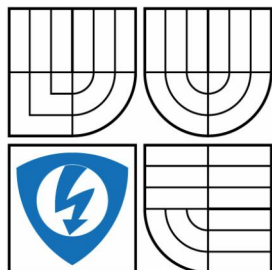


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

MODEL VĚTRNÉHO TUNELU S UNIVERZÁLNÍMI VSTUPY A VÝSTUPY

MODEL OF WIND TUNEL WITH UNIVERSAL INPUTS AND OUTPUTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

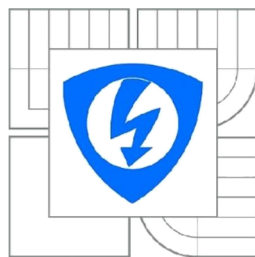
AUTOR PRÁCE
AUTHOR

Martin HOLČÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Radek ŠTOHL, Ph.D.

BRNO 2012



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Martin Holčík

ID: 134493

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Model větrného tunelu s univerzálními vstupy a výstupy

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvoření modelu větrného tunelu. Model bude řízen normalizovaným analogovým signálem, binární hodnotou nebo ručně. Výstup z modelu bude opět analogový signál nebo binární číslo. Do modelu budou vloženy základní bezpečnostní prvky.

1. Navrhněte model větrného tunelu pro řízení pomocí PLC
2. Nastudujte zapojení normalizovaných vstupů a výstupů, poruchového stavu
3. Aplikujte získané znalosti pro ovládání otáček vstupního ventilátoru, včetně detekce poruchy ventilátoru.
4. Navrhněte převodník otáček výstupního ventilátoru na analogovou normalizovanou hodnotu
5. Realizujte vyvedení digitálních reprezentací analogových hodnot.
6. Navrhněte desku plošných spojů.
7. Zhotovte výsledný model.
8. Ověřte funkčnost modelu na vzorové laboratorní úloze.

DOPORUČENÁ LITERATURA:

1. AMiNI4DS – vstupy/výstupy [online] http://amit.cz/docs/cz/amini/amini4ds_g_cz_100.pdf
2. 4-20 mA Digital to Process Current Transmitter Application Note [online] <http://application-notes.digchip.com/013/13-14780.pdf>

Termín zadání: 11.2.2013

Termín odevzdání: 27.5.2013

Vedoucí práce: Ing. Radek Štohl, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.

Předseda oborové rady

Abstrakt

Cílem práce je navrhnout a realizovat model větrného tunelu. Model je možné ovládat různými standardizovanými rozsahy signálů a to jak v analogové podobě, tak i v digitální. Model se skládá ze dvou ventilátorů – ventilátoru elektricky poháněného a ventilátoru poháněného proudem vzduchu. Výstupem z modelu jsou opět standardizované signály, nesoucí informaci o otáčkách druhého ventilátoru. Následující dokument pojednává o návrhu komponent, následné realizaci a technickým problémům, které nastaly, popisu konečného řešení a programové části řešení.

Klíčová slova

Analogový signál, digitální signál, napěťový signál, ATmega32, Hallova sonda.

Abstract

The aim of this work is to design and implement a model of wind tunnel. The model can be controlled in various ranges of standardized signals in both analog and digital form. The model consists two fans - an electrically driven fan and fan driven by air stream. The outputs from the model are standardized signals, carrying information about the speed of second fan. The following document discusses the design of components, and subsequent implementation of the technical problems that occurred, a description of the final solution and the software part of the solution.

Keywords

Analog signal, digital signal, voltage signal, ATmega32, Hall probe

Bibliografická citace:

HOLČÍK, M. Model větrného tunelu s univerzálními vstupy a výstupy. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 53s. Vedoucí bakalářské práce byl Ing. Radek Štohl, Ph.D.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma „Model větrného tunelu s univerzálními vstupy a výstupy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: **27. května 2013**

.....
podpis autora

Poděkování

Protože není v mých silách konkrétně vyjmenovat všechny, rád bych poděkoval všem těm, se kterými jsem se měl možnost v průběhu posledních tří let poznat a kteří se mne snažili pomoci, poradit a naučit něčemu novému.

Proto Vám všem nejupřímnější díky!

V Brně dne: **27. května 2013**

.....

podpis autora

Obsah

1	Úvod.....	9
1.1	Historie.....	9
1.2	Cíle.....	9
2	Větrný tunel	10
2.1	Model	10
2.2	Koncepce modelu.....	10
2.3	Blokové schéma	10
2.4	Ventilátor.....	11
2.4.1	Modulace	14
2.5	Ovládání modelu	15
2.5.1	Elektrické signály	16
2.6	Snímače otáček.....	19
2.7	Řídicí obvod	20
2.8	PLC	22
3	Návrh jednotlivých částí	24
3.1	Mikrokontrolér ATmega32	24
3.1.1	Čítače/časovače.....	25
3.1.2	AD převodník	40
3.2	DA převodník.....	43
3.2.1	Operační zesilovač.....	44
3.3	Číslicové vstupy/výstupy	45
3.4	Výkonová část.....	46
3.4.1	Tranzistorový spínač.....	46
3.5	Vývojový diagram.....	47
4	Program.....	48
5	Závěr	48

Seznam obrázků

Obr. 2.1 – Blokové schéma modelu větrného tunelu.....	11
Obr. 2.2 – BLDC motor – uspořádání	12
Obr. 2.3 – Halova sonda	13
Obr. 2.4 – Principiální schéma PWM	14
Obr. 2.5 – Průběhy signálů PWM.....	15
Obr. 2.6 – Rozdělení elektrických signálů.....	16
Obr. 2.7 – Typické uspořádání proudové smyčky [7]	17
Obr. 2.8 – Optický snímač otáček.....	19
Obr. 2.9 – Von Neumanova architektura - blokové schéma.....	20
Obr. 2.10 – Tok informací a vnitřní struktura PLC	22
Obr. 3.1 – Blokové schéma mikrokontroléru ATmega32	24
Obr. 3.2 – Blokové schéma čítače/časovače procesoru ATmega32	25
Obr. 3.3 – TCCR0 registr – rozložení bitů	29
Obr. 3.4 – Čítač/časovač 1	30
Obr. 3.5 – Rozložení bitů v registru TCCR1A	36
Obr. 3.6 – Rozložení bitů v registru TCCR1B	36
Obr. 3.7 – Blokové schéma AD převodníku ATmega32.....	40
Obr. 3.8 – Uspořádání AD převodníku ATmega32.....	42
Obr. 3.9 – AD převodník s postupnou aproximací, blokové schéma	42
Obr. 3.10 – DA převodník s odporovou sítí R-2R.....	43
Obr. 3.11 – Operační zesilovač.....	44
Obr. 3.12 – OZ v zapojení se zápornou zpětnou vazbou.....	44
Obr. 3.13 – Optron PC817 – uspořádání	45
Obr. 3.14 – Tranzistorový spínač.....	46
Obr. 3.15 – Vývojový diagram	47
Obr. 5.1 – Schéma zapojení	56

Seznam tabulek

Tab. 3.1 – Output compare – Normal/CTC Mode, nastavení.....	27
Tab. 3.2 – Compare output – Fast PWM mode, nastavení	27
Tab. 3.3 – Output compare – Phase Correct PWM mode, nastavení	28
Tab. 3.4 – Nastavení čítače/časovače 0, režimy	29
Tab. 3.5 – Output compare – Normal/CTC Mode, nastavení.....	32
Tab. 3.6 – Output compare – Fast PWM Mode, nastavení.....	34
Tab. 3.7 – Output compare –Frequency and Phase Correct PWM.....	35
Tab. 3.8 – Nastavení čítače/časovače 1, režimy	37
Tab. 3.9 – Nastavení dělicího faktoru předděličky	39

1 Úvod

1.1 Historie

20. století znamenalo ohromný rozvoj všech technických odvětví. Elektrotechnika byla jedním z těch, která zaznamenala největší růst nových technologií. Novým, velmi dynamicky se rozvíjejícím oborem se stala automatizace, kde se pro řízení procesů využívalo elektronických obvodů, nazývaných regulátory.

Pro kvalitní regulaci bylo zapotřebí co nejpřesněji nastavit regulátor. K nastavení regulátoru bylo zase potřeba identifikovat soustavu, kterou máme regulovat, řídit. To často není možné, nebo je to možné pouze zčásti, proto je v některých případech výhodné využít modelů soustavy. Takový model může potom velmi výrazně ovlivnit správné nastavení regulátoru a tím i celý regulační děj.

Tato práce se jedním takovým modelem zabývá. Ačkoliv není přímým modelem nějakého procesu, lze ho použít jako demonstračního modelu pro nastavování regulátoru otáček.

1.2 Cíle

Úkolem práce je sestavení modelu větrného tunelu, který bude řízen pomocí PLC nebo ručně ovládacími prvky přímo na modelu. Regulace otáček probíhá pomocí standardizovaných signálů a to jak v analogové, tak digitální podobě. Na modelu je osazeno STOP tlačítko, které zde představuje bezpečnostní prvek. Na výstupech jsou poskytovány informace o otáčkách ventilátoru a o stavu zapnuto-vypnuto.

Práce je rozdělena na několik částí, ve kterých je popsána struktura postupu návrhu a realizace tohoto modelu.

V první části je teoretický rozbor řešení jednotlivých částí a výběr toho nejvhodnějšího řešení.

Druhá část práce se věnuje návrhu hardwaru, jednotlivých částí obvodu, jejich nejvhodnějšímu obvodovému řešení a uspořádání.

Třetí, poslední část se zabývá popisem a rozбором programového řešení.

V přílohách lze najít schéma zapojení, návrh desky plošných spojů, osazovací výkres a rozpisku součástí.

2 Větrný tunel

2.1 Model

Model zpravidla slouží jako náhrada za nějakou fyzickou věc, princip nebo proces, který zjednodušuje a umožňuje získat hlubší informace o jeho vlastnostech a chování.

Vytváření objektů, nahrazujících fyzický objekt se nazývá modelování. Modelování je hojně využívané zejména ve stavebnictví, elektrotechnice, technickém inženýrství nebo ve strojírenských a konstrukčních aplikacích, kde je zapotřebí navrhnout parametry na dosud neznámý fyzický objekt. Postupně jsou navrženy parametry na abstraktní objekt. Ty jsou pak aplikovány na reálný objekt - to je výhodné z ekonomického, ekologického a mnohdy i bezpečnostního hlediska, kdy není možné si dovolit omyly v konstrukci či chybu v nastavení parametrů.

2.2 Koncepce modelu

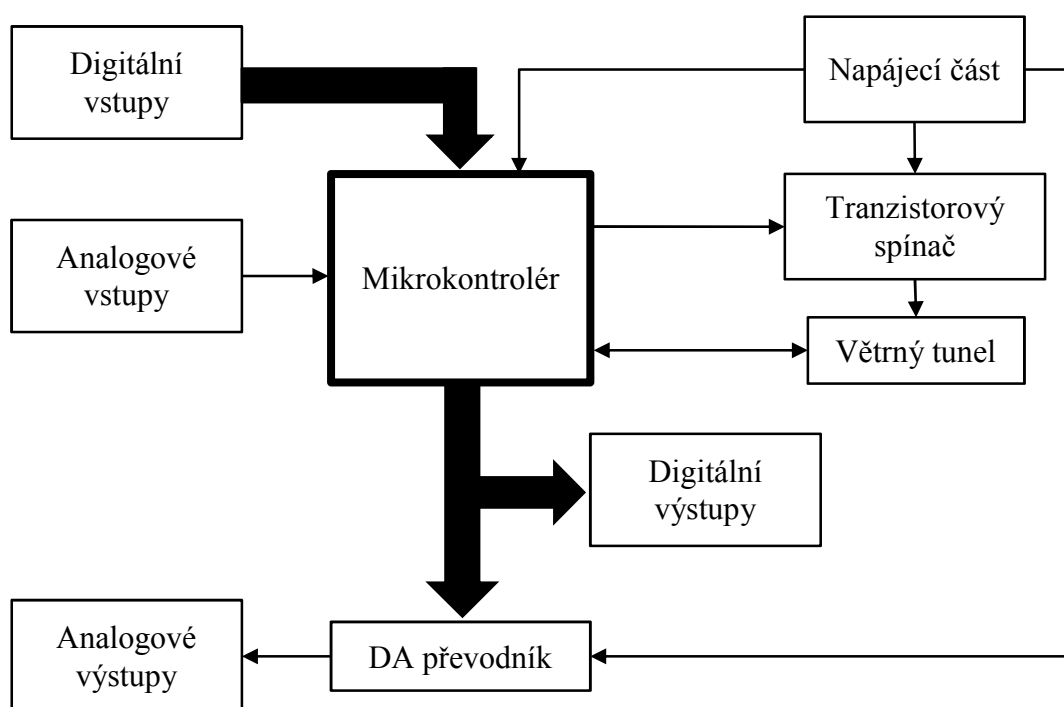
Model se skládá ze dvou ventilátorů. Jeden z nich je elektricky poháněný, který vytváří proud vzduchu pro druhý. Druhý ventilátor slouží ke snímání otáček ventilátoru prvního pomocí proudění vzduchu. V tomto smyslu je možné zde nalézt určitou analogii se zařízením pro měření rychlosti větru – anemometr.

Větrný tunel je koncipován jako model k PLC takže se pro jeho řízení předpokládá připojení k programovatelnému automatu, nicméně to není podmínkou, neboť přímo na modelu je osazen samostatný řídicí prvek. Tunel je možné ovládat číslicovými nebo analogovými signály. Na model je možné připojit standardizovaný analogový napěťový signál o úrovni $0 \div 10V$ a číslicový osmibitový signál o úrovni 24V.

Výstupním signálem z modelu je opět signál napěťový o odpovídajícím rozsahu vstupního (řídicího) signálu, opět i v číslicové podobě.

2.3 Blokové schéma

Na Obr. 2.1 je uvedeno zjednodušené blokové schéma celého obvodu. Skládá se z výkonové a řídicí části. Část řídicí se dále dělí na část analogovou a digitální. V další části textu budou jednotlivé bloky popsány.

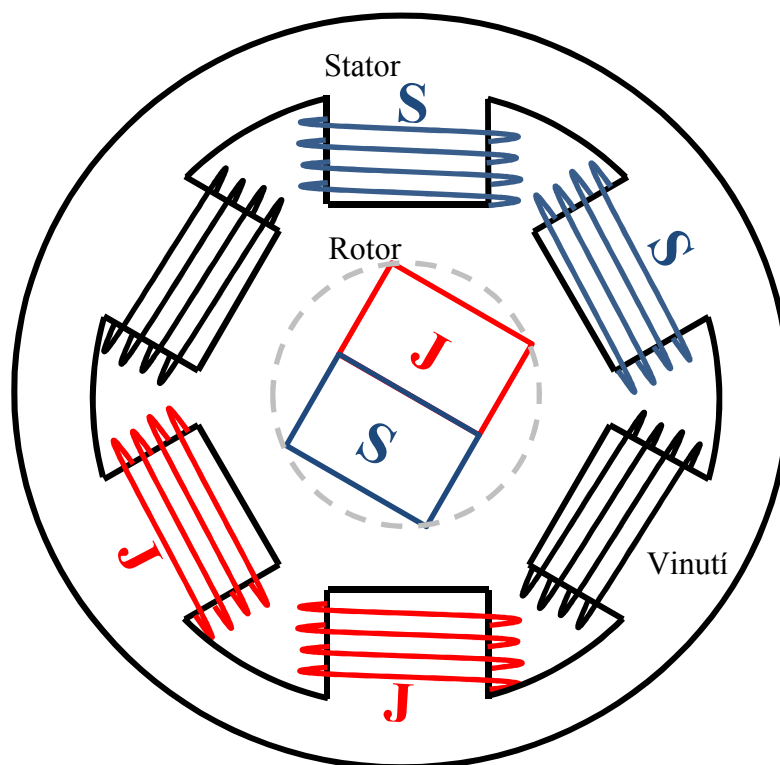


Obr. 2.1 – Blokové schéma modelu větrného tunelu

2.4 Ventilátor

Jako hlavní konstrukční prvek modelu je použit ventilátor, vytvářející proud vzduchu. Je proto potřeba rozhodnout o typu jeho pohonu. Protože je model napájený stejnosměrným napětím, bude řeč o stejnosměrném pohonu.

Pro účely modelu postačuje aplikace malých počítačových ventilátorů, které typově odpovídají stejnosměrným motorům s cizím buzením. Počítačové ventilátory jsou konkrétně motory s elektronickým komutátorem – BLDC motory (nebo také ECM motory) – bezkartáčové stejnosměrné motory [18].



Obr. 2.2 – BLDC motor – uspořádání

BLDC motory, jejichž principiální uspořádání je na Obr. 2.2 mají ve svém rotoru umístěn permanentní magnet. Stator je tvořený pólovými nástavci s napájeným vinutím. Vodič, kterým protéká proud, kolem sebe vytváří magnetické pole podle vztahu

$$\oint_C B \, dl = \mu_0 I_c \quad (1.1)$$

Rovnice (1.1) se nazývá Ampérův zákon celkového proudu a vyjadřuje závislost vzniku magnetického pole B na celkovém protékajícím proudu vodičem I_c [13].

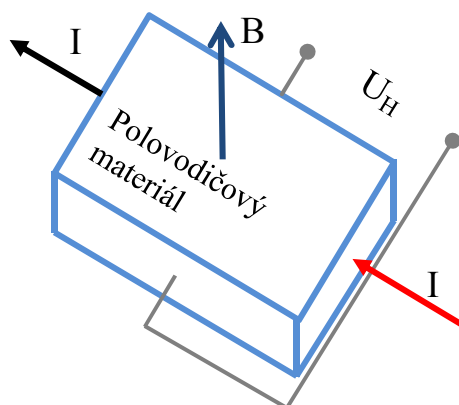
Jednotlivá vinutí pólových nástavců si lze představit jako cívky. Kolem nich, pokud jimi protéká proud, je vytvářeno magnetické pole. Sousedící vinutí mají vždy shodný směr toku magnetického pole, přičemž na protějších vinutích je vytvářen magnetický tok opačného směru. Tím vzniká moment sil působící na rotor tvořený permanentním magnetem.

Otáčení magnetu v rotoru je potom dosaženo přepínáním proudu jednotlivými vinutími. Magnetický tok vytvořený vinutím ve statoru za sebou táhne permanentní magnet v rotoru.

U stejnosměrných pohonů obecně je tedy zapotřebí zajistit, aby došlo k přepínání proudu vinutími. Nejjednodušší cestou je mechanický komutátor, který přivede proud vždy do toho vinutí, které má optimální směr vzniku krouticího momentu. Jedná se o přerušovaný kruhový kontakt, který je připojen přímo na jednotlivá vinutí v kotvě motoru. Takové motory mají vinutí v rotoru a ve statoru permanentní magnety. Jejich

velikou nevýhodou je jiskření při přepínání a problém s přenosem proudu přes sběrací kontakty do rotující kotvy.

Tyto problémy řeší právě bezkartáčový stejnosměrný motor s vinutím ve statoru [18]. Bez mechanického komutátoru se ztrácí informace o poloze rotou, která je zapotřebí pro optimální přepnutí proudu do dalšího vinutí. Poloha rotoru je proto snímána pomocí Halovy sondy, která má uspořádání podle Obr. 2.3. Pokud je polovodič, protékaný proudem vložen do magnetického pole, působícího kolmo na něj, potom na jeho protilehlých stranách vznikne rozdílný potenciál. Tento rozdíl potenciálů se nazývá Halovo napětí a je způsoben zakřivením domén v polovodiči k jedné z jeho stran. Díky signálu z Halovy sondy je k dispozici informace o poloze magnetu v rotoru, s jejíž pomocí lze ovládat přepínání proudu jednotlivými vinutími.



Obr. 2.3 – Halova sonda

Otáčky

Otáčky stejnosměrných motorů lze řídit několika způsoby [15]. Rovnice stejnosměrného motoru s cizím buzením je ve tvaru

$$\omega = \frac{U}{C\Phi} - \frac{RI}{C\Phi} \quad (1.2)$$

Z rovnice (1.2) tedy plyne možnost řídit otáčky změnou napětí U , odporu R nebo buzení Φ . Řízení otáček odporem je energeticky nevhodné, proto se této variantě vyhneme. Pokud bude motor řízen napětím, zachová se tím tvrdost jeho charakteristiky. Otáčky motoru lze řídit i změnou buzení. Toho ale nelze moc využívat z důvodů nasycení – proto se využívá odbuzování.

