

Digitální osciloskop s AT91SAM7S64

Bakalářská práce

Vedoucí práce:

Ing. David Matoušek

Autor práce:

Jakub Jelínek

Jihlava 2010

Anotace

Tato práce se zabývá vytvořením jednoduchého digitálního osciloskopu, jehož základem je mikroprocesor AT91SAM7S64. K zobrazování měřených signálů a nastavení je využit osobní počítač, se kterým zařízením komunikuje pomocí rozhraní USB. V textu je popsán použitý mikroprocesor a postup vývoje, včetně nejdůležitějších částí zdrojového kódu. V části věnované vývoji hardwaru je uvedeno konečné schéma zapojení osciloskopu, návrh desky plošných spojů a osazení součástkami.

Annotation

This thesis deals with creating a simple digital oscilloscope, which is based on a microprocessor AT91SAM7S64. For displaying of measured signals and setting is used a personal computer, which communicates with device using USB interface. The text describes used microprocessor and process development, including the most important parts of the source code. In the section devoted to the development of hardware is the final circuit diagram of oscilloscope, circuit board design and installation of components.

Zadání BP

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Souhlasím s umístěním bakalářské práce v knihovně VŠPJ a s jejím užitím k výuce nebo k vlastní vnitřní potřebě VŠPJ .

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje AZ, zejména § 60 (školní dílo).

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

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výtědku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne

.....

Podpis

Na tomto místě bych rád poděkoval panu Ing. Davidu Matouškovi za jeho pomoc, cenné rady a připomínky při vytváření této bakalářské práce.

Obsah

1	Úvod.....	8
2	Popis řešeného problému	10
2.1	Zadání práce	10
2.2	Požadavky	10
2.3	Cíle	10
3	Popis použitého mikroprocesoru	11
3.1	Základní popis	11
3.1.1	AT91SAM7S	11
3.1.2	Popis ARM architektury	13
3.1.3	Popis jádra ARM7TDMI	14
3.2	Popis použitých periférií	14
3.2.1	ADC – Analog to Digital Converter	15
3.2.2	UDP – USB Device Port.....	15
3.2.3	PIT – Periodic Interval Timer	16
3.2.4	PIO – Parallel Input/Output Controller.....	16
3.2.5	PMC – Power Management Controller	17
4	Analýza a návrh implementace	18
4.1	Co je to digitální osciloskop a jak funguje.....	18
4.2	Návrh osciloskopu.....	19
4.2.1	Vstupní zesilovače	19
4.2.2	A/D převodník	20
4.2.3	Komunikace s PC pomocí USB.....	21
4.2.4	Ovládací program	22
4.3	Použitá vývojová prostředí.....	22
4.3.1	Program pro mikroprocesor	22
4.3.2	Nahrávání programu do mikroprocesoru	23
4.3.3	Ovládací program	25
5	Popis implementace	26
5.1	Vývoj a výroba hardwaru.....	26
5.1.1	Vývoj zapojení.....	26
5.1.2	Návrh desky plošných spojů	29
5.2	Vývoj softwaru.....	32
5.2.1	Seznámení s mikroprocesorem	33
5.2.2	Nastavení ADC	35
5.2.3	Komunikace po USB	37
5.2.4	Vzorová aplikace A/D převodníku	38

5.2.5	Řídící příkazy osciloskopu.....	41
5.2.6	Časování.....	43
5.2.7	Aplikace digitálního osciloskopu.....	43
6	Závěr	45
6.1	Splnění cílů práce	45
6.2	Další možnosti rozšíření.....	45
7	Seznam použité literatury	46
8	Seznam obrázků.....	48
9	Seznam tabulek.....	49
10	Seznam zkratk.....	50
11	Seznam příloh	53

1 Úvod

Při rozhodování na jaké téma bude má bakalářská práce, jsem si vybíral mezi programováním a elektronikou. Jelikož z mé praxe, kterou jsem absolvoval v zahraničí u firmy zabývající se tvorbou webových aplikací, žádné téma nevzešlo a sám jsem žádné vhodné neměl, rozhodl jsem se vybrat si z prací vypsaných vyučujícími. Po prostudování nabízených možností a po konzultaci s panem Ing. Davidem Matouškem jsem se rozhodl pro jím nabízené téma Digitální osciloskop s AT91SAM7S64.

Konstrukce digitálního osciloskopu založeného na mikroprocesoru mě zaujala hned z několika důvodů. Za prvé to byla možnost spojení programování jak ovládací aplikace, tak samotného mikroprocesoru s konstrukcí hardwaru. Za druhé, po absolvování předmětů Mikroprocesorová technika 1 a 2, zájem o práci s mikroprocesory. V neposlední řadě pak možnost si na konečný výrobek sáhnout, což u témat čistě programátorských není možné.

Pro samotnou realizaci byl vedoucím práce zadán mikroprocesor AT91SAM7S64 od firmy Atmel a to zejména z důvodu, že disponuje mimo jiné vlastním zabudovaným A/D převodníkem a sběrnici USB, které budou pro konstrukci využity.

Cílem práce je nejdříve vytvořit vzorovou aplikaci A/D převodníku ovládaného přes USB pomocí ovládacího programu, který bude vytvořen ve vývojovém prostředí C++ Builder. Dále pak na základě této aplikace vytvořit digitální osciloskop. Úkolem mikroprocesoru je vzorkování vstupního signálu v přesně daných časových intervalech a přenos těchto dat do počítače pomocí USB, kde jsou posléze interpretovány ovládacím programem.

Pro zjednodušení vývoje periférií a programování mikroprocesoru byl zakoupen vývojový kit AT91SAM7S256-KIT od firmy Kramara s.r.o., který je určený pro základní vývoj softwaru pro řadu mikrokontrolérů AT91SAM7S. Tento kit obsahuje mikroprocesor AT91SAM7S256, který je řízen hodinovým signálem o kmitočtu 18,432MHz. Dále obsahuje zabudovaný konektor USB-B pro možnost programování přes program SAM-BA či pro ladění aplikací, které komunikují přes rozhraní USB. Všechny vstupně/výstupní piny a piny A/D převodníku jsou vyvedeny na dvě lišty, které ho umožňují připojit do nepájivého pole. Program pro mikroprocesor je psán

v jazyce C++ ve vývojovém prostředí Keil μ Vision 4. Ovládací program je psán také v jazyce C++ ve vývojovém prostředí Borland C++ Builder.

2 Popis řešeného problému

2.1 Zadání práce

Seznamte se s procesorem AT91SAM7S64 (Atmel). Vytvořte vzorovou aplikaci A/D převodníku ovládaného přes USB pomocí ovládacího programu, který bude vytvořen ve vývojovém prostředí C++ Builder. Využijte zabudované USB rozhraní a zabudovaný A/D převodník daného procesoru. Vytvořte digitální osciloskop na základě předchozí vzorové aplikace.

2.2 Požadavky

Z výše uvedeného cíle práce je patrné, že k realizaci má být použit mikroprocesor AT91SAM7S64 od firmy Atmel. Tento mikroprocesor disponuje mimo jiné zabudovaným A/D převodníkem, který bude použit na vzorkování měřeného signálu a zabudovaným USB rozhraním, které bude zajišťovat obousměrnou komunikaci osciloskopu s počítačem. Nejprve má být vytvořena vzorová aplikace A/D převodníku ovládaného přes USB pomocí ovládacího programu. Tato aplikace vyše do mikroprocesoru požadavek na převod vstupního napětí a následně z něj přečte změřenou hodnotu. Lze ji využít pro měření stejnosměrných nebo jen pomalu proměnných signálů. Dalším krokem má být využití poznatků z této aplikace a jejich další doplnění, aby bylo možné měřit a zobrazovat i časově proměnné signály, jako sinus, obdélník či trojúhelník.

2.3 Cíle

Cílem práce je vytvoření funkčního vzorku digitálního osciloskopu, který bude pro ovládání a zobrazování měřeného signálu využívat počítač. Jelikož zadání nijak nespecifikuje rozsah měřených napětí, frekvenční rozsah ani počet kanálů, limitujícím faktorem budou vlastnosti zabudovaného A/D převodníku. Ovládání by mělo obsahovat alespoň základní prvky, jako jsou volba hodnoty spouštěcího napětí, citlivost na hranu spouštěcího napětí, volba časové základny nebo volba měřených kanálů.

3 Popis použitého mikroprocesoru

Tato kapitola se bude věnovat popisu použitého mikroprocesoru, jeho architektury, jádra a periférií.

3.1 Základní popis

3.1.1 AT91SAM7S

AT91SAM7S je základní označení obvodů firmy Atmel, které jsou postaveny na architektuře jádra označované jako ARM7TDMI. Díky využití tohoto jádra jsou tyto mikroprocesory vhodné pro použití v široké škále aplikací. Jejich velikou výhodou je také jejich cena, která vzhledem k jejich výkonům, velikosti pamětí a velikou vybaveností zabudovaných periférií není nikterak vysoká. [8]

Základní vlastnosti řady AT91SAM7S:

- Integrovaná programová paměť Flash o velikosti až 512kB
Rychlá paměť pro uložení programu, umožňující až 10.000 zapisovacích cyklů.
- Datová paměť SRAM až 64kB
Velice rychlá datová paměť, umožňující jednoinstrukční přístup.
- Memory Controller (MC)
Integrovaný detektor poruchových stavů Flash.
- Reset Controller (RSTC)
Zajišťuje Power-on Reset a výrobcem kalibrovaný Low-power Brown-out detektor.
- Hodinový generátor (CKGR)
Integrovaný RC oscilátor, PLL násobič.
- Rozšířený kontrolér přerušení (AIC)

Umožňuje individuální maskování v až 8 úrovních přerušení, dva vnější a jeden rychlý zdroj přerušení.

- Ladící jednotka (DBGU)

Umožňuje ladění programu pomocí UARTu.

- Okénkový Watchdog (WDT)

Slouží k ochraně programu před zacyklením. Generuje buď přerušení, nebo reset.

- Až 11 kanálová DMA jednotka

Umožňuje přenos dat mezi perifériemi a datovou pamětí.

- SAM-BA™ Boot Assistant

Výchozí program uložený v ROM paměti, sloužící pro nahrávání programu pomocí USB

- IEEE® 1149.1 JTAG Boundary Scan

Rozhraní pro ladění programu.

- 5V tolerantní vstupně výstupní piny

4 s proudovým zatížením až 16 mA.

Základní integrované periferie:

- Periodic Interval Timer (PIT)

20bitový programovatelný čítač a 12bitový periodický čítač.

- Real-time Timer (RTT)

32bitový čítač s vlastním interním RC oscilátorem

- Paralelní I/O kontroler (PIOA)

Až 32 individuálně konfigurovatelných uživatelských pinů.

- USB 2.0 port

- 2x USART

Individuální nastavení, podpora různých standardů.

- 3kanálový 16bitový Čítač/Časovač (TC)

Každý kanál individuálně nastavitelný

- 4kanálový 16bitový PWM kontrolér (PWMC)

- Dvoudrátový interface (TWI)

- 8kanálový 10bitový A/D převodník

Až 533 kSPS při 8bitovém převodu.

3.1.2 Popis ARM architektury

ARM, někdy také označovaná jako „Advanced RISC Machine“, je 32 bitová architektura RISC mikroprocesorů, která byla vyvinuta firmou ARM Limited. Díky svým vlastnostem jako jsou nízká spotřeba nebo velký výpočetní výkon, jsou používány ve velkém množství vestavěných systémů. V dnešní době již tato architektura tvoří zhruba tři čtvrtiny trhu s 32 bitovými RISC procesory. S ARM technologií se můžeme setkat doslova na každém kroku, lze ji nalézt v mobilních telefonech, MP3 přehrávačích, herních konzolách a všude tam, kde je potřeba nízká spotřeba energie při zachování dostatečného výpočetního výkonu.

Velkou výhodou systémů s ARM architekturou je také fakt, že ARM je pouze jádro samotného mikroprocesoru, které je stejné u všech výrobců procesorů. Pro vývojáře je tedy poměrně snadné přejít z produktu jednoho výrobce k jinému. Díky tomu je také rozšířeno velké množství vývojových prostředí a vývojových kitů a vývojář tedy není nucen využívat produkt přímo toho výrobce.

