



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ŘÍZENÍ ROBOTA POMOCÍ JEDNODUCHÉ DIGITÁLNÍ KAMERY

APPLICATION OF SIMPLE DIGITAL CAMERA FOR ROBOT CONTROL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ KAKÁČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZBYNĚK FEDRA, Ph.D.

BRNO 2013



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Tomáš Kakáč

ID: 111063

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Řízení robota pomocí jednoduché digitální kamery

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi propojení jednoduchého obrazového senzoru s digitálním výstupem a procesoru (např. AVR). Zhodnoťte možnosti časování a použitelného rozlišení senzoru. Odhadněte množství procesorem zpracovávaných dat a s tím související optimální rozlišení. Otestujte základní komunikaci a získání dat, algoritmy pro zpracování dat získaných z obrazového senzoru a možnosti využití barevných nebo jasové složky obrazu.

Vytvořte knihovnu pro práci s obrazovým senzorem a jednoduchou dokumentaci k této knihovně. Otestujte připojení obrazového senzoru na polohovací rameno (servo motory). Otestujte využití pro řízení robotického podvozku např. aplikací sledující světelný či barevný objekt.

DOPORUČENÁ LITERATURA:

[1] MATOUŠEK, D. Práce s mikrokontroléry Atmel AVR. Praha: BEN - technická literatura, 2003.

[2] NOVÁK, P. MOBILNÍ ROBOTY - pohony, senzory, řízení. Praha: BEN - technická literatura, 2005.

Termín zadání: 11.2.2013

Termín odevzdání: 24.5.2013

Vedoucí práce: Ing. Zbyněk Fedra, Ph.D.

Konzultanti diplomové práce:

prof. Dr. Ing. Zbyněk Raida

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práce se zabývá návrhem komunikace s jednoduchým kamerovým modulem. Kamerový modul je propojen s mikroprocesorem pomocí sběrnice I²C. Dále se zaměřuje na získávání dat z kamery C3088 a nastavení řídicích registrů. Práce řeší získání černobílého snímku a pozorování bílého objektu. Pro polohování systému je použito servomechanismu. Pozorování objektu bylo využito pro řízení robotického podvozku. Kamera s podvozkem komunikuje přes rozhraní UART. Pozorovaný objekt je dán nastavením barvy RGB.

KLÍČOVÁ SLOVA

C3088, OV6620, SCCB, digitální kamera, zpracování obrazu, RP6

ABSTRACT

Master's thesis deals with communication with a simple camera module. The camera module is connected to the microprocessor via I2C bus. It then focuses on obtaining data from the camera C3088 and setting of the control registers. The work addresses the monochrome image acquisition and observation of the white object. Positioning servo system is used. Observed object was used to control the robot chassis. The camera communicates with the chassis via the UART interface. The observed object is given by setting the RGB colors.

KEYWORDS

C3088, OV6620, SCCB, digital camera, image processing, RP6

KAKÁČ, T. *Řízení robota pomocí jednoduché digitální kamery*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2013. 58 s., 17 s. příloh. Diplomová práce. Vedoucí práce: ing. Zbyněk Fedra, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Řízení robota pomocí jednoduché digitální kamery jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce ing. Zbyňku Fedrovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne

.....

(podpis autora)

OBSAH

| | |
|--|-------------|
| Seznam obrázků | viii |
| Seznam tabulek | x |
| Úvod | 1 |
| 1 Obrazové senzory | 2 |
| 1.1 Digitální kamerový modul C3088 | 2 |
| 1.1.1 CMOS snímač OV6620 | 4 |
| 1.1.2 Funkce modulu | 5 |
| 1.1.3 Video výstup | 6 |
| 1.1.4 Registry pro nastavení modulu | 7 |
| 1.2 Příklady projektů s modulem C3088 | 7 |
| 1.2.1 AVRcam | 8 |
| 1.2.2 CMUcam 4..... | 8 |
| 2 Mikrokontrolér | 10 |
| 2.1 PWM..... | 10 |
| 2.1.1 Fast PWM Mode..... | 10 |
| 2.1.2 Režim nulování časovače při komparaci (CTC)..... | 11 |
| 2.2 Sběrnice I ² C | 12 |
| 2.2.1 SCCB sběrnice..... | 12 |
| 3 Servomechanismus | 14 |
| 3.1 Konstrukce a funkce serva..... | 14 |
| 3.2 Řízení serva..... | 16 |
| 4 Robotický podvozek RP6 | 17 |
| 4.1 Řídící modul RP6 CONTROL M32 | 18 |
| 5 Návrh kamerového zařízení | 20 |
| 5.1 Polohovací rameno | 20 |
| 5.1.1 Vývojový kit | 21 |
| 5.1.2 Kamerový modul C3088..... | 22 |

| | | |
|----------|---|-----------|
| 5.1.3 | Ovládání serva | 22 |
| 5.1.4 | Paměť FIFO | 23 |
| 5.1.5 | Konstrukce | 23 |
| 5.2 | Řízení robotického podvozku | 24 |
| 5.2.1 | Konstrukce | 25 |
| 6 | Software | 27 |
| 6.1 | Sériová komunikace s PC | 27 |
| 6.2 | Zápis do paměti..... | 28 |
| 6.3 | Software podvozku | 29 |
| 6.4 | Software modulu RP6 CONTROL M32 | 30 |
| 6.5 | Vlastní programy pro μ P..... | 31 |
| 6.5.1 | Funkce photo..... | 31 |
| 6.5.2 | Funkce getcenter | 32 |
| 6.5.3 | Funkce track..... | 34 |
| 6.5.4 | Funkce pro ovládání podvozku..... | 35 |
| 7 | Dosažené výsledky | 36 |
| 7.1 | Průběhy z osciloskopu | 36 |
| 7.2 | Naměřené hodnoty | 37 |
| 7.3 | Získané obrázky z kamery | 38 |
| 8 | Závěr | 39 |
| | Literatura | 40 |
| | Seznam symbolů, veličin a zkratk | 41 |
| | Seznam příloh | 42 |

SEZNAM OBRÁZKŮ

| | | |
|-----------|---|----|
| Obr. 1.1: | Kamerový modul C3088 [7]. | 3 |
| Obr. 1.2: | Rozmístění pinů na desce C3088 [3]. | 3 |
| Obr. 1.3: | Rozložení pinů chipu OV6620 [4]. | 5 |
| Obr. 1.4: | Blokové schéma OV6620 [4]. | 5 |
| Obr. 1.5: | Výstupní signál [4]. | 6 |
| Obr. 1.6: | Formát 4:2:2 16 bitů a 8 bitů [4]. | 7 |
| Obr. 1.7: | AVRCam [17]. | 8 |
| Obr. 1.8: | CMUCam4 [15]. | 9 |
| Obr. 2.1: | Rozložení pinů pro pouzdro TQFP [5]. | 10 |
| Obr. 2.2: | Princip Fast PWM Modu [5]. | 11 |
| Obr. 2.3: | Princip CTC Modu [5]. | 11 |
| Obr. 2.4: | Zapojení sběrnice I2C. | 12 |
| Obr. 2.5: | Blokové schéma SCCB [11]. | 12 |
| Obr. 2.6: | Blokové schéma 2-vodičového SCCB [11]. | 13 |
| Obr. 2.7: | Časový diagram SCCB [11]. | 13 |
| Obr. 2.8: | Přenosová fáze [11]. | 13 |
| Obr. 3.1: | Řez serva od firmy Hitec [8]. | 14 |
| Obr. 3.2: | Řídící obvod serva. | 15 |
| Obr. 3.3: | Konektor serva [8]. | 16 |
| Obr. 3.4: | Šířky impulsů pro danou polohu serva [9]. | 16 |
| Obr. 4.1: | Robotický podvozek RP6 [13]. | 17 |
| Obr. 4.2: | Blokové schéma robotického podvozku. | 18 |
| Obr. 4.3: | Řídící deska M32 [13]. | 19 |
| Obr. 5.1: | Blokové schéma ramena. | 21 |
| Obr. 5.2: | Vývojový kit AVR EvB 4.3 [14]. | 21 |
| Obr. 5.3: | Zapojení modulu C3088. | 22 |
| Obr. 5.4: | Zapojení serva. | 22 |
| Obr. 5.5: | Blokové schéma paměti FIFO [16]. | 23 |
| Obr. 5.6: | Sestavený vývojový model s horizontálním natáčením. | 24 |
| Obr. 5.7: | Sestavený vývojový model s horizontálním a vertikálním natáčením. | 24 |

| | | |
|-----------|---|----|
| Obr. 5.8: | Blokové schéma návrhu robotnického podvozku s kamerou. | 25 |
| Obr. 5.9: | Sestavený robotický podvozek s kamerou. | 26 |
| Obr. 6.1: | Program HTerm. | 27 |
| Obr. 6.2: | Program CMUGUI..... | 28 |
| Obr. 6.3: | Konečný automat. | 30 |
| Obr. 6.4: | Schématicky autonomní chování. | 30 |
| Obr. 6.5: | Vývojový diagram funkce photo. | 31 |
| Obr. 6.6: | Vývojový diagram funkce getcenter. | 32 |
| Obr. 6.7: | Vývojový diagram funkce getcenter. | 33 |
| Obr. 6.8: | Vývojový diagram funkce track..... | 34 |
| Obr. 6.9: | Vývojový diagram hlavní funkce programu pro ovládání podvozku. | 35 |
| Obr. 7.1: | Časový průběh řídicího impulsu serva pro natočení 0°. | 36 |
| Obr. 7.2: | Časový průběh horizontální reference modulu C3088 HREF. | 36 |
| Obr. 7.3: | Časový průběh vertikální synchronizace modulu C3088 VSYN. | 37 |
| Obr. 7.4: | Časový průběh časování obrazových bodů modulu C3088 PCLK..... | 37 |
| Obr. 7.5: | Získaný obrázek č. 1 z kamerového modulu s rozlišením 352 x 244. | 38 |
| Obr. 7.6: | Získaný obrázek č. 2 z kamerového modulu s rozlišením 320 x 240. | 38 |

SEZNAM TABULEK

| | | |
|-----------|--|----|
| Tab. 1.1: | Přehled pinů na modulu [3]..... | 4 |
| Tab. 1.2: | Přehled registrů pro nastavení [4]..... | 7 |
| Tab. 7.1: | Přehled snímkové rychlosti jednotlivých řešení. | 37 |

ÚVOD

Cílem mé diplomové práce bylo seznámit se s možnostmi propojení jednoduchého obrazového senzoru s digitálním výstupem a procesorem. Ověřit možnosti získání dat číslcového výstupu, ovládání kamerového modulu přes sběrnici SCCB, komunikaci s nadřazeným prostředkem, algoritmy pro zpracování dat získané z obrazového senzoru. Dále jsem se zaměřil na polohování kamerového modulu panoramatickou a naklápěcí hlavou ovládanou modelářskými servomechanismy.

Robotika využívá technologie řady oborů. Velmi důležitou součástí robotů je sensorika, která zprostředkovává vazbu mezi robotem a okolním světem. Běžně se roboty osazují čidly základních fyzikálních veličin. Pro orientaci v okolním prostředí se využívají infračervené a ultrasonické snímače. Pro navigaci a stabilizaci polohy se využívá odometrie, elektronické kompas, akcelerometry, gyroskopy a ve venkovním prostředí GPS moduly.

Pro řadu robotických aplikací však uvedená sensorika nedostačuje. Pro správnou funkci aplikace je požadována přesnější znalost prostředí, ve kterém se robot pohybuje. Mezi typickým příkladem může být robotický fotbal, navigace mobilního robota po vytyčené cestě (například chodníku), vyhledávání a sběr předmětů, třídění objektů, vytváření mapy prostředí atd.

V robotice je pak nutné nahradit důležitý smysl člověka zrak. Tento smysl zprostředkovává až 80% informací o okolním prostředí. Zrak reaguje na světlo, různé barvy a tvary, které nás obklopují. K náhradě tohoto smyslu v robotice může sloužit tzv. kamerový systém (optický senzor), který se v určitých případech stále více používá jako hlavní informační kanál pro robota.

Práce je rozdělena do dvou částí. První část obsahuje teoretické poznatky potřebného hardwaru k funkčnosti obrazového senzoru. Na začátku se seznámíme s kamerovým modulem C3088 a jeho vlastnostmi a nastavením. Dále si popíšeme použití řídicí jednotky ATmega32 a její periférie PWM modulace a sběrnice I2C. Rozebereme funkci servomechanismu a jeho řízení pomocí procesoru. Na závěr teoretické části se seznámíme s robotickým podvozkem RP6 a jeho vlastnostmi.

V druhé části se zaměříme na praktické poznatky z realizace připojení kamerového modulu s 8-bitovým mikroprocesorem, přenosem a zpracováním dat na PC. Dále je rozebráno připojení modulu k robotickému podvozku a jeho softwarového odladění. Výsledky experimentů jsou vyhodnoceny a závěry jsou promítnuty do strategie postupu návrhu konstrukce mikroprocesorové jednotky strojového vidění.

1 OBRAZOVÉ SENZORY

System strojového (počítačového) vidění jsou relativně složité části řídicích systémů. Vedle složité technologie snímání obrazu (lokalizace na objekt, ostření, clona, vyvážení bílé a přizpůsobení intenzitě okolního osvětlení) je potřeba velký výpočetní výkon při sběru dat ze snímače a jejich analýzu.

V robotice můžeme najít řadu přístupů k aplikaci strojového vidění. Aplikace se liší podle určení a funkce robota. Rozdílnost jednotlivých konstrukcí je ve výběru snímače a využití výpočetních prostředků řídicího systému.

Portfolio optických kamerových snímačů je široké a základní pravidla výběru jsou dvě požadované vlastnosti a cena. Vzhledem ke skutečnosti, že jsou tato kritéria protikladná, hledá se vzájemný kompromis. Na pozici kamerového snímače můžeme použít:

- pohybové snímače (např. Microsoft Kinetics),
- IP kamery s výstupem dat přes Ethernet nebo bezdrátové technologii,
- webové a USB kamery,
- analogové kamery,
- digitální kamerové snímače,
- speciální levné kamery (původně součást hraček, herních periférií apod).

Je třeba mít na paměti, že u všechny výše uvedené kamerové snímače se mohou lišit formou snímání obrazu (barevné, černobílé – stupnice šedi nebo infračervené). Dalším důležitým parametry jsou rozlišení a snímková rychlost.

Využití výpočetního výkonu řídicí jednotky nabízí dvě základní řešení:

- přímé připojení kamerového snímače k I/O periférii řídicí jednotky,
- samostatná jednotka pro zpracování obrazu ze snímače.

Přímé připojení odebírá velký objem výpočetních prostředků řídicí jednotky, ale umožňuje snadnou modifikaci algoritmů strojového vidění pro potřeby aplikace. Samostatná jednotka sice vyžaduje další výpočetní prostředek, ale umožňuje vytvořit univerzální a flexibilní modul, který najde využití ve více aplikacích.

Semestrální práce se zabývá aplikací levného digitálního barevného kamerového modulu v samostatné jednotce pro zpracování obrazové informace. Pro spolupráci s nadřazeným prostředkem (PC, hlavní řídicí jednotka robota...) se uvažuje se sériovým přenosem příkazů a dat.

1.1 Digitální kamerový modul C3088

Modul C3088 je 0,25" barevná kamera s digitálním výstupem. Kamera využívá CMOS snímač obrazu OV6620 od firmy Omnivision. I přes nízkou cenu výrobku poskytuje na výstupu velmi kvalitní barevný obraz. Digitální video port dodává nepřetržitý tok obrazových dat se šířkou 4/8/16 bitů. Výstup obrazu můžeme zpracovávat ve formě digitálních dat nebo jako černobílý analogový video signál.



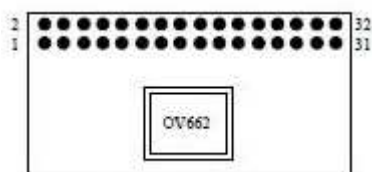
Obr. 1.1: Kamerový modul C3088 [7].

Kamerový modul má rozměry 40 x 28 mm a je napájen 5V. Modul má velmi malé nároky na spotřebu. Při aktivním režimu odebírá kamera 80mW a při režimu STAND BY pouze 30μW. Poměr signálu k šumu je 48dB.

Maximální rozlišení obrazu získaného z kamery je 356 x 292 obrazových bodů (pixelů). Velikost rozlišení lze měnit v rozmezí od 2 x 2 pixelu až po výše zmiňované maximum dané výrobcem. Standardně se používá rozlišení typu CIF (355 x 288) a QCIF (176 x 144). Další důležitou vlastností, ovlivňující zpracování obrazu v robotice, je rychlost snímání. Modul C3088 umožňuje snímací rychlost až 30 snímků/s. Rychlost snímání je ovlivňována hodinovým signálem. Vlastní modul je vybaven krystalem o kmitočtu 17,7344MHz. Pokud chceme přivést vlastní kmitočet lze využít pin EXCLK.

Obrazový snímač je opatřen objektivem s mechanicky nastavitelným ohniskem. Hloubka ostrosti se dá ovlivnit clonou, která se nastavuje řídicími registry snímače.

Kamerový modul má vyvedeno rozhraní s celkem 32 vývody pro napájení, ovládní kamery a získávání dat obrazu. K propojení je třeba kontaktní lišta 2x16 vývodů s roztečí 2,54 mm. Mechanické upevnění modulu na konstrukci se provádí pomocí konektoru a dvou montážních otvorů s průměrem 2,54 mm.



Obr. 1.2: Rozmístění pinů na desce C3088 [3].

Pro výstupní obrazová data z kamery obsahuje modul 16 pinů. Ve formátu YCrCb lze pomocí 8 pinů získat informace o jasové (Y) složce obrazu a pomocí zbylých 8 pinů informace o chromatických (UV) složkách obrazu. Díky rozdělení těchto pinů lze z kamery získat podle potřeby černobílý nebo barevný obraz. Popis jednotlivých funkcí pinů je popsáno v následující tabulce 1.1.

Tab. 1.1: Přehled pinů na modulu [3].

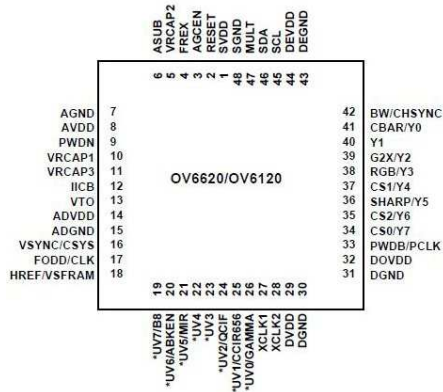
| Pin | Označení | Vlastnost |
|---------|-----------|---|
| 1 – 8 | Y0 – Y7 | Digitální výstup datové sběrnice Y |
| 9 | PWDN | Vstup ovládní režimu snížené spotřeby |
| 10 | RST | Reset |
| 11 | SDA | Sběrnice I ² C - linka sériových dat |
| 12 | FOOD | Příznak lichého obrazového pole |
| 13 | SCL | Sběrnice I ² C - linka sériových hodin |
| 14 | HREF | Výstup vertikální synchronizace |
| 15 | AGND | Analogová zem |
| 16 | VSYN | Výstup vertikální synchronizace |
| 17 | AGND | Analogová zem |
| 18 | PCLK | Výstup časování obrazových bodů |
| 19 | EXCLK | Vstup externího zdroje hodin |
| 20 | VCC | Napájecí stejnosměrné napětí 5V |
| 21 | AGND | Analogová zem |
| 22 | VCC | Napájecí stejnosměrné napětí 5V |
| 23 – 30 | UV0 – UV7 | Digitální výstup datové sběrnice UV |
| 31 | GND | Společná zem |
| 32 | VTO | Výstup analogového videa (75Ω) |

Nastavení parametrů a vlastností kamery se provádí zápisem do řídicích registrů obrazového snímače prostřednictvím rozhraní I²C. Mezi nejpoužívanější změny nastavení patří změna expozice, gama korekce, rozlišení, vyvážení bílé a další.

1.1.1 CMOS snímač OV6620

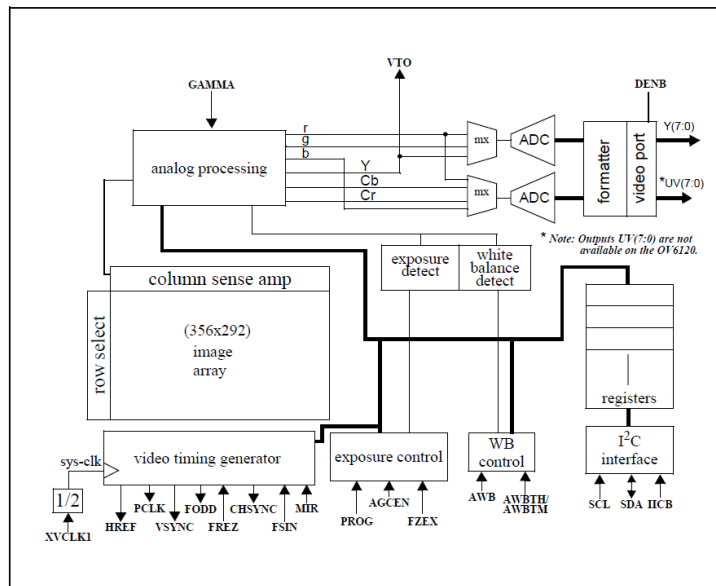
CMOS barevný obrazový snímač OV6620 je jednočipový video/obrazový kamerový obvod navržený tak, aby poskytoval vysokou úroveň funkčnosti v jediném malém pouzdru. Snímací část obvodu obsahuje obrazovou oblast 352x288 bodů, která je schopná zachytit až 60 snímků za sekundu. Patentovaná technologie snímače využívá moderní algoritmy k potlačení šumu s pevným motivem (FPN), eliminuje roztékání náboje a při expozici výrazně redukuje paprskové světlo. Všechny funkce kamery včetně řízení expozice, gama korekce, velikost zisku, vyvážení bílé, barevné matice, vytváření oken a další vlastnosti se programují přes rozhraní I²C. Obrazový procesor může být naprogramován tak, aby poskytoval výstup ve 4, 8 nebo 16 bitovém formátu obrazových dat. Snímač obsahuje obrazové pole 352x288 bodů, analogový signálový procesor, dvojici 8-bitových A/D převodníků, analogový video multiplex, blok časovače a další obvody viz. obrázek 1.4. Konstrukce snímače je založena na obrazovém poli s integrovaným systémem vyčítání a přenosem obrazových dat po řádcích a elektronickou závěrkou se synchronizovaným schématem vyčítání. Barevný filtr se skládá z barevné matice RG/GB.

