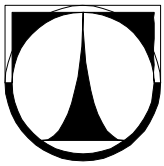
TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií



BAKALÁŘSKÁ PRÁCE

Připojení karty SD/MMC k jednočipovému mikroprocesoru.

Usage of the SD/MMC card with singlechip microcontroller

Liberec 2007 **Jan Mazura**

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídící systémy

Připojení karty SD/MMC k jednočipovému mikroprocesoru

Usage of the SD/MMC card with singlechip microcontroller

Bakalářská práce

Autor: **Jan Mazura**
Vedoucí BP práce: Ing. Tomáš Martinec
Konzultant: Ing. Tomáš Pluhař

V Liberci 3. 1. 2007 **Jan Mazura**

Připojení karty SD/MMC k jednočipovému mikroprocesoru

Abstrakt

Cílem této bakalářské práce je seznámit se paměťovými kartami SD/MMC, navrhnout jejich připojení k jednočipovému mikroprocesoru, vytvořit knihovnu funkcí pro práci s kartou a ukládání dat v nějakém jednoduchém vlastním formátu.

Dalším úkolem je vytvořit ukázkovou aplikaci, která bude využívat paměťovou kartu jako prostředek k ukládání dat. Na základě těchto požadavků byla v programovacím jazyce ANSI C vytvořena knihovna funkcí, obsahující funkce pro nezbytnou činnost karty

A nakonec je zhotovena ukázková aplikace pro otestování správné funkčnosti navrhnutého a implementovaného programu.

Klíčová slova: Jednočipový mikroprocesor, paměťová karta, SPI rozhraní.

Usage of the SD/MMC card with singlechip microcontroller

Abstract

The aim of this Bachelor Thesis is to be used the memory SD/MMC, to design their connection to single-chip microprocessor, to create function library for working with the card and for data storage in an own simple format.

The next task consists in creating of demo application that is to be used the memory card as a device for data storage. Based on these requirements there was created a function library in ANSI C program language.

Finally is made sample application for testing of the right functionality of the designed and implemented program.

Key words: Singlechip microcontroller, memory card, SPI interface

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé BP a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Poděkování

Touto cestou bych chtěl poděkovat především vedoucímu bakalářské práce Ing. Tomáši Martincovi za odborné vedení, trpělivost a pomoc při zpracování této bakalářské práce.

Samozřejmě mé díky patří také celé mé rodině za všestrannou podporu při mém vysokoškolském studiu.

Obsah

Abstrakt 5
Prohlášení 6
Poděkování 7
Obsah 8
Seznam obrázků 9
Úvod 10

1. PAMĚŤOVÉ KARTY 11

1.1 Paměťové karty - úvod 11

1.2 Popis SD a MMC karty 11
1.2.1 Popis pouzder a vývodů SD a MMC karty 13
SD Karta: 13
MMC Karta: 14
1.2.2 Popis komunikačního protokolu 14
Formát odpovědi R1 15
Průběh komunikace s kartou 16
Příkaz Reset(CMD0) 17
Příkaz Init(CMD1) 18
Příkaz Read CSD(CMD9) 18
Příkaz Read CID(CMD10) 19
Příkaz Read(CMD17) 20
Příkaz Write(CMD24) 20

2. ŘÍDÍCÍ PROCESOR 22

2.1 Volba řídícího procesoru 22

2.2 Popis zvoleného procesoru 22
2.2.1 Základní parametry procesoru 22
2.2.2 Programování procesoru pomocí UART 23
2.2.3 Popis vytváření programu 24

2.3 Popis komunikačního rozhraní mezi procesorem a ostatními obvody 24

3. POPIS VYTVOŘENÉ KNIHOVNY FUNKCÍ 27

3.1 Popis funkcí obsažených v knihovně 27

3.2 Popis systému ukládání dat 29

4. POPIS VZOROVÉ APLIKACE 31

4.1 Jednotlivé součásti 31

4.2 Ovládací software 33

4.3 Možné použití aplikace 33

5. ZÁVĚR 34
Použitá Literatura 35

Seznam obrázků

Obrázek 1.1 SD paměťová karta 11
Obrázek 1.2 MMC paměťová karta 11
Obrázek 1.3 SD karta 13
Obrázek 1.4 MMC Karta 14
Obrázek 1.5 Formát odpovědi R1 15
Obrázek 1.6 Průběh komunikace s kartou 16
Obrázek 1.7 Příkaz Reset(CMD0) 17
Obrázek 1.8 Příkaz Init(CMD1) 18
Obrázek 1.9 Příkaz CSD Read(CMD9) 19
Obrázek 1.10 Příkaz CID Read(CMD10) 19
Obrázek 1.11 Příkaz Read(CMD17) 20
Obrázek 1.12 Příkaz Write(CMD24) 21
Obrázek 2.1 Koncepce systému se sběrnicí SPI 25
Obrázek 2.2 Propojení obvodů Master a Slave 25
Obrázek 2.3 Význam parametrů CPOL a CPHA 26
Obrázek 3.1 Struktura dat na paměťové kartě 29
Obrázek 4.1 Úprava napájení a napěťových úrovní signálů 32

Úvod

Paměťové karty jsou velmi malé, přenositelné a rychlé prostředky pro ukládání různých dat. Základem karet je paměť typu flash. Vyznačuje se velmi nízkou spotřebou energie, což paměť předurčuje k používaní v bateriemi napájených aplikacích. Paměťové karty nacházejí uplatnění v digitálních fotoaparátech, digitálních kamerách, mp3 přehrávačích, a dalších podobných zařízeních. V této práci byly karty použity pro ukládání naměřených dat.

Jedním z cílů bakalářské práce bylo seznámit se s paměťovými kartami, konkrétně typu SD a MMC. Tím se zabýváme v první kapitole. Kapitola popisuje fyzické vlastnosti karet, komunikační protokol, průběh komunikace s kartou a také jednotlivé příkazy.

Následující kapitola popisuje výběr procesoru, kritéria výběru, vlastnosti vybraného procesoru a také komunikační rozhraní mezi procesorem a ostatními obvody.

Dalším cílem bylo vytvořit knihovnu funkcí pro práci s paměťovou kartou. Tato knihovna je popsána ve třetí kapitole. Obsahuje popis vstupních proměnných funkcí, návratové hodnoty a také ukázku použití funkcí. Dále popisuje princip ukládání dat.

Posledním cílem bylo vyrobit a naprogramovat ukázkovou aplikaci. Aplikace je tvořena zhotoveným přípravkem a ovládacím programem. Popisem této aplikace se zabývá poslední kapitola.

1. Paměťové karty

1.1 Paměťové karty - úvod

Existuje několik druhů paměťových karet. Mezi ty nejčastěji používané patři: CompactFlashI/II, Memory Stick, MultiMediaCard(MMC), SmartMedia, Secure Digital(SD), xD-Picture. Díky požadavkům na menší velikost (hlavně kvůli použití v mobilních telefonech) vznikly ještě zmenšené verze SD a MMC karet a to: RS-MMC/MMC mobile a miniSD/microSD. Co se týká rychlostí, nejvyšší přenosové rychlosti dosahuje karta CompactFlash, 40MB/s [3]. Naopak karta s nejpomalejším přenosem je SmartMedia, 2 MB/s. SD karta dosahuje přenosových rychlostí až 20MB/s. Srovnáme-li karty podle ceny, mezi nejlevnější patří SD i MMC karta. Naopak mezi nejdražší patří karta MemoryStick.

