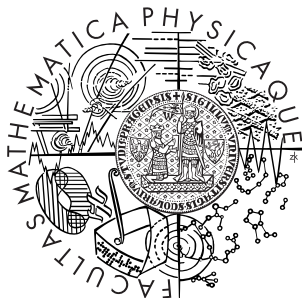


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

## BAKALÁŘSKÁ PRÁCE



Jana Ringelová

### **Procesorový systém pro řízení experimentu pro studium nabíjecích procesů prachových mikročástic**

Katedra fyziky povrchů a plazmatu

Doc. RNDr. Lubomír Přech, Dr.  
Studijní obor: Obecná fyzika

2007

Děkuji především panu docentu Přechovi za četné konzultace, pomoc při řešení problémů a jeho obětavost. Můj dík patří také Vojtěchu Hrubému za jeho cenné rady. Svým rodičům děkuji za podporu ve studiu.

Prohlašuji, že jsem svou bakalářskou práci napsala samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejněním.

V Praze dne 9.8.2007

Jana Ringelová

Název práce: Procesorový systém pro řízení experimentu pro studium nabíjecích procesů prachových mikročástic  
Autor: Jana Ringelová  
Katedra (ústav): Katedra fyziky povrchů a plazmatu  
Vedoucí bakalářské práce: Doc. RNDr. Lubomír Přech, Dr.  
e-mail vedoucího: Lubomir.Prech@mff.cuni.cz

Abstrakt: V předložené práci se zabývám možnostmi využití procesoru DSP Freescale 56F8014 pro řízení experimentu pro studium nabíjení prachových mikročástic. Práce popisuje generování referenčního signálu vhodného tvaru pro řízení kvadrupólové pasti jednak s využitím externího DA převodníku a jednak s využitím kanálů PWM ve funkci DA převodníků. Je zde též navržen způsob zahrnutí gravitační kompenzace a složky pro tlumení a stabilizaci kmitů částice. Na závěr práce jsou shrnuty praktické poznatky použitelné pro konstrukci skutečného řízení v nově budovaném experimentu.  
Klíčová slova: kvadrupól, DSP Freescale 56F8014, DA převodník, PWM.

Title: Microprocessor control system for the experimental study of charging processes of dust microparticles  
Author: Jana Ringelová  
Department: Department of Surface and Plasma Science  
Supervisor: Doc. RNDr. Lubomír Přech, Dr.  
Supervisor's e-mail address: Lubomir.Prech@mff.cuni.cz

Abstract: In the present work we study possibilities for utilization of the processor DSP Freescale 56F8014 for control of experiment dealing with charging processes of dust microparticles. The work describes generating a reference signal of the appropriate shape for control of the quadrupole trap using an external DA converter or PWM channels operating as DA converter. We also show an arrangement for gravitation compensation and particle oscillation stabilization and damping. Finally, practical knowledge is summarized, sufficient for the design of the real control device for currently built experiment.  
Keywords: quadrupole, DSP Freescale 56F8014, DA converter, PWM.

# Obsah

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Úvod</b>   | <b>5</b>  |
| <b>2</b> | <b>Teoretická část</b>  | <b>6</b>  |
| 2.1      | Experimentální aparatura pro měření nabíjecích procesů prachových částic                          | 6         |
| 2.2      | Kvadrupól a jeho buzení . . . . .   | 7         |
| <b>3</b> | <b>Cíl práce</b>  | <b>8</b>  |
| <b>4</b> | <b>Seznámení s použitými DSP a DAC, jejich vývojovými moduly a softwarem pro programování DSP</b> | <b>9</b>  |
| 4.1      | DSP 56F8014 . . . . .   | 9         |
| 4.2      | DAC8814 . . . . .   | 9         |
| 4.3      | Demonstrační deska s DSP 56F8014 . . . . .  | 11        |
| 4.4      | Vývojový modul DAC8814 EVM . . . . .  | 12        |
| 4.5      | Processor Expert– software pro programování DSP . . . . .   | 15        |
| <b>5</b> | <b>Popis vlastní konstrukce a programování</b>  | <b>19</b> |
| 5.1      | Program v C . . . . .   | 19        |
| 5.2      | Propojení modulů pro DSP a DAC . . . . .  | 19        |
| 5.3      | Program pro DSP . . . . .   | 23        |
| 5.4      | Gravitační kompenzace a tlumení kmitů prachové částice . . . . .                                  | 23        |
| <b>6</b> | <b>Výsledky a jejich diskuse</b>  | <b>25</b> |
| 6.1      | Řešení pomocí externího DAC . . . . .   | 25        |
| 6.2      | Řešení pomocí PWM . . . . .   | 26        |
| 6.3      | Možnosti vylepšení a zrychlení . . . . .  | 28        |
| <b>7</b> | <b>Závěr</b>  | <b>30</b> |
| <b>8</b> | <b>Příloha: Nastavení jednotlivých beanů</b>  | <b>31</b> |
| <b>9</b> | <b>Příloha: Kód programu pro DSP</b>  | <b>33</b> |
|          | <b>Literatura</b>   | <b>43</b> |

# 1 Úvod

Prachové částice jsou částice pevného skupenství o rozměrech řádově několika nanometrů až stovek mikrometrů. Vzhledem k jejich rozměrům má na jejich chování významný vliv jak gravitační, tak elektromagnetická síla.

Prachové částice se nacházejí jak na Zemi, tak ve vesmíru, kde tvoří přibližně 1% mezihvězdné hmoty. Jejich malé rozměry způsobují, že se mohou nabíjet na velké měrné náboje. Na tomto nabíjení se podílejí následující procesy: záchyt primárních částic (elektronů a iontů) na povrchu prachové částice, sekundární elektron-elektronová emise vyvolaná dopadem primárních elektronů, fotoemise vyvolaná fotony UV-zářením apod.

Pro zkoumání nabíjecích procesů prachových mikročástic byla na KFPP vyvinuta experimentální aparatura, která funguje na principu zachycení částice do elektrodynamického kvadrupólu a pozorování jejího nábojového stavu při ozařování elektronovým nebo iontovým svazkem o dané energii.

Pro poměr náboje a hmotnosti prachové částice uvádí Žilavý [1] přibližný vztah

$$\frac{|Q|}{m} \doteq \pi^2 r_0^2 \frac{f_{ac} f_z}{V_{ac}^{ef}}, \quad (1)$$

kde  $r_0$  je vnitřní poloměr prostřední elektrody kvadrupólu (viz. obrázek 1),  $f_z$  je měřená frekvence základních kmitů částice ve směru osy kvadrupólu (osa  $z$ ),  $f_{ac}$  frekvence referenčního signálu ze sinusového generátoru a  $V_{ac}^{ef}$  efektivní hodnota jeho napětí.  $|Q|$  je velikost náboje částice a  $m$  její hmotnost. Pro  $f_{ac}/f_z \geq 10$  udává tento vztah výsledek s přesností lepší než 1,5 %.

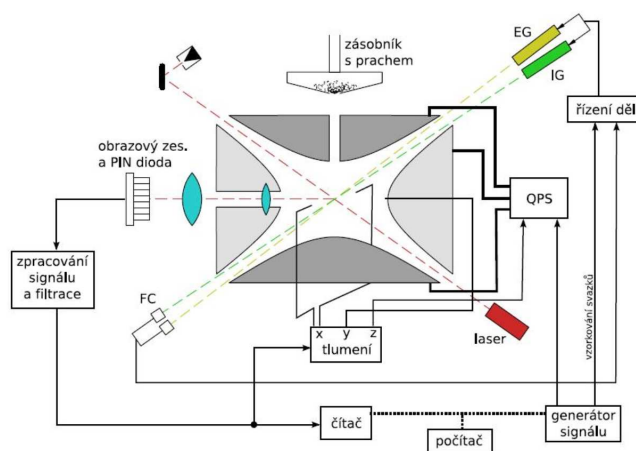
V současné době je na KFPP vyvíjena nová vylepšená aparatura, která by dovolovala širší rozsahy energií elektronů a iontů nabíjecích prachové částice a zkoumání fotoemise fotony UV nebo synchrotronového záření. Aby zařízení mohlo být použito v experimentech s fotoemisí, musí být pozměněna geometrie kvadrupólu.

Cílem této práce je ověřit možnosti konstrukce sinusového generátoru pro tuto novou aparaturu pomocí procesoru DSP 56F8014.

## 2 Teoretická část

### 2.1 Experimentální aparatura pro měření nabíjecích procesů prachových částic

Experimentální zařízení [1] vycházející z aparatury zkonstruované I. Čermákem [2], užívá pro generování sinusového referenčního napětí jednokanálový komerční generátor PHILIPS navíc doplněný o obvody vzorkování elektronového svazku a kvadrupólového napětí a další. Problémem tohoto sinusového generátoru je, že nedokáže zajistit kontinuitu fáze při změně frekvence. Ohledně celého experimentálního uspořádání Beránek et al. [3] uvádí, že v současné podobě nemůže být použito v experimentech s fotoemisí, pro které by byla kvadrupólová past příliš uzavřená, a tak by rozptýlené UV-fotony generovaly příliš velký fotoproud na elektrodách kvadrupólu. Proto se v současné době navrhuje nová podoba aparatury s odlišnou geometrií elektrod a to válcovou. Zjednodušené schéma aparatury, ještě však se "starým" tvarem kvadrupólu, ukazuje obrázek 1.(Převzato z [4].)



Obrázek 1: Schéma uspořádání experimentu. EG – elektronové dělo, IG – iontové dělo, FC – Faradayovy válce, QPS – vysokonapěťový zesilovač.

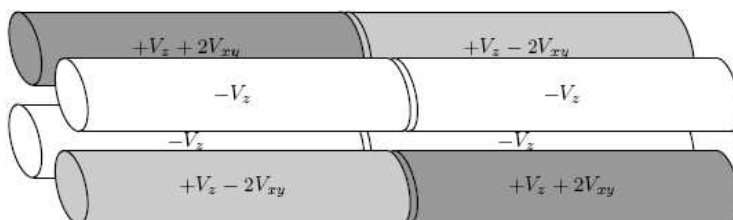
Vlastní měření probíhá tak, že se v elektrodynamickém kvadrupólu zachytí jediná částice a poté je pozorován její nábojový stav při ozařování elektronovým nebo iontovým svazkem o dané energii. Z parametrů pohybu částice a podmínek na elektrodách kvadrupólu se přitom kontinuálně určuje nábojový stav částice, přičemž podmínky na kvadrupólu jsou nastavovány řídicím systémem. Částice kmitá v kvadrupólu, přitom je osvětlena svazkem z He-Ne laseru a jednoduchým optickým systémem se zobrazuje na detektor polohy světelné stopy. Ze signálů tohoto detektoru získáváme jak frekvenci kmitů částice, tak signály pro tlumení a stabilizaci polohy částice uprostřed kvadrupólu.