Aplikace obrazového snímače OV6620 zahrnují: datový tok video konference, video telefon, video poštu, snímání stabilního obrazu, PC multimedia, sensoriku a počítačové vidění.



Obr. 1.3: Rozložení pinů chipu OV6620 [4].

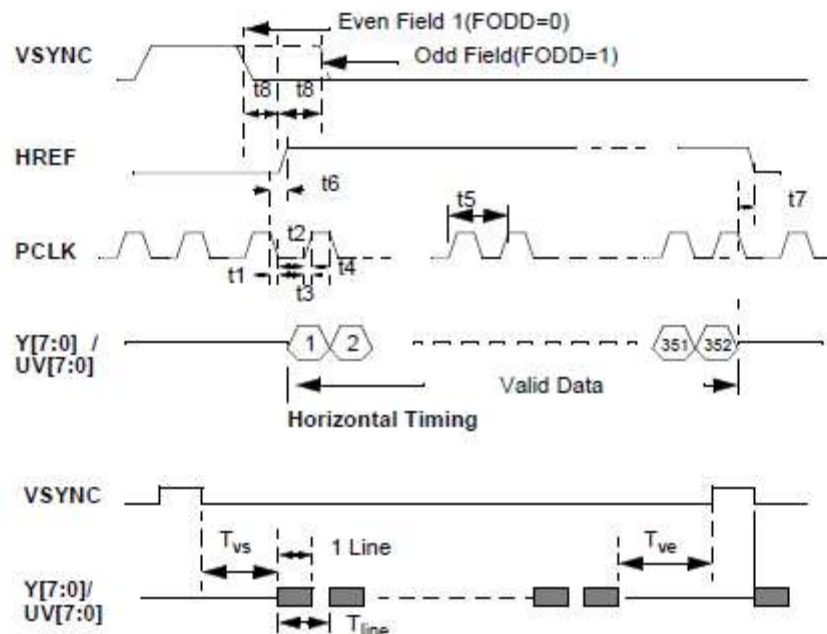
Pro přehlednost uvádím blokové schéma, které nám demonstruje, jakým způsobem modul pracuje a realizuje převod snímané scény na digitální data. Obraz je zachycen na matici 356 x 592 obrazových bodů a přiveden do části analogového zpracování, kde dochází k podstatnému zpracování signálu. Surový datový signál ve formátu YCrCb/RGB je přiváděn na dva integrované A/D převodníky. Zpracovaný signál se distribuuje na digitální video port a přes video multiplexer poslán na správné výstupní piny.



Obr. 1.4: Blokové schéma OV6620 [4].

1.1.2 Funkce modulu

Nejjednodušší funkce obvodu je v režimu pro formát Zoom Video Port Formát. V tomto formátu představuje 8 bitů složku Y jednoho obrazového bodu. Další 8 bitů představuje složky U a V. Informace obrazových bodů, které jsou přenášeny nepřetržitě, jsou časovány hodinovým signálem PCLK a synchronizovány signály HREF a VSYNC.



Obr. 1.5: Výstupní signál [4].

Signál VSYNC indikuje nový snímek, signál HREF indikuje, že je informace platná a provádí horizontální synchronizaci, PCLK je hodinový signál obrazových bodů (data jsou platná při náběžné hraně). Perioda PCLK lze měnit pomocí registru. To umožňuje mikroprocesoru přímé čtení snímků z kamery bez použití dalšího hardwaru.

1.1.3 Video výstup

Video výstup poskytuje řadu výstupních formátů a také standardních alternativ, aby datový tok vyhovoval řadě aplikačních požadavků. Obrazový prvek podporuje výstupní formáty CCIR601 i CCIR656 v následujících konfiguracích:

- 16-bitový formát 4:2:2,
- 8-bitový datový režim,
- 4-bitový nibble režim,
- formát 704 x 288.

Výstupní formát videa posílá data ve tvaru jasové složky (Y) a barevných složek (U,V). Pro převod těchto hodnot do známějšího tvaru RGB lze využít následující vztahy:

- $Y = 0,59G + 0,31R + 0,11B,$
- $U = R - Y,$
- $V = B - Y.$

| Data Bus | Pixel Byte Sequence | | | | | |
|----------|---------------------|----|----|----|----|----|
| Y7 | Y7 | Y7 | Y7 | Y7 | Y7 | Y7 |
| Y6 | Y6 | Y6 | Y6 | Y6 | Y6 | Y6 |
| Y5 | Y5 | Y5 | Y5 | Y5 | Y5 | Y5 |
| Y4 | Y4 | Y4 | Y4 | Y4 | Y4 | Y4 |
| Y3 | Y3 | Y3 | Y3 | Y3 | Y3 | Y3 |
| Y2 | Y2 | Y2 | Y2 | Y2 | Y2 | Y2 |
| Y1 | Y1 | Y1 | Y1 | Y1 | Y1 | Y1 |
| Y0 | Y0 | Y0 | Y0 | Y0 | Y0 | Y0 |
| UV7 | U7 | V7 | U7 | V7 | U7 | V7 |
| UV6 | U6 | V6 | U6 | V6 | U6 | V6 |
| UV5 | U5 | V5 | U5 | V5 | U5 | V5 |
| UV4 | U4 | V4 | U4 | V4 | U4 | V4 |
| UV3 | U3 | V3 | U3 | V3 | U3 | V3 |
| UV2 | U2 | V2 | U2 | V2 | U2 | V2 |
| UV1 | U1 | V1 | U1 | V1 | U1 | V1 |
| UV0 | U0 | V0 | U0 | V0 | U0 | V0 |
| Y FRAME | 0 | 1 | 2 | 3 | 4 | 5 |
| UV FRAME | 0 | 2 | 4 | | | |

| Data Bus | Pixel Byte Sequence | | | | | | | |
|----------|---------------------|----|----|----|----|----|----|----|
| Y7 | U7 | Y7 | V7 | Y7 | U7 | Y7 | V7 | Y7 |
| Y6 | U6 | Y6 | V6 | Y6 | U6 | Y6 | V6 | Y6 |
| Y5 | U5 | Y5 | V5 | Y5 | U5 | Y5 | V5 | Y5 |
| Y4 | U4 | Y4 | V4 | Y4 | U4 | Y4 | V4 | Y4 |
| Y3 | U3 | Y3 | V3 | Y3 | U3 | Y3 | V3 | Y3 |
| Y2 | U2 | Y2 | V2 | Y2 | U2 | Y2 | V2 | Y2 |
| Y1 | U1 | Y1 | V1 | Y1 | U1 | Y1 | V1 | Y1 |
| Y0 | U0 | Y0 | V0 | Y0 | U0 | Y0 | V0 | Y0 |
| Y FRAME | 0 | 1 | 2 | 3 | | | | |
| UV FRAME | 0 | 1 | 2 | 3 | | | | |

Obr. 1.6: Formát 4:2:2 16 bitů a 8 bitů [4].

1.1.4 Registry pro nastavení modulu

Modul obrazového snímače obsahuje celkem 80 registrů pro nastavení vlastností kamery. Přístup k registrům je, jak již bylo zmíněno výše, přes sběrnice I²C. Následující tabulka poskytuje seznam pouze některých registrů, které se využívají nejčastěji. Popis zbývajících registrů a význam jednotlivých bitů lze nalézt v katalogovém listu obvodu OV6620.

Tab. 1.2: Přehled registrů pro nastavení [4].

| Dílčí adresa (hex) | Registr | Výchozí hodnota (hex) | Popis |
|--------------------|----------|-----------------------|--------------------------------|
| 05 | Kontrast | 48 | Řízení kontrastu (FF – vysoký) |
| 06 | Jas | 80 | Řízení jasu (FF – vysoký) |
| 07 | Ostrost | C6 | Řízení ostrosti obrazu |
| 10 | AEC | 9A | Automatické řízení expozice |
| 12 | COMA | 24 | Nastavení formátu videa |
| 13 | COMB | 01 | Nastavení formátu videa |
| 14 | COMC | 00 | Nastavení formátu videa |
| 3D | COMN | 08 | Povolení analogového výstupu |

1.2 Příklady projektů s modulem C3088

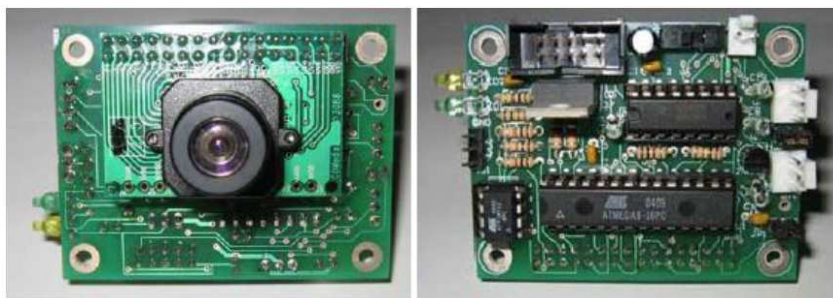
Na závěr této kapitoly uvádím některé již zpracované projekty s tímto kamerovým modulem. Nejznámější volně dostupné projekty jsou AVRcam a CMUcam. Modul CMUcam4 byl využit pro zprovoznění robotického podvozku.

1.2.1 AVRcam

AVRCam je plně programovatelný kamerový modul založený na procesoru Atmel AVR ATmega8. Obraz je snímán CMOS kamerou Omnivision OV6620 s rozlišením 88x143 pixelů. Spotřeba je 300mW.

Modul je dodáván se SW pro sledování, který je schopný najít až 8 objektů stejné barvy v obraze s rychlostí 30 fps. Dále je dodáván programátor s kompilátorem jazyka C a zkušební aplikace. Cena kompletního setu je \$100.

V současné době není tento produkt dostupný. V rámci semestrálního projektu jsem sestavil obvod modulu AVRcam na univerzální desce a odzkoušel jeho možnosti. Konstrukce byla vytvořena pro soutěž časopisu Circuit Cellar v roce 2004, ve které získala 2. cenu. Hlavní výhodou konstrukce je její cena. Funkčnost kamerového snímání je značně omezena (omezení datových sběrnic na 4 bity a nízké rozlišení obrazu). Motivační pro vznik modulu AVRcam bylo vytvoření levné alternativy k systému CMUcam.



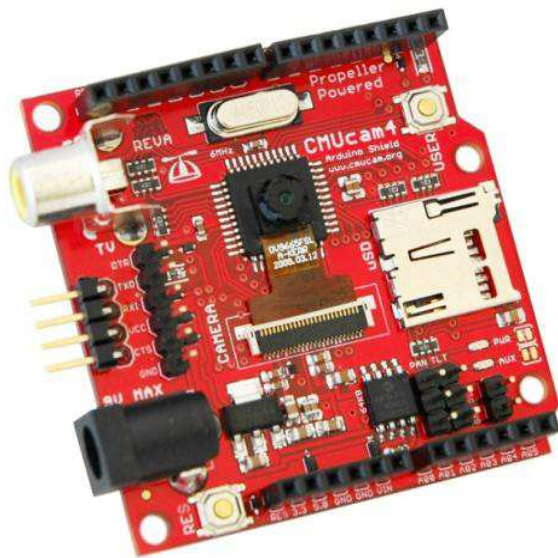
Obr. 1.7: AVRcam [17].

1.2.2 CMUcam 4

CMUcam4 je nejnovější model řady CMUcam představený v únoru 2012. Cílem projektu je poskytnout jednoduchý prostředek strojového vidění pro malé mikroprocesorové systémy v podobě inteligentního senzoru. CMUcam4 je plně programovatelný integrovaný počítačový kamerový snímač. Hlavní procesor je Parallax P8X32A (Propeller Chip) spojený s CMOS kamerovým snímačem OmniVision 9665.

CMUcam4 lze použít ke sledování barev nebo ke shromažďování základních statistických údajů o obraze. Nejlepší výkon lze dosáhnout, pokud v obraze existují vysoce kontrastní a intenzivní barvy. Například lze snadno sledovat červenou kuličku na bílém pozadí, ale těžko se vzájemně rozlišují různé odstíny hnědé v měnícím se světle. Sledování barevných objektů může být použito k lokalizaci trasovacích značek, sledování čáry nebo pronásledování pohyblivých světel. Pomocí barevných statistik umožňuje CMUcam4 sledovat scénu nebo detekovat určitou barvu. Pomocí "režimu line", může CMUcam4 generovat binární obrazy barevných objektů s nízkým rozlišením. To se může použít k sofistikovanějšímu zpracování obrazu, které zahrnuje sledování čáry s detekcí odbočky nebo dokonce rozpoznávání jednoduchých tvarů. Stejně jako v případě normální digitální kamery může tento typ zpracování obrazu vyžadovat výkonný počítač nebo alespoň rychlý mikrokontrolér.

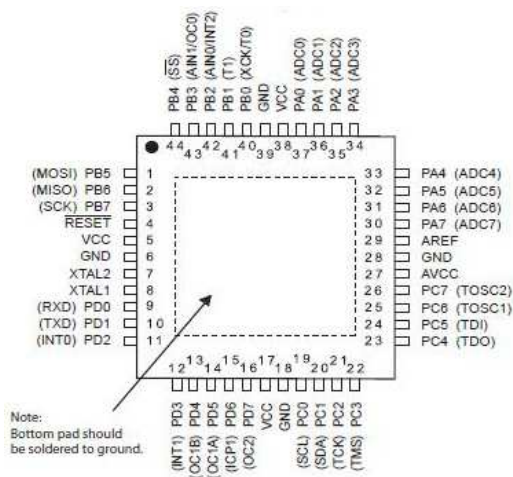
Nejběžnější konfigurace CMUcam4 je komunikace s hlavním procesorem pomocí standardního TTL sériového portu. Tento "master procesor" by mohl být počítač (připojený přes USB nebo RS232), Arduino, PIC, nebo podobný mikrokontrolér. Jeho komunikační protokol a přenosová rychlost je navržena tak, aby kapacitně vyhovovala i nejpomalejším procesorům. U ještě pomalejších procesorů, může CMUcam4 pracovat v "dotazovacím režimu". V tomto režimu, může nadřazený procesor požádat CMUcam4 o jediný paket dat. To pomalejším procesorům umožňuje snadnější synchronizaci dat. Srdcem kamerového modulu CMUcam4 je mikroprocesor Propeller od firmy Parallax. Původním záměrem součástky byl malý jedno jádrový procesor, ale v průběhu iteračního procesu návrhu, testování, sestavování software a vyhodnocování výsledků společně s klesající cenou křemíku skončili na osmi jádrovém řadiči na jediném čipu. Tento mikroprocesor je také snadno dostupný pro amatéry, za velmi rozumnou cenu – 5 US\$ z Digikey nebo Farnell. Dodává se ve standardním 40-pin DIL pouzdrů a povrchové QFP formátu pro povrchovou montáž.



Obr. 1.8: CMUcam4 [15].

2 MIKROKONTROLÉR

Pro řízení navrhovaného systému zpracování obrazových dat byl zvolen osmibitový mikroprocesor firmy Atmel typu AVR řady ATmega, konkrétně ATmega32 – 16PU. Tento procesor je vybaven 32kB pamětí programu FLASH, 2kB pamětí SRAM, 1kB EEPROM, 2 x 8bit a 1 x 16bit čítačem/časovačem, 4 PWM kanály, 8 kanálový AD převodník s rozlišením 10 bitů, integrovanými periferiemi USART, SPI a I²C. Rozmístění jednotlivých pinů procesoru lze sledovat na následujícím obrázku 2.1.



Obr. 2.1: Rozložení pinů pro pouzdro TQFP [5].

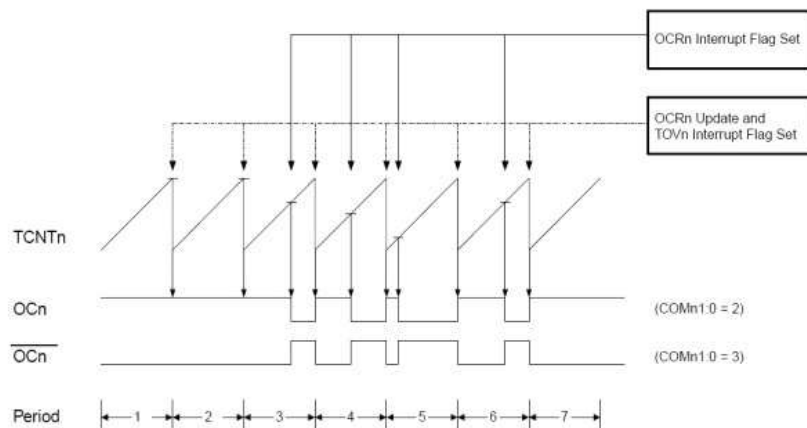
Projekt mimo porty, na které jsou přivedeny obrazové data z kamery, dále využívá integrovanou periférii impulzně šířkové modulace (PWM) pro ovládání pohybu serva. Další důležitou periférii, kterou využíváme je interface sběrnice I²C. Abychom se blíže seznámili s funkcí těchto periférii, popíšeme si v následujícím textu jejich činnost.

2.1 PWM

Pulzní šířková modulace (PWM) může být snadno generována pomocí čítače/časovače, který obsahuje také vybraný mikroprocesor. Generování signálu pulzně šířkovou modulací může být realizováno v různých modech a s různým nastavením časovače. Toto nastavení lze měnit pomocí řídicích registrů popsanych v katalogovém listu součástky. Pro ovládání serva je vhodný Fast Mode nebo CTC Mode generování PWM.

2.1.1 Fast PWM Mode

V tomto režimu může být ovlivňována pouze šířka impulzu, kterou nastavíme registrem OCRx. Výstupní průběh pulzu je uveden níže.



Obr. 2.2: Princip Fast PWM Modu [5].

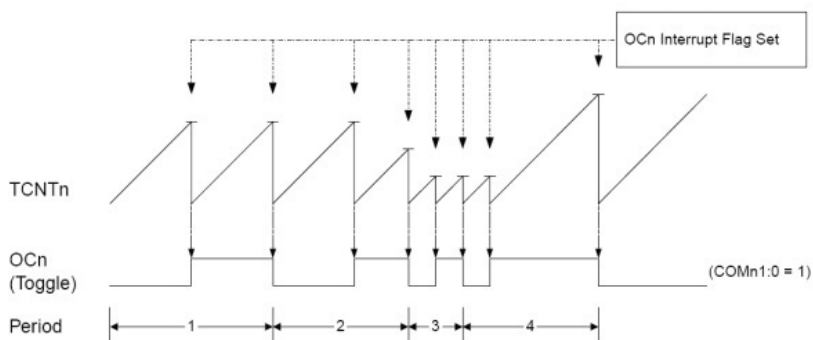
Frekvenci PWM nastavíme pomocí rovnice:

$$f = \frac{f_{clk}}{N \cdot 256}, \quad (2.1)$$

kde f_{clk} je frekvence krystalu mikroprocesoru a N je hodnota předřazené děličky.

2.1.2 Režim nulování časovače při komparaci (CTC)

Tento režim již umožňuje více nastavení vlastností generovaného PWM signálu. Základní režim je nulování čítače a vyvolání přerušování při shodě čítače a registru OCRx. Funkce je naznačena na obrázku.



Obr. 2.3: Princip CTC Modu [5].

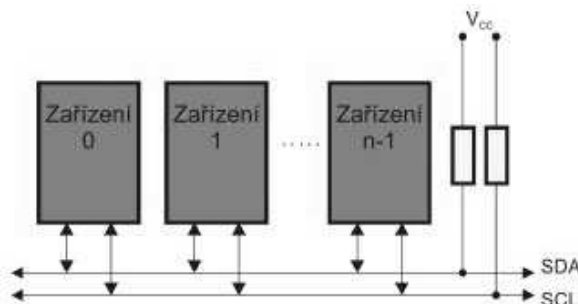
Frekvenci PWM nastavíme pomocí rovnice:

$$f = \frac{f_{clk}}{N \cdot OCRx}, \quad (2.2)$$

kde f_{clk} je frekvence krystalu mikroprocesoru, N je hodnota předřazené děličky a OCRx je komparační registr.

2.2 Sběrnice I²C

Sběrnice I²C byla vyvinuta firmou Philips Semiconductor pro komunikaci mezi dílčími funkčními bloky nebo obvody uvnitř jednoho zařízení. Komunikační protokol sběrnice umožňuje propojení až 128 zařízení pomocí dvou obousměrných vodičů SDA a SCL.



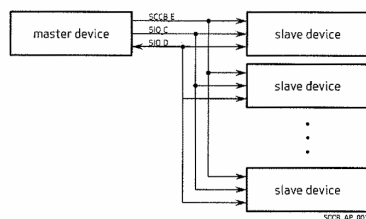
Obr. 2.4: Zapojení sběrnice I2C.

