zabývá návrhem I/O desky. V šesté kapitole je stručně naznačen princip použitých implementovaných programových funkcí a návrh možné realizace stavového automatu komunikace. V poslední sedmé kapitole je popsán a shrnut výsledek této práce.

## 2 Modelářská turbína

Modelářská turbína není v dnešní době už jen malý experiment, ale zcela běžně se s ní můžeme setkat ve formě pohonné jednotky ať letadla, či vrtulníku. Použití této pohonné jednotky s sebou ale přináší určitá specifika a odlišnosti oproti skutečné turbíně. Jedním výrazným specifikem je to, že oproti skutečné turbíně se u modelářské turbíny se zmenšující se velikostí turbíny musí zvyšovat počet otáček. To s sebou přináší vyšší nároky na použité materiály [12]. Další rozdíl oproti skutečné turbíně je ten, že na start turbíny se jako palivo používá propan-butan.

### 2.1 Princip funkce turbíny

Turbína pracuje na principu Newtonova zákona o akci a reakci, tedy spaliny vycházející z motoru působí opačnou silou na motor, který tím ženou vpřed. Schematicky je to znázorněno na obrázku 2.1



Obrázek 2.1: Principiální schéma turbíny (převzato a upraveno z [2])

Vzduch je nasáván vstupním otvorem do odstředivého (radiálního) kompresoru, kde je stlačen až na tlak 100 kPa. Usměrněný a zpomalený proud stlačeného vzduchu vstupuje do spalovací komory. Ta je tvořena pláštěm motoru. Zde se míchá tzv. primární

vzduch s rozprášeným palivem z palivových trysek a dochází k hoření při teplotě až 2000°C. Výstupní žhavé plyny dopadají pod správným úhlem na točící se lopatky axiální turbíny, která přes hřídel pohání kolo kompresoru. Pracovní otáčky jsou okolo 100 000 otáček/minutu. Za turbínou mají plyny o teplotě 500 až 600°C ještě dostatek energie k vyvození tahu [12].

## 2.2 Provozní režimy

V následující tabulce 1 jsou uvedeny základní pracovní režimy turbíny, které je nutné pro bezpečný chod turbíny dodržet.

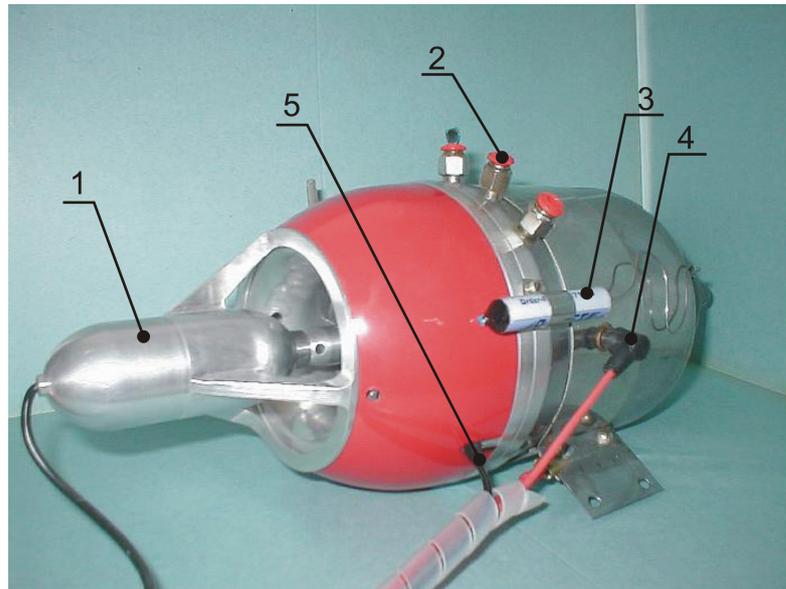
Pracovní režim	Otáčky [ot/min]
Volnoběh	25 000 - 30 000
Max. provozní výkon	80 000
Max. možný výkon	120 000

Tabulka 1: Tabulka pracovních režimů turbíny

Teplota výstupních plynů se pohybuje v rozmezí 450 - 800 °C, přičemž maximální možná teplota je 1000 °C.

## 2.3 Vstupy a výstupy

Vstupem turbíny jsou palivové trysky sloužící pro přívod paliva do spalovací komory. Dále je to žhavicí koncovka pro žhavicí svíčku, která slouží k počátečnímu zapálení plamene uvnitř spalovací komory. Dalším vstupem je startovací motor, který slouží během startovacího procesu k počátečnímu roztočení rotoru turbíny a při nevydařeném startu k vyfoukání propan-butanu nahromaděném ve spalovací komoře. To je obzvláště důležité, jinak by mohlo dojít k explozi turbíny. Startovací motor se používá také po zastavení turbíny ke chlazení. Výstupem turbíny je dopředný tah a měřené signály: teplota výstupních plynů a otáčky turbíny.



Obrázek 2.2: Vyznačení jednotlivých I/O částí turbíny

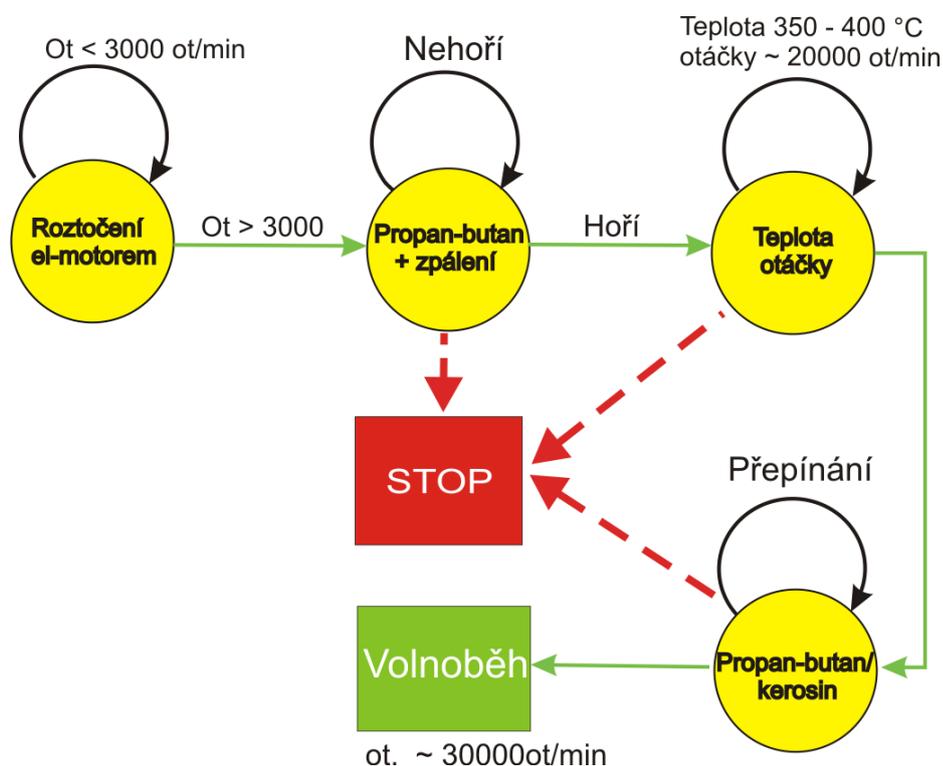
Na následujícím obrázku (obr. 2.2) jsou znázorněny jednotlivé vstupy a výstupy turbíny.

1. Startovací motor
2. Palivové trysky
3. Senzor teploty
4. Žhavicí koncovka
5. Snímač otáček

## 2.4 Start turbíny

Blokově je průběh startu znázorněn na obrázku 2.3. Start se provádí po roztočení turbíny startovacím motorem přibližně na 3000 ot/min. Poté je otevřen ventil s propan-butanem a za pomoci žhavicí svíčky se provede zapálení plynu ve spalovací komoře. Pokud se zapálení po určitém čase nepodaří, je proces zastaven, startovacím motorem se vyfouká přebytečný plyn ve spalovací komoře a začne se od začátku. Zapálení se detekuje nárůstem gradientu teploty výstupních plynů a otáček. Při podařeném zápalu

běží turbína na propan-butan dokud se teplota nepohybuje mezi 350 - 400 °C a otáčky okolo 20 000 ot/min. Poté se může za stálého sledování teploty a otáček provést přepnutí z propan-butanu na kerosin. Přepnutí je pozvolné, kdy se uzavírá ventil na propan-butan a otevírá na kerosin, přičemž teplota by měla stoupat k 500 °C a otáčky k 25 000 - 35 000 ot/min. Pokud je detekována jakákoliv chyba, je proces zastaven. Při úspěšném přepnutí běží turbína ve volnoběžných otáčkách.



Obrázek 2.3: Stavový diagram startu turbíny

## 2.5 Běh turbíny

Výkon turbíny se po úspěšném startovacím procesu ovládá přísunem paliva. Přísun paliva musí být ale pozvolný a musí odpovídat danému pracovnímu režimu. Například pokud bude přísun paliva moc veliký, palivo se nestačí spálit ve spalovací komoře a plameny začnou šlehat daleko z výstupní trysky. Naopak pokud bude množství paliva vzhledem k proudícímu vzduchu nedostatečné, tak proudící vzduch plamen "sfoukne" [12].