## 2.2 Kvadrupól a jeho buzení

Referenční harmonický signál pro vysokonapěťové zesilovače napájející elektrody kvadrupólu se vyrábí sinusovým generátorem řízeným počítačem. Cílem této práce je ověřit způsob generování vhodného signálu pomocí digitálního signálního procesoru (DSP). Harmonický signál vyrobený počítačem je poté 100krát zesílen a přiveden na kvadrupól.

V původní aparatuře má přiváděné napětí efektivní hodnotu až 840V a jeho frekvenci lze měnit až do velikosti 10 kHz. V nové aparatuře bychom rádi dosáhli až 100 kHz, ale s o něco menší amplitudou.

Dosud bylo zmiňováno kvadrupólové napětí sinusového průběhu. Tato informace není úplná, neboť kvadrupólové napětí se během periody po nastavitelný čas vypíná. Toto vypnutí nastává vždy v okolí průchodu sinu nulou. Je to potřeba z toho důvodu, že součástí aparatury jsou i obvody vzorkování, které za účelem omezení defokuse svazku polem kvadrupólu zapínají svazek právě v čase, kdy je kvadrupólové napětí vypnuto.



Obrázek 2: Návrh nového lineárního kvadrupólu.

Návrh nového lineárního kvadrupólu ukazuje obrázek 2. (Přejato z [4].) Elektrické pole ve středu má být superpozicí 3 dvoudimensionálních kvadrupólových polí s navzájem kolmými osami symetrie. Cílem je udržet kmity částice v rovině centrální štěrbině (kolmé na osu elektrod). Zmíněný obrázek ukazuje rozdělení napětí na osmi elektrodách. Máme zde 3 různé hodnoty poskládané z potenciálů  $V_z$  a  $V_{xy}$ . Potřebujeme tedy 3 kanály referenčního napětí (pomineme-li zatím gravitační kompenzaci). Napětí na všech těchto třech kanálech musí být synchronní – musí být tedy generováno pomocí jediného procesoru.

### 3 Cíl práce

Jak již bylo uvedeno výše, cílem práce je ověřit způsob generování referenčních signálů pro vysokonapěťové zesilovače napájecí elektrody kvarupólu pomocí digitálního signálního procesoru (DSP). K dispozici je demonstrační deska s DSP Freescale 56F8014 a příslušné softwarové vývojové prostředí (programování v jazyce C/C++/ASM). DSP je vybaven číslicovými vstupy a výstupy, čítači, časovači, 12-bitovými AD převodníky a kanály PWM (pulsně šířková modulace). Vše lze navíc rozšířit o další periferie pomocí sériových rozhraní SPI nebo I<sup>2</sup>C.

Je potřeba porovnat 2 možnosti konstrukce:

1. pomocí externího integrovaného DA převodníku nebo
2. pomocí kanálů PWM ve funkci DA převodníků (zde je třeba doplnit demonstrační desku potřebnými analogovými prvky).

Cílovým rozsahem frekvencí generátoru pro novou aparaturu je interval 100 Hz až 100 kHz.

Amplituda a frekvence se zadá do DSP po sériovém rozhraní RS232 z osobního počítače. DSP musí zajistit kontinuitu fáze a pлавnou změnu amplitudy a frekvence generovaných signálů při změně jejich charakteristik. Dále je třeba zajistit hradlování výstupních signálů v okolí průchodu nulou s programovatelnou šířkou.

Také je úkolem navrhnout možnost doplnění složek gravitační kompenzace a tlumení kmitů prachové částice a dále zhodnotit vhodnost použití konkrétního DSP 56F8014 pro cílovou aplikaci.



## 4 Seznámení s použitými DSP a DAC, jejich vývojovými moduly a softwarem pro programování DSP

### 4.1 DSP 56F8014

Datový list pro DSP 56F8014 [5] popisuje základní architekturu takto: DSP 56F8014 patří do skupiny digitálních signálových procesorů založených na jádru 56800E, které je složeno ze tří paralelně pracujících výkonných jednotek, dovolujících 6 operací za instrukční cyklus. Je to 16 bitový procesor pracující na frekvenci 32 MHz s výpočetním výkonem 32 MIPS (tzn. 32 milionů instrukcí za sekundu). Součástí jádra je také zvláštní jednotka schopná provádět instrukce MAC ("Multiply And Accumulate") 16x16 bit a čtyři 36-bitové akumulátory obsahující rezervní bity pro přetečení.

Pro DSP 56F8014 existují dva možné zdroje hodinového signálu:

1. vnitřní relaxační oscilátor (generující signál o frekvenci 8MHz)
2. vnější hodinový zdroj

56F8014 disponuje více než 26-ti linkami GPIO ("General Purpose Input/Output") – v závislosti na konfiguraci periférií. Dále obsahuje 1 pětikanálový modul PWM ("Pulse Width Modulator") pracující na frekvenci až 96 MHz. Tento modul zahrnuje 3 komplementární individuálně programovatelné páry výstupů PWM signálu a poskytuje referenční výstupy pro synchronizaci AD převodníku přes "Quad Timer", což je časovací obvod, který může vyrábět 4 nezávislé časovací funkce.

Tento procesor poskytuje celou řadu standardních programovatelných periférií, jako např. SCI ("Serial Communications Interface")<sup>1</sup>, SPI ("Serial Peripheral Interface")<sup>2</sup>, "Quad Timer" a rozhraní pro sběrnici I<sup>2</sup>C ("Inter-Integrated Circuit")<sup>3</sup>. Víceúčelové vývody DSP mohou pracovat buď jako GPIO, nebo na ně mohou být vyvedeny signály SPI, SCI apod.

DSP 56F8014 obsahuje paměť typu Flash (EEPROM) o velikosti 16KB pro programy a paměť typu RAM pro data o velikosti 4KB.

Architekturu procesoru 56F8014 ukazuje obrázek 3.(Převzato z [5].)

### 4.2 DAC8814

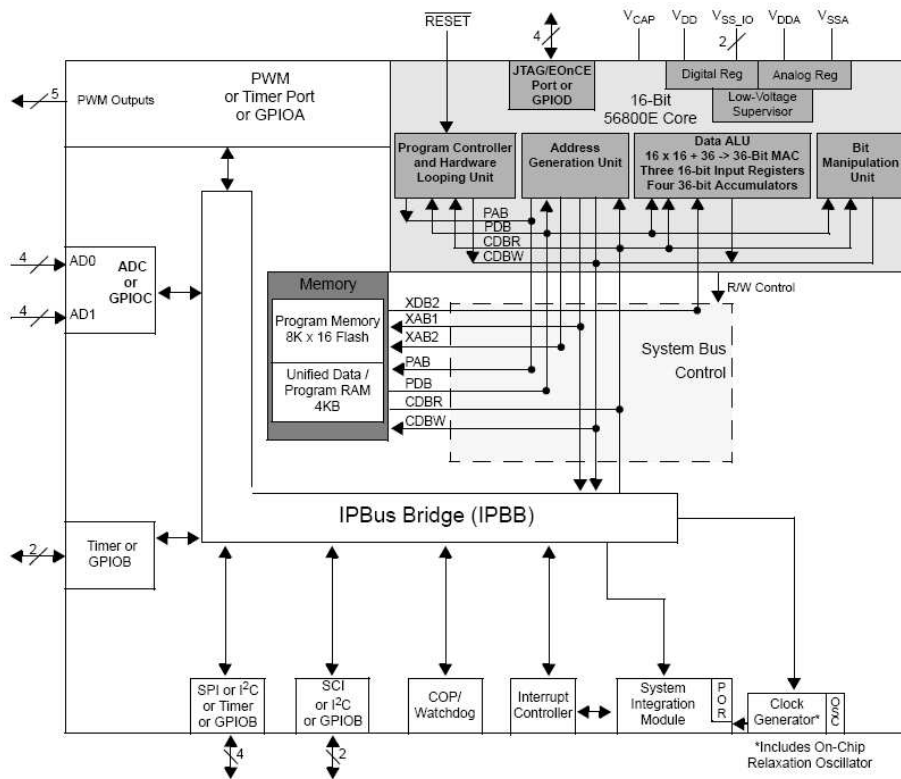
DAC8814 je čtyřnásobný 16 bitový digitálně-analogový převodník ("Digital-to-Analog Converter") s proudovým výstupem, podle [6] navržený pro práci s napájením 2,7 až 5,0 V.

---

<sup>1</sup>SCI je komunikační rozhraní umožňující sériovou asynchronní výměnu dat nejen mezi mikroprocesorem a perifériemi, ale i mezi dvěma mikroprocesory nebo mezi mikroprocesorem a externí sítí.

<sup>2</sup>Standardní sériové rozhraní mezi mikroprocesorem (master) a periferními integrovanými obvody (slave). Vedle dvou jednosměrných datových vodičů zahrnuje synchronizační hodinový signál a signál pro výběr podřízeného periferního obvodu. Přenosová rychlost: až desítky Mb/s.

<sup>3</sup>Sběrnice navržená firmou PHILIPS I<sup>2</sup>C obsahuje 2 obousměrné vodiče. Jeden přenáší hodinový signál SCL ("Synchronous Clock") a druhý data SDA ("Synchronous Data")



**56F8014 Block Diagram**

Obrázek 3: Blokové schéma DSP 56F8014

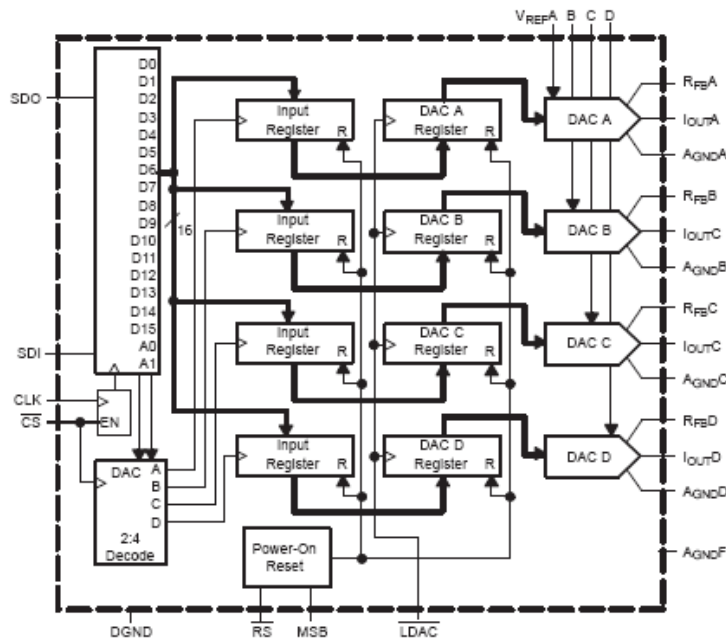
Použité externí referenční vstupní napětí  $V_{REF} = \pm 10\text{ V}$  určuje výstupní proud v plném rozsahu  $\pm 2\text{ mA}$ . Doba ustálení pro tento obvod je  $0,5\ \mu\text{s}$ , chyba linearity je menší než 1 LSB.

