

Rozhraní pro připojení výukového robota k vývojovému kitu M68EVB908GB60

Interface for Educational Robot Connection to Evaluation Board
M68EVB908GB60

Josef Doležel

Bakalářská práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Josef DOLEŽEL**
Osobní číslo: **A10011**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Rozhraní pro připojení výukového robota k vývojovému kitu M68EVB908GB60**

Zásady pro vypracování:

1. Zpracujte literární rešerši na téma modelářské servopohony a jejich řízení.
2. Prostudujte hardwarové vlastnosti vývojového kitu M68EVB908GB60 a navrhnete způsob připojení modulu rozhraní robota.
3. Provedte hardwarový návrh modulu rozhraní pro výukového robota Lynx 5 zajišťující kontrolu PWM řídicích signálů generovaných vývojovým kitem.
4. Realizujte prototyp modulu rozhraní, včetně jeho řídicího programového vybavení.
5. Vytvořte podpůrné programové vybavení pro obsluhu robota ve formě knihovny podprogramů v C jazyce.
6. Ověřte funkci rozhraní na vývojovém kitu M68EVB908GB60 a vytvořte ukázkové programové vybavení demonstrující funkci robota.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. AXIOM MANUFACTURING. M68EVB908GB60 Development Board for Freescale MC9S08GB60, Rev. C [online]. 2006. Dostupné z: [www.axman.com]
2. FREESCALE SEMICONDUCTOR. CPU08 Central Processor Unit Reference Manual [online]. 2001. Dostupné z: [www.freescale.com]
3. FREESCALE SEMICONDUCTOR. HCS08 Family Reference Manual, Rev.1. [online]. 2003. Dostupné z: [www.freescale.com]
4. FREESCALE SEMICONDUCTOR. MC9S08GB/GT Data Sheet, Rev.2.3. [online]. 2004. Dostupné z: [www.freescale.com]
5. JURÁNEK, Antonín a Miroslav HRABOVSKÝ. EAGLE pro začátečníky /: uživatelská a referenční příručka :. 2. vydání. Praha: BEN – technická literatura, 2007, 191 s. ISBN 80-730-0213-2.
6. NOVÁK, Petr a Miroslav HRABOVSKÝ. Mobilní roboty: pohony, senzory, řízení. Vyd. 1. Praha: BEN – technická literatura, 2004, 191 s. ISBN 80-730-0141-1.

Vedoucí bakalářské práce:

Ing. Petr Dostálek, Ph.D.

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

24. února 2013

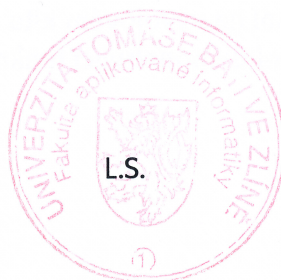
Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013

prof. Ing. Vladimír Vašek, CSc.

děkan



prof. Ing. Vladimír Vašek, CSc.

ředitel ústavu

ABSTRAKT

Cílem této práce je navrhnout a realizovat modul rozhraní pro připojení výukového robota Lynx 5 zajišťující kontrolu PWM řídicích signálů generovaných vývojovým kitem M68EVB908GB60. Návrh a realizace proběhne na základě zjištěných hardwarových vlastností tohoto vývojového kitu a robota. V teoretické části je vysvětlena PWM modulace a princip jednotlivých částí servomotoru. Dále je v teoretické části uveden základní popis vývojového kitu a robota. Stručný popis použitého mikropočítače, budiče sběrnice, programátoru, softwaru pro návrh desky plošného spoje a vývojového prostředí je uveden v praktické části. Praktická část se věnuje jednotlivým částem modulu rozhraní, jeho výrobě, tvorbě řídicího softwaru a ověření správné funkce rozhraní na vývojovém kitu vytvořením ukázkového programu pro řízení robota.

Klíčová slova: PWM, modelářské servopohony, vývojový kit, rozhraní, robot, Lynx 5, mikropočítač, Freescale

ABSTRACT

The goal of this bachelor's thesis is design and implement interface module for connect education robot Lynx 5 which controls the PWM signals generated development board M68EVB908GB60. Design and implementation will be done according to identified properties of this development board and robot. In the theoretical part is explained pulse-width modulation, principle of the servos, basic description of development board and robot. A brief description of the microcontroller in interface module, bus transceiver, programmer, PCB design software and development environment is given in the practical part. The practical part describes production of individual parts of the interface module, firmware for interface module, demonstration program for development board which controls the movement of the robot and verifies the correct function of the interface module.

Keywords: PWM, model servos, development board, interface, robot, Lynx 5, microcontroller, Freescale

Rád bych poděkoval panu Ing. Petru Dostálkovi, Ph.D. za jeho čas strávený nad touto prací, ochotu a rady, které mi velmi pomohly při řešení problémů, které se v průběhu práce vyskytly. Dále bych rád poděkoval rodině a přátelům, kteří mě po celou dobu podporovali.

„Představivost je důležitější než znalosti. Znalosti jsou omezené, ale představivost obklopuje celý svět.“

Albert Einstein

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....

podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PULSNĚ ŠÍŘKOVÁ MODULACE (PWM)	11
2 MODELÁŘSKÉ SERVOPOHONY A JEJICH ŘÍZENÍ	13
2.1 PRINCIP ČINNOSTI STEJNOSMĚRNÉHO MOTORU	13
2.2 ROZDĚLENÍ STEJNOSMĚRNÝCH MOTORŮ PODLE BUZENÍ (ZAPOJENÍ) VINUTÍ.....	15
2.3 MOTORY S CIZÍM BUZENÍM (S PERMANENTNÍMI MAGNETY)	18
2.4 ŘÍZENÍ OTÁČEK STEJNOSMĚRNÉHO MOTORU.....	20
2.5 SLOŽENÍ A PRINCIP MODELÁŘSKÉHO SERVOPOHONU	23
3 ROBOT LYNX 5	26
4 VÝVOJOVÝ KIT M68EVB908GB60	28
II PRAKTICKÁ ČÁST	31
5 POUŽITÝ MIKROPOČÍTAČ	32
6 BUDIČ SBĚRNICE 74HCT245N	35
7 NÁVRH A VÝROBA HARDWARE	37
7.1 BLOKOVÉ SCHÉMA	37
7.2 POPIS JEDNOTLIVÝCH ČÁSTÍ	38
7.2.1 Napájení	38
7.2.2 Zdroj hodinového signálu.....	39
7.2.3 BDM a tlačítko Reset	40
7.2.4 Zapojení budiče sběrnice.....	40
7.2.5 Připojení tlačítka pro volbu režimu ovládání	41
7.2.6 Připojení potenciometrů k AD převodníkům	42
7.2.7 Připojení LED diod	42
7.3 NÁVRH DESKY PLOŠNÉHO SPOJE V PROGRAMU EAGLE	43
7.4 VÝROBA DESKY PLOŠNÉHO SPOJE.....	44
7.5 POPIS OSAZENÉ DESKY MODULU ROZHRANÍ.....	44
8 PROGRAMOVÉ VYBAVENÍ A OVĚŘENÍ FUNKCE	48
8.1 VÝVOJOVÉ PROSTŘEDÍ	48
8.2 POPIS FIRMWARE V MODULU ROZHRANÍ	49
8.2.1 Hlavní funkce main	49
8.2.2 Obsluha přerušení při stisku tlačítka	51
8.2.3 Obsluha přerušení při přetečení časovače	52
8.2.4 Obsluha přerušení při zachycení vstupu.....	53

8.3	POPIS KNIHOVNY PRO PRÁCI S PWM A ROBOTEM.....	54
8.4	UKÁZKOVÝ PROGRAM PRO VÝVOJOVÝ KIT	59
8.4.1	Zadání příkladu	59
8.4.2	Řešení příkladu – funkce main.....	60
8.4.3	Řešení příkladu – obsluha přerušení při stisku tlačítka.....	63
8.4.4	Řešení příkladu – obsluha přerušení při přetečení časovače.....	64
8.5	OVĚŘENÍ FUNKCE MODULU A KNIHOVNY NA UKÁZKOVÉM PROGRAMU.....	65
ZÁVĚR		67
CONCLUSION		68
SEZNAM POUŽITÉ LITERATURY.....		69
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		71
SEZNAM OBRÁZKŮ		72
SEZNAM TABULEK.....		74
SEZNAM PŘÍLOH.....		75

ÚVOD

Průmyslové roboty mají široké uplatnění v automatizované výrobě. Používají se většinou k manipulaci s materiálem či rozpracovanými výrobky. Po připevnění vhodného nástroje lze robota použít k dalším činnostem jako je například měření, vrtání, leštění, svařování či lepení. Tyto roboty se používají hlavně v sériové výrobě, kde se pořizovací náklady rychle vrátí.

Robot Lynx 5 je zmenšeným modelem skutečného průmyslového robota s pěti stupni volnosti. Skládá se celkem z šesti servopohonů. Dva servopohony, které slouží pro ovládání ramene jsou zapojeny paralelně. Poloha každého z těchto servopohonů se řídí PWM signálem, kde šířka pulsu určuje úhel natočení servopohonu.

Cílem této bakalářské práce je vytvořit novou úlohu, na které by se studenti prakticky seznámili s řízením servomotorů pomocí PWM signálu a s modelem průmyslového robota. Student bude mít možnost řídit robota pomocí vývojového kitu, který se běžně používá k výuce v předmětu Programování mikropočítačů.

Mezi robotem a vývojovým kitem bude modul rozhraní, který provede kontrolu všech pěti PWM řídicích signálů, aby nedošlo k poškození jednotlivých servomotorů a kontrolu polohy celého robota aby nedošlo k poškození robota vlivem nárazu do podložky. Kromě kontroly signálů zajišťuje modul rozhraní také napájení robota, převod z 3,3V logiky, kterou používá vývojový kit na 5V logiku, se kterou pracují servopohony a umožňuje manuální řízení robota pomocí potenciometrů. Student tak bude mít možnost si robota vyzkoušet ještě předtím, než začne programovat.

Součástí práce je i knihovna funkcí pro vývojový kit napsaná v jazyce C. Tato knihovna umožní pomocí zavolání několika funkcí pohodlné a rychlé vytvoření programu pro ovládání robota bez nutnosti znát princip řízení servopohonů pomocí PWM signálu.

I. TEORETICKÁ ČÁST

1 PULSNĚ ŠÍŘKOVÁ MODULACE (PWM)

Pulsně šířková modulace (dále jen PWM, angl. Pulse Width Modulation - zkr.PWM) je způsob kódování dat, která se přenášejí z vysílajícího zařízení k přijímacímu zařízení zvolenou přenosovou cestou. Přenosovou cestou může být pevné spojení pomocí metalického vedení nebo bezdrátové spojení (např. při IR komunikaci) [1].

Princip PWM

Jde tedy o signál s konstantní periodou T , kde se mění střída (tj. poměr délky impulzu ku délce mezery uvažovaný v jedné periodě). Střída se uvádí někdy jako poměr (1:1, 2:1, 1:5 atd.), kdy je nutné uvést které číslo představuje impulz a které mezeru. Někdy se střída vyjadřuje procentuálně (100%, 50%, 0.1% atd.), kde 100% představuje ideální poměr 1:0, 50% poměr 1:1 atd. Poměr délky impulzu k délce mezery bývá v zahraniční literatuře nazýván Duty Cycle [1].

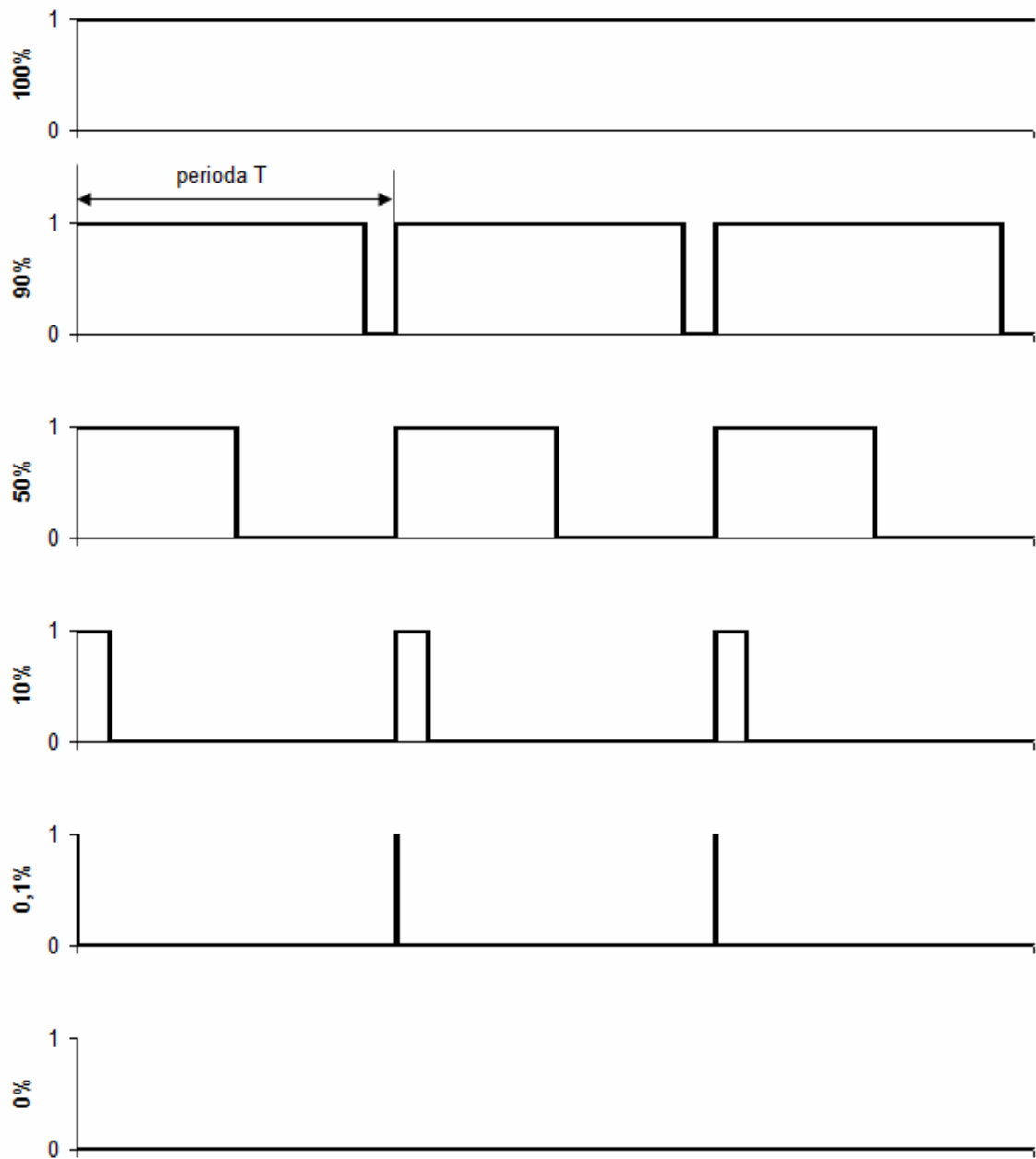
Využití PWM

Pulsně-šířková modulace PWM je hojně používaný způsob generování analogových signálů digitálními obvody. Obzvláště ve výkonové elektronice je oceňováno, že výstupní řídicí prvek rozeznává pouze dva stavy:

- sepnuto - minimální úbytek napětí - minimální ztrátový výkon
- rozepnuto - minimální zbytkový proud - minimální ztrátový výkon

Vlastní generování PWM je pak hardwarově velmi nenáročné, stačí prakticky pouze jeden čítač a jeden komparátor [2].

Střída Průběh napětí



Obr. 1 Průběh napětí při různých střídacích PWM modulace

2 MODELÁŘSKÉ SERVOPOHONY A JEJICH ŘÍZENÍ

V této kapitole je popsána funkce modelářských servopohonů a možnosti jejich řízení. Modelářské servo se většinou skládá ze stejnosměrného motoru, převodovky a řídicí elektroniky. Nejdříve bude popsán princip činnosti stejnosměrných motorů a jejich rozdělení podle zapojení vinutí. Další část se věnuje popisu motoru s cizím buzením (s permanentními magnety), který je velmi často hlavní součástí modelářských servopohonů. Následuje popis možností, jak řídit otáčky u stejnosměrných motorů a funkce převodovky. Na závěr je popsáno složení modelářského serva, funkce vnitřní elektroniky a signály, kterými lze modelářské servopohony řídit.

2.1 Princip činnosti stejnosměrného motoru

Pro pochopení principu stejnosměrného motoru bude na začátku uvedeno několik informací:

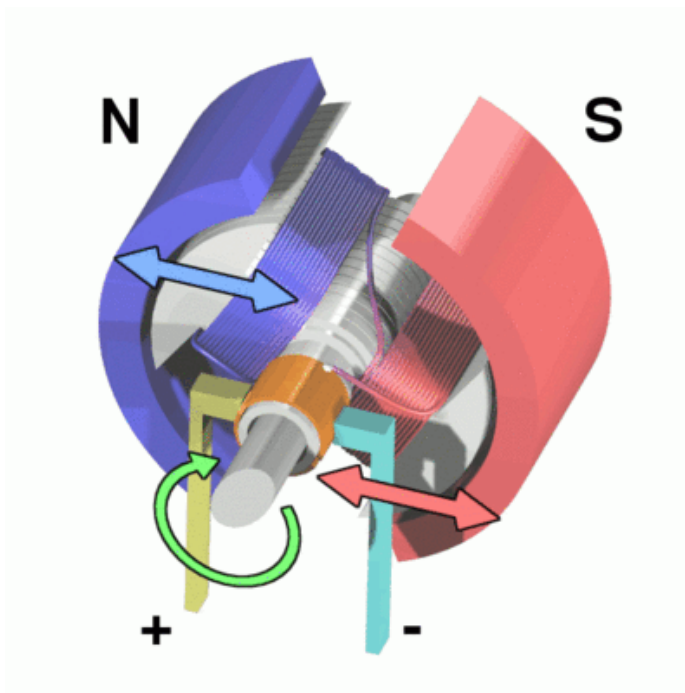
1. Kolem každého vodiče, kterým protéká elektrický proud, vzniká magnetické pole. Čím více vodičů je, tím bude magnetické pole silnější.
2. Každý magnet má dva póly (S a N). Stejně póly se odpuzují, opačné přitahují [3].

Stejnosměrný motor se skládá ze statoru (pevná, nepohyblivá část stroje), na kterém jsou umístěny hlavní póly s budícím vinutím a pomocné póly umístěné mezi hlavními póly pro zlepšení komutačních vlastností.

Rotor (pohyblivá část, nazývaná kotvou) otáčející se v magnetickém poli je složen z plechů v jehož drážkách je umístěno vinutí. Jednotlivé cívky vinutí rotoru jsou připojeny k měděným vzájemně izolovaným lamelám komutátoru. Na komutátor dosedají kartáče, umístěné ve speciálních držácích, jimiž se přivádí proud do vinutí rotoru.

Mechanický komutátor zajišťuje optimální polohu magnetických polí v motoru při každém zatížení a rychlosti, neboť řídí napájení cívek rotoru podle jeho polohy. Motor se rozběhne při pouhém připojení stejnosměrného napětí. Při otáčení rotoru se napětí na kartáčích komutátoru připojuje k těm cívkám rotoru, které vytvářejí magnetické pole směru optimálního pro vznik krouticího momentu. Největší proud teče do cívek rotoru v klidu, kdy tudíž na rotor působí největší magnetické síly a motor vyvozuje velký záběrný moment. Naproti tomu v rychle se otáčejícím rotoru se indukuje napětí zmenšující proud

tekoucí do cívek rotoru a krouticí moment s rostoucími otáčkami klesá. Motor s touto momentovou charakteristikou snadno překonává proměnné zatížení [4].



Obr. 2 Princip stejnosměrného motoru [3]

Modře je označen severní pól magnetu a konec vinutí připojené na záporný potenciál zdroje. Červeně je označen jižní pól magnetu a konec vinutí připojené na kladný potenciál zdroje. Oranžově je vyznačen komutátor, zelená šipka značí směr otáčení a modrá s červenou šipkou znázorňují odpuzivé síly mezi souhlasně polarizovanými permanentními magnety a elektromagnety.

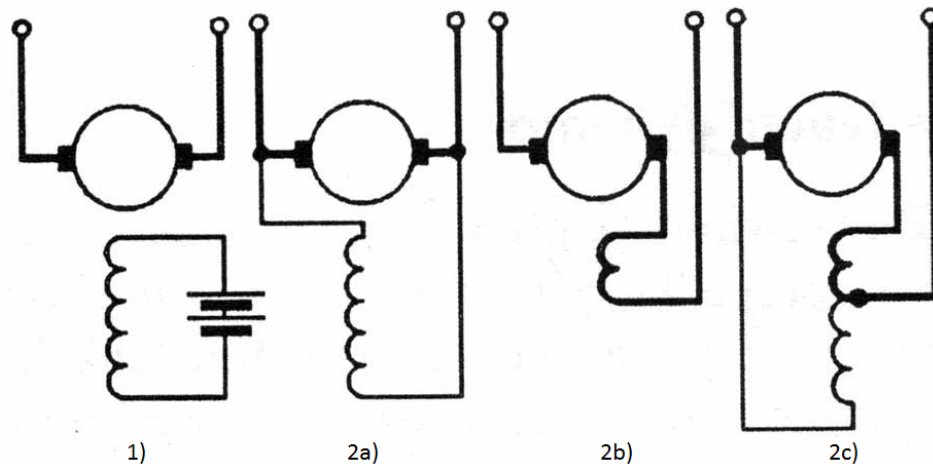
Už při pohledu na konstrukci klasického stejnosměrného motoru je jasně vidět, že se nejedná o žádnou složitou konstrukci, prostě dva magnety a mezi nimi dvě vinutí. K tomu jen komutátor, který je vyroben jako dvě třecí plochy, které pootáčením mění směr procházejícího proudu. Takže výhoda je jasná – jednoduchá výroba = nízká cena.

Nevýhodou je menší životnost způsobená právě přítomností komutátoru. Uhlíky, které zajišťují kontakt mezi přívodními vodiči a rotorem se po nějakém čase opotřebují a motor je nutné vyměnit. Existují i motory, kde se dají uhlíky měnit, ale jejich cena je poměrně vysoká a tudíž nad nimi nemá moc smysl uvažovat. Navíc tyto stejnosměrné motory nemají moc velkou účinnost, protože velká část příchozí energie se mění v teplo, které může být samo o sobě v uzavřeném modelu problémem. Je tedy nutné myslet i na chlazení motoru [3].

2.2 Rozdělení stejnosměrných motorů podle buzení (zapojení) vinutí

Podle způsobu buzení magnetického pole se stejnosměrné motory dělí na:

1. Stejnosměrné motory s cizím buzením – buzení vinutí umístěné ve statoru je napájeno elektrickou energií ze samostatného zdroje nebo motor využívá pevných magnetů s tvarově příslušnými pólovými nástavci umístěnými jako nedílná součást statoru. Rotorové vinutí je napájeno z jiného zdroje .
2. Stejnosměrné motory s vlastním buzením – buzení vinutí je určitým způsobem spojeno s vinutím rotoru – je tedy vinutí statoru i rotoru napájeno z jediného společného zdroje – podle druhu propojení rozlišujeme:
 - a) buzení derivační – vinutí rotoru a statoru jsou zapojeny paralelně a prochází jimi proud rozdílné velikosti.
 - b) buzení sériové – vinutí rotoru a statoru jsou zapojeny do série a prochází jimi proud shodné velikosti.
 - c) buzení kompaundní (sdružené) – kombinace obou předcházejících, kdy magnetizační statorové vinutí je složeno z (minimálně) dvou cívek, z nichž jedna je v sérii s rotorovým vinutím a druhá paralelně [5].



