

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA ELEKTROMECHANIKY A VÝKONOVÉ ELEKTRONIKY

BAKALÁŘSKÁ PRÁCE

Automatizovaný systém detekce barvy

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Luboš TRUNEC**
Osobní číslo: **E11B0261P**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a telekomunikace**
Název tématu: **Automatizovaný systém detekce barvy**
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

Navrhněte konstrukční řešení automatizovaného systému detekce barvy.

1. Navrhněte řídicí rozhraní systému.
2. Navrhněte řídicí algoritmy systému.
3. Implementujte algoritmy do řídicí jednotky a oživte systém.

Rozsah grafických prací: podle doporučení vedoucího
Rozsah pracovní zprávy: 20 - 30 stran
Forma zpracování bakalářské práce: tištěná/elektronická
Seznam odborné literatury:


Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: Ing. Petr Beneš
Katedra elektromechaniky a výkonové elektroniky

Datum zadání bakalářské práce: 14. října 2013
Termín odevzdání bakalářské práce: 9. června 2014


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 14. října 2013

Abstrakt

Tato bakalářská práce se zabývá řízením robotické ruky a automatickou detekcí barvy pomocí navrženého senzoru. Řízení zajišťuje univerzální řídicí jednotka EvB 5.1. Práce je zaměřena na odvětví robotiky a seznámení s problematikou přímé a inverzní kinematiky. Inverzní úloha kinematiky je navržena a řešena přímo pro použitý model robotické ruky a na základě získaných výsledků je robotická ruka schopna najet do libovolného bodu v prostoru. Je zde kladen důraz na řízení a návrh prostředků pro správnou funkci celého systému. Takto sestavený model manipulátoru by měl sloužit jako demonstrační model podobného manipulátoru v praxi.

Klíčová slova

Detekce barvy, mikropočítač, robotika, vývojový kit EvB 5.1, PWM, servo, kinematika, metody programování

Abstract

This bachelor thesis is concerned with control of a robotic arm and automatic color detection using the proposed sensor. Control provides universal control unit EvB 5.1. The work is focused on the robotics industry and on familiarity with the issues of direct kinematics. Inverse kinematics is designed and solved directly for used model of the robotic arm and on the basis of results obtained, the robotic arm is able to move to any point in space. There is an emphasis on control and design resources for proper functioning of the system. Thus assembled manipulator should serve as a demonstration model of a similar manipulator in practice.

Key words

Color detection, microcontroller, robotics, development kit EvB4.3, PWM, servo, kinematics, programming methods

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Litohlavech dne 6.6.2014

Jméno příjmení

.....

Obsah

SEZNAM ZKRATEK A SYMBOLŮ	8
ÚVOD	9
1 ROBOTIKA	10
1.1 STUPNĚ VOLNOSTI	10
1.2 METODY PROGRAMOVÁNÍ ROBOTŮ	11
1.3 KINEMATIKA	12
1.3.1 <i>Přímá úloha kinematiky</i>	12
1.3.2 <i>Inverzní úloha kinematiky</i>	12
2 MODEL AUTOMATIZOVANÉHO SYSTÉMU.....	13
2.1 KONSTRUKCE	13
2.2 POPIS SERVOMECHANIZMU	14
2.3 PRACOVNÍ ČINNOST MODELU.....	16
2.4 PRAKTICKÉ VYUŽITÍ	17
3 ŘÍDICÍ JEDNOTKA	17
3.1 MIKROPOČÍTAČ	17
3.1.1 <i>Základní části mikropočítače</i>	18
3.1.1.1 <i>Adresové prostory</i>	18
3.1.1.2 <i>Periferní obvody</i>	19
3.1.1.3 <i>Procesor</i>	19
3.1.2 <i>Přerušení</i>	21
3.2 VÝVOJOVÝ KIT EVB 5.1	23
4 ŘÍZENÍ.....	24
4.1 ŘÍZENÍ SERVA	24
4.2 VÝPOČET INVERZNÍ ÚLOHY KINEMATIKY PRO MODEL ROBOTICKÉ RUKY.....	28
4.3 SENZOR BARVY.....	32
4.4 PROBLÉMY A JEJICH ŘEŠENÍ.....	37
ZÁVĚR	38
POUŽITÁ LITERATURA.....	39

Seznam zkratk a symbolů

RAM.....	Random access memory, paměť s libovolným výběrem
GPR.....	General purpose register, pracovní registry pro krátkodobé ukládání výsledků
PC.....	Program counter, čítač instrukcí
ID.....	Instruction decoder, instrukční dekodér
RGB	Red, green, blue – barvy diody
OCR1A.....	Porovnávací registr čítače/časovače 1 mikrokontroléru ATmega 644P
TCNT1.....	Registr celkové hodnoty čítače/časovače 1 mikrokontroléru ATmega 644P
CTRL.....	Kontrolní (řídící) sběrnice
ADR.....	Adresová sběrnice
LED.....	Light emitting diode, světelná dioda
PWM.....	Pulse width modulation, pulzně šířková modulace

Úvod

Náhrada člověka ve fyzicky náročných a často také monotónních pracovních činnostech, je jedním z mnoha důvodů nasazení automatizovaných systémů do provozu. Využíváním robotů a manipulátorů v praxi, je také umožněno člověku vyhnout se nebezpečným a toxickým prostředím, která ho ohrožují na zdraví. Tyto důvody jsou důležité z hlediska bezpečnosti člověka, ovšem automatizované systémy mají přínos i z marketingového a technologického hlediska. Díky automatizaci technologických procesů se zvýší produktivita a mnohdy také kvalita výroby. Produktivita je důležitá z hlediska ekonomiky a marketingu, přičemž z technologického hlediska je důležitá kvalita výroby – robot je schopen vyrobit produkt s mnohem větší přesností a precizností než člověk, z čehož vyplývá také snížení nákladů na výrobu. Robotizace tedy dnes už neodmyslitelně patří k procesům výroby.

Automatizovaným systémem detekce barvy je míněno automatické rozpoznání barvy předmětu a jeho následné roztřídění. Rozlišování předmětů podle barev je v průmyslu velmi využíváno, jelikož se jedná o jeden ze základních a přirozených faktorů pro rozlišení předmětu. Třídění barev má uplatnění například při třídění PET lahví, podle barvy obalu, či etikety, třídění potravin, různých barevných výrobků a podobně. Na základě těchto argumentů vzniklo zadání mé bakalářské práce.

Cílem bakalářské práce je seznámit se s odvětvím robotiky, návrhem modelu, řídicí jednotkou a řízením automatizovaného systému detekce barvy. Na základě hodnot získaných z navrženého senzoru barvy, dokáže řídicí jednotka rozeznat danou barvu předmětu a pomocí robotické ruky roztřídit patřičný předmět do kontejneru odpovídající barvy. Základem je tedy řídicí jednotka, která vyhodnocuje rozeznanou barvu předmětu, řídí robotickou ruku pomocí zadaných prostorových souřadnic a v neposlední řadě komunikuje s okolním prostředím pomocí tlačítek, bzučáku a LCD displeje.

Text je rozdělen do čtyř hlavních částí. První část se zabývá oborem robotiky, kde je popsáno co vlastně slovo robotika a robot znamená a následně jsou zde uvedeny jednotlivé metody programování robotů, které se v dnešní praxi používají. Dále je tato část zaměřena na úvod do přímé a inverzní úlohy kinematiky, která je nutná pro určení koncového pracovního bodu robota.

Druhá část se zaměřuje na konstrukci modelu, pracovní činnost modelu, funkci servomotoru a praktické využití procesu třídění barvy v praxi.

Ve třetí části práce je popsána řídicí jednotka a základní části a funkce mikropočítače obecně, aby bylo možné pochopit funkci mikropočítače.

Ve čtvrté části je kladen důraz na řízení celého automatizovaného systému a výpočet inverzní úlohy kinematiky pro daný model robotické ruky. Také jsou zde uvedeny problémy, které nastaly při návrhu algoritmů pro řídicí jednotku.

1 Robotika

Robot je automatický stroj, vykonávající pohybové a manipulační činnosti, často podobné činnostem, které vykonává člověk tak, aby mohl být člověk nahrazen v průmyslu těmito stroji. Stroj musí znát své pohybové prostředí, které svou činností nějakým způsobem ovlivňuje (přemístění předmětů, svařování, montáž, lakování, apod.). Robot se neskládá pouze z manipulátoru jako takového, ale ke své činnosti potřebuje hlavně programovatelný řídicí systém, který řídí jeho pohyb.

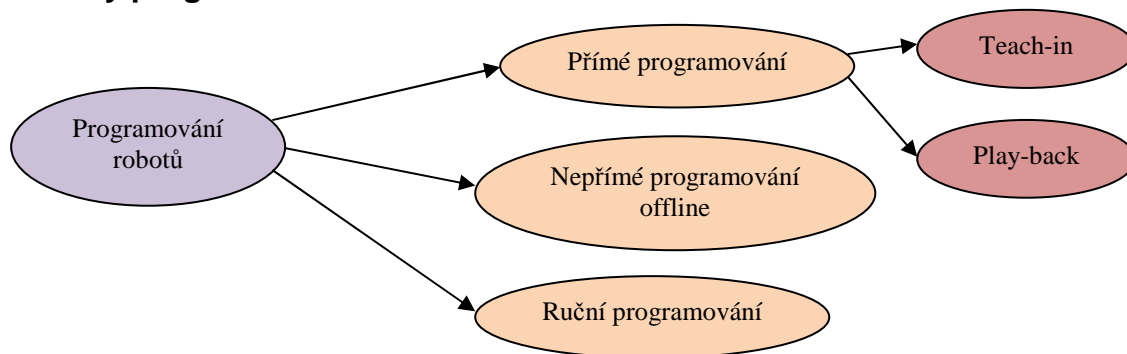
Robotika je odvětví, zabývající se návrhem, aplikacemi a konstrukcí automatických strojů (robotů). Je to obor, který je úzce svázán s mechanikou, elektronikou a teorií řízení, ovšem zabývá se i celou řadou dalších vědních oborů. Byly definovány tři zákony robotiky, na které je při návrhu robotů kladen výrazný důraz a to:

1. Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby bylo člověku ublíženo.
2. Robot musí poslechnout člověka, kromě případů, kdy je to v rozporu s prvním zákonem.
3. Robot se musí chránit před poškozením, kromě případů, kdy je to v rozporu s prvním, nebo druhým zákonem. [10]

1.1 Stupně volnosti

Model robota, použitého v bakalářské práci má šest stupňů volnosti a simuluje pohyby tzv. univerzálních průmyslových robotů, které mají stupňů stejný počet. Šest stupňů volnosti znamená, že robot může koncový pracovní bod nastavit do jakéhokoli bodu v prostoru, v kterém vystačí geometrické rozměry robota (tento bod je popsán šesti údaji). Jsou to tři hodnoty souřadnic referenčního bodu v základním kartézském systému x,y,z (pro pozici) a tři úhly natočení referenčního systému spojeného s tělesem, vzhledem k základnímu souřadnému systému (pro orientaci). Aby mohl tedy robot správně uchopit a manipulovat s předmětem, je třeba minimálně šest stupňů volnosti.

1.2 Metody programování robotů



Přímé programování – Teach-in

Pomocí této metody programování můžeme docílit velké přesnosti pohybu robota. Učení dráhy robota se provádí obsluhou pomocí tlačítek, kdy obsluha najede robotem do určité pozice a stiskne tlačítko „zapamatovat pozici“. Tímto způsobem se robot naučí všechny body, do kterých má najet. Není ovšem definována dráha, po které se má robot z jednoho bodu do druhého posouvat (jak jsou body propojeny v prostoru) – to se řeší dodatečným programováním, kdy se jednotlivé body propojují různými tvary křivek.