Dvojitě vyrovnávané sériové datové rozhraní je založeno na vysokorychlostním třívodičovém rozhraní kompatibilním s SPI. Používá signály SDI ("Serial Data In"), hodinový signál CLK ("Clock") a  $\overline{\text{CS}}$  ("Chip-Select"), který určuje, zda bude DAC8814 přijímat příchozí sériová data. (DAC přijímá data, pokud je  $\overline{\text{CS}}$  na úrovni "low".)

Společný vzorkovací vstup  $\overline{\text{LDAC}}$  umožňuje simultánní aktualizace všech výstupů DAC ze vstupních registrů, kam byla informace uložena již dříve. K aktualizaci dochází, je-li signál  $\overline{\text{LDAC}}$  na úrovni "low".

Při zapnutí systému nastavuje vnitřní nulování všechna výstupní napětí do základního stavu. Signál MSB umožňuje nulovací funkci  $\overline{\text{RS}}$  nastavovat všechny registry na nulu, pokud  $\text{MSB}=0$  (vhodné pro unipolární režim), nebo do poloviny rozsahu při  $\text{MSB}=1$  (vhodné pro bipolární režim).

Blokové schéma DAC8814 je na obrázku 4. (Převzato z [6].)



Obrázek 4: Blokové schéma DAC8814

### 4.3 Demonstrační deska s DSP 56F8014

Pro řešení této práce byla k dispozici demonstrační deska DSP 56F8014, která podle [7] obsahuje kromě samotného procesoru následující prvky:

- konektor pro připojení adaptéru paralelního rozhraní JTAG <sup>4</sup>
- rozhraní RS232<sup>5</sup>
- tzv. "Daughter Card Conector", což je konektor umožňující připojení vnějších periférií kompatibilních s SCI, SPI, PWM nebo GPIO rozhraním k procesoru
- diodu LED jako indikátor činnosti zařízení
- 6 dalších diod LED pro uživatelské aplikace
- tlačítko pro manuální nulování
- tlačítka pro přerušení #1 a #2

Připojení desky k PC vypadá takto: Z paralelního portu PC je veden (paralelní) kabel do adaptéru JTAG, z něhož jde pásový kabel do příslušného konektoru na demonstrační desce. Touto cestou se nahrává programový kód do paměti procesoru.

Do napájecího konektoru je připojeno stejnosměrné napětí 9 V.

Celé zapojení je zdokumentováno fotografií na obrázku 5. Druhý kabel na obrázku vyvedený z desky je sériový kabel. Je připojen k "9-pin" konektoru, odkud je vyvedena sběrnice RS232 (viz obrázek 3), odvozená od rozhraní SCI procesoru. Tímto kabelem se do procesoru dopravují data z terminálu PC pro změny parametrů generovaného signálu za běhu programu.

Podrobněji je demonstrační deska vidět na obrázku 6

Blokové schéma desky ukazuje obrázek 7. (Přejato z [7].) Na desce bylo třeba odstranit propojku JP4 (1-2), spojující víceúčelový vývod GPIO\_B3/MOSI/T3 s tlačítkem pro uživatelské přerušení #2 a kondenzátorem 100 nF připojeným na zem (viz obrázek 2-2 v Demonstration Board User Guide [7]). Tuto úpravu bylo nutno udělat, aby nedocházelo ke znehodnocování signálu SDI při nabíjení kondenzátoru.

## 4.4 Vývojový modul DAC8814 EVM

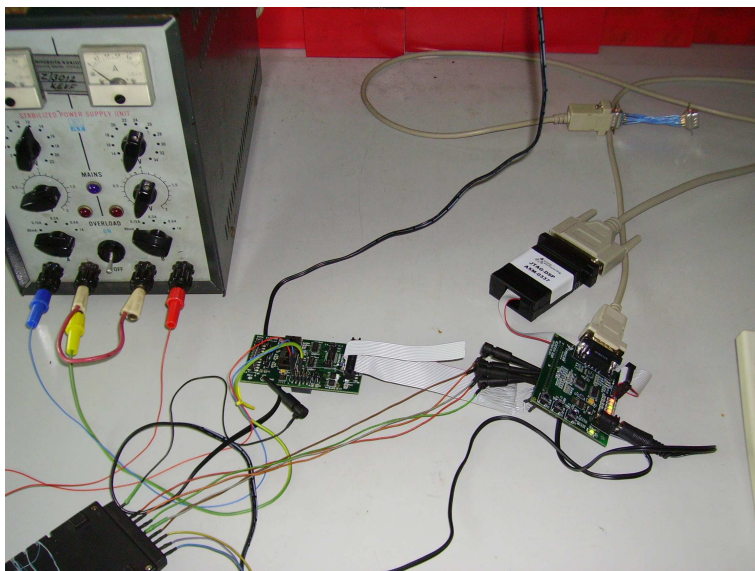
Pro ověření činnosti DA převodníku byl k dispozici vývojový modul EVM ("Evaluation Module"), který, jak uvádí DAC8803/14 User's Guide [8], obsahuje tyto prvky:

- integrovaný obvod DAC8814 se čtyřmi DA převodníky
- integrovaný zdroj referenčního napětí  $\pm 10$  V

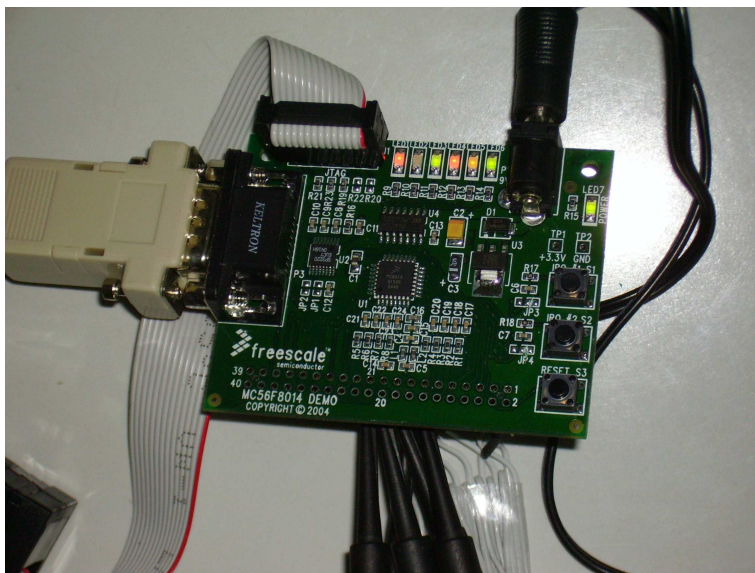
---

<sup>4</sup>"Joint Test Action Group" – standardní sériové rozhraní umožňující programování mikrokontrolérů a jiných obvodů přímo na desce plošných spojů s konkrétní aplikací, zároveň dovoluje trasování a ladění programu, případně diagnostiku jednotlivých částí čipu.

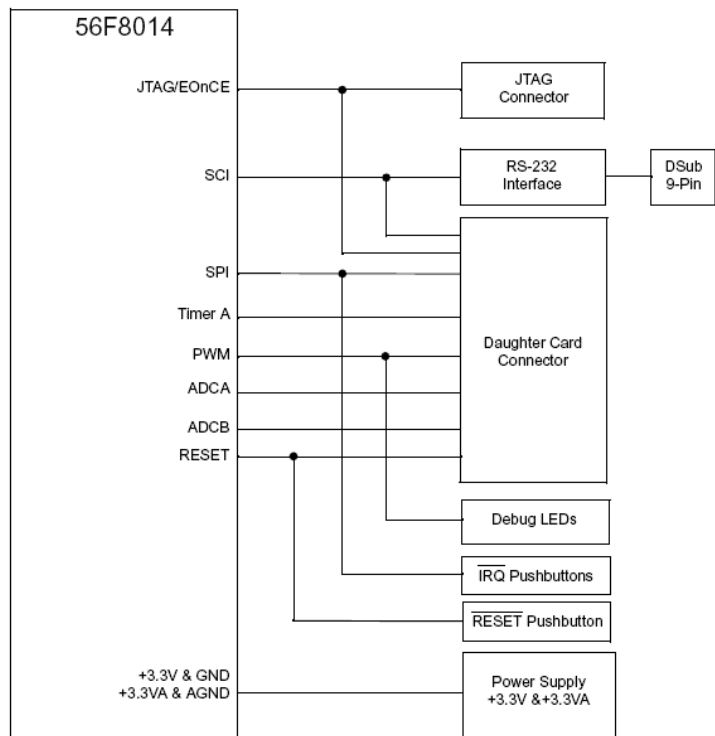
<sup>5</sup>Používá se jako komunikační rozhraní (COM) osobních počítačů, ale i další elektroniky. Umožňuje vzájemnou asynchronní sériovou komunikaci dvou zařízení po jediném vodiči. Bity se přenáší od LSB (nejméně významný) po MSB (nejvýznamnější bit). Počet datových bitů je volitelný (nejčastěji 8 bitů), logický stav je reprezentován bipolárně (nejčastěji  $\pm 12$  V).



Obrázek 5: Fotografie celkového zapojení s demonstrační deskou DSP 56F8014 a vývojovým modulem DAC8814 pro sinusový generátor řízený pomocí PC. (Demonstrační deska DSP je vpravo, deska vlevo je vývojový modul DAC.)



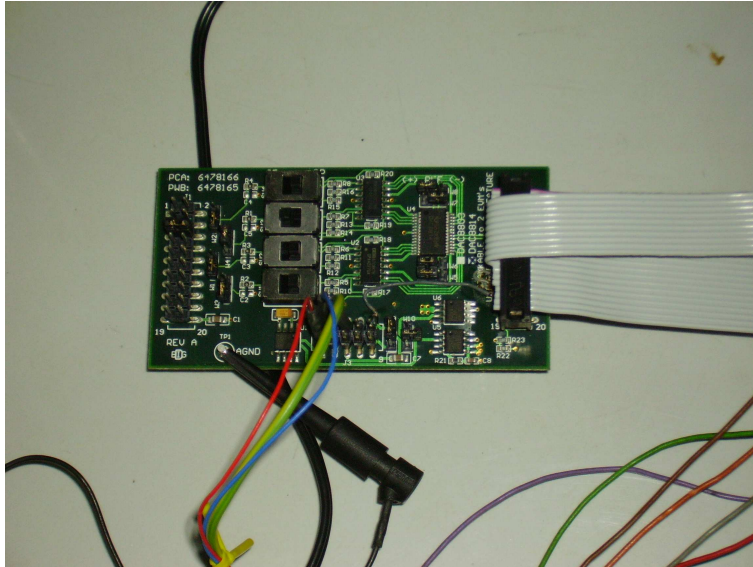
Obrázek 6: Fotografie demonstrační desky s DSP 56F8014.



