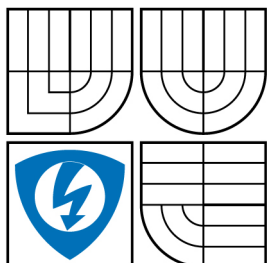


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

VÝVOJOVÝ MODUL S 32BITOVÝM PROCESOREM TYPU ARM

DEVELOPMENT BOARD WITH 32-BIT ARM-BASED PROCESSOR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

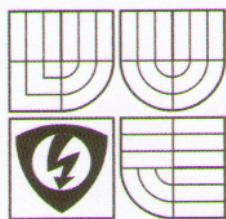
AUTOR PRÁCE
AUTHOR

Ing. VLADIMÍR POKORNÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ FRÝZA, Ph.D.

BRNO 2009



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ
Fakulta elektrotechniky
a komunikačních technologií
Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Pokorný Vladimír, Ing.

Ročník: 3

ID: 98651

Akademický rok: 2008/09

NÁZEV TÉMATU:

Vývojový modul s 32bitovým procesorem typu ARM

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte podrobný popis 32bitových procesorů typu ARM. Zpracujte možné postupy při vývoji nových aplikací pro tyto procesory v jazyce C (dostupná vývojová prostředí, překladače, apod.). Naprogramujte a odsimulujte jednoduchou aplikaci pro vámi zvolený procesor. Sestavte obvodové zapojení vývojové desky s procesorem typu ARM. Navrhněte a realizujte desky plošných spojů této desky.

Proveďte oživení celého zapojení. V jazyce C naprogramujte ukázkové aplikace s využitím vývojové desky. Zaměřte se především na interní periférie 32bitového procesoru.

DOPORUČENÁ LITERATURA:

[1] ARM Limited. ARM Architecture. [online]. 2007 – [cit. 20. prosince 2007]. Dostupné na WWW: <http://www.arm.com/>

[2] Atmel Corporation. AT91SAM 32-bit ARM-based Microcontrollers. [online]. 2007 – [cit. 20. prosince 2007]. Dostupné na WWW: <http://www.atmel.com/products/AT91/>

Termín zadání: 9.2.2009

Termín odevzdání: 5.6.2009

Vedoucí práce: Ing. Tomáš Frýza, Ph.D.

Konzultanti bakalářské práce:

prof. Dr. Ing. Zbyněk Raida

předseda oborové rady



UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení:	Ing. Vladimír Pokorný
Bytem:	Třešňová 1145, Strážnice, 696 62
Narozen/a (datum a místo):	22. ledna 1969 v Kutné Hoře

(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: Vývojový modul s 32bitovým procesorem typu ARM

Vedoucí/ školitel VŠKP: Ing. Tomáš Frýza, Ph.D.

Ústav: Ústav radioelektroniky

Datum obhajoby VŠKP: _____

VŠKP odevzdal autor nabyvateli*:

- | | |
|--|----------------------|
| <input checked="" type="checkbox"/> v tištěné formě | – počet exemplářů: 2 |
| <input checked="" type="checkbox"/> 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 užit, 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: 5. června 2009

.....
Nabyvatel

.....
Autor

Abstrakt

Práce se zabývá návrhem vývojového modulu s procesorem ARM. Nejprve je proveden stručný popis historie vývoje tohoto procesoru a jeho vnitřní struktury. Poté je krátce popsán vybraný procesor Atmel AT91SAM7SE512. V další části jsou představeny volně dostupné vývojové nástroje. V následující části je proveden návrh zapojení vývojového modulu. Na závěr byla podle tohoto návrhu deska vyrobena, oživena a otestována pomocí vzorového programu napsaném ve vývojovém prostředí Eclipse.

Abstract

The aim of my bachelor's thesis is creation of development module with 32 bit processor ARM. At first, it made a brief description of the history of the development of the processor and its internal structure. Then briefly described the chosen processor Atmel AT91SAM7SE512. The next section presents freely available development tools. In the following section is made a proposal of the development module. Finally, in accordance with this proposal the board was produced, revived and tested using the standard program written in a development environment Eclipse.

Klíčová slova

ARM, Atmel, AT91SAM7SE512, Eclipse, Yagarto, GNU, GCC, SAM-BA, TFT displej, MMC

Keywords

ARM, Atmel, AT91SAM7SE512, Eclipse, Yagarto, GNU, GCC, SAM-BA, TFT display, MMC

Bibliografická citace

POKORNÝ, V. *Vývojový modul s 32-bitovým procesorem typu ARM* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2009. 33 s., 10 s. příloh. Bakalářská práce. Vedoucí práce: ing. Tomáš Frýza, Ph.D.

Prohlášení

Prohlašuji, že svůj bakalářský projekt na téma **Vývojový modul s 32-bitovým procesorem ARM** jsem vypracoval samostatně pod vedením vedoucího bakalářského projektu 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 uvedeného bakalářského projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu 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 5. června 2009

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářského projektu Ing. Tomáši Frýzovi, Ph. D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mého bakalářského projektu.

V Brně dne 5. června 2009

.....
podpis autora

Obsah

Úvod.....	1
1 Popis procesoru ARM	2
1.1 Historie.....	2
1.1.1 ARM1.....	2
1.1.2 ARM2.....	2
1.2.3 ARM3.....	2
1.2.4 ARM4 a ARM5.....	2
1.2.5 ARM6.....	2
1.2.6 ARM7.....	3
1.2.7 ARM8.....	3
1.2.8 StrongARM.....	3
1.2.9 ARM9.....	3
1.2.10 ARM10E.....	3
1.2.11 ARM11.....	3
1.2.12 Cortex M3.....	4
1.2 Obecný popis.....	4
2 Jádro ARM7.....	5
2.1 Popis řady.....	5
2.2 Popis jádra ARM7TDMI.....	5
2.2.1 Zpracování instrukcí.....	6
2.2.2 Řízení paměti.....	7
2.2.3 Výkon, hustota kódu.....	7
2.2.4 Pracovní režimy.....	7
2.2.5 Registry procesoru.....	7
2.2.6 Stavový registr.....	9
2.2.7 Možnosti ladění.....	9
3 Procesor AT91SAM7SE512.....	12
4 Nástroje na vývoj.....	13
4.1 Překladače.....	13
4.2 Vývojové desky.....	14
4.3 Programátory – debuggery.....	14
4.4 Použité prostředky.....	14
4.4.1 Nastavení IDE Eclipse pro použití programu SAM-BA.....	15
5 Návrh vývojové desky.....	17
5.1 Napájecí zdroj.....	17
5.2 Komunikace s okolím.....	18
5.3 LCD displej, TFT displej a LED.....	19
5.4 Tlačítka.....	19
5.5 MMC, I2C.....	19
5.6 Analogové vstupy a výstup	19
5.7 Mikroprocesor.....	20
5.8 Rozšiřující konektory.....	20
6 Zkušební program.....	22
6.1 Nastavení procesoru.....	23
6.2 Nastavení SPI sběrnice.....	24
6.3 Ovládání TFT displeje.....	25
6.3.1 SPI.....	25
6.3.2 Adresování pixelů.....	25

6.3.4 Řízení barev pixelů.....	26
6.3.5 Inicializace	26
6.4 Ovládání karty MMC.....	27
6.4.1 Komunikace s kartou.....	27
6.4.2 Souborový systém FAT 16.....	28
6.5 Nastavení sériové linky pro ladění.....	29
6.6 Nastavení časovačů.....	30
6.7 Nastavení přerušení.....	31
7 Závěr.....	32
8 Seznam literatury.....	33
9 Seznam příloh.....	35
Příloha 1. Schéma.....	36
Příloha 2. Deska s plošnými spoji.....	39
Příloha 3. Seznam součástek.....	42

Seznam obrázků

Obr. 1: Blokové schéma jádra ARM7TDMI [5].....	6
Obr. 2: Příklad propojení zařízení při ladění.....	10
Obr. 3: Schéma toku dat při ladění.....	11
Obr. 4: Blokové schéma procesoru AT91SEM7SE[6].....	12
Obr. 5: Příklad nastavení spuštění externího programu SAM-BA v prostředí Eclipse.....	16
Obr. 6: Blokové schéma vývojového modulu.....	17
Obr. 7: Schéma zapojení napájecího zdroje +5V a +3,3V.....	18
Obr. 8: Schéma zapojení převodníků TTL/RS232.....	19
Obr. 9: Vývojový diagram zkušebního programu.....	23
Obr. 10: Průběh SPI komunikace - posílání příkazu a dat.....	25
Obr. 11: Průběhy SPI signálů při nastavení TFT displeje - výstup logického analyzátoru	27
Obr. 12: Přechytná struktura Boot Sektoru na paměťové kartě.....	28
Obr. 13: Záznam komunikace z jednotky DBGU	30
Obr. 14: Fotografie vyrobeného modulu za provozu.....	32

Seznam tabulek

Tabulka 1: Registry procesoru - stav ARM.....	8
Tabulka 2: Registry procesoru - stav Thumb.....	8
Tabulka 3: Význam bitů stavového registru CPSR.....	9
Tabulka 4: Seznam signálů na rozšiřujících konektorech	21
Tabulka 5: Příklad zápisu barev dvou pixelů.....	26

Úvod

Procesory ARM lze v současné době nalézt v mnoha elektronických zařízeních. Své uplatnění nalézají všude tam, kde je potřeba velký výpočetní výkon při malé spotřebě např. při zpracování obrazu, vyhodnocování GPS signálů, při mobilní komunikaci, atd.

Pro základní pochopení a využití schopností procesorů ARM je nutné mít k dispozici základní zapojení procesoru s možností jeho dalšího rozšiřování tzv. vývojovou desku.

Úkolem této práce je navržení takového vývojového modulu, jeho oživení a ověření jeho funkčnosti pomocí vzorového programu vyvinutého ve volně dostupných vývojových prostředcích. Tento modul bude sloužit především k výukovým účelům na seznámení s vlastnostmi procesorů typu ARM a jeho periferiemi a také bude vhodný pro vývoj nových zařízení využívající tento typ procesorů.

1 Popis procesoru ARM

1.1 Historie

Historie vzniku procesorů ARM sahá až ke konci 80-tých let 20. století kdy vznikl první RISC procesor ARM ve firmě Acorn RISC Machine. Následující úspěchy s procesory ARM1, ARM2 a ARM3 vedly k osamostatnění divize procesorů ARM pod novým názvem Advanced RISC Machine. Následovaly další procesory ARM6 -ARM 11 a také přejmenování firmy na ARM Ltd. Podrobněji v [1] a [2].

1.1.1 ARM1

Historicky prvním typem byl ARM1. Tento procesor byl použit v několika vývojových systémech pro BBC a PC ale především to byl prototyp, který byl rychle nahrazen typem ARM2. Procesoru ARM1 bylo vyrobeno řádově 100 ks, neobsahovaly instrukce násobení ani koprocessor a jeho instrukce.

1.1.2 ARM2

Nedlouho po ARM1 byl vyvinut typ ARM2, který již obsahoval 27 registrů z toho zároveň přístupných 16. Měl již čtyři pracovní módy (USR, IRQ, FIQ a SVC), umožňoval adresování až 64MB, měl třístavové zřetězení při zpracovávání instrukcí a jeho hodinový kmitočet byl 8MHz. Tomu odpovídá výpočetní výkon 4 -4,7 MIPS.

1.2.3 ARM3

Tento typ byl jen malým vylepšením předcházejícího typu. Oproti typu ARM2 byla přidána paměť cache (4k), koprocessor a zvýšena rychlost hodinového kmitočtu na 12 – 33 MHz. Výpočetní výkon při 12 MHz byl 7MIPS, při 24MHz 13,26 MIPS a při 33MHz 17,96 MIPS.

1.2.4 ARM4 a ARM5

Tyto procesory nebyly nikdy vyrobeny.

1.2.5 ARM6

První komerční procesor s plnou 32-bitovou adresovou sběrnici (adresovat lze až 4 GB), 32 32bitových registrů, 6 pracovních módů (USR, IRQ, FIQ, SVC, Abort a Undefined). Procesoru ARM6 bylo vyráběno několik verzí:

- ARM61 – zpětně kompatibilní s verzemi ARM2/3, neměl 32-bitovou adr. sběrnici
- ARM600 – přidán MMU a paměť typu cache
- ARM610 – 3 rychlostní varianty (20, 30 a 33 MHz – tomu odpovídá výpočetní výkon, 17, 20 a 27-28 MIPS)
- ARM60 – holý chip
- ARM650 – ARM6 + paměť RAM + periferie
- ARM6l – nízkopříkonová varianta ARM6
- ARM60l – nízkopříkonová varianta ARM60

1.2.6 ARM7

Identický s ARM6 ale s větší rychlostí. Některé varianty obsahují hardwarovou násobičku. Většina změn nastala v časování různých signálů. Typ ARM700 má zvětšenou paměť cache (8k) a větší účinnost s ohledem na spotřebu. Na 40 MHz má typ ARM710 výkon 36MIPS. Dostupné verze jsou:

- ARM7 – holý chip
- ARM7D – jádro s podporou ladění
- ARM7DM – ARM7D s hardwarovou násobičkou
- ARM7DMI – ARM7DM s podporou rychlého zpracování přerušení
- ARM70DM – ARM7DMI jako chip
- ARM700 – ARM7 + MMU + cache + WriteBack Buffer
- ARM7500 – ARM7 + MMU + cache + Writeback Buffer + IOMD + VIDC20

Uvedené typy mohou být i ve verzi Thumb (možnost zpracování 16-ti bitových instrukcí)

1.2.7 ARM8

Přímo kompatibilní s ARM6 a 7. Obsahuje 5-ti stavové zřetězení zpracovávání instrukcí. Cache paměť zůstává stejně velká ale slouží pro WriteBack, dále je přidána 64-bitová násobička. Chipy jsou vyráběny technologií 0,5 um, která umožňuje dosažení výkonu až 80MIPS při 3,3V a 80MHz.

1.2.8 StrongARM

Rychlá varianta rodiny ARM. Architektura je podobná jádru ARM8, také využívá 5-ti stavové zřetězení. Rozdíl je v rozdělení paměti cache na dvě části. Na cache pro instrukce a pro data (Harvardská architektura). Každá cache má velikost 16kB. Při výrobě je použita technologie 0,35 um a výkony jsou od 115 MIPS při 100MHz až po 230 MIPS při 200MHz.

1.2.9 ARM9

Opět vychází z předchozího typu ARM8, ze kterého si bere 5-ti stavové zřetězení ale vnitřní architektura je Harvardská, jako u StrongARM. Velikosti pamětí cache může být 0kB – ARM9TDMI, 4kB/4kB – ARM940T, 8kB/8kB – ARM922T a 16kB/16kB – ARM920T. Použitá technologie je opět 0,35 um a výkon je až 200MIPS při 180 MHz.

1.2.10 ARM10E

Cesta za dalším zrychlením vedla k procesoru se šestistupňovým zřetězením, rozdělenou pamětí cache (32k/32k) a technologií 0,13 um[21]

1.2.11 ARM11

Tento typ má již osmi-stavové zřetězení, podporu cache paměti (4 – 64 kB) a technologii výroby opět 0,13 um. Hodinová frekvence je přes 1GHz a výkony jsou 740 Dhrystone 2.1 MIPS. Spotřeba je 0.6mW/MHz (0.13um, 1.2V)[22].

1.2.12 Cortex M3

Jádro založeno na tří-stupňovém zřetězení s Harvardskou architekturou a s inovovanou instrukční sadou Thumb (Thumb-2). Technologie výroby je 0,18 um. Při testu Dhrystone dosahuje výkonu 1,25 DMIPS/MHz a spotřeba je 0,19mW/MHz[23].