Všechna zařízení připojená na sběrnici musí mít individuální 7bitovou adresu a implementovaný mechanismus komunikace, který splňuje specifikace sběrnice I²C. Komunikační protokol využívá dva typy zařízení a to master a slave.

Sběrnice I²C je v projektu využita pro ovládání kamery přes nastavení registrů modulu. Při důkladném studiu katalogového listu obvodu OV6620 zjistíme, že protokol používaný pro komunikaci není běžná specifikace I²C, ale že se jedná sběrniceový systém SCCB vyvinutý firmou Omnivision. Ze specifikace tohoto protokolu, můžeme vidět, že je v některých režimech stejný jako protokol I²C. Zásadní rozdíl je v tom, že používá tři vodiče. Je však možný také dvojvodičový režim. Přenos dat začíná stejně jako u sběrnice I²C vystavením podmínky START a končí podmínkou STOP. Přenosy jednotlivých byte se označují jako fáze přenosu. Posloupnost čtení a zápisu dat do registru v zařízení slave je stejný jako u I²C. Jediný rozdíl je v časování událostí na sběrnici a používání potvrzovacího bitu ACK.

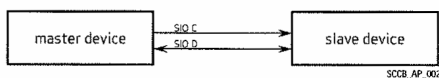
2.2.1 SCCB sběrnice

Společnost OmniVision Technologies, Inc. definovala a rozšířila 3-vodičovou sběrnici, určenou pro ovládání většiny funkcí snímačů OmniVision. U obvodů s omezeným počtem vývodů funguje v modifikovaném 2-vodičovém sériovém režimu. Snímače fungují pouze jako zařízení slave a spolupracující zařízení musí být master. Jedno zařízení master může ovládat více jednotek slave.



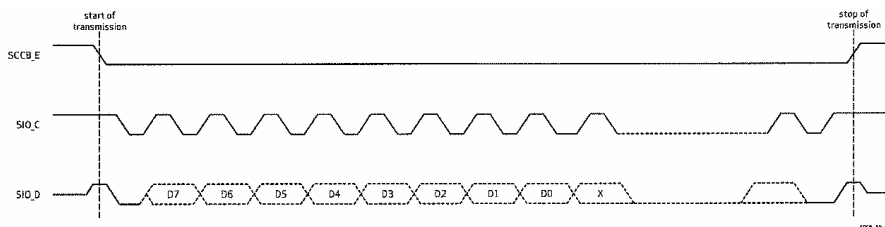
Obr. 2.5: Blokové schéma SCCB [11].

Modifikovaná 2-vodičová verze dokáže pracovat jen s jedním zařízením slave. Používá se především u snímačů s malým počtem vývodů.



Obr. 2.6: Blokové schéma 2-vodičového SCCB [11].

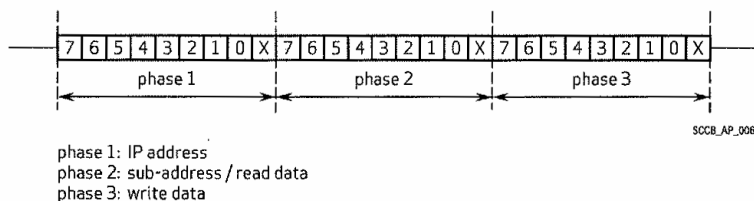
Jak je patrné z obrázků tak sběrnice pracuje se třemi signály. SCCB_E je jednosměrný řídicí signál aktivní v low, který musí ovládat zařízení master. Signál v logické 1 signalizuje nečinnost sběrnice. SIO_C je jednosměrný signál aktivní v high, který musí ovládat master. Signalizuje každý přenášený bit. SIO_D je obousměrný datový signál.



Obr. 2.7: Časový diagram SCCB [11].

Začátek přenosu dat je indikován přechodem signálu SCCB_E z high do low. Master však nejdříve vybudí signál SIO_D do logické 1. Tím je zabráněno neznámému stavu sběrnice. Konec přenosu je indikován přechodem signálu SCCB_E z low do high.

Sběrnice umožňuje sériovou přenosovou rychlost až 400kb/s. Fáze přenosu obsahuje celkem 9 bitů. Jsou složeny z 8-bitové posloupnosti datového přenosu, za kterou následuje devátý bit. Devátý bit je NA bit, který závisí na tom zda je přenos dat zápis nebo čtení. Maximální počet fází přenosu jsou tři.



Obr. 2.8: Přenosová fáze [11].

3-fázový cyklus zápisového přenosu je úplný zápisový cyklus. Za 2-fázovým cyklem zápisového přenosu následuje 2-fázový cyklus čtecího přenosu. Před 2-fázovým cyklem čtecího přenosu musí být vložen 3-fázový nebo 2-fázový cyklus zápisového přenosu. Doba přenosového cyklu jednoho bitu je 10μs.

3 SERVOMECHANISMUS

Modelářský servomechanismus (zkráceně servo) je levná a dostupná polohovací jednotka určená pro ovládání RC (rádiem řízených) modelů. Díky nízké ceně, snadnému ovládání, spolehlivé regulaci polohy natočení a velkému krouticímu momentu se modelářská serva hojně používají v robotice. Používají se k ovládání různých mechanických částí. Velké množství servomechanismů (18, 24 i více) najdeme na různých kráječících mobilních platformách (hexapod robot) nebo humanoidech.

Ve svém projektu jsem modelářská serva použil k realizaci panoramatického otáčení kamery. Předpokládám také rozšíření o naklápění kamery ve svislé ose. Polohování kamerového snímače v jedné nebo dvou osách je velmi užitečné při hledání zájmového objektu a jeho sledování. Nezastupitelnou úlohu má polohovací hlava při pořizování mapy okolí robota. Při navigaci lze z polohy panoramatické hlavy vypočítat směr dalšího pohybu mobilního podvozku.

3.1 Konstrukce a funkce serva

Servomechanismus se skládá ze stejnosměrného motorku, převodovky s výstupním hřídelem zpraženým na zpětnovazební potenciometr a obvodu elektronického řízení. Potenciometr slouží jako senzor úhlu natočení pro regulační obvod polohy. Mechanické uspořádání modelářského serva je patrné z obrázku 3.1.



Obr. 3.1: Řez serva od firmy Hitec [8].

Funkci a zapojení elektronického obvodu modelářského serva lze vysvětlit pomocí blokového schématu na obrázku 3.2. Elektronický obvod servomechanismu tvoří dvě části. Servo se ovládá PWM signálem s opakovací frekvencí 50 Hz a šířkou kladného impulsu 1 – 2 ms. Výkonová část, realizovaná zpravidla jako H – můstek, napájí motor. Napájení motoru musí mít dostatečný výkon pro otáčení v obou směrech. Součástí výkonového obvodu bývá také ochrana proti nadměrnému proudu při mechanickém zablokování motoru. Druhou částí je regulátor založený na porovnávání šířky řídicího impulsu a dobou kyvu vnitřního monostabilního klopného obvodu (MKO).

Elektronický regulační obvod zpracovává vstupní PWM signál. Ten se rozdělí do dvou větví. Náběžná hrana PWM signálu spustí monostabilní klopný obvod. Střední doba kyvu MKO je 1,5 ms. Potenciometr, který je mechanicky spojený s výstupním hřídelem serva dokáže změnit dobu kyvu vnitřního MKO v intervalu od 0,5 do 2,5 ms.

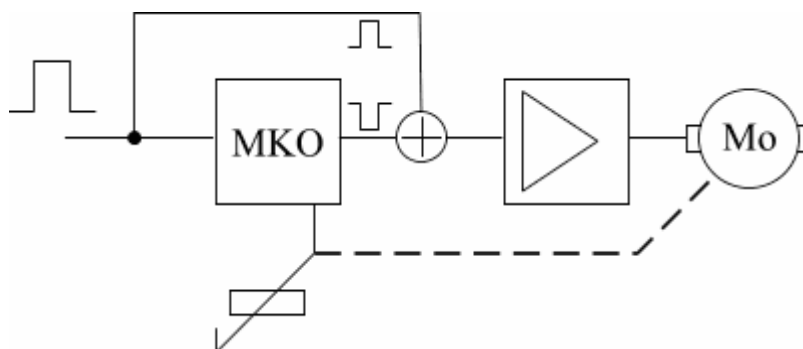
Výstupní signál MKO je vůči řídicímu impulsu negovaný. V součtovém členu se porovná šířka PWM signálu s dobou kyvu MKO. Výsledkem porovnání je pravoúhlý kladný nebo záporný impuls s dobou trvání:

$$T_M = |T_{PWM} - T_{MKO}|$$

Polarita signálu za rozdílovým členem určuje směr otáčení motoru, doba trvání impulsu T_M určuje velikost elektrické energie dodávané do motoru a tím i jeho rychlost. Celý regulační proces se opakuje podle frekvence řídicího signálu tj. 50x za sekundu.

Výstupní hřídel současně otáčí s běžcem potenciometru, který mění dobu kyvu MKO. Elektromechanický systém potenciometru je sestaven tak, aby vytvářel zápornou zpětnou vazbu. Při správné funkci se motor otáčí tak aby eliminoval rozdíl mezi požadovanou a skutečnou polohou výstupního hřídele. Současná změna hodnoty potenciometru ovlivňuje dobu kyvu MKO. Pokud se doba kyvu MKO rovná šířce vstupního PWM signálu, motor se zastaví.

Regulační obvod stabilizuje také mechanické vlivy, které působí na výstupní hřídel servomechanismu. Pokus se násilně pootočí výstupní hřídel, je šířka vstupního PWM signálu konstantní a natočení potenciometru vyvolá odchylku, která začne otáčet motorem. Výsledkem je návrt hřídele do původní polohy určené vstupním signálem. Tato schopnost se s výhodou používá pro kompenzaci vnějších sil, které působí na mechanické části systému.

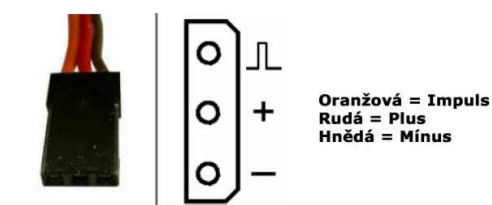


Obr. 3.2: Řídicí obvod serva.

Pro RC aplikace se prakticky využívá rozsah šířky vstupního impulsu 1,0 až 2,0 ms při natočení výstupního hřídele o 90 stupňů. Serva dovedou zpracovat řídicí signál s šířkou impulsu od 1,25 ms do 1,75 ms u kvalitních serv dokonce od 0,5 ms do 2,5 ms při natočení výstupního hřídele od 0° do 180°. Natočení serva je plynulé, klasická serva mají minimální otočení o 2° lepší serva o 1°. Pro ideální vstupní signál je frekvence opakování pulzu 50 Hz. Pásmo necitlivosti šířky vstupního signálu je přibližně 5 μ s, kvalitní serva mohou mít pásmo necitlivosti užší. Řízení serv v rozšířeném osahu však možné jen u servomechanismů některých výrobců a je třeba ho na konkrétním servu opatrně vyzkoušet, aby se nepoškodilo opakovaným najížděním na koncové dorazy převodovky. Serva je možné také upravit pro trvalé otáčení; PWM signálem se pak neřídí poloha výstupního hřídele, ale rychlost a směr jeho otáčení. Takto upravená serva se používají k pohonu kol mobilních robotů. Serva lze koupit v modelářských prodejnách. K dispozici jsou v různých provedeních a velikostech. Existují také varianty s rozsahem natočení 360°. Servomechanismy s větším kroutícím momentem mají také větší odběr proudu - proto je nutné dostatečně dimenzovat napájecí zdroj.

3.2 Řízení serva

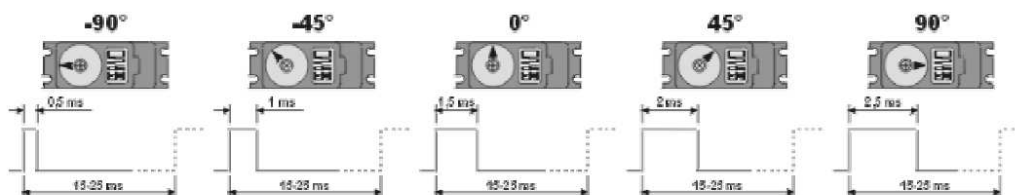
Z každého modelářského serva vede trojvodičový kablík a jednotlivé žíly jsou barevně odlišeny. Dva vodiče slouží k napájení a třetí k ovládání. Na ovládací vodič přichází v pravidelném intervalu 20ms (50Hz) impuls, jehož šířka určuje velikost natočení výstupního hřídele. Kabel je zakončen konektorem třemi dutinkami s roztečí 2,54 mm. Konektor lze nasunout na kontaktní lištu se třemi vývody. Barevné označení vodičů může být různé střední (červený) vodič je kladné napájení serva. Napájecí napětí se běžně může pohybovat v rozsahu 4,5 až 6 V. V robotice se obvykle používá napětí 5 V. Proud serva se značně liší podle zatížení výstupního hřídele. Pro standardní, mini a mikro serva by měl vyhovovat napájecí zdroj, který je schopen dodávat proud 1A. Pro velké servomechanismy musíme počítat s proudy až 5 A. Jeden krajní vodič má hnědou nebo černou barvu a slouží k přivedení záporného pólu napájecího napětí (GND). Na druhý krajní vodič (oranžový, žlutý nebo bílý) se přivádí ovládací signál. Proudový odběr je minimální a řídicí signál může být dodáván přímo z I/O portu mikroprocesoru.



Obr. 3.3: Konektor serva [8].

Při ovládání serva je nejdříve zapotřebí uložit konstanty pro hranice krajních poloh serv. Hranici zjistíme tak, že nastavíme výchylku do maximální polohy. Tuto polohu poznáme podle neměnicí se polohy a rostoucím hlukem zatíženého motoru. V této poloze také značně roste proud, a proto by natočení do této polohy nemělo trvat dlouho.

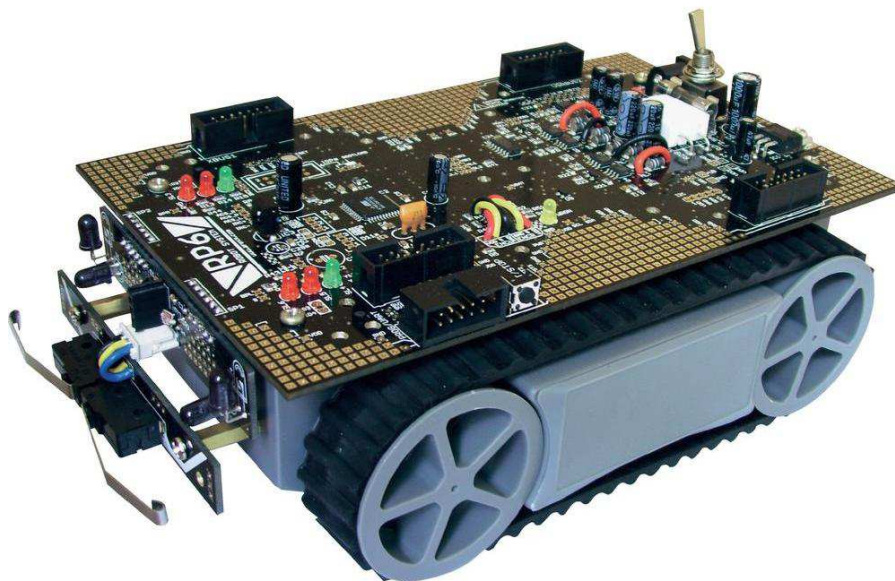
K řízení serva pomocí mikroprocesoru se principiálně využívá pulzní šířková modulace. Změnou šířky impulsu měníme natočení serva od -90° do $+90^\circ$, což dovoluje mechanický rozsah většiny serv. Šířky pulzů pro běžná serva jsou 1 až 2 ms. Střední poloze odpovídá šířka impulsu 1,5ms. Tyto hodnoty se však mohou u různých typů serv lišit. Při nastavování hodnot je zapotřebí pracovat opatrně, aby nedošlo ke zničení.



Obr. 3.4: Šířky impulsů pro danou polohu serva [9].

4 ROBOTICKÝ PODVOZEK RP6

Pro praktickou aplikaci kamerového modulu byla vybrána robotická platforma RP6 vyráběná holandskou firmou AREXX Engineering. V České Republice výrobek nabízí internetový obchod conrad.cz.



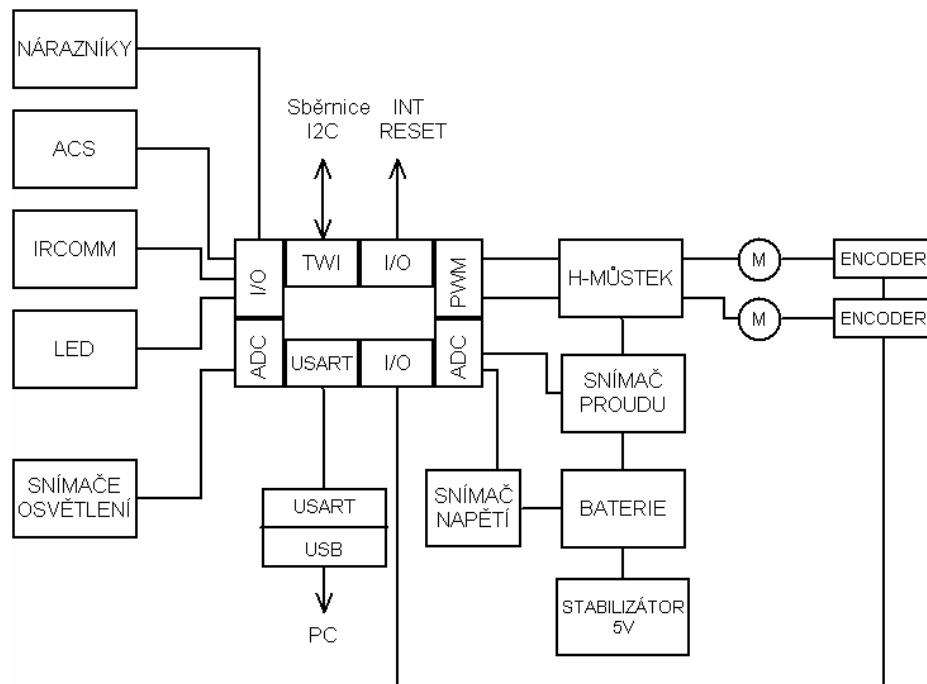
Obr. 4.1: Robotický podvozek RP6 [13].

Pro zajištění robustnosti, rozšiřitelnosti a možnosti řešení více cílů, využíváme při návrhu robotů metodiku založenou na různých architekturách. U mobilních robotů používáme zejména dvě architektury: Sence-plan-act a Subsumpční. Sence-plan-act architektura, jak již název napovídá, je založena na třech krocích: snímat (sence) okolí, naplánovat akce (plan) a nakonec provést naplánovanou činnost. Na této architektuře byl postaven první univerzální robot Shakey, vyvinutý SRI's Artificial Intelligence Center (dříve Stanford Research Institute), který měl podstatný vliv na vývoj umělé inteligence a robotiky.

Subsumpční architektura byla poprvé představena Rodneyem Brocksem v roce 1986. Jedná se o vrstvenou architekturu, kde jednotlivé vrstvy reprezentují určité chování. Vrstvy mají vzestupnou prioritu. Například robot může mít ve spodní vrstvě zabudované nouzové zastavení před překážkou a vyšší vrstva řeší, jak danou překážku objede. Jednotlivé vrstvy mohou pracovat na určitých cílech souběžně. Obsluha senzorů nemusí být součástí všech vrstev, vyšší vrstvy zpracovávají výsledky ze senzorů, zatímco nižší vrstvy dál senzory používají. Tím je zajištěn neustálý běh procesorů a není nutné čekat na zpracování měřených dat. Mobilní robot s touto architekturou je kdykoliv rozšiřitelný přidáním dalších vrstev nebo senzorů.

Robotická platforma RP6 je ideální pro realizaci Subsumpční architektury mobilního robotu. Firma AREXX nabízí vlastní mobilní podvozek vybavený

mikroprocesorovým řízením, diferenciálním pásovým pohonem, napájecí baterii a senzorickým systémem pro monitorování napájení, odometrii a antikolizní systém.



Obr. 4.2: Blokové schéma robotického podvozku.

Dále je k dispozici univerzální deska pro rozšiřování základního modulu, řídicí s mikroprocesorem ATMEL AVR ATmega32 a senzory.

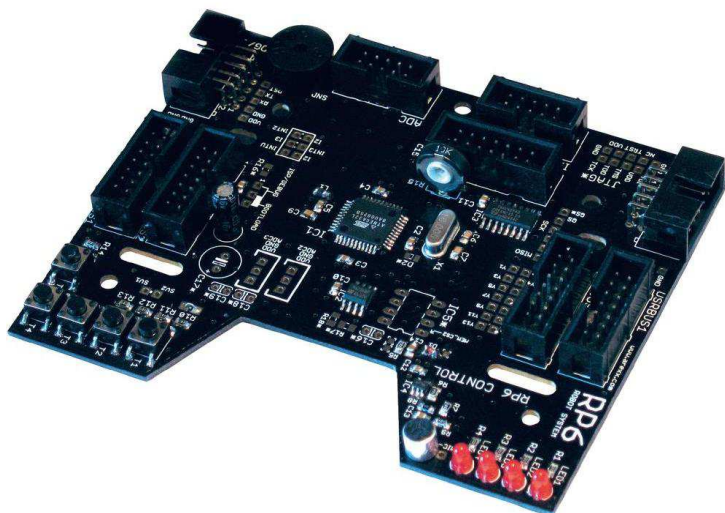
Firma AREXX poskytuje podrobný návod, který má v některých kapitolách učebnicový charakter a poskytuje cenné informace. Součástí balení je CD se softwarem, který tvoří knihovna funkcí a ukázkové příklady. Veškerý software je sestaven v programovacím jazyce C. Kód knihovny i příkladů je sestaven pomocí překladače GNU GCC WinAVR. Software je k dispozici ve zdrojových textech i kompilovaném tvaru Intel HEX. Mikroprocesory všech vrstev je možné programovat přes ISP nebo BOOTLOADER. V praxi se používá BOOTLOADER sériového rozhraní UART. K dispozici je PC aplikace pro nahrávání programu do paměti FLASH. Součástí této aplikace je také terminál.

