

České vysoké učení technické v Praze

Fakulta elektrotechnická



Bakalářská práce

Zařízení pro ukládání dat na SD kartu podporující souborový systém FAT32

Miroslav Roubal

Vedoucí práce: Ing. Pavel Kubalík, Ph.D.

Studijní program: Elektrotechnika a informatika, strukturovaný bakalářský

Obor: Výpočetní technika

Červen 2008

Poděkování

Tímto bych rád poděkoval vedoucímu práce panu Ing. Pavlu Kubalíkovi, Ph.D., za jeho čas a cenné rady. Dále pak děkuji všem, kteří mi pomohli ke zdárnému dokončení práce.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

Ve Valdově dne 30.6.2008

.....

Abstract

Design and implement a hardware device for data storage on SD cards. Measured values derived from RS232 serial bus will be stored in individual files, according to the type of measured data. The device will support the FAT32 file system. The device will be realized with the PIC18F4550 microcontroller.

Abstrakt

Navrhněte zařízení pro ukládání dat na SD kartu. Zařízení bude podporovat souborový systém FAT32. Naměřené hodnoty získané ze sériové sběrnice RS232 budou ukládány do jednotlivých souborů a to podle typu měřených dat. Zařízení bude realizováno s pomocí mikrořadiče PIC18F4550.

Obsah

Seznam obrázků.....	xii
Seznam tabulek.....	xiv
1 Úvod	1
2 Rešerše	2
2.1 Konečné rozhodnutí.....	3
3 Analýza	4
3.1 Média k ukládání dat	4
3.1.1 Compact Flash	4
3.1.2 Memory Stick	4
3.1.3 xD-Picture.....	4
3.1.4 MultiMedia	5
3.1.5 Secure Digital	5
3.1.6 Srovnání paměťových karet.....	6
3.2 Mikroprocesor	6
3.2.1 PIC18F4550.....	7
3.3 SPI	9
3.4 SD karta	10
3.4.1 Kompatibilita SD a SDHC	10
3.4.2 Chránění dat.....	11
3.4.3 Struktura SD karty	11
3.4.4 Komunikace s SD kartou.....	13
3.5 Souborový systém.....	19
3.5.1 FAT32.....	21
3.5.2 FAT32 & FAT16	24
3.6 RS232	24
3.7 I ² C.....	25
3.7.1 Adresování.....	25
4 Návrh řešení.....	27
4.1 Blokový Návrh	27
4.2 Návrh schématu a plošného spoje	28
5 Řešení.....	29
5.1 Hardware	29

5.1.1	Popis desky	29
5.1.2	Napájení	29
5.1.3	Propojení SD karty s CPU	30
5.1.4	Propojení RTC TIME s CPU přes I2C BUS	30
5.1.5	Propojení externího měření teploty	31
5.2	Firmware	31
5.2.1	Popis protokolu RS232	31
5.2.2	Implementace SW SPI	37
5.2.3	Implementace SW I ² C	39
5.2.4	Popis funkce zařízení	39
5.3	Testování	40
6	Testovací aplikace	41
7	Problémy při řešení	43
8	Závěr	44
9	Literatura	45
A	Schémata	47
B	Plošné spoje	48
C	Rozpiska součástek	50
D	Obsah CD	54

Seznam obrázků

Obrázek 2.1 Vývojová deska MMSD3.....	2
Obrázek 2.2 Vývojový kit stk1000.....	2
Obrázek 2.3 Čtečka paměťových karet	3
Obrázek 3.1 Pouzdro Compact Flash karty	4
Obrázek 3.2 Pouzdra SD karet	5
Obrázek 3.3 Vnitřní struktura PIC18F4550	8
Obrázek 3.4 Komunikace SPI – Master & Slave	10
Obrázek 3.5 Kompatibilita jednotlivých verzí	10
Obrázek 3.6 Logo SDHC.....	11
Obrázek 3.7 Ochrana dat na SD kartě	11
Obrázek 3.8 Pohled do vnitřní struktury SD karty	12
Obrázek 3.9 Command blok.....	15
Obrázek 3.10 Response blok	15
Obrázek 3.11 Diagram inicializace SPI módu.....	16
Obrázek 3.12 Inicializační tok.....	17
Obrázek 3.13 Čtení dat	18
Obrázek 3.14 Čtení dat – data error.....	18
Obrázek 3.15 Zápis dat.....	18
Obrázek 3.16 Struktura disku	19
Obrázek 3.17 Struktura FAT32	21
Obrázek 3.18 Struktura FAT tabulky	23
Obrázek 3.19 Základní průběhy na I ² C	25
Obrázek 3.20 Adresování I ² C	26
Obrázek 4.1 Blokové schéma zařízení	27
Obrázek 5.1 Napájení z 5V na 3,3V pro SD kartu	30
Obrázek 5.2 Propojení SD karty s CPU	30
Obrázek 6.1 GUI testovací aplikace	42

Seznam tabulek

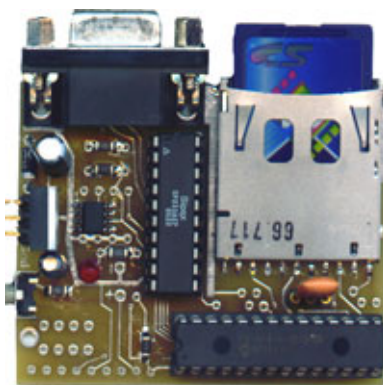
Tabulka 3.1 Rozměry SD karet	5
Tabulka 3.2 srovnání paměťových karet	6
Tabulka 3.3 Popis signálu SPI	9
Tabulka 3.4 Registry SD karty	13
Tabulka 3.5 Význam pinů v jednotlivých komunikačních módech	14
Tabulka 3.6 Partition tabulka v MBR.....	20
Tabulka 3.7 Některé položky z boot sektoru	22
Tabulka 3.8 Některé položky z boot sektoru specifické pro FAT32	22
Tabulka 3.9 Význam jednotlivých hodnot ve FAT tabulce.....	23
Tabulka 3.10 Obsah jednoho 32 bytového záznamu v adresářové struktuře	24
Tabulka 5.1 Ukládané informace do souboru.....	40

1 Úvod

V dnešní době při průmyslové výrobě je kladen důraz na monitorování procesů a záznam dat pro dokladování do systému řízení jakosti ISO 9001, tj. dodržení kvality a opakovatelnosti výroby. Mým úkolem je navrhnout zařízení, které bude ukládat získaná data do paměťového média. Tato paměť by měla být vyjímatelná. Čtení dat z média bude buď ze samotné paměti, která se připojí k PC, nebo po připojení PC k tomuto zařízení.

2 Rešerše

Vývojová deska MMSD3 (Obrázek 2.1) je osazena procesorem od firmy Microchip. Konkrétně se jedná o 8-bitový procesor typu PIC16F876. Dále obsahuje slot na MMC/SD karty. Komunikace je zajištěna přes RS232. Cena tohoto zařízení se pohybuje okolo 1000 Kč. Výhody: nízká pořizovací cena, velikost zařízení. Nevýhody: Malá kapacita paměti dat v CPU (jen 368 bytů).



Obrázek 2.1 Vývojová deska MMSD3

Vývojový kit stk1000 (Obrázek 2.2) je osazený procesorem AVR32. Tato vývojová deska má mnoho periférií jako například LCD displej QVGA (320x240), 2x RJ45, USB v2.0, SD/MMC slot a další. Dodává se také s operačním systémem Linux, který je uložen na SD kartě. Je to špičkové zařízení, a tudíž je jeho cena stanovena na 8500 Kč. Výhody: Mnoho periférií, LCD. Nevýhody: vyšší pořizovací cena.



Obrázek 2.2 Vývojový kit stk1000

Dalším typem zařízení jsou čtečky paměťových karet, které umožňují číst více typů karet. Jejich cena se pohybuje v rozmezí 200 Kč až 500 Kč. Čtečky paměťových karet neumožňují ukládání dat prostřednictvím RS232, ani jako například z čidla teploty. Jejich komunikace probíhá prostřednictvím USB. Nevýhody: komunikace pouze s PC přes USB.



Obrázek 2.3 Čtečka paměťových karet

2.1 Konečné rozhodnutí

Na našem trhu můžeme najít spoustu zařízení, která ukládají data do paměťových karet. Některé vývojové kity větších rozměrů jsou pro tuto aplikaci příliš složité, a tudíž také dražší. Naopak vývojový kit na Obrázek 2.1 je na implementaci souborového systému FAT32 s tímto procesorem PIC nevhodný z důvodů příliš malého bufferu. Na uložení čteného či zapisovaného sektoru je potřeba 512B a dále musíme mít k dispozici místo pro režii samotné aplikace a souborového systému. Proto jsem navrhl vlastní zařízení. Hlavním účelem tohoto zařízení by mělo být ověření některých funkcí pro pozdější využití ve větším zařízení, které bude sestaveno z podobně rozsáhlých a ověřených celků.

3 Analýza

3.1 Média k ukládání dat

Nyní potřebuji vybrat paměťové médium, na které budu ukládat data. Nejlépe takové, které po vyjmutí ze zařízení přečte i případná další existující zařízení, jako například počítače nebo čtečky paměťových karet. Nabízí se tedy USB klíčenka nebo karta do slotu. Mikroprocesor, který byl vybrán, sice nabízí částečnou hardwarovou podporu USB, ale implementace USB protokolu je natolik složitá, že by mohla být dalším pokračováním mé bakalářské práce. Z výše uvedených důvodů jsem se rozhodl pro jiné řešení než je připojení USB klíčenky. Vraťme se tedy k paměťovým kartám a na některé se podíváme blíže.

3.1.1 Compact Flash

Compact Flash používá jako záznamové médium typu flash. Její rozhraní je postaveno jako klasické IDE. Rychlosti CF karet byly postupně rozšiřovány o další přenosové režimy ATA. Karta byla naposledy upravena na verzi CF 4.0 podporující režim Ultra-DMA 5 a to v roce 2007. CF karty mohou dosahovat až 64GB, což je nejvíce z níže uvedených karet.



Obrázek 3.1 Pouzdro Compact Flash karty

3.1.2 Memory Stick

Je paměťová karta, jejíž kapacita dosahuje až 8GB. Vyrábí se v několika velikostních variantách.

3.1.3 xD-Picture

Paměťová karta je typu NAND flash memory, používá se hlavně ve fotoaparátech a dosahuje kapacity až 2GB při rozměrech 20 mm × 25 mm × 1,78 mm.

3.1.4 MultiMedia

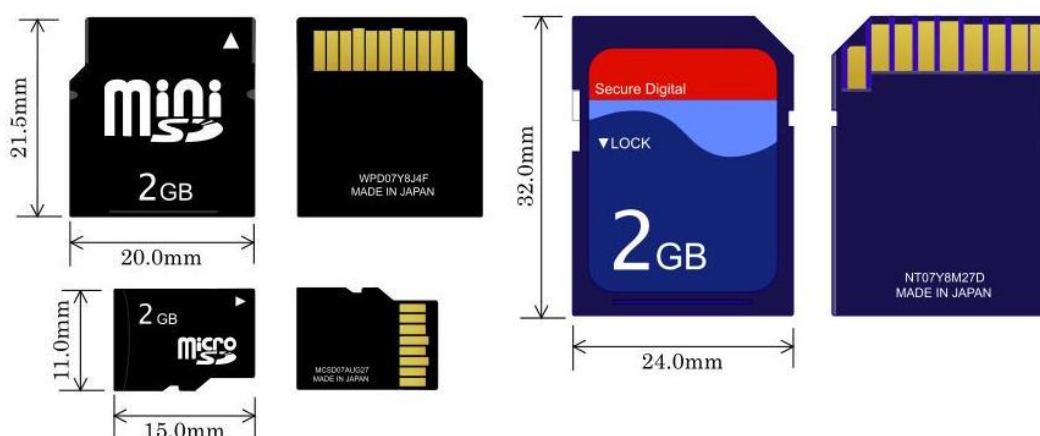
Je zde opět použita paměť typu NAND flash. MMC používá 1 bitové sériové rozhraní, nejnovější typy využívají až 8 bitové rozhraní a dosahují kapacity až 8GB. Podobá se velmi SD kartě a může být použita v mnoha zařízeních, která podporují SD karty.

3.1.5 Secure Digital

Jako ukládací médium je zde použita flash. Mají vestavěnou technologii zabezpečení dat. SD karta se používá v mnoha přenosných zařízeních a je velmi oblíbená. Vyrábí se ve třech velikostních variantách jako standart, mini a micro. Dosahuje kapacity až 2GB. Nyní nastupuje nová technologie SD karet označovaná jako SDHC nebo také SD 2.0. Výhodou této technologie je dosažení vyšší kapacity a to až k hranici 32GB. SDHC nejsou zpětně kompatibilní. Čtečky, které podporují novější formát, tak umějí číst a zapisovat i na starší typy SD karet.

	SD karta	miniSD karta	microSD karta
Šířka	24 mm	20 mm	11 mm
Výška	32 mm	21,5 mm	15 mm
Váha	Přibližně 2 g	Přibližně 1 g	Přibližně 0,5 g
Ochrana proti zápisu	Ano	Ne	Ne
Provozní napětí	2.7 ÷ 3.6 V	2.7 ÷ 3.6 V	2.7 ÷ 3.6 V
Počet pinů	9	11	8

Tabulka 3.1 Rozměry SD karet



Obrázek 3.2 Pouzdra SD karet

3.1.6 Srovnání paměťových karet

Všechny karty, které zde uvádím, nemají žádné mechanické součásti a uchovávají data i bez příkonu elektrického proudu. Jelikož SD a MMC karty jsou si velmi podobné a v některých případech i kompatibilní, podíváme se na některé jejich rozdíly:

- cena: SD karty bývají levnější
- rychlost: SD bývají 3- 4x rychlejší
- kapacita: SD karty dosahují větší kapacity
- rozměry: MMC je tenčí a lehčí

Typ karty	Zkratka	Cena karty za 2GB	# pinů	Max rychlost, MBytes/s	
				Čtení	Zápis
Secure Digital	SD	230 Kč	9	20	20
Compact Flash	CF	269 Kč	50	40	40
Multi Media	MMC	400 Kč	7	20	20
Memory Stick	-	650 Kč	10	20	20
xD-Picture	xD	700 Kč	18	15	9

Tabulka 3.2 srovnání paměťových karet

Z Tabulka 3.2 je vidět, že maximální hodnota zápisu nebo čtení je stejná až na xD kartu a nejnovější modely CF karet, které dosahují až o 20MB/s větší datový přenos. Ve skutečnosti velice záleží také na výrobci, jak si poradí s danou implementací. Rychlostní rozdíl bývá tedy i ve stejném druhu karet. Uvedeme například rychlost zápisu u Compact Flashky a porovnáme dva typy od jedné značky Ultra II a Extreme III. U první jmenované je zápis 9MB/s a čtení 10MB/s, u druhé karty Extreme III je zápis a čtení shodné 20MB/s. Vzhledem k počtu 50 pinů CF karty jsem ji nemohl použít, jelikož mikroprocesor má v pouzdře TQFP jen 44 pinů a to ještě některé piny musíme využít k napájení a k připojení hodinové frekvence. Kartu xD jsem nemohl zvolit, neboť firma, která stojí za vývojem, nemá dokumentaci veřejně přístupnou. Vzhledem k výše uvedeným skutečnostem jsem zvolil SD kartu.