1.2 Obecný popis

Architektura procesorů ARM (dříve Advanced RISC Machine a před tím Acorn RISC Machine) je 32-bitová typu RISC . Tento typ procesorů je široce použit u embeded zařízení. Charakteristickým rysem je i nízká spotřeba. Tato vlastnost je důležitá především u mobilních zařízeních.[4]

2 Jádro ARM7

2.1 Popis řady

Jedná se o základní řadu 32 bitového jádra ARM. Vzhledem k nízké spotřebě, je určen pro použití v mobilních telefonech, pagerech, digitálních diářích, automatizaci, přenosných audio přístrojích apod. Výpočetní výkon dosahuje hodnot 0,9MIPS/MHz při spotřebě 0,25 mW/MHz (platí pro technologii 0,13 um).

Výhodou je nejen nízká spotřeba, ale i podpora Thumb® instrukční sady podporující 16-ti bitovou instrukční sadu, v jejímž důsledku lze zhustit kód programu. Další výhodou je velká nabídka vývojových nástrojů ať už komerčního charakteru a nebo volně dostupného - GNU licence. Mezi další výhody patří i kompatibilita programů mezi různými řadami. Programy napsané pro jádro ARM7TDMI jsou plně kompatibilní s ostatními členy rodiny ARM7 a také s vyššími rodinami procesorů ARM (ARM9, ARM9E, ARM10, StrongARM a xscale). Řada ARM7 obsahuje jádra ARMTDMI, ARMTDMI-s, ARM720T a ARM7EJ-s.

- ARM7TDMI - Toto jádro je průmyslově nejrozšířeněji používaný 32 bitový mikroprocesor optimalizovaný na nízkou spotřebu, malou velikost a vysoký výkon potřebný v přenosných a embedded aplikacích.
- ARM7TDMI-s – syntezevatelná verze předchozího jádra. Optimalizován na flexibilitu a snížení času potřebného pro vývoj nové aplikace.
- ARM720T – Obsahuje jádro ARM7TDMI, 8k cache a MMU (Memory management unit – jednotka řízení paměti). Optimalizován pro použití s operačními systémy Windows CE, Linux, palm OS a Symbian OS.
- ARM7EJs – Syntezevatelné jádro se všemi benefity jádra ARM7TDMI rozšířené o technologii Jazelle pro Java aplikace.

2.2 Popis jádra ARM7TDMI

Jádro procesoru, na Obr. 1, je založeno na Von Neumanově architektuře s 32-bitovou datovou sběrnici společnou jak pro data, tak i pro instrukce. Data mohou mít velikost 8, 16 a 32 bitů.

2.2.2. Řízení paměti

Řízení paměti je vyvinuto k optimálnímu využití výkonu a minimalizaci paměťových nároků. Časově kritické řídicí signály jsou zřetězeny, což umožňuje funkcím řídicího systému využít mód „fast-burst“ přístupu podporovanému u mnoha pamětí.

Ukládání dat může být nakonfigurováno pro formát „big-endian“ nebo „little-endian“

2.2.3 Výkon, hustota kódu

Jak již bylo zmíněno výše jádro ARM7TDMI podporuje dva operační stavy a instrukční sady:

- ARM pro 32-bitové instrukce
- thumb pro 16-bitové instrukce

ARM instrukční sada umožňuje programu dosáhnout maximálního výkonu při minimálním počtu instrukcí.

Jednodušší thumb sada nabízí větší hustotu kódu při snížené paměťové náročnosti. K přepínání mezi ARM a thumb instrukční sadou může docházet při každém volání procedury.

Většina instrukcí je prováděna v jednom cyklu. K přepnutí jednotlivých stavů dochází při vykonání instrukce BX (Branch and Exchange), kdy operační stav procesoru určuje hodnota nultého bitu v operandu. Pokud je bit nastaven na 1, tak je zvolen Thumb stav a pokud 0 tak ARM.

2.2.4 Pracovní režimy

Jádro ARM7TDMI má sedm pracovních režimů

- User mode – obvyklý pracovní režim
- Fast interrupt FIQ – režim podporující velmi rychlé zpracování přerušení
- Interrupt IRQ – režim normálního zpracování přerušení
- Supervisor mode – chráněný režim pro operační systém
- Abort mode – režim spuštěný v případě, kdy se nepodaří dokončit vykonání instrukce
- System mode – výsadní uživatelský režim pro operační systém
- Undefined mode - spuštěn při pokusu o vykonání nedefinované instrukce

Většina aplikací běží v User módu. Ostatní tzv. privilegované módy slouží pro obsluhu hardwarových přerušení, výjimek a softwarových přerušení. Každý z privilegovaných módů je spojen s registrem SPSR (Saved Program Status Register). Tento registr slouží k zálohování CPSR (Current Program Status Register).

2.2.5 Registry procesoru

Procesor obsahuje celkem 37 registrů, každý o délce 32 bitů. Z toho je 31 základních a 6 stavových. Všechny registry nejsou přístupny ve stejný čas. Jejich dostupnost je určena stavem procesoru a pracovním režimem. Uspořádání registrů pro oba stavy a pro všech 7 módů je v následujících tabulkách 1 a 2. Vyšedlé pole znamenají, že základní registry byly přepsány alternativním registrem specifikovaným výhradním módem.

Tabulka 1: Registry procesoru - stav ARM

Módy						
	Privilegovaný					
		Výjimky				
User	System	Supervisor	Abort	Undefined	Interrupt	Fast Inter.
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
R8	R8	R8	R8	R8	R8	R8_FIQ
R9	R9	R9	R9	R9	R9	R9_FIQ
R10	R10	R10	R10	R10	R10	R10_FIQ
R11	R11	R11	R11	R11	R11	R11_FIQ
R12	R12	R12	R12	R12	R12	R12_FIQ
R13	R13	R13_SVC	R13_ABR	R13_UND	R13_IRQ	R13_FIQ
R14	R14	R14_SVC	R14_ABR	R14_UND	R14_IRQ	R14_FIQ
PC	PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_SVC	SPSR_ABR	SPSR_UND	SPSR_IRQ	SPSR_FIQ

Tabulka 2: Registry procesoru - stav Thumb

Módy						
	Privilegovaný					
		Výjimky				
User	System	Supervisor	Abort	Undefined	Interrupt	Fast Inter.
R0	R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7	R7
SP	SP	SP_SVC	SP_ABR	SP_UND	SP_IRQ	SP_FIQ
LP	LP	LP_SVC	LP_ABR	LP_UND	LP_IRQ	LP_FIQ
PC	PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR_SVC	SPSR_ABR	SPSR_UND	SPSR_IRQ	SPSR_FIQ

2.2.6 Stavový registr

Každý mód jádra obsahuje minimálně jeden stavový registr. Módy výjimek pracují se dvěma stavovými registry. Při vyvolání výjimek je obsah aktuálního stavového registru (CPSR) přesunut do „archivního“ stavového registru (SPSR). Po vykonání výjimky je zase původní obsah CPSR obnoven ze SPSR. Stavový registr je 32-bitový a obsahuje jak bity příznaků tak i řídicí bity viz tabulka 3.

Tabulka 3: Význam bitů stavového registru CPSR

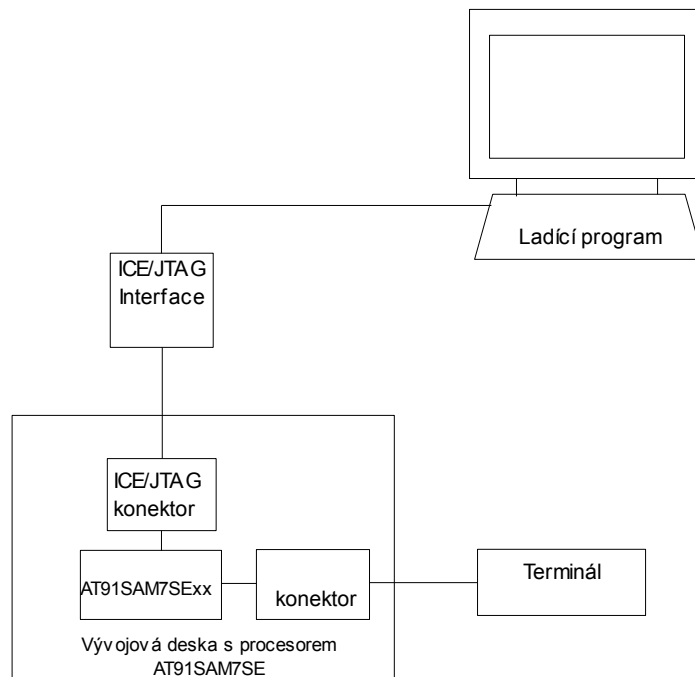
Příznaky (b31..b28)				(b27..b8)	Řídicí bity (b7..b0)			
N	Z	C	VRezerva.....	I	F	T	M0..4
N výsledek záporný nebo menší než Z nulový výsledek C přenos, výpůjčka V přetečení I zakázání přerušení IRQ F zakázání přerušení FIQ T stav Thumb M0..4... mód								

2.2.7 Možnosti ladění

Procesor obsahuje hardwarové rozšíření umožňující jednoduché ladění programů při vývoji. Mezi tato rozšíření patří JTAG/ICE (Embedded ICE) port, který slouží ke standardním ladícím operacím jako je nahrávání kódu do procesoru a jednokrokový průchod programem. Další pomocný nástroj pro ladění je Debug Unit. Tato jednotka vlastně představuje UART pomocí něhož je možné nahrát program do vnitřní paměti SRAM. Další využití této jednotky je při sledování běhu programu (posílání krátkých zpráv na připojený terminál). Největší výhody při ladění poskytuje JTAG/ICE port.

Typický ladící systém zahrnuje Obr.2.:

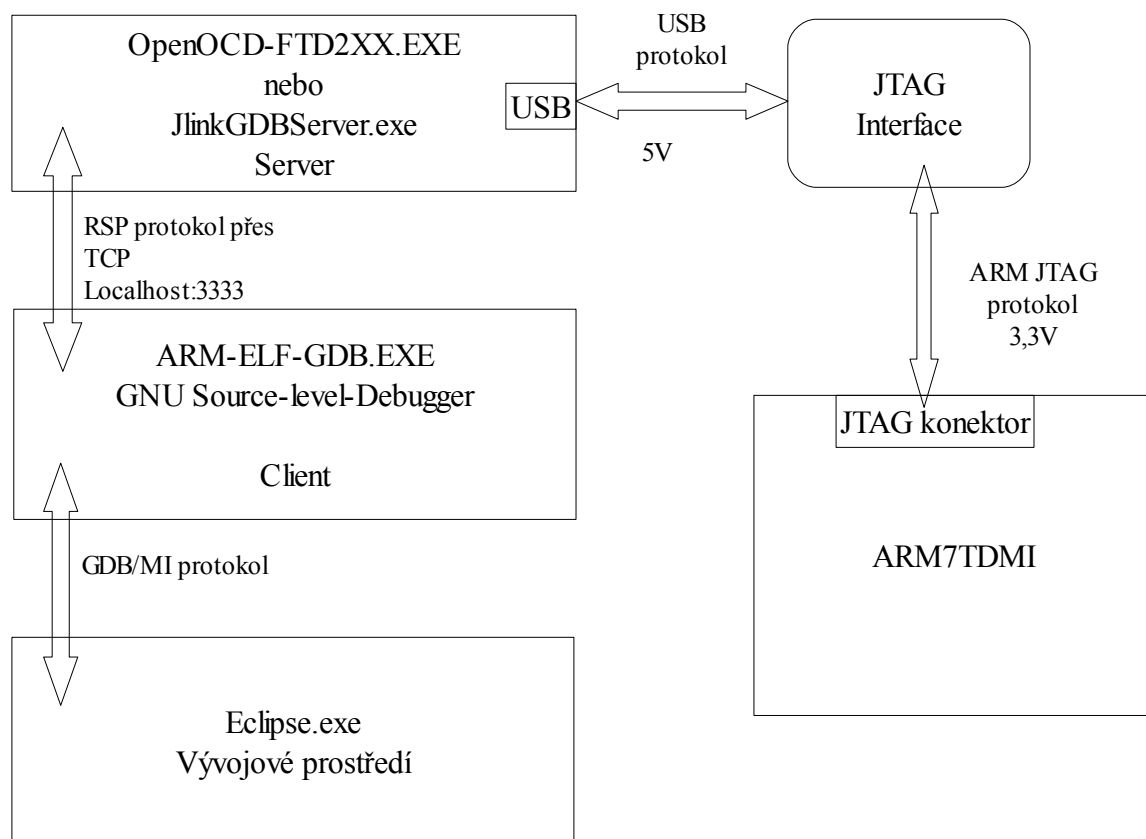
- ladící program – program na počítači (ladící nástroj)
- konvertor protokolu – slouží jako komunikační bod mezi ladícím programem a JTAG interface
- předmět ladění – jádro ARM7TDMI



Obr. 2: Příklad propojení zařízení při ladění

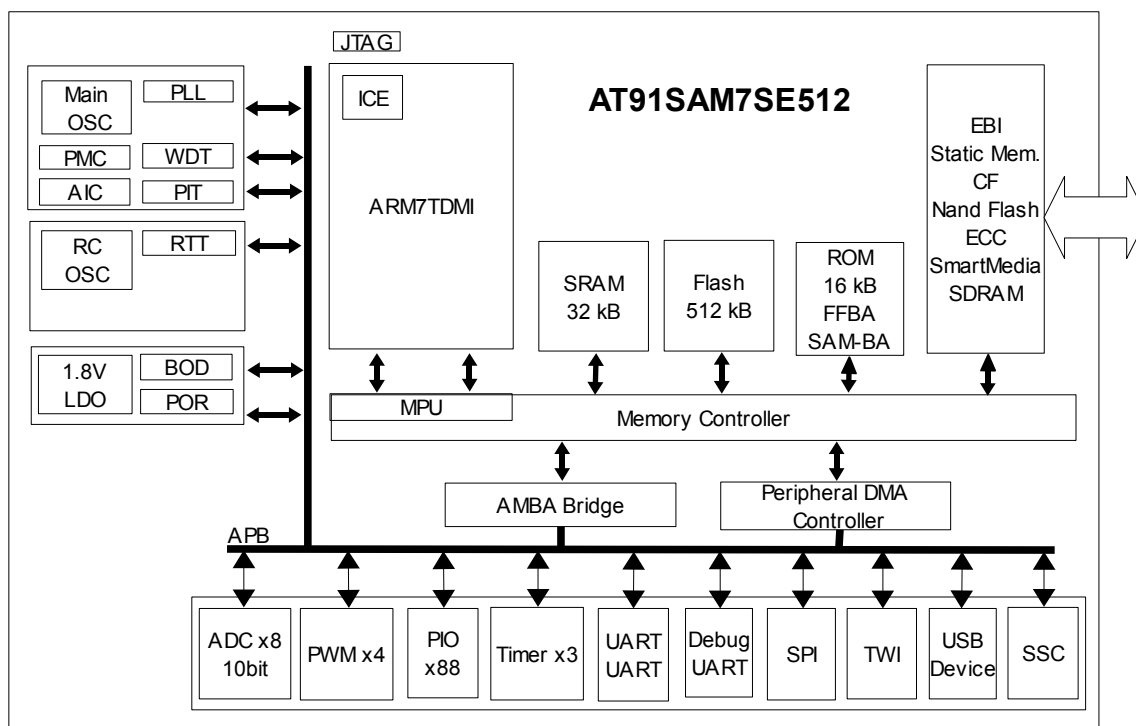
Vlastní ladění je trochu komplikovaný proces, při kterém musí mezi sebou komunikovat několik programů Obr.3. Při spuštění ladění z vývojového prostředí je spuštěna doplňková aplikace GNU GDB Source Level Debugger (arm-elf-gdb.exe) Vývojové prostředí s tímto programem komunikuje pomocí protokolu GDB/MI.