1.2 Popis SD a MMC karty

Obrázek 1.1 SD paměťová karta

Obrázek 1.2 MMC paměťová karta

SD karta je založena na základě formátu MMC karty, ale většina je trochu silnější než MMC. SD karty mají nejčastěji rozměry 32 x 24 x 2,1 mm, můžeme se ale setkat s SD kartou, jejíž tloušťka je jen 1,4 mm jako MMC karta. Obě karty mají stejný tvar i stejné rozmístění kontaktních ploch. Lze tedy v zařízeních, navržených pro SD kartu, použít místo SD karty MMC kartu a naopak. Existuje několik rozdílů ve kterých se karty liší.

K určení rychlosti se používá stejný systém jako u CDROM, tj. v násobcích 150kB/s. Základní rychlost karet je 6x, tedy 900kB/s. U lepších karet je udávána rychlost 66x(10MB/s). Nejlepší(také nejdražší) karty dosahují rychlostí 133x(20MB/s).

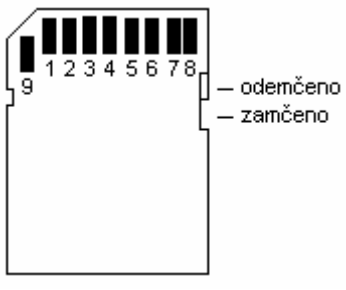
Silnější pouzdro a vhodné vnitřní uspořádání zajišťují lepší odolnost SD karty vůči elektrostatickému rušení, čímž se SD karta stává spolehlivější než MMC karta.

Pouzdro SD karty obsahuje přepínač se dvěma polohami: zamčeno a odemčeno. Funguje stejně jako u disket do PC, v poloze zamčeno není možné na kartu ukládat nebo z karty mazat data.

SD karta má navíc DRM(digitální správa práv), tato věc se však příliš nevyužívá.

1.2.1 Popis pouzder a vývodů SD a MMC karty

SD Karta:



Obrázek 1.3 SD karta

Pin	SD 4 - bitový mód		SD 1 – bitový mód		SPI mód	
	Označení	Význam	Označení	Význam	Označení	Význam
1	CD/DAT[3]	Data 3	NC	Nevyužito	CS	Výběr obvodu
2	CMD	Příkaz	CMD	Příkaz	IN(MOSI)	Datový vstup
3	VSS1	Zem	VSS1	Zem	VSS1	Zem
4	VDD	Napájení	VDD	Napájení	VDD	Napájení
5	CLK	Hodiny	CLK	Hodiny	SCK	Hodiny
6	VSS2	Zem	VSS2	Zem	VSS2	Zem
7	DAT [0]	Data 0	DATA	Data	DO(MISO)	Datový výstup
8	DAT [1]	Data 1	IRQ	Přerušení	IRQ	Přerušení
9	DAT [2]	Data 2	RW	Read Wait	NC	Nevyužito

Tabulka 1-1 Popis vývodů SD karty

MMC Karta:

Obrázek 1.4 MMC Karta

Pin	MultiMediaCard mód		SPI mód	
	Označení	Význam	Označení	Význam
1	RSV	Nevyužito	CS	Výběr obvodu
2	CMD	Příkaz	IN(MOSI)	Datový vstup
3	VSS1	Zem	VSS1	Zem
4	VDD	Napájení	VDD	Napájení
5	CLK	Hodiny	SCK	Hodiny
6	VSS2	Zem	VSS2	Zem
7	DAT0	Data	DO(MISO)	Datový výstup

Tabulka 1-2 Popis vývodů MMC karty

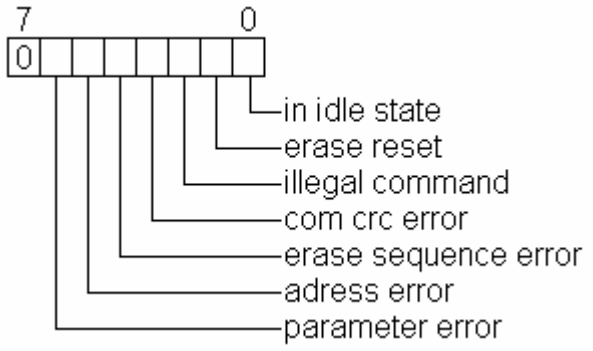
1.2.2 Popis komunikačního protokolu

V této kapitole je čerpáno z publikací [1] a [2], kde jsou jednotlivé vlastnosti vysvětleny podrobněji. U obou karet nás zajímá právě mód SPI, protože je u obou karet stejné rozvržení pinů. Tento mód je nejprve nutné aktivovat. Po zapnutí je totiž SD karta ve 4 – bitovém módu, MMC karta v MultiMediaCard módu. Komunikace s paměťovou kartou probíhá na základě příkazů. Každý příkaz má svůj význam. Příkaz se skládá ze šesti bytů. První byte určuje o jaký příkaz se jedná. Následující 4 byty obsahují data pro kartu jsou-li nutná (adresa žádaného sektoru), není-li tak jsou nulové. Poslední byte obsahuje tzv.

CRC(Cyclic Redundancy Codes). Slouží ke kontrole přenosu dat. Pro zjednodušení je (kromě příkazu Reset) roven 0xFF.

Po každém provedeném příkazu nás karta informuje o úspěchu nebo neúspěchu při provádění příkazu. Paměťová karta má tři různé formáty odpovědi R1, R2, R3. Typ odesílané odpovědi závisí na použitém příkazu. Protože v naší úloze používáme pouze základní příkazy, postačí nám znát pouze první formát odpovědi R1.