Obr. 3 Rozdělení stejnosměrných motorů podle zapojení vinutí [6]

Motory s cizím buzením

Vinutí statoru je připojeno na jiný zdroj el. energie než rotorové vinutí. Tím je zaručeno stálé buzení a tedy velikost statorového elektromagnetického pole v prostoru rotoru. Rotorové magnetické pole je závislé na napájení rotoru a není se statorovým polem přímo svázáno a proto lze tento typ motoru nejsnáze regulovat. Otáčky rotoru pak závisí pouze na velikosti napětí přivedeného na rotorové cívký. Moment závisí na proudu, který rotorem prochází. U velkých motorů se ke statorovému vinutí sériově přidává odpor, jehož velikostí lze regulovat rozběh motoru (ovlivněním velikosti magnetického pole).

Otáčková i momentová charakteristika jsou „tvrdé“ – otáčky se zátěží prakticky neklesají. Moment roste konstantně až do stavu nasycení.

Změnu smyslu otáčení lze realizovat velice snadno a to pouhým přepólováním (přepojením přívodu) konců statorového budicího vinutí v zapojení. Tím se změní smysl statorového budicího magnetického toku [5].

Motory se sériovým buzením

Vinutí statoru je připojeno sériově na stejný zdroj el. energie jako je rotorové vinutí. Tím je svázáno buzení statorového magnetického pole a tedy jeho velikost v prostoru rotoru s rotorovým magnetickým polem. Vlastnosti při rozběhu lze řídit sériovým odporem v kotvě motoru. Moment závisí na proudu, který rotorem prochází.

Otáčky se řídí dobře pomocí odporu zapojeného sériově k vinutím statoru a rotoru - zvětšováním odporu se zeslabuje statorové magnetické pole. Při rozběhu musí mít sériový odpor nulovou hodnotu. Otáčky při zvětšující se zátěži klesají.

Změnu smyslu otáčení lze realizovat velice snadno a to pouhým přepólováním (přepojením přívodu) konců statorového budicího vinutí v zapojení. Tím se změní smysl statorového budicího magnetického toku [5].

Motory s paralelním (derivačním) buzením

Vinutí statoru je připojeno paralelně na stejný zdroj el. energie jako je rotorové vinutí. Tím je svázáno buzení statorového a rotorového magnetického pole a tedy velikost elektromagnetického pole v prostoru rotoru. Otáčky rotoru závisí na velikosti napětí přivedeného na rotorové cívký. Moment závisí na proudu, který rotorem prochází a je velice stálý. Pokud není statorový magnetický tok ve stavu nasyceném (jmenovitém – tj. při

nižším napájecím napětí) je otáčkovou charakteristikou hyperbola a momentovou charakteristikou parabola (moment je závislý na kvadrátu procházejícího proudu). Otáčky se zatížením tedy následně klesají – otáčková charakteristika je „měkká“. V nasyceném stavu již otáčky se zatížením téměř neklesají a momentová charakteristika se napřimuje. Záběrový moment je veliký a proto se používá k pohonům s velmi častými rozběhy a to rozběhy s velkou zátěží (např. jeřáby - dělicí stroje, dopravní pásy , atd.).

Otáčky se řídí dobře pomocí odporu zapojeného sériově k budicímu vinutím statoru čímž se zeslabuje statorové magnetické pole. Sériovým odporem lze otáčky pouze snižovat.

Tyto derivační motory se nesmí použít tam, kde by se mohla při chodu odpojit zátěž. Vlivem tvaru charakteristik by se motor roztočil „nade všechny meze (nebo alespoň na hodnotu otáček, které by mohly ohrozit mechanickou stabilitu rotujících částí)“ a mohlo by dojít k mechanické havárii rotoru. Zátěž musí být připojena přes pevnou spojku nebo zubový převod [5].

Motory se smíšeným (kompaundním) buzením

Vinutí statoru je rozděleno na dvě části - připojeno sériově a paralelně na stejný zdroj el. energie jako je rotorové vinutí. Tím je buzení statorového magnetického pole a tedy jeho velikost v prostoru rotoru vázáno na obě budicí části. Podle toho které z budicích vinutí působí „silněji“ jsou určeny charakteristické vlastnosti tohoto typu motoru. Moment i otáčky mají charakter bližší sériovému motoru.

Působí-li obě části budicího vinutí souhlasně, má motor větší záběrový moment než derivační a otáčky tolik neklesají. Nebezpečí vysokých otáček při odlehčení ale není malé.

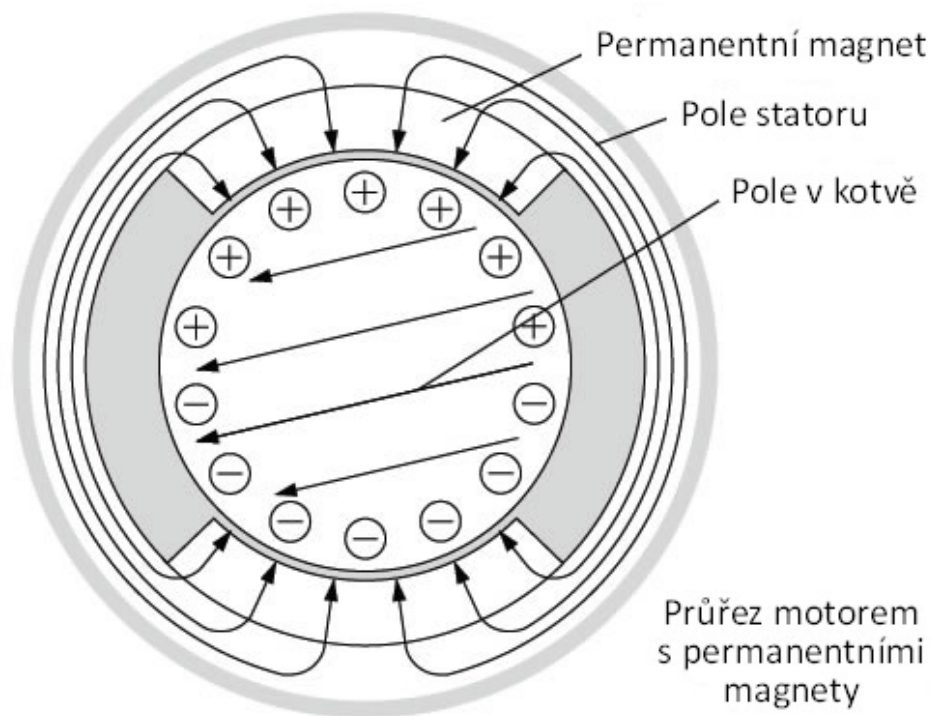
Působí-li obě části vinutí proti sobě, má motor velice tvrdé otáčky při změnách zatížení. Přičemž při vzrůstu zatížení otáčky klesají a protože tím pádem sériovou částí vinutí prochází větší proud a to zeslabí buzení – výsledkem je že otáčky opět o trochu stoupnou. Nebezpečí vysokých otáček při odlehčení stoupá.

Otáčky se řídí dobře pomocí odporu zapojeného paralelně k budicímu vinutím statoru čímž se zeslabuje statorové magnetické pole. Tyto sériové motory se nesmí použít tam, kde by se mohla při chodu odpojit zátěž. Vlivem tvaru charakteristik by se motor roztočil mimo povolené meze a mohlo by dojít k mechanické havárii rotoru. Zátěž musí být připojena přes pevnou spojku nebo zubový převod [5].

2.3 Motory s cizím buzením (s permanentními magnety)

Pro pohony v automatizaci a v modelářských servopohonech se dnes velmi často používá stejnosměrný motor s cizím buzením (s permanentními magnety) [4].

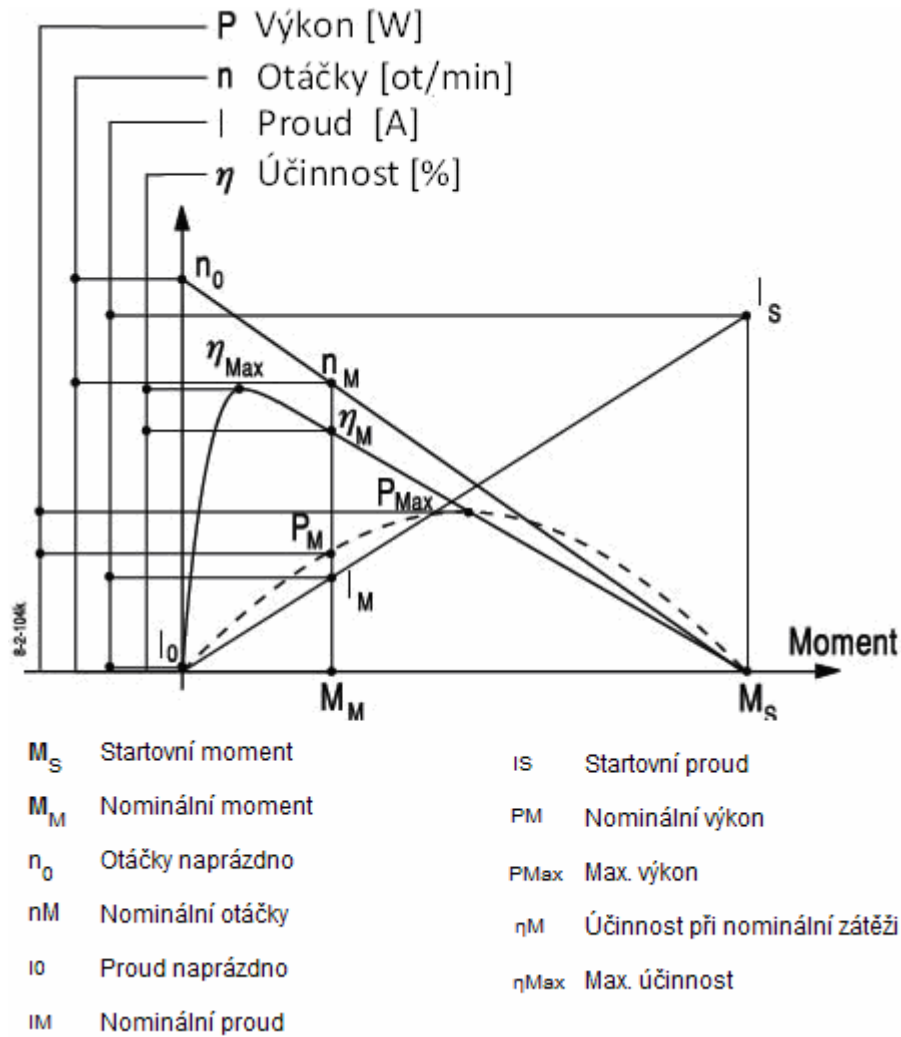
Pole permanentně magnetizovaného motoru (PM) se tvoří mezi dvěma nebo více trvalými magnety, tolika jako je počet pólů. Nejčastější jsou různé typy magnetů feritových. Vzhledem k tomu, PM motor nemá vinutí pro tvorbu pole. Pro připojení motoru stačí dva vodiče. Charakteristika PM vykazuje vlastnosti které jsou výhodné, což vedlo k tomu, že tento typ motoru se stal dominantním mezi stejnosměrnými motory s výkonem do 1 kW [7].



Obr. 4 Průřez motorem s permanentními magnety [7].

Nejvýznamnější vlastností PM motoru je, že při konstantním napětí, jsou otáčky a proud úměrné momentu. Je tedy jednoduché provádět výpočty a navrhnout jak motor tak i ostatní systémové komponenty.

Je také snadné měřit otáčky a točivý moment za provozu. Následující diagram popisuje charakteristiku PM motoru při konstantním napětí [7].

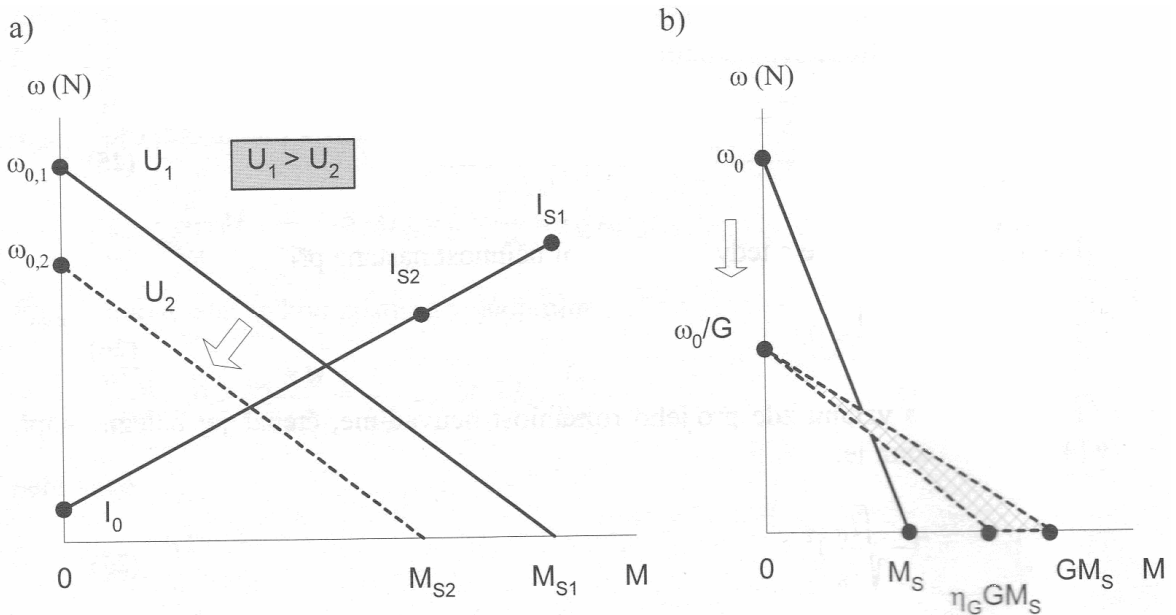


Obr. 5 Charakteristika permanentně magnetizovaného motoru [7]

Údaje, které jsou důležité, je rezistance R_A , indukčnost L a setrvačnost J_M . Můžete si všimnout, že rozběhový moment je obvykle 6-8 krát vyšší než nominální točivý moment. Vzhledem k tomu, že jmenovitý výkon je omezen tepelnými vlastnostmi, může být při občasných zatíženích použit motor s nižším výkonem než je požadováno. Například na 25% pracovní cyklus (např. 1 minuta práce následována 3 minutami klidu) používat motor, jehož jmenovitý výkon je poloviční než je požadováno. To samozřejmě předpokládá, že je třeba vzít v úvahu další klíčové charakteristiky, jako je životnost, otáčky motoru v závislosti na zatížení motoru, teplota, atd. Pokud je vysoký startovní moment u PM problémem, lze použít proudové omezení při jeho rozběhu [7].

2.4 Řízení otáček stejnosměrného motoru

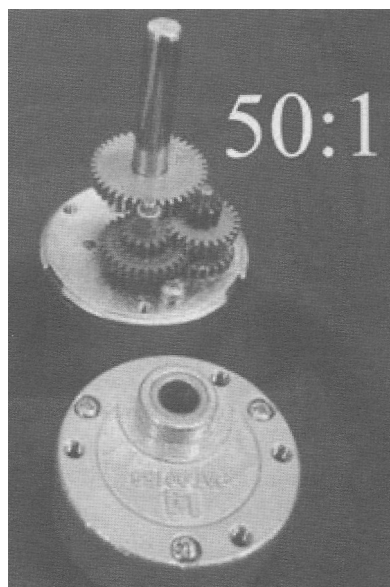
Stejnoseměrné motory mívají jmenovité otáčky v řádu jednotek tisíců za minutu, což je pro využití v robotice a automatizaci často velmi nepraktické. Otáčky můžeme snížit velikostí napájecího napětí, ale zároveň se snižuje i moment. Pokud je potřeba snížit otáčky motoru při současném zvýšení jeho momentu, což je v robotice častý požadavek, je nutno použít mechanickou převodovku [8].



Obr. 6 Změna otáček motoru vlivem napětí a pomocí převodovky [8]

Obr. 9 znázorňuje:

- vliv napětí na otáčky – kromě snížení otáček dochází v tomto případě také ke snížení momentu M_S .
- změnu (snížení) otáček pomocí převodovky s převodovým poměrem G . V tomto případě dojde ke snížení otáček z ω_0 na ω_0/G a k zvýšení momentu M_S na hodnotu $G M_S$, respektive na poněkud nižší reálnou hodnotu $\eta_G G M_S$, kde η_G je účinnost převodovky [8].



Obr. 7 Odkrytovaná převodovka [8]

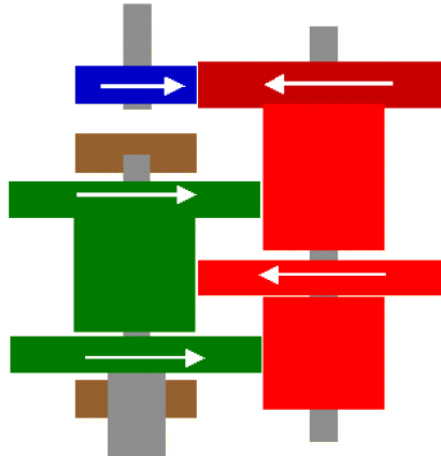
Převodovky se vyrábějí jako planetové s vysokým přenášeným mechanickým momentem nebo jako převodovky s předlohou. Pro nízké momenty existují levné varianty obou typů s plastovými koly. Postupně se zavádí výroba planetových převodovek s vysoce odolnými keramickými čepy.

Uspořádání převodovek ukazují obr. 8 a obr. 9. Převodovka s předlohou využívá k přenosu momentu v každém stupni jeden pár ozubených kol, planetová převodovka tři páry kol. Planetová převodovka přeneše proto vyšší moment. Vstupní pastorek obou typů převodovek je součástí motoru a zasune se do vstupního otvoru v převodovce při montáži pohonu. Další výhodou planetové převodovky je odstraněné radiální zatížení hřídele motoru, který přenáší pouze mechanický moment.

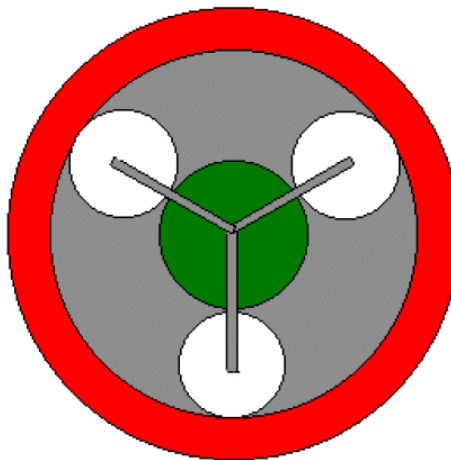
Převodovka tvoří s motorem konstrukční celek s vyváženými rozměry a výkonem. Ke každému motoru je možno vybrat převodovku z typů, které jsou k motoru přiřazeny v katalogu. Obdobná informace o přiřazených motorech je i na listu převodovky.

Obvyklý postup při návrhu pohonu vychází z požadovaného mechanického momentu a rychlosti na výstupu pohonu, které jsou rozhodující pro výběr velikosti převodovky a tím i pro výběr některého z přiřazených motorů podle jejich soupisu v katalogovém listu převodovky. Přitom se respektuje doporučená rychlost na vstupu do převodovky, tj. rychlost motoru a převodový poměr se zvolí tak, aby nebyla překročena. Doladění

požadované rychlosti na výstupu je jednoduché díky snadnému ovládní rychlosti motoru velikostí napájecího napětí. Rychlost motoru přitom může být vyšší nebo nižší než jmenovitá, ale nesmí přesáhnout nejvyšší přípustnou rychlost danou typem motoru, která je pro všechny varianty jmenovitých napětí stejná [9].

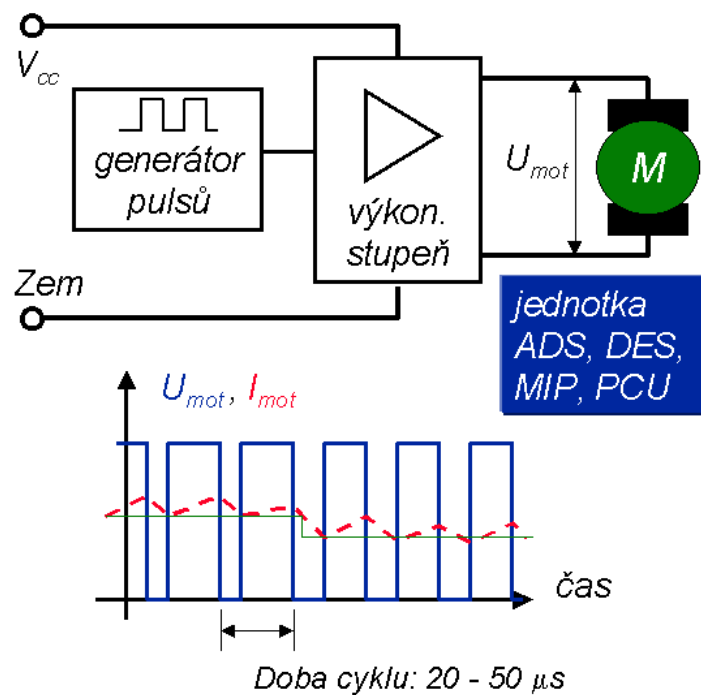


Obr. 8 Převodovka s předlohou [9]



Obr. 9 Schéma jednoho stupně
planetové převodovky [9]

Další možností jak řídit otáčky stejnosměrného motoru je použití PWM modulace. Konstantní napájecí napětí je časově rozděleno do impulsů s konstantní frekvencí a s řízenou šířkou. Protože výkonové tranzistory jsou buď zcela otevřeny nebo zcela zavřeny, nevznikají na nich téměř žádné ztráty. Indukčnost vinutí motoru způsobí, že proud do vinutí nestačí sledovat rychlé změny napětí a průběh proudu je více nebo méně zvlňný [9].



Obr. 10 Pulsní proudový regulátor (PWM) [9]

Mezi výhody PWM regulace otáček stejnosměrného motoru patří zejména nízké ztráty výkonu, vysoká účinnost a možnost použití pro vysoké jmenovité výkony. Mezi hlavní nevýhody můžeme zařadit elektromagnetické rušení na rozhlasových frekvencích, vysoké ztráty se zastaveným motorem a omezení minimální indukčnosti motoru [9].

2.5 Složení a princip modelářského servopohonu

Na obr. 14 je servo Hitec HS-303. Jeho velikost je 40x20x36mm, hmotnost 45g, rychlost pohybu 60 stupňů za 0,16s, krouticí moment 3kg na páce 1cm.

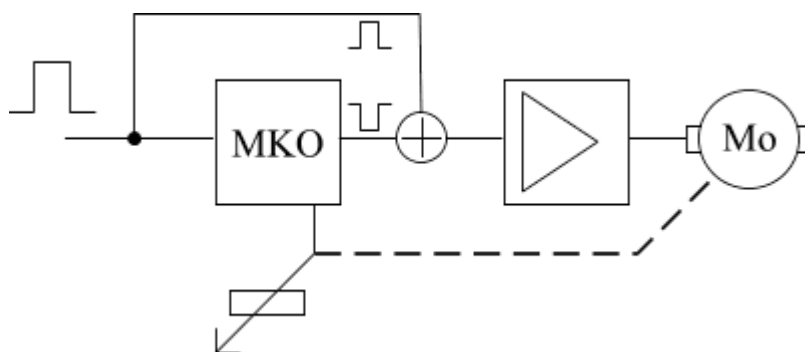


Obr. 11 Servo Hitec HS-303 [10]

Přívodní kabel je dlouhý 25cm, na konci je konektor, který má rozteč dutinek 2,54mm a jejich velikost takovou, že se jako protikus dá použít špiček pro jumpery (v katalogu GM electronic "Konektorové kolíky lámací", typ S1G20, S1G36 nebo S1G40). Konektor na kabelu serva se dá v katalogu GM najít pod označením KONPC-SPK-3. Zapojení konektoru je následující : černý vodič 0V, červený vodič +4,8V, žlutý vodič vstup řídicích impulsů.