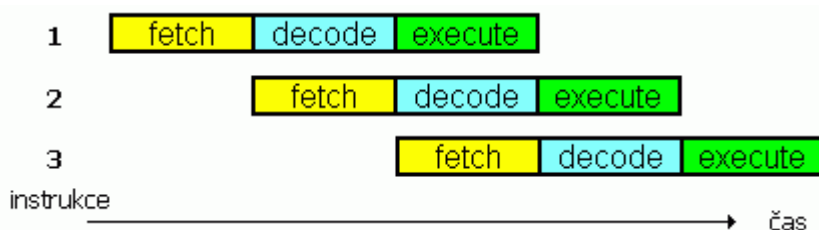
Další výhodou je také často velice bohaté vybavení periferiemi, jako jsou sériová linka, dvoudrátový interface, USB, ethernet, A/D převodník, čítače a časovače a další, které rozšiřují možnosti procesoru. [2]

3.1.3 Popis jádra ARM7TDMI

ARM7TDMI je jedno z jader rodiny ARM, které je speciálně navrženo pro 32 bitové mikroprocesory. Vychází z Von Neumannovy architektury se společnou 32bitovou sběrnici pro data i instrukce. Konstrukteři dokázali na čip velmi malých rozměrů vměstnat vysoce výkonné jádro s minimální spotřebou, které navíc podporuje silné a rychlé 32bitové ARM® instrukce. Pokud je zapotřebí uspořit programovou paměť, pak je k dispozici také zkrácená 16bitová instrukční sada Thumb®, obsahující nejpoužívanější instrukce.

Pro maximální zvýšení výkonu je u ARM7TDMI používán tzv. 3 stupňový pipeline, který umožňuje během jediného hodinového cyklu vykonat až 3 operace, kdy v jednom okamžiku dochází k vyzvednutí jedné instrukce z paměti, dekódování druhé a zpracování třetí instrukce.

- Fetch - vyzvednutí instrukce z paměti
- Decode - dekódování instrukce
- Execute – zpracování instrukce



Obrázek 1: Zpracování instrukcí jádrem ARM7TDMI [8]

Díky takovému řešení zpracování dochází vlastně ke zpracovávání 3 instrukcí souběžně, což dovoluje při maximální frekvenci 55MHz obvodu AT91SAM7S256 pracovat až s výkonem 49,5MIPS (0,9MIPS/MHz). [8]

3.2 Popis použitých periférií

V této části budou zmíněny pouze ty periférie, které byly použity pro řešení konstrukce digitálního osciloskopu.

3.2.1 ADC – Analog to Digital Converter

Jedná se asi o nejdůležitější část osciloskopu z pohledu jeho konečných vlastností.

Použitý mikrokontrolér obsahuje zabudovaný 8kanálový 10bitový převodník, který funguje na principu postupné aproximace. Rozsah měřeného napětí je od 0V až do napětí přivedeného na pin procesoru ADVREF, v tomto případě 3,3V (nesmí být vyšší než napájecí napětí procesoru). Pro realizaci bylo použito nastavení pro pouze 8bitový převod, což na úkor přesnosti převodu zvedlo poměrně podstatně jeho rychlost (10bit umožňuje maximálně 384 kSPS, zatímco 8bit umožňuje až 533 kSPS) a zároveň se tím zjednodušil i formát pro přenos změřených vzorků. Pro převod jednoho měřeného vzorku na jednom kanálu, je zapotřebí 15 hodinových cyklů A/D převodníku (8 MHz). Pět cyklů je zapotřebí pro navzorkování měřeného signálu a dalších deset pro samotný převod. Samotný aproximační převod probíhá tak, že v prvním kroku je výsledný registr vynulován a jeho nejvyšší bit je nastaven na hodnotu 1. Takto vzniklá digitální hodnota je poté převedena na analogovou pomocí D/A převodníku a porovnána s měřeným vzorkem. Pokud je měřený vzorek větší, pak je jednička v registru ponechána, pokud je vzorek menší, je příslušný bit nulován. Stejně se postupuje i s nižšími bity, až do dosažení požadované přesnosti, kdy aproximační registr již obsahuje požadovanou hodnotu převodu. [5]

3.2.2 UDP – USB Device Port

USB (Universal Serial Bus) je rychlé a flexibilní rozhraní, sloužící ke snadnému připojení rozličných zařízení k počítači. V dnešní době snad každý osobní počítač disponuje alespoň několika USB porty. Rozhraní je dostatečně univerzální na to, aby bylo použito pro standardní periférie jako klávesnice, tiskárny či přenosné disky, stejně tak jako pro speciální účelová zařízení. USB je navrženo tak, aby ho bylo snadné použít, bez nutnosti konfigurace hardwaru a softwaru uživatelem.

Jednoduše řečeno je USB poměrně rozdílné od jiných rozhraní, jako jsou sériová či paralelní linka, a také je postupně nahrazuje. Po připojení musí zařízení odpovědět na sérii dotazů, které umožní počítači zjistit jeho požadavky a určení a nastavit s ním komunikaci. K tomu je také zapotřebí, aby v počítači byl nainstalován potřebný ovladač, který odpovídá požadavkům zařízení. [3]

3.2.3 PIT – Periodic Interval Timer

Jak už název napovídá, slouží tento časovač pro přesné odměřování stejných časových úseků. Při realizaci osciloskopu byl využit pro přesné odměřování časových intervalů mezi jednotlivými A/D převody. Změnou tohoto intervalu se dosahuje měření pro různé časové základny.

PIT se skládá ze dvou čítačů. První je 20bitový čítač, který zvyšuje svou hodnotu každých 16 hodinových cyklů. Při dosažení hodnoty nastavené v registru PIT_MR (Mode Register) v poli PIV (Periodic Interval Value) je nastaven bit PITS (Periodic Interval Timer Status) registru PIT_SR (Status Register), nebo může nastat přerušení (pokud je tak nastaveno). Při každém přetečení čítače přes hodnotu PIV je inkrementován druhý, 12bitový čítač. K nulování obou čítačů dochází při přečtení registru PIT_PIVR (Periodic Interval Value Register). Pokud je třeba pouze přečíst hodnotu čítačů bez jejich nulování, je k dispozici registr PIT_PIIR (Periodic Interval Image Register). [5]

3.2.4 PIO – Parallel Input/Output Controller

PIO je určen k ovládání 32 plně programovatelných vstupně/výstupních portů. Každý z portů může být použit jako čistě vstupně/výstupní a nebo může být sdílen s některým ze zabudovaných periférií. Toto řešení umožňuje značně zredukovat počet pinů procesoru a tím i snížit náklady na jeho výrobu.

Pomocí PIO lze nastavovat pro každý port jeho povolení k používání, nastavení jako výstupní, zapnutí filtru zákmitů (impulzy kratší než je 1 hodinový cyklus jsou ignorovány a pokud je na daný port připojeno tlačítko, odpadá tak starost o ošetření zákmitů), nastavení logické hodnoty výstupu, nastavení přerušení, nastavení sdílení portu s perifériemi nebo připojení pull-up zabudovaných rezistoru s hodnotou 10kΩ. Pro každé z těchto nastavení jsou k dispozici hned 3 registry, které v názvu nesou Set/Enable, Clear/Disable nebo Status a každému z 32 portů je přidělen jeden bit v každém z registrů PIO. Toto rozdělení do třech registrů pak v praxi znamená zjednodušení některých operací. Když například budeme chtít vývod PA4 (který musí být předem nastaven jako výstupní) nastavit na log. 1, pak stačí do registru PIO_SODR (Set Output Data Register) zapsat hodnotu (1<<4). Tento zápis přitom nijak neovlivní jiné porty, jak tomu může být u jiných architektur. Pokud chceme naopak vývod přivést

do log. 0, použijeme registr PIO_CODR (Clear Output Data Register). Pokud chceme pouze zjistit stav nastavení, využijeme registru PIO_ODSR (Output Data Status Register). Díky tomuto řešení tak odpadá nutnost někdy nepřehledného maskování bitů.
[5]

3.2.5 PMC – Power Management Controller

PMC slouží k optimalizaci spotřeby mikroprocesoru. K tomu využívá kontroly zdroje hodin pro systém a periferie, přičemž zdroj hodin pro některé periferie lze zcela zakázat.
[5]

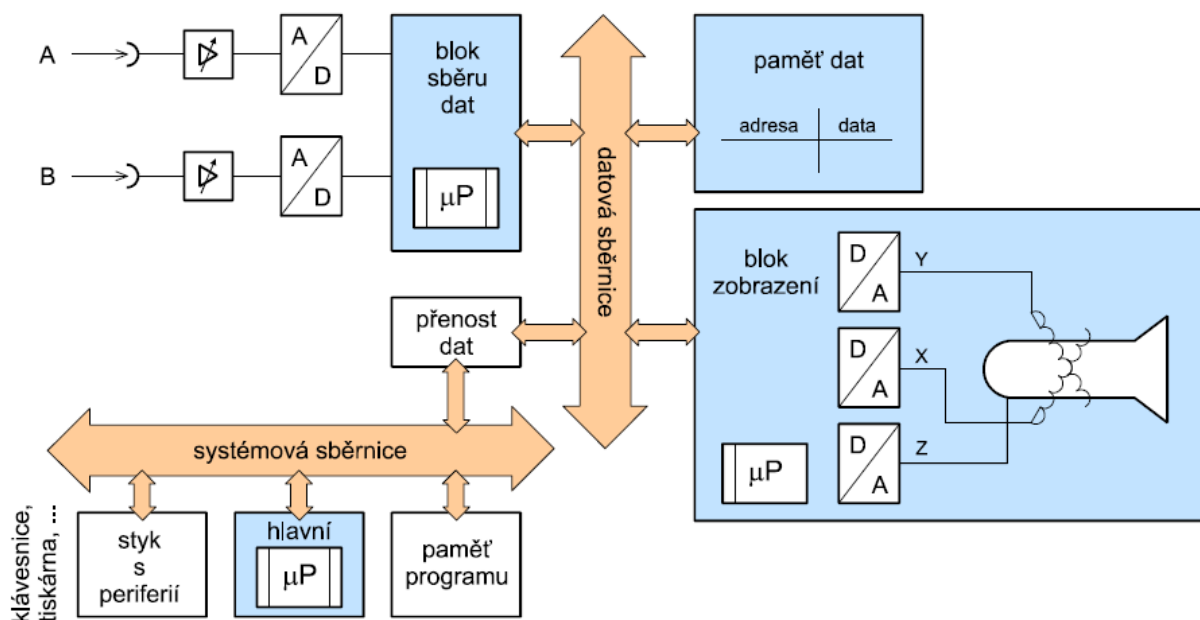
Power Management Controller poskytuje zdroj hodin pro:

- Hlavní zdroj hodin (MCK) – lze ho nastavovat od několika set Hz až po maximální frekvenci podporovanou procesorem (55MHz). Je přístupný částem, které jsou stále v provozu jako AIC (Advanced Interrupt Controller) nebo MC (Memory Controller).
- Hodiny procesoru (PCK)
- Hodiny periferií – vychází z hlavního zdroje hodin a v datasheetu se tak i nazývá (MCK), který je poskytován zabudovaným periferiím (USART, TWI, SPI, PIO, ADC ...). Periferie pak obsahují vlastní předděličky, které slouží k nastavení požadované frekvence.
- Hodiny pro UDP (UDPCK) – zdroj hodin potřebný pro funkci USB rozhraní.
- Programovatelné zdroje hodin – tři individuálně nastavitelné zdroje hodin, které mohou být vyvedeny na piny PCKx procesoru.

4 Analýza a návrh implementace

4.1 Co je to digitální osciloskop a jak funguje

Digitální osciloskop je elektronický měřicí přístroj, sloužící k zobrazování časového průběhu napětového signálu. Oproti analogovým osciloskopům má několik výhod, jako možnost uložení měřeného signálu a jeho následné zpracování a vyhodnocování, či tisk obrazovky. Jeho princip lze popsat tak, že vstupní signál je nejprve zesílen a upraven tak, aby mohl být navzorkován pomocí A/D převodníku. Takto získaná data jsou poté uložena do paměti, zpracována mikroprocesorem a zobrazena na displeji.