Formát odpovědi R1



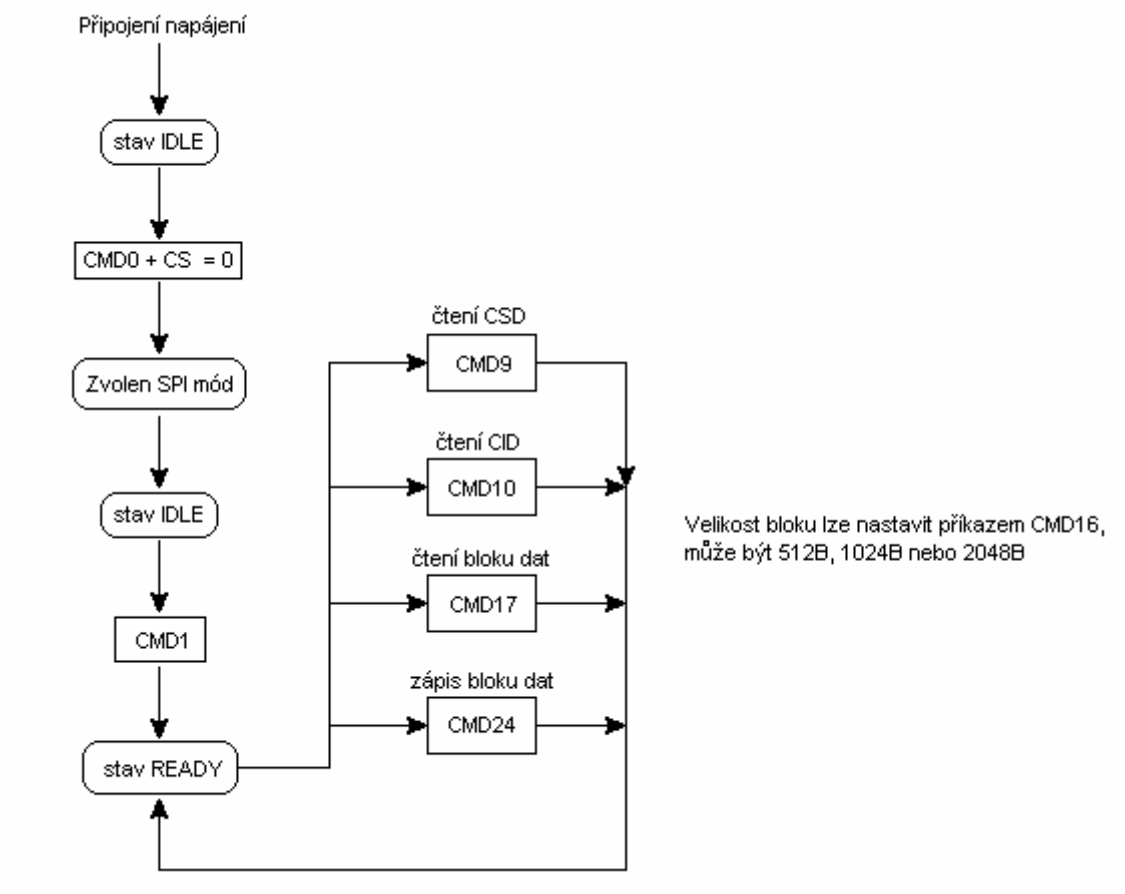
Obrázek 1.5 Formát odpovědi R1

- **In idle state:** Karta je ve stavu idle.
- **Erase reset:** Sekvence mazání nebyla provedena protože byl obdržen příkaz.
- **Illegal command:** Byl přijat neznámý kód příkazu
- **Communication CRC error:** Chybné CRC posledního příkazu
- **Erase sequence error:** Nastala chyba mazací sekvence.
- **Adress error:** Použita adresa ukazující na neexistující blok.
- **Parameter error:** Argument příkazu(adresa sektoru, velikost bloku) je mimo rozsah karty.

Průběh komunikace s kartou

Po zapnutí napájení vstoupí karta do režimu idle. Poté je nutné u karty zvolit komunikační rozhraní SPI. Dalším příkazem uvedeme kartu do stavu Ready.

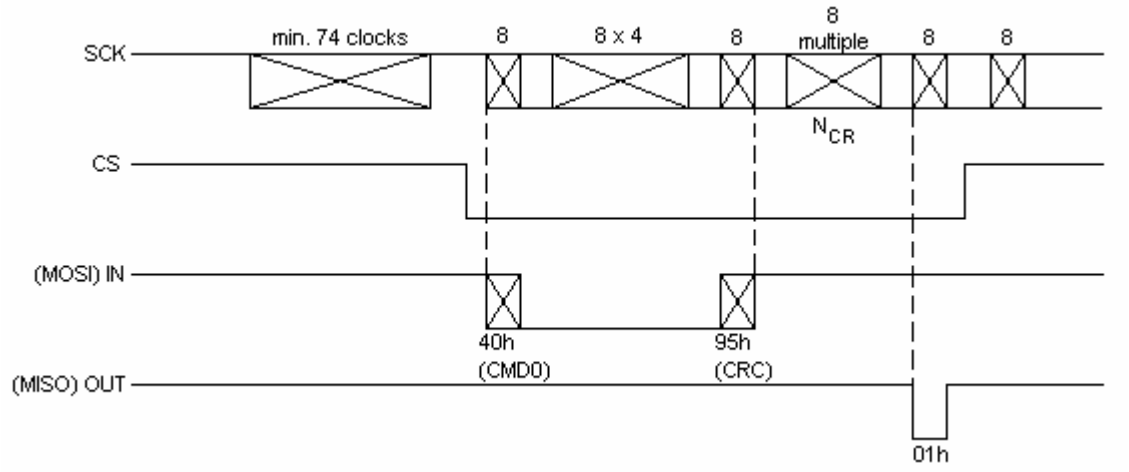
Z tohoto stavu je možné uskutečňovat čtení dat, zápis dat, čtení CSD a CID registru. Po každé z těchto činností se opět karta vrátí do režimu Ready.



Obrázek 1.6 Průběh komunikace s kartou

Příkaz Reset(CMD0)

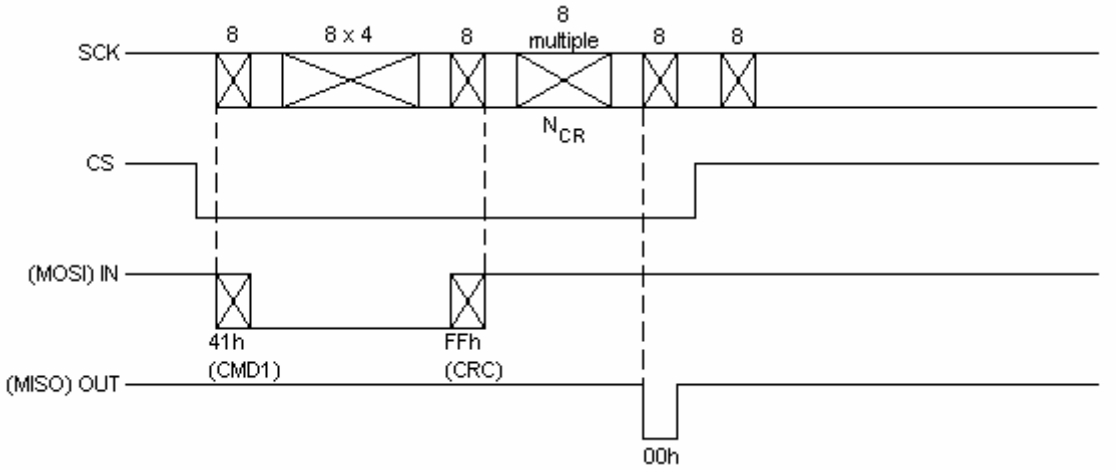
Tento příkaz zajišťuje inicializaci paměťové karty po zapnutí napájení a následné přepnutí komunikačního rozhraní do SPI módu. Po připojení napájení je nutné chvíli počkat než se ustálí napájecí napětí a napětí na paměťové kartě dosáhne hodnoty V_{DD} min. (u SD karty 2,7V). Poté procesor zahájí inicializační sekvenci. Ta se skládá ze souvislého proudu log. 1 na vodiči MOSI. Trvá maximálně 1 ms nebo 74 hodinových impulzů na vodiči SCK. 10 hodinových pulzů navíc odstraní synchronizační problémy(po 64 impulzech by měla být karta připravena pro komunikaci). Po této sekvenci odešle řídící procesor příkaz Reset(CMD0) přičemž přidrží CS na úrovni log.0. Tím je u karty zvoleno komunikační rozhraní SPI. Nakonec procesor čeká po dobu N_{CR} až karta potvrdí režim IDLE. U tohoto příkazu je nutné posílat správný byte CRC (0x95h) jinak dojde k chybě.



Obrázek 1.7 Příkaz Reset(CMD0)

Příkaz Init(CMD1)

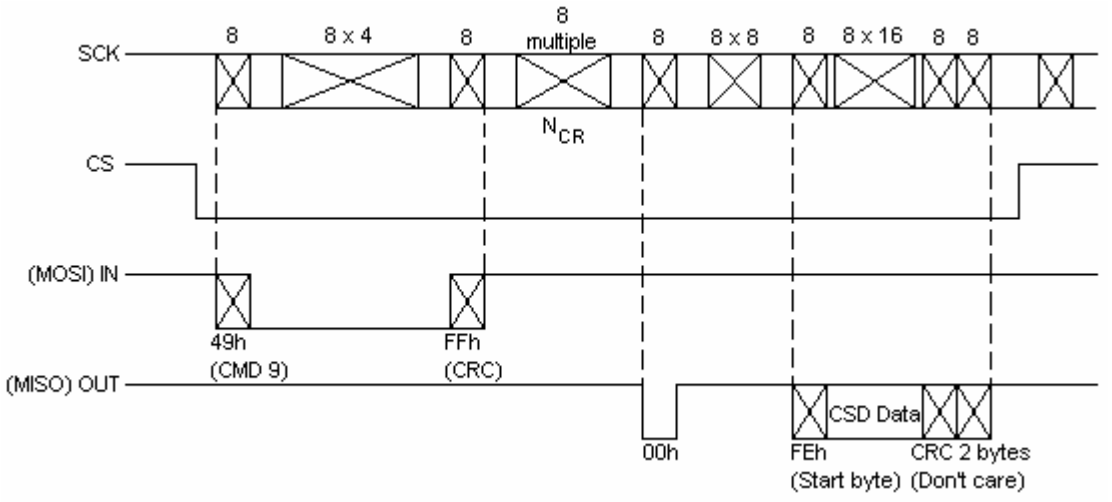
Tento příkaz uvede paměťovou kartu ze stavu idle do stavu Ready. Pokud není v tomto stavu, nelze s kartou komunikovat, karta ignoruje veškeré dění na sběrnici. Řídící procesor odešle příkaz CMD1. Poté čeká opět po dobu N_{CR} než karta potvrdí stav Ready odesláním odpovědi 0x00.