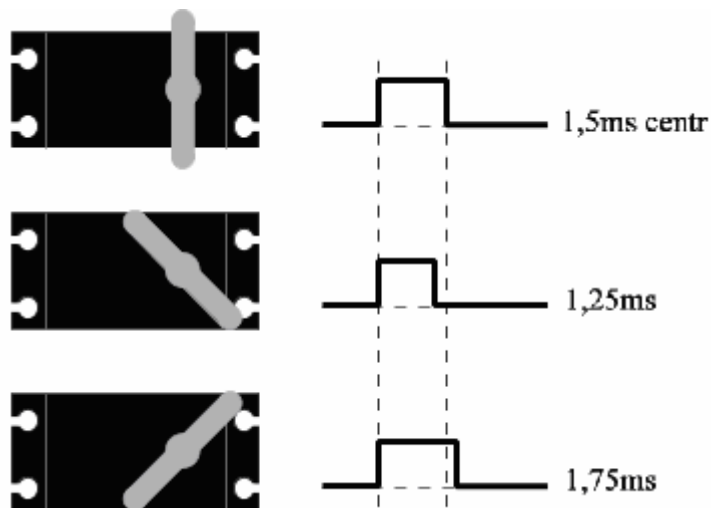
Samozřejmě typů serv je podstatně více, od různých výrobců, různé velikosti a provedení. Servo na obrázku je ukázka takového běžného typu. Existují serva silnější ("jumbo"), taková mají větší rozměry a hmotnost, naopak existují i typy miniaturní ("mikroserva"), ta mají rozměry srovnatelné s kovovou mincí a hmotnost kolem 10g. Všechny takové méně běžné typy jsou ovšem také až několikanásobně dražší ve srovnání s běžným typem serva.

Dnešní modelářská serva obsahují elektromotor s převodovkou a řídicí elektroniku. Zjednodušené zapojení elektroniky je na blokovém schématu [10].



Obr. 12 Blokové schéma elektroniky servomotoru [10]

Do vstupu přichází řídicí impuls, který spustí monostabilní klopný obvod (MKO), ten vygeneruje impuls o délce odpovídající momentální poloze serva a opačné polarity než je vstupní řídicí impuls. Tyto dva impulsy se porovnají a výsledkem je rozdílový impuls, který po zesílení přes můstkový spínač způsobí roztočení elektromotoru jedním nebo druhým směrem. Elektromotor přes převodovku otáčí výstupní hřídeli a současně i potenciometrem, který působí jako zpětná vazba do monostabilního klopného obvodu. Směr otáčení je takový, že impuls generovaný monostabilním klopným obvodem se svojí délkou přibližuje délce vstupního řídicího impulsu a až jsou oba impulsy stejně dlouhé, elektromotor se zastaví. Servo dosáhlo polohu, která odpovídá momentálně přijímanému řídicímu impulsu [10].



Obr. 13 Úhel natočení serva v závislosti na délce řídicího impulsu [10]

Dnes používaná serva pracují s kladnými řídicími impulsy o délce 1-2ms. Délce impulsu 1,5ms odpovídá střední poloha serva, 1ms je jedna a 2ms druhá krajní poloha. Mechanické provedení serva může být takové, že servo je schopno pohybu v rozsahu o něco větším než 180 stupňů, ale není to pravidlem. Většina serv má na koncích rozsahu pohybu mechanické blokování, na což je potřeba dávat pozor, protože při dojetí na doraz se výrazně zvýší proud odebíraný servem a může to skončit tím, že dojde k poškození elektroniky serva.

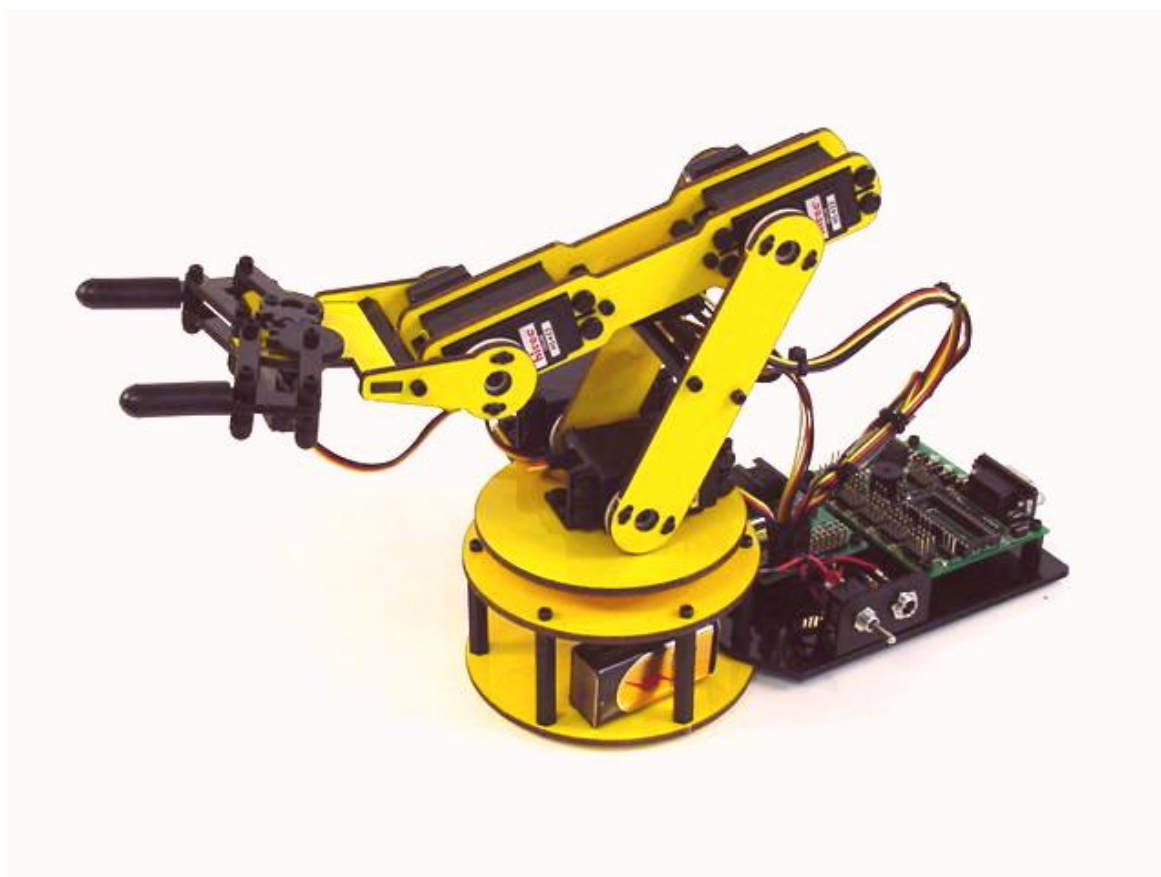
Jak už bylo o něco výše napsáno, dnes používaná serva pracují s kladnými řídicími impulsy o délce 1-2ms. Tyto impulsy se opakují většinou s frekvencí kolem 50Hz. Rychlost opakování není kritická, pokud budou řídicí impulsy přicházet méně často, pouze se zpomalí rychlost pohybu serva. Velikost impulsů je rovná velikosti napájecího napětí serva, což je 4,8-6V (napájení čtyřmi články NiCd $4 \times 1,2V = 4,8V$ nebo suchými články $4 \times 1,5V = 6V$) [10].

3 ROBOT LYNX 5

Robot Lynx 5 je v podstatě zmenšený model průmyslového robota. Robot byl ovládán pomocí modulu SOS-AT přes sériovou linku. Následuje popis robota Lynx 5 a původního ovládacího modulu SOS-AT, který bude nahrazen modulem rozhraní pro vývojový kit M68EVB908GB60.

Mechanické provedení robota

Robotická ruka LYNX 5 se skládá z rotační základny, ramena, loktu, zápěstí a chapadla. Robotická ruka s pěti stupni volnosti umožňuje rychlé a přesné pohyby. LYNX5 je dostupný systém s robustní a pevnou konstrukcí s dlouhou životností.



Obr. 14 Robot Lynx 5 [11]

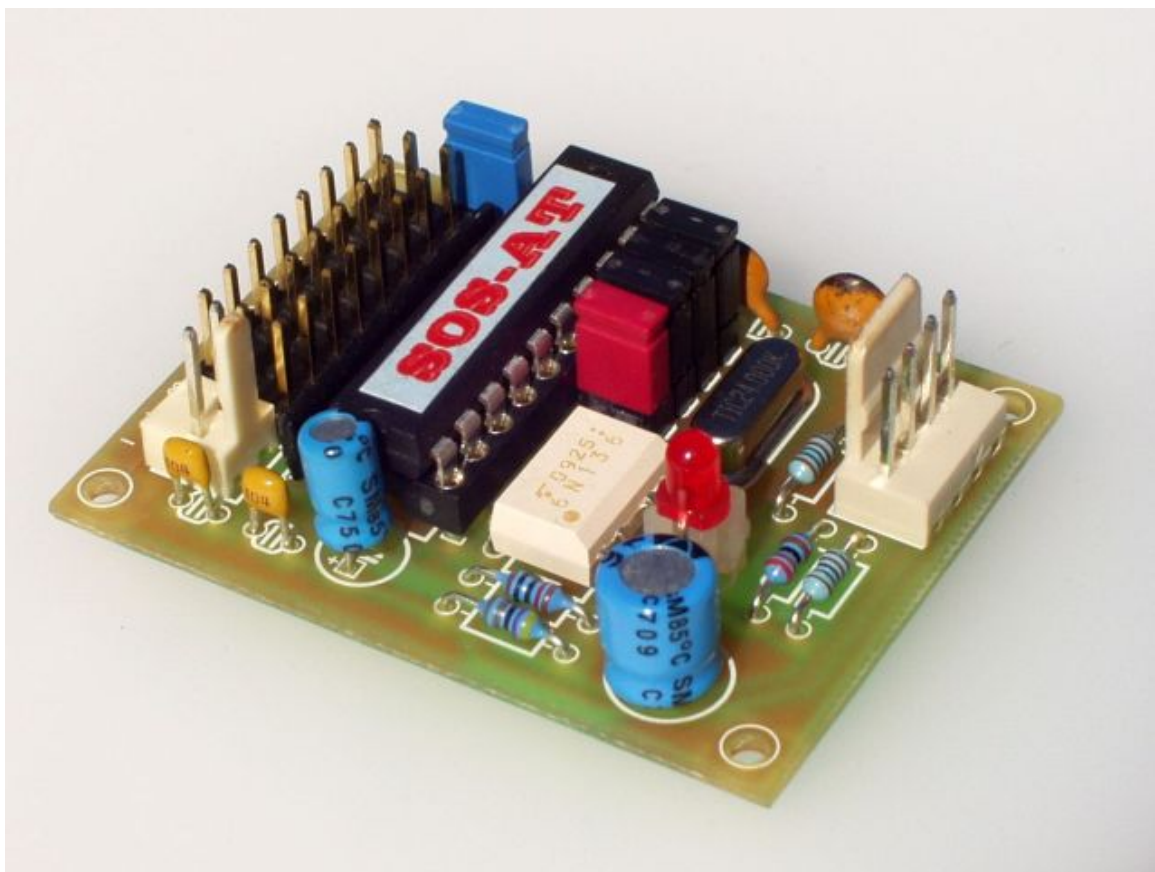
Celá ruka se skládá celkem z šesti modelářských servomotorů, které jsou řízeny PWM signálem (viz předchozí kapitoly).

Pět servomotorů Hitec HS-422 řídí pohyb ruky. Jeden otáčí základnou, dva ramenem, a zbylé dva hýbou loktem a zápěstím. Servomotor HS-81 slouží k řízení chapadla.

Rozsah pohybu pro každou osu je 180°, rozměr otevřeného chapadla je 5 cm. Robotická ruka dokáže manipulovat s předměty do hmotnosti cca 100 g [11].

Modul SOS-AT

Modul SOS-AT je určen pro ovládání polohy výstupního hřídele osmi modelářských servomechanismů podle povelů, předávaných sériovou komunikační linkou dle standardů RS232, RS422 nebo RS485. Přímé ovládání modulu sériovou linkou v úrovních TTL nebo CMOS je také možné. Komunikační linka je na vstupu modulu opticky oddělena od řídicího zařízení. Jeden modul SOS-AT může ovládat 8 servomechanismů, na jednu komunikační linku je možno připojit 8 modulů, naráz lze tedy ovládat pohyb až 64 servomechanismů nezávisle na sobě [12].

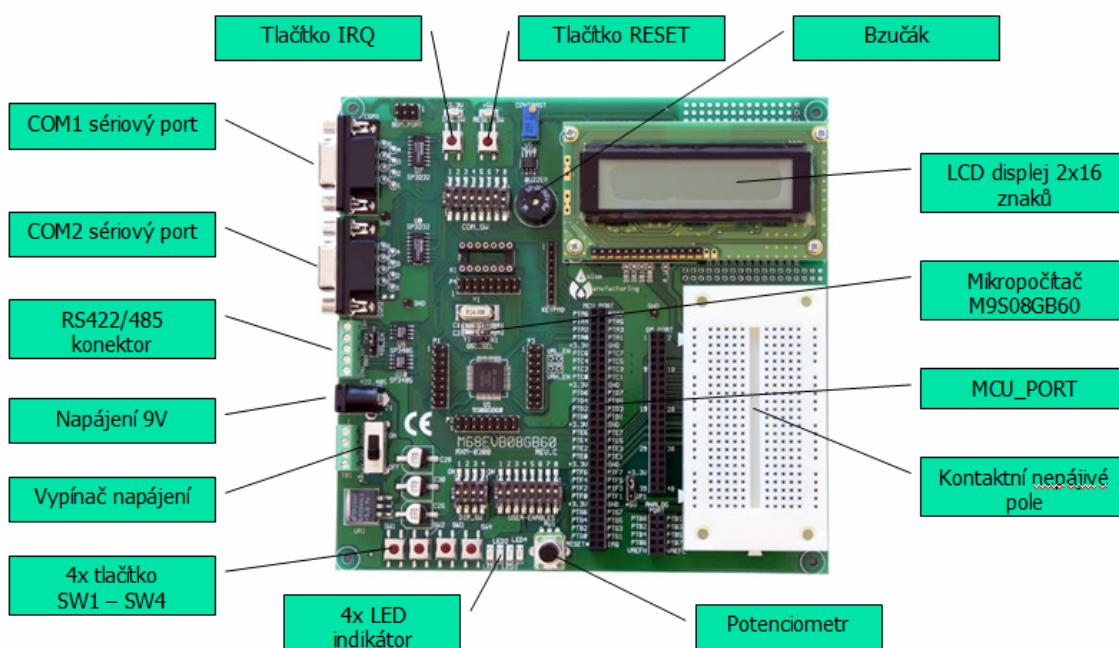


Obr. 15 Modul SOS-AT [13]

4 VÝVOJOVÝ KIT M68EVB908GB60

Výukový kit M68EVB908GB60 od firmy Freescale popsany v této kapitole je používán pro výuku předmětu Programování mikropočítačů. Jsou zde uvedeny základní hardwarové vlastnosti vývojového kitu a popis MCU portu, pomocí kterého bude modul rozhraní přijímat řídicí signály.

Periferie vývojového kitu



Obr. 16 Periferie vývojového kitu [14]

Napojení periférií na porty:

- LED indikátory – port F
 - ✓ LED1 – PTF0, LED2 – PTF1
 - ✓ LED3 – PTF2, LED4 – PTF3
- Tlačítka – port A
 - ✓ SW1 – PTA4, SW2 – PTA5
 - ✓ SW3 – PTA6, SW4 – PTA7
- Bzučák – port D
 - ✓ BUZZER – PTD0

- LCD modul – port G a port E
 - ✓ PTG3-7 a PTE6-7
- pólový DIP spínač – port B
 - ✓ PTB4-7
- Potenciometr – port B
 - ✓ RV1 – PTB0/AD0 [14]

Základní vlastnosti vývojového kitu

- 8 bitová centrální procesní jednotka HCS08, taktovací frekvence max. 40MHz
- 60KB FLASH paměti
- 4KB RAM paměti
- 56 vstupně/výstupních linek na 7 portech (porty A-G)
- 5 kanálový TPM2 časovač
- 3 kanálový TPM1 časovač
- 1x Synchronní sériové periferní rozhraní (SPI)
- 1x I2C rozhraní
- 2x Asynchronní sériové komunikační rozhraní (SCI)
- Interní generátor hodinového kmitočtu s FLL obvodem (32KHz – 20MHz frekvence sběrnice)
- 8 kanálový, 10 bitový A/D převodník
- COP watchdog systém s nastavitelnou časovou prodlevou na 218 nebo 213 cyklů sběrnice
- Systém kontroly napájecího napětí umožňující detekovat pokles napětí pod stanovenou mez
- Podpora režimů se sníženou spotřebou umožňující nasazení v bateriově napájených aplikacích
- BDM rozhraní pro pokročilé ladění a programování aplikací přímo v aplikaci [14]

MCU port

MCU PORT poskytuje přístup k vstupům a výstupům mikropočítače 9S08GB60 umístěném na vývojovém kitu [15].

Na obr. 20 jsou zvýrazněny piny, které budou použity pro připojení modulu rozhraní.

PTA6/KBD6	1	2	PTA7/KBD7
PTA4/KBD4	3	4	PTA5/KBD5
PTA2/KBD2	5	6	PTA3/KBD3
PTA0/KBD0	7	8	PTA1/KBD1
3.3V	9	10	GND
PTC6	11	12	PTC7
PTC4/CLKOUT	13	14	PTC5
PTC2/SDA	15	16	PTC3/SCL
PTC0/TXD2	17	18	PTC1/RXD2
3.3V	19	20	GND
PTD6/TPM2CH3	21	22	PTD7/TPM2CH4
PTD4/TPM2CH1	23	24	PTD5/TPM2CH2
PTD2/TPM1CH2	25	26	PTD3/TPM2CH0
PTD0/TPM1CH0	27	28	PTD1/TPM1CH1
3.3V	29	30	GND
PTE6	31	32	PTE7
PTE4/MOSI	33	34	PTE5/SPSCK
PTE2/SS*	35	36	PTE3/MISO
PTE0/TXD1	37	38	PTE1/RXD1
3.3V	39	40	GND
PTF6	41	42	PTF7
PTF4	43	44	PTF5
PTF2	45	46	PTF3
PTF0	47	48	PTF1
3.3V	49	50	GND
PTG6	51	52	PTG7
PTG4	53	54	PTG5
PTG2/EXTAL	55	56	PTG3
PTG0/BGND/MS	57	58	PTG1/XTAL
RESET*	59	60	IRQ

Obr. 17 Piny MCU portu [15]

II. PRAKTICKÁ ČÁST

5 POUŽITÝ MIKROPOČÍTAČ

Základní požadavek na modul rozhraní je kontrolovat řídicí PWM signály generované vývojovým kitem a v případě, že je signál v pořádku jej poslat k robotu. V případě chybného signálu na vstupu by bylo vhodné, aby tento chybový stav byl nějak signalizován. Robot se řídí pomocí pěti PWM signálů. Bude tedy potřeba pět kanálů časovače pro příjem signálů generovaných vývojovým kitem a pět kanálů časovače pro generování výstupních signálů pro robota. Časovač, který bude zpracovávat vstupní signály by měl mít funkci Input capture, která slouží k zachycení události na vstupu a časovač, pomocí kterého budeme generovat výstupní signál by měl podporovat funkci PWM.

Bude tedy zvolen mikropočítač, který je vybaven celkem deseti kanály od časovačů s funkcemi Input capture a generováním PWM a ještě několika výstupními piny pro signalizaci chyby vstupního signálu.

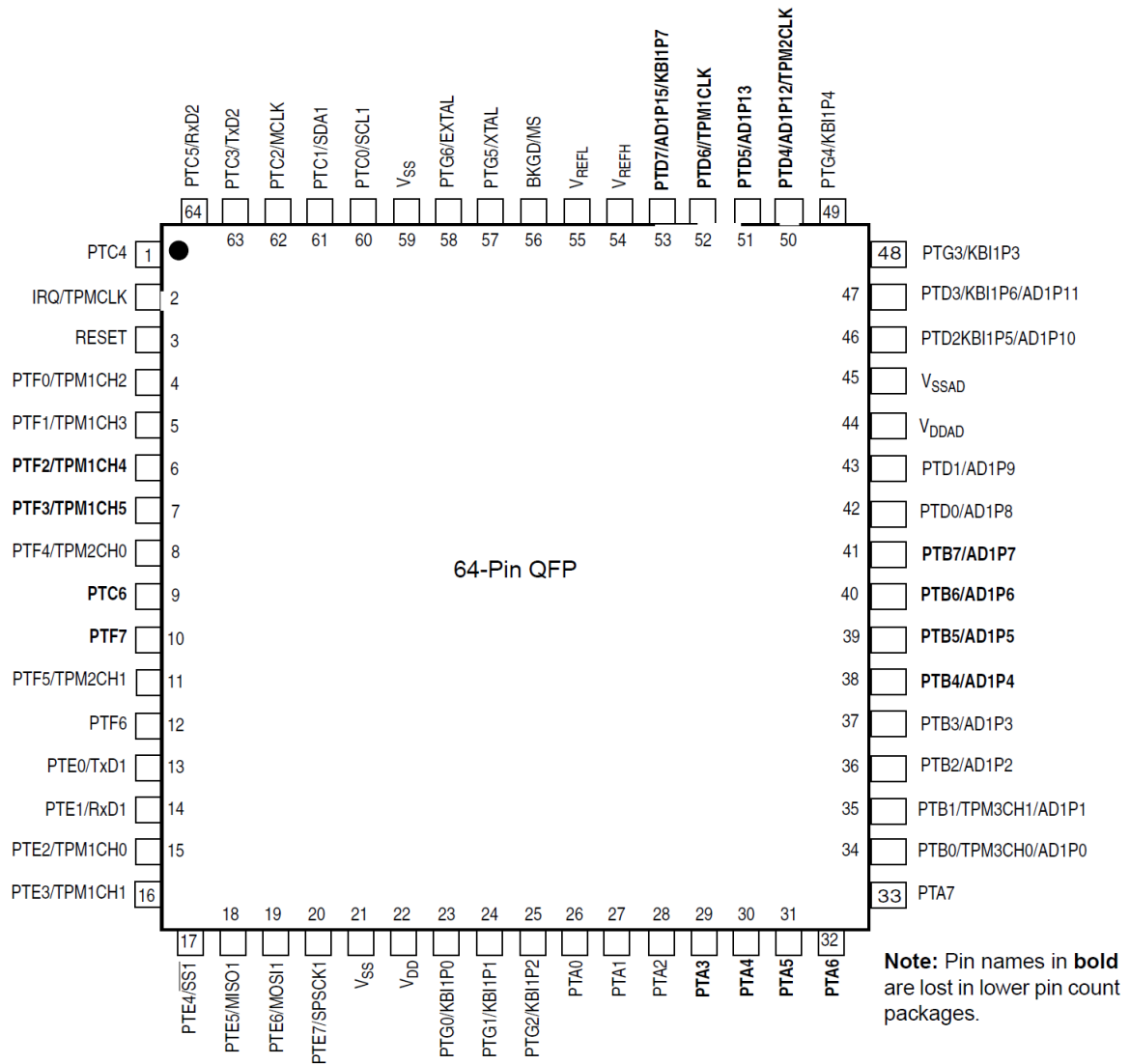
Po důkladném uvážení všech funkcí, které by měl modul rozhraní pro připojení výukového robota k vývojovému kitu splňovat, padlo rozhodnutí použít mikropočítač MC9S08AC128 od firmy Freescale v pouzdru 64-QFP.