Přímé programování – Play-back

Toto programování je velmi efektivní u robotů, které mají kopírovat složité tvary, nebo se dostat na špatně dostupná místa (touto metodou se programují například roboty pro lakování, či svařování). Metoda spočívá v tom, že obsluha vede chapadlo robota ručně po určité dráze a danou rychlostí. Při pohybu si řídicí systém zapamatovává vykonávaný pohyb. Informace o pohybu si ukládá v daných intervalech ve formě tabulky do paměti. Takto získaná data o pohybu poté robot provádí nezávisle na člověku. Nevýhodou může být nedokonalost pohybu člověka – tyto nedokonalosti se při této metodě programování přenáší totiž i na robota.

Nepřímé programování offline

Tento způsob programování je vhodný například pro pohyb svařovacího robota, kdy je popisována trajektorie pohybu v prostoru pomocí daných křivek tak, aby křivky kopírovali tvar objektu určeného pro svařování. Inverzní kinematika je zde řešena offline.

Ruční programování

Zde je pohyb robota dán, na základě ručně zadaných souřadnic koncového pracovního bodu a následujících pohybů. Najetí do koncového bodu je řešeno pomocí inverzní úlohy kinematiky a pro její výpočet slouží souřadnice, které jsou pevně zadány v programu.

1.3 Kinematika

Tělesa, z nichž je robot tvořen a kterými je celý robot spojen tak, že se mohou vzájemně pohybovat, se nazývají kinematické členy. Dva tyto členy, které jsou spolu spojeny, se nazývají kinematická dvojice a ty jsou spojeny klouby. Čím více kloubů robot má, tím jsou lepší jeho manipulační schopnosti. Poloha dvou členů spojených kloubem je určena počtem stupňů volnosti kinematické dvojice. Průmyslový roboti jsou sestaveny většinou z členů s rotačními, nebo translačními klouby (robot použitý v bakalářské práci má pouze rotační klouby). Členy, které jsou spojeny rotačním kloubem, se tak mohou vůči sobě otáčet pouze kolem své osy rotace a u členů s translačním kloubem se mohou vůči sobě pouze posouvat v ose.

Součástí analýzy robotů je úplný kinematický model systému, který popisuje všechny kinematické veličiny pro potřeby řízení (syntéza regulátorů polohy a rychlosti). U kinematické úlohy nás zajímá hlavně průběh polohy a orientace koncového pracovního bodu robota (efektoru) v čase a tomu odpovídající průběh kinematických členů. Vzájemná poloha kinematických členů je popsána zobecněnými souřadnicemi, které udávají vzájemné natočení u systémů s rotačním kloubem. Při řešení kinematické úlohy jsou využívány dvě roviny, do kterých se promítá robot tak, aby bylo možné popsat všechny zobecněné souřadnice. První rovinou je pohled na robota z boku, druhou potom pohled shora. [13]

1.3.1 Přímá úloha kinematiky

Přímá úloha kinematiky je taková úloha, kdy jsou známy všechny zobecněné souřadnice a je hledána poloha a orientace koncového pracovního bodu. Řešení přímé úlohy je poměrně jednoduché a lze řešit buď pomocí goniometrických funkcí, které popisují vztahy mezi jednotlivými členy systému, nebo pomocí lokálních souřadných systémů kinematických členů a transformačními maticemi pro přepočty mezi nimi.

1.3.2 Inverzní úloha kinematiky

Inverzní úloha kinematiky je taková úloha, kdy známe polohu a orientaci koncového pracovního bodu a je potřeba určit zobecněné souřadnice. Řešení inverzní úlohy je složité hlavně při použití na roboty s více stupni volnosti. Je zde třeba počítat s tím, že robot se může do daného bodu v souřadném systému natočit několika způsoby a ne každý způsob najetí do koncové polohy je pro nás vyhovující. V bakalářské práci je využívána inverzní úloha kinematiky, řešená pomocí goniometrických funkcí.

2 Model automatizovaného systému



Obr 2.1 Sestavený model

Model robotické ruky použitý v bakalářské práci vyrábí firma Lynxmotion s typovým označením AL5B. Disponuje šesti stupni volnosti a může být řízen pomocí přímo dodávané řídicí jednotky SSC-32, nebo jednotkou externí, což bylo zvoleno v mém případě.

2.1 Konstrukce

Konstrukce robotické ruky

Robotická ruka je dodávána jako stavebnice, takže veškeré komponenty jsou hotové, vyrobené z plastu, či hliníkových profilů. Robot byl tedy smontován podle návodu a doplněn o pohonné jednotky (servomotory) pro manipulaci. Při montáži je třeba dbát na správné ustavení serva a na natočení serva do správné polohy, jelikož při špatné montáži vznikají nepřesnosti, které se poté projevují při finálním řízení robota.

Konstrukce podstavce a zásobníku

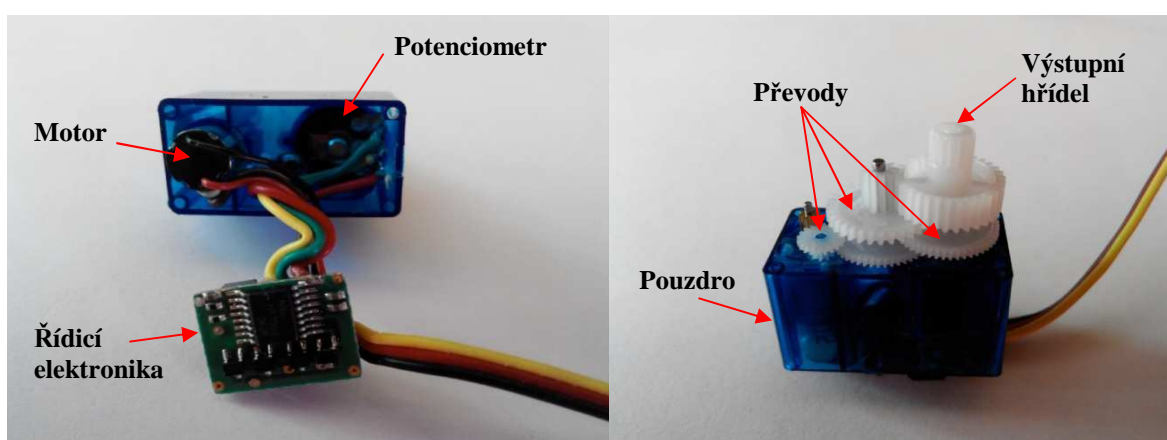
Podstavec je vyfrézován na CNC fríze ze silonu tak, aby se všechny komponenty daly pohodlně rozmístit a robotická ruka mohla dobře manipulovat s předměty. Zásobník je vyroben z hliníkového profilu a plexiskla, přičemž senzor barvy je uložen v silonovém nosníku, který slouží také jako podpěra pro předměty, které bude následně robot třídit.

Vývody ze senzoru barvy a servomotorů jsou vyvedeny na řídicí jednotku a rozvodnou desku napájení pro serva. Vše je propojeno dráty o průřezu 0,34 mm².

Pohonné jednotky modelu

K robotické ruce od firmy Lynxmotion jsou předepsána serva od společnosti Hitec s různými mechanickými a elektrickými vlastnostmi. Jedná se ale o serva s vyšší mechanickou pevností, tzn., že převody jsou na servech pracujících ve velké zátěži kovové.

2.2 Popis servomechanismu



Obr 2.2 Popis servomechanismu

Výstupní hřídel

Výstupní hřídel přenáší pohyb mechanismu na ovládací páku. Na kvalitě výstupní hřídele záleží hlavně z důvodu spolehlivosti a při vyšší zátěži serva. U některých servomechanismů může být výstupní hřídel doplněna o kuličková ložiska, což zpřesní chod serva a zvýší životnost.

Převody

Převodový poměr určuje poměr mezi rychlostí a tahem servomechanismu. Z hlediska konstrukce je to pak stejné jako u výstupních hřídelí – převody mohou být plastové, či kovové pro vyšší životnost.

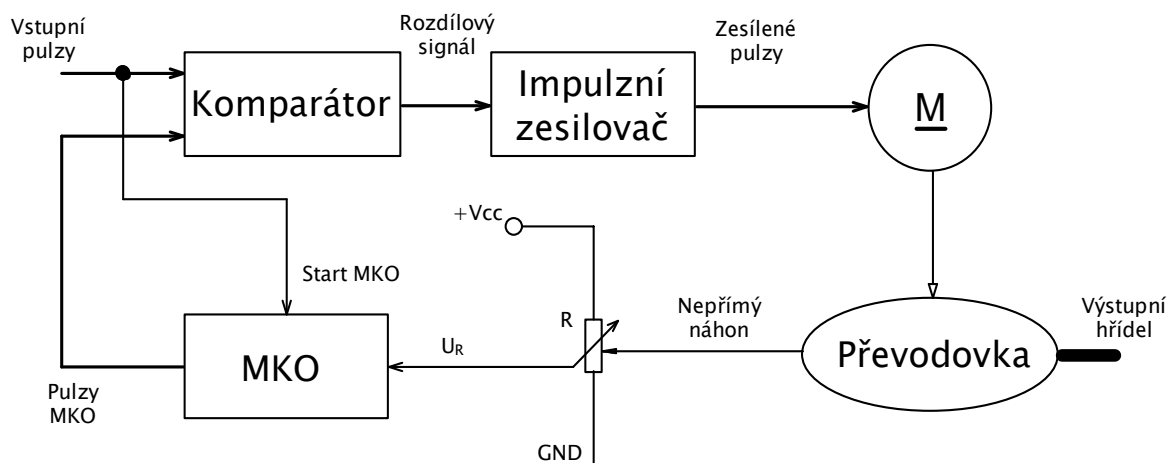
Potenciometr

Přes potenciometr se snímá poloha výstupní hřídele a uzavírá se zpětná vazba, bez níž by servomechanismus nepracoval správně. Konstrukce je u levnějších serv taková, že

potenciometr je připojen přímo na výstupní hřídel, kdežto u lepších serv je připojen potenciometr přes tzv. nepřímý náhon, čímž se velmi účinně odruší vibrace, které by mohli poškodit odporovou dráhu potenciometru. Poškození odporové dráhy má za následek špatné vyhodnocování polohy a projevuje se tak, že servo v určitých polohách přeskakuje.

Řídicí elektronika

Do serva vede třížilový kabel, kde červená žíla je +Vcc, černá žíla zem GND a žlutá je řídicí signál generovaný pulzně šířkovou modulací PWM. Větší serva mají třížilový kabel kroucený kvůli potlačení indukce rušivých napětí při průchodu vyšších proudů. Řídicí elektronika se skládá z monostabilního klopného obvodu, porovnávacího obvodu (komparátoru) a impulzního zesilovače. Jak je vidět na blokovém schématu servomechanizmu na Obr. 2.3, tak vstupní pulzy jsou přiváděny do komparátoru a zároveň jejich náběžná hrana spouští monostabilní klopný obvod MKO. Délku pulzu monostabilního klopného obvodu určuje úbytek napětí na potenciometru snímajícího polohu výstupní hřídele. Komparátor porovnává šířku vstupního (řídicího) pulzu a šířku výstupního pulzu MKO. Výstupem komparátoru je tedy rozdíl z těchto dvou pulzů, který je následně zesílen impulzním zesilovačem a poslán do elektromotoru. Šířka rozdílového pulzu nutí motor k natočení na určitou stranu.



Obr 2.3 Blokové schéma serva [16]

2.3 Pracovní činnost modelu

Model komunikuje s okolním prostředím pomocí LCD displeje a čtyř tlačítek, kterými je možno robotickou ruku ovládat.