Obrázek 1.8 Příkaz Init(CMD1)

Příkaz Read CSD(CMD9)

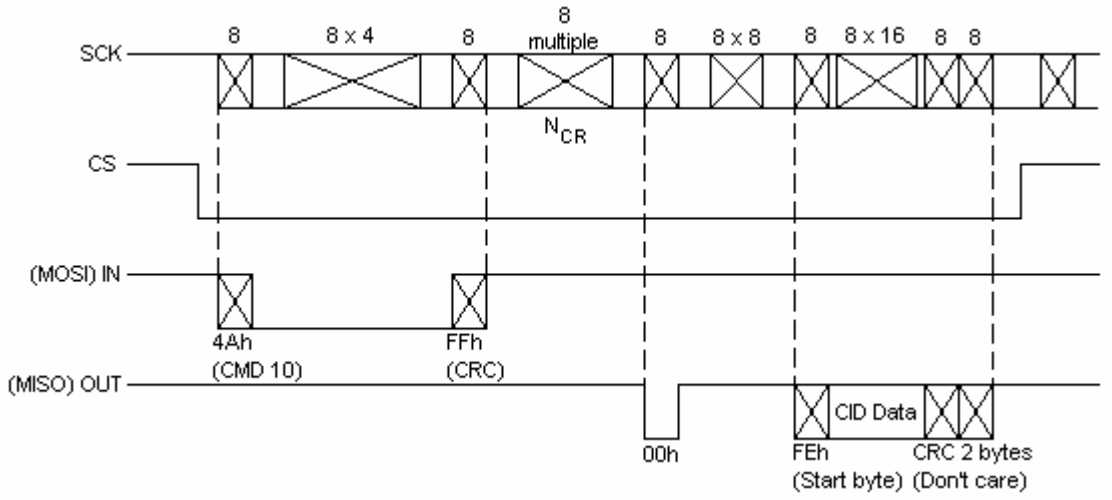
Registr CSD(Card Specific Data) obsahuje konfigurační informace potřebné pro přístup k datům na kartě. Například velikost paměťového prostoru, velikost bloku, atd. Nejprve procesor odešle příkaz CMD9, poté čeká opět po dobu N_{CR} dokud odpověď karty není 0x00. Dále čeká na příchod bytu 0xFE, následující byty jsou platnými daty které procesor ukládá do 16B dlouhého pole.



Obrázek 1.9 Příkaz CSD Read(CMD9)

Příkaz Read CID(CMD10)

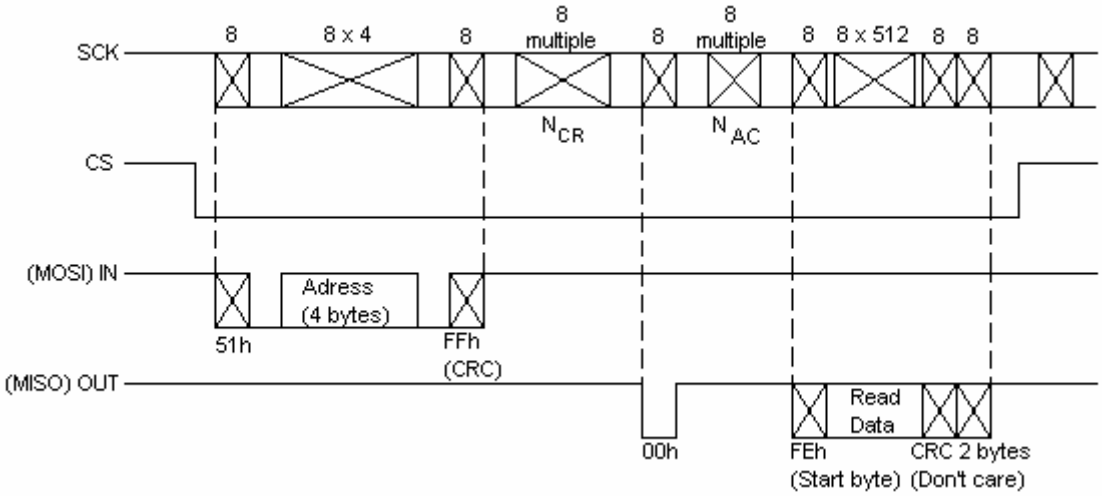
Registr CID(Card Identification) obsahuje informace výrobce. Sériové číslo, identifikační číslo, datum výroby, kód výrobce, atd. Nejprve procesor odešle příkaz CMD10, poté čeká opět po dobu N_{CR} dokud odpověď karty není 0x00. Dále čeká na příchod bytu 0xFE, následující byty jsou platnými daty které procesor ukládá do 16B dlouhého pole.



Obrázek 1.10 Příkaz CID Read(CMD10)

Příkaz Read(CMD17)

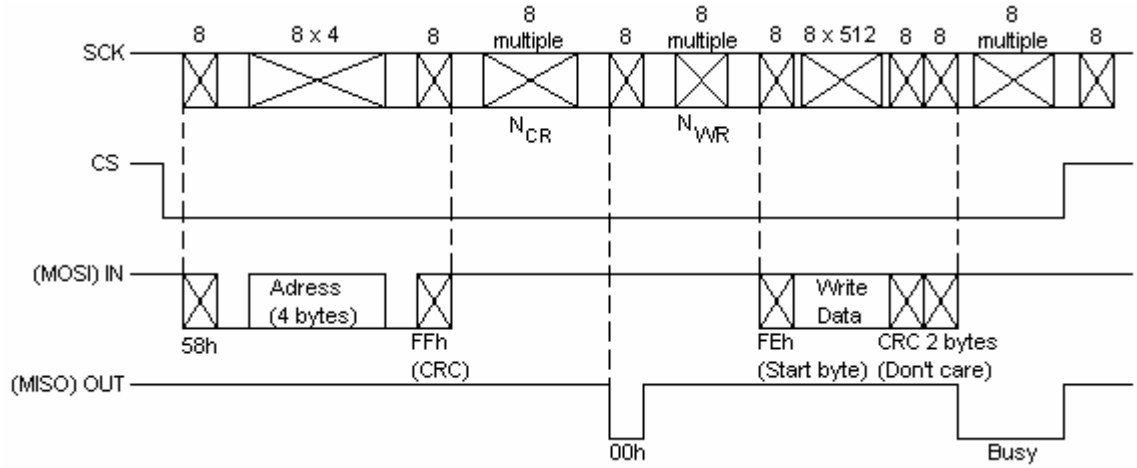
Příkaz CMD17 slouží ke čtení bloku dat (sektoru) z karty. Velikost tohoto bloku lze nastavit v rozsahu 512B,1024B nebo 2048B. Nastavení se provádí pomocí příkazu CMD16 kterým nastavíme proměnnou READ_BL_LEN která se nachází v registru CSD. Jako v předchozích případech procesor odešle příkaz pro čtení dat. Tentokrát je obsahem příkazu také adresa požadovaného sektoru. Dále čeká po dobu N_{CR} do odpovědi rovné 0x00. Následně čeká po dobu N_{AC} do příchodu startovacího bytu 0xFE. Poté ukládá platná data(počet závisí na nastavené velikosti bloku), vynechá 2 byty CRC.