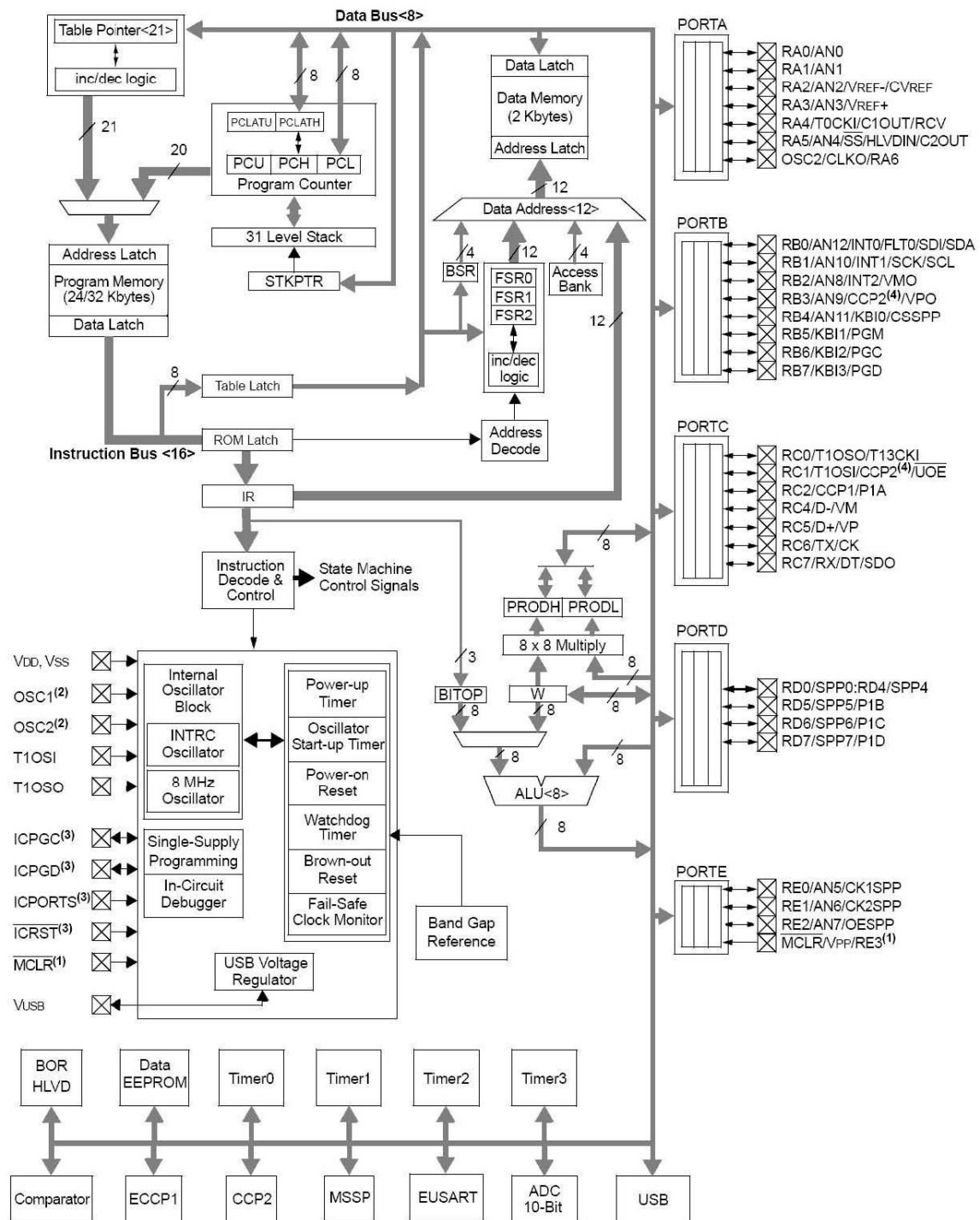
3.2 Mikroprocesor

Mikroprocesor nebo také procesor či mikrokontrolér je dnes mozkiem většiny zařízení od nejmenších až po ty největší jako například mp3 přehrávače, mobily, PDA zařízení,

fotoaparáty, navigační systémy, tiskárny, herní konzole, DVD přehrávače, počítače, automobily atd. V dnešní době je trend elektronická zařízení řídit, monitorovat a ukládat různá data. Někdy je potřeba i některou z těchto operací provádět vzdáleně. Segment s mikroprocesory jde velmi rychle kupředu, proto se mnoho firem snaží uspět v tomto odvětví. V oblasti jednočipových mikroprocesorů jsou u nás především známé dvě firmy: Microchip se svými procesory PIC a Atmel s řadou AVR. Já jsem se také rozhodl mezi těmito dvěma procesory. Ceny CPU od těchto dvou firem jsou srovnatelné i hardwarové vybavení je na stejné úrovni. S procesorem AVR jsem se setkal během studia. S procesorem PIC jsem pracoval při měření času. Jelikož jsem měl přístup k programátoru MPLAB REAL ICE In-Circuit Emulator, který je určen k ladění a debugování v reálném čase, rozhodl jsem se pro procesory od firmy Microchip. Tato firma vyrábí procesory počínaje 8-bitovými a konče 32-bitovými. Pro potřeby tohoto zařízení jsem zvolil 8bitový procesor PIC18F4550, který má mimo jiné i podporu USB a tím i široké využití v budoucnosti.

3.2.1 PIC18F4550

8 bitový procesor harvardské architektury s 10 bitovým A/D převodníkem a USB 2.0 rozhraním. Nízký odběr (aktivní mód 90 μ A typicky) s napájecím napětím 4,2V až 5,5V. Jeho maximální frekvence je 48 MHz. Na Obrázek 3.3 můžeme vidět vnitřní strukturu použitého procesoru. Povšimněme si zejména 31 úrovní zásobníku (stack), kde se ukládá návratová adresa a to při volání CALL, RCALL. Procesor má dvoustupňovou pipeline rozdělenou na FETCH a EXECUTE.



Obrázek 3.3 Vnitřní struktura PIC18F4550

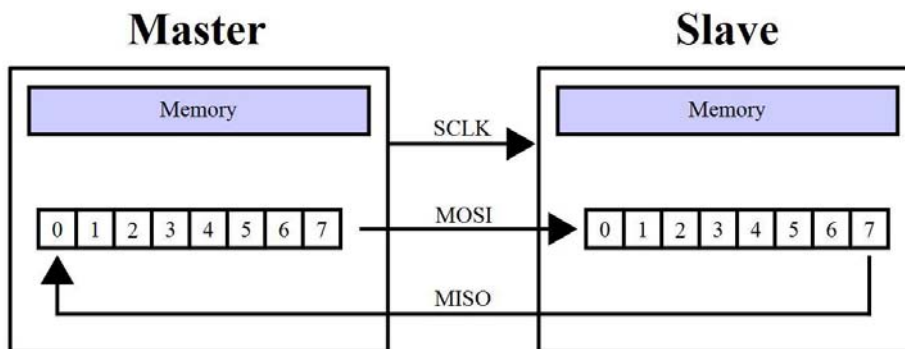
3.3 SPI

Zkratka SPI znamená Serial Peripheral Interface. Jak už anglický název napovídá, půjde o sériovou komunikaci. Používá se pro synchronní sériovou komunikaci mezi mikroprocesory a perifériemi. Výběr zařízení je prováděn pomocí signálu CS (chip select – výběr čipu), který aktivuje master neboli řídicí obvod celé sběrnice na logickou nulu. Na sběrnici jsou dva typy zařízení, master a slave. Master řídí komunikaci pomocí hodinového signálu většinou označovaného jako SCLK. Master také určuje, se kterým zařízením bude komunikovat pomocí signálu. Tento signál se označuje jako CS nebo SS-Slave Select. Zařízení typu slave může být na sběrnici více, jestliže zařízení má signál CS v logické nule, pak vysílá podle hodinového signálu. Tabulka 3.3 přehledně ukazuje popis signálů, které můžeme najít na sériové synchronní sběrnici SPI.

Název signálu	Popis signálu
SCLK, SCK	Hodiny
SDI, DI, SI	Sériová vstupní data
SDO, DO, SO	Sériová výstupní data
MOSI	Master Output Slave Input (Master výstup, Slave vstup)
MISO	Master Input Slave Output
CS, SS	Slave Select; aktivní LOW; výstup z mastru

Tabulka 3.3 Popis signálu SPI

Dále se podíváme na datovou komunikaci, která může probíhat mezi jedním zařízením typu master a jedním nebo více zařízeními typu slave. Jestliže máme pouze jedno zařízení slave, CS můžeme natvrdo připojit na logickou nulu. Zařízení slave má implementován jednoduchý posuvný registr, nejčastěji bývá 8-bitový, ale může být i jeho násobkem. Data jsou pumpována na vstup posuvného registru v jednotlivých taktech, který určuje master pomocí signálu hodin SCLK. Během každého signálu hodin SCLK odešle master na vodič MOSI jeden bit a slave přečte ze stejného vodiče také jeden bit, který nasune do posuvného registru. Současně s tím slave odešle jeden bit na vodič MISO a master přijme tento jeden bit na stejném vodiči. Tuto komunikaci ilustruje Obrázek 3.4.

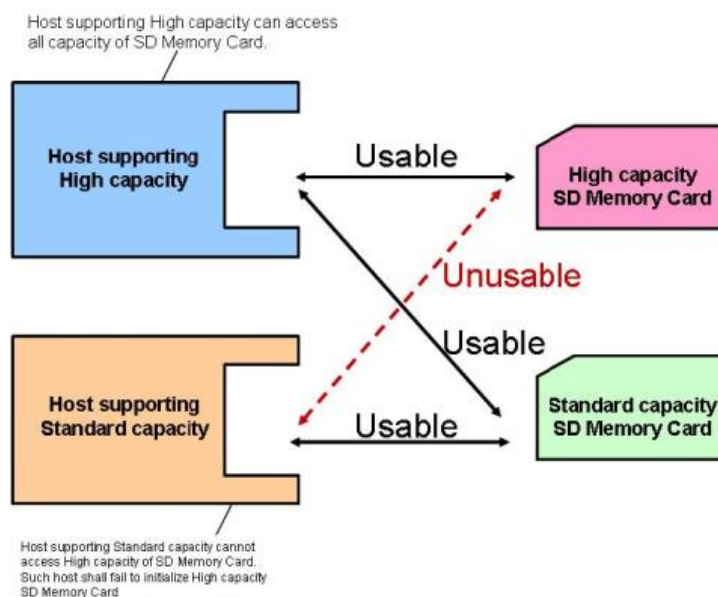


Obrázek 3.4 Komunikace SPI – Master & Slave

3.4 SD karta

3.4.1 Kompatibilita SD a SDHC

Nejprve se podíváme na kompatibilitu starého a nového formátu. Na prvním Obrázek 3.5 můžeme vidět, který SD host spolupracuje se kterými verzemi. Na dalším Obrázek 3.6 vidíme, jak poznáme správné označení a logo SDHC.



Obrázek 3.5 Kompatibilita jednotlivých verzí

CAUTION

About SDHC Memory Card

- Look for the SDHC logo when purchasing SDHC Memory Cards.
- The SDHC Logo is used for cards 4GB or larger.
- SD Speed Class is mandatory for SDHC Memory Card. It also appears on the card.

[more info](#) |

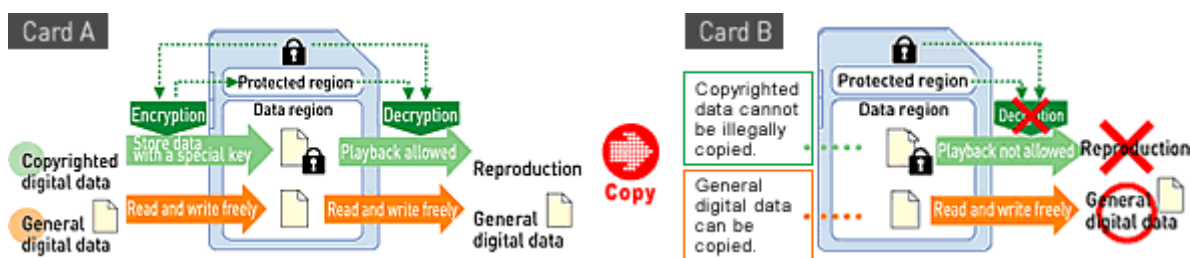
Correct	Wrong
	CLASS 2 CLASS 4 CLASS 6 SD Speed Class Logo

* SD Logo is a trademark.
* SDHC Logo is a trademark.

Obrázek 3.6 Logo SDHC

3.4.2 Chránění dat

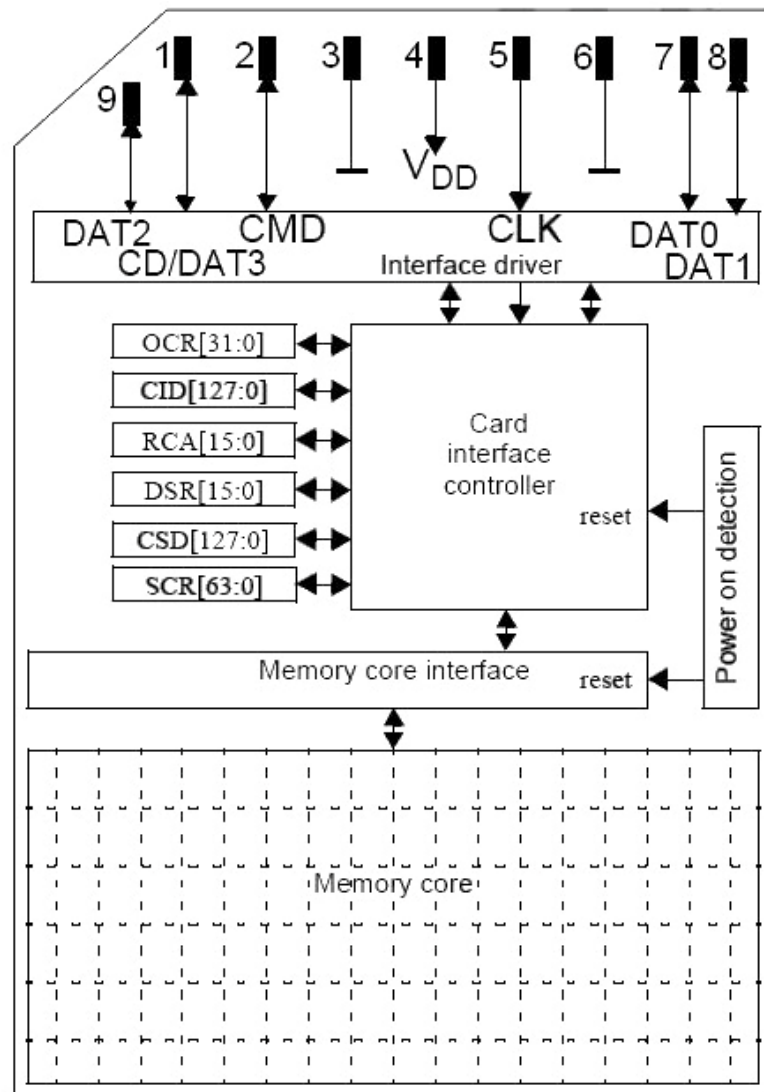
Data na kartě se nechají zamknout mechanicky, tj. přepínačem na kartě nebo heslem, které je uloženo na kartě v registru CMD42 - LOCK_UNLOCK. Této možnosti se využívá například při výrobě tím, že nahrávací společnost nahraje na kartu hudbu v mp3 a tu pak uzamkne a při kopírování se zabrání plagiátorství. Tento proces ilustruje Obrázek 3.7. Tím je karta vybavena nejvyšší ochranou ze standartu SDMI (Secure Digital Music Initiative).



Obrázek 3.7 Ochrana dat na SD kartě

3.4.3 Struktura SD karty

Na Obrázek 3.8 se můžeme podívat detailněji do hardware struktury SD karty. Obrázek znázorňuje vnitřní strukturu uspořádání ovládání a přístupu do paměťového prostoru SD karty.



Obrázek 3.8 Pohled do vnitřní struktury SD karty

Registry OCR, CID, CSD, SCR nesou specifické informace o kartě. Registry RCA, DSR jsou konfigurační registry, ve kterých jsou uloženy aktuální konfigurační parametry.