Popis tlačítek:

S1 – START

S2 – STOP

S3 – Referenční poloha

S4 – Menu

Pracovní činnost je následující: po zapnutí napájení mikrokontroléru se na LCD displeji zobrazí nápis „Zahájení třídění stiskněte S1“. Pokud se stiskne tlačítko S1, vypíše se „Zařízení v provozu! Skenuji“ a pokud je zásobník naplněn libovolným počtem kostek, senzor barvy naskenuje danou barvu kostky, která má být právě odebrána, vypíše rozpoznanou barvu na displej, jako např. „ŽLUTÁ Průběh třídění“ a robot najede do zásobníku, kde kostku uchopí. Když robot vytáhne kostku ze zásobníku, je roztříděna do příslušného kontejneru, který odpovídá její barvě. Po roztřídění robot najede zpět před zásobník a začne skenovat další barvu s nápisem na displeji „Skenuji“ a pokud barvu rozpozná, znovu ji roztřídí do příslušného kontejneru. Jestliže ale po pátém skenování senzor nic nerozpozná, vrátí se robot do referenční polohy, je vypsáno na displej „Zásobník prázdný“ a bzučák vydá přerušovaný signál s intervalem zapnutí/vypnutí 400 ms. Po doznění tónu bzučáku se znovu vypíše na displej „Zahájení třídění stiskněte S1“. Pokud se tak učiní a zásobník bude stále prázdný, senzor naskenuje pětkrát a znovu zahlásí, že je zásobník prázdný. Robot je přitom pořád v referenční poloze.

Tlačítko S2 funguje tak, že stlačením kdykoli při běhu programu se robot zastaví na pozici, kde zrovna byl a na displej se vypíše „STOP“, následně se může pokračovat dvěma způsoby. Prvním způsobem je pokračování v běhu programu pomocí tlačítka S1 – pokud je tedy naskenována určitá barva a bylo zmáčknuto tlačítko S2, robot dotřídí barvu do daného kontejneru a program pokračuje tak, jak bylo popsáno výše. Druhým způsobem je návrat robota do referenční polohy tlačítkem S3. Pokud bylo zmáčknuto tlačítko S2 a následně S3, na displej se vypíše „Referenční poloha“, robot rozpozná, zda-li má v chapadle kostku a pokud ano, položí kostku na zem před zásobník a vrátí se do referenční polohy. Jestliže kostka ale v chapadle není, robot najede do referenční polohy okamžitě. Při návratu do základní polohy je znovu vypsáno na displej „Zahájení třídění stiskněte S1“ a program může pokračovat od začátku. Při stlačení tlačítka S3 bez předchozího zastavení programu (tlačítkem S2) se nic neděje a robot pokračuje v dosavadní činnosti.

Tlačítkem S4 je možno zobrazit menu, kde je vypsán počet jednotlivých rozříděných barev, jako např. „Rozříděno černá: 3“. V menu se listuje tlačítkem S4.

2.4 Praktické využití

Praktické využití rozpoznávání předmětů podle barev je obrovské a v systémech s vysokým stupněm automatizace je jedním z hlavních kritérií pro rozpoznávání. Lze rozpoznávat barvu celého produktu, jeho obalu, etikety, či potisku, ale nejen to. Rozpoznávání barev se používá například i pro vyhodnocování kvality potravin, jako je ovoce, kde se kamerou snímá daný produkt (např. jablko) a podle jeho barvy je možno detekovat zralost ovoce a rozeznat povrchovou hnilobu. Při použití infračervené kamery, je možno ale detekovat i neobvyklý obsah cukru a kyselých látek a případně i nahnívání jablka pod povrchem a různé choroby.

Dále je možno pomocí kamer detekovat nežádoucí předměty v potravinách, jako jsou luštěniny, káva a podobně, díky různé odrazivosti a vlhkosti.

Rozpoznávání barev na principu jednoduššího bezkamerového snímače je využíváno například pro třídění výrobků na běžícím pásu, které mají sice různé barvy, ale stejného odstínu. Tyto snímače mají tedy v paměti uložené hodnoty pro jednotlivé barvy a podle těchto hodnot rozhodují o barvě, která je právě snímána. Na rozdíl od kamerových snímačů barvy, kde je možno nasnímat obraz daného předmětu a určit barvu podle dostatečného množství RGB pixelů v nasnímaném obrazu s pixely uloženými v referenční paměti. Díky tomu může snímač snímat barvy s malým rozdílem odstínů.

Snímač navržený v bakalářské práci by mohl být využíván v praxi pro detekci barev boxů, či krabic, které budou následně rozříděny do příslušných kontejnerů.

3 Řídicí jednotka

K řízení celého automatizovaného systému je použita řídicí jednotka v podobě vývojového kitu EvB 5.1 od firmy AND-TECH, která je osazena výkonným mikropočítačem ATmega644P od společnosti Atmel. V následujících podkapitolách je popsáno, z čeho se mikropočítač skládá a jak funguje a následně je popsán samotný vývojový kit.

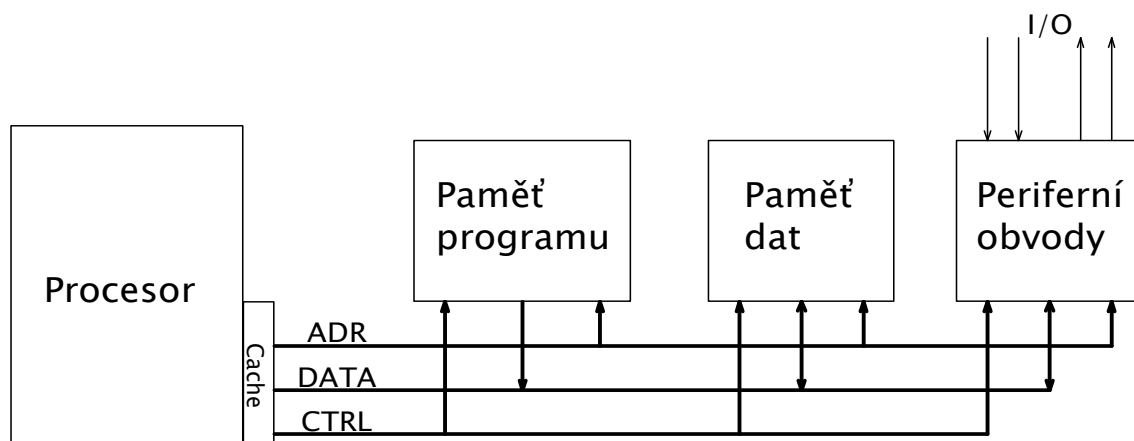
3.1 Mikropočítač

Pod názvem mikropočítač rozumíme jednočipový počítač, který není tvořen pouze mikroprocesorem, jak je běžně špatně označován mikropočítač. Mikroprocesor by bez

ostatních přídavných obvodů dokázal provádět pouze velmi málo operací. Jako mikropočítač je tedy nazýváno seskupení obvodů, které společně dokáží provádět v mnoha případech velmi složité operace. Mikropočítač dnes už najdeme téměř všude – většinou se používá pro jednorúčelové operace, jako je například řízení, regulace (pulzně šířková modulace, ovládání motorů), dálkové ovládání, apod.

3.1.1 Základní části mikropočítače

Základními částmi mikropočítače je mikroprocesor, paměť programu, paměť dat a periferní obvody. Všechny tyto bloky jsou propojeny třemi sběrnicemi pro komunikaci mezi sebou. U moderních mikropočítačů najdeme adresovou, datovou a kontrolní (řídící) sběrnici. Po adresové sběrnici se posílají všechny bity, které tvoří adresu (pro adresování paměti), po datové sběrnici bity, které reprezentují hodnotu vybavené paměťové buňky a po kontrolní (řídící) sběrnici CTRL jdou signály pro řízení ostatních obvodů, které jsou připojeny k mikroprocesoru a paměti. Kontrolní sběrnice navíc kontroluje, zda se signál dostal do cílového místa a zda signál daný obvod správně rozpoznal. Naznačení základního zapojení mikropočítače je na *Obr. 3.1*.



Obr 3.1 Základní části mikropočítače [1]

3.1.1.1 Adresové prostory

Adresové prostory jsou adresy, na které můžeme dosáhnout buď zápisem do paměti, či čtením z paměti. Díky tomu může existovat programový prostor, datový prostor, periferní prostor atd.

von Neumanova architektura

Tato architektura se dnes už příliš nepoužívá, ale stále se s ní můžeme setkat. Její nevýhodou je skutečnost, že nelze číst instrukce a zároveň zapisovat data. Spočívá v tom, že má mikropočítač pouze jeden adresový prostor, kde se nacházejí ostatní prostory, tedy

programový, datový a periferní. Výběr jednoho ze základních obvodů zajišťuje adresový dekodér, který se řídí nejvyššími bity adresy a podle nich se tedy i vybírá příslušná paměť, či periferní obvod. U této architektury se nachází dva řídicí signály a to čtení a zápis.

Harvardská architektura

Tato architektura se liší od von Neumanovy architektury tím, že mikro počítač obsahuje více adresových prostorů a má dvě soustavy sběrnic. V adresových prostorech se mohou nalézat stejné adresy, tzn., že k jejich rozlišení zde neslouží adresa, ale zvláštní řídicí signály, které se u předchozí architektury nenalézají – čtení instrukce, čtení dat, čtení z periferních obvodů, zápis dat a zápis do periferních obvodů. Díky tomu, že má tato architektura sběrnici pro čtení programu a sběrnici pro zápis, lze provádět dvě věci najednou. Výhodou je tedy rychlost.

3.1.1.2 Periferní obvody

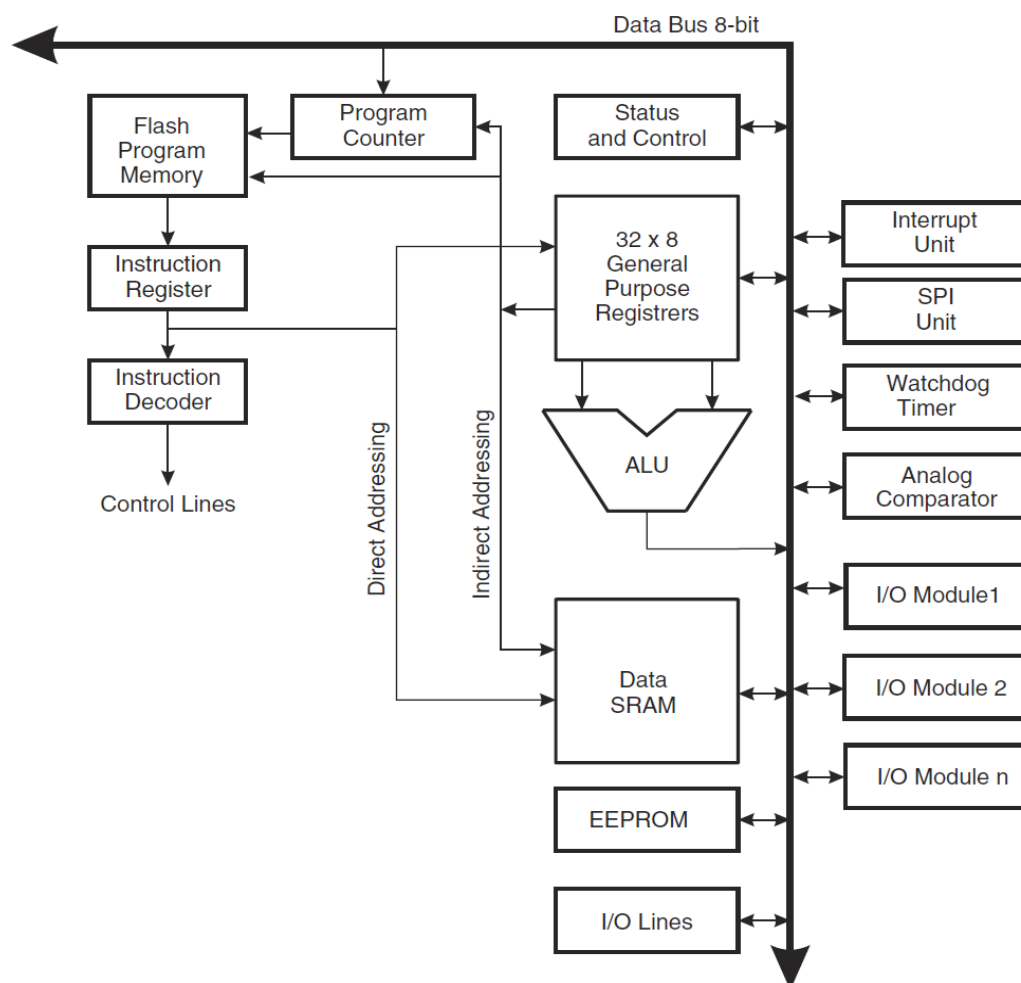
Periferní obvody umožňují jednočipovému mikro počítači styk s vnějším okolím. Mají poměrně složitou strukturu, jelikož v jedné periferní jednotce se nachází několik důležitých bloků pro správnou funkci obvodů. Nejdůležitějším blokem je blok zpracování signálů, který tvoří například omezovače napětí, oddělovací členy, ale také čítače nebo převodníky. Protože probíhá zpracování více signálů najednou (např. provádíme A/D převod, nebo čítáme impulzy pomocí čítače) potřebujeme tato data ukládat, k čemuž slouží datový registr. Abychom mohli nastavovat předděličky čítačů, musí blok pro zpracování signálů řídit řídicí registr, který komunikuje přes datovou sběrnici s mikroprocesorem. Posledním důležitým obvodem jsou stavové registry, díky nimž může procesor číst stav obvodů pro zpracování signálů (tzn. externí přerušení, přetečení čítače, porovnání hodnot v komparátoru, dokončení A/D převodu, příjem znaku při sériovém přenosu, atp.) a následně podle této události vytvořit přerušení, v kterém se definuje nějaká akce (rozsvícení LED, potvrzení zmáčknutím tlačítka, apod.).

