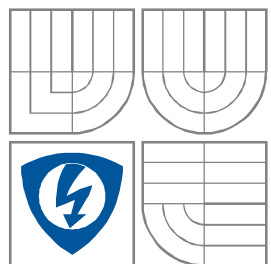


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

ZÁZNAMOVÉ ZAŘÍZENÍ PRO ŽELEZNIČNÍ PŘEJEZDY

MONITORING DEVICE FOR LEVEL CROSSINGS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

Tomáš Navrátil

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Dr. Ing. Zdeněk Kolka

BRNO, 2008

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Tomáš Navrátil
Bytem: Popovice 1, 664 61
Narozen/a (datum a místo): 1. listopadu 1977 v Brně

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 53, Brno, 602 00
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Dr. Ing. Zbyněk Raida, předseda rady oboru Elektronika a sdělovací technika
(dále jen „nabyvatel“)

Čl. 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☐ diplomová práce
- ☒ bakalářská práce
- ☐ jiná práce, jejíž druh je specifikován jako
(dále jen VŠKP nebo dílo)

Název VŠKP: Záznamové zařízení pro železniční přejezdy

Vedoucí/ školitel VŠKP: doc. Dr. Ing. Zdeněk Kolka

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- ☒ v tištěné formě – počet exemplářů: 2
- ☒ v elektronické formě – počet exemplářů: 2

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

* hodící se zaškrtněte

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☐ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☒ 10 let po uzavření této smlouvy
(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 6. června 2008

.....
Nabyvatel

.....
Autor

Abstrakt

Bakalářská práce (KBCE) se zabývá návrhem záznamového zařízení, které bude použito zejména na přejezdových zabezpečovacích zařízeních. Záznamové zařízení umožňuje sledování digitálních veličin pomocí základní desky s mikroprocesorem ATMEGA 128. Signály jsou vzorkovány každou milisekundu. Zařízení je vybaveno rozhraním USB, rozhraním RS 485, ethernetem a také SD kartou. Kapacita karty je navržena na 40 dní.

Pro otestování základních funkcí byla použita základní vývojová deska Ethernut, ke které byly připojeny další potřebné obvody. Záznamy se ukládají na vyjímatelnou paměťovou kartu SD ve formátu FAT 16. Po naplnění karty se nejstarší záznamy přepisují.

Projekt obsahuje popis komunikace pomocí rozhraní USB, dále technický návrh funkčního rozhraní USB, včetně fotografie hardwaru. Bakalářská práce popisuje vlastnosti SD karty, způsob komunikace a popis souborového systému FAT 16. Je zde také uveden návrh technického řešení připojení SD karty k mikroprocesoru přes rozhraní SPI.

Veškerý software je napsán v programovacím jazyce C. V projektu jsou popsány hlavní funkce programu a jejich parametry.

Klíčová slova

mikroprocesorová technika, vestavěné systémy, zabezpečovací zařízení

Abstract

Bachelor's thesis deals with a design of a safety appliance. The device will be used particularly at grade crossing. The record device allows monitoring of digital signals with the basic board and with microprocessor ATMEGA 128. Signals are patterned each second. The device is equipped with a USB interface, RS 485, ethernet and SD card. The capacity of SD card is designed for 40 days.

The basic development board Ethernut, to which additional circuits were connected, was used for testing basic functions. Records are saved on a removable memory SD card in FAT 16 format. Old records are overwritten when the card is full.

The project includes a description of communication with USB interface, a technical design of USB interface with photography of the hardware. Bachelor's thesis characterizes features of the SD card, communication and a description of the file system FAT 16. I also introduce a design of connecting SD card to the microprocessor within the SPI interface.

All software was written in programming language C. There are described main functions of the code and their parameters.

Key words

microprocessor technology, embedded systems, safety appliance

Bibliografická citace

NAVRÁTIL, T. *Záznamové zařízení pro železniční přejezdy: bakalářská práce*. Brno: FEKT VUT v Brně, 2008. 58 s., 6 příl.

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Záznamové zařízení pro železniční přejezdy jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne 6. června 2008

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Dr. Ing. Zdeňkovi Kolkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 6. června 2008

.....
podpis autora

Obsah

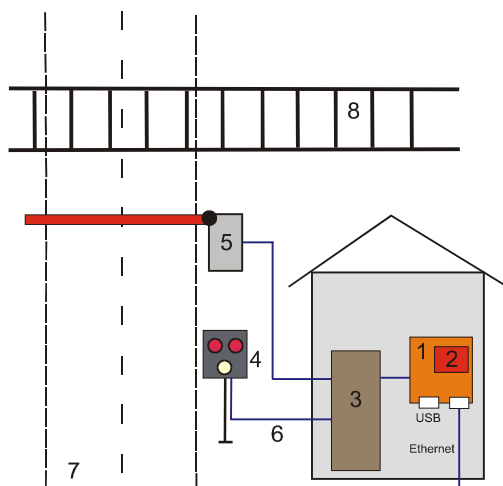
1. Úvod.....	9
2. Řešení částí záznamového zařízení	11
2.1 Ethernet	11
2.2 Rozhraní USB	13
2.3 Paměťová SD karta	16
2.4 Rozhraní RS 485	24
2.5 Modul optického oddělení vstupů	26
3. Softwarové vybavení záznamového zařízení.....	26
3.1 Vývoj a ladění programu	26
3.2 Testování základních softwarových funkcí Nut/OS.....	27
3.3 Popis zdrojového kódu programu a struktury vláken.....	28
3.4 Rychlé vyhledávání v záznamu SD karty.....	34
4. Záznam dat a měření	35
4.1 Nastavení připojení mezi PC a FTP serverem.....	35
4.2 Simulace signálů na vstupu	37
4.3 Záznam vstupních signálů a test přesnosti záznamu	39
4.4 Měření časové náročnosti zpracování jednoho vzorku.....	42
5. Závěr.....	44
6. Literatura.....	45
7. Seznam zkrátek, symbolů a příloh.....	46

8. Přílohy	47
Příloha č.1 Příkazy SD karty v SPI módu.....	47
Příloha č.2 Formáty odpovědí SD karty.....	48
Příloha č.3 Popis FAT Boot Recordu.....	49
Příloha č.4 Doplnění schéma zapojení záznamového zařízení	50
Příloha č.5 Vývojové diagramy hlavního programu.....	51
Příloha č.6 Vývojové diagramy funkcí	55

1. Úvod

Bakalářská práce se zabývá návrhem záznamového zařízení řízeného jednočipovým mikroprocesorem. Pro tuto aplikaci plně postačuje architektura AVR, 8-bitový mikroprocesor ATMEGA 128. Výhody jsou jednoduchost, dostupnost, softwarové vybavení volně stažitelné z internetu.

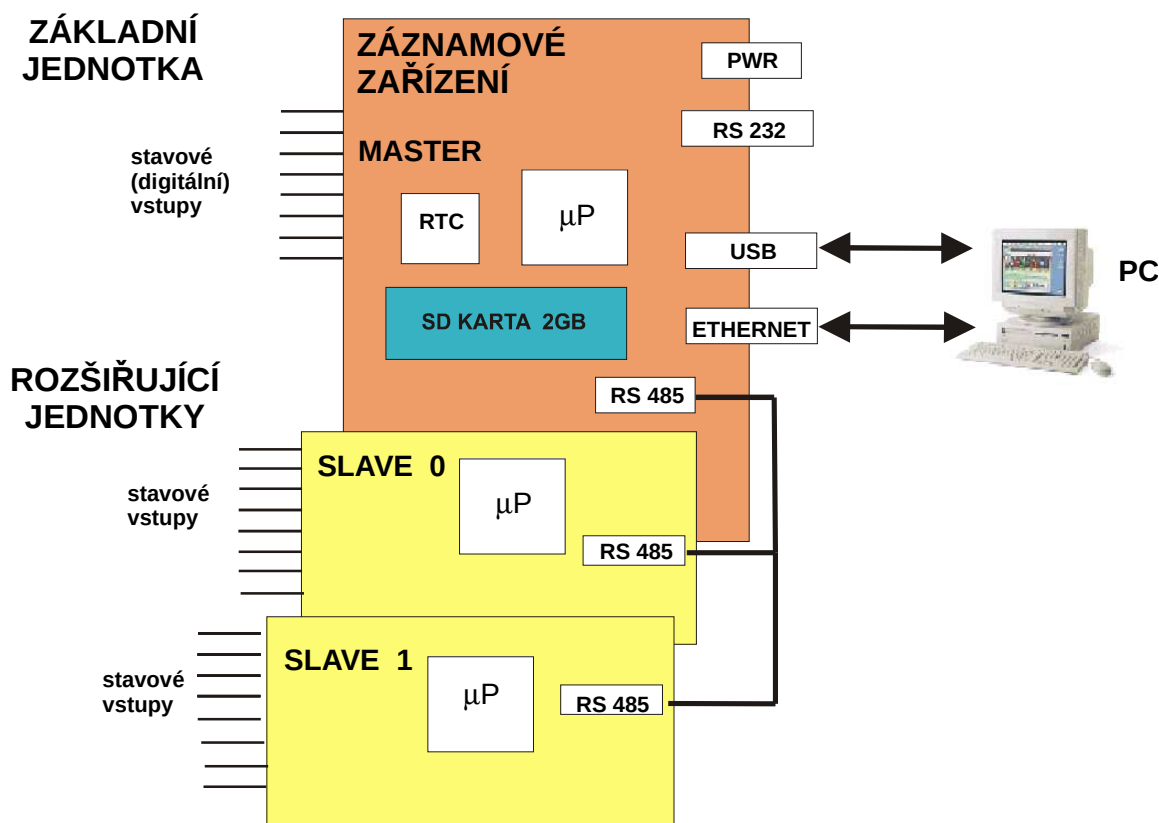
Jedná se o diagnostické zařízení, které bude implementováno zejména na přejezdová zabezpečovací zařízení, viz obr.1.



Obr. 1 Železniční přejezd

1 – záznamové zařízení, 2 – SD karta, 3 – zabezpečovací zařízení železničního přejezdu, 4 – semafor, 5 – závora, 6 – vedení signálů, 7 – silnice, 8 – koleje

Cílem je umožnit sledování digitálních veličin (především stavy relé) pomocí základní desky s mikroprocesorem. Signály jsou vzorkovány každou milisekundu. Záznamové zařízení má být vybaveno rozhraním USB (pro komunikaci s PC a také pro vyčítání uložených dat), dále rozhraním RS 485 (bude sloužit ke komunikaci s ostatními rozšiřujícími deskami, nebo s dalším zařízením) a ethernetem (podobná funkce jako USB). Blokové schéma záznamového zařízení je na obr. 2.



Obr. 2 Blokové schéma záznamového zařízení

Z časových a ekonomických důvodů bylo nejvýhodnější koupit základní vývojovou desku Ethernet (verze 2.1). K této desce jsem připojil ostatní obvody, viz obr.4. Základní deska je vybavena rozhraním RS 485 a ethernetem. Záznamy se ukládají na vyjímatelnou paměťovou kartu SD ve formátu FAT 16. Kapacita této karty je stanovena na 40 dní provozu na přejezdu. Po naplnění karty se nejstarší záznamy přepisují.

Bakalářská práce obsahuje popis komunikace pomocí rozhraní USB, RS 485 a také návrh technického řešení rozhraní USB. Dále jsou v práci popsány vlastnosti SD karty, způsob komunikace a popis souborového systému FAT 16. Následuje návrh technického řešení připojení SD karty k mikroprocesoru přes rozhraní SPI. Součástí práce je i popis základní vývojové desky.

Software jsem psal v programovacím jazyce C a bližší popis jeho jednotlivých částí je také součástí této práce.

Shrnutí požadavků na hardware záznamového zařízení

- architektura AVR
- zařízení bude obsahovat rozhraní USB, RS 485, Ethernet
- sledování digitálních veličin
- ukládání záznamů na vyjímatelnou kartu SD
- kapacita SD karty 40 dní

Shrnutí požadavků na software záznamového zařízení

- vzorkování vstupů každou 1 ms
- záznamy na kartě ve formátu FAT

2. Řešení částí záznamového zařízení

Navrhnul jsem jednotlivé části záznamového zařízení pro sledování 8 digitálních veličin (u první verze pouze 4 vstupy). Byl navržen také modul optického oddělení relé od vstupů procesoru, viz obr.4. Pro otestování funkce záznamového zařízení jsem signály z relé simuloval pomocí PC. Zařízení je řízeno jednočipovým mikroprocesorem ATMEGA 128, umožňující zápis a čtení dat z paměťové karty. Zařízení obsahuje rozhraní ethernet a rozhraní RS 485 pro připojení budoucích rozšiřujících karet. Byl zpracován návrh komunikačního protokolu přes rozhraní RS 485. V dalším textu uvádím návrhy, realizace a testování jednotlivých částí zařízení.

2.1 Ethernut

Projekt Ethernut implementuje moderní internetové technologie do embedded zařízení. Podrobnější informace jsou v [1]. Systém umožňuje současné zpracování více úloh, dynamickou správu paměti, implementaci standardních síťových protokolů a služeb. Zdrojové kódy jsou napsané v jazyku C, což zpřehledňuje celé řešení. Hlavní části operačního systému jsou Nut/OS a Nut/Net.

Nut/OS (Real-Time Operační systém)

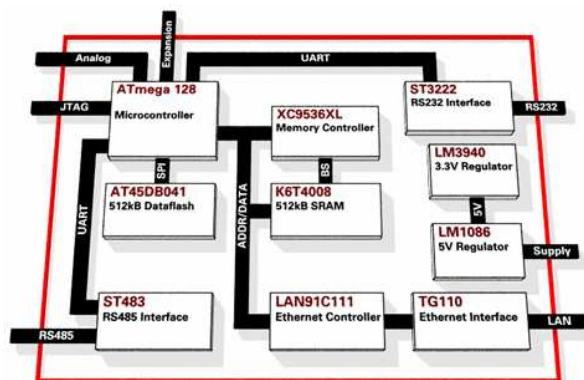
Nut/OS je napsán v jazyku C pro překladač AVR GNU C a ImageCraft AVR. Vše co potřebuje ke své činnosti, je součástí jeho zdrojových kódů. Jedná se o víceúlohový operační systém umožňující spouštět vlákna a přidělovat jim čas na zpracování. Také je zde integrován systém řízení priorit. Jednotlivé úlohy lze dynamicky vytvářet a ukončovat za běhu programu.

Nut/Net

Nut/Net je nadstavbou Nut/OS a představuje implementaci TCP/IP protokolového zásobníku v rámci projektu Ethernut. Na rozdíl od osobních počítačů jsme zde limitováni dostupnou výpočetní a paměťovou kapacitou. Nut/Net podporuje práci s TCP sokety.

2.1.1 Základní vývojová deska

Při řešení dané problematiky jsem použil základní vývojovou desku, založenou na platformě AVR Ethernut. Blokové schéma je na obr. 3, převzato z [1]. Na desce je k mikroprocesoru připojena vnější paměť, dále je zde ethernetový obvod LAN91C111, ethernetový interface TG 110, RS 485 interface, napájecí obvody a RS 232 interface. Rozhraní RS232 jsem využil při některých testech na ladění softwaru. Podrobný popis této desky Ethernut 2 je v [2].

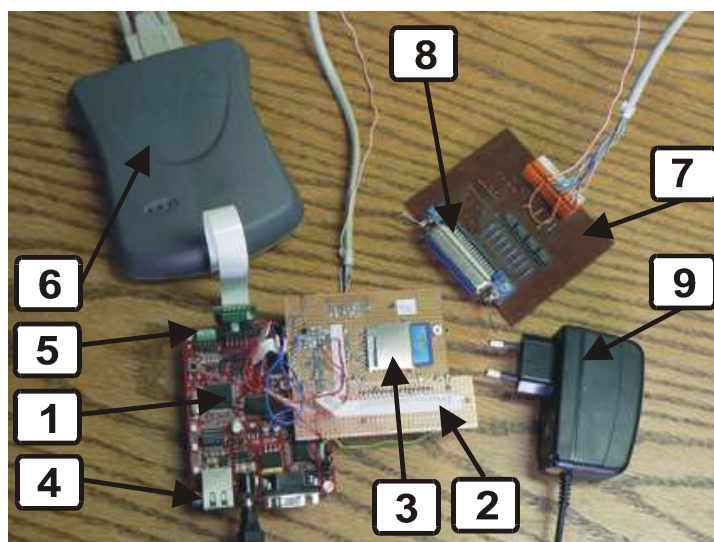


Obr. 3 Blokové schéma Ethernet 2

Procesor ATMEGA 128 [3] má sériové rozhraní SPI (Serial Peripheral Interface), které jsem využil k připojení SD karty v režimu SPI.