### 3 Princip funkce řídicí jednotky

Prakticky žádný proudový motor se v dnešní době neobejde bez řídicí jednotky - FADEC.<sup>2</sup> Jde o systém plně elektrického řízení chodu motoru [11]. Skládá se z řídicího počítače a nutného rozhraní na ovládací prvky motoru. Podstatou systému FADEC je zpracování pohybu plynové páky a jiných ovládacích prvků motoru v počítačové jednotce a až ta optimálním způsobem řídí samotný motor: přísun paliva, spouštěcí a zastavovací sekvenci. Výhody jednotky jsou:

- ochrana motoru v situacích spadající mimo jeho bezpečný pracovní režim
- změna chování motoru pouhým přeprogramováním jednotky
- automatické spouštění motoru
- monitoring a diagnostika

#### 3.1 Popis vstupů a výstupů navrhované řídicí jednotky FADEC

Na obrázku 3.1 je znázorněno blokově propojení řídicí jednotky se všemi vstupními a výstupními signály. Vstupními signály řídicí jednotky jsou:

- poloha plynové páky z modelářského přijímače (obrázek 3.2)
- teplota výstupních plynů turbíny
- otáčky turbíny

Výstupní signál tvoří:

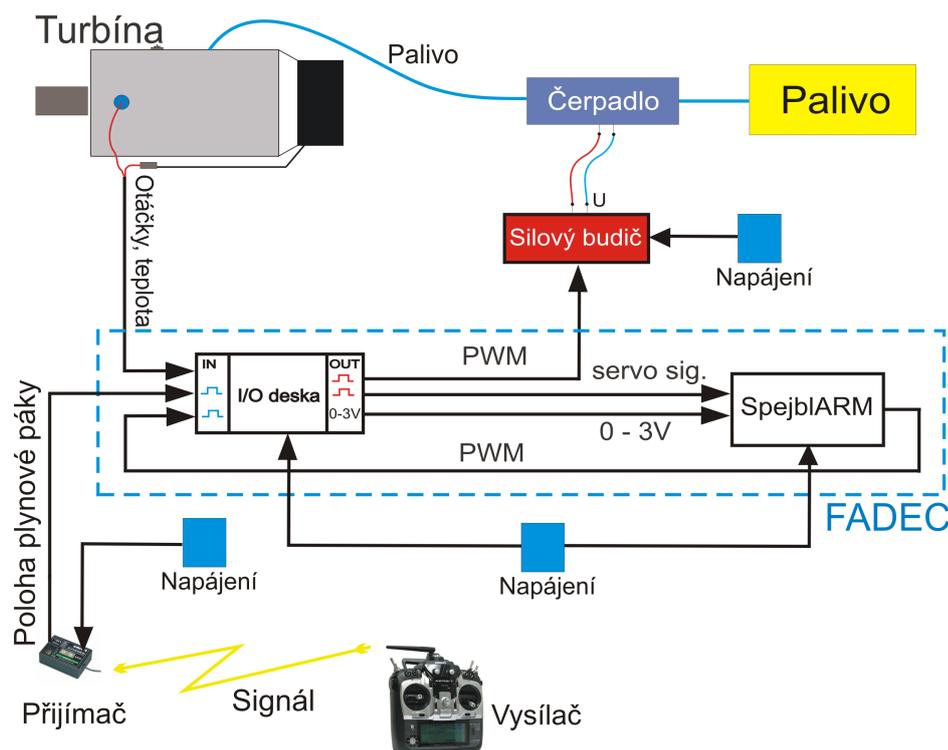
- PWM signál generovaný řídicí jednotkou do silového budiče, který ovládá napětí na motoru palivového čerpadla

V současné době se návrh řídicí jednotky provádí pouze pro jeden výstupní signál. Další možnost rozšíření výstupních signálů je například generování PWM pro řízení

---

<sup>2</sup>Full Authority Digital Engine Control - motorová řídicí jednotka

startovacího motoru, který se používá při startování na počáteční roztocení turbíny. Dále se tento startovací motor používá na vyfoukání propan-butanu při nepodařeném startu, aby se zabránilo možné explozi a pro chlazení turbíny po zastavení. Další výstup by mohl být spínání žhavicí svíčky, která slouží k počátečnímu zapálení plynu ve spalovací komoře. Tímto rozšířením by se pak startovací proces stal plně automatickým.

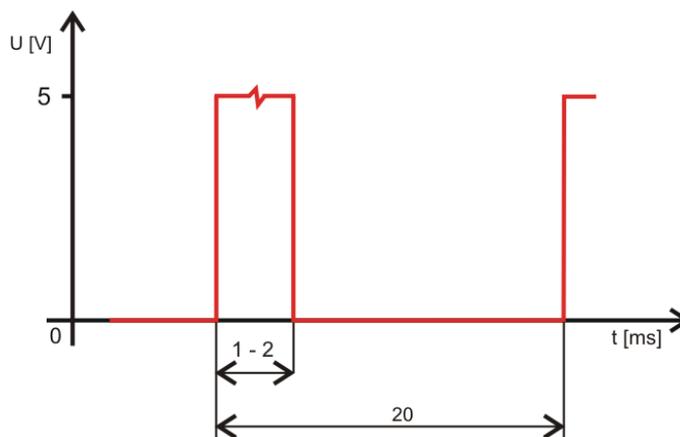


Obrázek 3.1: Blokové schéma navrženého zapojení řídicí jednotky

### 3.2 Princip řízení modelářské turbíny

Pro řízení turbíny je třeba sledovat gradient nárůstu/poklesu teploty a otáček. Dle těchto údajů pak dále ovládat palivové čerpadlo. Palivové čerpadlo je řízené pomocí silového budiče (obrázek 3.1), který je reprezentován modelářským regulátorem JES 05 [13]. Tento silový budič generuje výstupní napětí v rozmezí 0 - 5V, které tak ovládá otáčky čerpadla. Silový budič je řízen řídicí jednotkou, která generuje PWM signál na základě polohy plynové páky, počtu otáček a aktuální teploty. Řídicí signál silového budiče je stejný jako klasický servo signál modelářského přijímače (obrázek 3.2).

*Servo signál* je TTL signál o periodě  $20\text{ ms}$ , kde aktivní doba v logické "1" se pohybuje v rozmezí  $1 - 2\text{ ms}$  a to dle polohy páky na vysílači.



Obrázek 3.2: Servo signál modelářského přijímače

Pokud přijde požadavek na zvýšení/snížení výkonu (otáček) turbíny, je třeba k tomu úměrně generovat PWM signál pro silový budič. Řídicí jednotka generuje příslušný PWM signál za stálého sledování gradientu teploty výstupních plynů a otáček. Optimální regulace je taková, kdy je přísun palivové směsi nastaven tak, že změna k žádanému výkonu probíhá za konstantní teploty výstupních plynů. Pokud řídicí jednotka zaznamená nárůst gradientu teploty, nebo gradientu otáček opačným směrem vzhledem k žádaným, je požadavek na změnu výkonu pozastaven, provede se stabilizace chodu turbíny a až poté se pokračuje v požadavku na změnu výkonu.

## 4 Návrh HW řídicí jednotky

V této kapitole bude popsán výběr vhodné řídicí jednotky. Před samotným výběrem řídicí jednotky je třeba upřesnit si, co se vlastně od takové jednotky žádá. Hledaná jednotka by měla mít následující vlastnosti:

- mikrokontrolér by měl mít všechny potřebné periferie (A/D převodník, UART, PWM)
- musí poskytovat dostatečný výpočetní výkon pro řízení aplikace
- nízká spotřeba
- dostupnost
- SW podpora

### 4.1 Dostupné jednotky v laboratoři CAK

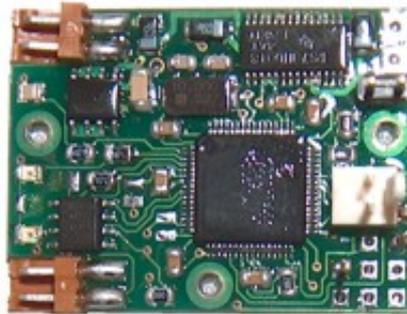
V této laboratoři se nachází řada hotových řídicích jednotek. Mezi nejpoužívanější patří jednotky s mikrokontroléry:

- *Motorola HC12D60*
- *H8S/2638F*
- *powerPC*
- *SH4*
- *LPC2119*

Pro tuto aplikaci byla vybrána řídicí jednotka s mikrokontrolérem LPC2119 [9]. S tímto mikrokontrolérem existuje ověřené fungující zapojení SpejblARM (obrázek 4.1) vytvořené pro projekt kráčejíci robot - SPEJBL [8]. Dále jsou na tento mikrokontrolér k dispozici vyzkoušené vývojové nástroje: cross-compiler GCC a systém OMK<sup>3</sup> vyvinutý

---

<sup>3</sup>OMK - OCERA make framework



Obrázek 4.1: Použitá řídicí deska SpejblARM

na katedře [7]. Tyto výhody vedly k upřednostnění LPC2119 před mikrokontrolérem H8S/2638F, na který není v současné době v laboratoři tak velká podpora. Další mikrokontroléry jako je powerPC nebo SH4 jsou velice výkoné, drahé a mají větší spotřebu. Jelikož řídicí aplikace bude ve finálním stavu napájena z baterií, nepřipadaly tyto mikrokontroléry v úvahu.

