



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

ALGORITMY ZPRACOVÁNÍ OBRAZU V OSMIBITOVÝCH PROCESORECH

IMAGE PROCESSING ALGORITHMS IN EIGHT-BIT MICROCONTROLLER

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MIROSLAV RÉBL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ZBYNĚK FEDRA, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Diplomová práce

magisterský navazující studijní obor
Elektronika a sdělovací technika

Student: Bc. Miroslav Rébl

ID: 72911

Ročník: 2

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Algoritmy zpracování obrazu v osmibitových procesorech

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte existující implementace propojení jednoduché kamery s procesorem Atmel AVR a vlastnosti obrazových dat získaných z kamery. Seznamte se s možnostmi uložení a adresování obrazových dat v paměti mikrokontroléru, s náročností na paměťový prostor a odhadněte limitace zpracování těchto dat.

Otestujte základní algoritmy práce s obrazovými daty (detekce výrazných objektů na základě monochromatického i barevného obrazu). Posuďte možnosti jednotlivých algoritmů z pohledy rychlosti, složitosti a implementace na platformě Atmel AVR.

Připravte ukázkovou aplikaci na základě předchozích projektů a konkrétních požadavků vedoucího. Sestavte potřebnou dokumentaci a aplikaci ověřte na dostupném zařízení.

DOPORUČENÁ LITERATURA:

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

[2] MANN, B. C pro mikrokontroléry. Praha: BEN - technická literatura, 2003.

Termín zadání: 7.2.2011

Termín odevzdání: 20.5.2011

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

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

Tato diplomová práce se zabývá implementací propojení jednoduché kamery a procesoru Atmel AVR. Dále se zaměřuje na získávání obrazových dat z kamery C3088 a manipulaci s řídicími registry modulu. Odhaduje možnou limitaci při zpracovávání dat touto cestou. Práce řeší zpracování ukázkové aplikace pro objektové vyhledávání prostřednictvím této implementace.

KLÍČOVÁ SLOVA

C3088, OV6620, digitální kamera, mikroprocesor, ATmega, AVR, I2C, JTAG, zpracování obrazu

ABSTRACT

The aim of master thesis was to investigate a simple camera and processor Atmel AVR unit implementation. Thesis describes how to get image from C3088 module and management over the bank of camera registers. Work includes limitation draft for this way of processing the information. Thesis also solves an exemplary model for object recognition using this implementation.

KEYWORDS

C3088, OV6620, digital camera, microprocessor, ATmega, AVR, I2C, JTAG, image processing

RÉBL, M. *Algoritmy zpracování obrazu v osmibitových procesorech*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 46s., 5s příloh. Vedoucí diplomové práce Ing. Zbyněk Fedra, Ph.D..

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Algoritmy zpracování obrazu v osmibitových procesorech 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 Mikroprocesor | 2 |
| 1.1 Obecné vlastnosti ATmega128 | 2 |
| 1.2 Sériová sběrnice TWI (I2C)..... | 3 |
| 1.2.1 Systém komunikace TWI..... | 3 |
| 1.2.2 Registry TWI | 4 |
| 1.3 Paměťová limitace MCU | 5 |
| 1.4 Propojení MCU a PC | 6 |
| 2 Digitální kameorový modul C3088 | 7 |
| 2.1 Základní vlastnosti | 7 |
| 2.2 Výstupní formáty digitálních dat | 9 |
| 2.3 Nastavení konfiguračních registrů | 12 |
| 3 Realizace ukázkové aplikace | 13 |
| 3.1 Konstrukce zařízení | 13 |
| 3.1.1 Návrh obvodu | 13 |
| 3.1.2 Převodník USB/USART | 13 |
| 3.1.3 Servomotory..... | 14 |
| 3.1.4 Externí SRAM | 14 |
| 3.1.5 Napájení | 15 |
| 3.1.6 Konektory periferií | 16 |
| 3.2 Software | 16 |
| 3.2.1 Sériová komunikace..... | 16 |
| 3.2.2 Vývojové prostředí programu | 17 |
| 3.2.3 Vlastní programy pro MCU | 17 |
| 3.2.4 Matlab | 22 |
| 4 Dosažené výsledky | 24 |

| | | |
|----------|---|-----------|
| 4.1 | Obrazová reprezentace..... | 24 |
| 4.1.1 | Chybné obrazové reprezentace | 27 |
| 4.2 | Vyhledávání objektů | 28 |
| 5 | závěr | 32 |
| | Literatura | 33 |
| | Seznam symbolů, veličin a zkratk | 35 |
| | Seznam příloh | 36 |

SEZNAM OBRÁZKŮ

| | | |
|------------|---|----|
| Obr. 1.1: | Rozložení pinů mikroprocesoru ATmega128 pouzdra TQFP64 (převzato z [3])..... | 2 |
| Obr. 1.2: | Zapojení sběrnice TWI (převzato z [17])..... | 3 |
| Obr. 1.3: | Adresní a datový formát paketu TWI (převzato z [17]) | 4 |
| Obr. 1.4: | Zapojení pinů konektoru JTAG (převzato z [6]) | 6 |
| Obr. 2.1: | Digitální kamera C3088 (převzato z [5])..... | 7 |
| Obr. 2.2: | Rozmístění pinů C0388..... | 7 |
| Obr. 2.3: | Data pixelů při 16-ti bitovém časování (převzato z [4])..... | 10 |
| Obr. 2.4: | Data pixelů při 8-mi bitovém časování (převzato z [4])..... | 11 |
| Obr. 2.5: | Výstupní signály C3088 (převzato z [4])..... | 11 |
| Obr. 3.1: | Blokové schéma zapojení | 13 |
| Obr. 3.2: | Převodník USB/USART (převzato z [8]) | 14 |
| Obr. 3.3: | Serva Hitec HS-422 (převzato z [13]) | 14 |
| Obr. 3.4: | Rozvržení paměti (převzato z [14]) | 15 |
| Obr. 3.5: | Nastavení paměti v překladači | 15 |
| Obr. 3.6: | Kabel 2x16 | 16 |
| Obr. 3.7: | Terminal 1.9b | 16 |
| Obr. 3.8: | Výpis jednoho kompletního snímku z výstupu kamery - program Imagegain | 17 |
| Obr. 3.9: | Popis podprogramu Photo..... | 18 |
| Obr. 3.10: | Čtení hodnoty registru..... | 19 |
| Obr. 3.11: | Výpis hodnot daných registrů po změně..... | 20 |
| Obr. 3.12: | Hlavní program - observer | 20 |
| Obr. 3.13: | Kontrolní fáze | 21 |
| Obr. 3.14: | Převod textového souboru na obraz | 22 |
| Obr. 3.15: | Výpis AVR studia o využití paměti | 22 |
| Obr. 3.16: | Výsledná obrazová reprodukce pro jasovou složku módu YUV v rozlišení 120x100..... | 23 |
| Obr. 4.1: | Obraz při rozlišení 120x100 obrazových bodů..... | 24 |
| Obr. 4.2: | Obraz při rozlišení 88x72 obrazových bodů..... | 24 |
| Obr. 4.3: | Obraz při rozlišení 44x36 obrazových bodů..... | 25 |

| | | |
|------------|---|----|
| Obr. 4.4: | Obraz při PCLK 69,25 kHz | 26 |
| Obr. 4.5: | Obraz při PCLK 260,8 kHz | 26 |
| Obr. 4.6: | Obraz při PCLK 4433 kHz | 26 |
| Obr. 4.7: | Frekvence se neshoduje s použitým krystalem | 27 |
| Obr. 4.8: | Příliš vysoký kmitočet PCLK | 27 |
| Obr. 4.9: | Špatná alokace paměti..... | 28 |
| Obr. 4.10: | uložený obraz s vyznačeným nalezeným objektem | 28 |
| Obr. 4.11: | Výstup v programu Terminal..... | 29 |
| Obr. 4.12: | Výsledek centrování programu observer | 29 |
| Obr. C.1: | Testovací zapojení s ATmega16..... | 39 |
| Obr. C.2: | finální zapojení s ATmega128..... | 39 |
| Obr. C.3: | Detail nepájivého pole | 40 |
| Obr. C.4: | Vstupní, Výstupní periferie a napájení | 40 |
| Obr. C.5: | Kamerový modul C3088 připevněný k servu Hitec | 41 |

SEZNAM TABULEK

| | | |
|-----------|---|----|
| Tab. 1.1: | Stavba komunikačního rámce I2C | 4 |
| Tab. 1.2: | Funkce jednotlivých bitů registru TWCR..... | 4 |
| Tab. 1.3: | TWPS - nastavení před-děličky | 5 |
| Tab. 1.4: | Označení vývodů programovacího rozhraní JTAG | 6 |
| Tab. 2.1: | Popis pinů digitální kamery C0388..... | 8 |
| Tab. 2.2: | 4:2:2 formát 16 bitů | 9 |
| Tab. 2.3: | 4:2:2 formát 8 bitů | 10 |
| Tab. 2.4: | Konfigurační rámec registru I ² C..... | 12 |
| Tab. 2.5: | Výpis vybraných registrů modulu..... | 12 |
| Tab. 4.1: | Závislost přenosové rychlosti na taktovacím kmitočtu PCLK - řádkové čtení | 25 |
| Tab. 4.2: | Proměnlivost jasových hodnot v rámci denní doby (aktuálního osvětlení). 30 | |
| Tab. 4.3: | Doba trvání jednoho cyklu programu observer - kompletní rozbor | 30 |

ÚVOD

Existují aplikace, kdy není možné použít pro přímé vyhodnocení obrazu výkonný počítač. Ať uvažujeme domovní videotelefon, robotiku či jiné přístroje využívající snímací techniku s nižším rozlišením. Budeme-li uvažovat robotické aplikace, může se jednat např. o robot, který má pomalou komunikační linku s řídicí centrálou a jehož fyzické rozměry a kapacita baterií neumožňují použít počítač přímo na těle přístroje.

Zde tedy přichází na řadu využití malých kamer propojených s mikropočítačem, který obraz vyhodnotí a po komunikační lince posílá pouze malý objem dat s potřebnými informacemi. Je zřejmé, že vzhledem k výkonu a paměťovým možnostem mikropočítačů není možné implementovat složité vyhodnocovací algoritmy, které je možné provádět na výkonných počítačích.

Současně je třeba vzít v úvahu, že není možné využít senzory či jiné snímací zařízení příliš vysoké rozlišovací schopnosti z důvodu menších paměťových kapacit jednočipových mikroprocesorů.

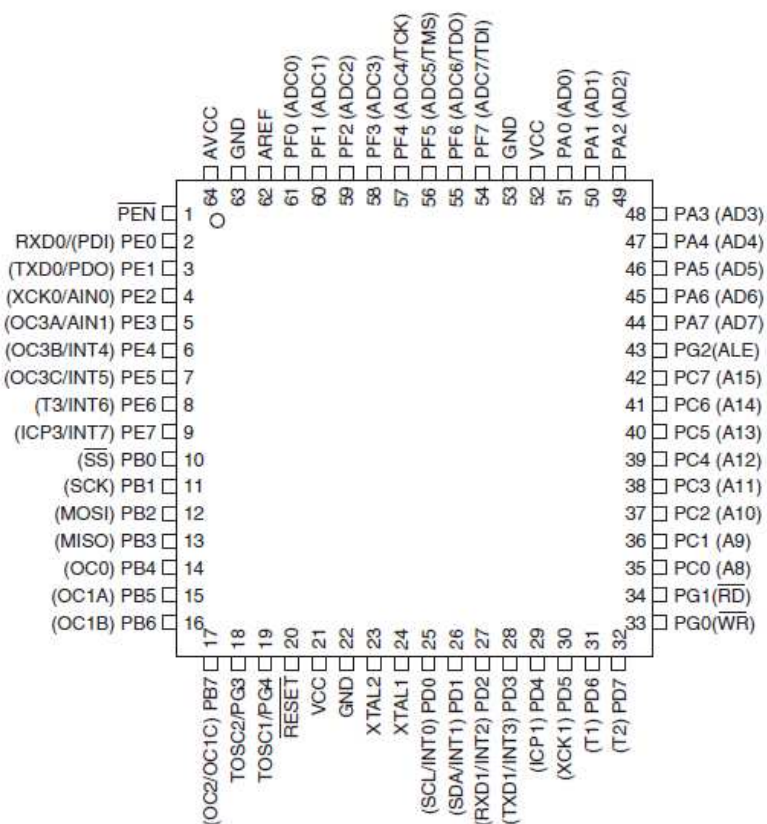
Tato práce rozebírá již existující implementaci propojení jednoduché kamery a mikroprocesoru Atmel AVR. Cílem je najít možnosti, jak efektivně implementovat základní algoritmy pro zpracování a vyhodnocování snímaného obrazu pomocí jednočipového mikroprocesoru. Rovněž ovšem použitím digitální kamery s přiměřenou rozlišovací schopností.

1 MIKROPROCESOR

Jakožto jádrem celé implementace byl zvolen osmibitový procesor od firmy AVR ATmega128, který bude mít za úkol přijímat a zpracovávat obrazová data z digitální kamery. Nejdůležitějšími kritérii pro volbu tohoto typu jsou schopnost práce s externí pamětí, dostatečný počet pracovních portů a komunikační sběrnice TWI. Pomocí mikroprocesoru bude taktéž možné měnit nastavení a vlastnosti kamerového systému jako je například rozlišovací schopnost, typ výstupních dat apod. Kompletní dokumentace pro ATmega128 je sepsána v datasheetu [3]. Tato kapitola popisuje obecné vlastnosti zvoleného typu MCU a jeho další využitelná rozhraní.

1.1 Obecné vlastnosti ATmega128

Jedná se o osmibitový mikroprocesor s RISC architekturou. Paměť je rozdělena samostatně pro data a zvláště pro program dle Harwardského přístupu. ATmega128 disponuje 133 využitelnými instrukcemi, jejichž přesný popis je k prostudování v [3]. Pracovat je možné až s 32 osmibitovými registry. Tyto programovatelné registry do šesti osmibitových portů a jsou označovány jako PORT A-F a pěti bitového portu G. Označení jednotlivých pinů portů pak vypadá takto PA2 (port A, pin 2). Rozložení těchto pinů je objasněno na obrázku.1.1.



