

# ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ FAKULTA ELEKTROTECHNICKÁ KATEDRA ŘÍDICÍ TECHNIKY



## BAKALÁŘSKÁ PRÁCE

Použití procesoru typu ARM řady Atmel AT91SAM  
pro řízení videokamery přes CAN sběrnici

Katedra řídicí techniky

Školní rok: 2006/2007

## Z a d á n í   b a k a l á ř s k é   p r á c e

Student:                    Michal N o v o t n ý  
Obor:                        Kybernetika a měření  
Název tématu:            Použití procesoru typu ARM řady Atmel AT91SAM pro řízení  
videokamery přes CAN sběrnici

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

1. Seznamte se s procesorem typu ARM řady Atmel AT91SAM.
2. Seznamte se s komunikačním rozhraním zadané kamery.
3. Navrhněte jednoduchý obvod realizující CAN rozhraní pro řídicí signály kamery.
4. Naprogramujte základní komunikaci mezi procesorem a kamerou po CAN sběrnici.

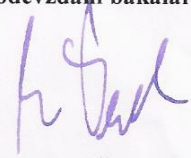
### *Seznam odborné literatury:*

1. Firemní literatura k procesoru Atmel AT91SAM. Online: <http://www.atmel.com/products/AT91/>
2. Firemní literatura ke kameře Sony série FCB-IX <http://www.sonybiz.net/vision>

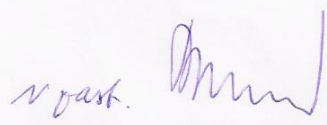
**Vedoucí bakalářské práce:** Ing. Zdeněk Hurák, Ph.D.

**Datum zadání bakalářské práce:** zimní semestr 2006/07

**Termín odevzdání bakalářské práce:** 15. 8. 2007

  
Prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry



  
Prof. Ing. Zbyněk Škvor, CSc.  
děkan

V Praze, dne 6. 3. 2007

## PROHLÁŠENÍ

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne .....

.....

podpis

## PODĚKOVÁNÍ

Touto cestou bych rád poděkoval kolektivu katedry řídicí techniky, který mi poskytl cenné rady a informace. Jmenovitě bych poděkoval vedoucímu práce Ing. Zdeňku Hurákovi, Ph.D. za možnost výběru této práce. Poděkování patří též všem, kteří mě ve studiu na vysoké škole podporují a nějak mi pomáhají.

## ANOTACE

Bakalářská práce je jednou z mnoha součástí projektu pohyblivé základny pro sledování terénu z bezpilotního letadla pomocí několika kamer. Veškeré řízené části uvnitř základny jsou k hlavnímu počítači připojeny přes kluzné kroužky. Ty je důležité použít pro možnost neomezené rotace. Tím vzniká problém přenosu informací pro jednotlivá zařízení. Většina kamer komunikuje přes rozhraní RS232. Proto je nutné omezit počet kabelů sjednocením komunikační sběrnice.

Tato práce se tedy zabývá řízením jedné z kamer pomocí řídicího mikroprocesoru umístěného mimo základnu. Ten vysílá příkazy po sběrnici CAN, které jsou na základně převedeny pro potřebu kamery. Tedy na sběrnici typu RS232. Převodník ze sběrnice typu CAN na RS232 je hlavní součástí této práce.

## ANNOTATION

The bachelor work is one of many parts of the project 'mobile base' for terrain monitoring from a pilotless plane (beeper) with the help of several video cameras. All controlled parts inside the base are connected to the main computer through slide rings. They are important to use for the possibility of unrestricted rotation. With this, a problem with information data transfer for the individual appliances occurs. Most video cameras communicate though the interface RS232. That is why it is important to reduce the number of cables by an integration communication data bus line.

This work addresses controlling of one of the video cameras by a control microprocessor, which is located outside the base. This processor transmits instructions though the data bus line CAN, which are transferred for the video camera use on the base, to the data bus line RS232. The conversion unit from CAN to RS232 is the main part of this bachelor diploma work.

# OBSAH

1. Úvod.....	- 1 -
1.1. Motivace .....	- 1 -
1.2. Cíl vlastní práce .....	- 1 -
1.3. Ideový návrh .....	- 1 -
2. CAN sběrnice.....	- 3 -
2.1. Úvod .....	- 3 -
2.2. Vlastnosti sběrnice.....	- 3 -
2.3. Jednotlivé typy zpráv .....	- 4 -
2.4. Synchronizace zpráv .....	- 4 -
3. Kamera FCB-X11.....	- 5 -
3.1. Úvod .....	- 5 -
3.2. Komunikační rozhraní .....	- 6 -
3.2.1. Připojení zařízení.....	- 6 -
3.2.2. Datové zprávy .....	- 7 -
3.2.3. Rozdělení zpráv .....	- 7 -
3.2.4. Vybrané příkazy a odpovědi kamery .....	- 8 -
4. Řídicí část .....	- 9 -
4.1. Procesor typu ARM AT91SAM7X256.....	- 9 -
4.2. Vývojový kyt Atmel AT91SAM7X-EK .....	- 10 -
4.3. Softwarové řešení řídicí části .....	- 10 -
4.3.1. Popis řešení .....	- 10 -
4.3.2. Jednotlivé kroky programu.....	- 11 -
4.3.3. Parametry nastavení CAN sběrnice:.....	- 11 -
4.3.4. Nastavení sériového rozhraní:.....	- 11 -
4.3.5. Popis programu .....	- 11 -
4.3.6. Výpis použitých funkcí programu .....	- 12 -
5. Rozhraní pro komunikaci mezi sběrnici CAN a RS232 .....	- 14 -
5.1. Návrh elektroniky převodníku .....	- 14 -
5.1.1. Použité integrované obvody .....	- 15 -
5.1.2. Návrh schématu zapojení a desky plošného spoje .....	- 15 -

5.2. Programové řešení.....	- 18 -
5.2.1. Popis hlavního principu programu.....	- 18 -
5.2.2. Popis důležitých funkcí programu .....	- 19 -
6. Závěrečné shrnutí.....	- 20 -
7. Literatura .....	- 23 -

## SEZNAM OBRÁZKŮ

Obr. 1 - Ideový model systému .....	- 2 -
Obr. 2 - Kamera Sony FCB-IX.....	- 5 -
Obr. 3 - Sériové zapojení VISCA zařízení .....	- 6 -
Obr. 4 - Datový rámec jedné zprávy VICSA protokolu.....	- 7 -
Obr. 5 - Vývojový kyt atmel AT91SAM7X + JLink debugger.....	- 9 -
Obr. 6 - Multiplex sériového rozhraní.....	- 14 -
Obr. 7 - Osazená deska rozhraní CAN - RS232 vrchní strana .....	- 17 -
Obr. 8 - Osazená deska rozhraní CAN - RS232 spodní strana.....	- 18 -
Obr. 9 - Kamerová základna.....	- 20 -
Obr. 10 - Řídicí mikroprocesor zapojený s převodníkem.....	- 21 -
Obr. 11 - výpis ovládacího programu s výpisem odpovědi od kamery -	22 -

# 1. ÚVOD

## 1.1. MOTIVACE

Hlavní motivací pro výběr této práce byla možnost seznámení se s nějakým hardwarovým problémem, který je nutné řešit pomocí mikroprocesoru naprogramovaného k určité aplikaci. Hlavním důvodem tedy bylo seznámení se s programováním mikroprocesorů, které lze v pozdější době využít na celou řadu zajímavých projektů, do kterých bych se bez této zkušenosti neodvážil pustit.

