

**UNIVERZITA PARDUBICE**  
Fakulta elektrotechniky a informatiky

**SERVOPOHON SE STEJNOSMĚRNÝM MOTOREM**

Filip Bellinger

Bakalářská práce

2020

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Filip Bellinger**  
Osobní číslo: **I17061**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Řízení procesů**  
Téma práce: **Servopohon se stejnosměrným motorem**  
Zadávající katedra: **Katedra řízení procesů**

### Zásady pro vypracování

Cílem práce je návrh a realizace konstrukce servopohonu se stejnosměrným motorem. Bude proveden návrh jak mechanické konstrukce, tak i elektronické řídicí jednotky pro ovládání stejnosměrného motoru pohonu. V teoretické části práce bude provedena řešerše zadaného tématu se zaměřením na obdobná, komerčně vyráběná zařízení. V praktické části práce bude proveden kompletní návrh a realizace konstrukce servopohonu se stejnosměrným motorem. Základním stavebním prvkem konstrukce bude zvolený typ mikropočítače z řady ATmega. Konstrukce zařízení bude navržena a řešena jako modulární, díky tomu bude umožněna její snadná modifikace a případná modernizace. V rámci zpracování práce bude realizován 3D model navržené konstrukce v CAD software DesignSpark Mechanical. Mechanické díly konstrukce budou realizovány s podporou 3D tisku. Nedílnou součástí práce bude kompletní výrobní dokumentace, včetně okomentovaných zdrojových kódů mikropočítače a podrobného uživatelského a montážního návodu.

Rozsah pracovní zprávy: **40**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

**Seznam doporučené literatury:**

MATOUŠEK, D., Práce s mikrokontroléry ATMEL AVR-3.díl, edice uP a praxe, 2. vydání, BEN – technická literatura, 2006, ISBN 80-7300-209-4  
ZÁHLAVA, V., Návrh a konstrukce DPS, BEN-technická literatura, 2010, ISBN 978-80-7300-266-4  
NOVÁK, P., Mobilní roboty: pohony, senzory, řízení. Vyd. 1. Praha: BEN – technická literatura, 2004, 247 s. ISBN 80-730-0141-1.  
MAIXNER, L. a kol., Mechatronika, Brno, Computer Press, 2006, ISBN 80-251-1299-3

Vedoucí bakalářské práce: **Ing. Libor Havlíček, Ph.D.**  
Katedra řízení procesů

Datum zadání bakalářské práce: **17. prosince 2019**  
Termín odevzdání bakalářské práce: **7. května 2020**

L.S.

---

**Ing. Zdeněk Němec, Ph.D.**  
děkan

---

**Ing. Daniel Honc, Ph.D.**  
vedoucí katedry

## **Prohlášení**

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne

Filip Bellinger

### **Poděkování**

Děkuji vedoucímu mé bakalářské práce Ing. Liborovi Havlíčkovi Ph.D, který se mi snažil vysvětlit, co jsem nechápal a v čem jsem potřeboval popostrčit, doc. Ing. Milanovi Javůrkovi CSc. RNDr., Evě Leinerové za výborně provedenou korekturu textu. Svým kamarádům, především Bc. Jakubovi Starému za možnost využití jeho 3D tiskárny a za rady ohledně 3D tisku a Michalovi Florianovi za jeho jiný úhel pohledu a několik dobrých nápadů, jak pokračovat. Samozřejmě svým rodičům za velkou podporu, a to nejen psychickou, ale i finanční a své přítelkyni, která mi pomáhala s focením.

V Pardubicích dne

Filip Bellinger

**Anotace**

*Bakalářská práce „Servopohon se stejnosměrným motorem“ se zabývá řízením stejnosměrného motoru pro kloub robotické paže pomocí integrovaného obvodu L298N řízeného pomocí mikropočítače z rodiny ATMEGA. Snahou je dosáhnout co dobré přesnosti a rychlosti. Součástí je návrh a tisk 3D modelu kloubu.*

**Klíčová slova**

*Ovládání stejnosměrného motoru, mikropočítač ATmega, pulzně šířková modulace, 3D model*

**Title**

*ACTUATOR WITH DC MOTOR*

**Annotation**

*The bachelor thesis “Actuator with DC motor” deals with controlling a DC motor powering a robot arm using a L298N circuit with microcomputer from the ATmega family. Endeavor to achieve great accuracy. Next part is design and printing of a 3D model of robot arm.*

**Keywords**

*DC motor control, microcomputer ATmega, pulse width modulation, 3D design*

# OBSAH

	SEZNAM ILUSTRACÍ.....	9
	SEZNAM TABULEK .....	11
	SEZNAM ZKRATEK A ZNAČEK .....	12
	ÚVOD.....	14
1	Teoretická část.....	15
1.1	Stejnoseměrné motory .....	15
1.1.1	Stejnoseměrné motory s kotoučovým rotorem.....	17
1.1.2	Převodovka .....	18
1.1.3	Odměřování polohy .....	19
1.1.4	Ovládání stejnosměrných motorů.....	21
1.1.5	Servopohon.....	25
1.2	Mikropočítač.....	26
1.2.1	Architektura .....	26
1.2.2	Procesor .....	29
1.2.3	Paměť.....	31
1.2.4	Speciální paměti .....	32
1.2.5	Periferie .....	33
1.2.6	Sběrnice .....	40
1.2.7	Přerušení .....	41
1.2.8	Programovací jazyk a program.....	42
1.3	Stejnoseměrné zdroje napětí.....	44
1.3.1	Lineární zdroj napětí.....	44
1.3.2	Spínaný zdroj napětí .....	46
1.4	3D model a tisk.....	47
1.4.1	Vytvoření modelu .....	47
1.4.2	Tisk .....	49
1.5	Komerčně vyráběné servomotory.....	50
2	Praktická část.....	51
2.1	Plánování a výběr součástí.....	51
2.2	Funkce jednotlivých součástí.....	53
2.2.1	Mikropočítač ATmega.....	53
2.2.2	Driver L298N .....	57

2.2.3	Motor od výrobce POLOLU.....	58
2.3	Návrh a provedení zapojení součástek .....	59
2.4	Návrh a tisk 3D modelu a podstavy.....	66
	ZÁVĚR.....	67
	LITERATURA .....	68
	PŘÍLOHY .....	73



## SEZNAM ILUSTRACÍ

Obrázek 1.1 – Stejnoseměrný motor .....	15
Obrázek 1.2 – Komutátor.....	16
Obrázek 1.3 – Stator a rotor motoru s cizím buzením .....	17
Obrázek 1.4 – Konstrukce stejnosměrného motoru s kotoučovým rotorem (Tkotz, 2006).....	18
Obrázek 1.5 – Ozubený převod s poměrem zubů 20:40.....	19
Obrázek 1.6 – Inkrementální magnetický enkodér .....	20
Obrázek 1.7 – Princip funkce enkodérů podle jejich konstrukce (PROFESS, nedatováno) ....	21
Obrázek 1.8 – Schéma zapojení řízení motorů pomocí tyristorů .....	22
Obrázek 1.9 – Schéma zapojení řízení motorů pomocí triaků.....	22
Obrázek 1.10 – Graf průběhu řízení motorů pomocí tyristorů a triaků, motor jede vpřed.....	23
Obrázek 1.11 – Graf průběhu řízení motorů pomocí tyristorů a triaků, motor jede vzad .....	23
Obrázek 1.12 – Jednoduchý H-můstek s kontaktními spínači a bez ochran.....	24
Obrázek 1.13 – H-můstek s bezkontaktními spínači (bipolární tranzistory) a s ochranami .....	25
Obrázek 1.14 – Blokové porovnání architektury mikropočítače (Palacký, 2007).....	27
Obrázek 1.15 – Blokové schéma mikroprocesoru (Tišnovský, 2011a).....	29
Obrázek 1.16 – 4bitový asynchronní čítač vpřed (Čítače, nedatováno).....	34
Obrázek 1.17 – 4bitový synchronní čítač vpřed (Čítače, nedatováno).....	34
Obrázek 1.18 – Asynchronní 2bitový vratný čítač (Čítače, nedatováno).....	35
Obrázek 1.19 – Převodník s postupnou aproximací (D/A a A/D převodníky, nedatováno) ....	37
Obrázek 1.20 – A/D převodník s čítačem (ČVUT, nedatováno).....	37
Obrázek 1.21 – Paralelní A/D převodník (D/A a A/D převodníky, nedatováno).....	38
Obrázek 1.22 – Integrační převodník (D/A a A/D převodníky, nedatováno; Dudka, 2017).....	39
Obrázek 1.23 – Převodník s váhovou rezistorovou sítí (D/A a A/D převodníky, nedatováno).....	39
Obrázek 1.24 – Převodník s žebříčkovou sítí rezistorů (D/A a A/D převodníky, nedatováno).....	40
Obrázek 1.25 – Blokové schéma lineárního zdroje napětí (Tkotz, 2006) .....	44
Obrázek 1.26 – Zapojení jednoduchého lineárního zdroje napětí (Tkotz, 2006) .....	45
Obrázek 1.27 – Zapojené jednoduchého spínaného zdroje (Tkotz, 2006) .....	46
Obrázek 1.28 – Schéma spínaného PC zdroje s ATX standardem (Pavouk, nedatováno).....	47
Obrázek 1.29 – Ukázka slicing (Spiegel, 2018) .....	48
Obrázek 1.30 – Servomotor od firmy Kollmorgen (Pohyb, pohony, převody, nedatováno) ...	50
Obrázek 2.1 – Počítačová skříň se zdrojem ATX standardem .....	51
Obrázek 2.2 – Mikropočítač ATmega32 .....	51

Obrázek 2.3 – Driver L298N .....	52
Obrázek 2.4 – DC motor od firmy Pololu a rozdělení pólů na magnetickém disku (Metal Gearmotor 25Dx66L, nedatováno) .....	52
Obrázek 2.5 – Blokové schéma funkce výrobku .....	53
Obrázek 2.6 – Vývojový diagram programu .....	55
Obrázek 2.7 – Schéma zapojení všech součástí .....	59
Obrázek 2.8 – Schéma hlavní desky .....	61
Obrázek 2.9 – Rozvržení plošného spoje hlavní desky .....	63
Obrázek 2.10 – Schéma desky s driverem .....	64
Obrázek 2.12 – Schéma přídatné desky .....	65
Obrázek 2.13 – Rozvržení plošného spoje přídatné desky pro lepší vyvedení ze skříně .....	65
Obrázek 2.14 – Osazené a připravené desky .....	66

## **SEZNAM TABULEK**

Tabulka 1.1 – Provádění instrukcí procesorem s RISC architekturou (Olivka, 2010) .....	28
Tabulka 1.2 – Provádění instrukcí procesorem s CISC architekturou (Olivka, 2010) .....	28
Tabulka 1.3 – Tabulka vektorů přerušení pro ATmega32A (ATmega32A, 2018) .....	42
Tabulka 1.4 – Plastové filamenty (WHAT ARE 3D PRINTED MATERIALS, nedatováno).49	
Tabulka 2.1 – Tabulka vývodů a jejich funkcí (první část) .....	56
Tabulka 2.1 – Tabulka vývodů a jejich funkcí (druhá část) .....	57
Tabulka 2.2 – Možné stavy driveru L298N (STMicroelectronics, 2000) .....	58
Tabulka 2.3 – Seznam a funkce použitých součástek.....	60
Tabulka 2.4 – Zapojení vývodů konektorů na hlavní desce (Pavouk, nedatováno) .....	62

## SEZNAM ZKRATEK A ZNAČEK

μPC	Microcomputer	mikropočítač
3D	3-Dimenzional	trojdimenzionální
A/D	Analog to Digital	analogově digitální
AC	Alternating Current	střídavý proud
ALU	Arithmetic Logic Unit	aritmeticko-logická jednotka
CISC	Complex Instruction Set Computer	procesory s kompletní instrukční sadou
D/A	Digital to Analog	digitálně analogový
DC	Direct Current	stejnoseměrný proud
DRAM	Dynamic Random-Access Memory	dynamická paměť s náhodným přístupem
EEPROM	Electrically Erasable Programmable Read-Only Memory	elektronicky vymazatelná paměť pouze pro čtení
EPROM	Erasable Programmable Read-Only Memory	mazatelná programovatelná paměť
FLASH	Non-volatile computer memory storage medium	paměť s libovolným přístupem
GND	Grounded	uzemnění
I <sup>2</sup> C	Inter-Integrated Circuit	multi-masterová počítačová sériová sběrnice
IGBT	Insulated Gate Bipolar Transistor	bipolární tranzistor s izolovaným hradlem
MISO	Master Out Slave In	výstup z master a vstup do slave
MOSFET	Metal Oxide Semiconductor Field Effect Transistor	polem řízený tranzistor
MOSI	Master In Slave Out	vstup do master a výstup ze slave
NC	Not Connected	Nepřipojeno
PCI	Peripheral Component Interconnect	počítačová sběrnice pro připojení periférií
PROM	Programable Read Only Memory	programovatelná paměť pouze pro čtení
PWM	Pulse Width Modulation	pulzně šířková modulace
RAM	Random-Access Memory	paměť s náhodným přístupem

RISC	Reduced Instruction Set Computer	procesory s redukovanou instrukční sadou
ROM	Read Only Memory	paměť pouze pro čtení
SCK	Clock Signal	hodinový signál
SPI	Serial Peripheral Interface	sériové periferní komunikace
SRAM	Static random-access memory	statická paměť s náhodným přístupem
USART	Universal Synchronous And Asynchronous Receiver-Transmitter	univerzální synchronní a asynchronní sériová komunikace
USB	Universal Serial Bus	univerzální sériová sběrnice
Vcc	Power supply for logic circuit	napájení pro logický obvod
Vss	Power supply for power circuit	napájení pro výkonový obvod

# ÚVOD

V dnešní době automatizace je doslova na pořadu dne co nejvíce nahradit lidskou pracovní sílu roboty, kteří jsou přesnější a jejich pohyby efektivnější. Protože se stále zvyšují požadavky na tyto stroje, je žádoucí zamýšlet se nad dalšími možnými konstrukcemi servopohonů se stejnosměrným motorem. Tato práce se zabývá jedním z možných řešení.

Výstupem práce bude část robotické paže, resp. část kloubu, který bude schopen pohybovat akčním členem s velkou přesností i dostačující rychlostí. Síla pohybu bude záležet na použitém motoru a konstrukčním materiálu použitém při výrobě. Celé řešení bude modulární a bere v potaz možnost rozšíření na více kloubů.

# 1 TEORETICKÁ ČÁST

## 1.1 STEJNOSMĚRNÉ MOTORY

Elektromotory patří mezi elektrické točivé stroje. Motory se rozdělují hlavně podle způsobu konstrukce a napájení. Střídavé motory neboli motory s točivým polem se dále rozdělují na synchronní a asynchronní. Komutátorové motory se rozdělují na stejnosměrné, kotoučové a univerzální (Tkotz, 2006).

Stejnospměrný motor (obrázek 1.1) se skládá z komutátoru, rotoru, statoru a hřídele a může obsahovat i další součásti, například převodovku a enkodér. Stejnospměrný motor převádí stejnosměrné napětí na točivý pohyb. Tyto motory se rozdělují na motory s vlastním a cizím buzením, což určuje konstrukci statoru. Elektromotor, který může být použit i jako generátor stejnosměrného napětí otáčením hřídele motoru a následným odebíráním napětí na svorkách napájecích kabelů, se nazývá dynamo (Tkotz, 2006).

Důležité parametry u stejnosměrných motorů jsou: výkon (W), napájecí napětí (V), otáčky za minutu (ot./min), moment síly (kg·m), doporučený pracovní proud vinutím motoru (A), maximální proud vinutím motoru (A), velikost a tvar výstupní hřídele a další (Tkotz, 2006; Metal Gearmotor 25Dx66L, nedatováno).

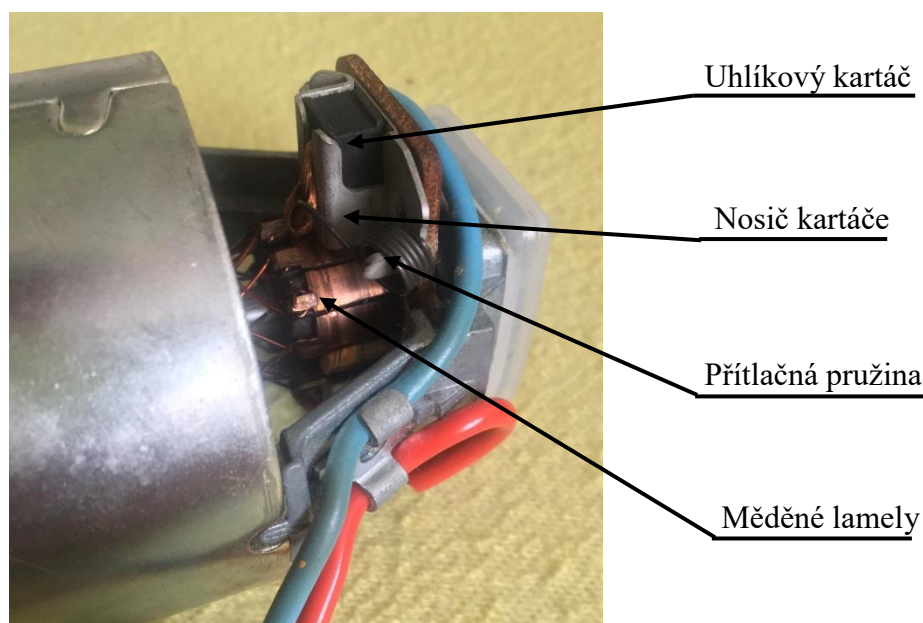


Obrázek 1.1 – Stejnospměrný motor

## Komutátor

Komutátor je jedna z rotujících částí stejnosměrného motoru. Funkcí komutátoru je zajištění přívodu a přepínání směru proudu do cívek navinutých na kotvě tak, aby se motor točil do určitého směru.

Komutátor (obrázek 1.2) je složený z lamel z tvrdé měděné slitiny ve tvaru klínů poskládaných vedle sebe do tvaru prstence. Lamely jsou elektricky navzájem odizolovány. Na ně doléhají kartáče, které jsou uloženy v nosiči a jsou stále přitlačovány k měděným lamelám. Na tyto kartáče se přivádí napájecí napětí (v případě dynama se zde odebírá napětí). Měděné lamely jsou spojeny s vinutím cívek, které jsou navinuty na rotoru (Tkotz, 2006; Komutátory, nedatováno).

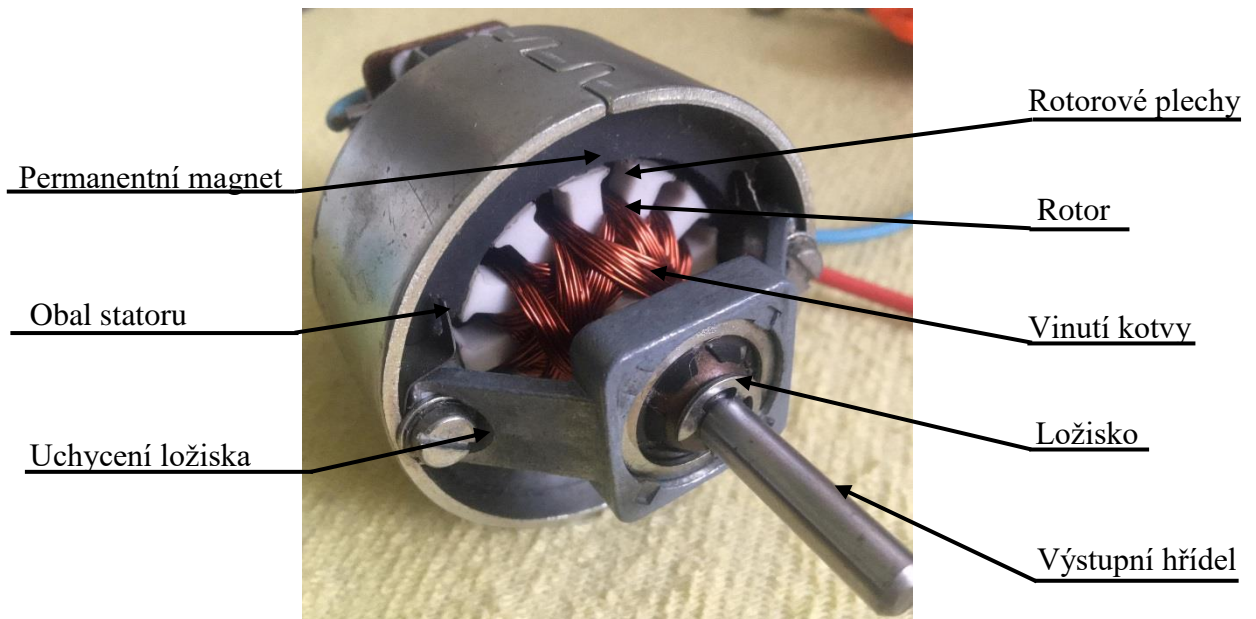


Obrázek 1.2 – Komutátor

## Stator

Stator (obrázek 1.3) je statická část stejnosměrného motoru. Až na speciální případy nemění své magnetické pole. Bývá tvořen plným kovovým prstencovým jádrem, ke kterému jsou připevněny trvalé magnety nebo cívky statoru (pro malé výkony se používají spíše trvalé magnety) (Tkotz, 2006).