Nejvýhodnější a dnes nejpoužívanější je řízení pomocí pulzní šířkové modulace. Její velká výhoda je v bezeztrátovém řízení výkonu motoru a díky tomu dosahuje vysoké účinnosti. Mimo jiné pracuje motor s konstantním momentem, což je také výhodné. Proto bude dále popsáno řízení otáček pomocí PWM modulace.

2.4.1 Modulace

Modulace obecně je proces, pomocí něhož se vysokofrekvenční nosný signál, mění (moduluje) pomocí nízkofrekvenčního signálu, obvykle nesoucí nějakou informaci [3]. Modulací existuje spousta typů, nejtypičtějšími případy jsou frekvenční modulace (používána v radiotechnice), amplitudová modulace a pulzní šířková modulace.

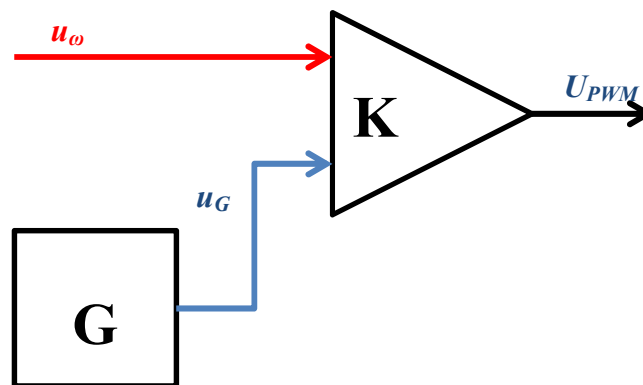
Zařízení, produkující takový signál se nazývá modulátor. Jeho výstupem je právě modulovaný signál. Každý takový modulátor musí obsahovat nelineární prvek, pomocí kterého právě k modulaci dochází – například dioda apod.

Z modulovaného signálu je většinou potřeba na koncové straně získat zpět informaci, jinými slovy signál demodulovat. Typickým příkladem je televizor, který přijímá kvadrurně modulovaný signál. Zařízení principiálně není nic jiného než demodulátor, který z modulovaného signálu zpětně získá signál nesoucí informaci pro vykreslování barevných pixelů na obrazovce. Demodulátory tedy převádějí (rekonstruují) signál do původní podoby, kterou měl před modulací.

2.4.1.1 Pulzní šířková modulace

Pulzní šířková modulace, (anglicky pulse wave modulation, zkr. PWM) je druh modulace, kdy je nízkofrekvenčním signálem měněna šířka pulzu vysokofrekvenčního signálu.

Klasicky se jedná o uspořádání dvou prvků, jak je uvedeno na Obr. 2.4. Sestava se skládá z generátoru G trojúhelníkového nebo pilovitého napětí u_G a komparátoru K [3].

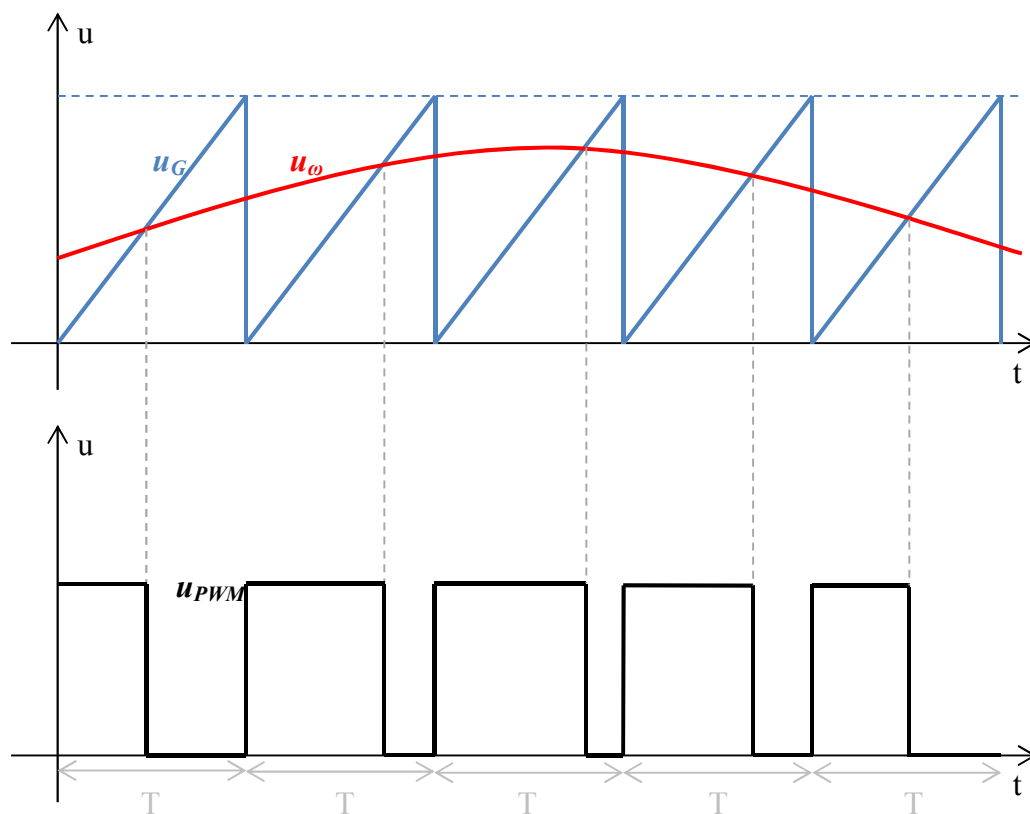


Obr. 2.4 – Principiální schéma PWM

Do tohoto komparátoru vstupuje signál z generátoru u_G , který je porovnáván s dalším analogovým signálem u_ω . V momentě jejich shody je překlopen výstup komparátoru.

Jak lze pozorovat na Obr. 2.5, výstupem komparátoru K je signál s proměnnou šířkou impulsu a konstantní periodou T .

Právě proměnná šířka impulsu je tedy nositelem informace – od toho taky vznikl název modulace (proměnná šířka pulzu).



Obr. 2.5 – Průběhy signálů PWM

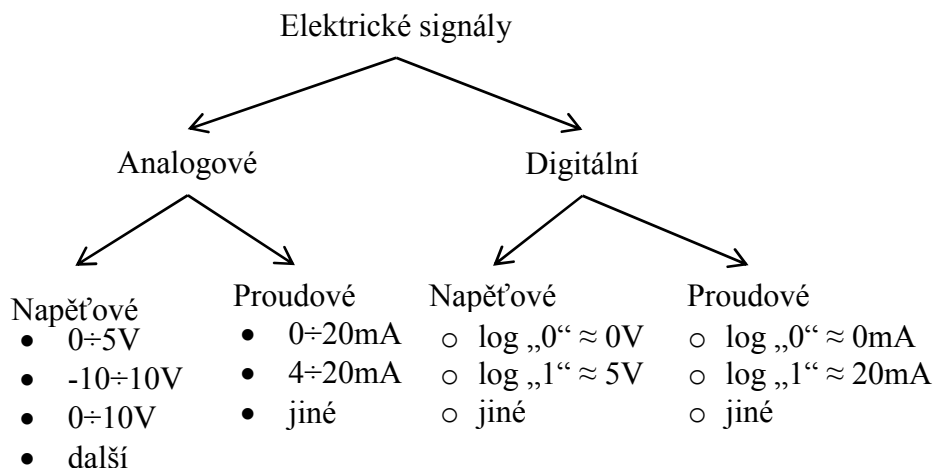
2.5 Ovládání modelu

Předpoklad řízení modelu pomocí PLC s sebou nese požadavky na možnost ovládat model různými typy signálů – číslicových, analogových a to v různých rozsazích. Proto je nutná unifikace použitých signálů.

Všechny procesory a elektronické obvody pracují s napěťovými signály (o různých úrovních, polaritách a tvarech). Proto je potřeba signály převést na napěťové, standardizovat různé rozsahy a teprve tyto signály přivádět do mikrokontroléru a tím řídit model.

2.5.1 Elektrické signály

Pod pojem elektrické signály spadá více typů signálů. Obecně jde ale o signály tvořené fyzikální veličinou typu napětí nebo proud [8]. Lze je rozdělit následovně



Obr. 2.6 – Rozdělení elektrických signálů

Elektrický signál je nejčastěji výstupem nějakého senzoru, čidla, řídicího členu nebo jiného systému apod. Každý typ je specifický použitím a má své výhody a nevýhody, existují ale určité standardy, které říkají, jaký typ signálu o jakém rozsahu se používá u různých zařízení.

Standardizované signály

Standardizace signálů v automatizaci dává konstruktérovi ohromnou výhodu při sestavování řídicího systému. Díky tomu lze pořídit akční člen a jednoduše ho připojit k regulátoru. Akční člen, pokud jsme ho vybrali správně podle typu signálu a jeho rozsahu, bude reagovat bez potřeby úpravy signálu. Typickou ukázkou standardizace je model ISO/OSI, kde na nejnižší fyzické vrstvě je popsáno, jak bude komunikace po fyzické stránce (nejen) signálu vypadat [2]. Je zde popsáno, jaké úrovně bude signál, jakou impedanci bude mít vodič, mechanické požadavky na konektory, jaké bude rádiové rozhraní včetně elektromagnetického spektra a intenzity signálu a dalších vlastností. Podobně tak jsou specifikovány například signály přenosového rozhraní RS-232 – datové signály: $\log „0“ = 3 \div 15V$ a $\log „1“ = -3 \div -15V$.

Podobně se ustálilo používání napětíových signálů o úrovních $0 \div 10V$ a proudových signálů $0 \div 20mA$ ($4 \div 20mA$) které se nazývají standardizované signály. Ty jsou používány ve většině průmyslových senzorů akčních členů a řídicích členů.

Napětíové signály

Napětíové signály jsou výhodné z hlediska zpracování, protože nepotřebují žádný další převod, lze je přímo připojit na určitý řídicí člen, po úpravě úrovně i na mikrokontrolér.

Jejich slabinou je ale rušení. Všude kolem nás se vyskytuje elektromagnetické pole, vytvořené různými přístroji nebo napájecími přívody, ze kterého se do vodiče indukují rušivá napětí. Ty potom zkreslují signály, někdy až do té míry, že se stávají nepoužitelnými.

Rušení je tím horší, čím delší je délka spojovacího vedení, souběh s jinými napájecími vodiči (většinou zakázáno, povoleno bývá pouze křížení) a další faktory [7]. Dalším problémem je nenulová impedance vodičů, která způsobuje úbytek napětí. Ten může být s velikou vzdáleností (stovky metrů) tak velký, že se signál prakticky „ztratí“ na vedení. Větší rušení také znamená zpomalení přenosu signálů, což může způsobovat další problémy.

Proudové signály

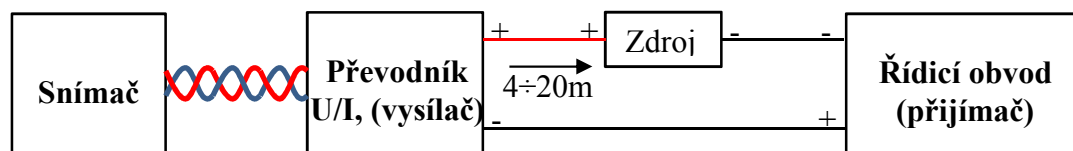
Z těchto důvodů je výhodnější použití proudových signálů, které jsou na rušení odolnější. Pokud je napájecí zdroj dostatečně dimenzován, odpadají i starosti s omezenou délkou vodiče – úbytek je malý, proto se prakticky neprojeví. Mimo jiné, lze senzory napájet přímo z proudové smyčky, čímž se ušetří náklady za napájecí pár vodičů. Proudová smyčka je nejčastěji složena z bloků naznačených na Obr. 2.7.

Senzorem snímaná neelektrická veličina je převáděna na odpovídající napěťový signál. Jeho výstup je připojen k vysílači, ve kterém je proveden převod z napěťového signálu na odpovídající proudový – 0% odpovídá proud 4mA a 100% proud 20mA. Výstup z vysílače je součástí proudové smyčky. Ve smyčce k řídicímu členu zpracovávajícímu proudový signál je ještě zařazen zdroj, který napájí komponenty připojené do smyčky. Před řídicím členem se provede pouze převod proudového signálu na napěťový a ten je dále zpracováván mikrokontrolérem nebo jinými komponentami.

Nejjednodušším převodníkem proudu na napětí je rezistor, na kterém podle Ohmova zákona ve tvaru

$$U = RI \quad (1.3)$$

vzniká úbytek napětí U odpovídající proudu I , procházejícímu přes rezistor R . Konstanta úměrnosti napětí U a proudu I je tedy elektrický odpor [13]. Hodnota rezistoru R (velikost elektrického odporu) se určí podle toho, jaký napěťový rozsah má odpovídat rozsahu proudovému.



Obr. 2.7 – Typické uspořádání proudové smyčky [7]

Typická proudová smyčka se tedy skládá ze

- Snímače – ten poskytuje nějaký signál, informaci k převodu na proudový signál
- Vysílače – převádí napěťový signál na proudový
- Zdroje – poskytuje napětí pro funkci přijímače a vysílače

Přijímače – přijímá proudové signály, převádí je zpět na napěťové a dále s nimi pracuje

Komponenty proudové smyčky mohou být dvojího typu

Aktivní (angl. SOURCE)

Pasivní (angl. SINK)

Jde o rozdělení z hlediska vlivu na smyčku. Komponenta typu SOURCE (aktivní) reaguje na změnu impedance ve smyčce (je zdrojem, dodává do smyčky elektrickou energii), zatímco SINK součást mění vlastní impedanci (vzniká na ní úbytek napětí, je ve smyčce spotřebičem).

V katalogových listech výrobci uvádějí maximální impedanci smyčky, pro kterou bude ještě schopen vysílač zajistit maximální rozkmit signálu 20mA. Je to z toho důvodu, aby bylo vysílači zajištěné dostatečné napájení – při signálu odpovídajícím 100% (20mA) musí být součet všech úbytků ve smyčce přinejhorším roven napětí zdroje. Je proto potřeba vzít v úvahu odpor vodiče, potřebné napájecí napětí pro vysílač a napěťový úbytek na přijímači (pokud jde o variantu přímého napájení z proudové smyčky). Lepší je zdroj lehce předdimenzovat s ohledem na maximální výkonový úbytek na komponentách proudové smyčky.

Proč je tedy proudová smyčka o tolik výhodnější? Je to z důvodu odlišného principu přenosu signálu. Vedení proudu je totiž tvořeno pohybem elektronů, a protože jde o smyčku, elektrony se pohybují v uzavřeném obvodu. Díky tomu se musí dostat elektrony z kladné části smyčky (kladný pól) do záporné části (záporný pól). Odolnost proti rušení je dána nízkým odporem vstupů zařízení, připojených do smyčky (řádově desítky Ω). Pokud je smyčka napájena z dostatečně tvrdého zdroje napětí, pak lze zanedbat i nenulový odpor vodičů a ztráty na nich nemají vliv na přenášenou hodnotu [7].

Další výhodou je, že nulové úrovni 0% odpovídá proud 4mA. Tento rozsah ($4 \div 20\text{mA}$) dává konstruktérovi možnost detekovat poruchu vysílače nebo vedení. Pokud bude vodič porušený nebo vysílač nebude fungovat, jednoduše nebude smyčka uzavřená, takže výsledkem bude nulový signál, zatímco při použití rozsahu $0 \div 20\text{mA}$ ztrácíme tuto možnost, neboť nulové úrovni odpovídá 0mA, tudíž nevíme, zda je na vodiči skutečná hladina, odpovídající 0% veličiny ze senzoru nebo nastala porucha vysílače/přenosové cesty signálu.

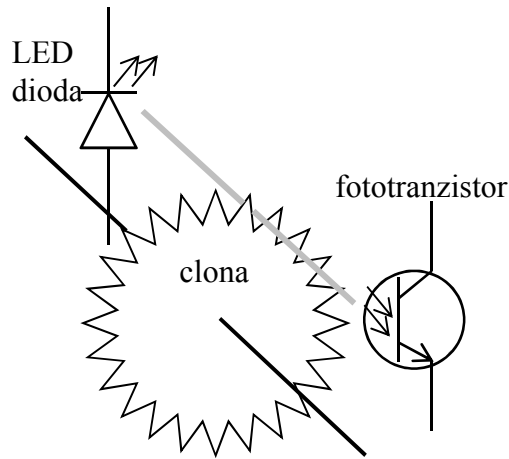
Analogové a digitální signály

Proudová smyčka může mít podobu jak analogovou, tak i digitální (podobně jako napěťová). Analogové signály vyjadřují nějakou funkční závislost, jsou v čase spojité.

Digitální signály jsou zase výhodnější z hlediska zpracování mikrokontrolérem nebo počítačem kde určují přímo číselnou hodnotu. Proto je výhodnější používat číslicovou podobu informace.

2.6 Snímače otáček

Hlavním výstupem z modelu je signál s informací o otáčkách ventilátoru, proto je potřeba jeho otáčky nějakým způsobem snímat. Pro snímání otáček je možné použít různé principy snímačů. Nejjednodušším typem je inkrementální čidlo.



Obr. 2.8 – Optický snímač otáček

Tento typ snímače využívá optoelektronického prvku (nejčastěji fototranzistoru nebo fotodiody) [4]. Typické uspořádání takového snímače je na Obr. 2.8.

LED diodou vytvářený světelný paprsek prochází přes clonku a dopadá na fototranzistor. Pokud je jeho báze dotována dopadajícími fotony, otevírá se. Pokud se clonka trochu pootočí, paprsek už přes clonku neprochází a fototranzistor se zavírá. Tímto způsobem dostaneme jednoduché inkrementální čidlo, které ovšem nezná směr otáčení. Je tedy schopné pouze reagovat na pohyb clonky – to nám ale pro snímání počtu otáček stačí

Další variantou je použití Halovy sondy z Obr. 2.3, která se už v použitém BLDC motoru vyskytuje – z důvodu přepínání proudu vinutími. Je to tedy konstrukční prvek BLDC motoru. Nejvýhodnější a nejjednodušší je využít její výstup, který po úpravě v programu poslouží jako signál pro programové počítadlo otáček.

2.7 Řídicí obvod

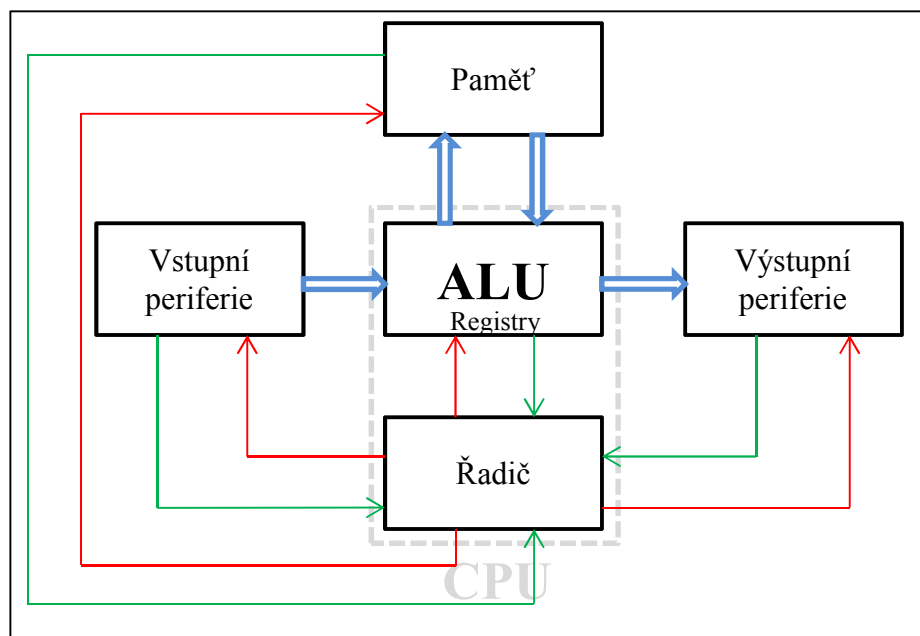
K řízení otáček ventilátoru je použito pulzní šířkové modulace a různých signálů včetně osmibitového čísla. To s sebou nese požadavek na obvod, který umí zpracovat číslíkovou hodnotu signálu nebo použít DA převodník. Výstupy z obvodu jsou opět osmibitové číslo nebo analogová hodnota, proto bude nutné i použití DA převodníku v modelu.