## 1.2. CÍL VLASTNÍ PRÁCE

Cílem této práce je řízení kamery SONY FCB-IX11, která pro komunikaci využívá sériové rozhraní, z mikroprocesoru Atmel AT91SAM7X, který bude příkazy vysílat po sběrnici typu CAN. Práce je tedy rozdělena do několika bloků a těmi jsou:

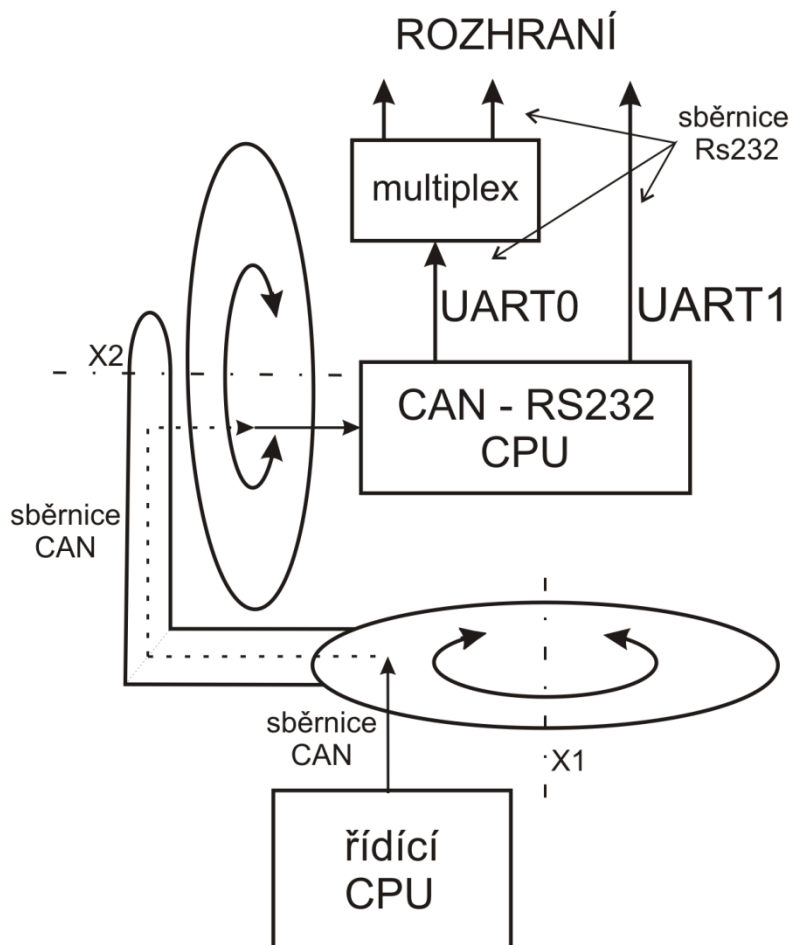
- Seznámení se s řídicím procesorem a naprogramování pro jednoduché řízení po sběrnici CAN.
- Seznámení se s komunikačním protokolem kamery SONY pro její základní nastavení a řízení.
- Navrhnutí jednoduchého převodníku mezi řídicím procesorem a řízenou kamerou, který by umožnil připojení sériového zařízení na sběrnici CAN s možností připojení více sériových zařízení.

## 1.3. IDEOVÝ NÁVRH

První ideou bylo nalezení integrovaného obvodu pro vytvoření převodníku CAN – RS232 a její zapojení mezi obě části. Toto se ale ukázalo jako nemožné z důvodu neexistence univerzálního převodníku. Ten by musel obsahovat několik možností nastavení jak sběrnice RS232 tak CAN. Proto bylo nutné použít mikroprocesor naprogramovaný pro tyto potřeby. Pro velkou rozšířenost ve školních projektech byl vybrán mikroprocesor Philips LPC 2119, který obsahuje CAN rozhraní, a dva sériové porty. Zjednodušený model aplikace je zobrazen na Obr. 1, kde je řídicí počítač, který pro připojení využívá dvoudrátové sběrnice CAN. Ta je postupně připojena k prvním kluzným kontaktům, které umožňují rotaci kamer kolem osy X1 a poté k druhým kluzným kontaktům pro rotaci kolem osy X2. Poté jsou data převedena na rozhraní RS232 a vyvedena k ostatním periferním zařízením. Přičemž zde už není problém s množstvím kabelového vedení. Pro možnost rozšíření je návrh ještě doplněn o multiplex, který umožňuje přepínání



jednoho sériového portu na několik zařízení. Toto řešení je možné použít u zařízení, které příkazy přijímá a okamžitě na ně reaguje. Nelze použít u zařízení, které pravidelně data vysílá. Tam by nastal problém se ztrátou vysílaných dat, které by nebyly nikým přijímány.



**OBR. 1 - IDEOVÝ MODEL SYSTÉMU**

## 2. CAN SBĚRNICE

V této kapitole jsou popsány základní vlastnosti a funkce protokolu CAN, který je velmi důležitý pro komunikaci mezi řídicím procesorem a ostatními periferními jednotkami.

### 2.1. Úvod

Controller Area Network (dále jen CAN) je sériová sběrnice, která byla vyvinuta firmou BOSCH pro automobilový průmysl. Postupem času se začala využívat i v ostatních průmyslových odvětvích. Její první specifikovaný standard je CAN 2.0A, ke kterému později přibývá standard CAN 2.0B. Tento nový formát zavádí standardní a rozšířený formát zpráv.

### 2.2. VLASTNOSTI SBĚRNICE

Hlavními rysy této sběrnice je možnost distribuovaného řízení při přenosových rychlostech do 1Mbit/s. Tato rychlost je ale velmi závislá na délce vedení a počtu připojených zařízení. Pro zabezpečení přenosu proti chybám je použito protokolu typu multi-mastr, který umožňuje rovnocenné řízení ze všech uzlů. K přístupu na sběrnici je použita metoda náhodného přístupu. Každá zpráva obsahuje identifikátor, který udává prioritu. Identifikátor určuje, pro které zařízení je zpráva vysílána, a při kolizi je použit k rozhodování, která zpráva bude potlačena a která bude dále pokračovat ve vysílání.

Sběrnice je tvořena dvěma vodiči označenými CAN-H a CAN-L. Tyto vodiče mají na koncích zakončovací rezistory. Ty slouží k eliminaci odrazů, které způsobuje rušení a snižuje možnost využití maximální přenosové rychlosti. Jednotlivé hodnoty na sběrnici jsou určeny rozdílem napětí mezi vodiči. Při klidovém stavu na sběrnici je na vodičích CAN-H a CAN-L stejné napětí 2.5V. Tedy nulový rozdílový potenciál. Při úrovni 1 stoupá napětí na vodiči CAN-H na 3.5V – 5V a klesá na vodiči CAN-L na hodnotu 1.5V – 0V. Tím vzniká různý potenciál.

Při kolizi zpráv dochází k porovnávání prvních bitů obou zpráv. V prvních bytech je přenášeno identifikační číslo. Zpráva, která má jako první dominantní úroveň, pokračuje ve vysílání. Proto je priorita zprávy dána identifikačním číslem, které při nižších hodnotách dosahuje dominantní úrovně dříve než vyšší identifikační číslo. Dominantní úroveň je dána rozdílem potenciálu sběrnice. Recesivní úroveň je dána nulovým rozdílovým potenciálem.

Jednotlivá zařízení jsou připojena paralelně k lince a všechna zařízení přijímají veškeré vysílané zprávy. Pro rychlost 1Mbit/s při délce sběrnice do 40m je doporučeno připojovat maximálně 30 zařízení.

Po sběrnici se komunikuje pomocí několika typů zpráv.

### **2.3. JEDNOTLIVÉ TYPY ZPRÁV**

CAN definuje 4 základní typy zpráv:

- Datová zpráva (DATA FRAME)

Je základní datovou zprávou, která přenáší požadovaná data. Každá datová zpráva obsahuje 8 bytů pro přenos dat, informaci o počtu platných bytů a identifikační číslo, které je pro standard CAN 2.0A 11bitové a zprávy jsou nazývány standard frame a pro standard CAN 2.0B 29 bitové, kde jsou nazývány extend frame.

- Žádost o data (REMOTE FRAME)

Tato zpráva obsahuje identifikační číslo a počet dat, o které žádá.

- Chybová zpráva ( ERROR FRAME)

Signalizuje chyby na sběrnici. Všechny uzly kontrolují všechny zprávy a v případě výskytu nějaké chyby vysílá chybovou zprávu. Která zprávu naruší a chybu zaznamenají i ostatní uzly.

- Zpráva o přetečení (OVERLOAD FRAME)

Je vysílána při nemožnosti uzlu přijmout data z důvodu nezpracování již přijatých dat.

### **2.4. SYNCHRONIZACE ZPRÁV**

K synchronizaci přenášených dat se využívá změna úrovně signálu na sběrnici. Kdy dochází k resynchronizaci. Doba trvání každého bitu je rozdělena do několika částí, jejichž délka je dána počtem tzv. Time quanta. To je nejmenší časový úsek a jeho celistvé násobky určují délku jednotlivých částí bitu. Ty jsou:

- SYNC\_SEG - v kterém probíhá synchronizace.
- PROG\_SEG - slouží ke kompenzaci zpoždění jak kabelovým vedením, tak zpožděním jednotlivých obvodů.
- PHASE\_SEG1 a PHASE\_SEG2 – určují pozici vzorkovacího času bitu, který se nachází mezi těmito úseky. Jejich délka se může při resynchronizaci měnit.

### **3. KAMERA FCB-X11**

V této části jsou popsány základní vlastnosti kamery a komunikační protokol, pro její ovládání. Samotná kamera je zobrazena na Obr. 2.