Obrázek 1.11 Příkaz Read(CMD17)

Příkaz Write(CMD24)

Tímto příkazem zapisujeme data na kartu. Ukládání probíhá stejně jako při čtení přenosem celého bloku. Velikost bloku můžeme opět nastavit pomocí proměnné READ_BL_LEN. Jako v předchozích případech procesor odešle příkaz pro zápis dat. Tentokrát je obsahem příkazu také adresa požadovaného sektoru. Dále čeká po dobu NCR do odpovědi rovné 0x00. Následně čeká po dobu NAC. Po uplynutí této doby procesor odešle startovací byte 0xFE. Následující byty považuje karta za platná data. Po odeslání procesor odešle startovací byte 0xFE. Následující byty považuje karta za platná data. Dále je ještě nutné odeslat 2 byty CRC. Oba jsou rovny hodnotě 0xFF.



Obrázek 1.12 Příkaz Write(CMD24)

Symbol	Počet hodinových impulzů		Popis
	Min	Max	
N_{CR}	2	64	Počet cyklů mezi příkazem a odpovědí
N_{AC}	2	TAAC+NSAC	Počet cyklů mezi příkazem read a začátkem přenosu.
N_{WR}	2	---	Počet cyklů mezi příkazem write a začátkem přenosu dat

TAAC a NSAC jsou proměnné, které jsou obsaženy v CSD registru.

2. Řídící procesor

2.1 Volba řídícího procesoru

Při výběru řídícího procesoru jsme se rozhodovali mezi různými klony procesoru 8051. Tento typ procesoru jsme zvolili proto, že s nimi mám několikaleté zkušenosti. Při výběru procesoru jsme museli zvážit několik nutných podmínek. Karty na které chceme zapisovat data umožňují zápis v blocích o velikosti nejméně 512B, z toho důvodu je nutné aby volený procesor měl dostatek paměti RAM (XRAM). Při volbě procesoru také rozhodovala přítomnost rozhraní SPI. S přihlédnutím na tato kritéria jsme zvolili procesor firmy Atmel AT89C51ED2, který má vestavěné rozhraní SPI a také 1792B paměti XRAM.

Tento procesor je řídící jednotkou celého zařízení. Zajišťuje obousměrnou komunikaci s počítačem, zpřístupňuje obsah karty. Na připojeném displeji zobrazuje informace pro uživatele. Generuje řídící signály pro teplotní senzor, pro paměťovou kartu a pro obvod reálného času.

2.2 Popis zvoleného procesoru

2.2.1 Základní parametry procesoru

Procesor AT89C51ED2 firmy Atmel je plně kompatibilní s procesory x51 [5]. Oproti např. AT89C51 má navíc některé periferie.

Základní parametry procesoru:

- Vývojově i instrukčně kompatibilní s 8051
- Verze procesoru v 64 a 68 pinovém pouzdře má kromě standardních 4 I/O bran (P0 – P3) ještě brány P4 a P5
- 256B interní RAM (obdobně jako např. AT89C52)
- 64kB programové paměti flash
- 1792B datové paměti XRAM(přístupná přes instrukci MOVX)
- 2kB paměti EEPROM(může posloužit k uložení nastavení)

- Dva registry DPTR
- X2 - jádro pracuje s rychlostí 6 period hodin na strojový cyklus (místo standardních 12), pro zachování kompatibility je možné zapnout děličku 2 mezi vstupem XTAL1 a vlastním jádrem procesoru(z hlediska programu se pak procesor chová, jako by měl 12 period hodin na strojový cyklus). Po resetu je dělička zapojena.
- PCA (programovatelné čítačové pole) – umožňuje generování PWM, jednotky capture/compare
- Možnost generování signálu ALE pouze během instrukcí MOVX a MOVC (pro snížení rušení)
- Rozhranní SPI
- Rozhraní pro připojení klávesnice k bráně P1

2.2.2 Programování procesoru pomocí UART

Programování procesoru pomocí UART zajišťuje bootloader, který je umístěn ve zvláštní paměti. Aby bylo možné procesor naprogramovat přes UART, musí se však nejprve zajistit aby se po resetu aktivoval bootloader a ne případný uživatelský program umístěný v paměti.

Vlastní programování probíhá zasíláním programovacích dat ve formátu Intel HEX z PC do procesoru. Data se posílají včetně počátečního znaku ':' a koncových CR a LF. Po každém přijatém řádku v Intel HEX formátu a naprogramování v něm obsažených dat procesor do PC odesílá výsledek operace.(vše je OK, špatné CRC, apod.). Formát Intel HEX slouží i pro příkazy, které např. čtou obsah paměti flash, programují stavový byte BSB nebo čtou ID výrobce procesoru.

Programovací software FLIP

K nahrávání programu do procesoru používáme program FLIP (Flexible In-System Programmer), který je zdarma ke stažení na stránkách firmy Atmel. Program podporuje procesory T89C51RD2, AT89C51RD2/ED2 a další

procesory firem Temic a Atmel , které lze programovat pomocí rozhraní UART, USB nebo CAN.

2.2.3 Popis vytváření programu

Program pro použitý procesor byl vytvářen v jazyce ANSI C [7], v editoru TM Make SDCC. K překládání zdrojové textu byl použit kompilátor SDCC (Small Device C Compiler). Procesor byl naprogramován rozhraním UART pomocí programu FLIP firmy Atmel.

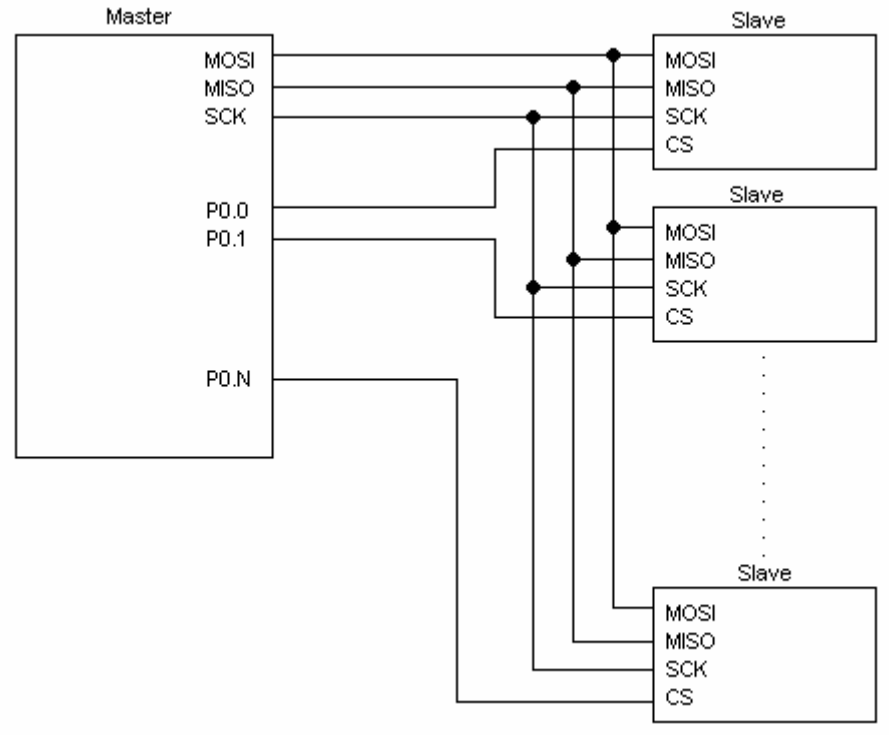
2.3 Popis komunikačního rozhraní mezi procesorem a ostatními obvody

V této podkapitole je čerpáno z publikace [6]. Rozhraní SPI je především určeno pro připojení vnějších pamětí, A/D převodníků a dalších obvodů k mikrokontroléru, případně pro vzájemnou komunikaci mezi mikrokontroléry. U některých klonů mikrokontrolérů je rozhraní SPI využíváno i pro programování jejich vnitřní paměti.