Jednou z možností by byla aplikace některého IO pro řízení PWM, například obvodu TL494CN, určeného pro řízení spínaných zdrojů. Pro číslicové zpracovávání informací se ale více hodí mikrokontroléry, u kterých odpadá potřeba převádět číslicový signál na analogový a ten zpracovat na PWM signál. Některé z nich mají v sobě zabudovány i další podpůrné obvody (AD převodník, komparátor aj.) a jejich použití je tedy značným zjednodušením celkového elektrického obvodu.

Mikroprocesory jako hlavní součást mikrokontrolérů, můžeme z hlediska architektury rozdělit na mikroprocesory s Von Neumannovou architekturou a Harvardskou architekturou [10].

Koncepce mikropočítače podle Von Neumanna Obr. 2.9 se skládá z

- Aritmeticko-logické jednotky (ALU)
- Řadiče
- Operační paměti
- Vstupních periférií
- Výstupních periférií



Obr. 2.9 – Von Neumanova architektura - blokové schéma

ALU je aritmeticko-logická výpočetní jednotka, zpracovává všechny aritmetické a logické operace (v ní se nacházejí sčítačky, násobičky, děličky, komparátory apod.).

Řadič slouží k řízení činnosti bloků (nachází se v něm registr instrukcí, dekodér instrukcí a čítač instrukcí - program counter). Paměť slouží k uchovávání instrukcí programu a dat. Vstupní a výstupní periferie jsou připojitelná zařízení, která umožňují interakci počítače s okolím, typicky se jedná například o klávesnici, myš, monitor a podobně. ALU společně s řadičem a registry tvoří procesor (zkr. CPU). Vytvořením všech tří bloků v jednom pouzdře vznikl mikroprocesor.

Základní cyklus takového počítače tvoří tyto činnosti

- Načtení instrukcí
- Dekódování instrukcí
- Načtení operandu
- Vykonávání instrukcí
- Uložení výsledných dat na výstup
- Test na přerušení

Většina dnešních procesorů využívá právě Von Neumannovy architektury. Harvardská architektura se liší od Von Neumannovy koncepce počítače oddělením paměti programu od paměti dat. Velikou výhodou této koncepce je fakt, že program nemůže přepsat sám sebe. Nevýhoda je ovšem v pevném (hardwarovém) rozdělení paměti. Programátor tedy nemůže plně administrovat a využít paměť.

Procesory lze dále rozdělit podle zpracovávaných instrukčních sad na RISC a CISC procesory. Procesory s RISC jádrem zpracovávají instrukce v redukované instrukční sadě. V praxi to znamená, že procesor zpracuje více fundamentálních instrukcí, které by jinak nahradila jedna složitější instrukce, která by ale zabrala více strojových cyklů. Tuto architekturu využívají procesory Atmel AVR, AMD, ARM a podobné.

CISC procesory zase používají komplexní sadu instrukcí, zpracovávající složitější operace, například násobení je v RISC procesorech implementováno jako sčítání s bitovým posunem, zatímco u CISC procesorů nikoliv.

Z preferenčního hlediska byla zvolena jako řídicí jednotka mikrokontrolér firmy Atmel AVR, typ ATmega32.

Mikrokontrolér ATmega32 disponuje dvěma osmibitovými čítači/časovači a jedním šestnáctibitovým čítačem/časovačem, čtyřmi kanály řízenými PWM, osmikanálovým AD převodníkem s maximálně desetibitovým rozlišením, 16kB FLASH paměti programu, 512B EEPROM paměti dat, taktováním externím krystalem a napájením 3,3-5V.

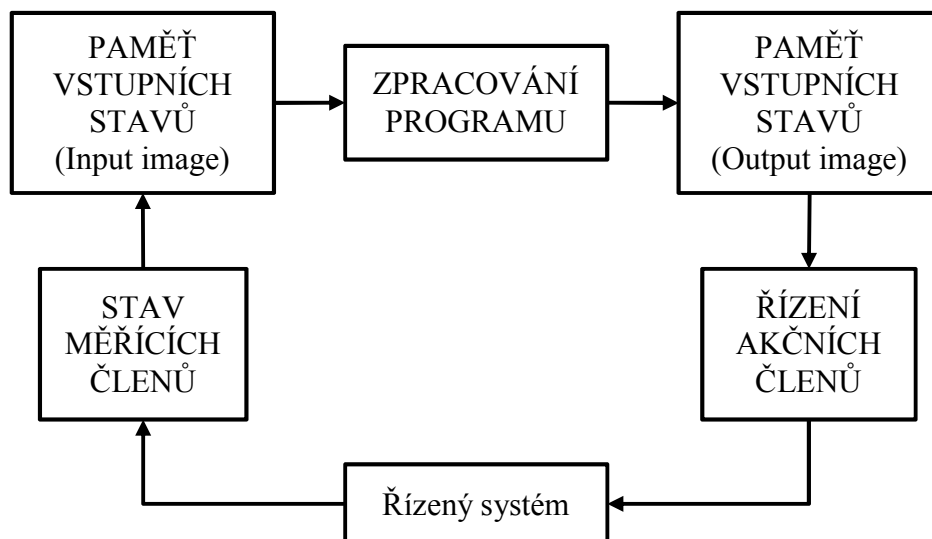
2.8 PLC

Programovatelné logické automaty (anglická zkr. PLC) jsou používány k řízení procesů a strojů [2].

Podle ČSN EN 61131 – 1 je PLC číslicově pracující elektronický systém konstruovaný pro použití v průmyslovém prostředí, využívající programovatelnou paměť pro interní ukládání uživatelsky orientovaných instrukcí pro provádění specifických funkcí (logických, sekvenčních, časových, čítacích, komunikačních, organizačních) za účelem řízení strojů či procesů, a to prostřednictvím digitálních nebo analogových vstupů a výstupů. Jak programovatelná řídicí jednotka, tak i periferní zařízení jsou konstruovány pro snadné začlenění do systému průmyslového řízení.

Zjednodušeně řečeno, jedná se o kombinaci procesoru se vstupy a výstupy (někdy i dalšími specializovanými obvody), který vykonává program ve své paměti v cyklech.

Základní cyklus PLC a tok informací v něm můžeme vidět na Obr. 2.10.



Obr. 2.10 – Tok informací a vnitřní struktura PLC

Po připojení napájení proběhne tzv. INPUT SCAN. Během něj se načtou do tabulky vstupů hodnoty vstupů (nejčastěji čidel). Dalším krokem je vykonávání programu PROGRAM EXECUTION. Probíhá sériově, cyklicky, a pokud jsou během vykonávání programu potřeba hodnoty vstupů, přistupuje se do tabulky vstupů. Po vykonání programu následuje OUTPUT SCAN, kdy jsou hodnoty, nahané po vykonání programu do tabulky výstupů, zapsány na příslušné výstupy.

Z konstrukčního hlediska dělíme PLC na dvě základní skupiny - na Modulární automaty a Kompaktní automaty.

Modulární logické automaty mají výhodu ve snadné rozšiřitelnosti, ekonomičnosti a variabilitě. Zpravidla se skládají z nějaké lišty (rack, základna), zdroje a procesoru. Další součásti už tvoří vstupně/výstupní karty, sběrníkové karty, čítače, speciální moduly a jiné zásuvné karty. Díky této konstrukci si může programátor přímo vybrat moduly podle toho, jaké komponenty bude připojovat do sestavy a tím minimalizovat

náklady. Nevýhodou tohoto řešení je větší cenová relace což ale vyvažuje možnost sestavy v budoucnu rozšiřovat, modifikovat.

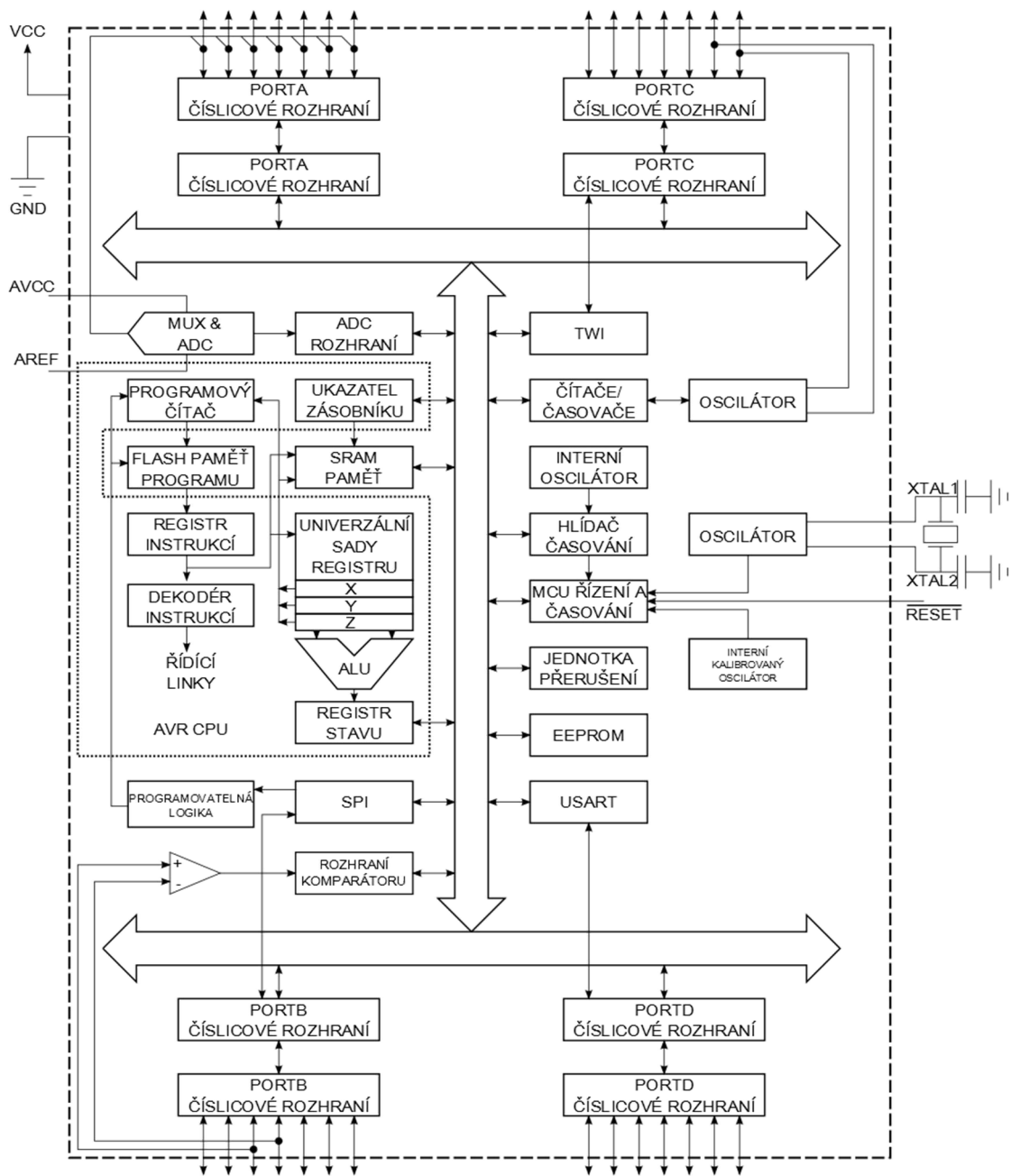
Kompaktní logické automaty jsou zpravidla levnější, protože mají jen omezený počet vstupů/výstupů a nedá se k nim dokoupit podobné příslušenství. Do určité míry je lze spojovat s dalšími moduly, avšak variabilita a modifikace není na takové úrovni jako u modulárních verzí.

Existují i speciální bezpečnostní verze programovatelných logických kontrolérů, které mají certifikaci pro použití ve zdravotnictví, extrémních podmínkách jako je výbušné prostředí, prostředí se zvýšeným výskytem polétavého prachu a podobně. Tyto verze jsou ještě dovybaveny různými bezpečnostními mechanismy, které zajišťují, že zařízení neselže.

3 Návrh jednotlivých částí

3.1 Mikrokontrolér ATmega32

Hlavním blokem celého modelu je mikrokontrolér ATmega32, který má za úkol generování signálu pro tranzistorový výkonový spínač, snímání otáček ventilátoru, A/D převod a signalizaci. Jeho blokové schéma je na Obr. 3.1 [5].



Obr. 3.1 – Blokové schéma mikrokontroléru ATmega32

Z hlediska ovládání je nejdůležitějším blokem čítač/časovač, mikrokontrolér ale nabízí i spoustu dalších periférií jako A/D převodník, SPI, USART, TWI aj. V další kapitole budou popsány čítače/časovače.

3.1.1 Čítače/časovače

Jak již bylo řečeno, pro řízení otáček ventilátoru se využívá PWM modulace. Princip tvorby takového pulzně šířkově modulovaného signálu je na Obr. 2.4.

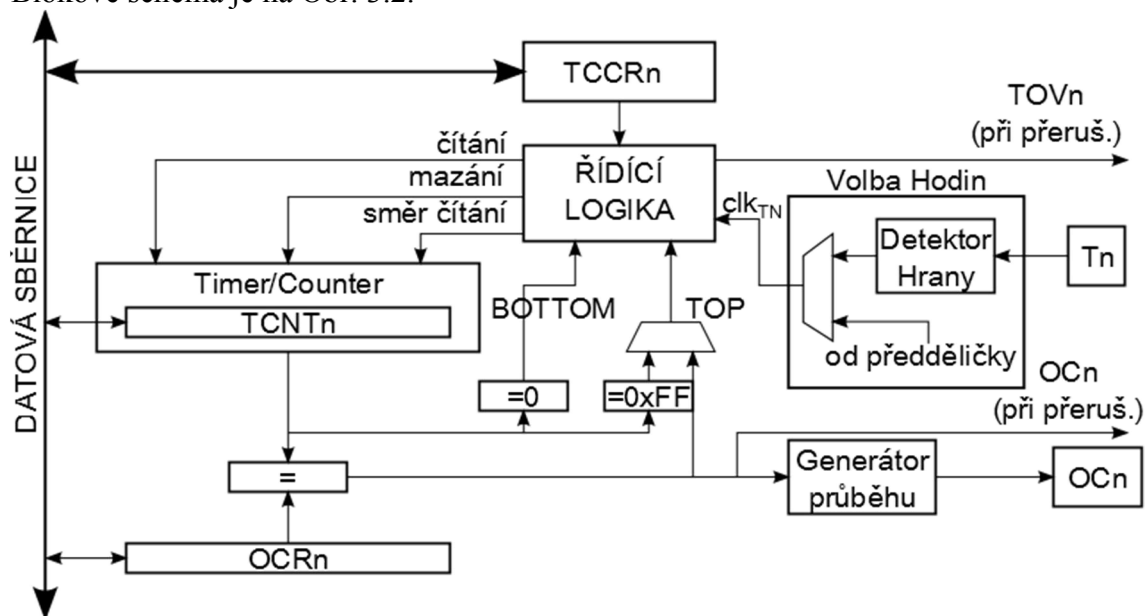
Pro vytvoření PWM signálu v číslicové podobě je zapotřebí časovač, podobně pro měření frekvence je použitý další časovač. Proto se jím budeme věnovat podrobněji.

Čítač/časovač 0

Protože model je ovládán osmibitovým číslem, vystačíme si s osmibitovou modulací – k tomu nám postačí čítač/časovač 0. Ten má následující vlastnosti:

- Je osmibitový, umožňuje tedy osmibitovou PWM
- Obsahuje nezávislou jednotku Output compare
- Dvojitě bufferované výstupní registry
- Jedna Input Capture jednotka + filtr
- Fázově korigovaný mód PWM
- Počítadlo externích událostí
- Přerušování od přetečení/jednotky compare match a další...

Blokové schéma je na Obr. 3.2.



Obr. 3.2 – Blokové schéma čítače/časovače procesoru ATmega32

Čítač/časovač 0 má definovány meze – MAX = 255 a BOTTOM = 0. Další mezí je TOP, určující vrchní mez čítání. Tu je možné měnit zápisem hodnoty do registru OCR0. Po přivedení hodinového signálu čítač čítá od hodnoty BOTTOM po hodnotu MAX (nebo TOP pokud je nastaveno). Následuje přetečení a nastavení příznaku TOV0.

Pokud je povoleno přerušení od čítače/časovače 0 (bit TOIE0 v registru TIMSK – Timer Interrupt Mask Register) následuje skok na obslužnou rutinu přerušení.

Hodinový signál může být buďto vnitřní, přiveden přes předděličku anebo externě přivedený signál na pin T0 (určují bity CS02:0 v registru TCCR0 – Timer Counter Control Register 0). Pokud není přiveden hodinový signál, je čítač deaktivován. Dělicí poměry hodinového signálu a jim odpovídající nastavení bitů je uvedeno v Tab. 3.9. Řídící logika určuje, zda bude čítač čítat směrem nahoru nebo směrem dolů (určuje režim, který se nastavuje bity WGM00 a WGM01 v registru TCCR0) a na které hrany signálu bude čítač reagovat. To úzce souvisí se signálem, který čítač generuje na pinu OC0.

Jednotka output compare je použita ke generování pulzně šířkové modulovaného signálu. Osmibitový komparátor cyklicky porovnává TCNT0 s registrem OCR0 (Output Compare Register 0). Při shodě obou registrů (TCNT0 a OCR0) dojde k nastavení příznaku OCF0 a pokud je povoleno přerušení od jednotky output compare (OCIE0 v registru TIMSK) dojde ke skoku do obslužné rutiny přerušení. Příznak shody obou těchto výše popsaných registrů využívá pro svou činnost generátor signálu

Normal mode

Jedná se o základní režim čítače/časovače 0. V tomto režimu čítač čítá směrem vzhůru (inkrementuje svůj obsah) s frekvencí hodinového signálu. Po dosažení hodnoty TOP přeteče a dojde k restartu. TOV0 příznak je nastavený, když čítač přeteče a je restartován (tj. ve chvíli, kdy je v něm hodnota 0x00). Někdy se využívá TOV0 jako 9. bit s tím rozdílem, že je pouze nastavitelný – o vynulování se musí postarat programátor. Pokud je povoleno i přerušení od přetečení čítače, je příznak TOV0 automaticky i vynulován.

CTC mode (Clear Timer on Compare Match)

V režimu CTC (WGM01:0=2) je registr OCR0 používán ke změně rozsahu čítače. Pokud nastane shoda registru OCR0 a TCNT0, potom je obsah registru TCNT0 vymazán (0x00) a čítač čítá od začátku. Tím definuje OCR0 registr TOP hodnotu čítače 0 a tím i jeho rozsahu. CTC režim dovoluje lepší kontrolu nad frekvencí výstupního signálu na pinu OC0 a zjednodušuje počítání externích událostí.

Pro vytváření signálu na OC0 výstupu v CTC režimu stačí měnit logickou hodnotu na výstupu OC0 při každé události output compare (shodě TCNT0 a OCR0) – nastavení bity WGM01:0=1. Frekvenci takto generovaného signálu lze vypočítat podle vztahu

$$f_{OCn} = \frac{f_{clk_I/O}}{2 \cdot N \cdot (1 + OCRn)} \quad (1.4)$$

kde N je dělicí poměr předděličky podle Tab. 3.9 a $f_{clk/I/O}$ je frekvence hodin. Nastavení bitů registru TCCR0 ilustruje tabulka Tab. 3.1

COM01	COM00	Popis funkce
0	0	Normální funkce portu, OC0 odpojeno
0	1	Změna úrovně OC0 při shodě OCR0 a TCNT0.
1	0	Reset úrovně OC0 při shodě OCR0 a TCNT0.
1	1	Set úrovně OC0 při shodě OCR0 a TCNT0.