Obrázek 1.3 – Stator a rotor motoru s cizím

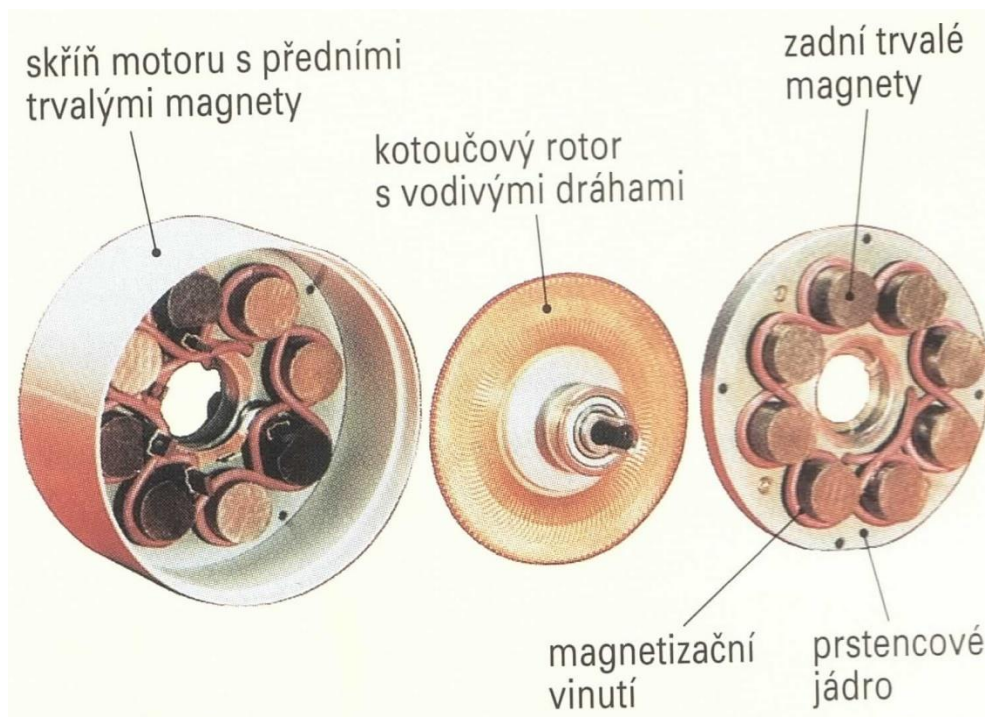
## **Rotor**

Rotor (obrázek 1.3) neboli kotva je druhá rotující část stejnosměrného motoru, která je napravo připojená ke komutátoru a hřídeli motoru. Jádro je složeno z tenkých plechů lepených k sobě tak, aby byly od sebe izolovány a netvořily se v nich vířivé proudy. V těchto plechách jsou vyfrézované drážky pro vinutí cívek, které jsou napájeny z komutátoru (Tkotz, 2006).

### **1.1.1 Stejnosměrné motory s kotoučovým rotorem**

Výhodou DC motorů s kotoučovým rotorem je, že se velmi rychle rozbíhají a zastavují, protože obsahují rotor bez kovového jádra. Proto mají malý setrvačný moment. Většina má cizí buzení. Výkon těchto motorů se pohybuje od jednotek wattů do jednotek kilowattů (Tkotz, 2006).

Rotor tohoto motoru (obrázek 1.4) je vyroben z plastového disku pokrytého tenkou vrstvou mědi, který má v sobě vyfrézované drážky tvořící vinutí rotoru. Na něj se kartáči přenáší napájecí napětí komutátoru (Tkotz, 2006).



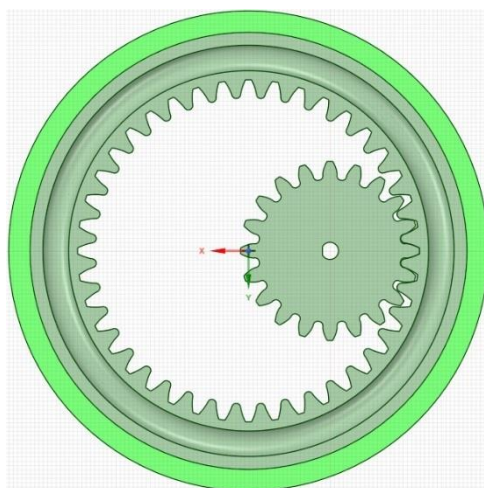
Obrázek 1.4 – Konstrukce stejnosměrného motoru s kotoučovým rotorem (Tkotz, 2006)

### 1.1.2 Převodovka

Existuje mnoho různých typů převodovek, avšak u stejnosměrných motorů se používají zejména převodovky s ozubenými koly a pevným převodem. Nejvíce používané jsou převodovky tvořené dvěma či více ozubenými koly napojenými na sebe s různým počtem zubů nebo planetové převodovky (MALÉ STEJNOSMĚRNÉ MOTORY MAXON, 2002).

Každá převodovka má daný převodový poměr, což je hodnota, která udává poměr otáček na jejím vstupu a výstupu. Jako příklad jsem vybral převodovku, která je součástí motoru použitého v praktické části méj bakalářské práce. Motor je převodovaný do síly a má šest převodů z kovových ozubených kol řazených za sebou. První pár ozubených kol má poměr zubů 22 : 12, další 4 páry 22 : 10 a poslední pár 23 : 10. Konečný převodový poměr motoru je 98,778 : 1 (zaokrouhleno na tisíciný) (MALÉ STEJNOSMĚRNÉ MOTORY MAXON, 2002).

$$\text{Převodový poměr} = \frac{22 * 22 * 22 * 22 * 22 * 23}{12 * 10 * 10 * 10 * 10 * 10} = 98,77794\bar{6}: 1 \quad (1. 1)$$



Obrázek 1.5 – Ozubený převod s poměrem zubů 20:40

### 1.1.3 Odměřování polohy

Pro odměřování polohy se nejčastěji používá enkodér, který měří délku posuvu nebo úhel natočení. Enkodéry můžeme rozdělit na inkrementální a absolutní, nebo podle způsobu odměřování na optické a magnetické. Enkodér má určité rozlišení, které je udáváno počtem pulzů na jednu otáčku motoru. U motorů se setkáme spíše s enkodéry úhlu natočení. Některé motory se už prodávají s vestavěným enkodérem (Absolutní a inkrementální enkodéry, nedatováno; PROFESS, nedatováno).

#### Absolutní enkodér

Absolutní enkodér udává polohu, ve které se motor nachází. Při odměřování úhlu natočení bude výstupem z enkodéru zakódovaná informace o poloze hřídele motoru. Absolutní enkodér je tvořen disky s otvory nebo reflexními ploškami, zdrojem světla a optickým snímačem. Výstup z enkodéru pak bude zapsán v binárním kódu, například 100011. Enkodéry s absolutním odměřováním využívají Greyův kód pro zmenšení chybovosti čtení. Výhodou absolutních enkodérů je, že nepotřebují prvotní kalibrační otáčku či natočení (Absolutní a inkrementální enkodér, nedatováno).

Princip Greyova kódu spočívá v tom, že při otočení disku do další polohy se výstup změní pouze v jednom bitu (Absolutní a inkrementální enkodéry, nedatováno)

Počítejme s výstupem 100011 z enkodéru s absolutním odměřováním, který pokrývá celý kruh. Znamená to, že enkodér má  $2^6$  kombinací (64 kombinací), to je rozlišení 64 impulzů na jednu otáčku. Jeden pulz představuje 5,625 stupně (Absolutní a inkrementální enkodéry, nedatováno).

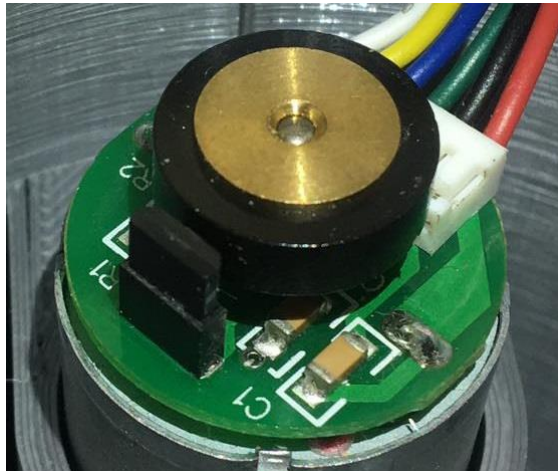
## **Inkrementální enkodér**

Inkrementální enkodér neudává svou polohu, ale její změnu. Takže pokud se hřídel, na které je enkodér, pootočí, bude okamžitě vyslán signál o změně polohy. Chceme-li odměřovat i na jakou stranu se točí, potřebujeme signál o poloze dvakrát a aby byl jeden z nich o 90 stupňů opožděný za druhým. Pak stačí jen sledovat, který pulz předbíhá a který se opoždí, a vyčíst z toho směr otáčení (Absolutní a inkrementální enkodéry, nedatováno).

## **Konstrukce enkodérů**

V praxi se nejčastěji používají enkodéry magnetické, indukční a optické (reflexní, okénkové). Všechny tyto enkodéry jsou bezkontaktní.

Magnetické enkodéry se vyrábějí jako inkrementální (obrázek 1.6). Fungují tak, že k hřídeli je připevněný magnet, nejčastěji neodymový prstěnek, který má po svém obvodu několik pólů. Vně disku jsou umístěny dvě Hallovy sondy. Ty snímají magnetické pole a pokaždé když se změní pól magnetu u sondy, změní se logický stav na výstupu. Sondy musí být od sebe vzdáleny tak, aby jejich výstupy byly posunuty o 90 stupňů (PROFESS, nedatováno).

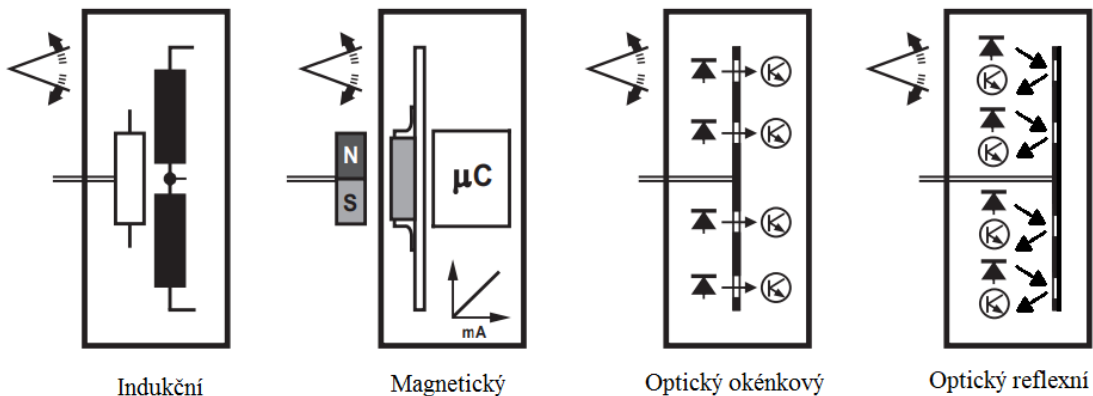


Obrázek 1.6 – Inkrementální magnetický enkodér

Indukční snímače úhlového natočení se používají na místech vystavených velkým vibracím nebo rázům a také v agresivním prostředí. Nejčastěji se s nimi setkáme při měření v energetických nebo chemických zařízeních (PROFESS, nedatováno).

Okénkové enkodéry se vyrábějí v provedení inkrementálním i absolutním. Na hřídeli je připevněn děrovaný disk a z jedné strany disku je umístěn zdroj světla (ne nutně okem viditelného) a z druhé optický snímač. Pulzy na výstupu závisejí na poloze okének ke zdroji

světla a optickému snímači. V případě absolutního snímače je na disku více drah okének a více zdrojů světla i optických snímačů (PROFESS, nedatováno).



Obrázek 1.7 – Princip funkce enkodérů podle jejich konstrukce

Reflexní enkodéry jsou konstrukcí i funkcí velmi podobné okénkovým enkodérům, pouze na disku jsou místo okének reflexní plošky. Zdroj světla i optický snímač jsou umístěny na jedné straně disku, oba otočené na jedno místo dráhy reflexních plošek. Pulzy na výstupu závisí na poloze reflexních plošek ke zdroji světla a optickému snímači (PROFESS, nedatováno).

#### 1.1.4 Ovládání stejnosměrných motorů

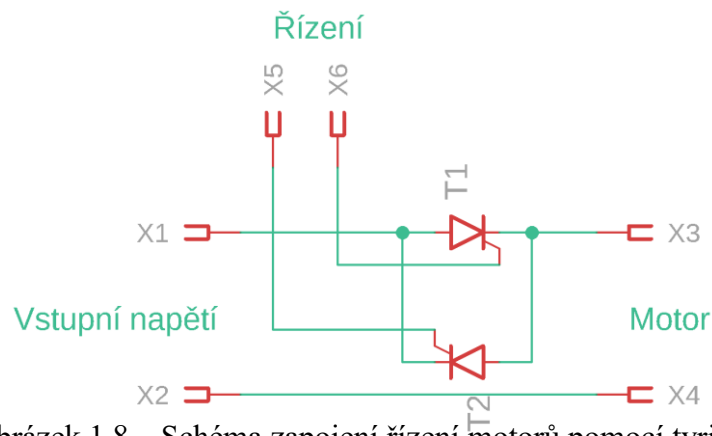
Pro ovládání motorů se používá obvod nebo součástka často nazývaná jako driver. Driver zajišťuje několik funkcí. U stejnosměrných motorů reguluje otáčky, procházející proud, chrání proti přepětí apod. Jedním z nejpoužívanějších obvodů je H-můstek, méně používané jsou například tyristory a triaky (DC Servomotor Controller, 2001; Kubie, 2014).

#### Řízení pomocí tyristorů a triaků

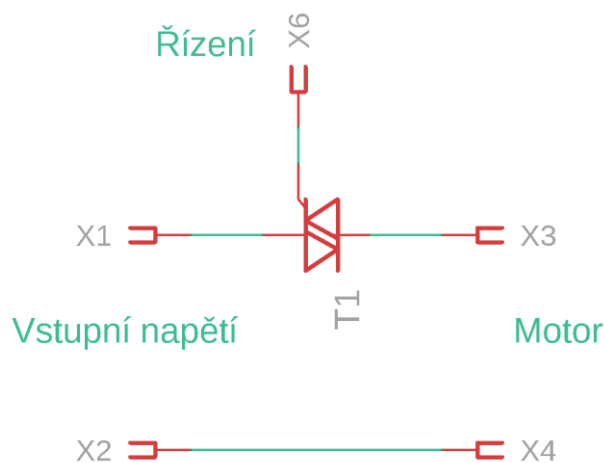
Tyristory a triaky jsou polovodičové prvky, kterými se spíná a vypíná elektrický proud. Řízení pomocí tyristorů nebo triaků probíhá velmi podobně, ale jeden triak dokáže nahradit dva tyristory (obrázek 1.8 a 1.9). Pokud chceme otáčet motorem oběma směry, zvolíme střídavé vstupní napětí. Výstupy jsou identické. Liší se jen řídicí signály: u tyristorů jsou dva, u triaků jeden (Kubie, 2014).

Pokud chceme při řízení tyristory otáčet motorem dopředu, budeme otevírat tyristor T1 při kladné půlvlně (obrázek 1.11), pokud chceme, aby se otáčel motor dozadu, budeme otevírat tyristor T2 při záporné půlvlně (obrázek 1.10). Výkon můžeme ovládat tím, že sepneme tyristor či triak v dané půlvlně. Na grafech jsem zvolil 50% výkon jednotlivých půlvln (Kubie, 2014).

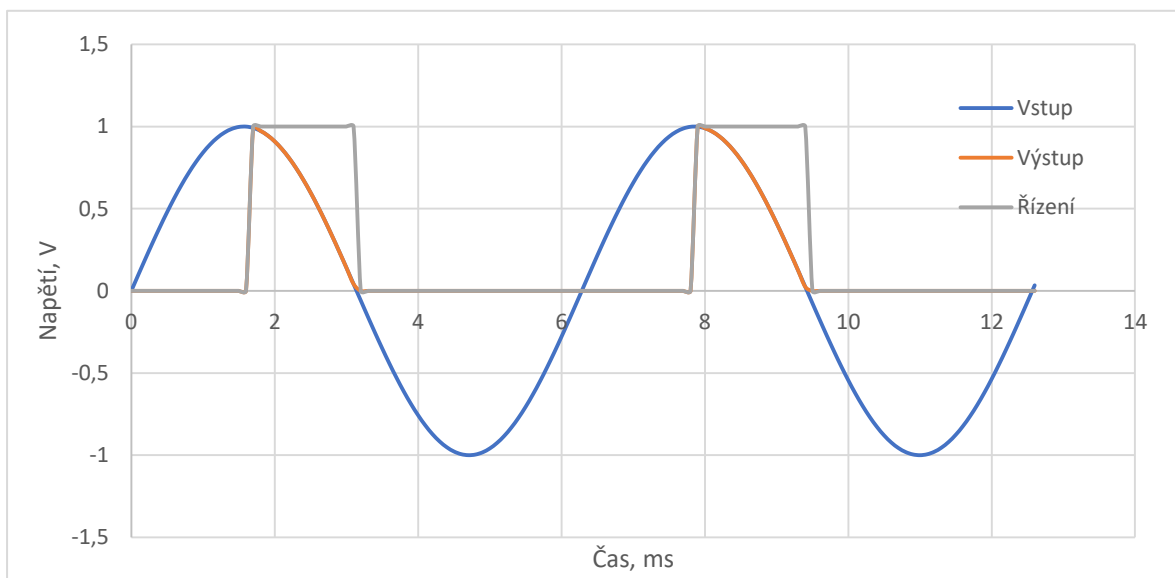
Při ovládnání triaky stačí jeden řídicí vstup. Pokud na něj přivedeme kladné napětí v kladné půlčlenně, triak se otevře a motor se bude točit vpřed. Pokud v záporné půlčlenně přivedeme na řídicí vstup záporné napětí, triak se otevře a motor se bude točit vzad (Kubie, 2014).



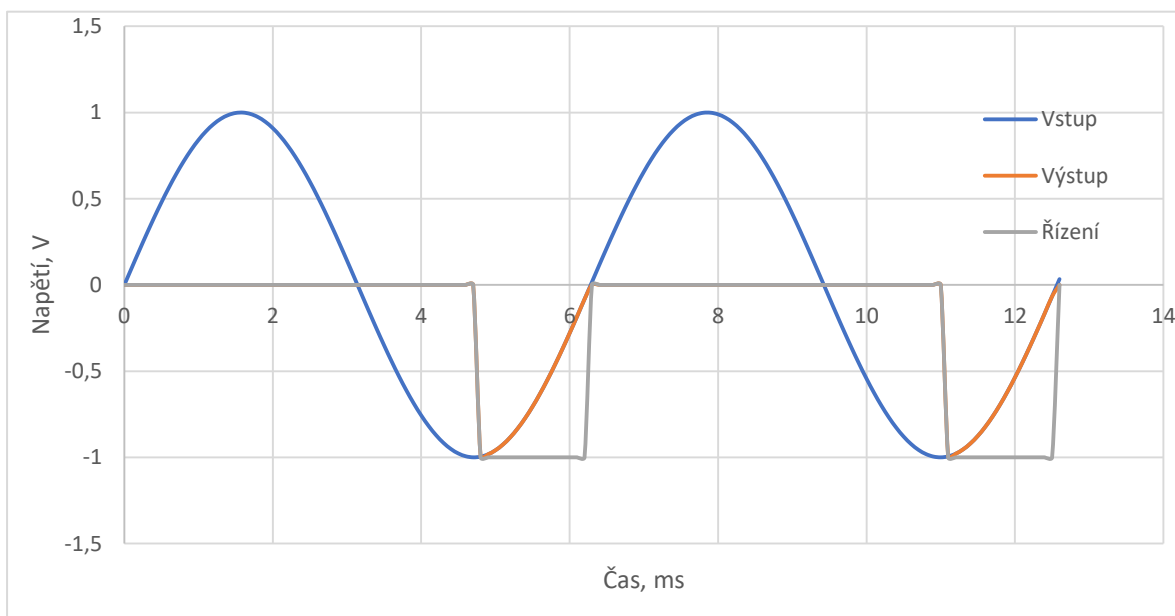
Obrázek 1.8 – Schéma zapojení řízení motorů pomocí tyristorů



Obrázek 1.9 – Schéma zapojení řízení motorů pomocí triaků



Obrázek 1.11 – Graf průběhu řízení motorů pomocí tyristorů a triaků, motor jede vpřed



Obrázek 1.10 – Graf průběhu řízení motorů pomocí tyristorů a triaků, motor jede vzad

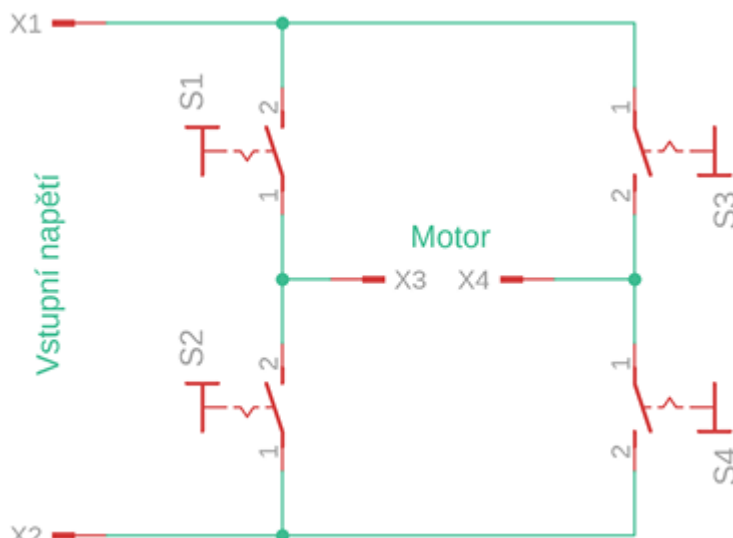
### H-můstek

H-můstek zajišťuje přepínání polarity napájecího napětí či odpojení přivedeného napětí na zátěž (motor). Je složen ze čtyř spínacích prvků a zátěže. Takové zapojení zajišťuje při připojení na stejnosměrný motor možnost změny směru otáčení nebo aktivní brzdu motoru a také umožňuje rozdělit obvod na řídicí a výkonovou část. Spínače mohou být kontaktní nebo bezkontaktní (DC Servomotor Controller, 2001; STMicroelectronics, 2000).