Název registru	Délka	Popis
CID	128	Card I Dentification registr – obsahuje identifikační informace: jméno výrobku, identifikace výrobce, datum výroby atd.
RCA	16	R elative Card Address – nastavení adresy karty, ale není přístupný v SPI módu.
DSR	16	D rive Stage Registr -
CSD	128	Card Specific Data - poskytuje informace týkající se

		přístupu do obsahu karty. Definuje: formát dat, maximální přístupový čas k datům atd.
SCR	64	SD Configuration Register – registr je nastavený při výrobě. Má tyto pole: SCR_STRUCTURE, SD_SPEC, DATA_STAT_AFTER_ERASE, SD_SECURITY, SD_BUS_WIDTHS
OCR	32	Operation Conditional Register – ukládá VDD napětí karty. 30 bit – Card Capacity Status: 1 High Capacity SD Memory Card 0 Standard Capacity SD Memory Card
SSR	512	SD Status Register
CSR	32	Card Status – informace o stavu karty

Tabulka 3.4 Registry SD karty

3.4.4 Komunikace s SD kartou

S SD kartou můžeme komunikovat pomocí tří komunikačních protokolů:

- Jednabitový SD mód
- Čtyřbitový SD mód
- SPI mód

Pin	SD 4-bitový mód		SD 1-bitový mód		SPI mód	
1	CD/DAT[3]	Datová linka 3	N/C	Není použit	CS	Vybrání karty
2	CMD	Command line	CMD	Command line	DI	Data vstup
3	VSS1	GND	VSS1	GND	VSS1	GND
4	VDD	Napájení	VDD	Napájení	VDD	Napájení
5	CLK	Hodiny	CLK	Hodiny	CLK	Hodiny
6	VSS2	GND	VSS2	GND	VSS2	GND
7	DAT[0]	Datová linka 0	DATA	Datová linka	DO	Data výstup
8	DAT[1]	Datová linka 1 nebo přerušení (volitelné)	IRQ	Přerušení	IRQ	Přerušení
9	DAT[2]	Datová linka 2 nebo čekání na čtení(volitelné)	RW	Čekání na čtení (volitelné)	NC	Nepoužito

Tabulka 3.5 Význam pinů v jednotlivých komunikačních módech

Tabulka 3.5 popisuje význam jednotlivých pinů ve všech třech komunikačních módech. Očíslování pinů je na Obrázek 3.8. Všechny paměťové karty musí podporovat tyto tři protokoly, kromě microSD, kde je SPI mód volitelný. V tomto zařízení je použita pro komunikaci mezi mikroprocesorem a kartou sběrnice SPI, která je popsána v kapitole 3.3.

3.4.4.1 SPI sběrniceový protokol

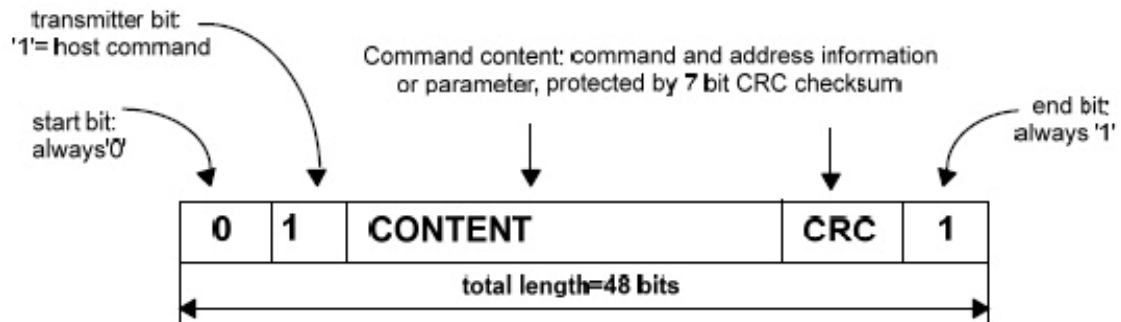
Základem komunikace jsou odesílané příkazy a přijímaná (vysílaná) data (datový bitový proud). Komunikace je inicializována start bitem a ukončena stop bitem.

- Command: příkaz je sada znaků, která startuje operaci. Příkaz je poslán z hosta jedné kartě jako adresní příkaz nebo všem kartám jako broadcast příkaz.
- Response: odpověď je sada znaků, které jsou odeslány z adresované karty nebo (synchronně) ze všech připojených karet do hosta jako odpověď na předtím přijatý příkaz.
- Data: data mohou být přenášena z hosta do karty nebo obráceně.

Na strukturu příkazu, na odpovědi a na data se podíváme v kapitole 3.4.4.2.

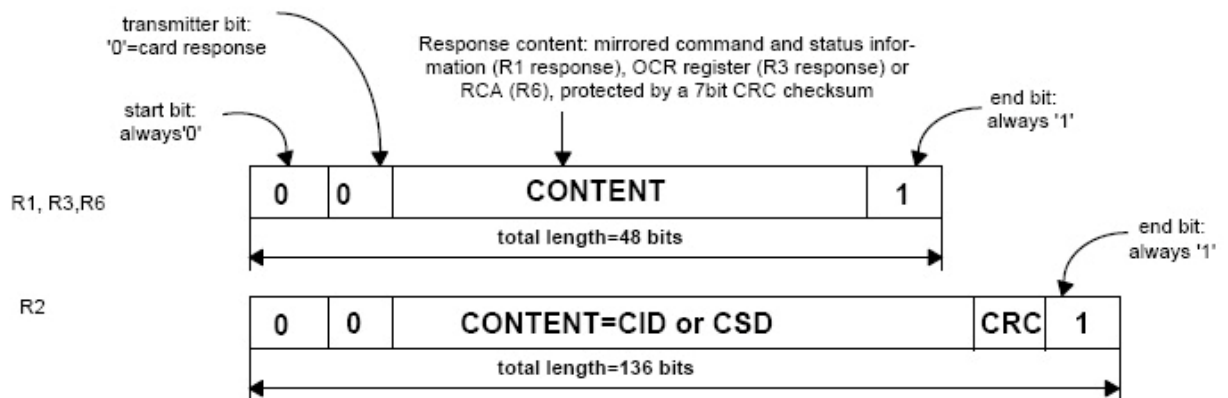
3.4.4.2 Struktura: Command , Response, Data

Každý příkazový (command - CMD) blok je předcházený start bitem (logická 0) a doprovázený koncovým bitem (logická 1). Toto ukazuje Obrázek 3.9.



Obrázek 3.9 Command blok

Odpovědní (Response) blok má vždy jedno ze 4 schémat. To je závislé na kontextu. Délka je pak 48 bitů jako u command bloku nebo 136 bitů. To můžeme vidět na Obrázek 3.10.

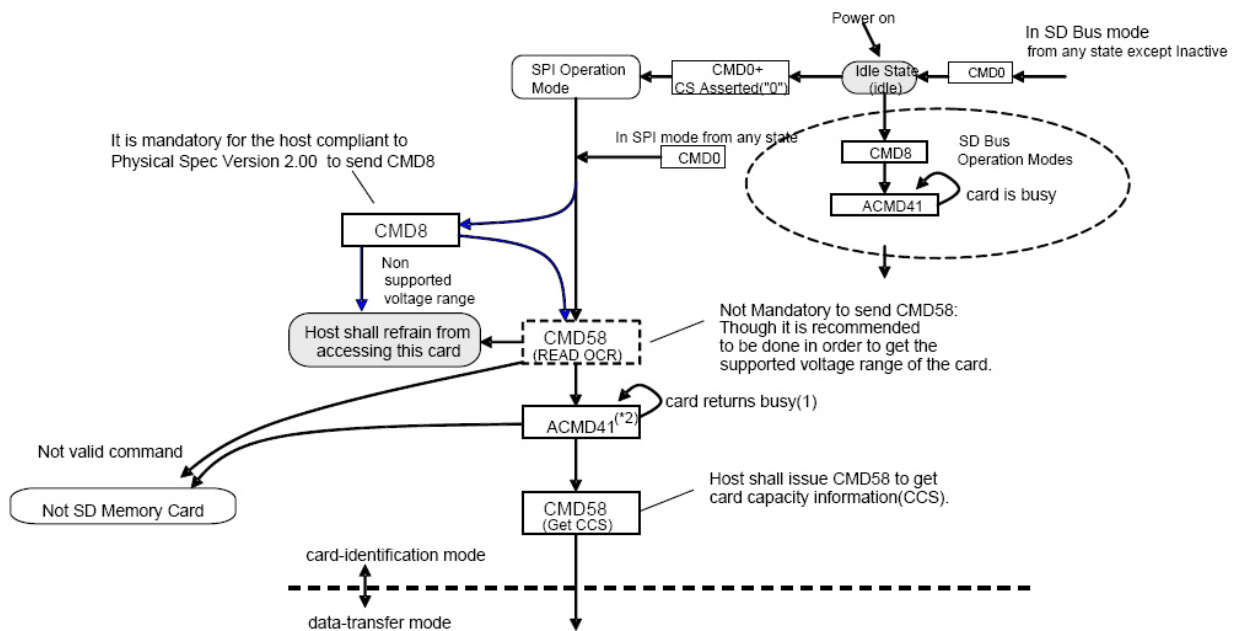


Obrázek 3.10 Response blok

Při posílání CMD je nejvýznamnější bit (MSB) vyslaný jako první. Význam CMD je detailně popsán v literatuře [13] v kapitole 7.3.1.3 Detailed Command Description.

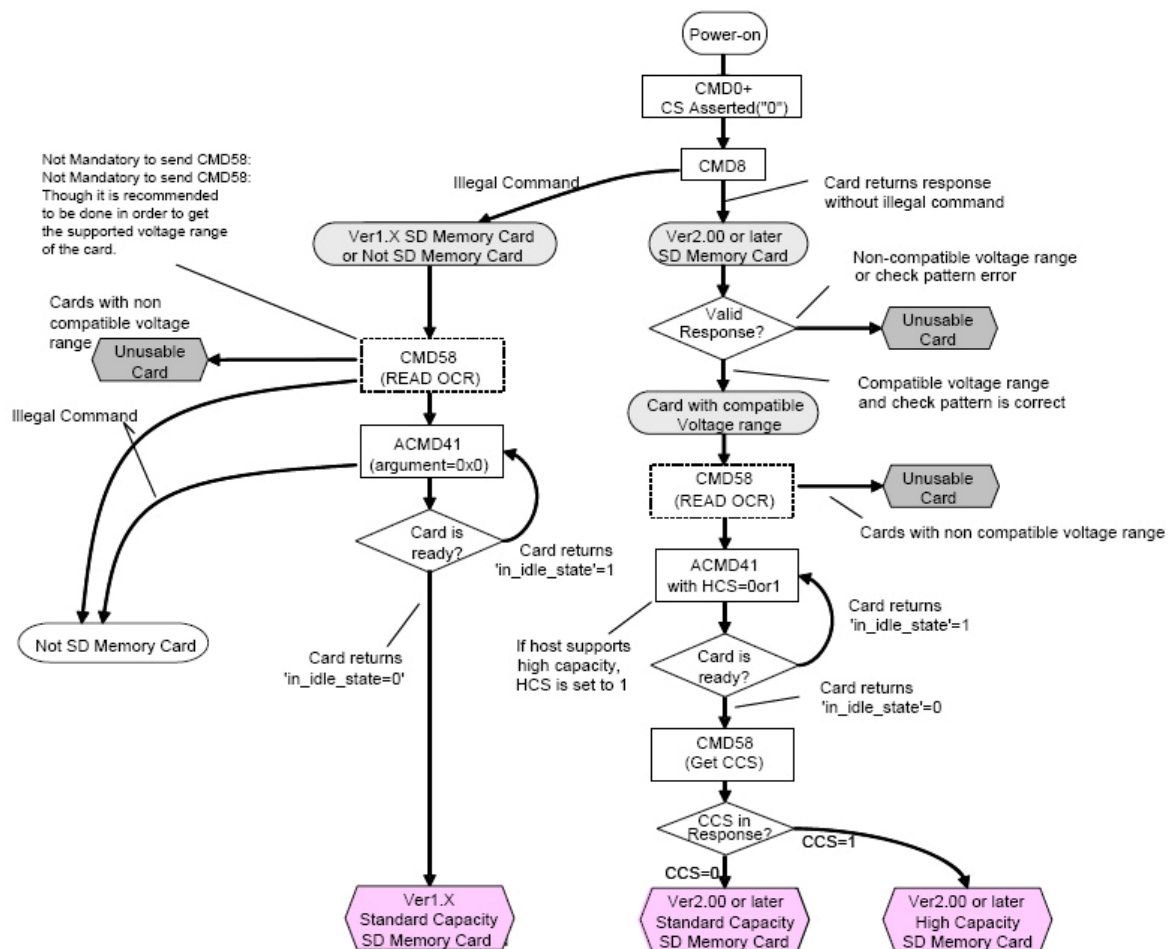
3.4.4.3 Inicializace

Po připojení napájení se SD karta nachází v SD módu. Následující diagram (Obrázek 3.11) zobrazuje přepnutí režimu z SD módu do SPI módu.



Obrázek 3.11 Diagram inicializace SPI módu

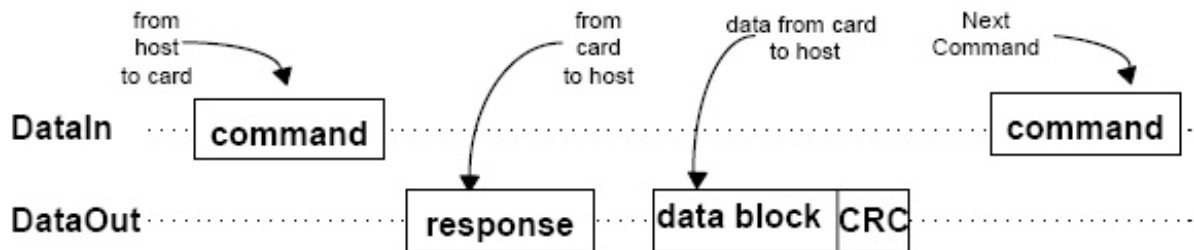
Poté co se přepneme do SPI módu, provedeme sekvenci inicializačních příkazů pro zjištění následujících informací o SD kartě: verze karty, rozsah operačního napětí. Tento inicializační tok je zobrazen na diagramu (Obrázek 3.12). V mém zařízení je naimplementována komunikace s SD kartou ve verzi 1.X. V době, kdy jsem pracoval na návrhu schématu DPS, nebyly ještě na trhu rozšířeny SD karty ve verzi 2.0. Za účelem testování a čtení z SD karty jsem, koupil tuto kartu a čtečku karet, která nepodporuje SDHC (verzi 2.0).



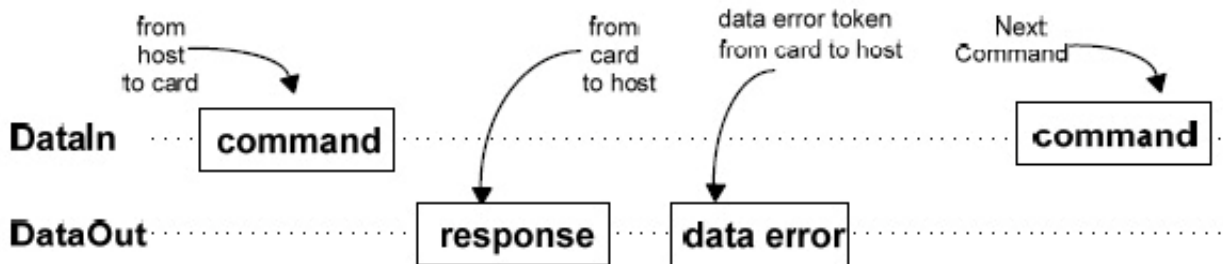
Obrázek 3.12 Inicializační tok

3.4.4.4 Čtení a zápis dat

Veškerá komunikace mezi hostem a kartou je řízena hostem (mastrem). Host startuje každou transakci. SPI mód podporuje přenos jednoho čteného nebo více čtených datových bloků (CMD17 nebo CMD18). Na příjem čtecího commandu karta odpoví response a pak následně data blokem. V případě standardní kapacity karty (verze 1.X) bude délka data bloku nastavena podle SET_BLOCKLEN (CMD16). V případě HDSC je velikost data bloku fixována na 512 Bytů bez ohledu na délku bloku, která je nastavena CMD16. Za data blok je připojen 16-bitový CRC součet generovaný podle polynomu $x^{16} + x^{12} + x^5 + 1$. Obrázek 3.13 ukazuje čtení jednoho data bloku. V případě, že přijmutý command je poškozen, karta nebude generovat žádná data na sběrnici, ale místo toho pošle speciální data error blok (Obrázek 3.14). V případě přenosu více datových bloků najednou vyšle host startovací příkaz (command) a na ukončení komunikace pošle příkaz CMD12.