Základní vlastnosti mikropočítače MC9S08AC128

MC9S08AC128 je levný, vysoce výkonný 8-bitový mikropočítač z rodiny HCS08. V MC9S08AC128 je použito vylepšené jádro HCS08 [16].

Základní vlastnosti použitého mikropočítače:

- Napájecí napětí 2,7 – 5,5 V
- Frekvence CPU až 40 MHz
- Flash 128 kB
- RAM 8 kB
- Modul přerušování KBI připojený na 8pinů
- 3 x časovač s podporou PWM (celkem 10 kanálů)
- A/D převodník s rozlišením 10 bitů (16 kanálů)
- Komunikace: 2x SCI, SPI, I2C, BDM [16]



Obr. 18 Mikropočítač MC9S08AC128 v pouzdru 64-QFP [16]

Programátor

Při programování mikropočítače byl použit programátor od firmy P&E Microcomputer systems.

P&E USB BDM Multilink je ladicí rozhraní, které umožňuje na PC přístup do režimu Background Debug (BDM) na mikropočítačích Freescale HCS08, HC(S)12(X), RS08 a ColdFire V1. Umožňuje spojení mezi USB portem na počítači se systémem Windows a standardním 6 pinovým BDM konektorem.

Pomocí USB BDM Multilink, může uživatel využít režimu ladění na pozadí a zastavit normální běh procesoru a pomocí PC běh procesoru řídit. Uživatel pak může přímo ovládat

vykonávání instrukcí, čtení a zápis do registrů a paměti, ladění kódu na procesoru a programovat interní nebo externí zařízení s flash pamětí. [17].



Obr. 19 Programátor USB BDM Multilink [17]

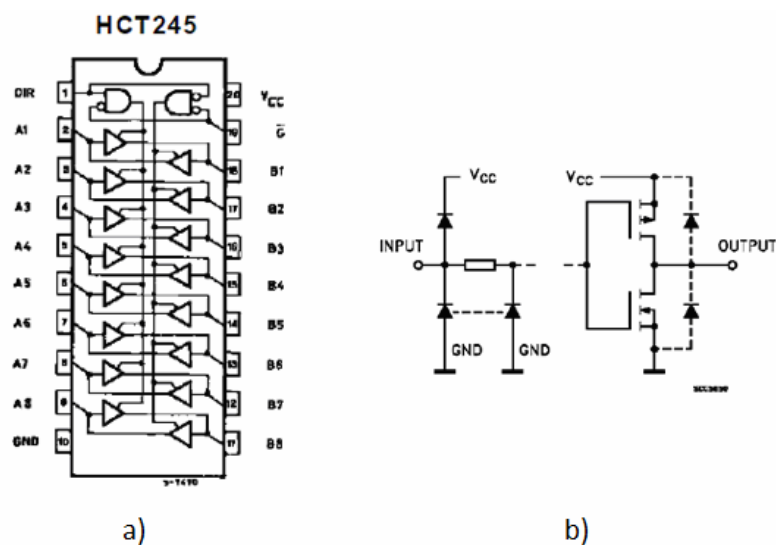
6 BUDIČ SBĚRNICE 74HCT245N

Protože vývojový kit generuje PWM signály o napěťové hodnotě 3,3V a vybraný MCU a serva pracují se signály s napětím 5 V, použil jsem jako převodník napěťových úrovní IO (integrováný obvod) 74HCT245N. Tento IO je logický, osmibitový budič sběrnice, který může fungovat jako vysílač i přijímač.

Základní vlastnosti IO 74HCT245N

- Napájecí napětí: 4,5V - 5,5 V
- Pracovní teplota: -40 to +85 °C
- Vstupní napětí pro logickou úroveň jedna: 2,0V – napájecí napětí.
- Vstupní napětí pro logickou úroveň nula: 0 – 0,8V
- Výstupní napětí pro log. 1 minimálně 4,13V při napájecím napětí 4,5V a odebíraném proudu -6mA a minimálně 4,4V při napájecím napětí 4,5V a odebíraném proudu -20μA.
- Výstupní napětí pro log. 0 maximálně 0,33V při napájecím napětí 4,5V a odebíraném proudu 6mA a maximálně 0,1V při napájecím napětí 4,5V a odebíraném proudu 20μA [18].

Zapojení a význam pinů:



Obr. 20 a) zapojení pinů, pohled shora b) náhradní schéma vstupu a výstupu [18]

Tab. 1 Význam a zapojení pinů HCT245 [18]

Číslo	Označení	Význam a funkce
1	<i>DIR</i>	Řízení směru
2,3,4,5,6,7,8,9	<i>A1 – A8</i>	Datové vstupy/výstupy
18,17,16,15,14,13,12,11	<i>B1 – B8</i>	Datové vstupy/výstupy
19	\overline{G}	Zapnutý výstup (aktivní při nízké logické úrovni)
10	<i>GND</i>	Zem (0V)
20	V_{cc}	Kladné napájecí napětí

Tab. 2 Pravdivostní tabulka IO HCT 245 [18]

Vstupy		Funkce		Výstup
\overline{G}	<i>DIR</i>	Sběrnice A	Sběrnice B	-
L	L	výstup	vstup	A=B
L	H	vstup	výstup	B=A
H	X	Z	Z	Z

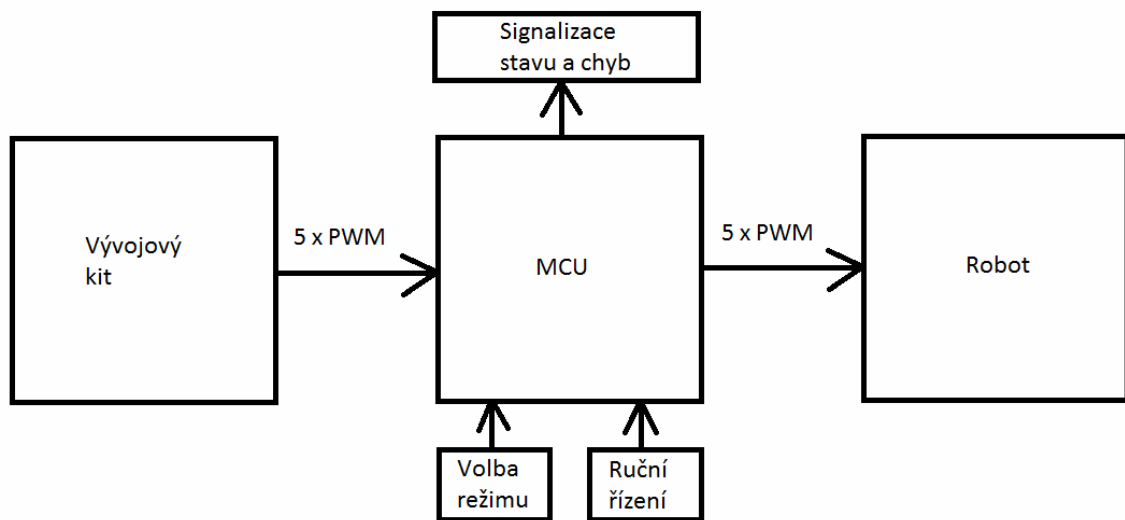
Význam písmen v pravdivostní tabulce: L – logická nula, H – logická jednička, X – libovolná logická úroveň, Z – stav vysoké impedance [18]

7 NÁVRH A VÝROBA HARDWARE

Po nastudování dokumentace k vybranému mikroprocesoru bylo zjištěno, že tento mikropočítač nabízí kromě deseti kanálů od třech časovačů ještě 16 kanálů od AD převodníku (2 z nich jsou sdílené s kanály od časovače TPM3), modul přerušení KBI připojený na 8 pinů a spoustu dalších vstupně výstupních pinů, které by mohly rozšířit modul rozhraní o další funkci.

7.1 Blokové schéma

Po různých úvahách bylo nakresleno následující blokové schéma:



Obr. 21 Blokové schéma modulu rozhraní

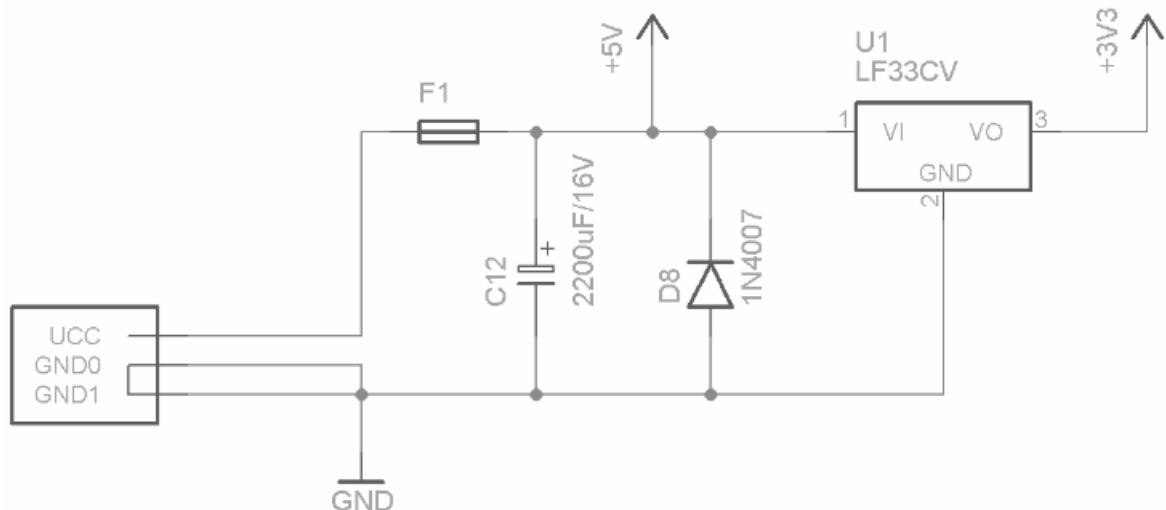
Mikropočítač bude kontrolovat pět vstupních PWM signálů a po úspěšné kontrole vygeneruje PWM signál na výstupu. V případě chybného vstupního signálu bude chyba signalizována pomocí pěti červených LED diod (pro každý signál jedna). K dispozici bude možnost ověřit správnou funkci robota bez vývojového kitu. K mikropočítači bude připojeno tlačítko na modul přerušení KBI. Tímto tlačítkem bude možné kdykoli přepnout mezi režimem řízení pomocí vývojového kitu, nebo potenciometrů připojených ke kanálům AD převodníku. Režim řízení robota bude signalizován žlutou LED diodou. Zelená LED dioda bude signalizovat připravenost modulu rozhraní.

7.2 Popis jednotlivých částí

Podle dokumentace k mikropočítači bylo z větší části navrženo schéma zapojení modulu. Následuje popis jednotlivých částí modulu, celkové schéma je v příloze P I.

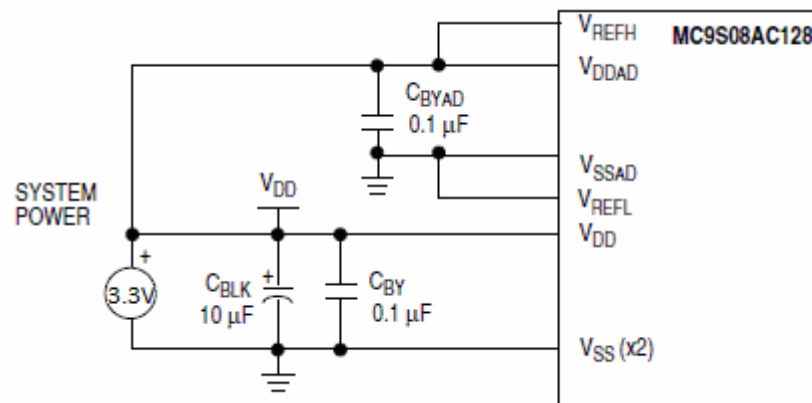
7.2.1 Napájení

Napájení modulu rozhraní je zajištěno externím napájecím síťovým adaptérem s výstupním napětím 5V, který je schopen dodat proud až 4A. Usměrnovací dioda D8 a rychlá pojistka F1 se jmenovitým proudem 3,15A slouží jako ochrana proti přepólování napájecího napětí a zkratu. Napětí 5V je použito k napájení budiče sběrnice a servomotorů robota. Stabilizované napětí 3,3V slouží k napájení mikropočítače a jako referenční napětí.



Obr. 22 Schéma zapojení napájení a pojistky proti zkratu a přepólování

V_{DD} a V_{SS} jsou napájecí piny mikropočítače. V_{DDAD} a V_{SSAD} jsou napájecí piny AD převodníku. Elektrolytický kondenzátor slouží ke stabilizaci napájení při vyšším odběru proudu. Keramické kondenzátory slouží k potlačení vysokofrekvenčního šumu. V_{REFL} a V_{REFH} jsou piny, na které jsou určeny k připojení referenčního napětí AD převodníku [16].

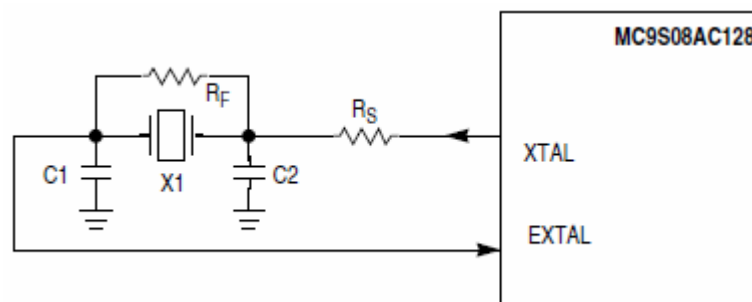


Obr. 23 Schéma napájení mikropočítače [16]

7.2.2 Zdroj hodinového signálu

Mikropočítač obsahuje dva interní zdroje hodinového signálu o frekvenci 8 MHz a 243 kHz, nebo je možné použít externí zdroj. Piny pro připojení krystalu jsou shodné s piny RTG5 a PTG6, takže pokud využijeme interní zdroj hodinového signálu, můžeme tyto piny využít pro jinou funkci. Na piny XTAL a EXTAL lze připojit krystal nebo rezonátor v rozsahu 32 – 100 kHz, nebo 1 – 16 MHz[16].

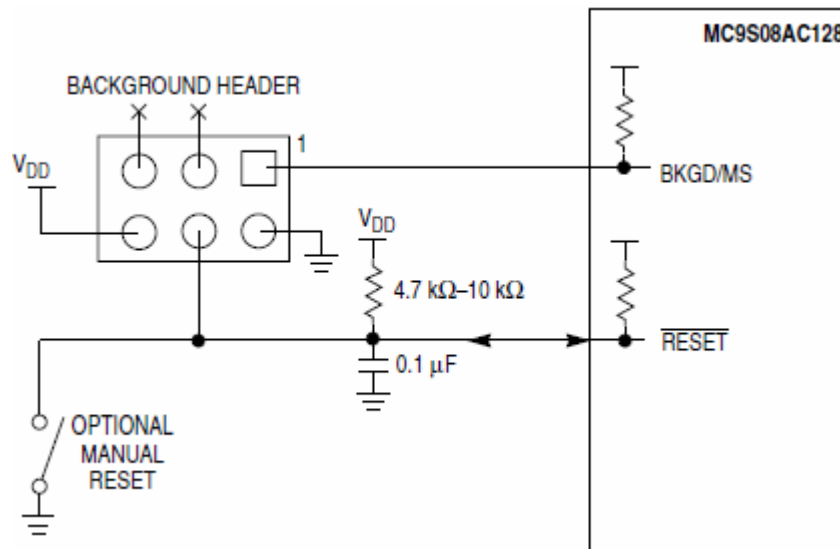
V zapojení je využito možnosti připojení externího zdroje hodinového signálu. Zapojení krystalu je realizováno podle schématu uvedeného v datasheetu. Krystal jsem zvolil o frekvenci 16 MHz. Hodnoty rezistorů $R_S = 0 \Omega$ a $R_F = 1 \text{ M}\Omega$ byly zjištěny z datasheetu k MCU [16] a hodnoty $C_1 = C_2 = 8,2 \text{ pF}$ byly uvedeny v datasheetu ke krystalu [19].



Obr. 24 Schéma připojení krystalu k mikropočítači [16]

7.2.3 BDM a tlačítko Reset

BDM konektor slouží pro programování, ladění programu a komunikaci na pozadí s mikropočítačem umístěným přímo v aplikaci. Jedná se o šestipinový konektor, na které je připojeno napájecí napětí, zem a piny RESET a BKGD/MS. Dva piny jsou nezapojeny. Pokud není na pin BKGD/MS nic připojeno, spustí se po restartu mikropočítač do normálního provozního režimu [16].

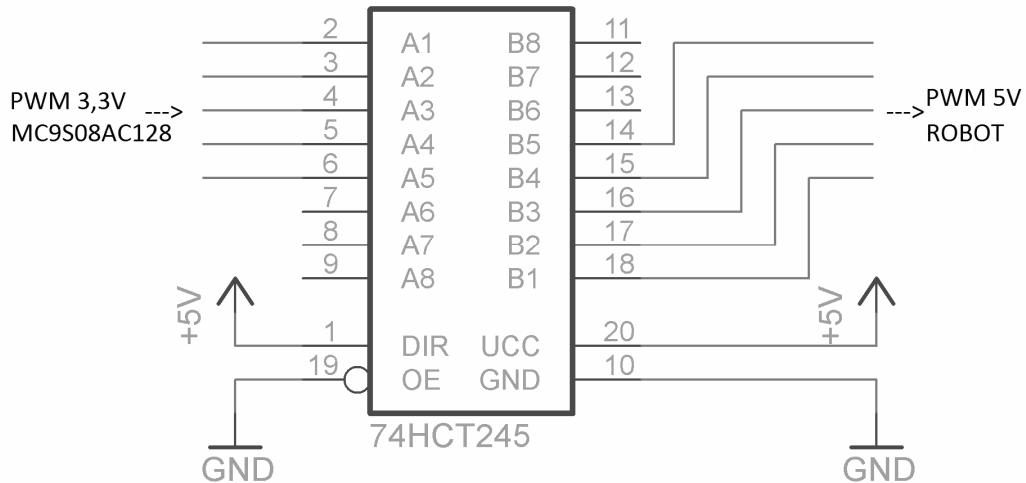


Obr. 25 Schéma zapojení BDM konektoru a tlačítka RESET k mikropočítači [16]

Pin RESET je připojen na konektor BDM, takže se může restartovat MCU přímo ve vývojovém prostředí. Restart mikropočítače může být také proveden manuálně připojením pinu RESET na zem pomocí připojeného spínače. K pinu je připojen RC filtr, který slouží k omezení rušivých signálů [16].

7.2.4 Zapojení budiče sběrnice

Vývojový kit generuje PWM signál s amplitudou 3,3 V, servopohony pracují s řídicím PWM signálem s amplitudou 5V. Vybraný mikropočítač pracuje s napájecím napětím 2,7 – 5,5V. Aby bylo možné řídit výstupním signálem z mikropočítače servomotory robota, musí se upravit amplituda PWM signálu z 3,3 na 5 V. Pro tuto funkci byl zvolen integrovaný obvod 74HCT245N. Zapojení je navrženo na základě informací zjištěných v datasheetu.

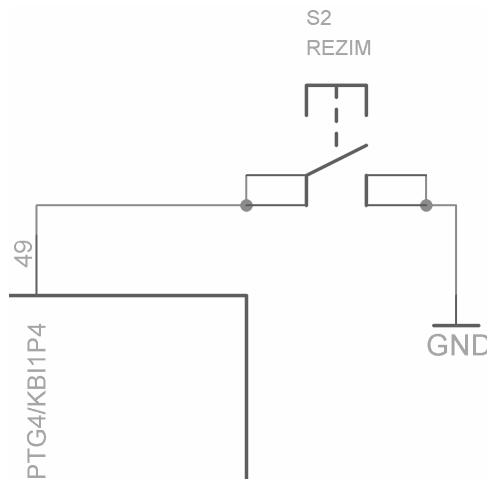


Obr. 26 Schéma zapojení budiče sběrnice

7.2.5 Připojení tlačítka pro volbu režimu ovládání

MC9S08AC128 má jeden KBI modul s několika vstupy pro přerušování od klávesnice. Počet vyvedených pinů závisí na pouzdru, ve kterém je mikroprocesor umístěn. V našem případě je z pouzdra 64-QFP vyvedeno celkem 8 pinů připojených na KBI modul [16].

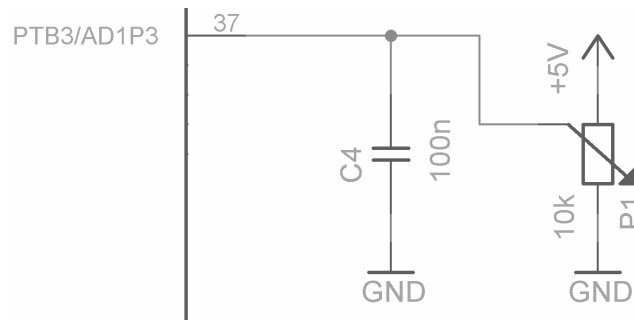
Jedním z pinů, na který je připojen modul přerušování KBI je pin č.49. Na tento pin bude připojeno tlačítko, které bude sloužit ke změně režimu řízení robota.



Obr. 27 Schéma připojení tlačítka k mikroprocesoru

7.2.6 Připojení potenciometrů k AD převodníkům

Jako zdroj referenčního napětí AD převodníku je zvoleno napájecí napětí mikropočítače 3,3V. Na napájecí napětí je též připojeno pět potenciometrů s odrušovacím kondenzátorem, jejichž výstupy jsou připojeny k jednotlivým kanálům AD převodníku. Takže na AD převodníku může být napětí v rozsahu od 0 V až do referenčního napětí 3,3V.



Obr. 28 Schéma připojení potenciometru ke kanálu AD převodníku mikropočítače

7.2.7 Připojení LED diod

LED diody signalizující stav rozhraní a správnost PWM řídicího signálu budou připojeny k mikropočítači na piny, které budou nastavené jako výstupní. Na těchto pinech bude napětí 0, nebo 5V. K LED diodě se připojí předřadný rezistor, aby nedošlo k jejímu zničení. Hodnota odporu předřadného rezistoru je vypočítána ze vztahu:

$$R_p = \frac{U_{zdroje} - U_f}{I_f}, \quad (1)$$

kde R_p je hodnota předřadného odporu v Ohmech, U_{zdroje} je napětí na pinu mikropočítače 5 V, U_f a I_f jsou katalogové hodnoty napětí a proudu LED diody. V tabulce níže jsou spočítané hodnoty odporů předřadných rezistorů pro použité LED diody:

Tab. 3 Katalogové hodnoty LED diod a hodnoty předřadných odporů

LED dioda	U_f [V]	I_f [mA]	R_p [Ω]
červená	2,3	2	500
žlutá	1,9	2	700
zelená	1,8	2	750

Aby bylo zaručeno, že nedojde ke zničení LED diod, jsou v zapojení použity rezistory s odporem 820Ω . Větší hodnota odporu předřadného rezistoru u LED diody není na závadu, způsobí pouze to, že dioda bude svítit o něco méně.