#### **3.1. Úvod**

Jedná se o barevnou kameru firmy SONY, která bude použita pro sledování terénu z letecké perspektivy. Hlavní nárok je tedy kladen na objektiv s možností co největšího přiblížení, které činí 10x pro optické přiblížení a dále možnost 4x digitální přiblížení. Její CCD snímač se zvýšenou citlivostí umožňuje snímání obrazu od velikosti osvětlení 1.5 Lux. Je navržena pro maximální úsporu elektrické energie. A díky protokolu VISCA umožňuje snadné ovládání přes sériové rozhraní. Další důležitý parametr je její velikost, která činí 40 x 45 x 60 mm.



**OBR. 2 - KAMERA SONY FCB-IX**

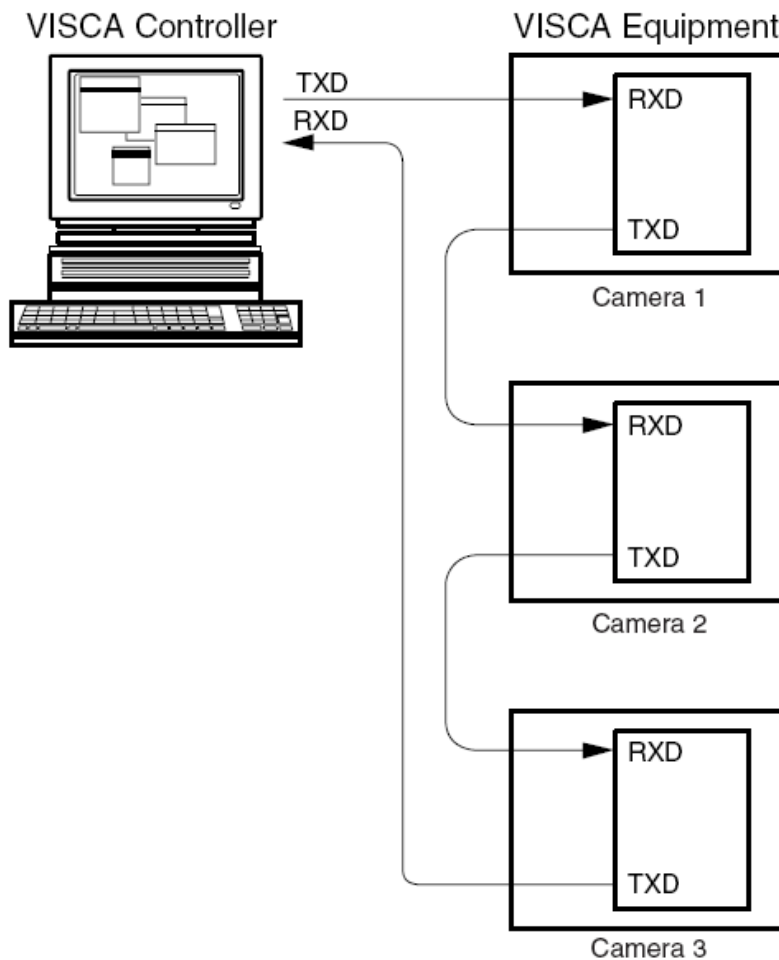
## 3.2. KOMUNIKAČNÍ ROZHŘANÍ

Komunikační rozhraní této kamery je nazýváno VISCA a slouží pro její ovládání. Toto rozhraní spočívá na komunikaci po sériové lince s možností připojení, jak k napětovým úrovním RS232 tak k napětovým úrovním TTL. Datová komunikace probíhá s následujícím nastavením:

- komunikační rychlosti 9600 kbps, 19200kbps nebo 38400kbps.
- 1 start bit
- 1/2 stop bit
- 8 data bits
- bez parity

### 3.2.1. PŘIPOJENÍ ZAŘÍZENÍ

Komunikační protokol VISCA umožňuje připojení až 7 zařízení s kompatibilním protokolem. Jednotlivá zařízení jsou připojena sériově, jejich zapojení je zobrazeno na Obr. 3.

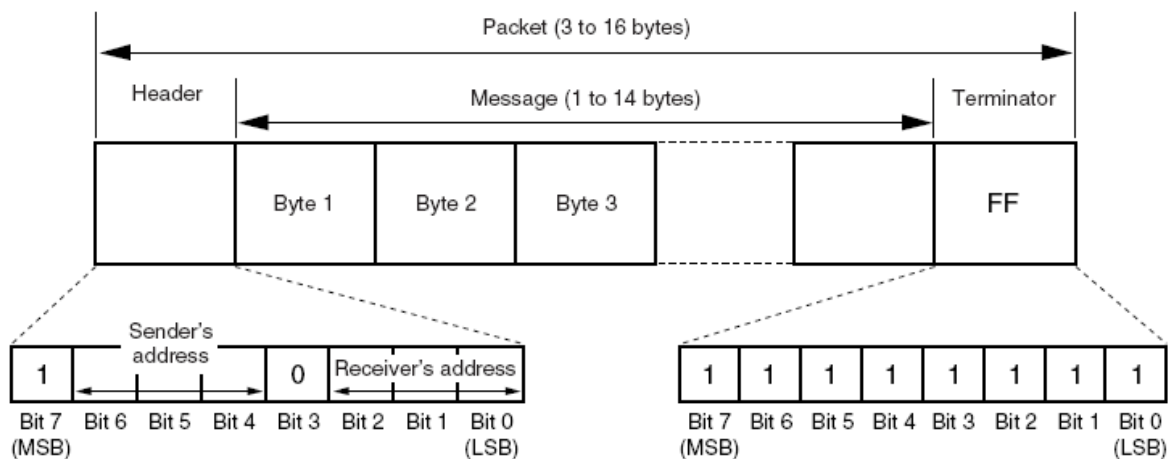


OBR. 3 - SÉRIOVÉ ZAPOJENÍ VISCA ZAŘÍZENÍ

Každé zařízení má svou adresu, která je přenášena v prvním bytu komunikačních zpráv. V případě nesouhlasící adresy zařízení a zprávy je zpráva přeposlána dál. Hlavní řídicí zařízení má adresu 0 a jednotlivé rozhraní 1 až 7.

### 3.2.2. DATOVÉ ZPRÁVY

Každý datový rámeček, viz Obr. 4, může dosahovat délky 3 až 16 bytů, kde první byt obsahuje adresu vysílacího zařízení (bit 4 – 6) a přijímacího zařízení (bit 0 – 2), přičemž 7 bit je vždy 1. Poslední byt má hodnotu FF a udává konec paketu. Ostatní určují daný příkaz nebo odpověď. Adresa zařízení je nastavena inicializační zprávou ve formátu 88 30 01 FF, kde 88 udává posláno od řídicího zařízení všem připojeným, 01 je nastavovaná adresa zařízení a FF ukončuje zprávu.



OBR. 4 - DATOVÝ RÁMEČEK JEDNÉ ZPRÁVY VICSA PROTOKOLU

### 3.2.3. ROZDĚLENÍ ZPRÁV

Jednotlivé zprávy jsou rozděleny do několika kategorií:

- Příkazová zpráva (Command message) - pro nastavení parametrů kamery
- Potvrzovací zpráva (ACK message) - kterými zařízení potvrzují přijetí příkazu
- Ukončovací zpráva (Completion message) - kterými zařízení potvrzují vykonání příkazu
- Dotazovací zpráva (Inquire message) – žádost o informaci o aktuálním nastavení
- Chybové zpráva (Error message) – při chybném příkazu nebo při nemožnosti vykonat příkaz.

- příkaz pro zrušení zprávy (Command execution cancel message) – umožňuje stornování vyslaného a nezpracovaného příkazu. Příkazy IF\_Clear (8x 01 00 01 FF, x adresa příjemce) ruší příkazy v obou soketech. Příkaz 8x 2y FF v soketu y, zařízení x.

Komunikace tedy probíhá zasíláním příkazů adresovanému zařízení. Při přijetí je vyslána potvrzovací zpráva ve formátu xy 4z FF, kde x udává adresu zařízení + 8, y je adresa příjemce tedy 0 pro řídicí počítač a z udává číslo soketu, kde byl příkaz uložen. Po zpracování příkazu je odeslána ukončovací zpráva (např. 90 51 FF kde 1 je číslo soketu). Tato zpráva není odeslána při zaslání dotazovací zprávy, na kterou zařízení odpovídá datovým paketem o informacích aktuálního nastavení.

### 3.2.4. VYBRANÉ PŘÍKAZY A ODPOVĚDI KAMERY

Seznam několika základních příkazů z každé kategorie příkazů.