3.1.1.3 Procesor

Procesor je nejdůležitější částí mikro počítače, jeho úkolem je zpracování instrukcí. Instrukce je nejmenší jednotka programu, kterou zadává programátor (je to například instrukce násobení, sčítání, apod.) při instrukci se tedy např. sečtou obsahy dvou registrů a následně se výsledek uloží do jiného registru. Spojením několika instrukcí vznikne program.

Procesor si přečte instrukci, následně ji provede a uloží výsledek → tomuto cyklu se říká *základní rytmus*. Instrukci procesor přečte tak, že zná adresu (díky programovému čítači) a

vyvolává čtecí cykly z programové paměti. Instrukci provede tak, že zjistí, co instrukce vlastně znamená – *dekódování*, postupně zpracuje data a uloží výsledek. K uložení výsledku musí procesor znát adresu pro hlavní paměť RAM, nebo adresu pro uložení do krátkodobé paměti – registrů GPR. RAM paměť je mimo procesor (rychlost procesoru je mnohem větší, než rychlost paměti) a musí spolupracovat s vyrovnávací pamětí cache, která se zařazuje mezi procesor a hlavní paměť. Cache paměť je konstruována tak, že stačí rychlosti procesoru (nesmí zdržovat), ale čím rychlejší vyrovnávací paměť je potřeba, tím menší kapacitu má.



Obr 3.2 Blokové schéma procesoru AVR ATmega644P [2]

Programový čítač (program counter PC)

Po uložení výsledku je třeba generovat adresu další instrukce pro programovou paměť, k tomu slouží tzv. programový čítač. Ten není přístupný přímo, ale jeho obsah se mění automaticky při provádění instrukcí.

Instrukční dekodér (instruction decoder ID)

Instrukční dekodér má za funkci správně dekodovat (rozlišit o jaký typ instrukce se jedná) a podle toho přiřadí počáteční stav řadiči.

Pracovní, datové registry (general purpose register GPR)

Slouží pro krátkodobé ukládání dat, která budou zakrátko využita. Výhodou těchto registrů je to, že jsou umístěny uvnitř procesoru, tzn., že je k nim mnohem rychlejší přístup, než k datové paměti. Registry slouží pro zápis i čtení. U moderních procesorů se využívají GPR, které mají dvojí čtení dat a díky tomu můžeme využívat tří-operandové instrukce (tří-operandová instrukce je např. součet obsahu registru 1 s obsahem registru 2 a zápis výsledku do registru 3).

Aritmeticko-logická jednotka (arithmetic logic unit ALU)

Je jedna ze základních komponent procesoru, jelikož provádí základní aritmetické operace, jako je součet, rozdíl, násobení, dělení, změna znaménka, bitové posuny a v neposlední řadě nastavení bitů příznakového registru. Operandy pro ALU jsou dodávány z datových (pracovních) registrů GPR, nebo z vnější paměti. Výsledky jsou ukládány znovu do GPR, či do paměti. O tom, jaký typ operace má ALU provádět rozhoduje řadič, který generuje kód, podle něhož se vybere příslušná kombinace obvodů, která následně provede požadovanou operaci.

Řadič (controller)

Řadič je realizován vždy jako pevně zapojená logika, či jako mikroprogramový automat. U mikroprogramového automatu je paměť mikroprogramů typu ROM z toho důvodu, aby uložené mikroprogramy nemohli být neúmyslně smazány. V této řídicí paměti jsou uloženy mikroprogramy, které provádí různé typy instrukcí.

3.1.2 Přerušení

Přerušení je vyvoláváno diagnostickými, nebo periferními obvody a díky němu může mikropočítač efektivně obsloužit nepravidelné události, na které má reagovat. Pokud se tedy vyskytne tato definovaná událost, mikropočítač přerušuje běh programu a provádí programovou obsluhu události, která přerušení vyvolala. Po dokončení obslužného podprogramu se počítač opět vrací na místo, kde byla vyvolána událost a pokračuje v běhu hlavního programu.

Událostí, která může přerušit vyvolat je například reset mikropočítače, dokončení A/D převodu, externí přerušit, dokončení sériového přenosu, přetečení čítače, porovnání analogovým komparátorem, atd.. Ve skutečnosti je ve stejném čase generováno více požadavků na přerušit a musí se tedy rozhodnout o tom, jaký požadavek se vykoná dříve – to provádí řadič přerušit. Řadič přerušit přiřadí prioritu jednotlivým aktivním požadavkům a na základě priority se následně provádí zpracování žádostí v jednom strojovém cyklu. Nejvyšší prioritu přerušit má vždy reset a následují externí přerušit.

Vector No.	Program Address	Source	Interrupt Definition
1	\$0000	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	\$0002	INT0	External Interrupt Request 0
3	\$0004	INT1	External Interrupt Request 1
4	\$0006	INT2	External Interrupt Request 2
5	\$0008	PCINT0	Pin Change Interrupt Request 0
6	\$000A	PCINT1	Pin Change Interrupt Request 1
7	\$000C	PCINT2	Pin Change Interrupt Request 2
8	\$000E	PCINT3	Pin Change Interrupt Request 3
9	\$0010	WDT	Watchdog Time-out Interrupt
10	\$0012	TIMER2_COMPA	Timer/Counter2 Compare Match A
11	\$0014	TIMER2_COMPB	Timer/Counter2 Compare Match B
12	\$0016	TIMER2_OVF	Timer/Counter2 Overflow
13	\$0018	TIMER1_CAPT	Timer/Counter1 Capture Event
14	\$001A	TIMER1_COMPA	Timer/Counter1 Compare Match A
15	\$001C	TIMER1_COMPB	Timer/Counter1 Compare Match B
16	\$001E	TIMER1_OVF	Timer/Counter1 Overflow
17	\$0020	TIMER0_COMPA	Timer/Counter0 Compare Match A
18	\$0022	TIMER0_COMPB	Timer/Counter0 Compare Match B
19	\$0024	TIMER0_OVF	Timer/Counter0 Overflow
20	\$0026	SPI_STC	SPI Serial Transfer Complete
21	\$0028	USART0_RX	USART0 Rx Complete
22	\$002A	USART0_UDRE	USART0 Data Register Empty
23	\$002C	USART0_TX	USART0 Tx Complete
24	\$002E	ANALOG_COMP	Analog Comparator
25	\$0030	ADC	ADC Conversion Complete
26	\$0032	EE_READY	EEPROM Ready
27	\$0034	TWI	2-wire Serial Interface
28	\$0036	SPM_READY	Store Program Memory Ready
29	\$0038	USART1_RX	USART1 Rx Complete
30	\$003A	USART1_UDRE	USART1 Data Register Empty
31	\$003C	USART1_TX	USART1 Tx Complete

Obr 3.3 Tabulka priorit přerušit ATmega644P [2]

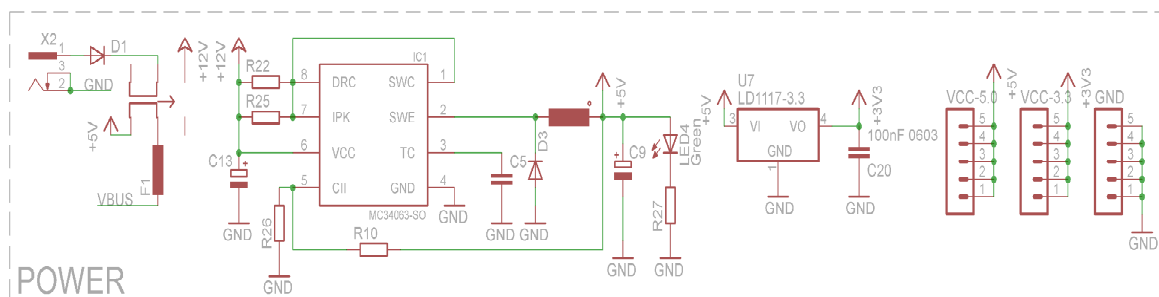
3.2 Vývojový kit EvB 5.1

Vývojový kit EvB 5.1 je univerzální kit, který slouží jako programátor pro mikrokontroléry od firmy Atmel, řady ATmega16,32, či 644. Ovšem neslouží pouze jako obyčejný programátor. Na vývojovém kitu je k dispozici mnoho dalších užitečných periférií, jako je například digitální senzor teploty, LED indikátory, tlačítka, potenciometry, bzučák, rozvod napájení (5 V, 3,3 V, GND), převodník USB-UART, výkonové výstupy, LED/LCD displej, hodiny reálného času, či infračervený přijímač/vysílač. Zde se budu zabývat pouze perifériemi, které jsou využívány v bakalářské práci.

Rozvod napájení

Napájení je dodáváno přes integrovaný napájecí obvod a kit lze napájet dvěma způsoby – externě z AC/DC adaptéru, který má výstupní napětí minimálně 9 V a maximální napětí 24 V, nebo pomocí USB portu počítače. V této práci je využívána jako zdroj napětí 9 V baterie.

Rozvod napájení na desce je 5 V, či 3,3 V. Toto napětí je vyvedeno pomocí hřebínek ven z desky. V práci je využíváno napětí 5 V, kterým je napájena RGB dioda a napěťový dělič tvořící senzor barvy.



Obr 3.4 Zapojení napájecího obvodu [4]

Převodník USB-UART

Vývojový kit obsahuje převodník USB-UART, který vytváří virtuální COM port při komunikaci s počítačem. V práci tedy nebyl využit sériový port jako takový pro nahrávání programu do mikrokontroléru.