Jaký typ kontaktních spínačů použijeme, závisí na velikosti proudu do motoru. Pro větší proudy použijeme spíše stykač, pro menší proudy relé. Kontaktní spínače jsou pomalejší.

Jako bezkontaktní spínače se nejčastěji používají tranzistory, méně často tyristory. U menších proudů do 3 A se nejčastěji setkáme s bipolárními tranzistory, u proudů od 3 do 10 A a u náročnějších operací s MOSFET tranzistory a u proudů nad 10 A s IGBT tranzistory. Velkou výhodou bezkontaktních spínačů je rychlost spínání, která je v řádech MHz (DC Servomotor Controller, 2001; STMicroelectronics, 2000).

Pro jednodušší představení zapojení spínačů jsem v programu EAGLE nakreslil jednoduché schéma (obrázek 1.12).



Obrázek 1.12 – Jednoduchý H-můstek s kontaktními spínači a bez ochran

### Ochrana H-můstku

H-můstek musíme chránit proti zpětnému indukovanému napětí, protože při zpomalování nebo zrychlování motor klade odpor změně elektrického pole v cívkách. To se řeší přidáním rychlých Schottkyho diod v závěrném směru přes jednotlivé spínací prvky. Některé H-můstky už mají integrovanou ochranu, ale ne všechny (TIPY&triky 01/12, 2012).

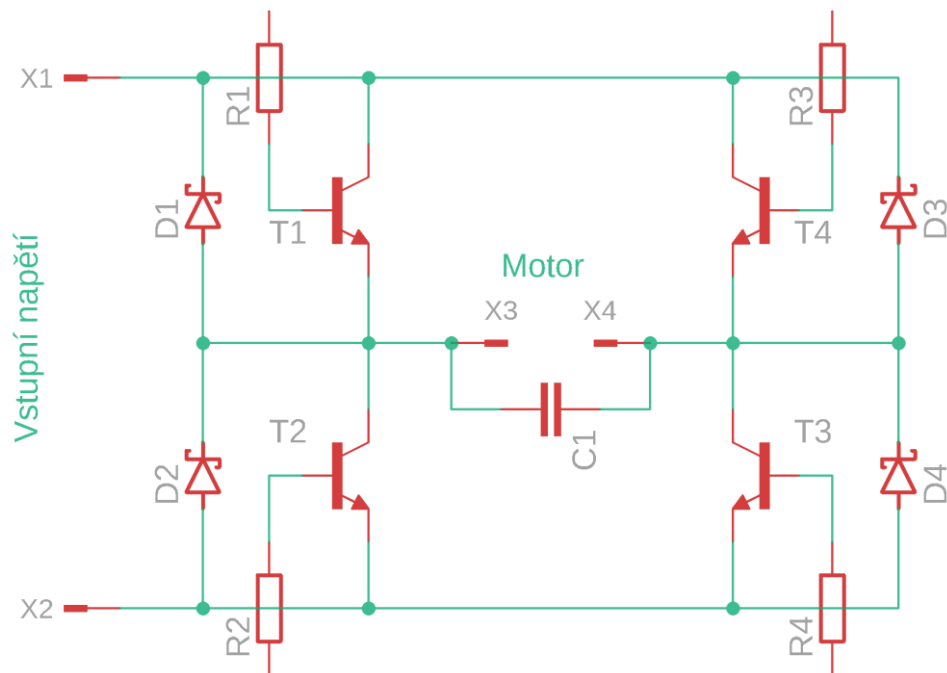
V katalogových listech se uvádí maximální možný výstupní proud například u integrovaného obvodu L298, který používám v praktické části, 2,5 A. Ovšem podle doporučení není vhodné překračovat proud 2 A (TIPY&triky 01/12, 2012).

Obvod je také nutné chránit proti přehřátí. To se realizuje buďto pasivním nebo aktivním chlazením (TIPY&triky 01/12, 2012).

H-můstek také chráníme proti elektrickému šumu přidáním kondenzátorů, které připojíme co nejbližší k vývodům motoru (TIPY&triky 01/12, 2012).



Další ochranou (v případě bezkontaktních spínačů) je připojení rezistorů, které chrání před velkými proudy do bází jednotlivých tranzistorů (TIPY&triky 01/12, 2012).



Obrázek 1.13 – H-můstek s bezkontaktními spínači (bipolární tranzistory) a s ochranami

### 1.1.5 Servopohon

Skládá se nejčastěji ze stejnosměrného elektromotoru (může být použit i krokový motor či stejnosměrný motor) s kotoučovým rotorem, z ovládání a napájení motoru a regulátoru otáček nebo polohy. Často se používají ještě snímače rychlosti a polohy otočení. Součástí regulátoru otáček nebo polohy je vždy zpětná vazba (Tkotz, 2006).

Regulací se myslí proces, při kterém je v reálném čase porovnávána požadovaná hodnota se skutečnou, kdy jejich rozdíl ovlivňuje výstup z regulátoru. To je řešeno uzavřenou zpětnou vazbou (Skalický, nedatováno; Tkotz, 2006).

U servopohonu existují dva typy polohových regulací, a to cílová a sledovací. Cílová regulace slouží k nastavování polohy např. u vrtaček, kdy je potřeba vypnout motor po dojezdu před cílovou polohu a až pak dojet do cílové polohy, aby se nezničil nástroj nebo obrobek. Sledovací polohová regulace se používá tam, kde je potřeba projet určitou trasou (Skalický, nedatováno).

Rychlostní neboli otáčkový servopohon dokáže velmi spolehlivě a přesně sledovat rychlost otáčení, ale neumí sledovat úhel natočení motoru (Skalický, nedatováno)

Polohový servopohon sleduje úhel natočení motoru nebo pomocí převodu vzdálenost na dráze (Skalický, nedatováno).

## 1.2 MIKROPOČÍTAČ

Mikropočítač tvoří několik integrovaných obvodů v jednom malém pouzdře, ve kterém najdeme procesor, paměť, periferie a sběrnice. Všechny tyto komponenty jsou jeho nedílnou součástí, bez kterých by nefungoval nebo nebyl použitelný podobně jako běžný stolní počítač (Jednočipové mikropočítače, nedatováno; Žoltá, nedatováno).

Mikropočítače jsou vyráběny hlavně za účelem menších aplikací, kde není potřeba velkého výpočetního výkonu. Typické aplikace pro mikropočítače v dnešní době jsou například u vstupních a výstupních zařízení (myš, klávesnice), zpracovávání signálů v inteligentních senzorech nebo při řízení motorů v automatizaci (Jednočipové mikropočítače, nedatováno; Žoltá, nedatováno).

### 1.2.1 Architektura

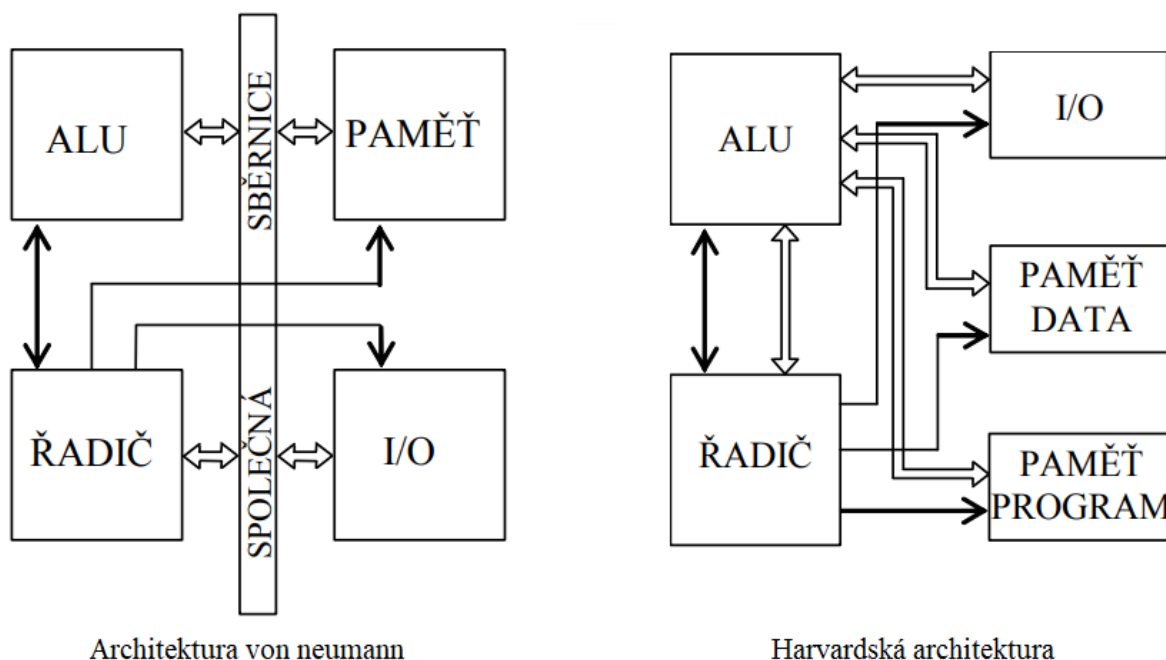
Architektura počítače udává konstrukci mikropočítače, způsob zpracovávání instrukcí, ukládání dat a programu. Vychází ze dvou koncepcí: harvardské a Von Neumannovské (obrázek 1.14) (Architektura počítačů a mikropočítačů, 2002).

#### Harvardská versus Von Neumannovská architektura

Harvardská architektura má rozdělenou paměť pro program a pro data, což znemožňuje přepis dat programem a naopak. Často se používá v mikropočítačích (Architektura počítačů a mikropočítačů, 2002; Palacký, 2007).

Von Neumannovská architektura se skládá z operační paměti, řadiče, aritmeticko-logické jednotky (ALU) a vstupně-výstupních periférií. To znamená, že má spojenou paměť pro program i data (Architektura počítačů a mikropočítačů, 2002; Palacký, 2007).

Princip činnosti: Jako první se pomocí ALU uloží program, se kterým se bude dále pracovat, do paměti RAM. Poté se stejným způsobem do paměti RAM uloží i data. Když jsou data uložena, proběhne jejich zpracování pomocí ALU, která je řízená řadičem a posílá mezivýsledky do paměti RAM. Po ukončení výpočtů ALU čte a odesílá výsledky na místo určení např. paměť nebo výstupní periferie (Architektura počítačů a mikropočítačů, 2002; Palacký, 2007).



Obrázek 1.14 – Blokové porovnání architektur mikro počítače (Palacký, 2007)

### RISC versus CISC architektura

RISC (Reduced Instruction Set Computer) je počítač s redukováným souborem instrukcí. U těchto počítačů je snaha převést co nejvíce činností mimo procesor a přenechat je programovému vybavení a kompilátoru (Architektura počítačů a mikro počítačů, 2002; Palacký, 2007).

RISC procesory využívají program flow control. Ten dokáže všechny kroky zpracování instrukce dělat najednou: v prvním časovém signálu vybrat instrukci, při druhém časovém signálu dekodovat první instrukci a vybírat druhou instrukci a tak dále (tabulka 1.2). Na ukládání a čtení dat využívá jen dvě instrukce: LOAD a STORE (Architektura počítačů a mikro počítačů, 2002; Palacký, 2007).

Výhodou je velké zrychlení díky flow control a malý instrukční soubor. Instrukce mají pevnou délku, počet instrukcí a způsobů adresování je malý. Mezi nevýhody patří zvětšení programu z důvodu omezeného počtu instrukcí, potřeba více pracovních registrů a větší riziko problémů typu hazardů nebo zřetěženého zpracování (Architektura počítačů a mikro počítačů, 2002; Palacký, 2007).

Tabulka 1.1 – Provádění instrukcí procesorem s RISC architekturou (Olivka, 2010)

Krok	Význam	Hodinový pulz											
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
1.	Výběr instrukce	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
2.	Dekódování		I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11
3.	Výpočet adresy			I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
4.	Výběr operandu				I1	I2	I3	I4	I5	I6	I7	I8	I9
5.	Provedení instrukce					I1	I2	I3	I4	I5	I6	I7	I8
6.	Uložení výsledku						I1	I2	I3	I4	I5	I6	I7

CISC (Complex Instruction Set Computer) je počítač se složitým souborem instrukcí. To znamená, že má v instrukční sadě takové instrukce, které by se daly nahradit jednoduššími (násobení opakovaným sčítáním a bitovým posunem). (Palacký, 2007; Žoltá, nedatováno)

Procesory CISC nepodporují flow control, takže vždy zpracují nejprve jednu instrukci a až poté začnou zpracovávat další (tabulka 1.3) (Architektura počítačů a mikropočítačů, 2002; Palacký, 2007).

Výhodou je jednodušší program a potřeba méně registrů, větší odolnost proti problémům se zřetěženým zpracováním. Nevýhodou je proměnlivá délka instrukcí (Architektura počítačů a mikropočítačů, 2002; Palacký, 2007).

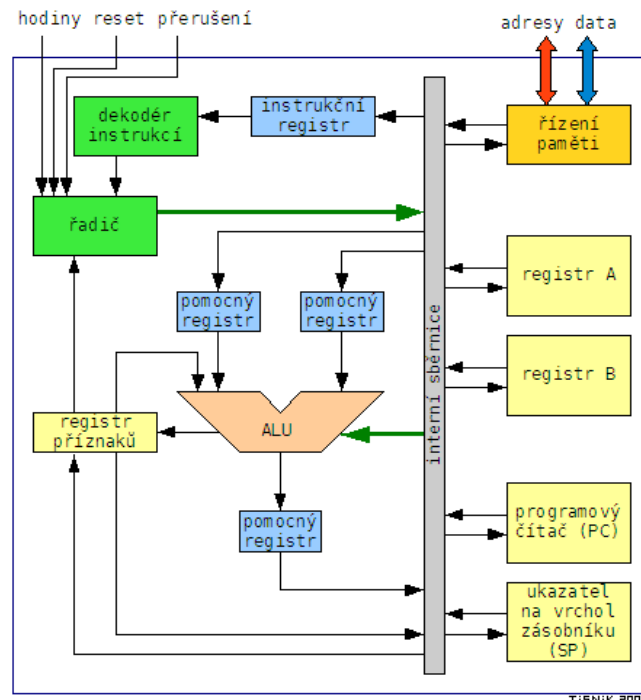
Tabulka 1.2 – Provádění instrukcí procesorem s CISC architekturou (Olivka, 2010)

Krok	Význam	Hodinový pulz											
		T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
1.	Výběr instrukce	I1						I2					
2.	Dekódování		I1						I2				
3.	Výpočet adresy			I1						I2			
4.	Výběr operandu				I1						I2		
5.	Provedení instrukce					I1						I2	
6.	Uložení výsledku						I1						I2

V dnešní době se nejčastěji používá kombinace RISC a CISC, která se více přiklání k RISC (Olivka, 2010)

## 1.2.2 Processor

Procesor je složený z aritmeticko-logické jednotky, řadiče instrukcí a programového čítače (obrázek 1.15). Umí zpracovávat bitové slovo, které má pevně danou délku, záleží na verzi procesoru, která je vždy násobkem 2. Setkat se můžeme s 4, 8, 16, 32 nebo 64bitovými verzemi procesoru. 4bitové verze už najdeme jen velmi zřídka a je to spíše historická záležitost. 8 a 16bitové verze najdeme v mikropočítačích. 32bitové a dnes už spíše 64bitové procesory se používají v osobních počítačích nebo laptotech (Mikropočítače, nedatováno).



Obrázek 1.15 – Blokové schéma mikroprocesoru (Tišnovský, 2011a)

### Aritmeticko-logická jednotka

Probíhají v ní jednoduché výpočty s vysokou rychlostí. ALU umí pouze logické a jednoduché aritmetické operace. Zpracovává dvě nebo jedno bitové slovo a výsledkem je jedno bitové slovo. Dokáže sčítat, ale nedokáže odčítat, avšak díky dvojkovému doplňku můžeme odčítání převést na sčítání (Palacký, 2007)

Instrukce pro zpracování jednoho slova jsou například negace nebo bitový posun a pro dvě slova je sčítání, inkrementace nebo porovnávání (Palacký, 2007).

## **Řadič instrukcí**

Instrukce je elementární operace pro procesor zakódovaná v instrukčním kódu. Skládá se z operačního kódu a operandu, kdy operační kód udává, jaká operace se bude provádět (sčítání, porovnávání atd.), a operand udává, s jakými daty se bude pracovat (číslo, odkaz do paměti/registru) (Tišnovský, 2011).

Řadič instrukcí určuje, jakým způsobem a jaká data se budou zpracovávat. Předchází mu dekodér instrukcí, který načítá instrukce z paměti nebo instrukčního registru. Každý řadič je ve skutečnosti konečným automatem, který se může nacházet vždy v jednom z konečného počtu stavů. Do řadiče je zaveden i hodinový signál z důvodu synchronizace. Klasické řadiče jsou realizované jako sekvenční obvod nebo jako mikroprogramový řadič (Tišnovský, 2011).

V klasickém řadiči se sekvenčním obvodem a stavovým automatem jsou jednotlivé stavy vždy binárně zakódovány do několika desítek až stovek bitů. Průběh instrukce začíná vysláním signálu z programového čítače, který zaktivuje řízení paměti, jež vystaví kód instrukce. Tento kód je přes instrukční registr dekodován v dekodéru instrukcí, poté je rozdělen na operační kód a operand a dále odeslán do řadiče. V klasickém čítači můžeme někdy narazit na řídicí registry, ve kterých jsou zakódované řídicí signály v kódu 1 z  $N$ , a pomocí multiplexoru se vybrané signály odesílají (Tišnovský, 2011a).

Mikroprogramový řadič se používá hlavně u složitějších mikropočítačů s větší instrukční řadou. Tento řadič má každou instrukci zakódovanou v několika jednodušších mikroinstrukcích. Mezi nejdůležitější části patří řídicí paměť a mikroprogramový čítač. V řídicí paměti jsou uloženy jednotlivé mikroinstrukce a mikroprogramový čítač obsahuje adresu další mikroinstrukce. Operační znak instrukce udává adresu první mikroinstrukce v řídicí paměti mikroinstrukcí (Tišnovský, 2011b).

## **Programový čítač**

Programový čítač, nazývaný také instrukční čítač nebo instrukční ukazatel, má za úkol adresovat instrukci v operační paměti počítače. Po přijetí většiny instrukcí dojde pouze k inkrementaci čítače pomocí ALU, která načte obsah programového čítače a přidá k němu jedničku, avšak v programu mohou být i instrukce nebo přerušení, které požadují skok na jinou část programu, v tomto případě se hodnota čítače přepíše na požadovanou hodnotu (Palacký, 2007).

### **1.2.3 Paměť**

Paměť slouží k uchování dat a programů. Dnes se používají spíše EEPROM, FLASH a RAM paměti, ale v menších zařízeních se můžeme setkat i s PROM pamětí, např. v mikrovlnné troubě nebo kalkulačce, kde není potřeba program měnit (Mikropočítače, nedatováno; Tkotz, 2006).

#### **ROM**

ROM (Read Only Memory) je elektronická paměť, která se naprogramuje přímo při výrobě, nelze v ní žádná data nebo program měnit, ale lze je číst. Na této paměti zůstávají všechna data i po odpojení napájení (Černá, 2012; Tkotz, 2006).

#### **PROM**

PROM (Programmable Read Only Memory) je paměť, kterou lze po výrobě naprogramovat, ale nelze na ní nic mazat ani přepisovat. Poté, co se paměť naprogramuje, už z ní lze jen číst. Na této paměti zůstávají všechna data i po odpojení napájení (Černá, 2012; Tkotz, 2006).

#### **EPROM**

EPROM (Erasable Programmable Read Only Memory) je paměť, kterou lze programovat, číst i mazat. Mazat se dá jen ultrafialovým zářením. Na této paměti zůstávají všechna data i po odpojení napájení (Černá, 2012; Tkotz, 2006).

#### **EEPROM**

EEPROM (Electrically Erasable Programmable Read Only Memory) je paměť, kterou lze programovat, číst i mazat. Vydrží více mazacích a programovacích cyklů než paměť FLASH, ale má složitější paměťové buňky, kvůli čemuž nelze EEPROM paměť vyrobit o tak malých rozměrech a velké kapacitě jako FLASH. Proto se používá u údajů, které nejsou moc objemné, ale se často mění. Paměť se maže elektronicky. Na této paměti zůstávají všechna data i po odpojení napájení (Černá, 2012; Tkotz, 2006).

## **FLASH**

FLASH paměť je elektronicky programovatelná i mazatelná. Vnitřek paměti je oproti ROM paměti organizován po blocích, což umožňuje zápis i čtení v jednotlivých blocích zároveň. Tuto paměť je možné vyrobit s velkou kapacitou a malými rozměry. Na této paměti zůstávají všechna data i po odpojení napájení (Černá, 2012; Tkotz, 2006).

## **RAM**

RAM (Random Access Memory) je paměť s náhodným přístupem. Dělí se na SRAM, DRAM, SDRAM a DDR SDRAM, ale se SDRAM a DDR SDRAM pamětí se setkáme spíše v osobních počítačích. RAM paměti jsou oproti všem ROM pamětem o poznání rychlejší, ale nezůstávají na nich data po odpojení napájení. Po odpojení od napájení a opětovném připojení je paměť prázdná (Černá, 2012; Tkotz, 2006).

## **SRAM**

SRAM paměť (statická RAM), která se používá spíše u cache pamětí, najdeme spíše v osobních počítačích. Každá paměťová buňka je realizována bistabilním obvodem 6 tranzistorů (Paměti počítače, nedatováno; Tkotz, 2006).

## **DRAM**

DRAM paměť (dynamická RAM) má složenou každou paměťovou buňku jen z jednoho tranzistoru a využívá parazitních kapacit. Jejich nevýhodou je, že se musí pravidelně obnovovat (Paměti počítače, nedatováno; Tkotz, 2006).