#### 4.1.1 LPC2119

##### Parametry:

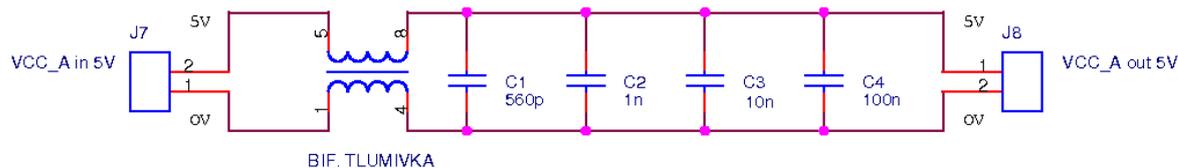
- Mikrokontrolér s 32 b jádrem ARM7TDMI-S
- Maximální frekvence CPU 60 MHz
- 16 kB on-chip Static RAM a 128 kB on-chip Flash
- In-System Programming (ISP), In-Application Programming (IAP)
- 2× UART rozhraní, 2× SPI a rychlé I<sup>2</sup>C rozhraní
- 2× 32 b timery a 6 PWM výstupů
- 8-kanálový 10 b AD převodník
- 1,8 V napájení jádra a 3,3 V napájení sběrnice

## 5 Návrh desky vstupů/výstupů

Tato kapitola se zabývá návrhem podpůrné I/O desky řídicí jednotky SpejblARM. Protože úroveň signálu jak teploty, tak otáček je malá, zvláště při vysokých otáčkách je nutné tyto signály patřičně upravit a až po té přivést ke zpracování řídicí jednotkou. Dále tato deska obsahuje galvanické oddělení signálu z modelářského přijímače do řídicí jednotky a PWM signálu pro silový budič (obrázek 3.1)

### 5.1 Napájecí obvod

Z obrázku 3.1 je patrné, že zdroj napájení je pro řídicí desku (SpejblARM) a I/O desku stejný. V řídicí desce je pak stabilizovaným napájením 5V (IO LE50CD<sup>4</sup>) napájen integrovaný obvod TPS73HD318, který vytváří 3,3V a 1,8V pro mikrokontrolér LPC2119 [8]. Pro I/O desku je napájecí část řešena tak, že do napájení je zařazen napájecí filtr (obrázek 5.1). Filtr se skládá z bifilární tlumivky a sady kapacitorů. Princip spočívá v tom, že bifilární tlumivka se pro střídavý signál chová jako zkrat (propustí stejnosměrný) a sada kondenzátorů slouží jako filtr na odsranění špiček v napájení. Jádro tlumivky je železo-prachové, na kterém je navinuto 17 závitů.

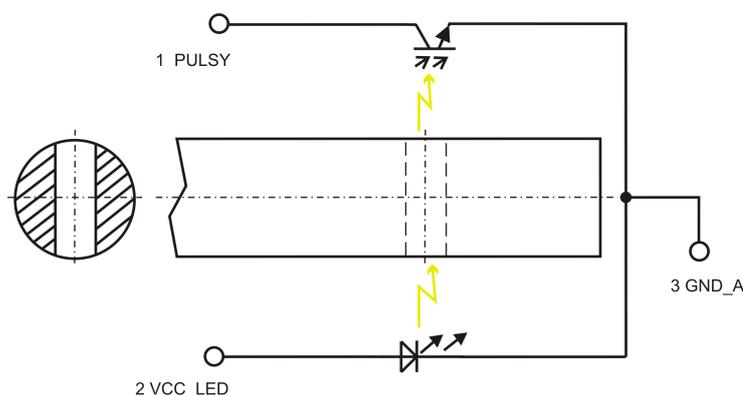


Obrázek 5.1: Schéma filtru napájení

### 5.2 Senzor otáček

Senzor otáček je tvořen infračervenou LED diodou a fototranzistorem. Na obrázku 5.2 je schematicky naznačeno měření otáček. V hřídeli turbíny je v podélné ose vyvrtán jeden otvor. V ose tohoto otvoru je umístěna na jedné straně infračervená dioda a na protější straně fototransistor. Tranzistor je tedy osvětlen každou otáčkou jednou.

<sup>4</sup>stabilizátor napětí, výstupní napětí je 5V



Obrázek 5.2: Princip měření otáček

### 5.2.1 Měřicí obvod senzoru otáček

Senzor otáček je připojen k měřicímu obvodu (obrázek 5.3) tak, že kolektor tranzistoru je připojen na PIN 1 konektoru J7. Tento pin je připojen také přes R7 (pull-up rezistor) k VCC. Operační zesilovač OZ1A je zapojen jako neinvertující zesilovač s napěťovým zesílením  $A = 49$ . Zesílení je voleno tak velké proto, že při maximálních otáčkách (okolo 100 000 ot/min) je úroveň osvětlení dopadající na fototranzistor velice malá a fototranzistor se nedokáže zcela plně otevřít. Za zesilovacím členem se nachází operační zesilovač OZ1B zapojený jako komparátor s hysterezí. Na výstupu tohoto komparátoru je obdélníkový signál o stejné frekvenci, jako je frekvence spínání fototranzistoru a amplitudě 5V. Tento výstup je pak následně připojen přímo na vstup registru CAP0.3<sup>5</sup>. Vstupy mikrokontroléru LPC2119 tolerují připojení 5V [9].

<sup>5</sup>CAP0.3 - záchytný vstupní registr mikrokontroléru LPC2119

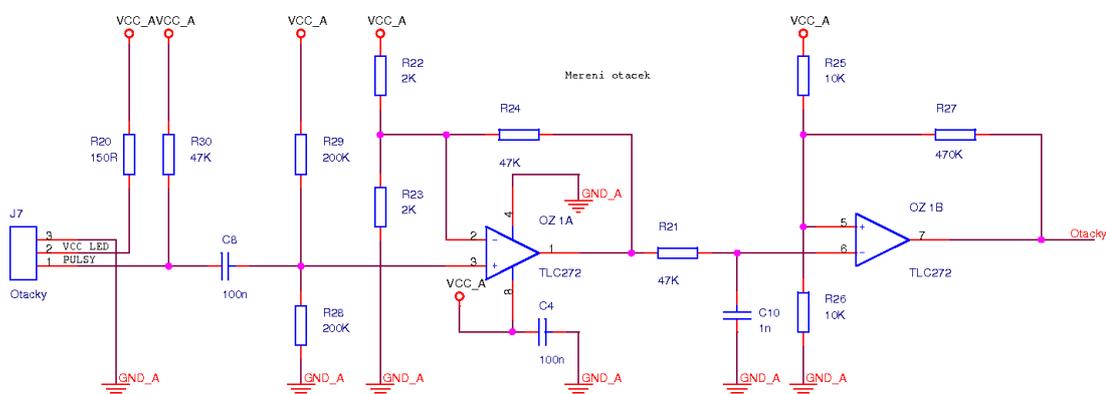
### 5.3 Senzor teploty

Jako senzor teploty je použit termoelektrický senzor teploty (termočlánek) typu K, jehož měřící konec je umístěn na plášti výstupní trysky. To je dobře patrné na obrázku 2.2. Při měření teploty spojem  $\sigma_M$  je nutné zaručit stejnou teplotu na obou srovnávacích spojích  $\sigma_S$ . Senzor teploty je zapojen pomocí dvou vodičového zapojení (obrázek 5.4). Při použití tohoto připojení je chyba měření nejvíce ovlivněna odporem přívodů k senzoru a vlivem kolísání srovnávacích teplot. Vliv kolísání teploty srovnávacích spojů se dá odstranit pomocí prodlužovacího vedení, kdy se posunou srovnávací spoje do míst kde je teplota konstantní, nebo pomocí kompenzační krabice [6]. Protože ale není potřeba měřit teplotu s tak velikou přesností, odchylka například  $\pm 10 \text{ }^\circ\text{C}$  je stále přijatelná, je dvou vodičové zapojení senzoru plně postačující.

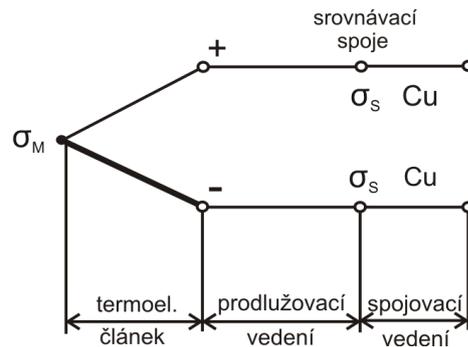
#### 5.3.1 Měřící obvod senzoru teploty

Měřící obvod pro měření teploty z termočlánu je na obrázku 5.5. Obvod se skládá z operačního zesilovače OZ3A, který je zapojen jako invertující zesilovač s napěťovým zesílením  $A = 96$ . Zesílení je voleno tak veliké, protože ze senzoru je zesilováno napětí o hodnotách řádově desítky mV. Senzor se připojí na konektor J1 na piny 3 a 1. Výstup zesilovače je připojen na vstupní pin mikrokontroléru LPC 2119 AIN0<sup>6</sup>.