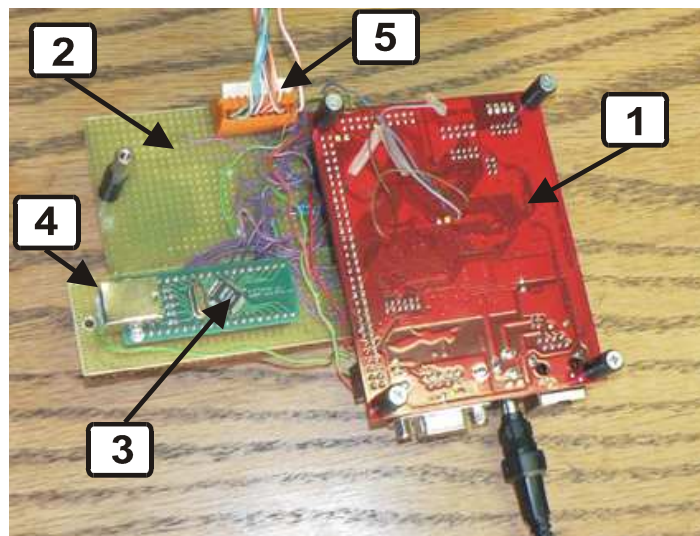
2.1.2 Zapojení a testování základní desky

Pohled na základní vývojovou desku je na obr. 4. Schéma zapojení základové desky (převzato z [1]) je k dispozici na CD ve složce „SCHÉMA“. K vývojové desce jsem připojil rozšiřující desku s univerzálním plošným spojem (viz obr.4, pozice 2) s modulem obvodu FT2232C (viz obr.5, pozice 3) a patičí na SD kartu (viz obr.4, pozice 3). Rozhraní USB, RS 485 a ethernet jsou také vidět na obr. 4 a 5. Zdroj 9V a JTAGICE programátor jsou na obr.4, pozice 6 a 4.



Obr. 4 Základní vývojová deska s periferiemi

1 – testovací deska Ethernet, 2 – univerzální plošný spoj s obvodem FT2232C a SD kartou, 3 – SD karta, 4 – rozhraní Ethernet, 5 – rozhraní RS 485, 6 – programátor JTAGICE, 7 – modul optického oddělení vstupů, 8 – konektor připojení PC, 9 – zdroj napájení



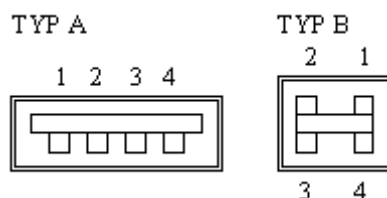
Obr. 5 Základní vývojová deska s periferiemi – pohled zespodu
 1 – testovací deska Ethernet, 2 – plošný spoj s obvodem FT2232C a SD kartou,
 3 – obvod FT2232C, 4 – rozhraní USB, 5 – konektor připojení modulu opt. oddělení vstupů

2.2 Rozhraní USB

USB je univerzální rozhraní pro sériový přenos dat [5]. Bylo vytvořeno s cílem jednoduché obsluhy periférií připojených k počítači. USB sběrnice podporuje připojování a odpojování zařízení za chodu počítače, včetně zavedení příslušných ovladačů do operačního systému. K připojení zařízení pomocí USB se používá čtyř vodičový stíněný kabel zakončený konektory typu A nebo B viz obr.6. Popis vývodů je v tabulce č.1. Maximální délka kabelu je 5 m. Konektor typu A je v počítači (hostiteli) a konektor typu B je v periferním zařízení.

Přenosové rychlosti rozhraní USB:

- Low Speed 1.5 Mb/s
- Full Speed 12 Mb/s
- High Speed 480 Mb/s



Obr. 6 USB konektory typu A a B

Tab.1 Popis vývodů USB konektoru

Číslo vývodu	Význam
1	+ 5 V (Ucc)
2	Data + (přímá data)
3	Data - (negovaná data)
4	GND

USB je jednomasterová sběrnice, všechny aktivity vycházejí z počítače. Data se vysílají v paketech o délce 8 až 64 bajtů. Připojené zařízení nemůže začít vysílat data bez vyzvání. Počítač vyšle token paket obsahující popis typu a směru přenosu dat a adresu USB zařízení. To pošle požadovaná data nebo paket s informací, že data nejsou. Nakonec se vyšle handshake paket, který informuje, jestli přenos proběhl úspěšně.

Zařízení může být napájeno přímo pomocí USB sběrnice, pokud je maximální odebíraný proud 100 mA (low power), nebo 500 mA (high power).

2.2.1 Typy přenosů na USB

Řídící přenos (Control transfer) slouží pro inicializaci a ovládání zařízení [5]. Má největší prioritu a automatické zabezpečení chyb. Maximální velikost paketu je 64 bytů.

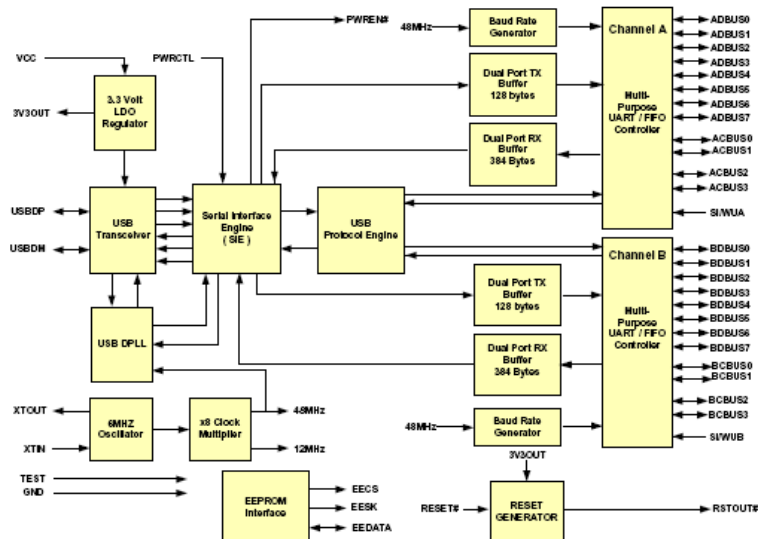
Přenos přes přerušení (Interrupt transfer) používají zařízení, která pravidelně posílají malé množství dat. Protože nemohou data vysílat bez vyzvání, počítač žádá o data v určitém časovém intervalu. Velikost paketu je 8 bytů. Používá se např. u počítačových myší a klávesnic připojených přes USB.

Hromadný přenos (Bulk transfer) slouží pro přenos velkého množství dat, u kterých nezáleží na rychlosti přenosu. Je důležitá bezchybnost, používá se kontrola dat. V praxi použití např. u externích USB disků, tiskáren atd.

Izochronní přenos (Isochronous transfer) je určen pro velké množství dat při konstantním datovém toku. Neprovádí se kontrola dat. Používá se např. u vnějších zvukových karet a web kamer.

2.2.2 Realizace rozhraní USB s obvodem FT2232C

Pro připojení zařízení k počítači přes USB rozhraní použijí obvod FT2232C. Obvod obsahuje dva nezávislé, individuálně konfigurovatelné kanály A a B, které mohou fungovat nezávisle, nebo společně. Funkce každého komunikačního kanálu je uživatelsky nastavitelná pomocí konfiguračních dat umístěných v externí paměti EEPROM. Blokové schéma je na obr.7.



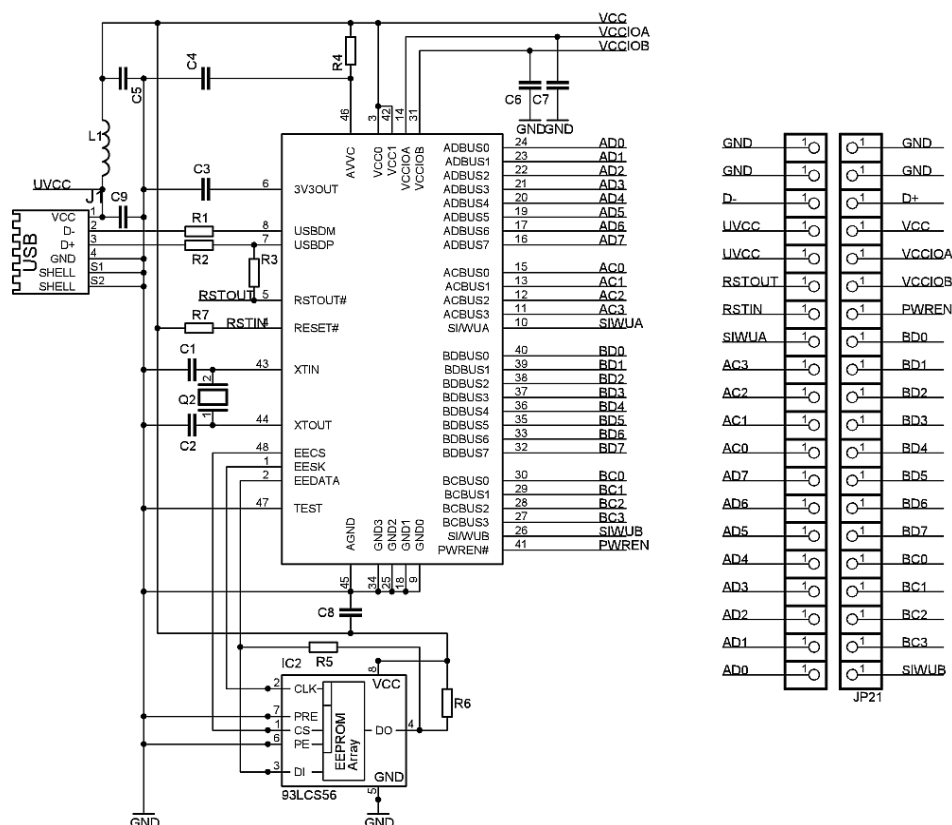
Obr. 7 Blokové schéma obvodu FT2232C

K dispozici je sériové rozhraní (kanál FT2232C se chová jako obvod FT232BM, podporuje RS-232, RS-422 a RS-485, dosahuje rychlosti až 3 Mbit/s), nebo paralelní rozhraní (kanál se chová jako součástka FT245BM, přenosová rychlost až 1 Mbyte/s) a asynchronní režim Bit-Bang (8 vývodů funguje jako 8-bitový paralelní port s možností individuálního nastavení vstup/výstup pro každý pin a volby kmitočtu pro obnovení stavu portu).

Další možnost použití je Multi-Protocol Synchronous Serial Engine Interface (MPSSE), který umožňuje implementaci různých synchronních sériových protokolů jako např. JTAG či SPI. Může být též s výhodou využito pro konfigurování programovatelných hradlových polí (FPGA) založených na technologii pamětí SRAM, jako např. Xilinx a Altera. Kombinací FT2232C a FPGA tak vzniká flexibilní systém, jehož funkce může být neomezeně modifikována či upgradována. MPSSE je k dispozici pouze na kanálu A. Rychlost toku dat může být až 5,6 Mbit/s.

Pro moji aplikaci jsem zvolil paralelní režim (pro konverzi mezi USB a paralelní 8 bitovou vstupně výstupní branou) z důvodu rychlejší komunikace. Pro USB aplikaci využiji kanál B, kanál A zůstane volný pro další verzi desky – komunikace s programovatelným polem.

Obvod 2232C je zapojený podle doporučení výrobce, viz [6]. Použil jsem již hotový modul s připojenou pamětí EEPROM, viz obr.8. Označení vývodů tohoto modulu se 40-pinovým konektorem je v pravé části obrázku. Pomocí patice je tento modul připojen k desce záznamového zařízení, viz obrázek č. 5, pozice 3. Na celkovém schématu v příloze č.4 je tato patice označena jako KON2.



Obr. 8 Zapojení modulu s obvodem FT2232C

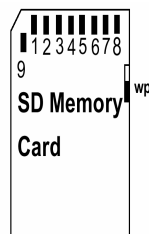
K mikroprocesoru je obvod připojen, jak již bylo řečeno, přes kanál „B“. Data (D0 až D7) jsou v nastaveném paralelním režimu připojena na vývody obvodu označené jako BD0 až BD7. Řídící signály RXF, TXE, RD, WR jsou pak v tomto režimu označeny jako BC0 až BC3

2.2.3 Testování rozhraní USB

Obvod FT2232C bylo potřeba nejprve nakonfigurovat prostřednictvím programu MProg. Tento software byl stažen od výrobce obvodu z [7]. Bližší popis postupu konfigurace viz [8]. Pro otestování správné funkce rozhraní USB jsem použil program popsáný v kapitole 5.2.2.

2.3 Paměťová SD karta

SD karta (Secure Digital) je nástupce karty MMC, kterou vyvinuly v roce 1999 firmy Sandisk, Matsushita a Toshiba [9]. Kmitočet hodinových impulsů je max. 25 MHz. Nejkratší doba trvání log. 1 a log. 0 je 10 ns a nejdelší doba trvání nástupné a sestupné hrany je 10 ns. Rozmístění vývodů karty je na obr. 9 a popis vývodů SD karty je v tab. 2.



Obr. 9 Rozmístění vývodů SD karty

Tab.2 Popis vývodů SD karty

Číslo vývodu	Industrial Grade SD Mód	SPI Mód
1	Card Detect / Data Line (Bit 3)	Chip Select
2	Command / Response	Host to Card Commands and Data
3	Supply voltage ground (U_{SS})	Supply Voltage Ground (U_{SS})
4	Supply voltage (U_{DD})	Supply Voltage (U_{DD})
5	Clock	Clock
6	Supply voltage ground (U_{SS})	Supply Voltage Ground (U_{SS})
7	Data Line (Bit 0)	Card to Host Data and Status
8	Data Line (Bit 1)	Reserved
9	Data Line (Bit 2)	Reserved

SD karta umožňuje několik módů komunikace:

- 1-bitový SD mód kompatibilní s MMC kartami
- 4-bitový SD mód
- SPI mód

V SD módech jsou příkazy kartě a její odpovědi mikroprocesoru vedeny po jednom vodiči (CMD) a bloky dat pro čtení nebo zápis po jednom (1-bitový mód) nebo čtyřech (4-bitový mód) datových vodičích (DAT0-3). V SPI módu jsou data do karty přiváděna jedním a data z karty do mikroprocesoru druhým vodičem.

Pro komunikaci s SD kartou volím SPI mód, protože použitý mikroprocesor podporuje komunikaci po SPI sběrnici. Zatímco SD módy jsou založené na příkazech a bitovém toku začínající start bitem a končící stop bitem, v SPI módu je kanál bytově orientovaný. Všechny příkazy a bloky dat jsou uspořádány po 8-bitech v bytu. V SD i SPI módu je podobný komunikační protokol založený na příkazech, odpovědích a datových blocích.

SPI módu SD karta odpoví na každý příkaz a je zde možné nastavit tři různé druhy odpovědi. Jestliže karta zjistí chybu dat, ihned odpoví chybovým hlášením (toto hlášení nahradí očekávaná data), zatímco v SD módu dojde k časové prodlevě.

2.3.1 Formát příkazů v SPI módu

Každý příkaz se skládá z 6-ti bytů. První byt označuje příkaz, první dva bity určují začátek příkazu, následuje šest bitů s číslem příkazu. V následujících čtyřech bytech je parametr příkazu, např. adresa bloku dat. Poslední byt slouží pro kontrolu chyb CRC, poslední bit v tomto bytu je vždy jedna, tzv. stop bit. Formát příkazu je znázorněn v tab. 3.