Obr. 1.1: Rozložení pinů mikroprocesoru ATmega128 pouzdra TQFP64 (převzato z [3])

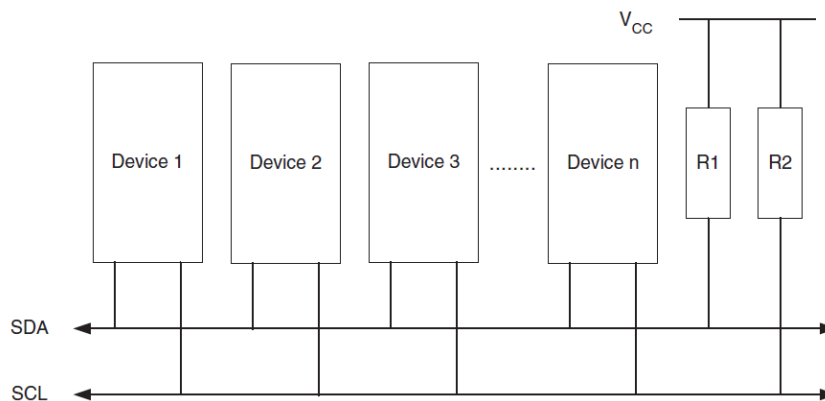
K účelu taktování MCU je možné zvolit mezi interním kalibrovaným oscilátorem s kmitočtem 1 až 8 MHz nebo použít externího krystalu. Paměť mikroprocesoru je dělena na 128kB programovatelné flash, 4kB interní SRAM a 4kB EEPROM paměti. Kontrolér podporuje připojení externí paměti přes vyhrazené adresní piny A0-A15.

Podporuje rozhraní JTAG pro přehlednější programovací a ladící fázi. Pro připojení periferních zařízení (EEPROM, AD/DA převodníky) je implementována sběrnice TWI, jejíž koncepce je shodná s I2C sběrnici.

Dále je možné využívat interních či externích přerušení MCU, což je reakce na příslušnou událost. K nejdůležitějším patří INTO a INT1. Jedná se o externí přerušení vyvolané vysokou úrovní na stanoveném pinu. Při záměru použití některého přerušení ve skriptu programu, musí být nejprve povoleno globální přerušení obecně a současně daný typ využívaného přerušení. Kompletní souhrn všech přerušení je k nalezení v [3].

1.2 Sériová sběrnice TWI (I2C)

Tato sběrnice je vhodná pro typické mikroprocesorové aplikace. V této práci je využita pro zprostředkování komunikace mezi digitální kamerou a MCU. I2C bude dále v textu označována taktéž jako TWI, protože firma Atmel tento typ komunikace označuje ve svých zařízeních takto. TWI protokol dovoluje připojit až 128 různých zařízení pouze dvou obousměrných vodičů sběrnice, jeden pro hodinový signál (SCL) a druhý pro data (SDA). Z důvodu vysoké úrovně v klidovém stavu jsou oba vodiče připojeny přes pull-up rezistory ke kladné svorce napětí viz. obrázek 1.2. Všechna zařízení připojená na sběrnici mají individuální sedmibitovou adresu, podle níž jsou rozpoznávány.



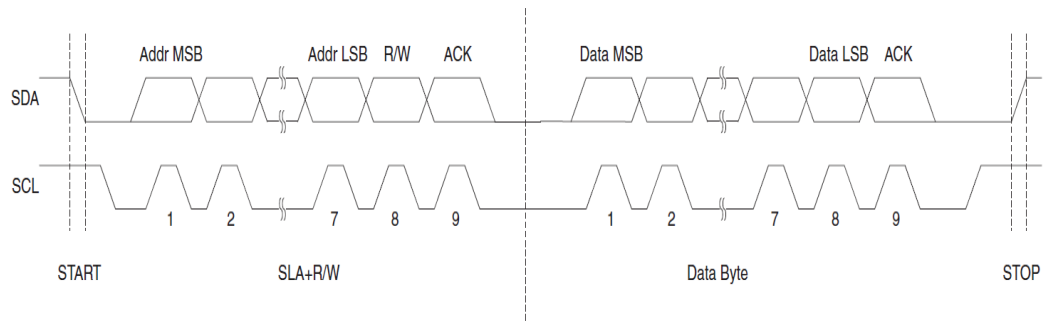
Obr. 1.2: Zapojení sběrnice TWI (převzato z [17])

Připojená zařízení se dělí na dva typy a to master a slave. Master je zařízení, které začíná a končí komunikaci. Master také generuje hodinový signál. Slave je zařízení adresované masterem. Dále je možné je dělit na přijímač a vysílač. Vysílač zapisuje data na sběrnici, zatímco přijímač je z ní čte. Používané frekvence SCL jsou 100 a 400kHz.

1.2.1 Systém komunikace TWI

Přenos každého bitu je řízen jedním pulzem hodinového signálu SCL. Úroveň datové

linky SDA musí být v okamžiku vysoké úrovně SCL konstantní. Výjimkou jsou generované impulsy START a STOP. Start i stop je signalizován změnou stavu datové linky při vysoké úrovni SCL jak je vidět na obrázku 1.3.



Obr. 1.3: Adresní a datový formát paketu TWI (převzato z [17])

Všechny datové pakety přenášené po I2C jsou devíti bitové viz. Tab. 1.1. Obsahují 1 byte dat a jeden potvrzovací bit (ACK), kterým odpovídá slave. Pokud SDA dosahuje kladné úrovně, slave hlásí nepotvrzený příjem NACK. Nejvýznamnější bit se zasílá jako první. Typická komunikace začíná událostí start, následuje adresa zařízení, R/W bit a ACK. Druhý již datový paket (či více paketů, vždy ukončených potvrzovacím bitem) a ukončení celé komunikace podmínkou stop.

Tab. 1.1: Stavba komunikačního rámce I2C

| Start | Adresa zařízení | R/W | ACK | Časová prodleva | Data | ACK | Stop |
|--------------|-----------------|-------|-------|-----------------|--------|-------|---------------|
| 1bit (změna) | 7 bitů | 1 bit | 1 bit | | 8 bitů | 1 bit | 1 bit (změna) |

1.2.2 Registry TWI

Bližší popis registrů ovládajících funkci sběrnice TWI.

TWBR - řídí periodu hodinového signálu a tím ovlivňuje přenosovou rychlost. Kmitočet hodinového signálu se vypočítá dle vzorce

$$SCL = \frac{CPU}{16+2*(TWBR)*4^{TWPS}} \quad (1.1)$$

kde CPU je taktovací frekvence procesoru v Hz, v TWBR se uvádí hodnota TWI bit rate registru a TWPS je nastavení před-děličky (podrobně v tabulce 1.3) z registru TWSR.

TWCR - slouží k řízení funkcí TWI, povoluje uplatnění START podmínky jako master komunikátor, generovat impuls pro potvrzení příjmu či STOP podmínku. Podrobný popis v tabulce 1.2.

Tab. 1.2: Funkce jednotlivých bitů registru TWCR

| Bit | Označení | Funkce |
|-----|----------|--|
| 7 | TWINT | Nastaví se na 1 pokud sběrnice dokončí aktuální úkol a očekává odezvu software aplikace. |
| 6 | TWEA | Generuje ACK bit. |
| 5 | TWSTA | Generuje START podmínku. |
| 4 | TWSTO | Generuje STOP podmínku. |
| 3 | TWWC | Nastavuje se na 1 při pokusu o zápis do TWDR, při hodnotě TWINT 0. Jinak nastaven na 0. |
| 2 | TWEN | Hodnota 1 aktivuje TWI rozhraní a přebírá kontrolu nad piny SCL a SCA. |
| 1 | | Rezervovaný bit, vždy 0 |
| 0 | TWIE | Povolení přerušování TWI. |

TWSR - Dělí se na 2 části. 5 bitů definuje stav rozhraní (bit 7-3) a 2 bity (bit1-0) jsou využity k nastavení před-děličky.

Tab. 1.3: TWPS - nastavení před-děličky

| Hodnota před-děličky | TWPS1 | TWPS0 |
|----------------------|-------|-------|
| 1 | 0 | 0 |
| 4 | 0 | 1 |
| 16 | 1 | 0 |
| 64 | 1 | 1 |

TWDR - Ve vysílacím módu obsahuje následující byte, který má být přenesen. V módu přijímacím obsahuje poslední přijatý byte.

TWAR - Bity 7-1 slouží k uložení adresy pokud mikroprocesor pracuje jako slave. Bit 0 slouží, jako povolení volání General Call.

1.3 Paměťová limitace MCU

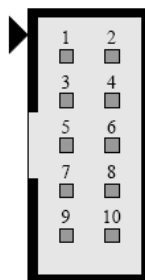
Uvažujeme-li standardní přednastavené rozlišení kamerového modulu 356x292 obrazových bodů, při 8-mi bitovém výstupním modelu celkem tak připadá na jeden snímek 103424 bitů (101 kB). Vzhledem k tomu, že MCU obsahuje paměť SRAM o kapacitě 4kB je nepřípustné toto rozlišení využít. K využití pro účely práce musí být celková velikost jednoho snímku menší než je paměť MCU z důvodu předpokládaných prací s obrazovými daty.

Proto by obrazová data měla být přiměřené velikosti, aby bylo možné je adresovat a zároveň s nimi pracovat v reálném čase. Uvažovaným vhodným rozlišením je např. 60x40 obrazových bodů odpovídající přibližně třetině rozlišení formátu QCIF, kde je třeba manipulovat s obrazem o velikosti 2400 B.

Ani po připojení nejvyšší možné hodnoty externí sram pro ATmega128 (64kB) není možné pracovat s plným rozlišením. Postupy v této práci dovolují pracovat s maximálním odhadovaným rozlišením, jehož celková velikost nepřekročí 60kB (pro kapacitu externí paměti 64kB). Doba odezvy mezi vyhledávacími cykly s velikostí paměti roste. Problémem není trvání uložení obrazu v paměti, ale jeho zpracování vyhledávacím algoritmem.

1.4 Propojení MCU a PC

Z důvodu absence sériových portů u novějších PC je místo dříve používaného MAX232 využít obvod FT232R od společnosti FTDI . Toto spojení se využívá při identifikaci, koordinaci desky s programovacím software a je taktéž využito jako komunikační kanál mezi MCU a PC. Pro programování samotné je využito rozhraní JTAG-ice 2x5 pinů pro programování a ladění přímo na desce. Zapojení pinů JTAG konektoru je zobrazeno na obrázku 1.4.



Obr. 1.4: Zapojení pinů konektoru JTAG (převzato z [6])

Tab. 1.4: Označení vývodů programovacího rozhraní JTAG

| Vývody | JTAG | Funkce |
|--------|------|---------------------------------------|
| 4,7 | +5V | Napájecí napětí programovacího kabelu |
| 2,10 | GND | Nulový potenciál |
| 1 | TCK | Vstup hodinového signálu |
| 3 | TDO | Sériový datový výstup |
| 9 | TDI | Sériový datový vstup |
| 5 | TMS | Řídící signál |
| 6 | RST\ | Resetovací signál |

2 DIGITÁLNÍ KAMEROVÝ MODUL C3088

Nejčastěji užívané obrazové senzory se v dnešní době dělí na dva typy a to CCD a CMOS. Zatímco CCD disponují velkou citlivostí a malým šumem, CMOS senzory se pyšní malou pořizovací cenou, větším rozsahem intenzit a mnohem menší spotřebou.

Tato kapitola se zabývá snímacím zařízením označovaným C0388. Je dělena na 3 části, první část rozebírá základní vlastnosti a skladbu modulu, druhá část se zabývá výstupními daty kamerového modulu a poslední část popisuje konfigurační registry modulu a jejich nastavení. Veškeré údaje o C0388 jsou čerpány z datasheetu modulu [4].

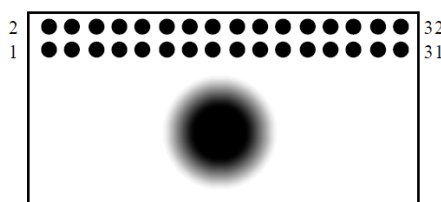
2.1 Základní vlastnosti

C0388 je 0,25" barevný kamerový modul s digitálním výstupem (viz obrázek 2.1). Využívá OmniVision CMOS obrazového senzoru. Kombinace CMOS technologie spolu s jednoduchým přístupem digitálního rozhraní v sobě snoubí dostatečnou kvalitu obrazu spolu s rozumnou cenovou relací. I přes velmi malé rozměry (40x28mm) a nízký odběr, který v aktivním režimu činí 80mW a v režimu Standby 30 μ W, disponuje modul dobrým rozlišením 356x292 pixelů až 60 snímků za sekundu. Poměr signál šum daného modulu činí 48dB. Zařízení je napájeno napětím 5 V stejnosměrného průběhu.



Obr. 2.1: Digitální kamera C0388 (převzato z [5])

Deska modulu má vyvedeno 32 funkčních pinů jejichž popis je uveden v tabulce 2.1. Rozmístění těchto pinů je dobře znázorněno na obrázku 2.2.



Obr. 2.2: Rozmístění pinů C0388

Výstupní data kamery je možné odebírat analogově i digitálně. Analogový výstup (75Ω monochromatický) přítomný na pinu 32 je možné přímo připojit ke kompozitnímu video vstupu. Digitální výstup využívá 16 pinů, 8 pro jasové (Y) a 8 pro chromatické (UV) složky. Modul umožňuje rozsáhlou volbu výstupních formátů videa. Konfigurace se provádí přes I2C rozhraní. Nastavování probíhá změnou stavů v 76 registrech C0388.