Například chceme zobrazit obsah proměnné x. Vývojové prostředí pošle žádost do GDB. GDB má přístup do souboru main.out, kde jsou uloženy informace o symbolech a instrukcích. Z tohoto souboru získá informaci o proměnné x, např. long integer na adrese 0x2006D4. GDB nyní vyše příkaz „read memory“ sériovým protokolem RSP. Tento příkaz přijme speciální serverový program, který jej přeloží do ARM JTAG protokolu a vyše do ARMu.



Obr. 3: Schéma toku dat při ladění

3 Procesor AT91SAM7SE512



Obr. 4: Blokové schéma procesoru AT91SEM7SE[6]

Pro návrh vývojového modulu byl vybrán procesor firmy Atmel AT91SAM7SE512 Obr.4 . Řada AT91SAM7SE je členem rodiny ARM procesorů založených na 32-bitových ARM7 RISC procesorech. Vybraný typ procesoru AT91SAM7SE512 obsahuje 512 kB flash paměti a 32 kB SRAM. Dále je tento procesor rozšířen o periferie:

- USB 2.0
- External Bus Interface (EBI) umožňující připojení SDRAM, NAND Flash, SmartMedia a CF karet
- USART
- SPI, I2C
- PWM
- timer
- obvod reálného času
- AD převodník
- watchdog
- vlastní Reset generátor + sledovač poklesu napájení
- JTAG rozhraní pro programování a odlad'ování programů
- SAM-BA boot program pro nahrávání programu přes USB nebo UART

4 Nástroje na vývoj

4.1 Překladače

V dnešní době je v oblasti vývojových nástrojů široká nabídka. Je možné si vybrat jak komerční produkty tak i volně dostupné. Jako příklad komerčních produktů bych zmínil balíčky firem Keil, IAR, RealView, CrossWorks, Green Hills. Produkty těchto firem nabízejí komplexní řešení vývoje bez náročného nastavování. Jejich použití je velmi snadné i díky příkladům, které výrobci zveřejňují na svých stránkách. Toto řešení má však ale velkou nevýhodu ve své finanční náročnosti. Pro první seznámení se svými nástroji umožňují někteří výrobci alespoň si stáhnout omezené verze svých programů. Omezení bývá většinou ve velikosti výsledného kódu anebo v časovém omezení.

Alternativou ke komerčním nástrojům jsou volně šiřitelné programy tzv. open-source, které nabízejí srovnatelné možnosti. Hlavním nedostatkem u tohoto řešení je neexistence komplexního balíku, který by zahrnoval jak překladač tak i vývojové prostředí, simulátor, debugger i programátor. Příklad volně dostupných nástrojů je např. GNUARM, WinARM, ARMGCC.

V současné době se na internetu dají nalézt první pokusy o vytvoření takového balíčku. Je jím YAGARTO. Na svých stránkách [7] nabízí autor ke stažení jak překladač, tak i nástroje pro ladění. Výhodou překladače YAGARTO je jeho přímá návaznost na IDE Eclipse. Nevýhoda tohoto řešení je především ve složitějším nastavování prostředí a nedostatku informací, které se ale díky internetu dají snadněji získat.

Na ukázkou je uveden výpis z krátkého programu napsaném ve volně dostupném vývojovém prostředí Eclipse. Tento program nejprve rozsvítí LED1-4 a pak v nekonečné smyčce bliká LED1.

```
/*
 * main.c
 *
 * Created on: 30.12.2008
 * Author: Ing. Vladimír Pokorný
 */

#include "AT91SAM7SE512.h"

#define LED_MASK 0x0000FF00
#define LED1      0x00000100
#define LED2      0x00000200
#define LED3      0x00000400
#define LED4      0x00000800
#define LED5      0x00001000
#define LED6      0x00002000
#define LED7      0x00004000
#define LED8      0x00008000

int main (void)
{
    UINT i;
    volatile AT91PS_PIO pPIO;

    pPIO = AT91C_BASE_PIOB; // PIO B
    pPIO->PIO_PER = LED_MASK; // PIO B povolení portu I/O;
```

```

pPIO->PIO_OER = LED_MASK;      // PIO B vybrane piny budou vystupni
pPIO->PIO_SODR= LED_MASK;      //PB8-PB15 nastavit na "1" -> LEDnesviti

pPIO->PIO_CODR = LED1;          // Rozsvitit LED1
pPIO->PIO_CODR = LED3;          // Rozsvitit LED3

CPU_Init();
// Inicializace procesoru AT91SAM7SE512 (watchdog, PLL clock, atd.)

pPIO->PIO_CODR = LED2;          // Rozsvitit LED2
pPIO->PIO_CODR = LED4;          // Rozsvitit LED4

while (1)                       //nekonecna smycka
{
    pPIO->PIO_CODR = LED1;        // rozsvitit LED1
    for( i = 0; i < 0xFFFF; i++ ); // cekej
    pPIO->PIO_SODR = LED1;        // zhasnout LED1
    for( i = 0; i < 0xFFFF; i++ ); // cekej
}
}

```

4.2 Vývojové desky

Komerční vývojové desky má ve své nabídce velká spousta výrobců. Jedná se jak o výrobce vlastních procesorů - Atmel, Philips – NXP, Freescale,....., tak i výrobce překladačů – Keil a výrobce vývojových prostředků (desek, programátorů, a pod.) – Olimex, Kramara

4.3 Programátory – debugery

Pro programování je možné využít jak vlastních prostředků procesorů (USB, UART), tak i JTAG rozhraní s možností programování a ladění. Programování pomocí USB nebo UART je možné díky bootovacímu programu (V případě procesoru Atmel AT91SAM7SE512 umístěnému v ROM paměti a mapovaném do paměti typu Flash na adrese 0x0 když 2.bit registru GPNVM je nulový). Použití JTAG rozhraní přináší výhodu v možnosti nejen programování ale hlavně v možnosti odlaďování aplikací přímo na desce. Na světě existuje mnoho výrobců, kteří vyrábějí tato rozhraní, například: Atmel, Keil, Olimex, Kramara, Wiggler, Turtelizer, ... Dříve byly velmi rozšířené programátory připojované k paralelnímu portu. Tyto programátory obsahovaly převodník LPT->JTAG. Jejich obliba byla ve své jednoduchosti a možnosti jejich rychlé stavby (standard Wiggler). V současné době začínají převládat programátory připojované k USB. I zde lze nalézt konstrukce, které lze postavit svépomocí (Turtelizer)

4.4 Použité prostředky

Ve své práci jsem se rozhodl pro použití nástroje YAGARTO[7]. Tento nástroj obsahuje GCC překladač, assembler, linker a GDB debugger. Jako další doplňky jsem zvolil integrované vývojové prostředí Eclipse a program OpenOCD pro komunikaci mezi GDB a JTAG programátorem. Pro naprogramování procesoru, bez možností ladění, využiji USB připojení (procesor obsahuje bootloader v nepřepisovatelné části paměti a pomocí programu SAM-BA lze procesor naprogramovat bez potřeby speciálního programátoru).

4.4.1 Nastavení IDE Eclipse pro použití programu SAM-BA

Po napsání programu a jeho úspěšném překladu zbývá jen tento program nahrát do paměti procesoru. K tomuto slouží program SAM-BA. Komunikace mezi procesorem a programem SAM-BA probíhá přes USB konektor. Pro příjemnější práci je možné v IDE Eclipse nastavit přímé spuštění programu SAM-BA takže vlastní naprogramování procesoru je pak jen otázkou jednoho kliknutí myši. Před vlastním nastavením je potřeba si vytvořit skriptový soubor pro program SAM-BA, který obstará vlastní automatické naprogramování procesoru. Buď si jej napíšeme sami podle vzoru a nebo si můžeme zapnout záznamník operací v programu SAM-BA, který tento skript automaticky vytvoří. V tomto případě byla zvolena druhá možnost:

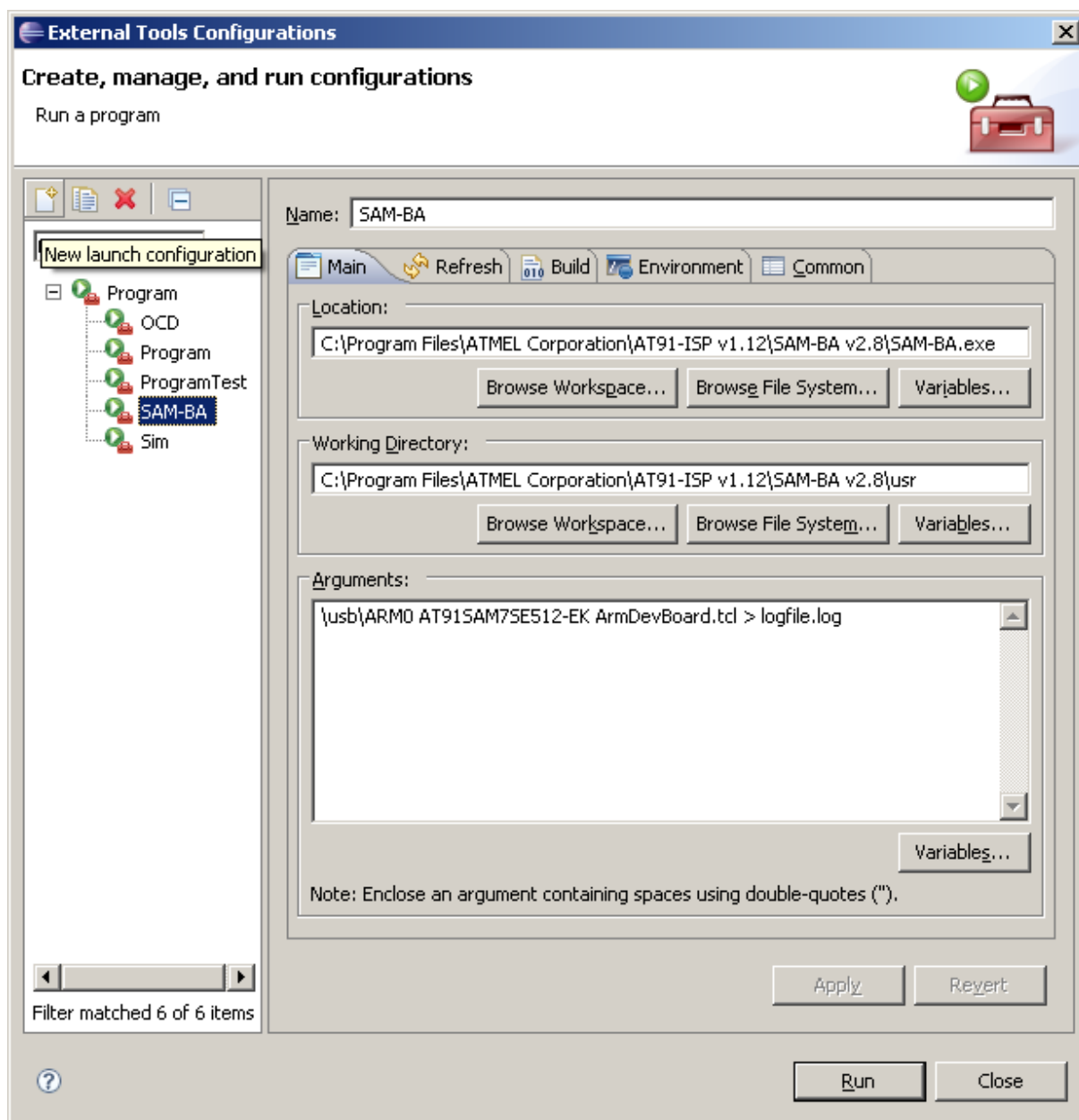
```
ArmDevBoard.tcl
#####

send_file {Flash} "C:/Documents and
Settings/Vláda/workspace/ARMDevBoard/ARMTDevBoard_rom.bin"
0x100000 0

FLASH::ScriptGPNMV 4
#####
```

Význam uvedených příkazů skriptu je celkem zřejmý. Udává, že se bude ukládat soubor ARMTDevBoard_rom.bin do paměti Flash, na adresu 0x100000 s nulovým offsetem. Na závěr je povolen BrownOut reset.

Vlastní nastavení IDE Eclipse je velmi jednoduché. V menu zvolíme „Run“, „External Tools“ a „External Tools Configuration“. Kliknutím na ikonu „stránka s plusem“ vytvoříme nový nástroj. V poli „Name“ zapíšeme jeho jméno – v případě jak je zobrazeno na Obr. 5 jsem jej pojmenoval SAM-BA. V poli „Location“ uvedeme název programu včetně cesty. V poli „Working directory“ zapíšeme adresář kde se nachází skriptový soubor pro SAM-BA. Nejdůležitější informace se nachází v poli „Arguments“. Do něj zapisujeme argumenty se kterými je program spuštěn. V případě programu SAM-BA se jedná o port připojení, typ procesoru, jméno skriptového souboru a název souboru pro záznam výsledku operací.



Obr. 5: Příklad nastavení spuštění externího programu SAM-BA v prostředí Eclipse

5 Návrh vývojové desky

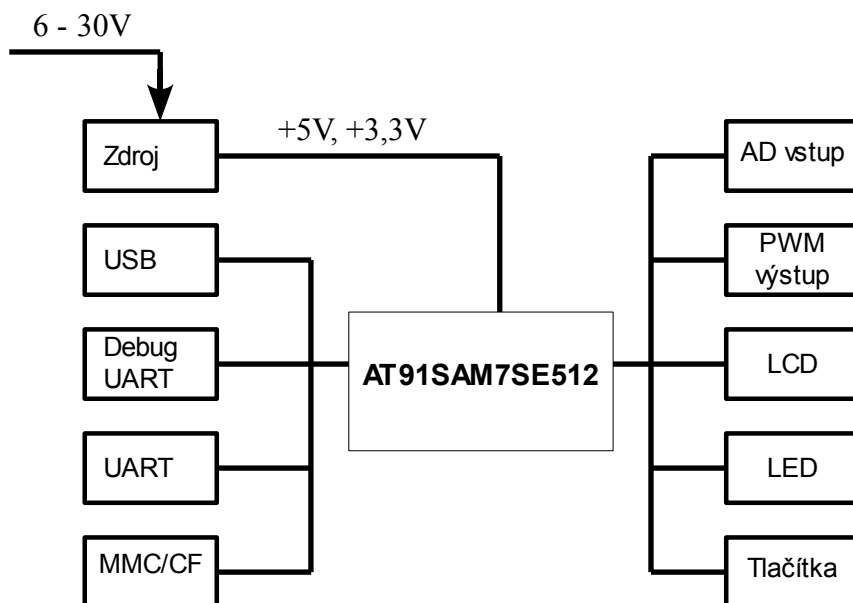
Při návrhu vývojového modulu jsem vycházel z vývojových desek firem Atmel [9] a Olimex [8].