### **1.2.4 Speciální paměti**

#### **Zásobník**

Má datovou strukturu LIFO (Last In – First Out) a řídí se pouze dvěma příkazy: PUSH a POP. Příkazem PUSH uložíme hodnotu na vršek zásobníku a příkazem POP odebereme a načteme hodnotu z vršku zásobníku. Zásobník se ukládá na paměť RAM, nejčastěji na konec. Do zásobníků se většinou ukládají data návratových adres při volání podprogramů, procedur nebo funkcí (*Jednočipové mikropočítače – a, nedatováno*).



## **Registr**

Registrů najdeme v mikropočítači mnoho a můžeme je rozdělit na univerzální, příznakové, pomocné a další (*Jednočipové mikropočítače – a, nedatováno*).

Univerzální registry slouží k ukládání dat pro aritmetické operace a výsledků. Například mikroprocesory AVR mají sadu 32 8bitových registrů, kdy šest z nich může být použito jako tři 16bitové registry (*Jednočipové mikropočítače – a, nedatováno*).

Příznakový registr slouží k ukládání stavu procesoru. Například v něm najdeme, zdali je povoleno globální přerušování, zdali byl poslední výsledek nulový, zdali došlo k přenosu z vyššího řádu a další příznaky (*Jednočipové mikropočítače – a, nedatováno*).

Pomocný registr nejčastěji slouží ke chvilkovému uchování dat. Například když ALU dopočítá aritmetickou instrukci, předá výsledek pomocnému registru na svém výstupu, který jej podrží a pošle dál, když je to možné (*Jednočipové mikropočítače – a, nedatováno*).

### **1.2.5 Periferie**

Periferie jsou obvody zajišťující komunikaci mikropočítače s okolím.

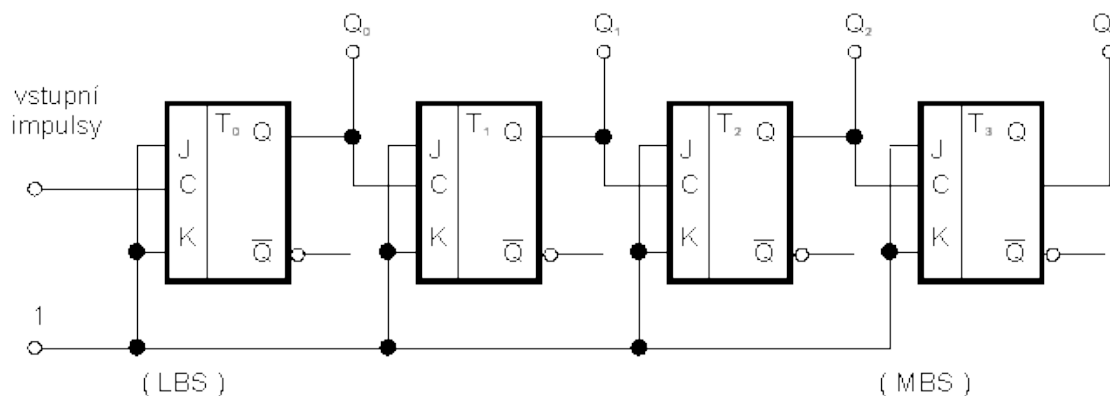
#### **Vstupní a výstupní brány**

Často se označují jako port. Jsou tvořeny skupinou vývodů z mikropočítače v počtu násobku 2, nejčastěji po osmi nebo po čtyřech, a každý vývod signalizuje jeden bit. Na jednotlivé vývody lze současně zapisovat logické hodnoty 0 a 1 a číst je. U bran lze nastavit, který bit bude sloužit jako vstupní a který jako výstupní. Na vstupu každého bitu je Schmittův klopný obvod. U většiny mikropočítačů jsou brány implementovány tak, že s nimi instrukční soubor může pracovat jako s jednotlivými bity nebo jako s množinou vývodu (*Žoltá, nedatováno*).

#### **Čítače a časovače**

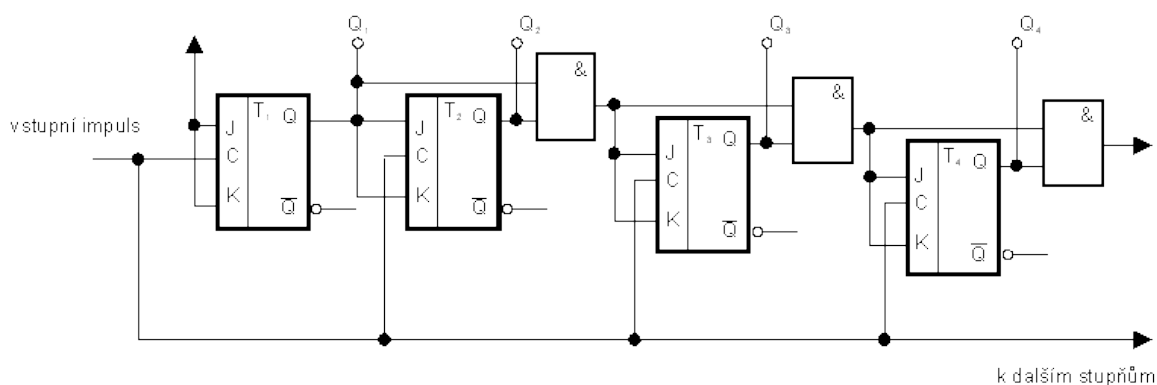
Hlavní funkcí čítače je určení počtu pulzů, které se objevily na vstupu. Čítače mohou být asynchronní a synchronní a skládají se z klopných obvodů JK. Tyto JK obvody jsou zapojeny za sebou a tvoří řetězec. Při každém vstupním pulzu do čítače dojde k inkrementaci nebo dekrementaci výsledku. Výsledek čítače je určen výstupy jednotlivých klopných obvodů, kdy první je vždy ten, který určuje bit s nejnižší vahou. Při dosažení maximální hodnoty se dalším pulzem čítač vynuluje (*čítače, nedatováno*).

U asynchronního čítače to znamená, že výstup prvního JK obvodu je zapojen do vstupu časového signálu druhého (viz obrázek 1.16) (Čítače, nedatováno).



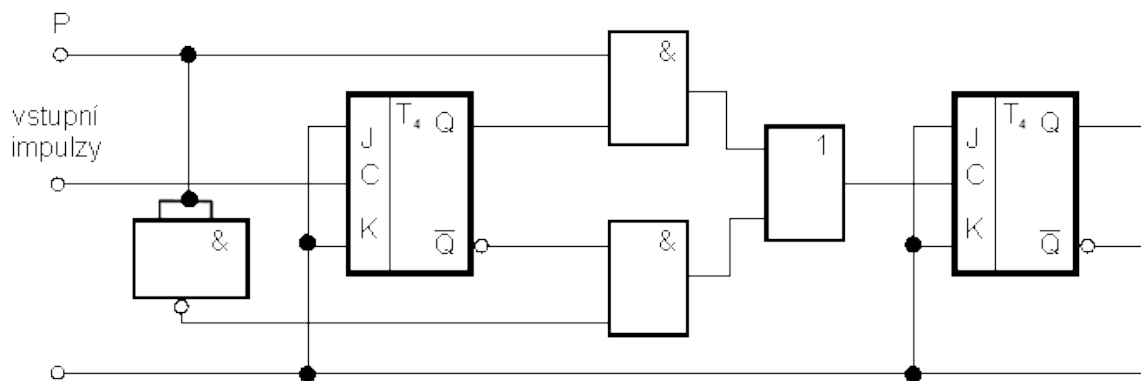
Obrázek 1.16 – 4bitový asynchronní čítač vpřed (Čítače, nedatováno)

U synchronního čítače je potřeba pro realizaci použít i jiné logické obvody (obrázek 1.17). Každý čítač má určitý počet bitů, který určuje počet JK obvodů. Výstupy klopných obvodů tvoří výsledné slovo čítače (Čítače, nedatováno).



Obrázek 1.17 – 4bitový synchronní čítač vpřed (Čítače, nedatováno)

Čítače také dělíme podle směru čítání. Čítač vpřed při vstupním pulzu výsledek inkrementuje, čítač vzad dekrementuje. Vratný čítač při vstupním pulzu výsledek inkrementuje nebo dekrementuje. Zdali bude vratný čítač přičítat, nebo odečítat, záleží na ovládacím vstupním signálu, viz vstup P na obrázku 1.18 (Čítače, nedatováno).



Obrázek 1.18 – Asynchronní 2bitový vratný čítač (Čítače, nedatováno)

Časovač je čítač, který není inkrementován vnějším signálem, ale přímo vnitřním hodinovým signálem, použitým pro chod mikropočítače (Čítače, nedatováno).

### Datové linky

Datové linky slouží pro přenos dat.

Sériové datové linky posílají jeden bit po druhém. V dnešní době jsou čím dál více používané v praxi, a to zejména pro menší počet vodičů, který je potřeba na realizaci přenosu. Proto se používají hlavně na větší vzdálenost. Ovšem informace se před odesláním musí zakódovat a po přenosu zase dekodovat, což ve srovnání s paralelními linkami omezuje jejich přenosovou rychlost. Nejčastěji používané sériové linky jsou USART, I<sup>2</sup>C a SPI (Žoltá, nedatováno).

Rozhraní USART zpřístupňuje mnoho komunikačních protokolů mikropočítačům, jako například RS232, RS422, RS485, a také komunikaci s počítačem pomocí USB-UART. U každého z těchto protokolů slouží ke komunikaci dvě linky, a to Rx a Tx. Některé protokoly obsahují ještě linky pro řízení toku dat. Rx je vždy vstupní linkou a Tx výstupní linkou. Při komunikaci mezi dvěma mikropočítači se Tx z prvního  $\mu$ PC zapojí na Rx u druhého  $\mu$ PC. Komunikace u rozhraní USART probíhá plně duplexně, to znamená, že oba mikropočítače mohou vysílat i přijímat data současně (Dudka, 2016c).

I<sup>2</sup>C je datová sběrnice sloužící ke komunikaci a přenosu dat mezi jednotlivými integrovanými obvody. Nejčastěji pracuje v režimu jedno řídicí zařízení (master) a několik podřízených (slave) obvodů, kdy master si může určovat, s kým právě bude komunikovat, a nikdo mu nemůže komunikaci narušit. Ke komunikaci slouží dvě linky, a to SDA a SCL, kdy na SDA se přenášejí data a SCL slouží jako hodinový signál. Z toho vyplývá, že celá linka funguje synchronně (Dudka, 2016a).

Rozhraní SPI slouží pro komunikaci mezi mikropočítači nebo mezi mikropočítačem a periferiemi. Typicky se používá duplexní přenos, ale může fungovat i simplexně. Simplexní přenos spočítá v tom, že se data odesílají pouze jedním směrem. Ke komunikaci se používají vždy 3 linky, a to SCK, MOSI a MISO. MOSI neboli Master Out Slave In odesílá data od masteru k slave obvodům, MISO neboli Master In Slave Out odesílá data od slave obvodů k masterovi. Na lince SCK je hodinový signál, který je generován masterem a jen pro synchronizaci. Z toho vyplývá, že SPI je synchronní datová linka (Dudka, 2016b).

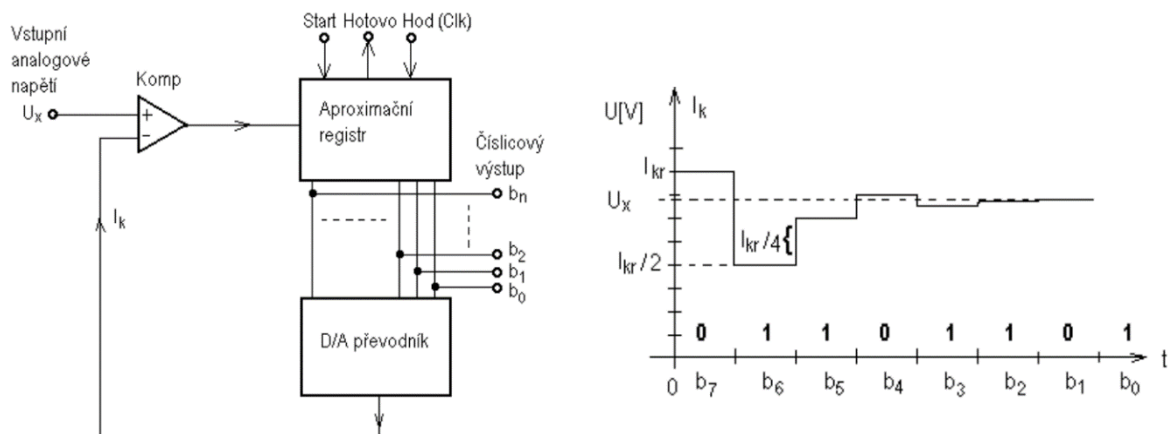
Paralelní datové linky posílají několik bitů v jeden okamžik, což umožňuje vyšší přenosovou rychlost. Pro každý bit však musí mít samostatný vodič. Typicky má paralelní linka tři řídicí vodiče, které obsahují hodiny a signály pro řízení toku a směru přenosu dat. V mikropočítači se s paralelními linkami moc neseťkáme. Nejznámější paralelní linky využívané v osobním počítači jsou PCI, SATA, ATA, ISA (Sběrnice pro PC).

## **Převodníky**

Převodníky slouží pro převádění signálu mezi analogovými a digitálními hodnotami. U převodníků nás zajímá rychlost, přesnost a cena. Existují dva typy převodníků: A/D převodníky a D/A převodníky.

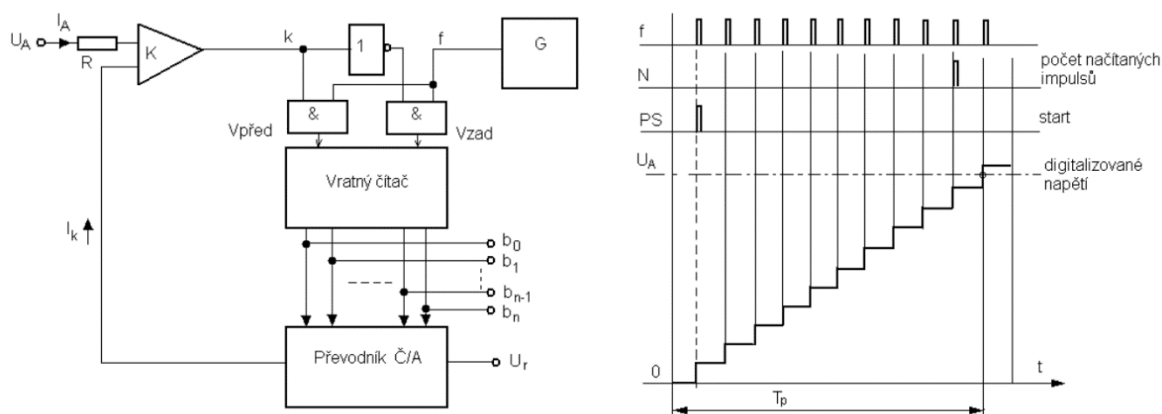
A/D převodníky převádějí analogovou hodnotu na digitální. Nejčastěji se používají převodníky s postupnou aproximací nebo integrační převodníky (Dudka, 2017; Tkotz, 2006).

A/D převodník s postupnou aproximací (obrázek 1.19) obsahuje pouze jeden komparátor, D/A převodník a aproximační registr. Funguje na principu půlení intervalu a zjišťuje výsledek postupně bit po bitu. Nejdříve zjišťuje, zdali je hodnota větší, nebo menší než polovina intervalu, pokud je větší, zjišťuje, zdali je větší, nebo menší než tři čtvrtě intervalu, pokud je menší, zjišťuje, zdali je větší, nebo menší než čtvrtina intervalu a tak dále. Rychlost tohoto převodníku je lineární s počtem bitů. Pro  $n$ -bitový převodník je potřeba vždy  $n$  taktů (Dudka, 2017; D/A a A/D převodníky).



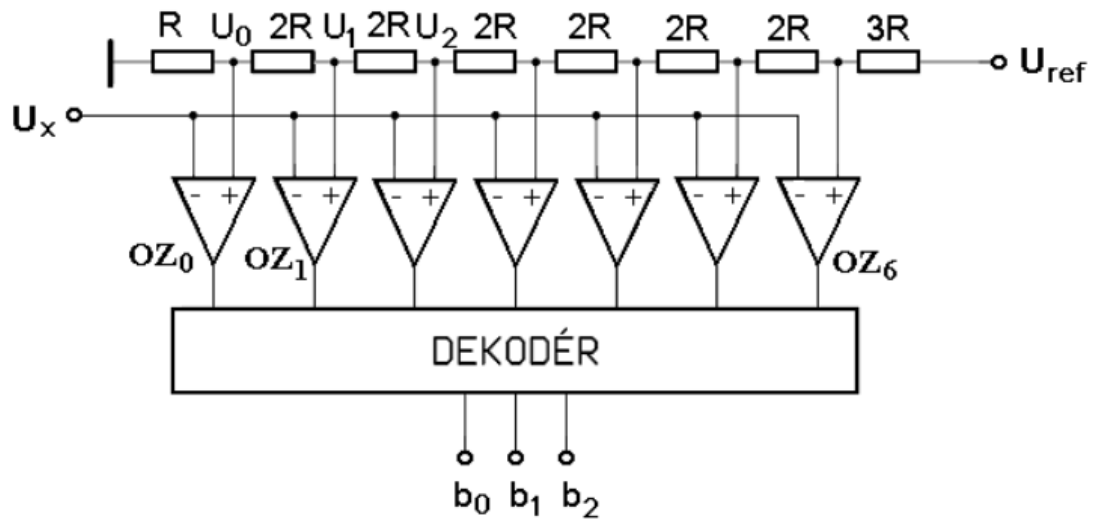
Obrázek 1.19 – Převodník s postupnou aproximací (D/A a A/D převodníky, nedatováno)

A/D převodník s čítačem (obrázek 1.20) obsahuje jeden komparátor, D/A převodník a čítač. Funguje velmi podobně jako převodník s postupnou aproximací, jen místo půlení intervalu postupně čítá po celém intervalu, dokud nepřekročí hodnotu vstupního napětí, a zaznamená hodnotu čítače. Tato hodnota je výstupem A/D převodníku. Některé převodníky mají vratný čítač a díky tomu mohou začít odečítáním z maximální hodnoty. Rychlost tohoto převodníku je lineárně závislá na velikosti vstupního napětí a exponenciálně závislá na velikosti rozsahu převodníku. Pokud by čítač čítal od 0 směrem nahoru a hodnota vstupního napětí by byla maximum intervalu, bude doba trvání u  $n$  bitového převodníku  $2^n$  taktů (Dudka, 2017; D/A a A/D převodníky, nedatováno).



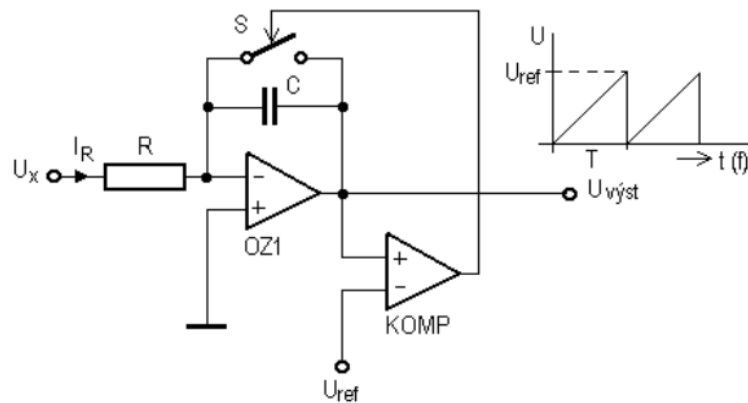
Obrázek 1.20 – A/D převodník s čítačem (ČVUT, nedatováno)

A/D paralelní převodník (obrázek 1.21) se skládá z mnoha komparátorů. Například u  $n$ -bitového převodníku bude  $2^n - 1$  komparátorů,  $2^n$  rezistorů a dekodér. Na rezistorech se rozdělí referenční napětí. Vstupní napětí se porovnává s napětími na jednotlivých rezistorech a poté se dekóduje na digitální hodnotu. Tento převodník patří mezi nejrychlejší A/D převodníky. Převod trvá vždy pouze jeden takt. Ovšem patří mezi nejdražší, protože potřebuje mnoho komparátorů (Dudka, 2017; D/A a A/D převodníky).



Obrázek 1.21 – Paralelní A/D převodník (D/A a A/D převodníky, nedatováno)

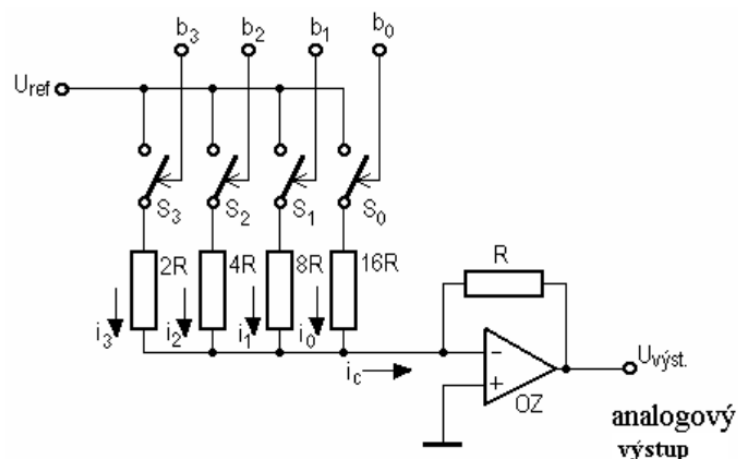
A/D integrační převodník (obrázek 1.22) je složen z operačního zesilovače, zapojeného jako integrátor, a komparátoru. Při měření se integruje určité referenční napětí, které určuje rozsah převodníku a které se neustále porovnává se vstupním napětím komparátorem. V okamžiku, kdy se komparátor sepne, integrované napětí přesáhne vstupní napětí a dojde k takzvanému odintegrování neboli vybití integračního kondenzátoru. Výstupem je signál pilovitého průběhu s určitou frekvencí, která určuje velikost vstupního napětí. Pro získání digitální hodnoty je potřeba změřit frekvenci výstupního napětí například čítačem. Tyto převodníky mají vysokou přesnost za poměrně nízkou cenu (Dudka, 2017; D/A a A/D převodníky, nedatováno).