Tab. 2.1: Popis pinů digitální kamery C0388

| Pin | Označení | Funkce |
|---------|-----------|--|
| 1 - 8 | Y0 - Y7 | Digitální výstup jasové linky |
| 9 | PWDN | Power down mód |
| 10 | RST | Reset |
| 11 | SDA | Data I ² C |
| 12 | FODD | Indikace lichého snímku |
| 13 | SCL | Hodinový signál I ² C |
| 14 | HREF | Vysoká úroveň značí validní data na digitálním výstupu |
| 15 | AGND | Analogové uzemnění |
| 16 | VSYN | Výstup vertikálního synchronizačního signálu |
| 17 | AGND | Analogové uzemnění |
| 18 | PCLK | Hodinový signál pro určení pixelů na Y a UV linkách |
| 19 | EXCLK | Vstup pro externí hodinový signál |
| 20 | VCC | Napájecí napětí 5V |
| 21 | AGND | Analogové uzemnění |
| 22 | VCC | Napájecí napětí 5V |
| 23 - 30 | UV0 - UV7 | Digitální výstup chromatické linky UV |
| 31 | GND | Uzemnění |
| 32 | VTO | Analogový výstup |

Konfigurováním registrů modulu je možné využívat a měnit nastavení mnoha různých funkcí jako separace barvy, automatické nastavení citlivosti (AGC), gama korekce, korekce barvy a nastavení obrazového luminiscenčního, chromatického a anti-aliasingového filtrování. Rozlišení je pevně nastaveno na 356x292 pixelů, ale lze jej libovolně měnit od 2x2 do 356x292 pixelů. Nejpoužívanějšími a jedinými módy nastavitelnými přímo z registru jsou CIF (352x288) a QCIF (176x144). Modul je taktován vlastním krystalem o kmitočtu 17,7344MHz, ten je možné odpojit a nahradit ho přivedením externího hodinového signálu na pin EXCLK. Pro úsporu energie je možné kameru přivést do režimu spánku přivedením aktivní úrovně na pin PWDN. Odběr senzoru se tím sníží z 80mW na pouhých 10μA, při tomto jsou všechny interní registry resetovány (kromě I²C registrů).

2.2 Výstupní formáty digitálních dat

Digitální video výstup čítá několik formátů dat z důvodu pokrytí různých aplikačních požadavků. S těch nejpoužívanějších barevných modelů jsou to 4:2:2 YUV, obsahující jasovou složku Y a barevné složky UV. Či formát RGB, skládající se z hodnot červené, zelené a modré barvy. Mezi soustavami YUV a RGB je možný přepočít dle vzorců:

$$Y = 0,59G + 0,31R + 0,11B \quad (2.1)$$

$$U = R - Y \quad (2.2)$$

$$V = B - Y \quad (2.3)$$

Tyto modely jsou programovatelné přes rozhraní I²C. Dále je možné pracovat v 16-ti (viz tabulka 2.2), 8-mi (viz tabulka 2.3) nebo 4 (nibble) bitovém módu a také zvláště pro formát 704x288. Dle zvoleného módu a formátu je pak možné určit a zmapovat výstupní data modulu. Podrobnější popis v [4].

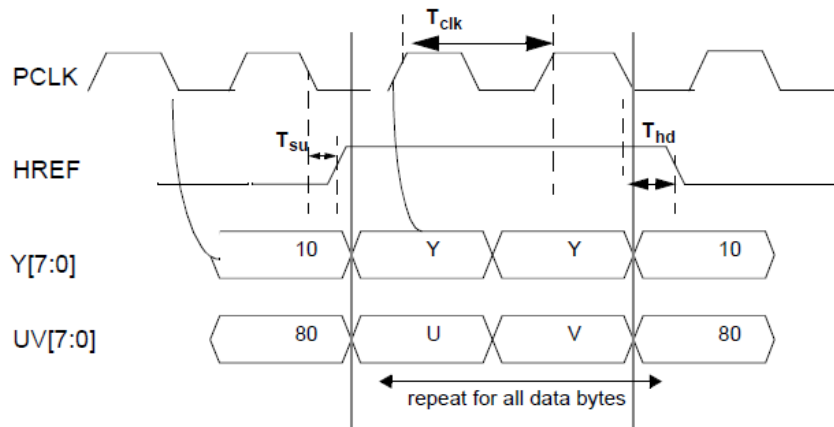
Tab. 2.2: 4:2:2 formát 16 bitů

| Piny | Datová sekvence pixelů | | | | | |
|------------|------------------------|----|----|----|----|----|
| 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 rámeček | 0 | 1 | 2 | 3 | 4 | 5 |
| UV rámeček | 0 | | 2 | | 4 | |

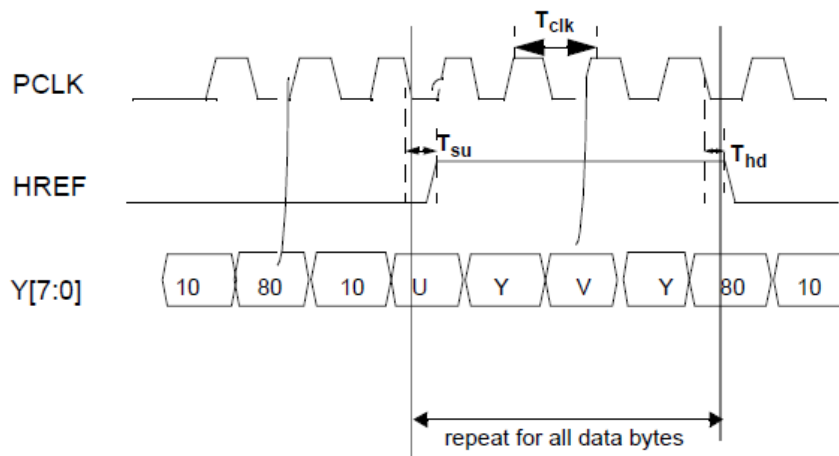
Tab. 2.3: 4:2:2 formát 8 bitů

| Piny | Datová sekvence pixelů | | | | | | | |
|------------|------------------------|----|----|----|----|----|----|----|
| 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 rámeček | 0 | | 1 | | 2 | | 3 | |
| UV rámeček | 0 | | | | 2 | | | |

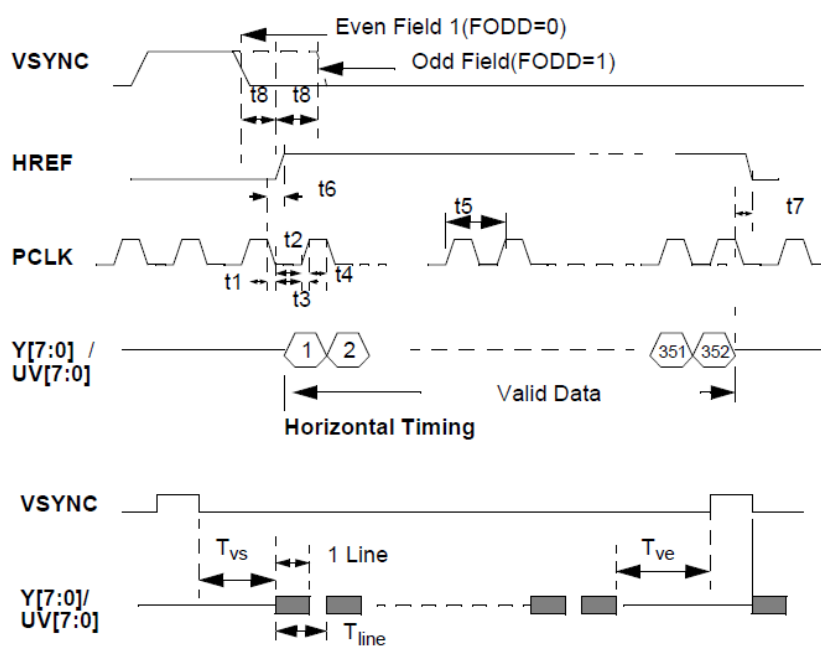
Každý následující snímek je možné pozorovat indikací na pinu VSYNC. Jak ukazují časové průběhy signálů na obrázcích 2.3 a 2.4, o hodnotě bitu výstupního signálu rozhoduje sestupná (případně nástupná dle nastavení) hrana hodinového signálu PCLK. Jestli jsou data obrazová či jiného charakteru řídí úroveň signálu HREF, kterou je taktéž možné nastavit. Obrázky jsou uvedeny jen pro YUV zobrazení. Pro zobrazení RGB by signály vypadaly stejně s malou změnou. Pro 16-ti bitové časování by místo Y přicházely data G R a místo U V data B G.



Obr. 2.3: Data pixelů při 16-ti bitovém časování (převzato z [4])



Obr. 2.4: Data pixelů při 8-mi bitovém časování (převzato z [4])



Obr. 2.5: Výstupní signály C3088 (převzato z [4])

Při použití interního krystalu 17,7344MHz odpovídá kmitočet hodinového signálu PCLK pro 16-ti bitové časování polovině, tedy 8,8672MHz. Perioda jednoho pulzu je tak 112,8ns. Pro 8-mi bitové časování je zůstává kmitočet hodinového signálu 17,7344MHz a perioda jednoho pulzu tedy poloviční než u 16-ti bitů a to 56,39ns. Tyto údaje jsou taktéž nastavitelné. Náběžná i sestupná hrana signálu HREF musí přijít do 15-ti ns po předchozí hraně PCLK.

Ve výstupní sekvenci dat odpovídá Y7 nejvíce důležitému bitu (MSB) a Y0 nejméně důležitému bitu (LSB). Tuto posloupnost je možné úpravou hodnot registrů negovat.

2.3 Nastavení konfiguračních registrů

Jak bylo poznamenáno výše v textu, vlastnosti modulu se dají rozmanitě měnit pomocí 81 konfiguračních registrů prostřednictvím příkazů sběrnice I²C. Modul se připojí jako slave, jeho adresa je již stanovena součástkami daného obvodu. V tomto případě C0 pro čtení a C1 pro zápis dat.

Při komunikaci přes I²C musí master generovat Start/stop podmínku, hodinový signál na SCL. Samotný komunikační rámec se poté skládá z adresy cílového zařízení, příkazu read/write, adresy registru, nové hodnoty registru a několika potvrzovacích bitů, tak jak ukazuje tabulka 2.4.

Tab. 2.4: Konfigurační rámec registru I²C

| Start | Adresa | R/W | ACK | Adresa registru | ACK | Data | ACK | Stop |
|---------------|--------|-------|-------|-----------------|-------|--------|-------|---------------|
| 1 bit (hrana) | 7 bitů | 1 bit | 1 bit | 8 bitů | 1 bit | 8 bitů | 1 bit | 1 bit (hrana) |

Registry jsou číslovány hexadecimálně a to od hodnoty 00 až 50. Jejich kompletní přehled je dostupný v [4]. Tabulka 2.5 popisuje ukázkou některých registrů modulu.

Tab. 2.5: Výpis vybraných registrů modulu

| Adresa registru | Označení | Přednastavená hodnota | Popis |
|-----------------|-----------|-----------------------|---|
| 0x05 | Cnt | 0x48 | Vyšší hodnota -> vyšší kontrast |
| 0x06 | Brt | 0x80 | Vyšší hodnota -> vyšší jas |
| 0x07 | Sharpness | 0xC6 | Změna ostrosti obrazu |
| 0x10 | AEC | 0x9A | Automatické řízení expozice |
| 0x11 | CLKRC | 0x00 | Polarita signálu CLK a nastavení předděličky |
| 0x12 | COMA | 0x24 | Formát výstupních dat a další úpravy |
| 0x13 | COMB | 0x01 | Volba n-bitového modelu |
| 0x14 | COMC | 0x00 | Změna rozlišení QCIF a parametry signálu HREF |
| 0x3D | COMN | 0x08 | Povolení analogového výstupu |

3 REALIZACE UKÁZKOVÉ APLIKACE

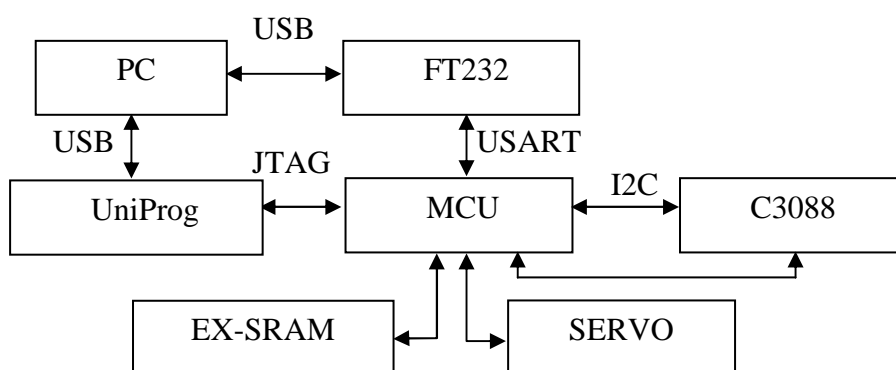
Tato kapitola se zabývá vlastním řešením obvodu. Realizace spočívá v sestavení dle schématu zapojení (viz. příloha A) na nepájivém poli. Tak aby bylo možné uskutečnit komunikaci s kamerovým modulem po sběrnici I2C pro práci s registry kamery, odebírat výstupní data modulu, komunikovat s PC a ovládat servomotory. Dále zahrnuje popis aplikovaných algoritmů v rámci řešené aplikace, potřebného programového vybavení a jeho nastavení.

3.1 Konstrukce zařízení

Podkapitola obsahuje informace týkající se konstrukce zapojení. Seznam použitých součástek v zapojení je uveden v příloze B. Fotografie výsledného zapojení jsou uvedeny v příloze C.

3.1.1 Návrh obvodu

Rozložení komunikačních rozhraní mezi individuálními bloky zapojení je znázorněno v blokovém schématu na obrázku 3.1. Podrobné schéma, které bylo vytvořeno v programu Eagle je k dispozici v příloze A. Nepopsané kontakty jsou přímé spoje tvořené n-žilovými izolovanými dráty.