Software je vhodné sestavovat v AVR studio, který je volně k dispozici na webových stránkách výrobce mikroprocesorů ATMEL.

4.1 Řídicí modul RP6 CONTROL M32

Mobilní robot vytvořený pro tento projekt je navržen podle subsumptické architektury. Robotický podvozek obsahuje všechny HW a SW prostředky pro napájení a pohon robota, senzoriku pro regulaci rychlosti, odometrii a antikolizní systém. V mikroprocesoru řízení podvozku je implementován program I2C SLAVE.

Řídící deska M32 musí plnit funkci I2C zařízení typu MASTER pro ovládání podvozku. Kromě této nezbytné funkce je v řídicí jednotce implementován program pro řízení robotu na základě informací získané z kamerového snímače. Více procesorová koncepce robotu poskytuje více systémových prostředků (paměť RAM, I/O vývody, AD převodník, USART a další periferie).



Obr. 4.3: Řídící deska M32 [13].

Vlastnosti a technické parametry:

- výkonný 8-bitový mikroprocesor Atmel ATmega32
- externí paměť 32kB SPI EEPROM
- rozšiřující konektor sběrnice I2C
- mikrofonní snímač
- piezo bzučák
- 4 stavové LED
- 5 ovládacích tlačítek
- port pro připojení LCD displeje
- 14 volně dostupných I/O portů
- 3 vstupy externího přerušení
- UART rozhraní

5 NÁVRH KAMEROVÉHO ZAŘÍZENÍ

Součástí projektu je návrh řešení kamerového systému pro zpracování obrazu. Modul kamerového snímače by měl být univerzální systém pro možnost dalších libovolných úprav a jednoduché změny konfigurace pro aplikace zpracování obrazu, strojového vidění nebo navádění mobilních robotických systémů. Systém má být jednoduchý levný a reprodukovatelný pomocí běžných vývojových a programovacích nástrojů. Při návrhu můžeme diskutovat o dvou možnostech provozních režimů.

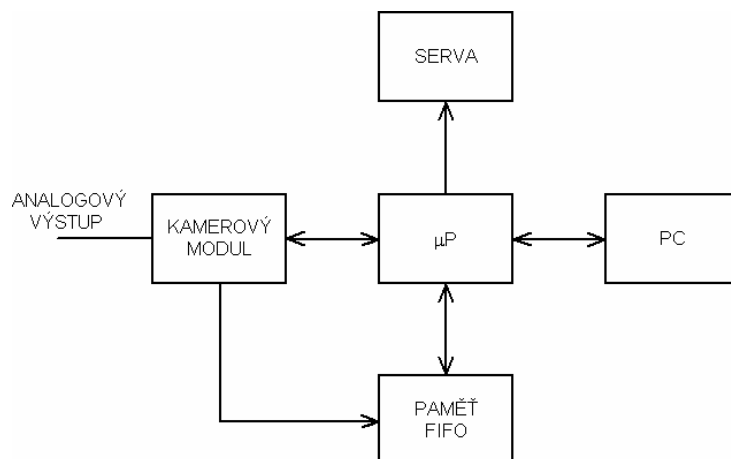
Přímé připojení kamery se zdá být použitelný v systému, který je určen pro použití "on-fly" společného sběru dat a zpracování obrazu, který běží s nejvyšší prioritou a v takovém systému obvykle není potřeba žádný jiný hardware pro operace s daty. Rozlišení obrazu je v tomto případě omezeno rozlišením použitého kamerového snímače a výpočetní schopností mikroprocesoru společně s požadovanou obnovovací frekvencí snímání obrazu. V případě souběžného provádění několika aplikací, běžících na stejné úrovni priority jako zpracování obrazu a sběr dat, může dojít k nežádoucímu chování při přidělení obsluhy přerušení pro jiné aplikace. Což znamená, že některé hodnoty obrazových bodů mohou chybět a pak budou získaná data nepřesná, přesto můžeme více či méně říci, že získaná data budou v pořádku. Takže v případě použití více aplikací, musí být použity různé konstrukce systému.

Nepřímé připojení s vyrovnávací pamětí FIFO na rozdíl od přímého spojení, vede datová cesta z kamery (nastavené do režimu master) do paměťového úložiště vhodného pro kamerové aplikace – použití vyrovnávací paměti FIFO. Také časování obrazových bodů a signál horizontální synchronizace z kamery, společně se signálem povolení zápisu z mikroprocesoru mohou být použity pro nastavení zápisu dat do FIFO pro úspěšné ukládání platných dat do paměti (pro řízení FIFO je nutná externí logika sestavená z hradla AND). Mikroprocesor přistupuje k paměti a vyžaduje nejstarší data se snímkovou rychlostí podle schopnosti zpracování dat. PIC32 (nastavený do režimu master) může používat rozhraní přímého přístupu k paměti (DMA) pro přesměrování dat z kamery do integrované paměti SRAM a pak také k vyzvedávání uložených dat z paměti SRAM a přenosu do mikroprocesoru na zpracování. DMA běží na stejné rychlosti jako mikroprocesor a je navržen tak, aby ukládání nezatěžovalo výpočetní kapacitu a zvýšil se celkový výkon, protože při všech přenosech DMA zůstává mikroprocesor nečinný nebo můžete provádět další výpočty, aniž by došlo k přerušení (DMA pracuje nezávisle).

Při návrhu bylo zpracování problematiky rozděleno do dvou kroků. Nejprve bylo navrženo polohovací rameno s horizontálním natáčením a poté i s vertikálním. Po odladění byl navržen systém pro řízení robotického podvozku pomocí barevného objektu.

5.1 Polohovací rameno

Vlastní návrh polohovacího ramene s kamerovým modulem lze popsat níže uvedeným blokovým schématem na obrázku 5.1.

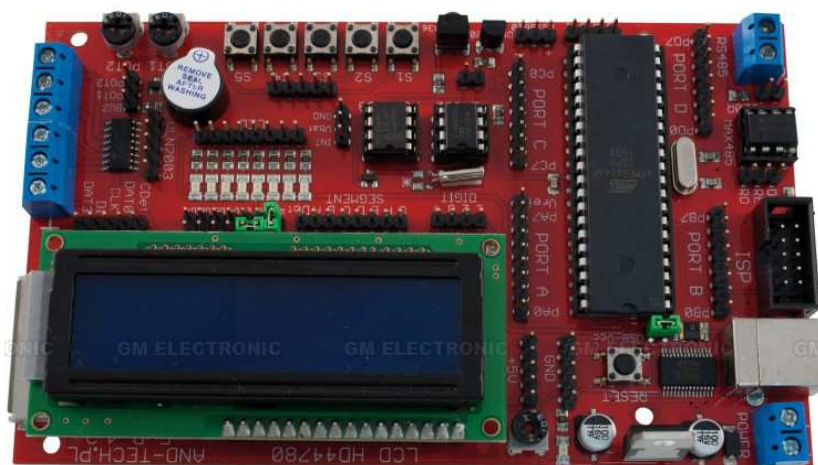


Obr. 5.1: Blokové schéma ramena.

Propojení kamery s mikroprocesorem je pomocí 8-bitového portu pro čtení dat obrazu. Komunikace mezi těmito prvky je zajištěna pomocí protokolu I2C. Komunikace s PC může být zajištěna pomocí komunikace UART. Servo je připojeno na vývod čítače/časovače procesoru. Datová cesta vede z kamery (nastavené do režimu master) do paměťového úložiště vhodného pro kamerové aplikace – použití vyrovnávací paměti FIFO. Mikroprocesor přistupuje k paměti a vyžaduje nejstarší data se snímkovou rychlostí podle schopnosti zpracování dat.

5.1.1 Vývojový kit

Pro lepší odladění komunikace mikroprocesoru s modulem kamery jsem využil vývojový kit AVR EvB 4.3 od firmy AND-TECH. Tento kit je vybaven mikroprocesorem ATmega32 (lze nahradit ATmega16 nebo ATmega644P) s vyvedenými porty a dalšími potřebnými prvky.

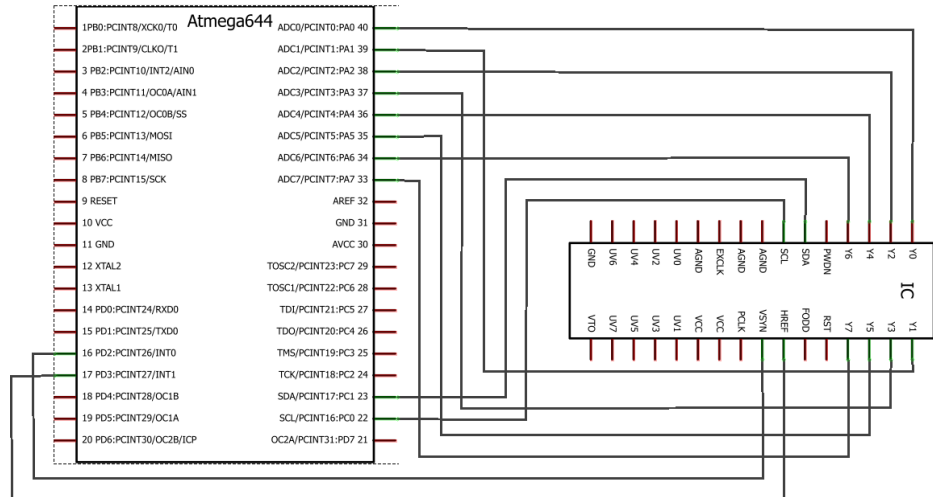


Obr. 5.2: Vývojový kit AVR EvB 4.3 [14].

Vývojová deska lze připojit k PC pomocí rozhraní USB (RS232). Nahrávání programu lze pomocí konektoru ISP.

5.1.2 Kamerový modul C3088

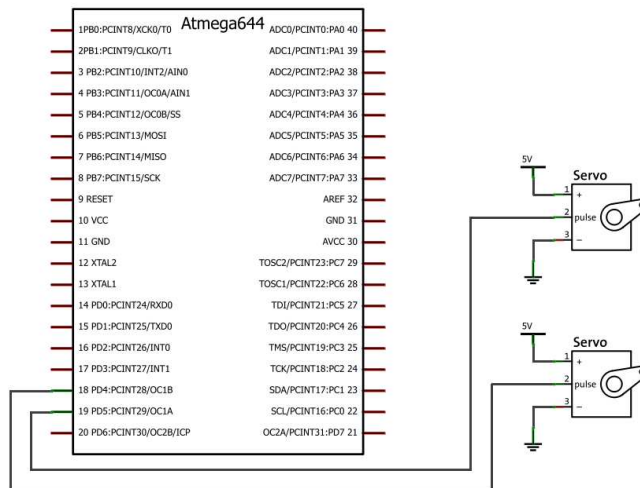
Zapojení kamery využívá 12 pinů mikroprocesoru. Sběrnice jasové složky Y je připojena na port A (PA0 – PA7). Linky SDA a SCL pro komunikaci jsou připojeny na piny PC1 a PC0. Synchronní vývody kamery jsou připojeny na piny PD2, PD3. Vývody 20 a 22 jsou napojeny přímo na napájení 5V.



Obr. 5.3: Zapojení modulu C3088.

5.1.3 Ovládání serva

Serva jsou připojena na vývody PD4, PD5, protože využívají komparátor čítače/časovače TIMER1. Serva jsou dále připojena na napájení a zem.

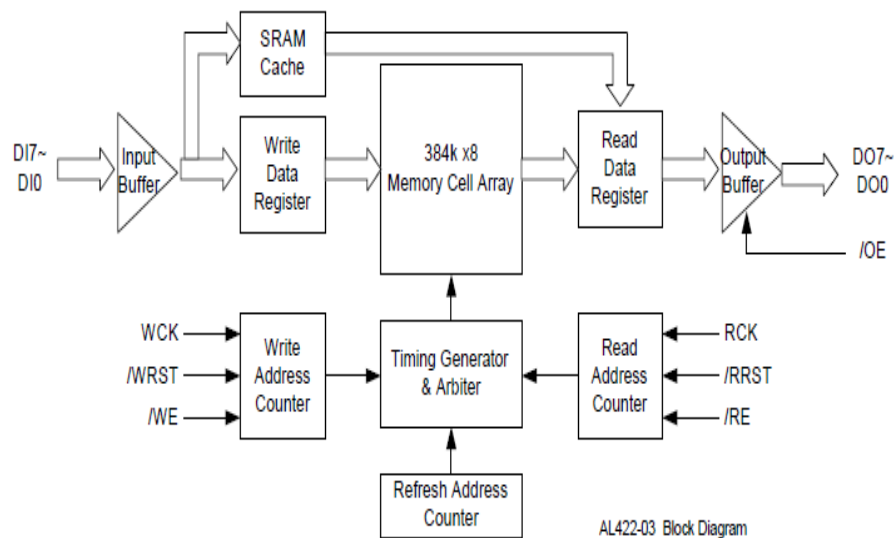


Obr. 5.4: Zapojení serva.

5.1.4 Paměť FIFO

V návrhu byl vybrána externí FIFO obvod AL422B od firmy AVERLOGIC. AL422B obsahuje 3M-bit DRAM, která je konfigurovaná jako 393 216 slov x 8 bitová vyrovnávací paměť FIFO. Rozhraní je uživatelsky velmi příjemné, protože všechny komplikované operace paměti DRAM obsahuje vnitřní řadič.

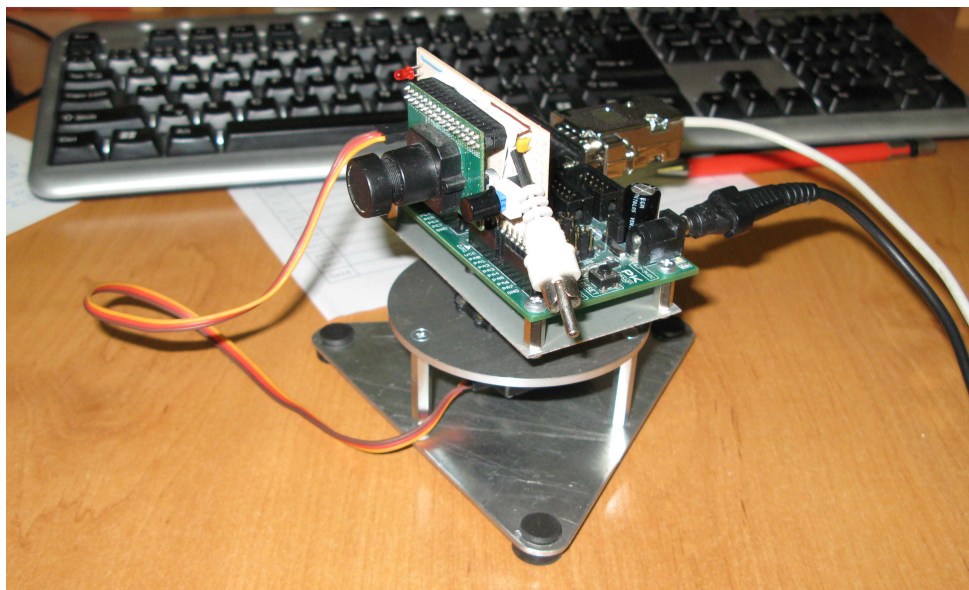
V obvodu kamerového modulu se paměť FIFO používá pro zachycení jednoho snímku z kamerového snímače. Napájení obvodu je 5V se zapojením podle doporučení v katalogovém listu. Refresh dynamické paměti zajišťuje trvalé připojení signálu PCLK z kamerového snímače. Zachycením pouze jednoho snímku je zajištěno vhodným zapojením hardwaru a softwarovým spouštěním ukládání dat do FIFO. Hardware ovládání zápisu do FIFO tvoří hradlo NAND, Na jeden vstup hradla je přiveden synchronizační signál kamerového snímače HREF. Druhý vstup hradla je ovládán signálem WEE, který generuje spouštěcí funkce software implementovaného v mikroprocesoru modulu.



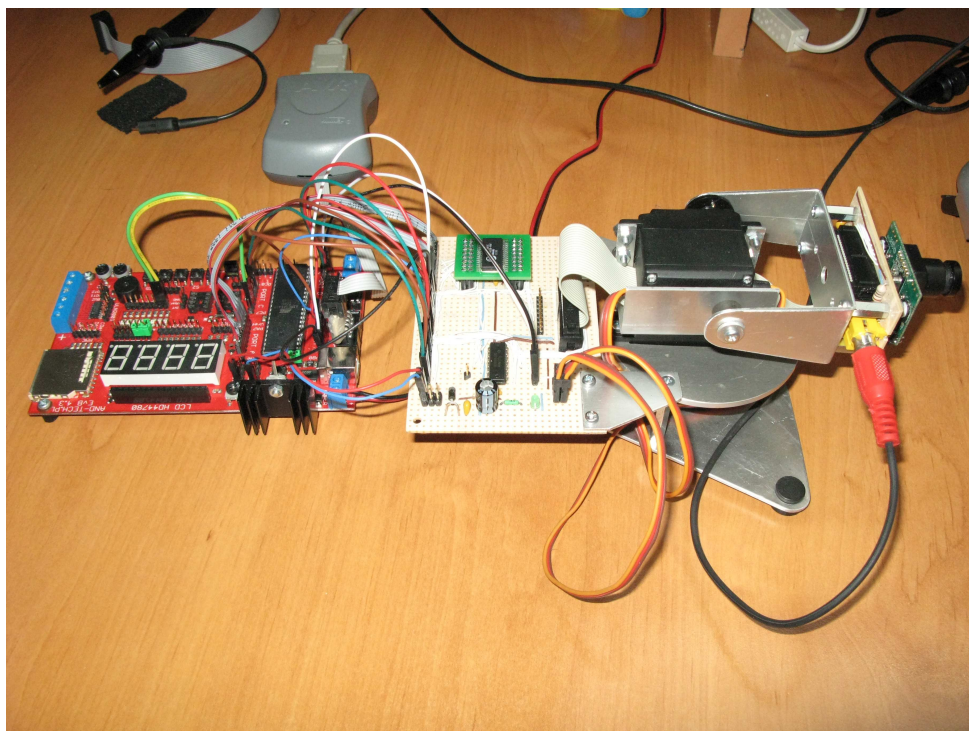
Obr. 5.5: Blokové schéma paměti FIFO [16].

5.1.5 Konstrukce

Zkušební model jak již bylo zmíněno je realizován pomocí vývojového kitu. Tato deska je připojena k řídicí desce s vyvedenými potřebnými piny. Deska je připevněna k servomechanismu, který je uchycen do vyrobeného stojanu s hliníkového plechu. Jeho díly a rozměry jsou zaznamenány v příloze. Kamerový modul je připojen přes propojovací DPS k vertikálnímu servu.



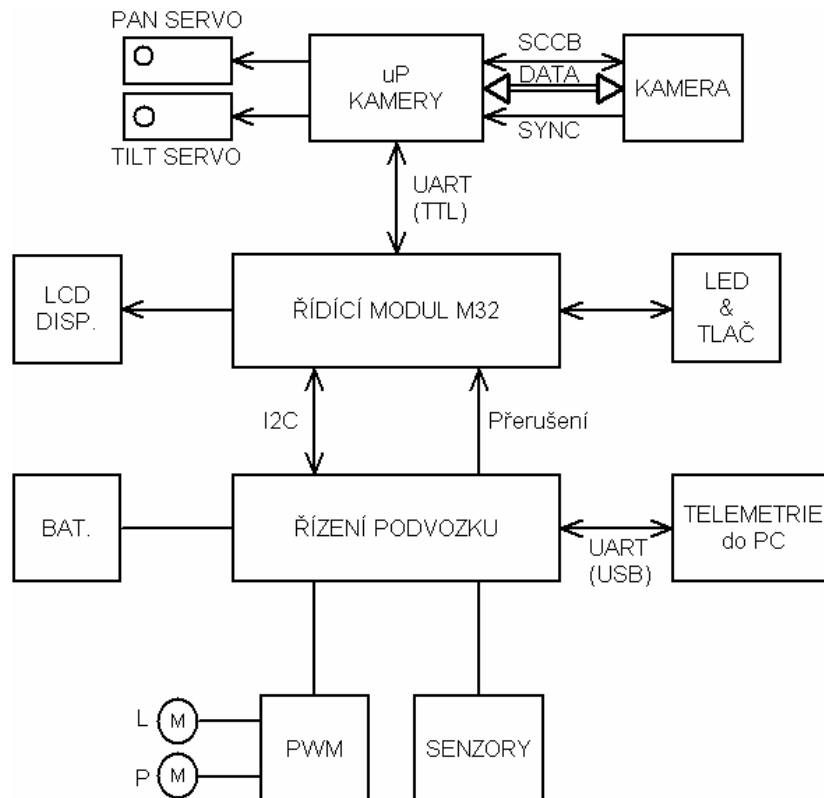
Obr. 5.6: Sestavený vývojový model s horizontálním natáčením.