Příkaz	Příkazová zpráva	Odpověď	Poznámka
Hlavní příkazy	81 01 04 38 02 FF	90 41 FF(ACK) 90 51 FF(completion)	Vrací ACK při přijetí a Completion při vykonání.
	81 01 04 38 FF	90 60 02 FF(špatný příkaz)	Je zadán špatný příkaz nebo příkaz se špatným parametrem.
	81 01 04 38 02 FF	90 60 03 FF (příkazový zásobník je plný)	Při vykonávání dvou příkazů nemůže být další přijat.
Dotazovací příkaz	81 09 04 38 FF	90 50 02 FF (completion)	ACK zpráva není u dotazovacích příkazů vysílána.
	81 09 05 38 FF	90 60 02 FF	Neplatný příkaz
Nastavení adresy	88 30 <u>01</u> FF	88 30 <u>02</u> FF	Vrací adresu + 1
IF_Clear (všem)	88 01 00 01 FF	88 01 00 01 FF	Vrací stejný příkaz
IF_Clear (pro x)	8x 01 00 01 FF	z0 50 FF (completion)	Nevrací ACK.
Zrušení příkazu v daném soketu	8x 2y FF	Z0 6y 04 FF (příkaz zrušen)	Při úspěšném zrušení není poslána completion zrušeného příkazu.
		Z0 6y 05 FF	Při neúspěšném zrušení nebo při špatném zadání soketu.

TABULKA 1 - UKÁZKA ZÁKLADNÍCH PŘÍKAZŮ K OVLÁDÁNÍ KAMERY

## 4. ŘÍDICÍ ČÁST

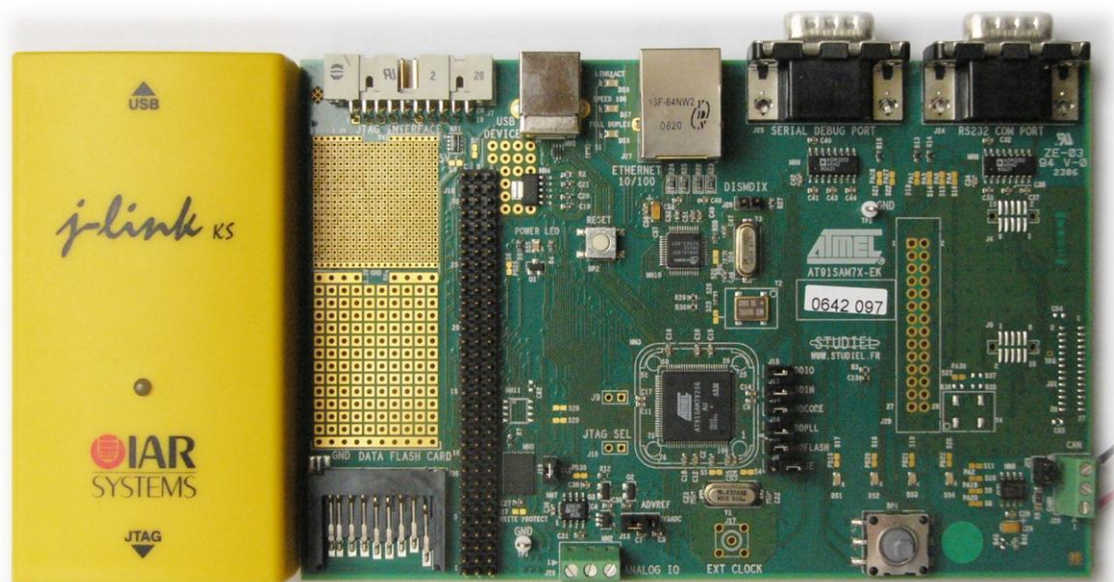
Tato část popisuje realizaci řídicí části, základní informace o použitém hardwaru a softwaru k této části.

### 4.1. PROCESOR TYPU ARM AT91SAM7X256

Tento procesor je určen v zadání práce. Z důvodu jeho nasazení pro řízení celé kamerové základny. Byl vybrán z důvodů vysoké výkonnosti a velkému množství periférií. 32 bitový procesor je založen na jádře ARM7TDMI RISC. Obsahuje integrovanou paměť 256 kB typu flash a 64 kB SRAM.

K jeho perifériím patří USB 2.0, Ethernet MAC 10/100, CAN 2.0A a 2.0B kontrolér, 2x USART port, SPI, SCC, TWI, 8 kanálový 10 bitový AD převodník. Díky tomu se velmi snižuje počet dalších potřebných čipů a tím i energetická náročnost. Tím, že jsou periférie integrovány, snižuje se riziko špatného zapojení nebo nějakého přerušení spoje. Pracovní frekvence je v nejhorších teplotních podmínkách 55MHz. Napájení jádra je typických 1.8V, vstupních a výstupních portů 1.8V nebo 3.3V s 5V tolerancí. Procesor obsahuje i stabilizátor napětí.

Procesor je určen k real-time řídicích aplikacích, kde je zapotřebí propojení pomocí USB portu, Ethernetu nebo CAN sběrnice. Což odpovídá i našemu problému řízení kamerové základny.



OBR. 5 - VÝVOJOVÝ KYT ATMEL AT91SAM7X + JLINK DEBUGGER



## **4.2. VÝVOJOVÝ KYT ATMEL AT91SAM7X-EK**

Tato vývojová deska je založena na výše zmíněném procesoru a slouží pro testování navržených aplikací před jejím samotným osazením. Obsahuje USB port, DBGU sériový komunikační port, RS232 sériové rozhraní se signály RTS/CTS, JTAK/ICE ladící rozhraní, sériové CAN komunikační rozhraní, Ethernet 10/100, AD převodník a PWM výstup, napájecí led a univerzální LED diody, slot pro paměťové karty atd. Proto plně vyhovoval pro naše účely řídicí části.

K programování vývojové desky je díky vyvedenému JTAK rozhraní určen JLink debugger od firmy IAR systems. Vývojový kyt s ladícím rozhraním je zobrazen na Obr. 5.

## **4.3. SOFTWAREVÉ ŘEŠENÍ ŘÍDICÍ ČÁSTI**

Programování samotného programu bylo provedeno v programu IAR Embedded Workbench. Tento program je dodán k vývojové desce a pomocí ladícího zařízení JLink debugger umožňuje nahrávání a testování navržené aplikace. To je velmi vhodné při hledání chyb, kterým se při vývoji softwaru nedá vyhnout.

### **4.3.1. POPIS ŘEŠENÍ**

Po seznámení se s procesorem a programováním jeho některých částí jsem pro řídicí algoritmus využil 3 základní části.

- Jedná se o rozhraní CAN pro řídicí příkazy.
- Sériové rozhraní pro vizuální sledování prováděných operací a přijímaných odpovědí kamery.
- Joystick integrovaný na naší destičce, kterým jsou vybírány příkazy kamer.

Pro řízení kamery je nutné vysílání datových zpráv. Jednou možností tohoto řešení je tedy tyto zprávy pomocí PC posílat přes rozhraní sériového portu. Při použití by výsledný systém zapojený s převodníkem a kamerou představoval jakýsi přenos sériových dat přes rozhraní CAN. Na jednom konci vysíláme data a na druhém přijímáme v nezměněné podobě. Toto řešení je možné, ale zakládá se na nutnosti vysílání dat z dalšího řídicího počítače. Proto jsem se rozhodl tyto příkazy definovat už v řídicím mikroprocesoru a odesílat je za pomoci joysticky. Pro výběr mezi nimi jsem využil možnost výběru pomocí výpisu na sériovém portu.

### **4.3.2. JEDNOTLIVÉ KROKY PROGRAMU.**

Spuštění programu, po kterém se inicializují potřebné registry pro povolení přerušení, nastavení vstupních a výstupních portu pro vizualizaci pomocí LED diod a načítání za pomoci joysticku, nastavení message registrů procesoru pro příjem a odesílání zpráv a vypsaní hlavního výběru na sériové rozhraní.

Poté následuje vybrání určitého příkazu za pomoci joysticku a výpisu na sériovém rozhraní. Program si ukládá vybranou pozici a při potvrzení zasílá příslušný příkaz. Při odeslání zprávy je následná odpověď vypsaná v HEX formátu. Tyto zprávy nebyly nijak zpracovány. Tímto systémem je možno demonstrovat několik předdefinovaných příkazů příslušné kamery, které k ukázce funkce a prvnímú seznámení postačuje.

### **4.3.3. PARAMETRY NASTAVENÍ CAN SBĚRNICE:**

- Rychlost sběrnice 500kbit/s
- 4 message boxy nastaveny na příjem zpráv kdy poslední message box umožňuje přepis nepřečtené zprávy. Tato možnost je možná ale sloužila spíše jen k testování programových schopností rozhraní CAN. Jeden message box by pro příjem zpráv byl dostačující.
- 4 message boxy nastaveny na odesílání se stejným ID. Toto nastavení opět sloužilo k testování rychlosti zpracování odesílaných dat.
- Chybové zprávy a přerušení zatím nebyly nijak odchyťávány a zpracovávány. Při testování přenos fungoval 100procentně a jejich odchyťávání a zpracování závisí na konkrétní aplikaci.