Tab.3 Formát příkazů v SPI módu

Byt 1			Byt 2			Byt 3			Byt 4			Byt 5			Byt 6		
7	654321	0	7	654321	0	7	654321	0	7	654321	0	7	65431	0	7	654321	0
0	1	Příkaz	Parametr příkazu												CRC		1

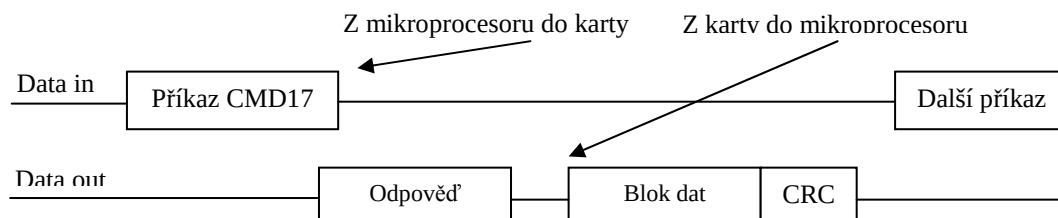
2.3.2 Výběr módu

SD karta je po připojení napájení v SD módu. Pro použití karty v SPI módu je nutné v průběhu odpovědi na výběr karty poslat resetovací příkaz GO_IDLE_STATE (CMD0). Po přepnutí karty do SPI módu se odešle odpověď v základním formátu R1. Pokud tato odpověď nepřijde, karta zůstala v SD módu. Pro úplné inicializování SPI módu se musí ještě poslat příkaz SEND_OP_COND (CMD1). Karta odpovědí nula oznámí, že dokončila inicializaci.

2.3.3 Čtení dat

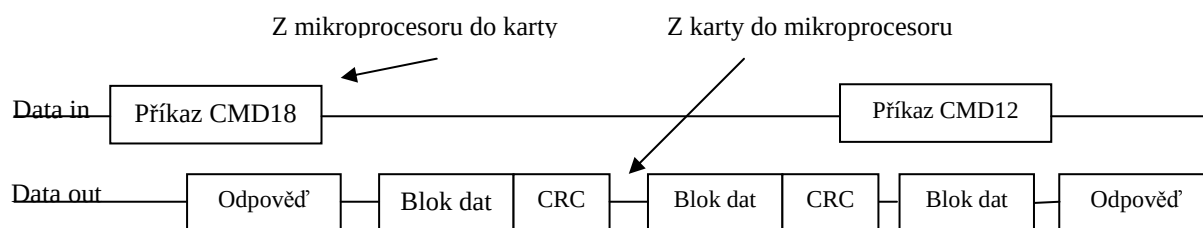
SPI mód umožňuje čtení jednoho nebo více bloků dat. Maximální velikost bloku READ_BLOCK_LEN je definována v registru CSD a je nastavena na 512 bytů. Pokud je nutné číst menší bloky dat, musí se nastavit READ_BLOCK_PARTIAL do jedničky a pomocí příkazu CMD16 SET_BLOCKLEN nastavit délku přenášeného bloku.

Pro čtení jednoho bloku se používá příkaz CMD17 READ_SINGLE_BLOCK. Princip čtení jednoho bloku dat je zobrazen na obr. 10. Po zadání tohoto příkazu karta potvrdí příkaz odpovědí a pošle blok dat zakončený 16-ti bity CRC cyklického kódu zabezpečení dat.



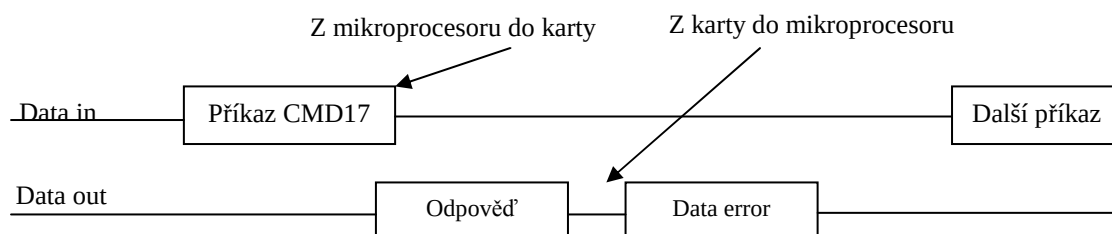
Obr. 10 Princip čtení jednoho bloku dat

Na obr. 11 je zobrazen princip čtení více bloků, pro který se používá příkaz CMD18 READ_MULTIPLE_BLOCK. Po vyslání tohoto příkazu karta potvrdí přijetí odpovědi a začne vysílat bloky dat. Počet bloků není definován. Karta data vysílá, dokud neobdrží příkaz pro konec přenosu CMD12 STOP_TRANSMISSION.



Obr. 11 Princip čtení více bloků

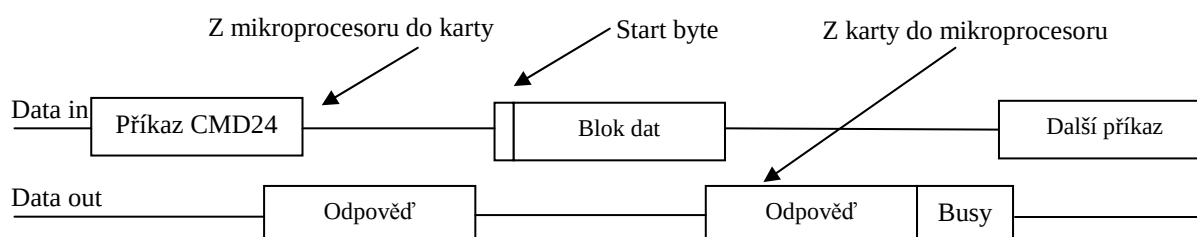
Pokud se v průběhu vysílání dat vyskytne chyba, karta nepokračuje ve vysílání dat. Mikroprocesor přijme chybový byte, viz obr.12. Když nastane chyba při čtení více bloků dat, karta přepíše libovolnou sekvenci dat chybovým hlášením. Mikroprocesor pak ukončí přenos příkazem konec přenosu CMD12 STOP_TRANSMISSION.



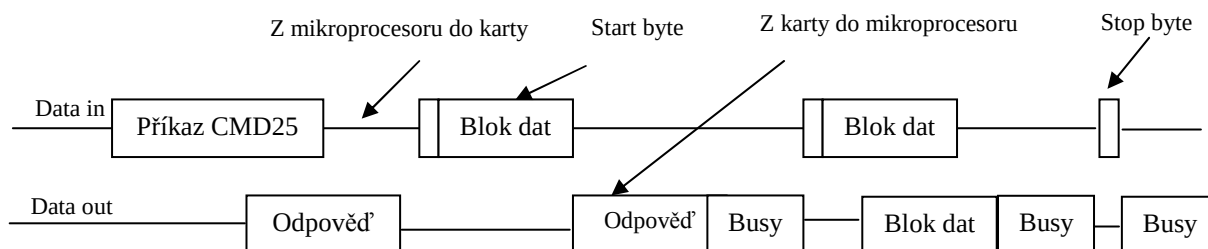
Obr. 12 Popis chyby při čtení jednoho bloku dat

2.3.4 Zápis dat

V SPI módu karta umožňuje zápis jednoho nebo více bloků dat, obr. 13 a 14. Pomocí příkazu CMD24 WRITE_BLOCK zápis jednoho bloku a pomocí CMD25 WRITE_MULTIPLE_BLOCK zápis více bloků dat. Po vyslání příkazu karta potvrdí přijetí příkazu odpovědí a čeká na data, která má zapsat. Přidání CRC, adresa bloku je stejná jako u čtení dat. Data začínají start bytem a po jejich vyslání karta pošle odpověď o příchodu dat a signál busy, během kterého zapíše data. Pokud karta zjistí chybu CRC, nezapíše data a informaci o tom pošle v odpovědi o příchodu dat. Při zápisu více bloků není počet bloků definován. Karta přijímá data, dokud se nepošle stop byte. Když je zápis dokončen, lze pomocí příkazu CMD13 SEND_STATUS zkontrolovat průběh zápisu.



Obr. 13 Zápis jednoho bloku dat



Obr. 14 Zápis více bloků

2.3.5 Registry SD karty v SPI módu

a/ Operation Conditions Register (OCR) 4-byty

V tomto registru je uložena hodnota napájecího napětí karty. Je v něm také stavový bit, který se nastaví po skončení inicializační procedury po připojení karty k napájecímu napětí.

b/ Card Identification Register (CID) 16-bytů

Tento registr obsahuje jedinečné identifikační číslo, které je nastaveno během výroby a nedá se měnit.

c/ Card Specific Data Register (CSD) 16 bytů

V tomto registru jsou uloženy informace pro přístup k datům na kartě, např. počet bloků pro čtení nebo zápis více bloků dat. Podrobný popis registrů je uveden v [9].

2.3.6 Příkazy pro SD kartu v SPI módu

Příkazy pro komunikaci s SD kartou v SPI módu a formáty odpovědí viz příloha č.1 a 2.

2.3.7 Souborový systém FAT

U souborového systému FAT, viz [10] je paměť rozdělena na sektory, bloky paměti o velikosti 512 bytů. Dále se paměť dělí na určité celky, o různém počtu sektorů, jak je vidět v tab. 4.

První část paměti, pouze u paměti s větší kapacitou, je vyhrazena pro Master Boot Record. Následuje FAT Boot Record, FAT tabulka a tabulka kořenového adresáře (tabulka je pouze u souborových systémů FAT 12 a FAT 16). U souborového systému FAT 32 jsou položky (soubory a adresáře) kořenového adresáře uloženy v oblasti dat, viz [11]. Na konci je oblast paměti, ve které jsou uložena data.

Tab.4 Způsob rozdělení paměti při použití FAT

Master Boot Record
FAT Boot Record
FAT tabulka
Tabulka kořenového adresáře
Data

▪ Master Boot Record

Neobsahuje příliš mnoho informací o paměťovém zařízení a jeho popis je v tab. 5. Nachází se v prvním sektoru paměti. Jeho hlavním účelem je zavedení operačního systému počítače. Obsahuje partition table, tabulky rozdělení (jsou čtyři) a jejich bližší popis je v tab. 6

Tab.5 Popis Master Boot Recordu

offset	Délka (bytů)	Popis
0x0	446	Consistency check routine (zaváděcí kód)
0x1be	16	1 Blok tabulky partition
0x1ce	16	2 Blok tabulky partition
0x1de	16	3 Blok tabulky partition
0x1ee	16	4 Blok tabulky partition
0x1fe	2	Kontrolní signatura (0x55aa)

Tab.6 Popis jednotlivých bloků tabulky partition

offset	Délka (bytů)	Popis	Hodnota/Rozsah
0x0	1	Určuje ze které oblasti se zavádí systém	0x00 – neaktivní 0x80 – aktivní
0x1	3	První sektor partition	Adresa prvního sektoru partition
0x4	1	Popis souborového systému	1 = DOS FAT12 4 = DOS FAT16 < 32 MB 5 = Extended DOS 6 = DOS FAT16 > 32 MB
0x5	3	Poslední sektor partition	Adresa posledního sektoru partition
0x8	4	Informace o umístění začátku disku	Číslo prvního sektoru
0xc	4	Počet sektorů partition	Počet sektorů které partition zabírá

- FAT Boot Record

U pamětí s větší kapacitou je uložen v sektoru určeném Master Boot Recordem. Ve FAT Boot Recordu jsou uloženy nejdůležitější informace o souborovém systému. Popis FAT Boot Recordu je v příloze 3.

- FAT tabulka

U souborového systému FAT je oblast paměti dat rozdělena na tzv. clustery. Každý cluster obsahuje několik sektorů, počet je určen ve FAT Boot Recordu. Clustery mají v paměti své fyzické umístění a s tím souvisí obsah FAT tabulky. Každá buňka v tabulce FAT odpovídá umístění příslušného clusteru. Počet buněk odpovídá počtu clusterů. Velikost buňky závisí na typu FAT, např. u FAT 16 má buňka FAT velikost 16 bitů. Oblast paměti, ve které je uložena tabulka FAT, začíná na sektoru za FAT Boot Recordem. Její konec je určen počtem sektorů, které tabulka FAT zabírá. Tato hodnota je uložena ve FAT Boot Recordu. Za FAT tabulkou může být uložena jedna nebo více jejích záložních kopií, počet udává FAT Boot Record. Označení FAT, 12, 16 a 32, určuje kolika bity je možné cluster adresovat. Popis významu hodnoty uložené v buňce FAT, pro FAT 16, je v tab. 7.

Tab. 7 Popis hodnot jednotlivých buněk FAT 16

Hodnota	Popis
0x0000	Označuje prázdný cluster
0x0002 - 0xffff	Vstupní bod následujícího clusteru souboru
0xffff0 - 0xffff6	Rezervované
0xffff7	Cluster obsahuje vadné sektory
0xffff8 - 0xfffff	Označuje poslední cluster souboru

- Tabulka kořenového adresáře

Je pouze u FAT 12 a FAT 16. Je to oblast paměti, která začíná v sektoru za poslední FAT tabulkou a končí v sektoru před začátkem oblasti dat. V tabulce kořenového adresáře jsou uloženy informace o položkách, souborech a adresářích, uložených v kořenovém adresáři paměti. Způsob uložení těchto informací je stejný pro soubory i adresáře v tabulce kořenových adresářů, ale i podadresářů. Informace, které se o položkách ukládají, jsou jméno, přípona, atributy, datum a čas vytvoření nebo poslední změny položky, první cluster obsazený položkou a velikost v bytech. Podrobný popis způsobu uložení informací o položkách je v tab. 8, nebo případně v [10].

Tab.8 Popis uložení informací o položkách

Offset	Délka (bytů)	Popis
0x0	8	Jméno souboru
0x8	3	Přípona souboru
0xb	1	Atributy souboru
0xc	9	Rezervováno
0x16	2	Čas vytvoření nebo změny souboru
0x18	2	Datum vytvoření nebo změny souboru
0x1a	2	První cluster obsazený souborem
0x1c	4	Velikost souboru v bytech

▪ Data

V této oblasti paměti jsou uložena vlastní data. Data souborů nemusí být uložena v clusterech hned za sebou, ale v různých částech paměti.

2.3.8 Adresářová struktura

Souborový systém FAT používá stromovou strukturu adresářů. Začátek této struktury je kořenový adresář, který je u FAT 12 a FAT 16 pevně daný a má přesně stanovený počet položek (ve FAT Boot Recordu). Tento kořenový adresář se může větvit pomocí podadresářů, které se mohou dále také větvit. V každém adresáři (podadresáři) mohou být uloženy adresáře nebo soubory. Podadresáře jsou podobné souborům (liší se pouze atributem) a podobně se s nimi i pracuje. Ve FAT se potom zjistí všechny cluster, ve kterých jsou uloženy položky, které podadresář obsahuje. Číslo FAT určuje počet bitů, kterými jsou adresovány jednotlivé cluster na kartě, a určuje tak maximální možnou kapacitu paměti. Velikost clusteru může být od 1 do 64 sektorů (sektor je 512 bytů).

2.3.9 Připojení SD karty

Maximální napájecí napětí SD karty i maximální napětí na datových vstupech karty je 3,6 V. SD karta bude pracovat v SPI módu. Pro připojení SD karty k mikroprocesoru byly použity čtyři vodiče:

- hodiny (CLK)
- výběr obvodu (CS)
- data z mikroprocesoru do SD karty (MO)
- data z SD karty do mikroprocesoru (MI)

Schéma připojení SD karty k mikroprocesoru je na celkovém schématu v příloze č.4. Karta je připojena na piny SPI sběrnice mikroprocesoru, PB7 (SCK-hodiny), PB6 (MISO-data z karty), PB5 (MOSI-data kartě), PB4 (OE-připojení karty), PB3 (CS-výběr karty). Pro převod z 5 V na 3 V jsem použil na desce odporový dělič.

2.3.10 Přednastavení SD karty

Před vlastním záznamem na kartu je potřeba na ní vytvořit datový soubor „DATA.DTA“ o patřičné velikosti a uložit do něj nuly. Tento bude sloužit pro vlastní záznam dat. Dále vytvoříme soubor „INDEX.DTA“ s uloženými nulami. Tento soubor bude sloužit pro lepší vyhledávání požadovaného rozsahu dat. Třetí soubor „CONFIG.DTA“ je určen k ukládání konfiguračních dat na přejezdu (popis připojených zařízení atd.).

2.4 Rozhraní RS 485

Rozhraní RS 485 se používá pro multipoint komunikace (více zařízení může být připojeno na jedno signálové vedení) [12]. Většina RS 485 systémů používá Master/Slave architekturu, kde má každá slave jednotka svůj unikátní adresu a odpovídá pouze na jí určené pakety. Tyto pakety generuje Master (hlavní deska) a periodicky oběsílá všechny připojené slave jednotky. V Master/Slave architektuře slave nikdy nezačíná komunikaci. RS 485 existuje ve dvou variantách Single TwistedPair RS 485 a Double TwistedPair RS 485.