Obr. 5.7: Sestavený vývojový model s horizontálním a vertikálním natáčením.

5.2 Řízení robotického podvozku

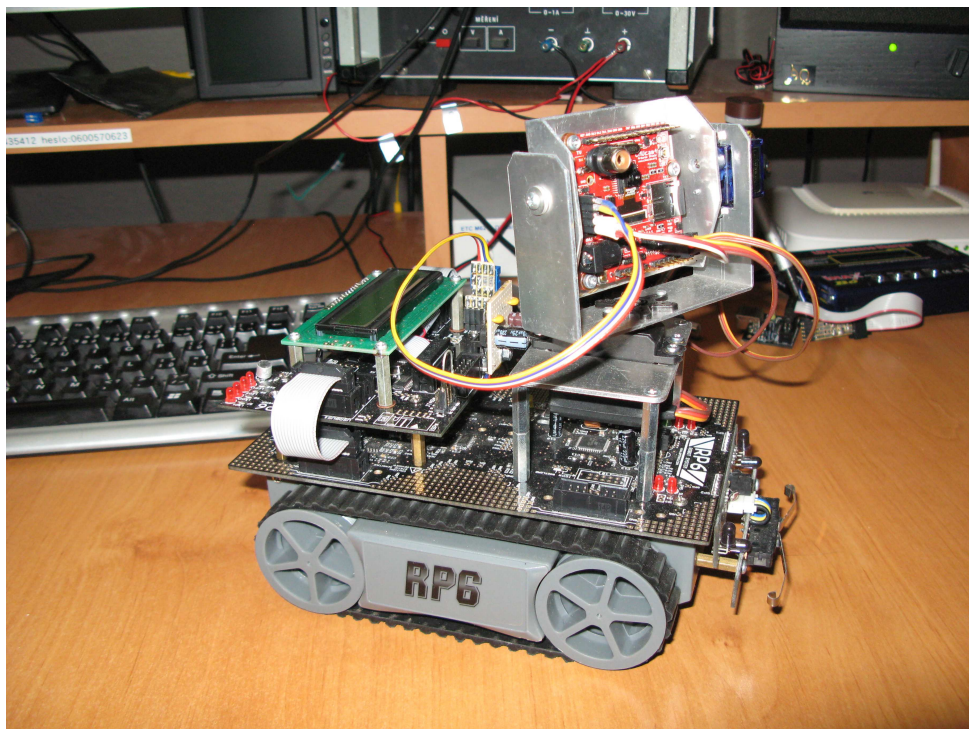
Mobilní robot realizovaný pro aplikaci vizuální sensoriky má následující uspořádání viz obrázek 5.6.



Obr. 5.8: Blokové schéma návrhu robotnického podvozku s kamerou.

5.2.1 Konstrukce

Modul M32 má mechanickou konstrukci, která se snadno umístí na robotický podvozek. Propojení elektrických signálů zajišťují dva ploché kabely se 16 vodiči. Podvozek byl doplněn konstrukcí pro uchycení horizontálního a vertikálního serva. Dále byl vyroben držák pro kamerový modul. Modul M32 byl doplněn znakovým LCD displejem. Pro zprovoznění chování podvozku byl využit modul CMUcam4, který je dostupný ve vybraných internetových obchodech. Kamerový modul je napájen stabilizovaným napětím 5V. Napájení servomechanismu se muselo provést samostatným zdrojem 5V odvozeným přímo z baterie. Napájení panoramatického a naklápěcího serva přímo z modulu způsobovalo RESET a rušení celého řídicího systému. Zdroj napjení serv je mechanicky spojeno s modulem M32. Datové připojení kamerového modulu je realizováno rozhraním UART na úrovni TTL. Přizpůsobení 5V úrovní signálů M32 a 3,3V úrovní signálů kamerového modulu zajišťuje odporová síť.



Obr. 5.9: Sestavený robotický podvozek s kamerou.

6 SOFTWARE

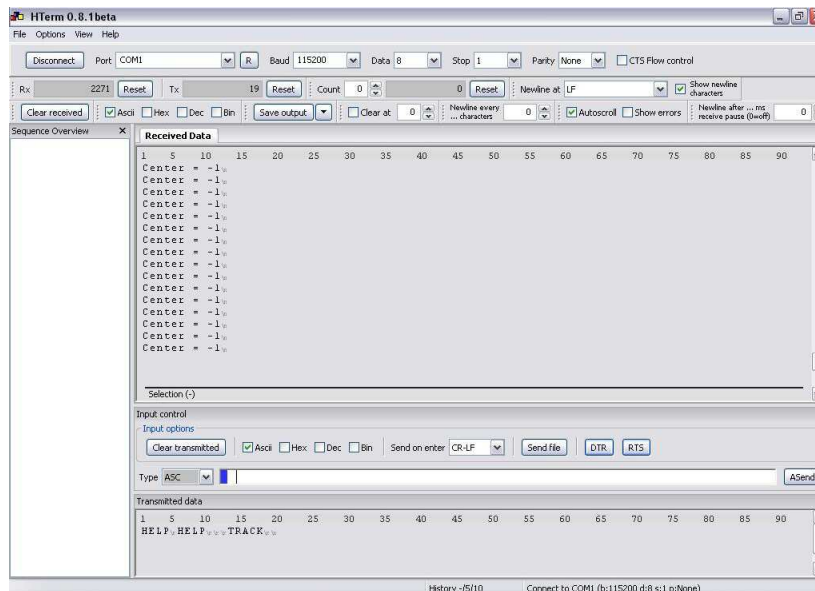
K sestavení kódu byl použit překladač jazyku C pro mikroprocesory. Zvolil jsem volně dostupný překladač WIN AVR. Jedná se o řádkový překladač jazyka C bez omezení velikosti zdrojového i cílového kódu distribuovaný jako GCC produkt. Překladač je možné volně stáhnout z webových stránek výrobce ATMEL. Prostředí umožňuje pohodlné psaní, překlad, sestavování a ladění programů. Vedle vestavěného softwarového simulátoru umožňuje přímé ovládání ISP programátoru a dalších ladících prostředků.

Programové vybavení jednotky řeší dvě úlohy, které můžeme shrnout do následujícího přehledu:

- komunikace a ovládání kamery
- ovládání servomechanismu

6.1 Sériová komunikace s PC

Pro komunikaci s PC a sledování výstupních dat modulu jsem využil program HTerm. Program byl využit při zprovoznění polohovacího ramene. Pomocí programu lze testovat funkci serva a nastavovat jeho polohu. Dále byla programem ukládána data o snímaném obrazu do vlastního souboru, aby mohly být informace dále zpracovány.

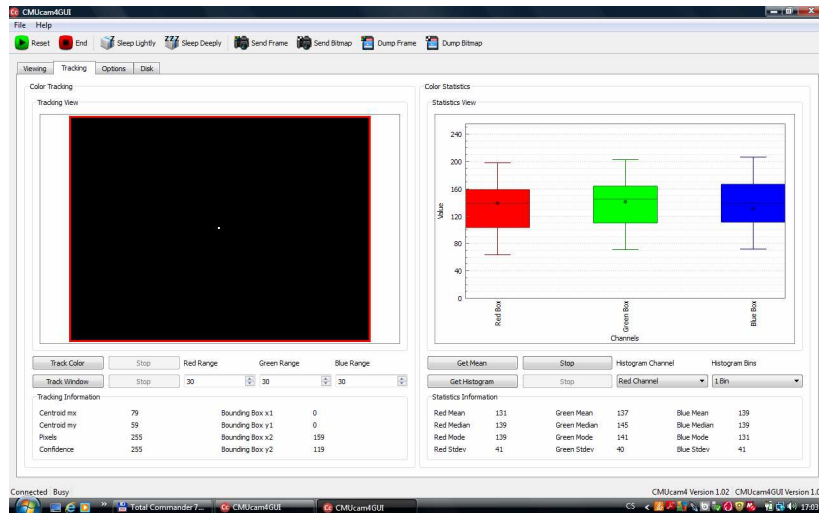


Obr. 6.1: Program HTerm.

Příkazy využity pro ovládání polohovacího ramene

- SERVO arg1 – ovládání serva, arg1 je hodnota natočení od -900 do 900
- PHOTO – příkaz zaznamená data pro získ obrázku ve formátu bmp
- TRACK – příkaz pro sledování bílého objektu

Pro zprovoznění modulu CMUcam4 bylo využito programu CMUGUI. Tento software je volně stažitelný na internetu. Jeho předností je přesné nastavení sledované barvy, sejmutí obrázku a další různé nastavení kamery.



Obr. 6.2: Program CMUGUI.

Příkazy využity pro ovládání robotického podvozku. Další příkazy a jejich vysvětlení lze nalézt v command listu CMUcam4, který je přiložen na CD.

- **GV – Get Version** (Získání verze) – <GV ‘\r’>
Příkaz tiskne řetězec s verzí kamerového modulu.
- **GS – Get Servo Position** (Získání polohy serva) - <GS[servo–Address]‘\r’>
Příkaz získá polohu pan nebo tilt serva.
- **TC – Track Color** (Trasování barvy) - <TC [červená min – Integer] [červená max – Integer] [zelená min – Integer] [zelená max – Integer] [modrá min – Integer] [modrá max – Integer] ‘\r’ >
Příkaz vrací trasovací data o obrazu.

6.2 Zápís do paměti

Mechanismus spouštěcí funkce využívá synchronizační signály VSYNC a HREF kamerového snímače OV6620. Signál VSYNC je přiveden na vstup externího přerušování mikroprocesoru. Tento vstup se ve spouštěcí funkci využívá alternativně ke spuštění události přerušování a jako univerzální číslicový vstup k testování úrovně signálu VSYNC. Algoritmus spouštěcí funkce lze popsat následujícím postupem:

1. na začátku funkce se vstup signálu VSYNC nastaví jako univerzální vstupní port,
2. v cyklu WHILE se čeká na přechod signálu VSYNC z vysoké úrovně do nízké – ukončení snímku,
3. pokud se signál VSYNC nachází v nízké úrovni dochází k expozici nového snímku,
4. další cyklus WHILE čeká na náběžnou hranu signálu VSYNC,

5. v tomto místě je možné pomocí monitorování signálu HREF potlačit první řádky snímku,
6. generuje se vysoká úroveň výstupního signálu WEE, který otevře hradlo NAND ovládání zápisu,
7. zároveň se povolí externí přerušení a nastaví spuštění při sestupné hraně,
8. obsluha události externího přerušení způsobí přechod signálu WEE do nízké úrovně,
9. nízká úroveň WEE ukončí zápis do FIFO,
10. na konci obsluhy události se zakáže externí přerušení a nastaví vstupní režim vývodu.

6.3 Software podvozku

Software řízení robotického podvozku RP6-BASE je sestaven na základě Subsumpční architektury. Nejvyšší vrstvu tvoří funkce pro realizaci pulsní šířkovou modulaci pro signály s H-můstky pohonného systému. Analýzu měření napětí baterie, proudů tekoucích do motorů a intenzitu osvětlení. V této vrstvě se také zpracovávají číslicové signály snímačů polohy, nárazníků a ovládají infračervenou sensoriku a indikátory LED.

Střední vrstva řeší regulaci rychlosti otáčení odometrie a zpracování signálů sensorického systému. Prostřednictvím datových struktur dokáže tato vrstva řídit pohon podvozku a předávat výsledky snímačů. Patří sem také obsluha sériové komunikace UART s nadřazeným prostředkem (PC, bezdrátová telemetrie atd.).

Nejvyšší vrstva subsomptického softwaru realizuje I2C typu SLAVE. Fyzickou vrstvu rozhraní I2C využívá hardwarovou periférii TWI mikroprocesoru. Komunikace po sběrnici I2C je obsloužena v přerušení od TWI. Softwarová část I2C SLAVE tvoří kód umístěný v nekonečné smyčce. Klíčovou součástí softwaru jsou registry I2C zařízení typu SLAVE. Adresa I2C SLAVE je trvale nastavena na hodnotu 10.

Robotický podvozek RP6 s implementací I2C zařízením typu SLAVE poskytuje 30 registrů určených pouze ke čtení, ve kterých jsou aktuální hodnoty robotického podvozku. Všechny registry jsou 8-bitové. Pokud je hodnota parametru větší než 8 bitů, ukládá se do více registrů. I2C SLAVE předává informace o:

- provozním stavu řízení
- stav pohonu
- výkon levého a pravého motoru
- rychlost levého a pravého pásu
- ujetá vzdálenost levého a pravého pásu
- intenzitu levého a pravého snímače
- proud levého a pravého motoru
- napětí baterie
- stav LED

Ovládání se provádí přes příkazové registry, do kterých může zařízení MASTER zapisovat 8 bitové hodnoty. Implementován jeden příkazový registr a šest registrů je vyhrazeno pro parametry příkazu. Podobně jako jiná I2C zařízení typu SLAVE, generuje robotický podvozek signál přerušení přivedený do I2C zařízení typu SLAVE. Událost, která přerušení vyvolala, lze zjistit ve stavovém registru.

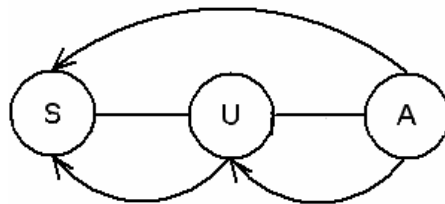
Kód programu I2C SLAVE pravidelně aktualizuje obsah registrů přístupných přes sběrnice I2C. Kontroluje a zpracovává příkazy vyslané I2C zařízením typu MASTER. Součástí programu je časový dozor (watchdog) zpracování přerušení v zařízení I2C MASTER. Pokud MASTER nereaguje na signál přerušení déle než 3s, pak se zastaví veškerá činnost podvozku. Tento časový dozor je nejlepší způsob ochrany neřízeného podvozku před poškozením (náraz do stěny, pád ze schodů atd.).

6.4 Software modulu RP6 CONTROL M32

Software je opět navržen podle subsumpční architektury. Nejnížší vrstva provádí obsluhu HW obvodů modulu (LED, tlačítka, LCD displej, mikrofon).

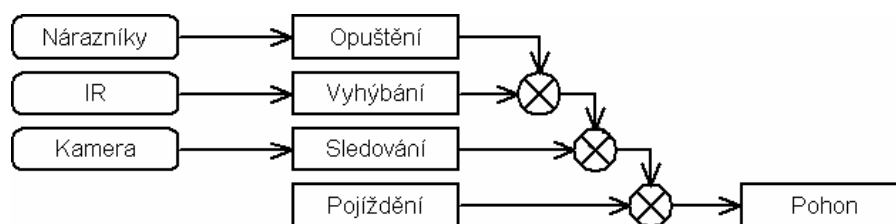
Střední vrstva řeší ovládání podvozku přes sběrnici I2C obsluhu přerušení a další výpočty nezbytné pro správnou funkci systému.

Nejvyšší vrstva implementovaného softwaru realizuje konečný automat autonomního ovládání robotické platformy a rozhraní s uživatelem. Konečný automat má tři základní stavy (klidový, interakce s uživatelem, autonomní).



Obr. 6.3: Konečný automat.

Autonomní chování musí rozhodovat o chování robota na základě informací senzoričského systému. Chování je celkem složité a musí mít určenou prioritu a další fáze chování, aby mohl optimálně řešit všechny možné situace. Schématicky můžeme autonomní chování vyjádřit podle obrázku 6.3.



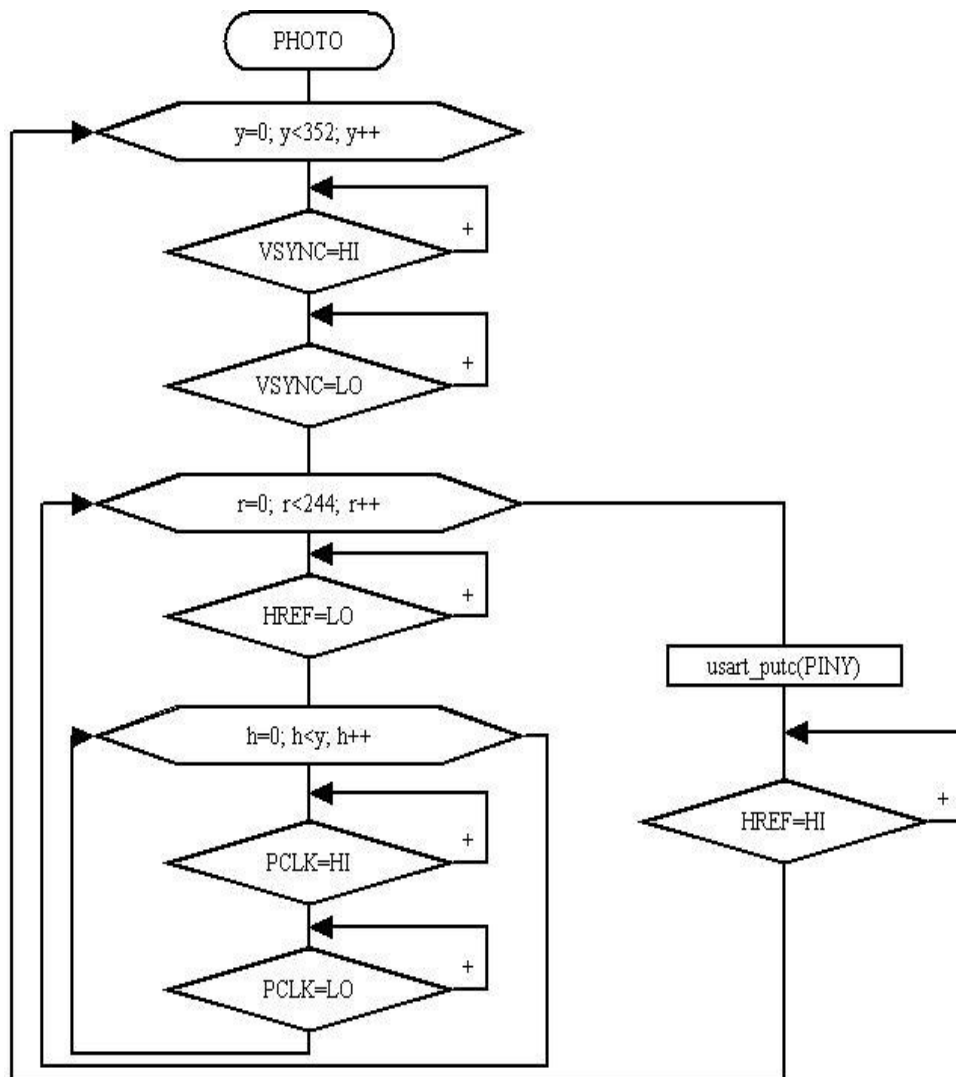
Obr. 6.4: Schématicky autonomní chování.

V programovacím jazyce se tento konečný automat vytvořil pomocí zdrojového kódu přiloženém v příloze C.

6.5 Vlastní programy pro μP

6.5.1 Funkce photo

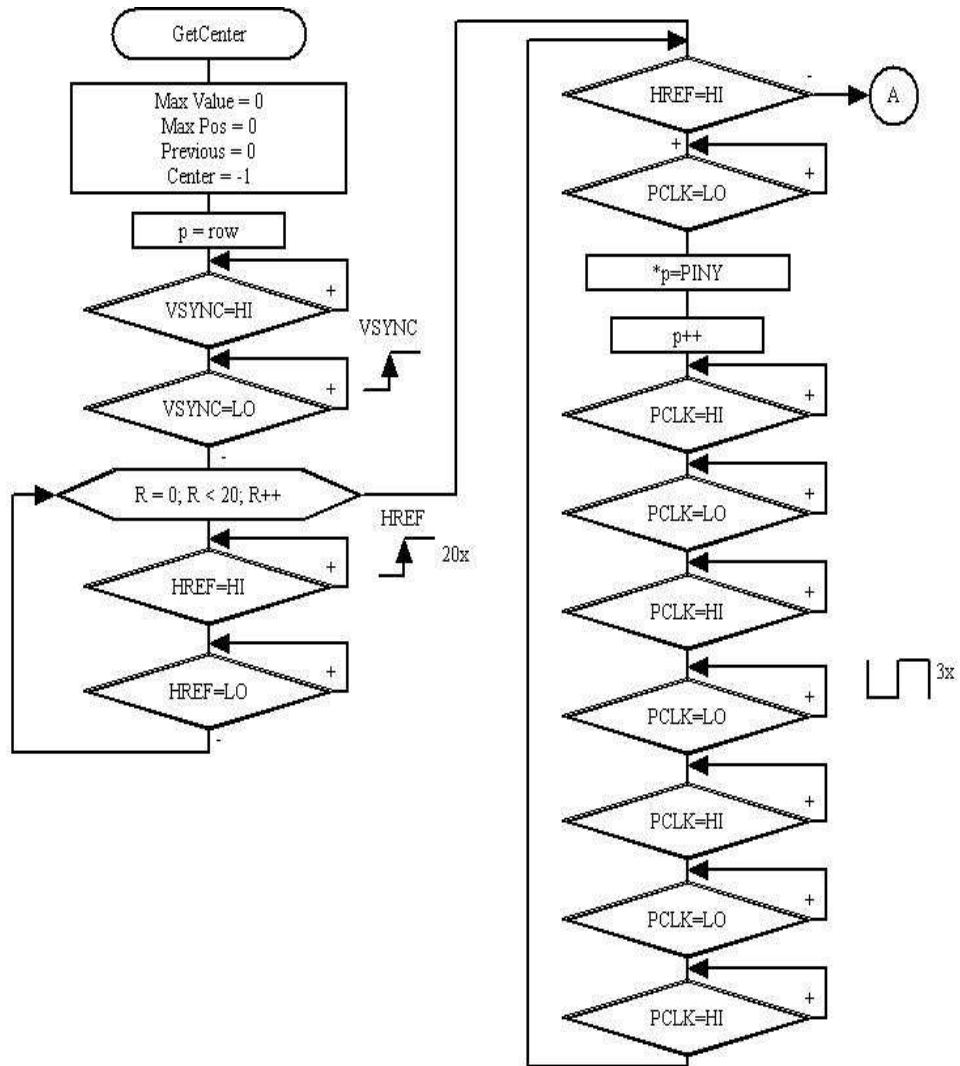
Funkce photo slouží pro zachycení fotografie z kamerového modulu. Čtení obrazu probíhá tak, že se vezme první sloupec snímku a posílá se bod po bodu do PC, potom druhý sloupec a tak dále, než se přenese celý obraz.