Na sběrnici mohou být zapojeny dva nebo více obvodů. Jeden z obvodů, obvykle řídící mikrokontrolér, je typu Master, ostatní jsou typu Slave. Komunikace probíhá pomocí 4 vodičů:

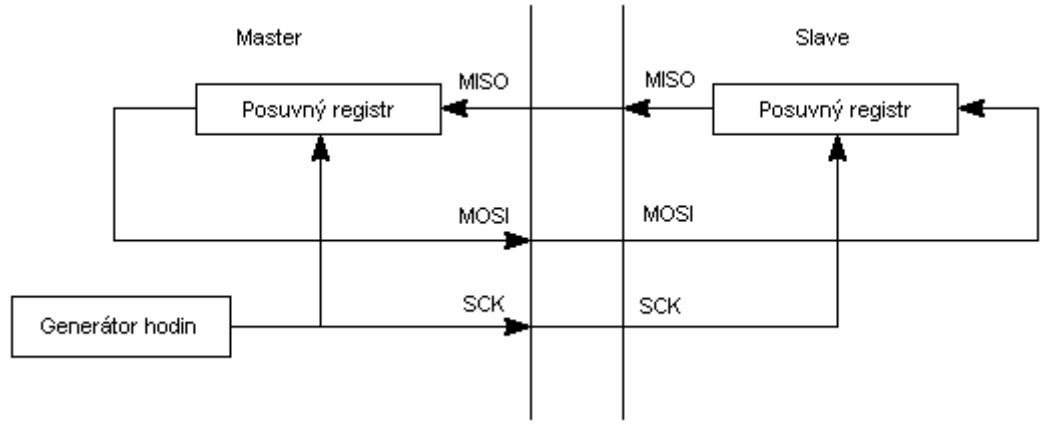
- Datový výstup MOSI(Master Out, Slave In) je připojen na vstupy MOSI všech obvodů Slave.
- Datový výstup MISO(Master In, Slave Out) obvodu Master je propojen s výstupy MISO všech obvodů Slave.
- Výstup hodinového signálu SCK je připojen na vstup SCK všech obvodů Slave.
- CS(Chip Select) je vývod pro výběr obvodu. Je-li nastaven v neaktivní úrovni, je příslušný Slave obvod odpojený od sběrnice, jeho vstupy jsou ve vysoko-impedančním stavu. Je-li obvodem Master mikrokontrolér,

bývají vstupy připojeny na některou z jeho bran. Tímto způsoben lze snadno vybírat obvod, s nímž má být vedena komunikace.



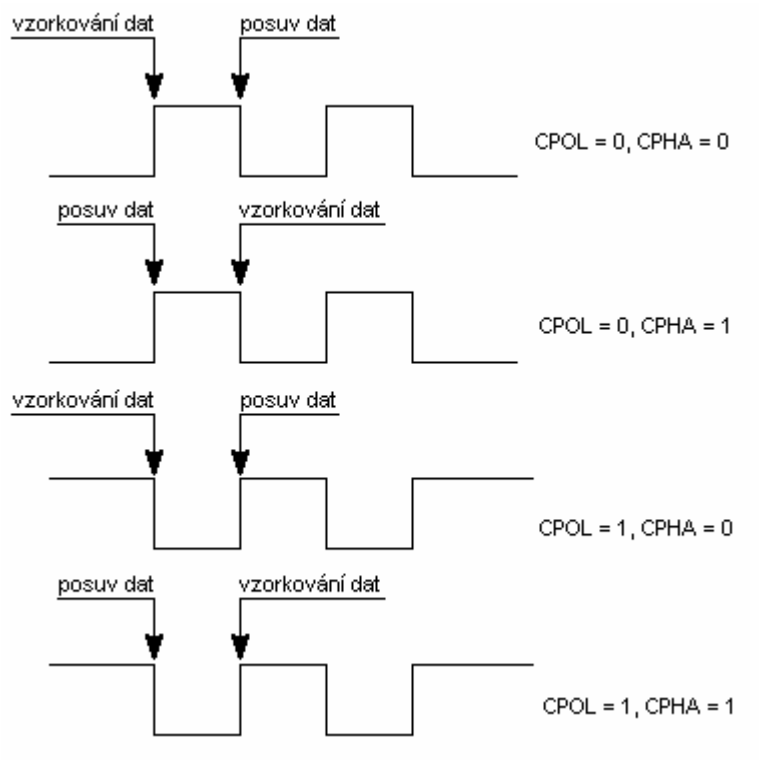
Obrázek 2.1 Koncepce systému se sběrnicí SPI

Přenosy po sběrnici SPI probíhají vždy mezi obvodem Master a obvodem Slave. Oba obvody obsahují posuvné registry které jsou propojeny tak, jak je schématicky naznačeno na Obr. 2.2



Obrázek 2.2 Propojení obvodů Master a Slave

Obvod Master generuje hodinový signál, který řídí posouvání obou posuvných registrů. Klidová úroveň signálu SCK a vztah mezi datovým a hodinovým signálem je dán parametry CPOL a CPHA (viz. Obr.2.3) . Pokud je rozhraní SPI realizováno specializovaným řadičem, je obvykle možné tyto parametry v řadiči nastavit. Je-li rozhraní SPI realizováno programově, musí být okamžiky změny úrovně datových a hodinových signálů zvoleny tak, aby přijímací obvod vzorkoval platná data.



Obrázek 2.3 Význam parametrů CPOL a CPHA

3. Popis vytvořené knihovny funkcí

3.1 Popis funkcí obsažených v knihovně

SD_cmd(char c1,char c2,char c3,char c4,char c5,char c6)

Tato funkce slouží k odeslání příkazu paměťové kartě. Jejím argumentem je 6 bytů příkazu. Funkce vrací hodnotu odpovědi ve formátu R1.

Příklad užití:

while (SD_cmd(0x40,0x00,0x00,0x00,0x00,0x95)!=0x01);

Init_SD()

Tato funkce zajistí správnou inicializaci karty. Přepnutí rozhraní na SPI, uvedení do stavu Ready. Funkce nemá žádné operandy. Po užití této funkce je karta připravena ke komunikaci. Funkce opět informuje o úspěšnosti provedení operace vrácením hodnoty 0x00. Funkci je nutné provádět pouze jednou po spuštění aplikace nebo po vložení paměťové karty.

Příklad užití:

if (PCON & 0x10){while(Init_SD()!=0);PCON &= 0xEF;}

SD_read_block(char c1,char c2,char c3,char c4,char c5,char c6,int bytes,char pole[])

Funkce SD_Read_block skouží ke čtení všech dat z paměťové karty. Proměnné c1 až c6 určují jaký příkaz bude nejprve odeslán, nebo-li jaká data od karty požadujeme. Proměnná bytes určuje kolik bytu se bude číst. Nakonec uvedeme cílové pole, které musí mít délku rovnu proměnné bytes + 1. Tato procedura je využita při čtení registrů CSD, CID i při čtení sektoru.