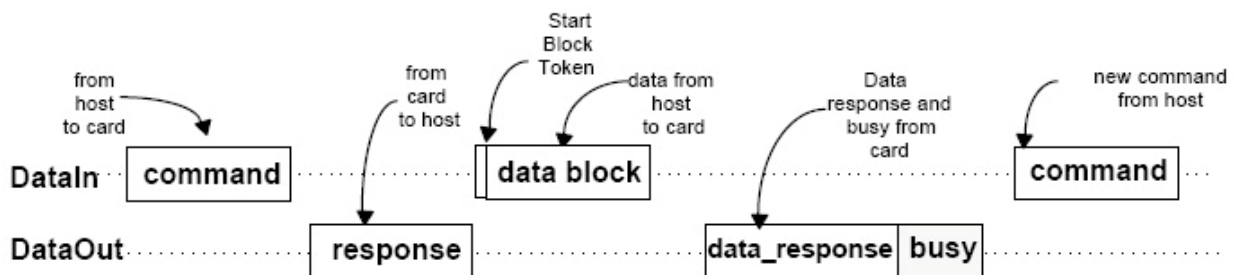


Obrázek 3.13 Čtení dat



Obrázek 3.14 Čtení dat – data error

Pro zápis dat (Obrázek 3.15) SPI podporuje jeden anebo více příkazů (CMD24 a CMD25), za kterým následuje blok dat. Když karta přijme command blok, odpoví response blokem a následně očekává od hosta data blok. Každý data blok má na začátku jeden startovní bit. Po přijetí data bloku odpoví karta odpovědním blokem. Jestliže data přijdou do karty bez chyb, budou zapsána do SD karty. Po dobu zaměstnání karty bude generován busy blok na výstup (DataOut). Po busy bloku by měl host zaslat příkaz SEND_STATUS (CMD13). Chyby jako například adresování mimo rozsah, porušení proti ochraně, proti zápisu atd. jsou detekovány pouze během zapisování dat na kartu.



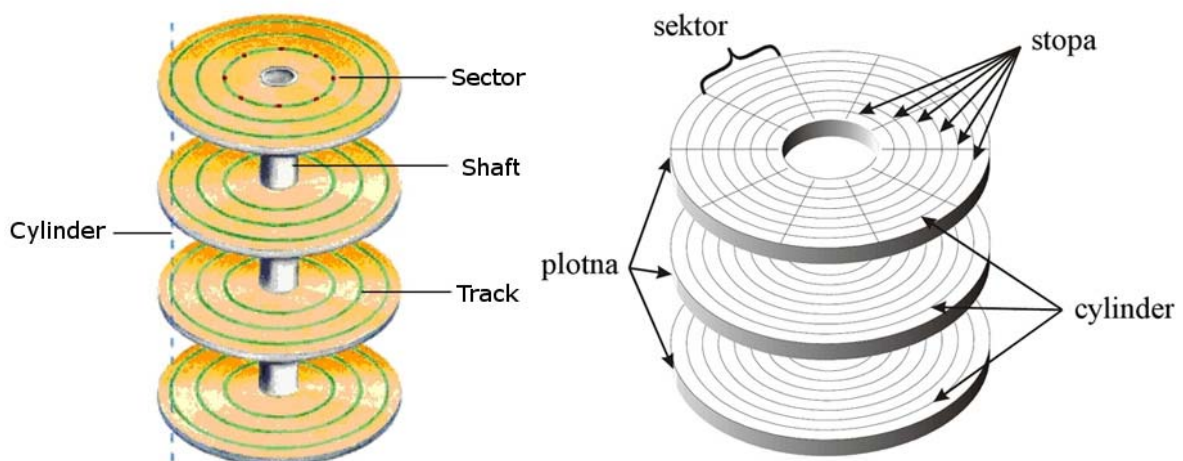
Obrázek 3.15 Zápis dat

3.5 Souborový systém

S příchodem a rozvojem počítačů a počítačových systémů byla potřeba ukládat a načítat data. Aby ukládání a následné čtení dat bylo vůbec možné rozumně provádět, byly vyvinuty souborové systémy, které definují způsob ukládání a indexování dat (souborů) v paměti (v PC nejčastěji HDD). Druhy některých souborových systémů:

- FAT12, FAT16, FAT32 (File Allocation Table)
- NTFS (New Technology File System)
- Ext2, ext3 (extended filesystem)

Nejmodernější souborové systémy jako např. NTFS, ext3 jsou žurnálové souborové systémy. Ty zapisují změny do záznamu nazývaného se žurnál (anglicky journal). Jeho účelem je ochrana dat před ztrátou integrity v případě například náhlého přerušení toku elektrického proudu nebo pádu systému. Jednotlivé operace jsou řazeny do transakcí.



Obrázek 3.16 Struktura disku

Nyní se tedy podívejme na vnitřní uspořádání disku. Nejmenší adresovatelnou jednotkou je sektor. Ve většině případů bývá jeho velikost 512 bytů. Z několika sektorů složíme alokační jednotku (cluster). Počet sektorů v clusteru je vždy násobkem 2^n (kde n je z množiny \mathbb{N}^+). Obrázek 3.16 ukazuje pomyslné uspořádání na disku, avšak ve skutečnosti na vnější části je jiná hustota sektorů na stopu než ve vnitřní části. Dnešní disky toto uspořádání tají (dokonce i operačnímu systému) a mají v sobě implementovanou dekódovací logiku na to, aby si určily pozici sektoru na disku. Každý hard disk musí mít shodný „startovací bod“, kde jsou uloženy klíčové informace o disku, jako je například počet oddílů a jejich seřazení atd. Na disku mohou být definovány až 4 primární oddíly, přičemž jeden z nich může být rozšířený. Toto rozložení a umístění oddílů je popsáno

v MBR (Master Boot Record), který se nachází v prvních 512 bytech na disku. MRB obsahuje následující:

- MASTER PARTITION TABLE: tato malá tabulka obsahuje popisy z oddílů, které jsou obsaženy na disku. Je v ní místo pouze pro informace popisující 4 oddíly. Proto tedy disk může mít pouze 4 oddíly, také nazývané PRIMARY PARTITIONS. Další oddíly jsou LOGICAL PARTITIONS, které jsou nalinkované v jednom z primárních oddílů. Z nichž je jeden označen jako aktivní (počítač ho může použít pro bootování).
- MASTER BOOT CODE: MRB obsahuje malý počáteční boot program (bios load), někdy také nazývaný jako zavaděč. Boot program uloží oddíl, který je použit k bootování (pozn. většinou je zde operační systém).

S rozdělením disku souvisí i pojem logického a fyzického adresování sektorů na disku. U každého oddílu je logické číslování sektorů počítáno od nuly, naopak fyzické číslování se začíná číslovat od začátku disku (tedy MBR) a je inkrementováno přes celý disk. Následující rozdělení ukazuje, k čemu je dobré takové rozdělení disku:

- instalace více operačních systémů
- použití různých souborových systémů
- oddělení dat od operačního systému

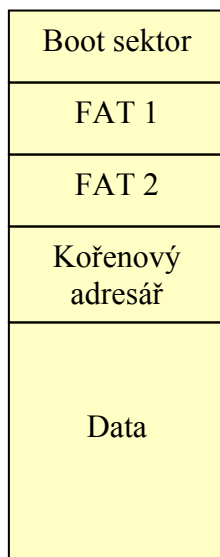
Partition table – tabulka rozdělení oddílů je umístěna na 446. bytu a její velikost je 64 bytů, což je právě 4*16 bytů. Obsah těchto 16 bytů je popsán v Tabulka 3.6

Ofset	Velikost	Popis
0x00	1 byt	stav (0x80 = systémová, 0x00 = nesystémový)
0x01	1 byt	Začátek oddílu – hlava
0x02	2 byty	Začátek oddílu – cylindr (válec) a sektor
0x04	1 byt	Typ oddílu
0x05	1 byt	Konec oddílu – hlava
0x06	2 byty	Konec oddílu – cylindr a sektor
0x08	4 byty	Počet sektorů mezi MBR a prvním sektorem v oddílu
0x0C	4 byty	Počet sektorů v oddílu

Tabulka 3.6 Partition tabulka v MBR

3.5.1 FAT32

Číslo za názvem souborového systému FAT určuje, kolika bytová položka je ve FAT tabulce. Základní myšlenky a principy jsou u všech těchto verzí stejné. Liší se pouze v detailech. Na Obrázek 3.17 vidíme základní strukturu souborového systému FAT32.



Obrázek 3.17 Struktura FAT32

Na začátku oddílu je boot sektor obsahující základní údaje o souborovém systému, který je nainstalován v tomto oddílu. Některé základní zajímavé informace o systému FAT32, které jsou uloženy v boot sektoru, můžeme vidět v následující Tabulka 3.7.

Název	Ofset	Velikost	Popis
BS_jmpBoot	0	3 byty	Skoková instrukce boot kódu. jmpBoot[0] = 0xEB, jmpBoot[1] = 0x??, jmpBoot[2] = 0x90 nebo jmpBoot[0] = 0xE9, jmpBoot[1] = 0x??, jmpBoot[2] = 0x??
BPB_BytsPerSec	11	2 byty	Počet bytů na sektor.
BPB_SecPerClus	13	1 byt	Počet sektorů na alokační jednotku (cluster).
BPB_RsvdSecCnt	14	2 byty	Počet rezervovaných sektorů. Počet sektorů před první FAT tabulkou od začátku oddílu.
BPB_NumFATs	16	1 byt	Počet FAT tabulek. Téměř vždy jsou 2.
BPB_RootEntCnt	17	2 byty	Tato hodnota slouží pro FAT12 a FAT16. Pro FAT32 musí být 0.
BPB_TotSec32	32	4 byty	Množství sektorů v oddílu.

Tabulka 3.7 Některé položky z boot sektoru

Název	Ofset	Velikost	Popis
BPB_FATsSz32	36	4 byty	Velikost jedné FAT tabulky v sektorech
BPB_RootClus	44	4 byty	Zde je cluster, kde začíná kořenový adresář. Obvykle 2, ale není požadovaný.

Tabulka 3.8 Některé položky z boot sektoru specifické pro FAT32

Poslední 2 byty v boot sektoru musí být 0xAA55. V tabulkách Tabulka 3.7 a Tabulka 3.8 jsou nejzákladnější parametry souborového systému, které jsou nutné pro operace s tímto systémem. Tyto parametry se nastavují při formátování média. Některé z nich se stanovují v závislosti na velikosti média a typu ukládaných dat (velké či malé soubory). Při inicializaci nejprve zjistíme, jaké verze je souborový systém. Spočítáme počet alokačních jednotek (CountOfCluster):

```

if (BPB_FATsSz16 != 0)
    FATsSz = BPB_FATsSz16;
else
    FATsSz = BPB_FATsSz32;

if (BPB_TotSec16 != 0)
    TotSec = BPB_TotSec16;
else
    TotSec = BPB_TotSec32;

DataSec = TotSec - (BPB_ResvdSecCnt + (BPB_NumFATs * FATsSz) + RootDirSectors);
CountofClusters = DataSec / BPB_SecPerClus;

if (CountofClusters < 4085) { /* Volume is FAT12 */ }
else
    if (CountofClusters < 65525) { /* Volume is FAT16 */ }
    else { /* Volume is FAT32 */ }

```

Nyní si vypočteme, kde začíná kořenový adresář (root directory):

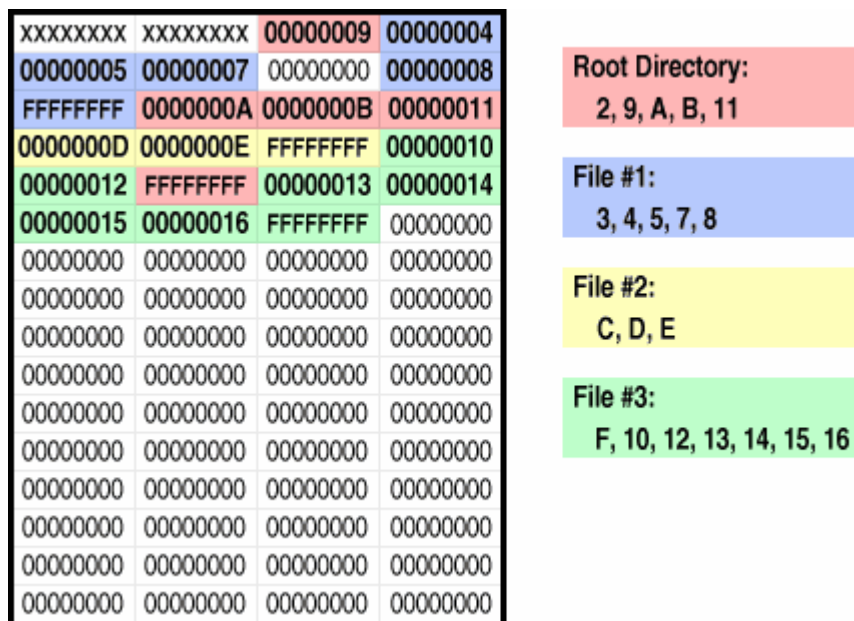
$$\text{RootDirSectors} = ((\text{BPB_RootEntCnt} * 32) + (\text{BPB_BytsPerSec} - 1)) / \text{BPB_BytsPerSec};$$

$$\text{FirstDataSector} = \text{BPB_ResvdSecCnt} + (\text{BPB_NumFATs} * \text{FATsSz}) + \text{RootDirSectors}.$$

Nyní máme v FirstDataSector číslo sektoru, kde začíná kořenový adresář. Hodnota RootDirSector vychází obvykle rovné nule. Alokační jednotku kořenového adresáře vyčteme z

Tabulka 3.8. Nejčastěji začíná na druhém clusteru. Výše jsem popsal boot sektor a pak také provedl nezbytné výpočty. Dále bude pozornost soustředěna na Obrázek 3.17 zobrazující FAT tabulku. Druhá FAT tabulka je kopií té první. Velikost FAT tabulky je určena

BPB_FATSz32 a začíná na sektoru BPB_RsvdSecCnt. Pro každou alokační jednotku na disku existuje ve FAT tabulce 32 bitový integer záznam. Hodnota v něm může indikovat další cluster, koncový cluster nebo vadný cluster. Tato 32 bitová hodnota je uložena ve formátu little endian. Princip FAT tabulky je ukázán na Obrázek 3.18. První dva cluster jsou rezervované (BPB_RootClus). V diskutovaném obrázku si můžeme povšimnout fragmentace disku.



Obrázek 3.18 Struktura FAT tabulky

Hodnota	Význam
0x?0000000	Volný cluster
0x?0000001	Rezervovaná hodnota; není použita
0x?0000002 - 0x?FFFFFFE	Použitý cluster; hodnota ukazuje na další cluster
0x?FFFFFF0 - 0x?FFFFFF6	Rezervovaná hodnota; není použita
0x?FFFFFF7	Vadný nebo rezervovaný cluster
0x?FFFFFF8 - 0x?FFFFFFF	Poslední cluster v souboru

Tabulka 3.9 Význam jednotlivých hodnot ve FAT tabulce

Adresářová struktura začíná v kořenovém adresáři. Odtud se dostaneme ke každému souboru či adresáři uloženém na tomto oddílu. Každý záznam má 32 bytů. Detailně je tento záznam popsán v Tabulka 3.1. Soubor od adresáře poznáme že má nulovou velikost.