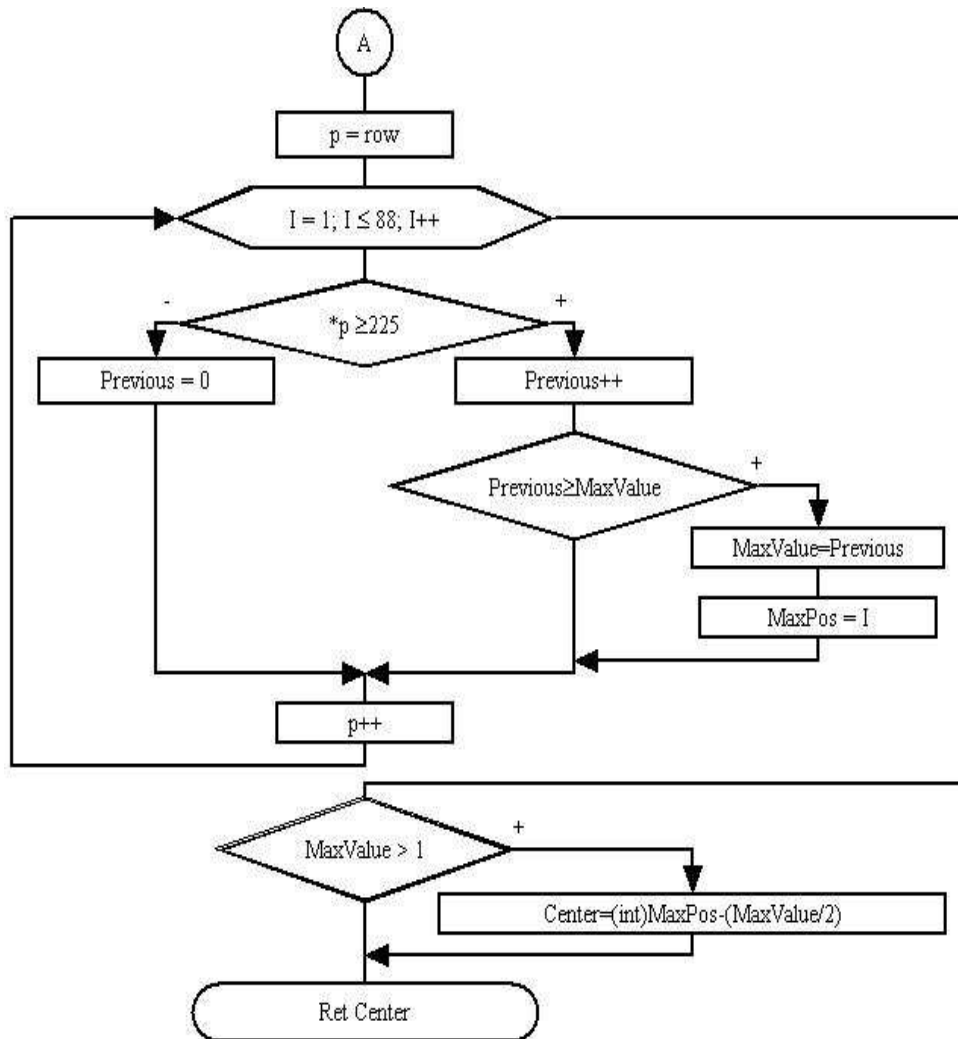
Obr. 6.5: Vývojový diagram funkce photo.

6.5.2 Funkce getcenter

Funkce getcenter určuje střed sledovaného objektu. Funkce vrací číslo sloupce, který je ve středu objektu. Z kamery se přečte horizontální řádek a uloží se do paměti. Poté čteme každý bod a zjišťujeme, zda jsou jasnější než prahová hodnota. Nakonec se zaměří na zónu, která je jasnější než prahová hodnota a vypočte se její střed.



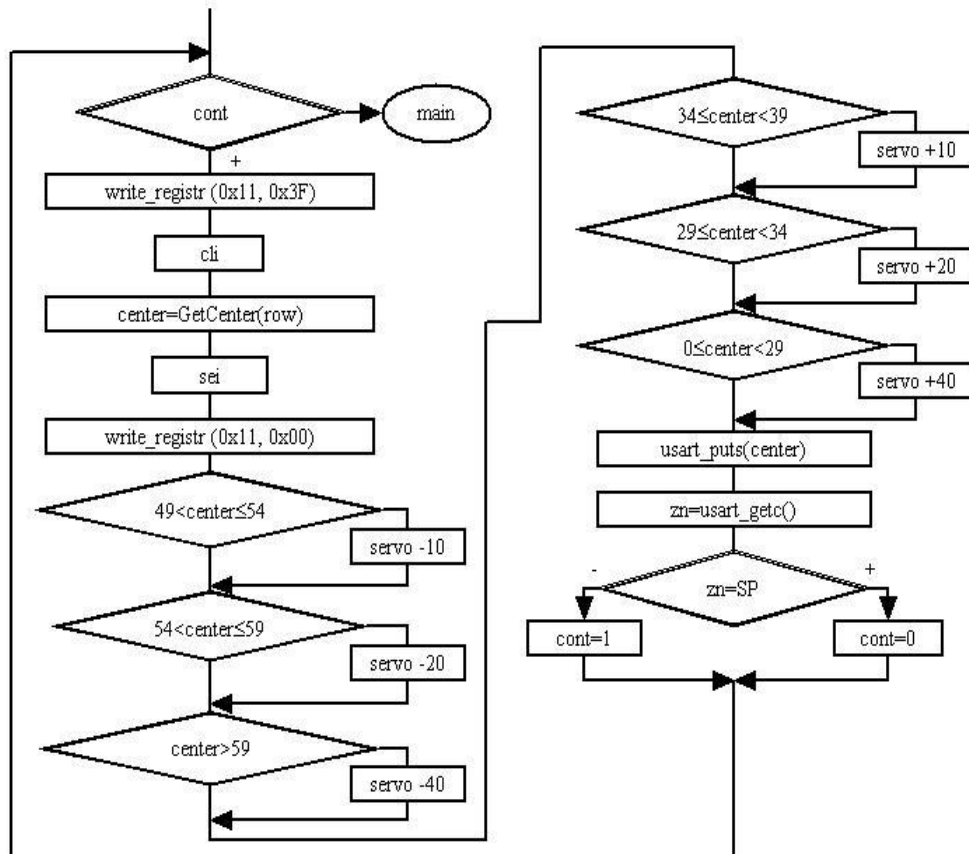
Obr. 6.6: Vývojový diagram funkce getcenter.



Obr. 6.7: Vývojový diagram funkce getcenter.

6.5.3 Funkce track

Funkce track je určena pro sledování bílého objektu. Využívá funkci getcenter a určuje polohu serva.

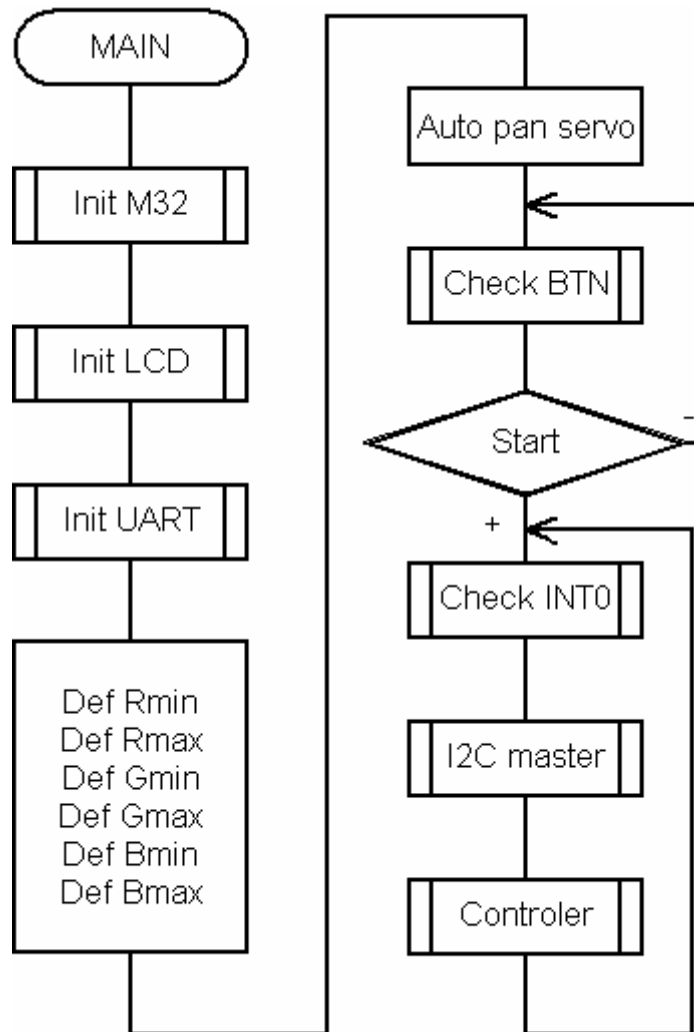


Obr. 6.8: Vývojový diagram funkce track.

Důležitá část zdrojového kódu je uvedena v příloze C.

6.5.4 Funkce pro ovládání podvozku

Po inicializaci systému a nastavení intervalů barevných složek sledovaného bodu přejde program do nekonečné smyčky. Ve smyčce se opakovaně provádí tři úlohy. Kontrola a obsluha externího přerušení (stav baterie, nárazníky). Druhým úkolem je softwarová obsluha zařízení I2C master telemetrie a ovládání podvozku. Posledním bodem je ovladač chování robotu (konečný automat).



Obr. 6.9: Vývojový diagram hlavní funkce programu pro ovládání podvozku.

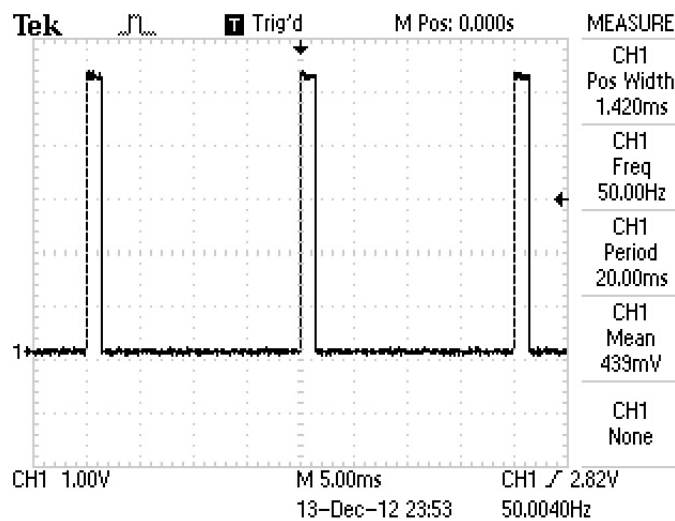
Funkce controller obsahuje definici a nastavení šesti struktur (slabá baterie, čekání na start, sledování objektu, náraz do překážky). V každé struktuře jsou proměnné pro rychlost motorů, stav senzorů atd. Struktury mají různou prioritu. Jedna z důležitých proměnných je state, který zaznamenává, která struktura se zrovna vykonává.

Funkce follow (sledování objektu) nejdříve zahájí komunikaci kamery s podvozkem. Nejdříve se zjistí zda je před kamerou sledovaný objekt. Pokud je dopočítají se hodnoty tak, aby byl objekt na střed kamery. Podle toho se natočí podvozek.

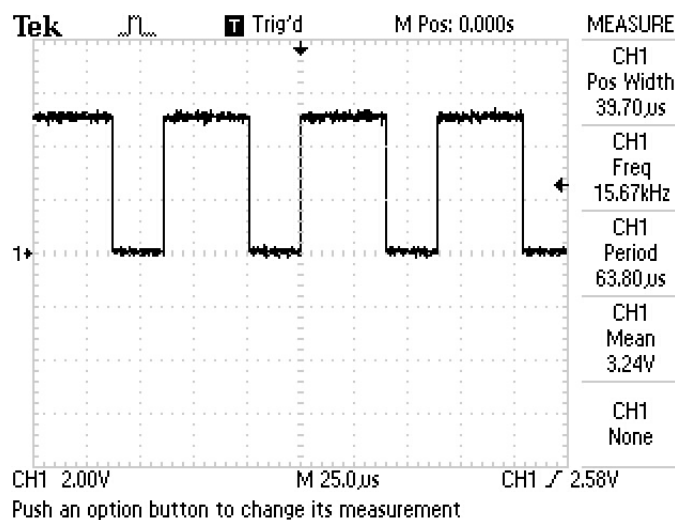
7 DOSAŽENÉ VÝSLEDKY

7.1 Průběhy z osciloskopu

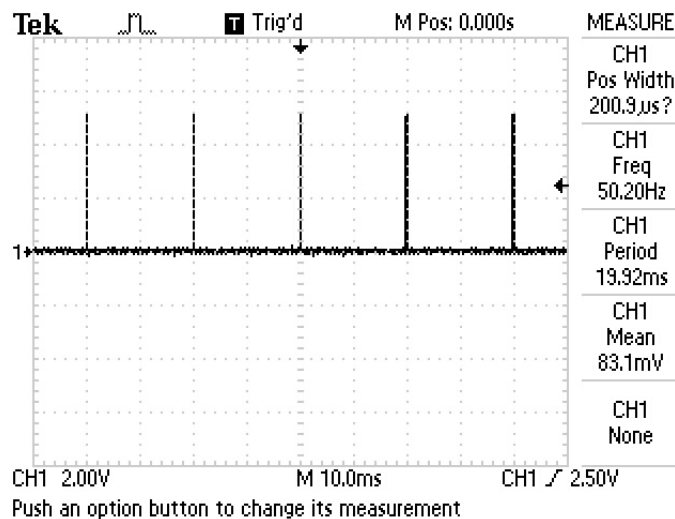
Pomocí osciloskopu jsem zaznamenal průběhy a odměřil hodnoty potřebných signálů. Nejprve jsem otestoval chování serva. Poté jsem zobrazil signály kamerového modulu a to signál HREF, VSYN a PCLK viz. níže.



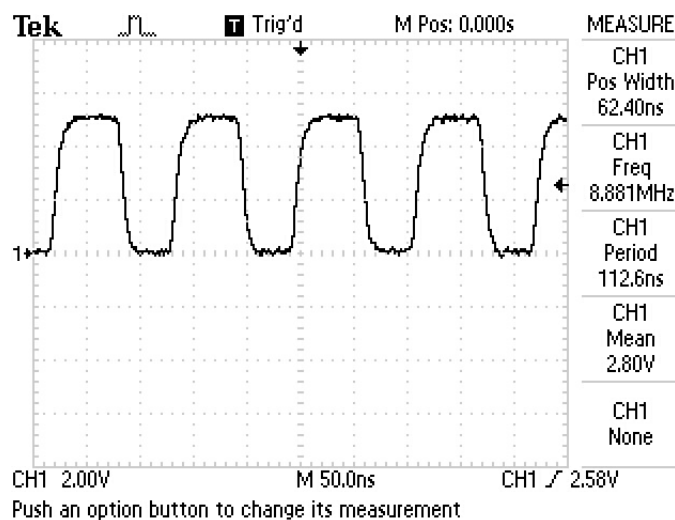
Obr. 7.1: Časový průběh řídicího impulsu serva pro natočení 0°.



Obr. 7.2: Časový průběh horizontální reference modulu C3088 HREF.



Obr. 7.3: Časový průběh vertikální synchronizace modulu C3088 VSYN.



Obr. 7.4: Časový průběh časování obrazových bodů modulu C3088 PCLK.

7.2 Naměřené hodnoty

Tab. 7.1: Přehled snímkové rychlosti jednotlivých řešení.

| Konstrukce | Rozlišení | Snímková frekvence [FPS] |
|------------------------|-----------|--------------------------|
| rameno bez FIFO paměti | QCIF MONO | 25 |
| rameno s FIFO pamětí | QCIF MONO | 13,70 |
| robotický podvozek | QCIF RGB | 8 |

Naměřené hodnoty časování s výstupních signálů kamerového modulu:

PCLK:

- frekvence = 8,881MHz => perioda jednoho pulzu = 112,6ns
- z hodnot vyplývá, že je nastaveno 16-ti bitové časování

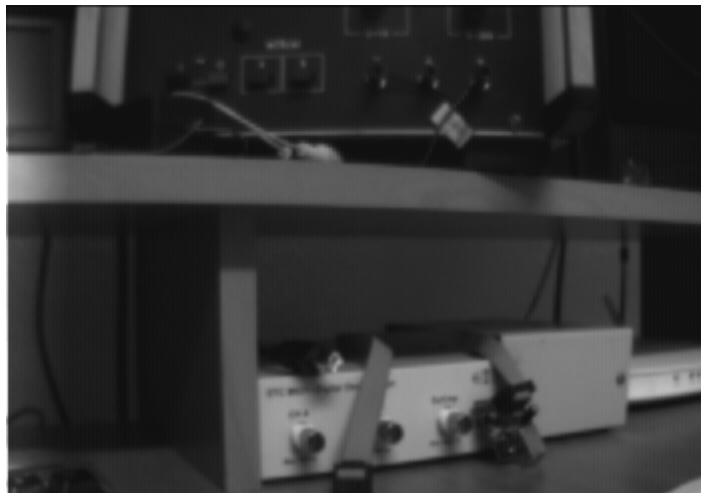
VSYNC:

- frekvence = 50,20Hz => perioda jednoho pulzu = 19,92ms
- šířka synchronního pulzu = 200 μ s
- periody vyplývá, že po 19,92ms může přijmout nový snímek

HREF

- frekvence = 15,67kHz => perioda jednoho pulzu = 63,80 μ s
- tento signál určuje typ dat (obrazová či jiného charakteru)

7.3 Získané obrázky z kamery



Obr. 7.5: Získaný obrázek č. 1 z kamerového modulu s rozlišením 352 x 244.



Obr. 7.6: Získaný obrázek č. 2 z kamerového modulu s rozlišením 320 x 240.

8 ZÁVĚR

Diplomová práce navazuje na práci semestrální jejíž úkolem bylo seznámení s možnostmi propojení jednoduchého obrazového senzoru s digitálním výstupem a procesoru. Otestování základní komunikace a získání dat pro zpracování obrazu.

Provedl jsem rozbor možností a řešení s dostupným kamerovým modulem C3088. Uvedl jsem některé již řešené projekty, z kterých se dalo čerpat hodně informací. Seznámil jsem se s možnostmi komunikace modulu s mikroprocesorem a zvolil jsem pro zpracování mikroprocesor ATMEL řady ATmega. Rozebral jsem také možnosti práce s modelářskými servy. Popsal jsem vlastnosti a možnosti použitého robotického podvozku RP6

Při prvotním seznamování s modulem C3088 jsem vycházel s dostupného projektu AVRcam. Při sestavení obvodu a propojení s PC jsem narazil na problém s komunikací. Systém sice navázal spojení s PC, ale při zasílání obrazových dat modul přestal komunikovat. Zkušenosti z konstrukce tohoto projektu a poznatky získané při testech jsem použil pro návrh vlastního systému. Nově sestavený kamerový systém ovládá mikroprocesor ATmega32 později ATmega644. Z kamerového modulu získáváme data pouze na sběrnici Y, a proto je výsledný obraz černobílý. Propojení s PC je pomocí sériové komunikace USART signálově převedené na rozhraní RS232 a data jsou pozorovány pomocí terminálového programu HTerm.

Při realizaci a měření prototypu jsem pomocí osciloskopu zaznamenal několik signálových průběhů. První průběh ukazuje řídicí impuls, který nastavuje pozici serva. Zbýlé tři průběhy ukazují vlastnosti synchronizačních a vzorkovacích signálů kamerového modulu. Jedná se o synchronizační signály VSYN, HREF a časový signál PCLK, který signalizuje platná data na sběrnici. Při komunikaci a zpracování obrazových dat se mi podařilo pomocí kamerového modulu C3088 získat fotografii s rozlišením 352 x 244. Další funkce, která se mi povedla naprogramovat a zprovoznit je pozorování bílého předmětu. Při pohybu bílého předmětu před objektivem kamery se snímací modul natáčí pomocí serva na střed bílého předmětu. Celý program byl napsán v jazyce C a funkce pro zpracování obrazu je možné studovat na vývojových diagramech v kapitole 6. Část zdrojového kódu je uvedena v příloze C.

Dále byl realizován robotický podvozek s kamerou. Pro oživení komunikace s podvozkem jsem použil dostupný modul CMUcam4. Jeho prvotní funkci jsem zkusil pomocí programu CMUGUI na PC. Při správné funkci se kamera připojila k podvozku pomocí rozhraní UART. Při výběru pozorovacího objektu bylo zjištěno, že kamera lépe reaguje na předmět tvaru koule než kvádr. Dalším aspektem byla osvětlenost místnosti. Při malém osvětlení nebyl objekt kamerou zaznamenán. Při velkém osvětlení se objekt naopak leskl a kamera nedokázala určit sledovaný bod objektu. Barva objektu byla zvolena žlutá, která měla nejlepší vlastnosti snímání.