Tab. 3.1 – Output compare – Normal/CTC Mode, nastavení

Fast PWM mode

Režim fast PWM umožňuje generování vysokofrekvenčního PWM signálu. Od jiných režimů se liší jednostrannou pilou, proto je také rychlejší. Čítač čítá cyklicky od hodnoty BOTTOM po hodnotu MAX a restartuje se. V neinvertujícím režimu (COM01:0=2) viz. Tab. 3.2 se OC0 vynuluje při shodě TCNT0 a OCR0. V invertujícím režimu (COM01:0=3) Tab. 3.2 dojde při shodě k nastavení OC0. Díky dvojnásobné frekvenci oproti režimu Phase correct PWM je Fast PWM režim vhodný pro regulaci napájení, DAC aplikace apod. Vyšší kmitočet také umožňuje zmenšení součástí a tím i snížení celkové ceny. Kmitočet lze vypočítat ze vztahu

$$f_{OCnPWM} = \frac{f_{clk_I/O}}{N \cdot 256} \quad (1.5)$$

kde N je dělicí faktor předděličky a $f_{clk_I/O}$ je frekvence hodin.

COM01	COM00	Popis funkce
0	0	Normální funkce portu, OC0 odpojeno
0	1	Vyhrazeno
1	0	Reset úrovně OC0 při shodě OCR0 a TCNT0, Set OC0 při hodnotě BOTTOM. (neinvertující režim)
1	1	Set úrovně OC0 při shodě OCR0 a TCNT0, Reset OC0 při hodnotě BOTTOM. (invertující režim)

Tab. 3.2 – Compare output – Fast PWM mode, nastavení

Phase correct PWM mode

Režim fázově korektní PWM (WGM01:0=1) je založen na generování oboustranné pily. Čítač cyklicky čítá od hodnoty BOTTOM po hodnotu MAX. Po jejím dosažení dochází k čítání směrem dolů (od hodnoty MAX po hodnotu BOTTOM). V neinvertujícím módu (COM01:0=2) viz. Tab. 3.3 dochází k vynulování OC0 při shodě registrů TCNT0 a OCR0 a čítání směrem nahoru. Při čítání směrem dolů dojde k nastavení OC0. V invertujícím módu (COM01:0=3) při shodě registrů a čítání nahoru dojde k nastavení OC0 a při shodě a čítání dolů k vynulování OC0. Kmitočet signálu je možné vypočítat podle vztahu

$$f_{OCnPCPWM} = \frac{f_{clk_I/O}}{N \cdot 510} \quad (1.6)$$

kde N je dělicí faktor předděličky a $f_{clk_I/O}$ je frekvence hodin.

COM01	COM00	Popis funkce
0	0	Normální funkce portu, OC0 odpojeno
0	1	Změna úrovně OC0 při shodě OCR0 a TCNT0.
1	0	Reset úrovně OC0 při shodě OCR0 a TCNT0 čítání nahoru, Set OC0 při shodě OCR0 a TCNT0 čítání dolů. (neinvertující režim)
1	1	Set úrovně OC0 při shodě OCR0 a TCNT0 čítání nahoru, Reset OC0 při shodě OCR0 a TCNT0 čítání dolů. (invertující režim)

Tab. 3.3 – Output compare – Phase Correct PWM mode, nastavení

Bit	7	6	5	4	3	2	1	0
TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
Zápis/čtení (R/W)	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

Obr. 3.3 – TCCR0 registr – rozložení bitů

Režim	WGM01 (CTC0)	WGM00 (PWM0)	Režim čítače/časovače	TOP hodnota	Aktualizace OCR0	TOV0 příznak nastaven při
0	0	0	Normální	0xFF	Immediate	MAX
1	0	1	PWM, Phase correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Tab. 3.4 – Nastavení čítače/časovače 0, režimy

TCCR0—

Bit 7 – FOC0: Force Output Compare

Bit 7. Je aktivní pouze pokud WGM00 bit specifikuje mód, při kterém není vytvářen PWM signál.

Bity 6,3 – WGM0[1:0]: Waveform Generation Mode

Tyto bity nastavují, ve kterém registru bude hodnota určující maximum (TOP), směr čítání čítače a jaký typ signálu bude na výstupu.

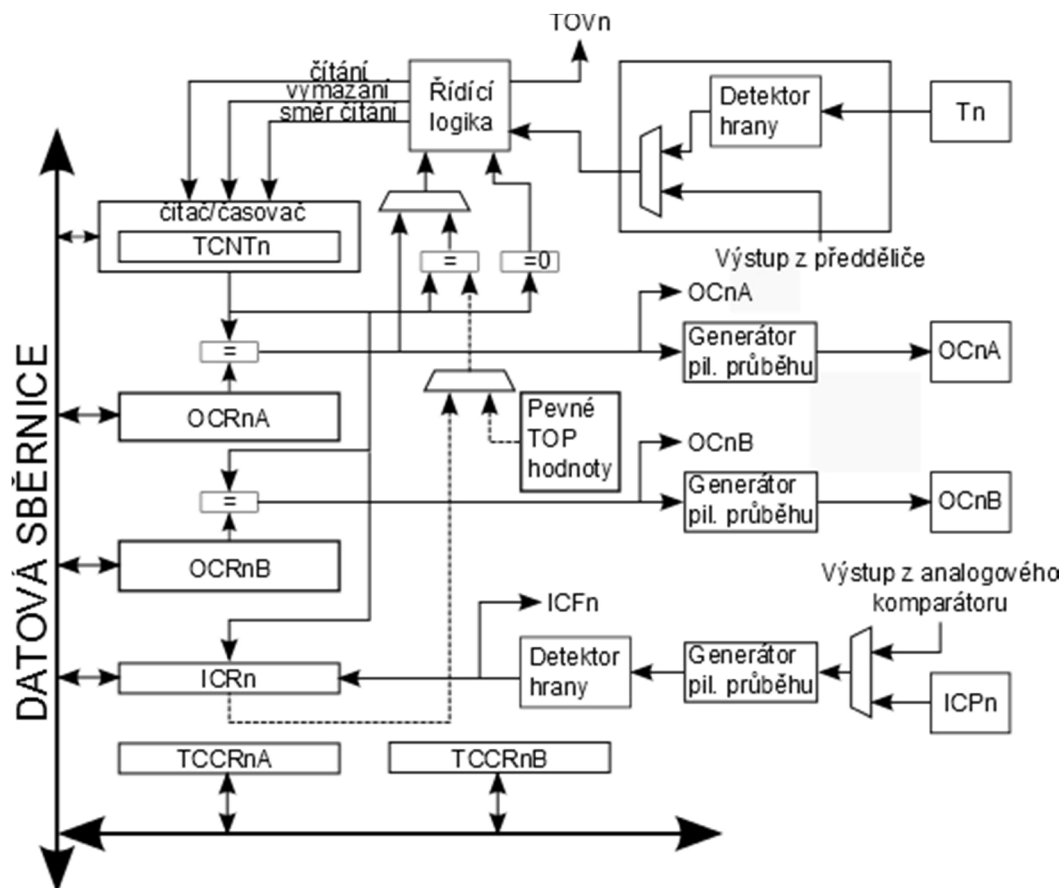
Bity 5:4 – COM0[1:0]: Compare Match Output Mode

Bity nastavují chování OC0 výstupu. Aby se zde nějaký signál objevil, musí být nastavený Data Direction Register příslušného pinu jako výstupní.

Bity 2:0 – CS02:0: Clock select

Kombinace bitů určuje dělicí poměr předděličky a tím i frekvenci výstupního signálu z pinu OCR0 viz. Tab. 3.9.

Čítač/časovač 1



Obr. 3.4 – Čítač/časovač 1

Čítač/časovač 1 umožňuje přesné programové časování (plánování událostí) a generování s časováním signálů. BOTTOM hodnota odpovídá číselně hexadecimálnímu číslu 0x0000, hodnota MAX je 0xFFFF (65536 dekadicky, čítač/časovač 1 je šestnáctibitový). Hodnota TOP udává horní hranici čítače a lze ji měnit. Může být buď

fixní 0x00FF, 0x01FF, 0x03FF nebo může být nastavena pomocí registrů OCR1A nebo ICR1.

Jednotka typu Input Capture zachycuje události přicházející na pin 20 (port D pin 6 – ICP1) a dokáže jim přiřadit i časový údaj. Díky tomu můžeme zjistit frekvenci, střidu a další vlastnosti signálu přicházejícího na pin ICP1. Toho využijeme pro získání údaje o otáčkách výstupního ventilátoru.

Pro události typu Input Capture je možné hardwarově nastavit úroveň spouštěcího signálu. To znamená, že jednotka na signál nezareaguje, pokud je nižší než nějaká daná úroveň. Úroveň takového spouštěcího signálu můžeme nastavit připojením signálu na vstup analogového komparátoru AIN po povolení bitu ACIC (Analog Comparator Input Capture) v registru ACSR (Analog Comparator Control and Status Register). Tuto variantu lze samozřejmě provést i softwarově.

Další užitečnou funkcí je potlačení rušení na vstupu Input Capture jednotky. Funkci povolíme zápisem logické jedničky do bitu ICNC1 v registru TCCR1B (Obr. 3.6). Potom je ale zapotřebí, aby se signál na pinu ICP1 neměnil po dobu 4 taktů mikrokontroléru, neboť filtrace funguje jako zpoždění signálu o čtyři hodinové takty mikrokontroléru.

Output Compare jednotka porovnává aktuální hodnotu čítače/časovače TCNT1 a registru OCR1x. Pokud nastane rovnost těchto dvou registrů, dojde v dalším hodinovém cyklu k nastavení příznaku Output Compare Flag (OCF1x). Pokud je povoleno přerušení (bitem OCIE_x = 1) dojde k přerušení typu Output Compare. Po návratu z přerušení je příznak Output Compare Flag (OCF_x) vynulován. Speciální funkce Output Compare jednotky A umožňuje nastavit TOP hodnotu čítače/časovače 1. Měníme tím tedy jeho rozlišení a zároveň periodu signálu z generátoru.

Procesor ATmega32 nabízí 4 režimy chování šestnáctibitového čítače/časovače 1

- Normal mode
- Clear timer on compare match - CTC mode
- Fast PWM
- Phase correct PWM
- Frequency and phase correct PWM

Zvolený režim nastavíme pomocí bitů WGM13÷10 v řídicích registrech čítače/časovače 1 TCCR1A (Obr. 3.5) a TCCR1B (Obr. 3.6).

Normal Mode

Nejjednodušším režimem funkce je Normal Mode. V tomto režimu je vždy směr čítání vzestupný a není zde použito nulování – čítač jednoduše přeteče, dosáhne-li své maximální hodnoty (je šestnáctibitový) takže pokud dosáhne hodnoty 65536 (TOP = 0xFFFF), nastaví příznak přetečení TOV1 (Timer Overflow Flag) a resetuje načítanou hodnotu (v registru TCNT1). Po návratu z rutiny přerušení se opět pokračuje v čítání od hodnoty BOTTOM (0x0000). Hodnotu čítače TCNT1 lze měnit během bez problémů i během čítání.

Funkce jednotky Input Capture v tomto režimu je velmi jednoduchá, nicméně je potřeba nepřekračovat rozlišení čítače velkým časovým intervalem mezi vstupními událostmi. Pokud je časový interval mezi vstupními událostmi příliš velký, je potřeba použít přerušení od přetečení nebo předděličku pro zvětšení rozlišení čítače.

Jednotku output compare lze využít ke generování přerušení v libovolném daném čase. Nastavení provedeme pomocí Tab. 3.5.

Clear Timer on Compare Match (CTC) Mode

V tomto režimu se používá OCR1A nebo ICR1 registrů pro nastavení rozlišení čítače.

Čítač startuje od hodnoty BOTTOM (0x0000) a čítá směrem nahoru. Při shodě registrů TCNT1 a OCR1A (při nastavených bitech WGM13:0=4) nebo ICR1 (při nastavených bitech WGM13:0=12) se nastaví do registru TCNT hodnota BOTTOM (0x0000). OCR1A nebo ICR1 tedy reprezentují TOP hodnotu čítače a určují tím rozlišení. V tomto režimu je možné lépe řídit frekvenci průběhu na pinu OC1A. Zjednodušuje také počítání externích událostí, kdy změnou hodnoty v ICR1 registru změníme rozlišení a tím frekvenci signálu.

Příznak shody OCF1A při shodě TCNT1 s OCR1A nebo ICF1 při shodě TCNT1 a ICR1 je nastaven při rovnosti aktuální hodnoty čítače s jedním z registrů. Díky tomu lze generovat přerušení pro oba způsoby definování TOP hodnoty čítače. Je zapotřebí dbát na to, aby nová definovaná TOP hodnota byla vyšší než aktuální hodnota v registru TCNT1. Pokud by tomu tak nebylo, čítač by zmeškal událost compare match (shodu registrů) a postupně by čítal až do přetečení (0xFFFF).

Pro generování signálu na výstupu OC1A je zapotřebí nastavit hodnotu bitů COM1A1:0 (Obr. 3.5). Výstup OC1A může mít podle nastavení těchto bitů různé chování. Nastavení popisuje Tab. 3.5, kde je popsáno i chování výstupního signálu na OC1A.

Frekvenci signálu OC1A vypočítáme ze vztahu

$$f_{OCnA} = \frac{f_{CLK_I/O}}{2N(1 + OCRnA)} \quad (1.7)$$

kde N určuje dělicí faktor (hodnota nastavená v předděličce), $f_{CLK_I/O}$ je frekvence oscilátoru a $OCRnA$ je číselná hodnota z registru Output Compare Register n kanál A.

COM1A1/ COM1B1	COM1A0/ COM1B0	Popis funkce
0	0	Normální funkce portu, OC1A/OC1B odpojeno
0	1	Změna úrovně OC1A při shodě OCR1A/OCR1B a TCNT1.
1	0	Reset OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1. (Nastavení výstupu na nulovou hladinu)
1	1	Set OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1. (Nastavení výstupu na jedničkovou hladinu)

Tab. 3.5 – Output compare – Normal/CTC Mode, nastavení

Fast PWM Mode

V režimu Fast PWM je možné generovat průběhy o vysoké frekvenci. Zásadním rozdílem oproti ostatním PWM režimům je generování jednostranné pily (čítání probíhá pouze jedním směrem - nahoru). Čítač začne čítat od hodnoty BOTTOM a čítá nahoru po hodnotu TOP. Potom se nastaví příznak přetečení čítače (TOV1). Pokud je povoleno přerušení při přetečení (TOIE=1 v registru TIMSK) nastane jeho obsluha, po návratu z něj restart čítače a opět nové čítání od hodnoty BOTTOM.

V neinvertujícím režimu Compare Output je při shodě registrů OCR1x a TCNT1 výstupní pin OC1x vynulován. Po dosažení hodnoty BOTTOM (resetu čítače) je nastaven. Při invertujícím režimu (řídí se bity COM1A1:0) je nastaven při shodě a vynulován při hodnotě BOTTOM. Díky jednostranné pile je možné nastavit dvakrát větší kmitočet než u módů Phase Correct a Frequency and Phase Correct PWM. Díky vyššímu kmitočtu lze použít menší součástky (cívky, kondenzátory) a tím zredukovat cenu a zabírající prostor použitých komponentů.

Rozlišení PWM v tomto módu je fixních 8, 9 nebo 10 bitů, popřípadě definované registrem ICR1 nebo OCR1A (definují TOP hodnotu). Pokud používáme registr ICR1 pro definování TOP hodnoty je opět potřeba zajistit, aby nová TOP hodnota byla větší než aktuální hodnota čítače. Pokud by se tak nestalo, čítač by zmeškal událost Output Compare (shodu registrů), čítal by až do MAX hodnoty 0xFFFF a potom by přetekl.

U registru OCR1A je odlišný mechanismus naplňování hodnotou. OCR1A registr je totiž dvojitě bufferovaný, což nám dává možnost kdykoliv změnit jeho hodnotu. Změna se projeví nejdříve v bufferu a až s dalším hodinovým cyklem a shodou TCNT1 a TOP hodnoty se provede v registru, takže nemůže dojít ke zmeškání Output Compare události a následnému přetečení čítače.

Minimální rozlišení jsou 2 bity (ICR1/OCR1A = 0x0003) a maximální rozlišení je 16 bitů (ICR1/OCR1A = MAX). Kmitočet PWM signálu můžeme vypočítat ze vztahu

$$f_{OCn\text{PWM}} = \frac{f_{CLK_I/O}}{N \cdot (1 + TOP)} \quad (1.8)$$

$f_{CLK_I/O}$ je frekvence oscilátoru, N je hodnota předděličky a TOP je maximální hodnota. Nastavení chování výstupu OC1A provádíme opět zápisem bitů COM1A1:0 viz Tab. 3.6.

COM1A1/ COM1B1	COM1A0/ COM1B0	Popis funkce
0	0	Normální funkce portu, OC1A/OC1B odpojeno
0	1	WGM13:0=15; Změna úrovně OC1A při shodě OCR1A/OCR1B a TCNT1, OC1B odpojeno Pro všechny ostatní nastavení WGM13:0, normální funkce portu, OC1A/OC1B odpojeno
1	0	Reset OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1. Set OC1A/OC1B při hodnotě TCNT1 = BOTTOM. (neinvertující režim)
1	1	Set OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1. Reset OC1A/OC1B při hodnotě TCNT1 = BOTTOM. (invertující režim)

Tab. 3.6 – Output compare – Fast PWM Mode, nastavení

Phase Correct PWM Mode

Režim Phase Correct je založený na obousměrném čítání – produkuje tedy symetrickou pilu. Čítač v tomto režimu cyklicky čítá od hodnoty BOTTOM do hodnoty TOP a zpět. V neinvertujícím režimu Output Compare je OC1x vynulován při shodě TCNT1 a OCR1x při čítání nahoru a nastavený při shodě a čítání směrem dolů. Pro invertující režim je chování OC1x opačné. Díky oboustrannému čítání je frekvence PWM menší, nicméně tento režim je výhodný pro řízení motorů.

Rozlišení může být opět fixních 8, 9 nebo 10 bitů, nebo nastavitelné pomocí registrů ICR1A nebo OCR1A. Minimální rozlišení jsou 2 bity (ICR1/OCR1A = 0x0003) a maximální rozlišení je 16 bitů (ICR1/OCR1A = MAX).

V módu Phase Correct je čítač inkrementován, dokud jeho hodnota nedosáhne jedné z fixních hodnot (0x00FF, 0x01FF nebo 0x03FF) nebo hodnoty v jednom z registrů ICR1 nebo OCR1A. Čítač tedy dosáhl TOP hodnoty a změnil směr čítání.

TOV příznak přetečení čítače je nastaven při dosažení čítače hodnoty BOTTOM. Pokud měníme TOP hodnotu, je potřeba zajistit, aby nová TOP hodnota byla větší než aktuální. Pokud by byla stará TOP hodnota menší než hodnota v jednom z registrů ICR1 nebo OCR1A, událost Output Compare (shoda registrů) by nikdy nenastala.

Protože se OCR1x registr aktualizuje při hodnotě TOP (viz. Tab. 3.8), může nastat situace, kdy dostaneme nesymetrický výstup. Pokud čítač běží a je změněna hodnota v OCR1x registru, může se stát, že je výstup nastaven/vynulován na událost Output Compare pro starou hodnotu a vynulován/nastaven pro novou hodnotu registru OCR1x. Řešením tohoto problému je použití Frequency and Phase Correct módu nebo změna hodnoty.

Kmitočet pulzně-šířkově modulovaného signálu určíme podle vztahu

$$f_{OCnPCPWM} = \frac{f_{CLK_I/O}}{2 \cdot N \cdot TOP} \quad (1.9)$$

kde $f_{CLK_I/O}$ je frekvence oscilátoru, N je hodnota předděličky a TOP je maximální hodnota. Nastavení bitů COM1A1:0 je uvedeno na Tab. 3.7.