Obrázek 2: Blokové schéma digitálního osciloskopu

Vstupní zesilovače slouží ke změně měřícího rozsahu, úpravě signálu pro převodníky a zároveň zajišťují i velký vstupní odpor osciloskopu.

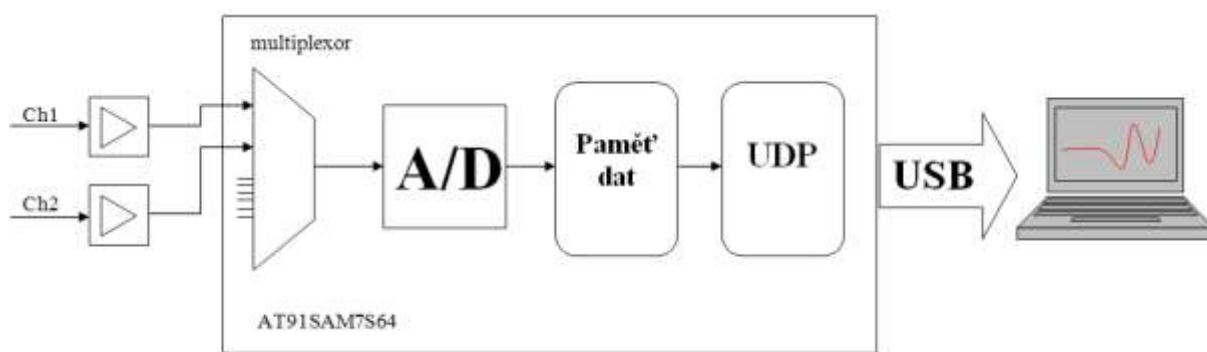
Výsledné vlastnosti osciloskopu velice ovlivňují vlastnosti použitých analogově/číslcových převodníků. Přesnost měření je ovlivněna rozlišením převodníku. Čím více bitový je, tím přesnější je výsledek. Dalším důležitým parametrem je vzorkovací frekvence, se kterou je vstupní signál snímán. Jelikož se zobrazování signálu neděje jeho přímým převodem na obrazovku, ale rekonstruuje se z dat uložených v paměti, je zřejmé, že výsledný obraz bude tím věrnější, čím více vzorků na periodu

bude k dispozici. Maximální počet vzorků je tedy omezen vzorkovací frekvencí, jak je to ovšem s minimálním počtem? Na tuto otázku dává odpověď tzv. Shannon-Kotělnikovův teorém. Ten říká, že pro spolehlivou rekonstrukci signálu musí být vzorkovací frekvence více jak dvakrát větší než frekvence měřená. [14]

4.2 Návrh osciloskopu

Osciloskop s AT91SAM7S64 bude oproti klasickému digitálnímu osciloskopu fungovat trochu odlišně, jelikož se nebude jednat o osciloskop jako samostatný měřicí přístroj, ale o periférii připojenou k počítači pomocí USB. K zobrazování a ovládání bude tedy sloužit ovládací program. Skládat se bude ze čtyř základních částí:

- Vstupní zesilovače
- A/D převodník
- Komunikace s PC pomocí USB
- Ovládací program



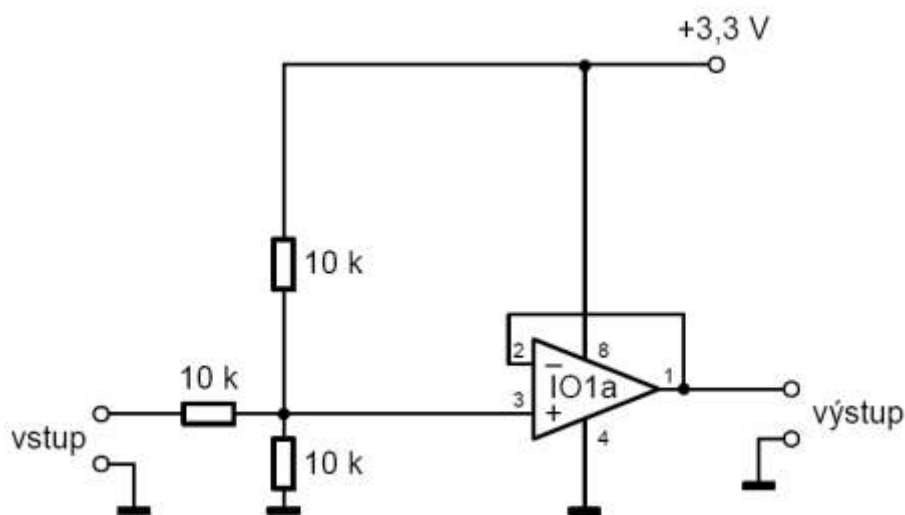
Obrázek 3: Blokové schéma osciloskopu s AT91SAM7S64

4.2.1 Vstupní zesilovače

Vstupní zesilovače osciloskopu slouží pro úpravu rozsahu měřeného signálu. Jelikož v mikroprocesoru zabudovaný převodník umí zpracovávat pouze signály od 0V do 3,3V, je třeba vstupní signál posunout, jinak by docházelo ke vzorkování pouze kladných půlperiod signálu.

K úpravě vstupního signálu byl použit integrovaný obvod MCP602, který obsahuje dvojici operačních zesilovačů, pro každý kanál jeden, které lze napájet z nesymetrického zdroje napětí a jejichž frekvenční přenos je až 2,8MHz [15], což je zcela dostačující. Na neinvertující vstup operačních zesilovačů zapojených jako sledovače, bylo přes odpory o velikosti připojeno:

- Napětí 3,3V (4,878k Ω)
- Měřený signál (10k Ω)
- Zem (10k Ω)



Obrázek 4: Schéma zapojení vstupního zesilovače jednoho kanálu

Vstupní signál je díky odporovým děličům zeslabený a k němu je přičtena polovina napájecího napětí. Díky tomu může být A/D převodníkem snímán signál v rozmezí od 6V do -6V.

4.2.2 A/D převodník

Zabudovaný převodník umožňuje převod až 8 kanálů v rozlišení 8 nebo 10 bitů.

Mikroprocesor ve skutečnosti neobsahuje 8 převodníků, ale pouze jeden, před nímž je předřazen multiplexor, který k měření přepíná požadovaný kanál. To způsobuje, že nelze vzorkovat oba kanály současně a mezi jednotlivými měřeními vzniká časový

posuv o velikosti 15 cyklů ADCLK ($1,875\mu\text{s}$). U pomalých signálů není tento posuv vůbec patrný, ale u signálů okolo 10kHz již ano. Proto je při zobrazování prováděna korekce, která spočívá v posunutí vykreslovaných vzorků.

Kanály 1 – 4 mají piny procesoru sdíleny s portem PIO a kanály 5 – 8 jsou vyvedeny přímo jako samostatné piny procesoru. V aplikaci budou použity kanály 5 a 6, odpadá tak nastavování jejich sdílení.

Rozlišení bylo použito pouze 8bitové, což sice znamená horší přesnost měření, ale zjednodušil se tak přenos dat do počítače, neboť 8bitů se přenáší snadněji než 10. Díky nižšímu rozlišení se také podstatně zvedla vzorkovací frekvence z 384 na 533kSPS. Takto rychlé vzorkování postačuje při měření na dvou kanálech a počtu 25 vzorků na periodu, snímání signálů kolem frekvence 10kHz. [5]

4.2.3 Komunikace s PC pomocí USB

Komunikace pomocí USB se ukázala jako nejnáročnější část celé práce.

V první řadě byla snaha vyhnout se nutnosti vytvářet pro zařízení ovladače, což vedlo k využití standardu HID (Human Interface Device). Tento standard, který se nejčastěji používá u zařízení, jako jsou klávesnice, polohovací zařízení, webkamery či sluchátka, se vyznačuje tím, že pro jeho použití není potřeba instalovat žádný speciální ovladač, neboť ten je již od Windows 98 jejich součástí. Toto řešení pomohlo vyřešit problém s ovladačem, znamenalo však zpomalení komunikace.

Jedním z dalších problémů je to, že zařízení samo nemůže zahájit komunikaci po USB. Ta je totiž řízena přímo procesorem počítače. Tento problém se ve většině případů s úspěchem obchází tak, že počítač se pravidelně dotazuje zařízení, jestli nemá nějaká data k odeslání. [3] Toto řešení se však v tomto případě nedalo použít. Při USB komunikaci totiž dochází v zařízení k přerušení s největší možnou prioritou, což by například při současném vzorkování vstupního signálu A/D převodníkem, znamenalo ztrátu nebo znehodnocení měřených dat. Z tohoto důvodu je v mikroprocesoru během vzorkování zcela zakázáno přerušení od UDP a ovládací program v počítači je „uspán“ po dobu nezbytně nutnou pro navzorkování signálu. Sám A/D převodník je značně náchylný na rušení a pokud by během převodu došlo k pokusu o komunikaci po USB, může také dojít k chybě.

4.2.4 Ovládací program

Ovládací program slouží k nastavování měřicích parametrů osciloskopu, jako jsou výběr kanálu (pouze kanál 1, pouze kanál 2 nebo dvoukanálový režim), nastavení spouštěcího napětí, spouštění na náběžnou/sestupnou hranu signálu a časová základna v rozmezí od 100ms/dílek po 20 μ s/dílek obrazovky. Jeho dalším úkolem je vykreslování měřeného signálu a měření jeho frekvence.

Pomocí menu Soubor->Uložit lze obsah obrazovky osciloskopu uložit do souboru, který může sloužit například k tisku.

4.3 Použitá vývojová prostředí

4.3.1 Program pro mikroprocesor

S vývojovým kitem AT91SAM7S256-KIT dodaným firmou Kramara s.r.o., je na přiloženém CD dodáván program Eclipse + Open OCD, který je doporučován k vývoji aplikací pro tento kit. Bohužel se ale tento program a přídatné balíky nepovedlo zprovoznit a proto bylo třeba se poohlédnout po jiném prostředí.

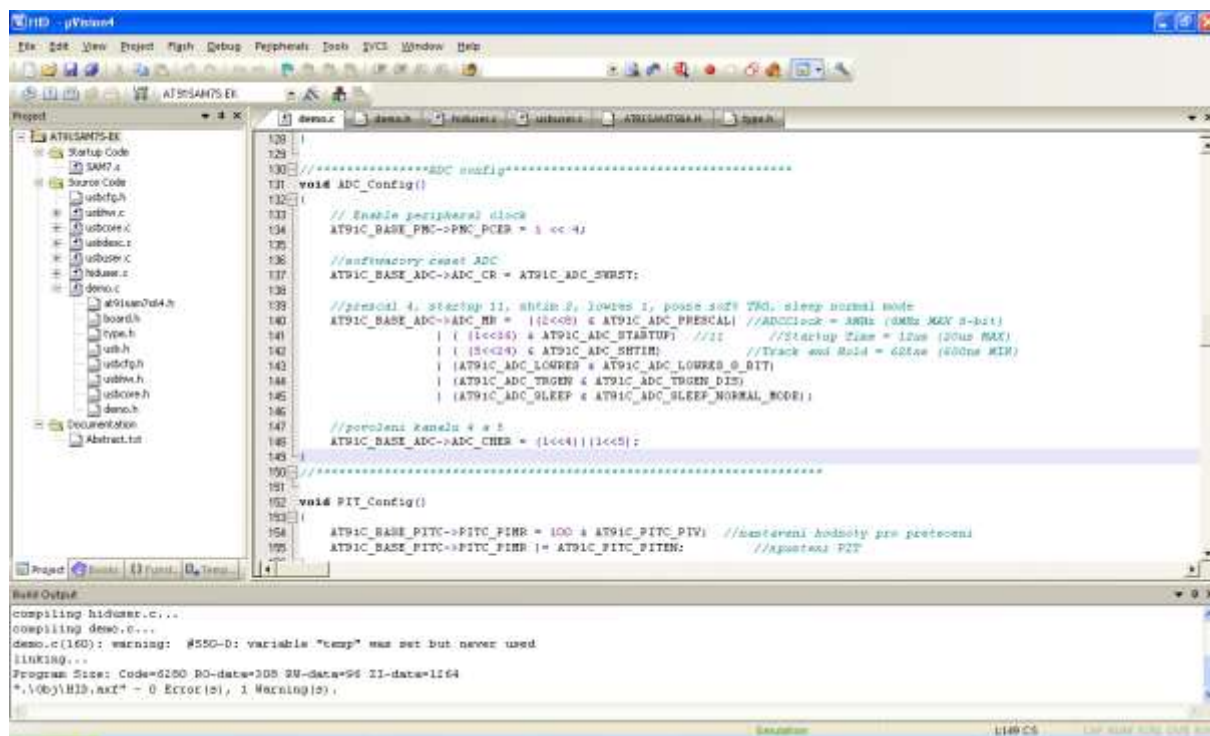
Výrobce procesoru doporučuje v softwarovém balíku [13], který je možné stáhnout na www.atmel.com, vývojová prostředí od čtyř výrobců:

- Segger
- KEIL
- IAR
- Codesourcery

Po domluvě a na doporučení vedoucího práce bylo zvoleno vývojové prostředí KEIL μ Vision 4. Velkou výhodou využití tohoto programu bylo i to, že jeho dřívější verze se využívala během výuky předmětu Mikroprocesorová technika 1.