Obrázek 7: Blokové schéma demonstrační desky s DSP 56F8014.

- vysokorychlostní sériové rozhraní

Schéma zapojení na vývojové desce ukazují obrázky 9 a 10. (Přejato z [8].) V tomto zapojení byly však třeba dodatečné úpravy, které jsou na obrázcích vyznačeny červenou barvou.



Obrázek 8: Fotografie vývojového modulu DAC8814.

Tento modul je k vidění jednak na již zmíněné fotografii na obrázku 5 a dále pak podrobněji na obrázku 8.

## 4.5 Processor Expert – software pro programování DSP

Pro vývoj programů pro DSP sloužilo programové prostředí firmy Metrowerks, CodeWarrior IDE, což je integrované vývojové prostředí pro procesory firmy Freescale Semiconductor vybavené překladači C/C++. Jeho součástí je doplněk Processor Expert (od firmy UNIS), který využívá formulářové komponenty zvané "Embedded Beans" (dále jen beans), samostatně generující kostru zdrojového kódu aplikace. Pomocí beanů lze ovládat jak periferie, tak samotné jádro procesoru.

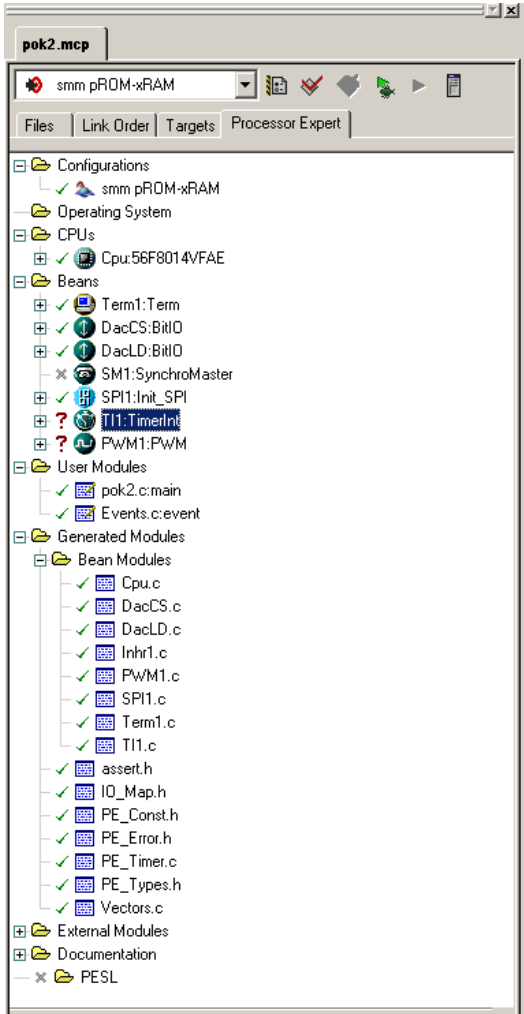
Například pro náš sinusový generátor byl využit bean pro PWM, bean časového přerušení ("TimerInt"), bean pro inicializaci SPI ("Init\_SPI"), bean pro terminál PC ("Term") a 2 beany pro bitový vstup a výstup ("BitIO"). Obrázek 11a zachycuje okno vývojového prostředí Processor Expert, kde je vypsána struktura celého projektu.

Konfigurace beanů se provádí pomocí okna "Bean Inspector". Toto okno je na obrázku 11b. Nastavují se zde vlastnosti ("Properties"), metody ("Methods") a události ("Events"). Konkrétní nastavení jednotlivých beanů pro náš program jsou v příloze.

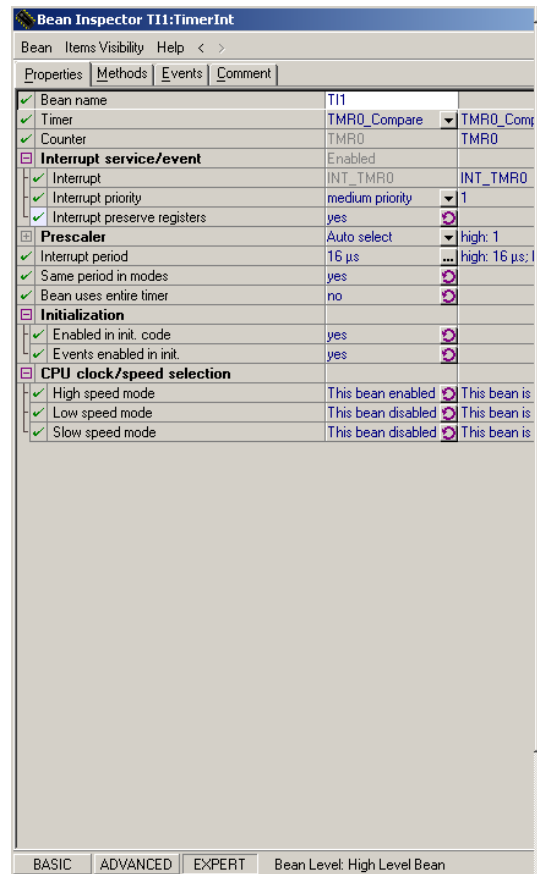








a



b

Obrázek 11: (a) Přehled projektu v prostředí Processor Expert. (b) Okno "Bean Inspector". Na obrázku je bean "TimerInt", generátor přerušení mikroprocesoru v pravidelných časových intervalech.

## 5 Popis vlastní konstrukce a programování

### 5.1 Program v C

Protože ve skutečnosti změny amplitudy a frekvence probíhají příliš rychle na to, aby se jejich plynost dala ověřit na osciloskopu, bylo třeba ověřit správnost algoritmů zápisem vygenerovaných dat do souboru a následným vykreslením do grafu. Plynulý přechod mezi amplitudami a frekvencemi ukazuje obrázek 12, který byl vytvořen programem Gnuplot. Tato část byla prováděna na osobním počítači s překladačem Watcom C.

Potřebujeme dosáhnout co největší cílové frekvence s dostatečným počtem vzorků v periodě. Proto potřebujeme co nejvíce zkrátit výpočet. Náš typ DSP nemá FPU (tj. jednotku pro výpočty s pohyblivou řádovou čárkou), a tak výpočty s těmito datovými typy (např. single) emuluje v celočíselné jednotce ALU, což je značně pomalé. Proto v programech pro DSP používáme jen celočíselné datové typy. Také se potřebujeme vyhnout náročnějším operacím jako je dělení a tím spíše nelze pomoci této jednotky počítat funkci sinus rozvojem do řady. (Tudíž nemůžeme sinus počítat ani pomocí nějaké matematické knihovny.)

Proto jsme zvolili tento způsob: uložit hodnoty sinu pro argument měnící se po určitém malém kroku (konkrétně 0,1 stupně) do pole. Když počítáme sinus pro jeden konkrétní argument, nalezneme v tomto poli předchůdce a následníka tohoto argumentu a pro výpočet funkční hodnoty provedeme lineární interpolaci. Abychom se vyhnuli pomalým operacím s pohyblivou řádovou čárkou, přeskálovali jsme hodnoty v tabulce tak, aby byly celočíselné. Abychom se vyhnuli dělení při interpolaci, přenásobili jsme fázi (argument sinu) číslem 80, čímž se změnil krok z 0,1 stupně na 8 LSB. Číslo 8 je mocnina dvou ( $8 = 2^3$ ), a tak lze namísto dělení a násobení provádět bitové posuvy, které jsou rychlejší. Vztah pro interpolaci je dán rovnicí 2.

$$x = (((f_i - (vstup1 \ll 3)) * (vystup2 - vystup1)) \gg 3) + vystup1, \quad (2)$$

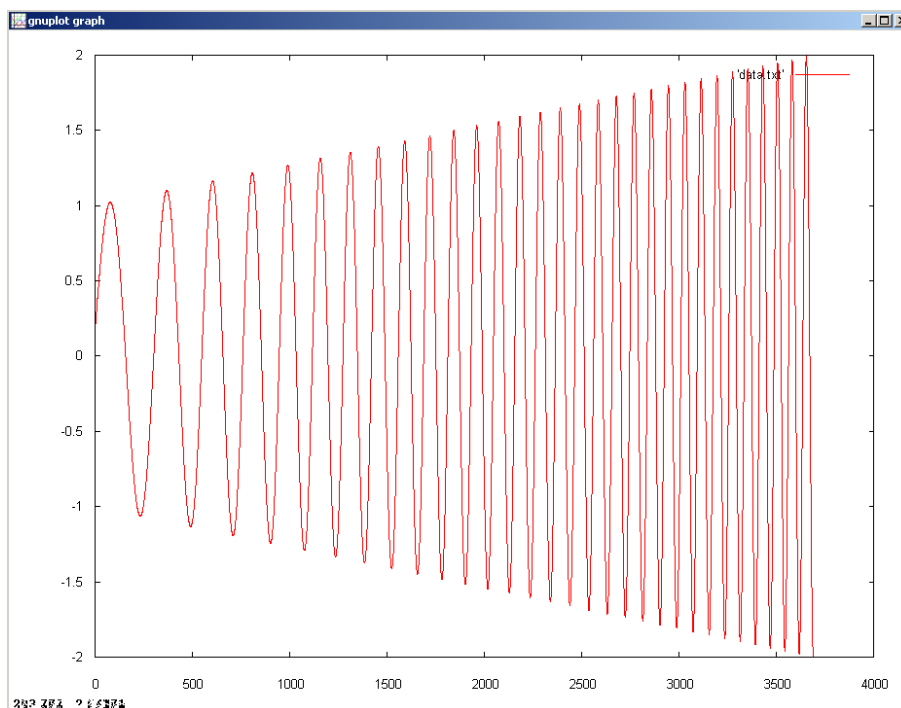
kde  $x$  je návratová hodnota funkce,  $f_i$  je fáze,  $vstup1$  hodnota argumentu nejbližší fázi – směrem dolů v tabulce,  $vystup1$  hodnota sinu pro nejbližší argument k dané fázi – směrem dolů v tabulce,  $vystup2$  hodnota sinu pro nejbližší argument k dané fázi – směrem nahoru v tabulce a  $\ll$ ,  $\gg$  jsou operátory bitového posuvu.

Máme-li přejít od jedné amplitudy k druhé, program postupně po kladném či záporném kroku mění hodnotu proměnné, kterou se násobí celkový výsledek funkce. Změny frekvence se dosahuje pomocí změny kroku pro fázi. Každý výpočet funkce sinus pro nový argument se provádí v rámci rutiny obsluhy časového přerušování, ihned po odeslání minulé hodnoty na DA převodník.

### 5.2 Propojení modulů pro DSP a DAC

Propojení obou modulů bylo již jako fotografie ukázáno na obrázku 5. Jak jsou propojeny konektory, je schématicky zachyceno na obrázku 13.