COM1A1/ COM1B1	COM1A0/ COM1B0	Popis funkce
0	0	Normální funkce portu, OC1A/OC1B odpojeno
0	1	WGM13:0=9 nebo 14; Změna úrovně OC1A při shodě OCR1A a TCNT1, OC1B odpojeno (normální funkce portu). Pro všechny ostatní nastavení WGM13:0 normální funkce portu, OC1A/OC1B odpojeno
1	0	Reset OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1 při čítání nahoru. Set OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1 při čítání dolů.
1	1	Set OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1 při čítání nahoru. Reset OC1A/OC1B při shodě OCR1A/OCR1B a TCNT1 při čítání dolů.

Tab. 3.7 – Output compare –Frequency and Phase Correct PWM

Frequency and Phase Correct PWM Mode

Tento režim je podobný předešlému, opět generuje obousměrnou pilu. Čítač čítá cyklicky od hodnoty BOTTOM k hodnotě TOP a zpět opačným směrem. Pro neinvertující režim je výstup OC1x vynulován při shodě TCNT1 a OCR1x při čítání nahoru a vynulován při čítání směrem dolů. Při invertujícím režimu bude chování pinu OC1x opačné.

Hlavním rozdílem oproti Phase Correct Mode je moment, ve kterém dojde k aktualizaci registru OCR1x viz Tab. 3.8.

Rozlišení je dáno opět fixně – minimálně 2 bity (ICR1 nebo OCR1A 0x0003), maximálně 16 bitů (ICR1 nebo OCR1A nastaveno na MAX).

V módu Frequency and Phase Correct je čítač inkrementován, dokud jeho hodnota nedosáhne jedné z fixních hodnot (0x00FF, 0x01FF nebo 0x03FF) nebo hodnoty v jednom z registrů ICR1 nebo OCR1A. Čítač tedy dosáhl TOP hodnoty a bude měnit směr čítání.

Pokud měníme TOP hodnotu, je potřeba zajistit, aby nová TOP hodnota byla větší než aktuální. Pokud by byla stará TOP hodnota menší než hodnota v jednom z registrů ICR1 nebo OCR1A, událost Output Compare (shoda registrů) by nikdy nenastala.

Díky aktualizaci registru OCR1x při hodnotě BOTTOM je zaručeno, že úsek čítače čítajícího směrem nahoru bude trvat stejnou dobu jako úsek čítače čítajícího směrem dolů. Pulzy jsou tedy symetrické a frekvenčně korektní.

Kmitočet pulzně-šířkově modulovaného signálu lze vypočítat ze vztahu

$$f_{OCnxFPCPWM} = \frac{f_{CLK_I/O}}{2 \cdot N \cdot TOP} \quad (1.10)$$

kde $f_{CLK_I/O}$ je frekvence oscilátoru, N je hodnota předděličky a TOP je maximální hodnota. Nastavení chování výstupu OC1A provádíme zápisem bitů COM1A1:0 viz Tab. 3.6.

Bit	7	6	5	4	3	2	1	0
TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
Zápis/čtení (R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

Obr. 3.5 – Rozložení bitů v registru TCCR1A

Bit	7	6	5	4	3	2	1	0
TCCR1B	ICNC	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
Zápis/čtení (R/W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Počáteční hodnota	0	0	0	0	0	0	0	0

Obr. 3.6 – Rozložení bitů v registru TCCR1B

Režim	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Režim čítače/časovače	TOP hodnota	Aktualizace OCR1X	TOV1 příznak nastaven při
0	0	0	0	0	Normální	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR1	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR1A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR1	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR1A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR1	Immediate	MAX
13	1	1	0	1	Reserved	-	-	-
14	1	1	1	0	Fast PWM	ICR1	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR1A	BOTTOM	TOP

Tab. 3.8 – Nastavení čítače/časovače 1, režimy

TCCR1A

Bit 7:6 – COM1A1:0: Compare Output Mode pro kanál A

Bit 5:4 – COM1B1:0: Compare Output Mode pro kanál B

Tyto bity ovlivňují chování Output Compare jednotky. Pokud dojde k nastavení jednoho nebo obou z bitů COM1A1:0 nebo COM1B1:0, dojde k potlačení normální funkčnosti I/O portu, ke kterému je připojen. Nicméně, je ještě potřeba definovat port jako výstupní nastavením příslušných bitů v registru DDR (Data Direction Register), portům, odpovídajících pinům OC1A nebo OC1B. Nastavení bitů je uvedeno v popisu k jednotlivým módům na Tab. 3.5, Tab. 3.6 a Tab. 3.7.

Bitů 1:0 – WGM10:11 slouží ke zvolení správného režimu podle tabulky Tab. 3.8.

Bitů 3:2 – FOC1A a FOC1B jsou příznaky událostí Output Compare pro příslušné kanály (A/B)

TCCR1B

Bit 7 – ICNC1: Input Capture Noise Celler – Nastavením jedničky do tohoto bitu je aktivován filtr jednotky Input Capture. Aby filtr správně fungoval, vyžaduje čtyři shodné vzorky na vstupu Input Capture jednotky (ICP1) pro změnu hodnoty výstupu. Jednotky je tedy o čtyři hodinové cykly zpožděná pokud je potlačení rušivého signálu (filtr) u jednotky Input Capture povoleno.

Bit 6 – ICES1: Input Capture Edge Select – Tímto bitem můžeme nastavit spouštěcí úroveň pro příchozí události na pin ICP1. Pokud je bit nastaven na nulu, je spouštěcí úroveň sestupná hrana, pokud je bit nastaven na jedničku, spouštěcí úroveň je vzestupná úroveň. Pokud událost na vstupu je propuštěna skrze spouštěcí úroveň odpovídající podmínce v registru ICES1, hodnota čítače je zkopírována do ICR1 registru (Input Capture Register). Tato událost nastavuje příznak Input Capture Flag (ICF1), který můžeme využít ke generování přerušení typu Input Capture Interrupt pokud je toto přerušení povoleno (bit TICIE1 v registru TIMSK). Pokud se registr ICR1 využívá jako definice TOP hodnoty pro čítač, potom je pin ICP1 odpojen a funkce Input Capture je s ním deaktivována.

Bit 5 – Rezervovaný bit, slouží pro budoucí použití. Musí v něm být zapsána nula pro zajištění kompatibility, pokud používáme registr TCCR1B.

Bitů 4:3 – WGM13:12: Waveform Generation Mode viz popis na Obr. 3.5.

Bitů 2:0 – CS12:10: Clock select – Pomocí nich je nastavován dělicí faktor předděličky viz Tab. 3.9.

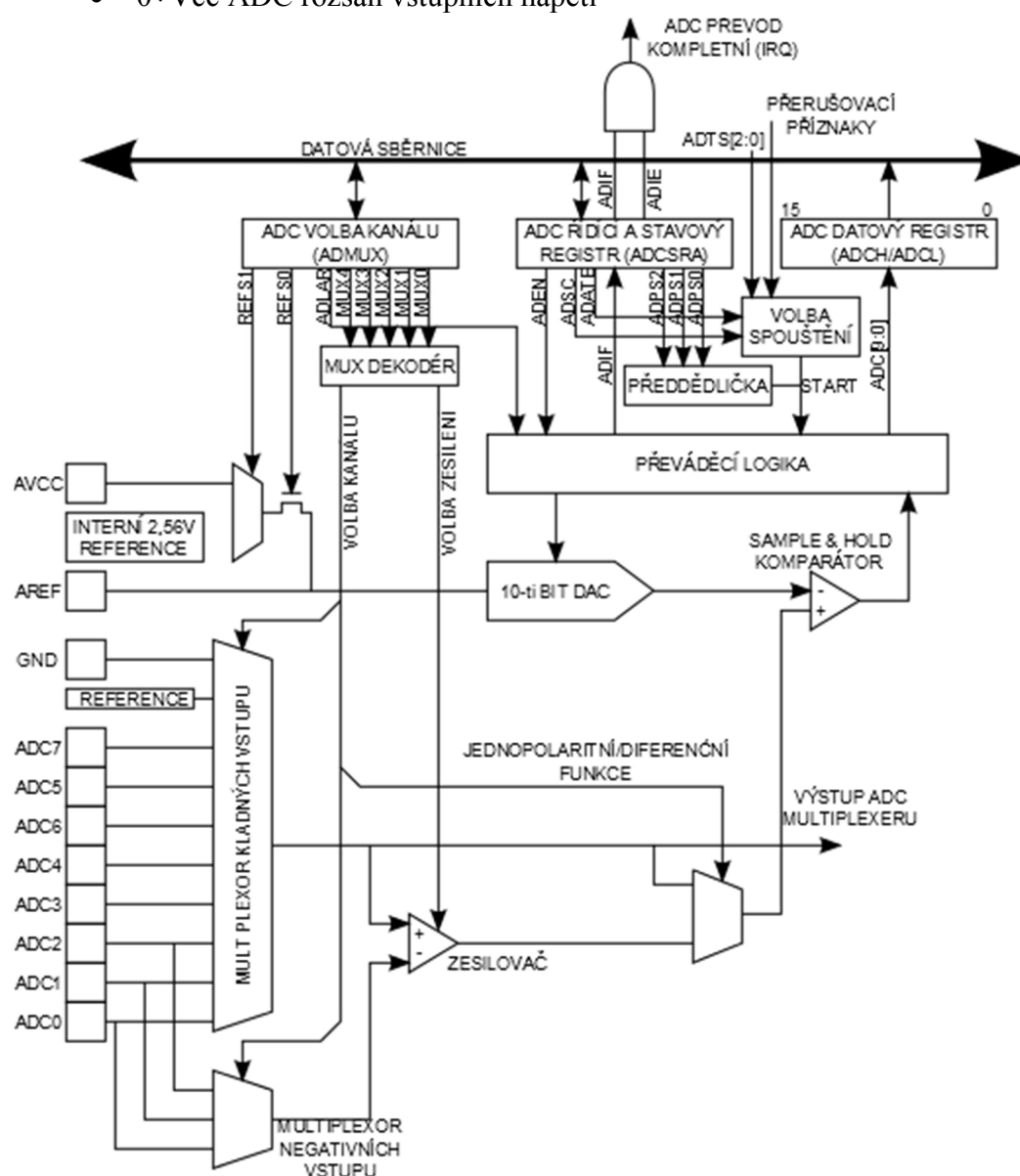
CS12	CS11	CS10	Popis
0	0	0	Bez zdroje hodinového signálu (čítač/časovač zastaven)
0	0	1	clk _{IO} /1 (Z předděliče)
0	1	0	clk _{IO} /8 (Z předděliče)
0	1	1	clk _{IO} /64 (Z předděliče)
1	0	0	clk _{IO} /256 (Z předděliče)
1	0	1	clk _{IO} /1024 (Z předděliče)
1	1	0	Externí zdroj hodinového signálu na Tn pinu. Reakce na sestupnou hranu.
1	1	1	Externí zdroj hodinového signálu na Tn pinu. Reakce na náběžnou hranu.

Tab. 3.9 – Nastavení dělicího faktoru předděličky

3.1.2 AD převodník

Pro řídicí analogové signály je použit AD převodník. Jeho vlastnosti v krátkém shrnutí jsou uvedeny níže:

- 10-ti bitové rozlišení
- Nelinearita 0,5 LSB
- $\pm 2\text{LSB}$ přesnost
- Doba převodu 13-260 μs
- 8 multiplexovaných jednopolaritních vstupů
- 7 diferenčních vstupů
- 2 diferenční vstupy s volitelným zesílením
- Volitelné zarovnání výsledku
- $0 \div V_{\text{cc}}$ ADC rozsah vstupních napětí



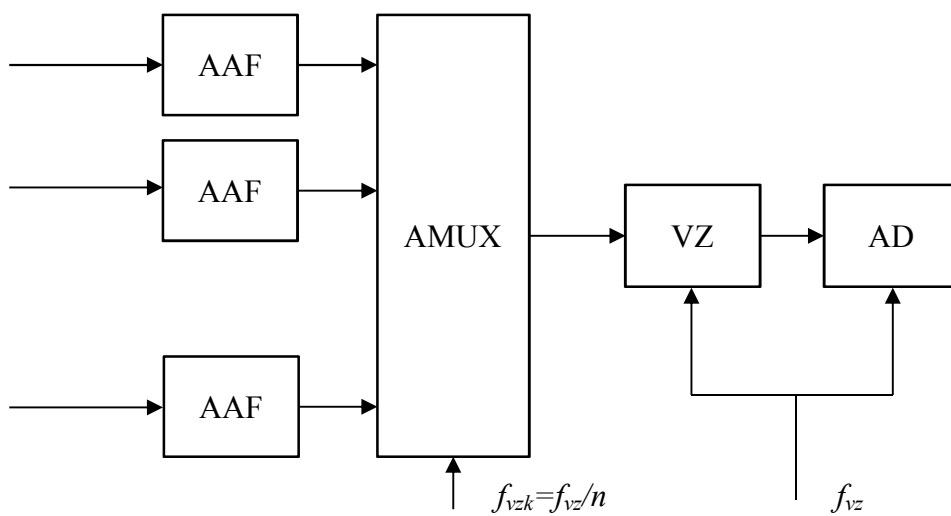
Obr. 3.7 – Blokové schéma AD převodníku ATmega32

Převodník u mikrokontroléru ATmega32 nalezneme na portu A, piny A0÷A7. K dispozici je 8 vstupů vztažených k 0V, 7 diferenčních vstupů vztažených ke sdílenému negativnímu vstupu ADC1 (pin A1, ostatní vstupy mohou být kladné) a další dva samostatné, diferenční (ADC1, ADC0; ADC3, ADC2). Tyto dva diferenční jsou ještě navíc vybaveny volitelným zesílením 0,20 a 46dB (1x, 10x nebo 200x (pouze 7-bit. rozliš.)). Převodník obsahuje, tzv. Sample & hold obvod, který navzorkuje signál a po danou dobu (než přijde další vzorek) drží jeho hodnotu během převodu. Mimo jiné je také potřeba zajistit, aby se napájecí napětí převodníku (má rozdílný pin (AREF)) od napájecího pinu mikrokontroléru Vcc) nelišilo o více jak $\pm 0,3V$.

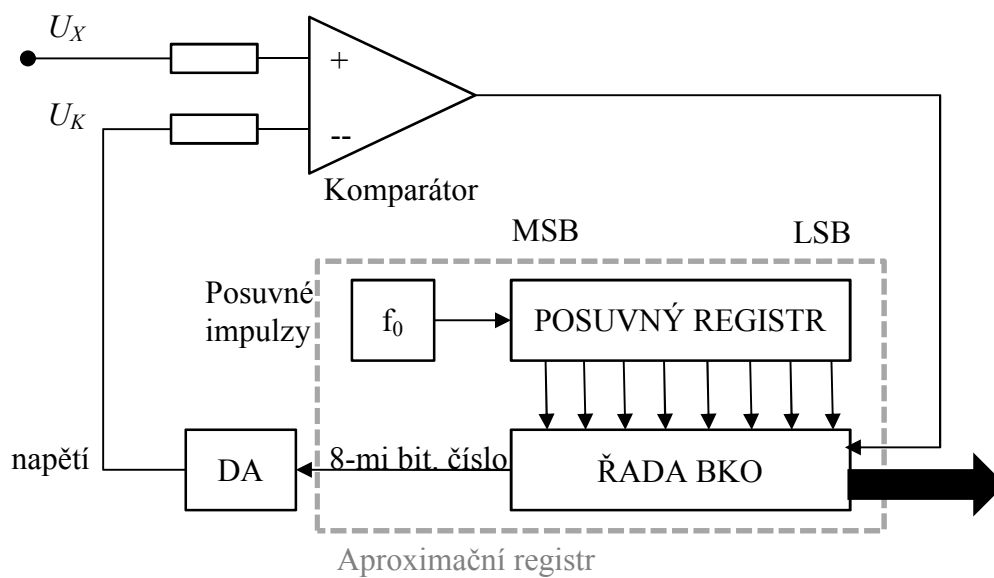
AD převodník je připojen na multiplexer, blokové schéma je vyobrazeno na Obr. 3.8, ve skutečnosti je fyzicky jen jeden [16]. AAF je antialiasing filtr, AMUX je multiplexor, VZ vzorkovač a AD konečná jednotka převodníku s postupnou aproximací Obr. 3.7. Postupně přepínané kanály jsou navzorkovány a vzorky převedeny do digitální podoby. Výhodou tohoto řešení je omezení celkového počtu potřebných obvodů a tím i celkové ceny takového převodníku. Nevýhodou je fázový posuv signálů. Existují i drobné modifikace v zapojení, které eliminují tuto nevýhodu. Jedním takovým je například řešení, kdy jsou vzorkovače umístěny na každý kanál samostatně. Potom nedochází k fázovému posuvu mezi signály.

Z hlediska typů se jedná o převodník s postupnou aproximací [15]. Principiální schéma je na Obr. 3.9. Takový převodník „posouvá“ jedničku od největšího (MSB) bitu k nejmenšímu. Následuje zpětný DA převod a test, zda napětí nastavené na spínačích (například řada BKO) je větší nebo menší než referenční (vztažné) napětí. Test provádí komparátor K, kde se obě napětí porovnávají, podle něj se dále nastavují příslušné bity. Posloupnost může být zhruba následující:

- $$U_x \geq \frac{U_{REF} \max}{2} \Rightarrow MSB = 1$$
- $$U_x \leq \frac{U_{REF} \max}{2} \Rightarrow MSB = 0$$
1. Krok
- $$U_x \geq \frac{U_{REF} \max}{4} \Rightarrow bit\ n-1 = 1$$
- $$U_x \leq \frac{U_{REF} \max}{4} \Rightarrow bit\ n-1 = 0$$
2. Krok
- .
- .
- .
- atd.



Obr. 3.8 – Uspořádání AD převodníku ATmega32



Obr. 3.9 – AD převodník s postupnou aproximací, blokové schéma

3.2 DA převodník

Pro převod výstupního signálu odpovídajícímu otáčkám vzduchem poháněného ventilátoru je použito osmibitového DA převodníku firmy Texas Instruments s odporovou sítí R-2R viz Obr. 3.10 [9].

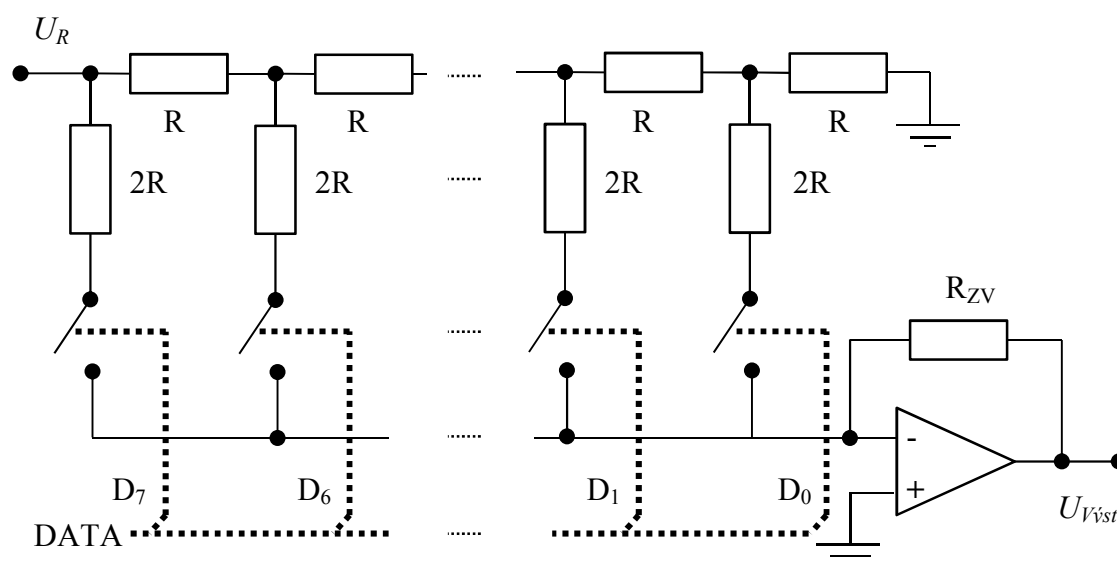
Činnost takového převodníku je následující:

Jednotlivé bity ovládají spínače (v tomto případě CMOS tranzistory) které postupně připojují další větve R-2R sítě (podle binárního čísla). Čím dále od zdroje referenčního napětí U_R se nachází sepnutý spínač, tím méně proudu přes něj teče (tím více odporů R je před sepnutým spínačem zařazeno). Jednotlivé bity jsou tedy převedeny na R-2R sítí odstupňované proudy. Ty jsou vedeny do operačního zesilovače. OZ je v tomto případě zapojen jako převodník I/U, kde převádí součet proudů přicházejících do invertujícího vstupu podle vztahu (1.11) na napětí U_{Vyst} [15],[13].

$$U_{Vyst} = \frac{I_{Vst}}{R_{ZV}} \quad (1.11)$$

Integrovaný obvod DAC0832LCN je průmyslovým převodníkem, vyráběným přímo pro použití s mikroprocesory. Jeho unikátní vlastností je dvojitě bufferovaný vstup. To znamená, že DA převodník obsahuje pomocný registr, ve kterém se vyskytuje vždy předchozí hodnota digitálního čísla na vstupu. Signálem WR1 je povolen zápis dat do vstupního registru, signálem WR2 pak ovlivňován výstup. Signály CS a XFER je obvod řízen, pokud se v zapojení vyskytuje vícekrát a je zapotřebí adresovat jednotlivé převodníky.