Příklad užití:

SD_read_block(0x51,c0,c1,c2,0x00,0xFF,511,pole);

SD_Read_CSD(char pole[])

Funkce SD_Read_CSD slouží ke čtení registru CSD. Jako proměnnou je nutné zadat pole o délce 16B. V tomto poli se nachází obsah registru CSD po provedení procedury.

Příklad užití:

SD_Read_CSD(pole); kde pole je typu unsigned char pole[16];

SD_Read_CID(char pole[])

Funkce SD_Read_CID pracuje stejně jako předchozí procedura. Jako proměnnou je nutné zadat pole o délce 16B. V tomto poli se nachází obsah registru CID po provedení procedury.

Příklad užití:

SD_Read_CID(pole); kde pole je typu unsigned char pole[16];

SD_Read_Sektor(long adr,char pole[512])

Pomocí této funkce čteme žádaný sektor. Vstupem do procedury je adresa požadovaného sektoru typu long a pole o délce 512B kam chceme uložit obsah sektoru.

Příklad užití:

SD_Read_Sektor(0,pole); kde pole je typu unsigned char pole[512];

SD_Write_Sektor(long adr,char pole[512])

Tato funkce zapisuje data do sektorů na paměťovou kartu. Opět nejprve uvedeme adresu sektoru do kterého budeme zapisovat. Další proměnnou je pole o délce 512B jehož obsah bude zapsán do určeného sektoru.

Příklad užití:

SD_Write_Sektor(0,pole);

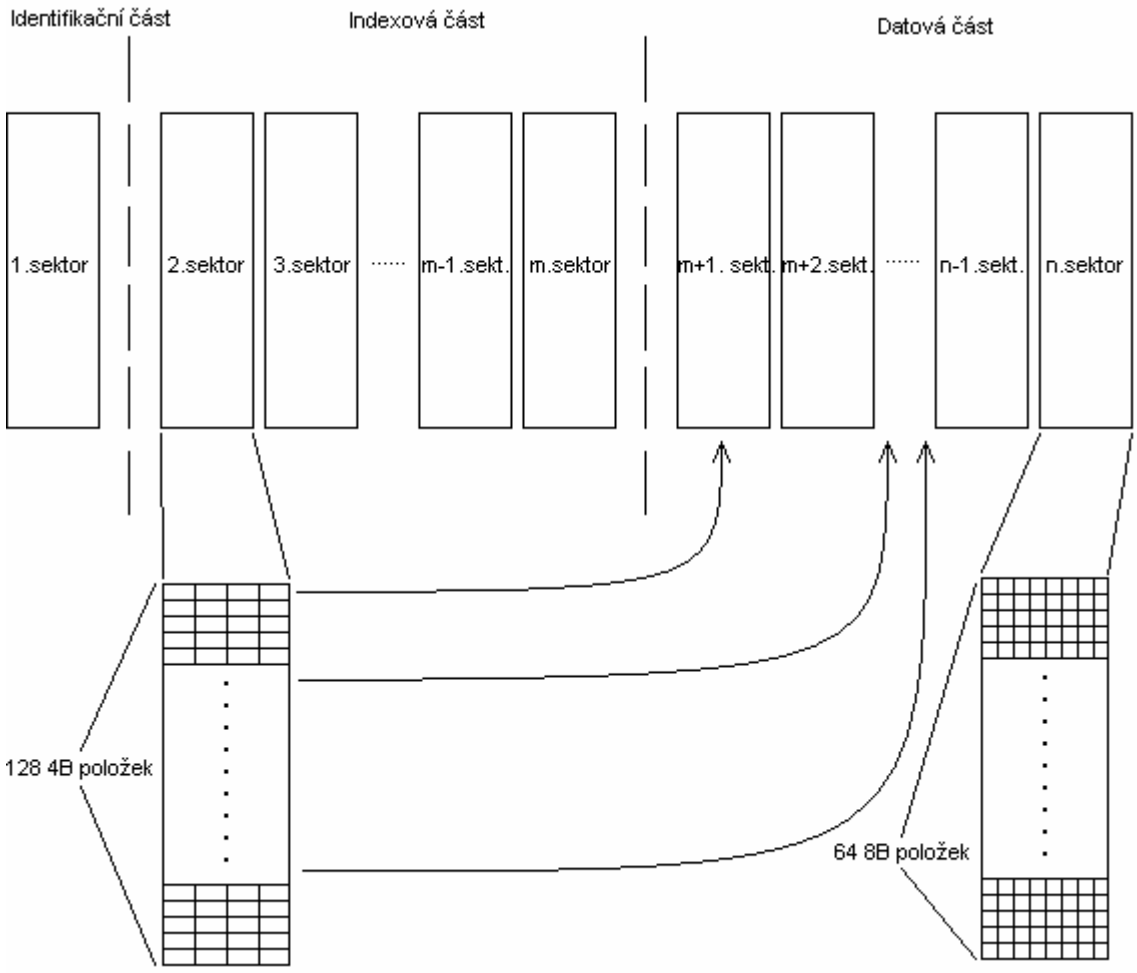
unsigned long SD_size()

Tato funkce zjistí velikost paměťové karty a vrátí ji v typu long. Funkce nemá žádný argument.

Příklad užití:

Velikost = SD_size();

3.2 Popis systému ukládání dat



Obrázek 3.1 Struktura dat na paměťové kartě

$$n = \frac{Velikost\_karty}{512} \tag{3.1}$$

$$m = \frac{n}{128} \tag{3.2}$$

Celá oblast paměti je rozdělena do tří částí. První (Identifikační) část zabírá pouze první sektor karty. Obsahuje informace o systému, intervalu ukládání dat a také ukazatele na poslední položky v indexové a datové části. Druhá (indexová část) obsahuje ukazatele, které ukazují na počáteční sektory jednotlivých bloků v datové oblasti. Blokem rozumíme množinu sektorů mezi jednotlivými starty ukládání, v našem případě mezi jednotlivými starty přípravku. Velikost indexové

části závisí na celkové velikosti paměťové karty. Třetí (Datová) oblast je tvořena sektory od indexové oblasti až po konec paměťové karty.

Po spuštění aplikace řídící procesor přečte první sektor karty. Podle prvního bytu sektoru zjistí jestli se jedná o známý systém nebo je na kartě systém jiný. Jestliže karta obsahuje neznámý systém, procesor zjistí velikost karty ze které určí podle vztahu 3.1 a 3.2 velikost indexové oblasti, poté nastaví výchozí hodnoty ukazatelů pro jednotlivé bloky. Zjistí-li procesor přítomnost známého systému, přečte hodnoty ukazatelů z prvního sektoru. Následně zahájí měření hodnoty teploty, které ukládá včetně času měření do 512B dlouhého pole. Po naplnění tohoto pole zapíše obsah pole do prvního volného sektoru v datové oblasti. Adresa prvního sektoru bloku dat je uložena do indexové části a také jsou aktualizovány hodnoty ukazatelů v prvním sektoru.

Po příchodu dotazu od ovládací aplikace, přejde procesor k datům v indexové oblasti. Ke každému indexu (adrese počátečního sektoru bloku dat) vyhledá hodnoty data/času první a poslední položky, které následně odešle ovládací aplikaci. Uživatel zvolí blok dat, které procesor odešle po přijetí hodnoty ukazatele v indexové oblasti.

4. Popis vzorové aplikace

Pro zjednodušení byl využit školní výukový modul, který obsahuje mikrokontrolér, displej, podpůrné obvody pro programování a obvody zdroje. V návrhovém systému EAGLE byl vytvořen modul který obsahuje pouzdro pro paměťovou kartu, teplotní senzor, obvod reálného času a záložní baterii. Modul je k mikroprocesoru připojen pomocí portu P1. Všechny obvody byly zvoleny takové, které pro komunikaci používají rozhraní SPI. To dovoluje všechny součástky připojit paralelně na jednu sběrnici. Tím snižujeme počet potřebných vývodů mikroprocesoru na minimum.