### **4.3.4. NASTAVENÍ SÉRIOVÉHO ROZHRAŇÍ:**

- Přenosová rychlost 115200 kbps
- 1 start bit
- 1 stop bit
- 8 data bit
- Bez parity

### **4.3.5. POPIS PROGRAMU**

Veškeré datové zprávy, jak ze sériového portu, tak CAN portu, jsou ukládány do cyklického zásobníku, kde je zaznamenána pozice prvního a posledního platného bytu. Poté při rozdílné pozici prvního a posledního platného bytu je informace z prvního bytu přenesena na druhé rozhraní, kde při posílání na sériový port je okamžitě odeslána a při posílání na CAN sběrnici uložena do datového pole message registru. Při naplnění celého

datového pole je paket odeslán. Při vyčerpání zásobníku je nastaven počet platných bytů datového pole message registru a následně odeslán. Test plnosti obou zásobníků je cyklicky opakován až do vypnutí procesoru.

#### **4.3.6. VÝPIS POUŽITÝCH FUNKCÍ PROGRAMU**

##### **int main( void )**

- Hlavní vlákno programu - vyvolává funkce pro nastavení CAN a UART, nastavuje vstupní a výstupní porty, nastavuje a povoluje přerušování. Poté v cyklu testuje plnost zásobníku a v případě obsahujících dat zajišťuje jejich přeposlání. V případě příjmu dat po CAN sběrnici provádí výpis dat pro vizuální kontrolu.
- Výstupní parametr:
  - 1 - v případě úspěšného dokončení programu

##### **void pio\_c\_irq\_handler( void )**

- Obsluha přerušování vyvolaná použitím joysticku. Testuje, který směr vyvolal přerušování, a podle toho mění polohu v menu, nebo vysílá příkaz. Při odeslání příkazu pro přiblížení čeká na uvolnění joysticku pro vyslání příkazu zastavení přiblížování.

##### **void write\_menu( void )**

- Vypiše položky menu pro výběr příkazu.

##### **void uart\_init( void )**

- Nastavuje registry pro komunikaci po sériovém portu a nastavuje obsluhu přerušování.

##### **void uart\_handler( void )**

- Obsluha přerušování. Ukládá příchozí data a zachycuje chyby na sériovém rozhraní.

##### **void can\_init( void )**

- Nastavuje registry pro komunikaci CAN. Nastavuje rychlost a typy message boxu a jejich využití.

##### **void send\_string( unsigned char \* buffer, int size )**

- Odesílá data po sběrnici CAN.
- Vstupní parametry:
  - buffer – udává adresu prvního znaku přenášeného řetězce
  - size - udává délku přenášeného řetězce

**void send( int dataL, int dataH, int size )**

- Odesílá data po sběrnici CAN
- Vstupní parametry:
  - dataL – první 4 byty vysílané zprávy
  - dataH – byty 5-8 přenášené datové zprávy
  - size – udává počet platných přenášených bytů

**void can\_c\_irq\_handler( void )**

- Vlákno přerušení vyvolané při příjmu dat po CAN sběrnici. Data ukládá do zásobníku, jež jsou dále zpracovávána v hlavním vláknu.

**int push\_to\_buffer( int character, unsigned char buffer[], unsigned int \* start, unsigned int \* end )**

- Přidává data do zásobníku.
- Vstupní parametry:
  - character – přidávaný byte
  - buffer – pole s uloženými byty
  - start - ukazatel na první platný byt
  - end – ukazatel na poslední platný byt
- Výstupní parametr:
  - 1 - při úspěšném vložení
  - 0 - při plném zásobníku

**int size\_of\_buffer( unsigned int \* start, unsigned int \* end )**

- Vypočítává velikost volného místa v zásobníku v závislosti na prvním a posledním platném bytu
- Výstupní parametr:
  - velikost volného místa

**int pop\_from\_buffer( unsigned char buffer[], unsigned int \* start, unsigned int \* end )**

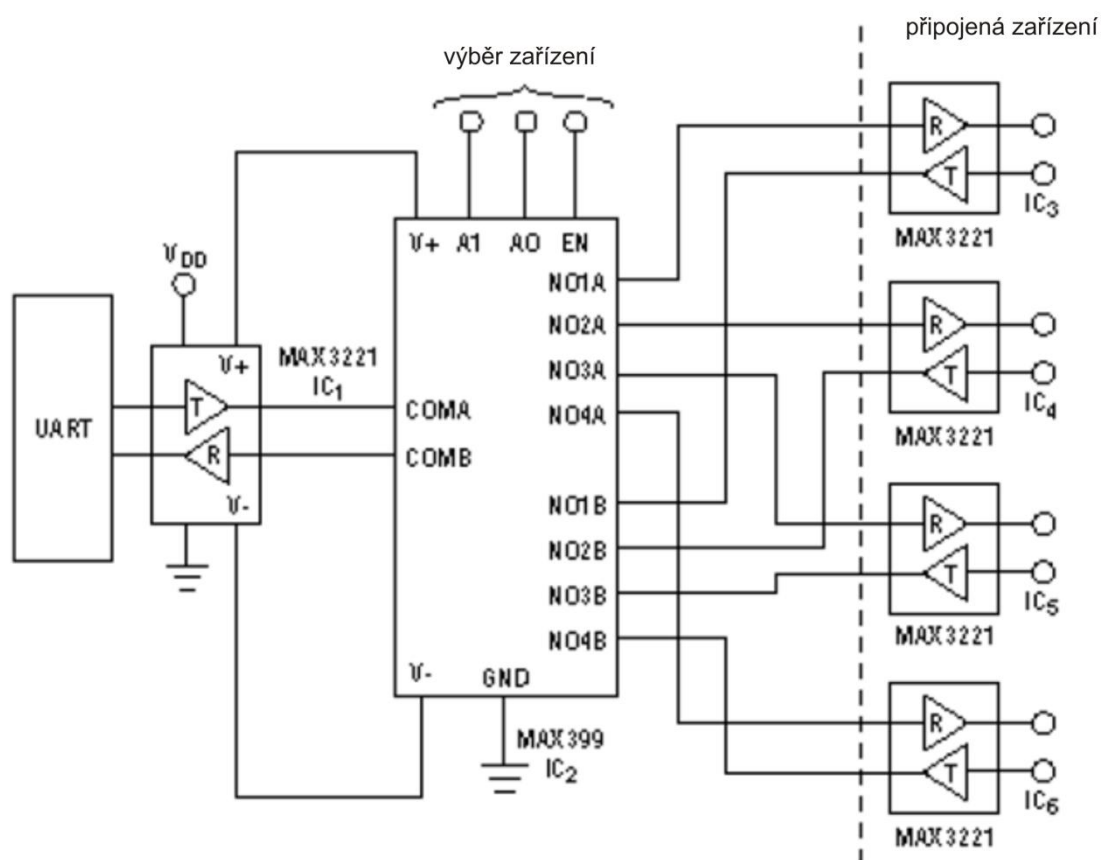
- Vyjímá první platný byt a posunuje ukazatel na další.
- Vstupní parametry:
  - buffer – pole s uloženými byty
  - start - ukazatel na první platný byt
  - end – ukazatel na poslední platný byt
- Výstupní parametr:
  - vrací hodnotu odebraného bytu
  - -1 - v případě prázdného zásobníku

## 5. ROZHRANÍ PRO KOMUNIKACI MEZI SBĚRNICEMI CAN A RS232

V této části je popsáno řešení pro komunikaci mezi řídicím procesorem a řízenou kamerou. Je to nejdůležitější část této práce, bez kterého by komunikace nebyla možná. Je zde popsán postup při návrhu a následné softwarové řešení.

### 5.1. NÁVRH ELEKTRONIKY PŘEVODNÍKU

Pro zhotovení rozhraní pro komunikaci mezi sběrnicemi CAN a RS232 neexistuje žádný univerzální obvod. Je to asi ovlivněno poměrně složitou komunikací CAN sběrnice s nutností nastavení více parametrů. Z tohoto důvodu musel být použit mikroprocesor naprogramovaný pro tyto účely. Jako hlavní částí je mikroprocesor Philips LPC2119, na kterém stojí několik školních projektů a samozřejmě i mnoho zkušeností. Jako podklad byla použita destička SPEJB pro ovládání kloubů noh robota, z které byla využita napájecí část a řadič pro CAN sběrnici. K této části byl přidán dvojitý budič sběrnice RS232 pro využití obou UART portů procesoru.