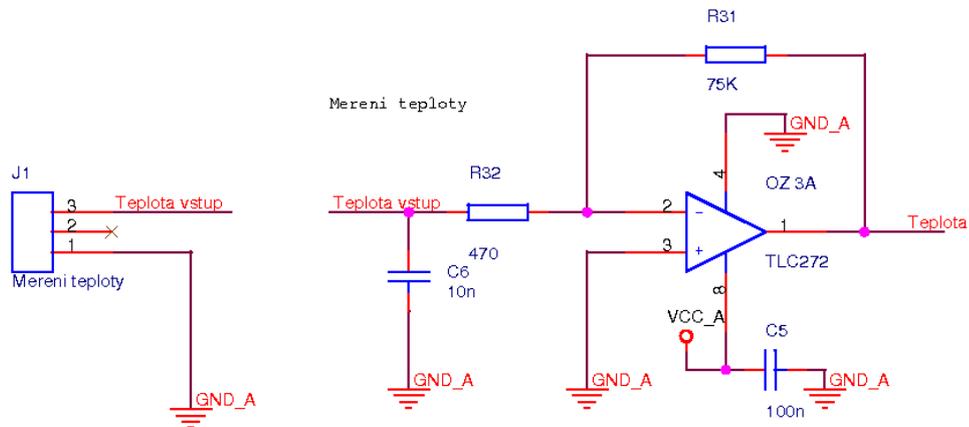
<sup>6</sup>AIN0 - vstupní pin registru PINSEL1



Obrázek 5.3: Měřící obvod pro senzor otáček



Obrázek 5.4: Dvou vodičové připojení senzoru teploty



Obrázek 5.5: Měřící obvod pro senzor teploty

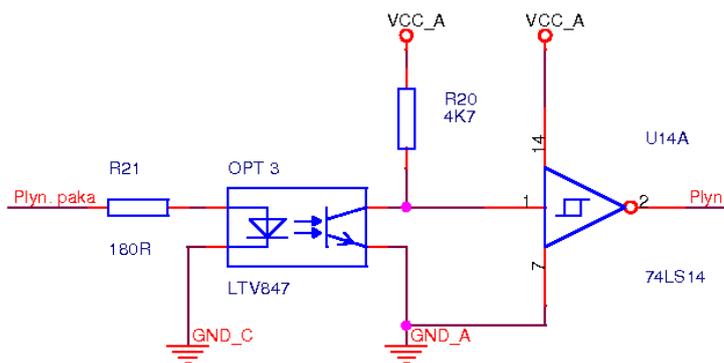
## 5.4 Galvanické oddělení řídicích signálů

Galvanickým oddělením RC přijímače od řídicí jednotky a silového budiče od zbytku elektroniky eliminujeme rušivé vlivy jednotlivých částí v napájení. Galvanicky jsou odděleny signály:

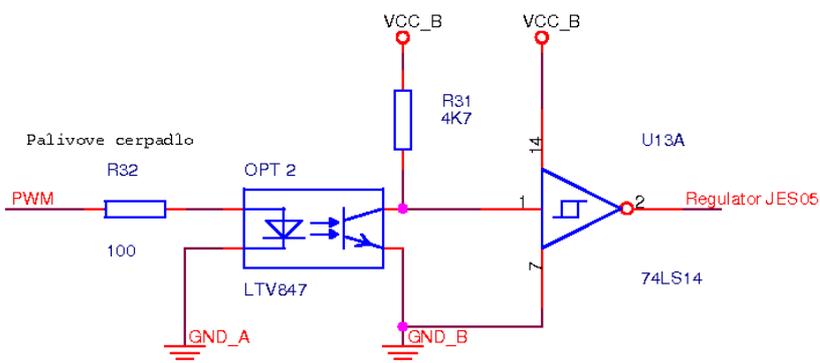
- signál z modelářského přijímače (obr.3.2) - poloha plynové páky (signál přiveden na vstup řídicí jednotky)
- signál PWM generovaný řídicí jednotkou (obr.3.2) - PWM signál (signál přiveden na vstup silového budiče)

### 5.4.1 Zapojení galvanického oddělení

Na obrázcích 5.6 a 5.7 jsou jednotlivá schémata zapojení pro oddělení signálů. Pro tvarování výstupního signálu je použit Schmittův klopný obvod.



Obrázek 5.6: Galvanické oddělení polohy plynové páky



Obrázek 5.7: Galvanické oddělení PWM

## 6 SW vybavení řídicí jednotky

V následující kapitole je popsán návrh a základní implementace SW vybavení pro řídicí jednotku. Dále je zde popsán komunikační protokol a návrh možné implementace komunikace řídicí jednotky s PC přes RS232. Jako vývojové prostředí byl zvolen OS Linux, cross-compiler GCC a OMK [7].

*Cross-kompilace* je proces, kdy se překládá zdrojový kód programu na jednom typu architektury pro jiný typ architektury. Výsledkem tohoto překladu je pak binární soubor spustitelný na cílové architektuře. V případě této aplikace se provádí *cross-kompilace* z architektury Intelx86 na architekturu ARM7

### 6.1 Inicializační funkce

Funkce, které provádí počáteční inicializaci použitých periférií. Jedná se o periferie:

- časovač TIMER0 a TIMER1
- A/D převodník
- UART0 - seriová linka
- PWM kanál

#### Implementace inicializačních funkcí:

`void timer_init ( uint32_t prescale0, unsigned irq_vect0, uint32_t prescale1, unsigned irq_vect1, uint32_t period )` Provádí inicializaci časovače TIMER0 a TIMER1. Parametry *prescale0* a *prescale1* nastavují předděličku časovače TIMER0 a TIMER1 na frekvenci čítání 10 MHz a 1 kHz. Parametry *irq\_vect0* a *irq\_vect1* představují velikost bazového posunu od základní adresy registru *VICVectAddr0*. Na tuto adresu je pak umístěna příslušná rutina obsluhy přerušení. Čím menší je bazový posun, tím je priorita dané obsluhy vyšší. Parametr *period* nastavuje hodnotu, do které registr T1MR0 čítá.

Registru T1MR0 stále čítá do nastavené hodnoty ( $period = 40$ ). Při přetečení této hodnoty provede reset registru, nastaví svůj výstup do logické "1" a vyvolá přerušení (obslužná rutina `timer1_irq_handler( )`). Výsledný efekt je, že přerušení je voláno periodicky s frekvencí 25 Hz.

`void pwm_channel ( int n )`

Funkce z knihovny `pwm.h` (Marek Peca). Argument  $n$  nastavuje příslušný PWM kanál. Na generování PWM signálu je v této aplikaci používán kanál PWM2 ( $n = 2$ ). Funkce automaticky nastaví daný registr jako výstupní. Více o možnosti PWM viz datasheet [9].

`void pwm_init ( uint32_t prescale, uint32_t period )`

Funkce z knihovny `pwm.h` (Marek Peca). Argument `prescale` představuje hodnotu předděličky PWM časovače a argument `period` velikost periody generovaného PWM signálu. Pro tuto aplikaci je perioda nastavena na 50 Hz ( $prescale = 1$ ) a PWM časovač čítá frekvencí 10 MHz ( $period = 200\ 000$ ).

`void pwm_set ( int_t n, uint32_t when )`

Funkce z knihovny `pwm.h` (Marek Peca). Argument  $n$  představuje výběr PWM kanálu a argument `when` hodnotu, do kdy je výstup daného PWM kanálu v logické "1". Poté je výstup v logické "0". Tento PWM signál se opakuje stále do té doby, dokud se nezmění argumenty této funkce.

`void adc_init ( uint32_t clock_div )`

Inicializační funkce pro A/D převodník. Argument `clock_div` nastavuje frekvenci vzorkování, která je v této aplikaci volena 4.5 MHz (maximální frekvence). A/D převodník je nastaven tak, aby se automaticky spouštěl při změně výstupu registru T1MR0 z logické "0" na logickou "1".

```
void UART_init ( uint8_t uart_num, uint32_t baud_rate, unsigned rx_isr_vect );
```

Funkce z knihovny *uart\_zen.h* (Ondřej Špínka) provádí inicializaci komunikačního rozhraní. Argument *uart\_num* označuje výběr konkrétního rozhraní (UART0, UART1), argument *baud\_rate* označuje rychlost komunikace v bps (bit/s) a argument *rx\_isr\_vect* nastavuje prioritu přerušení dané periférie. Pro tuto aplikaci je používán pro komunikaci UART0, rychlost přenosu 38 400 bps a priorita přerušení 13.

## 6.2 Nízkoúrovňové funkce

Funkce zpracovávají vstupní signály (teplota, otáčky, poloha plynové páky) do řídicí jednotky (SpejblARM). Hodnoty zpracovaných signálů tvoří vstupní informace pro řízení turbíny.