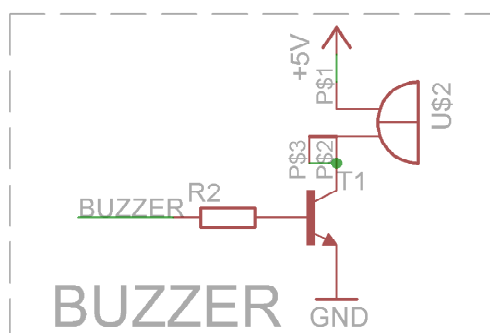
Tlačítka

Pro komunikaci s okolním prostředím nejsou využita tlačítka na vývojovém kitu, ale tlačítka vhodná pro zasazení do řídicího panelu. Tlačítka jsou zapojena se společnou zemí, tzn., že se musí zapnout pull-up rezistor na vstupu mikropočítače. Nejsou nijak ošetřena proti zákmitům z výroby, a proto je třeba ošetřit zákmity dodatečně. Ošetření zákmitů tlačítek je

řešeno hardwarově, využitím 10 nF kondenzátoru, připojeného k tlačítku. Tím jsou částečně filtrovány zákmity projevující se při stisku. Tento způsob ošetření zákmitů byl zvolen z důvodu zásoby součástek a jednoduchosti řešení. Další možností ošetření je použití Schmittova klopného obvodu a RC článku, či softwarového ošetření opakovaným skenováním stavu pinu, na kterém je připojeno tlačítko.

Bzučák

Bzučák je využíván pro oznámení prázdného zásobníku tónem. Bzučák budí NPN řídicí tranzistor, který má bázi připojenou přímo na pin mikrokontroléru.



Obr 3.5 Zapojení bzučáku [4]

LCD displej

Pro komunikaci s okolním prostředím je využit 32 znakový podsvětlený LCD displej, řízený standardním řadičem HITACHI HD44780. Pro řízení chodu displeje je využita knihovna, určená přímo pro tyto typy displejů.

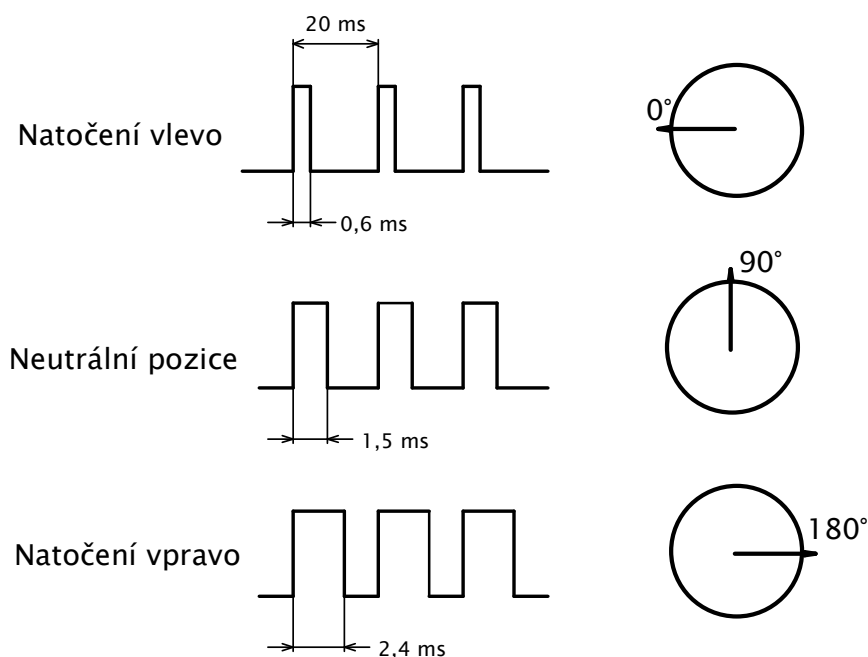
4 Řízení

Řízení robotické ruky se provádí na základě vypočtené inverzní úlohy kinematiky přímo pro tento model. Tyto výpočty jsou zalgoritmovány a nahrány v mikropočítači, který výsledky inverzní kinematiky převádí do podoby vhodné pro servomotory. K řízení senzoru barvy je využíván jeden kanál A/D převodníku a tři výstupy pro spínání LED – funkce je popsána níže v kapitole 4.3 Senzor barvy.

4.1 Řízení serva

Poloha výstupní hřídele serva je řízena pulzně šířkovou modulací PWM a to tak, že každý pulz musí být vyslán na vstup komparátoru s frekvencí 50 Hz (tedy každých 20 ms). Šířka pulzu je u většiny serv pevně dána, a to 0,6 ms až 2,4 ms (orientační natočení serva

demonstruje *Obr. 4.1*). Ovšem tyto hodnoty jsou pouze orientační a pro přesný chod serva (který u robotické ruky potřebujeme) je třeba šířku pulzů změřit pro různá natočení. V tomto případě byl zvolen krok po pěti stupních. Každé servo v originálním stavu se otáčí od 0° do 180° , takže na každé servo vychází 36 různých natočení. Tyto hodnoty se měří při natáčení serva zleva doprava, zprava doleva a následně se hodnoty průměrují. Zprůměrované hodnoty se prokládají navíc lineárním proložením a takováto výsledná přesnost je pro naše účely dostačující.



Obr 4.1 Natočení serva [3]

Generování PWM signálu na více než dva piny mikropočítače se musí řešit softwarově pomocí jednoho hardwarového čítače/časovače, jelikož ATmega644P má pro PWM signál předurčeny pouze dva piny. Používán je režim 14 Fast PWM („rychlá PWM“) s maximální hodnotou uloženou v registru ICR1 (více viz. datasheet Atmel ATmega644P [2]). Funkci generování PWM signálu ilustruje *Obr. 4.2* a vývojový diagram na str. 27.

Na robotické ruce je šest serv a na každé musí přicházet pulz s frekvencí 50 Hz. U čítače/časovače je tedy zvolena frekvence 300 Hz, což znamená, že čítač bude přetékat každých 3,33 ms. Je zde využíváno přerušování při přetečení a přerušování při shodě porovnávacího registru OCR1A s registrem celkové hodnoty čítače TCNT1. Při volbě inkrementace čítače nebyla zvolena žádná předdělička z toho důvodu, aby bylo zajištěno co největší rozlišení pulzně šířkové modulace. Rozlišení při řízení tímto způsobem dosahuje 15,7 bitů, přičemž maximální rozlišení, kterého lze dosáhnout je 16 bitů. Rozlišení lze vypočítat dle následujícího vztahu:

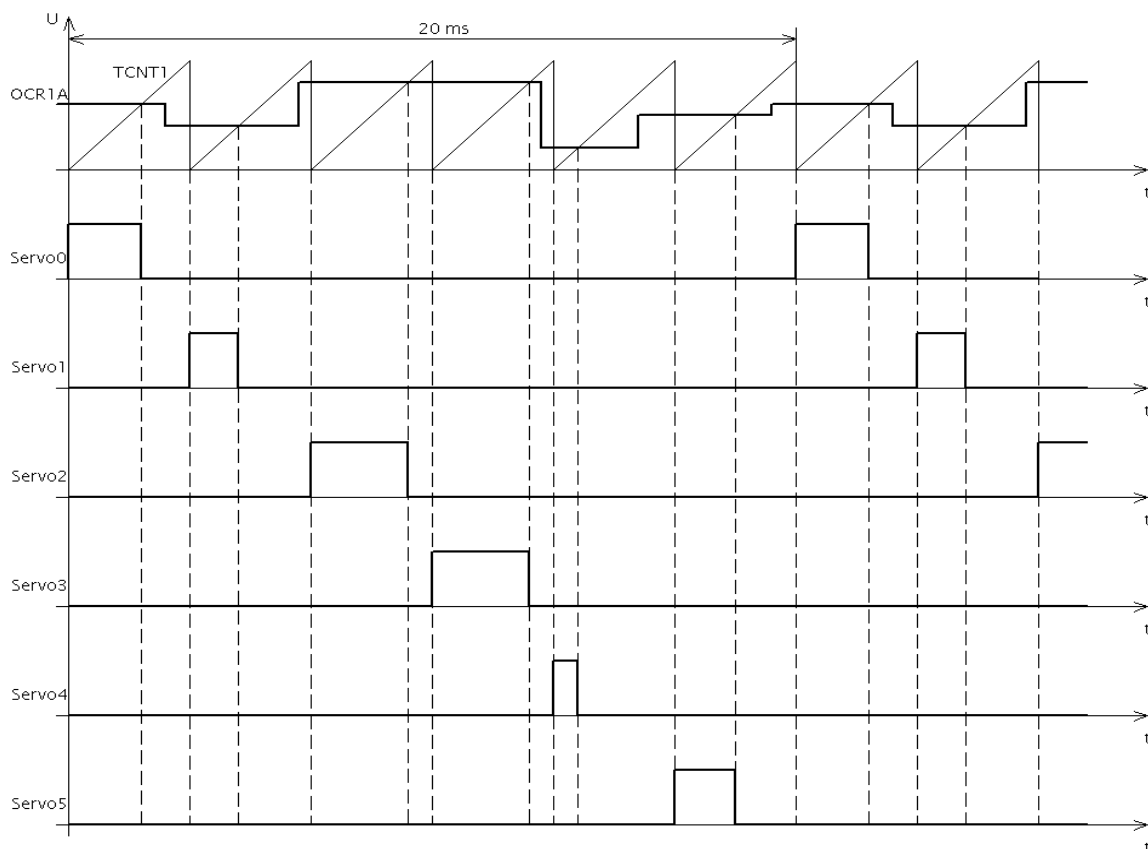
$$R_{FPWM} = \frac{\log(TOP + 1)}{\log 2} = \frac{\log(53332 + 1)}{\log 2} = 15,7 \text{ bitů} \quad 1$$

,kde TOP je maximální hodnota čítače, která je zvolena tak, aby výstupní frekvence PWM odpovídala 300 Hz. TOP hodnotu lze vypočítat podle vztahu:

$$TOP = \frac{f_{CLK}}{f_{OCR1A} \cdot N} - 1 = \frac{16 \cdot 10^6}{300 \cdot 1} - 1 = 53332,33 \cong 53332 \quad 2$$

,kde f_{CLK} je frekvence hodin procesoru (16 MHz), f_{OCR1A} je výstupní frekvence PWM a N je předdělička.

V přerušení při přetečení se začne vysílat kladný pulz na první pin mikropočítače, resp. první servo. Při vyvolání přerušení shody registru OCR1A s TCNT1 se pulz vysílaný na první servo shodí a nastavuje se indexace dalšího pinu. Při dalším přerušení při přetečení se vysílá kladný pulz na druhý pin resp. druhé servo a takto se opakuje obsluha při shodě registrů až do doby, než se vyšle kladný pulz na poslední servo. Poté se celý proces opakuje znovu od prvního pinu.