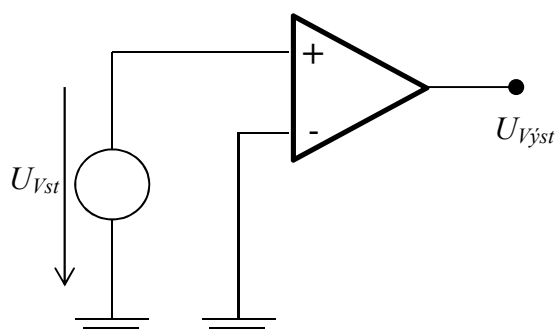
V modelu je použito tzv. Flow-Through zapojení, kdy změna na vstupu má okamžitý dopad na změnu napětí na výstupu. Nemí tudíž využito dvojitě bufferovaných vstupů.



Obr. 3.10 – DA převodník s odporovou sítí R-2R

3.2.1 Operační zesilovač

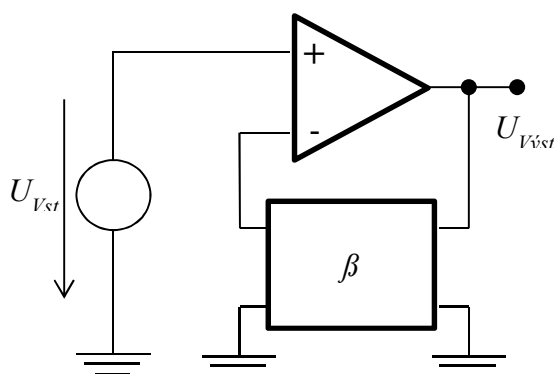
Pro zajištění činnosti DA převodníku je v zapojení použito několik operačních zesilovačů. Operační zesilovač je analogový elektronický obvod, zpracovávající napětí nebo proudy na vstupních svorkách [14]. Řídící napětí takového OZ je dáno rozdílem napětí na vstupních svorkách. Ty jsou invertující (-) nebo neinvertující (+). Výstupní napětí je tedy dáno rozdílem napětí neinvertujícího a invertujícího vstupu a na výstupu se objeví se zesilením A . Změna napětí na neinvertujícím (invertujícím) vstupu způsobí změnu napětí na výstupu v totožném (opačném) smyslu.



Obr. 3.11 – Operační zesilovač

Operační zesilovač má velké zesílení s otevřenou smyčkou zpětné vazby, proto většina zapojení využívá zpětnou vazbu. Ta umožňuje zlepšení nebo nastavení některých důležitých parametrů. Veliký vliv na vlastnosti zapojení má přenos zpětné vazby. Celkový přenos uzavřené zpětnovazební smyčky podle Obr. 3.12 je

$$A_U = \left| \frac{U_2}{U_1} \right| = \frac{A_o}{1 + \beta \cdot A_o} \quad (1.12)$$



Obr. 3.12 – OZ v zapojení se zápornou zpětnou vazbou

Zpětná vazba může mít stabilizující účinky. Pokud je přenos zpětnovazebního členu kladný ($\beta \cdot A > 0$), potom zpětnovazební napětí působí proti napětí na neinvertujícím vstupu U_{Vst} a výsledné zesílení se snižuje. Výhodou je ale stabilita

proti změnám parametrů, navíc záporná zpětná vazba zvyšuje šířku pásma, zmenšuje nelineární zkreslení a má vliv na vstupní a výstupní impedanci.

Zpětnovazební napětí u kladného zpětnovazebního zapojení OZ se naopak přičítá k napětí na neinvertujícím vstupu a tím zvyšuje výstupní napětí celého obvodu U_{Vyst} . Zpravidla je nežádoucí (zvyšování výstupního napětí nad určitou mez vede k destrukci dalšího obvodu) ale nemusí tomu tak být vždy. Kladná zpětná vazba se používá u oscilátorů, například v zapojení OZ jako komparátoru s hysterezí.

3.3 Číslicové vstupy/výstupy

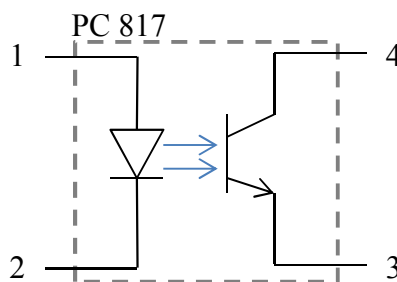
Model může být ovládán i číslicově, paralelním osmibitovým číslem. Pokud by byl propojen společný vodič modelu a řídicího prvku (např. PLC) a propojeny příslušné vodiče odpovídající jednotlivým bitům na svorky, vznikla by na společném vodiči zemní smyčka. Na dlouhém vedení, jehož odpor už není nulový, by vznikl úbytek napětí, který by se mohl projevit chybným potenciálem společného vodiče (země) [17]. Proto je zapotřebí model galvanicky oddělit. K tomu je možné využít několika způsobů:

- Elektromechanicky (relé)
- Elektromagneticky (transformátor)
- Opticky (optočlen)

První způsob nevyhovuje pro vysokofrekvenční aplikace, kde je časová konstanta relé velmi vysoká, navíc není relé mechanicky konstruováno na tak četná spínání a rozpínání.

Druhý způsob je pro VF techniku vhodnější, z hlediska prostoru trochu náročnější. Používá se zejména pro oddělení analogového signálu, jenž je pomocí optronu z důvodů jeho nelinearity obtížněji oddělitelný.

Nejvýhodnějším a často používaným způsobem galvanického oddělení ve slaboproudé elektrotechnice je použití optronu. Optron je nejčastěji integrovaný obvod obsahující světelný prvek (LED diodu) a fotocitlivý prvek (fotodioda, fototranzistor). Takový optron je zobrazen na Obr. 3.13. Skládá se z LED diody a fototranzistoru. Pokud prochází proud LED diodou D , dopadá světlo z ní emitované na bázi fototranzistoru T a ten se otevírá [14].



Obr. 3.13 – Optron PC817 – uspořádání

Samotný fototranzistor není konstruován na větší výkony, pokud chceme spínat větší proudy, je zapotřebí propojit tranzistory v Darlingtonově zapojení nebo jinak proudově posílit výstup.

3.4 Výkonová část

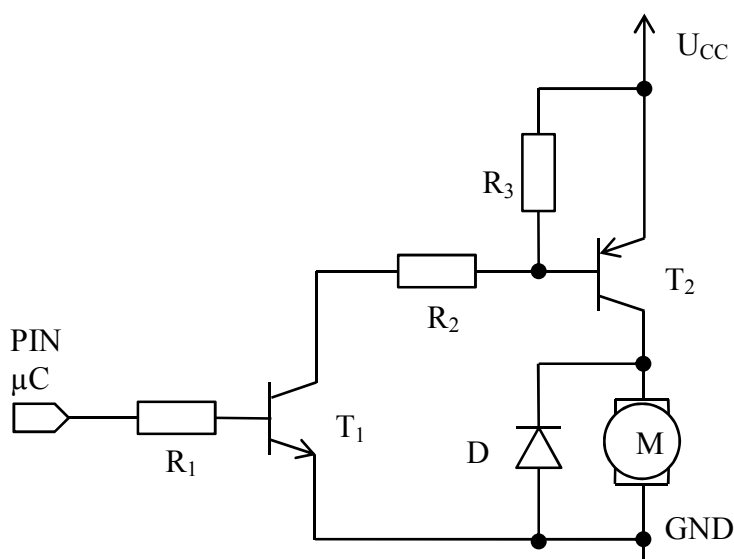
Napájení na desce zajišťuje lineární, kladný 5V stabilizátor L7805CV. Před ní je ještě zařazen 27V transil pro ochranu obvodu před napěťovými špičkami. V případě přetížení nebo zkratu na přípravku je osazena indikační LED dioda. Přepólování zabraňuje dioda, zapojená v propustném směru. Kondenzátory 330nF a 100nF (keramika) zde slouží jako lokální zásobníky energie zamezující zákmitům. Obvod obsahuje omezovač proudu, teplotní pojistku, a pokud je osazen odpovídajícím chladičem, je schopen dodat do zátěže přes 1A.

3.4.1 Tranzistorový spínač

Pro spínání ventilátoru se jmenovitým odběrem je použitý tranzistorový spínač na Obr. 3.14. Dále bude rozebrána jeho činnost.

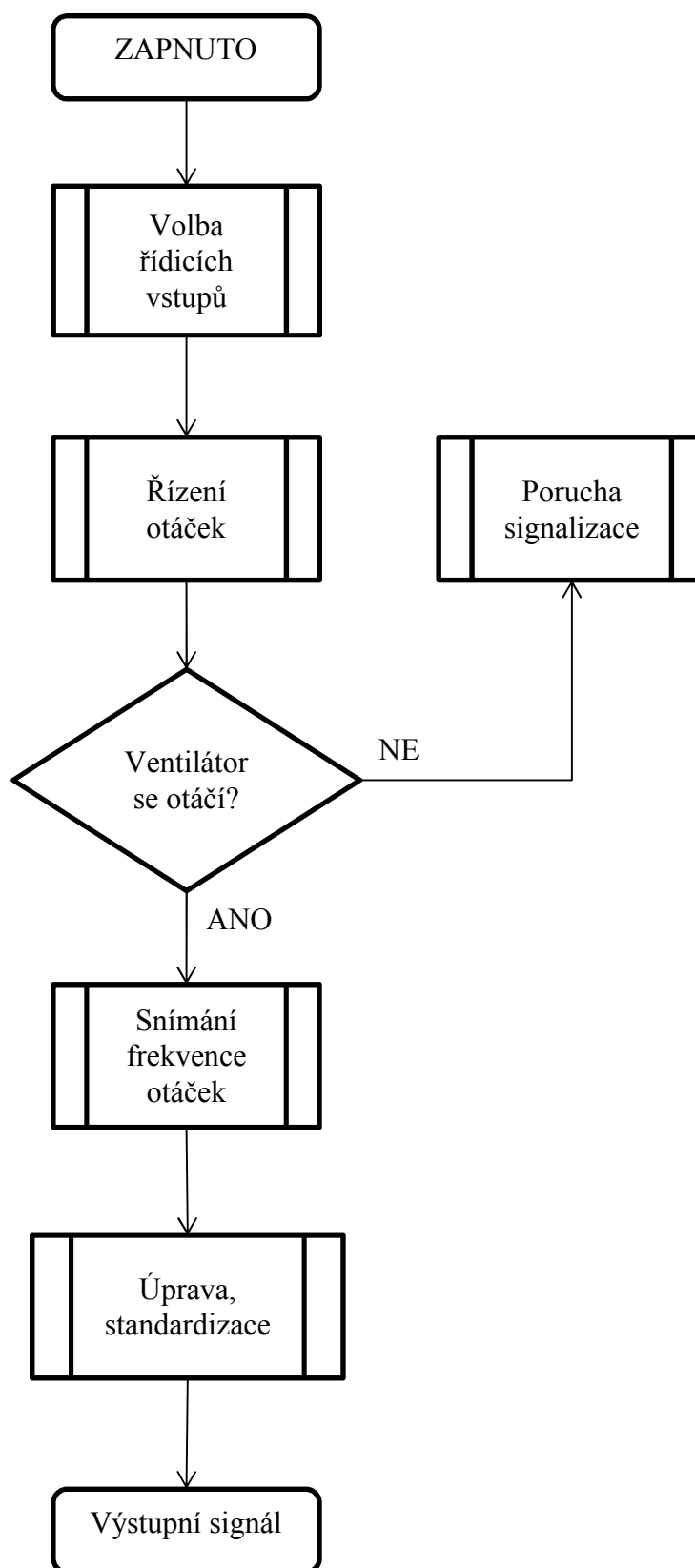
Pokud je na pinu mikrokontroléru 0V, na bázi tranzistoru T_1 neteče žádný proud - je uzavřen (nevede proud - typ NPN). Na bázi tranzistoru T_2 teče přes rezistor R_3 proud, je tedy uzavřen (jde o PNP typ). Na zátěži (motor) tedy není žádné napětí, ani přes ni neprotéká žádný proud.

Opačná situace nastane, když se objeví napětí na výstupu mikrokontroléru. Na bázi tranzistoru T_1 již poteče proud, tranzistor bude sepnut. Tím je ale stahován potenciál báze tranzistoru k T_2 zemi, proto i tranzistor T_2 přechází do saturace a zátěží začne protékat proud. Ochranná dioda D zamezuje napěťovým špičkám, které by se mohly objevit na zátěži induktivního charakteru. Cívka má totiž tu vlastnost, že po odpojení napětí se snaží udržet velikost a směr protékajícího proudu. Tím vznikne chvilkově přepětová špička, která by mohla zničit tranzistor.



Obr. 3.14 – Tranzistorový spínač

3.5 Vývojový diagram



Obr. 3.15 – Vývojový diagram

4 Program

V řídicím bloku probíhá ve smyčce se opakující program. Na začátku jsou nastaveny globální proměnné a určeny jejich datové typy. Obsluhy přerušení jsou uvedeny hned pod tímto blokem počátečních inicializací. Následuje hlavní rutina main a v ní se nachází nekonečná smyčka.

V ní jsou uvedeny části pro nastavení AD převodníku, čítačů, časovačů, ICP jednotky, generátorů signálů a jejich příslušných portů.

Každý blok je samostatně popsán, ke každému řádku je uveden komentář. Následující kód je nahrán v mikrokontroléru a zajišťuje obsluhu a řízení celého modelu. Software se nachází v Příloze 4.

5 Závěr

Cílem práce bylo navrhnout a sestavit model větrného tunelu, který by bylo možno ovládat několika způsoby. V modelu jsem se zabýval aplikací analogových a číslicových vstupů, výstupů a jeho řízením. Model by měl sloužit jako výukový přípravek v laboratorních programovatelných automatech. Tuto práci jsem si zvolil proto, abych si rozšířil své znalosti z konstrukce elektronických zařízení a aplikace mikroprocesorů v reálných situacích.

Práci lze rozdělit do několika částí.

V první části jsem navrhoval koncepci modelu. Zadání bylo jednoznačné a tak jsem se inspiroval již existujícím modelem v laboratoři číslicové řídicí techniky. Bylo zvoleno podobné řešení s odstraněním některých chyb tohoto modelu.

V další fázi jsem se zabýval volbou řídicího obvodu. Nejprve jsem se snažil model řídit pomocí integrovaných obvodů, specializovaných pro jiné účely. To se ukázalo jako špatná varianta, neboť tyto obvody neuměly zpracovávat číslicové signály. Další komplikací by bylo snímání otáček z druhého ventilátoru. Proto jsem se začal zabývat aplikací mikroprocesorů. Protože s konstrukcí zařízení nemám zkušenosti, musel jsem postupně nastudovat jednotlivé funkce mikroprocesoru, jeho programování a zapojení jeho vstupů a výstupů. Tato volba se ukázala jako správná, protože použitý mikrokontrolér obsahuje spoustu komponent (AD převodník a podobné) které by jinak bylo nutno připojovat.

Dále následovala tvorba řídicího programu, který zajišťuje řízení modelu.

Po navržení konceptu a volbě řídicího obvodu a jeho programování následoval návrh schéma elektrického zapojení. Přitom jsem vycházel z katalogových listů použitých obvodů a literatury. Při kreslení elektrického schéma jsem se snažil dodržovat návrhových pravidel a bezpečnostních opatření jak pro ochranu zařízení, tak uživatele.

Dalším stádiem byl návrh desky plošného spoje - layoutu. Při rozmístování jsem se snažil o co nejmenší rozměry desky plošného spoje a jednostrannou variantu, což umožnilo vyrobit desku v domácích podmínkách. První návrhy a vyrobené DPS

byly velké, ale s mnoha úpravami se mi podařilo velikost DPS zmenšit na nynější velikost 10x13cm.

Následovalo vyvrtání a osazení desky součástkami.

Dalším bodem bylo vytvoření vzorové laboratorní úlohy a ověření funkčnosti.

Při práci jsem postupoval samostatně a kladl důraz na vlastní řešení, které se ne vždy ukázalo jako nejlepší. Například při konstrukci desky plošného spoje, jejíž výroba je v domácích podmínkách velmi složitá, vznikly výrobní problémy. Používal jsem technologii podobnou sítotisku. Postup spočíval ve vytisknutí obrazce laserovou tiskárnou na hlazený papír, očištění a odmaštění cuprextitu a následném nažehlení tohoto papíru s obrazcem na cuprextit. Poté následuje odstranění papíru ve vodě (obrazec tvořený tonerem na cuprextitu zůstane) a leptání chloridem železitým. Konečné odstranění toneru probíhá za pomoci odmašťovadla. Nezřídka se stalo, že se deska podleptala nebo cesta „odplavala“. Další problémy nastaly při ověřování, kde model funkční zkouškou neprošel. Nefunkčnost může být způsobena právě vadou při výrobě DPS. Nicméně model sestavený na nepájivém poli fungoval správně, podle zadání. Zadání tedy bylo splněno.