Ofset	Velikost	Popis
0x00	8 bytů	Název souboru
0x08	3 byty	Přípona souboru
0x0B	1 byt	Atributy souboru
0x0C	1 byt	Rezervováno
0x0D	1 byt	Čas vytvoření; jednotka je 10ms a její rozsah 0 až 199
0x0E	2 byty	Čas vytvoření; hodina (0-23), minuta (0-59), sekunda/2 (0-29)
0x10	2 byty	Datum vytvoření; rok (0 = 1980, 127 = 2107), měsíc (1 = leden, 2 = únor, 12 = prosinec), den (1-31)
0x12	2 byty	Datum posledního přístupu
0x14	2 byty	Vyšší dva byty z prvního clusteru kde se nacházejí data
0x16	2 byty	Čas poslední modifikace
0x18	2 byty	Datum poslední modifikace
0x1A	2 byty	Nižší dva byty z prvního clusteru kde se nacházejí data
0x1C	4 byty	Velikost souboru

Tabulka 3.10 Obsah jednoho 32 bytového záznamu v adresářové struktuře

3.5.2 FAT32 & FAT16

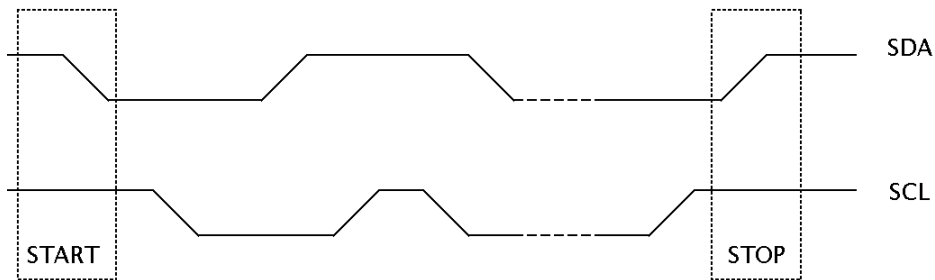
FAT32 je nástupcem FAT16. Hlavní rozdíl mezi těmito souborovými systémy je ve velikosti adresování clusteru, které jsou uloženy ve FAT tabulce. FAT32 používá 32-bitové respektive je využito 28-bitů. Z toho vyplývá že FAT16 končí na velikosti 2GB zatímco FAT32 má maximální velikost až 8TB. FAT16 má pevně dané umístění kořenového adresáře. Další vylepšení FAT32 je v tom, že se dá uložit více souborů do adresáře. FAT32 také začal podporovat delší názvy souboru než je 8 znaků. Proto jsem zvolil implementaci FAT32.

3.6 RS232

Rozhraní UART obsahuje 9 interních registrů. Konfigurace se provádí zápisem do těchto registrů. Těmito registry se ovlivňuje například přenosová rychlost, parita, stop bit. Paralelním zápisem 8 bitové hodnoty do vysílacího registru se automaticky aktivuje sériové vysílání dat jednotlivých bitů i včetně start bitu, parity a stop bitu. Přijatá data získáme paralelním čtením 8 bitové hodnoty z přijímacího registru. Vygenerování přerušení indikuje naplnění tohoto registru. Detailnější informace v dokumentaci [33].

3.7 I²C

Sběrnice I²C (Inter Integrated circuit Bus) je sériová sběrnice firmy Philips určená pro komunikaci mezi integrovanými obvody v rámci jednoho zařízení. Sběrnice má dva vodiče: datový SDA a hodinový SCL, přičemž přenosová rychlost se obvykle pohybuje na 100 kbitech/s. Podmínky START a STOP jsou na Obrázek 3.19. Během přenosu dat musí být data stabilní v době, kdy SCL=1.



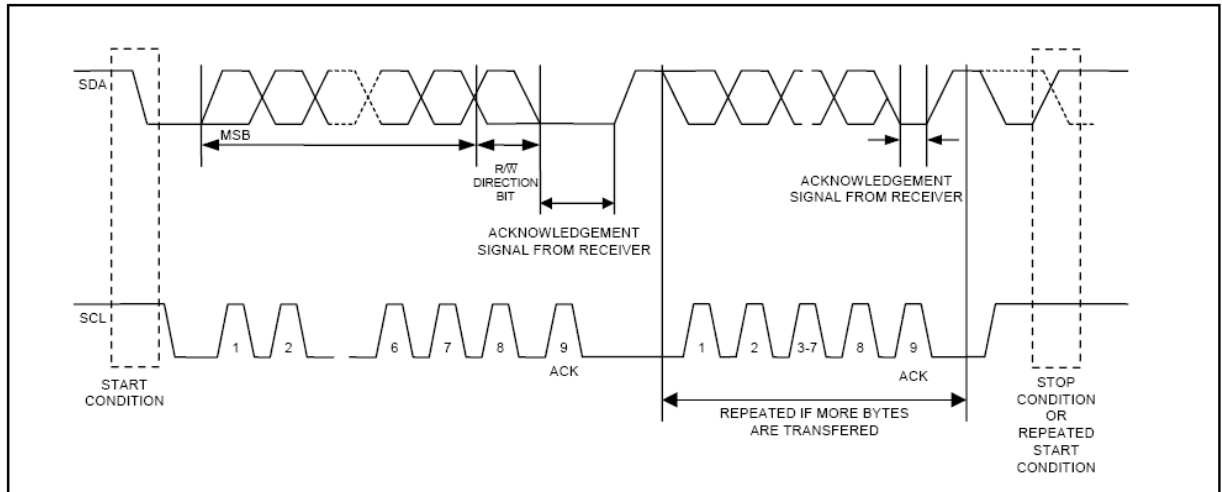
Obrázek 3.19 Základní průběhy na I²C

Za každou přenášenou slabikou (8 bitů) musí následovat potvrzení SDA, vysílané příjemcem dat. Synchronizační impuls generuje hlavní jednotka, potvrzovací datový bit vytváří příjemce dat.

3.7.1 Adresování

Přenos sběrnicí I²C začíná vždy vysláním adresy ihned po podmínce START. První vyslaná slabika (obr Obrázek 3.20) obsahuje 7 bitů adresy a osmý bit určuje R/\overline{W} . Když $R/\overline{W} = 0$, hlavní jednotka bude zapisovat data do vedlejší jednotky.

Čtyřbitová adresa ve skupině 1 rozděluje adresový prostor do 16 kategorií integrovaných obvodů. Kategorie 0000 a 1111 jsou vyhrazeny pro speciální účely. Tříbitová adresa ve skupině 2 je programovatelnou částí adresy vedlejší jednotky.



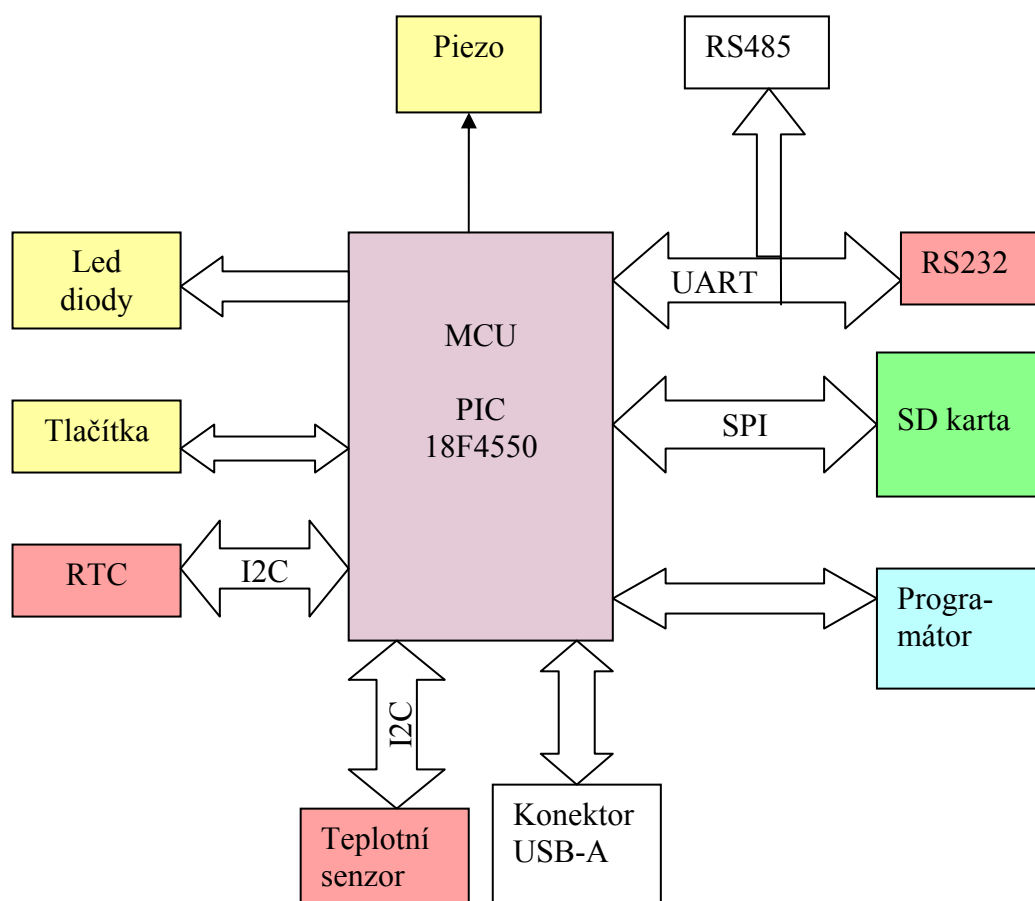
Obrázek 3.20 Adresování I²C

4 Návrh řešení

V této kapitole popíšu blokové schéma tohoto zařízení.

4.1 Blokový Návrh

Veškeré řízení bude obstarávat mikroprocesor PIC18F4550. Měl by zpracovat dotaz, který přijde po sériové lince a zpět odeslat zpracovanou odpověď po totožné sběrnici. Při návrhu jsem počítal, že by zařízení mohlo přijímat po sériové lince přicházející dotazy na aktuální čas a teplotu v zařízení. Dále také odbavovat dotazy po sériové lince na jednotlivé bloky dat z SD karty. Na desku jsem přidal další vstupně výstupní jednotky, které by měly z prvopočátku sloužit k ladění celého zařízení. V další fázi pak jako indikátory různých událostí (na Obrázek 4.1 jsou tyto bloky vyznačeny žlutě). Na Obrázek 4.1 je vidět blokově rozkreslené schéma zařízení.



Obrázek 4.1 Blokové schéma zařízení

4.2 Návrh schématu a plošného spoje

Návrh schématu bude proveden tak, aby bylo možné použít dvouvrstvého plošného spoje. Spodní vrstva (bottom) bude primárně sloužit jako podzemnění celého zařízení. To je nutné především z důvodů EMC (elektromagnetické kompatibility). Signály ve vrstvě bottom budou tahány, jen když by se ve vrstvě top (vrchní) měly křížit. Návrh byl proveden s pomocí programu Ranger.

5 Řešení

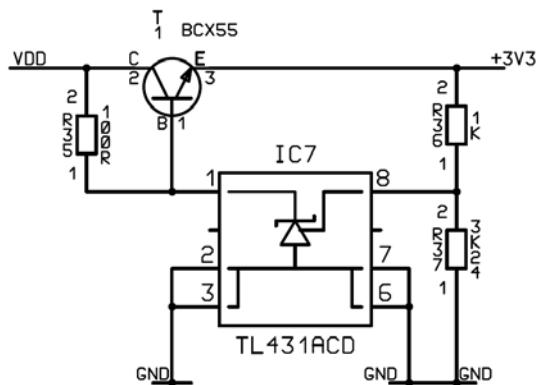
5.1 Hardware

5.1.1 Popis desky

Na svorky K1 se připojuje napájení celé DPS. Vstupní napětí je mezi 18V a 30V. Konektor K3 slouží k připojení programátoru MPLAB ICD 2 nebo debugeru MPLAB REAL ICE. Konektor K7 (devítipinový cannon) slouží pro připojení RS232. Pro výběr mezi komunikačními sběrnicemi RS 232 a RS 485 slouží propojky (jumpy) K4 až K6 (v mém případě jsem navolil K4, K5, K6 do pozic 2, 3 pro výběr RS232). Konektor K10 slouží k připojení sběrnice RS 485. Pomocí jumperů K8 a K9 je možno nastavit zakončení vedení a klidové stavy komunikační linky. Konektor K11 se skládá s jumperů, kde je vyvedeno na dvou pinech napětí +5V a GND. Na další piny byly vyvedeny nepoužité vývody z procesoru. Diody D7 až D10 jsou volně použitelné. Další 3 diody se nacházejí uvnitř tlačítek, která mohou signalizovat jejich stisknutí. Tlačítka S2 až S4 jsou určena k volnému použití. Tlačítko S1 je použito jako RESET procesoru. SD karta se zasunuje do připraveného konektoru, který nese označení IC2.

5.1.2 Napájení

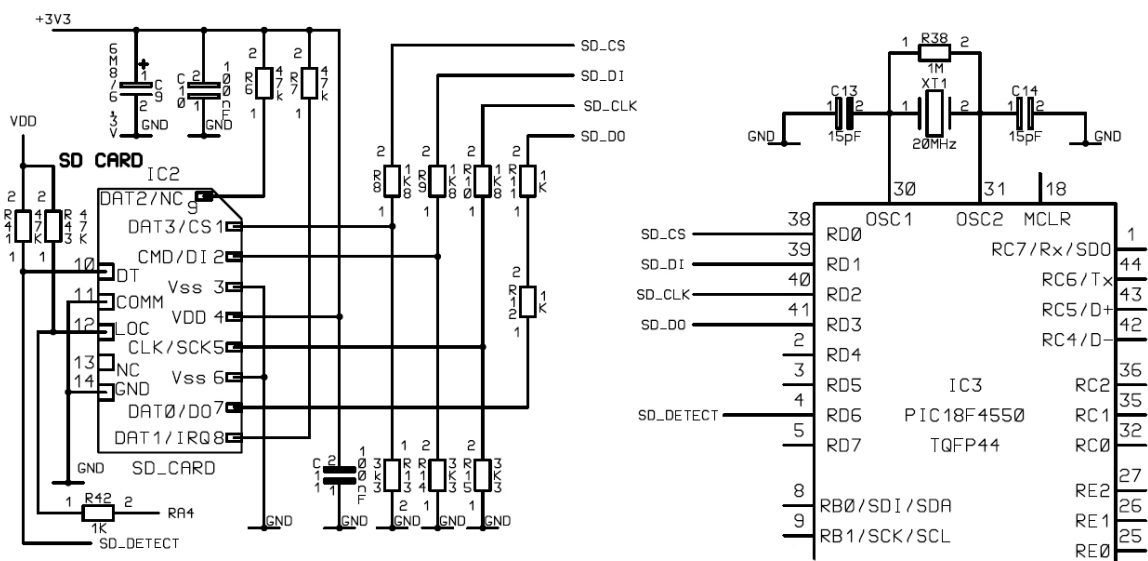
Všechny obvody jsou napájené 5V kromě SD karty, která potřebuje 3,3V. Pro celé zařízení bylo vytvořeno napájení z 18-30V (svorky K1 přičemž pin č.2 je zem) na 5V se zvlněním zhruba 40 mV špička-špička. K vytvoření tohoto spínaného napájecího zdroje byl použit integrovaný obvod MC34063AD. Tento obvod se stará o řízení spínaného zdroje. Zapojení zdroje bylo navrženo podle doporučení v dokumentaci od řídicího obvodu (viz použitá literatura [22]). Napájení pro SD kartu bylo z 5V na 3,3V obvodem na Obrázek 5.1. Odběr celého zařízení je zhruba 21 mA. Ve chvíli kdy se zahájí komunikace po RS232 se spotřeba zvýší zhruba na 30mA.



Obrázek 5.1 Napájení z 5V na 3,3V pro SD kartu

5.1.3 Propojení SD karty s CPU

Procesor PIC18f4550 má hardwarovou podporu SPI, ale na těchto portech je zároveň I2C, SPI a USART (konkrétně na RC7 je RX/DT/SDO). Na port RC7 jsem připojil RS232, a proto jsem připojil SD kartu podle Obrázek 5.2 a implementoval softwarové SPI. Napěťové přizpůsobení karty a procesoru jsem provedl děličem napětí (z 5V na 3,3V).