Obrázek 1.22 – Integrovní převodník (D/A a A/D převodníky, nedatováno; Dudka, 2017)

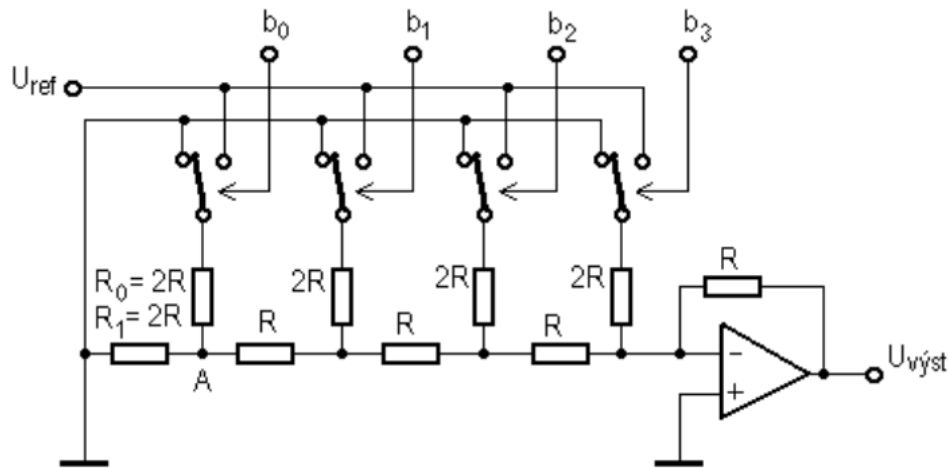
D/A převodníky převádějí digitální hodnotu na analogovou. Mohou být konstruovány pomocí váhové sítě rezistorů, žebříčkové sítě rezistorů a pulzně šířkové modulace (PWM). Mají určité rozlišení, které udává, kolik hodnot mezi minimální a maximální dokážou zpracovat na výstup (Dudka, 2017; Tkotz, 2006).

D/A převodník s váhovou rezistorovou sítí (obrázek 1.23) obsahuje pro každý bit vlastní rezistor, který má pro bit s nejvyšší vahou hodnotu  $R$ , pro bit s druhou nejvyšší vahou hodnotu  $2R$ , to znamená, že každá další hodnota je vždy dvojnásobkem předcházející. Nevýhodou tohoto převodníku je nutnost zapojení velké škály rezistorů. V  $n$ -bitovém převodníku je potřeba  $n$  rezistorů s  $2^{n-1}$  rozsahem, takže do 16bitového převodníku musíme zapojit 16 rezistorů, kde největší a nejmenší musí být v poměru 32 768:1 (D/A a A/D převodníky, nedatováno).



Obrázek 1.23 – Převodník s váhovou rezistorovou sítí (D/A a A/D převodníky, nedatováno)

D/A převodník s žebříčkovou sítí rezistorů (obrázek 1.24), často nazývaný jako R-2R, funguje na principu napěťového děliče. Pro výpočet výstupního napětí se používá Theveninův teorém (D/A a A/D převodníky).



Obrázek 1.24 – Převodník s žebříčkovou sítí rezistorů (D/A a A/D převodníky, nedatováno)

D/A převodník na principu PWM slouží pro přenos analogového signálu pomocí dvou hodnot výstupního signálu. K tomu se nejčastěji používá napětí, méně často proud nebo světelný tok. Výstupem jsou pulzy se stejnou frekvencí a jediné, co se mění, je střída pulzu. Střída pulzu je poměr délky pulzu (zapnuto) ku délce mezery (vypnuto) během jedné periody. Nejčastěji se tato hodnota udává v procentech. Pro představu, při připojení rezistoru jako zátěže na výstupu by měl rezistor při střídě 50 % a výstupním napětí PWM modulátoru 10 V vydávat teplo, jako kdybychom na něj připojili stálé napětí 5 V (D/A a A/D převodníky, nedatováno).

## 1.2.6 Sběrnice

Používá se pro přenos dat. Nevýhodou sběrnice je, že na ní může vysílat pouze jeden počítačový komponent v jeden časový okamžik (Palacký, 2007; Komunikace pro sběrnici, nedatováno).

### Datová sběrnice

Datová sběrnice zajišťuje pohyb dat mezi jednotlivými komponenty mikropočítače. Má určitou šířku, která určuje rychlost průtoku dat. Vždy může pouze jeden komponent vysílat a jeden přijímat, jinak by došlo k neurčitosti signálu nebo poškození vysílačů. Pro její ochranu se používají třístavové budiče, které zabraňují, aby začalo na jedné sběrnici komunikovat více komponentů najednou (Palacký, 2007).



## **Adresová sběrnice**

Adresová sběrnice slouží k přenosu adres. Pokud chce mikroprocesor číst data z nějakého bloku nebo je zapisovat, musí sdělit místo čtení i zápisu. Šířka této sběrnice určuje maximální velikost adres, takže u  $n$ -bitové adresové sběrnice je možné adresovat  $2^n$  adres. Tomuto číslu se říká adresovatelný prostor. V mikroprocesoru nemusí být jen jedna adresová sběrnice. Velká část z nich má dva adresové prostory: jeden pro adresování paměti a druhý pro adresování vstupů a výstupů. Každý komponent, se kterým mikroprocesor komunikuje, musí být umístěn v některém z těchto prostorů. Tyto dva prostory mívají rozdílnou velikost (Palacký, 2007).

## **Řídicí sběrnice**

Řídicí sběrnice zajišťuje přenos řídicích instrukcí pro čtení, zápis, reset a další. V podstatě je spíš souhrnem individuálních signálů s různým významem. Jednotlivé signály mohou mít různé zdroje (Palacký, 2007).

### **1.2.7 Přerušení**

Přerušení je událost, při které mikroprocesor přeruší vykonávání sledu instrukcí, uloží hodnotu z programového čítače do zásobníku, provede obsluhu přerušení, načte zpět hodnotu ze zásobníku do programového čítače a poté pokračuje dál v původním sledu instrukcí. Přerušení se rozděluje na vnější a vnitřní (Čítače, nedatováno; Karas, 2006).

Vnitřní přerušení vyvolá sám procesor. Signalizuje tím chybu zpracování instrukce. Vnitřním přerušením může být třeba pokus o dělení nulou (Čítače, nedatováno).

Vnější přerušení přichází ze vstupně-výstupních obvodů a většinou je signál o přerušení poslán přímo do řadiče instrukcí (Čítače, nedatováno).

Vektor přerušení určuje prioritu přerušení a adresu, kde najdeme, co se má při obsluze přerušení vykonat. Vektory přerušení jsou zpravidla uloženy na začátku paměti mikropočítače. Priorita přerušení je dána tím, jak vysoko je vektor v tabulce přerušení. Důležité je, že pokud už probíhá nějaké přerušení a dojde k dalšímu jinému přerušení, záleží na prioritách přerušení. Pokud má probíhající přerušení menší prioritu, tak se přeruší a proběhne přerušení s větší prioritou, po kterém se dokončí přerušení, které předtím probíhalo. Pokud má probíhající přerušení větší prioritu, proběhne beze změny (Čítače, nedatováno; Karas, 2006).

Tabulka 1.3 – Tabulka vektorů přerušení pro ATmega32A (ATmega32A, 2018)

Číslo vektoru	Adresa programu	Zdroj přerušení	Definice přerušení
1	\$000	RESET	Reset mikropočítače
2	\$002	INT0	Externí přerušení 1
3	\$004	INT1	Externí přerušení 2
4	\$006	INT2	Externí přerušení 3
5	\$008	TIMER2 COMP	Časovač/čítač2 porovnání shody
6	\$00A	TIMER2 OVF	Časovač/čítač2 přetečení
7	\$00C	TIMER1 CAPT	Časovač/čítač1 zachycení události
8	\$00E	TIMER1 COMPA	Časovač/čítač1 porovnání shody A
9	\$010	TIMER1 COMPB	Časovač/čítač1 porovnání shody B
10	\$012	TIMER1 OVF	Časovač/čítač1 přetečení
11	\$014	TIMER0 COMP	Časovač/čítač0 porovnání shody
12	\$016	TIMER0 OVF	Časovač/čítač0 přetečení
13	\$018	SPI, STC	Dokončení sériového přenosu
14	\$01A	USART, RXC	USART, Rx dokončen
15	\$01C	USART, UDRE	USART prázdný registr dat
16	\$01E	USART, TXC	USART, Tx dokončen
17	\$020	ADC	ADC převod dokončen
18	\$022	EE_RDY	EEPROM paměť připravena
19	\$024	ANA_COMP	Analogový komparátor
20	\$026	TWI	Sériová dvoulinka
21	\$028	SPM_RDY	Připraveno uložit programovou paměť

## 1.2.8 Programovací jazyk a program

### Program

Počítačový program je posloupnost několika instrukcí nebo algoritmů podle předepsané syntaxe, které určují, co má mikropočítač dělat. Ve většině programů najdeme proměnné a konstanty, které mají určitý datový typ, jenž určuje, je-li proměnná text, kladné celé číslo nebo reálné číslo apod. Dále můžeme v programech najít pole, funkce a další (Tkotz, 2006; Počítačový program (Computer program), nedatováno).

### Programovací jazyk

Mikropočítače i počítače umí zpracovávat pouze strojový kód, a protože napsat nějaký algoritmus pro zpracování by bylo ve strojovém kódu velmi náročné, používají se programovací jazyky, které se píšou v určité syntaxi bližší lidské řeči a je poté převedena do strojového kódu (Bechyňová, nedatováno).

Programovací jazyky se dělí například podle míry abstrakce na vyšší a nižší (neboli vysokoúrovňové a nízkoúrovňové) a podle způsobu překladu a spuštění na kompilované a interpretované (Bechyňová, nedatováno).

Kompilované programovací jazyky se před spuštěním musí kompletně přeložit kompilátorem do strojového kódu. Pro tyto jazyky je charakteristická větší rychlost, ale také větší náročnost na programování (Bechyňová, nedatováno).

Interpretované programovací jazyky se překládají za chodu programu nebo se překládají pouze do mezikódu (Bechyňová, nedatováno).

Vyšší programovací jazyky se snaží přesunout od strojového kódu k lidskému myšlení. Pro většinu lidí jsou vyšší programovací jazyky srozumitelnější. Příkladem takového programovacího jazyka je Java, C++, Python a další. Vyšší programovací jazyky lze rozdělit na procedurální a neprocedurální. U procedurálního jazyka se popisuje výpočet pomocí posloupnosti příkazů, což určuje přesný postup, jak se má celý algoritmus řešit. Neprocedurální neboli deklarativní jazyk popisuje programování pomocí definic, co se má udělat, a ne jak se to má udělat (Bechyňová, nedatováno).

U nižších programovacích jazyků existují minimální rozdíly mezi programovacím jazykem a strojovými instrukcemi a je nutné, abychom při programování znali velmi dobře hardware, na který budeme program nahrávat. Můžeme je rozdělit na jazyky první a druhé generace. První generace je v podstatě strojový kód, zatímco u druhé generace už zapisujeme jen instrukce. Nižší programovací jazyky je možné lépe odladit a mohou zabírat méně paměti (Bechyňová).

Programovacích jazyků je mnoho, ale jelikož ve své práci používám pouze programovací jazyk C, který je hodně spjatý s assemblerem. Jsou to nejpoužívanější programy pro programování pro AVR procesory, a proto popíši jen tyto dva (Bechyňová, nedatováno).

## **Programovací jazyk C**

Programovací jazyk C je často považován za nízkoúrovňový, avšak na rozdíl od ostatních nízkoúrovňových jazyků je nezávislý na počítačové platformě. Poskytuje pouze nízkoúrovňový přístup k operační paměti a programové konstrukce, které je snadné přímo převést do strojového kódu. Má možnost takzvaného inline assembleru, což znamená, že do kódu v jazyce C můžeme připsat řádky v assembleru (Fabian, nedatováno; Úvod do jazyka C, nedatováno).

Jazyk C má pět hlavních bloků pro implementaci programu, a to editor, preprocesor, kompilátor, linker a debugger. V editoru se tvoří a upravuje zdrojový soubor. Preprocesor je

částí překladače a předzpracovává zdrojový soubor dřív, než se dostane do kompilátoru. Kompilátor překládá předzpracovaný kód z preprocesoru do relativního kódu, který je téměř hotovým programem. Linker mu pak přiděluje absolutní adresy a provádí všechny odkazy na dosud neznámé identifikátory. Výstupem z linkeru je hotový spustitelný program. Debugger je tak pouze pomocný ladící program pro nalezení chyb při běhu programu (Fabian, nedatováno; Úvod do jazyka C, nedatováno).

## Assembler

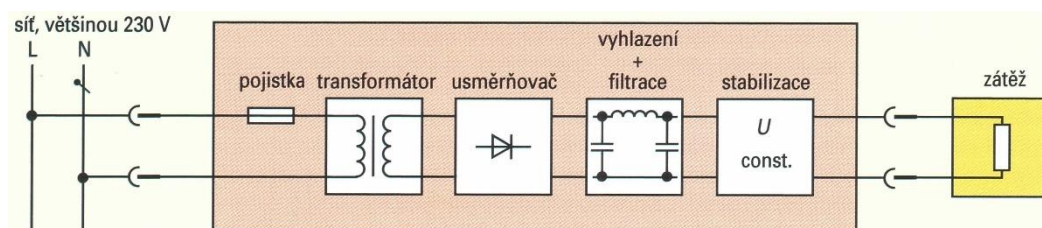
Assembler neboli jazyk symbolických adres je nástroj, který překládá program zapsaný v symbolických jménech do jednotlivých instrukcí ve strojovém kódu. Často je ještě potřeba linker pro přidělení absolutních adres. Výstupem linkeru je hotový program ve strojovém kódu (Malý, 2013; Malý, 2014).

## 1.3 STEJNOSMĚRNÉ ZDROJE NAPĚTÍ

Stejnoseměrné zdroje napětí rozdělujeme na laboratorní regulovatelné, regulovatelné a pevné. Laboratorní regulovatelné zdroje jsou určeny především pro laboratoře. Regulovatelné zdroje se používají hlavně u méně náročných operací. Pevné zdroje se používají všude, kde potřebujeme stálé stejnosměrné napětí. Dále se rozdělují na lineární a spínané (Bulka, 2002).

### 1.3.1 Lineární zdroj napětí

Každý lineární zdroj napětí (obrázek 1.25) má čtyři části, a to transformátor, usměrňovač, vyhlazení s filtrací a stabilizaci. Před transformátorem ještě často najdeme pojistku pro případnou ochranu zdroje, kdyby došlo ke zkratu. Lineární zdroje mívají účinnost mezi 20 – 50 % (Tkotz, 2006).



Obrázek 1.25 – Blokové schéma lineárního zdroje napětí (Tkotz, 2006)

Transformátor převádí vstupní střídavé napětí na nižší střídavé napětí, aby s ním mohly další součástky snáze pracovat. Také tvoří galvanické oddělení obvodů od sítě jako ochranu před úrazem elektrickým proudem (Tkotz, 2006).

Usměrňovač se používá k převodu střídavého napájení na stejnosměrné. Usměrňovačů je více druhů. Dělíme je na jednocestné, dvoucestné a zapojené do můstku, se kterými se setkáme nejčastěji, a na kontaktní a polovodičové, kdy kontaktní jsou složené z vypínačů a polovodičové z polovodičových součástek. Polovodičové dělíme dále na řízené a neřízené, kdy u řízených můžeme ovládat dobu sepnutí kontaktů (SPŠ, SOŠ a SOU Hradec Králové, Usměrňovače, nedatováno)

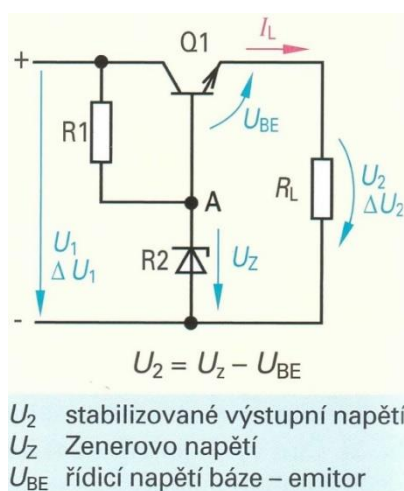
Jednocestný usměrňovač obsahuje jeden spínací prvek a propouští každou kladnou půlvlnu (SPŠ, SOŠ a SOU Hradec Králové, Usměrňovače, nedatováno).

Dvoucestný usměrňovač obsahuje dva spínací prvky a propouští kladnou i zápornou půlvlnu. Ovšem pro realizaci dvoucestného usměrňovače je potřeba transformátor s rozděleným sekundárním vinutím, který je dražší (SPŠ, SOŠ a SOU Hradec Králové, Usměrňovače, nedatováno).

Můstkový usměrňovač obsahuje čtyři spínací prvky a propouští obě půlvlny stejně jako dvoucestný, ale stačí mu obyčejný transformátor. Nevýhodou je, že vykazuje úbytek napětí dvakrát větší než předešlé usměrňovače, nejčastěji cca 1,4 V (0,7 V na každé diodě) (SPŠ, SOŠ a SOU Hradec Králové, nedatováno).

Vyhlazení a filtrace mají za úkol co nejvíce odstranit střídavou složku z usměrněného napětí. Většinou se řeší pomocí kondenzátorů a cívek (Skryja, 2016).

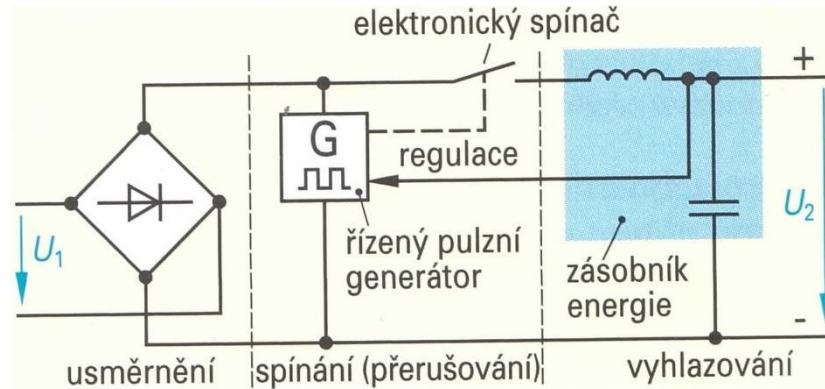
Stabilizační obvody udržují vyhlazené a vyfiltrované napětí na stálé hodnotě. Výstupem stabilizátoru je výstupní napětí zdroje (Skryja, 2016).



Obrázek 1.26 – Zapojení jednoduchého lineárního zdroje napětí

### 1.3.2 Spínaný zdroj napětí

Spínaný zdroj napětí (obrázek 1.27) se nejčastěji skládá z usměrňovače, měniče, řídicích obvodů a zásobníku energie s vyhlazením. Účinnost spínaných zdrojů se pohybuje mezi 60 – 90 % (Vlček, 2007).



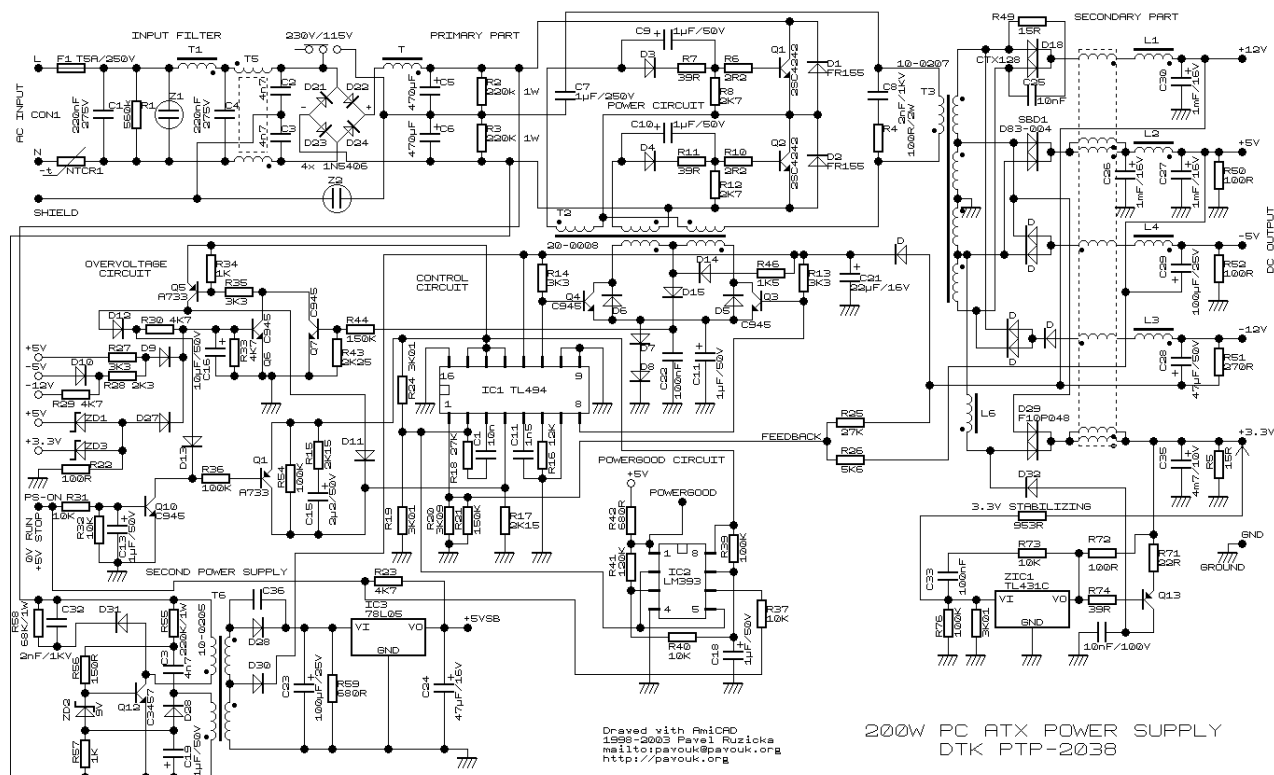
Obrázek 1.27 – Zapojené jednoduchého spínaného zdroje (Tkotz, 2006)

Ve spínaných zdrojích napětí se používá pouze můstkový usměrňovač. Hned po usměrnění dochází k prvnímu vyhlazení signálu (Vlček, 2007).