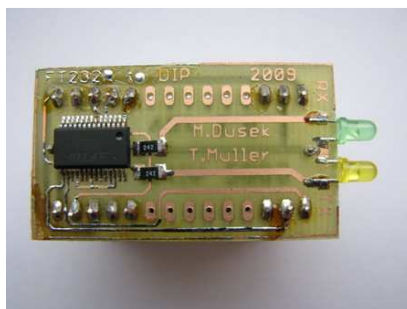


Obr. 3.1: Blokové schéma zapojení

K samotnému programování procesoru je použit programátor UniProg od firmy PK-Design [6] využívající rozhraní JTAG. Díky tomu je možné ladit aplikaci přímo za chodu v debug módu. Podrobné informace o propojení MCU a PC jsou uvedeny v kapitole 1.4.

3.1.2 Převodník USB/USART

K propojení PC a MCU po stránce komunikační byl využit zapůjčený KIT FT235 (obrázek 3.2) z předmětu MMIA . K němuž jsou přístupné všechny informace v [8]. Tento obvod plní funkci převodníku USB/USART. Zapojení napájení FT232 je možné vyřešit mnoha cestami zde bylo provedeno ve stylu vlastního napájení. Napájecí vodič rozhraní USB je přes odporový dělič přiveden na vstup FT232-RST a poté na zem (viz. schéma obvodu - příloha A).



Obr. 3.2: Převodník USB/USART (převzato z [8])

3.1.3 Servomotory

Pro pohyb kamery do stanovené polohy jsou použity dvě modelářská mikroserva typu Hitec HS-422 (obrázek 3.3). Jsou využity dvě serva tak, aby bylo možné nastavení pohybu ve vertikální i horizontální rovině. K nastavení pozice servomotorů je využit upravený skript, který vznikl za účelem projektu do předmětu MMIA na ústavu radioelektroniky spolu s napájecími obvody (viz. [13]).

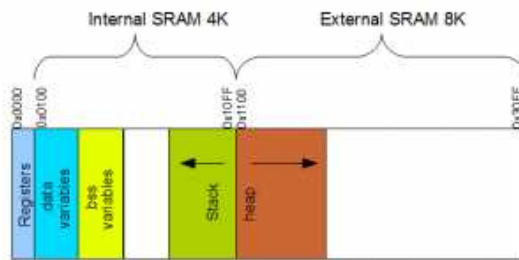


Obr. 3.3: Serva Hitec HS-422 (převzato z [13])

3.1.4 Externí SRAM

Interní datová paměť mikrokontroléru je pro zpracovávání přehlednějších obrazů nedostatečná. Podrobněji v kapitole 1.3. Aby bylo možné pracovat s vyšším rozlišením snímaného obrazu byla připojena a alokována externí paměť typu SRAM o velikosti 32k x 8bitů. Detailní informace k použité paměti jsou k nahlédnutí v datasheetu [15]. V případě potřeby může být zvětšena neboť ATmega128 je limitována schopností přijmout paměť maximální velikosti 64kB.

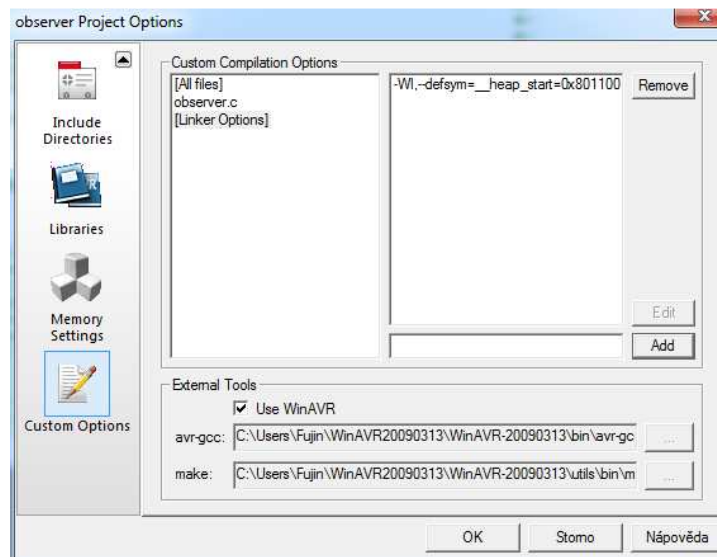
Pro správnou a zajištěnou funkci externí SRAM je mimo nastavení bitu SRE v registru MCUCR a XMCRA nutné v překladači nastavit informace o velikosti paměti a způsobu rozvržení. To je možné dvěma cestami. Externí paměť může obsahovat proměnné (.data/.bss) a heap paměť nebo budou proměnné ukládány v interní SRAM a externí zůstane vyhrazena pro heap paměť.



Obr. 3.4: Rozvržení paměti (převzato z [14])

V tomto projektu je využita druhá možnost. Tento typ rozvržení je rozkreslen na obrázku 3.4. Princip je shodný pro jakoukoli velikost, změna je pouze v hodnotě konečné adresy pro heap. V programu AVR studio je třeba otevřít nastavení Configuration options, v záložce Custom options vybrat Linker Options (viz. obrázek 3.5) a přidat řádek :

"-Wl,--defsym=__heap_start=0x801100,--defsym=__heap_end=0x8090ff".



Obr. 3.5: Nastavení paměti v překladači

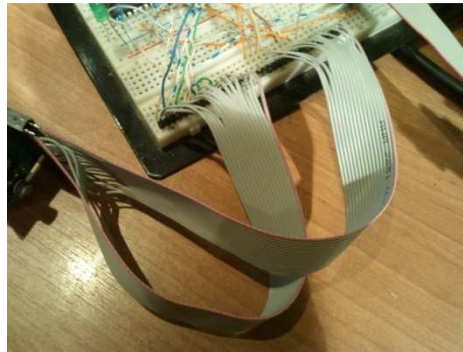
K připojení externí paměti je nezbytné zapojit před adresní piny (AD0-AD7) záchytnou paměť LATCH. Protože paměťový řadič používá multiplex sběrnice, kdy se na jedné sběrnici postupně objeví adresa i data k zápisu. Datasheet použitého obvodu je ke shlédnutí v [16].

3.1.5 Napájení

K napájení zařízení je možné užít transformátor s výstupním ss. napětím 6,5V nebo obyčejnou 9V baterii. Vstupní napětí je stabilizováno na hodnotu VCC = 5V prostřednictvím integrovaného obvodu 7805. V případě napájení baterií může dojít po 1-2 hodinách provozu k poklesu napětí VCC a tak znemožnit správnou funkci některých obvodů zvláště FT232.

3.1.6 Konektory periférií

Z přípravku je vyveden kabel USB A/Male pro sériovou komunikaci a konektor JTAG k programování MCU. Pro propojení pole a kamery byl vyroben kabel 2x16 žil se soklem pro piny C3088 a žebříčkem pinů pro zasunutí do pole (viz. obrázek 3.6). Zbylé spoje jsou provedeny jednožilovými vodiči.



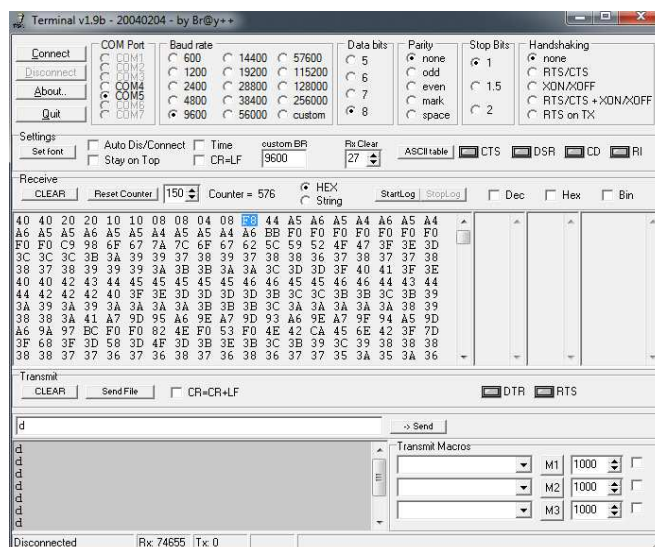
Obr. 3.6: Kabel 2x16

3.2 Software

Tato podkapitola přibližuje programovou část zařízení a součásti potřebné k jejímu vývoji.

3.2.1 Sériová komunikace

Aby bylo možné vzájemně komunikovat s MCU po sériové lince využitím rozhraní USB, musel být vytvořen virtuální COM port pomocí ovladačů od firmy FTDI získaných z [10]. Virtuální port se chová, jako obyčejný sériový port. Pro manipulaci dat na tomto portu byl využit freeware Terminal 1.9b, který byl stažen z [11]. Prostředí programu je představeno na obrázku 3.7.



Obr. 3.7: Terminal 1.9b

3.2.2 Vývojové prostředí programu

Pro tvorbu software bylo využito AVR studio 4.14 build-589 (vývojové prostředí od firmy Atmel), které je spolu s kompilátorem AVR GCC z balíčku WinAVR volně ke stažení z webové stránky (<http://www.mikrocontroller.net/articles/AVR-Studio>). Společně fungují, jako veškerý software potřebný k tvorbě programu v syntaxi jazyka C pro MCU z rodiny AVR, který může být následovně naprogramován do mikrokontroléru.

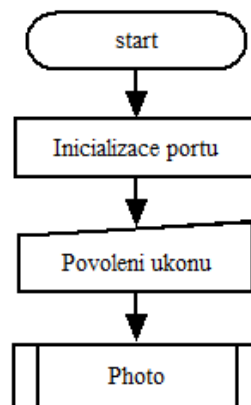
Prostřednictvím AVR studia je možné měnit základní nastavení (fuses) mikrokontroléru a nahrát do něj zhotovený program. Studio je také vybaveno nástrojem debugger, pomocí kterého je možné krokovat aplikaci v chodu a tím ji i ladit.

Po zprovoznění AVR studia, volbě mikrokontroléru a připojení programátoru ve správném módu bylo otestováno zda se na poli nachází výše zvolený typ mikrokontroléru vyčtením signatury ze známé lokace.

3.2.3 Vlastní programy pro MCU

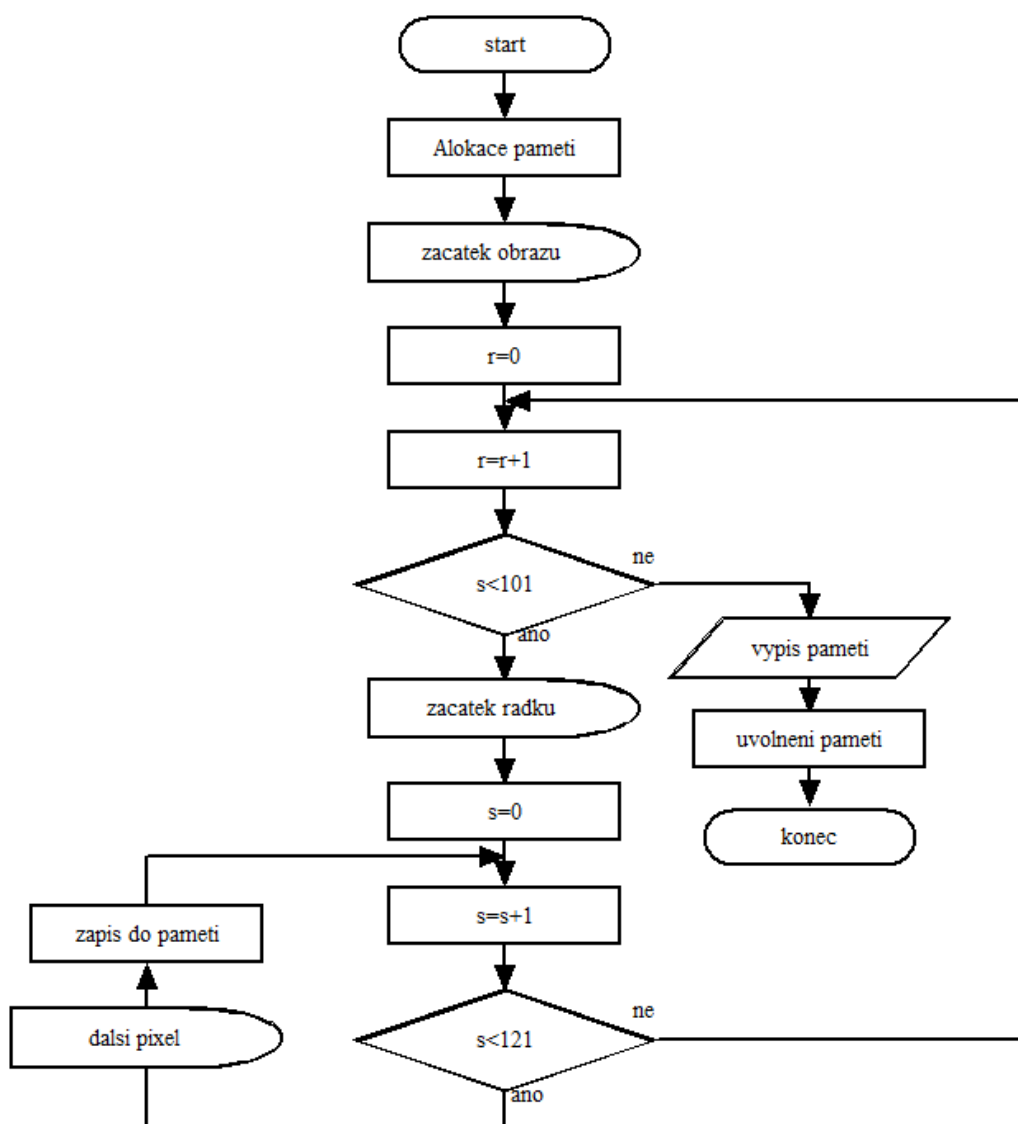
Na přiloženém nosiči jsou k dispozici všechny využité vlastní i upravené programy z uvedených zdrojů s komentářem.