Obrázek 13 ukazuje propojení mezi konektorem P2 na desce s DSP a konektorem J2T na desce s DAC. K pinu 15 konektoru J2T byla přiletována propojka pro přívod napájení



Obrázek 12: Postupné zdvojnásobení frekvence a amplitudy.

$V_{DD}$ , díky čemuž deska s DAC nepotřebuje vlastní napájení a bere ho přes desku s DSP. Sériová data SDI z DSP jsou přivedena na pin 13 konektoru J2T, ke kterému rovněž vede dodatečný letovaný spoj (viz obrázek 9).

Realizaci propojení mezi DAC A a DAC B, jednotlivými DA převodníky obvodu DAC8814, vystihuje obrázek 14. DAC A zde pracuje v unipolárním režimu s pevným referenčním napětím na vstupu, DAC B pracuje v bipolárním násobícím režimu. Tomuto schématu odpovídají následující propojky na EVM:

- 5-6 na konektoru J1
- 2-3 na jumperu W7
- 1-2 na jumperu W6

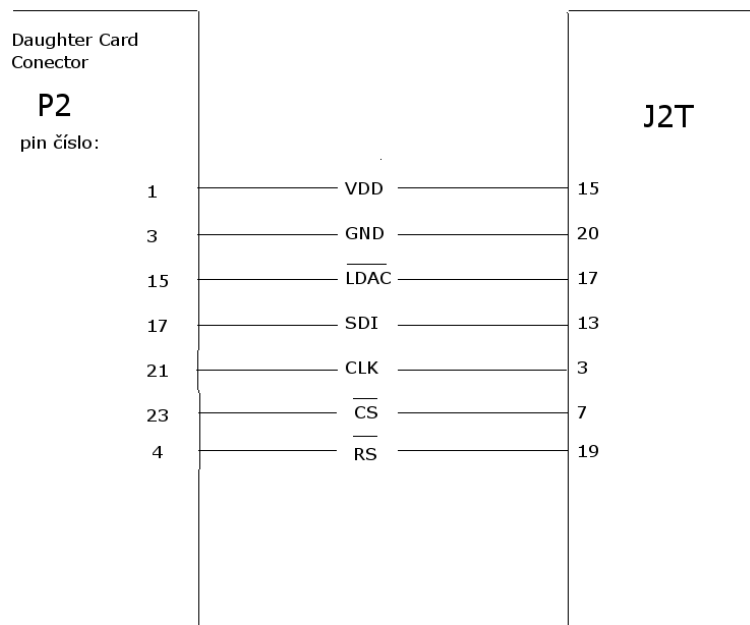
Převodník DAC A přijímá údaj o amplitudě, převodník DAC B údaj o vypočtené hodnotě sinu. Tímto se ušetří jedno násobení mezi oběma těmito veličinami.

Celkové schéma zapojení generátoru a jeho připojení ke kvadrupólu je na obrázku 15.

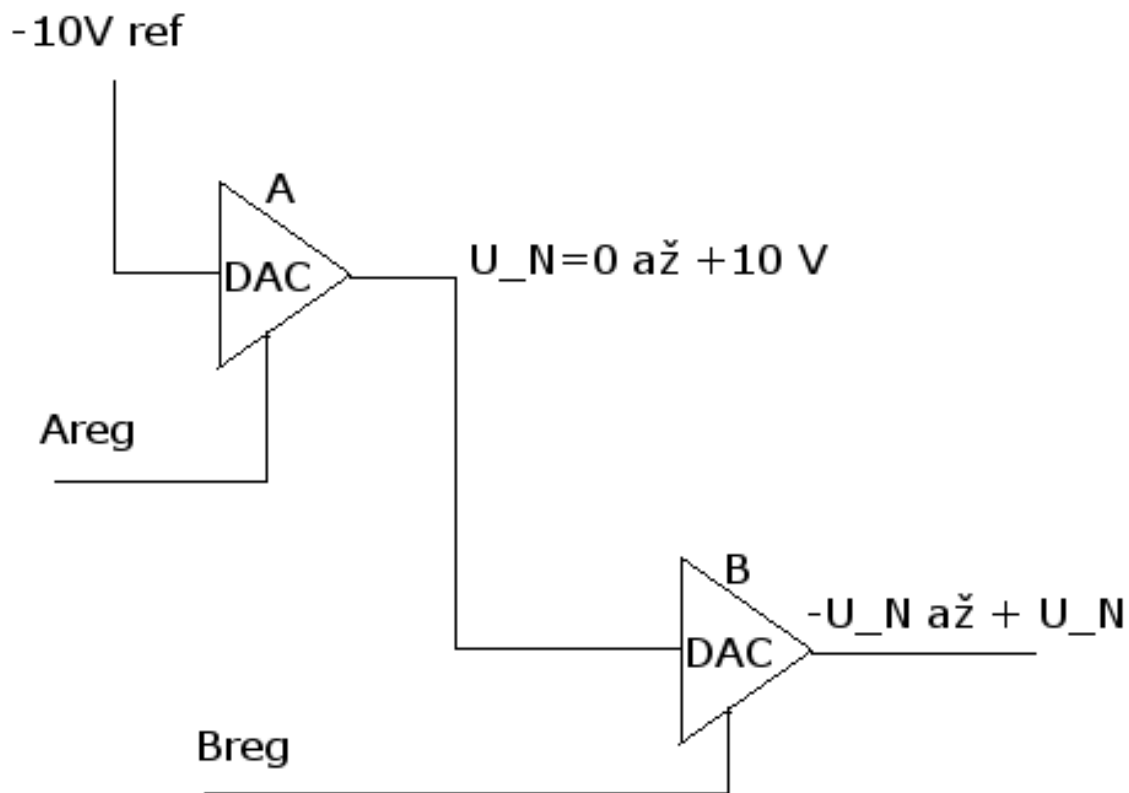
Pro účely ladění postačí vyrábět hodinový signál pro procesor pomocí vnitřního relaxačního oscilátoru, pro účely měření je však takovýto hodinový signál značně nepřesný. Přesnost, s jakou jsme schopni určit  $Q/m$ , závisí na přesnosti, s jakou známe frekvenci generovaného referenčního napětí. (Je to vidět ze vztahu 1.) Ta se odvíjí od přesnosti frekvence hodinového signálu. Proto ve výsledné konstrukci bude třeba využít externí hodinový zdroj.

Demonstrační deska  
DSP 56F8014

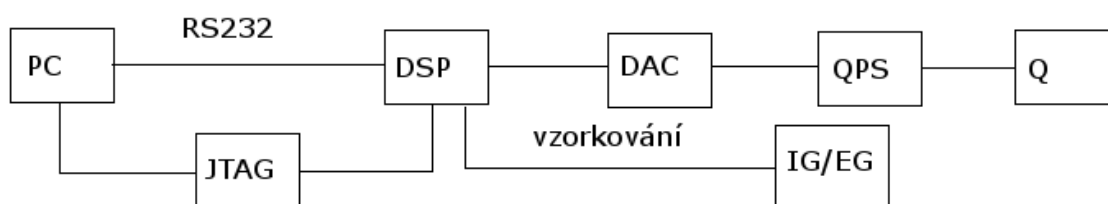
Deska s DAC8814



Obrázek 13: Schéma propojení modulů pro DSP a DAC.



Obrázek 14: Realizace propojení DAC A a DAC B.



Obrázek 15: Celkové schéma zapojení generátoru a jeho připojení ke kvadrupólu. PC – počítač, JTAG – volitelný adaptér paralelního rozhraní pro ladění, DSP – digitální signální procesor, DAC – obvod digitálně-analogových převodníků, QPS – vysokonapěťový zesilovač, Q – kvadrupól, IG/EG – iontové nebo elektronové dělo.

## 5.3 Program pro DSP

Výpis programového kódu je k shlédnutí v příloze.

Algoritmus pro výpočet sinu a plyné změny amplitudy a frekvence byl již nastíněn v sekci 5.1.

Po celou dobu běhu programu probíhají v procesoru časová přerušení s předem nastavenou periodou. Byla zvolena perioda  $16 \mu\text{s}$ , protože při kratších periodách se mezi jednotlivými přerušeními nestačil vykonat celý programový kód. (Delší perioda by vedla ke zbytečnému zpomalení programu, čímž bychom se připravili o část frekvenčního rozsahu.)

Toto časové přerušení vyvolá událost `TI1_0nInterrupt`, která je ošetřena příslušným programovým kódem. Tento programový kód obsahuje výpočet sinu a případné přenastavení amplitudy a frekvence. Přenastavení amplitudy (resp. frekvence) proběhne, je-li proměnná `newamp` (resp. `newf` nastavena na hodnotu "1"). Hodnota těchto proměnných je měněna v základní nekonečné smyčce v hlavním modulu, která čeká na pokyny z terminálu.

Pokyny z terminálu, na které smyčka reaguje, představují znaky "a", "f", "s", "p". Po detekci jednoho z těchto znaků se načte hodnota příslušné veličiny. Pomocí volby "a" se mění amplituda, pomocí "f" frekvence. (Znaky "s" a "p" budou vysvětleny dále.)

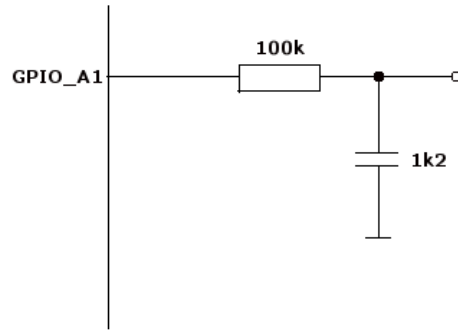
V zapojení s externími DA převodníky se vypočtená hodnota sinu zapisuje po částech (po 9 bitech) do registru označeného v kódu jako `Breg`, ze kterého je signál veden až na registr "DAC B" integrovaného obvodu DAC8814 (viz obrázek 4). Obdobným způsobem se zapisuje i změna amplitudy do registru "DAC A" (viz též obrázek).

Pokud namísto externího převodníku využijeme kanály PWM (ve funkci DA převodníků), využije se volba "p" pro přenastavení délky základní periody signálu na PWM. (Tuto periodu je třeba zadat v mikrosekundách.) Při ošetření časového přerušení se pomocí vypočtené hodnoty sinu nastaví příkazem `PWM1_SetRatio16(i)` nový poměr dob, kdy bude signál PWM mít úroveň "0" a kdy "1". Signál PWM je vyveden na výstup `GPIO_A1` a odtud je veden přes RC-filtr na osciloskop. RC-filtr vyrábí z šířkově modulovaného obdélníkového signálu na `GPIO_A1` střední hodnotu. Připojení RC-filtru ukazuje obrázek 16.

Ještě je třeba zmínit hradlování signálu v okolí 0 V. Je provedeno jednoduše přímo v programovém kódu pomocí příkazu `if`. Délka nulování se nastavuje volbou "s" z terminálu. Nacházíme-li se v zadaném intervalu, pak se díky příkazu `if` namísto uložení hodnoty sinu do výstupní proměnné uloží nula.