Obrázek 5.2 Propojení SD karty s CPU

5.1.4 Propojení RTC TIME s CPU přes I2C BUS

Procesor PIC18f4550 má hardwarovou podporu I2C, ale porty jsou využity na komunikaci přes RS232 s PC, proto je rozhraní I2C obslouženo softwarově.

5.1.5 Propojení externího měření teploty

Na volné porty vyvedené na konektor K11 jsem připojil externí aplikaci. Jedná se o obvod který měří teplotu. S procesorem komunikuje po sběrnici I²C, která je obsloužena pomocí softwarového I²C.

5.2 Firmware

5.2.1 Popis protokolu RS232

Jednou z hlavních součástí aplikace je komunikační protokol. Protokol slouží pro ladění aplikace a přenos dat mezi SD kartou a PC.

5.2.1.1 Obecná struktura protokolu

Komunikace ve směru nadřízený systém (master) → zařízení (slave)

a) zpráva bez datového pole

SD1	DA	SA	FC	FCS	ED
------------	-----------	-----------	-----------	------------	-----------

b) zpráva s datovým polem

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

Komunikace ve směru zařízení → nadřízený systém (potvrzení, odpověď)

a) odpověď bez datového pole (potvrzení)

SD1	DA	SA	FC	FCS	ED
------------	-----------	-----------	-----------	------------	-----------

b) odpověď s datovým polem

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

Význam použitých symbolů

SD1 začátek rámce (Start Delimiter), kód 10H

SD2 začátek rámce (Start Delimiter), kód 68H

LE délka informačního pole (Length) začíná bytem DA a končí bytem před FCS. Délka pole 4 - 249.

LEr opakování bytu délky informačního pole (Length repeat)

DA adresa cílové stanice (Destination Address)

SA adresa zdrojové stanice (Source Address)

FC řídící byte (Frame Control)

DATA pole dat maximálně 246 bytů
FCS kontrolní součet (Frame Check Sum)
ED konec rámce (End Delimiter), kód 16H

LE, LEr - Délka informačního pole

Oba byty v hlavičce zprávy s proměnnou délkou informačního pole obsahují počet bytů informačního pole. Je v tom započítáno DA, SA, FC a DATA. Nejnižší hodnota LE je 4, nejvyšší 249. Tím lze přenést 1 - 246 bytů dat.

DA, SA - Adresa stanice (DA - cílová, SA - zdrojová)

Adresy mohou ležet v rozmezí 0 - 126.

FC – Řídící byt

Řídící byt v hlavičce rámce obsahuje přenosovou funkci a informaci zabráňující ztrátě resp. zdvojení zprávy.

b8	b7	b6	b5	b4	b3	b2	b1
RES	1	1	0	FUNKCE			
	0	0	0				

RES – rezervováno

FUNKCE: b7 = 1 – rámeček výzvy

Kód	funkce
0x03	Send Dat with Acknowledge poslání dat s potvrzením
0x09	Request FDL – Status With Reply dotaz na Status
0x0C	Send and Request Data poslání a požadavek na data

FUNKCE: b7 = 0 – rámeček potvrzení nebo odpovědi

Kód	funkce
0x00	Acknowledgement positive kladné potvrzení
0x01	Error checksum chyba kontrolního součtu

0x02	Acknowledgement negative záporné potvrzení
0x08	Response FDL / FMA - Date vyslání dat

FCS - kontrolní součet

Kontrolní součet je dán aritmetickým součtem dat informačního rámce DA, SA, FC a DATA modulo 256 (100h) se zanedbáním vyšších řádů vzniklých přenosem 256 (100h).

$$25h = (24h + 30h + 37h + 52h + 48h) \text{ MOD } 100h$$

$$\text{Pro } SD1: \sum_{DA}^{FC} \text{ mod } 256 \quad \text{a pro } SD2: \sum_{DA}^{FCS-1} \text{ mod } 256$$

Při chybě kontrolního součtu je vyslána zprava s pevnou délkou a FC = 1.

5.2.1.2 Aplikační vrstva

Aplikační vrstva (datová část) implementuje protokol. K dispozici jsou následující služby:

INX	Popis	Poznámka
0x00	Čtení identifikace přístroje (verze a typ)	odpověď 42 znaků
0x03	Čtení stavu zařízení	
0x10	Nastavení nové adresy zařízení	zápis
0x20	Zápis dat do SD karty	zápis
0x2E	Nastavení času RTC obvodu	zápis
0x33	Čtení DIR ROOT	
0x34	Čtení DIR FILE	
0x35	Čtení SEND FILE	
0x36	Čtení sektoru SD karty	

Zahájení komunikace – navázání spojení

dotaz

SD1	DA	SA	FC	FCS	ED
------------	-----------	-----------	-----------	------------	-----------

$$FC = 0x69$$

kladná odpověď

SD1	DA	SA	FC	FCS	ED
------------	-----------	-----------	-----------	------------	-----------

$$FC = 0x00$$

Příklad: Navázání komunikace mezi přístrojem s nastavenou adresou 2 a nařízeným systémem s nastavenou adresou 4.

dotaz: 0x10, 0x02, 0x04, 0x69, 0x6F, 0x16

odpověď: 0x10, 0x04, 0x02, 0x00, 0x06, 0x16

Zjištění typu zařízení – identifikace

zkrácený dotaz (délka pole LE, LEr = 4)

SD2	LE	LEr	SD2	DA	SA	FC	INX	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	------------	-----------

FC 0x6C

INX INDEX 0x04

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

FC 0x08

DATA Typ zařízení + verze zařízení 21 +21 znaků

Zápis to RTC – nastavení času

dotaz

SD2	LE	LEr	SD2	DA	SA	FC	INX	SEC	MIN	HOD	FCS	ED
							DT	MES	ROK			

FC 0x63

INX INDEX 0x2E

SEC SEKUNDY 0-59

MIN MINUTY 0-59

HOD HODINY 0-23

DT DEN 1-31

MES MĚSÍC 1-12

ROK ROK 0-59

kladná odpověď

SD1	DA	SA	FC	FCS	ED
------------	-----------	-----------	-----------	------------	-----------

FC = 0x00, v případě chyby FC = 0x02

Čtení sektoru SD karty

dotaz

SD2	LE	LEr	SD2	DA	SA	FC	INX	SECu32	OFS	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	---------------	------------	------------	-----------

FC 0x6C

INX INDEX 0x36

SECu32 SEKTOR číslo sektoru (unsigned long 32 bit)

OFS OFFSET posuv v sektoru po 64 bytech

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

DATA 64 byte

Čtení kořenového adresáře (root dir)

- inicializace čtení root directory

dotaz

SD2	LE	LEr	SD2	DA	SA	FC	INX	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	------------	-----------

FC 0x6C

INX INDEX 0x33

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

FC 0x08

DATA datové pole v ASCII ... "Root Directory Entry " (21 znaků)

- požadavek na další data

NDR

NDR 0x33

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	INX	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	-------------	------------	-----------

INX 0 nejsou k dispozici další data

DATA 32 byte

Čtení adresáře (dir file)

- inicializace čtení zadaného adresáře začínajícím sektorem

dotaz

SD2	LE	LEr	SD2	DA	SA	FC	INX	SECu32	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	---------------	------------	-----------

FC 0x6C

INX INDEX 0x34
 SECu32 SEKTOR číslo sektoru (unsigned long 32 bit)

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

FC 0x08

DATA datové pole v ASCII ... "Directory Listing " (21 znaků)

➤ požadavek na další data v zadaném adresáři

NDF

NDF 0x34

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	INX	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	-------------	------------	-----------

INX 0 nejsou k dispozici další data

DATA 32 byte

Čtení souboru

➤ inicializace čtení souboru

dotaz

SD2	LE	LEr	SD2	DA	SA	FC	INX	SECu32	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	---------------	------------	-----------

FC 0x6C

INX INDEX 0x35

SECu32 SEKTOR číslo sektoru (unsigned long 32 bit)

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

FC 0x08

DATA datové pole v ASCII ... " Send file " (21 znaků)

➤ požadavek na další data

NSF

NSF 0x35

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	INX	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	------------	-------------	------------	-----------

INX 0 nejsou k dispozici další data

DATA 64 byte

Čtení stavu zařízení

dotaz

SD2	LE	LEr	SD2	DA	SA	FC	INX PB	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	---------------	------------	-----------

FC 0x6C

INX INDEX 0x03

PB POČET_BYTE počet byte v tabulce

kladná odpověď

SD2	LE	LEr	SD2	DA	SA	FC	DATA	FCS	ED
------------	-----------	------------	------------	-----------	-----------	-----------	-------------	------------	-----------

FC 0x08

DATA 1 - n byte dle tab.

význam	rozsah	typ	počet byte
RTC sekundy	0-59	char	1
RTC minuty	0-59	char	1
RTC hodiny	0-23	char	1
RTC den	1-31	char	1
RTC měsíc	1-12	char	1
RTC rok	0-99	char	1
Teplota	teplota	float	4
Aktuální sektor		int	4
Počet sektorů na cluster		int	2
Začátek root directory		int	4

5.2.2 Implementace SW SPI

Nyní se podíváme na softwarovou implementaci SPI sběrnice z důvodů, které jsem uvedl v kapitole 5.1.3. Při spuštění zařízení provedu inicializaci portů. Nastavím je jako vstup nebo výstup. Hlavní funkcí SPI je `sw_send_SPI`. Nejprve otestuji nejvyšší bit vstupní proměnné a podle ní přepnu na vstup nebo výstup. Pak provedu v cyklu následující operace: rotaci o jeden bit doprava vstupní proměnné a před a za vložím čekací cyklus, abych dosáhl frekvence 100kHz. Osciloskop jsem použil, abych zjistil počet čekacích cyklů, které mám vložit. Na konci funkce vrátím vstupní proměnnou.

```
unsigned char sw_send_SPI( unsigned char output)
{
```

```

char BitCount;                // Bit counter
static char input;
input = output;
// SCK idles low
// Data output after falling edge of SCK
// Data sampled before rising edge of SCK

if(input&0x80)                // test on highest bit
    SW_DOUT_PIN = 1;
else
    SW_DOUT_PIN = 0;          // Set Dout to MSB of data
BitCount = 8;                // Do 8-bits

do                            // Loop 8 times
{
    Nop();
    Nop();
    Nop();
#ifdef __100KHZ
    dly(3);
#endif
    SW_SCK_PIN = 1;          // Set the SCK pin
    STATUSbits.C = 0;        // Set the carry bit according
    if(SW_DIN_PIN)          // to the Din pin
        STATUSbits.C = 1;
    _asm
        rlcfc input,1,1     // Rlcf(input); Rotate the carry into the data
byte
    _endasm
#ifdef __100KHZ
    dly(3);
#endif
    Nop();                  // Produces a 50% duty cycle clock
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    Nop();
    SW_SCK_PIN = 0;         // Clear the SCK pin
    Nop();
    if(input&0x80)          // the MSB of data
        SW_DOUT_PIN = 1;
    else
    {
        SW_DOUT_PIN = 0;    // Set Dout to the next bit according to
        Nop();
    }
}

```

```

    BitCount--; // Count iterations through loop
} while(BitCount);
return(input); // Return the received data
}

```

5.2.3 Implementace SW I²C

Při navrhování této sběrnice jsem nejprve vyřešil START a STOP bit protokolu I2C. Dále jsem napsal obsluhu SCL a signály AKC a NACK. Nad těmito elementárními funkcemi jsem napsal datovou vrstvu, která zajišťuje komunikaci s obvodem. Na závěr jsem zkontroloval na osciloskopu šířku a prodlevu jednotlivých impulsů na sběrnici a doladil prodlevy podle příslušné dokumentace k obvodu.

5.2.4 Popis funkce zařízení

Pro demonstrování správné funkce souborového systému jsem zvolil ukládání teploty okolí, která je získávána z čidla a je zapisována do souboru. Při každém zápisu se do souboru ukládá teplota a čas, kdy bylo toto měření provedeno. Tyto informace jsou uloženy v pořadí, které ukazuje

Tabulka 5.1. Jsou uloženy v hexadecimálním tvaru z důvodu úspory místa. Jelikož za jediný den soubor s uloženými daty může být velký až 23kB. Zhruba za jedenáct dní stálého ukládání je vyčerpána kapacita na 256MB SD kartě. Adresářový strom je uspořádán tak, že se každý den vytvoří nový soubor. Tento soubor se ukládá do složky roku, ve kterém byl vytvořen. V této složce vyhledá název měsíce, kam následně soubor uloží. Při stisknutí tlačítka S4 se provede jeden zápis do souboru. Při stisknutí tlačítka S3 se aktivuje zapisování do souboru každou minutu. Deaktivace se provede stejným tlačítkem. Tato funkce je indikována rozsvícením tlačítka S3. Tlačítko S1 slouží k resetování zařízení. Tlačítko S2 není využito. Dioda D7 signalizuje příjem z komunikační linky. Dioda D8 signalizuje vysílání do komunikační linky. Dioda D9 signalizuje čtení/zápis z/do SD karty. Jestliže svítí dioda D10, tak vložená SD karta v slotu je naformátována na podporovaný souborový systém tohoto zařízení.

Pořadí bytu	Popis uložených dat
0	Sekunda (0 až 59)
1	Minuta (0 až 59)
2	Hodina (0 až 23)
3	0x20 (mezera)
4	0x20

5	Den (1 až 31)
6	Měsíc (1 až 12)
7	Rok (rok 2000 je 00)
8	0x20
9	0x20
10	Naměřená teplota (32-bitový float)
11	
12	
13	
14	0x0D
15	0x0A

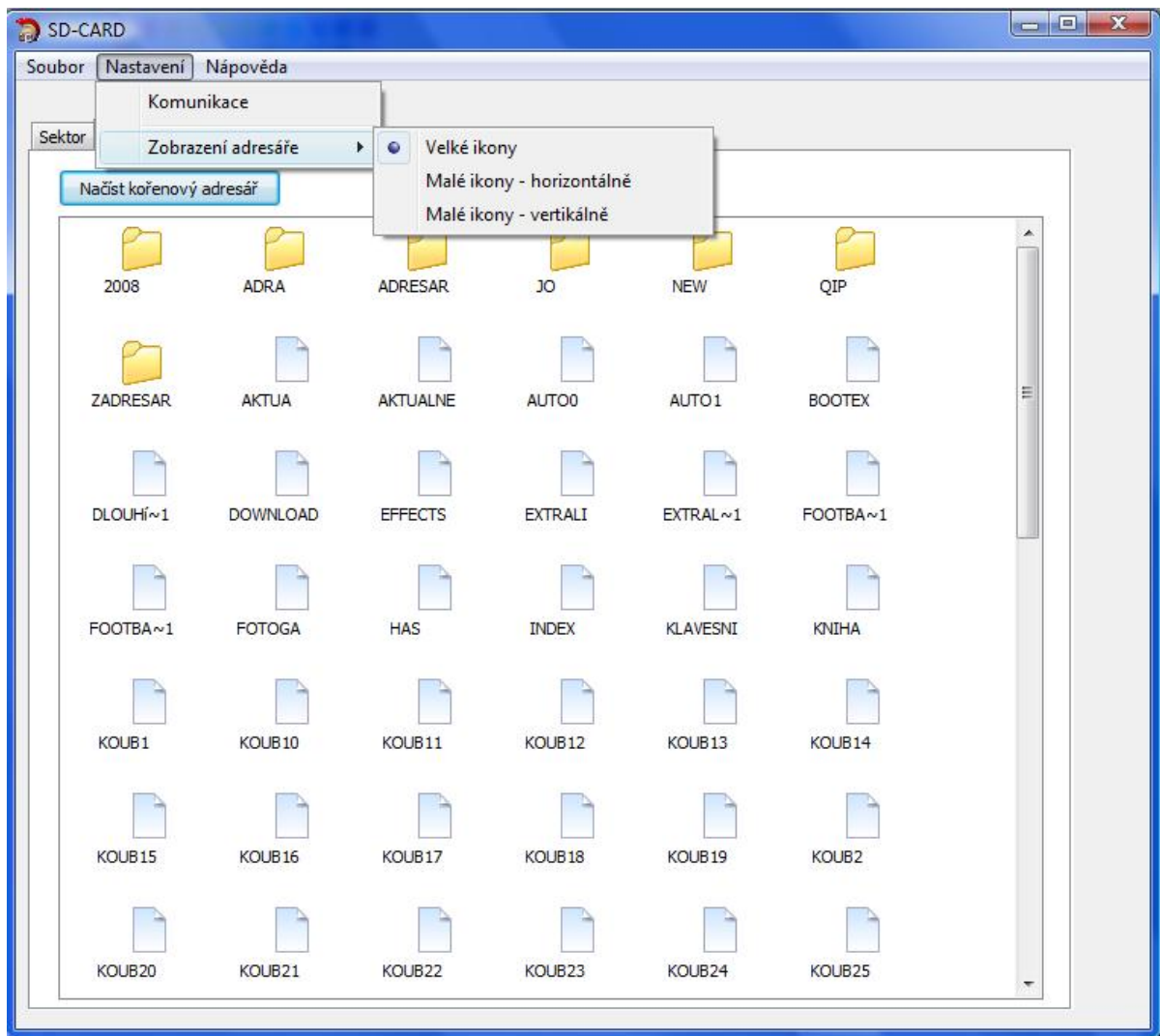
Tabulka 5.1 Ukládané informace do souboru

5.3 Testování

Zapsaný soubor s naměřenými údaji jsem přečetl pomocí čtečky paměťových karet. Dále jsem nahrál náhodný adresářový strom na SD kartu a ten jsem následně procházel pomocí testovací aplikace na PC, která komunikovala se zařízením pomocí RS232. Ke zkoumání souborového systému jsem dále použil program winhex. Velmi mi pomohl při psaní kódu, který implementuje FAT32.