K zobrazení výstupních dat kamery je implementován program *imagegain*. Skrz něj jsou výstupní obrazová data nejprve uložena v paměti ve formě hexadecimálního vektoru a následovně bod po bodu zapisovány na port. Data jsou zapisována systémem vyčítání po řádcích, které k validnímu chodu programu vyžaduje nižší hodinový signál kamery než vyčítání po sloupcích, ale z důvodu potřeby sloupcového čtení přijmout tolik obrazů kolik je aktuálně nastaveno sloupců v jednom obrazu, je systém řádkového vyčítání časově mnohem hospodárnější. Způsob jakým čtení pracuje je přiblížen vývojovými diagramy. Na obrázku 3.4 je hlavní cyklus. Jehož výstup je patrný z obrázku 3.8.



Obr. 3.8: Výpis jednoho kompletního snímku z výstupu kamery - program Imagegain

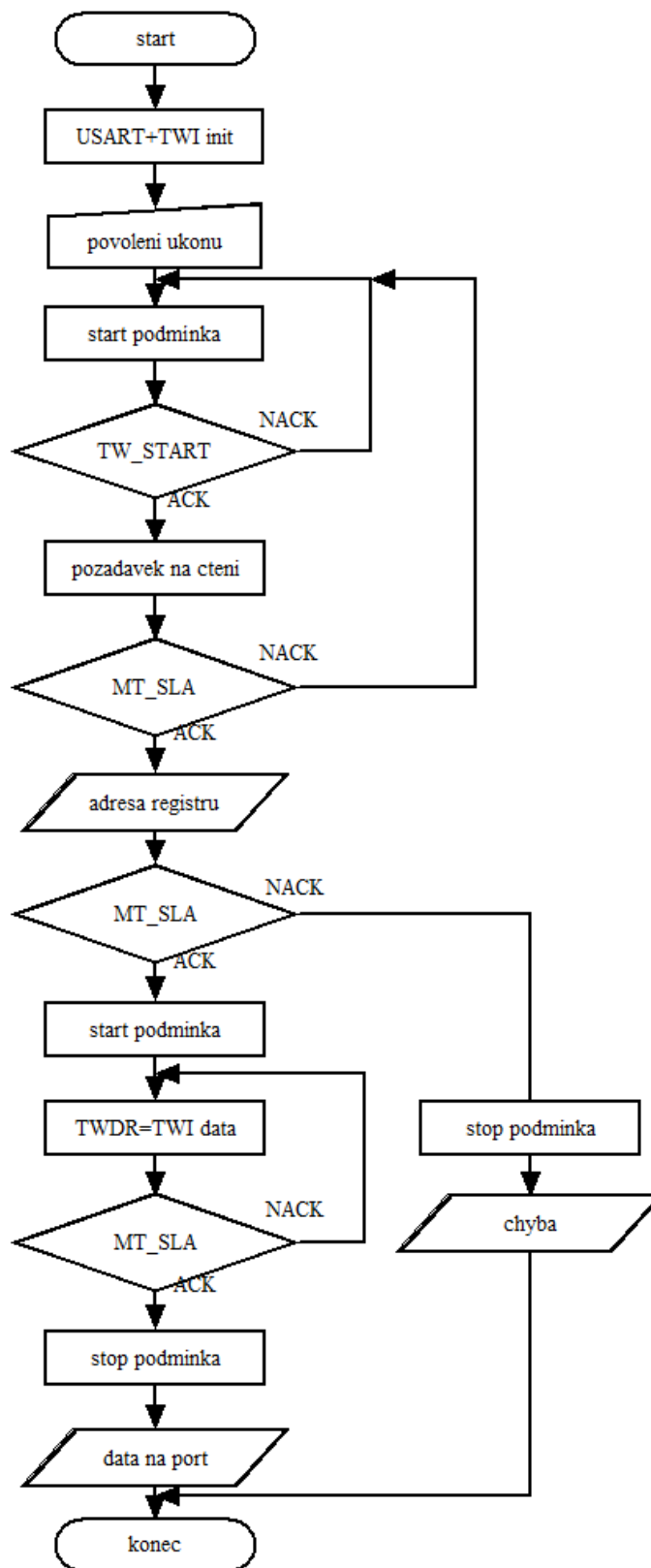
Podprogram Photo je podrobně rozebrán na vývojovém diagramu (viz. obrázek 3.9).



Obr. 3.9: Popis podprogramu Photo

Program pro ovládání servomotorů byl převzat z [13] a upraven. Úprava spočívala v negaci polarity některých řídicích signálů a úprav konstant pro ATmega128. Jádrem skriptu je externí přerušení, které je generováno prostřednictvím čítače/časovače 1 v režimu CTC. Tento způsob umožňuje přesné nastavení servomotoru ovládajícího horizontální pohyb od 0 do 180 stupňů a servomotoru ovládajícího pohyb vertikální od 0 do 90 stupňů.

Pro čtení a změnu hodnot registrů EEPROM kamery je používán program *regwork*. Na čtení registrů je použit algoritmus, který je popsán vývojovým diagramem na obrázku 3.10. Vývojový diagram pro zápis registru je podobný a proto zde nebude uveden.



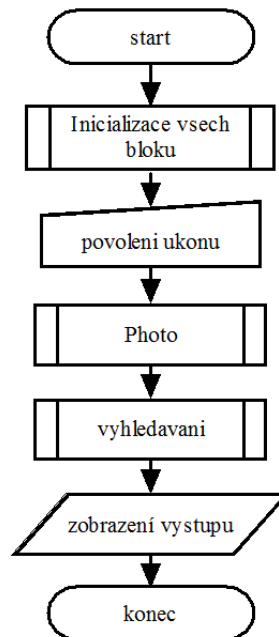
Obr. 3.10: Čtení hodnoty registru

Tato procedura vypíše hodnotu daných registrů po změně na port a ty jsou poté zobrazeny prostřednictvím terminálu. Příklad výsledku pro změnu hodnot registrů 0x17-0x1A (rozlišení), 0x11(PCLK) a 0x14 (QCIF formát) je na obrázku 3.11.

| Register Address | Register Value | Binary Representation |
|------------------|----------------|-----------------------|
| 88 | E0 | 10001000 |
| 48 | 90 | 11100000 |
| 15 | 24 | 01001000 |
| 20 | | 10010000 |
| 136 | | 00010101 |
| 224 | E0 | 00100100 |
| 72 | 48 | 00100100 |
| 144 | 90 | 00100100 |
| 21 | 15 | 00100100 |
| 36 | 24 | 00100100 |
| 32 | 20 | 00100000 |

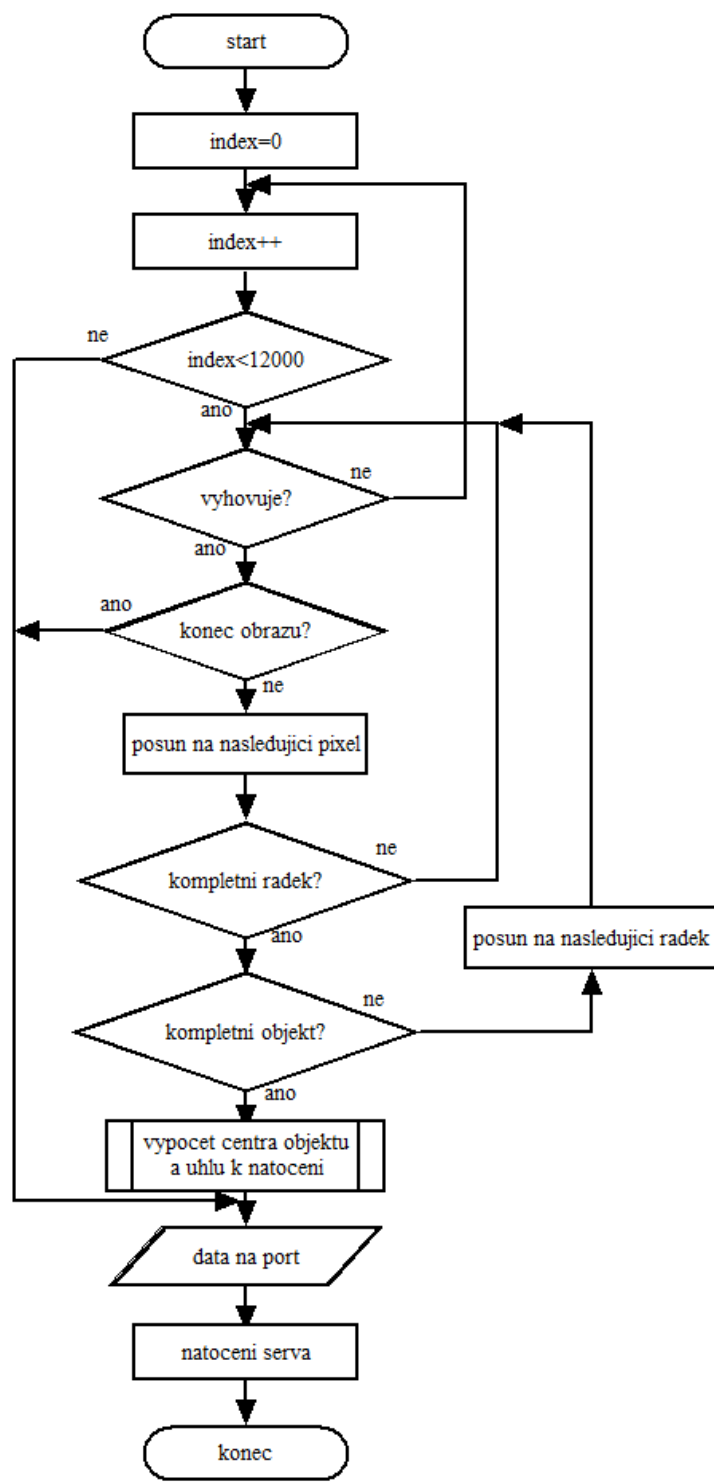
Obr. 3.11: Výpis hodnot daných registrů po změně

Výsledným programem je sjednocení několika zmíněných procedur, kde po příkazu k aktivitě dojde k inicializaci datových portů, řídicích signálů kamery, servomotorů, USARTU i externí paměti a její alokaci. Následuje funkce uložení pořízené fotografie v paměti a poté kontrolní fáze, kde algoritmus rozhoduje nachází-li se v obrazu předem definovaný objekt. Tento program má název *observer* a jeho struktura je znázorněna v obrázku 3.12.



Obr. 3.12: Hlavní program - observer

Samotná rutina pro vyhledávání předem definovaného objektu, je naznačena ve vývojovém diagramu na obrázku 3.13. V této rutině jsou kontrolovány chronologicky všechny pixely, jakožto potenciální počátky hledaného objektu, tato kontrola je postupně hloubena a pokud se prokáže irelevance zkoumaného počátku je postoupeno na pixel následující. V případě, že rutina nalezne předepsaný objekt, následuje výpočet středu objektu. Z pozice středu jsou vypočteny odchylky vertikální a horizontální od aktuálního středu kamerového pohledu a z nich vypočteny úhly pro natočení servomotorů v příslušných směrech.

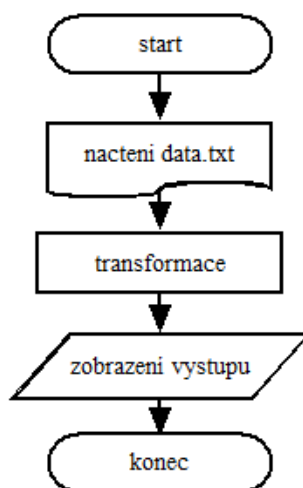


Obr. 3.13: Kontrolní fáze

3.2.4 Matlab

Z důvodu kontroly obrazového výstupu dat, jsou po proceduře čtení data v podobě hexadecimálních hodnot každého obrazového bodu seřazeny bod po bodu postupem uvedeným pomocí výše zmíněných vývojových diagramů. Záznam dat je uložen do souboru s data.txt a je dále zpracováván pomocí matematického programu Matlab.

Manipulaci s daty v aplikaci Matlab popisuje obrázek 3.14.