### Implementace nízkoúrovňových funkcí:

```
void timer0_irq_handler ( void )
```

Obsluha přerušení od časovače TIMER0. Přerušení se odchytlí při náběžné a sestupné hraně na vstupu registru CAP0.2 (vstupní signál - poloha plynové páky). Výstupem je tedy hodnota šířky pulsu v logické "1", která se uloží do globální proměnné *uint32\_t width\_pulse*. Dále je toto přerušení odchyceno každou náběžnou hranou na vstupu registru CAP0.3 (vstupní signál - otáčky). Výstupem je tedy počet otáček / minutu.

```
void timer1_irq_handler ( void )
```

Obsluha přerušení slouží k synchronizaci jednotlivých periférií. Je odchycena každým přetečením registru T1MR0 (tj. každých 25 Hz). V obsluze se nastaví příznak *timer1\_int\_flag = 1* a aktuální naměřená data (šířka pulsu, počet otáček, teplota) se pošlou pomocí funkce **write\_UART\_data** (*uint8\_t uart\_num*, *uint8\_t data*) přes UART0 do monitorovacího počítače.

*void* **write\_UART\_data** ( *uint8\_t* *uart\_num*, *uint8\_t* *data* )

Funkce z knihovny *uart\_zen.h* (Ondřej Špínka) pošle data přes rozhraní UART. Argumet *uart\_num* označuje, které rozhraní je pro přenos vybrané (UART0, UART1), a argument *data* představuje samotná data typu *char* pro přenos.

*int32\_t* **read\_AD** ( *int32\_t* *channel* )

Funkce vrací zkonvertovanou hodnotu teploty. Argument *channel* představuje vybraný A/D převodník, a tím i vstupní registr A/D převodníku. V této aplikaci je použit registr AIN0.

*int* **i2str** ( *char* \**s*, *long val*, *int len*, *unsigned base* )

Funkce slouží pro převod čísla datového typu *long* na řetězec *char*. Vstupní argument *char* \**s* je pointer na pole typu *char*, kde je uloženo překonvertované číslo, argument *long val* představuje číslo pro konverzi, argument *int len* je minimální délka řetězce doplněného nulami a *unsigned base* představuje soustavu, do které se převod provádí.

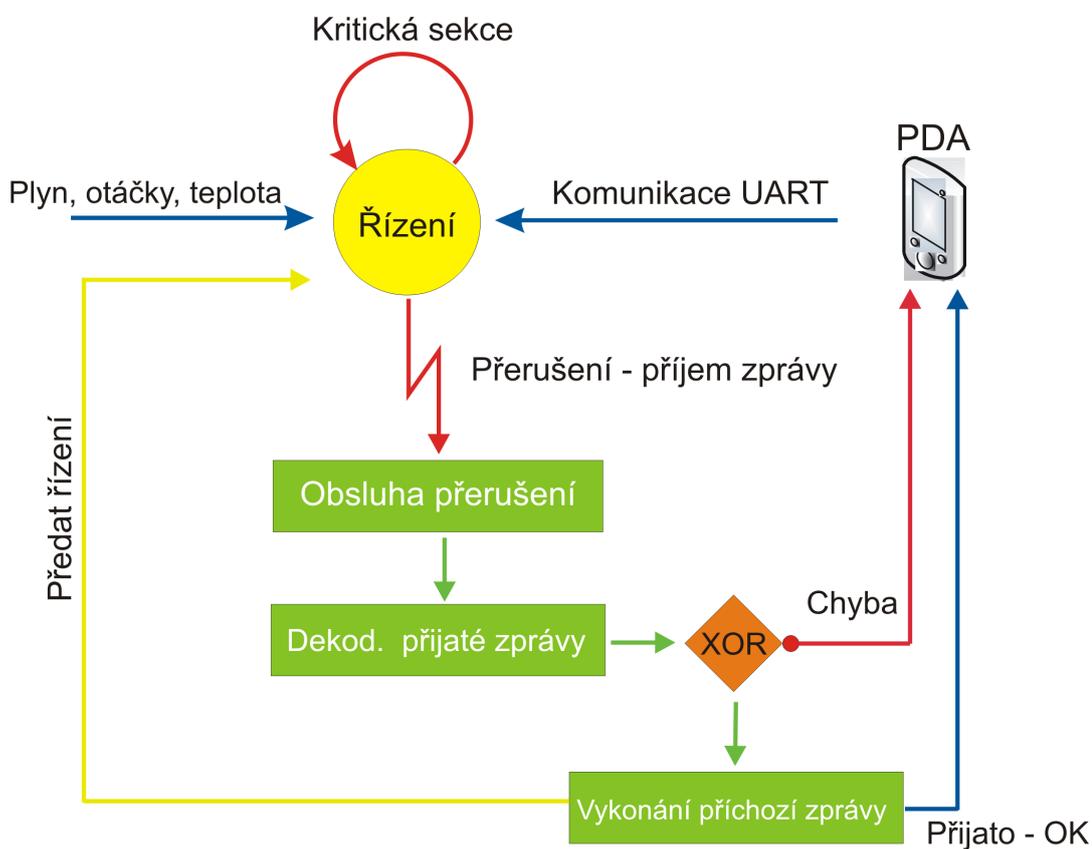
### 6.3 Komunikace řídicí jednotky s uživatelským počítačem

Komunikace řídicí jednotky s uživatelským počítačem je nejjednodušší způsob jak lze provádět inicializaci, diagnostiku, monitoring a nastavení jednotlivých parametrů řídicí jednotky. Uživatelský program je zpracováván M. Hájkem a je navrhován pro osobní počítač a PDA. Dále byl navržen komunikační protokol mezi uživatelským počítačem a řídicí jednotkou, který je podrobně popsán v příloze. Na straně řídicí jednotky je třeba vyřešit způsob změny chování řídicí jednotky dle přijatých zpráv komunikačním kanálem z PC nebo PDA.

#### 6.3.1 Návrh stavového automatu komunikace

Nabízenou možností implementace změny chování řídicí jednotky je pomocí stavového automatu (obrázek 6.1). Řídicí jednotka provádí algoritmus řízení do té doby, dokud

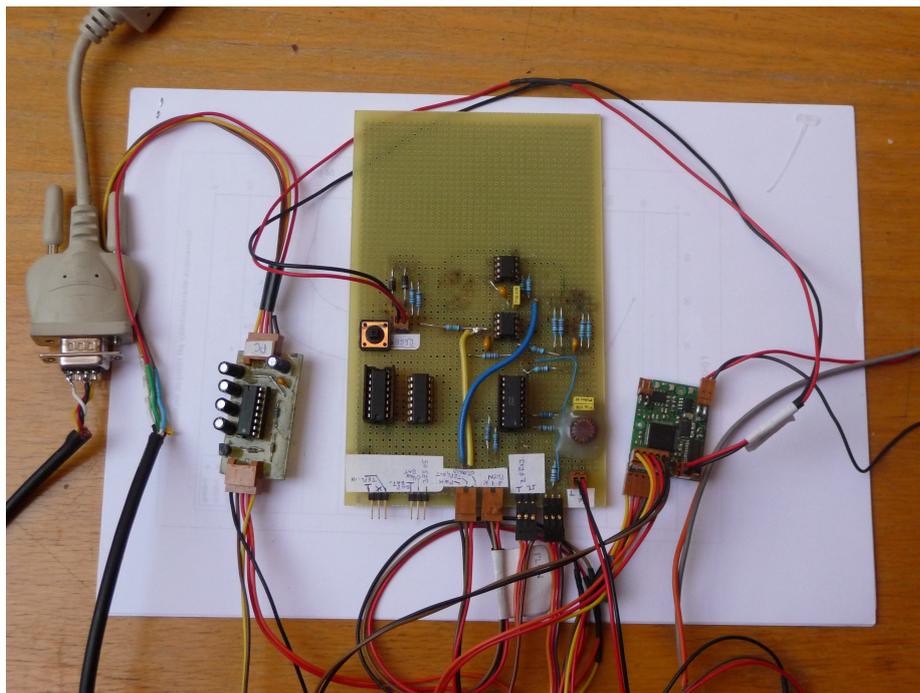
nepřijde přerušení přijetí zprávy z periferie UART0. Po vyvolání přerušení se zavolá obslužná rutina přerušení, která musí být co nejrychlejší. V této obsluze se bude testovat, zda přijatá zpráva je či není kompletní. To lze velice jednoduše, protože další dva znaky za úvodním znakem udávají délku zprávy. Pokud je zpráva kompletní, může se přejít na další stav. *Dekódování přijaté zprávy* se může tedy pouze provádět až poté, co je zpráva kompletní. Po dekodování se vyhodnotí příčná kontrolní parita a v případě chyby se vyše uživatelskému počítači zpráva o chybě. Ten vyše danou zprávu znova a celý cyklus se opakuje od začátku. Pokud vše proběhlo v pořádku, provede se dle dekodované zprávy nastavení příslušných parametrů.



Obrázek 6.1: Stavový automat komunikace

## 7 Realizace a výsledky

I/O deska byla realizována pouze na univerzálním plošném spoji kvůli snadné změně zapojení. Zhotovená deska a její propojení desky s řídicí jednotkou (SpejblARM) a uživatelským počítačem je na obrázku 7.1.