DC/DC měnič vytvoří z vyhlazeného signálu obdélkový o poměrně vysoké frekvenci, většinou 20–200 kHz, která je stálá. Jediný parametr, který se mění, je střída obdélkového signálu. Velikost střídy určují řídicí obvody. Obdélkový signál je zaslán do zásobníku energie (Vlček, 2007).

Řídicí obvody zjišťují zatížení zdroje změřením poklesu výstupního napětí na výstupu zdroje napětí. Pokud výstupní napětí klesne, nařídí měniči, aby se zvětšila střída a naopak (Dvořák, 2008).

Zásobník energie neboli také nábojová pumpa zajišťuje dostatek náboje na výstupu při rychlejší zatížení, aby nebyl pokles napětí moc velký, než řídicí obvody přikážou zvětšení střídy (Dvořák, 2008).



Obrázek 1.28 – Schéma spínaného PC zdroje s ATX standardem (Pavouk, nedatováno)

## 1.4 3D MODEL A TISK

Před zhotovením výrobku na 3D tiskárně je potřeba podstoupit několik kroků od vytvoření modelu přes přeložení kódu pro tiskárnu až po výběr materiálu samotného tisku a konečné úpravy po vytištění. Modelování a tisk ve 3D se v posledních letech rozmohly, a nejen ve strojírenském průmyslu, ale také ve zdravotnictví, zbrojním průmyslu, stavebnictví, umění a v domácím využití (What is 3D printing, nedatováno).

### 1.4.1 Vytvoření modelu

Vytvoření modelu může probíhat dvěma způsoby, a to buď návrhem modelu v 3D editoru, nebo naskenováním tělesa.

#### 3D skenování

Nejpřesnějším, ale také nejdražším způsobem vytvoření modelu je použití 3D skeneru, ale za použití fotogrammetrického softwaru je možné vytvořit model i kamerou nebo fotoaparátem. Následně se doporučuje použít editor na upravení a doladění modelu. Ovšem jak

při použití 3D skeneru, tak při použití fotoaparátu je nutné mít fyzický výrobek, jinak nezbývá než jej navrhnout ve speciálním editoru. Tímto způsobem získáme 3D model rychle, ale s menší přesností (Enfroy, 2020).

## Editor

V 3D editoru můžeme pomocí přímek, křivek, kružnic, ploch a těles, ze kterých se skládá výsledný model, vytvořit objekt podle potřeb, ovšem občas ho musíme rozdělit na více částí buď z důvodu velikosti, protože tiskárna má jen omezený prostor pro tisknutí, nebo z důvodu jednoduššího provedení tisku (potřeba méně podpor). Editorů pro návrh 3D modelů je hodně, a proto uvedu jen pár nejpoužívanějších (Francesco, 2013).

Autodesk Fusion 360 je jeden z nejznámějších 3D editorů. Nabízí mnoho možností pro rychlé a kvalitní vytváření modelů (Francesco, 2013).

Autodesk AutoCAD je jeden z nejpoužívanějších softwarů pro 3D modeling v průmyslu. Nabízí zajímavé možnosti, jako import dokumentů z pdf a další (Francesco, 2013)

Ultimaker Cura je velmi jednoduchý editor, vhodný pro začátečníky. Je kompletně zdarma, určen spíše pro domácí použití (Francesco, 2013).

Creo má výborně zpracované nástroje. Používá se hojně v průmyslu (Francesco, 2013)

SketchUP se používá hlavně v architektuře (Francesco, 2013).

## Slicing

Po dokončení návrhu je potřeba „rozsekat“ model na jednotlivé vrstvy tak tenké, aby je zvládla tiskárna vytisknout a aby to vyhovovalo požadavkům na kvalitu výrobku. Nejčastěji se tloušťka vrstvy pohybuje v desetínách milimetru (Spiegel, 2018).



Obrázek 1.29 – Ukázka slicingů (Spiegel, 2018)



## G-kód

V G-kódu je zapsána každá akce 3D tiskárny, která má být provedena: například dráha a směr pohybu trysky v prostorových souřadnicích, typ pohybu (zda je lineární nebo kruhový), jeho rychlost, kolik filamentu se má z trysky vytlačit, zapnutí a nastavení teploty podložky nebo trysky a další (Lineární motory, lineární pohyb, nedatováno).

Při převodu modelu do G-kódu je velmi důležité zadat správný materiál, kterým budeme tisknout, protože se tím do kódu zapíše, na jakou teplotu se má nahřát tryska, aby se materiál správně tavil a nepřepaloval, a na jakou teplotu se má nahřát podložka, aby nedošlo k deformaci materiálu z důvodů různé tepelné roztažnosti materiálů při chladnutí filamentu (Lineární motory, lineární pohyb, nedatováno).

### 1.4.2 Tisk

Po kompletním převodu objektu do 3D modelu už zbývá jen předat tiskárně data k tisku: vybrat na tiskárně program, kterým se má řídit, ujistit se, že má v zásobníku připojený filament a spustit program. Po spuštění programu bude nejdříve tiskárna nahřívát podložku a trysku. Když teploty dosáhnou požadované hodnoty, tiskárna začne tisknout. Tisknout dnes lze z plastů, kovu a dalších materiálů (WHAT ARE 3D PRINTED MATERIALS AND HOW ARE THEY USED?, nedatováno).

Tabulka 1.4 – Plastové filamenty (WHAT ARE 3D PRINTED MATERIALS, nedatováno)

Materiál	Teplota pro tisk, °C	Vlastnosti
ABS	220 - 240	Odolný vůči nárazům, silný a lehce flexibilní
PLA	180 - 200	Malá odolnost proti teplotám, chemikáliím a vodě, trochu křehké
PVA	200	Uvolňuje nepříjemné chemikálie při vyšších teplotách, rozpustné ve vodě
Nylon	250	Má vysokou pevnost v tahu, dobrá odolnost proti chemikáliím, špatně doléhá na podložku
PET	230	Schválen pro použití s potravinami, silný a odolný, ale musí být tištěn pomalu, aby k sobě přilehly vrstvy

### Konečné úpravy po ukončení 3D tisku

Po odstranění výrobku z podložky je nutné provést posledních pár úprav, jako odříznutí podpor, opílování nebo vyčištění. Pokud jsme museli výrobek tisknout po částech, slepíme

jednotlivé části k sobě. Někdy je potřeba, aby měl výrobek hladký povrch, stříká se buď fillerem (plničem), nebo se brousí. Ovšem ne všechny materiály jsou vhodné pro broušení. V případě potřeby se výrobky i lakují (WHAT ARE 3D PRINTED MATERIALS AND HOW ARE THEY USED?, nedatováno).

## 1.5 KOMERČNĚ VYRÁBĚNÉ SERVOMOTORY

Servomotory se vyrábějí jak s DC motorem, tak AC motorem o výkonu od jednotek wattů až po desítky kilowattů a krouticím momentu až několik stovek newtonmetrů (N·m). Jsou připraveny do různých prostředí a pro specifické operace. Nejčastější napájení je 12, 24, 48 V stejnosměrného napětí a 120, 240, 480 V střídavého napětí. Mohou mít různé tvary hřídele, a to i dutou, vřetenovou či hřídel vycházející z obou stran elektromotoru. Vyrábějí se s různou rychlostí otáčení, od stovek po lehce přes 10 tisíc otáček za minutu, a dosahují až několika desítek i stovek inkrementů na výstupu enkodéru za jednu otáčku (Raveo s.r.o.; REM-technik).

Servopohony se prodávají jako otočné i lineární. Otočné mění úhel kolem osy hřídele motoru, zatímco lineární pohybují objektem po přímce (Lineární motory, lineární pohyb, nedatováno).

Servopohony mají i své certifikace, které určují, do jakého prostředí se dané servopohony hodí a do jakého ne. Certifikace ATEX je evropská norma určující stálost a bezzávadnost ve výbušném prostředí. JAT jsou servomotory používané při velmi náročných operacích a specifických úlohách (Pohyb, pohony, převody, nedatováno).

Servomotory s DC motorem vyrábějí například firmy Kollmorgen, Moons, Stöber, ECOSPEED, TG Drives (Pohyb, pohony, převody / Motory a servomotory / Servomotory, nedatováno).



Obrázek 1.30 – Servomotor od firmy Kollmorgen (Pohyb, pohony, převody, nedatováno)

## 2 PRAKTICKÁ ČÁST

### 2.1 PLÁNOVÁNÍ A VÝBĚR SOUČÁSTEK

Pro praktickou část jsem použil počítačovou skříň se zdrojem ATX standardu. Tento zdroj byl vlastně úplně ideální pro mou práci, protože na výstupech má napětí jak 12 V pro připravený motor, tak 5 V pro logické obvody s maximálním skokovým výkonem 400 W a stálým výkonem 350 W, dokonce se dá zapínat uzemněním POWER\_ON, má oznámení POWER\_GOOD, zda zdroj běží, jak má, ochranu proti zkratování a aktivní chlazení. Déle jsem chtěl použít stejnosměrný motor o výkonu 60 W s napájecím napětím 12 V (nejspíše motorek stěrače z automobilu Škoda MB 1000, rok výroby 1965). Tento motor byl ale pro moje účely moc velký, těžký, zbytečně silný, s příliš vysokými otáčkami a bez enkodéru.



Obrázek 2.1 – Počítačová skříň se zdrojem ATX standardem

Použil jsem mikropočítač ATmega32 v pouzdře DIP40. ATmega32 disponuje 32 kilobajty FLASH paměti pro program, 1024 bajty SRAM paměti, čtyřmi PWM kanály, osmi 10bitovými A/D převodníky, 32 vstupně-výstupními piny, 32 pracovními registry a dalšími součástkami (Microchip Technology, 2018).



Obrázek 2.2 – Mikropočítač ATmega32

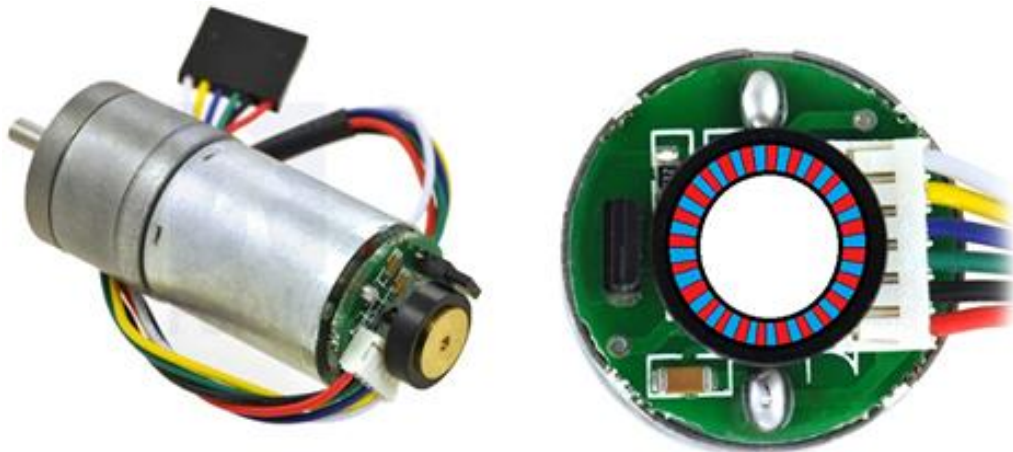
Jako driver pro ovládání motorů jsem použil L298N. Tento driver má 2 kanály a na každém z nich může mít maximální stálý výstup 2 A. U každého kanálu lze změnit směr otáčení či zapnout elektronickou brzdu motoru. Je možné přivést PWM signál z mikropočítače na vstup enable motoru pro ovládání otáček (STMicroelectronics, 2000).



Obrázek 2.3 – Driver L298N

Pro oddělení řídicí a výkonové části jsem se rozhodl použít DC/DC měnič TBA 2-0511.

Otáčení konstrukce je realizováno pomocí stejnosměrného motoru od firmy Pololu s cizím buzením. Součástí motoru je i převodovka a magnetický enkodér. Může být napájen až 12 V, při kterých má na výstupu 5600 otáček za minutu (Metal Gearmotor 25Dx66L, nedatováno).



Obrázek 2.4 – DC motor od firmy Pololu a rozdělení pólů na magnetickém disku (Metal Gearmotor 25Dx66L, nedatováno).

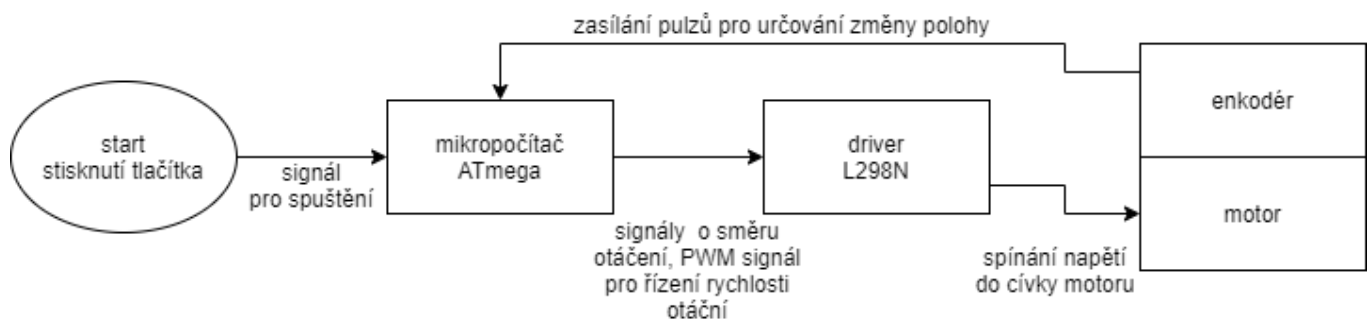
Převodovka je složená z šesti párů kovových ozubených kol s celkovým převodovým poměrem  $1 : 98,77794\bar{6}$ , což činí 56,6928 otáčky za minutu bez zátěže. Vstupní proud

naprázdno při 12 V je 100 mA a při největším možném zatížení, což je 0,8 N/m, dosáhne motor odběru proudu 1,1 A (Metal Gearmotor 25Dx66L, nedatováno).

Magnetický enkodér je vybavený dvěma Hallovy sondami a magnetickým diskem, který je zmagnetizovaný, aby měl po obvodu 48 pólů (obrázek 2.4, červeně označený kladný pól a modře záporný pól). Hallovy sondy jsou vůči sobě signálově posunuté o 90 stupňů, díky čemuž můžeme určit směr otáčení motoru (Metal Gearmotor 25Dx66L, nedatováno).

## 2.2 FUNKCE JEDNOTLIVÝCH SOUČÁSTEK

Na následujícím blokovém schématu (obrázek 2.5) je popsána komunikace mezi jednotlivými součástkami.



Obrázek 2.5 – Blokové schéma funkce výrobku

### 2.2.1 Mikropočítač ATmega

V mikropočítači jsou nahrány souřadnice pohybů a program, který má za úkol nejen zaznamenat aktuální polohu, ale i vypočítat zbývající vzdálenost a řídit rychlost a směr otáčení motoru.

Jako první dojde k inicializaci proměnných, přerušení, portů a časovače 1.

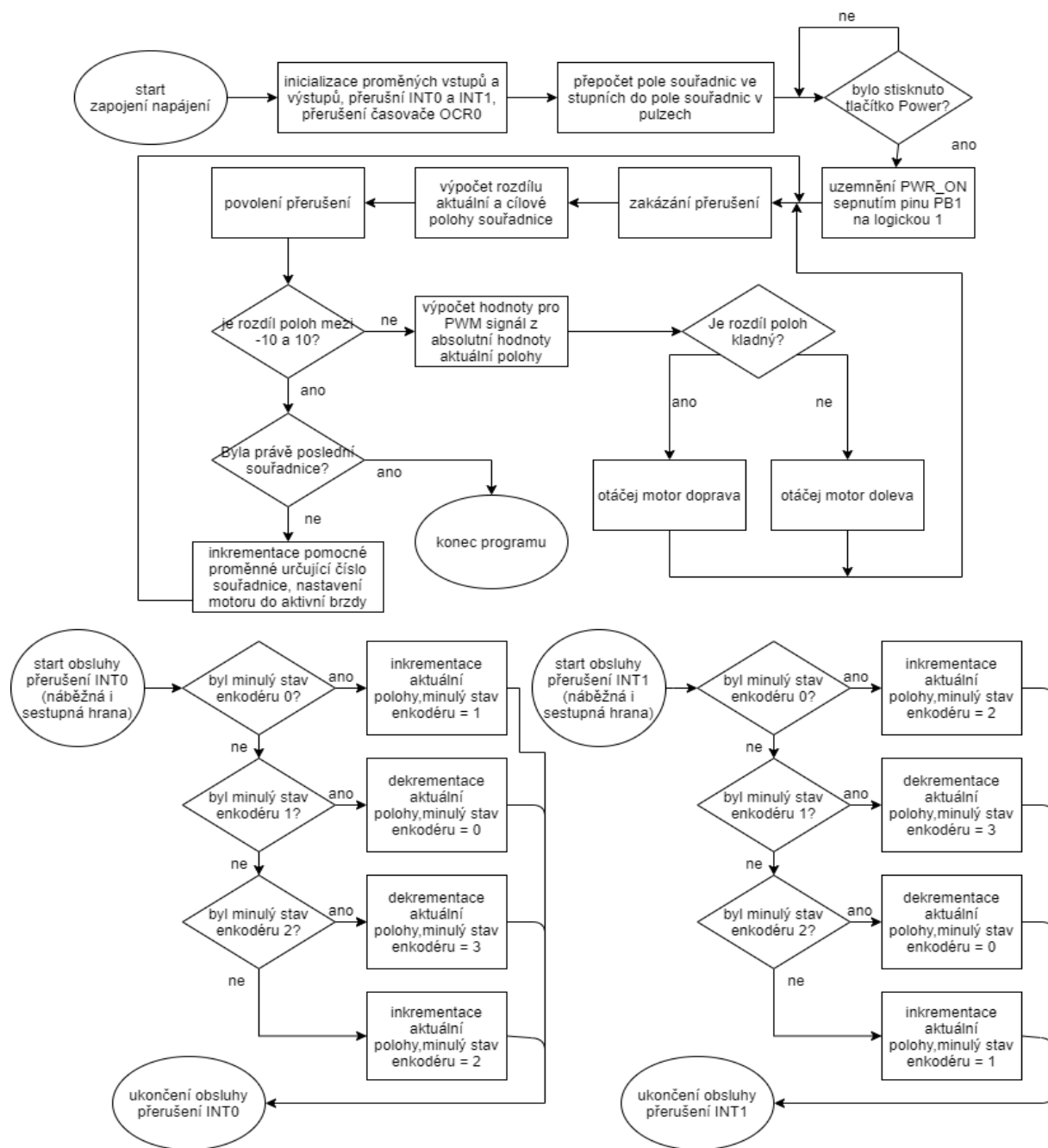
Souřadnice jsou v mikropočítači zapsány v poli jako proměnné ve stupních, ale mikropočítač počítá s pulzy z enkodéru, takže je potřeba přepočítat souřadnice ze stupňů do počtu pulzů. Mikropočítač přepočítává jednu souřadnici po druhé, dokud nejsou přepočítány všechny. Souřadnici ve stupních vynásobí počtem pulzů za otáčku a výsledný součin vydělí počtem stupňů za otáčku. Pulzy se vypočítají pomocí rychlosti otáčení motoru (ot./s), počtu pólů na disku magnetického enkodéru a převodu za motorem, což činí 8920 pulzů na jednu otáčku kloubu.

$$\text{pulzusaot} = \frac{5600 * 48 * 2}{60} = 8960 \quad (2.1)$$

K odečítání polohy je využito přerušení INT0 a INT1, u kterých čip porovnává, jaké přerušení nastalo a jaký byl minulý stav vstupů přerušení (Microchip Technology, 2018).

Řízení směru probíhá pomocí porovnání aktuální polohy k nule: zdali je kladná nebo záporná. Pokud je kladná, nastaví se PC0 na logickou nulu a PC1 na logickou jedničku. Pokud je záporná, nastaví se PC0 na logickou jedničku a PC1 na logickou nulu (Microchip Technology, 2018).

K řízení rychlosti otáčení je použit časovač TIMER0 pro vytvoření PWM signálu. Pulzně šířkovou modulaci využívám na neinvertujícím kanále v režimu phase correct, kde mohu řídit střidu od 0 do 100 %. Střida PWM signálu závisí na hodnotě proměnné, se kterou se časovač porovnává, a kolikabitový časovač používáme. TIMER0 má osm bitů, takže čítá od 0 do 255. Zanesením čísla z tohoto intervalu do registru OCR0 nastavuji střidu. Do registru OCR0 je zanášena hodnota rozdílu aktuální a cílové polohy dělena čtyřmi. Pokud je potřeba ujet ještě tisíc pulzů, motor začne zpomalovat. Pokud by chybělo ujet více než 1000 pulzů, tak by bylo zaneseno do registru OCR0 číslo větší než 255, to ale nesmí nastat. Proto, pokud je rozdíl hodnot dělený čtyřmi větší než 255, nastaví se hodnota OCR0 na 255 (Microchip Technology, 2018).