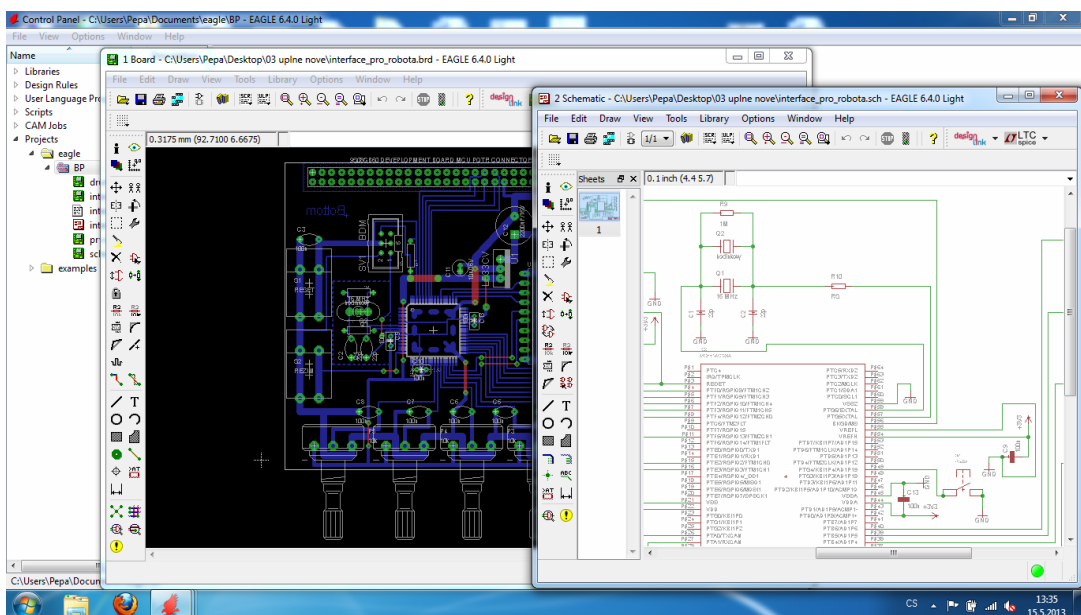
7.3 Návrh desky plošného spoje v programu Eagle

Pro návrh desky modulu rozhraní byl použit software Eagle 6.4.0 od firmy Cadsoft. Program je možné používat jako freeware ve verzi Light, která je volně ke stažení na stránkách společnosti.

Eagle Light je omezen následujícími vlastnostmi:

- Použitelná plocha desky je omezena na 100 x 80 mm (4 x 3.2 palce).
- Mohou být použity jen dvě signálové vrstvy spojů (vrchní a spodní strana).
- Editor schématu může vytvořit schéma pouze na jednom listu.
- Podpora je dostupná pouze pomocí e-mailu nebo prostřednictvím fóra uživatelů (ne faxem nebo telefonem).
- Použití je omezeno na nevýdělečné aplikace nebo studijní účely [20].

S ohledem na počet použitých součástek a jejich rozměry je omezená velikost desky 10 x 8 cm zcela dostačující. Pro propojení součástek na desce budou stačit dvě signální vrstvy. Použití programu Eagle ve verzi Light tedy nebude ničím omezovat návrh desky.



Obr. 29 Návrh DPS v programu Eagle

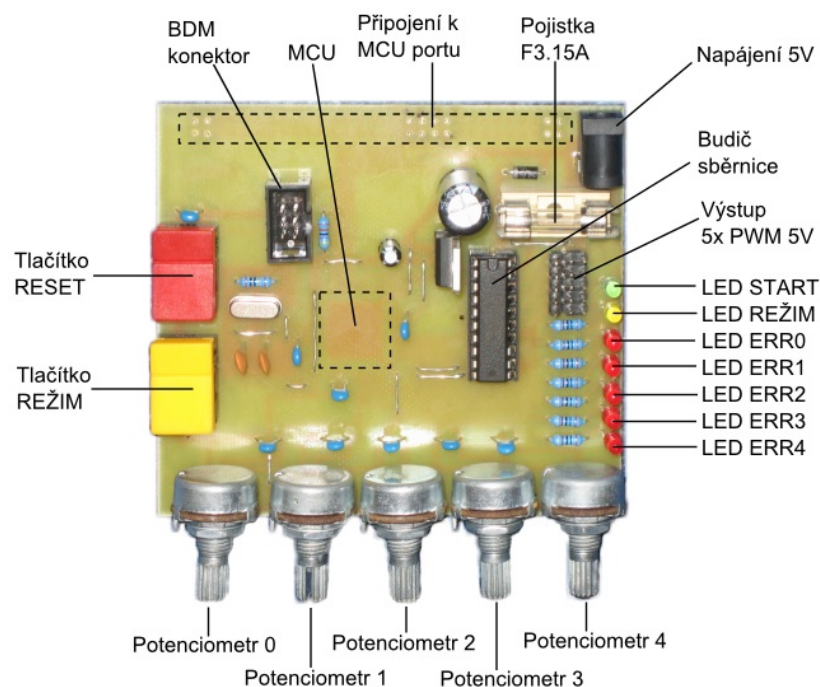
Při návrhu desky plošného spoje byla má představa o umístění součástek na modulu rozhraní následující: MCU konektor je umístěn na desce tak, aby deska překryla kontaktní nepájivé pole na vývojovém kitu. Tlačítka a potenciometry jsou umístěny u okraje desky tak, aby se modul pohodlně ovládal. Signalizační LED diody jsou umístěny tak, aby byly hned pod displejem, aby případný výpis z vývojového kitu a signalizace stavu rozhraní byly pěkně u sebe. Napájecí konektor je samozřejmě umístěn, tak, aby kabel od napájecího adaptéru nikde nepřekážel.

7.4 Výroba desky plošného spoje

Po návrhu desky plošného spoje byl obrazec vytištěn na průhlednou fólii. Tato fólie byla přiložena na cuprexitovou desku s fotocitlivou vrstvou a následně osvětlena UV lampou. Poté byla deska vyvolána v roztoku hydroxidu sodného (NaOH) a vyleptána v roztoku chloridu železitého (FeCl₂). Po vyleptání byla deska opět osvětlena UV lampou, vyvolána a očištěna od zbytku fotocitlivé vrstvy. Posledním krokem bylo vyvrtání děr pro součástky a osazení desky.

7.5 Popis osazené desky modulu rozhraní

Níže jsou popsány možnosti ovládání, signalizace stavu a chyb modulu rozhraní:



Obr. 30 Osazená deska modulu rozhraní

Tlačítko RESET – slouží k restartu mikropočítače modulu rozhraní

Tlačítko REŽIM – slouží pro výběr ovládání servopohonů pomocí signálů z vývojového kitu, nebo manuálního ovládání pomocí pěti potenciometrů.

Potenciometr 0 – v manuálním režimu slouží pro ovládání pohonu základny robota.

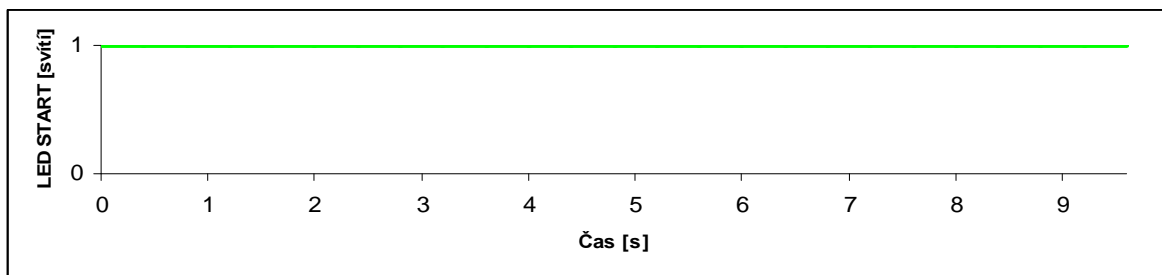
Potenciometr 1 – v manuálním režimu slouží pro ovládání pohonu ramena robota.

Potenciometr 2 – v manuálním režimu slouží pro ovládání pohonu loktu robota.

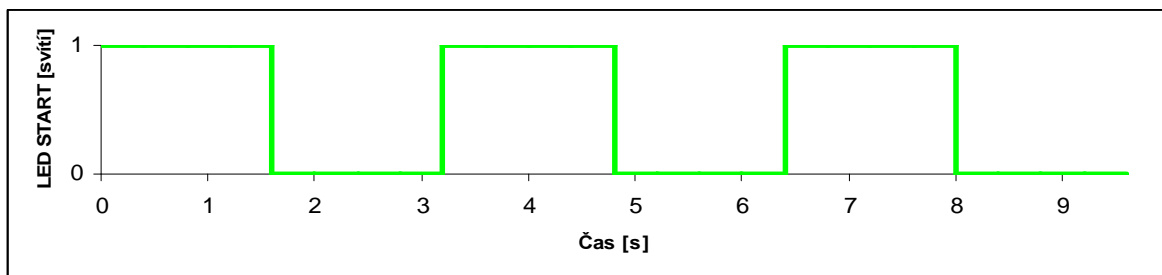
Potenciometr 3 – v manuálním režimu slouží pro ovládání pohonu zápěstí robota.

Potenciometr 4 – v manuálním režimu slouží pro ovládání pohonu chapadla robota.

LED START – signalizuje správnou činnost modulu a polohu servopohonů robota. Po inicializaci modulu LED svítí.

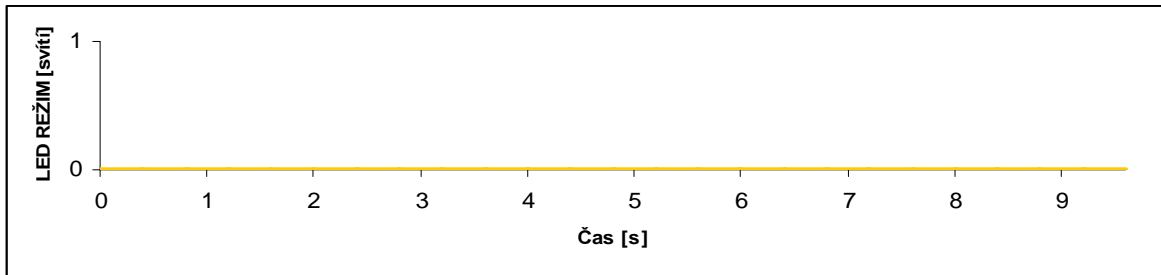


Obr. 31 Časový průběh signalizace LED START – robot je v povolené poloze

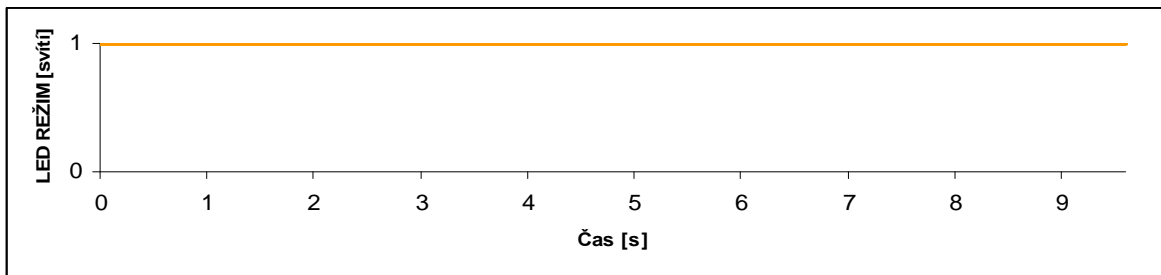


Obr. 32 Časový průběh signalizace LED START – robot není v povolené poloze

LED REŽIM – signalizuje režim ovládání robota.

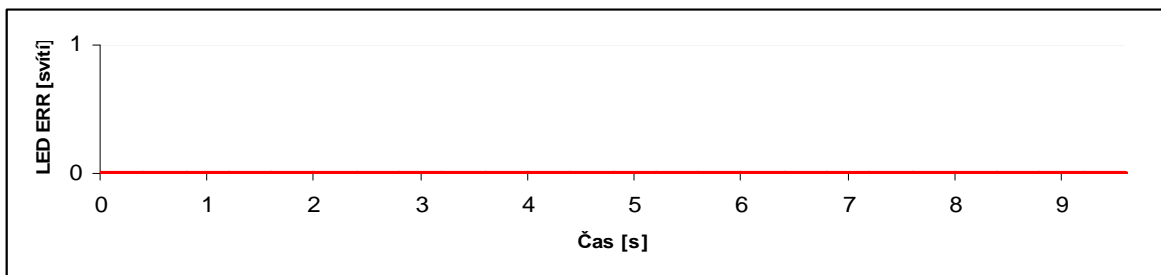


Obr. 33 Časový průběh signalizace LED REŽIM – robot je řízen z vývojového kitu

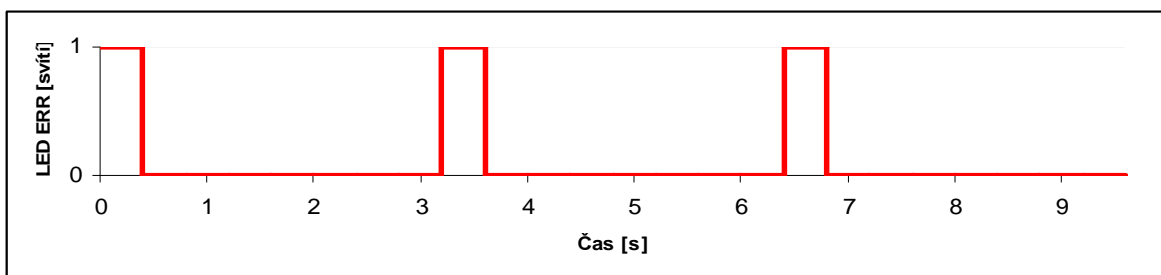


Obr. 34 Časový průběh signalizace LED REŽIM – robot je řízen potenciometry

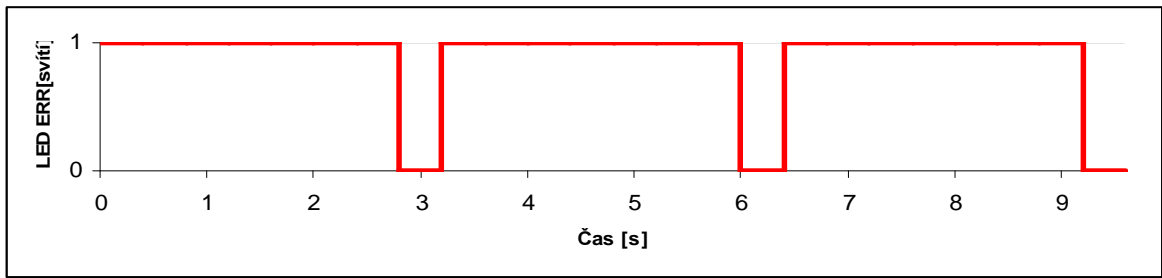
LED ERR0-4 – v automatickém režimu slouží pro signalizaci chyby PWM signálu generovaného vývojovým kitem. Číslování LED odpovídá pořadí kanálů TPM2 na vývojovém kitem (ERR0 signalizuje chybu TPM2CH0, ERR1 signalizuje chybu TPM2CH1, atd.).



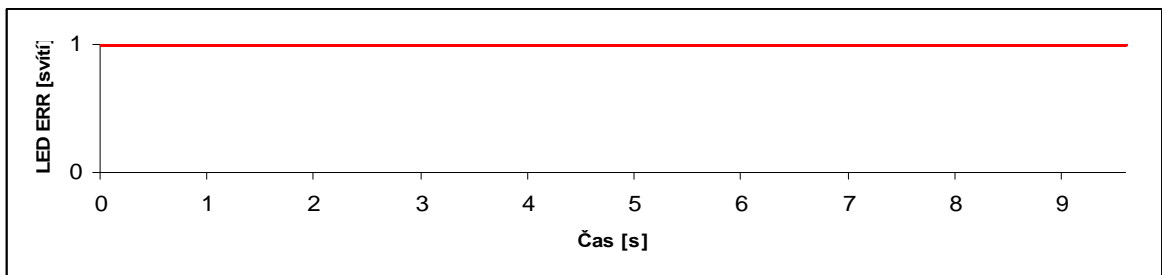
Obr. 35 Časový průběh signalizace LED ERR – signál je v pořádku



Obr. 36 Časový průběh signalizace LED ERR – signál má špatnou délku pulsu



Obr. 37 Časový průběh signalizace LED ERR – signál má špatnou délku periody



Obr. 38 Časový průběh signalizace LED ERR – signál s chybnou délkou periody i pulsu

Pro správnou funkci rozhraní je potřeba servopohony robota správně připojit. Pokud by byly servopohony připojeny jinak, než je znázorněno na obrázku níže, bude sice fungovat kontrola jednotlivých PWM signálů, ale nebude fungovat kontrola povolených poloh celého robota.



Obr. 39 Připojení servopohonů k modulu rozhraní

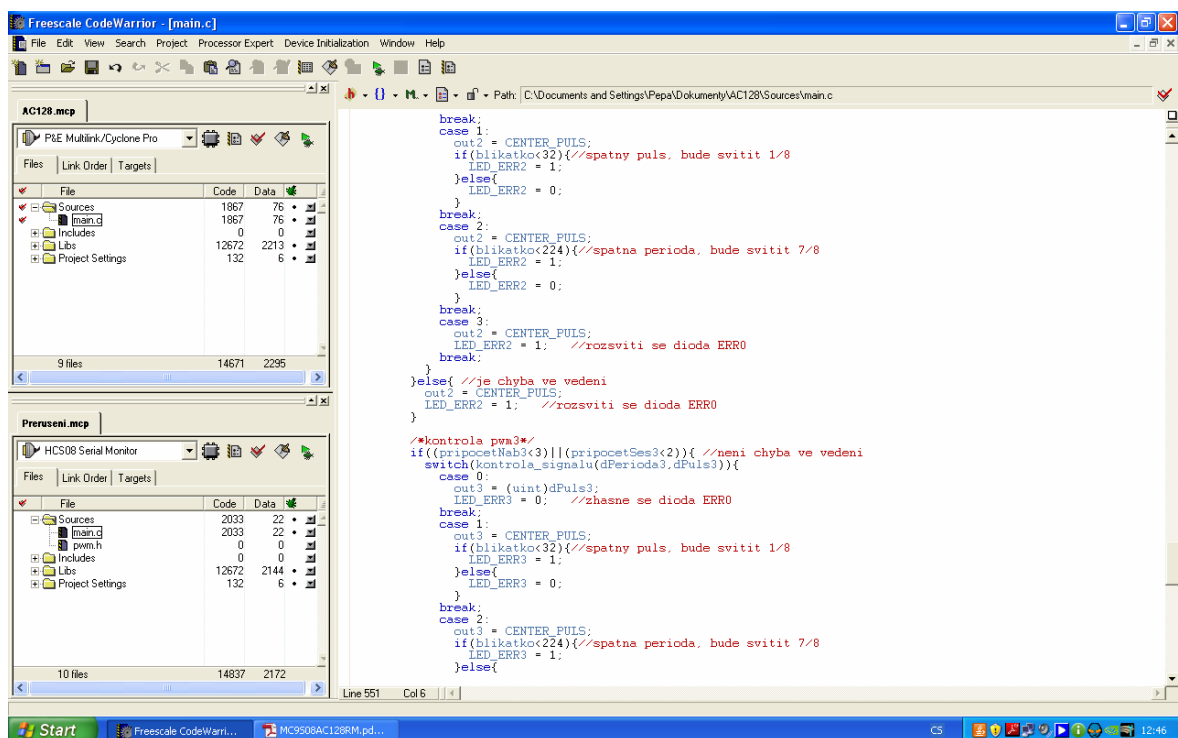
8 PROGRAMOVÉ VYBAVENÍ A OVĚŘENÍ FUNKCE

V této kapitole bude popsán způsob a postup tvorby firmware pro modul rozhraní a knihovny podprogramů pro vývojový kit, která umožní snadnou programovou obsluhu robota. Správná funkce bude ověřena na ukázkovém příkladu na konci kapitoly.

8.1 Vývojové prostředí

Při vytváření programového vybavení pro modul rozhraní a vývojový kit bylo použito vývojové prostředí CodeWarrior 6.3 SE.

Freescale CodeWarrior 6.3 pro mikro počítače je jediná, integrovaná sada nástrojů navržena tak, aby umožnila rychlý vývoj aplikací na mikro počítačích řady RS08, S08 a ColdFire V1 od firmy Freescale. Ať už navrhujete 8-bitovou aplikaci na nízké úrovni (např. detektor kouře) nebo 32-bitové aplikaci na vysoké úrovni (např. požární poplachovou ústřednu), CodeWarrior poskytuje optimalizované nástroje plně použitelné pro mikro počítače Freescale [21].



Obr. 40 Vývojové prostředí CodeWarrior 6.3 SE

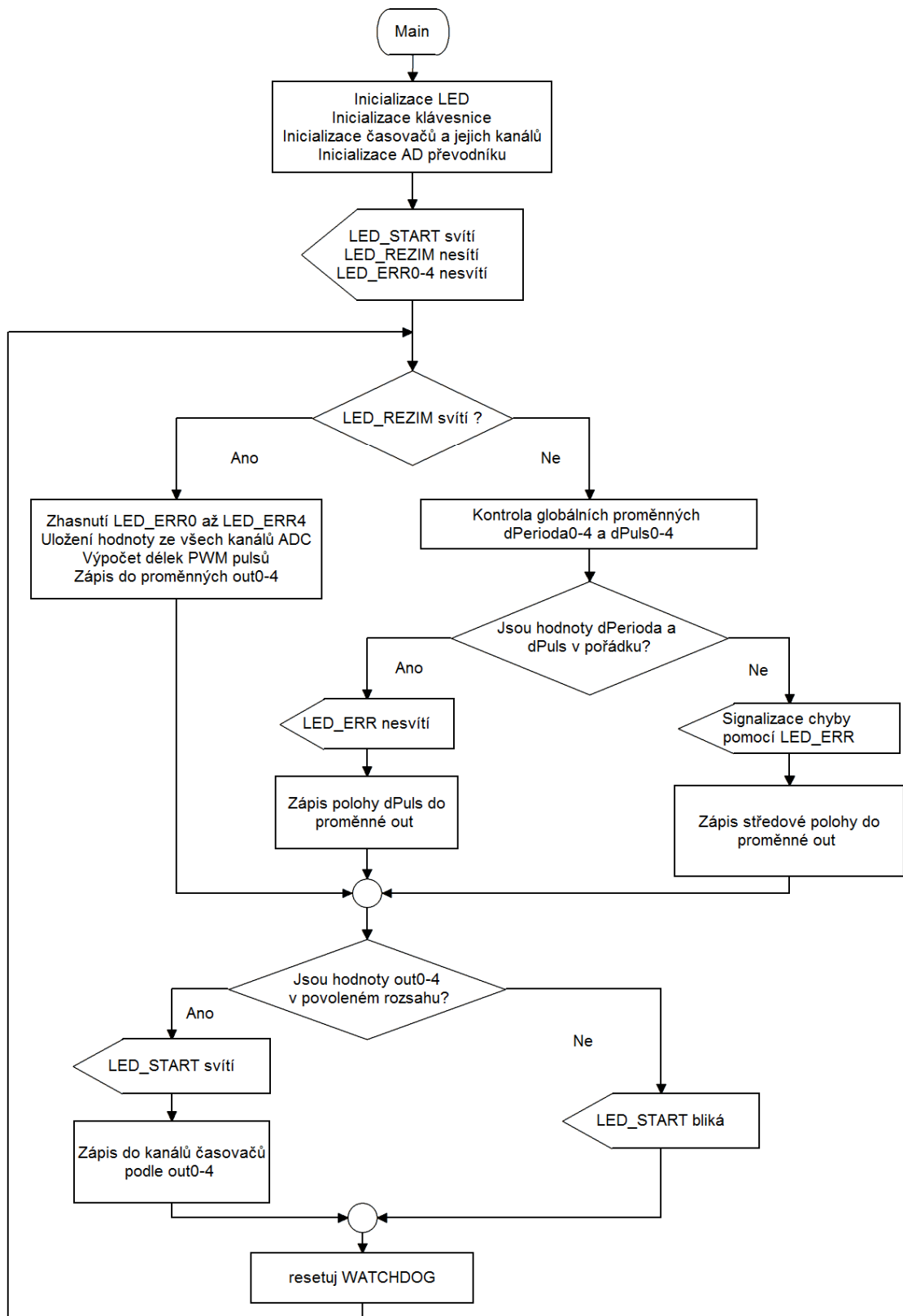
8.2 Popis firmware v modulu rozhraní

8.2.1 Hlavní funkce main

Nejdříve proběhne inicializace oscilátoru a nastavení frekvence sběrnice na hodnotu 4 MHz. Následuje inicializace portu A jako výstup pro připojené LED, inicializace tlačítka pro změnu režimu - nastavení jako KBI vstupu a nastavení všech časovačů, tak aby jejich doba přetečení byla 20 ms. Pět kanálů časovačů připojených k vývojovému kitu přes MCU_PORT je nastaveno do režimu „zachycení vstupu“ a pět kanálů časovače připojených na výstup přes budič sběrnice je nastaveno do režimu „PWM“. Piny mikropočítače, na které jsou připojeny potenciometry, jsou nastaveny jako vstupy pro AD převodník. Když je vše potřebné nastaveno, rozsvítí se LED START. Po inicializaci všech potřebných periférií se zjistí zda je aktivní režim ovládání robota pomocí potenciometrů, nebo pomocí vývojového kitu.