Obrázek 7.1: Realizované zapojení

Velký nárok byl kladen na použité konektory, protože právě konektory mohou být největším potencionálním zdrojem poruchy. Proto byly vybrány konektory *PSH02-05W* které svou houževnatostí provedení tyto požadavky splňují. Protože výsledná aplikace je napájena napětím 5 V, bylo nutné vybrat operační zesilovač, který dovolí nesymetrické napájení při zaručeném správném chodu. V návrhu byl použit operační zesilovač TLC272 [14], který požadavek splňuje a dále je to zesilovač typu *rail to rail*. To znamená, že saturační napětí se blíží napájecímu.

Zapojení bylo testováno v laboratoři za použití dvou laboratorních napájecích zdrojů a generátoru TTL signálu. Generátor byl použit k emulaci senzoru otáček a servo signálu (otáčky turbíny, poloha plynové páky) a jeden napájecí zdroj k emulaci senzoru teploty.

Všechny tyto vstupní signály byly připojeny na I/O desku a dále pak zpracovávány řídicí jednotkou, výsledek pak přes rozhraní RS232 vyslán do uživatelského počítače. Přesnost měření všech signálů se ukázala jako plně dostačující, a tudíž výsledky vhodné jako vstupní parametry pro řízení turbíny.

## 8 Závěr

Pro řízení aplikace byla vybrána řídicí jednotka SpejblARM s mikrokontrolérem LPC2119. Na jednotku jsou vytvořené vývojové prostředky poskytuje dostatečný výpočetní výkon, který se dá použít pro testování nevhodnějších řídicích algoritmů a má nejpriznivější poměr výpočetní výkon/spotřeba.

V dalším bodě dle zadání byla vytvořena podpůrná I/O deska řídicí jednotky (kapitola 5). Vzhledem k malé úrovni výstupních signálů z turbíny (otáčky, teplota) je realizace této desky nutná. Deska je navržena tak, aby byl minimalizován možný rušivý vliv v napájení od řídicí jednotky a silových částí a tím zabráněno chybovému běhu celé jednotky. Návrh a realizace se ukázaly jako vhodné a použitelné.

Byl implementován základní software pro zpracování telemetrických dat turbíny a vstupních dat od modelářského vysílače. Dále byl proveden návrh možné implementace komunikace řídicí jednotky s uživatelským počítačem nebo PDA zařízením.

Celé navržené propojení (I/O deska, řídicí jednotka) bylo testováno v laboratoři, kde vstupní signály otáčky a poloha plynové páky byly emulovány generátorem TTL signálu a senzor teploty emulován napěťovým zdrojem. Pro zjištění správnosti měřených dat byly zpracované naměřené údaje posílány přes RS232 do uživatelského počítače, kde se výsledek vypisoval do konzole. Propojení se ukázalo jako funkční a zaznamenané údaje správné.

Vyhlídky do budoucna pro tuto práci jsou: otestovat stávající zapojení na reálné turbíně, doimplementovat komunikaci s uživatelským počítačem nebo PDA a doimplementovat regulátor. Po zvládnutí všech těchto úkolů se může provést rozšíření výstupních řídicích signálů jednotky o PWM signál pro startovací motor a spínací signál pro žhavicí svíčku (kapitola 3.1). Poté se tato aplikace může stát zajímavým zdrojem informací pro testování různých řídicích algoritmů.

## Reference

- [1] HÁJEK M., *Identifikace proudového motoru pro model letadla* [online]. bakalářská práce, Praha 2006. [cit. 2008-03-09].  
URL: <[http://dce.felk.cvut.cz/knihovna/diplomky/2006/bp\\_2006\\_hajek\\_miroslav/bp\\_2006\\_hajek\\_miroslav.pdf](http://dce.felk.cvut.cz/knihovna/diplomky/2006/bp_2006_hajek_miroslav/bp_2006_hajek_miroslav.pdf)>.
- [2] WIKIPEDIA, *Proudový motor* [online]. 2008 [cit. 2008-06-30].  
URL: <[http://cs.wikipedia.org/wiki/Proudov\\_motor](http://cs.wikipedia.org/wiki/Proudov_motor)>.
- [3] JETCAT, *JetCat Germany* [online]. 18.června 2008. [cit. 2008-04-09].  
URL: <<http://www.jetcat.de/>>.
- [4] WREN, *WREN - turbine technology* [online]. 2008. [cit. 2008-07-09].  
URL: <<http://wrenturbines.co.uk/>>.
- [5] GASPAR ESPIELL, *Gaspar Espiell FADEC* [online]. 2008. [cit. 2008-06-09].  
URL: <<http://www.uknets.co.uk/rcplansguide/faau05.pdf>>.
- [6] RIPKA P., ĎADO S., KREIDEL M., NOVÁK J., *Senzory a převodníky*, strana 84, Vydavatelství ČVUT, Praha 2005 ISBN 80-01-03123-3
- [7] SOJKA M., PÍŠA P., *OMK Make-System Homepage* [online]. 28. dubna 2008. [cit. 2008-07-05].  
URL: <<http://rtime.felk.cvut.cz/omk/>>.
- [8] PECA M., *Spejbl ARM* [online]. 2008. [cit. 2008-06-15].  
URL: <<http://www.duch.cz/spejbl/elektronika.html#3-3>>.
- [9] LPC2119, *Datasheet LPC2119*, NXP (Philips) [online]. 2008. [cit. 2008-03-08]  
URL: <[http://www.nxp.com/acrobat\\_download/usermanuals/UM\\_LPC21XX\\_LPC22XX\\_2.pdf](http://www.nxp.com/acrobat_download/usermanuals/UM_LPC21XX_LPC22XX_2.pdf)>.
- [10] LPC2119 ERRATA, *Errata sheet LPC2119*, NXP (Philips) [online]. 17. května 2006, version 2.1. [cit. 2007-08-18].

- URL: <http://www.standardics.nxp.com/support/documents/microcontrollers/pdf/errata.lpc2119.pdf>.
- [11] KUSSIOR Z., *Samostatná digitální řídicí jednotka* [online]. 2008. [cit. 2008-07-02].  
URL: <http://www.mzak.cz/pojmy.php?print>.
- [12] AMATÉRSKÉ TUBÍNY, *Stránky pro příznivce modelářských turbín* [online]. 2003. [cit. 2008-06-09].  
URL: <http://mklipence.wz.cz/turbiny.htm>.
- [13] JES 05, *Elektronický regulátor otáček elektromotoru* [online]. 2008. [cit. 2008-06-09].  
URL: [http://www.jetimodel.cz/cze/navody/jes\\_05\\_14comp\\_10\\_18.pdf](http://www.jetimodel.cz/cze/navody/jes_05_14comp_10_18.pdf).
- [14] TLC272, *Datasheet TLC272* [online]. 2008. [cit. 2008-03-06].  
URL: <http://www.hep.upenn.edu/SNO/daq/parts/tlc272.pdf>.
- [15] LTV847, *Datasheet LTV847* [online]. 2008. [cit. 2008-03-09].  
URL: [http://www.datasheetcatalog.com/datasheets\\_pdf/L/T/V/8/LTV847.shtml](http://www.datasheetcatalog.com/datasheets_pdf/L/T/V/8/LTV847.shtml).

## **A Schémata**

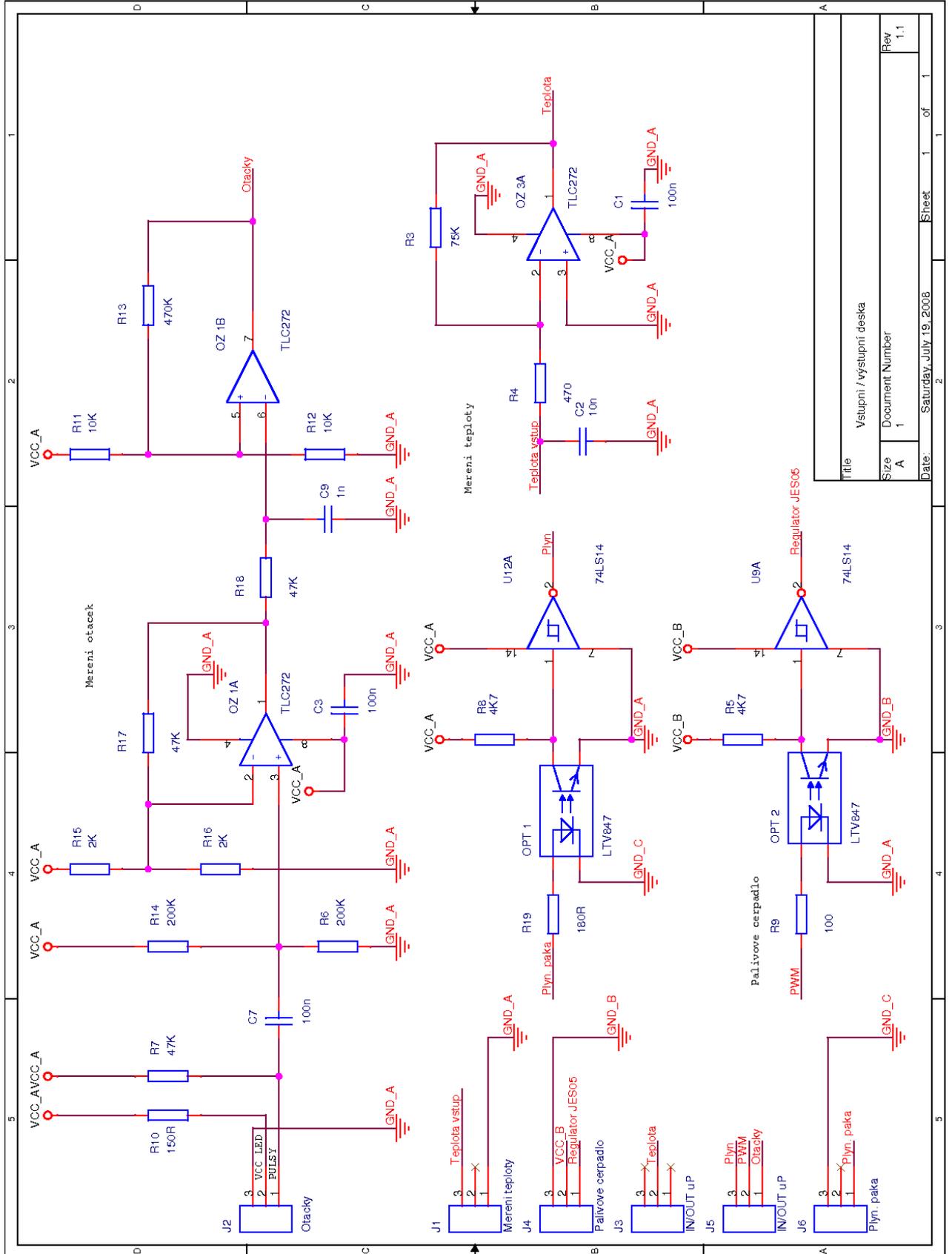
Na následujících stránkách jsou tyto schémata:

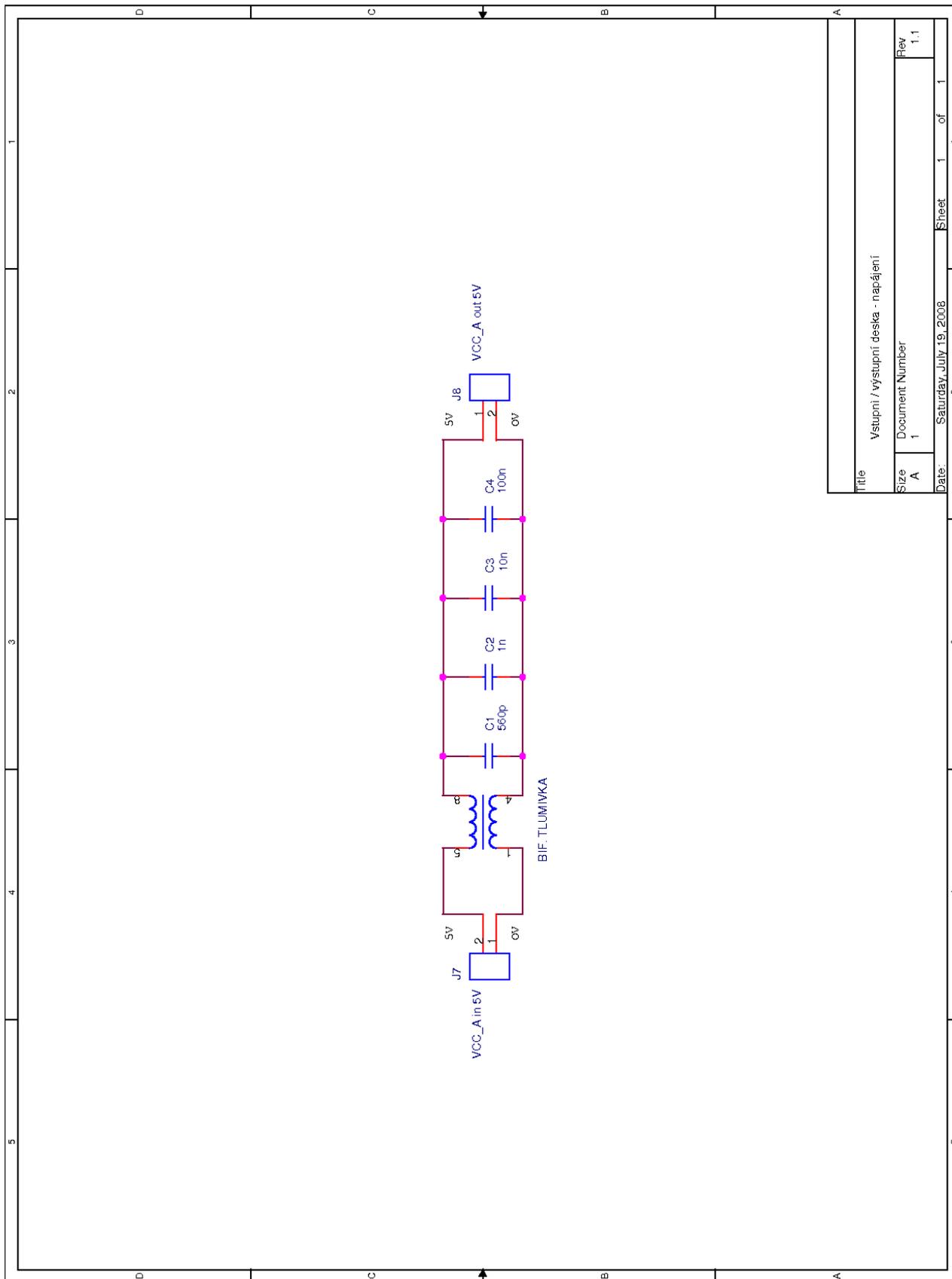
**Deska vstupů/výstupů**

**Napájecí obvod**

**SpejblARM**

**Turbína**

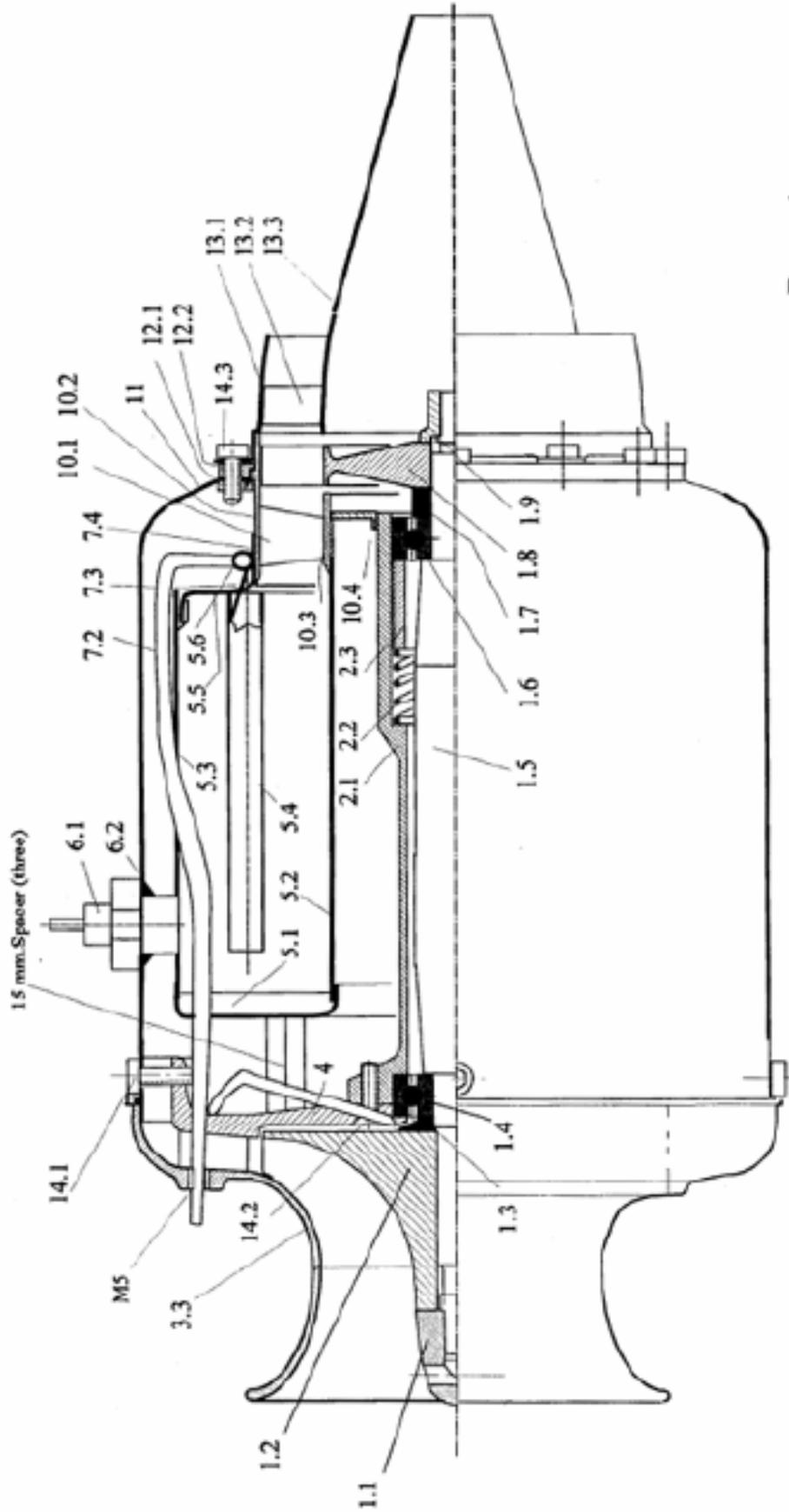




Title		Vstupní / výstupní deska - napájení	
Size	A	Document Number	1
Rev	1.1	Date:	Saturday, July 19, 2008
Sheet 1 of 1		2	



# Microturbine KJ-66



## B Komunikační protokol

Tento protokol byl navržen Miroslavem Hájkem jako komunikační protokol mezi softwarovým vybavením osobního počítače, PDA a řídicí jednotkou pro řízení proudového motoru. Komunikace probíhá po seriové lince RS232, která je popřípadě emulována na PC rozhraním USB. Data jsou posílána ve formátu *big-endian*. Zpráva je posílána jako posloupnost znaků, kde čísla jsou reprezentována řetězcem hexadecimálních čísel o různé délce.