6 Testovací aplikace

Jelikož zařízení nemá žádnou zobrazovací jednotku, jeho funkce souborového systému byla ověřována přes komunikační linku a některé stavy byly signalizovány pomocí LED diod. Testovací aplikace byla naprogramována v C++. Ke snadné obsluze přispívá také GUI. V této aplikaci bylo implementováno procházení adresářové struktury SD karty, ve které byl souborový systém FAT32. Na procházení adresářové struktury byla ověřena správná implementace procházení sektorů a FAT tabulky v procesoru. Dále je možné přijmout aktuální stav některých proměnných v zařízení (např. datum, teplota, informace o souborovém systému). Testovací aplikace dokáže vyčíst libovolný sektor, který se nachází na SD kartě. Z SD karty je možné také přečíst libovolný soubor z adresářové struktury. Dále se dá nastavit čas v RTC obvodu na desce. Zjištění, na kterém portu a jakou má zařízení adresu je možné automatickou detekcí v menu výběrem nastavení a pak dále kliknutím na komunikaci. Při vývoji této aplikace jsem zjistil, že když komunikuji po sériové lince, celá aplikace během komunikace zamrzne. Když by použil tuto aplikaci někdo další, mohl by si myslet, že program přestal fungovat. Z tohoto důvodu jsem implementoval komunikaci do samostatného vlákna. Jestliže v aplikaci nastane nějaký požadavek na zařízení, které je připojeno přes RS232, aplikace přidá do fronty požadavek. Vlákno si tento požadavek převezme. Po zpracování předá přijatá data funkce, která je zpracuje. Tato funkce je určena při vkládání požadavku do fronty. Na tomto je postavena celá aplikace.



Obrázek 6.1 GUI testovací aplikace

7 Problémy při řešení

Při ladění aplikace se objevily některé nedostatky v hardwarovém zapojení. Kontakty tlačítek nebyly přitaženy do klidové polohy odporem 10 K na +5 V. Tyto odpory (R44 až R46) byly doplněny do aplikace dodatečně. Dále nebyla zakončena sběrnice I2C pro RTC TIME signály SCL a SDA nebyly přitaženy na plus, proto byla sběrnice dodatečně zakončena odpory 1 K (R47 a R48). Kondenzátory (C18 a C19) u převodníku na RS232 byly do schématu nakresleny (navrženy) dle doporučeného zapojení podle literatury [29], kde byly nakresleny s obrácenou polaritou. Nejprve jsem zprovoznil komunikaci I²C na DS75 (obvod teploty). Pak jsem tento odladěný kód použil na komunikaci s RTC, který je taktéž připojen na tuto sběrnici. Tyto dva obvody vyrábí jedna firma, proto jsem předpokládal, že komunikace bude pracovat. Připojil jsem osciloskop a podle manuálu [30] jsem prodloužil intervaly mezi jednotlivými bity a délku impulsu. Při restartu procesoru je defaultně nastaven port RB0 a RB1 jako analogový vstup, bylo nutno tento port přestavit z analogového na digitální port pro sběrnici I²C. Dalším velice zásadním problémem byl překladač z jazyka C do assembleru. Uvedu jeden příklad za všechny. Při volání funkce jsem ji posílal jako parametr pointer. V těle funkce jsem chtěl zvětšit proměnnou, na kterou ukazoval pointer a tedy jsem napsal „*prom ++; “. Tato instrukce zvětšila pointer a ne proměnnou, na kterou ukazoval.

8 Závěr

Zařízení pro ukládání dat na SD kartu jsem celé navrhl v rámci své bakalářské práce. Začal jsem kreslením schématu. Potom přišlo na řadu navržení desky plošného spoje. Následně proběhlo osazení DPS součástkami. Poté jsem se věnoval psaní firmwaru a softwaru. Implementoval jsem souborový systém, který je uložen na SD kartě. Komunikaci mezi procesorem a SD kartou. Komunikaci po sběrnici mezi počítačem a zařízením. V zadání jsem se odchytil od bodu, kdy jsem měl ukládat naměřené hodnoty z RS232. Místo toho jsem připojil externí obvod na měření teploty. Tu pak ukládám do souboru, který je ve složce podle aktuálního data. Udělal jsem to proto, aby zařízení mohlo nezávisle pracovat na jakémkoliv místě. Po několika dnech je možné k zařízení přijít a vyčíst naměřená data. Je to možné dvěma způsoby. Jedna možnost je připojení PC přes RS232. Druhá možnost je vyjmutí SD karty a vložení do jakékoliv čtečky paměťových karet.

Programování v jazyce C je velice příjemné a velmi urychluje vývoj aplikace oproti assembleru, je zároveň přehlednější a přenositelné na jiné typy procesorů. Při další konstrukci by bylo vhodnější vybrat procesor s větší kapacitou paměti jak RAM tak CODE. Vhodné by bylo postavit zařízení na 3,3 V logice z důvodů snížení nákladů na napájení. Vlastní spotřeba zařízení je do 1W.

9 Literatura

[1] MMC PROJECT

http://www.compsys1.com/workbench/On_top_of_the_Bench/MMC_Project/mmc_project.html

[2] STK1000

http://www.avrfreaks.net/index.php?module=Freaks%20Tools&func=viewItem&item_id=709

[3] ACER http://ap.apacer.com/ap/products/Internal_Card_Reader_AE161.htm

[4] AVR 8-Bit RISC – Devices

http://www.atmel.com/dyn/products/devices.asp?family_id=607

[5] Atmel Products -Microcontrollers - AVR32 32-bit MCU

<http://www.atmel.com/products/avr32/default.asp>

[6] AVR freaks - Stránka milovníků MCU firmy atmel <http://www.avrfreaks.net/>

http://www.avrfreaks.net/index.php?module=Freaks%20Tools&func=viewItem&item_id=709

[7] Wikipedia – Compact flash <http://en.wikipedia.org/wiki/CompactFlash>

[8] Wikipedia – MultiMediaCard <http://en.wikipedia.org/wiki/MultiMediaCard>

[9] Wikipedia – Comparison of memory cards

http://en.wikipedia.org/wiki/Comparison_of_memory_cards

[10] Wikipedia – Secure Digital card http://en.wikipedia.org/wiki/Sd_card

[11] SD Card Association home page <http://www.sdcard.org/home>

[12] SD Card Association – About the SD Memory Card http://www.sdcard.org/about/memory_card/

[13] SD Card Association – SD Specifications – Physical Layer – datasheet

http://www.sdcard.org/about/memory_card/pls/Simplified_Physical_Layer_Spec.pdf

[14] SD Specifications - SD Host Controller nebo Simplified Version of SD Host Controller Spec

http://www.sdcard.org/about/host_controller/simple_spec/Simplified_SD_Host_Controller_Spec.pdf

[15] Wikipedia – Serial Peripheral Interface Bus

http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[16] SPI - Serial Peripheral Interface <http://www.mct.net/faq/spi.html>

[17] FAT32 Structure Information - Written by Jack Dobiash

<http://home.teleport.com/~brainy/fat32.htm>

[18] How to Read and Write to a Hard Drive <http://home.teleport.com/~brainy/diskaccess.htm>

[19] Paul's 8051 Code Library: Understanding the FAT32 Filesystem

<http://www.pjrc.com/tech/8051/ide/fat32.html>

[20] Microsoft - FAT32 File System Specification

<http://www.microsoft.com/whdc/system/platform/firmware/fatgen.msp>

[21] Embedded SD Card/MMC Card http://www.freelabs.com/~whitis/sd_card/

[22] Step–Down Switching Regulators MC34063A

http://alpha.octopart.com/ON_Semiconductor_MC34063AD.pdf

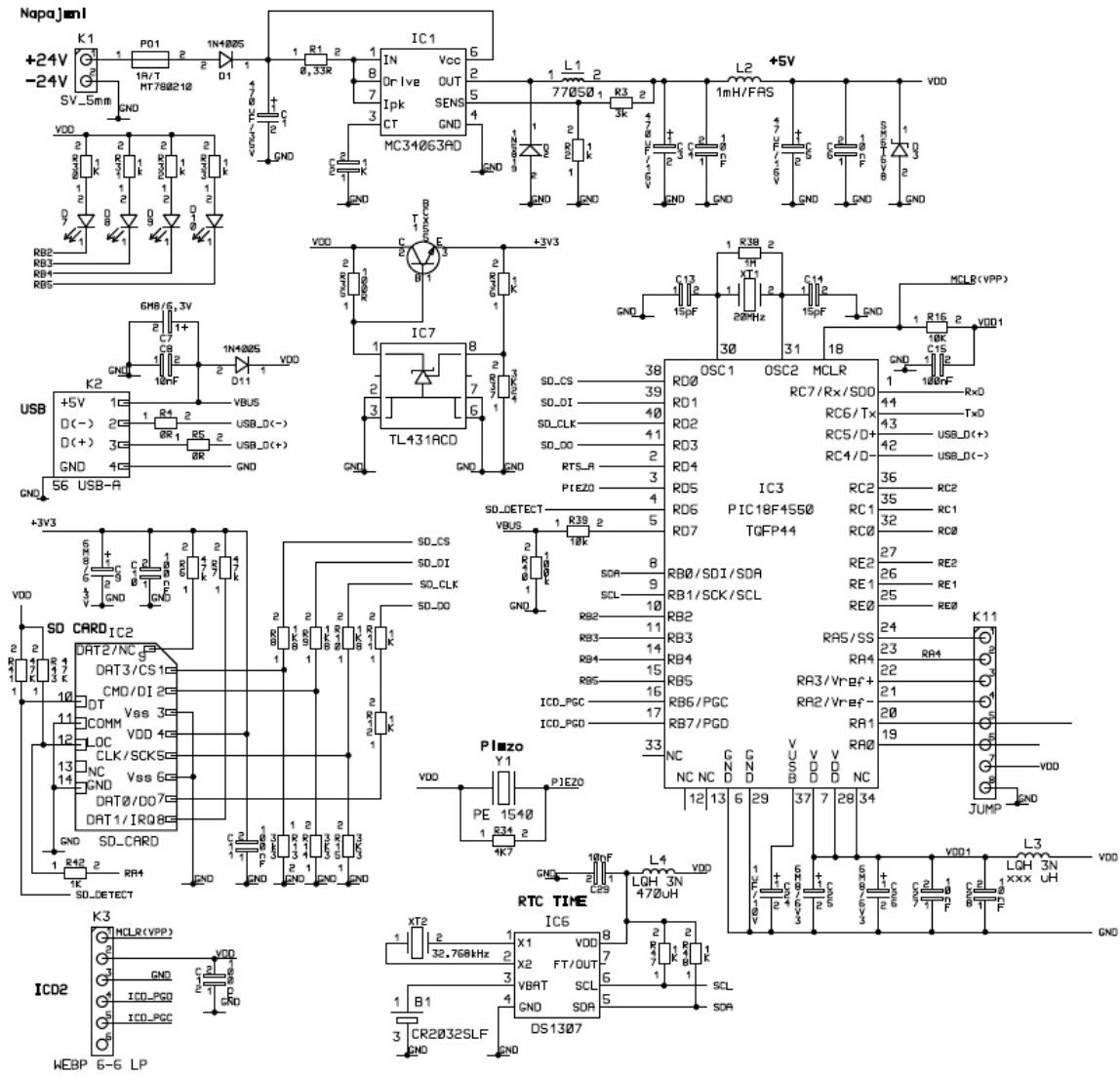
[23] <http://www.stud.u-szeged.hu/Toth.Peter.3/30400g.pdf>

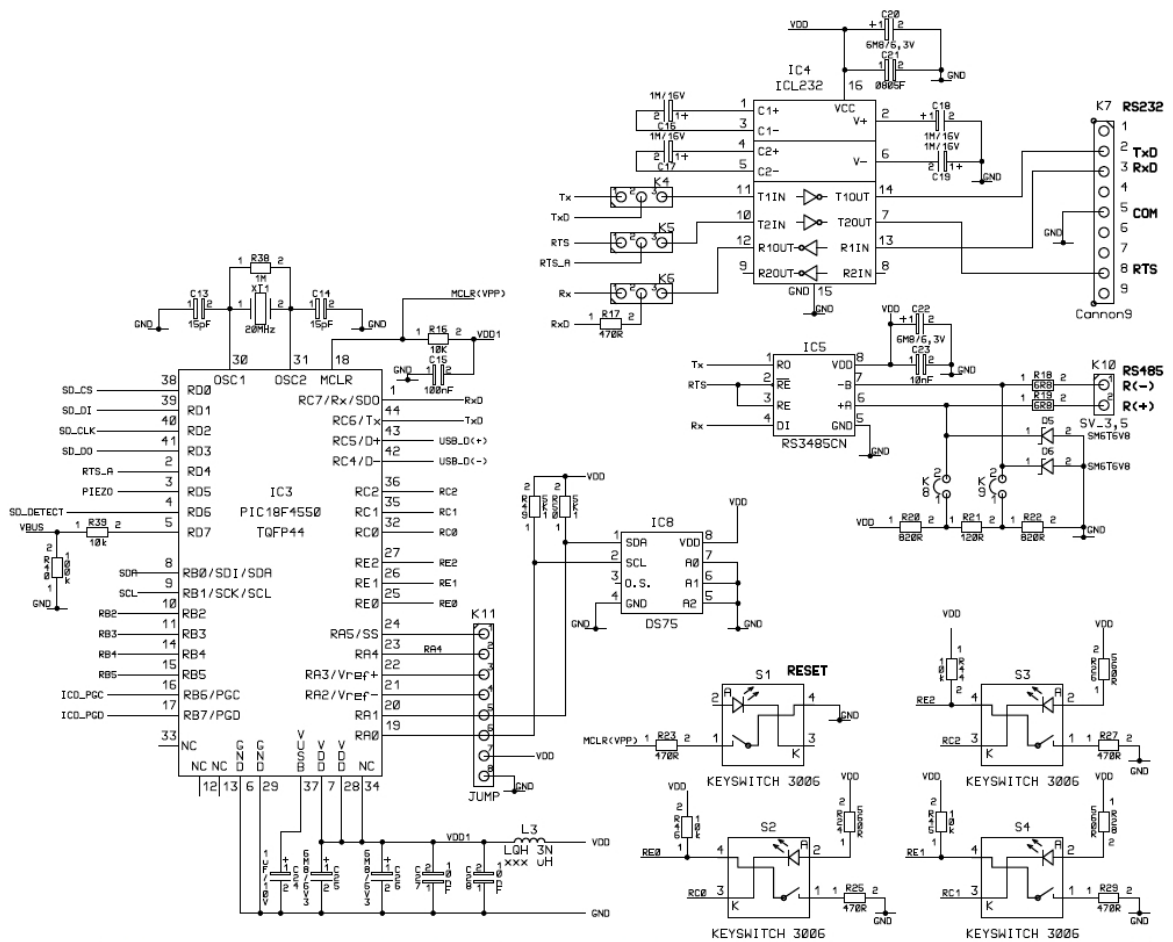
[24] MPASM™ Assembler, MPLINK™ Object Linker, MPLIB™ Object Librarian