Obr. 3.14: Převod textového souboru na obraz

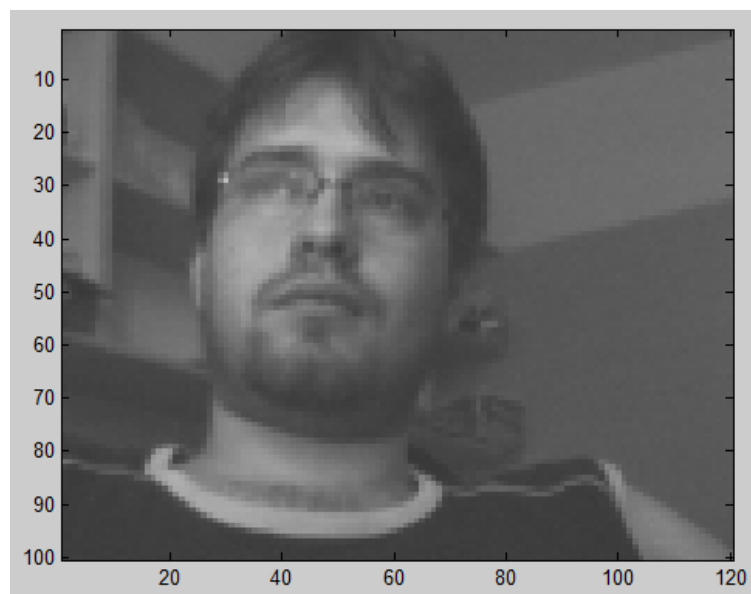
Do proměnné datamatrix jsou načtena data zachycená programem Terminal a jsou zpracována v matici hexadecimálních hodnot zadaných rozměrů. V případě řádkového vyčítání je nutné v této fázi matici transponovat. Pomocí funkce colormap je definována barevná paleta, která každé hodnotě jednotlivých bodů v matici přidělí příslušnou obrazovou reprezentaci. Takto transformovaná matice je finálně reprodukována funkcí image. Výsledek této transformace je zachycen na obrázku 3.16. Paměťové vytížení finálního programu pro mikrokontrolér popisuje obrázek 3.15

```
AVR Memory Usage
-----
Device: atmega128

Program: 13148 bytes (10.0% Full)
(.text + .data + .bootloader)

Data: 12301 bytes (300.3% Full)
(.data + .bss + .noinit)
```

Obr. 3.15: Výpis AVR studia o využití paměti

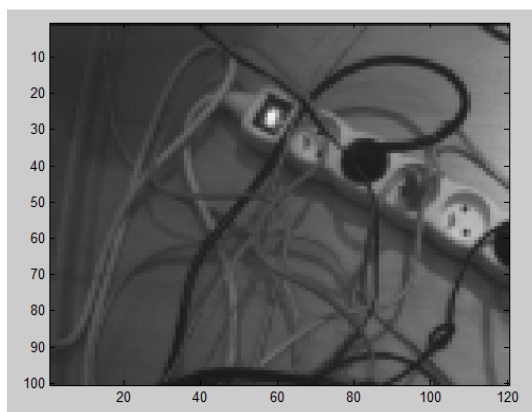


Obr. 3.16: Výsledná obrazová reprodukce pro jasovou složku módu YUV v rozlišení 120x100

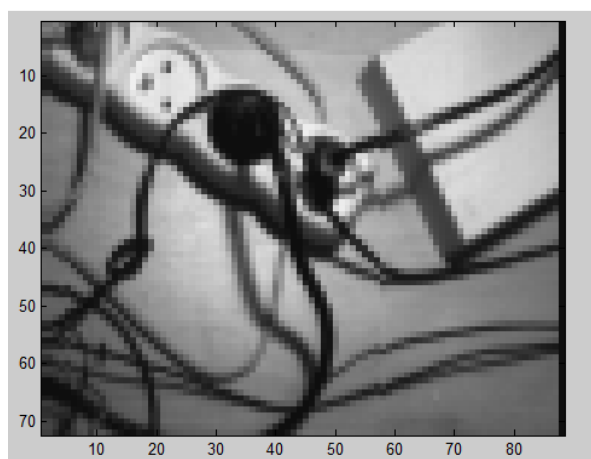
4 DOSAŽENÉ VÝSLEDKY

4.1 Obrazová reprezentace

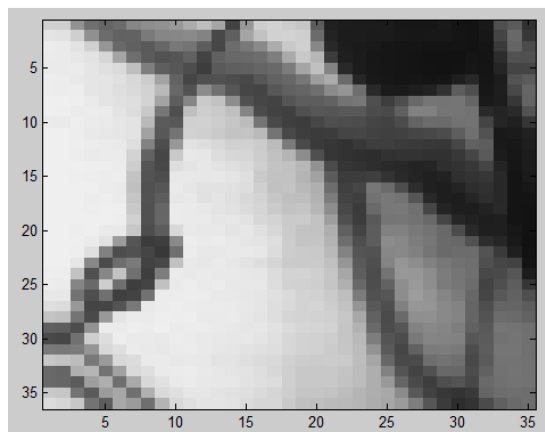
Použitým módem pro získání výstupních dat modulu byl YUV. Z důvodu přítomnosti jasové složky v jeho barevném modelu. Rozlišení kamery je měnitelné pomocí registrů 0x17-0x1A, kde se přímo nastaví hranice obrazového rámce. Na obrázcích 4.1-4.3 je dokázáno, že změna rozlišení je pouhý výřez z celkového obrazu.



Obr. 4.1: Obraz při rozlišení 120x100 obrazových bodů



Obr. 4.2: Obraz při rozlišení 88x72 obrazových bodů



Obr. 4.3: Obraz při rozlišení 44x36 obrazových bodů

Data se podařilo získat způsobem vertikálního (sloupcového) i horizontálního (řádkového) čtení. V případě sloupcového jsou pixely čteny po sloupcích a na jeden kompletní obrázek je tedy potřeba přijmout stejný počet celých obrázků kolik je sloupců v nastaveném rozlišení. To činí čtení větších rozlišení velmi zdlouhavou záležitostí. Řádkové čtení je klasickým přístupem k obrazu vždy kompletní řádek a po něm následující. Doba uložení jednoho obrázku je tedy:

$$V_p = \frac{D}{t_p}, \quad [\text{bps}; \text{bit}, \text{s}] \quad (4.1)$$

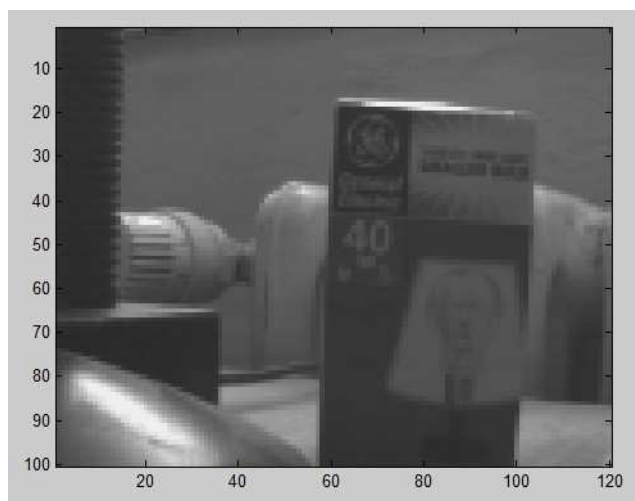
kde V_p je přenosová rychlost, t_p doba přesunu kompletního snímku do paměti a D objem transferovaných dat.

Jaký má vliv změna kmitočtu PCLK modulu na přenosovou rychlost popisuje tabulka 4.1. Hodnoty byly měřeny pro rozlišení obrazu 120x100 pixelů pro řádkové čtení využitě v projektu.

Tab. 4.1: Závislost přenosové rychlosti na taktovacím kmitočtu PCLK - řádkové čtení

| Hodnota registru 0x11 | PCLK [kHz] | t_p [s] | V_p [kbps] |
|-----------------------|------------|-----------|--------------|
| 0x3F | 69,26 | 1,21 | 77,48 |
| 0x30 | 90,46 | 0,802 | 116,9 |
| 0x20 | 134,3 | 0,660 | 142,0 |
| 0x10 | 260,8 | 0,356 | 263,3 |
| 0x00 | 4433 | <0,050 | >1875 |

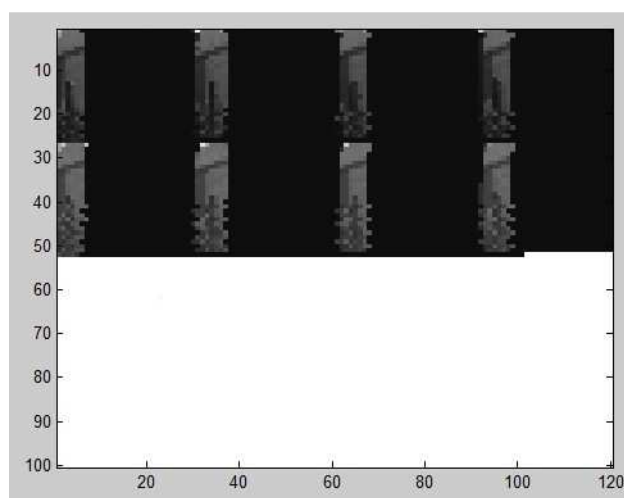
Na obrázcích 4.4-4.6 je znatelné jaký vliv má změna PCLK na výstupní obrazovou kvalitu při rozlišení 120x100 obrazových bodů řádkového čtení. Kvalita zůstává přiměřeně zachována do kmitočtu 200kHz, u vyšších kmitočtů je ztrátovost průkazná.



Obr. 4.4: Obraz při PCLK 69,25 kHz



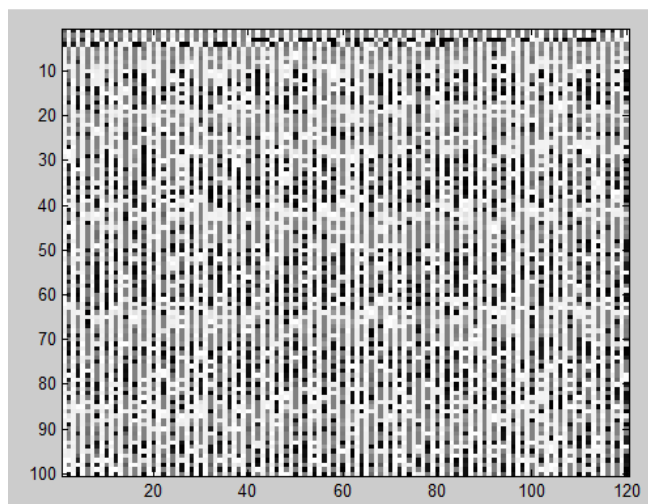
Obr. 4.5: Obraz při PCLK 260,8 kHz



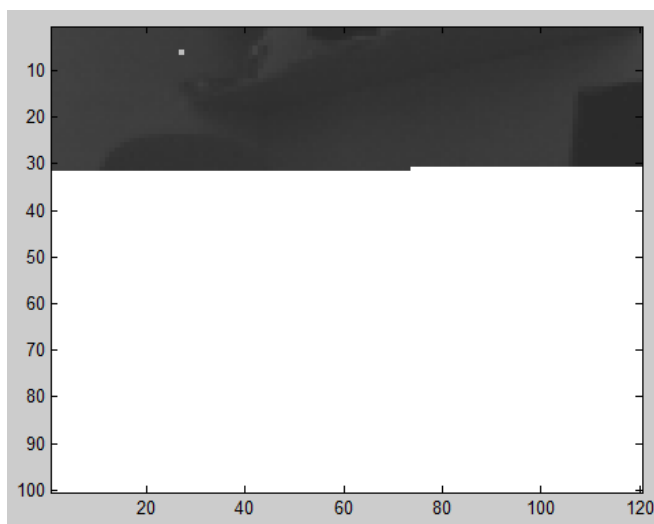
Obr. 4.6: Obraz při PCLK 443,3 kHz

4.1.1 Chybné obrazové reprezentace

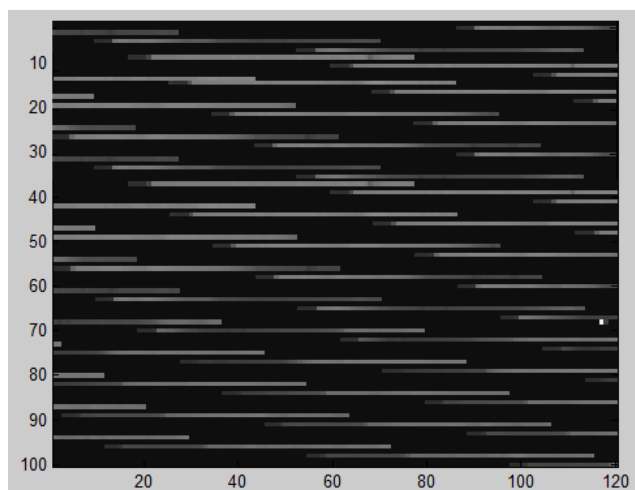
Podkapitola poukazuje na různé typy nesprávně rekonstruovaných obrazů a uvádí jejich příčiny.



Obr. 4.7: Frekvence se neshoduje s použitým krystalem



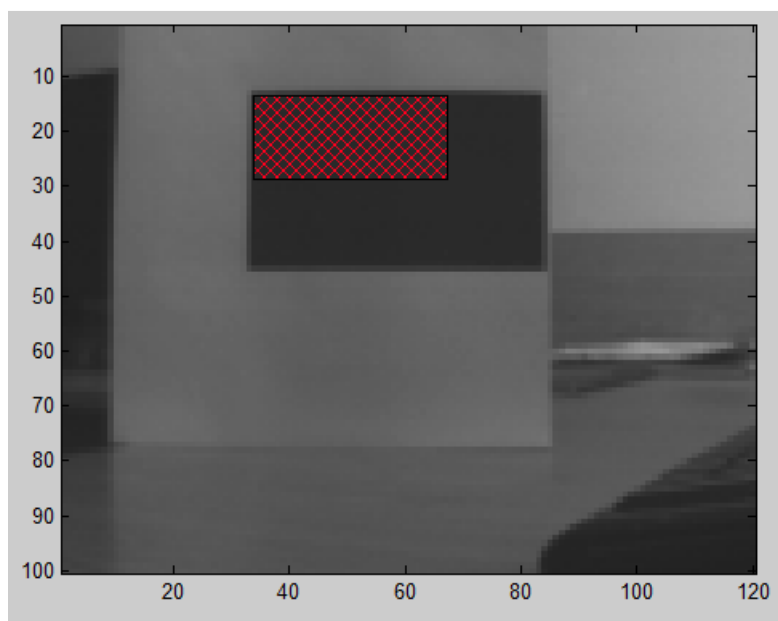
Obr. 4.8: Příliš vysoký kmitočet PCLK



Obr. 4.9: Špatná alokace paměti

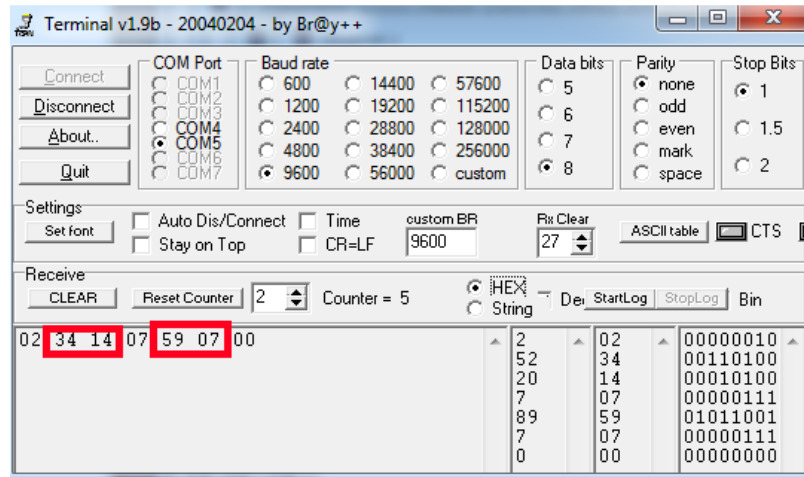
4.2 Vyhledávání objektu

Finální program s názvem *observer* je schopen nalézt předem definovaný objekt a na svém výstupu udává pozici středu nalezeného objektu v obraze z pohledu kamerového senzoru a vycentrováním senzoru na nalezený objekt. Příklad pozitivního nálezu je uveden na obrázku 4.11, který byl pořízen při rozboru snímaného pohledu 4.10. Na němž je ukázkově vyznačen (červená vyšrafovaná oblast) nalezený objekt odpovídající výstupním údajům získaných z programu Terminal.