K diplomové práci je přiloženo CD se všemi zdrojovými kódy. Dále disk obsahuje potřebné programy na PC, katalogové listy použitých modulů a součástí, výkresy konstrukčních dílů a pár fotografií z měření.

LITERATURA

- [1] MATOUŠEK, D. *Práce s mikrokontroléry Atmel AVR*. Praha: BEN – technická literatura, 2003.
- [2] NOVÁK, P. *Mobilní roboty – pohon, senzory, řízení*. Praha: BEN – technická literatura, 2005.
- [3] Modul C3088 [online]. 1999 – [cit. 2012-09-30].
Dostupné na www: <http://www.electronics123.net/amazon/datasheet/c3088.pdf>.
- [4] Omnivision OV6620 [online]. 1999 – [cit. 2012-09-30].
Dostupné na www: <http://shop.snainstruments.com/docs/ov6620.pdf>.
- [5] ATmega32 [online]. 2012 – [cit. 2012-11-08].
Dostupné na www: <http://www.gme.cz/dokumentace/958/958-120/dsh.958-120.1.pdf>.
- [6] Robotika [online]. 2012 – [cit. 2012-11-25].
Dostupné na www: <http://www.robotika.cz>
- [7] JROBOT [online]. 2004 – [cit. 2012-10-16].
Dostupné na www: <http://www.jrobot.net>.
- [8] Robotik systém [online]. 2010 – [cit. 2012-12-01].
Dostupné na www: http://www.robotiksystem.com/servo_motor_types_properties.html.
- [9] Hobbyrobot [online]. 2001 – [cit. 2012-10-15].
Dostupné na www: <http://www.hobbyrobot.cz>
- [10] McCOMB, G., PREDKO, M. *Robot builder's bonanza United States of America: The McGraw-Hill Companies*, 2006.
- [11] SCCB Bus [online]. 2007 – [cit. 2012-10-16].
Dostupné na www: <http://www.ovt.com/support/>
- [12] Kit MB-ATmega16/32 [online]. 2012 – [cit. 2012-11-08].
Dostupné na www: http://pk-design.net/HtmlCz/MB_ATmega16.html
- [13] E-shop Conrad [online]. 2013 – [cit. 2013-04-18].
Dostupné na www: <http://www.conrad.cz>
- [14] Hardwarové pomůcky [online]. 2012 – [cit. 2013-04-20].
Dostupné na www: <http://www.hw.cz>
- [15] CMUcam [online]. 2011 – [cit. 2013-03-20].
Dostupné na www: <http://www.cmucam.org>
- [16] Paměť FIFO AL422B [online]. 1999 – [cit. 2013-04-30].
Dostupné na www: <http://www.averlogic.com/AL422B.asp>
- [17] AVRcam [online]. 2011 – [cit. 2013-03-20].
Dostupné na www: <http://www.jrobot.net>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

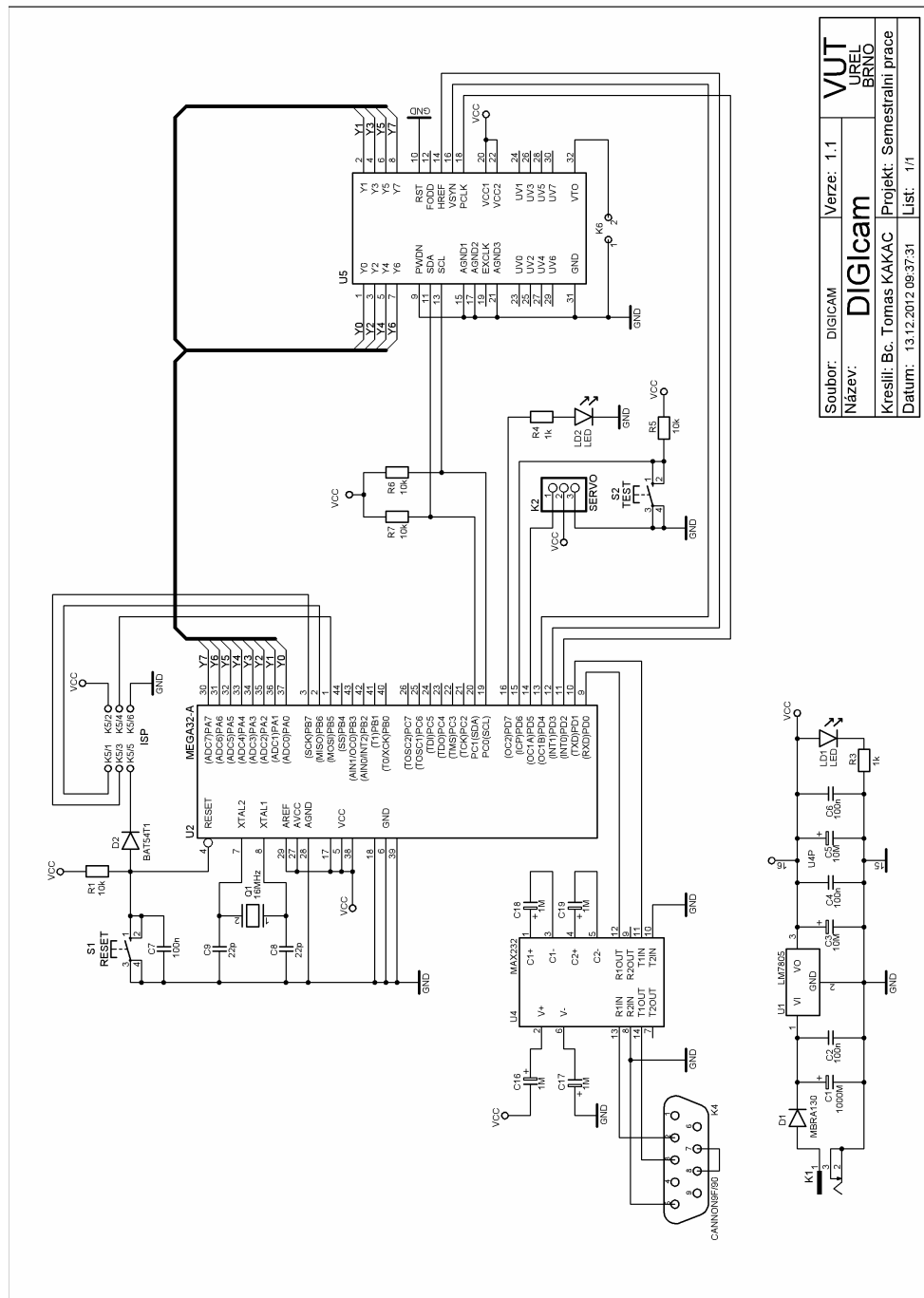
| | |
|------------------|---|
| f | frekvence |
| mW | [miliwatt], jednotka výkonu |
| ms | [milisekunda], jednotka času |
| MHz | [megahertz], jednotka frekvence |
| V | [volt], jednotka elektrického napětí |
| A | [ampér], jednotka elektrického proudu |
| px | obrazový bod |
| | |
| I ² C | Inter-Integrated Circuit <i>sběrnice komunikace mezi obvody</i> |
| CMOS | Complementary Metal-Oxide-Semiconductor <i>technologie výroby integrovaných obvodů</i> |
| PWM | Pulse Width Modulation <i>pulsní šířková modulace</i> |
| SCCB | Serial Camera Control Bus <i>sériová sběrnice ovládání kamery</i> |
| CTC | Clear Timer on Compare <i>nulování časovače komparátorem</i> |
| USART | Universal Synchronous/Asynchronous Receiver and Transmitter <i>univerzální synchronní/asynchronní přijímač a vysílač</i> |
| SPI | Serial Peripheral Interface <i>rozhraní sériových periférií</i> |
| RGB | RED – GREEN – BLUE <i>barevné složky červená – zelená – modrá</i> |
| IP | Internet Protocol <i>komunikační protokol Internetu</i> |
| USB | Universal Serial Bus <i>univerzální sériová sběrnice</i> |
| MKO | Monostabilní klopný obvod |

SEZNAM PŘÍLOH

| | | |
|----------|---|-----------|
| A | Návrh zařízení | 43 |
| A.1 | Obvodové zapojení pro polohovací rameno s 1 servem..... | 43 |
| A.2 | Obvodové zapojení řídicí desky se 2 servy | 44 |
| A.3 | Obvodové zapojení paměti FIFO..... | 45 |
| B | Konstrukční díly | 46 |
| B.1 | Díly pro sestavení polohovacího ramene..... | 46 |
| B.2 | Díly k robotickému podvozku | 49 |
| C | Zdrojové kódy | 52 |
| C.1 | Cam.h..... | 52 |
| C.2 | Cam.c | 53 |
| C.3 | Autonomní chování robotnického podvozku..... | 55 |
| D | Fotografie z měření | 57 |

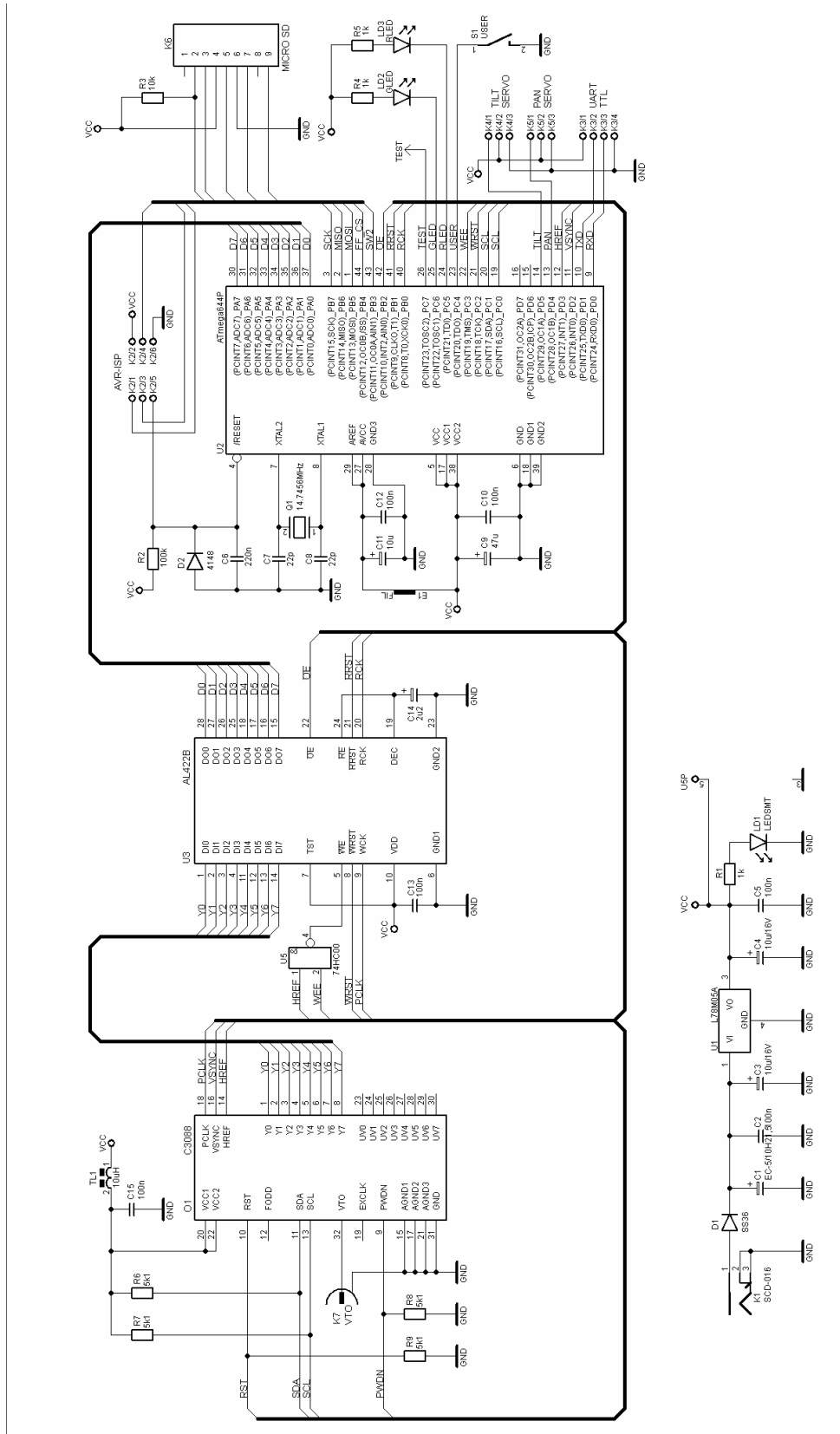
A NÁVRH ZAŘÍZENÍ

A.1 Obvodové zapojení pro polohovací rameno s 1 servem

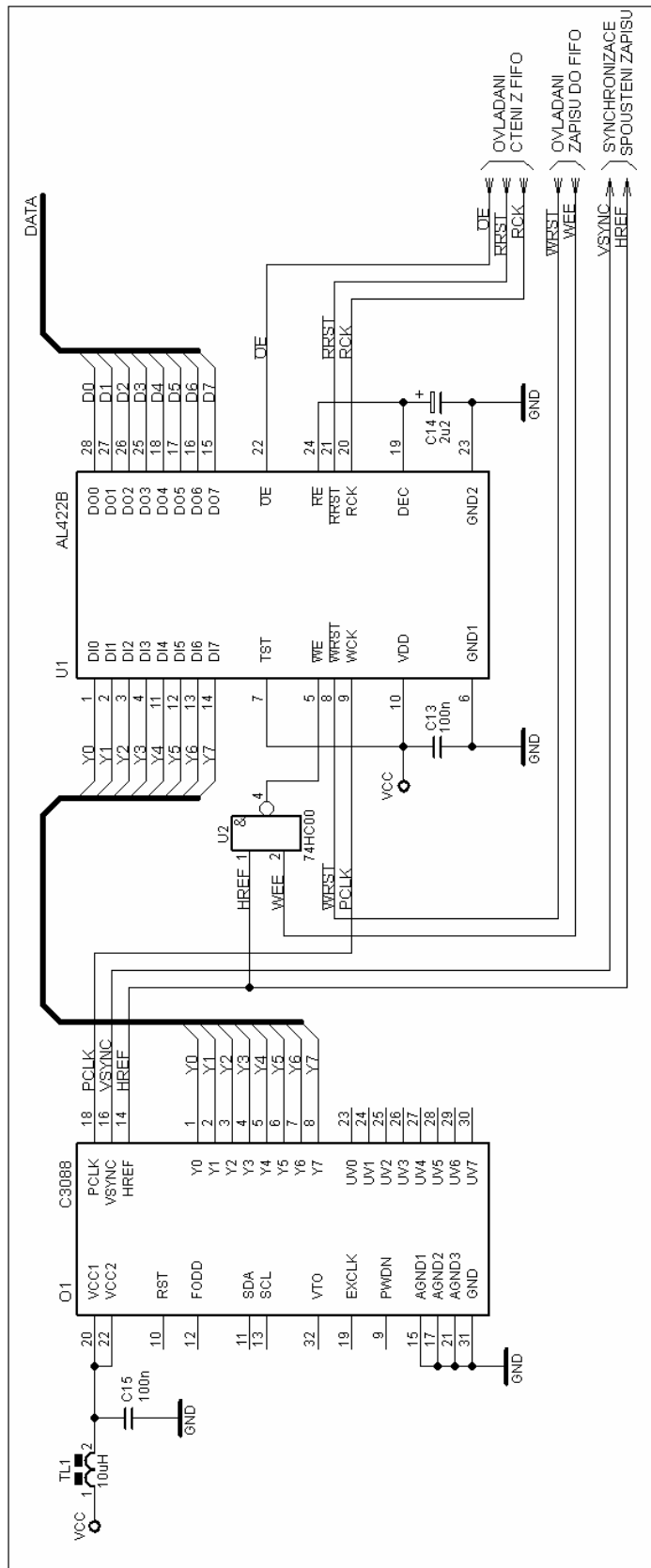


| | |
|----------------------------|----------------------------|
| Soubor: DIGICAM | Verze: 1.1 |
| Název: DIGICAM | VUT UREL BRNO |
| Kreslí: Bc. Tomas KAKAC | Projekt: Semestrální práce |
| Datum: 13.12.2012 09:37:31 | List: 1/1 |