## 5.4 Gravitační kompenzace a tlumení kmitů prachové částice

Podle Žilavého [1] dochází zejména pro málo nabitě hmotnější částice (tj. částice s malým  $\frac{Q}{m}$ ) vlivem tíhové síly k posunu středu jejich oscilací vůči geometrickému středu kvadrupólu, čehož následkem je nestabilita kmitů částice. Pro vyřešení tohoto problému tedy stačí udržet částici v geometrickém středu kvadrupólu, k čemuž je potřeba působení síly ve vertikálním směru proti gravitaci. Nejschůdnější je řešení se silou elektrickou. Žilavý za tímto účelem navrhuje přivést na horní a dolní elektrodu kvadrupólu stejnosměrné napětí  $V_{gc}$ .



Obrázek 16: Připojení RC-filtru k DSP.

Jiná možnost je využít přímo náš sinusový generátor a z terminálu PC přímo zadávat, o kolik je třeba celý signál posunout ve vertikálním směru (tedy jakou konstantu přičítat). Program pro DSP by se rozšířil pouze o příkaz, který ke každé spočtené výstupní hodnotě signálu přičítá zadanou hodnotu parametru.

V novém návrhu kvadrupólu (viz obrázek 2) by bylo potřeba vytvořit ještě dodatečný potenciálový rozdíl mezi horními a dolními elektrodami, což by pro referenční napětí namísto tří vyžadovalo 6 až 8 kanálů. Také by bylo ovšem nutné volit několikanásobně rychlejší DSP.



## 6 Výsledky a jejich diskuse

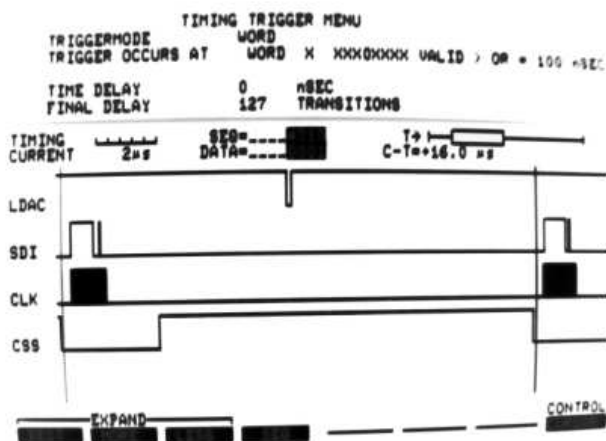
### 6.1 Řešení pomocí externího DAC

Následující obrázky vznikly úpravou fotografií z osciloskopu a logického analyzátoru programem GIMP.

#### Vysvětlivky k obrázkům s průběhem napětí na jednotlivých kanálech DSP

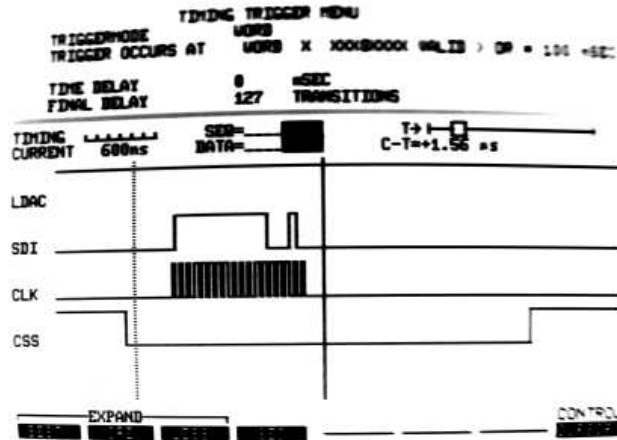
- LDAC – Když je tento signál ve stavu "low", přenáší se data postupně načtená na vstupní registry DAC synchronizovaně do registrů "DAC A,B,C,D" (viz obrázek 4).
- SDI – přenášená sériová data.
- CLK – hodinové pulsy.
- CS – "Chip-Select". Je-li ve stavu "low", přijímá DAC8814 data na vstupní posuvný registr.

Jak ukazuje obrázek 17, podařilo se periodu výstupních vzorků pro DAC úpravami programu zkrátit na  $16 \mu\text{s}$ .



Obrázek 17: Průběh signálů pro řízení DAC na rozhraní SPI – celý cyklus.

Mezi počátkem nulování  $\overline{\text{CS}}$  a pulsem na  $\overline{\text{LDAC}}$  uběhne přibližně  $8 \mu\text{s}$ . V této době je zahrnut jednak vlastní výpočet hodnoty generovaného signálu a jednak přenos dat na vstupní registry DAC8814. Zbytek padá jednak na režii vlastního přerušení (úklid registrů, uložení a zpětný výběr návratové adresy apod.) a jednak na ponechání pracovního času pro hlavní program (komunikace s terminálem).



Obrázek 18: Průběh signálů pro řízení DAC na rozhraní SPI – část cyklu.

Obrázek 18 ukazuje podrobněji průběh SDI a CLK. Můžeme z něj usoudit, že DAC8814 přijímá data přibližně po dobu  $2 \mu\text{s}$  (po tento čas je CS na nule).

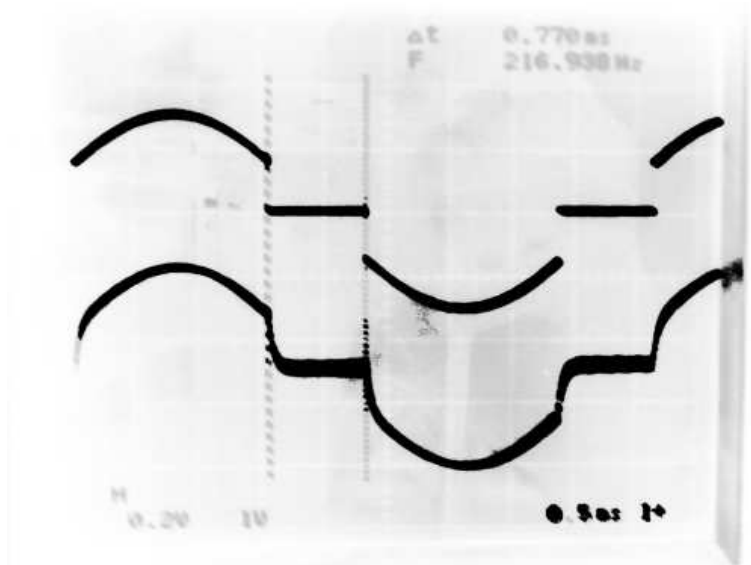
Jak ukazují obrázky 19, 20 a 21, při zvětšující se frekvenci generovaného signálu přicházíme o jeho hladkost. Projevuje se zde totiž délka výpočtu, a tak se signál zdrží na jedné hodnotě po dobu  $16 \mu\text{s}$ , což je právě celková doba trvání výpočtu. Takto se v signálu udělá "schod" (viz obrázek 21). Signál získaný s využitím externích DAC je ten nahoře. U skutečného generátoru bude třeba tyto "schody" vyhladit.

Při frekvenci větší než  $2 \text{ kHz}$  je takto získaný signál bez další speciální filtrace již nepoužitelný, na jednu periodu již zbývá méně než 32 vzorků.

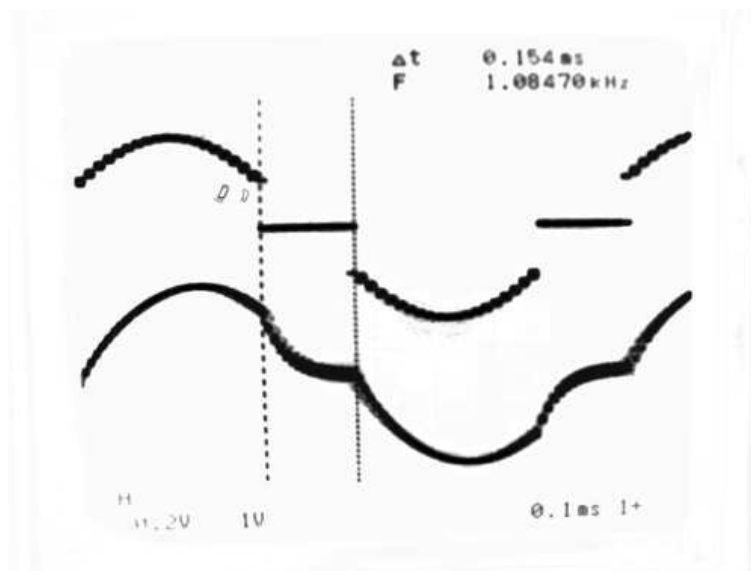
Program dovoluje změnu amplitudy s rychlostí přibližně  $10 \text{ V/s}$ , což lze dle potřeby dále zrychlit. Při změně frekvence se s každým vzorkem ( $16 \mu\text{s}$ ) upraví fázový krok o  $1/80$  stupně. Takto např. přechod ze  $100 \text{ Hz}$  na  $1 \text{ kHz}$  zabere přibližně  $6,6 \text{ ms}$ .

## 6.2 Řešení pomocí PWM

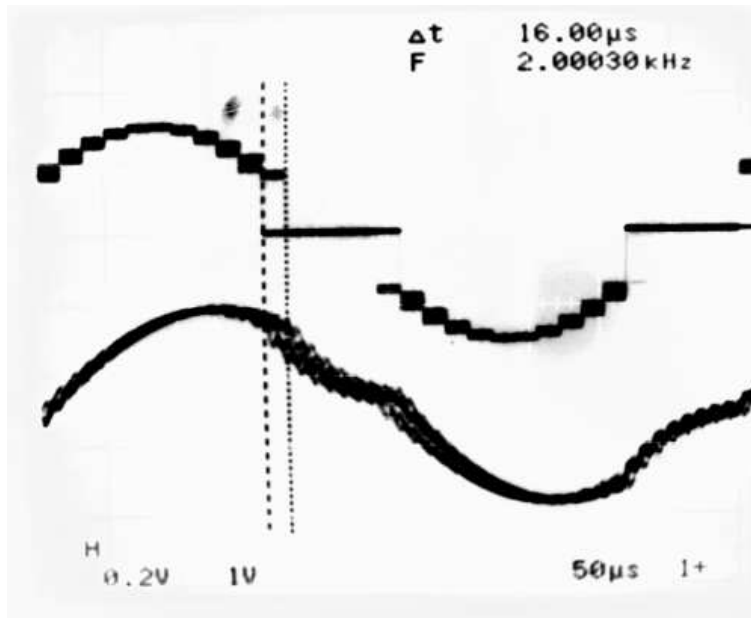
Výsledek využití PWM ukazují opět obrázky 19, 20 a 21, přičemž tentokrát se jedná o signál zobrazený dole. Tento signál byl získaný filtrováním obdélníkového signálu s proměnnou střídou z GPIO\_A1 přes RC-člen. Jak ukazují obrázky, se zvyšující se frekvencí se tu uplatňuje více pro změnu jiný nepříznivý efekt, a to čas potřebný pro nabití resp. vybití kondenzátoru výstupního filtru. To znamená, že v krátkých časových měřících namísto požadovaného skoku vidíme na osciloskopu vybíjecí křivku kondenzátoru. Časová konstanta výstupního RC-filtru je významná zejména pro vyšší generované frekvence a kratší hradlovací intervaly. Vidíme, že ani tímto způsobem se nepodaří vygenerovat signál požadovaného tvaru s větší frekvencí než  $1 \text{ kHz}$ .