Seznam použitých zkratk, veličin a symbolů

Zkratky

A	zesílení operačního zesilovače
AC	střídavá veličina
ADC	analogově – číslicový převodník (Analog Digital Converter)
ALU	aritmeticko–logická výpočetní jednotka mikroprocesorů
BLDC	bezkartáčový DC motor (Brushless DC)
CISC	procesor s komplexní instrukční sadou
DAC	číslicově – analogový převodník (Digital Analog Converter)
DC	stejnoseměrná veličina
DPS	deska plošných spojů
EC	motor s elektronickým komutátorem
OZ	operační zesilovač (ang. OA – operation amplifier)
PC	počítač (Personal Computer)
PLC Controller)	programovatelný logický kontrolér (Programmable Logical
PWM	pulzní šířková modulace
RISC	procesor s redukovanou instrukční sadou
VF	vysokofrekvenční technika

Seznam použité literatury

- [1] ROUBAL, Jiří a Petr HUŠEK. Regulační technika v příkladech. 1. vyd. Praha: BEN – technická literatura, 2011, xxii, 276, ii s. ISBN 978-80-7300-260-2.
- [2] CHLEBNÝ, Jan. *Automatizace a automatizační technika 3: Prostředky automatizační techniky*. 4. aktualizované vydání. Brno: Computer Press, 2009. ISBN 978-80-251-2523-6.
- [3] NOBILIS, Jiří, Josef PLETÁNEK, Jaroslav KUCHARA a Jiří KAZDA. *Teorie elektronických obvodů: Základní modulační principy*. 1. vydání. Pardubice: SPŠE, Karla IV. 13, 2005.
- [4] ĎAĎO, Josef a Marcel KREIDL. *Senzory a měřicí obvody*. Praha: ČVUT, 1996. ISBN 80-01-01500-9.
- [5] Katalogový list: ATmega 32A. Rev. 2466T-07/10. 2325 Orchard Parkway San Jose, CA 95131: Atmel, 2010. Dostupné z: <http://www.atmel.com/Images/doc2466.pdf>
- [6] Application note: 4-20mA Current Loop Primer DMS-AN-20. [online]. 2009, s. 3. Dostupné z: <http://www.murata-ps.com/data/meters/dms-an20.pdf>
- [7] VOJÁČEK, Josef. Proudová smyčka 4-20mA. [online]. 2005-11-29 [cit. 2013-01-04]. Dostupné z: <http://automatizace.hw.cz/clanek/2005112901>
- [8] JURA, Pavel. Signály a systémy: Část 1: Spojité signály. 2. opravené vydání. Brno: VUTIMUM, 2010.
- [9] Katalogový list: DAC0832LCN. Ver. SNAS534B May 1999, rev. March 2013. Texas Instruments, P.O. Box 655303, Dallas, Texas 75265. Dostupné z: <http://www.ti.com/lit/ds/symlink/dac0830.pdf>
- [10] MACHO, Tomáš. Mikroprocesory. 1. vydání. Brno: VUTIMUM, 2011.
- [11] MATOUŠEK, David. Práce s mikrokontroléry ATMEL AVR: [měření, řízení a regulace pomocí několika jednoduchých přípravků]. 1. vyd. Praha: BEN – technická literatura, 2006, 319 s. ISBN 80-730-0174-8.
- [12] Programujeme AVR v jazyku C: Zoznámte sa s MCU rady Atmel AVR. 1. vyd. 2012. Dostupné z: <http://svetelektro.com/clanky/kniha-programujeme-avr-v-jazyku-c-557.html>

- [13] VOBECKÝ, Jan a Vít ZÁHLAVA. Elektronika: součástky a obvody, principy a příklady. 3., rozš. vyd. Praha: Grada Publishing, 2005, 220 s. ISBN 80-247-1241-5.
- [14] MAŤÁTKO, Jan. Elektronika. 5. vyd., V [nakl.] Idea servis 3., rozš. vyd. Praha: Idea servis, 2002, 327 s. ISBN 80-859-7042-2.
- [15] LOSENICKÝ, Miroslav. SPŠE PARDUBICE, Karla IV. 13. *Automatizace*. Pardubice, 2008.
- [16] ČEJKA, Miloslav. Elektronické měřicí systémy 1. vydání. Brno: VUTIUM, 2008.
- [17] VRBA, Kamil. Konstrukce elektronických zařízení 1. vydání. Brno: VUTIUM, 2008.
- [18] KOLÁČNÝ, Josef. Elektrické pohony. 1. vydání. Brno: VUTIUM, 2003.

Seznam příloh

Příloha 1. DPS

Příloha 2. Schéma zapojení

Příloha 3. Rozpiska součástí

Příloha 4. Zdrojový kód

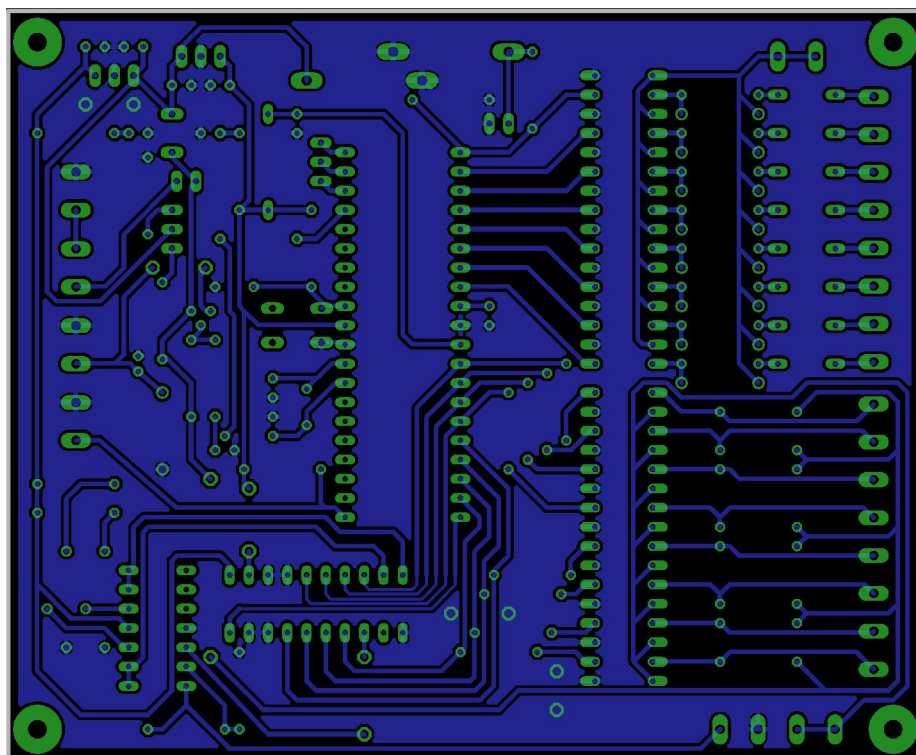
Příloha 5. Fotodokumentace

Příloha 6. CD - dokumentace

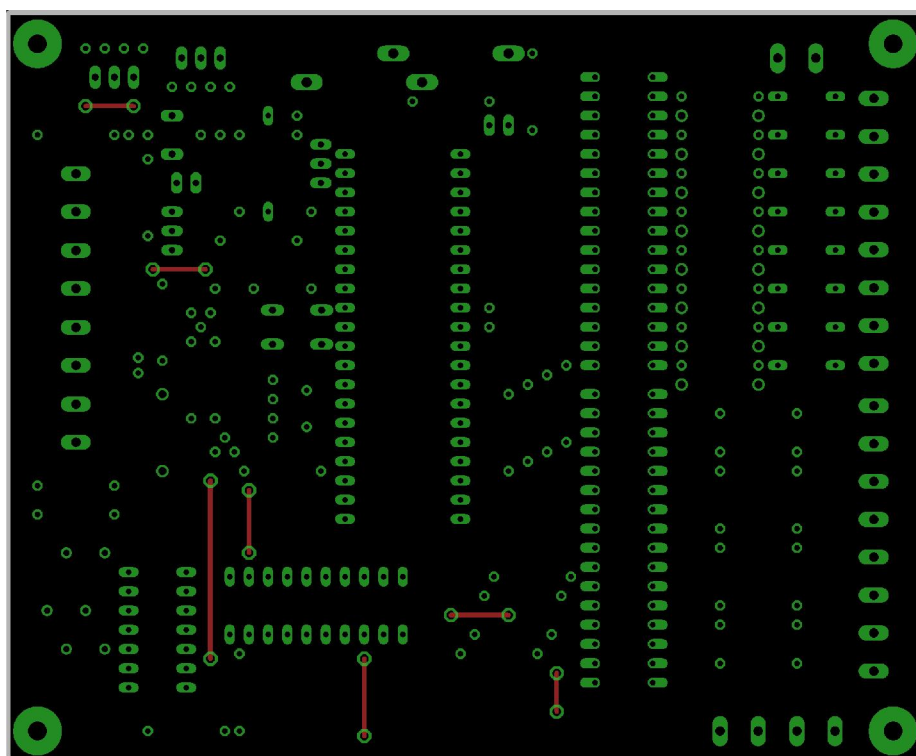
- schéma zapojení
- výkres DPS
- program
- vzorová laboratorní úloha

Příloha 1

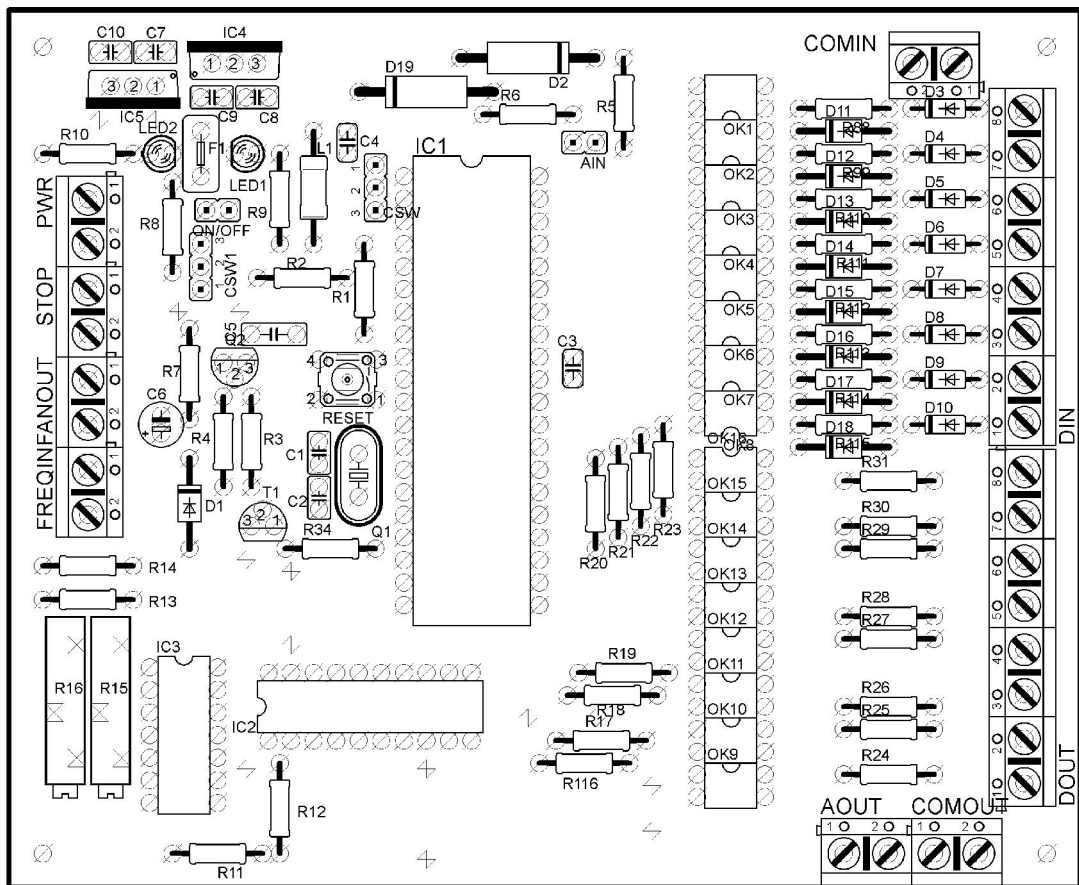
DPS – Spodní strana



DPS – Vrchní strana

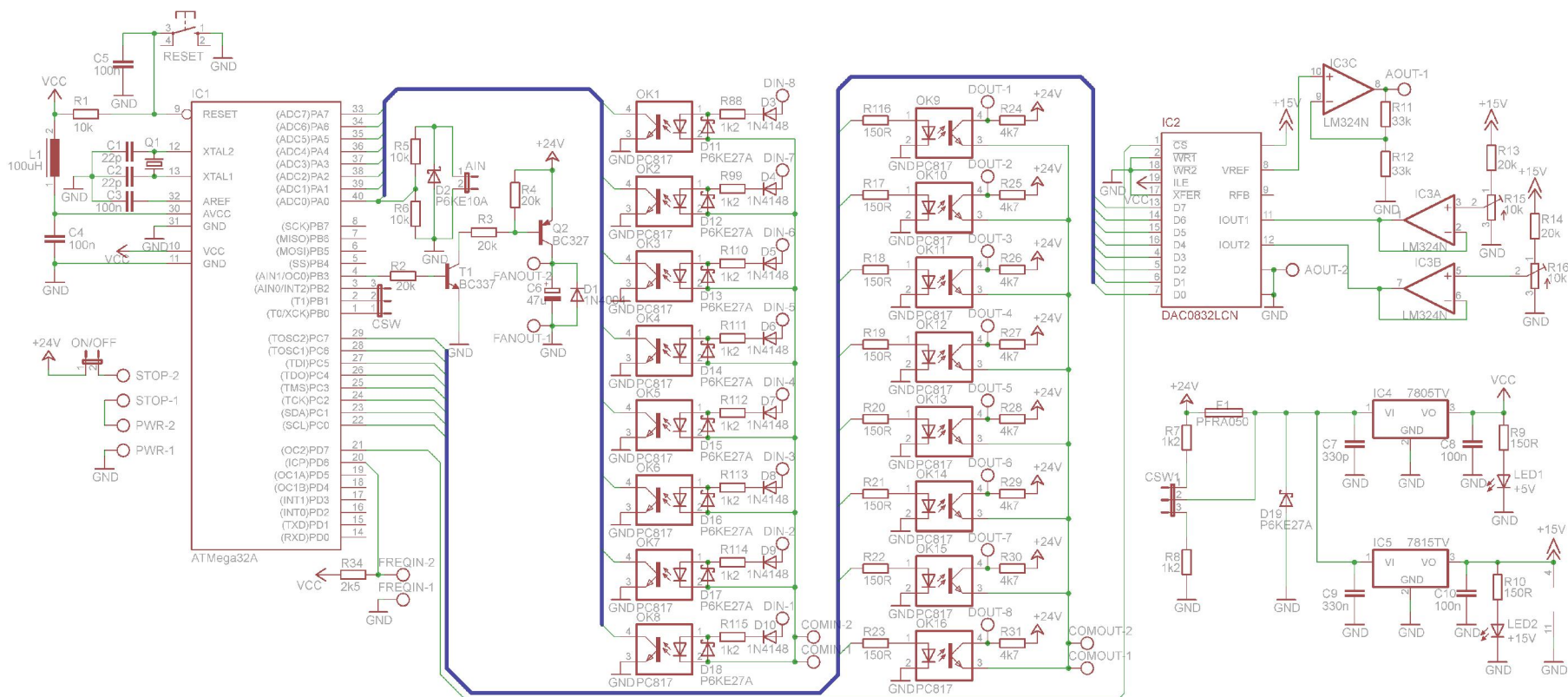


Osazovací výkres



Příloha 2

Schéma zapojení



Obr. 5.1 – Schéma zapojení

Příloha 3

Rozpiska součástek

Part	Value	Device	Package	Description
AIN		JP1E	JP1	JUMPER
AOUT		AK500/2	AK500/2	CONNECTOR
C1	22p	C-EU025-025X050	C025-025X050	CAPACITOR, European symbol
C2	22p	C-EU025-025X050	C025-025X050	CAPACITOR, European symbol
C3	100n	C-EU025-024X044	C025-024X044	CAPACITOR, European symbol
C4	100n	C-EU025-024X044	C025-024X044	CAPACITOR, European symbol
C5	100n	C-EU050-025X075	C050-025X075	CAPACITOR, European symbol
C6	47u	CPOL-EUE2-5	E2-5	POLARIZED CAPACITOR, ES
C7	330p	C-EU025-025X050	C025-025X050	CAPACITOR, European symbol
C8	100n	C-EU025-025X050	C025-025X050	CAPACITOR, European symbol
C9	330n	C-EU025-025X050	C025-025X050	CAPACITOR, European symbol
C10	100n	C-EU025-025X050	C025-025X050	CAPACITOR, European symbol
COMIN		AK500/2	AK500/2	CONNECTOR
COMOUT		AK500/2	AK500/2	CONNECTOR
CSW		JP2E	JP2	JUMPER
CSW1		JP2E	JP2	JUMPER
D1	1N4004	1N4004	DO41-10	DIODE
D2	P6KE10A	P6KEXXA	C1702-15	DIODE
D3	1N4148	1N4148DO35-7	DO35-7	DIODE
D4	1N4148	1N4148DO35-7	DO35-7	DIODE
D5	1N4148	1N4148DO35-7	DO35-7	DIODE
D6	1N4148	1N4148DO35-7	DO35-7	DIODE
D7	1N4148	1N4148DO35-7	DO35-7	DIODE
D8	1N4148	1N4148DO35-7	DO35-7	DIODE
D9	1N4148	1N4148DO35-7	DO35-7	DIODE
D10	1N4148	1N4148DO35-7	DO35-7	DIODE
D11	P6KE27A	P4KEXX	DO41-10	DIODE
D12	P6KE27A	P4KEXX	DO41-10	DIODE
D13	P6KE27A	P4KEXX	DO41-10	DIODE
D14	P6KE27A	P4KEXX	DO41-10	DIODE
D15	P6KE27A	P4KEXX	DO41-10	DIODE
D16	P6KE27A	P4KEXX	DO41-10	DIODE
D17	P6KE27A	P4KEXX	DO41-10	DIODE
D18	P6KE27A	P4KEXX	DO41-10	DIODE
D19	P6KE27A	P6KEXX	CB417-15	DIODE
DIN		AK500/8	AK500/8	CONNECTOR
DOUT		AK500/8	AK500/8	CONNECTOR
F1	PFRA050	TE5	TE5	FUSE
FANOUT		AK500/2	AK500/2	CONNECTOR
FREQIN		AK500/2	AK500/2	CONNECTOR
IC1		MEGA32-P	DIL40	MICROCONTROLLER

IC2		DAC0832LCN	DIL20	
IC3	LM324N	LM324N	DIL14	OP AMP
IC4	7805TV	7805TV	TO220V	Positive VOLTAGE REGULATOR
IC5	7815TV	7815TV	TO220V	Positive VOLTAGE REGULATOR
L1	100uH	MICC12	MICC12	INDUCTOR
1.01	+5V	LED3MM	LED3MM	LED
1.02	+15V	LED3MM	LED3MM	LED
OK1	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK2	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK3	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK4	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK5	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK6	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK7	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK8	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK9	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK10	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK11	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK12	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK13	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK14	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK15	PC817	PC817	DIL04	SHARP OPTO COUPLER
OK16	PC817	PC817	DIL04	SHARP OPTO COUPLER
ON/OFF		JP1E	JP1	JUMPER
PWR		AK500/2	AK500/2	CONNECTOR
Q1		CRYSTALHC49U70	HC49U70	CRYSTAL
Q2	BC327	BC327	TO92	PNP Transistor
R1	10k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R2	20k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R3	20k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R4	20k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R5	10k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R6	10k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R7	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R8	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R9	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R10	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R11	33k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R12	33k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R13	20k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R14	20k	R-EU_0207/10	0207/10	RESISTOR, European symbol
R15	10k	R-TRIMMT18	RTRIMT18	Trimm resistor
R16	10k	R-TRIMMT18	RTRIMT18	Trimm resistor
R17	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R18	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R19	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol

R20	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R21	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R22	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R23	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
R24	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R25	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R26	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R27	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R28	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R29	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R30	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R31	4k7	R-EU_0207/10	0207/10	RESISTOR, European symbol
R34	2k5	R-EU_0207/10	0207/10	RESISTOR, European symbol
R88	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R99	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R110	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R111	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R112	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R113	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R114	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R115	1k2	R-EU_0207/10	0207/10	RESISTOR, European symbol
R116	150R	R-EU_0207/10	0207/10	RESISTOR, European symbol
RESET		10-XX	B3F-10XX	OMRON SWITCH
STOP		AK500/2	AK500/2	CONNECTOR
T1	BC337	BC337	TO92	NPN TRANSISTOR