2.4.1. Single TwistedPair RS 485

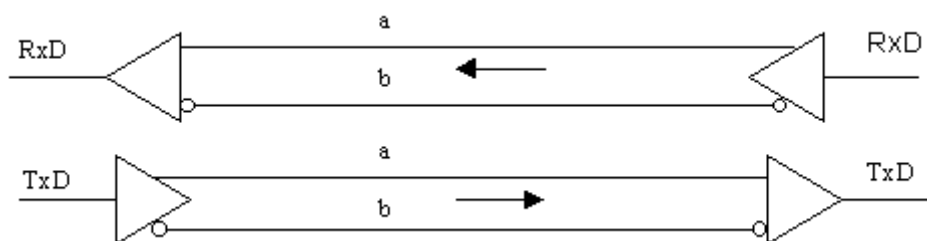
V této verzi jsou všechna zařízení připojená na jeden TwistedPair. Zařízení musí mít třístavové budiče (včetně Master). Komunikace probíhá po tomtéž vedení v obou směrech. Je tedy důležité, aby nezačalo vysílat více zařízení najednou. To je zajištěno softwarem.

2.4.2. Double TwistedPair RS 485

U této varianty Master nemusí mít třístavový výstup, protože Slave zařízení vysílají do druhého twistedpairu, určeného pro komunikaci od slave zařízení k masteru. Software na straně Master posílá ke všem Slave zařízením dotazovací pakety. Výhodou je také zvýšení datové propustnosti při větších objemech dat.

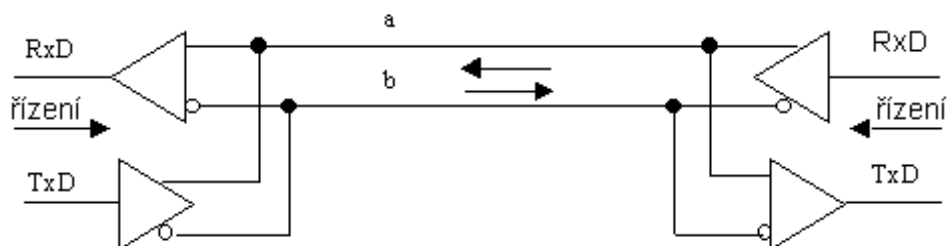
2.4.3. Provedení linky RS485 a RS422

Provedení linky RS422 je na obr. 15. Obě linky přenášejí pouze data a nepoužívají žádné řídicí signály.



Obr. 15 Vedení nevětvené linky RS422

Linka RS422 používá jeden pár vodičů pro signál RxD a druhý pro signál TxD, viz obr. 16.



Obr. 16 Vedení nevětvené linky RS485

Linka RS485 používá jeden pár vodičů pro oba směry toku dat, viz obr. 19. Je tedy třeba směr komunikace přepínat.

2.4.4. Návrh komunikačního protokolu rozhraní RS485

Navrhl jsem komunikační protokol pro rozhraní RS 485 podle doporučení „Point-to-Point Protocol“ uvedeného na CD s názvem „rfc1331.txt“. Na obrázku 17 a 18 je zobrazena sekvence komunikace Master-Slave a Slave-Master. V místě řídicího znaku CTL je informace o tom, jaká data se posílají (např. žádost o nové události, čas atd.).

STX	ADR	LEN	CTL	DATA	CS	ETX
-----	-----	-----	-----	------	----	-----

Obr. 17 Sekvence komunikačního protokolu Master-Slave

STX	ADR	ADR_sl	LEN	DATA	CS	ETX
-----	-----	--------	-----	------	----	-----

Obr. 18 Sekvence komunikačního protokolu Slave-Master

Zkratky:

STX - uvozující znak; ADR - adresa master; ADR_sl - adresa slave; LEN - délka dat;
CTL - řídicí znak; DATA - data; CS - zabezpečení; ETX - ukončující znak

Uvozujícím a ukončujícím znakem bude příznak „0x7e“. Komunikační protokol zaručuje, že uvozující znak 0x7e se nemůže objevit uvnitř sekvence. Byl by totiž nahrazen znaky 0x7D03 a potom zpětně dekodován funkcí XOR.

2.4.5. Návrh synchronizace času mezi Masterem a Slave deskami

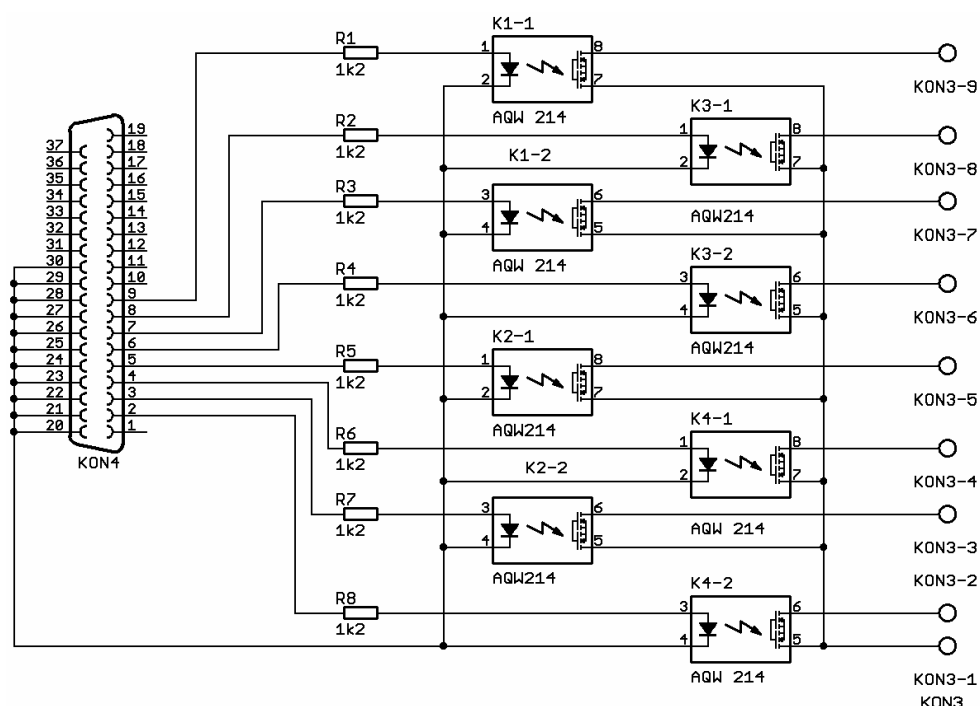
Všechny desky „Slave“ budou mít vlastní časové základny, které budou mít čas synchronizovaný cyklicky od mastera. Nejprve master pošle všem deskám čas, kterým se bude synchronizovat. Potom nastavením synchronizačního bitu, který bude rozveden ke všem deskám, se rozeslaný čas zapíše ve stejný okamžik do jednotlivých „Slave“ desek.

2.5 Modul optického oddělení vstupů

Z důvodu ochrany vstupů procesoru a většího rozsahu napětí vstupního signálu relé 10 - 24V byl navržen modul optického oddělení vstupů. Schéma zapojení modulu optického oddělení vstupů je na obr. 19. Funkční realizace je na obr. 4 a 5.

Tento modul lze připojit např. přes paralelní port k PC konektorem KON4. Pomocí programu „Simulace.exe“ jsem simuloval signály z relé. Technické parametry optických oddělovačů jsou uvedeny v katalogovém listu na CD ve složce „MANUAL“ s názvem „aqw21.pdf“.

Ke konektoru KON3B na celkovém schématu záznamového zařízení (viz příloha č.4) je připojen modul optického oddělení vstupů pomocí kabelu a konektoru KON3 (Obr.19). Pro simulaci digitálních signálů jsou v první verzi hardwaru zapojeny pouze 4 vstupy.



Obr. 19 Schéma zapojení modulu optického oddělení vstupů

3. Softwarové vybavení záznamového zařízení

3.1 Vývoj a ladění programu

Veškerý software jsem psal v jazyce C v aplikaci Win AVR - Programmer's Notepad. Pro překlad zdrojového kódu jsem použil volně dostupný kompilátor AVR GNU C. Vlastní překlad souboru s příponou „c“ jsem prováděl pomocí souboru makefile. Makefile mi generoval kromě souborů s příponou „hex“ také soubory pro ladění s příponou „elf“.

Po překladu jsem vytvořenou aplikaci nahrál přes AVR-studio do mikroprocesoru. Mikroprocesor ATmega128 je možné programovat přes rozhraní JTAG. Vygenerované

soubory s příponou „elf“, se používají k trasování aplikací přímo na čipu. Existují však jistá omezení, jako je např. nemožnost trasování jádra Nut/OS. K programování mikroprocesoru jsem použil adaptér JTAGICE (viz obr. 20), který je připojen přes rozhraní RS 232 k PC.



Obr. 20 Adaptér JTAGICE

3.2 Testování základních softwarových funkcí Nut/OS

Základní informace o možnostech operačního systému Nut/OS jsou uvedeny v kapitole 3.1. Použil jsem připravenou část zdrojového kódu dostupného v [1]. Nejprve jsem si vyzkoušel fungování operačního systému se třemi vlákny v testu1. Hlavní program střídavě vypisuje 3 písmena podle aktuálního spuštěného vlákna, viz popis níže.

3.2.1 Test 1 – Test tří vláken

- zdrojový kód je ve složce TESTY s názvem „test1_thread.c“ na přiloženém CD
- nejprve bylo nutné připojit testovací port RS 232 na základové desce k portu COM 1 na PC
- v programu „Hyperterminál“ byly nastaveny parametry portu COM1 (rychlost 115 200 b/s, parita - žádná, stop bit - 1)
- po přeložení a nahrání spustitelného kódu do aplikace byl spuštěn hlavní program
- program střídavě přepíná řízení mezi vlákny a zobrazuje písmeno „H“ (vlákno hlavního programu), „V“ (vlákno s vyšší prioritou) a „N“ (vlákno s nižší prioritou)
- hlavní funkce:

<code>putchar('H');</code>	-> výpis znaku “H” na monitor
<code>NutThreadCreate("t1", Thread1, 0, 512);</code>	-> definice vlákna s názvem Thread1
<code>NutThreadSetPriority(128);</code>	-> nastavení priority vlákna
<code>NutSleep(125);</code>	-> funkce pozastavuje dané vlákno na dobu 125 ms a předává řízení dalším vláknům čekajícím ve frontě

V tomto testu jsem ověřil, že Nut/OS umožňuje na předem definovanou dobu pozastavit vlákna a předávat mezi nimi řízení. Dále jsem ověřil možnost nastavení priorit vláken.

3.2.2 Test 2 - FTP server

Hlavní program tohoto testu spouští jedno vlákno ftp serveru, které je přesně každou milisekundu přerušeno časovačem a bliká s LED diodou. Zdrojový kód je ve složce TESTY s názvem „test2_ftp.c“ na přiloženém CD.

Nejprve bylo potřeba nastavit aktuální IP adresu:

```
#define MY_IPADDR "192.168.192.35"
```

Dále byla definována obsluha přerušování od časovače a funkce ftp serveru:

```
SIGNAL (SIG_OUTPUT_COMPARE1A)  
FtpService();
```

V testu 2 jsem ověřil funkci FTP serveru v programu „Total commander“. Ověřil jsem přesnost generování přerušování od časovače každou milisekundu (osciloskopem), dále funkci předávání řízení jádra a přerušování.

3.3 Popis zdrojového kódu programu a struktury vláken

Kód programu je rozdělen do knihoven, které zahrnují obsluhu jednotlivých funkčních bloků systému, viz tabulka č.9.

Tab. 9 Rozdělení knihoven

Funkční blok	Název knihovny	
Hlavní program	main.c	main.h
SD karta	sdcard.c	sdcard.h
USB	usb.c	usb.h
FTP server	ftpserv.c	ftpserv.h

Každá knihovna se skládá ze dvou souborů. Zdrojové kódy jsou v souborech s příponou „c“. Konstanty, proměnné a hlavičky funkcí jsou definovány v souborech s příponou „h“. Popis registrů a funkcí mikroprocesoru je v [3]. Knihovny jsou k hlavnímu programu připojeny pomocí direktivy #include. Všechny zdrojové kódy jsou na přiloženém CD ve složce SW_C. Součástí je také soubor „ftpd.c“ a „ftpd.h“ převzaté z [1] a také soubor „Makefile“. V následujících kapitolách je uveden popis významných funkcí z jednotlivých knihoven. Vývojový diagram hlavního programu je v příloze č.5 a vývojové diagramy složitějších funkcí jsou v příloze č.6.

3.3.1 Software k sériové komunikaci USB

Napsal jsem knihovnu **usb.c** a **usb.h** pro komunikaci s obvodem FT 2232C, který zprostředkovává USB rozhraní.

Definice a popis funkcí v knihovně „usb.c“:

(a) Vysílání znaku

unsigned char Znak_Vyslat(unsigned char znak)

Tato funkce posílá jeden byte přes rozhraní USB do PC. Funkce má jeden parametr znak, který se má vysílat.

(b) Čtení příkazů

void Read_Command(void)

Tato funkce slouží ke čtení a interpretování příkazů ze sériové linky USB. Funkce nemá žádný parametr.

(c) Odeslání zprávy

unsigned char Send_Msg(void)

Tato funkce zahajuje odeslání zprávy z bufferu přes USB. Funkce nemá žádný parametr.

(d) Čtení znaků z EEPROM

*void Read_EEPROMstr(void * adr, unsigned char * str, unsigned char n)*

Tato funkce přečte blok znaků z EEPROM od zadané adresy. Funkce má tři parametry (adresu, pointer a počet znaků).

3.3.2 Software pro komunikaci s SD kartou

Vytvořil jsem knihovnu pro ovládání SD karty, soubory **sdcard.c** a **sdcard.h**.

Definice a popis funkcí v knihovně „sdcard.c“:

(a) Zápis bytu

unsigned char SPI_WriteByte(unsigned char val)

Tato funkce zapíše 1 byte do bufferu SPI rozhraní. Funkce má jeden parametr zapisovaný byte.

(b) Reset SD karty

unsigned char MMC_SD_Reset(void)

Tato funkce resetuje SD kartu. Nejprve pošle „idle“ příkaz a potom „active“ příkaz. Funkce nemá parametry.

(c) Načtení bloku dat z SD karty

unsigned char MMC_SD_ReadSingleBlock(uint32 sector, uint8 buffer)*

Funkce čte jeden sektor z SD karty. Funkce má 2 parametry (číslo sektoru a místo uložení dat).

(d) Zápis bloku dat na SD kartu

MMC_SD_WriteSingleBlock(uint32 sector, uint8 buffer)*

Funkce zapíše blok dat na jeden sektor SD karty. Funkce má 2 parametry (číslo sektoru a zdroj dat).

(e) Reset SD karty

void MMC_SD_Init(void)

Funkce inicializuje SD kartu a nemá parametry.

(f) Uložení informací z Fat Boot Recordu

unsigned char Nacti_Fat(void)

Tato funkce slouží k uložení důležitých informací z Fat Boot recordu, např. počet sektorů na cluster, počet bytů na sektor, počet rezervovaných sektorů atd. Funkce je bez parametrů.

(g) Ověření požadovaného souboru

unsigned char Init_Over_Soubor(unsigned char Jm[11])

Tato funkce ověřuje, zda je na SD kartě požadovaný soubor. Případně načte jeho parametry, např. první cluster souboru, velikost souboru. Funkce má jako parametr název souboru v proměnné typu pole Jm[11].

(h) Vysílání příkazu SD kartě

unsigned char MMC_SD_SendCommand(uint8 cmd, uint32 arg, uint8 crc)

Tato funkce posílá příkazy SD kartě. Funkce má 3 parametry (kód příkazu, adresa, crc).

Příklad kódu funkce MMC_SD_SendCommand:

```
unsigned char MMC_SD_SendCommand (unsigned char cmd, unsigned long arg,
                                  unsigned char crc)
{
    unsigned char r1;
    unsigned char timeout=0; //hlidani max. casu pro vykonani
    //.....
    SPI_WriteByte(0xff);
    SPI_WriteByte(0xff);
    SPI_WriteByte(0xff);
    SPI_WriteByte(0xff);
    SPI_WriteByte(0xff);
    SPI_WriteByte(0xff);
}
```

```

    SPI_WriteByte(0xff);
//.....
    SPI_CS_Assert(); //hlidani signalu CS behem posilani prikazu a odpovedi

    SPI_WriteByte(cmd | 0x40); //pocatecni 1 a 0 ; posli prikaz
    SPI_WriteByte(arg>>24); //argument , adresa nejvyssi byte (celkem max. 4 byty)
    SPI_WriteByte(arg>>16);
    SPI_WriteByte(arg>>8);
    SPI_WriteByte(arg);
    SPI_WriteByte(crc); //CRC

    while((r1 = SPI_WriteByte(0xff)) == 0xff)
        if(timeout++ > 100) break;
    SPI_CS_Deassert();
    return r1;
}

```

3.3.3 Popis hlavního programu

Kód hlavního programu (hlavního vlákna) je uložen v souboru „main.c“. Konstanty, proměnné, definice a hlavičky funkcí jsou v „main.h“.