Obrázek 19: Průběh výsledného vygenerovaného signálu při frekvenci 200 Hz. Signál získaný řešením pomocí externího DAC je nahoře, signál na výstupu z jednoduchého RC filtru napojeného na kanál PWM dole. Přerušovanými čarami je vyznačen interval hradlování na výstupu, určený pro ozařování částice elektronovými nebo iontovými svazky.



Obrázek 20: Průběh výsledného vygenerovaného signálu při frekvenci 1 kHz.



Obrázek 21: Průběh výsledného vygenerovaného signálu při frekvenci 2 kHz.

### 6.3 Možnosti vylepšení a zrychlení

Z toho, co zde již bylo uvedeno, vyplývá, že vlastní výpočet nové výstupní hodnoty pro jeden kanál trvá cca  $6 \mu\text{s}$ . Tato doba by se dala zkrátit, kdybychom namísto čtvrtperrody sinu uložili do paměti procesoru pole s hodnotami celé periody. Tím by jedna část kódu odpadla. Avšak vzrostly by nároky na paměť. V současné verzi má pole s hodnotami čtvrtperrody sinu celkem 901 položek typu `unsigned short`, což zabírá přibližně 1,8 KB. Vzhledem k tomu, že celková velikost paměti RAM procesoru je 4 KB, tak by se do ní čtyřnásobně dlouhé pole nevešlo. (Nehledě na to, že potřebujeme uložit do paměti i ostatní proměnné.) Odhadem bychom tedy potřebovali procesor s pamětí RAM o velikosti 16 KB.

Pro splnění požadavků na paměť by byl vhodný například procesor DSP Freescale 56F8156. Na webových stránkách Freescale Semiconductor [10] se uvádí, že tento procesor je rovněž založen na jádru 56800E. Obsahuje 256 KB paměti typu Flash pro programy, 16 KB paměti typu RAM pro data a 16 KB bootovací paměti typu Flash. Pinů má pro naše účely dostatek (více než 62 vývodů GPIO). Pracuje na frekvenci 60 MHz při výkonu 60 MIPS, což by zkrátilo výpočet téměř na polovinu. Protože v cílovém zapojení je potřeba tří resp. šesti až osmi výstupních kanálů, bude podle skutečného frekvenčního rozsahu potřeba hledat ještě rychlejší mikroprocesor, nebo zvolit řešení s řádově větší kapacitou paměti pro uložení kompletního předem spočteného průběhu jedné periody výstupního signálu.

Dalšího zrychlení výpočtu by bylo možno dosáhnout paralelním vedením dat z DSP do DAC. K tomu by byl potřeba obvod DAC s paralelním rozhraním a DSP s dostatečným počtem vývodů (16 pro data + 2 ovládací). Jak lze zjistit na webových stránkách Texas In-

struments [11], vhodný DAC by byl například DAC8822, což jedvoukanálový 16-bitový DA převodník vybavený paralelním rozhraním. Jeho vzorkovací frekvence dosahuje až 50 MHz, doba ustálení a linearita je zde stejná jako u DAC8814.

Pro účely ladění stačilo použít pro hodinový signál procesoru vnitřní relaxační oscilátor ( $8 \text{ MHz} \pm 0,3\%$ ). Jak již bylo uvedeno výše, pro účely měření je však takovýto hodinový signál značně nepřesný. Vztah 1 ukazuje, že je třeba znát frekvenci  $f_{ac}$  co nejpřesněji. Tím pádem potřebujeme co nejpřesnější údaj o frekvenci hodinového signálu. Proto jako další vylepšení určitě nesmíme opomenout využití externího hodinového zdroje namísto vnitřního. Krystalové oscilátory, využívané za tímto účelem, mohou generovat stabilní hodinový signál s relativní přesností až  $10^{-6}$ .

## 7 Závěr

V předložené práci bylo ukázáno, že se během experimentů s vývojovými moduly podařilo generovat signál požadovaného tvaru s možností plynulé změny amplitudy i frekvence a zachováním kontinuity fáze na frekvencích do 2 kHz. Získaný signál byl při těchto frekvencích vyhovující při generování pomocí externího DAC. Při využití kanálů PWM vyhovoval signál o frekvencích do 1 kHz.

Pro novou aparaturu je však požadován signál s o řád větší frekvencí. Bylo však ukázáno, že procesor DSP 56F8014, který je nyní k dispozici, pro tyto frekvence nestačí, ať již použijeme externí DAC, anebo kanály PWM. Rozdíl mezi oběmi možnostmi dobře ukazuje obrázek 21.

Pro dosažení požadovaných frekvencí bylo navrženo užití rychlejšího procesoru např. DSP 56F8156 (také od firmy Freescale Semiconductor), jehož paměť RAM je čtyřnásobná oproti DSP 56F8014. Další zrychlení by přineslo užití DAC8822 s paralelním rozhraním namísto DAC8814 se sériovým rozhraním.

Dále byla diskutována nová varianta gravitační kompenzace a tlumení kmitů prachových částic vycházející přímo z možností sinusového generátoru (tudíž není třeba dodatečné stejnosměrné napětí na elektrodách kvadrupólu ze zvláštního zdroje).

## 8 Příloha: Nastavení jednotlivých beanů

Pro jednotlivé beany byly nastaveny následující vlastnosti ("Properties"):

### **Term1:Term**

Baud rate: 19200 baud  
Parity: none  
Width: 8 bit  
Stop bit: 1  
RxD: GPIOB6\_RxD\_SDA\_CLKIN  
TxD: GPIOB7\_TxD\_SCL

### **DacCS:BitIO**

Pin for I/O: GPIOB1\_SS\_B\_SDA  
Direction: Output  
Init. value: 1

### **DacLD:BitIO**

Pin for I/O: GPIOB5\_T1\_FAULT3  
Direction: Output  
Init. value: 1

### **SPI1:Init\_SPI**

Device: SPI  
MISO pin: GPIOB2\_MISO\_T2  
MOSI pin: GPIOB3\_MOSI\_T3  
SCLK pin: GPIOB0\_SCLK\_SCL  
SS pin: GPIOB1\_SS\_B\_SDA  
SPI frequency: 16 MHz  
Data size: 9 bit  
Data shift order: MSB first

### **TI1:TimerInt**

Timer: TMRO\_Compare  
Counter: TMRO  
Interrupt period: 16  $\mu$ s

**PWM1:PWM**

Output pin: GPIOA1\_PWM1

Period: 10  $\mu$ s

Starting pulse width: 5  $\mu$ s

**CPU:56F8014VFAE**

PWM clock rate: 3x system clock (96 MHz)

System clock: 32 MHz



## 9 Příloha: Kód programu pro DSP

```
/** #####
**      Filename   : pok2.C
**      Project    : pok2
**      Processor  : 56F8014VFAE
**      Version    : Driver 01.11
**      Compiler   : Metrowerks DSP C Compiler
**      Date/Time  : 12.6.2007, 12:04
**      Abstract   :
**          Main module.
**          Here is to be placed user's code.
**      Settings   :
**      Contents   :
**          No public methods
**
**      (c) Copyright UNIS, spol. s r.o. 1997-2006
**      UNIS, spol. s r.o.
**      Jundrovska 33
**      624 00 Brno
**      Czech Republic
**      http       : www.processorexpert.com
**      mail       : info@processorexpert.com
** #####*/
/* MODULE pok2 */
/* Including used modules for compiling procedure */
#include "Cpu.h"
#include "Events.h"
#include "Term1.h"
#include "Inhr1.h"
#include "DacCS.h"
#include "DacLD.h"
#include "SPI1.h"
#include "TI1.h"
#include "PWM1.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
```

```

#include <stdlib.h>
extern unsigned short Ampl,Ampl0,Ampl1;
extern unsigned short freqx,freq1,freq0,s,p;
extern unsigned char newamp,newf;
extern unsigned short deltaphase;
void main(void)
{
    /* Write your local variable definition here */
    unsigned char volba;
    unsigned long data;
    char line[100];
    char *endptr;
    //unsigned short i=0;
    /*unsigned short Ampl0=32768;
    unsigned short Ampl;
    unsigned char newamp=0; */
    /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
    PE_low_level_init();
    /*** End of Processor Expert internal initialization.          ***/
    //newamp=0;
    //newf=0;
    SPI1_Init();
    setRegBits(GPIO_B_PER,1+8);
    DacLD_ClrVal(); //LDAC to low
    DacCS_SetVal();
    Ampl=Ampl0;
    freqx=freq0;
    Ampl1=Ampl0;
    freq1=freq0;
    newamp=1;
    newf=1;
    /* Write your code here */
    Term1_SendStr("Terminal ready.\n\r");
    for(;;) {
        //Term1_SendStr("Poslat data?"); /* Display a message */
        //while(!Term1_KeyPressed()) {;} /* Wait for any char */
        //Term1_CRLF();
        //Term1_ReadChar(&dum); /* Move cursor to new line */
        //Term1_SendNum(i);
        /*Send2DAC(Breg,i);
        // DacLD_ClrVal(); //LDAC to low
        // DacLD_SetVal(); //LDAC to high
        //phase += 500; //phase je nasobena 100, 500=5 deg.