Toto prostředí umožňuje simulaci běhu mikroprocesoru a většiny jeho periférií, bez nutnosti jeho připojení. Tato schopnost zejména v začátcích velice usnadnila vývoj aplikace. Například při nastavování A/D převodníku, kdy při zadání referenčního napětí

a měřeného napětí pro jednotlivé kanály, bylo možno odečítat převedenou hodnotu přímo z registrů a ověřit tak jeho funkci.



Obrázek 5: Vývojové prostředí KEIL μVision 4

4.3.2 Nahrávání programu do mikroprocesoru

Jelikož samo vývojové prostředí μVision nedokáže zprostředkovat nahrání vytvořeného programu do mikroprocesoru, je třeba použít speciální program. Firma Atmel pro tento účel dodává program SAM-BA (součástí zakoupeného kitu byla jeho verze v2.8).

Jelikož program SAM-BA umí do procesoru nahrát pouze soubory typu .bin, ale program μVision jej nevytváří, je třeba soubor s příponou .axf na binární převést. K tomu slouží program fromelf.exe, který je součástí instalace μVision 4. Pro převedení stačí program spustit s patřičnými parametry:

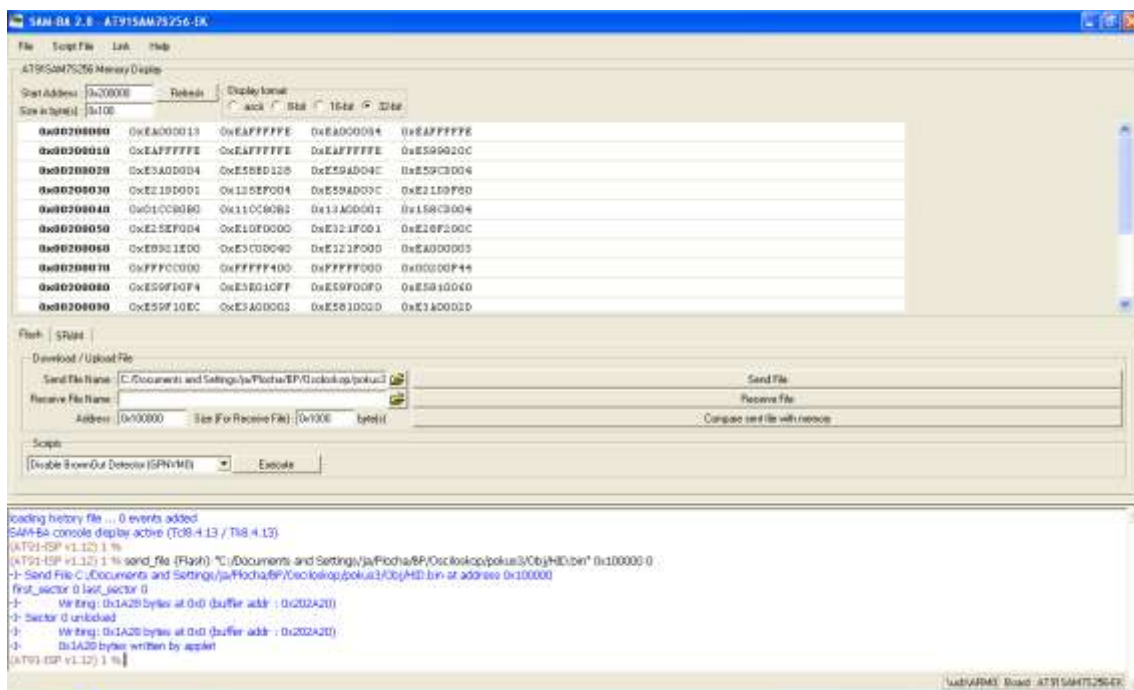
```
fromelf.exe --bin --output=test.bin test.axf
```

Pro nahrávání aplikace do samotného mikroprocesoru se využívá sběrnice USB. Aby jí ale bylo možno využít k nahrání aplikace do programové paměti, je zapotřebí aktivace tzv. bootloaderu podle následujícího postupu [7]:

1. Odpojte napájení
2. Zkratujte jumper J2
3. Připojte napájení
4. Vyčkejte 10 – 30 sekund
5. Odpojte napájení
6. Rozpojte jumper J2
7. Připojte USB kabel
8. Připojte napájení

Po této proceduře počítač detekuje zařízení „at6124.Sys ATMEL AT91 xxxxx Test Board“ a může být nahrán program [7]:

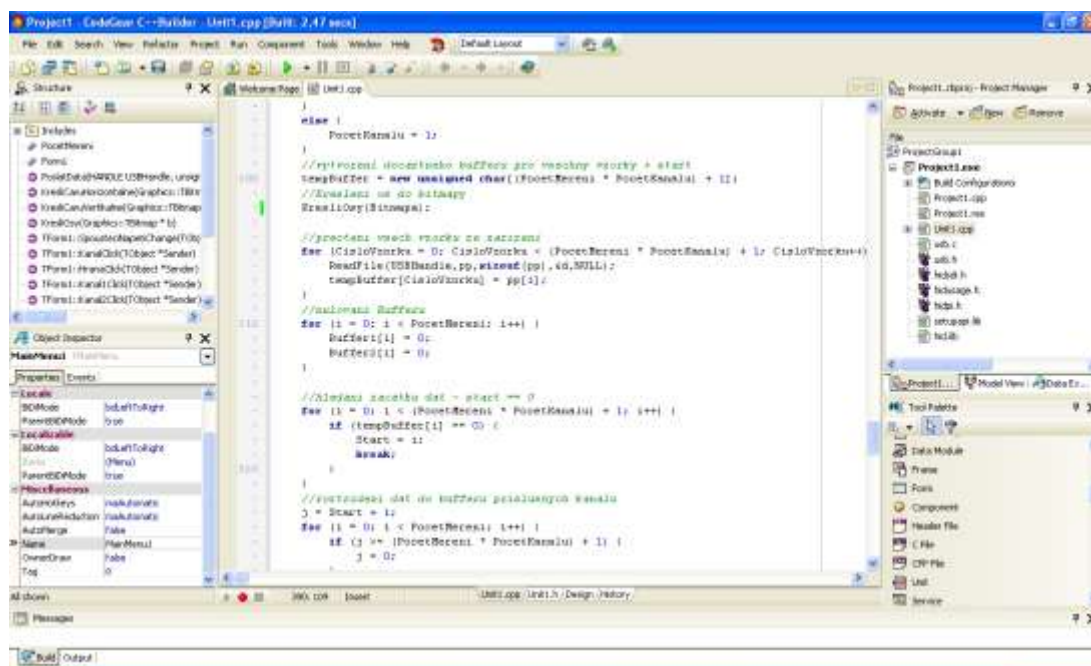
1. Spustíte program SAM-BA
2. V položce Select the connection vyberte: \usb\ARM0
3. V položce Select your board vyberte: AT91SAM7S256-EK
4. Klikněte na Connect
5. Vyberte záložku Flash
6. Vyberte v Send File Name jméno programu pro nahrání do kitu
7. Stiskněte Send File
8. Restartujte mikroprocesor



Obrázek 6: Program SAM-BA

4.3.3 Ovládací program

Pro psaní ovládacího programu bylo již v zadání uvedeno použití programu C++ Builder od společnosti Borland. Toto prostředí umožňuje snadné a rychlé vytváření aplikací pro Windows.



Obrázek 7: Vývojové prostředí C++ Builder

5 Popis implementace

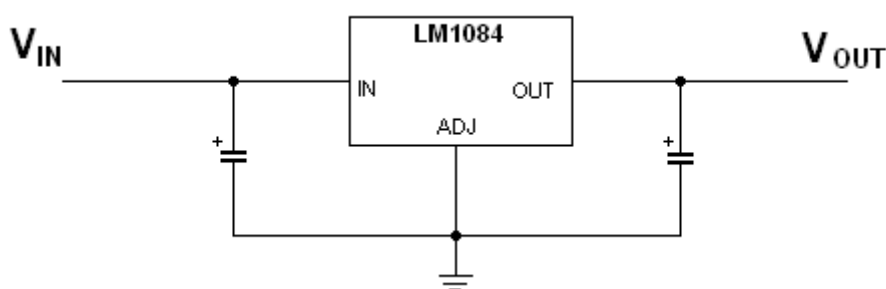
V této kapitole bude popsán postup při návrhu hardwaru a jeho výrobě, vývoj softwaru pro mikroprocesor a vývoj ovládacího programu.

5.1 Vývoj a výroba hardwaru

5.1.1 Vývoj zapojení

Při vývoji hardwarové části práce bylo jako základu použito zakoupeného, již hotového a funkčního, vývojového kitu (IO3), který obsahuje mikroprocesor AT91SAM7S256, řízeného krystalem o kmitočtu 18,432MHz, USB-B konektor a další. To pomohlo zjednodušení celého postupu, neboť tento kit je konstruován tak, že všechny piny mikroprocesoru jsou vyvedeny na dvě lišty, které jsou v rastru 2,54 mm. Díky tomu bylo možné kit umístit přímo na nepájivé pole a snadno připojovat další periferie. [7]

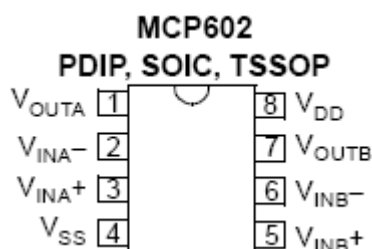
Jelikož vývojový kit potřebuje externí napájení o velikosti 3,3V, byl tento problém řešen jako první. Jako napájecí zdroj byl použit síťový adaptér 5V 1,3A, který je připojen přes konektor X7. Dále byl použit napěťový regulátor LM1084 (IO1), který dává na výstupu požadované napětí 3,3V a maximální teoretický proud 5A. Pro svoji činnost potřebuje pouze 2 kondenzátory (C1, C2).