Příloha 4

Zdrojový kód

```
/*
 * GccApp13WindTune1.c
 *
 * Created: 10.10.2012 14:42:28
 * Author: Martin
 */

#include <avr/io.h>
#include <avr/interrupt.h>

/*
 * Initialization of global variables
 */
volatile uint8_t freq = 0, pwmValue = 0;
uint16_t pulse;

/*
 * AD converter A Interrupt Service Routine/Interrupt Handler
 */

ISR(ADC_vect)
{
    pwmValue = ADCH; // if AD conversion finished, set new value for
    generating PWM
}

/*
 * Timer Counter 0 Overflow A Interrupt Service Routine/Interrupt Handler
 */

ISR(TIMER0_OVF_vect)
{
    OCR0 = pwmValue; // if TCNT0 matches BOTTOM, set pwmValue to OCR0
}

/*
 * Timer Counter 1 Input Capture A Interrupt Service Routine/Interrupt
 * Handler
 */

ISR(TIMER1_CAPT_vect)
{
    pulse = ICR1;
    TCNT1 = 0;
    PORTB ^= (1 << PORTB6); // ICP interrupt test
}

/*
 * Main routine
 */

int main(void)
{
    // GLOBAL SETTINGS (PORT DIRECTIONS, PULL-UPS)
    DDRA |= (1 << DDRA0); // pin A0 as input for ADC0
    DDRB &= ~(1 << DDRB0); // pin B0 as input (analog/digital control
    switch)
    PORTB &= ~(1 << PORTB0); // pull-up resistor
}
```

```

DDRB &= ~(1 << DDRB1); // pin B1 as input (analog/digital feedback
switch)
PORTB &= ~(1 << PORTB1); // pull-up resistor
DDRC = 0xff; // PORTC 8-bit & analog feedback out
DDRD |= (1 << DDRD7); // pin D7 as output (CS for analog
feedback)

sei(); // global interrupt enable function

//PWM settings (waveform generator)
//*****
DDRB |= (1 << DDRB3); // OC0 (output compare 0) pin B3 output
TCCR0 |= (1 << WGM00) | (1 << COM01) | (1 << COM00) | (1 << CS00); //
phase correct PWM mode, fclk
TIMSK |= (1 << TOIE0); // overflow interrupt enable from TCNT0

//Counter 1 settings (speedometer)
//*****
DDRD &= ~(1 << DDRD6); // pin D6 (ICP1) as input
//PORTD |= (1 << PORTD6); // pin D6 (ICP1) pull-up resistor
TIMSK |= (1 << TICIE1); // interrupt from ICP1 unit
TCCR1B |= (1 << CS12) | (1 << CS10); // (1 << ICES1) | rising edge
trigger, fclk/1024 => Tcnt1=64us

while(1)
{
    if(PINB & (1 << PINB0))
    {
        //AD conversion settings (analog controlling)
        //*****
        ADMUX |= (1 << REFS0) | (1 << ADLAR); // 5V reference,
        left adjustment of result
        ADCSRA |= (1 << ADEN) | (1 << ADSC) | (1 << ADATE) | (1
        << ADIE) | (1 << ADPS0) | (1 << ADPS1) | (1 << ADPS2); //
        AD enable = 1; AD start conversion = 1; AD auto trigger
        enable = 1; AD interrupt enable = 1; AD prescaler 0,1,2
        = 1 => 16M/128 = 125kHz AD conversion frequency
        SFIOR |= (1 << ADTS2); // trigger source = TCNT0
        overflow
    }
    else
    {
        //Settings for 8bit number on PORTB (digital 8bit
        controlling)
        //*****
        ADCSRA = 0x00; // AD converter off
        DDRA = 0x00; // PORTA as an input port (for 8bit number)
        PORTA = 0xff; // pull-up resistor
        pwmValue = ~PINA; // value PINA to waveform generator
    }
    //Settings for digital feedback (8 bit out)
    //*****
    if (pulse)
    {
        PORTC = (PINB & (1 << PINB1)) ? 15625/(2*pulse) :
        ~(15625/(2*pulse));
        pulse = 0;
    }

    //Settings for analog feedback (analog out)
    //*****
    if (PINB & (1 << PINB1))
        PORTD |= (1 << PORTD7); // chip select ON

```

```
        else
            PORTD &= ~(1 << PORTD7); // chip select OFF
            //never ending loop
    }
}
```

Příloha 5

Vzorová laboratorní úloha

ÚKOL

Zjistěte účinnost větrného tunelu. Regulujte otáčky vstupního ventilátoru od nulové hodnoty po maximální a odečtěte otáčky výstupního. Vyzkoušejte regulaci jak s digitálními vstupy a výstupy, tak pro analogové verze a zjistěte, který způsob je lepší.

POPIS EXPERIMENTU

1. Spuštěním programu RSLinx zjistíme IP adresu našeho automatu a odečteme typy, verze a revize karet použitých pro řešení úlohy.
2. Po spuštění RSLogix 5000 nastavíme cestu k požadovanému automatu, přidáme karty s odpovídajícím počtem vstupů/výstupů použité v naší úloze a nastavíme jejich verze, popř. revize.
3. Vytvořte program podle popisu.

Program má za úkol nastavování různých hodnot otáček v rozsahu MIN-MAX, přičemž 0 ot/min odpovídá (0)_H a 2600 ot/min odpovídá hodnotě (FF)_H (v dekadické soustavě 0 a 255). Proved'te standardizaci a tímto osmibitovým číslem regulujte model.

Z výstupního ventilátoru odečítejte frekvenci otáček a proved'te programový přepočet. Ze zjištěných rozdílů otáček vstupního a výstupního ventilátoru určete účinnost takového modelu (ztráty v tunelu, ložiska ventilátorů apod. budou zmenšovat otáčky výstupního ventilátoru).

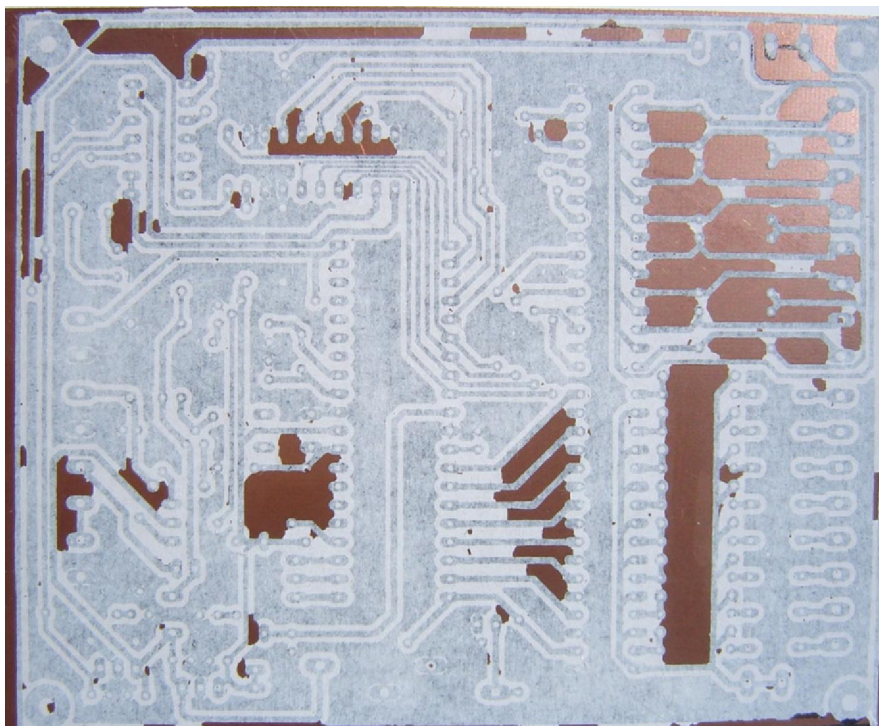
Podobně tak pro analogový vstup a výstup platí, že 0ot/min odpovídá 0V a 2600ot/min zase 10V. Proved'te opět standardizaci a zopakujte. Rychlost zvyšujte postupně, se zpožděním cca 5 sekund.

Seznam vstupů a výstupů

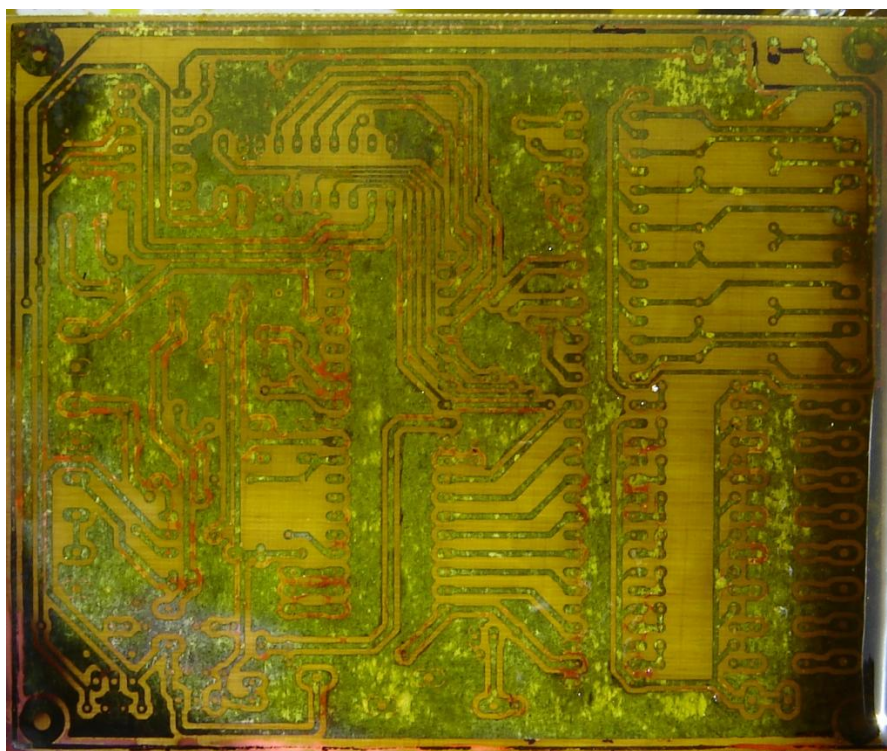
karta	alias	typ karty	popis
DI	DIN0	Digital Input	data vstup 0. bit
DI	DIN1		data vstup 1. bit
DI	DIN2		data vstup 2. bit
DI	DIN3		data vstup 3. bit
DI	DIN4		data vstup 4. bit
DI	DIN5		data vstup 5. bit
DI	DIN6		data vstup 6. bit
DI	DIN7		data vstup 7. bit
DO	DOUT0	Digital Output	data výstup 0. bit
DO	DOUT1		data výstup 0. bit
DO	DOUT2		data výstup 0. bit
DO	DOUT3		data výstup 0. bit
DO	DOUT4		data výstup 0. bit
DO	DOUT5		data výstup 0. bit
DO	DOUT6		data výstup 0. bit
DO	DOUT7		data výstup 0. bit
AI	AIN	Analog input	analogový vstup
AO	AOUT	Analog output	analogový výstup

Příloha 6

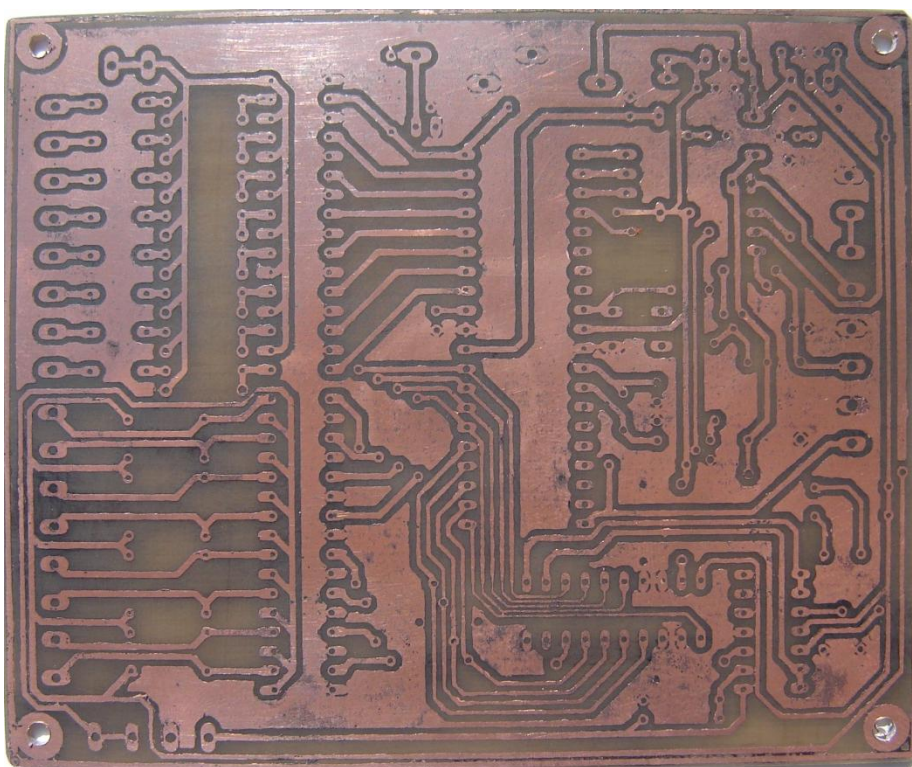
Fotodokumentace



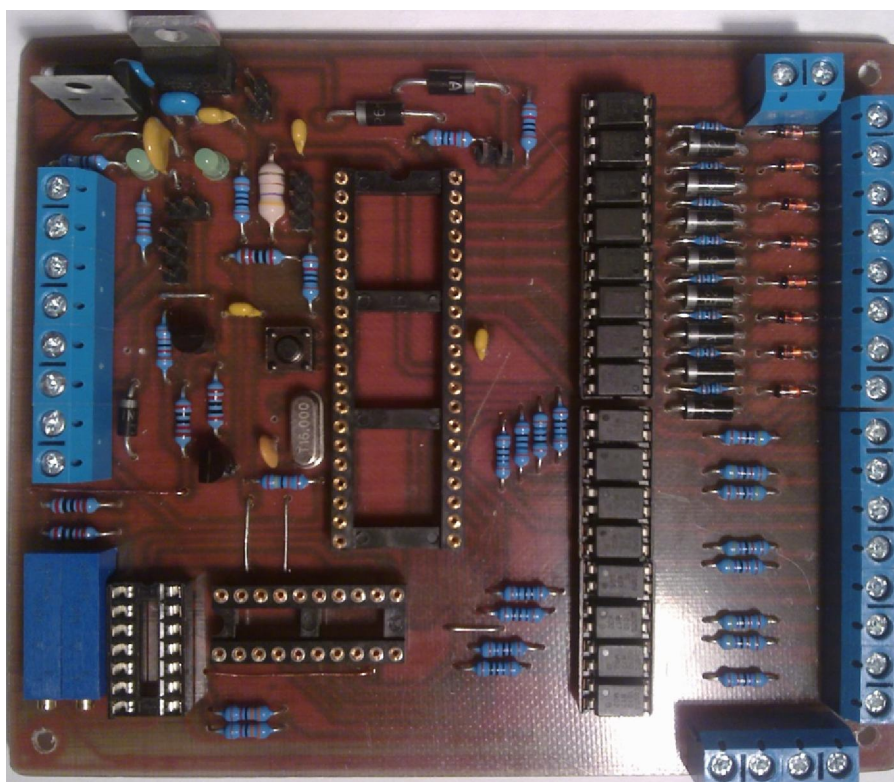
Sítotisk leptacího obrazce



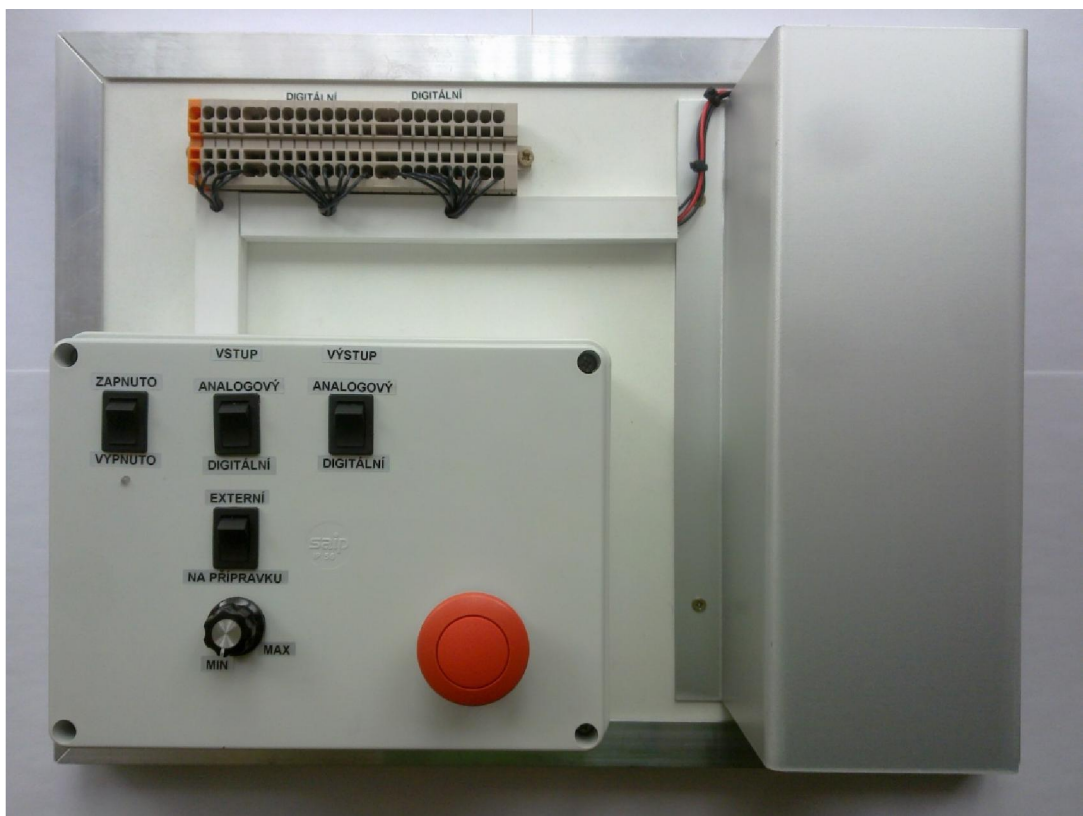
Leptání v chloridové lázni



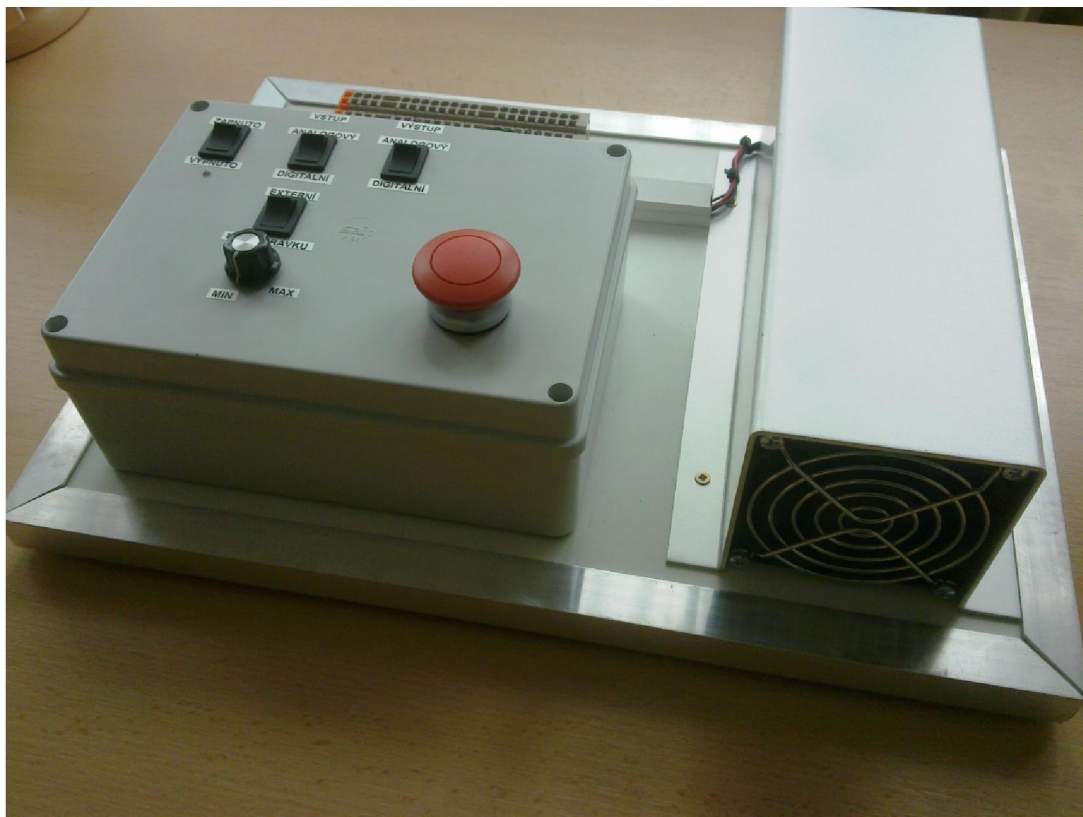
Výsledná deska



Osazená deska plošného spoje



Konečná sestava modelu



Model větrného tunelu