OBR. 6 - MULTIPLEX SÉRIOVÉHO ROZHRANÍ

K UART1 byl ještě připojen analogový multiplex umožňující přepínání mezi čtyřmi periferními zařízeními komunikujících na sběrnici RS232. Toto zapojení pochází ze stránek firmy MAXIM a je zobrazeno na Obr. 6. Díky tomu vznikl nápad přepínání mezi několika periferními zařízeními. Obvod MAX399, který tvoří zmíněný multiplex, byl nahrazen obvodem HCF 4052 CMOS, který má stejnou funkci a je běžně k dostání.

### **5.1.1. POUŽITÉ INTEGROVANÉ OBVODY**

- Mikroprocesor **Philips LPC2119FBD64** - tento mikroprocesor je založen na jádře ARM7TDMI-S s 16kB SRAM paměti 128 kB flash paměti, pracovní frekvenci dosahuje hodnot 60MHz. Programování je možné provádět přes rozhraní ISP nebo CAN zavaděče, CAN rozhraní s možností filtrování vstupních zpráv. 2 UART porty, PWM výstupy, 10 bitový AD převodník, 32 bitový čítač mnoho IO portu s možností přerušování.
- Budič CAN sběrnice **Philips PCA82C250** - pro vysokorychlostní komunikaci 1Mbit/s na standardu CAN 2.0A a CAN 2.0B.
- Budič RS232 sběrnice **Maxim MAX3232CSE** - obsahující dvojitý převodník úrovní TTL pro sběrnici RS232.
- Multiplex **ST HCF4052** - s 2 vstupy a 8 výstupy. A možností odpojení všech výstupů. Tento obvod využívá napájení z obvodu MAX3232.
- Stabilizátor napětí **ST LE50CD** - stabilizuje vstupní napětí na hodnotu 5V s možností vstupního napětí do 16V a maximálním proudem 150mA.
- Stabilizátor napájecího napětí **TI TPS73HD318** - pro mikroprocesor a ostatní integrované obvody. Výstupní napětí je 3.3V a 1.8V. Signál reset sloužící pro resetování procesoru při připojení napětí.

### **5.1.2. NÁVRH SCHÉMATU ZAPOJENÍ A DESKY PLOŠNÉHO SPOJE**

Na návrh schématu a desky plošného spoje byl využit program EAGLE 4.16r2 light. Tento program jsem zvolil pro jeho velkou rozšířenost jak ve školních projektech, tak v široké veřejnosti, a pro možnost doplnění programu o balíček pro generování dat pro výrobu plošného spoje na školní CNC frézce. Tato metoda je velmi výhodná pro testovací účely pro její rychlost a snadnost oproti zadání zakázky profesionální firmě. Metodu výroby jsem si osvojil hned při prvním návrhu. Při pozdějším osazení a naprogramování se ukázaly nějaké problémy se zapojením, a tím bylo nutno vyrobit nový plošný spoj, což ale nebyl problém 10 dnů, ale jen několika hodin.

Schéma zapojení je složeno z několika bloků:

- Napájecí část – obsahuje vstupní stabilizátor a stabilizaci napětí pro napájení integrovaných obvodů.
- Hlavní mikroprocesor.
- Komunikační část - pro CAN sběrnici.
- Komunikační část - pro sběrnici RS232 obsahující multiplex.
- Odrušovací kondenzátory - pro mikroprocesor.
- Sada led diod - sloužících pro vizuální kontrolu stavu procesoru, přičemž dvě signalizují vybraný výstupní port multiplexu.
- Konektory – pro připojení napájení, vývodu RESET a ISPSEL a jednotlivých sériových zařízení. Jejich popis, viz Tabulka 2 - Zapojení vyvedených konektorů. Tx1a, Tx1b a Tx1c označují vyvedený UART port 1 s možností přepínání mezi vývody a, b a c. Mezi každými konektory je vyvedena zem. Toto zapojení umožňuje prohození vývodů Tx a Rx, které jsem využil při testování z důvodů různých zapojení propojovacích kabelů (křížené, nekřížené atd.).

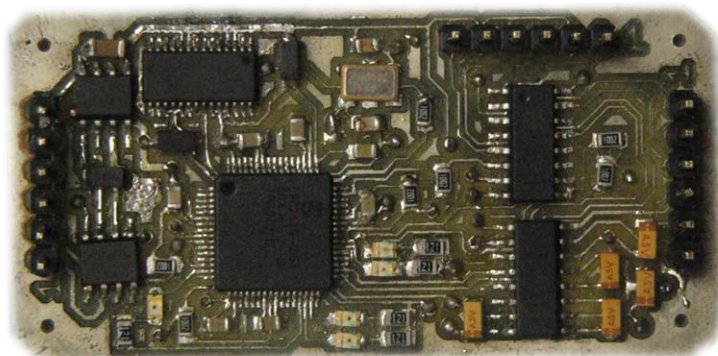
Konektor	Pin					
	1	2	3	4	5	6
<b>SV1</b>	CAN_L	CAN_H	ISPSEL	RESET	GND	VCC
<b>SV2</b>	Tx1a	GND	Rx1a	Rx1b	GND	Tx1b
<b>SV3</b>	Tx1c	GND	Rx1c	Rx0	GND	Tx0

**TABULKA 2 - ZAPOJENÍ VYVEDENÝCH KONEKTORŮ**

Navrhnuté schéma zapojení, viz příloha 1. Deska plošného spoje, viz příloha 2, je navrhnutá pro dvoustranný plošný spoj. Součástky jsou osazeny z horní strany. Tato deska, jak již bylo zmíněno, byla vyrobena na CNC frézce. K povrchové úpravě bylo použito chemické pocínování. Kompletní seznam použitých součástek, jejich hodnot a velikostí pouzder, viz Tabulka 3. Veškeré součástky byly vybrány pro montáž povrchového pájení SMD z důvodu úspory místa a snadného získání požadovaných součástek. Výsledná osazená deska, viz Obr. 7 přední strana a Obr. 8 zadní strana.

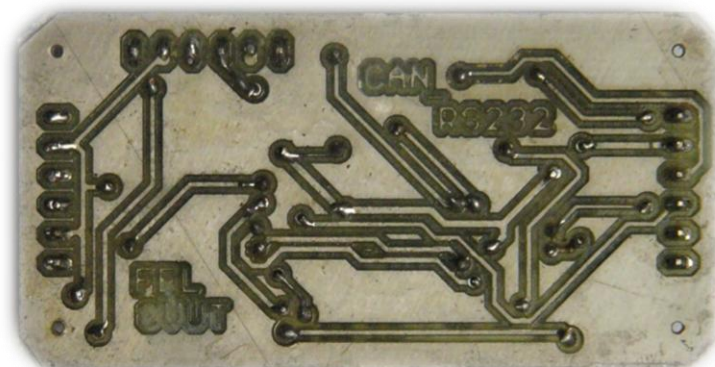
Označení	typ/hodnota	pouzdro	poznámka
<b>C1</b>	2u2	1206	keramický
<b>C2,C3</b>	56p	0805	keramický
<b>C4,C5,C6,C7,C8, C9,C10</b>	u1	0805	keramický
<b>C11,C12,C13</b>	4u7	A	tantalový
<b>C14,C15,C16,C17, C18</b>	u1	A	tantalový
<b>SV1,SV2,SV3</b>		Konektorová lišta s roztečí 2.5 mm	
<b>R1 - R7</b>	10k	0805	
<b>IC1</b>	LE50CD	SO08	ST
<b>IC2</b>	TPS73HD318	TSSOP28	TI
<b>IC3</b>	PCA82C250	SO08	Philips
<b>IC4</b>	HCF 4052	SO16	ST
<b>IC5</b>	MAX3232CSE	TSSOP20	Maxim
<b>IC6</b>	LPC2119FBD64	LQFP64	Philips
<b>X1</b>	12MHz	5032	
<b>LED1 - LED5</b>		0805	2x zelená, 2x žlutá , 1x červená
<b>R8 - R12</b>	120	0805	

**TABULKA 3 - SEZNAM POUŽITÝCH SOUČÁSTEK**



**OBR. 7 - OSAZENÁ DESKA ROZHRANÍ CAN - RS232 VRCHNÍ STRANA**





**OBR. 8 - OSAZENÁ DESKA ROZHRAŇÍ CAN - RS232 SPODNÍ STRANA**

## **5.2. PROGRAMOVÉ ŘEŠENÍ**

Mikroprocesor obsahuje rozhraní ISP, a tak byl procesor programován na osazené desce pomocí vyvedených signálů RESET a ISPSEL přes sériové rozhraní UART0. Toto programování bylo zprostředkováno programem Philips Flash Utility určeným pro programování procesoru značky Philips poskytnutým výrobcem. Tento program využívá přeloženého programu a jeho výstupu ve formátu HEX. Samotné programování bylo prováděno v programovacím prostředí Keil  $\square$  Vision3. Tento program je omezen maximální velikostí kódu, kterou při této aplikaci nebylo problém dodržet. Pro tento programovací jazyk je poskytnuto velké množství ukázkových programů pro různé aplikace a velké množství driverů pro komunikaci po sběrnicích. Proto jsem využil knihovnu pro komunikaci CAN, která obsahovala veškeré možnosti nastavení sběrnice.