### B.1 Protokol

**Obecný formát zpráv:** Zde je obecný popis, z jakých částí se zpráva skládá.

```
<zpráva> <XOR>
<start byte><tělo zprávy><XOR>
<start byte><délka zprávy><datablok><XOR>
<start byte><délka zprávy><id zprávy><data><XOR>
```

**Použité znaky:**

*start byte*

Velikost	Znak	Popis
1	§	Úvodní znak

*délka zprávy*

Velikost	Znak	Popis
2		Počet dat v databloku včetně hodnoty XOR

*id zprávy*

Velikost	Znak	Popis
1	V	Data
1	C	Příkaz viz tabulka příkazů
1	R	Regulátor
1	E	Error
1	O	Ok
1	J	Provede skok o definované výšce
1	U	Rampa nahoru

1	D	Rampa dolů
1	P	Nastaví vzorkovací periodu
1	S	Otáčky

*XOR*

Velikost	Znak	Popis
1		XOR

### Směr zpráv/příkazů

PC ⇒		⇐ Řídicí deska	
Zpráva	Odpověď	Odpověď	Zpráva
			<b>Z</b> Data
<b>Z</b> Regulátor		Ok / Error	
<b>Z</b> Otáčky		Ok / Error	
<b>Z</b> Skok		Ok / Error	
<b>Z</b> Rampa		Ok / Error	
<b>Z</b> Vz. perioda		Ok / Error	
<b>P</b> Regulátor		<b>Z</b> Regulátor	
<b>P</b> Program		Ok / Error	
<b>P</b> Manuál		Ok / Error	
<b>P</b> Stop		Ok / Error	
<b>Z</b> Vz. perioda		<b>Z</b> Vz. perioda	

**Z** - zpráva, **P** - příkaz

### Příkazy

<kód>

§	0	3	C	Id	XOR
---	---	---	---	----	-----

*Délka: 3*

Příkazy jsou zprávy, které obsahují pouze *id* zprávy (znak C), po kterém následuje jeden znak reprezentující daný příkaz. Příkaz tak neobsahuje blok data. V následující tabulce

je seznam všech použitých příkazů.

#### *Tabulka příkazů*

Příkaz	Kód	Popis
regulátor	R	Řídicí deska pošle aktuální nastavení regulátoru
program	A	Předá řízení programu
manual	B	Přepne řízení na manualní
stop	C	Zastaví turbínu
vz. peri- oda	S	Odešle do PC nastaveno vz. periodu

## Zprávy

### Data

<id dat><napětí na čerpadle><otáčky><teplota>

§	2	2	V	Id	P	S	T	XOR
---	---	---	---	----	---	---	---	-----

*Délka: 34*

*Formát*

Znak	Formát	Jednotky	Popis
Id	int32	-	
P	int32	[ mV ]	Napětí na čerpadle
S	int32	[ ot/min ]	Otáčky
T	int32	[ °C ]	Teplota

### Popis

Na prvním místě je úvodní znak, následuje znak, který identifikuje datablok jako data. Délka zprávy je 34 (22h). *Id* odpovídá číslu vzorku a je inkrementováno s každým následujícím vzorkem. Pokud poslaná hodnota XOR neodpovídá vypočtené, informuje se uživatel.

**Nastavení regulátoru**

&lt;parametry&gt;&lt;limity&gt;

§	C	2	R	k	$w_I$	$w_D$	b	c	N	$I_L$	$I_H$	$D_L$	$D_H$	$O_L$	$O_H$	XOR
---	---	---	---	---	-------	-------	---	---	---	-------	-------	-------	-------	-------	-------	-----

*Délka: 192**Formát*

Znak	Formát	Jednotky	Popis
k	double	-	
$w_I$	double	-	
$w_D$	double	-	
b	double	-	
c	double	-	
N	double	-	
$I_L$	double	-	Spodní mez integrátoru
$I_H$	double	-	Horní mez integrátoru
$D_L$	double	-	Spodní mez derivátoru
$D_H$	double	-	Horní mez derivátoru
$O_L$	double	-	Spodní mez výstupu
$O_H$	double	-	Horní mez výstupu

**Vzorkovací perioda**

&lt;násobitel&gt;

§	0	A	P	M	XOR
---	---	---	---	---	-----

*Délka: 10**Formát*

Znak	Formát	Jednotky	Popis
M	int32		Násobitel základní vz. periody procesoru

*Hodnoty*

Násobitel	Vzorkovací perioda [ ms ]
1	16,384
2	32,768
3	49,152
4	65,536
...	...

**Otáčky**

&lt;otáčky&gt;

§	0	A	S	X	XOR
---	---	---	---	---	-----

*Délka: 10**Formát*

Znak	Formát	Popis
X	int32	Otáčky. Například pro nastavení volnoběžných otáček. Turbína přejde plynule na danou hodnotu otáček (v daných mezích).

**Skok**

&lt;napětí&gt;

§	0	A	J	U	XOR
---	---	---	---	---	-----

*Délka: 10**Formát*

Znak	Formát	Popis
X	int32	Napětí v mV

**Rampa nahoru a dolu**

&lt;počet kroků&gt;&lt;Výška jednoho kroku&gt;&lt;čas jednoho kroku&gt;

§	1	A	U	C	H	T	XOR
---	---	---	---	---	---	---	-----

*Délka: 26*

*Formát*

Znak	Formát	Popis
C, H, T	int32	Napětí v mV

**Popis**

Po přijetí tohoto příkazu řídicí jednotka začne generovat rampu směrem nahoru (U), popřípadě dolů (D) definované výšky (CxH) a délky (CxT). Výška je napětí v *mV*, čas je v *ms*.

**Potvrzující a chybové zprávy**

Potvrzující zprávy se posílají:

- pouze z řídicí jednotky

Potvrzují se:

- příkazy
- nastavení regulátoru

Chybové zprávy se posílají:

- pokud nastala „nějaká“ chyba dle tabulky chyb

**Potvrzující zprávy**

<id potvrzované zprávy>

§	0	3	O	Id	XOR
---	---	---	---	----	-----

*Délka: 3*

*Formát*

Znak	Formát	Popis
id	char	Id zprávy, která má být potvrzena. Napr. pro regulátor je <i>Id</i> „R“

**Příklad**

Pokud je vyžadována potvrzující zpráva, měla by být do určitého časového limitu doručena zpět straně, která ji vyžádala. Pokud se tak nestane:

- informuje se uživatel
- opakuje se automaticky pokus (např. max 3x → informuje se uživatel)

### Chybové zprávy

<id>

§	0	A	E	Id	XOR
---	---	---	---	----	-----

*Délka: 10*

Chybové zprávy se řídí dle následující tabulky:

*Tabulka chyb*

Id	Formát	Popis
??	int32	Id chybové zprávy

### Zabezpečení zpráv - XOR

<zpráva><XOR>

Kontrolní příčná sudá parita se počítá jako exkluzivní OR. Vždy přes celou zprávu.

**Příklad**

§	2	2	V	Id	P	S	T	XOR
---	---	---	---	----	---	---	---	-----

$x$	$y$	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

### Formát dat

#### Int32

Velikost čísla typu *int32* je  $4B \Rightarrow 8$  znaků

#### Double

Velikost čísla typu *double* má délku  $8B \Rightarrow 16$  znaků

#### Char

Představuje jeden libovolný znak

**Příklad zprávy****Data**

§22V000000020000157C000151E400000282[XOR]

*Id zprávy:* V = data  
*Id (číslo) vzorku:* 2  
*Napětí na čerpadle:* 5500 mV  
*Otáčky:* 86500 ot/min  
*Teplota:* 642 °C

## C Obsah CD

K práci je přiložené CD s tímto obsahem:

**datasheety** (dir) Datasheety nejdůležitějších součástek použitých v zapojení.

**Schémata** (dir) Shémata navrženého zapojení.

**turbina** (dir) Zdrojové kódy programu.

**LaTeX** (dir) Zdrojové soubory bakalářské práce pro L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>.

**bp\_2008\_michal\_vosecky.pdf** (file) Elektronická verze bakalářské práce.