Základní požadavky na desku jsem zvolil:

- široký rozsah napájecího napětí
- možnost programování bez speciálních programátorů
- možnost připojení rozšiřujících desek
- základní vstupní a výstupní periferie (tlačítka a LED)

K využití speciálních periférii jsem zvolil:

- MMC kartu, I2C EEPROM a teploměr
- vstup pro elektretový mikrofón (obecně nf vstup)
- nf zesilovač
- fotoodpor a trimr
- grafický LCD modul
- barevný grafický TFT LCD modul



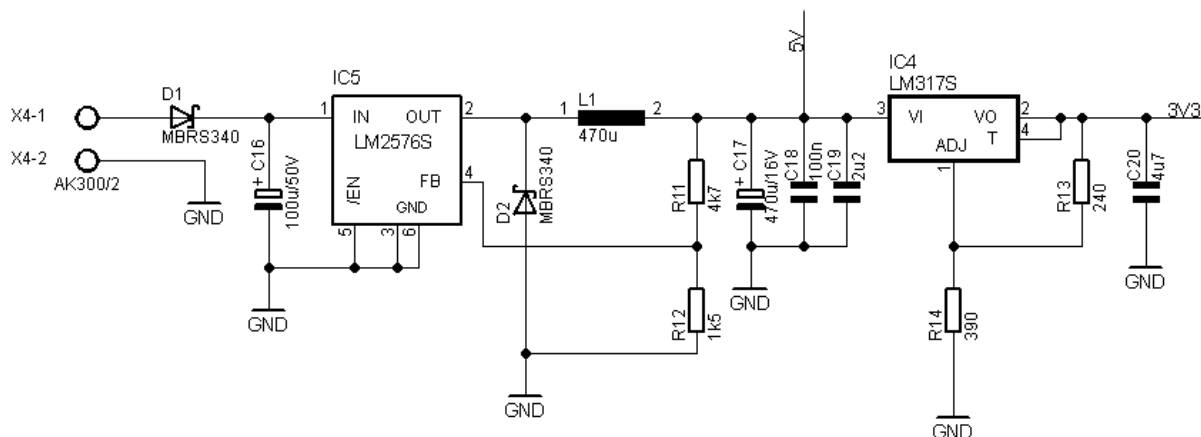
Obr. 6: Blokové schéma vývojového modulu

Kompletní návrh jsem provedl v programu Eagle v3.55. Při návrhu jsem se snažil co nejvíce používat SMD součástky. Výsledné schéma i návrh DPS je v příloze 1 a 2.

5.1 Napájecí zdroj

Na ochranu před přepólováním je za vstupním konektorem zapojena schotky dioda. Napájecí

napětí je zvoleno v rozsahu 6-30V. S ohledem na takto velký rozsah napájecího napětí, byl zvolen jednoduchý step-down spínaný zdroj. Bylo použito katalogové zapojení integrovaného obvodu LM2576S-ADJ výrobce National Semiconductors[10]. Tento obvod umožňuje sestavit s minimem vnějších součástek kompletní spínaný zdroj. Jeho výhodou je široký rozsah vstupního napětí (4-40 V) a velký výstupní proud (až 3A). Na vývojové desce je jej použit jako zdroj +5V. Pro napájení procesoru je potřeba napětí 3,3 V. Toto napětí bylo získáno z +5V použitím lineárního regulátoru LM317 viz Obr.7.



Obr. 7: Schéma zapojení napájecího zdroje +5V a +3,3V

Jádro procesoru pracuje při napětí 1,8V které si procesor vytváří sám z napětí 3,3V. Pro napájení podsvětlení TFT displeje je potřeba napětí 6,5V, které je získáváno z 3,3V pomocí step-up spínaného zdroje tvořeného obvodem LM2731Y opět v katalogovém zapojení[11]. Pro možnost vypnutí podsvětlení, je využit vstup /SHD, kterým v případě nízké úrovně lze zablokovat spínaný zdroj. Řízení kontrastu velkého grafického LCD displeje potřebuje napětí -18V. Toto napětí získáme z +5V pomocí DC/DC měniče P2AU-0518 firmy PEAK elektronik [12] .

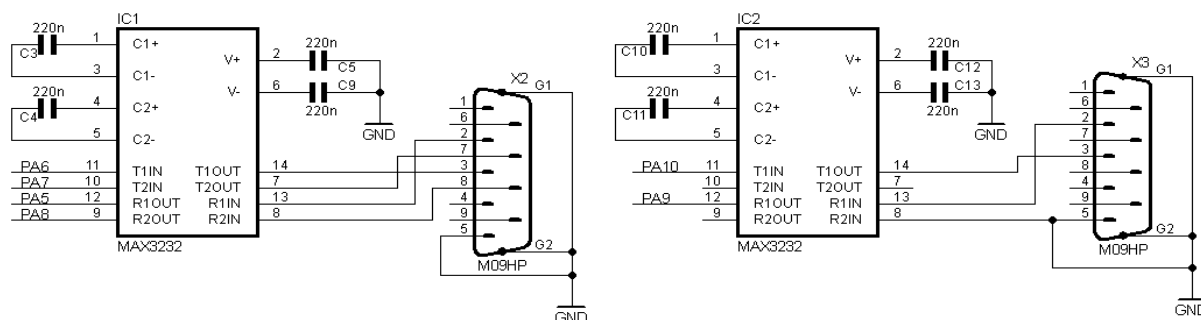
Při snaze o minimalizaci počtu zařízení potřebných k provozu vývojové desky, byla navržena i možnost napájení vývojové desky z USB konektoru. Z konektoru lze napájet zařízení až do odběru 500mA což pro první pokusy je dostatečné.

Aby bylo možné měřit spotřebu vývojového modulu, je na desce osazeno pět propojek na které se můžou připojit multimetry pro měření proudu:

- JP1 – napájení jádra (1,8V)
- JP2 – napájení I/O portů (3,3V)
- JP3 – napájení PLL (1,8V)
- JP4 – napájení procesoru (3,3V)
- JP5 – napájení desky přes USB (5V)

5.2 Komunikace s okolím

Pro styk s okolím jsou určeny 2 UART obvody, USB a JTAG. Pro použití UART obvodů je potřeba použít převodníků z úrovně 3,3V na úroveň RS232 ($\pm 12V$). Jako převodníky jsou použity obvody MAX3232 od firmy MAXIM[13] v klasickém zapojení viz Obr.8 .



Obr. 8: Schéma zapojení převodníků TTL/RS232

Konektory USB, JTAG a jeden UART obvod jsou určeny pro programování a s výjimkou USB i pro odlaďování programů.

5.3 LCD displej, TFT displej a LED

Jako zobrazovací prvky byl zvolen velký grafický monochromatický displej 240x128 bodů, malý barevný TFT displej 128x128 bodů a řadu osmi LED. Oba LCD displeje mají řízení jasu (podsvětlení) a velký displej i kontrastu, pomocí PWM signálu. Pro převod PWM signálu na DC napětí je použit jednoduchý RC článek zapojený jako dolní propust. Časová konstanta tohoto článku byla zvolena 2,2ms, z toho plyne, že perioda PWM signálu by měla být maximálně 220 us. Za DP potom následuje operační zesilovač jako sledovač. Vzhledem k vysoké ceně grafického LCD displeje, není displej osazen napevno ale přes konektory.

LED jsou zapojeny standardně tj. katody jsou spínány přes procesor k zemi. Protože každým z výstupů může protékat proud max 8 mA, byly vybrány modré vysocesvítivé LED pro proud do 20 mA. Velikost protékajícího proudu byla zvolena na 0,35 mA což při úbytku napětí na LED 2,75 V a napájecím napětí 3V znamená použít omezovací rezistor o odporu 680 Ω.

5.4 Tlačítka

Tlačítka jsou zapojeny s pull-up rezistory tj stisknutý stav je indikován nízkou úrovní. Jsou použita SMD mikrotlačítka 3FSH9 firmy Mec[14].

5.5 MMC, I2C

Na ověření sběrnice I2C jsou použity 2 obvody. EEPROM paměť 1 Mbit 24C1024 od firmy Atmel[15] a teploměr LM75 od firmy National Semiconductors [16]. Paměti EEPROM může být na sběrnici až 4 – základní adresa je 0xAX, teploměrů může být až 8 – základní adresa je 0x9X.

Pro připojení paměťové karty MMC je zvolen konektor FPS009-2405-0 od firmy YAMAICHI. Čtení a zápis do karty bude prováděn pomocí sběrnice SPI.

5.6 Analogové vstupy a výstup

Procesor obsahuje 8-mi kanálový 10-ti bitový AD převodník. Na desce jsou použity jen tři kanály. Jeden kanál jako nf vstup s možností připojení elektretového mikrofónu. Toto zapojení

je zamýšleno pro zpracování nf signálu (vzorkování, dig. filtrace, FFT a následný DA převod). Zbývající 2 kanály jsou využity pro snímání napětí na víceotáčkovém trimru a na fotorezistoru zapojeného jako dělič napětí.

Analogový výstup je vytvořen z PWM signálu – podobně jako bylo řízení jasu LCD displejů. Na zesílení výstupního nf signálu je použit nf výkonový zesilovač TDA7267 od firmy STMicroelectronics [17].

5.7 Mikroprocesor

Zapojení mikroprocesoru je dle datového listu. Hodinová frekvence připojeného krystalu je 18,432 MHz. Tato frekvence je potom vnitřními obvody upravena na 48 MHz. Za zdroj resetu byl zvolen obvod MAX809.

5.8 Rozšiřující konektory

Na rozšíření desky jsou použity tři konektory 2x20 pinů. Na konektory jsou přivedeny všechny signály z mikroprocesoru tj signály PA0-31, PB0-31, PC0-23 a AD4-7. Dále jsou na každý konektor přivedena napájecí napětí +5 , +3,3V a zem. Seznam signálů dostupných na rozšiřujících konektorech je v tabulce 4.

Tabulka 4: Seznam signálů na rozšiřujících konektorech

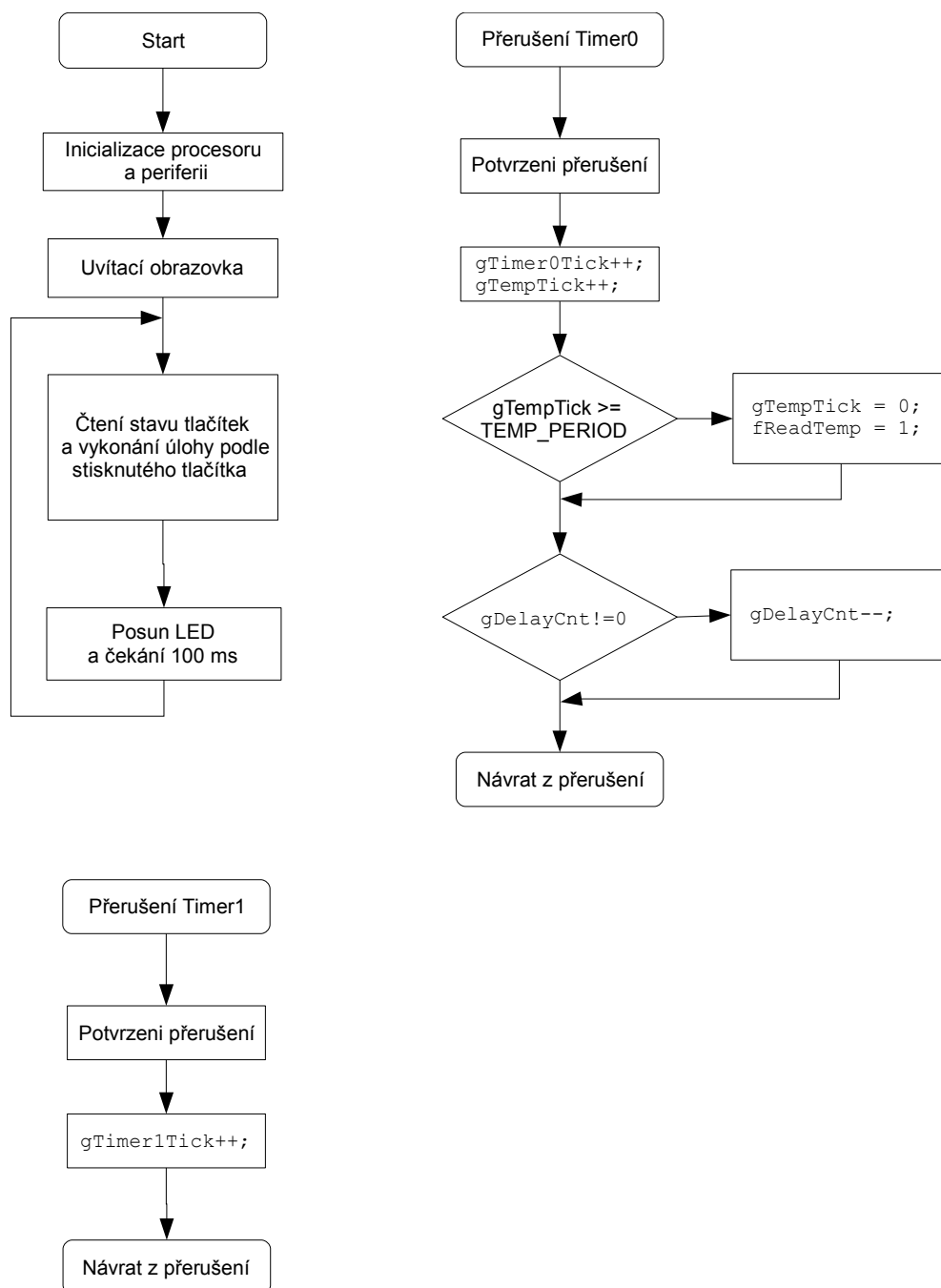
PIO	Alternativní funkce			PIO	Alternativní funkce			PIO	Alternativní funkce		
PA0	PWM0	NBS0_A0		PB0	TIOA0	A0_NBS0		PC0	D0		
PA1	PWM1	NBS2_A1		PB1	TIOB0	A1_NBS2		PC1	D1		
PA2	PWM2	A2		PB2	SCK0	A2		PC2	D2		
PA3	TWD	A3		PB3	NPCS3	A3		PC3	D3		
PA4	TWCK	A4		PB4	TCLK0	A4		PC4	D4		
PA5	RXD0	A5		PB5	NPCS3	A5		PC5	D5		
PA6	TXD0	A6		PB6	PCK0	A6		PC6	D6		
PA7	RTS0	A7		PB7	PWM3	A7		PC7	D7		
PA8	CTS0	A8		PB8	ADTRG	A8		PC8	RTS1	D8	
PA9	DRXD	A9		PB9	NPCS1	A9		PC9	DTR1	D9	
PA10	DTXD	A10		PB10	NPCS2	A10		PC10	PCK0	D10	
PA11	NPCS0	A11		PB11	PWM0	A11		PC11	PCK1	D11	
PA12	MISO	A12		PB12	PWM1	A12		PC12	PCK2	D12	
PA13	MOSI	A13		PB13	PWM2	A13		PC13	D13		
PA14	SPCK	A14		PB14	PWM3	A14		PC14	NPCS1	D14	
PA15	TF	A15		PB15	TIOA1	A15		PC15	NCS3_NANDCS	D15	
PA16	TK	BA0_A16		PB16	TIOB1	A16_BA0		PC16	NWAIT	A18	
PA17	TD	BA1_A17	AD0	PB17	PCK1	A17_BA1		PC17	NANDOE	A19	
PA18	RD	NBCS3_CFI0W	AD1	PB18	PCK2	D16		PC18	NANDWE	A20	
PA19	RK	NCS4_CFCS0	AD2	PB19	FIQ	D17		PC19	A21		
PA20	RF	NCS2_CFCS2	AD3	PB20	IRQ	D18		PC20	NCS7	A22	
PA21	RXD1	NCS6_CFCE2		PB21	PCK1	D19		PC21	NWR0_NWE_CFWE		
PA22	TXD1	NCS5_CFCE1		PB22	NPCS3	D20		PC22	NRD_CFOE		
PA23	SCK1	NWR1_NBS1_CFIOR		PB23	PWM0	D21		PC23	NCS0	CFRNW	
PA24	RTS1	SDA10		PB24	PWM1	D22					
PA25	CTS1	SDCKE		PB25	PWM2	D23					
PA26	DCD1	SDCS_NCS1		PB26	TIOA2	D24					
PA27	DTR1	SDWE		PB27	TIOB2	D25					
PA28	DSR1	CAS		PB28	TCLK1	D26					
PA29	RI1	RAS		PB29	TCLK2	D27					
PA30	IRQ1	D30		PB30	NPCS2	D28					
PA31	NPCS1	D31		PB31	PCK2	D29					

6 Zkušební program

Dalším úkolem mého projektu bylo naprogramování jednoduché aplikace, která by ověřila funkčnost navrženého modulu. V rámci řešení byly programově oživeny jen tyto periferie:

- I2C sběrnice – snímání teploty
- DBG UART – ladící textové výpisy na lince RS232
- SPI sběrnice – karta MMC a TFT displej
- AD převodník – snímání osvětlení pomocí fotoodporu a napětí na trimru
- PWM převodník – řízení podsvětlení TFT displeje
- Timer
- LED
- Tlačítka

Při psaní programu jsem čerpal informace ze zdrojů dostupných na internetu. Hlavní zdrojem byl balíček programů poskytovaný výrobcem procesoru firmou Atmel [9]. Dalším zdrojem byl velmi pěkný článek [18] popisující ovládání TFT displeje a [20] obsahující základní funkce pro zápis a čtení z karty MMC. Pro implementaci FAT16 jsem použil knihovnu [19]. Celý program byl koncipován jako nekonečná smyčka, ve které jsou testovány stavy tlačítek a podle stavu tlačítek jsou vykonávány jednotlivé úkoly. Názorně to vyjadřuje vývojový diagram na Obr.9 .