### **5.2.1. POPIS HLAVNÍHO PRINCIPU PROGRAMU**

Celý program spočívá ve čtení přicházejících dat po sběrnicích CAN a RS232. Tato data jsou ukládána do zásobníků stejně jako u řídicí části. Pro komunikaci se sběrnicí CAN byl použit zmíněný driver, který umožňuje nastavení filtru přijímaných zpráv a celé inicializace CAN sběrnice. Největším problémem mezi těmito sběrnicemi je rozdíl ve velikosti přenášených dat. Přijatá data z CAN sběrnice obsahuje 8 bytů dat. Ty jsou ukládány a později byt po bytu odesílány na sériové rozhraní. Pro opačnou stranu jsou jednotlivé byty ukládány do pole, které je následně ukládáno do datových polí CAN zprávy. Při naplnění zprávy je paket odeslán. Při nenaplnění celého datového pole je nastaven počet platných bytů datového pole a následně odeslán. Tento převodník tedy zatím převádí jen na jeden sériový port. Později bude doplněn o přepínání i na ostatní zařízení. To může být zajištěno například rozlišováním zpráv pomocí identifikační značky v CAN paketu. Ale pro zadaný problém jedné kamery toto nastavení postačuje.

### 5.2.2. POPIS DŮLEŽITÝCH FUNKCÍ PROGRAMU

Jednotlivý popis funkcí, jejich vstupní a výstupní parametry, které byly naprogramovány pro potřeby převodníku.

#### **int main( void )**

- Hlavní vlákno programu - vyvolá funkce inicializace a poté cyklicky testuje plnost zásobníku. Při nalezení dat zajišťuje přeposílání dat. Obsahuje také část výpisu chybových stavů CAN sběrnice.
- Výstupní parametr:
  - 1 - při úspěšném ukončení

#### **void init( void )**

- Nastavuje veškeré registry procesoru. Nastavuje výstupní IO, CAN , UART porty, volá funkce pro nastavení CAN sběrnice pro inicializaci a nastavení datové rychlosti, pracovní mód CAN sběrnice a nastavuje vstupní filtr CAN sběrnice. Nastavuje parametry sběrnice UART. Poté povoluje přerušení pro potřebné porty.

#### **void IRQ\_UART0( void )**

- Odchytává příchozí data a ukládá je do zásobníku.

#### **void IRQ\_CAN1Rx( void )**

- Odchytává přerušení vzniklé příchozími daty, nebo chybovými zprávami. Data ukládá do zásobníku, která jsou zpracována v hlavním vláknu.

#### **int push\_to\_uart\_buffer( int character )**

- Vkládá data do zásobníku
- Vstupní parametr:
  - character – byte ukládaný do zásobníku
- Výstupní parametr:
  - 1 - při úspěšném vložení
  - 0 - při plném zásobníku

#### **int size\_uart\_buffer(void )**

- Vypočítává velikost volného místa v zásobníku
- Výstupní parametr:
  - velikost volného místa v zásobníku

**int pop\_from\_uart\_buffer( void )**

- Vrací první vložený byte.
- Výstupní parametr:
  - hodnota prvního platného bytu
  - -1 - při prázdném zásobníku

## **6. ZÁVĚREČNÉ SHRUTÍ**

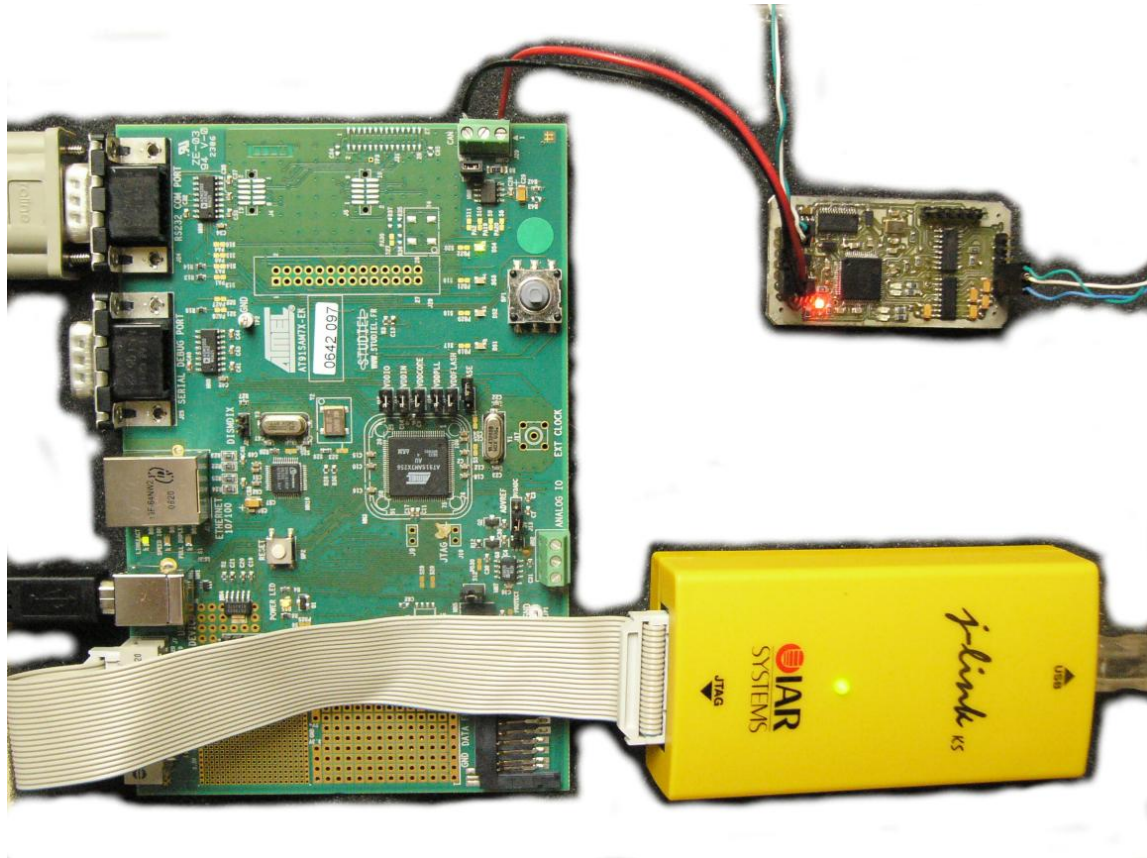
Při výsledném ověření požadované funkce byl celkový obvod propojen a připojen ke konektorům kamery prozatím bez použití kluzných kroužků mezi procesory na CAN sběrnici, které ještě nebyly připojeny.

Při testování bohužel nebyla možnost připojení výstupního signálu obrazu, a tak ověření funkce spočívalo jen na komunikaci mezi kamerou a řídicím mikroprocesorem, kde bylo možno sledovat odpovědi kamery na jednotlivé příkazy. Jedinou vizuální kontrolou kamery zůstala možnost měnit přiblížení, při němž bylo možné vidět pohybující se objektiv kamery. Výsledné zapojení je možné vidět na třech obrázcích. Obr. 9 ukazuje kamerovou základnu, v jejíž levé části je zabudovaná naše řízená kamera. Obr. 10 ukazuje propojení řídicí desky s převodníkem a vyvedeným kabelem do kamery a PC pro výpis dat ukázaný na Obr. 11.