V kódu hlavního programu jsou v úvodu uvedeny funkce, jejichž popis je dále v textu. Nejprve se inicializuje ftp server, pak následuje alokace bufferů (dynamická paměť), inicializace jednotlivých částí zařízení, spuštění vlákna ftp serveru a vlákna obsluhy USB rozhraní. Následuje načtení informací o FAT SD karty, smazání starých dat ve čtecím bufferu USB obvodu FTDI, spuštění časovače 1, povolení zápisu událostí a zápis události RESET.

Vlastní smyčka hlavního programu obsahuje test změny vyhledávacího indexu při změně času, dále funkci „Process“ (testuje, zda v kruhovém bufferu je nový vzorek určený ke zpracování) a nakonec předání řízení na 1 ms dalšímu vláknu.

Hlavní program je každou milisekundu přerušen od časovače 1, jehož obsluha má nejvyšší prioritu a vzorkuje vstupní data (jsou generovány pomocí PC přes paralelní port). Vzorky se zaznamenávají do kruhového mezi-bufferu (slouží pro vyrovnávání zpoždění vzniklých operačním systémem). Z mezi-bufferu jsou čteny a zpracovávány funkcí „Process“. Tato funkce volá funkci „BitStatProc“, která generuje události a pomocí dalších funkcí zaznamenává tyto události na SD kartu. Zatím jsou v kódu tři typy událostí (nástupná hrana-0x20, sestupná hrana-0x30 a reset událost). Reset událost je generována vždy po resetu záznamového zařízení a má kód 0x10. V budoucnu je možné rozšiřovat typy událostí např. změna času, kmitání atd. Po naplnění karty se nejstarší záznamy přepisují.

Definice a popis obsluhy přerušování časovače v „main.c“:

- (a) Obsluha přerušování od časovače 1

SIGNAL (SIG_OUTPUT_COMPARE1A)

V obsluze přerušování je provedeno vzorkování vstupů každou ms a uložení do kruhového bufferu, viz vývojový diagram v příloze č.5.

Definice a popis funkcí v „main.c“:

- (b) Čtení dat z portu

unsigned char Read_PortL(void)

Funkce čte data z portu. Parametry funkce nemá.

- (c) Počáteční nastavení mikroprocesoru

void HW_init(void)

Tato funkce nastavuje softwarově některé obvody mikroprocesoru do výchozího stavu. Funkce je bez parametru.

- (d) Zápis do souboru

unsigned char WriteFileData(void)

Tato funkce zapisuje do volného sektoru SD karty (do souboru DATA.DTA) obsah 'BufferWR' s vygenerovanými událostmi. Funkce je bez parametru.

- (e) Zápis dat do vyhledávací tabulky INDEX

*void WriteINDEX(unsigned long *index_h, unsigned long *sector_h, unsigned char repeat_h)*

Tato funkce zapisuje 8 bytů informací pro rychlé vyhledávání v datech na SD kartě. Funkce volá funkci Write_SD_byte_IND.

- (f) Zápis 1 bytu dat do vyhledávací tabulky INDEX

void Write_SD_byte_IND(unsigned char data)

Funkce zapisuje do bufferu „BufferWR_IND“ 1 byte dat.

- (g) Zápis do souboru INDEX.DTA

unsigned char WriteFileINDEX(void)

Tato funkce zapisuje do příslušného sektoru SD karty (do souboru INDEX.DTA) obsah „BufferWR_IND“.

- (h) Vyhledání sektoru

unsigned long Find_Sector(unsigned long time_h)

Funkce zjistí z indexovací tabulky sektor pro vyčítání dat

- (i) Zápis 1 bytu dat

```
void Write_SD_byte_ptr(unsigned char data)
```

Funkce zapisuje do „BufferWR“ data (za poslední událost v rámci sektoru).

- (j) Zápis události

```
void WriteEvent(unsigned char EventChannel, unsigned char EventCode, volatile  
unsigned long* EventTime1, unsigned int* EventTime2, unsigned char param1,  
unsigned char param2)
```

Funkce zapíše na SD kartu podle parametrů vygenerovanou událost. Využívá funkci „Write_SD_byte_ptr“.

- (k) Zpracování vzorků

```
void BitStatProc(unsigned char bit_n, unsigned char bit, unsigned char bit_s)
```

Funkce zpracuje naměřený vzorek a generuje události.

- (l) Čtení vzorků z mezi-bufferu

```
void Process(void)
```

Tato funkce testuje, zda je nějaký nový vzorek v kruhovém mezi-bufferu. Vzorek načte a zpracovává ho pro všechny vstupy funkcí „BitStatProc“. Funkce inkrementuje systémový čas. V této funkci byl proveden test měření doby zpracování vzorků a generování událostí.

- (m) USB komunikace

```
void USB_com(void)
```

Funkce obsluhuje komunikaci PC se záznamovým zařízením přes rozhraní USB.

3.3.4 Popis struktury vláken

Vedle hlavního vlákna jsou spuštěna ještě další dvě. První vlákno řídí vyčtení požadovaného rozsahu dat z karty (zadané rozsahem požadovaných sektorů) a posílá tato data přes USB rozhraní do počítače. Druhé vlákno obsluhuje ftp server (v základní verzi) umožňující zatím jen bez narušení vzorkování kopírovat menší soubory z PC do záznamového zařízení a mazat je.

(a) USB vlákno

THREAD(UsbThread, arg)

Smyčka obsluhy USB vlákna slouží ke čtení dat uložených na SD kartě přes USB rozhraní. Vlákno používá funkci „USB_com“. Je zde nastavena priorita vlákna 16, tedy vyšší priorita než u FTP vlákna.. Jestliže se nezapisuje na kartu, je možné číst data do PC. Vlákno je pozastaveno na 200 ms. Vývojový diagram je v příloze č.5.

(b) FTP vlákno

THREAD(FtpThread, arg)

Smyčka obsluhy FTP vlákna slouží k obsluze FTP serveru. Vlákno využívá funkci „FtpService“. Je zde nastavena priorita 18. Vlákno je pozastaveno na 100 ms. Vývojový diagram je v příloze č.5.

3.4 Rychlé vyhledávání v záznamu SD karty

Navrhl jsem a naprogramoval funkci rychlého vyhledávání čísla sektoru začátku a konce požadovaného záznamu. Ve zdrojovém kódu je tato funkce pod názvem „Find_Sector“. Struktura každého řádku vyhledávací tabulky je uvedena na obrázku č. 21.

číslo bytu	0	1	2	3	4	5	6	7
1. řádek	INDEX - 3 Byty			Číslo sektoru - 4 Byty				Opakování

Obr. 21 Struktura vyhledávací tabulky v souboru INDEX.DTA

První tři byty obsahují index, popisovaný níže. Další čtyři byty obsahují číslo sektoru (přímo sektor v souboru DATA.DTA). Poslední 7 byte obsahuje informaci o tom, zda záznam do vyhledávací tabulky byl proveden jednou nebo několikrát (lze využít pro další vyhodnocení).

K vyhledávání slouží funkce „Find_Sector“, která používá níže uvedené vztahy:

INDEX lze vypočítat:

$$INDEX = \frac{čas}{4096} \quad (1)$$

Číslo hledaného sektoru:

$$sektor = \frac{INDEX}{64} \quad (2)$$

Číslo bytu ve vypočítaném sektoru:

$$\text{číslo bytu} = (\text{INDEX} \% 64) * 8 \quad (3)$$

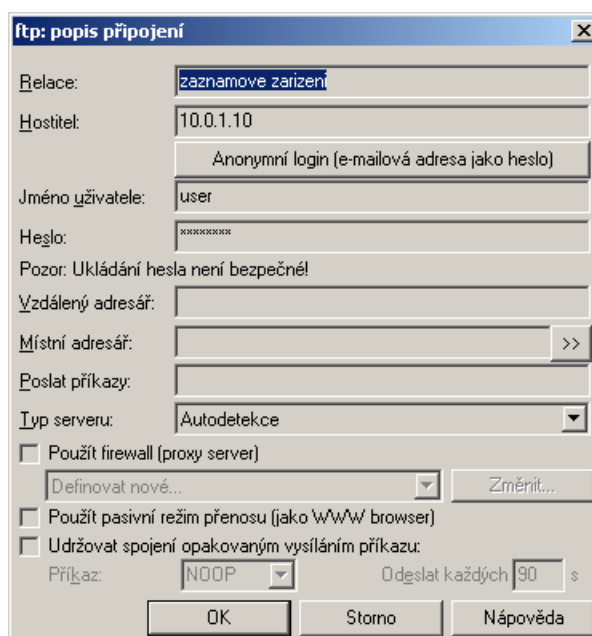
Princip rychlého vyhledávání je navržen tak, že uživatel uvede rozsah času, ve kterém ho zajímají zaznamenaná data na SD kartě. Hlavní program záznamového zařízení dostane tento čas v binární podobě od PC. Čas je dělen konstantou 4096 a vypočítá se vyhledávací index. Podle indexu se v tabulce uložené v souboru „INDEX.DTA“ vyhledá přímo číslo sektoru, kde začínají požadovaná data. Stejným postupem se získá i číslo sektoru, kde požadovaná data končí.

4. Záznam dat a měření

4.1 Nastavení připojení mezi PC a FTP serverem

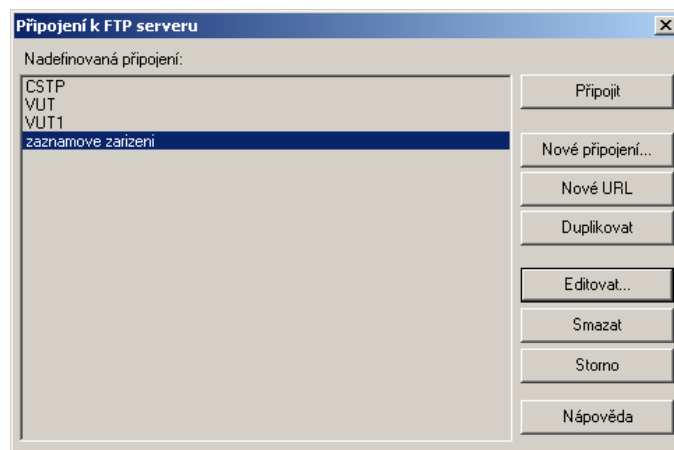
Pro připojení PC k FTP serveru (realizovaný záznamovým zařízením) je nutné nejprve vytvořit nové připojení v menu „Sít“. Nastavil jsem v Total Commanderu potřebné parametry (viz obr.22):

- název připojení (libovolný-např. záznamové zařízení)
- hostitel (IP adresa)
- jméno uživatele (user)
- heslo



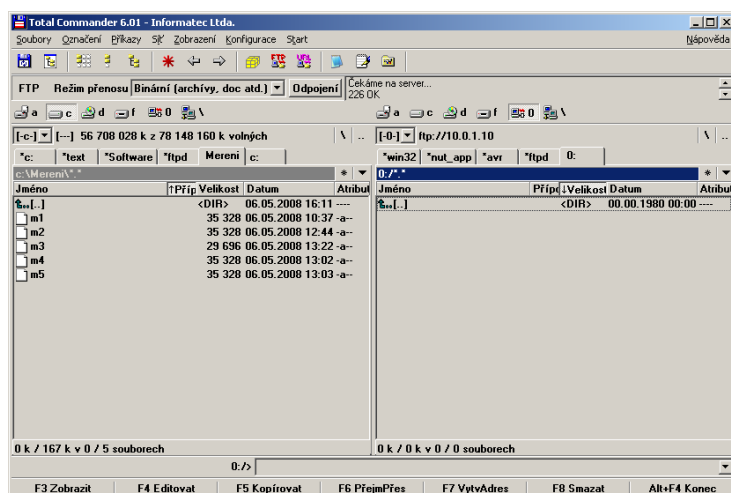
Obr. 22 Nastavení připojení k FTP serveru v Total Commanderu

Dále v menu FTP připojení jsem vybral definované připojení a stisknul tlačítko OK , viz obr č.23.

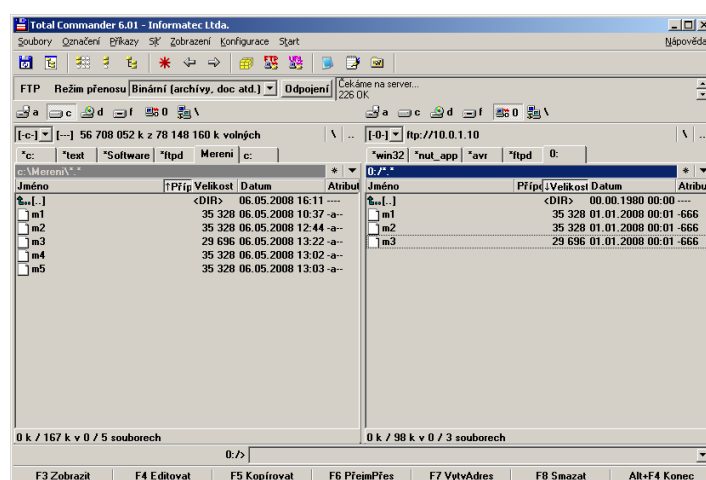


Obr. 23 Připojení k FTP serveru v Total Commanderu

Na obrázku č.24 je zobrazen obsah hlavní složky FTP serveru po připojení. Na obr č.25 je zobrazen obsah hlavní složky po přenosu dat.



Obr. 24 Obsah složky FTP serveru v Total Commanderu



Obr. 25 Obsah složky FTP serveru v Total Commanderu - po přenosu dat

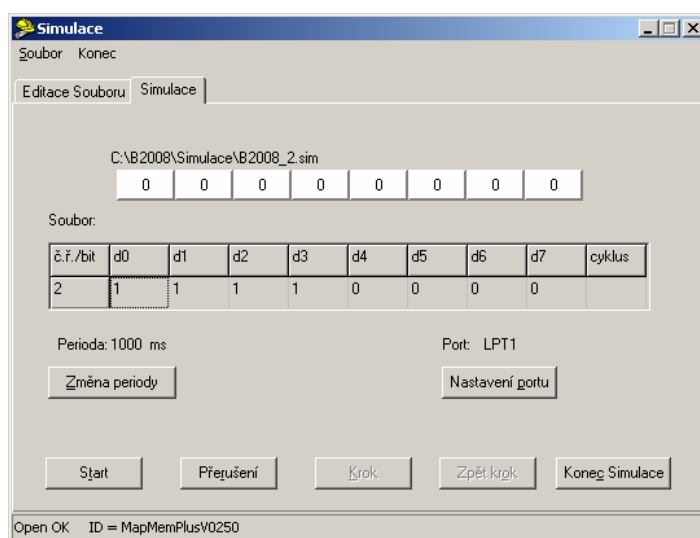
FTP server byl testován pouze v základních funkcích jako je kopírování souborů, složek a mazání souborů. Těmito funkcemi jsem zatěžoval systém při vlastním snímání a záznamu dat ze vstupů, současně s vláknem pro obsluhu čtení dat přes USB rozhraní.

4.2 Simulace signálů na vstupu

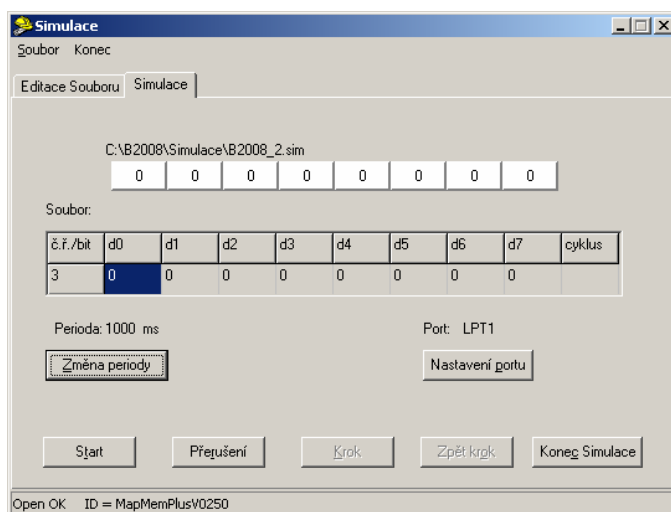
4.2.1 Program Simulace

Pro účely simulace digitálních signálů na vstupu záznamového zařízení jsem použil program (od firmy pro kterou je tento projekt) „Simulace.exe“.