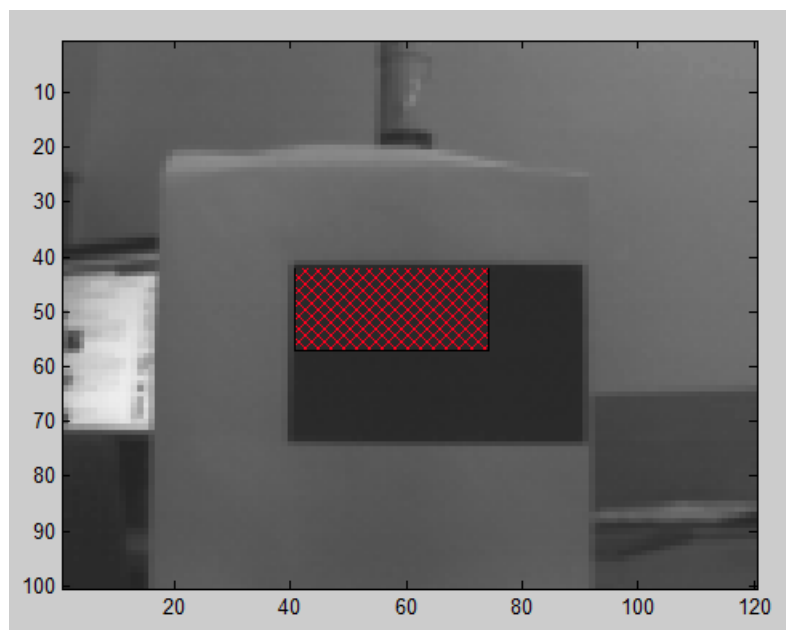
Obr. 4.10: uložený obraz s vyznačeným nalezeným objektem

Z výstupu terminálu můžeme číst pozitivní nález objektu. Tuto informaci podává přítomnost znaku 2, za nímž následují údaje o vypočteném středu. Prvním údajem je horizontál (sloupec) a druhým vertikál (řádek). Znak 7 přítomný bezprostředně po horizontálu a vertikálu nese informaci o tom, že následující dvě hodnoty udávají úhly natočení servomotorů. První údaj udává horizontální a druhý vertikální pozici servomotoru. V případě negativního nálezu se v terminálu vždy po skenování kompletního obrazu vypíše pouze úhly re-centrování senzoru. Znak 0 na konci oznamuje uvolnění bloku paměti heap a cyklus programu tím končí.



Obr. 4.11: Výstup v programu Terminal

Po skončení re-centrování obrazu programem *observer* se nalezený objekt nachází ve středu kamerového pohledu. Výsledek tohoto procesu je očividný z obrázku 4.12.



Obr. 4.12: Výsledek centrování programu observer

Vyhledávání je řešeno přímo pro obdélník přesně stanovených rozměrů a stanovených rozmezí jasových hodnot. Tyto hodnoty jsou proměnlivé a závažně závisí na aktuálním osvětlení při snímkování. V tabulce 4.2 jsou nastíněny hodnoty jasu hledané černé barvy v závislosti na denní době.

Tab. 4.2: Proměnlivost jasových hodnot v rámci denní doby (aktuálního osvětlení)

| Osvětlení | Jasová hodnota [dec] |
|------------------------|----------------------|
| Šero | 57-70 |
| Denní sluneční svit | 45-47 |
| Noc-přímé světlo lampy | 31-42 |

Celková doba kompletního rozboru jedné scény uvedeným postupem při negativním nálezů odpovídá maximální době trvání běhu programu. Hodnota byla měřena statisticky a vždy pro negativní nález. Od těchto dob se odečtením doby za kterou MCU uloží obraz do paměti získá doba trvání vyhledávací fáze.

Tab. 4.3: Doba trvání jednoho cyklu programu observer - kompletní rozbor

| Č. měření | t_{kc} [s] | t_{vf} [s] |
|-----------|--------------|--------------|
| 1 | 12,4 | 11,7 |
| 2 | 13,1 | 12,4 |
| 3 | 12,5 | 11,8 |
| 4 | 13,2 | 12,5 |
| 5 | 12,6 | 11,9 |

kde t_{kc} je dobou trvání jednoho kompletního cyklu programu observer a t_{vf} je dobou trvání vyhledávací fáze. Měření probíhalo statistickou metodou při PCLK = 134,3kHz (0x20). Při pozitivním nálezů je doba kratší než zde uvedené t_{kc} . Pro zde uvedený příklad (viz. obrázky 4.7-4.9) doba trvání běhu algoritmu kolísala kolem 3 sekund.

Z tohoto měření vyplývá (kompletní cyklus s negativním nálezem):

- minimální trvání jednoho cyklu činí 12,4 s
- maximální trvání jednoho cyklu činí 13,2 s
- průměrné trvání jednoho cyklu činí 12,8 s

- minimální trvání vyhledávací fáze činí 11,7 s
- maximální trvání vyhledávací fáze činí 12,5 s
- průměrné trvání vyhledávací fáze činí 12,1 s

Rozhodování je prováděno na principu rozdílu polohy středu nalezeného objektu od polohy středu aktuálního kamerového pohledu. Z horizontálu a vektorálu je vypočten úhel i směr natočení senzoru, tak aby se po změně polohy servomotorů oba středy protnuly.

Algoritmus sledování objektu je kompletně funkční pro jedno cyklický průběh. K funkčnímu sledování objektu v nekonečné smyčce je třeba modifikovat natáčení servomotorů relativním způsobem. Aktuálně funguje na bázi přímého výpočtu absolutních hodnot natočení.

S pokusem uvést do provozu nekonečnou smyčku sledování obrazu, se uplatňují nevyřešené problémy. Dostí nízký taktovací kmitočet kamery (PCLK) zajišťuje minimální chyby v obrazové rekonstrukci v paměti, kde stále zůstává malá možnost chybného uložení. Největším problémem je archivace dat v externí paměti, která musí být správně alokována a po jednom kontrolním cyklu uvolněna. Pokud dojde k pádu programu mezi těmito dvěma procedurami, může to mít za následek chybné vyčítání uložených hodnot a v nejhorším případě pád celého programu. Při pokusu o smyčkové sledování docházelo k těmto pádům pravidelně. Většina příčin byla odstraněna modifikací skriptu, ale z důvodu časového nedostatku nebyl funkční smyčkový program vytvořen.

5 ZÁVĚR

Cílem práce bylo aplikovat jednoduché algoritmy pro získání a práci s obrazovými daty digitální kamery C0388 pomocí mikrokontroléru Atmel AVR ATmega128 dle limitací využitelných parametrů a prostřednictvím této implementace otestovat možnost detekce specifikovaného objektu ve snímaném obrazu.

První dvě kapitoly jsou věnovány popisu využitého hardware a jeho parametrů. Kapitola 1 popisuje použitý procesor a systém TWI využitý k komunikaci s EEPROM modulu C3088. Druhá kapitola obsahuje informace o modulu C3088 osazeném digitálním senzorem OV6620, přibližuje správu funkcí řídicích registrů a podobu formátů a výstupních obrazových dat kamery.

Třetí kapitola se věnuje vlastní realizaci kompletního zapojení ke komunikaci s bankou registrů, získávání obrazových dat z kamery, rozšíření datové paměti MCU, komunikaci po sériové lince a ovládání servomotorů.

Následující kapitola popisuje vlastnosti získaných obrazových dat, diskutuje dosažené výsledky a zjištěné parametry přenosu. Vyhodnocování obrazu je založeno na sledování hodnot jasových složek jednotlivých pixelů metodou horizontálního čtení. Touto metodou je možné dosáhnout přenosové rychlosti 200 kbps bez výrazné změny kvality obrazu. Převážně bylo pracováno s rozlišením 120x100 obrazových bodů pro nastavení PCLK = 134,3 kHz. K taktování MCU byl použit externí krystal 16 MHz což je nejvyšší možná hodnota použitelná pro zvolený typ ATmega128. Finální program (observer) je schopen nalézt předem definovaný objekt ve tvaru černého obdélníku o rozměrech 32x15 obrazových bodů, vyhodnotit, kde se nachází střed hledaného objektu z aktuálního pohledu kamery a ten zacentrovat. Úspěšnost nalezení objektu není 100%, což může být následek špatně reprezentovaného obrazového bodu nebo jasové citlivosti, která velmi závisí na aktuální úrovni osvětlení. Častým problémem je též chyba při alokaci či uvolnění heap paměti, jenž má za následek pád programu.

Z důvodu nedostatku času nebylo odladěno automatické centrování v nekonečné smyčce. Kde je nutné modifikovat systém motorické odpovědi systému na změnu z absolutního na relativní.

Dalším postupem v práci by mohlo být rozšíření manipulace obrazových dat o složky chromatické a implementace komplexnějších algoritmů k vysledování pohybu určitého objektu s odladěnou reakcí sledování kamery zaměřeného objektu v nekonečné smyčce.

LITERATURA

- [1] MATOUŠEK, D. *Práce s mikrokontroléry Atmel AVR*. Praha: BEN - technická literatura, 2003.
- [2] MANN, B. *C pro mikrokontroléry*. Praha: BEN - technická literatura, 2003.
- [3] ATMEL Corporation. *ATmega128/L* [online]. 2011 [cit.2011-05-13]. Dostupný na WWW: <http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf>.
- [4] OmniVision. *OV6620/OV6120* [online]. 1999 [cit.2011-05-13]. Dostupný na WWW: <<http://www.hobbyrobot.cz/PDF/ov6620.pdf>>.
- [5] Elektronik123. *C3088 1/4" color camera module with digital output* [online]. 1999 [cit. 2011-05-13]. Dostupný na WWW: <<http://www.electronics123.net/amazon/datasheet/c3088.pdf>>.
- [6] PK-DESIGN. *Uniprog-USB v1.0*. [online]. 2010 [cit.2011-05-13]. Dostupný na WWW: <http://www.pk-design.net/Datasheets/UniProgUSB_v10_doc_ver_v13_20100320.pdf>.
- [7] FTDI LTD. Glasgow. *FT232R USB UART I.C* [online]. 2005 [cit.2010-12-28]. Dostupný na WWW: <http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf>.
- [8] Dušek Martin, Müller Tomáš. *MMIA - FTDI FT235* [online]. 2009 [cit.2011-05-13]. Dostupný na: WWW: <<http://www.urel.feec.vutbr.cz/MIA/2009/dusek/index.html>>.
- [9] Orlando John Royce. *Building robots insted sleeping* [online]. 2004 [cit.2011-05-13]. Dostupný na WWW: <<http://www.jrobot.net/>>.
- [10] FTDI LTD. Glasgow. *FTDI drivers instalation guide* [online]. 2009 [cit.2010-12-28]. Dostupný na WWW: <http://www.ftdichip.com/Support/Documents/AppNotes/AN_119_FTDI_Drivers_Installation_Guide_for_Windows7.pdf>.
- [11] *HW-server* [online]. Praha : HW-server s.r.o., 2004 [cit.2010-12-28]. Dostupné na WWW:< <http://hw-server.com/software/termv19b.html>>.
- [12] Navarro Oiza. *Digital camera interface* [online]. 2004 [cit.2010-12-28]. Dostupný na WWW: < <http://www.robozes.com/inaki/dproject/report.pdf>>.
- [13] Mik Šimon, Juhás Miloš. *Pozicování pomocí dvou serv* [online]. 2010 [cit.2011-05-13]. Dostupný na: WWW: < <http://www.urel.feec.vutbr.cz/MIA/2010/Juhas/index.html#ka2>>.
- [14] Science-Prog. *Adding external memory to Atmega128* [online]. 2010 [cit.2011-05-13]. Dostupný na: WWW: <<http://www.scienceprog.com/adding-external-memory-to-atmega128/>>.
- [15] AMIC Technology,Inc. *32K X 8BIT CMOS SRAM* [online]. 2000 [cit.2011-05-13]. Dostupný na WWW: <http://www.gme.cz/_dokumentace/dokumenty/410/410-019/dsh.410-019.1.pdf>.
- [16] Philips Semiconductors. *Datasheet 74HC/HCT573* [online]. 1990 [cit.2011-05-13]. Dostupný na WWW: <http://www.gme.cz/_dokumentace/dokumenty/426/426-081/dsh.426-081.1.pdf>.
- [17] ATMEL Corporation. *AVR315: Using the TWI module as I2C master* [online]. 2010 [cit.2011-05-13]. Dostupný na WWW:

<http://www.atmel.com/dyn/resources/prod_documents/doc2564.pdf>.

- [18] FRÝZA, T. FEDRA, Z. ŠEBESTA, J. *Mikroprocesorová technika*: Skriptum pro laboratorní cvičení předmětu Mikroprocesorová technika. Brno: FEKT VUT v Brně, [cit.2011-05-13]. 50 s.