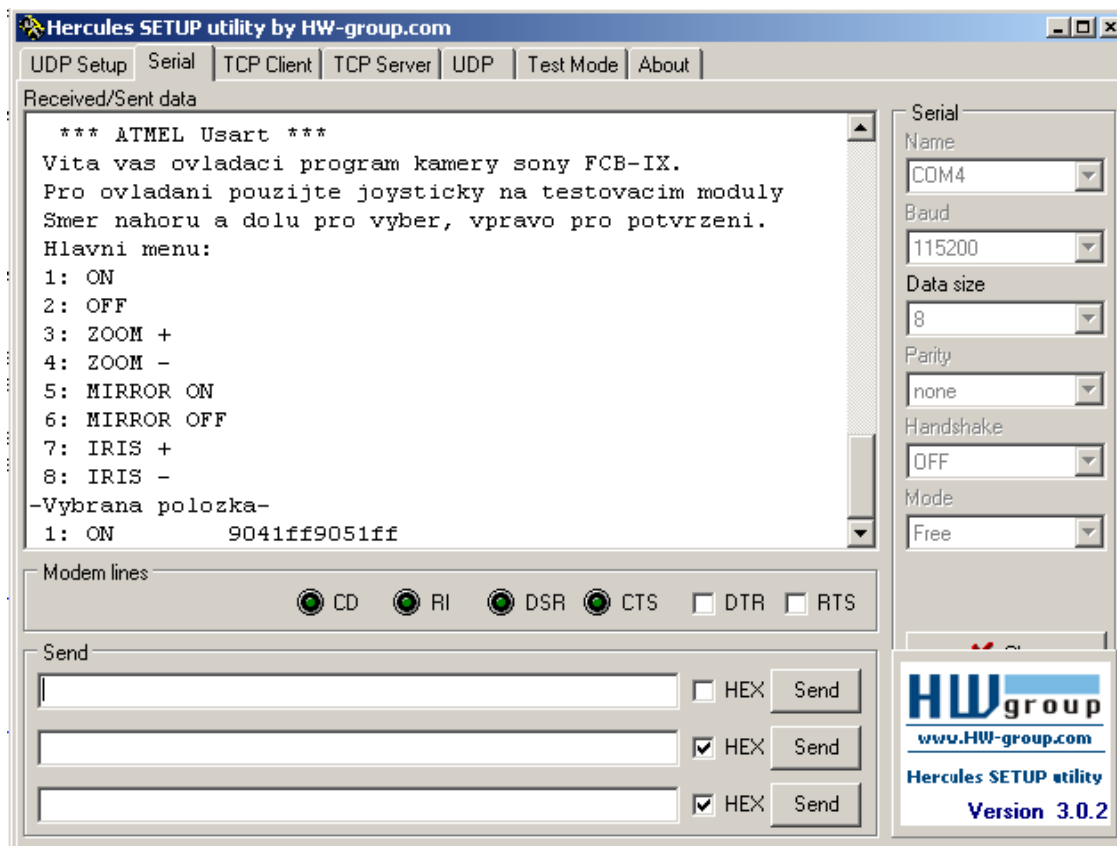


**OBR. 9 - KAMEROVÁ ZÁKLADNA**

Díky této práci jsem se seznámil s hardwarovým problémem, který jsem musel řešit pomocí mikroprocesoru naprogramovaného k určité aplikaci, seznámil jsem se s programováním mikroprocesorů a zároveň jsem si ověřil, že dokážu vyřešit určitý aplikační problém. Seznámil jsem se s výrobou plošných spojů a značně jsem si rozšířil znalost technických výrazů v anglickém jazyce.



**OB. 10 - ŘÍDÍČÍ MIKROPROCESOR ZAPOJENÝ S PŘEVODNÍKEM**

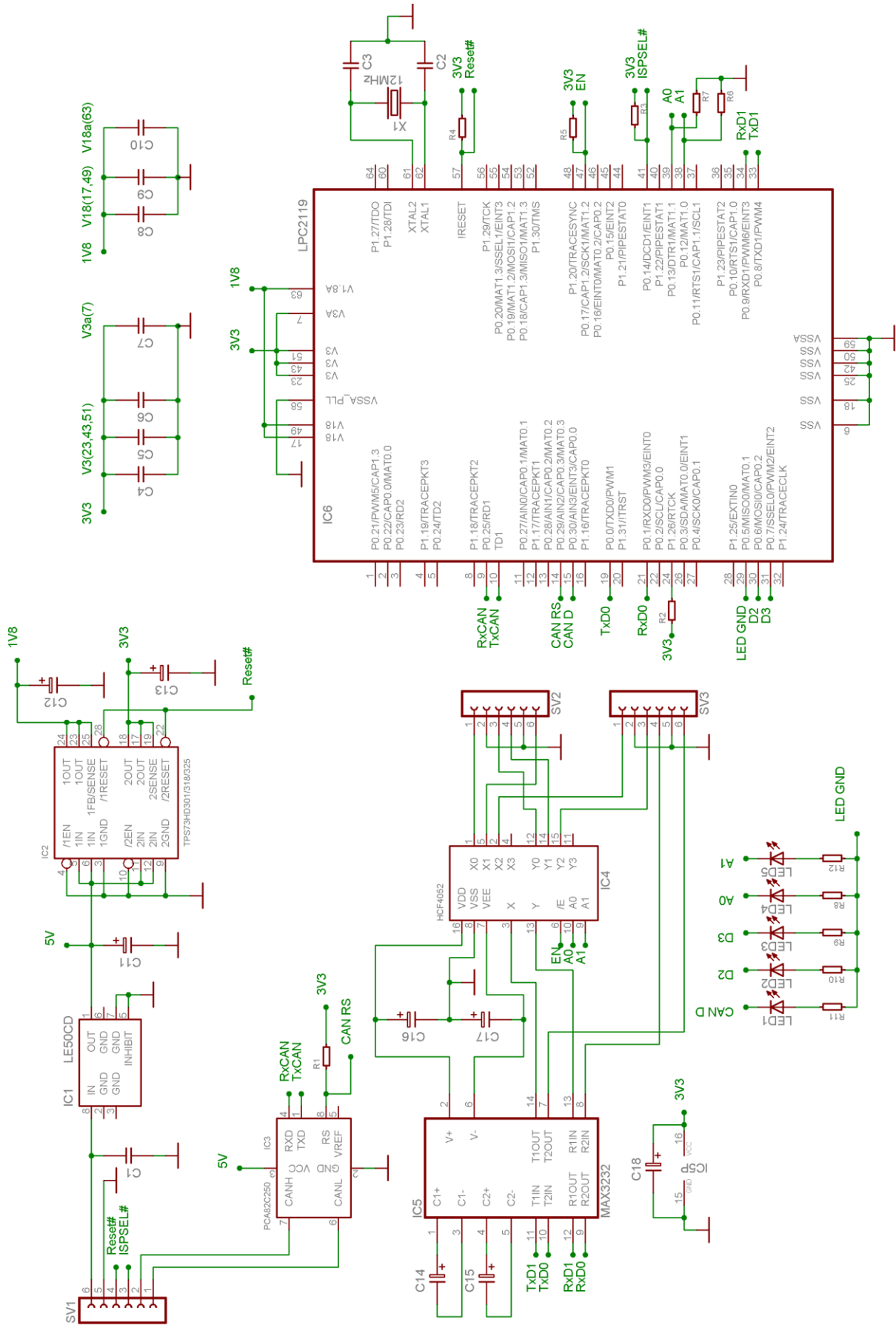


**OBR. 11 - VÝPIS OVLÁDACÍHO PROGRAMU S VÝPISEM ODPOVĚDI OD KAMERY**

## 7. LITERATURA

1. Atmel Data sheet AT91SAM7X  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc6120.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc6120.pdf)
2. Informace o CAN protokolu  
<http://hw.cz/rozhrani/ART1173-Aplikovani-sbernice-CAN.html>
3. Základní informace o program EAGLE  
<http://www.fj.tul.cz/~kes/zipprogil/des/maneag.html>
4. Philips Data sheet LPC 2119  
[http://www.nxp.com/acrobat/download/datasheet/LPC2109\\_2119\\_2129\\_5.pdf](http://www.nxp.com/acrobat/download/datasheet/LPC2109_2119_2129_5.pdf)
5. Philips User Manual LPC2119  
<http://www.nxp.com> – UM\_LPC21XX\_LPC22XX\_2;  
LPC2119/2129/2194/2292/2294 USER MANUAL
6. MAXIM Data sheet MAX3222,MAX3232  
<http://datasheets.maxim-ic.com/en/ds/MAX3222-MAX3241.pdf>
7. MAXIM Application note 588, RS232 multiplex  
<http://pdfserv.maxim-ic.com/en/an/AN588.pdf>
8. ST Data sheet HCF 4052  
<http://www.st.com/stronline/books/pdf/docs/8214.pdf>
9. ST Data sheet LE50CD  
<http://www.ortodoxism.ro/datasheets/stmicroelectronics/2573.pdf>
10. Philips Data sheet PCA82C250  
<http://www.ortodoxism.ro/datasheets/philips/PCA82C259.pdf>
11. TI Data sheet TPS73HD318  
<http://www.ortodoxism.ro/datasheets/texasinstruments/tps73hd318.pdf>
12. Návod pro kameru FCB-IX11  
FCB-IX11AP\_navod.pdf – získaný ve škole.
13. Kračmar S., Vogel J., Programovací jazyk C, ČVUT Fakulta strojní

# PŘÍLOHA 1 - SCHÉMA ZAPOJENÍ ROZHRAŇÍ CAN - RS232



## PŘÍLOHA 2 - DESKA PLOŠNÉHO SPOJE ROZHRANÍ CAN - RS232