V režimu ovládání robota pomocí potenciometrů budou zhasnuty všechny LED ERR, protože se na výstupu nemůže objevit špatný PWM signál. Proběhne získání hodnot z pěti kanálů AD převodníku, jejich přepočítání na délku pulsu PWM signálu a uložení do pomocných proměnných out0-4 k dalšímu vyhodnocení.

V režimu ovládání robota pomocí vývojového kitu proběhne kontrola hodnot délek period a pulsů všech pěti vstupních PWM signálů z vývojového kitu. Tyto hodnoty byly spočítány v obsluhách přerušování při události „zachycení vstupu“ na jednotlivých kanálech časovačů a uloženy do globální proměnné. Pokud jsou zjištěné parametry u daného PWM signálu v pořádku, zkopíruje se délka pulsu do příslušné pomocné proměnné out. Pokud šířka pulsu, nebo doba periody či dokonce obě hodnoty nesplňují podmínky korektního PWM signálu pro řízení servopohonů, je tato skutečnost signalizována pomocí LED ERR a v proměnné out zůstává poslední korektní hodnota.

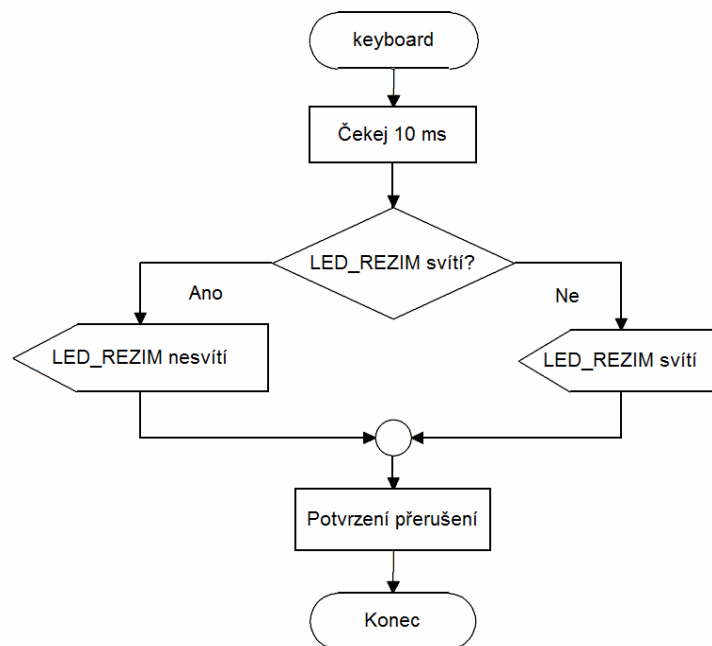


Obr. 41 Vývojový diagram – modul rozhraní – funkce main

Když už jsou všechny hodnoty odpovídající poloze servopohonů uloženy v proměnných out0-4, může proběhnout kontrola povolené polohy robota. Tím je zajištěno, že stejná kontrola povolené polohy proběhne nezávisle na zvoleném režimu. Kontrola probíhá tak, že se nejdříve vyhodnotí zda poloha ramene robota, může vést k havárii celého robota. Pokud ne, nastaví se pomocná proměnná povolenaPoloha na hodnotu 1, pokud ano, nastaví se povolenaPoloha na hodnotu 0, pokud nelze rozhodnout na základě polohy ramene, testuje se stejným způsobem poloha loktu a pokud ani na základě polohy loktu nelze rozhodnout o možné havárii robota, testuje se ještě poloha zápěstí. Posledním kontrolovaným pohonem je chapadlo robota, které nesmí být úplně sevřené či rozevřené, aby nedošlo k mechanickému poškození tohoto pohonu. Povolené a zakázané polohy byly zjištěny experimentálně. Pokud je po kontrole povolené polohy robota nastavená povolenaPoloha na hodnotu 1, dojde k překopírování hodnot z proměnných out0-out4 do příslušných registrů kanálů časovačů a tím i k nastavení střídý PWM řídicích signálů. Správná poloha robota je signalizována nepřerušovaným rozsvícením LED START. Pokud je po kontrole povolené polohy robota nastavená povolenaPoloha na hodnotu 0, nedochází ke změně střídý výstupních PWM signálů. Tento stav je signalizován blikáním LED START. Proběhne reset WATCHDOG a celý program se opakuje.

8.2.2 Obsluha přerušování při stisku tlačítka

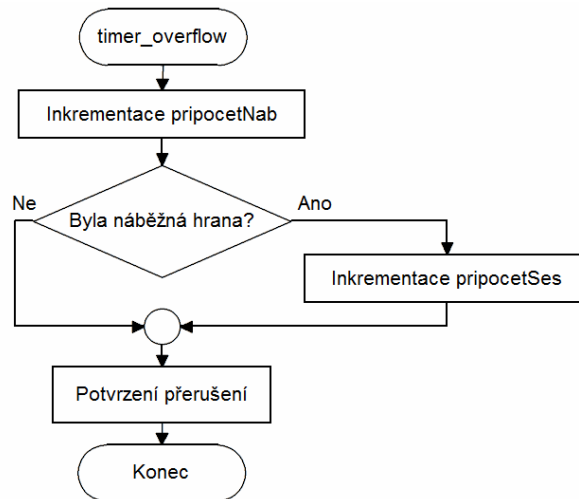
Při stisknutí tlačítka REŽIM se rozsvítí LED REŽIM, pokud byla zhasnutá. V opačném případě, pokud LED REŽIM svítila, pak zhasne. Obsluha přerušování končí po potvrzení příjmu přerušování.



Obr. 42 Vývojový diagram – modul rozhraní – obsluha přerušení při stisku tlačítka

8.2.3 Obsluha přerušení při přetečení časovače

Při přetečení časovače se spustí obsluha přerušení, kde dojde k inkrementaci pomocných proměnných pripocetNab0-4 a pokud je na vstupu kanálu časovače logická úroveň 0 (byla sestupná hrana), dojde navíc k inkrementaci pomocných proměnných pripocetSes0-4. Smysl toho způsobu inkrementace pomocných proměnných bude vysvětlen níže v popisu obsluhy přerušení při události „zachycení vstupu“. Obsluha přerušení končí po potvrzení příjmu přerušení.



Obr. 43 Vývojový diagram – modul rozhraní
– obsluha přerušení při přetečení časovače

8.2.4 Obsluha přerušení při zachycení vstupu

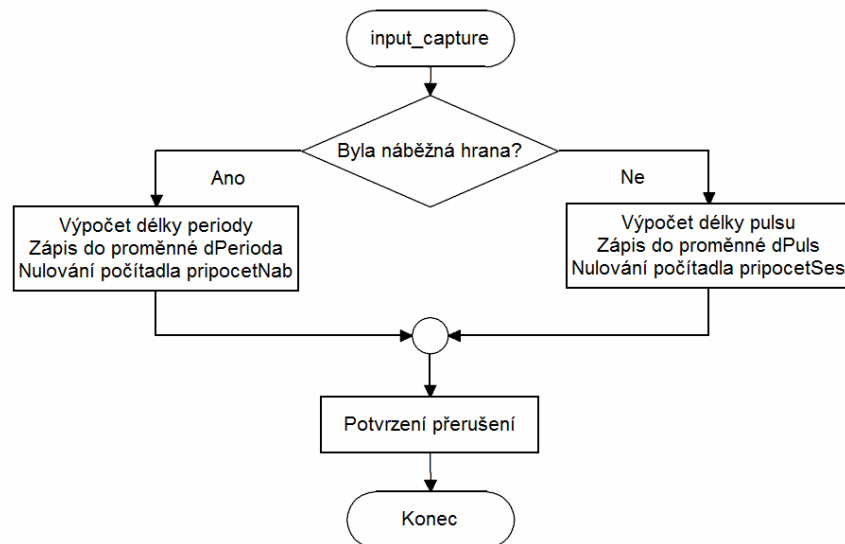
V obsluze přerušení při události „zachycení vstupu“ probíhá měření délky periody a pulsu vstupního PWM signálu z vývojového kitu. Na začátku obsluhy se zjistí, zda byla na vstupu kanálu časovače náběžná hrana a provede se výpočet délky periody, nebo byla sestupná hrana a provede se výpočet délky pulsu.

Při náběžné hraně se vezme hodnota v proměnné $tStart$, která byla v časovači při minulé náběžné hraně a přičte se k ní hodnota při které dochází k přetečení časovače vynásobená proměnnou $pripocetNab$, která se inkrementuje v obsluze přerušení při přetečení časovače. Po odečtení aktuální hodnoty na kanálu časovače, se získá číslo, které odpovídá počtu pulsů, které přišly na časovač, během trvání periody. Po výpočtu se uloží do proměnné $tStart$ hodnota časovače, při které došlo k přerušení a vynuluje se pomocná proměnná $pripocetNab$.

Měření délky pulsu probíhá podobně. Pokud byla na vstupu sestupná hrana PWM signálu, tak se jednoduše od hodnoty $tStart$ odečte aktuální hodnota časovače. Pokud by během doby trvání řídicího pulsu nastalo přetečení časovače, inkrementuje se v obsluze přerušení pomocná proměnná $pripocetSes$ a k hodnotě $tStart$ se připočítá ještě hodnota při které dochází k přetečení časovače. Pomocná proměnná $pripocetSes$ se inkrementuje v obsluze přerušení při přetečení časovače, pouze v případě, když je na vstupu kanálu časovače řídicí puls PWM signálu. Pokud by se pomocná proměnná $pripocetSes$ inkrementovala při

každém přerušení, docházelo by při výpočtu pulsu velmi často (vždy když by přetečení časovače nastalo mimo dobu trvání řídicího pulsu) k chybě odpovídající délce celé periody PWM signálu. Na závěr se vynuluje pomocná proměnná `pripocetSes`.

Obsluha přerušení končí po potvrzení příjmu přerušení.



Obr. 44 Vývojový diagram – modul rozhraní – obsluha přerušení při zachycení vstupu

8.3 Popis knihovny pro práci s PWM a robotem

Byla nasána knihovna funkcí, s níž lze na vývojovém kitu jednoduše generovat až pět PWM signálů o libovolné periodě od 1 do 419 ms. Knihovna také obsahuje funkce pro generování PWM signálů, které jsou vhodné pro řízení serv a funkce, které umožňují plynulou změnu jejich poloh. Následuje popis funkcí v knihovně `pwm.h`. Tato knihovna pracuje s časovačem TPM2 a používá všechny jeho kanály. Proto by se měl programátor ve svém programu vyhnout použití tohoto časovače a pinů PTD3 – PTD7, které sdílejí kanály tohoto časovače.

Čekej 100 ms

```
void cekej_100ms(void);
```

Tato funkce zastaví běh programu asi na 100 ms. Tato funkce pouze spotřebovává cykly mikropočítače.

Inicializace PWM

```
int pwm_init(int perioda);
```

Tato funkce slouží k nastavení periody generovaného PWM signálu. Jako parametr funkce můžete zadat číslo v rozsahu 1 – 419, které reprezentuje délku trvání periody v ms. Automaticky se nastaví prescaler a modulo registr. Všechny kanály TPM2 se nastaví do módu PWM. V případě úspěšného nastavení funkce vrací hodnotu 1. Pokud je špatně zadán vstupní parametr, vrací funkce hodnotu 0.

Nastavení střídání PWM

```
int pwm_duty_cycle(int kanal, int sirka);
```

Před použitím této funkce je potřeba provést inicializaci PWM. Pomocí této funkce můžete nastavit střídání PWM signálu na kterémkoli z pěti kanálů TPM2. V prvním parametru zadáte číslo kanálu, které může být v rozmezí 0 – 4 a v druhém parametru zadáte střídání v rozsahu 0 – 10000, kde 0 odpovídá hodnotě 0% a 10000 odpovídá 100%. Na základě hodnoty v modulo registru a druhého parametru funkce se provede výpočet hodnoty, která se uloží do příslušného registru vybraného kanálu časovače, který určuje střídání PWM signálu. Pokud byl některý z parametrů funkce zadán špatně, vrací funkce hodnotu 0. Po výpočtu a nastavení registru kanálu časovače vrátí funkce hodnotu 1.

Inicializace PWM pro řízení polohy serva

```
int servo_init(void);
```

Pokud budete chtít řídit servo, můžete použít pro inicializaci tuto funkci. Po zavolání této funkce se nejdříve zavolá funkce „pwm_init(20)“, která nastaví dobu přetečení TPM2 a tím i periodu PWM signálu na 20 ms a všechny kanály tohoto časovače do režimu PWM. Poté se nastaví střídání signálu na všech jeho kanálech na 7,5%. Délka pulsu je tedy 1,5 ms, což odpovídá středové poloze u většiny serv. Pokud nastavení časovače TPM2 proběhlo v pořádku, funkce vrací hodnotu 1. Pokud nastavení časovače TPM2 neproběhlo v pořádku funkce vrací hodnotu 0.

Nastavení střídání PWM pro řízení polohy serva

```
int servo_set(int kanal,int poloha);
```

Před použitím této funkce je nutné provést inicializaci PWM s délkou trvání periody o hodnotě 20 ms. V nejlepším případě použít inicializaci pomocí funkce „int servo_init(void);“. Jako první parametr funkce je číslo kanálu v rozsahu 0 – 4, na kterém chcete změnit střídání PWM signálu. Druhým parametrem nastavíte šířku řídicího pulsu. 0 odpovídá délce řídicího signálu 1 ms, tj. plné natočení serva vlevo a 500 odpovídá délce řídicího signálu 2 ms, tj. plné natočení serva vpravo. Funkce zkontroluje zadané parametry a pokud jsou mimo povolené meze, vrátí se hodnota 0. Pokud proběhne kontrola parametrů v pořádku nastaví se střídání PWM signálu zavoláním funkce „pwm_duty_cycle (kanal,poloha+500)“. Pokud nastavení střídání signálu proběhlo v pořádku, vrací funkce hodnotu 1.

Získání informace o poloze serva

```
int servo_get(int kanal);
```

Pomocí této funkce můžete snadno zjistit v jaké poloze se servo právě nachází. Pozor, tuto funkci můžete použít, pouze pokud jste provedli správnou inicializaci PWM pro řízení serva, v opačném případě vám bude funkce vracet nesmyslné hodnoty. Funkce vrací hodnoty v rozsahu 0 – 500, kde hodnota 0 odpovídá levé krajní poloze serva a hodnota 500 odpovídá pravé krajní poloze serva. Podle zadaného parametru funkce se z registru kanálu časovače spočítá poloha serva odpovídající generovanému PWM signálu. Pokud je parametr funkce zadán mimo povolený rozsah vrací funkce hodnotu -1.

Nastavení rychlosti pohybu serv při plynulé změně jejich polohy

```
int servo_speed(char speed);
```

Tato funkce slouží k nastavení rychlosti pohybu serv v případě, že využijete některou z funkcí pro plynulou změnu polohy serv „int servo_move(int kanal, int pozPoloha)“ nebo „int robot_move(int kan0, int kan1, int kan2, int kan3, int kan4)“. Jako parametr funkce zadáte číslo v rozsahu 0 – 5. Při hodnotě parametru 0 dojde k zastavení pohybu serv. Rychlost při hodnotě parametru 1 je nejnižší možná. Při této rychlosti se servo otočí z jedné krajní polohy do druhé za 10 sekund. Každý vyšší stupeň rychlosti je dvakrát

rychlejší. Rychlostní stupně a časy potřebné k tomu aby se servo otočilo z jedné krajní polohy do druhé jsou uvedeny v následující tabulce:

Tab. 4 Rychlosti pohybu serv

Rychlost [-]	Čas [s]
0	Serva se nepohybují
1	25
2	12,5
3	8,33
4	6,25

Pokud je zadán parametr funkce mimo povolený rozsah vrací funkce hodnotu 0. Pokud je parametr funkce v pořádku nastaví se globální proměnná rychlost na příslušnou hodnotu a funkce vrátí hodnotu 1.

Plynulá změna polohy serva

```
int servo_move(int kanal, int pozPoloha);
```

Umožňuje plynule natočit servo do požadované polohy. Před zavoláním této funkce je potřeba nastavit rychlost pohybu pomocí funkce „int servo_speed(char speed)“. Mohlo by se stát, že bude rychlost nastavena na nulu a k žádnému pohybu by nedocházelo. Jako první parametr funkce zadáte kanál časovače, který řídí činnost serva a jako druhý parametr polohu serva v rozsahu 0 – 500. V této funkci se zapíše požadovaná hodnota do globální proměnné pozPol a nastaví se poloha serva podle globální proměnné novaPol, která je pravidelně přepočítávána v obsluze přerušení při přetečení časovače.

Plynulý pohyb celým robotem

```
int robot_move(int kan0, int kan1, int kan2, int kan3, int kan4);
```

Tato funkce slouží pro plynulou změnu polohy pěti serv současně. To je výhodné při ovládání robotické paže složené z pěti servopohonů. Uživatel nemusí volat pětkrát funkci „int servo_move(int kanal, int pozPoloha)“, ale zavolá pouze tuto funkci, která má pět parametrů. Pořadí parametrů funkce odpovídá pořadí čísel kanálů časovače. Rychlost

pohybu serv se nastavuje úplně stejně jako u funkce „int servo_move(int kanal, int pozPoloha)“. Na začátku funkce proběhne kontrola vstupu. Pokud je některá z požadovaných hodnot mimo rozsah, vrátí funkce hodnotu 0. Pokud je vše v povolených mezích, zavolá se pro každý kanál funkce „int servo_move(int kanal, int pozPoloha)“. Tím dojde k pohybu celého robota.

Test zda už je robot v požadované poloze

```
int is_final_position(void);
```

Tato funkce testuje zda se už robot dostal do polohy zadané v parametrech funkce „int robot_move(int kan0, int kan1, int kan2, int kan3, int kan4)“. Funkce pouze porovná globální proměnné pozPol0-4 s hodnotami získané pomocí funkce „int servo_get(int kanal)“. Pokud se robot nachází v požadované poloze vrací funkce hodnotu 1, pokud se robot v žádané poloze nenachází, vrací funkce hodnotu 0.

Výpočet postupné změny polohy servopohonu

Při zavolání funkce „int servo_move(int kanal, int pozPoloha)“, nebo „int robot_move(int kan0, int kan1, int kan2, int kan3, int kan4)“ se provede zápis do globální proměnné pozPol a kanál časovače se nastaví pomocí globální proměnné novaPol, která se počítá v obsluze přerušení od přetečení časovače. V této obsluze se nejdříve zjistí aktuální poloha servopohonu pomocí funkce „int servo_get(int kanal)“. Pokud je aktuální hodnota menší, než požadovaná hodnota uložená v globální proměnné pozPol, dojde k jejímu navýšení o hodnotu uloženou v globální proměnné rychlost. Pokud je aktuální hodnota větší, než požadovaná hodnota uložená v globální proměnné pozPol, dojde k jejímu ponížení o hodnotu uloženou v globální proměnné rychlost. Pokud je absolutní hodnota rozdílu požadované a aktuální hodnoty menší než hodnota v proměnné rychlost, dojde přímo k nastavení nové polohy na žádanou polohu. Tím je ošetřen stav, kdy by rychlost byla větší než absolutní hodnota rozdílu aktuální a požadované hodnoty, kde by neustále nová poloha kmitala na dvou hodnotách pod a nad požadovanou hodnotou.

8.4 Ukázkový program pro vývojový kit

Protože výsledkem této práce by měla být nová závěrečná úloha do předmětu Programování mikropočítačů, bylo vymyšleno ukázkové zadání projektu i s jeho řešením, na kterém je ověřena správná činnost knihovny pwm.h i modulu rozhraní.

8.4.1 Zadání příkladu

Vytvořte program pro vývojový kit 68EVB08GB60, pomocí kterého bude možné ovládat činnost robota LYNX 5. Tlačítko SW1 bude sloužit pro výběr režimu. Režim může být buď manuální, nebo automatický.

➤ Manuální režim:

- ✓ Na prvním řádku displeje bude zobrazen text „Manuální“.
- ✓ Potenciometr bude sloužit k ovládání pohonu základny robota. Na konci prvního řádku bude vypsána délka pulsu PWM signálu, který řídí otáčení základnou. Ostatní pohony robota se budou ovládat pomocí tlačítek.
- ✓ Tlačítko SW3 bude sloužit pro volbu pohonu, který se bude ovládat. Na druhém řádku displeje bude zobrazen název ovládaného pohonu („chapadlo“, „zápěstí“, „loket“, „rameno“) a délka řídicího pulsu PWM signálu pro vybraný pohon.
- ✓ Tlačítkem SW4 se bude přepínat mezi pohybem doleva (první krajní poloha) a pohybem doprava (2. krajní poloha).
- ✓ Pokud bude aktivní pohyb vybraného pohonu doleva bude tento stav signalizován pomocí rozsvícení LED2. V opačném případě, pokud bude aktivní pohyb pohonu doprava bude svítit LED3.
- ✓ Tlačítkem SW2 se zastaví / spustí vybraný pohon.
- ✓ Pokud vybraný pohon běží bude tento stav signalizován pomocí rozsvícení LED1, pokud bude pohon zastaven bude LED1 zhasnuta.