Nejprve jsem vytvořil šablonu signálů, která generovala signál 1111 a 0000 na čtyřech bitech (d0,d1,d2,d3) s periodou nastavitelnou v milisekundách (viz obr.č.26 a 27).



Obr. 26 Program „Simulace.exe“- signál 1111

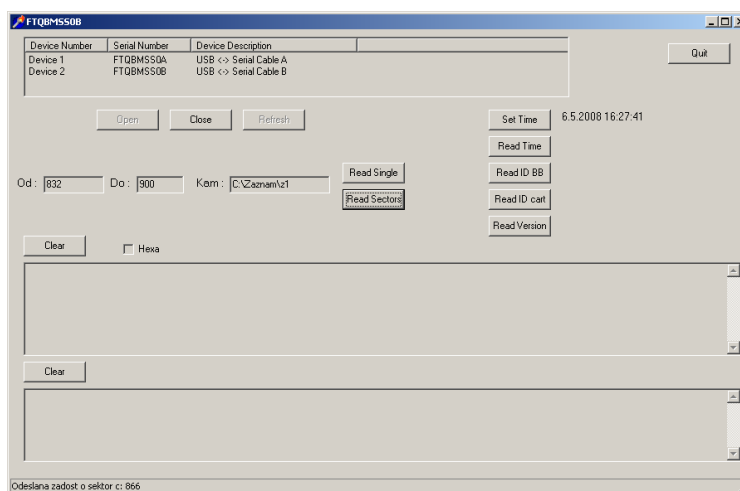


Obr. 27 Program „Simulace.exe“ - signál 0000

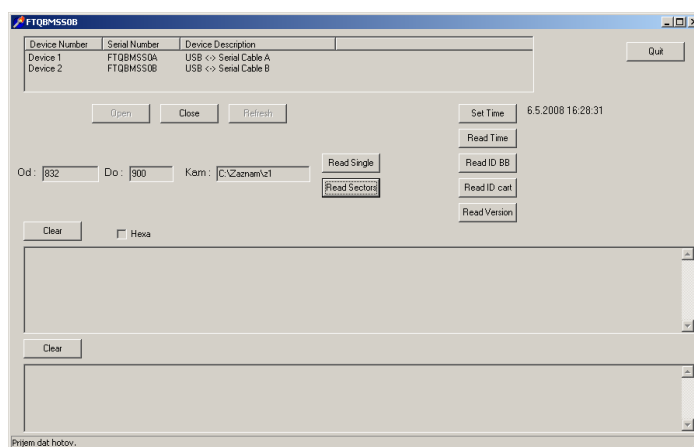
4.2.2 Program pro komunikaci mezi PC a záznamovým zařízením přes USB rozhraní

Komunikaci mezi PC a mikroprocesorem přes rozhraní USB lze rozdělit na dvě části. První část je komunikace mezi obvodem FTDI a mikroprocesorem ATMEGA128 (můj navržený kód v C), viz popis USB vlákna.

Druhou částí je komunikace mezi PC a obvodem FTDI. Zde jsem použil dále popisovaný program navržený firmou, u které jsem dělal bakalářskou práci. Na obr.28 je zobrazeno hlavní okno programu. Nejprve jsem vybral v menu kanál B, dále jsem zadal rozmezí sektorů a název souboru, kam se budou načtená data z SD karty ukládat. Po stisku tlačítka „Read Sectors“ se v dolním menu objeví zpráva, že byla odeslána žádost o sektor č. XX (viz obr.27). Po odeslání všech sektorů se zobrazí text „Příjem dat hotov“ (viz obr.29).



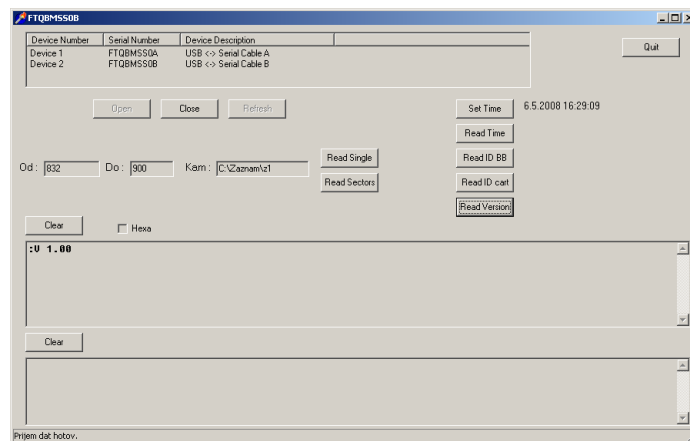
Obr. 28 Program pro komunikaci záznamového zařízení s PC přes rozhraní USB - tlačítko „Read Sectors“



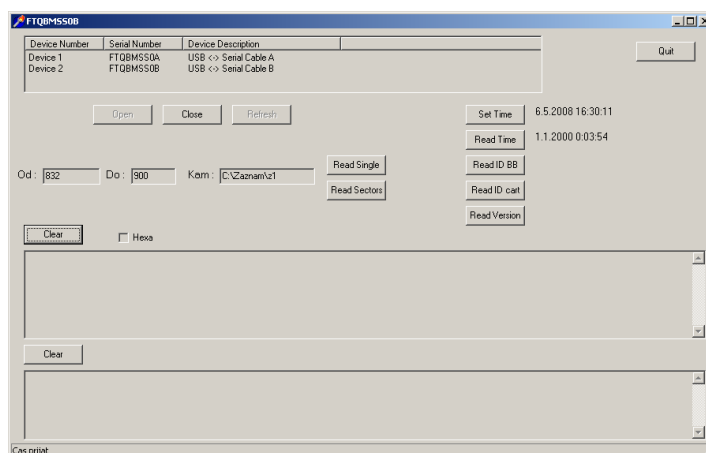
Obr. 29 Program pro komunikaci záznamového zařízení s PC

Další funkce tohoto programu – načtení a zobrazení:

- verze firmwaru tlačítkem „Read Version“ (viz obr.30)
- ID záznamového zařízení tlačítkem „Read ID BB“
- ID SD karty tlačítkem „Read ID card“
- interního času záznamového zařízení tlačítkem „Read Time“ (viz obr.31)
- nastavení interního času záznamového zařízení podle času PC tlačítkem „Set Time“



Obr. 30 Program pro komunikaci záznamového zařízení s PC
- tlačítko „Read Version“



Obr. 31 Program pro komunikaci záznamového zařízení s PC
- tlačítko „Read Time“

4.3 Záznam vstupních signálů a test přesnosti záznamu

V následujícím textu je uveden popis testu funkčnosti záznamového zařízení a testu přesnosti záznamu.

4.3.1 Postup při testování funkčnosti záznamového zařízení:

- zkontrolovat, zda je vložena SD karta s předem vytvořeným souborem o požadované velikosti „DATA.DTA“ a „INDEX.DTA“ (soubory obsahují pouze nuly) v patci na SD kartu
- zkontrolovat, zda jsou připojeny všechny kabely – napájení, propojení s PC (USB, ethernet, paralelní port, příp.RS 232)
- spustit na PC program č.1 „simulace.exe“, který generuje vstupní signál (dále P1) a nastavit paralelní port
- v P1 nastavit vybranou posloupnost 1111/0000 (viz článek 4.2.1) v menu „Editace souboru“, dále nastavit potřebnou periodu generování signálu v „ms“ tlačítkem „změna periody“ a spustit generování tlačítkem „Start“
- spustit na PC program č.2 pro komunikaci se záznamovým zařízením přes USB rozhraní (dále P2)
- v P2 zadat rozsah sektorů na vyčítání (já jsem použil čtení od prvního sektoru karty 832 do 900)

- v P2 zadat název souboru a stisknout tlačítko „Read Sectors“
- po zobrazení zprávy „Příjem dat hotov“ lze čtení obsahu záznamového zařízení opakovat

Záznam dat jsem opakoval pro periodu vstupního digitálního signálu v rozsahu 200 ms až 20 s. Při generování vstupního signálu s periodou 200ms a 1000 ms jsem použil místo programu „simulace.exe“ (signály s frekvencí větší než 1 Hz zde byly nepřesné) generátor AGILENT 33120A, 15MHz, viz obr. 32.

Při každém měření jsem vstupní signál kontroloval pomocí osciloskopu Tektronix THS 710, 60MHz. Měření jsem prováděl vždy po restartu záznamového zařízení, aby data byla uložena vždy na stejných sektorech pro lepší vyhodnocení měření.

Naměřené soubory dat jsou uvedeny na přiloženém CD ve složce „Naměřená data“ s příponou *.hex. Jsou zde dvě složky, jedna obsahuje měření pouze pro jeden kanál a druhá měření na čtyřech kanálech. Data lze prohlížet např. v Total Commanderu v „hex“ formátu.



Obr. 32 Generátor AGILENT 33120A, 15MHz

4.3.2 Popis záznamového souboru *.hex

Jednotlivé záznamy se v souboru skládají z událostí. Jedna událost má pevnou velikost 16 bytů. Struktura záznamu událostí je v tabulce 10.

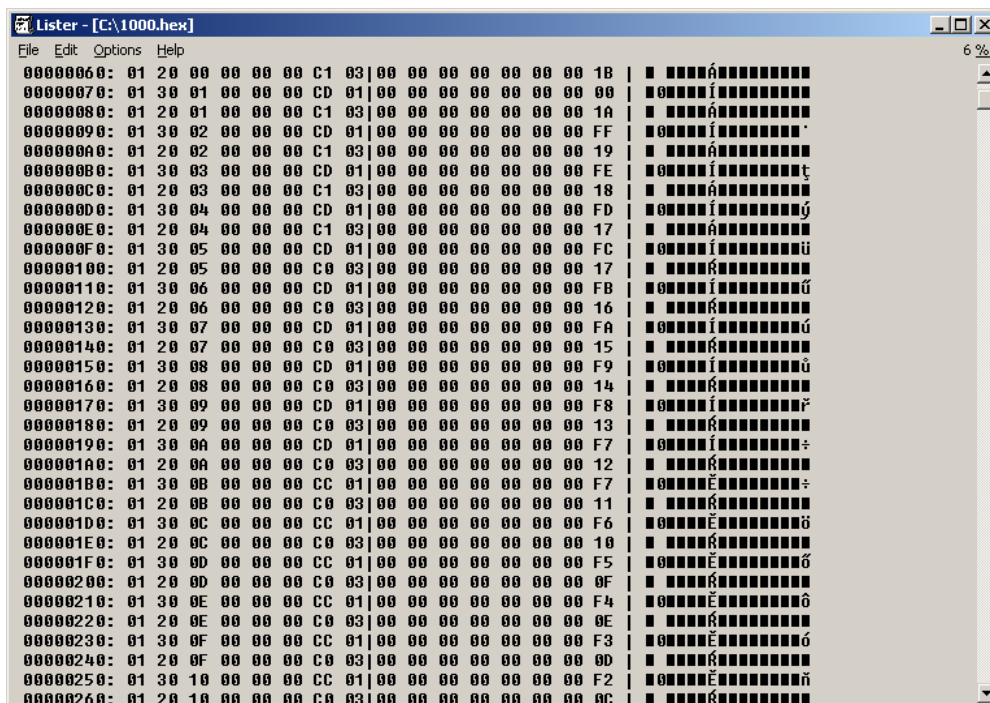
Číslo bytu	1	2	3	4	5	6	7	8	9	10	11--16
Popis události	Vstup (kanál)	Kód	Čas- sekundy				Čas- milisekundy		Par.1	Par.2	Kontrolní součet
			1 byte	2 byte	3 byte	4 byte	1 byte	2 byte			
nástupná hrana	0-3	20h	čas 1	čas 2	čas 3	čas 4	čas 5	čas 6			cs
sestupná hrana	0-3	30h	čas 1	čas 2	čas 3	čas 4	čas 5	čas 6			cs

Tab.10 Struktura záznamu události

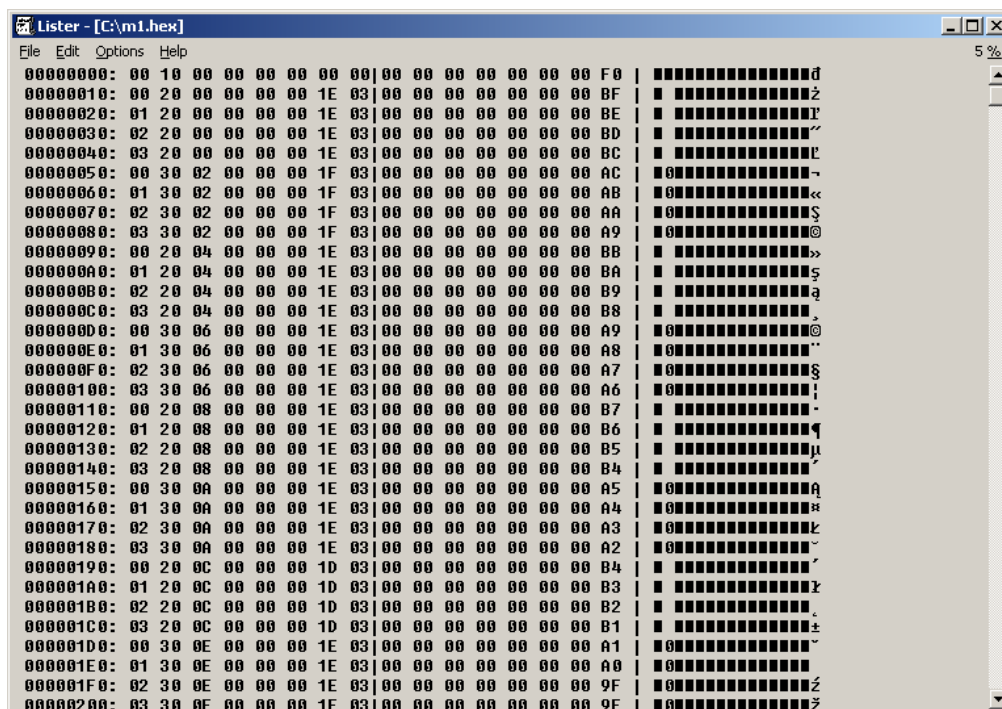
První byte je číslo vstupu (kanálu), dále následuje kód události, 4 byty času v sekundách, 2 byty času v milisekundách, parametr1, parametr 2 (některé typy událostí mohou obsahovat až 2 parametry), rezervní byty a kontrolní součet.

4.3.3 Testování přesnosti záznamového zařízení:

Zaznamenaná data uvedená v předchozím textu jsem vyhodnotil ručně. Ukázka měření na jednom a čtyřech vstupech (kanálech) je uvedena na obr. 33 a 34. V praxi budou tato data vyhodnocena softwarem pod WinXP. Tento software je však už nad rámec bakalářské práce.



Obr. 33 Záznam naměřených dat z jednoho vstupu



Obr. 34 Záznam naměřených dat ze čtyř vstupů

4.3.4 Metodika vyhodnocení přesnosti záznamu

- zkontrolovat, zda čas v [s] mezi příslušnými událostmi odpovídá vstupní periodě signálu
- v rozmezí času od 0x001 [s] do 0x04D[s] zjistit rozdíl času v [ms] a stanovit nepřesnost záznamu

4.3.5 Příklad vyhodnocení přesnosti záznamu při periodě vstupního signálu 1 s

V uvedeném příkladu, na obrázku 33 měl vstupní signál na kanálu č.01 periodu 1 s, časy jednotlivých událostí odpovídají periodě.