Obrázek 8: Schéma zapojení regulátoru

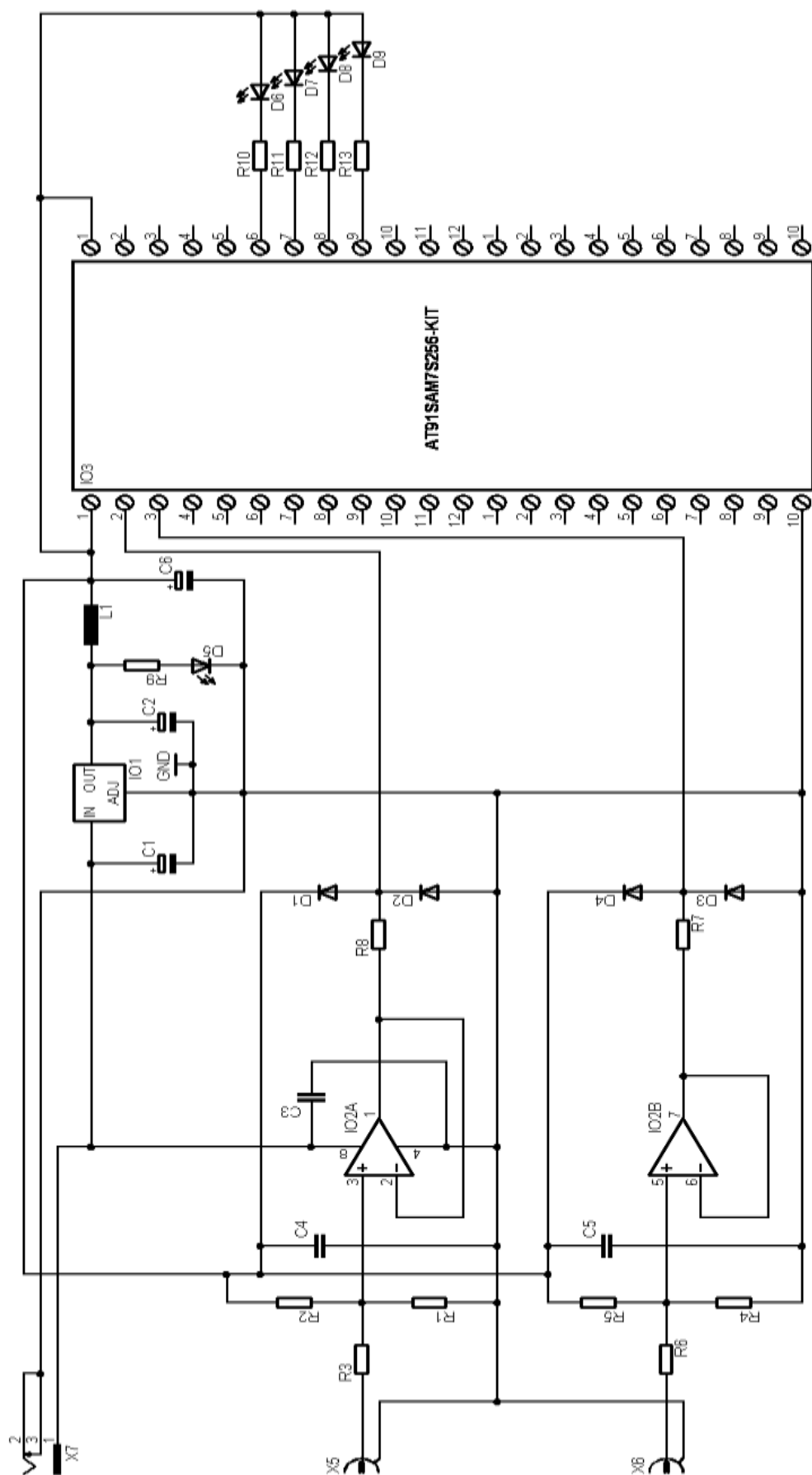
Výstupní napětí regulátoru je dále filtrováno pomocí LC článku (L1, C6), aby jeho zvlnění bylo co nejmenší. Ze stejného důvodu jsou na desce umístěny ještě další tři keramické kondenzátory (C3, C4, C5).

Další částí nutnou pro správnou funkci osciloskopu bylo přidání vstupních zesilovačů, pro každý kanál jeden. K tomu posloužil integrovaný obvod MCP602 (IO2), který obsahuje dvojici operačních zesilovačů s asymetrickým napájením, pracujících až do frekvence 2,8MHz. Obvod se nachází v osmi vývodovém pouzdře, které je usazeno do patice. Operační zesilovače jsou zapojeny jako sledovače, na jejichž neinvertující vstup je přes odpory (R1-R6) přiveden měřený signál (konektory X5 a X6), referenční napětí 3,3V a zem. Základní schéma zapojení jednoho zesilovače již bylo uvedeno na obrázku číslo 4. Toto zapojení bylo na výstupu doplněno o rezistor o velikosti 1,5k Ω (R7, R8) a dvojicí ochranných diod (D1a D2, D3 a D4), pro případ přepětí. Následující obrázek ukazuje rozmístění pinů IO MCP602.



Obrázek 9: Popis vývodů IO MCP602

Pro snadnou kontrolu nastavení osciloskopu bylo přidáno 5 LED. Jedna pro kontrolu napájení (D5), dvě pro indikaci aktivace kanálu (D6, D7), jedna pro indikaci citlivosti spouštění na hranu (D8, svítí – náběžná, nesvítí – sestupná) a poslední LED (D9) se rozsvítí, pokud právě probíhá snímání vzorku.

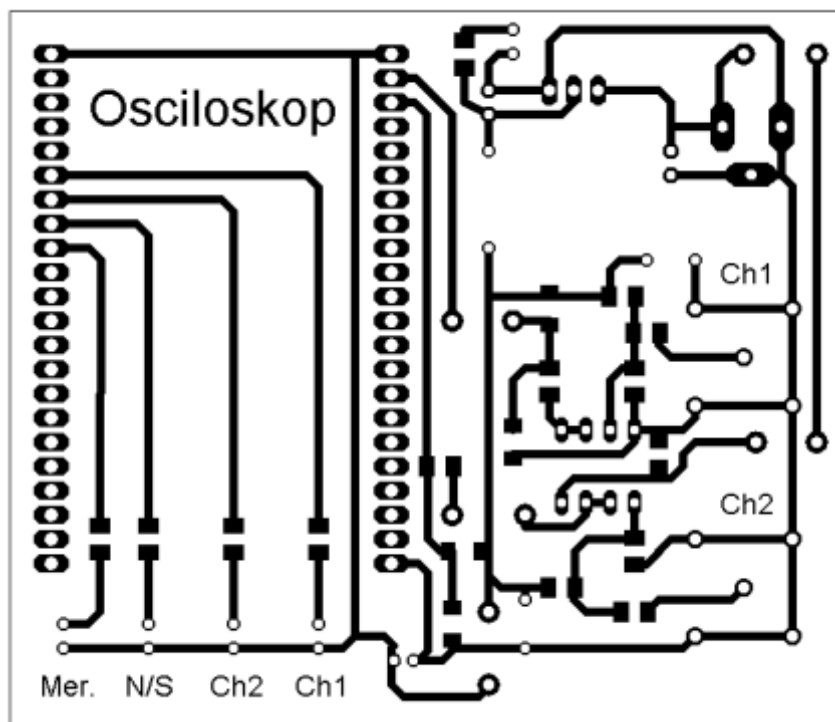


Obrázek 10: Schéma zapojení

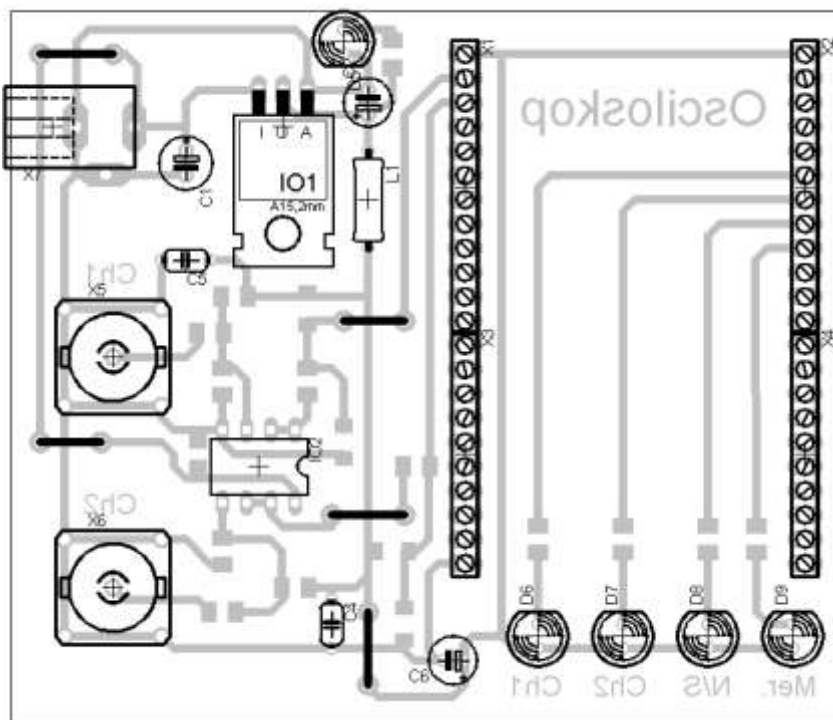
5.1.2 Návrh desky plošných spojů

Po odzkoušení zapojení v nepájivém poli došlo na výrobu desky plošných spojů. K návrhu byl použit program Eagle. V tomto prostředí probíhá návrh desky ve dvou krocích. Nejprve je nakresleno celé schéma zapojení, s využitím schematických značek. Při jejich výběru ze seznamu je třeba dodržet odpovídající rozměry a rozteče vývodů skutečně použitých součástek a konektorů. V dalším kroku jsou součástky s již skutečnými rozměry vhodně rozmístěny na desku. Zároveň je třeba myslet na to, aby se daly snadno pospojovat a pokud možno nedošlo ke křížení drah, které se jinak musí řešit propojkami.

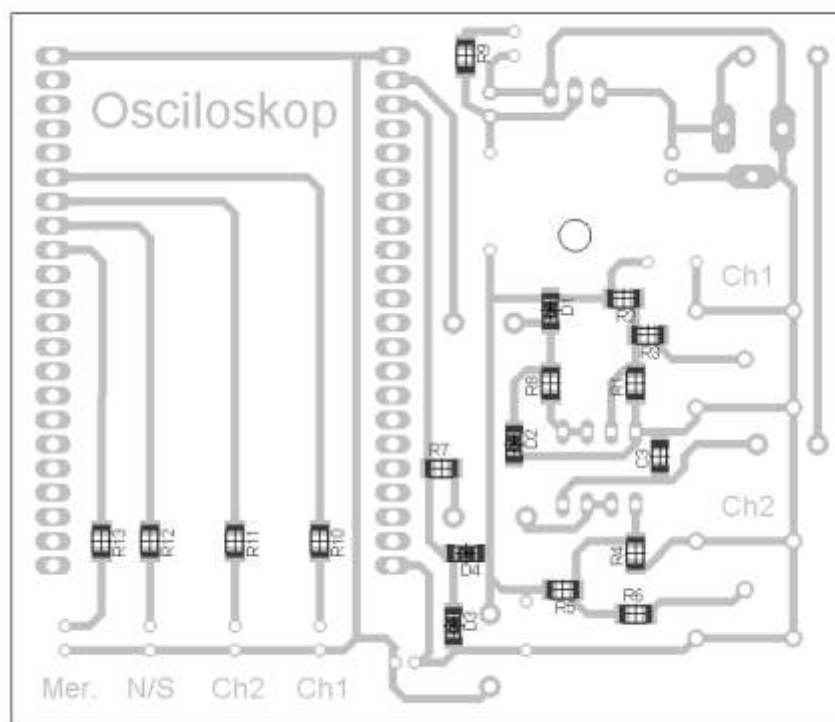
Ve snaze ušetřit místo na desce, byly odpory, ochranné diody a některé kondenzátory navrženy jako součástky pro povrchovou montáž (SMD). Výsledné rozměry desky jsou zhruba 7,5 x 8,5 cm. Dále bude uveden výkres desky plošných spojů, osazovací plánek ze strany součástek (horní strana) a osazovací plánek ze strany spojů obsahující součástky SMD (spodní strana).