Obr 4.2 Generování PWM na šest serv

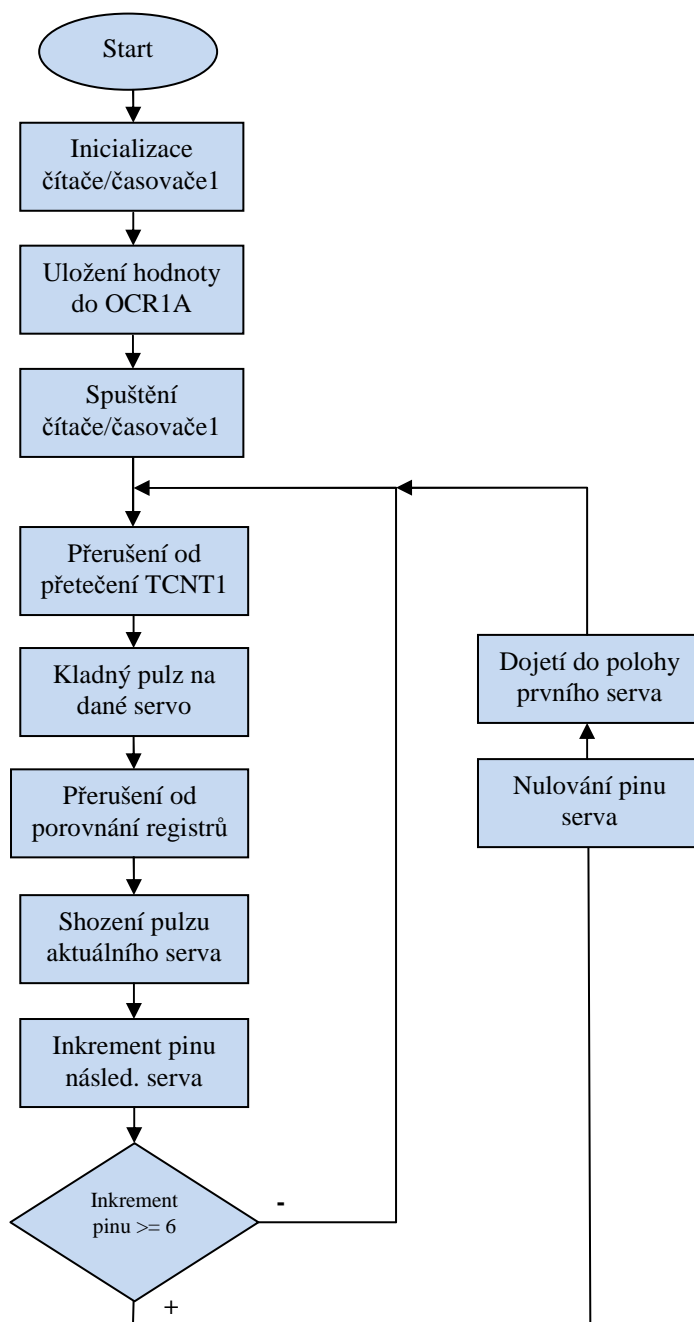
Zpomalení chodu serva

Zpomalení chodu serva lze provést tak, že není na servo poslán okamžitě plný pulz pro natočení, ale je určen inkrement, který se ke stávajícímu pulzu buď přičítá (pokud je předchozí hodnota menší, než následující), anebo odečítá (pokud je předchozí hodnota větší, než následující). Inkrement je počítán z minimální hodnoty serva, která odpovídá natočení serva vlevo, maximální hodnoty, která odpovídá natočení serva vpravo, času jednoho průběhu nekonečné smyčky v hlavním programu a času, který je zadán jako požadovaný čas pro natočení serva. Všechny tyto hodnoty jsou uloženy v jednorozměrných polích. Vzorec pro výpočet inkrementu je:

$$\text{Inkrement} = \frac{\text{max} - \text{min}}{\text{casCyklu} \cdot \text{casNatoceni}} \quad 3$$

Čas natočení serva (je udáván v milisekundách) není přesný, stejně jako není přesně daný čas jednoho průběhu nekonečnou smyčkou (ten je zadán jako 1 ms). V ideálním případě by byl tedy čas průběhu nekonečnou smyčkou 1 ms, ovšem v nekonečné smyčce jsou další příkazy (jako jsou například výpisy na displej, podmínky pro určení barvy, čtení stavů tlačítek, apod.), které průběh smyčkou zpomalují, a ve skutečnosti je tedy tento čas delší. To vede k tomu, že nelze nastavit přesný čas natočení serva.

Vývojový diagram pro generování PWM na šest serv

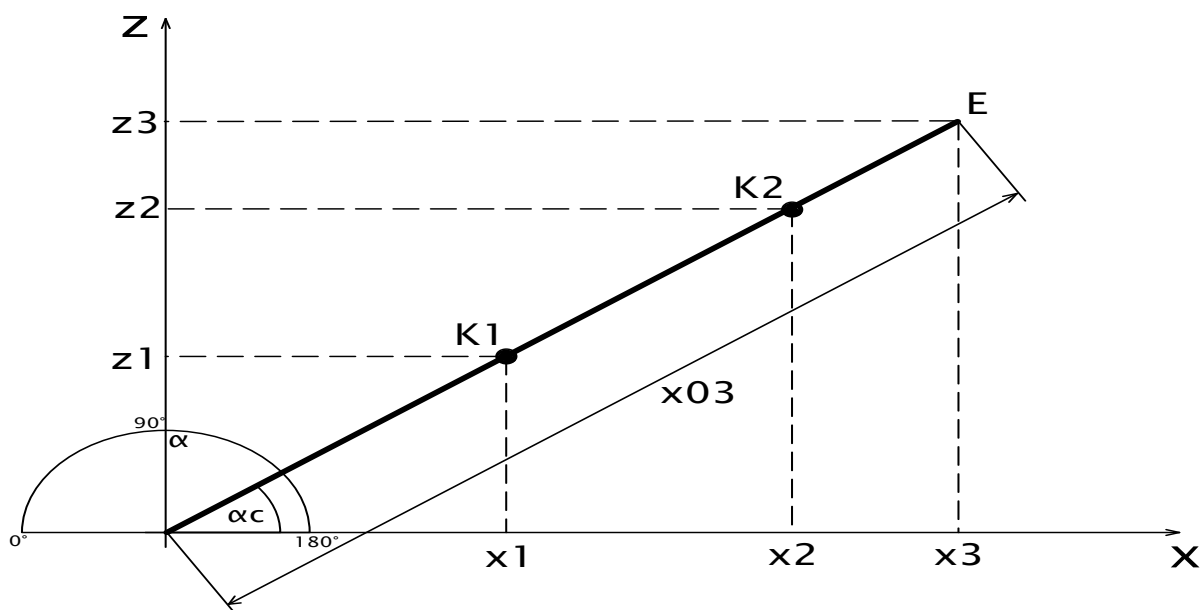


4.2 Výpočet inverzní úlohy kinematiky pro model robotické ruky

Při výpočtu inverzní kinematiky známe polohu koncového pracovního bodu (vrchol efektoru), což jsou ve výpočtu souřadnice x_3 , y_3 , z_3 . Dále jsou dány délky jednotlivých ramen robota a , b , c a úhel σ . Úhel σ je úhel, který je zadáván přímo a určuje natočení chapadla vůči zemi (kolmice k zemi v kloubu K_2 , úhel který vede od kolmice k rameni c je úhel σ).

Z těchto zadaných údajů, je třeba určit úhel natočení α kolem osy y , úhel natočení β ramene a , úhel natočení γ ramene b vůči rameni a a úhel natočení ψ ramene c vůči rameni b .

Výpočet je prováděn pomocí goniometrických funkcí s vhodným použitím sin a cos. Obrázky ilustrující rozložení robotické ruky pro výpočet inverzní kinematiky jsou *Obr. 4.3* a *Obr. 4.4*.



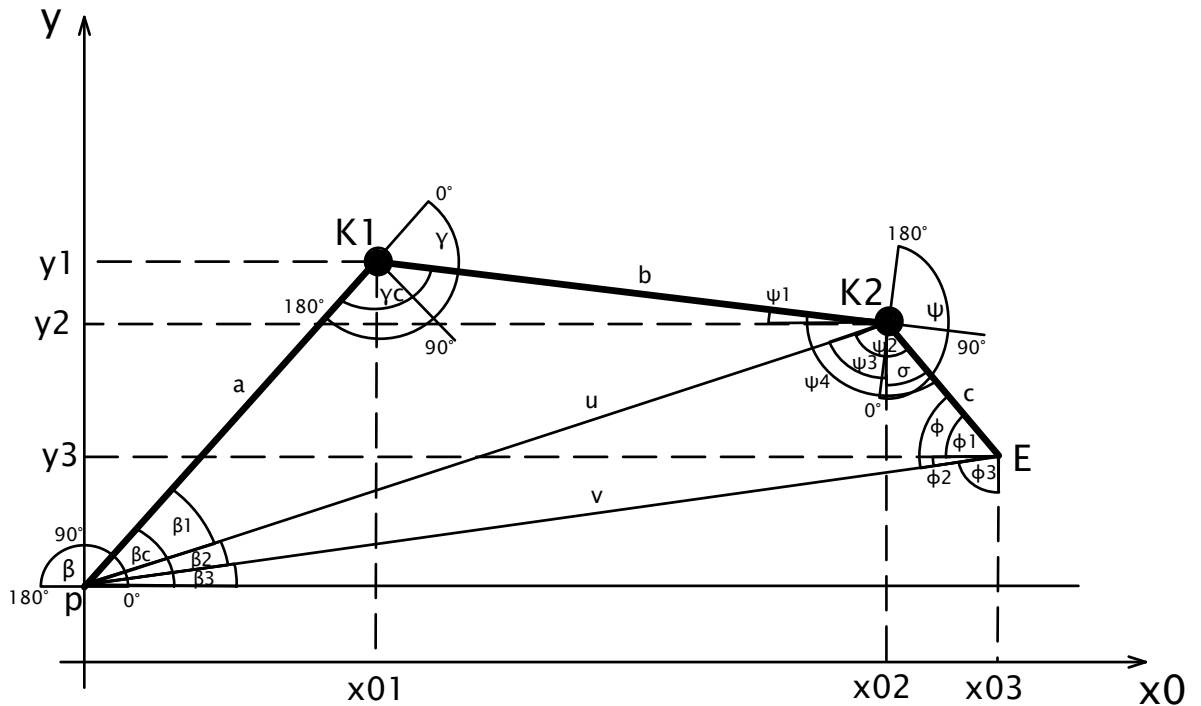
Obr 4.3 Výpočet inverzní kinematiky – pohled shora

Výpočet úhlu natočení α

Celková délka všech ramen $x_{03} = \sqrt{x_3^2 + z_3^2}$ 4

Úhel α_c vůči ose x $\cos \alpha_c = \frac{x_3}{x_{03}} \rightarrow \alpha_c = \arccos\left(\frac{x_3}{x_{03}}\right)$ 5

Úhel α $\alpha = 180 - \alpha_c$ 6



Obr 4.4 Výpočet inverzní kinematiky – pohled z boku

Výpočet úhlu Φ

Pomocná
přímka v

$$v = \sqrt{x_{03}^2 + (y_3 - p)^2} \quad 7$$

Úhel
 Φ_1, Φ_2, Φ_3

$$\Phi_1 = 90 - \sigma, \quad \Phi_3 = \arcsin\left(\frac{x_{03}}{v}\right), \quad \Phi_2 = 90 - \Phi_3 \quad 8$$

Úhel Φ

$$\Phi = \Phi_1 + \Phi_2 = 180 - \sigma - \Phi_3 = 180 - \sigma - \arcsin\left(\frac{x_{03}}{v}\right) \quad 9$$

Výpočet úhlu natočení β

Pomocná
přímka u

$$u^2 = v^2 + c^2 - 2 \cdot v \cdot c \cdot \cos\varphi \quad 10$$

$$u = \sqrt{v^2 + c^2 - 2 \cdot v \cdot c \cdot \cos\varphi}$$

Úhel β_1

$$b^2 = a^2 + u^2 - 2 \cdot a \cdot u \cdot \cos\beta_1 \quad 11$$

$$\beta_1 = \arccos\left(\frac{a^2 + u^2 - b^2}{2 \cdot a \cdot u}\right)$$

$$c^2 = u^2 + v^2 - 2 \cdot u \cdot v \cdot \cos\beta_2$$

$$\text{Úhel } \beta_2 \quad \beta_2 = \arccos\left(\frac{u^2 + v^2 - c^2}{2 \cdot u \cdot v}\right) \quad 12$$

$$\text{Úhel } \beta_3 \quad \sin\beta_3 = \frac{y_3 - p}{v} \quad \rightarrow \quad \beta_3 = \arcsin\left(\frac{y_3 - p}{v}\right) \quad 13$$

$$\text{Úhel } \beta \quad \beta = \beta_c = \beta_1 + \beta_2 + \beta_3 \quad 14$$