Příklad výpočtu přesnosti záznamu:

Čas pro událost nástupná hrana s kódem „20h“:

0x000 [s] -> 0x03C1 [ms]

0x04D [s] -> 0x03BB [ms]

Čas: $t = 0x04D = 77 \text{ s}$

Výpočet rozdílu času:

$$\Delta t = t_2 - t_1 \quad (4)$$

$$\Delta t = 0x03C1 - 0x03BB = \underline{\underline{6ms}}$$

Chybu záznamu K na SD kartě lze vypočítat dle vztahu:

$$K = \frac{\Delta t}{t} \quad (5)$$

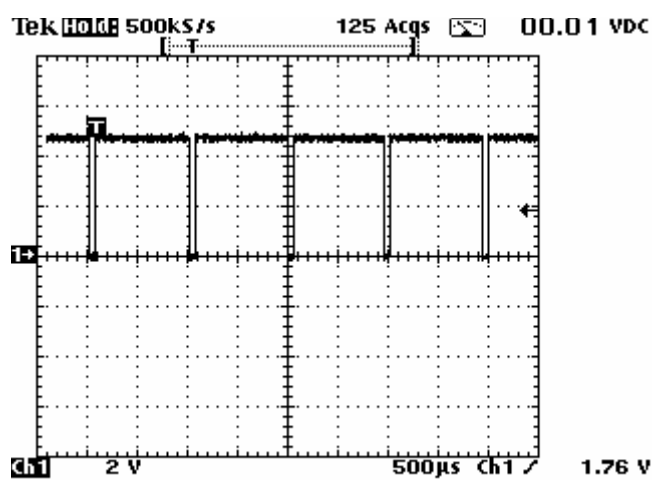
$$K = \frac{6ms}{77s} = \underline{\underline{0,0779 \frac{ms}{s}}}$$

Chyba záznamu K byla pro ostatní naměřená data téměř stejná, tato nepřesnost je dána pouze nepřesností krystalu časové základny mikroprocesoru.

4.4 Měření časové náročnosti zpracování jednoho vzorku

Úkolem tohoto testu bylo změřit čas, který potřebuje procesor na zpracování jednoho naměřeného vzorku. Použil jsem diodu LED připojenou na port procesoru, která byla ovládaná pomocí maker uvedených dále.. Ve funkci „Process“ volané opakovaně v hlavním programu (soubor „main.c“) bylo pro tento test vloženo makro (LED_ON_1) v místě těsně před zpracováním naměřeného vzorku a po zpracování makro (LED_OFF_1). Výsledný průběh napětí na LED v čase je na obrázku 35. Zpracování jednoho naměřeného vzorku trvá asi 70 mikrosekund (signál má $U = 0 \text{ V}$). Během této doby je pro každý kanál volána funkce

„BitStatProc“, která vyhodnotí měřený vzorek a zapíše na SD kartu příslušnou událost (např. sestupná hrana).



Obr. 35 Záznam z osciloskopu – měření doby zpracování naměřeného vzorku

5. Závěr

V bakalářské práci jsem navrhl koncepci záznamového zařízení pro snímání a záznam digitálních signálů na železničních přejezdech. Záznamy se ukládají na vyjímatelnou paměťovou kartu SD ve formátu FAT 16. Kapacita této karty je stanovena na 40 dní provozu na přejezdu (2GB). Po naplnění karty se nejstarší záznamy přepisují. Vyčítání záznamů bude v praxi prováděno za účelem oprav a údržby diagnostikovaného zařízení na přejezdu. Záznam urychlí detekci poruchy a zkracuje dobu opravy. Dále se vyčítání záznamů uplatní při vyšetřování nehodové události.

Koncepce je realizována základní vývojovou deskou Ethernet, ke které je konektorem J5 připojena druhá deska plošných spojů (USB rozhraní, SD karta) ke konektoru KON1 ve schématu uvedeném v příloze č.4. Celkové schéma desky Ethernet je na CD ve složce „Schéma“.

Navrhl jsem technické řešení připojení SD karty k mikroprocesoru přes sériové rozhraní SPI a také potřebné softwarové vybavení. Dále jsem řešil komunikaci základní desky záznamového zařízení s PC přes rozhraní USB. K tomu jsem použil obvod FT232C v paralelním režimu.

Zařízení je také vybaveno rozhraním RS 485 pro připojení dalších desek umožňujících rozšířit počet digitálních vstupů. Po dohodě s vedoucím bakalářské práce se upustilo od návrhu a realizace přídatné karty s rozšiřujícími vstupy v této práci a od sledování analogových veličin. Byl zpracován pouze návrh komunikačního protokolu přes rozhraní RS 485.

Napsal jsem základní verzi softwaru. Naučil jsem se pracovat s vlákny operačního systému. Vstupní signály jsou vzorkovány každou milisekundu a ukládají se ve formě událostí na SD kartu (událost nástupná hrana, sestupná hrana), podrobnější popis je uveden v kapitole 3.3.3 a 4.3.2. Po restartu se ukládá událost „reset“.

Vyčtení dat ze zařízení je zatím možné pouze přes rozhraní USB. Rozhraní ethernet jsem použil pro testování FTP serveru, který byl testován pouze v základních funkcích jako je kopírování souborů, složek a mazání souborů. Těmito funkcemi jsem zatěžoval systém při vlastním snímání a záznamu dat ze vstupů, současně s vlákem pro obsluhu čtení dat přes USB rozhraní.

Přesnost záznamu byla ověřena v kapitole 4.3. Nepřesnost záznamu je dána pouze nepřesností krystalu časové základny mikroprocesoru. Zatím je možné synchronizovat čas ručně přes rozhraní USB.

Závěrem lze říci, že záznamové zařízení splňuje požadavky uvedené v úvodu práce. Na železničním přejezdu je především důležitá kontinuita dat bez přerušení a v práci uvedená nepřesnost je tedy pro praxi vyhovující.

V dalším vývoji nad rámec bakalářské práce je možné přidat ukládání dalších typů událostí (např. změna času, kmitání, výpadek napájení atd.). Dalším rozšířením by mohla být pravidelná synchronizace časové základny např. přes ethernet.

6. Literatura

- [1] Egnite Software GmbH. *Ethernut info* [online]. Castrop-Rauxel, Německo, 2007 - [cit. 10.10.2007]. Dostupné na WWW: <http://www.ethernut.de/>.
- [2] Egnite Software GmbH. *Ethernut Version 2.1 - Hardware User`s Manual* [online], Castrop-Rauxel, Německo, 2005 - [cit. January 2005]. Dostupný na WWW: <http://www.ethernut.de/pdf/ethwm21e-1.pdf>.
- [3] Atmel Corporation. *ATmega128*. 2006. [cit. říjen 2006]. Katalogové listy. Dostupné na WWW: http://www.atmel.com/dyn/resources/prod_documents/doc2467.pdf
- [4] HEROUT, P., *Učebnice jazyka C*. České Budějovice: KOPP, 1994. 269 s. ISBN 80-901342-1-1.
- [5] MATOUŠEK, D. *USB prakticky s obvody FTDI -1.díl*. Praha: BEN, 2003. 270 s. ISBN 80-7300-103-9.
- [6] Future Technology Devices International. *FT2232C Dual USB UART / FIFO I.C.* [online]. Glasgow, 2004. Katalogové listy. Dostupné na WWW: <http://www.asix.cz/download/ftdi/ft2232c/ds2232c12.pdf>
- [7] Future Technology Devices International. *MProg - utilita pro programování EEPROM, ver. 2.3* [online]. Glasgow, 2004 - [cit. 11.8.2004]. Dostupné na WWW: <http://www.asix.cz/ftdrivers.htm>.
- [8] Future Technology Devices International. *MProg 3.0 Manual* [online]. Glasgow, 2006 Dostupné na WWW: <http://www.ftdichip.com/Resources/Utilities/MProg.pdf>
- [9] SanDisk. *SD Memory Card Specifications / Part 1 PHYSICAL LAYER SPECIFICATION*. [online]. 2001 - [cit. 9.1.2005]. Dostupné na WWW: www.sandisk.com/industrial/manuals.asp.
- [10] DataRAID. *FAT12, FAT16 and FAT32 Windows File System* [online]. 2006 - [cit.20.5.2006]. Dostupné na WWW: <http://www.dataraid.com/storage-u/fat12-fat16-fat32-file-system/>.
- [11] Microsoft Corporation. *Microsoft Extensible Firmware Initiative FAT32 File System Specification* [online]. 2000 - [cit. 20.5.2006]. Dostupné na WWW: <http://www.microsoft.com/whdc/system/platform/firmware/fatgendown.msp>.
- [12] VOJÁČEK, A., *Základní informace o RS-485 a RS-422 pro každého* [online]. HW server s.r.o., 2007 - [cit. červenec, 2007]. Dostupné na WWW: <http://automatizace.hw.cz/zakladni-informace-o-rs-485-rs-422-pro-kazdeho>
- [13] POUCHA, P., *Přenos dat po linkách RS485 a RS422* [online]. HW server s.r.o., 1999 – [cit. 25. srpen 1999]. Dostupné na WWW: <http://hw.cz/Teorie-a-praxe/Dokumentace/ART821-RS-485-422.html>.
- [14] ZELNÍČEK, Z., *Využití paměťových karet v přístrojových aplikacích*. Diplomová práce. Brno: FEKT VUT v Brně, 2006.
- [15] Egnite Software GmbH. *Ethernut Software Manual* [online], Castrop-Rauxel, Německo, 2005 - [cit. November 2005]. Dostupný na www: <http://www.ethernut.de/pdf/enswm24e.pdf>.

7. Seznam zkratek, symbolů a příloh

7.1 Seznam zkratek a symbolů

Init_Over_Soubor()	název funkce
VelSoub	název proměnné, parametru funkce
LED_OFF_1()	makro
0x1ff	hexadecimální číslo

7.2 Seznam příloh

Příloha č.1	Příkazy SD karty v SPI módu
Příloha č.2	Formáty odpovědí SD karty
Příloha č.3	Popis FAT Boot Recordu
Příloha č.4	Doplnění schéma zapojení záznamového zařízení
Příloha č.5	Vývojové diagramy hlavního programu
Příloha č.6	Vývojové diagramy funkcí

8. Přílohy

Příloha č.1 Příkazy SD karty v SPI módu

Příkaz	Parametr	Odpověď	Zkratka	Popis
CMD0		R1	GO_IDLE_STATE	Reset karty.
CMD1		R1	SEND_OP_COND	Aktivuje inicializační proces karty.
CMD9		R1	SEND_CSD	Karta pošle registr CSD.
CMD10		R1	SEND_CID	Karta pošle registr CID.
CMD12		R1	STOP_TRANSMISSION	Karta zastaví vysílání v průběhu čtení více bloků.
CMD13		R2	SEND_STATUS	Karta pošle stavový registr.
CMD16	[31:0]len	R1	SET_BLOCKLEN	Nastaví délku bloku pro blokové operace.
CMD17	[31:0]adr	R1	READ_SINGLE_BLOCK	Čtení jednoho bloku dat.
CMD18	[31:0]adr	R1	READ_MULTIPLE_BLOCK	Čtení více bloků dat.
CMD24	[31:0]adr	R1	WRITE_BLOCK	Zápis jednoho bloku dat.
CMD25	[31:0]adr	R1	WRITE_MULTIPLE_BLOCK	Zápis více bloků dat.
CMD27		R1	PROGRAM_CSD	Nastavení nastavitelných bitů reg.CSD.
CMD28	[31:0]adr	R1b	SET_WRITE_PROT	Nastaví ochranu proti zápisu na adrese.
CMD29	[31:0]adr	R1b	CLR_WRITE_PROT	Zruší ochranu proti zápisu na adrese.
CMD30	[31:0]adr	R1	SEND_WRITE_PROT	Pošle stav ochrany proti zápisu na adrese.
CMD32	[31:0]adr	R1	TAG_SECTOR_START	Nastaví adr. prvního sektoru na smazání.
CMD33	[31:0]adr	R1	TAG_SECTOR_END	Nastaví adr. posledního sektoru na smazání.
CMD34	[31:0]adr	R1	UNTAG_SECTOR	Odstraní sektor z výběru pro mazání.
CMD35	[31:0]adr	R1	TAG_ERASE_GROUP_START	Nastaví adresu začátku mazací oblasti.
CMD36	[31:0]adr	R1	TAG_ERASE_GROUP_END	Nastaví adresu konce mazací oblasti.
CMD37	[31:0]adr	R1	UNTAG_ERASE_GROUP	Odstraní jednu mazací oblast.
CMD38	[31:0]	R1b	ERASE	Smaže předem vybrané oblasti, sektory.
CMD42	[31:0]	R1b	LOCK_UNLOCK	Nastavení hesla, zamčení/odemčení karty.
CMD58		R3	READ_OCR	Přečte registr OCR.
CMD59	[31:1]CRC	R1	CRC_ON_OFF	Zapnutí/vypnutí CRC.

Příloha č.2 Formáty odpovědí SD karty

▪ Formát R1

7	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---

- 7: Nejvyšší bit je nastaven do 0 a je vysílán jako první.
- 6: Chyba v parametru.
- 5: Chyba adresy.
- 4: Chyba v sekvenci příkazů k mazání.
- 3: Chyba v CRC.
- 2: Byl zadán chybný příkaz.
- 1: Mazací sekvence zrušena, došlo k chybě příjmu příkazu.
- 0: SD karta byla inicializovaná.

▪ Formát R1b

Stejný jako formát R1, ale na konci je přidán signál „busy“. Pokud je tento signál nula, karta zpracovává data. Není-li nula, karta je připravena na příchod dalšího příkazu.

▪ Formát R2

15	Formát R1	8	7	6	5	4	3	2	1	0
----	-----------	---	---	---	---	---	---	---	---	---

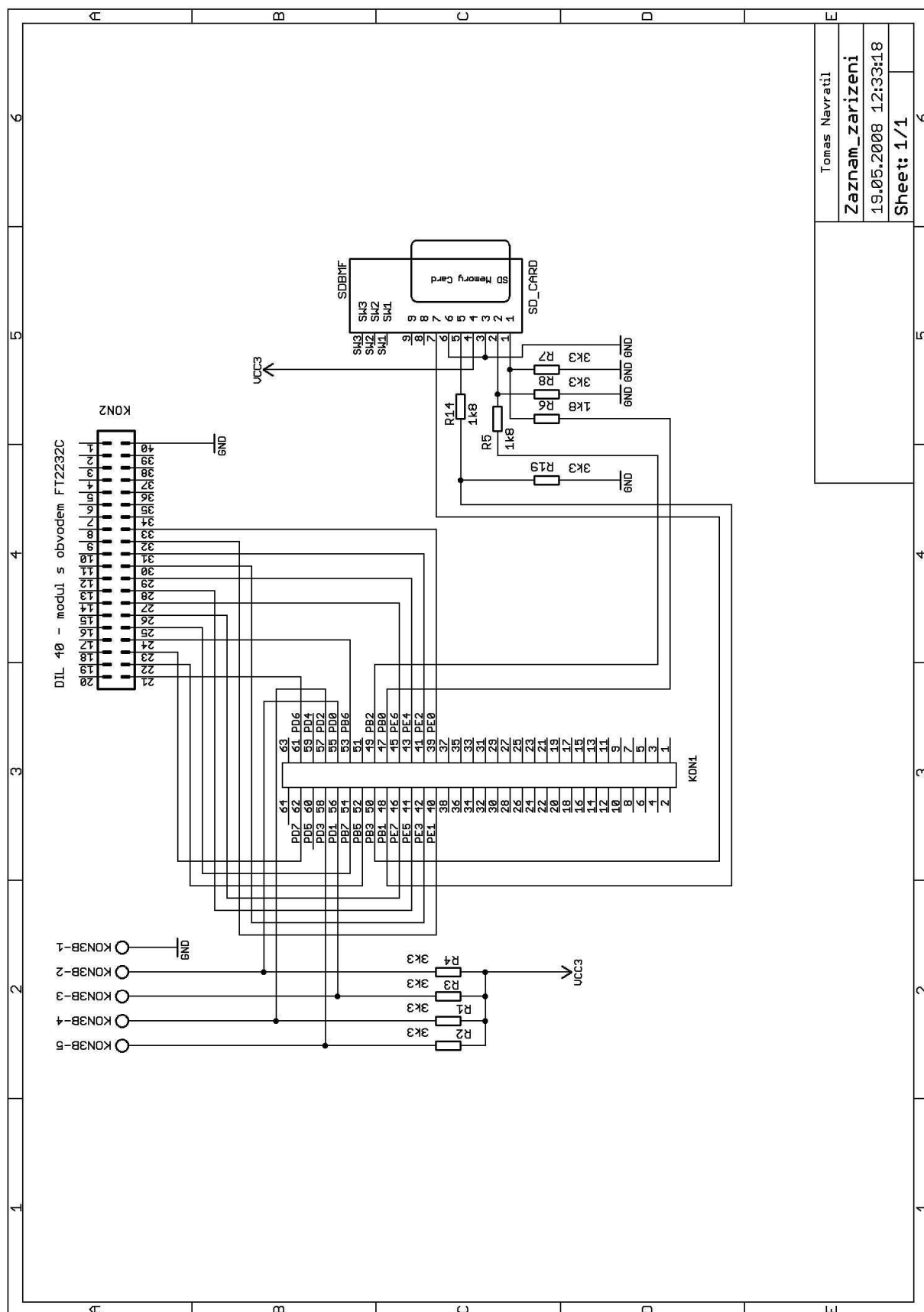
- 15-8: První byt odpovídá formátu R1.
- 7: Tento bit je nastaven, když má příkaz chybný parametr nebo když dojde k pokusu o zásah do ROM
- 6: Špatný výběr sektoru pro mazání.
- 5: Pokus o zápis na chráněný blok.
- 4: Chyba ECC
- 3: Chyba vnitřního procesoru karty.
- 2: Vyskytla se chyba v průběhu operace.
- 1: Tento bit je nastaven, když dojde k pokusu smazat chráněná data nebo dojde k chybě hesla nebo v průběhu zamykání/odemykání karty.
- 0: Když je tento bit v 1, karta je zamčená. Po resetu je 0.