Obr. 9: Vývojový diagram zkušebního programu

6.1 Nastavení procesoru

Po připojení napájení je spuštěn vnitřní oscilátor o frekvenci přibližně 32 kHz. Pro přepnutí na hlavní oscilátor je třeba nastavit příslušné registry. Povolení hlavního oscilátoru se provede nastavením bitu MOSCEN v registru PMC_MOR

```
PMC_MOR = ( AT91C_CKGR_OSCOUNT & (8 << 8)) | AT91C_CKGR_MOSCEN ;
```

Tímto příkazem je nastaven hlavní oscilátor a zároveň je nastaveno jeho zapnutí po 64

pomalých pulsech (přibližně 1,9 ms).

Nyní je zapnut hlavní oscilátor. Jeho kmitočet je dán připojeným externím krystalem. V tomto případě 18,430 MHz. Pro nastavení vyšší frekvence je v procesoru přítomen PLL závěs. S pomocí PLL je možné měnit kmitočet v širokých mezích (max. 220 MHz). K nastavení PLL je třeba natavit vstupní děličku a PLL násobičku. S ohledem na plánované použití USB je také nutno nastavit výstupní děličku (USB pracuje pouze na 48 MHz).

```
PMC_PLLR = AT91C_CKGR_USBDIV_1      // USB-CLK = MCK/2 (96/2 = 48 MHz)
           | AT91C_CKGR_OUT_0
           | (AT91C_CKGR_PLLCOUNT & (40 << 8))
           | (AT91C_CKGR_MUL & (124 << 16)) // PLL-OUT = 768000*125 = 96 MHz
           | (AT91C_CKGR_DIV & 24);    // PLL-IN = (18432000/24 = 768kHz
```

Nyní je hlavní oscilátor nastaven na 48MHz.

6.2 Nastavení SPI sběrnice

SPI řadič v procesoru umožňuje komunikaci až 4 zařízení při přímém připojení anebo až 16 zřízení při připojení přes dekodér výběrového signálu. Na desce je zvoleno přímé připojení. Povolení SPI komunikace se provede nastavením bitu SPIEN v registru SPI_CR, deaktivování nastavením bitu SPIDIS. Pokud jsou oba bity nastaveny, pak je SPI komunikace vypnuta. Volba módu SPI, typu CS, zpoždění mezi dvěma CS se provádí zapsáním příslušných hodnot do registru SPI_MR. Rychlost komunikace každého zařízení může být různá stejně tak i neaktivní úroveň signálu SPCK a okamžik platnosti dat. Tyto parametry se nastavují registry SPI_CSR[0..3]. V případě, že je zvoleno připojení zařízení přes dekodér signálu CS, platí nastavení pro danou čtveřici zařízení tj SPI_CSR[0] pro CS0..CS3, SPI_CSR[1] pro CS4..CS7 atd. Na krátkém výpisu zdrojového kódu je ukázáno nastavení SPI komunikace použité ve vzorové aplikaci.

```
SPI_CR      = 0x81;                //Povolení SPI, Software reset

SPI_MR      =
(AT91C_SPI_DLYBCS & (0xF0 << 24))| // Zpoždění mezi dvěma CS (5 us)
(AT91C_SPI_MODFDIS & (0x1 << 4))|  // Deaktivace dekodování chyby
(AT91C_SPI_PS & (0x1 << 1))|       // Proměnný CS
(AT91C_SPI_MSTR & (0x01 << 0));    // Mod Master

SPI_CSR[0] =
(AT91C_SPI_DLYBCT & (0x02 << 24))| // Zpoždění mezi 2 přenosy (1,33us)
(AT91C_SPI_DLYBS & (0x01 << 16))|  // Zpoždění mezi CS a SPCK ( 21 ns )
(AT91C_SPI_SCBR & (0x02 << 8))|    // Přenosová rychlost = 24 Mbaud
(AT91C_SPI_BITS & (AT91C_SPI_BITS_9))| // Délka dat (9 bitů)
(AT91C_SPI_CSAAT & (0x0 << 3))|    // Uvolnění CS po dokončení přenosu.
(AT91C_SPI_NCPHA & (0x0 << 1))|    // Data platná při 2. hraně SPCK
(AT91C_SPI_CPOL & (0x01 << 0));    // Neaktivní stav SCK při logické 1

SPI_CSR[1] =
(AT91C_SPI_DLYBCT & (0x04 << 24))| // Zpoždění mezi dvěma přenosy (2,7us)
(AT91C_SPI_DLYBS & (0x30 << 16))|  // Zpoždění mezi CS a SPCK (1 us )
(AT91C_SPI_SCBR & (0xF0 << 8))|    // Přenosová rychlost = 250 kbaud
(AT91C_SPI_BITS & (AT91C_SPI_BITS_8))| // počet přenesených bitů (8 bitů)
```

```

(AT91C_SPI_CSAAT & (0x1 << 3))|    // CS aktivni dokud neni jiny CS
(AT91C_SPI_NCPHA & (0x1 << 1))|    // Data platna pri 1. hraně
(AT91C_SPI_CPOL & (0x0 << 0));    // Neaktivni stav SPCK pri logicke 0

SPI_CR      = 0x01;                //Povoleni SPI

```

Volba parametru komunikace a tím i typ zařízení se kterým se bude komunikovat se provádí při zápisu posílaného byte do registru SPI_TDR, kde posílaná data jsou v prvních 16-ti bitech a adresa zařízení v bitech 16-19 viz ukázka zdrojového kódu.

```

//Posli data
// pockej dokud SPI_TDR a posuvny registr nejsou prazdny
while ((s_pSpi->SPI_SR & AT91C_SPI_TXEMPTY) == 0);
// Do registru zapiseme posilana data + adresu zarizeni
s_pSpi->SPI_TDR = (data & 0xFFFF) | (MMC_CS << 16 );
//pockej dokud data z SPI_TDR byla predana k vyslani
while ((s_pSpi->SPI_SR & AT91C_SPI_TDRE) == 0);

// Cekej dokud neni prijimaci registr plny
while((s_pSpi->SPI_SR & AT91C_SPI_RDRF) == 0);
// Precti prijaty znak
spib = ((s_pSpi->SPI_RDR) & 0xFFFF);

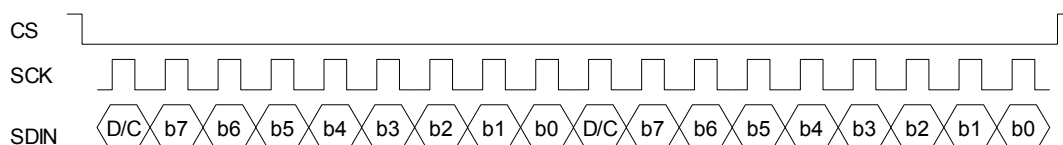
```

6.3 Ovládání TFT displeje

Jednou z periférií, která byla zvolena v tomto vývojovém modulu je TFT displej. Ze široké nabídky byl vybrán typ z mobilního telefonu Nokia 6100. K volbě tohoto typu bylo přistoupeno vzhledem k jeho dostupnosti jako náhradní díl i spolu s miniaturním konektorem ale také i pro dostatek informací k tomuto displeji. Uvedený displej má rozlišení 132x132 bodů, 12 bitů pro barvu (RGB, na každou složku 4 bity), napájení 3,3V a 9-ti bitovou SPI komunikaci.

6.3.1 SPI

Jedinou nepříjemností při psaní ovládacích procedur je fakt, že řídicím driverem může být typ PCF8833 od firmy Philips(nyní NXP)a nebo typ S1D15G00 od firmy EPSON. V mém případě byl v displeji použit driver PCF8833. Jak již bylo zmíněno dříve, komunikace probíhá po SPI sběrnici s délkou 9 bitů. 9. bit indikuje zda je posílán příkaz nebo data. Přenos začíná 9. bitem. Maximální frekvence hodinového signálu může být až 6 MHz (Při testech byla ověřena funkčnost komunikace i při frekvenci hodinového signálu 24 MHz).



Obr. 10: Průběh SPI komunikace - posílání příkazu a dat

6.3.2 Adresování pixelů

Při zápisu jednotlivých bodů displeje je důležité znát vnitřní uspořádání displeje. Řadič

PCF8833 obsahuje vnitřní paměť o velikosti 17424 (132x132) slov. Každé slovo má velikost 12 bitů (4 bity červená, 4 bity zelená a 4 bity modrá složka). Požadovaná adresa bodu se specifikuje příkazy PAGE ADDRESS SET (řádek) a COLUMN ADDRESS SET (sloupec). Oba příkazy udávají dvě věci - počáteční a koncový bod. To má význam při kreslení pravoúhlých objektů. Při využití příkazů WRAP – ROUND a automatické inkrementace můžeme velmi snadno zobrazovat znaky a vyplňovat obdélníky.

6.3.4 Řízení barev pixelů

Řadič PCF8833 umožňuje specifikovat barvu bodu třemi způsoby:

- 12 bitů na pixel
- 8 bitů na pixel
- 16 bitů na pixel

Ve vzorové aplikaci byl použit mód **12 bitů na pixel**. Tento mód se zapne posláním příkazu 0x3A (Color Interface Pixel Format) a následně jeden datový byte s hodnotou 3. Pro zapsání dat specifikující barvu pixelu je třeba poslat do displeje 2,5 byte viz tabulka 5.

Tabulka 5: Příklad zápisu barev dvou pixelů

0	0	1	0	1	1	0	0	příkaz RAMWR
R	R	R	R	G	G	G	G	červená a zelená složka 1 pixelu
B	B	B	B	R	R	R	R	modrá složka 1. pixelu a červená 2. pixelu.
G	G	G	G	B	B	B	B	zelená a modrá složka 2. pixelu

Pokud zapíšeme jen dva byte tzn. u druhého pixelu jen červenou složku, řadič druhý pixel ponechá beze změny. Změnu provede jen při obdržení kompletních dat.

6.3.5 Inicializace

Po zapnutí napájení je třeba provést inicializaci displeje. První, co musíme udělat je provést hardwarový reset, pak je nutno displej probudit, nastavit provozní mód, typ přístupu k paměti kontrast a zapnout displej. Příklad inicializace je v následujícím výpisu zdrojového kódu a Obr. 11.

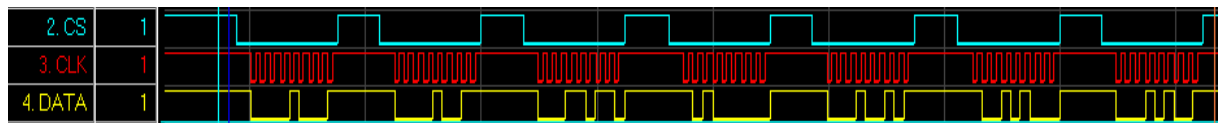
```
// Hardware reset
LCD_RESET_LOW;
Delaya(2000);
LCD_RESET_HIGH;
Delaya(2000);

///// Inicializace /////
WriteSpiCommand(SLEEP_OUT); // probuzeni displeje (0x11)
WriteSpiCommand(NORMAL_MODE); // nastaven normalni mod (0x13)
WriteSpiCommand(ACCESS_CTRL); // nastaveni pristupu k pameti (0x36)
WriteSpiData(0x40); // prevraceni x
//WriteSpiData(0x20); // prevraceni y

WriteSpiCommand(CONTRAST); // nastaveni kontrastu (0x25)
```

```
WriteSpiData(0x28); // kontrast nastaven na +1600 mV Voffset LCD

// Displej je nastaven
WriteSpiCommand(DISPLAYON); // Zapnutí displeje (0x29)
```



Obr. 11: Průběhy SPI signálů při nastavení TFT displeje - výstup logického analyzátoru

6.4 Ovládání karty MMC

Další z periférií použitých ve vývojovém modulu je paměťová karta. V současné době jsou paměťové karty využívány jako paměťové médium v digitálních fotoaparátech, některých videokamerách, mobilních telefonech, MP3 přehrávačích atd. Kapacita těchto karet neustále stoupá. Nyní lze běžně koupit karty o kapacitě až 32 GB. Ve vývojovém modulu byla použita karta typu MMC o velikosti 512MB.

6.4.1 Komunikace s kartou

Karta MMC může komunikovat s procesorem ve dvou režimech. Buď v režimu MMC, kdy komunikace běží po třech vodičích (DAT \leftrightarrow , CMD \leftrightarrow a CLK) a v režimu SPI kdy se komunikuje po 4 vodičích (DataIn, DataOut, CLK a CS). S ohledem na použití i karet SD, které komunikují jen SPI protokolem, byla zvolena SPI komunikace. Nastavení módu SPI se provede vysláním příkazu CMD0 (GO_IDLE_STATE - resetování karty) při stavu CS signálu v nízké úrovni. Po resetu se karta nachází v IDLE stavu ze kterého se dostane vysláním příkazu CMD1 (SEND_OP_COND – zapnutí normálního stavu). Tento příkaz je třeba vyslat několikrát, dokud karta nevrátí hodnotu 0 – viz příklad zdrojového kódu. Tím je karta nastavena.

```
for(i=0;i<10;i++)
    spiSendByte(0xff, 0); // Vysleme 80 pulsu na linku CLK

response = 0;
for( i = 0; i < 5; i++)
{
    mmcSendCmd(MMC_GO_IDLE_STATE,0,0x95);
    // provedeme RESET karty, pri CS = 0,
    // se karta nastavi do SPI modu

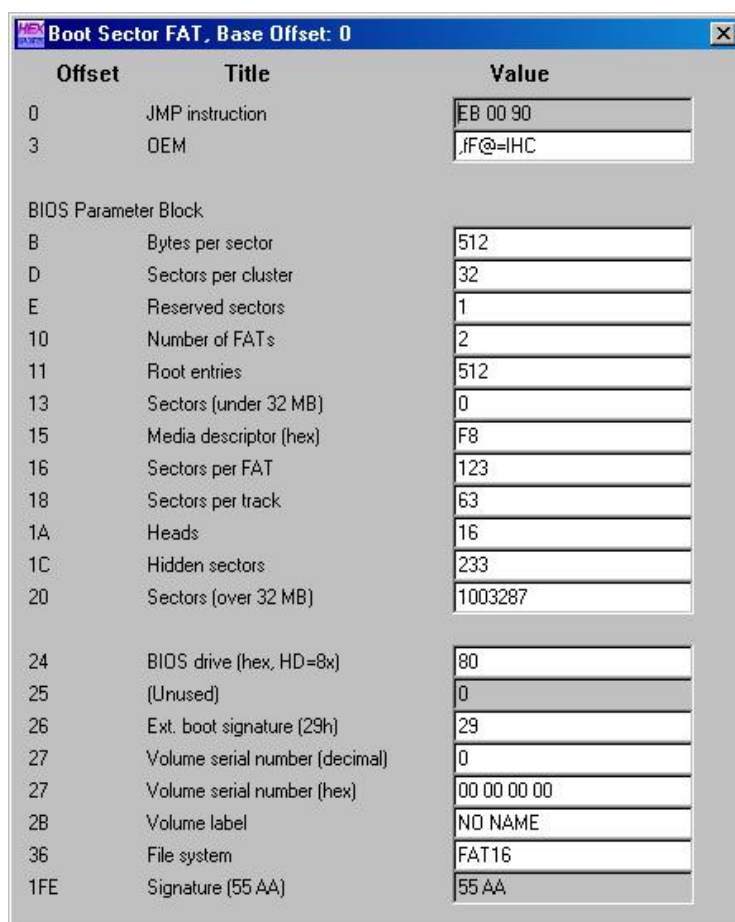
    response = mmcGetResponse();
    TRACE_MMC("response = 0x%X \r\n", response);
    // pomocny vypis na terminal

    Delay( 100);
}
response = 0;
while(response==0x01)
{
    TRACE_MMC("Sending Command 1\r\n");
    spiSendByte(0xff,0);
    mmcSendCmd(MMC_SEND_OP_COND,0x00,0xff); // prepnutí do normal. stavu
    response=mmcGetResponse();
    TRACE_MMC("response = 0x%X \r\n", response);
}
```