A.2 Obvodové zapojení řídicí desky se 2 servy

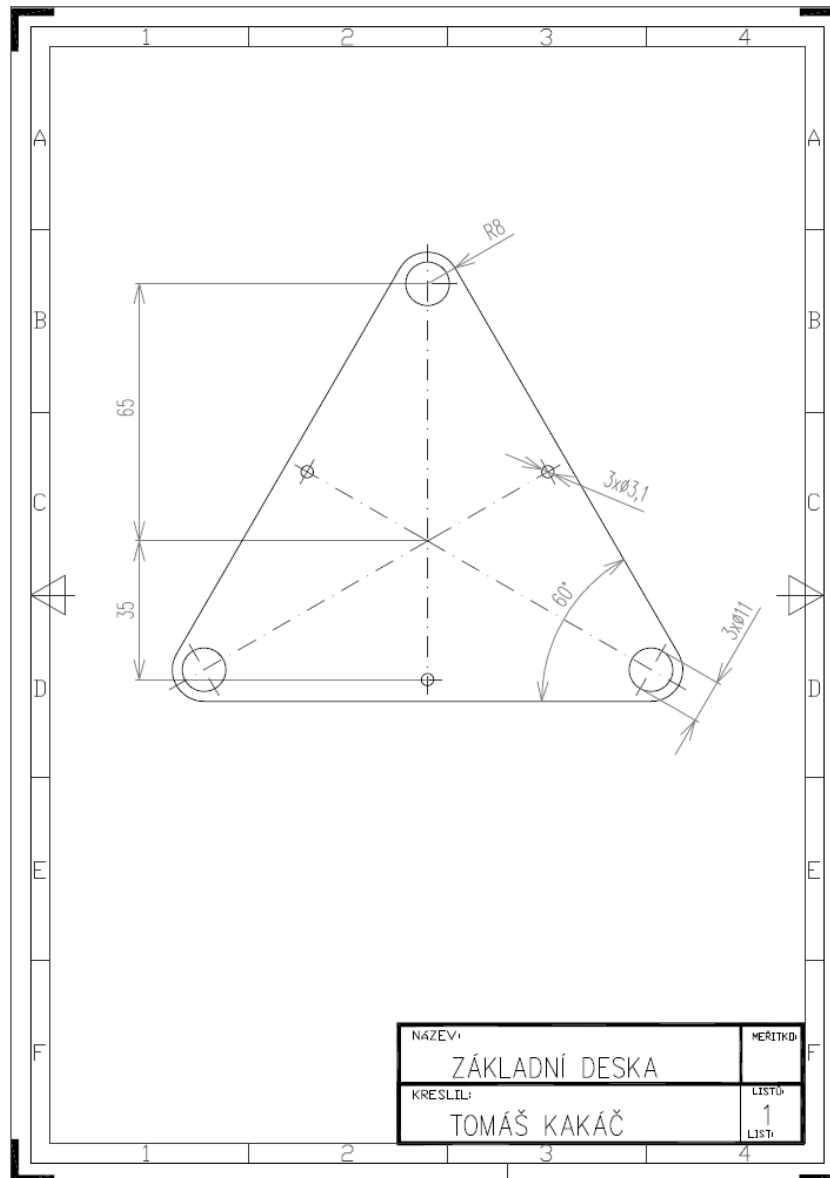


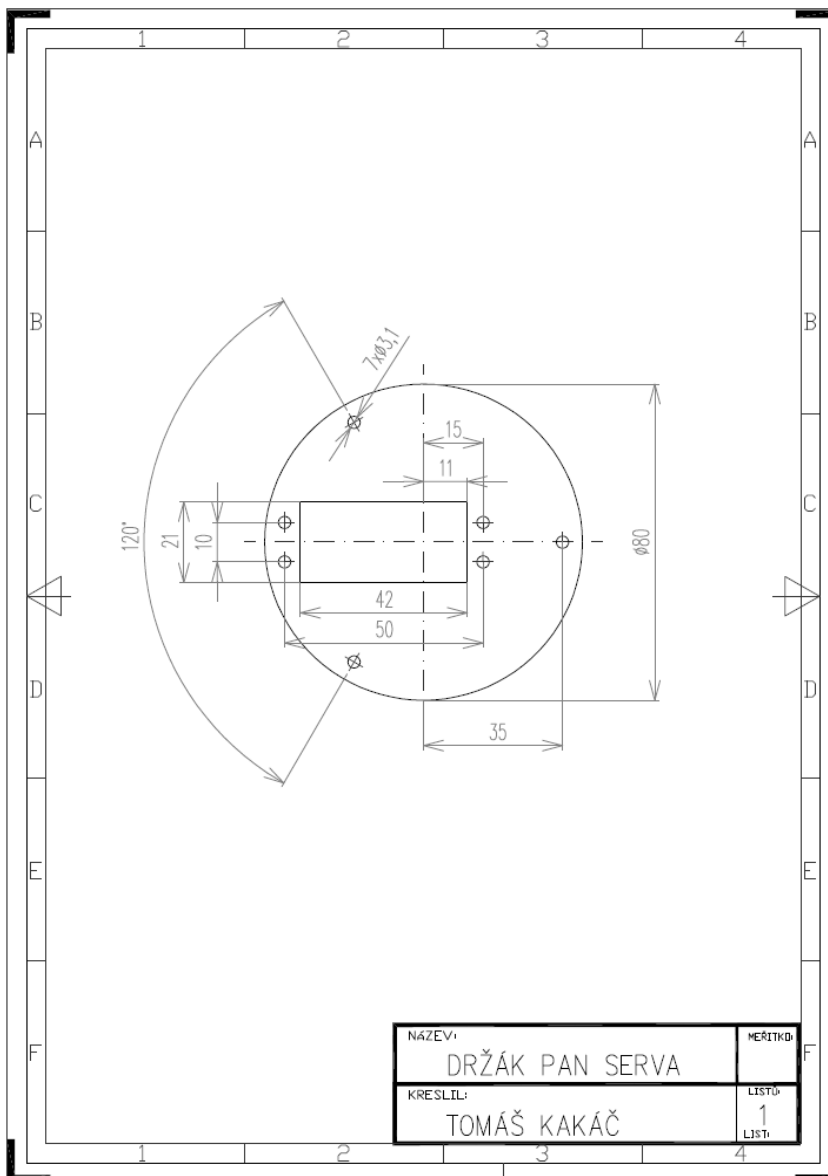
A.3 Obvodové zapojení paměti FIFO

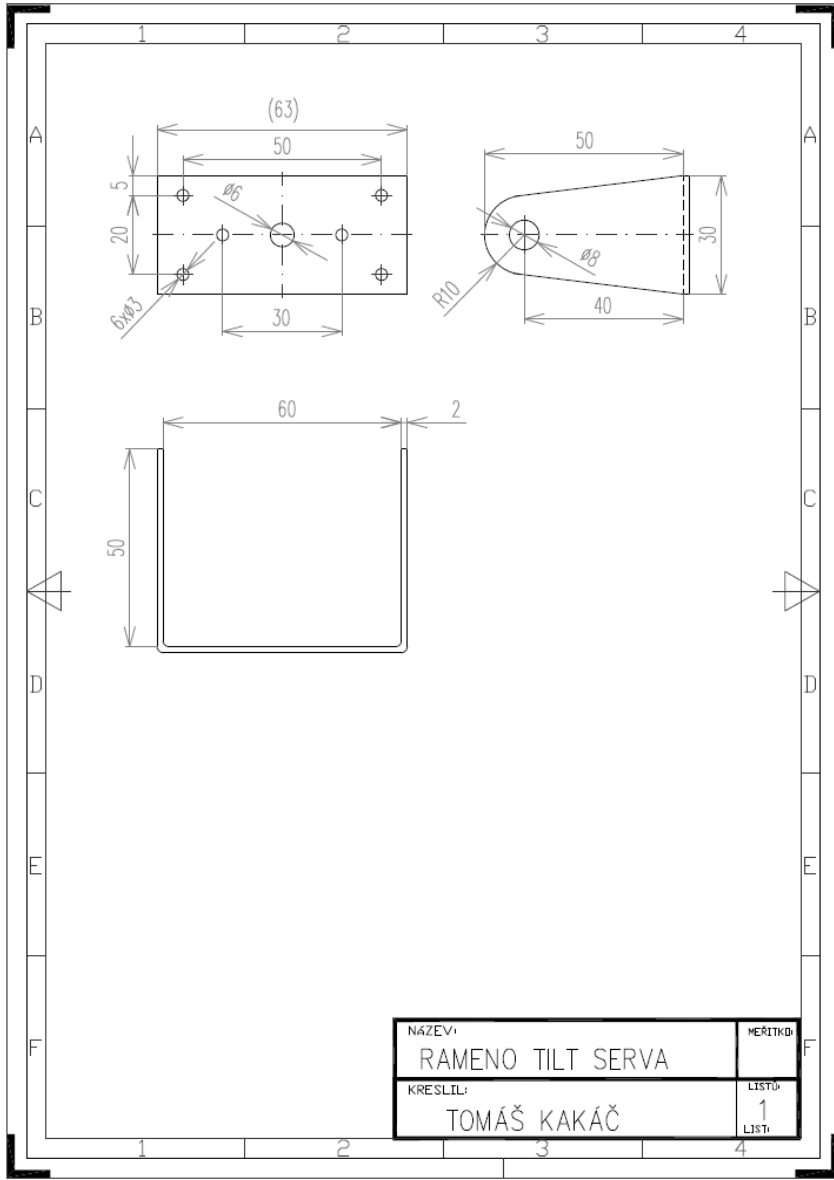


B KONSTRUKČNÍ DÍLY

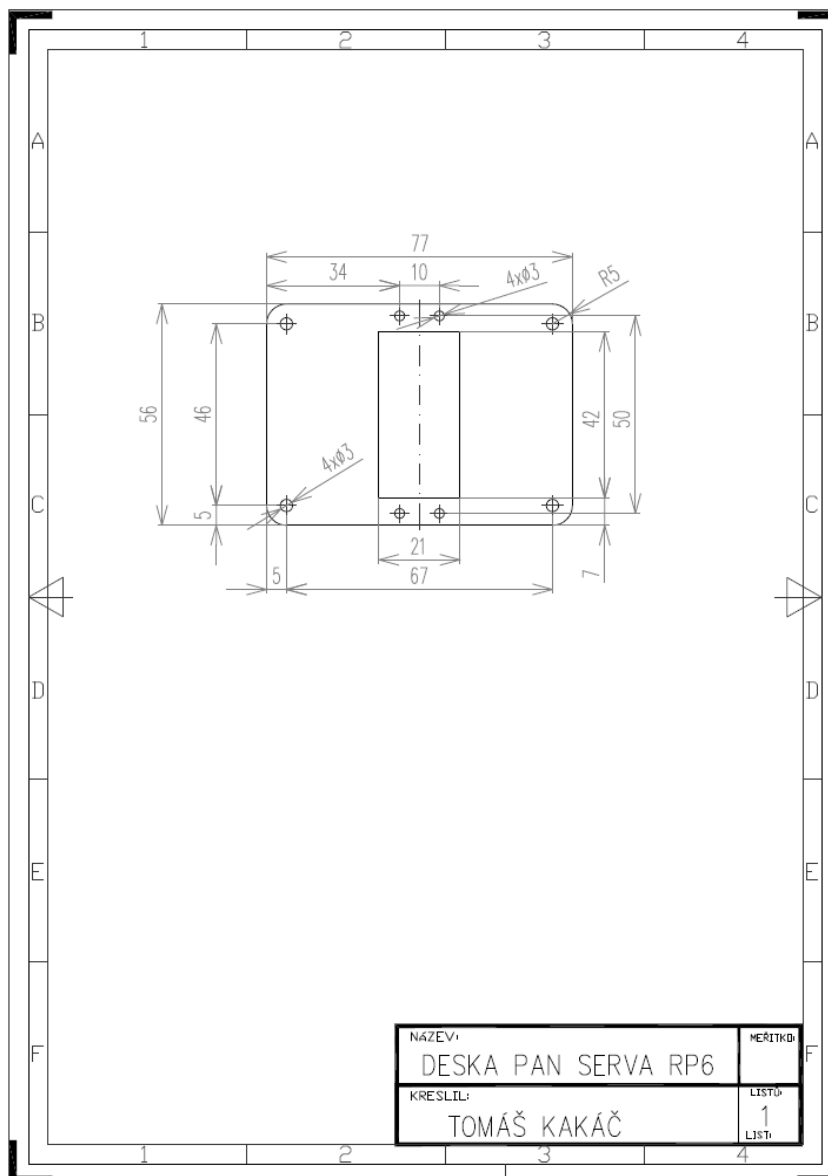
B.1 Díly pro sestavení polohovacího ramene

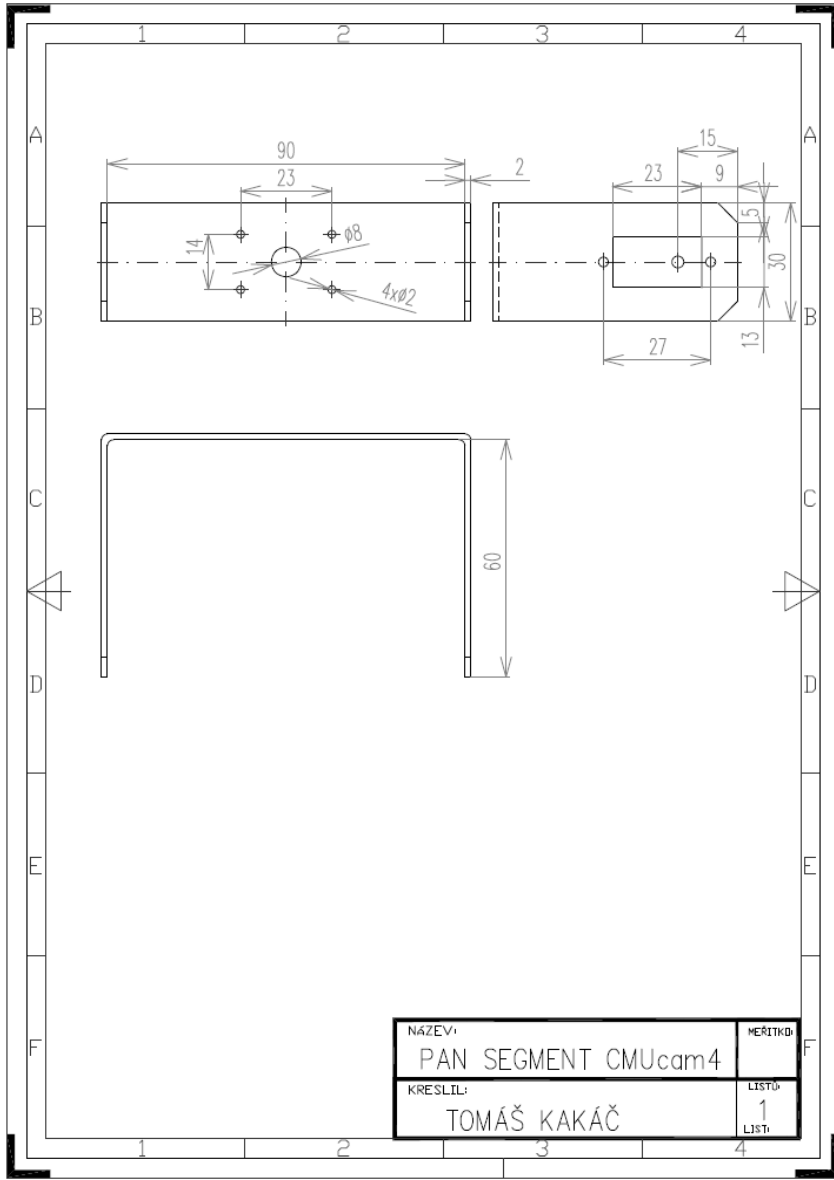


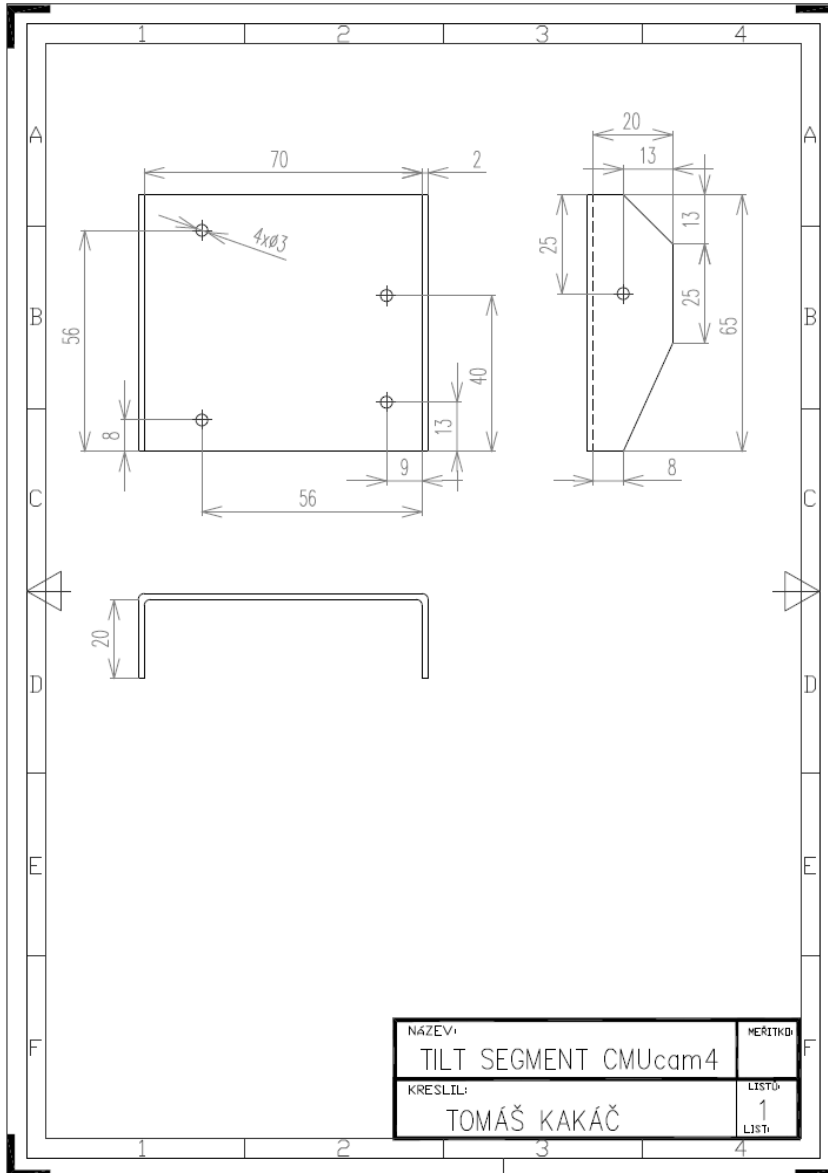




B.2 Díly k robotickému podvozku







C ZDROJOVÉ KÓDY

C.1 Cam.h

```
/*
*****
* Nazev: CAM - hlavickovy soubor modulu kamery
* Verze: 1.0
* Popis: knihovna pro praci s kamerovym modulem C3088.
*****
*/

/* ----- */
/*          Definice portu datove sbernice Y          */
/* ----- */

#define PORTY PORTA
#define PINY PINA
#define DDRY DDRA

/* ----- */
/*          Definice maker pro sledovani signalu      */
/* ----- */

#define isPCLKdown bit_is_clear(PIND, 2)
#define isPCLKup bit_is_set(PIND, 2)
#define isHREFdown bit_is_clear(PIND, 3)
#define isHREFup bit_is_set(PIND, 3)
#define isVSYNdown bit_is_clear(PIND, 4)
#define isVSYNup bit_is_set(PIND, 4)

/* ----- */
/*          Prototypy sdilenych funkci                */
/* ----- */

void camports_init(void);
void photo(void);
int getcenter(char *row);
```

C.2 Cam.c

```
/*
*****
* Nazev: CAM - modul kamery
* Prekl: AVR-GCC
* MCU:   ATMEL ATmega32 @16.000 MHz
* Verze: 1.0
* Zmena: -
* Popis: knihovna pro praci s kamerovym modulem C3088.
*****
*/

#include "cam.h"
#include "delay.h"
#include "usart.h"
#include "servo.h"

/* -----
* Nazev: camports_init
* Popis: funkce inicializuje port kameroveho modulu C3038
* Vstup: -
* Vraci: -
* -----
*/
void camports_init(void)
{
    DDY = 0x00;
    DDRD = (DDRD & 0xE3);
}

/* -----
* Nazev: photo
* Popis: funkce zachyti fotografii z kameroveho modulu C3038
* Vstup: -
* Vraci: -
* -----
*/
void photo(void)
{
    for (int y = 0; y < 352; y++)
    {
        while (isVSYNup);           // cekani na zacatek snimku
        while (isVSYNdown);
        for (int r=0; r<244; r++) {   // cyklus 244 radku
            while (isHREFdown);
            for (int h=0; h<y; h++) { // cyklus 352 bodu na radku
                while (isPCLKup);    // vzorkovani sbernice Y
                while (isPCLKdown); // nabeznou hranou PCLK
                usart_putc(PINY);     // cteni a vysilani dat
            }
            while (isHREFup);        // konec radku
        }
    }
}
```

```

/* -----
* Nazev: getcenter
* Popis: funkce urci stred sledovaneho bileho objektu
* Vstup: -
* Vraci: cislo sloupce, který je uprostred bileho objektu
* -----
*/
int getcenter(char *row)
{
    int maxvalue = 0;           // maxi. pocet bodu oblasti
    int maxpos = 0;
    int previous = 0;
    int center = -1;

    char *p;                   // ukazatel na pole pixelu
    p = row;                   // pole bodu jednoho radku

    while (isVSYNup);          // snimkova synchronizace
    while (isVSYNdown);
    for (int r=0;r<20;r++) {    // vertikalni nastaveni
        while (isHREFup);      // na 21. radek
        while (isHREFdown);
    }

    while (isHREFup) {         // cyklus vzorkovani radku
        while(isPCLKdown);     // vzorkovani nabeznou hranou
        *p = PINY;             // jas bodu se ulozi do pole
        p++;                   // posunuti ukazatele
        while(isPCLKup);
        while(isPCLKdown);
        while(isPCLKup);
        while(isPCLKdown);
        while(isPCLKup);
        while(isPCLKdown);
        while(isPCLKup);       // snima se kazdy 5. pixel
    }

    p = row;                   // ukazatel na zacatek dat
    for (int i=1;i<=88;i++) {  // v poli je 88 pixelu
        if (*p >= 225) {       // prahova hodnota bile
            previous++;        // pocitadlo bilych bodu
            if (previous >= maxvalue) { // hleda se nejvetsi souvisla
                maxvalue = previous; // oblast bilych bodu
                maxpos = i;     // pravy okraj oblasti
            }
        }
        else {                 // pixel pod prah. hodnotou
            previous = 0;      // konec souvisle oblasti
        }
        p++;                   // posun ukazatele
        if (maxvalue > 1) {    // souvisla oblast vice bodu
            center = (int)(maxpos - (maxvalue/2)); // vypocet stredu oblasti
        }
    }

    return center;            // vraci stred bile oblasti
}

```


C.3 Autonomní chování robotnického podvozku

```
/* -----  
* Nazev: behaviourController()  
* Popis: Radic chovani ovlada jednotlivé ukoly subsumpcni  
* architektury.  
* Implementuje urovne preruseni pro ruzné chovani robotu.  
* Radic take zobrazuje aktivni chovani robotu na LCD  
* displeji.  
* Vstup: -  
* Vraci: -  
* -----  
*/  
void behaviourController(void)  
{  
    // Vola vsechny ukoly chovani robotu:  
    behaviour_checkLowBattery();  
    behaviour_waitForStart();  
    behaviour_follow();  
    behaviour_cruise();  
    behaviour_avoid();  
    behaviour_escape();  
  
    // Zpracuje prikazy podle urovne priority:  
    if(checkLowBattery.state != IDLE) { // Nejvyssi priorita - 6  
        displayBehaviour(7);  
        moveCommand(&checkLowBattery);  
    }  
    else if(waitForStart.state != IDLE) { // Priorita - 5  
        displayBehaviour(6);  
        moveCommand(&waitForStart);  
    }  
    else if(escape.state != IDLE) { // Priorita - 4  
        displayBehaviour(5);  
        moveCommand(&escape);  
    }  
    else if(avoid.state != IDLE) { // Priorita - 3  
        displayBehaviour(4);  
        moveCommand(&avoid);  
    }  
    else if(follow.state != IDLE) { // Priorita - 2  
        displayBehaviour(3);  
        moveCommand(&follow);  
    }  
    else if(cruise.state != IDLE) { // Priorita - 1  
        displayBehaviour(2);  
        moveCommand(&cruise);  
    }  
    else { // Nejnizsi priorita - 0  
        displayBehaviour(1); // Implicitni prikaz - nedela nic  
        moveCommand(&STOP); // Tento stav normalne nenastane  
    }  
}
```

```

/* -----
* Nazev: behaviour_follow()
* Popis: Chovani pri sledovani barevneho objektu - stredni poloha
*       X = 80
* Vstup: -
* Vraci: -
* -----
*/
void behaviour_follow(void)
{
    writeString_P("TC ");           // Vyslani povelu TC
    writeInteger(rMin,DEC);         // Rmin
    writeString_P(" ");
    writeInteger(rMax,DEC);         // Rmax
    writeString_P(" ");
    writeInteger(gMin,DEC);         // Gmin
    writeString_P(" ");
    writeInteger(gMax,DEC);         // Gmax
    writeString_P(" ");
    writeInteger(bMin,DEC);         // Bmin
    writeString_P(" ");
    writeInteger(bMax,DEC);         // Bmax
    writeChar('\r');                // CR
    if (ackRx()) {                  // Test ACK
        enterString();              // Cteni prijateho paketu
        if (rxBuf[0] == 'T') {      // Test paketu T
            xPos = getXpos();        // Prevod X-souradnice
        }
    }
    switch(follow.state)
    {
        case IDLE:
            if (xPos != 0) {
                setStopwatch2(0);
                startStopwatch2();
                follow.state = FOLLOW;
            }
            break;
        case FOLLOW:
            if (getStopwatch2() > 100) {
                // Vypocet rychlosti motoru podle polohy stredu telesa:
                int16_t dif = ((int16_t)(xPos - 80)>>1);
                if (dif > 40)
                    dif = 40;
                if (dif < -40)
                    dif = -40;

                setCursorPosLCD(1,11);
                writeIntegerLCD(dif,DEC);
                follow.speed_left = X_CENTER + dif;
                follow.speed_right = X_CENTER - dif;
            }
            else {
                stopStopwatch2();
                follow.state = IDLE;
            }
            break;
    }
}

```

D FOTOGRAFIE Z MĚŘENÍ