Obrázek 11: Deska plošných spojů

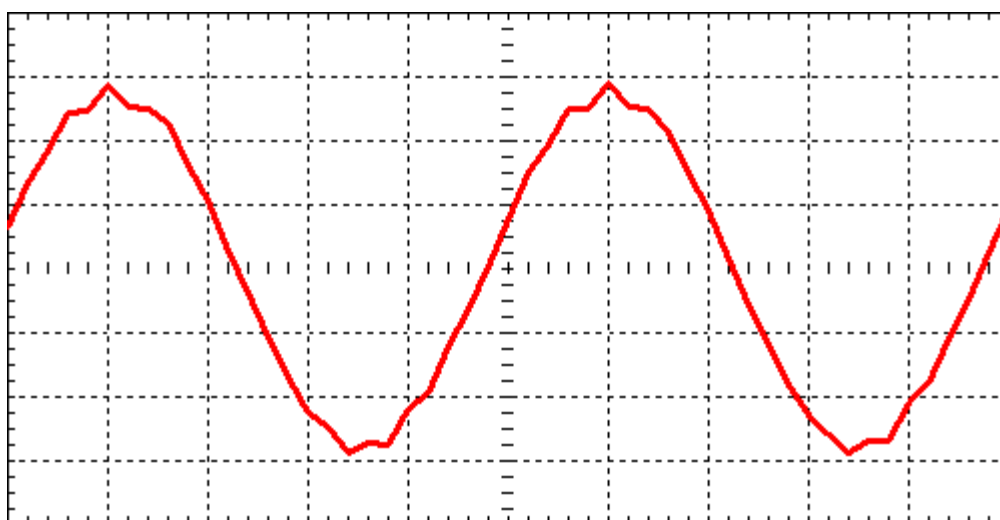


Obrázek 12: Osazovací plánec (strana součástek)



Obrázek 13: Osazovací plánec (Strana spojů)

Po vyleptání a napájení původního návrhu desky, který vycházel přesně ze zapojení na nepájivém poli, se při měření začalo projevovat zkreslení signálu (signál neměl hladký průběh, ale mírně schodovitý), viz Obrázek 14. Toto zkreslení bylo způsobeno nedostatečnou filtrací referenčního napětí pro A/D převodník. V původním zapojení se tato chyba zřejmě neprojevila z důvodu lepšího umístění součástek. K odstranění tohoto nedostatku došlo po přidání filtračního LC článku a filtračních kondenzátorů. Výše uvedené schéma, deska plošných spojů a osazovací plánky již tyto součástky obsahují.



Obrázek 14: Zkreslení signálu vlivem špatné filtrace referenčního napětí

Tabulka 1: Seznam použitých součástek

Označení ve schématu	Popis	Počet kusů
R1, R3, R4, R6	10k Ω SMD	4 ks
R2, R5	4,878k Ω SMD (10k-10k-200k paralelně)	2 ks
R7, R8	1,5k Ω SMD	2 ks
R9 – R13	170k Ω SMD	5 ks
L1	330 μ H	1 ks
C1, C2	10 μ F	1 ks
C3	CK0402 100N X7R SMD	1 ks
C4, C5	CK 100N X7R	2 ks
C6	470 μ F	1 ks
D1-D4	1N4148 SMD	4 ks
D5-D9	LED 5mm	5 ks
IO1	LM 1084	1 ks
IO2	MCP 602	1 ks
IO3	AT91SAM7S256-KIT	1 ks

5.2 Vývoj softwaru

Vývoj softwaru pro digitální osciloskop se skládá ze dvou částí. Za prvé to je software pro samotný mikrokontrolér a za druhé je to ovládací program zajišťující komunikaci a zobrazování na straně počítače. Obě tyto části musí spolupracovat, a proto je bylo zapotřebí vyvíjet současně. V této kapitole budou popsány kroky od oživení samotného mikroprocesoru až po konečný produkt.

5.2.1 Seznámení s mikroprocesorem

Z počátku bylo třeba se s daným mikroprocesorem nejprve seznámit, neboť se jedná o architekturu ARM, která je oproti architekturám známým z výuky poměrně složitější a umožňuje daleko více nastavení.

První program měl velice jednoduchý úkol, nastavení frekvence procesoru a rozblikání LED připojené k I/O portu.

Jelikož po restartu je procesor taktován na pouhých 32kHz (zabudovaný RC oscilátor), je třeba nastavit hlavní oscilátor a PLL (Phase Lock Loop). Obojí se nastavuje pomocí PMC. Nejprve je potřeba povolit hlavní oscilátor a vyčkat na jeho ustálení, které je indikováno nastavením bitu MOSCS registru PMC_SR (Status Register).

```
AT91C_BASE_PMC->PMC_MOR = (AT91C_CKGR_OSCOUNT & (0x8 << 8))
                          | AT91C_CKGR_MOSCEN;
while (!(AT91C_BASE_PMC->PMC_SR & AT91C_PMC_MOSCS));
```

Jakmile dojde ke stabilizaci, může se pokračovat dalším nastavením a to parametrů DIV a MUL registru PMC_PLLR (Phase Lock Loop Register). Pole DIV udává hodnotu, kterou je frekvence oscilátoru dělena a hodnotou v poli MUL je násobena. Po nastavení těchto hodnot je třeba vyčkat na nastavení zámku PLL.

```
AT91C_BASE_PMC->PMC_PLLR = AT91C_CKGR_OUT_0
                          | (AT91C_CKGR_PLLCOUNT & (40 << 8))
                          | (AT91C_CKGR_MUL & (72 << 16))
                          | (AT91C_CKGR_DIV & 14);
while (!(AT91C_BASE_PMC->PMC_SR & AT91C_PMC_LOCK));
```

Posledním krokem je nastavení předděličky a nastavení PLL výstupu. Důležité je zachovat pořadí těchto dvou kroků, aby nedošlo k tomu, že procesor bude taktován vyšší frekvencí, než je maximální povolená hodnota.

```
AT91C_BASE_PMC->PMC_MCKR = AT91C_PMC_PRES_CLK_2;
while (!(AT91C_BASE_PMC->PMC_SR & AT91C_PMC_MCKRDY));
AT91C_BASE_PMC->PMC_MCKR |= AT91C_PMC_CSS_PLL_CLK;
while (!(AT91C_BASE_PMC->PMC_SR & AT91C_PMC_MCKRDY));
```

Po dokončení této procedury je procesor konfigurován k chodu na nastavené frekvenci (MCK-Master Clock). V tomto případě to je $MCK = 18,432\text{MHz} / 14 (\text{DIV}) * 73$

$(\text{MUL} + 1) / 2$ (předdělička) = 48,05MHz. Tato frekvence byla zvolena záměrně, neboť je třeba pro správnou funkci UDP. [6]

Po restartu mikroprocesoru je také aktivní WatchDog, takže pokud není potřeba, je vhodné ho vypnout pomocí zápisu do registru WDTC_WDMR (Watch Dog Timer Controller – Watch Dog Mode Register).

```
AT91C_BASE_WDTC->WDTC_WDMR = AT91C_WDTC_WDDIS;
```

Pro použití PIO je třeba nejprve povolit jeho zdroj vnitřních hodin a nastavit pro používané porty jejich povolení k užívání a výstupu. LED dioda bude spínána proti napájecímu napětí (k rozsvícení dojde při logické hodnotě 0 výstupu). Výsledný program pak vypadá takto:

```
#include <AT91SAM7S256.h>
void delay (unsigned long time);
int main (void) {
...      -nastavení oscilátoru a WatchDogu uvedeno výše
//povolení hodin pro PIO
AT91C_BASE_PMC->PMC_PCER = 1 << 2;
//povolení portu PA0
AT91C_BASE_PIOA->PIO_PER = 0x00000001;
//povolení výstupu na portu PA0
AT91C_BASE_PIOA->PIO_OER = 0x00000001;
while(1){
    //rozsvítit LED
    AT91C_BASE_PIOA->PIO_CODR = 0x00000001;
    delay(1000000);
    //zahesnout LED
    AT91C_BASE_PIOA->PIO_SODR = 0x00000001;
    delay(1000000);
}
void delay (unsigned long time) {
    while(time--);
}
```

5.2.2 Nastavení ADC

Dalším krokem bylo rozchození A/D převodníku a jednoduché zobrazení výsledku pomocí LED.

Při nastavování ADC se začíná podobně jako u většiny dalších periférií, a sice povolením zdroje hodin pro danou periférii. V případě ADC je vhodné tento krok následovat softwarovým resetem převodníku. K jeho vyvolání stačí zapsat log. 1 do příslušného bitu registru ADC_CR (Control Register).

```
//povolení zdroje hodin pro ADC
AT91C_BASE_PMC->PMC_PCER = 1 << 4;
//softwarový reset ADC
AT91C_BASE_ADC->ADC_CR = AT91C_ADC_SWRST;
```

Následuje nastavení parametrů samotného převodníku, které se dělá pomocí registru ADC_MR (Mode Register). Hodnota PRESCAL slouží pro nastavení předděličky pro zdroj hodin $ADCClock = MCK / ((PRESCAL + 1) * 2) = 8\text{MHz}$. Minimální čas potřebný pro start převodu je definován polem Startup Time = $(STARTUP + 1) * 8 / ADCClock = 12\mu\text{s}$. Pro správné navzorkování měřeného signálu je třeba určitý čas Sample and Hold, který se nastavuje pomocí SHTIM. Sample and Hold Time = $SHTIM / ADCClock = 625\text{ns}$. Pomocí pole LOWRES se určuje rozlišení převodu (0 = 10bit, 1 = 8bit). Dále je třeba nastavit, jakým způsobem bude převod spouštěn. Na výběr je buď softwarové, nebo několik hardwarových možností spuštění. V tomto případě bylo zvoleno softwarové spuštění, ovládané přímo z programu pomocí zápisu log. 1 do pole START registru ADC_CR (Control Register). Dokončení převodu každého kanálu je pak indikováno nastavením příslušného bitu registru ADC_SR (Status Register). V poslední části je ještě nastaven Sleep na hodnotu Normal Mode. Pokud je tato položka nastavena na hodnotu Sleep Mode, je převodník „uspán“ a sníží se tak spotřeba celého mikroprocesoru.

Při nastavování hodnot je potřeba dodržet jejich rozsahy dané charakteristikou ADC viz tabulka 2.

```
//prescal 2, startup 1, shtim 5, lowres 1, pouze soft TRG, sleep
normal mode
AT91C_BASE_ADC->ADC_MR = ((2<<8) & AT91C_ADC_PRESCAL)
| ((1<<16) & AT91C_ADC_STARTUP)
```

```

| ( (5<<24) & AT91C_ADC_SHTIM)
| (AT91C_ADC_LOWRES & AT91C_ADC_LOWRES_8_BIT)
| (AT91C_ADC_TRGEN & AT91C_ADC_TRGEN_DIS)
| (AT91C_ADC_SLEEP & AT91C_ADC_SLEEP_NORMAL_MODE);

```

Na závěr je ještě třeba zápisem do registru ADC_CHER (CHanel Enable Register) povolit potřebné kanály převodníku. V tomto případě to jsou kanály 4 a 5, neboť jsou přímo vyvedeny na piny mikroprocesoru a nejsou sdíleny s žádnou jinou periferií. [5]

```

//povoleni kanalu 4 a 5
AT91C_BASE_ADC->ADC_CHER = (1<<4) | (1<<5);

```

Tabulka 2: Mezní hodnoty nastavení ADC [5]

Parametr	Podmínky	Min	Typ	Max	Jednotky
Frekvence hodin ADC	10-bit rozlišení			5	MHz
Frekvence hodin ADC	8-bit rozlišení			8	MHz
Startovací čas (Startup Time)	Návrat z Idle Mode			20	μs
Vzorkovací čas (Track and Hold)		600			ns
Čas převodu	ADCClock = 5MHz			2	μs
Čas převodu	ADCClock = 8MHz			1,25	μs
Vzorkovací frekvence	ADCClock = 5MHz			384	kSPS
Vzorkovací frekvence	ADCClock = 8MHz			533	kSPS

Výsledný program, který sloužil k otestování správnosti nastavení převodníku, prováděl v nekonečné smyčce A/D převod. Pro zobrazování byly použity 4 LED, které indikovaly stav horních čtyř bitů převodu. Volba, který kanál bude zobrazen, byla původně možná pouze softwarově, což pro otestování stačilo. V pozdější verzi byla možnost volby zobrazovaného kanálu pomocí tlačítka, pro názornost je ovšem uvedena pouze první z variant.

```

...
BYTE ch1,ch2;
while(1){
//zahajit AD prevod
AT91C_BASE_ADC->ADC_CR = AT91C_ADC_START;
//cekani na dokonceni prevodu na kanalu 4 a 5
while(!(AT91C_BASE_ADC->ADC_SR & AT91C_ADC_EOC4) &&
      !(AT91C_BASE_ADC->ADC_SR & AT91C_ADC_EOC5));

ch1 = AT91C_BASE_ADC->ADC_CDR4;
ch2 = AT91C_BASE_ADC->ADC_CDR5;

//zhasnuti vseh LED
AT91C_BASE_PIOA->PIO_SODR = 0x0000000f;
//rosviceni LED podle hornich 4bitu prevodu
AT91C_BASE_PIOA->PIO_CODR = (ch1>>4) & 0xf;
}

```

5.2.3 Komunikace po USB

Pro řešení programu pro mikroprocesor, komunikujícího s PC a umožňující využití standardu HID, byl použit vzorový příklad z instalace programu μ Vision. Tento program je navržen tak, že po připojení mikroprocesoru k počítači dojde k jeho rozpoznání jako zařízení standardu HID a je použit příslušný ovladač. Dále je s jeho pomocí možno ovládat 4 LED a monitorovat stav 4 spínačů, připojeným k portům PIO. Tento příklad obsahuje soubor s vlastní definicí datových typů, ten bude uveden v příloze.