}

6.4.2 Souborový systém FAT 16

Při použití paměťové karty, bylo zamýšleno i její případné čtení na stolním počítači s instalovanou čtečkou karet. Proto byl zvolen systém FAT16. Po inicializaci karty je nutné inicializovat i FAT tabulku, tzn. zjistit Boot sector, počet FAT tabulek, pozici adresářového sektoru, datového sektoru atd. Na internetu lze nalézt velké množství řešení, od jednoduchých, které umožňují pouze čtení, přes středně složitá, které již mají i zápis až po velmi složitá, které umožňují nejen čtení a zápis, ale i přidávání, mazání vyhledávání apod. Jako výchozí řešení bylo vybráno řešení z [19] popisující implementaci FAT16 v MCU typu MSP430F449. Při úpravě tohoto řešení velmi pomohl program WinHex, který umožňuje zobrazit obsahy jednotlivých sektorů na disku a dokonce rozkódovat sektor na strukturu viz Obr. 11



Offset	Title	Value
0	JMP instruction	EB 00 90
3	OEM	.IF@=IHC
BIOS Parameter Block		
B	Bytes per sector	512
D	Sectors per cluster	32
E	Reserved sectors	1
10	Number of FATs	2
11	Root entries	512
13	Sectors (under 32 MB)	0
15	Media descriptor (hex)	F8
16	Sectors per FAT	123
18	Sectors per track	63
1A	Heads	16
1C	Hidden sectors	233
20	Sectors (over 32 MB)	1003287
24	BIOS drive (hex, HD=8x)	80
25	(Unused)	0
26	Ext. boot signature (29h)	29
27	Volume serial number (decimal)	0
27	Volume serial number (hex)	00 00 00 00
2B	Volume label	NO NAME
36	File system	FAT16
1FE	Signature (55 AA)	55 AA

Obr. 12: Přečtená struktura Boot Sektoru na paměťové kartě

V ukázce zdrojového kódu je zobrazena inicializace souborového systému.

```
TRACE_FAT("FATInit\r\n");
FATLibInit();
if(mmcSetBlockLength(512)== MMC_SUCCESS) //nastavení velikosti bloku
```



```

        sectorZero = getPartitionOffset(); //zjisteni hodnoty 0. log. sektoru
else
    return 3;
identifyFileSystem(buf); //identifikace souboroveho systemu
buf[0] = readFileSystemInfo(buf); //zjisteni parametru soub.systemu

```

6.5 Nastavení sériové linky pro ladění

Pokud chceme využít periferie DBGU – Debug Unit je nutné ji správně nastavit. Nejprve je potřeba přiřadit piny procesoru k této jednotce tzn. nastavit příslušné bity v registru PDR, po té resetovat přijímač a vysílač, nastavit parametry přenosu a nakonec uvolnit přijímač a vysílač. Tyto kroky jsou názorně ukázány na následujícím zdrojovém kódu.

```

*AT91C_PIOA_PDR =   DBG_MASK; // Povoleni pinu DRxD
                        // Povoleni pinu DTxD
pDBG->DBGU_CR =   AT91C_US_RSTRX | // Resetovani prijimace
                  AT91C_US_RSTTX | // Resetovani vysilace
                  AT91C_US_RXDIS | // Prijem zakazan
                  AT91C_US_TXDIS; // Vysilani zakazano

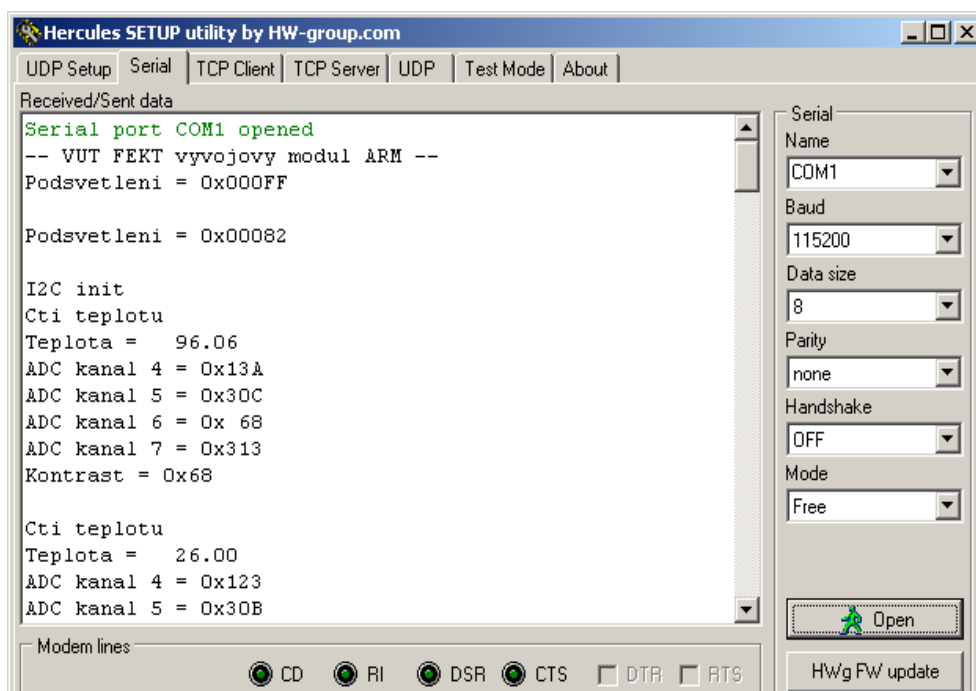
pDBG->DBGU_MR =   AT91C_US_USMODE_NORMAL | // Nastaveni normalniho modu
                  AT91C_US_CLKS_CLOCK | // Zdroj hodin je MCK
                  AT91C_US_CHRL_8_BITS | // 8-bitova data
                  AT91C_US_PAR_NONE | // Bez parity
                  AT91C_US_NBSTOP_1_BIT; // 1 Stop bit

pDBG->DBGU_BRGR = BAUDRATE_DIV; // Baudrate delicka -115200 baud/s

pDBG->DBGU_CR =   AT91C_US_RXEN | // Povoleni Prijmu a vysilani
                  AT91C_US_TXEN; //

```

Poslání znaku je pak jednoduché, stačí zapsat posílaný byte do registru DBGU_THR.



Obr. 13: Záznam komunikace z jednotky DBGU

6.6 Nastavení časovačů

Důležitou součástí každého procesoru jsou časovače a čítače. Procesor AT91SAM7SE512 obsahuje celkem tři 16-ti bitové časovače/čítače. V programu jsou využity dva: TC0 a TC1. Časovač TC0 je nastaven na generování přerušení každou milisekundu a časovač TC1 na 44 ms. Nastavení časovače TC0 je ukázáno v následující ukázce zdrojového kódu.

```
// Nastaveni timeru 0
pPMC = AT91C_BASE_PMC; // ukazatel na PMC strukturu
pPMC->PMC_PCER = (1<<AT91C_ID_TC0); // Povoleni periferie Timer0

// Deaktivovani timeru0 a preruseni
pTC = AT91C_BASE_TC0;
pTC->TC_CCR = AT91C_TC_CLKDIS;
pTC->TC_IDR = 0xFFFFFFFF;

dummy = pTC->TC_SR; // nulovani stavoveho bitu
dummy = dummy; // Odstraneni varovani: "dummy" was set but never
used

// Nastaveni modu casovace
pTC->TC_CMR = TC_CLKS_MCK32| // Delicka vstupniho signalu MCK (MCK/32)
             WAVESEL2| // casovac je inkrementovan od 0 do RC
             WAVE; // Nastaveni modu casovace
pTC->TC_RC = 1500; //preruseni je vyvolano kazdou 1 ms
                //( 48000000/32/1500 = 1000 Hz )
pTC->TC_CCR = AT91C_TC_CLKEN; // Aktivovani hodin - spusteni
```

6.7 Nastavení přerušení

V ověřovací aplikaci je využito přerušení od časovačů TC0 a TC1. Před spuštěním časovačů je nutno nastavit parametry pro obsluhu přerušení. Především je potřeba u vektoru přerušení nastavit ukazatel na funkci obsluhující přerušení, nastavit prioritu a typ přerušení. Nastavení přerušení pro časovač TC0 je opět v ukázce zdrojového kódu:

```
//Nastaveni obsluhy preruseni
oldHandler = pAIC->AIC_SVR[AT91C_ID_TC0];
mask = 0x1 << AT91C_ID_TC0;
//Deaktivovani daneho vektoru preruseni
pAIC->AIC_IDCR = mask;
//Ulozeni ukazatele na obsluhu preruseni
pAIC->AIC_SVR[AT91C_ID_TC0] = (unsigned int) timer0_c_irq_handler;
// Nastaveni priority a typu preruseni
pAIC->AIC_SMR[AT91C_ID_TC0] = AT91C_AIC_SRCTYPE_INT_HIGH_LEVEL |
                             TIMER0_INTERRUPT_LEVEL;

//Nulovani daneho preruseni
pAIC->AIC_ICCR = mask;

AT91C_BASE_TC0->TC_IER = AT91C_TC_CPCS; // IRQ enable CPC
pAIC->AIC_IECR = 0x1 << AT91C_ID_TC0;
```

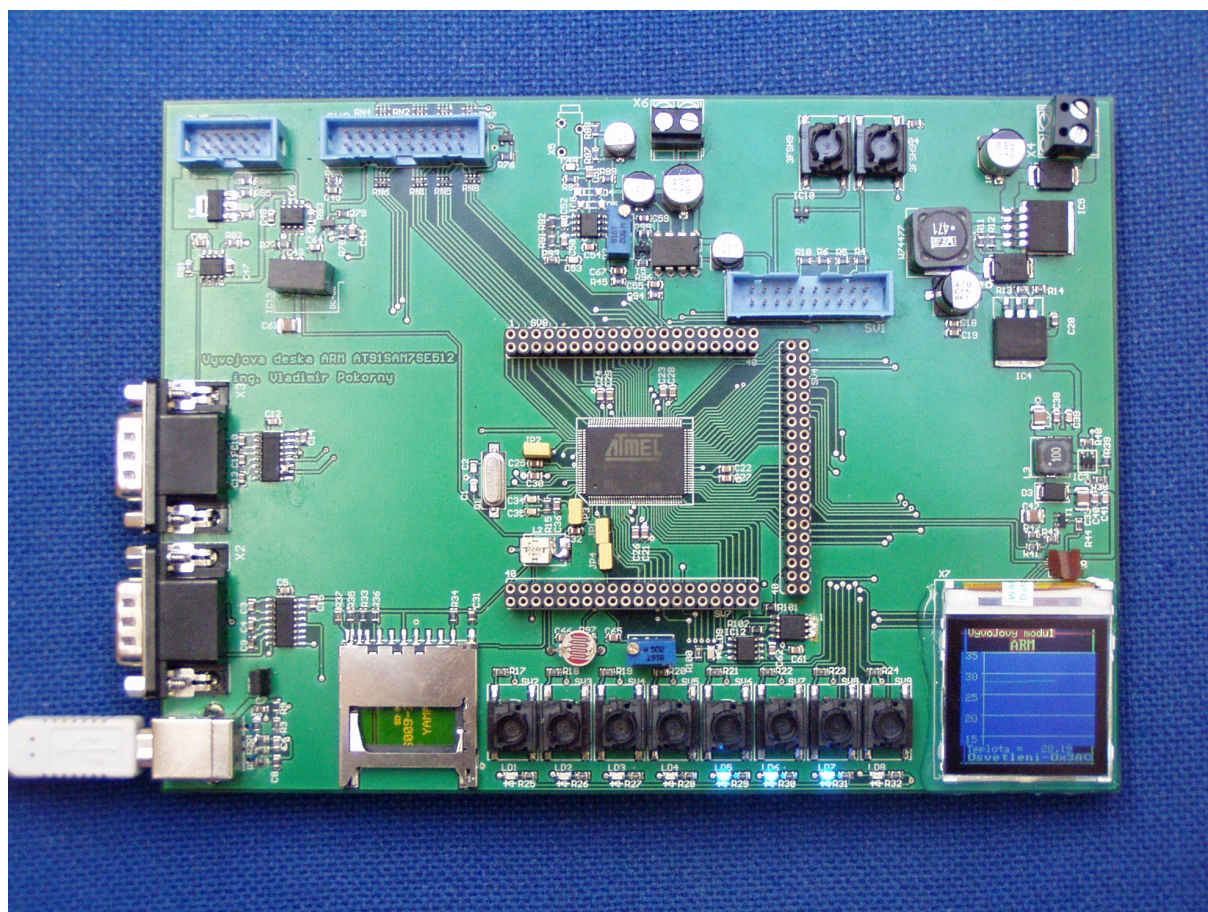
7 Závěr

Na základě zadání bakalářské práce, bylo navrženo zapojení vývojového modulu pro 32 bitové procesory typu ARM. Zároveň byl i vytvořen návrh desky s plošnými spoji o rozměrech 195x142 mm. Vzhledem k velkému počtu spojů je navržena deska čtyřvrstvá. Deska byla následně osazena a oživena. Součástí tohoto projektu bylo i naprogramování jednoduché aplikace využívající periferie procesoru, např. SPI, PWM, TWI(I2C), ADC.

Použité volně dostupné vývojové nástroje, Eclipse a YAGARTO, se přes počáteční problémy s nastavením osvědčily jako velmi vhodné programy pro vývoj aplikací určené pro procesory ARM.

Velkou výhodou tohoto modulu je, že k práci není potřeba dalšího zařízení jako je např. napájecí zdroj a programátor, obě tyto funkce obstará USB připojení - jen pro zajímavost, spotřeba modulu v okamžiku, kdy jsou rozsvíceny všechny LED a TFT displej je 320 mA, spotřeba jádra ARM7TDMI je 26 mA a spotřeba I/O portů a PLL je 3 mA.

Tímto modulem se dostává do rukou všem zájemcům, kteří se chtějí seznámit s 32bitovým procesorem ARM, velmi vhodný nástroj na experimenty s tímto typem procesoru. Díky rozšiřujícím konektorům není modul vázán jen na periferie přítomné na desce ale s jejich pomocí lze modul rozšířit o další periferie např. SDRAM, NAND, CF apod.