Obrázek 2.6 – Vývojový diagram programu

Na obrázku 2.6 se při zapnutí napájení spustí horní část programu ve vývojovém diagramu. Při změně signálu na vstupech přerušení INT0 a INT1 se zastaví hlavní program a mikropočítač začne provádět obsluhu přerušení. Po dokončení se vrátí zpět k hlavnímu programu, kde skončil.

Jak je uvedeno výše, ATmega32 je v pouzdře DIP40. To znamená, že má 40 vývodů a piny mikropočítače se upevňují zapájením do otvorů v plošném spoji. Každý pin má jednu nebo více funkcí (Microchip Technology, 2018)

Tabulka 2.1 – Tabulka vývodů a jejich funkcí

číslo pinu	název	funkce	využití pinu v mojí práci
1	PB0	pin portu B	ovládání zdroje napětí
2	PB1	pin portu B	tlačítko zapínání
3	PB2	pin portu B, vývod přerušeni INT2	-
4	PB3	pin portu B, výstup časovače0	výstup PWM signálu do enable driveru
5	PB4	pin portu B	-
6	PB5	pin portu B, pin MOSI pro SPI programování	vývod MOSI pro SPI programování
7	PB6	pin portu B, pin MISO pro SPI programování	vývod MISO pro SPI programování
8	PB7	pin portu B, pin SCK pro spi programování	vývod SCK pro SPI programování
9	RESET	při uzemnění resetuje mikropočítač	reset mikropočítače
10	VCC	vstup napájecího napětí (2,7 V - 5,5 V)	přívod napájecího napětí
11	GND	uzemnění	uzemnění
12	XTAL2	vývod pro externí krystal udávající hodinový signál mikropočítače	připojení 16MHz krystalu
13	XTAL1	vývod pro externí krystal udávající hodinový signál mikropočítače	připojení 16MHz krystalu
14	PD0	pin portu D, RX pro USART	-
15	PD1	pin portu D, TX pro USART	-
16	PD2	pin portu D, vývod pro přerušeni INT0	vstup z enkodéru do přerušeni INT0
17	PD3	pin portu D, vývod pro přerušeni INT1	vstup z enkodéru do přerušeni INT1
18	PD4	pin portu D, výstup časovače1A	-
19	PD5	pin portu D, výstup časovače1B	-
20	PD6	pin portu D	-
21	PD7	pin portu D, výstup časovače2	-
22	PC0	pin portu C	informace o směru otáčení do IN1 driveru
23	PC1	pin portu C	informace o směru otáčení do IN2 driveru
24	PC2	pin portu C	-
25	PC3	pin portu C	-
26	PC4	pin portu C	-
27	PC5	pin portu C	-
28	PC6	pin portu C	-
29	PC7	pin portu C	-
30	ACC	pomocný vstup napájení pro port A a A/D převodník	přívod napájecího napětí



Tabulka 2.1 – Tabulka vývodů a jejich funkcí (pokračování)

číslo pinu	název	Funkce	využití pinu v mojí práci
31	GND	Uzemnění	uzemnění
32	AREF	referenční napětí pro A/D převodník	-
33	PA0	pin portu A, vstup ADC0 A/D převodníku	vstup informace o proudu protékajícím motorem
34	PA1	pin portu A, vstup ADC1 A/D převodníku	-
35	PA2	pin portu A, vstup ADC2 A/D převodníku	-
36	PA3	pin portu A, vstup ADC3 A/D převodníku	-
37	PA4	pin portu A, vstup ADC4 A/D převodníku	-
38	PA5	pin portu A, vstup ADC5 A/D převodníku	-
39	PA6	pin portu A, vstup ADC6 A/D převodníku	-
40	PA7	pin portu A, vstup ADC7 A/D převodníku	-

Všechny vývody portů A, B, C, D mají možnost pull-up rezistorů a v tabulce nejsou všechny funkce vývodů (Microchip Technology, 2018).

### 2.2.2 Driver L298N

L298N je v provedení H-můstku, který má dva kanály a zvládne spínat až 46 V, 2 A (každý kanál) a snese maximální výstupní výkon 25 W. Nachází se v pouzdru multiwatt15 a má 15 vývodů, vždy 6 pro každý kanál a 3 pro napájení. Při použití jednoho či obou kanálů je nutné zapojit napájení, a to uzemnění na GND, napájecí napětí pro logické obvody (4,5 – 7 V) na Vcc a napájecí napětí pro výkonovou část (Vcc + 2 – 46 V) na Vss. Napájecí napětí pro logickou a výkonovou část musí mít spojené uzemnění. Každý kanál má dva vstupy IN1 a IN2 (IN3 a IN4) pro řízení směru či aktivní brzdy, ENABLE\_A (ENABLE\_B), kterým se spíná napětí na výstup, SENSE\_A (SENSE\_B), kterým se určuje proud výstupem, a OUT1 a OUT2 (OUT3 a OUT4), které jsou výstupy a připojují se přímo na cívku motoru (STMicroelectronics, 2000).

Pokud bychom mezi SENSE\_A a zem připojili 1  $\Omega$  rezistor, tak můžeme na rezistoru změřit přesně takové napětí, jaký je proud cívku motoru (STMicroelectronics, 2000).

Tabulka 2.2 – Možné stavy driveru L298N (STMicroelectronics, 2000)

Spínače				Výsledná funkce
S1	S2	S3	S4	
0	0	0	0	Motor není napájen "běží na volnoběh"
0	0	0	1	
0	0	1	0	
0	1	0	0	
1	0	0	0	
1	0	0	1	Motor se otáčí doprava
0	1	1	0	Motor se otáčí doleva
0	1	0	1	Aktivní brzda
1	0	1	0	
x	x	1	1	Zakázaný stav (zkrat)
1	1	x	x	

### 2.2.3 Motor od výrobce POLOLU

Motor napájím 12 V, takže mohu dosáhnout plného výkonu. Na hřídeli motoru je nasazeno ozubené kolo, které doléhá na vnitřní ozubení otáčené konstrukce s převodem 1 : 2, takže konečný převod je 1 : 197,555983 a konečné otáčky 28,3464 za minutu.



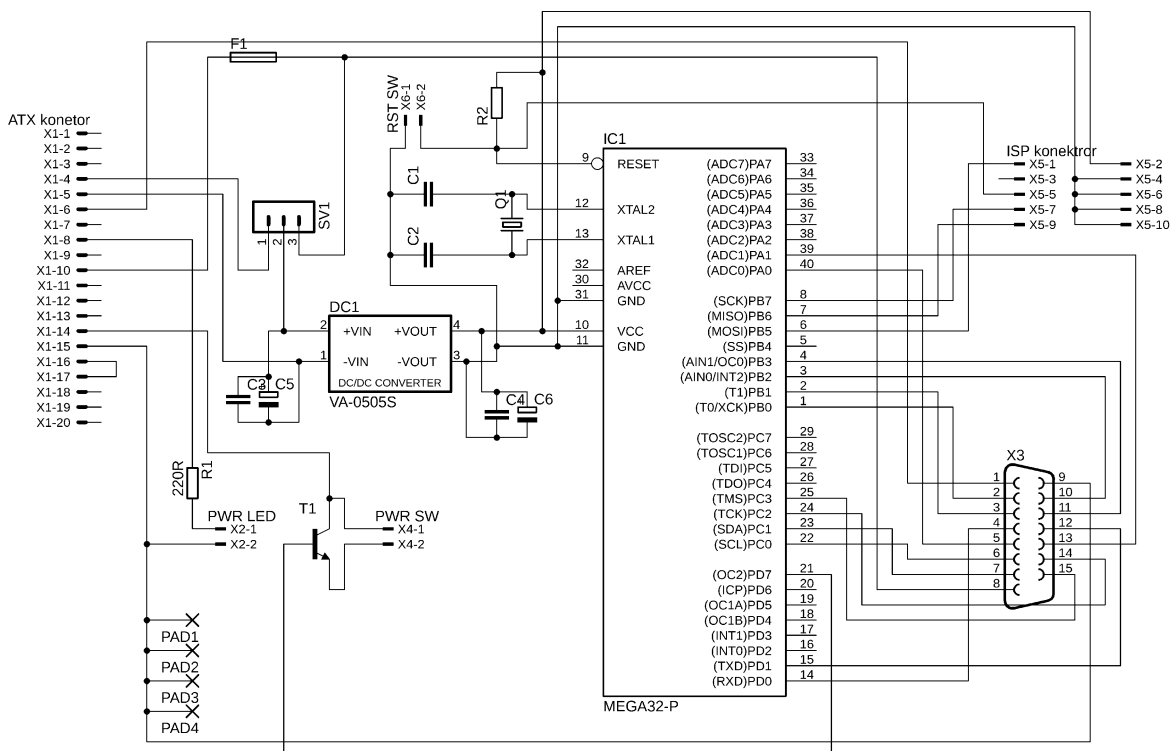
Tabulka 2.3 – Seznam a funkce použitých součástek

Označení	Hodnota	Funkce
R1	220 $\Omega$	omezení proudu do LED diody
R2	10 k $\Omega$	připojení resetu na +5 V
R3	1 $\Omega$	odpor pro měření proudu do motoru
R4	1 k $\Omega$	ochrana výstupů $\mu$ PC
R5	1 k $\Omega$	ochrana výstupů $\mu$ PC
R6	1 k $\Omega$	ochrana výstupů $\mu$ PC
R7	51 $\Omega$	snížení proudu tranzistorem T1
R8	1 k $\Omega$	odpor pro omezení proudu do báze tranzistoru T1
C1	220 pF	filtrace rušení
C2	220 pF	filtrace rušení
C3	100 nF	filtrace rušení
C4	100 nF	filtrace rušení
C5	1 $\mu$ F	filtrace rušení
C6	1 $\mu$ F	filtrace rušení
C7	100 nF	filtrace rušení
Q1	16 MHz	krystal udávající hodinový signál $\mu$ PC
DC1	5 V - 5 V	galvanické oddělení napájení $\mu$ PC
T1	-	ovládání zdroje napětí
D1	-	ochrana proti zpětnému indukovanému napětí z motoru do driveru
D2	-	ochrana proti zpětnému indukovanému napětí z motoru do driveru
D3	-	ochrana proti zpětnému indukovanému napětí z motoru do driveru
D4	-	ochrana proti zpětnému indukovanému napětí z motoru do driveru
IC1	-	mikropočítač ATmega32
IC2	-	driver motorů L298N
PAD1	-	uzemnění šroubu a bedny
PAD2	-	uzemnění šroubu a bedny
PAD3	-	uzemnění šroubu a bedny
PAD4	-	uzemnění šroubu a bedny

Toto zapojení je rozděleno na 2 desky. Pro lepší vyvedení vodičů ze skříně byla navržena ještě třetí deska.

Hlavní deska (obrázek 2.8 a obrázek 2.9) je osazena mikropočítačem ATmega32, DC/DC měničem pro galvanické oddělení obvodů, kondenzátory pro odfiltrování šumu, rezistory pro ochranu vstupů/výstupů a krystalem pro udání frekvence hodinového signálu mikropočítače.

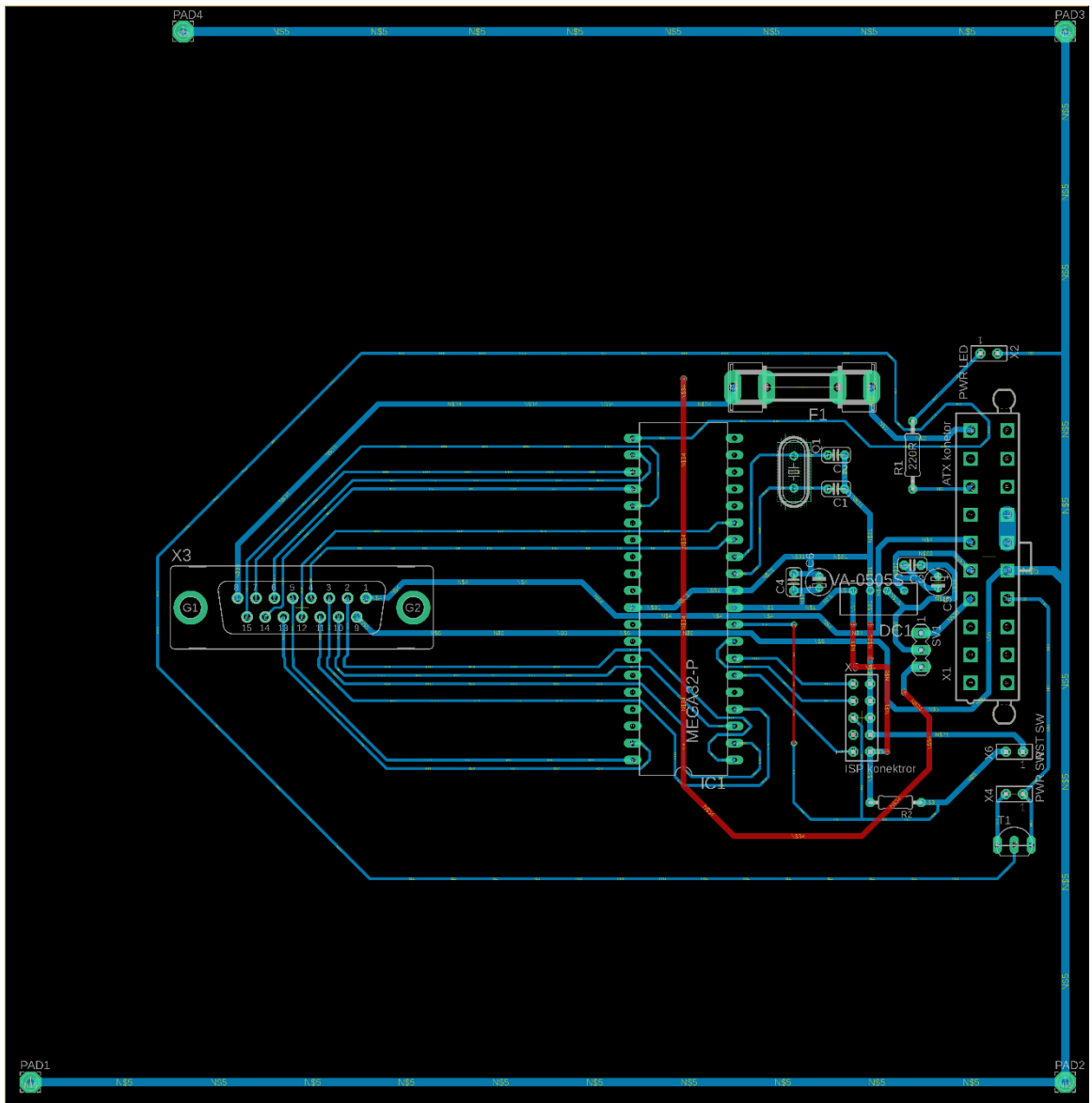
Na desce je umístěno několik konektorů. Hlavními konektory je dvacetipinový ATX konektor, který přivádí napájecí napětí pro logické obvody i motory a vodič pro zapnutí zdroje i vodič oznamující stav zdroje, a patnáctipinový D-SUB konektor, který přenáší napájení a řídicí signály pro driver na druhou desku a přivádí signály z enkodéru. Umístěn je zde i konektor pro SPI rozhraní za účelem možnosti programování mikropočítače, konektor pro větrák, konektory pro tlačítka, konektor pro diodu a čtyři upevňovací otvory pro usazení plošného spoje na místo. Rozložení montážních otvorů je stavěno podle rozměrů základové desky stolního počítače typu mATX a ITX.



Obrázek 2.8 – Schéma hlavní desky

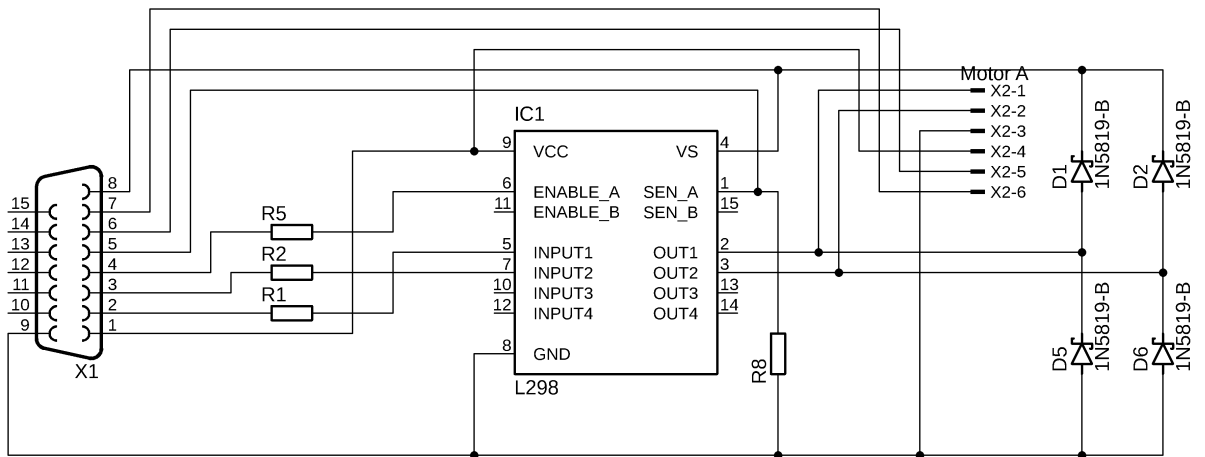
Tabulka 2.4 – Zapojení vývodů konektorů na hlavní desce (Pavouk, nedatováno)

číslo pinu	konektor		
	ATX	D-SUB	SPI
1	+3,3 V	Vss (+12 V)	MOSI
2	+3,3 V	ENABLE	Vcc
3	GND	GND	NC
4	+5 V	Vcc (+5 V)	GND
5	GND	INPUT2	RESET
6	+5 V	INPUT1	GND
7	GND	SENSE	SCK
8	PWR_GOOD	ENC_A	GND
9	+5V STNDBY	ENC_B	MISO
10	+12 V	NC	GND
11	+3,3 V	NC	-
12	-12 V	NC	-
13	GND	NC	-
14	PWR_ON	NC	-
15	GND	NC	-
16	GND	-	-
17	GND	-	-
18	NC	-	-
19	+5 V	-	-
20	+5 V	-	-

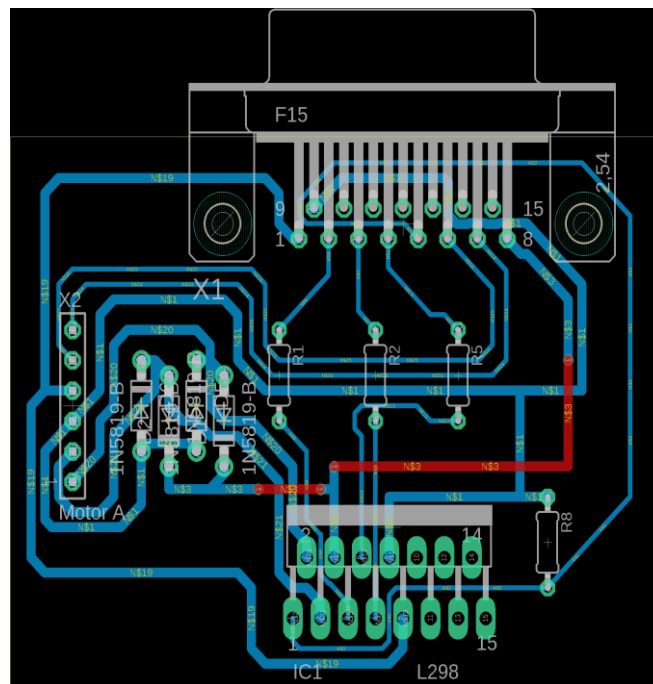


Obrázek 2.9 – Rozvržení plošného spoje hlavní desky

Na další desce (obrázek 2.10 a 2.11) nalezneme pouze driver L298N s ochrannými diodami, konektor se signály z hlavní desky a konektor s vývody přímo pro motor.



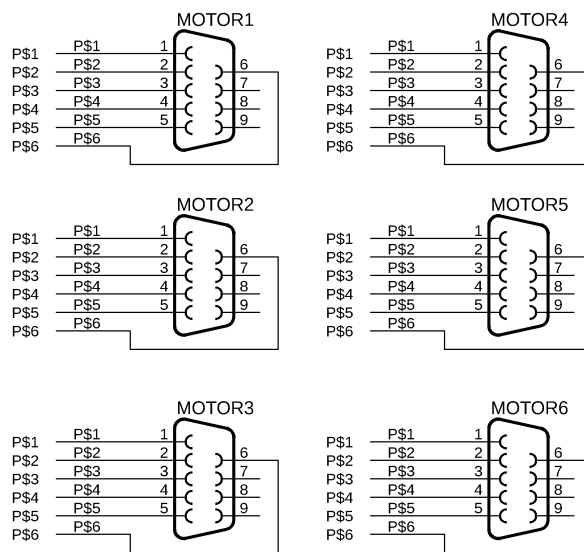
Obrázek 2.10 – Schéma desky s driverem



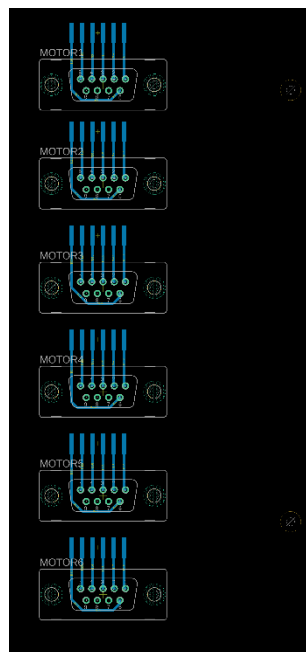
Obrázek 2.11 – Rozvržení plošného spoje s driverem

Poslední, třetí deska je přídatná (obrázek 2.12 a obrázek 2.13), aby netrčely kabely ze skříně, ale byly vyvedeny pomocí konektorů. Tato deska je připravená pro 6 motorů, aby byla možná případná expanze výrobku.