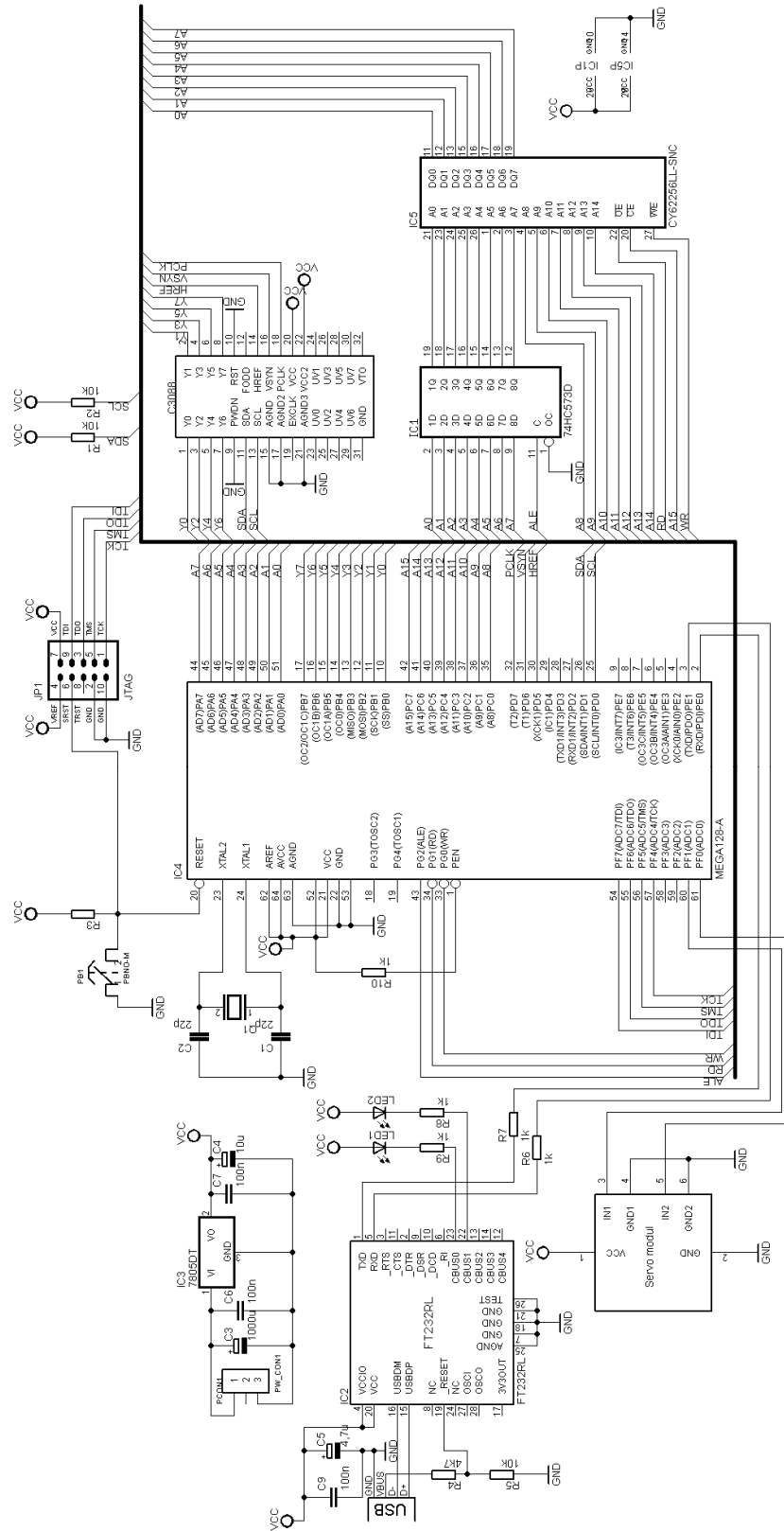
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

| | |
|------------------|---|
| AD/DA | Převodníky analog->digital / digital->analog |
| AVR | Označení pro rodinu 8bitových mikročipů typu RISC s harvardskou architekturou od firmy Atmel |
| CCD | Charge-Coupled Device, zařízení s vázanými náboji |
| CIF | Common Intermediate Format, formát 352x288 |
| CMOS | Complementary Metal–Oxide–Semiconductor, doplňující se kov-oxid-polovodič |
| DPS | Deska plošného spoje |
| EEPROM | Electrically Erasable Programmable Read-Only Memory, jedná se o elektricky mazatelnou nevolatilní paměť |
| I ² C | Inter-Integrated circuit, sériová sběrnice |
| JTAG | Join Test Action Group |
| MCU | Microcontroller Unit |
| QCIF | Quarter CIF (174x144) |
| PC | Personal Computer |
| PTH | Plated Through-Hole, součástky instalované skrz otvor v DPS |
| RISC | Reduced Instruction Set Computer, procesory s redukovanou instrukční sadou |
| SCL | Signal CLock, hodinový signál |
| SDA | Signal DAta, datový signál |
| SRAM | Static Random Access Memory, polovodičová paměť, která k uchování svých dat nepotřebuje jejich periodickou obnovu |
| TWI | Two-Wire Interface, AVR označení pro I ² C |

SEZNAM PŘÍLOH

| | | |
|----------|--------------------------------------|-----------|
| A | Obvodové zapojení | 37 |
| B | Seznam součástek | 38 |
| C | Fotografie sestaveného obvodu | 39 |

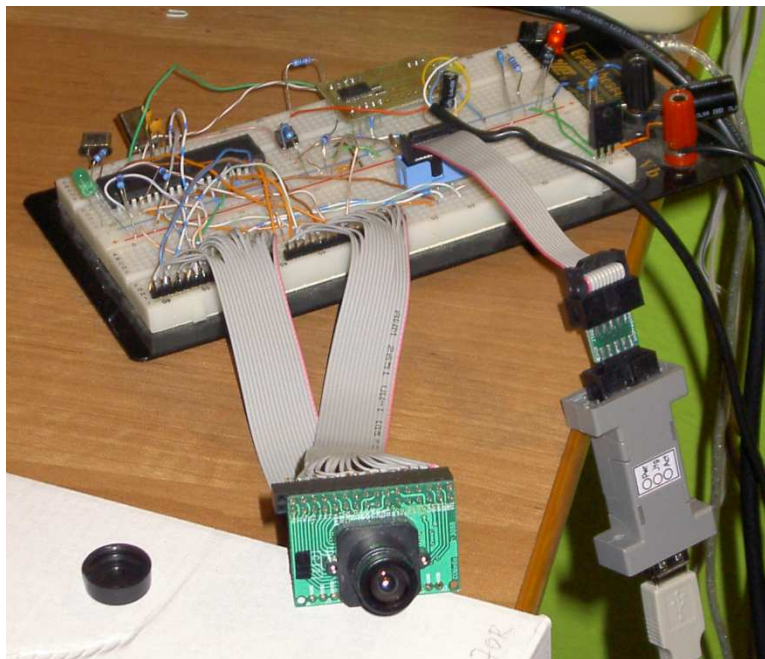
A OBVODOVÉ ZAPOJENÍ



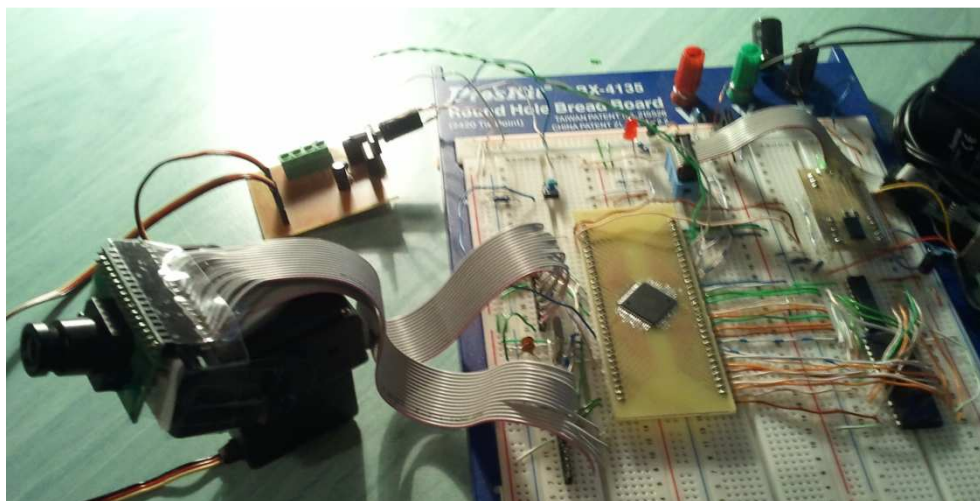
B SEZNAM SOUČÁSTEK

| Označení | Hodnota | Pouzdro | Popis |
|----------|--------------|---------|----------------------------|
| C1 | 22p | 5mm | Keramický kondenzátor |
| C2 | 22p | 5mm | Keramický kondenzátor |
| C3 | 1000u | 13x21mm | Elektrolytický kondenzátor |
| C4 | 10u | 5x7mm | Elektrolytický kondenzátor |
| C5 | 4,7u | 4x7mm | Elektrolytický kondenzátor |
| C6 | 100n | 5mm | Keramický kondenzátor |
| C7 | 100n | 5mm | Keramický kondenzátor |
| C9 | 100n | 5mm | Keramický kondenzátor |
| IC4 | ATmega128-AU | TQFP64 | Mikrokontrolér |
| IC2 | FT232RL | SSOP28 | Převodník USB/USART |
| IC3 | 7805 | TO220 | Stabilizátor napětí |
| LED1 | YELLOW | 5mm | LED dioda |
| LED2 | GREEN | 5mm | LED dioda |
| Q1 | 16MHz | HC49-U | Krystal |
| R1 | 10k | 0207 | rezistor |
| R2 | 10k | 0207 | rezistor |
| R3 | 10k | 0207 | rezistor |
| R4 | 4k7 | 0207 | rezistor |
| R5 | 10k | 0207 | rezistor |
| R6 | 1k | 0207 | rezistor |
| R7 | 1k | 0207 | rezistor |
| R8 | 1k | 0207 | rezistor |
| R9 | 1k | 0207 | rezistor |
| JP1 | JTAG | S2G20 | 2x5 pinů |
| U1 | | TM070 | Mikrospínač |
| C3088 | OV6620 | 2x16 | Konektor kamery |
| USB | FEMALE | A | Konektor USB |
| Kab1 | 0,5m | USB/A/M | USB kabel |
| Kab2 | 0,5m | USB/A/M | USB kabel |
| IC1 | 74HCT573 | DIP20 | Latch (flip-flop) |
| IC5 | 32kx8 | DIP28 | SRAM |

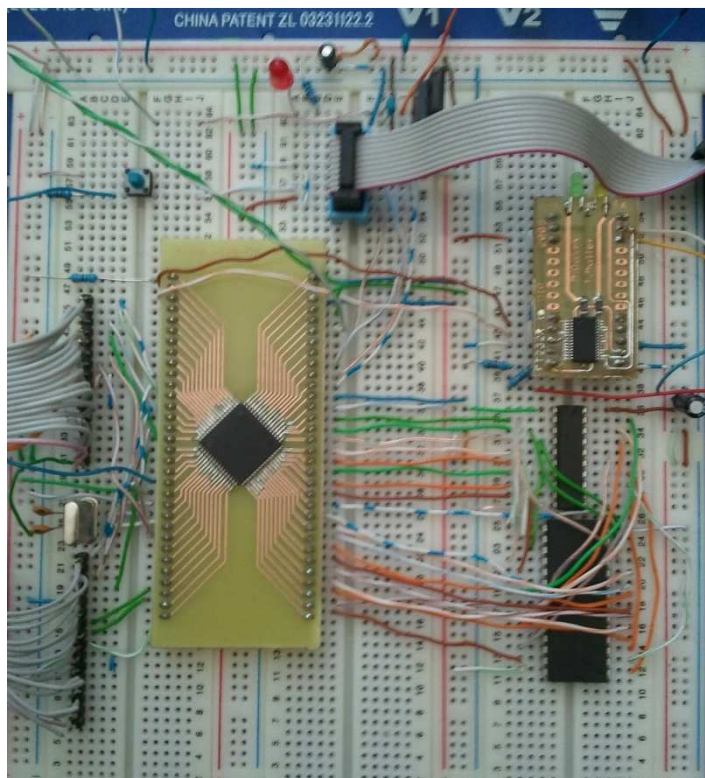
C FOTOGRAFIE SESTAVENÉHO OBVODU



Obr. C.1: Testovací zapojení s ATmega16



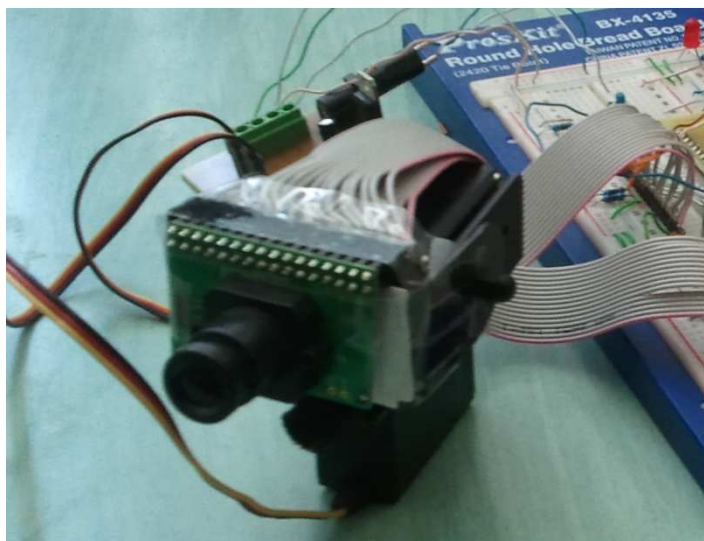
Obr. C.2: finální zapojení s ATmega128



Obr. C.3: Detail nepájivého pole



Obr. C.4: Vstupní, Výstupní periferie a napájení



Obr. C.5: Kamerový modul C3088 připevněný k servu Hitec