Obr. 14: Fotografie vyrobeného modulu za provozu

8 Seznam literatury

[1] MURRAY, Richard *Procesor types* [online]. 2004 [cit. 20. prosince 2008]. Dostupné na WWW: <http://www.heyrick.co.uk/assembler/proctype.html>

[2] MCU Server. *Atmel AT91SAM7X* [online]. 2008 [cit. 20. prosince 2008]. Dostupné na WWW: <http://www.mcu.cz/news.php?extend.1058.3>

[3] Wikipedia. *ARM architecture* [online]. 2008 [cit. 20. prosince 2008]. Dostupné na WWW: http://en.wikipedia.org/wiki/ARM_architecture :

[4] ARM Limited. *ARM Architecture Reference Manual* [online]. 2005 [cit. 20. prosince 2008]. Dostupné na WWW: <http://www.arm.com/miscPDFs/14128.pdf>

[5] Arm Limited. *ARM7TDMI (Rev 3) Core Processor* [online]. 2001 [cit. 20. prosince 2008]. Dostupné na WWW: http://infocenter.arm.com/help/topic/com.arm.doc.dvi0027b/_DVI0027A_ARM7TDMIPO.pdf

[6] MSC. *Die SAM7S und SAM7SE-Serie* [online] [cit. 20. prosince 2008]. Dostupné na WWW: <http://www.msc-ge.com/de/1771-www.html>

[7] YAGARTO. *Yet another GNU ARM toolchain*. [online]. 2008[cit. 20. prosince 2008]. Dostupné na WWW: <http://www.yagarto.de/>

[8] Olimex Ltd.. *SAM7-EX256 DEVELOPMENT BOARD FOR AT91SAM7X256 ARM7TDMI-S MICROCONTROLLER* [online]. 2007[cit. 20. prosince 2008]. Dostupné na WWW: <http://www.olimex.com/dev/images/ARM/ATMEL/SAM7-EX256-REV-Bsch.gif>

[9] Atmel Corporation. *AT91SAM7SE-EK Hardware files* [online]. 2006[cit. 20. prosince 2008]. Dostupné na WWW: http://www.atmel.com/dyn/resources/prod_documents/AT91SAM7SE-EK_Hardware_files.zip

[10] National Semiconductors. *LM2576/LM2576HV Series SIMPLE SWITCHER® 3A Step-Down Voltage Regulator* [online]. 2004[cit. 20. prosince 2008]. Dostupné na WWW: <http://cache.national.com/ds/LM/LM2576.pdf>

[11] National Semiconductors. *LM2731 0.6/1.6 MHz Boost Converters With 22V Internal FET Switch in SOT-23* [online]. 2007[cit. 20. prosince 2008]. Dostupné na WWW: <http://cache.national.com/ds/LM/LM2731.pdf>

[12] PEAK electronics. *P2AU-XXXXE 1kV ISOLATED 0,25W UNREGULATED SINGLE OUTPUT SIP4*[online]. 2001[cit. 20. prosince 2008]. Dostupné na WWW: <http://www.peak-electronics-gmbh.de/data/ap2auxe.pdf>

[13] Maxim Integrated Products. *3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1μF External Capacitors* [online]. 2007[cit. 20. prosince 2008]. Dostupné na WWW: <http://datasheets.maxim-ic.com/en/ds/MAX3222-MAX3241.pdf>

- [14] MEC a/s. *3F + IGA/IGC* [online]. 2006[cit. 20. prosince 2008]. Dostupné na WWW: <http://www.mec.dk/files/webGA-GC.pdf>
- [15] Atmel Corporation. *Two-wire serial EEPROM 1M (131,072 x 8) AT24C1024B with Two Device Address Inputs* [online]. 2008[cit. 20. prosince 2008]. Dostupné na WWW: http://www.atmel.com/dyn/resources/prod_documents/doc5194.pdf
- [16] National Semiconductors. *LM75 Digital Temperature Sensor and Thermal Watchdog with Two-Wire Interface* [online]. 2005[cit. 20. prosince 2008]. Dostupné na WWW: <http://cache.national.com/ds/LM/LM75.pdf>
- [17] STMicroelectronics. *TDA7267 2W MONO AMPLIFIER* [online]. 1998[cit. 20. prosince 2008]. Dostupné na WWW: <http://www.st.com/stonline/products/literature/ds/1569/tda7267.pdf>
- [18] LYNCH, James P., *Nokia 6100 LCD Display Driver* [online]. 2007 [cit. 15. března 2009]. Dostupné na WWW: <http://www.sparkfun.com/tutorial/Nokia%206100%20LCD%20Display%20Driver.pdf>
- [19] SHAM, I. HUE, W. RIZUN, P. *Portable FAT Library for MCU Applications*. Circuit Cellar, March 2005, no. 176, p. 18-27. [cit. 15. března 2009]. Source code: ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2005/176/Sham176.zip
- [20] OLIMEX, Demo code LCD write, Joystick and buttons read [online]. 2007 [cit. 15. března 2009]. Dostupné na WWW: <http://www.olimex.com/dev/soft/arm/SAM7/SAM7EX256.zip>
- [21] ARM, ARM10, *ARM10E Family*[online]. 2009 - [cit. 20.května 2009]. Dostupné na WWW: <http://www.arm.com/products/CPUs/families/ARM10EFamily.html>
- [22] ARM, ARM11, *ARM11 Family*[online]. 2009 - [cit. 20.května 2009]. Dostupné na WWW: <http://www.arm.com/products/CPUs/families/ARM11Family.html>
- [23] ARM, Cortex , *ARM Cortex-M3*[online]. 2009 - [cit. 20.května 2009]. Dostupné na WWW: http://www.arm.com/products/CPUs/ARM_Cortex-M3.html

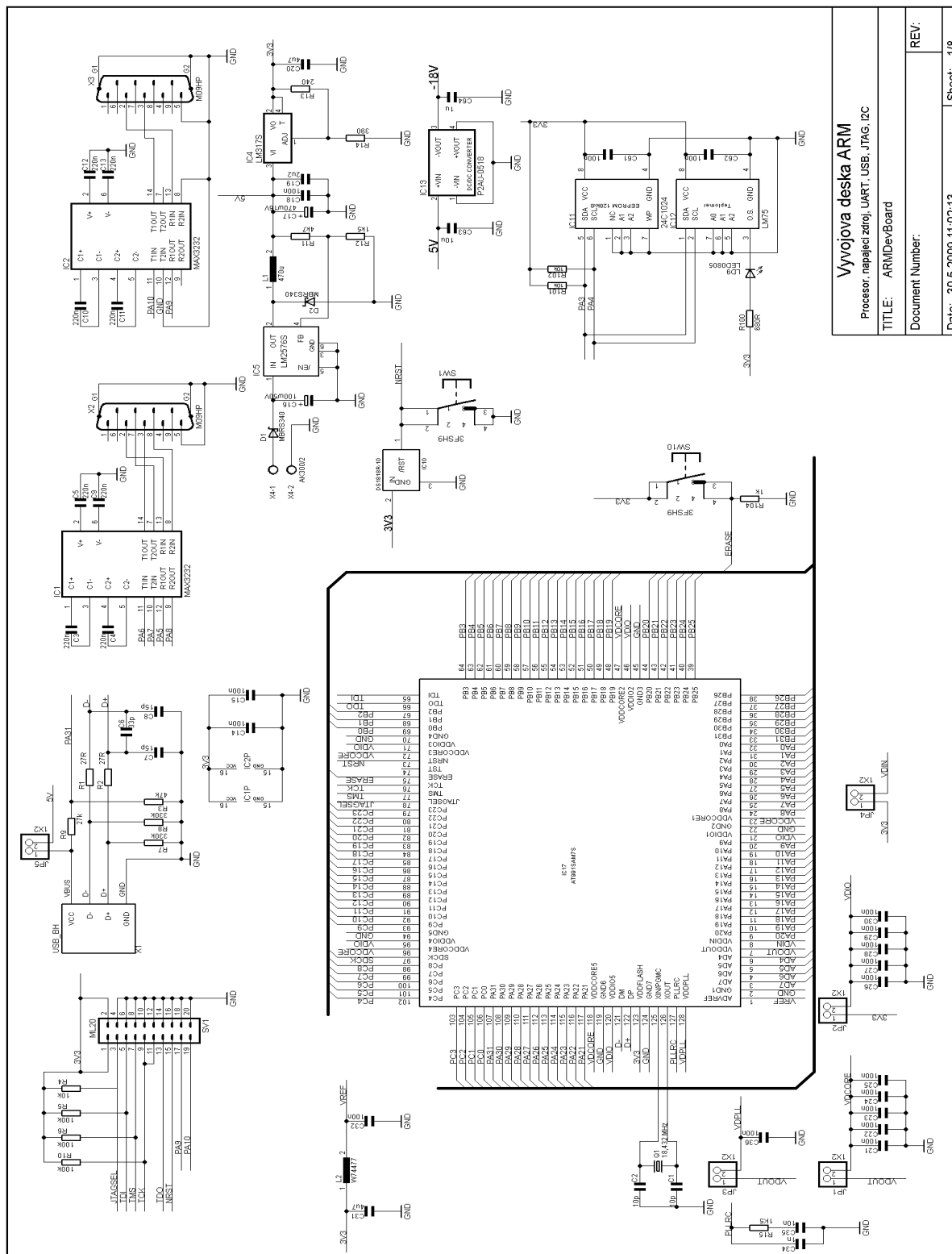
9 Seznam příloh

Příloha1. Návrh zapojení vývojové desky

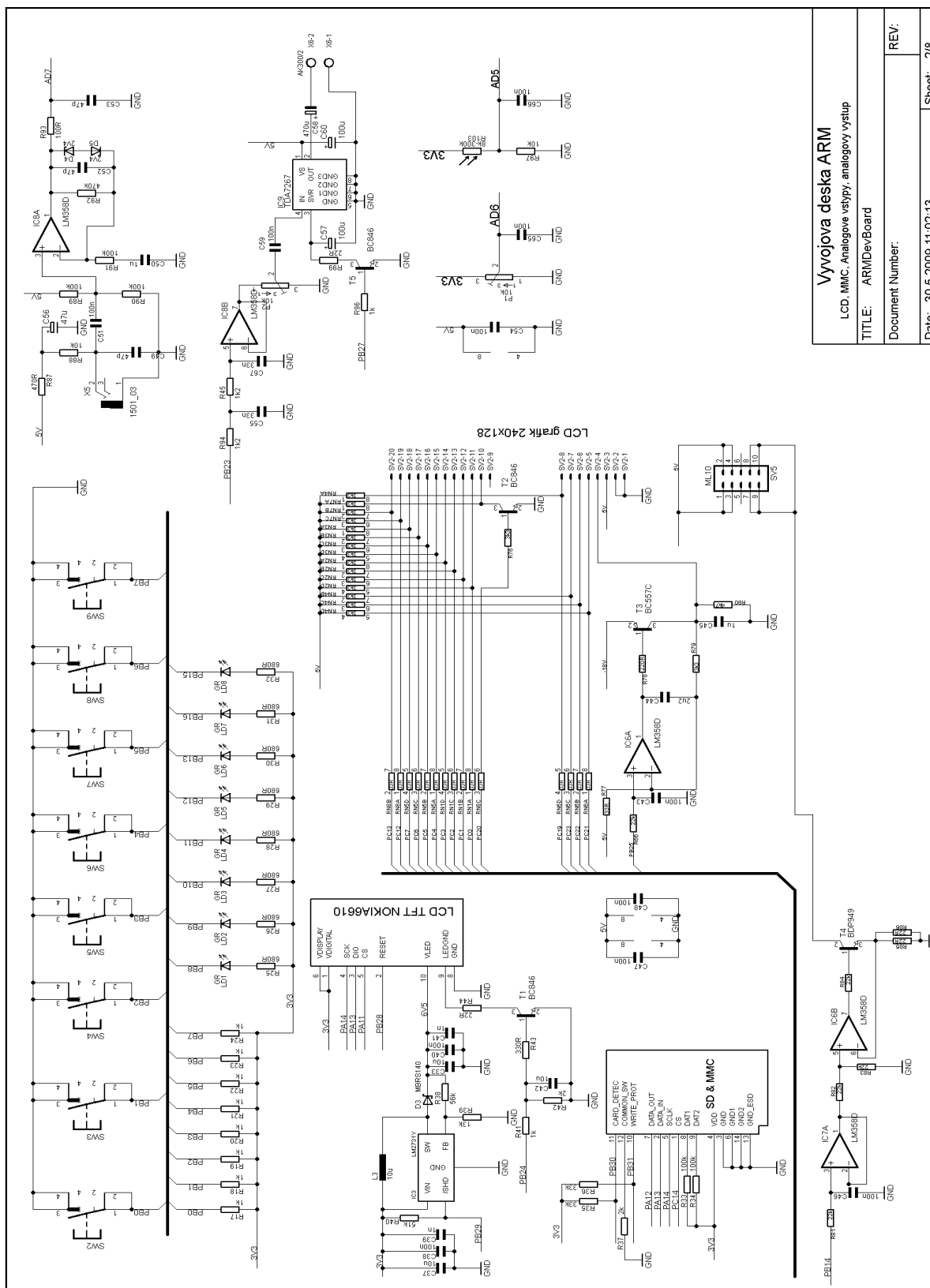
Příloha2. Návrh desky s plošnými spoji

Příloha3. Seznam součástek

Příloha 1. Schéma



Příloha 1a – Schéma vývojové desky – Procesor, napájecí zdroj, USB, UART, JTAG, I2C



Vývojová deska ARM

LCD, MMC, Analogové vstupy, analogový výstup

TITLE: ARMDevBoard

Document Number:

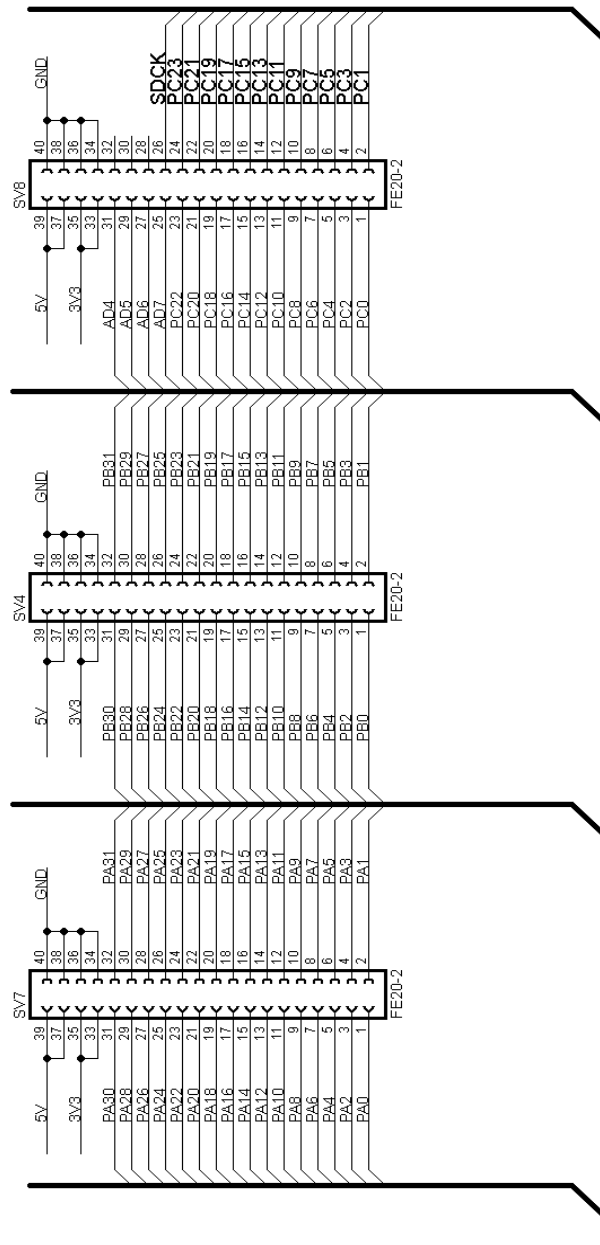
REV:

Date: 30.5.2009 11:02:13

Sheet: 2/8

Příloha 1b – Schéma vývojové desky – Tlačítka, LED, LCD, analogové vstupy a výstup

Příloha 1c – Schéma vývojové desky – Rozšiřující konektory



Vývojová deska ARM

Rozšiřující konektory

TITLE: ARMDevBoard

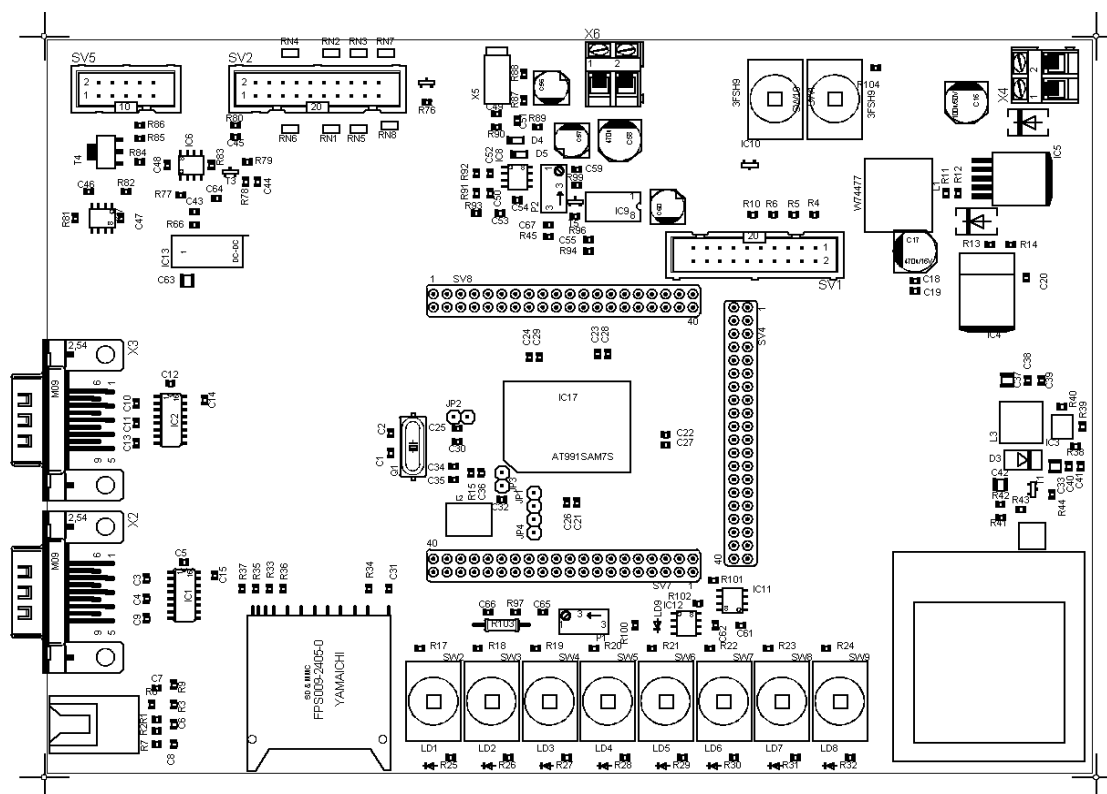
Document Number:

REV:

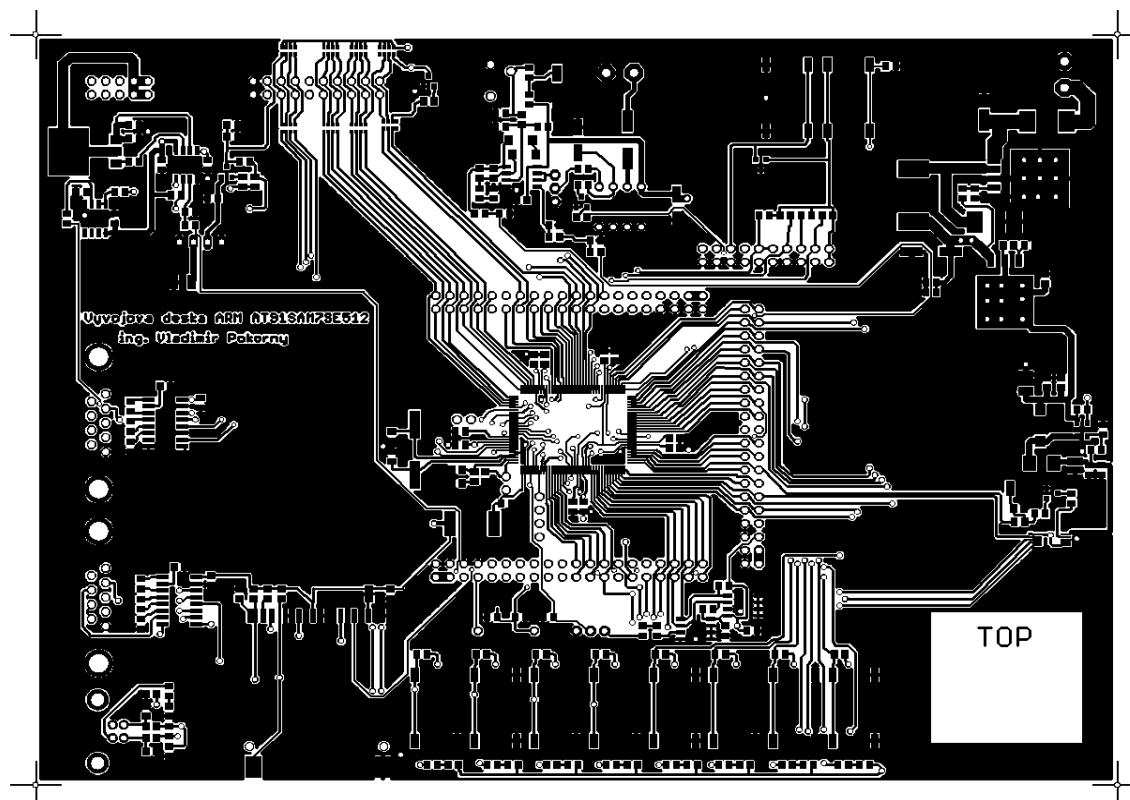
Date: 30.5.2009 11:02:13

Sheet: 3/8

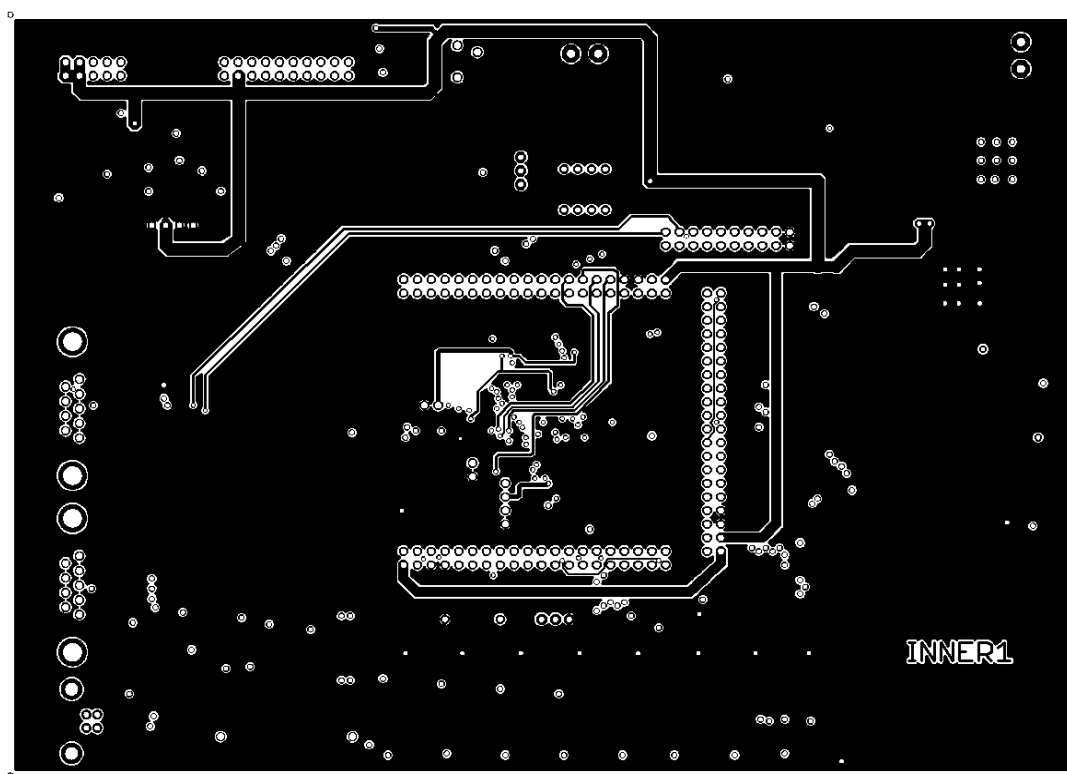
Příloha 2. Deska s plošnými spoji



Příloha 2a – Návrh DPS – Rozmístění součástek



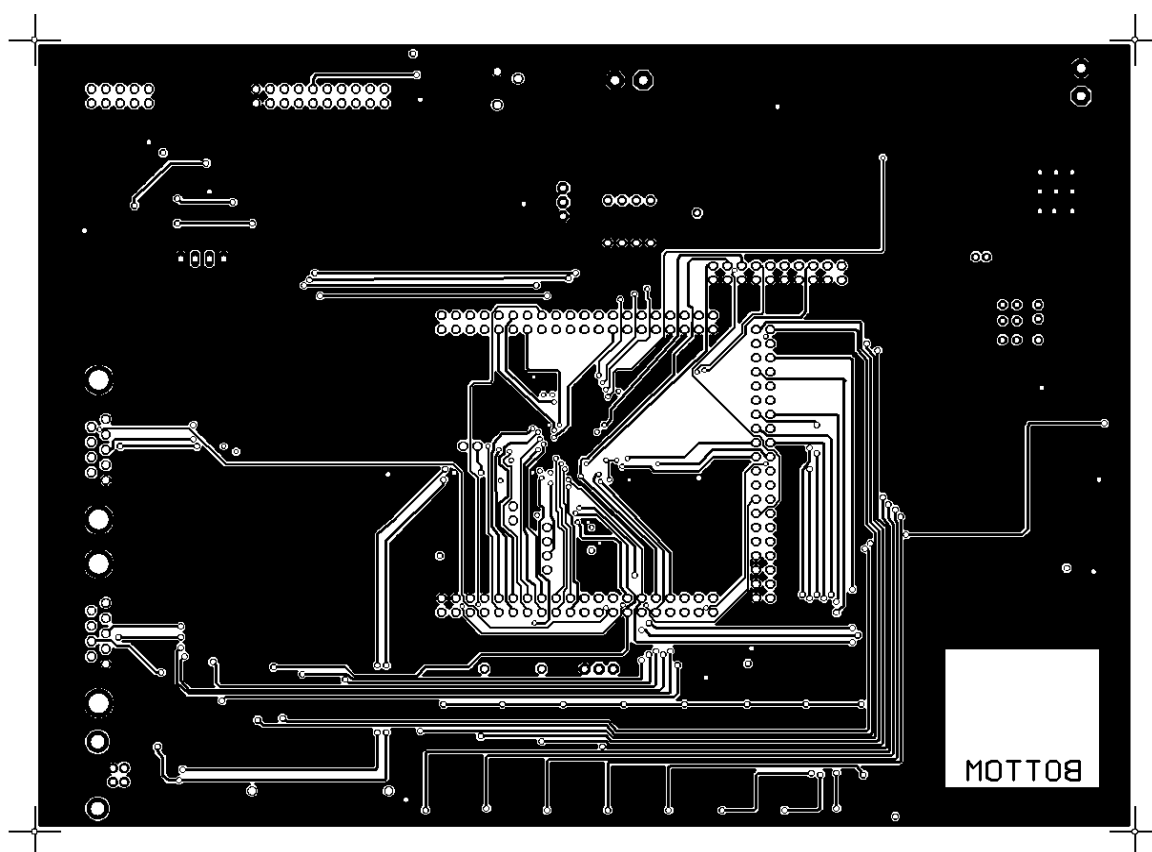
Příloha 2b – Návrh DPS – vrchní vrstva



Příloha 2c – Návrh DPS – 1. vnitřní vrstva



Příloha 2d – Návrh DPS – 2. vnitřní vrstva



Příloha 2e – Návrh DPS – Spodní vrstva

Příloha 3. Seznam součástek

Rezistory

22R/0805	R44,85,86,99
27R/0805	R1,2
33R/0805	R77
33R/CAY16	RN1,5,6,8
100R/0805	R93
220R/0805	R78
240R/0805	R13
330R/0805	R43
390R/0805	R14
470R/0805	R87
680R/0805	R25,26,27,28,29,30,31,32,100
1k/0805	R17,18,19,20,21,22,23,24,41,96
1k2/0805	R45,94
1k5/0805	R12,15
2k/0805	R37,42
3k3/0805	R76,76
3k3/CAY16	RN2,3,4,7
4k7/0805	R11,80
10k/0805	R4,88,97,101,102
13k/0805	R39
22k/0805	R66,81,82,83,84
27k/0805	R9
33k/0805	R35,36
47k/0805	R3
56k/0805	R38
51k/0805	R40
100k/0805	R5,6,10,33,34,89,90,91
330k/0805	R7,8
470k/0805	R92
fotorezistor 1k-1M	R103
10k/RJ9W	P1,2

Kondenzátory

10p/50V/0805	C1,2
15p/50V/0805	C7,8
33p/50V/0805	C6
47p/50V/0805	C49,52,53
1n/50V/0805	C34,39,41
10n/50V/0805	C35
33n/50V/0805	C55, 67
100n/50V/0805	C14,15,18,21,22,23,24,25,26,27, 28,29,30,32,36,38,40,43,46,47, 48,51,54,59,61,62,65,66
1u/25V/0805	C45,50,64
2u2/25V/0805	C19,44
4u7/25V/0805	C20,31
10u/25V/1210	C33,37,42,43
47u/16/SMD-AL-C	C56
100u/16V/SMD-AL-D	C57,60
100u/50V/SMD-AL-F	C16
220n/50V/0805	C3,4,5,9,10,11,12,13
470u/16V/SMD-AL-F	C17,58

Indukčnosti

10uH/WE-PD-M	L3
220uH/2220	L2
470uH/WE-PD-XL	L1

Polovodiče

MBRS340/SMC	D1,2
MBRS140/SMB	D3
ZD2V4/SOD-80	D4,5
MAX3232/SO16	IC1,2
DS1818R-10/SOT23	IC10
AT24C1024/SO08	IC11
LM75/SO8	IC12

P2AU-0518/SIL4		IC13
AT91SAM7SE512/LQFP-128		IC17
LM2731Y/SOT23-5		IC3
LM317S/TO263-3		IC4
LM2576S-ADJ/TO263-5		IC5
LM358D/SO8		IC6,7,8
TDA7267/DIL8		IC9
LED 2mA/0805		LD1,2,3,4,5,6,7,8,9
BC846/SOT23	T1,2,5	
BC557C/SOT232	T3	
BDP949/SOT223	T4	
Krystal		
18,432MHz/SM49		Q1
Konektory		
propojky (jumpery)		JP1,2,3,4
FPS009-240		SD1
ML20		SV1,2
FE20-2		SV4,7,8
ML10		SV5
3FSH9		SW1,2,3,4,5,6,7,8,9,10
USB-B		X1
M09HP		X2,3
AK300/2		X4,6
stereo jack/1501-03		X5
DF23		X7