Výpočet úhlu natočení γ

$$u^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos\gamma_c$$

$$\text{Úhel } \gamma \quad \gamma_c = \arccos\left(\frac{a^2 + b^2 - u^2}{2 \cdot a \cdot b}\right) \quad 15$$

$$\gamma = 180 - \gamma_c$$

Výpočet úhlu natočení ψ

$$a^2 = u^2 + b^2 - 2 \cdot u \cdot b \cdot \cos\psi_1$$

$$\text{Úhel } \psi_1 \quad \psi_1 = \arccos\left(\frac{u^2 + b^2 - a^2}{2 \cdot u \cdot b}\right) \quad 16$$

$$v^2 = u^2 + c^2 - 2 \cdot u \cdot c \cdot \cos\psi_2$$

$$\text{Úhel } \psi_2, \psi_3 \quad \psi_2 = \arccos\left(\frac{u^2 + c^2 - v^2}{2 \cdot u \cdot c}\right) \quad 17$$

$$\psi_3 = \psi_2 - \sigma$$

$$\text{Úhel } \psi_4, \psi \quad \psi_4 = \psi_1 + \psi_2 \quad 18$$

$$\psi = \psi_4 - 90$$

Takto jsou vypočítány jednotlivé úhly natočení, které je třeba následně dosadit do rovnice regrese příslušného serva. Díky tomuto dosazení, řídicí program může vypočítat hodnoty pro serva, které odpovídají vypočtenému úhlu natočení pomocí inverzní kinematiky. Rovnice regrese je získána z grafu závislosti úhlu natočení na hodnotě posílané do serva. Měření této závislosti je popsáno v kapitole 4.1 Řízení serva

4.3 Senzor barvy

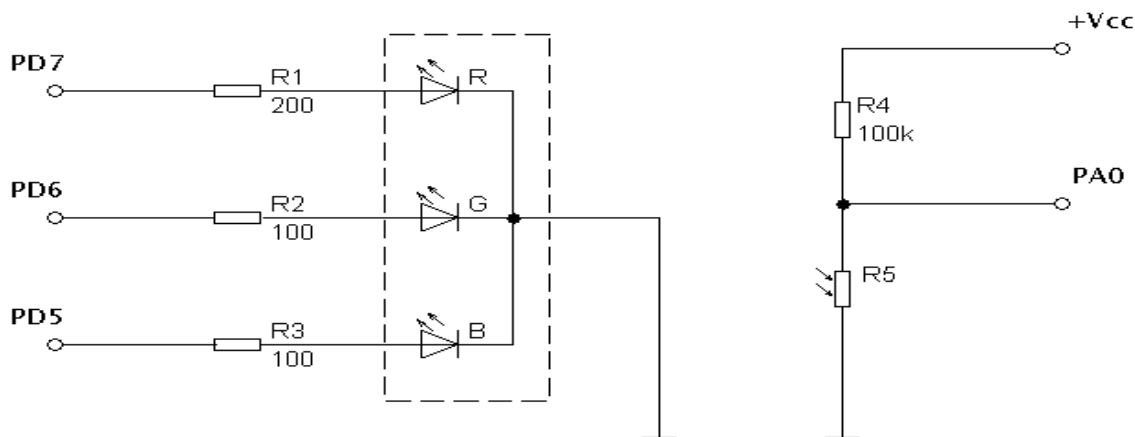
Senzor barvy tvoří jedna RGB dioda a napěťový dělič složený z pevného rezistoru a jednoduchého CdS (sulfid kadmia) fotorezistoru. Rozpoznávání barvy funguje na principu odrazu světla od barevného předmětu. Předmět se osvětluje jednotlivými barvami RGB diody, to znamená, že každá barva svítí na barevný předmět určitou dobu a nikdy nesvítí dvě barvy najednou. Intenzitou světla dopadající na fotorezistor se mění jeho odpor, tzn., že se mění úbytek napětí na fotorezistoru a toto napětí je následně přivedeno na A/D převodník, který převádí hodnotu napětí na číslicový signál, jenž je rozpoznáván programově pomocí mikropočítače. RGB dioda je použita z důvodu přesnosti senzoru, jelikož při použití pouze jedné barvy diody a následném osvětlení barevného předmětu by nemusel senzor barvu správně rozpoznat. U RGB diody je využíváno toho, že červenou, zelenou a modrou barvu například černý předmět pohlcuje jinak, než bílý a podle toho se poté vyhodnocuje odražená intenzita světla fotorezistorem.



Obr 4.5 Vyrobený senzor barvy

Nevýhodou použití takto sestaveného senzoru je, že obyčejný fotorezistor má určité zpoždění, to znamená, že pro získání konečné hodnoty úbytku napětí na fotorezistoru je potřeba svítit na fotorezistor určitou dobu. Experimentálním měřením se ukázalo, že tato doba je přibližně 300 ms, takže rozpoznání jedné barvy trvá 900 ms (osvit třemi barvami), což se ukazuje jako poměrně dlouhá doba, pokud by se tento senzor měl používat v některých rychlých aplikacích. Ovšem pro naše účely třídění barevných kostek postačuje. Další nevýhodou je, že fotorezistor po určitém čase degraduje, takže jeho hodnota odporu se mění

nezávisle na intenzitě světla s postupujícím časem. Tato degradace je umocněna navíc tím, když je fotorezistor uložen trvale v temném prostředí. Rychlost nezávislé změny odporu můžeme tedy zpomalit tak, že budeme skladovat fotorezistor na světle.



Obr 4.6 Schéma zapojení senzoru barvy

Důležité parametry použitých součástek:

RGB dioda

4 vývody (společný vývod Katoda)

Vyzařovací úhel 25°

Červená barva - vlnová délka: 625 nm, svítivost: 2000 mcd, proud: 20mA, napětí: 1,9 V

Zelená barva - vlnová délka: 520 nm, svítivost: 3000 mcd, proud: 20 mA, napětí: 3,2 V

Modrá barva - vlnová délka: 460 nm, svítivost: 1800 mcd, proud: 20 mA, napětí: 3,2 V

Provozní teplota: -40° C až 80°C

Fotorezistor

Odpor při 0 lx: 5 MΩ

Odpor při 10 lx: 55 ÷ 65 kΩ

Odpor při 100 lx: 4 ÷ 10 kΩ

Max. napětí: 150 V

Max. ztrátový výkon: 100 mW

Provozní teplota: -30°C až 70°C

Ovšem naměřené hodnoty odporu u fotorezistoru byly velmi odlišné od katalogových hodnot. Hodnota odporu při intenzitě světla 0 lx (tedy tma) byla přibližně 10 MΩ a odpor při svítivosti 100 lx (běžné osvětlení vnitřních prostor) byl 3,5 kΩ.

Výpočet předřadných odporů RGB diody:

Červená barva:

$$R1 = \frac{U - U_{LED}}{I_{LED}} = \frac{5 - 1,9}{20 \cdot 10^{-3}} = 155 \Omega \cong 200 \Omega$$

Zelená barva:

$$R2 = \frac{U - U_{LED}}{I_{LED}} = \frac{5 - 3,2}{20 \cdot 10^{-3}} = 90 \Omega \cong 100 \Omega$$

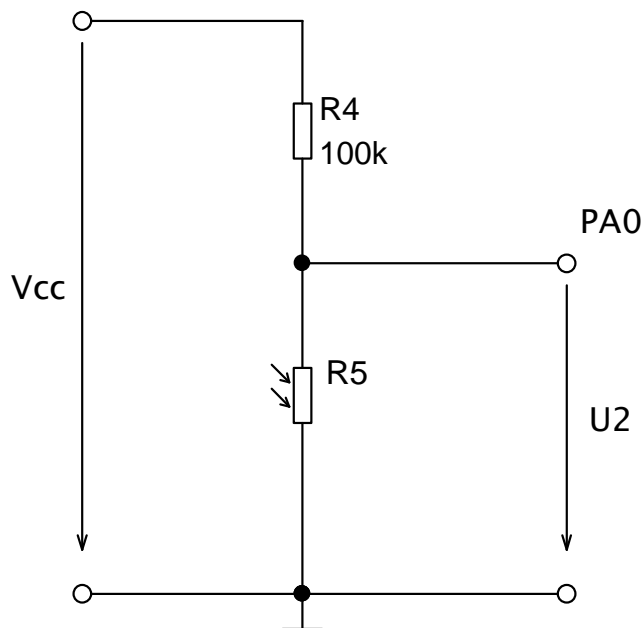
Modrá barva:

$$R3 = \frac{U - U_{LED}}{I_{LED}} = \frac{5 - 3,2}{20 \cdot 10^{-3}} = 90 \Omega \cong 100 \Omega$$

Pro jednotlivé barvy byly zvoleny odpory 200 Ω , 100 Ω a 100 Ω z důvodu blízkosti od vypočtených hodnot, zásob a větší šetrnosti k RGB diodě.

Výpočet pevného odporu napěťového děliče:

Pro volbu správné hodnoty pevného odporu je vycházeno z obvyčejného vzorce pro napěťový dělič. Odpor je zvolen tak, aby rozdíl úbytku napětí na fotorezistoru U2 při intenzitě světla 0 lx (tedy ve tmě) a intenzitě světla 100 lx byl co největší. Tím je zajištěn největší rozsah napětí, které je následně přivedeno na A/D převodník, který toto napětí převede na číslíkový signál.

**Obr 4.7** Napěťový dělič

Pro odpor R4=10 kΩ:

$$V_{CC} = 5 \text{ V}$$

$$R_5 = 3,5 \text{ k}\Omega \div 10 \text{ M}\Omega$$

$$U_{2_{100lx}} = V_{CC} \cdot \frac{R_5}{R_4 + R_5} = 5 \cdot \frac{3,5 \cdot 10^3}{10 \cdot 10^3 + 3,5 \cdot 10^3} = 1,296 \text{ V}$$

$$U_{2_{0lx}} = V_{CC} \cdot \frac{R_5}{R_4 + R_5} = 5 \cdot \frac{10 \cdot 10^6}{10 \cdot 10^3 + 10 \cdot 10^6} = 4,995 \text{ V}$$

$$\text{Rozdíl napětí: } \Delta U = U_{2_{0lx}} - U_{2_{100lx}} = 4,995 - 1,296 = 3,699 \text{ V}$$

Pro odpor R4 = 100 kΩ:

$$V_{CC} = 5 \text{ V}$$

$$R_5 = 3,5 \text{ k}\Omega \div 10 \text{ M}\Omega$$

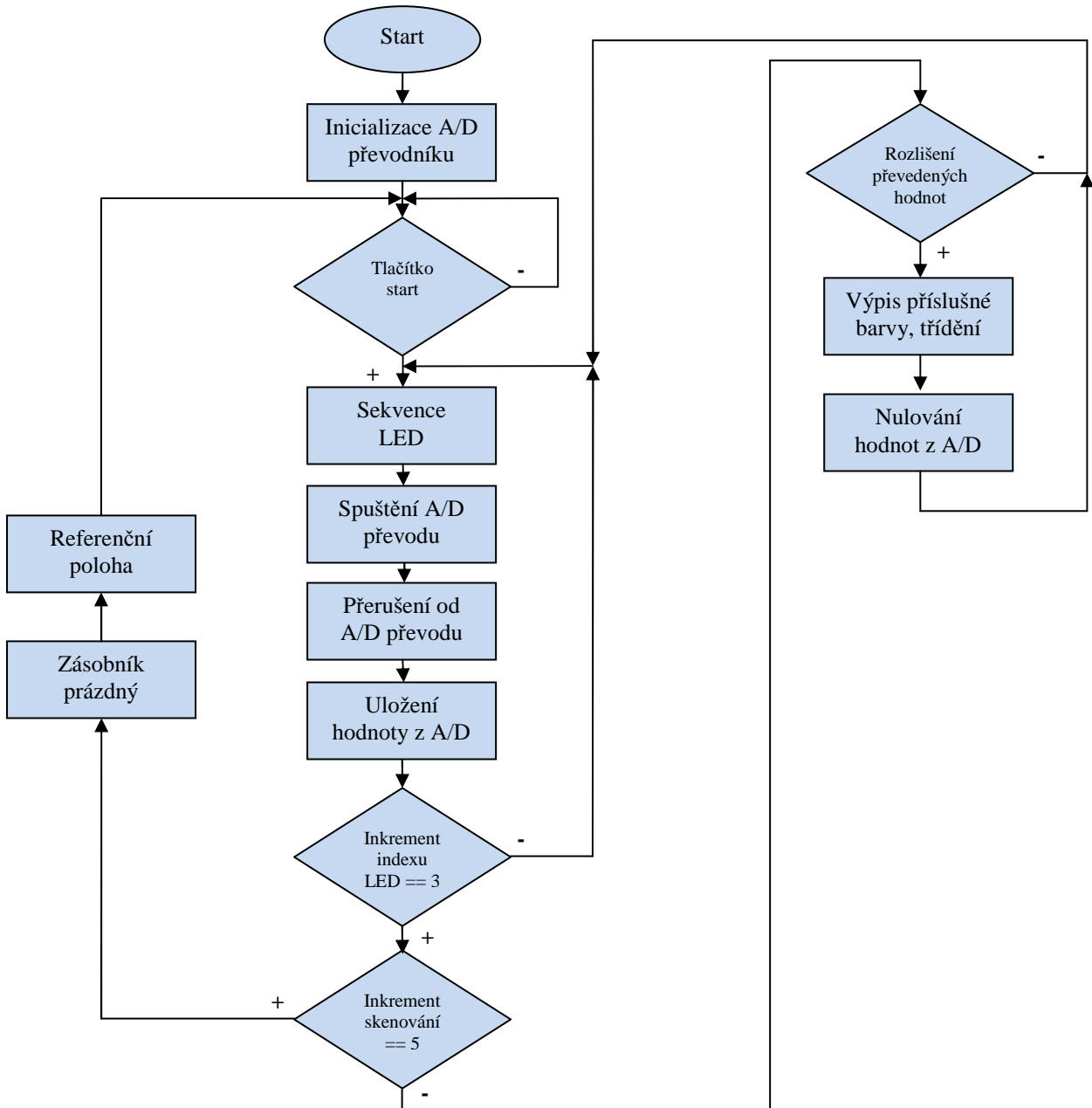
$$U_{2_{100lx}} = V_{CC} \cdot \frac{R_5}{R_4 + R_5} = 5 \cdot \frac{3,5 \cdot 10^3}{100 \cdot 10^3 + 3,5 \cdot 10^3} = 0,169 \text{ V}$$

$$U_{2_{0lx}} = V_{CC} \cdot \frac{R_5}{R_4 + R_5} = 5 \cdot \frac{10 \cdot 10^6}{100 \cdot 10^3 + 10 \cdot 10^6} = 4,950 \text{ V}$$

$$\text{Rozdíl napětí: } \Delta U = U_{2_{0lx}} - U_{2_{100lx}} = 4,950 - 0,169 = 4,781 \text{ V}$$

Pokud by byl tedy zvolen odpor R4 = 10 kΩ, rozdíl napětí by byl o více než 1 V menší, než při volbě odporu R4 = 100 kΩ. A/D převodníkem bychom převáděli tedy menší napětí a byla by snížena jeho citlivost.

Vývojový diagram řízení rozpoznávání barvy



4.4 Problémy a jejich řešení

Problém, který se vyskytl při řízení šesti serv pomocí mikropočítače Atmel ATmega644P byl ten, že tento mikropočítač dokáže generovat pomocí jednoho čítače/časovače PWM signál pouze na dva výstupní piny. Bylo tedy nutné vymyslet generování PWM signálu na více pinů, ne k tomuto účelu určených. Jako první způsob byl zvolen jeden hardwarový čítač a jeden softwarový čítač. Softwarový čítač čítal tak dlouho, dokud nedosáhl 20 ms – v tomto čase se vynuloval a na všechna serva se vyslal kladný pulz. Hardwarový čítač byl nastaven tak, že přetékal každých 10 μ s a tvořil tak rozlišení pulzně šířkově modulovaného signálu, jelikož při každém přetečení hardwarového čítače se testovala šířka pulzu posílaného na dané servo. Rozlišení PWM při takovémto způsobu řízení bylo nedostatečné, jelikož přidání několika dalších akcí, které by měl v tomto případě mikropočítač provádět (např. rozeznávání barev, výpis na displej, atd.) nebylo možné z důvodu přeskokování serv. Tento způsob by byl vhodný pouze pro hýbání robotem do daných pozic. Další věci by nebylo možné provádět, a proto bylo nutné navrhnout jiný způsob generace PWM signálu na ostatní piny mikropočítače.

Jako druhý a finální způsob řízení serva je posílání pulzů na jednotlivá serva postupně. Tento způsob je popsán v kapitole 4, podkapitola 4.1 Řízení serva.

Dalším problémem se ukázalo snímání barvy za chodu jakéhokoli serva. Jelikož při snímání se musí počkat 300 ms pro každou osvětlenou barvu RGB, než se ustálí hodnota na fotorezistoru (jedno snímání trvá tedy 900 ms) a snímání se provádí pětkrát za sebou, dokud není správně rozpoznána příslušná barva, generováním přerušení od přetečení čítače/časovače, shody registru čítače a porovnávacího registru a navíc ještě přerušení od A/D převodu je přerušovací jednotka mikropočítače zaneprázdněna a při chodu serva s A/D převodem servo přeskakuje. Tento problém je vyřešen tak, že snímání barvy probíhá tehdy, když je robot v poloze před zásobníkem, nebo v referenční poloze a není tak v pohybu.

Závěr

V této bakalářské práci byl realizován model manipulátoru, jeho řídicí jednotka a příslušné periferie. Celý automatizovaný systém je schopen plnit všechny na něj kladené požadavky, pro které byl navržen a mohl by být využit v praxi pro třídění barev boxů, či krabic. Díky použití inverzní kinematiky pro řízení pohybu manipulátoru (tedy ručnímu zadávání prostorových souřadnic), lze jednoduše modifikovat program a manipulátor by bylo možné použít v praxi i pro jiné aplikace. U použitého modelu je snížena přesnost najetí do koncové pozice vlivem nedokonalé konstrukce modelu a použití základních servomotorů. Tento problém by bylo možno odstranit zpevněním konstrukce, použitím kvalitních průmyslových servomotorů a hlavně použitím ložisek do všech kloubů. Ovšem pro splnění požadovaných funkcí manipulátoru je sestavený model více než dostačující.

K rozpoznání barvy byl navržen senzor, který osvětluje předmět jednotlivými základními barvami (červená, modrá, zelená) a odrazem světla od předmětu se snímá intenzita pomocí fotorezistoru. Tento princip rozpoznávání by byl v praxi vhodný u časově méně náročných procesů, jelikož čím déle je osvětlován fotorezistor, tím lépe se ustálí jeho hodnota odporu a tedy i úbytek napětí, který je převáděn A/D převodníkem. Tím je zajištěna větší přesnost určení barvy. Nepřesnosti jsou dány také nedokonalým nastříkáním kostek (kostky stejné barvy, ale odlišného odstínu). Pro rychlé procesy by mohl být použit senzor, který osvětluje předmět bílým světlem a odrážené světlo prochází přes barevné filtry na tři fotorezistory. Tím by bylo zrychleno snímání barvy třikrát. Nicméně navržený senzor disponuje kompaktními rozměry a při miniaturizaci (použití SMD součástek), by bylo možno umístit senzor i na malých a špatně přístupných místech.

Při návrhu řídicích algoritmů pro mikropočítač (programováno v jazyce C, vývojové prostředí AVR Studio 4) byl kladen důraz na správné generování PWM signálu pro jednotlivá serva, výpočet inverzní úlohy kinematiky, co nejpřesnější rozpoznávání barvy, řešení krizových situací při provozu manipulátoru a v neposlední řadě také na přehlednost a uživatelskou přívětivost výsledného programu. Řešením krizových situací manipulátoru je míněno například stisknutí tlačítka STOP v jakékoli pozici efektoru. Mikropočítač sám vyhodnocuje v jaké pozici je efektor manipulátoru (nejsou na manipulátor implementovány žádné senzory snímající jeho pohyb – vše je řešeno programově) a na základě toho musí při dané krizové situaci mikropočítač rozhodnout o tom, jak se má vrátit manipulátor například do referenční polohy a zda má v chapadle nějaký předmět, či ne. Řešení krizových situací je v praxi velmi důležité, jelikož je nutno zajistit bezpečnost všech lidí pohybujících se v manipulačním prostoru a také bezpečnost samotného manipulačního zařízení.

Použitá literatura

- [1] PINKER, Jiří. *Mikroprocesory a mikropočítače*. Praha: BEN – technická literatura, 2004. ISBN 80-7300-110-1
- [2] © 2013 ATMEL CORPORATION. ALL RIGHTS RESERVED. *Datasheet – ATmega644P* [online]. 2013, 505 str. [cit. 2014-03-02]. Dostupné z: http://www.atmel.com/Images/Atmel-8011-8-bit-AVR-Microcontroller-ATmega164P-324P-644P_datasheet.pdf
- [3] *Servocity – How do servos work?* [online]. [cit. 2014-03-25]. Dostupné z: https://www.servocity.com/html/how_do_servos_work_.html
- [4] AND-TECH.pl Electronics. *Datasheet – EvB 5.1 v5* [online]. 2013, 34 str. [cit. 2014-04-11]. Dostupné z: <http://and-tech.pl/wp-content/uploads/downloads/2013/10/Instrukcja-EvB5.1-v1-cze.pdf>
- [5] PINKER, Jiří. *Mikroprocesory a počítače*. [přednáška]. Plzeň: ZČU, 12. března 2014
- [6] *Robotroom – Cadmium-Sulfide color sensor* [online]. [cit. 2014-03-30]. Dostupné z: <http://www.robotroom.com/ColorSensor.html>
- [7] *Ifm – Senzor pro detekci barev s vysokým rozlišením* [online]. [cit. 2014-04-01]. Dostupné z: <http://www.ifm.com/ifmcz/web/news/colour-sensor-in-a-compact-o5-housing.html>
- [8] *Automatizace – Detekce a třídění produktů podle barev s kamerovými snímači Sick CVS1 a CVS2* [online]. [cit. 2014-04-25]. Dostupné z: <http://automatizace.hw.cz/detekce-a-trideni-produktu-podle-barev-s-kamerovymi-snimaci-sick-cvs1-a-cvs2>
- [9] SKÁLA, Jiří. Úvod do elektroniky. [přednáška]. Plzeň: ZČU, 24. únor 2014
- [10] BENEŠ, Petr. *Vybrané statě z elektrických pohonů – Úvod do robotiky*. [přednáška]. Plzeň: ZČU, 20. května 2014

- [11] BENEŠ, Petr. Vybrané statě z elektrických pohonů – Kinematické struktury robotických systémů. [přednáška]. Plzeň: ZČU, 20. května 2014
- [12] BENEŠ, Petr. Průmyslová elektronika a mechatronika – Průmyslové roboty a manipulátory. [přednáška]. Plzeň: ZČU, 23. května 2014
- [13] ŠOLC, František, ŽALUD, Luděk. Robotika. [přednáška]. Brno: FEKT, 1. října 2002. In: *Robotika, kybernetika Matescb* [online]. [cit. 2014-05-15]. Dostupné z: http://matescb.skvorskmal.cz/robotika_kybernetika/VUT_Brno_Robotika.pdf
- [14] MOSTÝN, Vladimír, KRYŠ, Václav. *Mechatronika průmyslových robotů*. Ostrava: VŠB – Technická univerzita Ostrava, 2010. ISBN 978-80-248-2610-3
- [15] *Lynxmotion – AL5B* [online]. [cit. 2014-01-04]. Dostupné z: <http://www.lynxmotion.com/c-126-al5b.aspx>
- [16] *RC modely cevaro – spínač (svetiel či niečoho iného) z elektroniky serva – 1. část* [online]. [cit. 2014-04-16]. Dostupné z: <http://rcmodely.cevaro.sk/index.php?id=434&n=spinac-svetiel-ci-niecoho-ineho-z-elektroniky-serva-%E2%80%93-1cast>