Pokud je ze strany počítače odeslán požadavek na čtení, je volána procedura `GetInReport(void)`, která slouží pro obsluhu tohoto požadavku. Data, která mají být přečtena, je třeba uložit do globální proměnné `InReport` typu `BYTE`. Podobně pak pokud přijde požadavek na zápis, je volána procedura `SetOutReport(void)` a data jsou přístupná v globální proměnné `OutReport` typu `BYTE`. Označení `In` a `Out` zde může být zavádějící, ale je třeba ho chápat jako vstupní (`in`) a výstupní (`out`) ze strany počítače.

Na straně ovládacího programu bylo základním úkolem získání platného handlu zařízení, aby bylo možné použít funkce `ReadFile` a `WriteFile`. K tomu byl využit kód dostupný na www.alanmacek.com/usb, který využívá knihoven z Windows DDK.

Skládá se ze dvou souborů – usb.h a usb.c. Autor si k užívání a šíření tohoto kódu klade jedinou podmínku, a sice tu, že chce být o jejich využití informován emailem (jeho text je součástí přílohy).

Dále bylo ještě třeba vytvořit knihovny importů .lib dynamických knihoven HID.dll a SetupAPI.dll. To lze snadno udělat pomocí programu IMPLIB.EXE, který je součástí instalace C++ Builderu, stačí ho spustit s patřičnými parametry.

```
implib.exe SetupAPI.lib SetupAPI.dll
```

K získání handle je k dispozici funkce connectToUSBHIDDevice, jejímiž parametry jsou VID a PID zařízení. Návrátovou hodnotou je buď platný handle zařízení nebo INVALID_HANDLE_VALUE v případě jeho nenalezení. Tato funkce však měla jeden nedostatek a sice ten, že postupně kontrolovala seznam zařízení HID a pokud kterékoliv z nich nešlo otevřít (funkce OpenFile vrátila hodnotu INVALID_HANDLE_VALUE), bylo prohledávání ukončeno a hledané zařízení nebylo nalezeno. Tato chyba byla způsobena tím, že u některých zařízení systém nepovolí jejich handle získat. Proto byl cyklus prohledávání upraven a sice tak, aby prohledal prvních 10 zařízení bez ohledu na to, jestli je lze otevřít. Pokud dojde k jeho nalezení dříve, cyklus je ukončen.

5.2.4 Vzorová aplikace A/D převodníku

V první části zadání bakalářské práce se hovoří o vytvoření vzorové aplikace A/D převodníku. Ta by měla sloužit ke kontinuálnímu snímání hodnoty napětí na vstupu převodníku a její zobrazování v počítači.

Okno aplikace obsahuje tyto části:

- Tlačítko, které slouží ke spuštění nebo zastavení měření
- Okno pro zobrazení hodnoty měřeného napětí (pod ním se pro kontrolu zobrazuje hodnota převodu v rozsahu 0 – 255)
- RadioGroup pro výběr měřeného kanálu
- Timer pro zajištění pravidelného měření



Obrázek 15: Vzorová aplikace A/D převodníku

Získání handlu zařízení je zajištěno již v konstruktoru. Pokud funkce `connectToUSBHIDDevice` vrátila platný handle, je do pole Caption formuláře vepsáno OK, pokud je vrácena hodnota `INVALID_HANDLE_VALUE`, je vepsáno Nepřipojeno.

Při stisknutí tlačítka dojde ke spuštění/zastavení timeru, který zajišťuje každých 100ms přečtení nové hodnoty převodu, její přepoččet na napětí a zobrazení. Pro čtení je použita funkce `ReadFile` s těmito parametry:

- `USBHandle` – handle zařízení získaný funkcí `connectToUSBHIDDevice`
- `pp` – ukazatel na dvouprvkové pole typu `unsigned char` (čteny jsou dva bajty, z nichž první je vždy nulový a druhý obsahuje přečtenou hodnotu)
- `sizeof(pp)` – počet bajtů, které mají být přečteny
- `&d` – ukazatel na proměnnou, do které je uložen počet skutečně přečtených bajtů
- `NULL` – zde může být uveden ukazatel na strukturu `OVERLAPPED`

```
DWORD d;
float V;
unsigned char pp[2];
if (ReadFile(USBHandle, pp, sizeof(pp), &d, NULL)) {
    V = (3.3 / 255) * pp[1];
    Edit1->Text = ceil(V * 100) / 100;
}
else{
    Edit1->Text = "Chyba! ";
}
//zobrazeni hodnoty prevodu
```

```
Label1->Caption = pp[1];
```

Při změně kanálu je automaticky volána procedura `RadioButton1Click`. Druhá položka pole `pp` je nastavena podle toho, který kanál je zvolen a pomocí funkce `WriteFile` jsou data odeslána. Parametry funkce `WriteFile` jsou obdobné funkci `ReadFile` a proto již nejsou uváděny.

```
DWORD d;  
unsigned char pp[2];  
pp[0] = 0;  
if (RadioButton1->Checked == true) {  
    pp[1] = 1;  
}  
if (RadioButton2->Checked == true) {  
    pp[1] = 2;  
}  
if (!WriteFile(USBHandle, pp, sizeof(pp), &d, NULL)) {  
    MessageBox(Handle, "Odeslani selhalo", "chyba", MB_OK);  
}
```

Na straně mikroprocesoru je procedura `GetInReport` napsána tak, že je zahájen A/D převod, vyčkáno na jeho dokončení a do proměnné `InReport` je nahrána hodnota buď z kanálu 1, nebo z kanálu 2. To záleží na hodnotě logické proměnné `kanal1`.

```
BYTE ch1, ch2;  
BOOL kanal1;  
  
void GetInReport (void) {  
    //zahajit AD prevod  
    AT91C_BASE_ADC->ADC_CR = AT91C_ADC_START;  
    //cekani na dokonceni prevodu na kanalu 4 a 5  
    while(!(AT91C_BASE_ADC->ADC_SR & AT91C_ADC_EOC4) &&  
          !(AT91C_BASE_ADC->ADC_SR & AT91C_ADC_EOC5));  
  
    ch1 = AT91C_BASE_ADC->ADC_CDR4;  
    ch2 = AT91C_BASE_ADC->ADC_CDR5;  
    InReport = kanal1 ? ch1 : ch2;  
}
```

Procedura SetOutReport slouží k nastavení proměnné kanal1 (true = ch1, false = ch2) a k indikaci zvoleného kanálu pomocí dvou LED.

```
void SetOutReport (void) {  
    if (OutReport == 1) kanal1 = TRUE;  
    else if (OutReport == 2) kanal1 = FALSE;  
    if (OutReport & 0x01) pPIO->PIO_CODR = LED1;  
    else pPIO->PIO_SODR = LED1;  
    if (OutReport & 0x02) pPIO->PIO_CODR = LED2;  
    else pPIO->PIO_SODR = LED2;  
}
```

Zbytek programu jako je nastavení ADC již byl uveden dříve.

5.2.5 Řídící příkazy osciloskopu

Jelikož pro nastavování osciloskopu se nedá vystačit pouze s výběrem kanálu, jako u aplikace A/D převodníku, bylo zapotřebí nejprve vymyslet systém řídicích příkazů, které umožní složitější nastavení. Po vzoru [1] byl zvolen 3bajtový formát dat, pro aplikaci tohoto typu je to zcela dostačující a obsluha těchto příkazů není nikterak složitá. První bajt obsahuje vždy ID příkazu, zbytek pak nese data.

Příkaz s ID = 0 slouží k nastavování kanálu, citlivosti na hranu a spouštěcího napětí. První bajt tedy nese hodnotu 0, druhý spouštěcí napětí a třetí má ve spodních třech bitech uvedeno nastavení kanálů a citlivost na hranu.

Příkaz s ID = 1 nastavuje hodnotu zpoždění mezi snímáním jednotlivých vzorků a startuje samotné snímání. První bajt má hodnotu 1, druhý spodních 8 bitů a třetí horních 8 bitů zpoždění.

Více příkazů nebylo zapotřebí, ale díky použití takového formátu je přidání dalšího poměrně snadné. Pro názornost jsou ještě uvedeny v následující tabulce.

Tabulka 3: Seznam řídicích příkazů

	ID = 0	ID = 1
data 1	0	1
data 2	spouštěcí napětí	spodních 8 bitů zpoždění
data 3	kanál 1 kanál 2 citlivost na hranu	horních 8 bitů zpoždění

Pro zasílání řídicích příkazů byla v ovládacím programu vytvořena jednoduchá procedura PoslatData. Její návratová hodnota typu bool indikuje úspěšnost odeslání všech dat, volána je s těmito parametry:

- USBHandle – handle zařízení
- *data – ukazatel na pole typu unsigned char, které obsahuje data k odeslání
- pocet – počet dat k odeslání

```
bool PoslatData(HANDLE USBHandle, unsigned char * data,
               unsigned char pocet){
    DWORD d;
    unsigned char buf[2], i;
    buf[0] = 0;
    for (i = 0; i < pocet; i++) {
        buf[1] = data[i];
        if (!WriteFile(USBHandle, buf, sizeof(buf), &d, NULL)) {
            MessageBox(Form1->Handle, "Odeslani selhalo",
                       "chyba", MB_OK);
            return false;
        }
    }
    return true;
}
```

5.2.6 Časování

Z důvodu potřeby pokud možno přesné rekonstrukce měřeného signálu a možnosti měnit časovou základnu signálu, bylo zapotřebí, aby ke vzorkování docházelo v přesně daných časových intervalech. Původně byl pro jednoduchost použit pouze prázdný for cyklus. Pro otestování funkčnosti to bylo dostačující, ovšem pro potřeby osciloskopu bylo třeba najít lepší řešení. Pro účely přesného odměřování času je u mikroprocesorů řady AT91SAM7S k dispozici hned 5 časovačů (Periodic Interval Timer, Real Time Timer a třikrát Timer Counter). Jako nejvhodnější k potřebnému úkolu se jevil první z nich PIT.

PIT se vlastně skládá hned ze dvou čítačů. První z nich je 20bitový a čítá MCK/16. Čítání probíhá od nuly až do hodnoty definované v poli PIV registru PIT_MR. Při dosažení této hodnoty dojde k indikaci přetečení a je inkrementován druhý 12bitový čítač. [5]

Nastavení PIT je velice jednoduché, stačí nastavit požadovanou hodnotu pro přetečení a spustit samotný čítač.

```
//nastaveni hodnoty pro pretečení
AT91C_BASE_PITC->PITC_PIMR = 100 & AT91C_PITC_PIV;
//spusteni PIT
AT91C_BASE_PITC->PITC_PIMR |= AT91C_PITC_PITEN;
```

V programu je pak na požadovaném místě vložen cyklus čekající na nastavení příznaku přetečení. Tento příznak je poté třeba vynulovat přečtením registru PITC_PIVR.