➤ Automatický režim

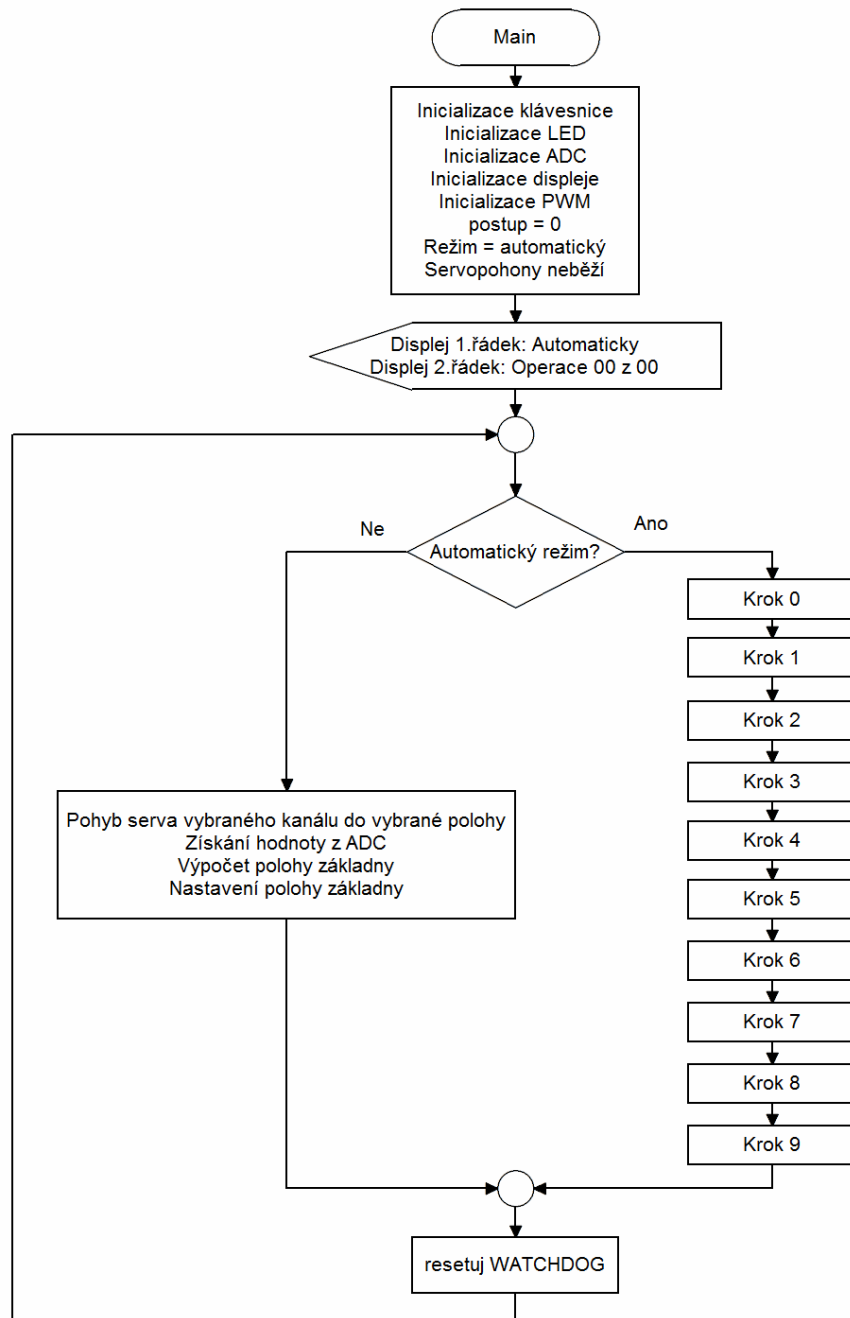
- ✓ Využijte svou představivost a naprogramujte robota na libovolnou automatickou opakující se činnost (přemísťování předmětu z bodu A do bodu B). Fantazii se meze nekladou.
- ✓ Na prvním řádku displeje bude zobrazen text „Automatický“.
- ✓ Na druhém řádku displeje bude zobrazena informace o postupu prováděné činnosti (např. „Postup: 42%“, nebo „Operace 12 z 17“).
- ✓ Tlačítkem SW2 bude možnost robota pozastavit. Při opětovném spuštění tohoto tlačítka bude robot pokračovat ve své činnosti přesně tam kde skončil.
- ✓ Signalizace zastavení robota bude signalizována stejně jako v manuálním režimu pomocí LED1.
- ✓ LED2 a LED3 nebudou mít v automatickém režimu žádnou signalizační funkci a proto budou zhasnuté.
- ✓ Tlačítka SW3 a SW4 nebudou mít na chování robota v automatickém režimu žádný vliv.

8.4.2 Řešení příkladu – funkce main

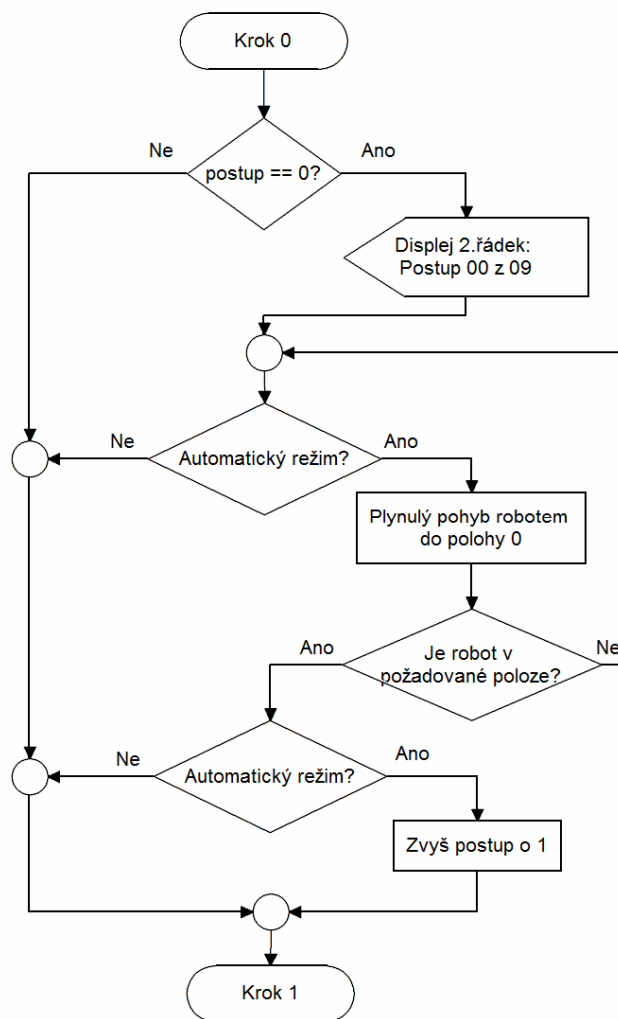
Na začátku hlavní funkce proběhne inicializace všech potřebných periférií mikropočítače. Režim je nastaven na automatický a servopohony jsou zastaveny. Podle zvoleného režimu řízení se vykonává jedna ze dvou hlavních větví programu.

V manuálním režimu se provádí plynulý pohyb vybraného serva do zvolené krajní polohy, dokud nedojde v obsluze přerušení od tlačítka ke změně, nebo zastavení pohonu. Z AD převodníku se získá hodnota, podle které se nastaví poloha základny robota. Tato činnost se cyklicky opakuje dokud nedojde k přepnutí režimu.

Automatický režim se skládá z několika za sebou jdoucích kroků, které se postupně vykonávají. Po skončení posledního kroku se zjistí, zda nedošlo během automatického režimu k přepnutí na manuální režim. V případě, že je automatický režim stále aktivní pokračuje se opět nultým krokem.



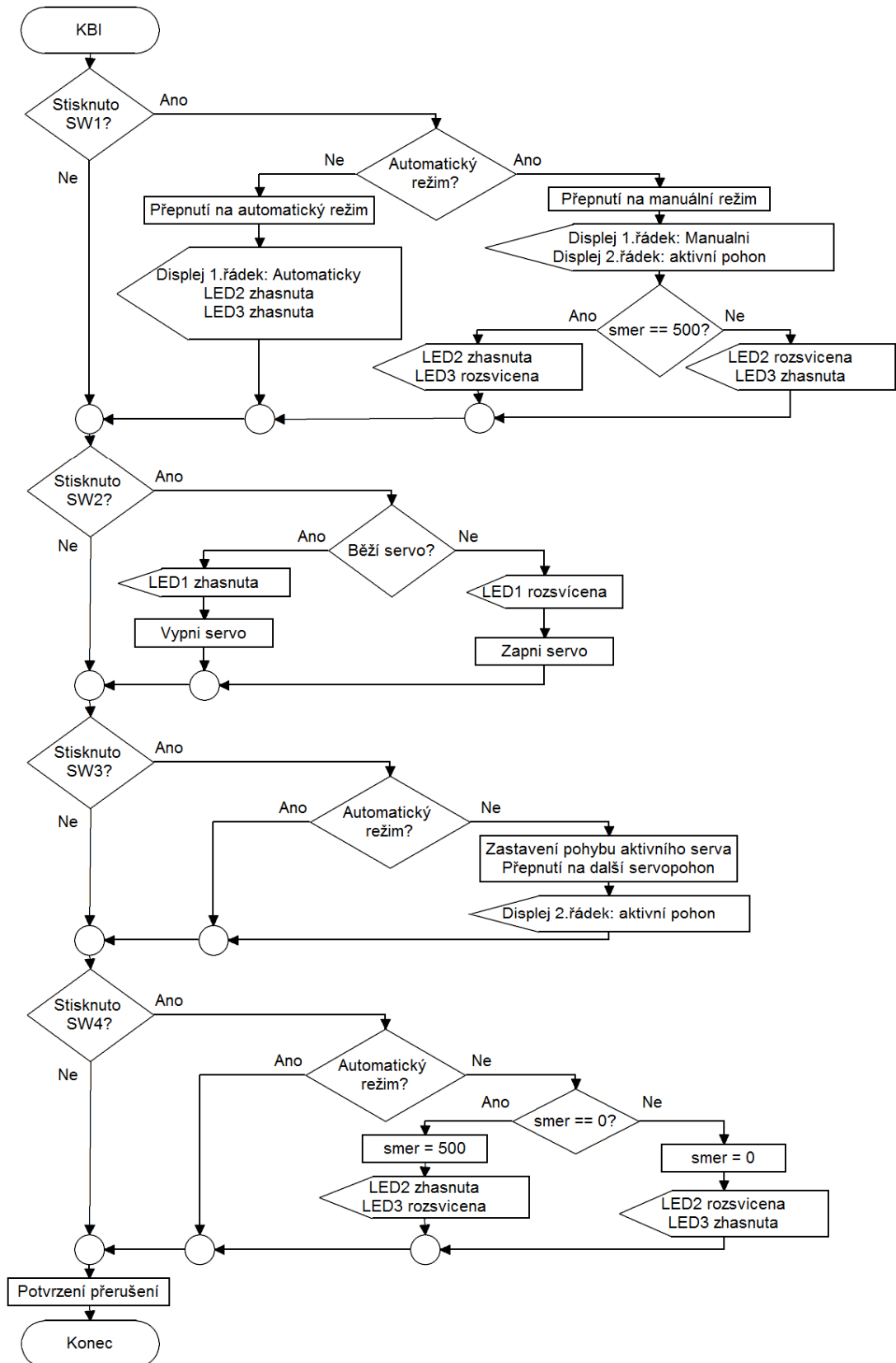
Obr. 45 Vývojový diagram – vývojový kit – funkce main



Obr. 46 Vývojový diagram – vývojový kit – krok v automatickém režimu

V proměnné postup je uložena hodnota, která odpovídá aktuálně prováděnému kroku. Na konci každého kroku se tato hodnota zvýší o jedničku, což umožní pokračovat na dalším krokem. V posledním kroku dojde k nulování proměnné postup čímž je zajištěno cyklické vykonávání pohybů robota. Pokud dojde ke změně režimu na manuální režim, zůstane aktuální krok uložen a při zpětném přepnutí do automatického režimu robot přeskočí již vykonané bloky a pokračuje do polohy, kde se měl dostat před změnou režimu. V každém kroku se robot pohybuje do požadované polohy, dokud požadované polohy nedosáhne, nebo nedojde ke změně režimu.

8.4.3 Řešení příkladu – obsluha přerušení při stisku tlačítka



Obr. 47 Vývojový diagram – vývojový kit – obsluha přerušení při stisku tlačítka

V obsluze přerušení od modulu KBI se nejdříve otestuje, které tlačítko bylo stisknuto a podle toho se provede příslušná akce.

Po stisknutí SW1 dojde k přepnutí režimu. Tento stav je signalizován pomocí výpisu na 1.řádek displeje. Pokud byl režim přepnut do manuálního režimu, je na druhém řádku displeje zobrazen název vybraného servopohonu a rozsvítí se LED2, nebo LED3, podle toho do které krajní polohy se bude vybraný pohon po zapnutí pohybovat. V automatickém režimu jsou LED2 a LED3 zhasnuty.

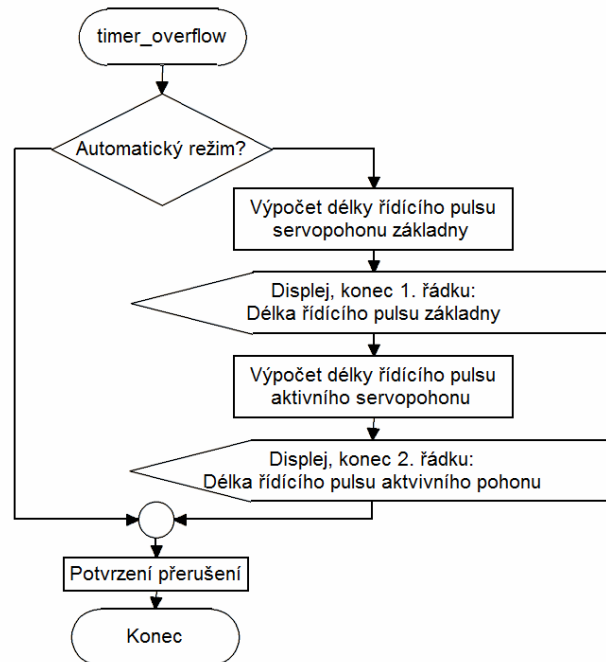
Při stisku SW2 se pozastaví, nebo opět spustí pohyb všech serv nastavením rychlosti na hodnotu „0“, nebo „2“.

Tlačítko SW3 slouží k volbě servopohonu v manuálním režimu. Pokud je po stisku zjištěno, že je aktivní automatický režim, obsluha končí a nic dalšího se neděje. V opačném případě, pokud je aktivní manuální režim, dojde k přepnutí na další pohon a výpisu názvu zvoleného pohonu na druhý řádek displeje.

Tlačítko SW4 slouží k volbě krajní polohy v manuálním režimu, do které se bude zvolený pohon pohybovat. Pokud je po stisku zjištěno, že je aktivní automatický režim, obsluha končí a nic dalšího se neděje. V opačném případě, pokud je aktivní manuální režim, dojde k přepnutí požadované krajní polohy a signalizaci pomocí právě jedné LED2, nebo LED3.

8.4.4 Řešení příkladu – obsluha přerušení při přetečení časovače

V obsluze přerušení časovače je aktivní pouze automatickém režimu výpočet délek řídicích pulsů PWM signálů pro základnu a další vybraný pohon a jejich tisk na konec prvního a druhého řádku displeje. Po potvrzení přerušení obsluha končí.



Obr. 48 Vývojový diagram – vývojový kit –
obsluha přerušení při přetečení časovače

8.5 Ověření funkce modulu a knihovny na ukázkovém programu

Po Správnou funkci modulu a knihovny „pwm.h“ jsem ověřil na ukázkovém programu pro vývojový kit. Pomocí tlačítek jsem si v ukázkovém programu zvolil zda chci manuální, nebo automatické řízení.

V manuálním režimu se snímá hodnota z AD převodníku a po přepočtu se ihned nastaví poloha základny pomocí funkce „int servo_set(int kanal,int poloha)“. Pomocí dalších dvou tlačítek se volí jeden ze čtyřech pohonů a cílová poloha aktivního pohonu pomocí funkce „int servo_move(int kanal, int pozPoloha)“. Zvolený servopohon se postupně otáčí do požadované polohy. Před přepnutím kanálu se nejdříve zavolala funkce „int servo_get(int kanal)“ pro současný kanál a takto získaná hodnota aktuální polohy serva se nastavila pomocí „int servo_move(int kanal, int pozPoloha)“ na požadovanou hodnotu, čímž došlo k zastavení serva. Pomocí posledního tlačítka byla možnost zastavit a spustit pohon pomocí funkce pro změnu rychlosti servopohonů „int servo_speed(char speed)“, která se zavolala s parametrem 0 pro zastavení, nebo s parametrem 2 pro pohyb.

V automatickém režimu se robot pohyboval do zadané polohy pomocí funkce „int robot_move(int kan0, int kan1, int kan2, int kan3, int kan4)“, dokud nebyla vyhodnocena

konečná poloha pomocí funkce „int is_final_position(void)“. Poté se robot pohyboval do další polohy dokud opět nebyla splněna podmínka konečné polohy a tak pořád dokola. Tímto způsobem je naprogramována sekvence pohybů robota. Při každém pohybu v automatickém režimu si robot pamatuje aktuální konečnou polohu a kontroluje se zda nebyl přepnut režim na manuální ovládání. Pokud dojde k přepnutí z automatického režimu na manuální a opět zpátky na automatický režim, robot pokračuje v pohybu na poslední požadovanou polohu, na kterou se nestihl před přepnutím režimu dostat.

Správná funkčnost kontroly jednotlivých PWM signálů byla odzkoušena zvlášť v jiném projektu tak, že se pomocí funkcí „int pwm_duty_cycle(int kanal, int sirka)“ a „int pwm_init(int perioda)“ měnila jejich perioda a střída. Kontrola povolených poloh byla ověřena tak, že byl pohyb robota řízen do havarijní polohy nejdříve v manuálním režimu modulu rozhraní a poté ještě pomocí vývojového kitu. Kontrola PWM řídicích signálů a povolených poloh robota fungovala bezchybně.

ZÁVĚR

Cílem bakalářské práce bylo navrhnout a realizovat modul rozhraní mezi vývojovým kitem M68EVB908GB60 a robotem Lynx 5, zajišťující kontrolu PWM řídicích signálů včetně jeho programového vybavení.

Protože je robot složen z modelářských servopohonů, bylo nezbytné si nejdříve nastudovat princip a fungování těchto servopohonů a způsob, jakým lze tyto pohony řídit.

Podle hardwarových požadavků bylo třeba vybrat vhodný mikropočítač, který by řídil činnost celého rozhraní. Byl zvolen mikropočítač MC9S08AC128 od firmy Freescale. Po návrhu základního zapojení byl modul ještě rozšířen o pět potenciometrů a jedno tlačítko, aby bylo možné ovládat robota i bez vývojového kitu.

Pro návrh zapojení a desky plošného spoje byl použit program Eagle 6.4 ve verzi Lite. Po návrhu následovala výroba a osazení desky. Jakmile byla deska hotová, byl napsán firmware, který byl následně pomocí programátoru nahrán do osazeného mikropočítače. Po doladění a odzkoušení správné funkce modulu rozhraní, bylo vytvořeno podpůrné programové vybavení ve formě knihovny podprogramů v jazyce C, která umožňuje snadné nastavení a generování PWM pro řízení servopohonů. Pomocí této knihovny lze snadno řídit činnost robota bez nutnosti rozumět řízení servopohonů pomocí PWM signálu. Veškeré programové vybavení bylo vytvořeno ve vývojovém prostředí CodeWarrior 6.3 SE.

Výsledkem celé práce je modul rozhraní kontrolující řídicí signály PWM generované kitem a polohy robota Lynx 5, aby nemohlo dojít k jeho mechanickému poškození a knihovna podprogramů, která usnadňuje generování PWM signálu pro řízení pohybu robota. Dále pak dokumentace k modulu včetně základního popisu knihovny pro práci s PWM, ukázkovým zadáním a řešením závěrečného projektu do předmětu Programování mikropočítačů.

CONCLUSION

The goal of this bachelor's thesis was design and implement interface module and its firmware which controls the PWM signals between development board M68EVB908GB60 and education robot Lynx 5.

First, it was necessary to study the principle and method of control servos of the robot which is composed.

According hardware requirements has been chosen microcomputer, which will control the interface. Microcontroller MC9S08AC128 from Freescale was chosen. After designing the basic draft of the module has been expanded to include five potentiometers and one button to control the robot without the development board.

For the design of the electric circuit and PCB has been used Eagle software in version 6.4 Light. After the designing followed the manufacturing of PCB and soldering of components on it. As soon as the PCB was done, firmware was loaded into the memory of microcontroller by programmer. After fine-tuning and testing the correct function of interface module has been created supporting software in the form of a subroutine library in C language, which allows easy setup and generation of PWM signals to control the servos. With this library it is possible to easily manage the actions of the robot without having to understand the control servos using PWM signal. All the software has been created in development environment CodeWarrior 6.3 SE.

The result of this work is the interface module checking the PWM control signals generated by the development board and the position of the robot Lynx 5, to prevent mechanical damage and subroutine library that makes it easy to generate a PWM signal to control motion of the robot. As well as documentation for the interface module including a basic description of the library to work with PWM and sample final project and its solution for the subject Microcomputer programming.