▪ Formát R3 Je dlouhý 40 bitů. První část je jako R1. Druhá část je ORC registr.

Příloha č.3 Popis FAT Boot Recordu

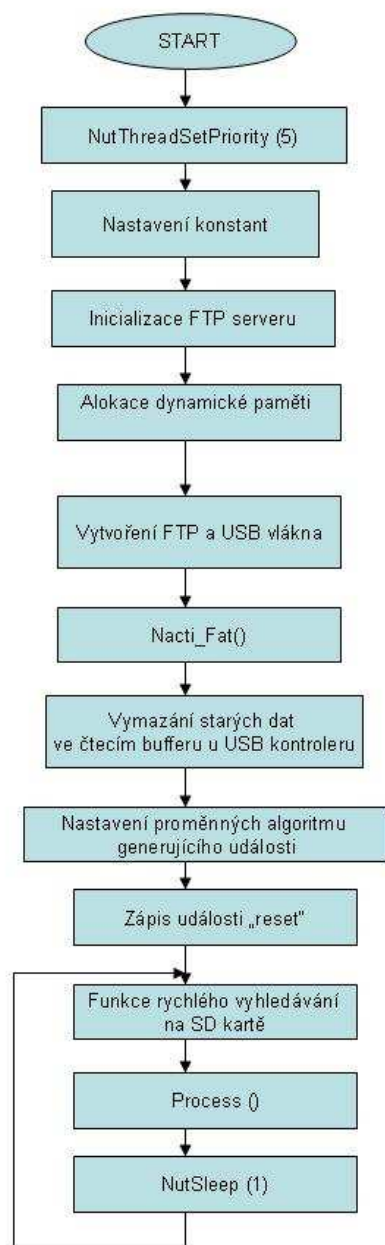
Offset	Délka (bytů)	Popis	Hodnota/Rozsah
0x0	3	Instrukce skoku	XXX
0x3	8	Jméno systému	XXX
0xb	2	Počet bytů na sektor	512
0xd	1	Počet sektorů na cluster	1-64
0xe	2	Počet rezervovaných sektorů	1
0x10	1	Počet FAT	2
0x11	2	Počet kořenových adresářů	512
0x13	2	Celkový počet sektorů	Je-li 0 platí údaj v 0x20-0x23
0x15	1	Typ média	0xf8 (hard disk)
0x16	2	Počet sektorů které zabírá FAT	XXX
0x18	2	Počet sektorů na stopu	32
0x1a	2	Počet hlav disku	2
0x1c	4	Počet skrytých sektorů	0
0x20	4	Celkový počet sektorů	XXX
0x24	1	Číslo jednotky	0x80
0x25	1	Rezervovaný	0
0x26	1	Identifikace správnosti rozšířené informace	0x29
0x27	4	Identifikační sériové číslo	XXX
0x2b	11	Jméno disku	V ASCII
0x36	8	Typ souborového systému	FAT12, FAT16 (v ASCII)
0x3e	448	Kód zavaděče	XXX
0x1fe	2	signatura	0xaa55(0xfe-0x55,0xff-0xaa)

Příloha č.4 Doplnění schéma zapojení záznamového zařízení

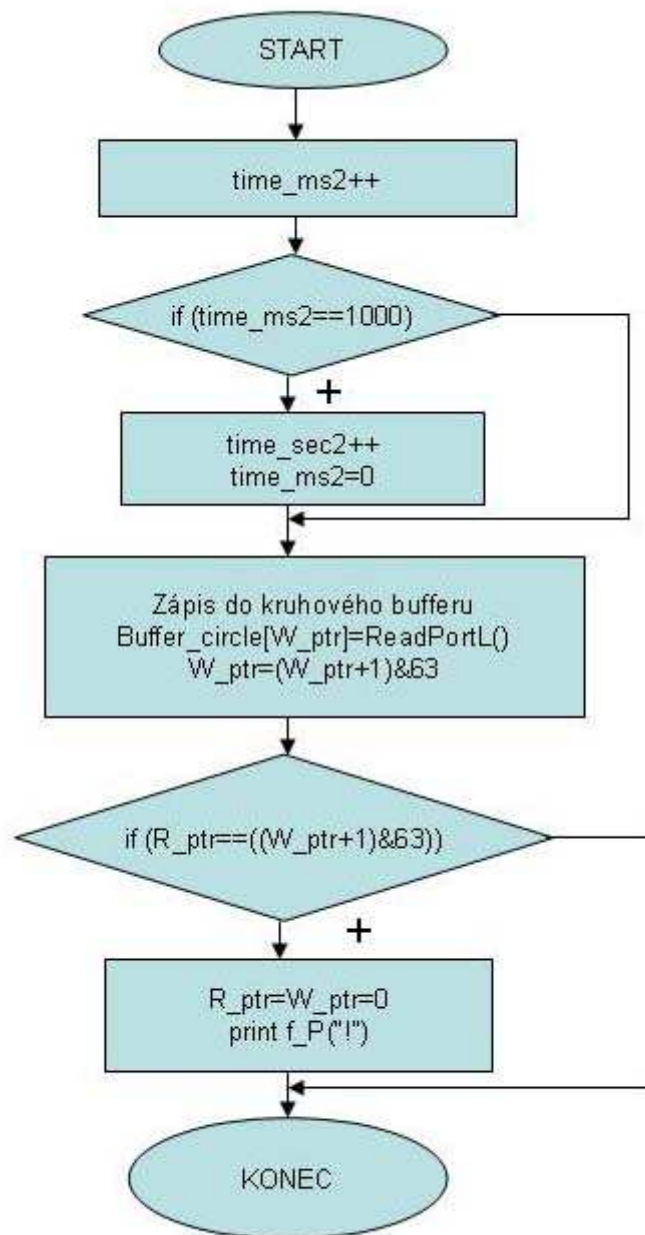


Příloha č.5 Vývojové diagramy hlavního programu

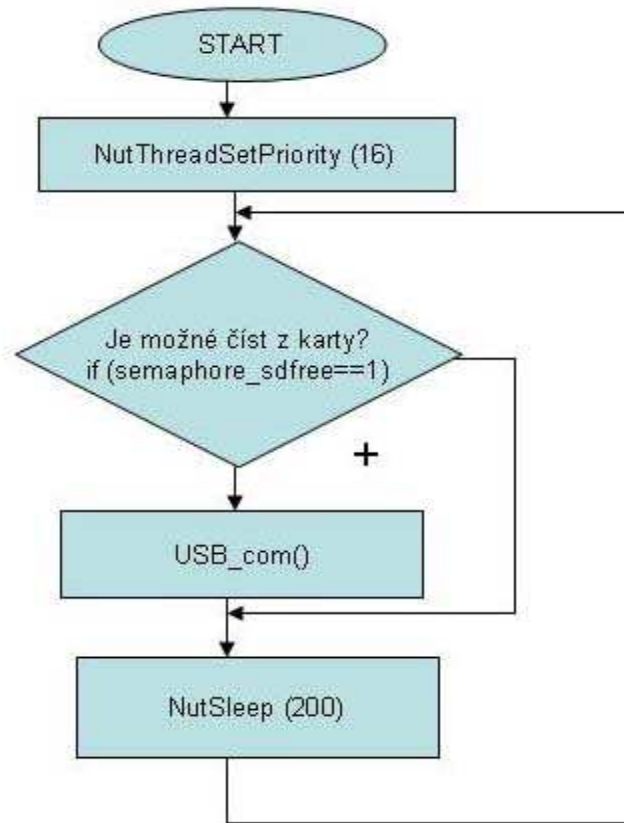
- Hlavní program



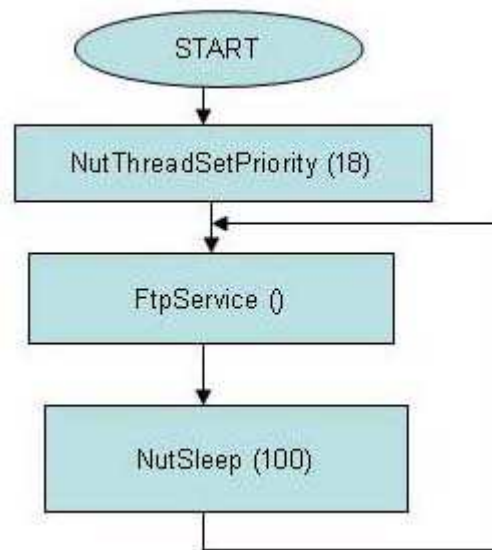
- Obsluha přerušení od časovače 1 - SIGNAL (SIG_OUTPUT_COMPARE1A)



- **THREAD (UsbThread, arg)**

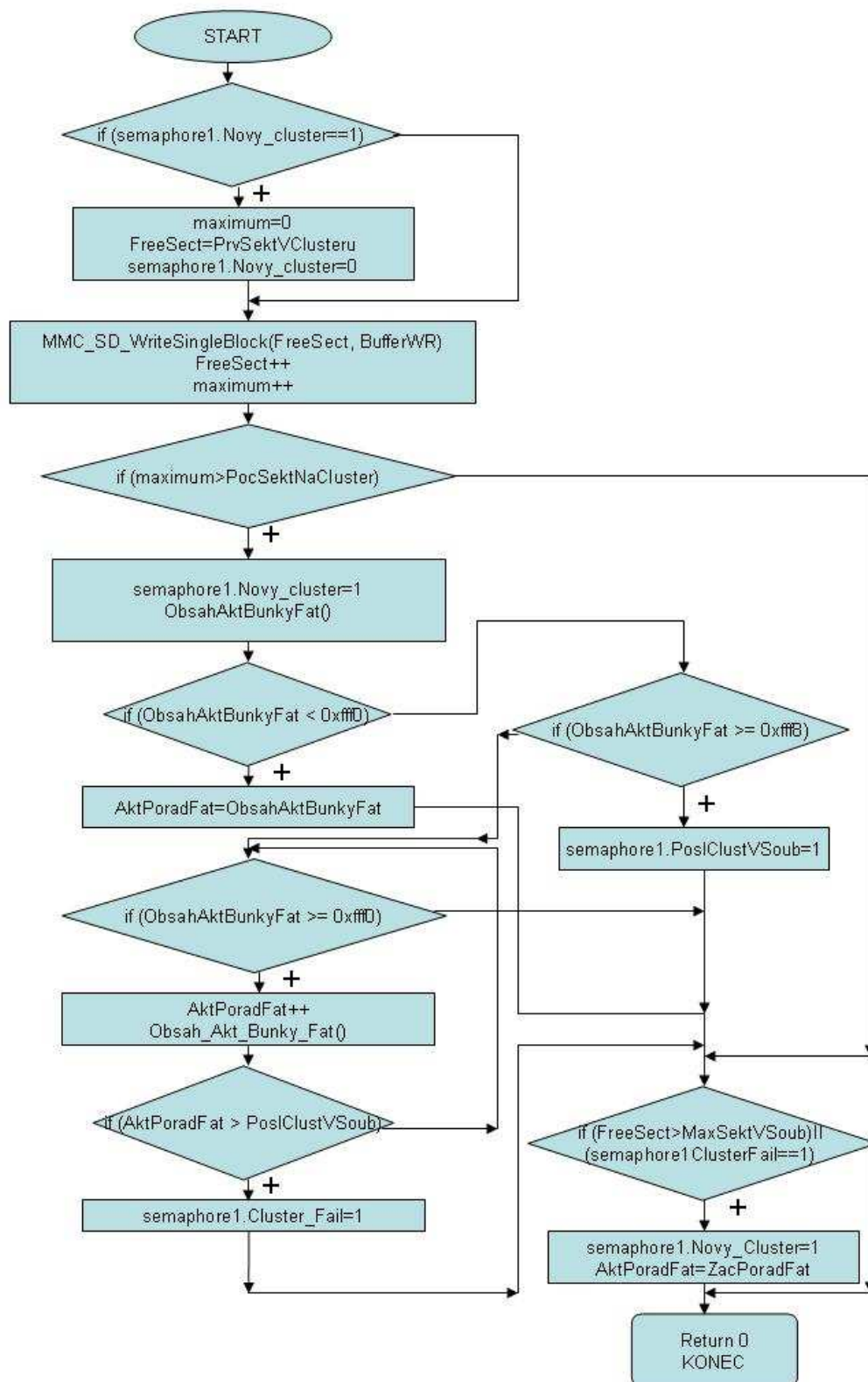


- **THREAD (FtpThread, arg)**

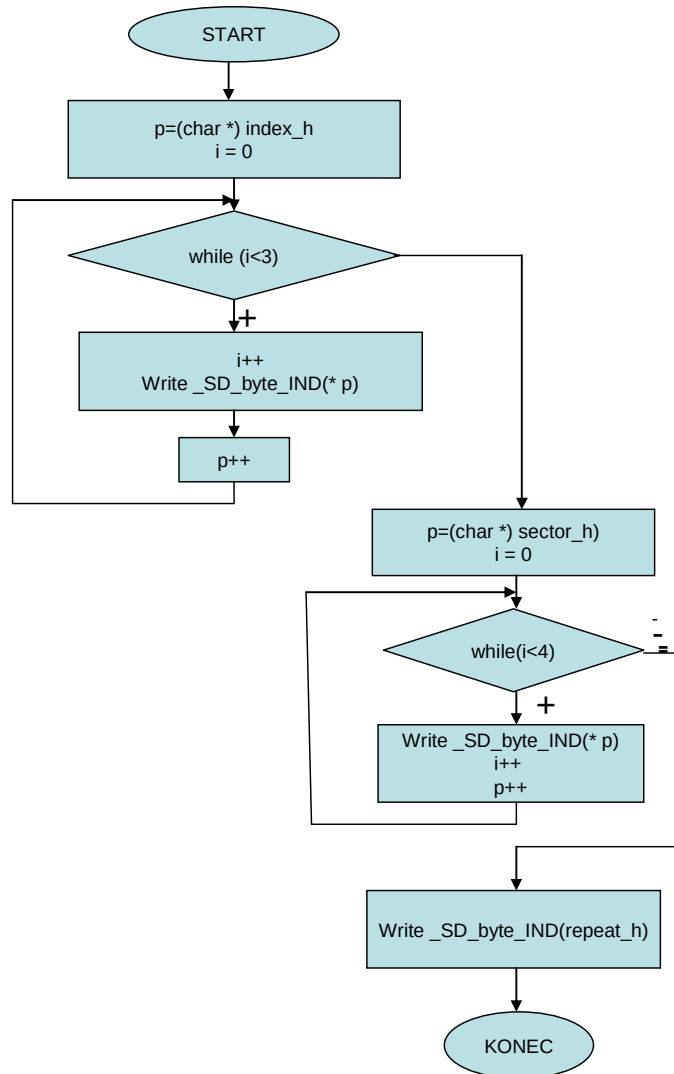


Příloha č.6 Vývojové diagramy funkcí

- Funkce WriteFileData (void)



- Funkce WriteINDEX (unsigned long *index_h,unsigned long *sector_h,unsigned char repeat_h)



- Funkce Find_Sector (unsigned long time_h)