4.1 Jednotlivé součásti

Řídící procesor:

AT89C51ED2, stručný popis uveden výše.

Displej:

Použit standardní znakový displej 2 řádky po 16 znacích s řadičem Hitachi HD44780.

Teplotní senzor:

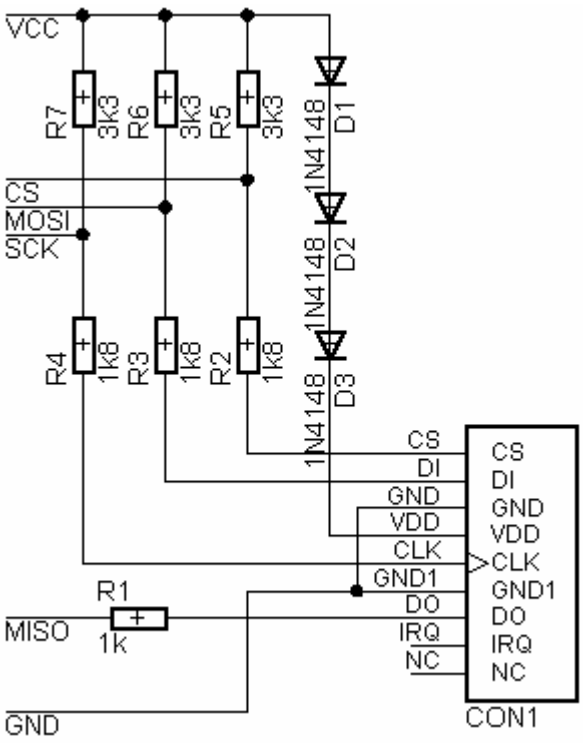
DS1722, měří teplotu od -55°C do 120°C s přesností ± 2°C, rozlišení 8 až 12 bitů. Napájení 5V. Komunikační rozhraní 3-wire nebo SPI [8].

Obvod reálného času:

DS1305, udržuje sekundy, minuty, hodiny, den v měsíci, měsíc, rok, kalibrován do roku 2100. Dále obsahuje 96 bytu paměti RAM zálohované baterií. Napájení 5V. Komunikační rozhraní 3-wire a SPI [9].

Paměťová karta:

Patice pro SD kartu. Zvolen typ, který podporuje kartu SD i MMC. Obě paměťové karty (SD i MMC) používají k napájení napětí 2,7V až 3,6V. Toto napětí získáváme zařazením tří standardních usměrňovacích diod mezi napájení modulu(5V) a napájecí vývod paměťových karet. Měřením bylo potvrzeno že se napájecí napětí karty pohybuje okolo 3,3V. Dále je také nutné snížit napětí signálových vodičů. To je zajištěno připojením vstupů karty přes odporové děliče.



Obrázek 4.1 Úprava napájení a napěťových úrovní signálů

4.2 Funkce software v mikroprocesoru

Po spuštění aplikace provede procesor nastavení všech obvodů včetně paměťové karty. Následně zjistí pozice ukazatelů pro kartu. Pak vstoupí do nekonečné smyčky ve které neustále obnovuje informace na displeji. Také kontroluje zda uplynul čas pro další měření nebo byl přijat požadavek o data od ovládací aplikace. Po zjištění uplynutí času provede měření a hodnoty uloží do paměti. Po přijetí požadavku o data, procesor nejprve odešle aplikaci seznam bloků nacházejících se na kartě. Teprve po výběru jednoho z bloků procesor odešle platná data.

4.3 Ovládací software

Ovládací program byl vytvořen ve vývojovém prostředí Delphi. Ke komunikaci po sériové lince bylo nutno doinstalovat knihovnu CPORT. Pomocí tohoto ovládacího programu lze nastavovat velikost intervalu mezi jednotlivým snímáním hodnot, nastavit správný datum a čas. Dále stahovat údaje z paměťové karty, mazat kartu. Naměřené hodnoty lze zobrazovat v grafu a ukládat do souboru.

4.4 Možné použití aplikace

Přípravek slouží pro ukládání (logování) různých hodnot. Je možné k němu připojit jakýkoli senzor (teploty, tlaku, vlhkosti,…), či převodník k měření napětí, proudu, příkonu. Jedná se pouze o ukázkovou aplikaci. V praxi by bylo nutné umístit měřící prvek mimo modul, jinak by mohlo docházet k problémům díky omezení pracovních podmínek ostatních součástek.

5. Závěr

Bylo ukázáno že SD a MMC karta jsou téměř identické. Liší se pouze v síle pouzdra. Lze tedy SD kartu nahradit MMC kartou. Opačně to však nahradit nelze. Pro práci s těmito kartami byla vytvořena knihovna funkcí v jazyce ANSI C. Funkčnost knihovny ověřuje ukázková aplikace, jejímž účelem je čtení teploty z čidla a ukládání dat na paměťovou kartu. Rychlost přenosu dat v této aplikaci však není vysoká, pohybuje se v řádech desítek kilobytů za sekundu. Navržený systém ukládání dat je velice jednoduchý. Čtení a zápis dat na kartu je možno pouze po sektorech. Prohledávání celé karty by bylo vzhledem k nízké přenosové rychlosti časově velmi náročné.

K nepatrnému zlepšení by určitě pomohlo použití krystalu s vyšší rezonanční frekvencí. Dále by bylo vhodnější implementovat nějaký ze známých souborových systému např. FAT16. Nynější způsob ukládání dat vylučuje přítomnost jiných než naměřených dat na kartě.

Použitá Literatura

[1] SD specifications Part1 PHYSICAL LAYER Simplified Specification, SD card Association, 2006, [online] [cit. 8.9.2006] Dostupné na www <http://www.sdcard.org/sd_memorycard/Simplified%20Physical%20Layer%20Specification.PDF>

[2] SanDisk MultiMediaCard and Reduced-Size MultiMediaCard, SanDisk Corporation, 2006, [online] [cit. 8.9.2006] Dostupné na www <http://www.sandisk.com/Assets/File/OEM/Manuals/ProdManRS-MMCv1.3.pdf>

[3] Comparison of memory cards, Wikipedia, 2005, [online] [cit. 5.9.2006] Dostupné na www <http://en.wikipedia.org/wiki/Comparison_of_memory_cards>

[4] Skalický P.: Mikroprocesory řady 8051, BEN 2002, ISBN 80-86056-39-2

[5] AT89C51RD2/ED2, Atmel Corporation, 2005, [online] [cit. 8.9.2006] Dostupné na www <http://www.atmel.com/dyn/resources/prod_documents/doc4235.pdf>

[6] Dudáček K.:Sériová rozhraní SPI, Microwire, I2C a CAN, 2002, [online] [cit. 8.9.2006] Dostupné na www <http://home.zcu.cz/~dudacek/NMS/Seriova_rozhrani.pdf>

[7] Mann B.: C pro mikrokontroléry, BEN 2003, ISBN 80-7300-077-6

[8] DS1722 Digital Thermometrer with SPI/3-wire interface, Dallas Semiconductor, 2005, [online] [cit. 8.9.2006] Dostupné na www <http://datasheets.maxim-ic.com/en/ds/DS1722.pdf>

[9] DS1305 Serial Alarm Real-Time Clock, Dallas Semiconductor, 2005, [online] [cit. 8.9.2006] Dostupné na www <http://datasheets.maxim-ic.com/en/ds/DS1305.pdf>