- User's Guide <http://ww1.microchip.com/downloads/en/DeviceDoc/33014J.pdf>
- [25] PICmicro MCU C - An introduction to programming
The Microchip PIC in CCS C - By Nigel Gardner
http://old.disco.unimib.it/informaticaindustriale/PDF_Utli/eBook%20-%20PIC%20Programming%20with%20C.pdf
- [26] PICmicro® 18C MCU Family
Reference Manual <http://ww1.microchip.com/downloads/en/DeviceDoc/39500a.pdf>
- [27] PIC18F2455/2550/4455/4550 Data Sheet
<http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>
- [28] PIC18FXXX Hands On Workshop
http://www.microchip.com/conference/PIC18Fxxx_comprehensive_tutorial_containing_7Mb_of_info.pdf
- [29] PICDEM™ FS USB DEMONSTRATION BOARD USER'S GUIDE
<http://ww1.microchip.com/downloads/en/devicedoc/51526a.pdf>
- [30] DS1307, 64 x 8, Serial, I2C Real-Time Clock <http://datasheets.maxim-ic.com/en/ds/DS1307.pdf>
- [31] TEMP SENSORS - Design Guide
http://www.maxim-ic.com/design_guides/en/TEMP_SENSORS_13.pdf
- [32] DS75 Digital Thermometer and Thermostat <http://datasheets.maxim-ic.com/en/ds/DS75.pdf>
- [33] Šnorek, Miroslav. 1992. Standardní rozhraní PC. Praha: Grada a.s.
- [34] Device Using a PIC® MCU <http://ww1.microchip.com/downloads/en/AppNotes/01003a.pdf>

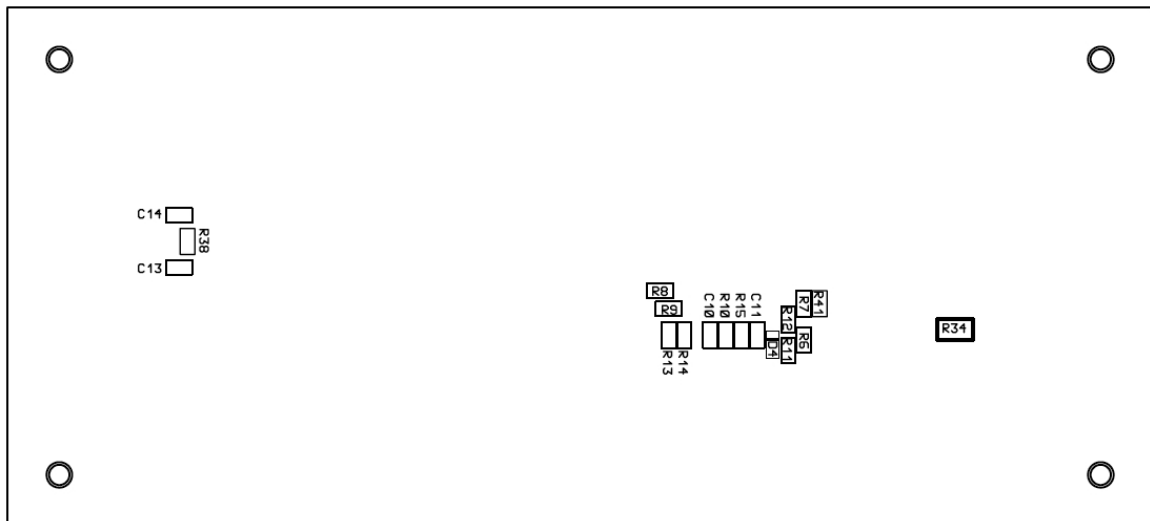
Internetové odkazy byly ověřeny ke dni 15. 6. 2008.

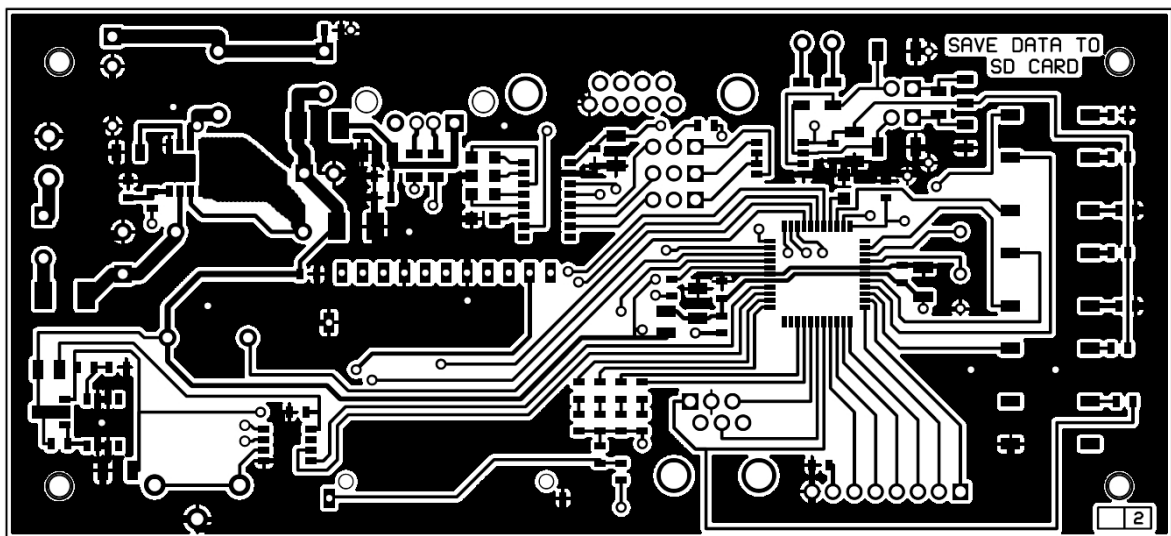
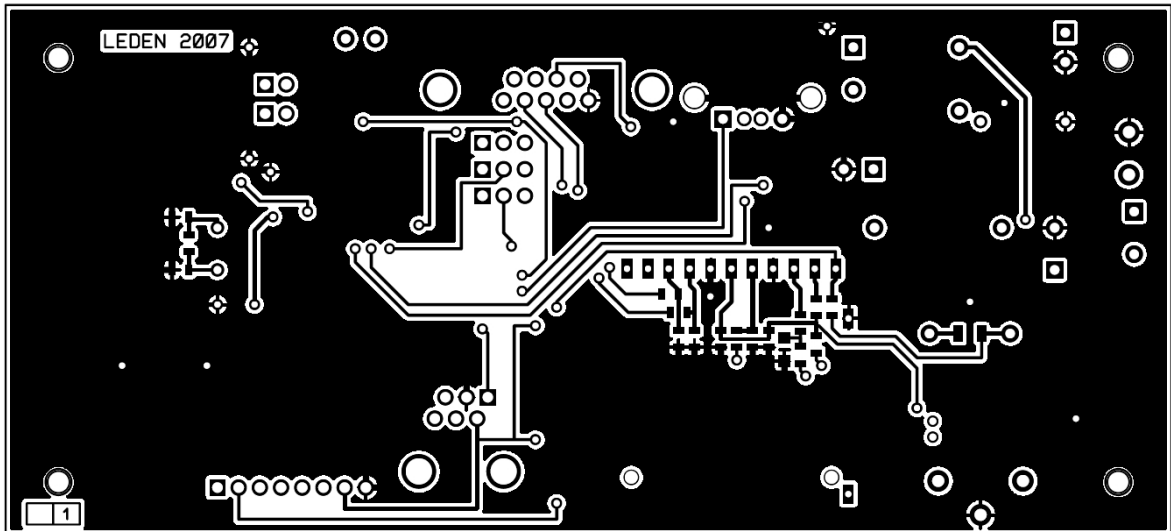
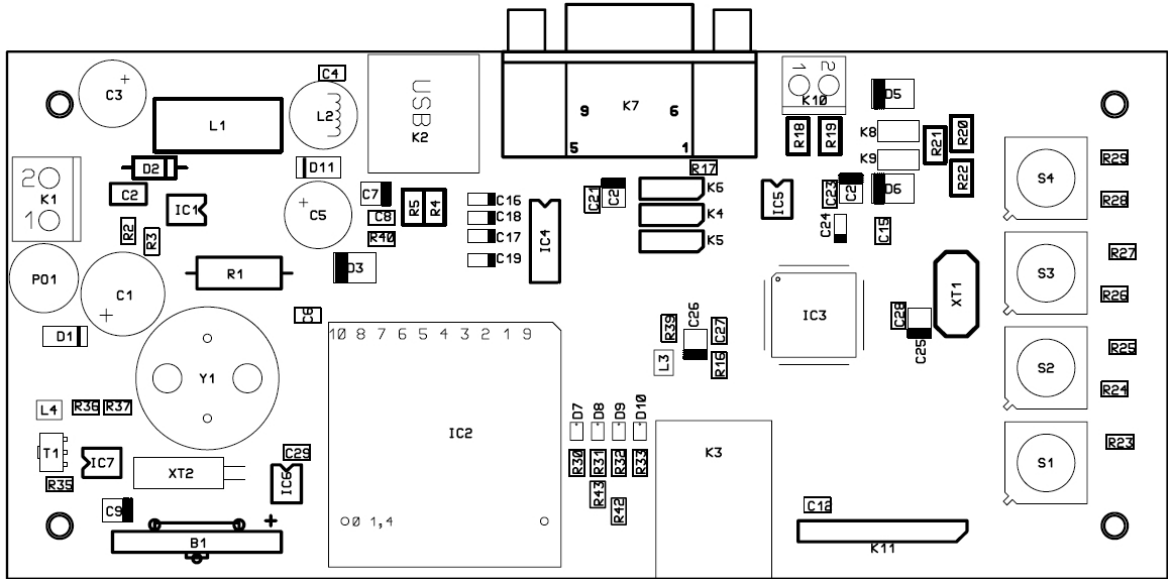
A Schémata





B Plošné spoje





C Rozpiska součástek

Označení	Specifikace	Poznámka
DPS	SD CARD 63,5x140	Plošný spoj dvouvrstvý
C 1	CE 470 uF/35V	
C 2	C0805 1 nF	SMD
C 3	CE 470 uF/16V	
C 4	C0805 10 nF	SMD
C 5	CE 470 uF/16V	
C 6	C0805 10 nF	SMD
C 7	CTS-B 6M8/6,3V B	SMD
C 8	C0805 10 nF	SMD
C 9	CTS-B 6M8/6,3V B	SMD
C 10	C0805 100 nF	SMD
C 11	C0805 100 nF	SMD
C 12	C0805 100 nF	SMD
C 13	C0805 15 nF	SMD
C 14	C0805 15 nF	SMD
C 15	C0805 100 nF	SMD
C 16	CTS-A 1 M/16V	SMD
C 17	CTS-A 1 M/16V	SMD
C 18	CTS-A 1 M/16V	SMD
C 19	CTS-A 1 M/16V	SMD
C 20	CTS-B 6M8/6,3V B	SMD
C 21	C0805 10 nF	SMD
C 22	CTS-B 6M8/6,3V B	SMD
C 23	C0805 10 nF	SMD
C 24	CTS-A 1 M/16V A	SMD
C 25	CTS-B 6M8/6,3V B	SMD
C 26	CTS-B 6M8/6,3V B	SMD
C 27	C0805 10 nF	SMD

C 28	C0805	10 nF		SMD
C 29	C0805	10 nF		SMD
R 1		0,33R		
R 2	R0805	1 K	1%	SMD
R 3	R0805	3 K	1%	SMD
R 4	R0805	0 R		SMD
R 5	R0805	0 R		SMD
R 6	R0805	47 K		SMD
R 7	R0805	47 K		SMD
R 8	R0805	1K8		SMD
R 9	R0805	1K8		SMD
R 10	R0805	1K8		SMD
R 11	R0805	1 K		SMD
R 12	R0805	1 K		SMD
R 13	R0805	3K3		SMD
R 14	R0805	3K3		SMD
R 15	R0805	3K3		SMD
R 16	R0805	10 K		SMD
R 17	R0805	470 R		SMD
R 18	R0805	6R8		SMD
R 19	R0805	6R8		SMD
R 20	R0805	820 R		SMD
R 21	R0805	120 R		SMD
R 22	R0805	820 R		SMD
R 23	R0805	470 R		SMD
R 24	R0805	560 R		SMD
R 25	R0805	470 R		SMD
R 26	R0805	560 R		SMD
R 27	R0805	470 R		SMD
R 28	R0805	560 R		SMD
R 29	R0805	470 R		SMD
R 30	R0805	1 K		SMD

R 31	R0805	1 K		SMD
R 32	R0805	1 K		SMD
R 33	R0805	1 K		SMD
R 34	R0805	4K7		SMD
R 35	R0805	100 R		SMD
R 36	R0805	1 K	1%	SMD
R 37	R0805	3K24	1%	SMD
R 38	R0805	1 M		SMD
R 39	R0805	10 K		SMD
R 40	R0805	100 K		SMD
R 41	R0805	47 K		SMD
R 42	R0805	1 K		SMD
R 43	R0805	47 K		SMD
R 44	R0805	10 K		SMD
R 45	R0805	10 K		SMD
R 46	R0805	10 K		SMD
R 47	R0805	1 K		SMD
R 48	R0805	1 K		SMD
R 49	R0805	5K1		SMD
R 50	R0805	5K1		SMD
D 1		4005		SMD (MELF)
D 2		1N5819		
D 3	transil	SM6T6V8		SMD
D 4	zenerova dioda	ZMM	3V3	SMD (MINIMELF)
D 5	transil	SM6T6V8		SMD
D 6	transil	SM6T6V8		SMD
D 7	LED			SMD
D 8	LED			SMD
D 9	LED			SMD

D 10	LED			SMD
D 11		4005		SMD
L 1		77050		
L 2	Fastron	1mH		
L 3	tlumivka	LQH3N4R7M04	4,7uH	SMD
L 4	tlumivka	LQH3N4R7M04	4,7uH	SMD
IC 1		MC34063AD		SO8
IC 2		SD CARD		slot
IC 3		PIC18F4550		TQFP44
IC 4		ICL232		SO16
IC 5		RS485CN		SO8
IC 6	RTC	DS1307		SO8
IC 7		TL431ACD		SO8
IC 8		DS75		SO8
PO 1	pojistka	MT780210	1 A/T	
XT 1	Krystal	HC-49	20 MHz	
XT 2	Krystal		32,786 kHz	
T 1	tranzistor	BCX55		SMD
S 1	keyswitch	3006		SMD
S 2	keyswitch	3006		SMD
S 3	keyswitch	3006		SMD
S 4	keyswitch	3006		SMD
Y 1	Piezo	PE 1540		
K 1	svorky			Rozteč 5 mm
K 2		USB-A		
K 3		WEBP 6-6 LP		
K 4	jump			
K 5	jump			
K 6	jump			
K 7	Cannon9	CAN 9 V 90		
K 8	jump			
K 9	jump			

K 10	svorky	Rozteč 3,5 mm
K 11	jump	

D Obsah CD

-  [DPS]
 -  [Kód]
 -  [Literatura]
 -  [Překladač]
 -  [WINHEX]
 -  Bakalářská práce
 -  Bakalářská práce
- doc
pdf