SEZNAM POUŽITÉ LITERATURY

- [1] DHservis *Pulzní šířková modulace* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.dhservis.cz/psm.htm>
- [2] ČVUT v Praze – Fakulta elektrotechnická, Katedra měření *Řízení jasů LED displeje pomocí PWM* [online]. 2013 [cit. 2013-03-10]. Dostupné z: http://measure.feld.cvut.cz/cs/vyuka/predmety/x38prm/pwm_output
- [3] Modelfun.cz *Motory: Stejnoseměrné vs. Brushless* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.modelfun.cz/rcmagazin/motory-stejnosmerne-vs-brushless/>
- [4] e-Automatizace *Akční členy – Elektrické pohony* [online]. 2013 [cit. 2013-03-10]. Dostupné z: http://www.e-automatizace.cz/ebooks/ridici_systemy_akcni_cleny/Akc_el.html
- [5] Vysoké učení technické v Brně – Fakulta stavební *Základy elektrotechniky: stejnosměrné motory* [online]. 2013 [cit. 2013-06-05]. Dostupné z: www.fce.vutbr.cz/tst/rada.v/ELTECH/EL-MOTT-SS.DOC
- [6] Filip Sikora, Elektrotechnika *Stejnoseměrné stroje* [online]. 2013 [cit. 2013-06-05]. Dostupné z: <http://elektrotechnika.netstranky.cz/stejnosmerne-stroje/stejnosmerne-stroje.html>
- [7] PohonnaTechnika.cz *Princip – Servo motor* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/servo-motor>
- [8] NOVÁK, Petr *Mobilní roboty: pohony, senzory, řízení. 1. vydání* Praha: BEN – technická literatura, 2005, 248 s. ISBN 80-7300-141-1
- [9] UZIMEX *Malé stejnosměrné motory Maxon* [online]. 2013 [cit. 2013-03-10]. Dostupné z: http://www.uzimex.cz/soubory/20070103_maxon_serial.pdf
- [10] vlastikd Bastlír & spol *Jak fungují modelářská serva* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://vlastikd.webz.cz/bastl/serva.htm>
- [11] RobotBuy.ca *Lynxmotion L5AC-KT 5 Robotic Arm Combo Kit for PC* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.robotbuy.ca/products>

/Lynxmotion_L5AC_KT_5_Robotic_Arm_Combo_Kit_for_PC_serial_port-233-0.html

- [12] GM electronic *Datasheet SOS-AT* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.gme.cz/dokumentace/764/764-368/dsh.764-368.1.pdf>
- [13] Snail Instruments *SOS-AT* [online]. 2013 [cit. 2013-03-10]. Dostupné z: http://shop.snailinstruments.com/index.php?main_page=product_info&products_id=252
- [14] UTB ve Zlíně, Výuka na FAI, kurz Mikropočítače *Vývojový kit Freescale M68EVB908GB60* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://vyuka.fai.utb.cz/mod/resource/view.php?id=1621>
- [15] Freescale Semiconductor *M68EVB908GB60 Development Board for Freescale MC9S08GB60* [online]. 2013 [cit. 2013-03-10]. Dostupné z: http://cache.freescale.com/files/soft_dev_tools/doc/user_guide/M68EVB908GB60UM.pdf
- [16] Freescale Semiconductor *MC9S08AC128 8-Bit Microcontroller Data Sheet* [online]. 2013 [cit. 2013-03-10]. Dostupné z: http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC9S08AC128.pdf
- [17] P&E Microcomputer Systems *USB BDM Multilink* [online]. 2013 [cit. 2013-03-30]. Dostupné z: http://www.pemicro.com/products/product_viewDetails.cfm?product_id=33
- [18] GM electronic *Datasheet 74HCT245* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.gme.cz/dokumentace/426/426-051/dsh.426-051.1.pdf>
- [19] GM electronic *Datasheet Dip Crystal 11.4x4.8 mm* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.gme.cz/dokumentace/131/131-075/dsh.131-075.1.pdf>
- [20] Eagle Online *Freeware* [online]. 2013 [cit. 2013-03-10]. Dostupné z: <http://www.eagle.cz/freeware.htm>
- [21] Software Informer *CodeWarrior Development Studio for Microcontrollers 6.3* [online]. 2013 [cit. 2013-03-30]. Dostupné z: <http://codewarrior-development-studio-for-micro.software.informer.com/6.3/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AD	Analogově digitální
BDM	Background Debug Mode
COP	Computer Operating Properly
CPU	Central Processing Unit
DPS	Deska plošných spojů
FLL	Frequency-locked loop
I2C	Inter-Integrated Circuit
IO	Integrovaný obvod
IR	Infrared radiation
KBI	Keyboard Interrupt Module
LED	Light-Emitting Diode
MCU	Microprocessor Control Unit
MKO	Monostabilní klopný obvod
PC	Personal computer
PCB	Printed Circuit Board
PM	Permanentně magnetizovaný
PWM	Pulse Width Modulation
RAM	Random-access memory
SCI	Serial Communication Interface
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
UV	Ultraviolet

SEZNAM OBRÁZKŮ

Obr. 1 Průběh napětí při různých střídacích PWM modulace	12
Obr. 2 Princip stejnosměrného motoru	14
Obr. 3 Rozdělení stejnosměrných motorů podle zapojení vinutí.....	15
Obr. 4 Průřez motorem s permanentními magnety	18
Obr. 5 Charakteristika permanentně magnetizovaného motoru.....	19
Obr. 6 Změna otáček motoru vlivem napětí a pomocí převodovky.....	20
Obr. 7 Odkrytovaná převodovka.....	21
Obr. 8 Převodovka s předlohou	22
Obr. 9 Schéma jednoho stupně planetové převodovky.....	22
Obr. 10 Pulsní proudový regulátor (PWM)	23
Obr. 11 Servo Hitec HS-303	23
Obr. 12 Blokové schéma elektroniky servomotoru	24
Obr. 13 Úhel natočení serva v závislosti na délce řídicího impulsu.....	25
Obr. 14 Robot Lynx 5	26
Obr. 15 Modul SOS-AT.....	27
Obr. 16 Periferie vývojového kitu.....	28
Obr. 17 Piny MCU portu	30
Obr. 18 Mikroprocesor MC9S08AC128 v pouzdru 64-QFP	33
Obr. 19 Programátor USB BDM Multilink	34
Obr. 20 a) zapojení pinů, pohled shora b) náhradní schéma vstupu a výstupu.....	35
Obr. 21 Blokové schéma modulu rozhraní	37
Obr. 22 Schéma zapojení napájení a pojistky proti zkratu a přepólování.....	38
Obr. 23 Schéma napájení mikroprocesoru	39
Obr. 24 Schéma připojení krystalu k mikroprocesoru	39
Obr. 25 Schéma zapojení BDM konektoru a tlačítka RESET k mikroprocesoru.....	40
Obr. 26 Schéma zapojení budiče sběrnice	41
Obr. 27 Schéma připojení tlačítka k mikroprocesoru	41
Obr. 28 Schéma připojení potenciometru ke kanálu AD převodníku mikroprocesoru.....	42
Obr. 29 Návrh DPS v programu Eagle	43
Obr. 30 Osazená deska modulu rozhraní	44
Obr. 31 Časový průběh signalizace LED START – robot je v povolené poloze.....	45

Obr. 32 Časový průběh signalizace LED START – robot není v povolené poloze.....	45
Obr. 33 Časový průběh signalizace LED REŽIM – robot je řízen z vývojového kitu	46
Obr. 34 Časový průběh signalizace LED REŽIM – robot je řízen potenciometry	46
Obr. 35 Časový průběh signalizace LED ERR – signál je v pořádku.....	46
Obr. 36 Časový průběh signalizace LED ERR – signál má špatnou délku pulsu.....	46
Obr. 37 Časový průběh signalizace LED ERR – signál má špatnou délku periody	47
Obr. 38 Časový průběh signalizace LED ERR – signál s chybnou délkou periody i pulsu	47
Obr. 39 Připojení servopohonů k modulu rozhraní	47
Obr. 40 Vývojové prostředí CodeWarrior 6.3 SE.....	48
Obr. 41 Vývojový diagram – modul rozhraní – funkce main	50
Obr. 42 Vývojový diagram – modul rozhraní – obsluha přerušení při stisku tlačítka	52
Obr. 43 Vývojový diagram – modul rozhraní – obsluha přerušení při přetečení časovače.....	53
Obr. 44 Vývojový diagram – modul rozhraní – obsluha přerušení při zachycení vstupu	54
Obr. 45 Vývojový diagram – vývojový kit – funkce main	61
Obr. 46 Vývojový diagram – vývojový kit – krok v automatickém režimu	62
Obr. 47 Vývojový diagram – vývojový kit – obsluha přerušení při stisku tlačítka	63
Obr. 48 Vývojový diagram – vývojový kit – obsluha přerušení při přetečení časovače.....	65

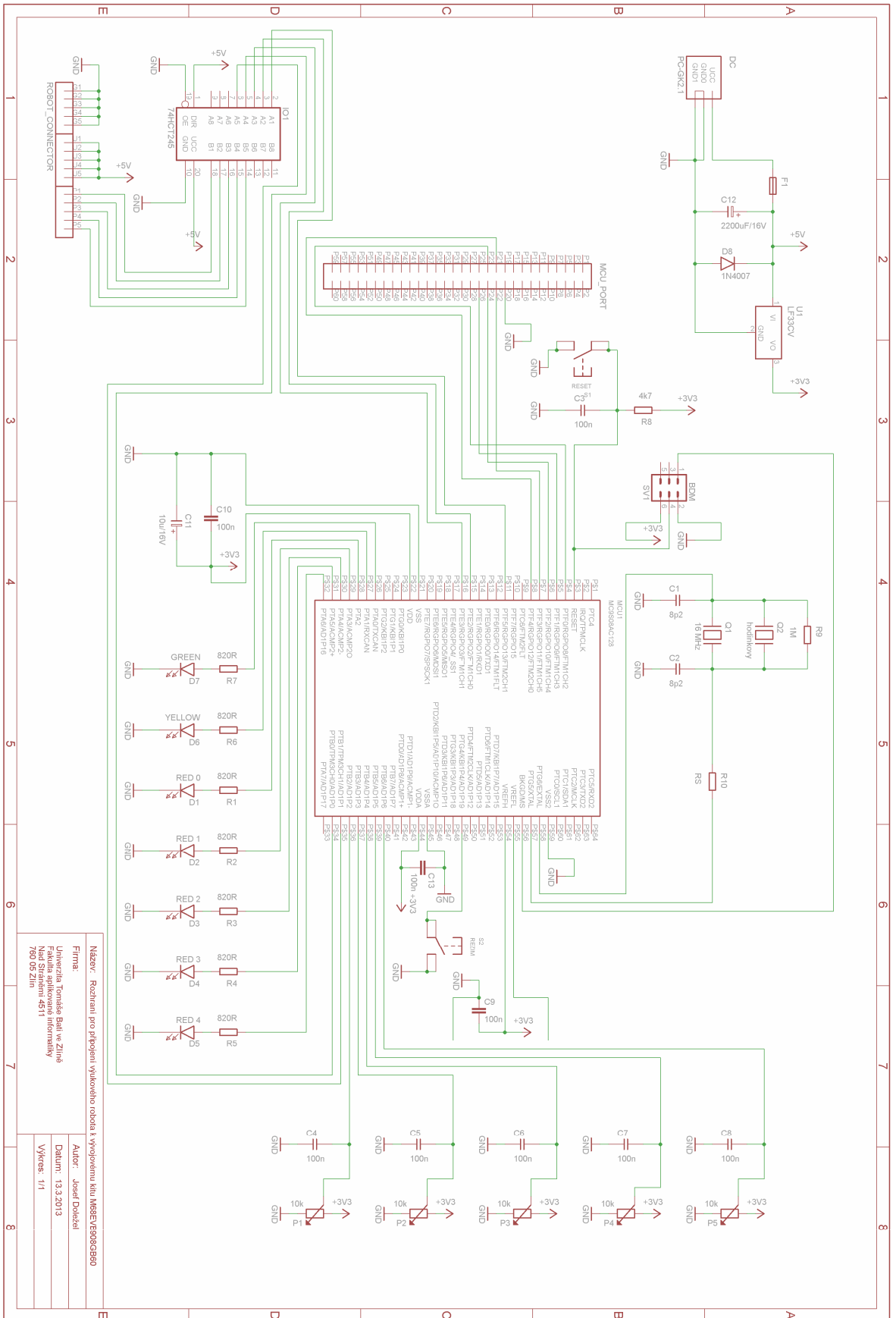
SEZNAM TABULEK

Tab. 1 Význam a zapojení pinů HCT245	36
Tab. 2 Pravdivostní tabulka IO HCT 245	36
Tab. 3 Katalogové hodnoty LED diod a hodnoty předřadných odporů	42
Tab. 4 Rychlosti pohybu serv	57

SEZNAM PŘÍLOH

- P I Schéma zapojení.
- P II Osazovací plánec a obrazec DPS
- P III Seznam součástí pro stavbu modulu
- P IV Fotodokumentace
- P V CD s bakalářskou prací, firmwarem pro robota, knihovnou pwm.h, zadáním závěrečné práce z MPC a ukázkovým řešením závěrečné práce z MPC.

PŘÍLOHA P I: SCHÉMA ZAPOJENÍ



Název: Rozhraní pro přípravu výukového robota k vývojevnému kitu MIBET/VS90SG380

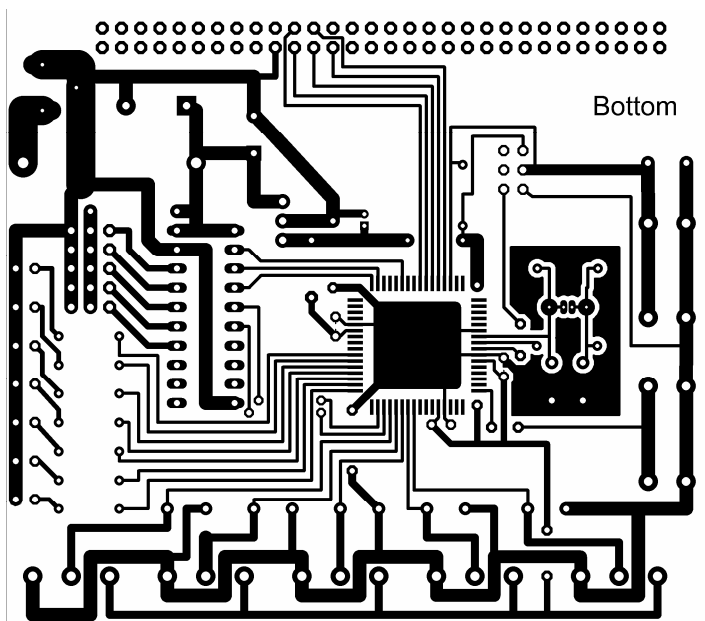
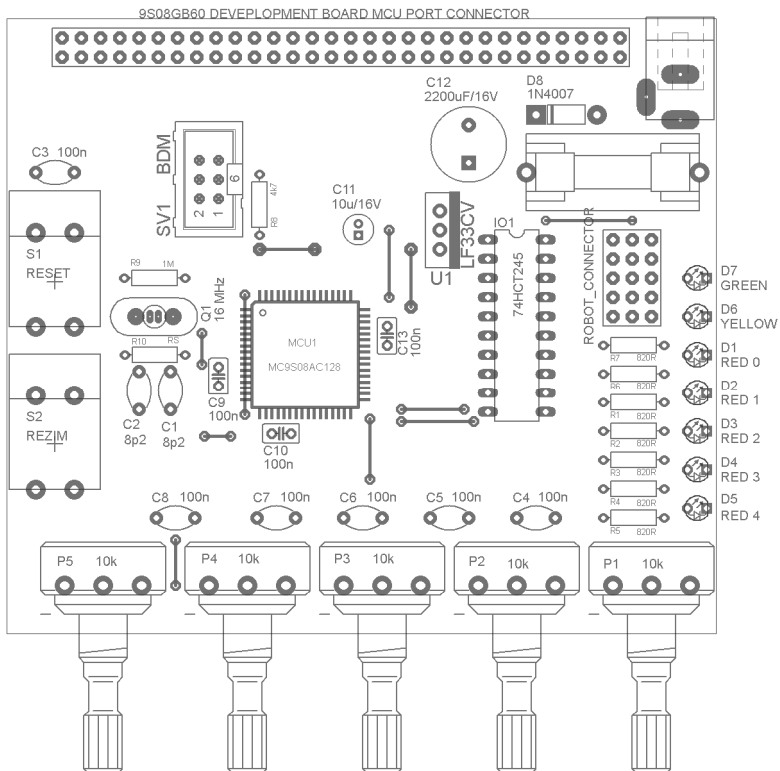
Firma: Univerzita Tomáše Bati ve Zlíně
 Fakulta aplikované informatiky
 760 05 Zlín

Autor: Josef Dožal

Datum: 13.3.2013

Výkes: 1/1

PŘÍLOHA P II: OSAZOVACÍ PLÁNEK A OBRAZEC DPS

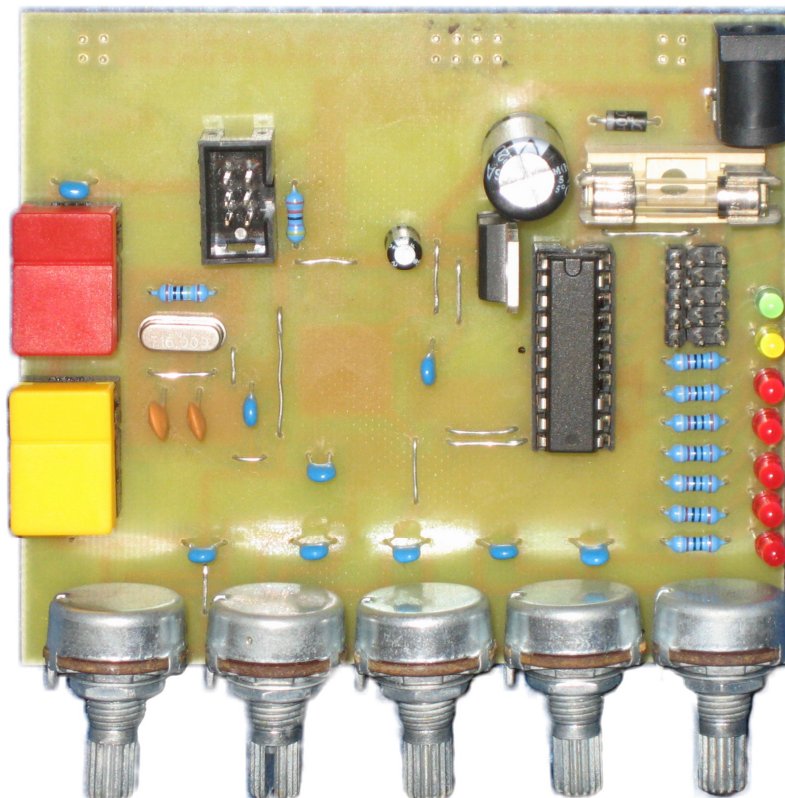
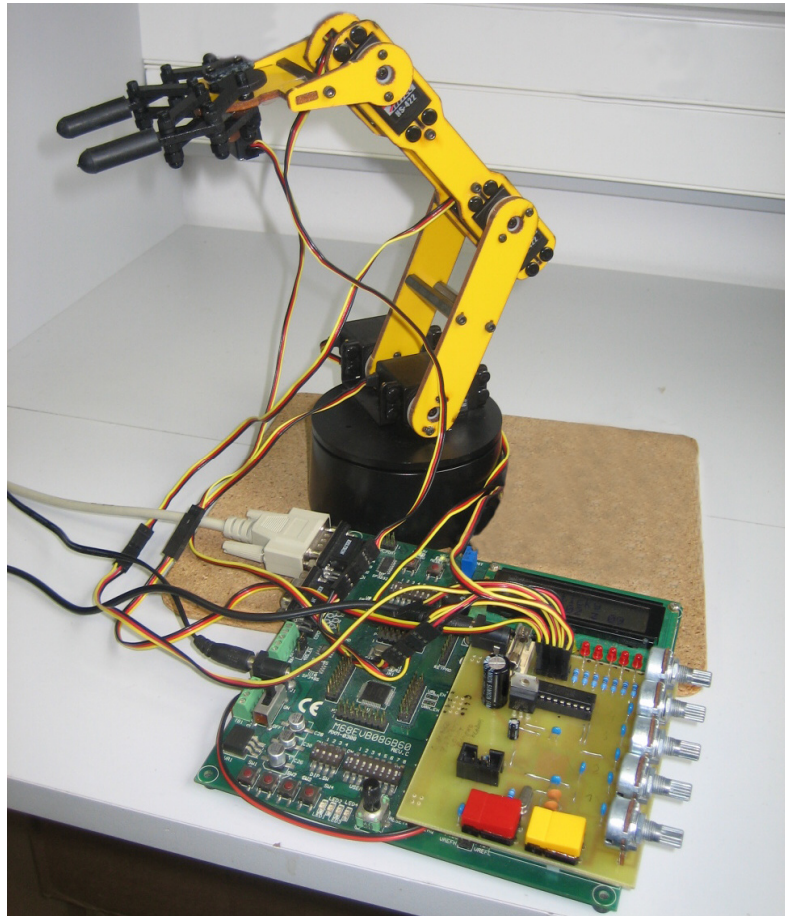


Rozměry desky: 8,12 x 9,39 cm

PŘÍLOHA P III: SEZNAM SOUČÁSTEK PRO STAVBU MODULU

<u>Název</u>	<u>Součástka</u>	<u>Popis</u>	<u>Počet</u>
Mikropočítač	MCU1	MC9S08AC128	1 ks
Budič sběrnice	IO1	74HCT245	1 ks
Patice pro IO	IO1	SOKL 20 (pro DIL 20)	
Stabilizátor napětí	U1	LF33CV (3,3V)	1 ks
Rezistor	R1-R7	820Ω/0,25 W	7 ks
	R8	4,7kΩ/0,25 W	1 ks
	R9	1MΩ/0,25 W	1 ks
	RS	0Ω/ drátová propojka	1 ks
Potenciometr	P1-P5	10kΩ/0,125 W	5 ks
Kondenzátor	C1,C2	22pF/50V	2 ks
	C3-C10,C13	100nF/50V	9 ks
	C11	10μF/16V	1 ks
	C12	2200μF/16V	1 ks
Krystal	Q1	16.000MHZ	1 ks
	Q2	hodinkový (neosazeno)	0 ks
Usměrňovací dioda	D8	1N4007	1 ks
LED dioda	D7	Zelená, 2mA, Ø3mm	1 ks
	D6	Žlutá, 2mA, Ø3mm	1 ks
	D1-D5	Červená, 2mA, Ø3mm	5 ks
Tlačítko	S1	Červené	1 ks
	S2	Žluté	1 ks
Konektor	BDM	MLW06G	1 ks
	MCU PORT	XINYA S2G20 (2x10 lámací)	1 ks
	ROBOT CONNECTOR	XINYA S1G20 (1x20 lámací)	1 ks
Zdroj	-	5V/4A, konektor Ø2,1 mm	1 ks
DC konektor	DC	Souosý 5,5/2,1 mm	1 ks
Pojistka	F1	5x20mm 250V/3,15A	1 ks
Pojistkové pouzdro	F1	SI-HA#112000	1 ks
DPS	-	Cuprexit jednostranný 94x82mm	1 ks

PŘÍLOHA P IV: FOTODOKUMENTACE



PŘÍLOHA P V: CD – ADRESÁŘOVÁ STRUKTURA

Bakalářská práce

Plný text práce v PDF

CD:/fulltext.pdf

Seznam součástí, schéma, DPS, osazovací plánek

Seznam součástí

CD:\01_Soubory_ze_softwaru_Eagle\Seznam_soucastek.pdf

Knihovna součástí, která byla vytvořena při návrhu modulu

CD:\01_Soubory_ze_softwaru_Eagle\Eagle\#_dolezel.lbr

Schéma vytvořené v programu Eagle 6.4

CD:\01_Soubory_ze_softwaru_Eagle\Eagle\interface_pro_robota.sch

DPS vytvořená v programu Eagle 6.4

CD:\01_Soubory_ze_softwaru_Eagle\Eagle\interface_pro_robota.brd

Schéma v grafickém formátu PNG

CD:\01_Soubory_ze_softwaru_Eagle\Export\Schema.png

Osazovací plánek v grafickém formátu PNG

CD:\01_Soubory_ze_softwaru_Eagle\Export\Osazovaci_planek.png

DPS ve formátu PNG v rozlišení 600dpi

CD:\01_Soubory_ze_softwaru_Eagle\Export\DPS_-_600dpi.png

Firmware pro modul rozhraní, knihovna a ukázkové projekty pro vývojový kit

Firmware pro modul rozhraní – projekt vytvořený v CodeWarrior 6.3

CD:\02_Software\Interface_robot.zip

Ovladač v jazyce C pro obsluhu robota a generování PWM signálů

CD:\02_Software\Ovladac_Lnyx5.zip

Ukázkový program, který je popsán v dokumentaci – projekt vytvořený v CodeWarrior 6.3

CD:\02_Software\Demo_projekt.zip

Ukázková závěrečná práce z předmětu MPC – projekt vytvořený v CodeWarrior 6.3

CD:\02_Software\Zaverecny_projekt.zip

Fotodokumentace a návod

Dokumentace a návod k závěrečnému projektu

CD:\03_Dokumentace_a_navod\Dokumentace_a_navod_MPC_Lynx_5.pdf

Modul rozhraní

CD:\03_Dokumentace_a_navod\Modul_rozhrani.jpg

Obrázek celé úlohy

CD:\03_Dokumentace_a_navod\Kompletni.jpg