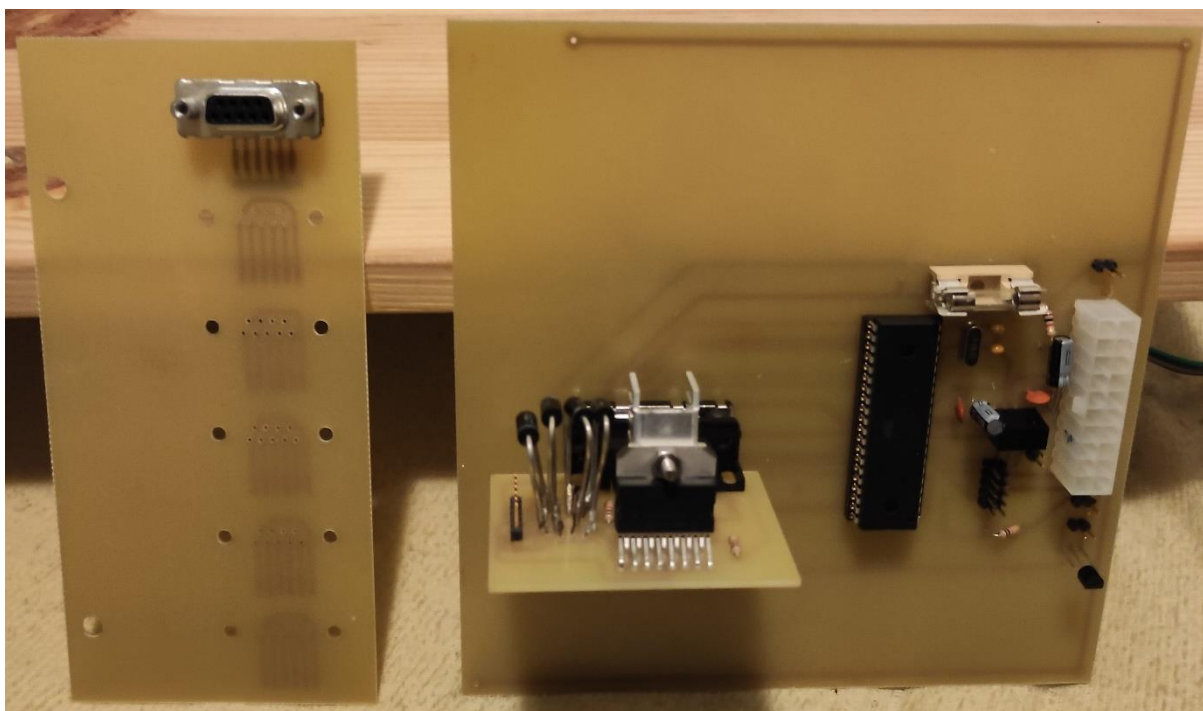




Obrázek 2.12 – Schéma přídavné desky



Obrázek 2.13 – Rozvržení plošného spoje přídavné desky pro lepší vyvedení ze skříně



Obrázek 2.14 – Osazené a připravené desky

## 2.4 NÁVRH A TISK 3D MODELU A PODSTAVY

3D model jsem navrhoval v programu DesignSpark Mechanical. Po navržení v editoru byl každý díl vyexportován ve formátu .stl a převeden pomocí programu Ultimate Cura do G-kódu.

Celý produkt se skládá z 5 dílů, které jsou tisknuty ze dvou materiálů s celkovou dobou tisku všech dílů cca 26,5 hodiny. 2 díly jsou přilepeny přímo k sobě a jeden díl jsem musel pro tisk rozpůlit a po tisku slepit. Po vytisknutí jsem ještě vyhladil dráhy na kuličky a musel jsem trochu zbrousit vnitřní ozubení, protože špatně doléhalo na sebe. K tomu jsem použil brusný papír s hrubostí 240 a 600. Do ložiska jsem použil kvalitnější airsoftové kuličky o průměru 6 mm.

Celý model je připevněn šrouby k dřevěné desce o rozměrech 800 × 600 mm s otvorem a má 4 kovové nohy vysoké kolem 20 cm. Do desky je vyvrtán otvor o průměru 80 mm, díky kterému bude možné pohodlně vyvést kabely spodní částí modelu.

## ZÁVĚR

V tomto projektu se mi úspěšně podařilo vytvořit modulární konstrukci, ovládat motor přes driver L298N s dostatečnou rychlostí otáčení a navrhout, vytisknout a sestavit 3D model kloubu robota.

Zjistil jsem, že je ovládání mikropočítačem nestálé. Občas nastane chyba, která je pokaždé jiná a kvůli tomu se nedá jednoduše odhadnout a odladit. Zřejmě je čip moc pomalý a nestíhá čítat kolem devíti tisíc přerušení za vteřinu a nebyla možnost vyměnit čip za rychlejší, aby bylo splněno zadání, které vysloveně požaduje použití mikropočítače z rodiny ATmega.

## LITERATURA

- 17 Best 3D Printing Software of 2020, 2020 (CAD and Modeling Tools). *Adam Enfroy* [online]. Adam Enfroy Ventures [cit. 2020-07-29]. Dostupné z: <https://www.adamenfroy.com/3d-printing-software>
- 200W ATX PC POWER SUPPLY. *Pavouk* [online]. [cit. 2020-07-28]. Dostupné z: <http://www.pavouk.org/hw/atxps.html>
- Architektura počítačů a mikropočítačů* 2002. [online]. HW server, [cit. 2020-07-27]. Dostupné z: <https://vyvoj.hw.cz/teorie-a-praxe/dokumentace/architektura-pocitacu-a-mikropocitacu.html>
- ATmega32A: megaAVR Data Sheet* 2018. [online]. Microchip Technology [cit. 2020-07-28]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega32A-DataSheet-Complete-DS40002072A.pdf>
- BECHYŇOVÁ, Marta. 14. Programovací jazyky. *Stránky k výuce informatiky* [online]. [cit. 2020-07-28]. Dostupné z: <http://www.ivt.mzf.cz/seminar/14-programovaci-jazyky/>
- BULKA, Miroslav, 2002. Zdroje, zdroje a zase zdroje. *ELEKTRO: časopis pro elektroniku* [online]. FCC Public, **2002**(10) [cit. 2020-07-28]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/zdroje-zdroje-a-zase-zdroje--14729>
- CAROLO, Lucas, 2020. WHAT IS A 3D SLICER?: Simply explained. *All3DP* [online]. [cit. 2020-07-29]. Dostupné z: <https://all3dp.com/2/what-is-a-3d-slicer-simply-explained/>
- ČERNÁ, Monika, 2012. Úvod do paměti počítače: vnitřní paměti. *Metodický portál* [online]. [cit. 2020-07-27]. Dostupné z: <https://clanky.rvp.cz/clanek/a/15075/15065/UVOD-DO-PAMETI-POCITACE-VNITRNI-PAMETI.html/>
- ČERNÁ, Monika, 2012. Úvod do paměti počítače: vnitřní paměti. *Metodický portál inspirace a zkušenosti učitelů* [online]. [cit. 2020-08-19]. Dostupné z: <https://clanky.rvp.cz/clanek/a/15075/15065/UVOD-DO-PAMETI-POCITACE-VNITRNI-PAMETI.html/>
- Čítače* [online]. [cit. 2020-08-19]. Dostupné z: [https://physics.mff.cuni.cz/kfpp/skripta/elektronika/kap6/6\\_7.html](https://physics.mff.cuni.cz/kfpp/skripta/elektronika/kap6/6_7.html)
- D/A a A/D převodníky. *ČVUT* [online]. [cit. 2020-07-27]. Dostupné z: [https://www.fd.cvut.cz/personal/janes/zdt/prednasky/3\\_prevodniky.pdf](https://www.fd.cvut.cz/personal/janes/zdt/prednasky/3_prevodniky.pdf)

- DC Servomotor Controller, 2001. [online]. [cit. 2020-08-19]. Dostupné z: [http://elchan.org/works/smc/report\\_e.html?fbclid=IwAR2fZtgxI7LFeJgEuBZ-yG0u47zPiK00WNw1hShy6AbrKhIO0aidfbQe\\_20](http://elchan.org/works/smc/report_e.html?fbclid=IwAR2fZtgxI7LFeJgEuBZ-yG0u47zPiK00WNw1hShy6AbrKhIO0aidfbQe_20)
- DUDKA, Michal, 2016. I2C – Relativně jednoduše. *Tajned* [online]. [cit. 2020-08-19]. Dostupné z: <http://www.tajned.cz/2016/10/i2c-relativne-jednoduse/>
- DUDKA, Michal, 2016. SPI rozhraní. *Tajned* [online]. [cit. 2020-07-27]. Dostupné z: <http://www.tajned.cz/2016/12/spi-rozhrani/>
- DUDKA, Michal, 2016. USART u AVR 1. Díl – Teoretický popis. *Tajned* [online]. [cit. 2020-08-19]. Dostupné z: <http://www.tajned.cz/2016/09/usart-u-avr-1-dil-teoreticky-popis/>
- DUDKA, Michal, 2017. Začínáme s mikrokontroléry ATmega – A/D převodník 1. díl (teoretický základ). *Tajned* [online]. [cit. 2020-07-27]. Dostupné z: <http://www.tajned.cz/2017/05/zaciname-s-mikrokontrolery-atmega-ad-prevodnik-1-dil-teoreticky-zaklad/>
- ENFROY, Adam, 2020. *17 Best 3D Printing Software of 2020 (CAD and Modeling Tools)* [online]. 2020 [cit. 2020-08-19]. Dostupné z: <https://www.adamenfroy.com/3d-printing-software>
- FABIAN, David. *Úvod do C* [online]. [cit. 2020-07-27]. Dostupné z: <http://kmlinux.fjfi.cvut.cz/~fabiadav/cecko/poznamky-k-jazyku-c/uvod-do-c>
- FRANCESCO, Fassi, 2013. *Comparison between laser scanning and automated 3d modelling techniques to reconstruct complex and extensive cultural heritage areas* [online]. [cit. 2020-07-29]. Dostupné z: [https://www.researchgate.net/publication/235742814\\_Comparison\\_between\\_laser\\_scanning\\_and\\_automated\\_3d\\_modelling\\_techniques\\_to\\_reconstruct\\_complex\\_and\\_extensive\\_cultural\\_heritage\\_areas](https://www.researchgate.net/publication/235742814_Comparison_between_laser_scanning_and_automated_3d_modelling_techniques_to_reconstruct_complex_and_extensive_cultural_heritage_areas)
- G-CODE, jádro 3D tisku. *Vše pro 3D tisk* [online]. [cit. 2020-07-29]. Dostupné z: <https://www.vsepro3dtisk.cz/s/gcode1>
- JAROSLAV, Dvořák, 2008. *Spínaný zdroj* [online]. Brno [cit. 2020-07-28]. Dostupné z: [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=8575](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=8575)
- Jednočipové mikropočítače* [online]. [cit. 2020-07-27]. Dostupné z: [https://www.fd.cvut.cz/personal/janes/ZC/prednasky/prednaska10/HW\\_prednaska\\_10.pdf](https://www.fd.cvut.cz/personal/janes/ZC/prednasky/prednaska10/HW_prednaska_10.pdf)
- Jednočipové mikropočítače* [online]. [cit. 2020-08-19]. Dostupné z: <https://physics.mff.cuni.cz/kfpp/skripta/elektronika/kap9/jednocpoc.html>

- KARAS, Ondřej, 2006. AVR – přerušení. *Programujte* [online] [cit. 2020-07-27]. Dostupné z: <http://programujte.com/clanek/2006092402-avr-preruseni/>
- Komunikace pro sběrnici. *Mendelova Univerzita v Brně* [online]. Brno [cit. 2020-07-27]. Dostupné z: [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=9947](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=9947)
- Komutátory. *Internetová prodejna s elektromotory* [online]. [cit. 2020-07-29]. Dostupné z: <https://www.elektro-motory.cz/komutatory-c2721/>
- KUBIE, Ivan, 2014. *Způsoby řízení stejnosměrných motorů* [online]. [cit. 2020-08-19]. Dostupné z: <https://elektrika.cz/data/clanky/zpusoby-rizeni-stejnosmernihych-motoru>
- Lineární motory, lineární pohyb* [online]. SEW-EURODRIVE [cit. 2020-08-19]. Dostupné z: [https://www.sew-eurodrive.cz/vyrobky/motory/linearni\\_pohyb/linearmotoren.html](https://www.sew-eurodrive.cz/vyrobky/motory/linearni_pohyb/linearmotoren.html)
- MALÉ STEJNOSMĚRNÉ MOTORY MAXON*, 2002 [online]., UZIMEX PRAHA, spol. s r.o. [cit. 2020-08-19]. Dostupné z: [https://www.uzimex.cz/soubory/20070103\\_maxon\\_serial.pdf](https://www.uzimex.cz/soubory/20070103_maxon_serial.pdf)
- MALÝ, Martin, 2014. *Naučte se assembler 8080, Z80, 6502* [online]. [cit. 2020-08-19]. Dostupné z: <https://web.archive.org/web/20140318081547/http://strojak.cz/trocha-assemblerove-teorie/>
- MALÝ, Martin, 2014. *Trocha assemblerové teorie* [online]. [cit. 2020-07-27]. Dostupné z: <https://web.archive.org/web/20140318081547/http://strojak.cz/trocha-assemblerove-teorie/>
- Martin Maly. *Naučte se assembler 8080, Z80, 6502: Možná vám to bude k něčemu dobré!* [online]. [cit. 2020-08-19]. Dostupné z: <https://web.archive.org/web/20140414124839/http://strojak.cz/zakladni-pojmy>
- MEGATRON – komponenty pro automatizaci, snímače polohy, tenzometry, Absolutní a inkrementální enkodéry pro monitoring úhlového natočení. [online]. MEGATRON [cit. 2020-04-10]. Dostupné z: <https://www.megatron.cz/absolutni-inkrementalni-ekodery-pro-vytahy>
- Metal Gearmotor 25Dx66L* [online]. Pololu Corporation [cit. 2020-08-19]. Dostupné z: <https://www.pololu.com/product/3267>
- Mikropočítače: Mikropočítače* [online]. [cit. 2020-07-27]. Dostupné z: <https://physics.mff.cuni.cz/kfpp/skripta/elektronika/kap9/mikropo.html>
- Mikropočítače. *MATEMATICKO-FYZIKÁLNÍ FAKULTA UNIVERZITY KARLOVY: Elektronika* [online]. [cit. 2020-07-27]. Dostupné z: <https://physics.mff.cuni.cz/kfpp/skripta/elektronika/kap9/mikropo.html>

- OLIVKA, Petr, 2010. *Procesory CISC a RISC: Studijní materiál pro předmět Architektury počítačů* [online]. Ostrava [cit. 2020-07-27]. Dostupné z:  
<http://poli.cs.vsb.cz/edu/arp/down/procrisc.pdf>
- PALACKÝ, Petr, 2007. *MIKROPOČÍTAČOVÉ ŘÍDICÍ SYSTÉMY I: učební text* [online]. Ostrava [cit. 2020-07-27]. ISBN 978-80-248-1494-0. Dostupné z:  
[http://www.elearn.vsb.cz/archivcd/FEI/MRS1/Palacky\\_MRS\\_elerning.pdf](http://www.elearn.vsb.cz/archivcd/FEI/MRS1/Palacky_MRS_elerning.pdf)
- Paměti počítače* [online]. [cit. 2020-07-27]. Dostupné z:  
<https://mamut.spseol.cz/nozka/elektro/epo/pameti/>
- Počítačový program (Computer program). *Managment Mania* [online]. [cit. 2020-07-28]. Dostupné z: <https://managementmania.com/cs/pocitacovy-program.pdf>
- Pohyb, pohony, převody / Motory a servomotory / Servomotory* [online]. REM-Technik s.r.o [cit. 2020-08-19]. Dostupné z: <https://www.rem-technik.cz/pohyb-pohony-prevody/motory-a-servomotory/servomotory/>
- PROFESS. *Přehled snímačů úhlového natočení: Snímače úhlového natočení* [online]. [cit. 2020-08-19]. Dostupné z: [https://www.profess.sk/dynamic/produkty/dokument/FSG-Snimace\\_uhloveho\\_natoceni.pdf](https://www.profess.sk/dynamic/produkty/dokument/FSG-Snimace_uhloveho_natoceni.pdf)
- Program Counter (PC), 2017. <https://www.techopedia.com> [online]. Techopedia, [cit. 2020-07-27]. Dostupné z: <https://www.techopedia.com/definition/13114/program-counter-pc>
- REM-technik, Servomotory REM-technik. [online]. [cit. 2020-08-17]. Dostupné z:  
<https://www.rem-technik.cz/pohyb-pohony-prevody/motory-a-servomotory/servomotory/servomotor-ed-176.html>
- Sběrnice pro PC. *Fakulta informatiky Masarykovy univerzity* [online]. [cit. 2020-07-27]. Dostupné z: <https://www.fi.muni.cz/usr/pelikan/ARCHIT/TEXTY/BUSPC.HTML>
- SKALICKÝ, Jiří. *Elektrické servopohony* [online]. Brno [cit. 2020-07-28]. Dostupné z:  
[https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj84eSOwe\\_qAhXITcAKHZWRAU4QFjABegQIAxAB&url=https%3A%2F%2Fwww.vutbr.cz%2Fwww\\_base%2Fpriloha.php%3Fdpid%3D33400&usg=AOvVaw2DdgXx0-OK69WfvANKPUCg](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj84eSOwe_qAhXITcAKHZWRAU4QFjABegQIAxAB&url=https%3A%2F%2Fwww.vutbr.cz%2Fwww_base%2Fpriloha.php%3Fdpid%3D33400&usg=AOvVaw2DdgXx0-OK69WfvANKPUCg)
- SKRYJA, Petr, 2016. *REGULOVATELNÝ LABORATORNÍ ZDROJ* [online]. Brno [cit. 2020-07-28]. Dostupné z:  
[https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=127240](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=127240)
- SPŠ, SOŠ a SOU Hradec Králové, Usměrňovače. [Vyuka.hradebni.cz](http://Vyuka.hradebni.cz) [online]. [cit. 2020-07-01]. Dostupné z: <https://vyuka.hradebni.cz/file.php/168/zdroje/usmernovace.pdf>

- STMicroelectronics, 2000. *L298N Datasheet* [online]. [cit. 2020-08-19]. Dostupné z:  
<https://pdf1.alldatasheet.com/datasheet-pdf/view/22440/STMICROELECTRONICS/L298N.html>
- TIPY&triky 01/12: *Aplikační poznámky k integrovanému H-můstku L298* [online], 2012. [cit. 2020-08-19]. Dostupné z: <http://www.hobbyrobot.cz/wp-content/uploads/TT0112.pdf>
- TIŠNOVSKÝ, Pavel, 2010. *Mikrořadiče a jejich aplikace v jednoduchých mikropočítačích (3)* [online]. [cit. 2020-08-19]. Dostupné z: <https://www.root.cz/clanky/mikroradice-a-jejich-aplikace-v-jednoduchych-mikropocitacich-3/>
- TIŠNOVSKÝ, Pavel, 2011a. Mikroprogramové řadiče a mikroprogramování. *Root.cz* [online]. [cit. 2020-07-27]. Dostupné z: <https://www.root.cz/clanky/mikroprogramove-radice-a-mikroprogramovani/>
- TIŠNOVSKÝ, Pavel, 2011b. *Princip činnosti řadiče v mikroprocesorech* [online]. [cit. 2020-08-19]. Dostupné z: <https://www.root.cz/clanky/princip-cinnosti-radice-v-mikroprocesorech/>
- TKOTZ, Klaus, 2006. Příručka pro elektrotechnika. 2., dopl. vyd. Praha: Europa-Sobotáles,, ISBN 80-86706-13-3.
- Usměrňovače. *Vyuka.hradebni.cz* [online]. [cit. 2020-07-28]. Dostupné z:  
<https://vyuka.hradebni.cz/file.php/168/zdroje/usmernovace.pdf>
- Úvod do jazyka C. *Matematická sekce: Matematicko-fyzikální fakulta Univerzita Karlova* [online]. [cit. 2020-07-27]. Dostupné z:  
[https://www2.karlin.mff.cuni.cz/~halas/Pyth/Dokumentace/uvod\\_do\\_jazyka\\_c.pdf](https://www2.karlin.mff.cuni.cz/~halas/Pyth/Dokumentace/uvod_do_jazyka_c.pdf)
- VLČEK, Jiří, 2007. *Měniče napětí a spínané zdroje - teoretická základna* [online]. [cit. 2020-07-28]. Dostupné z: <https://elektro.tzb-info.cz/teorie-elektrotechnika/4263-menice-napeti-a-spinane-zdroje-teoreticka-zakladna>
- WHAT ARE 3D PRINTED MATERIALS AND HOW ARE THEY USED?* [online]. [cit. 2020-08-19]. Dostupné z: <https://www.makerbot.com/stories/design/3d-printing-materials/>
- What is 3D printing? *Online 3D Printing Service* [online]. [cit. 2020-07-29]. Dostupné z:  
<https://i.materialise.com/en/learn/what-is-3d-printing>
- Základní post-processing vytištěných modelů. *MarkerBot* [online]. [cit. 2020-07-29]. Dostupné z: <https://www.vsepro3dtisk.cz/s/post-processing>
- ŽOLTA, Lucka. *Základní vlastnosti monolitických počítačů a jejich typické integrované periférie*. [online]. [cit. 2020-08-19]. Dostupné z:  
<http://lucie.zolta.cz/index.php/statnice-vs/175-zakladni-vlastnosti-monolitickyh-pocitacu-a-jejich-typicke-integrované-periferie>



**PŘÍLOHY**  
**A – CD**

**Příloha k bakalářské práci**  
Servopohon s DC motorem  
Filip Bellinger

**CD**

## **OBSAH**

1. Text bakalářské práce ve formátu PDF
2. Schémata zapojení a plošných desek
3. Program mikropočítače
4. Technická dokumentace 3D modelu
5. 3D model
6. Video s ukázkou výsledné činnosti