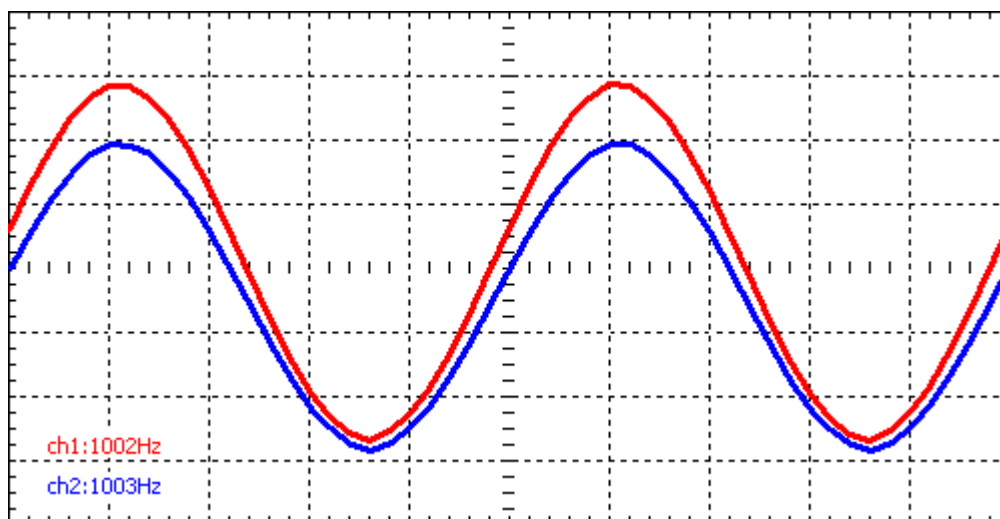
```
//cekani na pretečení PIT
while (!AT91C_BASE_PITC->PITC_PISR & AT91C_PITC_PITS);
//prečtení hodnoty kvůli nulování příznaku přetečení
temp = AT91C_BASE_PITC->PITC_PIVR;
```

5.2.7 Aplikace digitálního osciloskopu

Konečná aplikace digitálního osciloskopu spojuje všechny výše uvedené poznatky a ještě něco navíc.

Snímáno je celkem 51 vzorků pro každý kanál (definováno konstantou PocetKanalů), takže po každém měření je třeba do počítače přenést 52 nebo 103 vzorků (51 pro každý aktivní kanál + 1 pro označení začátku). Stahovaná data jsou nejprve uložena do pomocného pole tempBuffer, ze kterého jsou poté roztržena do příslušných bufferů (Buffer1 a Buffer2).

Vykreslování probíhá do pomocné bitmapy (Bitmapa), která má stejné rozměry jako použitá komponenta TImage formuláře. Nejprve je do ní pomocí procedury KresliOsy nakreslena základní mříž, pak aktivní kanály (kanál 1 červeně, kanál 2 modře). Takto vzniklá bitmapa je překreslena do Image1. Díky tomu, že vykreslování neprobíhá přímo do image, není patrné blikání. Pomocí položky v menu Soubor->Uložit lze zobrazenou bitmapu, doplněnou o frekvence signálů, uložit do souboru.



Obrázek 16: Obrázek uložený pomocí položky menu

6 Závěr

6.1 Splnění cílů práce

Všechny cíle dané v zadání práce se podařilo splnit. Při realizaci byl místo zadaného mikroprocesoru AT91SAM7S64 použit mikroprocesor AT91SAM7S256, který se od zadaného liší pouze velikostí pamětí. Tato změna byla konzultována s vedoucím práce. Pro měření byl použit jeho zabudovaný A/D převodník a pro komunikaci zabudované USB rozhraní. Vytvořený digitální osciloskop umožňuje měření signálů v rozpětí $\pm 6V$ do frekvence až 10kHz (při 25 vzorcích na periodu). Pro jeho ovládání jsou k dispozici základní nastavení, jako je hodnota a hrana spouštěcího napětí, časová základna či volba měřených kanálů. Funkce osciloskopu byly dokonce oproti zadání rozšířeny o výpočet frekvence měřeného signálu a možnost ukládání obrazovky jako bitmapu.

6.2 Další možnosti rozšíření

Jako další možné rozšíření osciloskopu by mohlo být zvýšení maximální možné měřené frekvence, kterého by se dosáhlo zvýšením vzorkovací frekvence (např. pro snímání signálu s frekvencí 100kHz, při potřebě alespoň 25 snímků na periodu, by bylo třeba vzorkování 2,5MSPS). To by ovšem znamenalo nutnost použití externích A/D převodníků. Dalším přínosem pro použití by bylo využití možností napájení ze sběrnice USB.

7 Seznam použité literatury

- [1] MATOUŠEK, David . *Udělejte si z PC v Delphi*. 1. vydání. Praha : Nakladatelství BEN - technická literatura, 2003. 271 s. ISBN 80-7300-111-X.
- [2] VÁŇA, Vladimír . *ARM pro začátečníky*. 1. vydání. Praha : Nakladatelství BEN - technická literatura, 2009. 196 s. ISBN 978-80-7300-246-6.
- [3] AXELSON, Jan . *USB Complete*. 2. vydání, Madison : Lakeview Research, 2001. 523 s. ISBN 0-9650819-5-8.
- [4] ANDERSON, Don; DZATKO, Dave. *Universal Serial Bus System Architecture*. 2. vydání. USA : Addison-Wesley, 2001. 505 s. ISBN 0-201-30975-0.
- [5] Atmel Corporation. *AT91SAM7S Series Preliminary* [online]. ©2009 [cit. 2009-05-09]. Dostupný z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc6175.pdf>.
- [6] Atmel Corporation. *Getting Started with AT91SAM7S Microcontrollers* [online]. ©2007 [cit. 2009-05-10]. Dostupný z WWW:
<http://www.atmel.com/dyn/resources/prod_documents/doc6293.pdf>
- [7] Kramara. *AT91SAM7S256KIT Microcontrollers* [online]. ©2008 [cit. 2009-05-10]. Dostupný z WWW: http://kramara.com/podklady/AT91SAM7S256-kit/doku_r1.pdf.
- [8] AT91SAM7S - 1.díl: *Seznámení s obvody*. Elektronický magazín [online]. ©20.1.2009, [cit. 2010-05-20]. Dostupný z WWW:
<http://pandatron.cz/?606&at91sam7s_-_1.dil:_seznameni_s_obvody>. ISSN 1803-6007.
- [9] AT91SAM7S - 2.díl: *Základní výbava a instalace*. Elektronický magazín [online]. ©6.2.2009, [cit. 2010-05-20]. Dostupný z WWW:
<http://pandatron.cz/?636&at91sam7s_-_2.dil:_zakladni_vybava_a_instalace>.
- [10] AT91SAM7S - 3.díl: *PIO a první příklady*. Elektronický magazín [online]. ©4.3.2009, [cit. 2010-05-20]. Dostupný z WWW:
<http://pandatron.cz/?668&at91sam7s_-_3.dil:_pio_a_prvni_prikklady>.

- [11] AT91SAM7S - 4.díl: *Clock Generator a PMC* Elektronický magazín [online]. ©3.4.2009, [cit. 2010-05-20]. Dostupný z WWW: <http://pandatron.cz/?718&at91sam7s_-_4.dil:_clock_generator_a_pmc>.
- [12] AT91SAM7S - 5.díl: *Interrupt* Elektronický magazín [online]. ©1.5.2009, [cit. 2010-05-20]. Dostupný z WWW: <http://pandatron.cz/?764&at91sam7s_-_5.dil:_interrupt>.
- [13] Atmel Corporation. *AT91SAM7S-EK Software Package for IAR 5.2, Keil and GNU* [online]. ©2010 [cit. 2009-05-09]. Dostupný z WWW: <http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3784>.
- [14] BRTNÍK, Bohumil. *Elektrická měření* : Skriptum Vysoké školy polytechnické Jihlava. Vydavatelství VŠP Jihlava 2007.
- [15] Microchip Technology Inc.: *Datasheet MCP601/2/3/4* [online]. ©2004, [cit. 2010-05-18]. Dostupný z WWW: <http://www.datasheetcatalog.org/datasheet/microchip/21314e.pdf>

8 Seznam obrázků

Obrázek 1: Zpracování instrukcí jádrem ARM7TDMI	14
Obrázek 2: Blokové schéma digitálního osciloskopu	18
Obrázek 3: Blokové schéma osciloskopu s AT91SAM7S64	19
Obrázek 4: Schéma zapojení vstupního zesilovače jednoho kanálu.....	20
Obrázek 5: Vývojové prostředí KEIL μ Vision 4.....	23
Obrázek 6: Program SAM-BA	25
Obrázek 7: Vývojové prostředí C++ Builder.....	25
Obrázek 8: Schéma zapojení regulátoru	26
Obrázek 9: Popis vývodů IO MCP602	27
Obrázek 10: Schéma zapojení.....	28
Obrázek 11: Deska plošných spojů.....	29
Obrázek 12: Osazovací plánec (strana součástek).....	30
Obrázek 13: Osazovací plánec (Strana spojů).....	30
Obrázek 14: Zkreslení signálu vlivem špatné filtrace referenčního napětí	31
Obrázek 15: Vzorová aplikace A/D převodníku.....	39
Obrázek 16: Obrázek uložený pomocí položky menu.....	44

9 Seznam tabulek

Tabulka 1: Seznam použitých součástek	32
Tabulka 2: Mezní hodnoty nastavení ADC	36
Tabulka 3: Seznam řídicích příkazů	42

10 Seznam zkratek

A/D – analog to digital

ADC – analog to digital converter

ADC_CHER – analog to digital converter channel enable register

ADC_CR – analog to digital converter control register

ADC_MR – analog to digital converter mode register

ADC_SR – analog to digital converter status register

ADCLK – analog to digital clock

ADVREF – analog to digital voltage reference

AIC – advanced interrupt controller

ARM – advanced RISC machine

CD – compact disc

CKGR – clock generator

DBGU – debugger unit

DIV – division

DMA – direct memory access

HID – human interface device

I/O – input/output

ID – identity

IEEE – institute of electrical and electronical engineers

IO – integrovaný obvod

JTAG – joint test action group

LED – light emitted diode

MC – memory controller

MCK – master clock

MIPS – million instructions per second

MOSCS –

MUL – multiplex

OCP – on chip debugger

PC – personal computer

PCK – processor clock

PID – product identification

PIO – parallel input output

PIO_CODR – parallel input output clear output data register

PIO_ODSR – parallel input output output data status register

PIO_SODR – parallel input output set output data register

PIT – periodic interval timer

PIT_MR – periodic interval timer mode register

PIT_PIIR – periodic interval timer periodic interval image register

PIT_PIVR – periodic interval timer periodic interval value register

PIT_SR – periodic interval timer status register

PITS – periodic interval timer status

PIV – periodic interval value

PLL – phase lock loop

PMC – power management controller

PMC_PLLR – power manager controller phase lock loop register

PMC_SR – power manager controller status register

PWM – pulse width modulator

PWMC – pulse width modulator controller

RISC – reduced instruction set computer

ROM – read only memory

RSTC – reset controller

RTT – real time timer

SHTIM – sample and hold time

SMD – surface mount device

SPI – serial peripheral interface

SPS – samples per second

SRAM – static random access memory

TC – timer counter

TWI – two wire interface

UART – universal asynchronous receiver transmitter

UDP – USB device port

UDPCK – USB device port clock

USART – universal synchronous asynchronous receiver transmitter

USB – universal serial bus

VID – vendor identification

WDT –watchdog timer

WDTC_WDMR – watchdog timer counter watchdog mode register

11 Seznam příloh

Příloha 1: Type.h - vlastní definice datových typů pro program mikroprocesoru

Příloha 2: informační email o používání zdrojového kódu adresáta

Příloha 1: Type.h - vlastní definice datových typů pro program mikroprocesoru

```
#ifndef __TYPE_H__
#define __TYPE_H__

#ifndef NULL
#define NULL ((void *)0)
#endif

#ifndef FALSE
#define FALSE (0)
#endif

#ifndef TRUE
#define TRUE (1)
#endif

typedef unsigned char  BYTE;
typedef unsigned short WORD;
typedef unsigned long  DWORD;
typedef unsigned int   BOOL;

#endif /* __TYPE_H__ */
```

Příloha 2: informační email o používání zdrojového kódu adresáta

Od: "Jakub Jelínek" <jakub.jelda@centrum.cz>

Komu: <al@alanmacek.com>

Předmět: I am using your code

Datum: 21.05.2010 17:01

Velikost: 727 B

Hi,

I am using your code found at <http://www.alanmacek.com/usb/> to get a handle of USB HID device. I am writing an application to communicate with Atmel's AT91SAM7S256 microprocessor, configured as a HID device. It should be a simple digital oscilloscope at the end. Your code is very helpful for me, thanks for it.

With many thanks

Jakub Jelinek