```

```

phase += 4100; //phase je nasobena 80, 400=5 deg.
if (phase>=28800)
{
    phase -= 28800;
}
else
{
    phase -= 0;
}
DacCS_SetVal();
i=sinus(phase);*/
Term1_ReadChar(&volba);
if (volba=='a')
{
    Term1_SendChar(volba);
    Term1_SendNum(deltaphase);
        Term1_SendChar('\n');
    Term1_SendChar('\r');
    Term1_ReadLine(line);
    Term1_SendStr(line);
    Term1_SendChar('\n');
    Term1_SendChar('\r');
    data=strtoul(line, &endptr, 10);
    Ampl1=(unsigned short) data;
    newamp=1;
}
if (volba=='f'){
    Term1_SendChar(volba);
    Term1_SendNum(deltaphase);
        Term1_SendChar('\n');
    Term1_SendChar('\r');
    Term1_ReadLine(line);
    Term1_SendStr(line);
    Term1_SendChar('\n');
    Term1_SendChar('\r');
    data=strtoul(line, &endptr, 10);
    freq1=(unsigned short)data;
    newf=1;
}
if (volba=='s')
{
    Term1_SendChar(volba);
    Term1_SendChar('\n');
}

```

```

        Term1_SendChar('\r');
        Term1_ReadLine(line);
        Term1_SendStr(line);
        Term1_SendChar('\n');
        Term1_SendChar('\r');
        data=strtoul(line, &endptr, 10);
        s=(unsigned short)data;
    }
if (volba=='p')
    {
        Term1_SendChar(volba);
        Term1_SendChar('\n');
        Term1_SendChar('\r');
        Term1_ReadLine(line);
        Term1_SendStr(line);
        Term1_SendChar('\n');
        Term1_SendChar('\r');
        data=strtoul(line, &endptr, 10);
        p=(unsigned short)data;
        PWM1_SetDutyUS(p);
    }
//Term1_CRLF(); /* Display a message */
}
}
/* END pok2 */
/*
** #####
**
** This file was created by UNIS Processor Expert 2.98 [03.79]
** for the Freescale 56800 series of microcontrollers.
**
** #####
**
*/

/** #####
** Filename : Events.C
** Project : pok2
** Processor : 56F8014VFAE
** Beantype : Events
** Version : Driver 01.03
** Compiler : Metrowerks DSP C Compiler

```

```

**      Date/Time : 19.6.2007, 10:37
**      Abstract  :
**          This is user's event module.
**          Put your event handler code here.
**      Settings  :
**      Contents  :
**          TI1_OnInterrupt - void TI1_OnInterrupt(void);
**
**      (c) Copyright UNIS, spol. s r.o. 1997-2006
**      UNIS, spol. s r.o.
**      Jundrovska 33
**      624 00 Brno
**      Czech Republic
**      http       : www.processorexpert.com
**      mail       : info@processorexpert.com
** #####*/
/* MODULE Events */
#include "Cpu.h"
#include "Events.h"
/*
** =====
**      Event      : TI1_OnInterrupt (module Events)
**
**      From bean  : TI1 [TimerInt]
**      Description :
**          When a timer interrupt occurs this event is called (only
**          when the bean is enabled - "Enable" and the events are
**          enabled - "EnableEvent").
**      Parameters : None
**      Returns    : Nothing
** =====
*/
#pragma interrupt called /* Comment this line if the appropriate 'Interrupt
preserve registers' property */
/* is set to 'yes' (#pragma interrupt saveall is generated
before the ISR) */
unsigned short phase=0;
unsigned short deltaphase=4100;
unsigned short i=0;
    unsigned short Ampl0=32768;
    unsigned short Ampl,Ampl1;
    unsigned short freqx,freq1;
    unsigned short freq0=7;

```

```
unsigned short s=2400;
unsigned short p;
unsigned char newamp=0;
unsigned char newf=0;
unsigned short pole[901]={
0,57,114,171,228,285,343,400,457,514,
571,629,686,743,800,857,914,972,1029,1086,
1143,1200,1257,1314,1372,1429,1486,1543,1600,1657,
1714,1771,1829,1886,1943,2000,2057,2114,2171,2228,
2285,2342,2399,2456,2513,2570,2627,2684,2741,2798,
2855,2912,2969,3026,3083,3140,3197,3254,3311,3368,
3425,3481,3538,3595,3652,3709,3766,3822,3879,3936,
3993,4050,4106,4163,4220,4276,4333,4390,4446,4503,
4560,4616,4673,4730,4786,4843,4899,4956,5012,5069,
5125,5182,5238,5295,5351,5408,5464,5520,5577,5633,
5689,5746,5802,5858,5915,5971,6027,6083,6139,6196,
6252,6308,6364,6420,6476,6532,6588,6644,6700,6756,
6812,6868,6924,6980,7036,7092,7147,7203,7259,7315,
7370,7426,7482,7538,7593,7649,7704,7760,7816,7871,
7927,7982,8037,8093,8148,8204,8259,8314,8370,8425,
8480,8535,8591,8646,8701,8756,8811,8866,8921,8976,
9031,9086,9141,9196,9251,9306,9361,9415,9470,9525,
9580,9634,9689,9744,9798,9853,9907,9962,10016,10071,
10125,10179,10234,10288,10342,10397,10451,10505,10559,10613,
10667,10721,10775,10829,10883,10937,10991,11045,11099,11153,
11206,11260,11314,11368,11421,11475,11528,11582,11635,11689,
11742,11796,11849,11902,11955,12009,12062,12115,12168,12221,
12274,12327,12380,12433,12486,12539,12592,12645,12697,12750,
12803,12855,12908,12960,13013,13065,13118,13170,13222,13275,
13327,13379,13431,13484,13536,13588,13640,13692,13744,13796,
13847,13899,13951,14003,14054,14106,14158,14209,14261,14312,
14364,14415,14466,14518,14569,14620,14671,14722,14773,14824,
14875,14926,14977,15028,15079,15130,15180,15231,15282,15332,
15383,15433,15484,15534,15584,15635,15685,15735,15785,15835,
15885,15935,15985,16035,16085,16135,16185,16234,16284,16333,
16383,16433,16482,16531,16581,16630,16679,16728,16778,16827,
16876,16925,16974,17023,17071,17120,17169,17218,17266,17315,
17363,17412,17460,17509,17557,17605,17653,17702,17750,17798,
17846,17894,17942,17989,18037,18085,18133,18180,18228,18275,
18323,18370,18417,18465,18512,18559,18606,18653,18700,18747,
18794,18841,18887,18934,18981,19027,19074,19120,19167,19213,
19259,19306,19352,19398,19444,19490,19536,19582,19628,19673,
19719,19765,19810,19856,19901,19947,19992,20037,20083,20128,
```

20173, 20218, 20263, 20308, 20353, 20397, 20442, 20487, 20531, 20576,  
20620, 20665, 20709, 20753, 20798, 20842, 20886, 20930, 20974, 21018,  
21062, 21105, 21149, 21193, 21236, 21280, 21323, 21367, 21410, 21453,  
21497, 21540, 21583, 21626, 21669, 21712, 21754, 21797, 21840, 21882,  
21925, 21967, 22010, 22052, 22094, 22137, 22179, 22221, 22263, 22305,  
22347, 22388, 22430, 22472, 22513, 22555, 22596, 22638, 22679, 22720,  
22761, 22802, 22843, 22884, 22925, 22966, 23007, 23048, 23088, 23129,  
23169, 23210, 23250, 23290, 23330, 23371, 23411, 23451, 23490, 23530,  
23570, 23610, 23649, 23689, 23728, 23768, 23807, 23846, 23886, 23925,  
23964, 24003, 24042, 24080, 24119, 24158, 24196, 24235, 24273, 24312,  
24350, 24388, 24426, 24465, 24503, 24540, 24578, 24616, 24654, 24691,  
24729, 24766, 24804, 24841, 24878, 24916, 24953, 24990, 25027, 25064,  
25100, 25137, 25174, 25210, 25247, 25283, 25320, 25356, 25392, 25428,  
25464, 25500, 25536, 25572, 25608, 25643, 25679, 25714, 25750, 25785,  
25820, 25855, 25890, 25925, 25960, 25995, 26030, 26065, 26099, 26134,  
26168, 26203, 26237, 26271, 26305, 26339, 26373, 26407, 26441, 26475,  
26508, 26542, 26576, 26609, 26642, 26676, 26709, 26742, 26775, 26808,  
26841, 26873, 26906, 26939, 26971, 27004, 27036, 27068, 27100, 27132,  
27164, 27196, 27228, 27260, 27292, 27323, 27355, 27386, 27418, 27449,  
27480, 27511, 27542, 27573, 27604, 27635, 27665, 27696, 27727, 27757,  
27787, 27818, 27848, 27878, 27908, 27938, 27968, 27997, 28027, 28057,  
28086, 28116, 28145, 28174, 28203, 28232, 28261, 28290, 28319, 28348,  
28376, 28405, 28433, 28462, 28490, 28518, 28546, 28574, 28602, 28630,  
28658, 28686, 28713, 28741, 28768, 28796, 28823, 28850, 28877, 28904,  
28931, 28958, 28984, 29011, 29038, 29064, 29090, 29117, 29143, 29169,  
29195, 29221, 29247, 29272, 29298, 29324, 29349, 29375, 29400, 29425,  
29450, 29475, 29500, 29525, 29550, 29574, 29599, 29623, 29648, 29672,  
29696, 29721, 29745, 29768, 29792, 29816, 29840, 29863, 29887, 29910,  
29934, 29957, 29980, 30003, 30026, 30049, 30071, 30094, 30117, 30139,  
30162, 30184, 30206, 30228, 30250, 30272, 30294, 30316, 30337, 30359,  
30380, 30402, 30423, 30444, 30465, 30486, 30507, 30528, 30549, 30569,  
30590, 30610, 30631, 30651, 30671, 30691, 30711, 30731, 30751, 30771,  
30790, 30810, 30829, 30849, 30868, 30887, 30906, 30925, 30944, 30963,  
30981, 31000, 31018, 31037, 31055, 31073, 31091, 31109, 31127, 31145,  
31163, 31180, 31198, 31215, 31233, 31250, 31267, 31284, 31301, 31318,  
31335, 31351, 31368, 31384, 31401, 31417, 31433, 31449, 31465, 31481,  
31497, 31513, 31528, 31544, 31559, 31575, 31590, 31605, 31620, 31635,  
31650, 31665, 31679, 31694, 31708, 31723, 31737, 31751, 31765, 31779,  
31793, 31807, 31821, 31834, 31848, 31861, 31874, 31888, 31901, 31914,  
31927, 31939, 31952, 31965, 31977, 31990, 32002, 32014, 32026, 32038,  
32050, 32062, 32074, 32086, 32097, 32109, 32120, 32131, 32142, 32153,  
32164, 32175, 32186, 32197, 32207, 32218, 32228, 32238, 32249, 32259,  
32269, 32279, 32288, 32298, 32308, 32317, 32326, 32336, 32345, 32354,

```

32363,32372,32381,32389,32398,32407,32415,32423,32431,32440,
32448,32455,32463,32471,32479,32486,32494,32501,32508,32515,
32522,32529,32536,32543,32549,32556,32562,32569,32575,32581,
32587,32593,32599,32604,32610,32616,32621,32626,32632,32637,
32642,32647,32652,32656,32661,32665,32670,32674,32678,32683,
32687,32691,32694,32698,32702,32705,32709,32712,32715,32719,
32722,32725,32727,32730,32733,32735,32738,32740,32742,32744,
32747,32748,32750,32752,32754,32755,32757,32758,32759,32760,
32762,32762,32763,32764,32765,32765,32766,32766,32766,32766,
32767};
unsigned short Areg=0 <<7;
unsigned short Breg=1 <<7;
void TI1_OnInterrupt(void)
{
    //9bit version
    unsigned char dum;
    unsigned short vstup1;
    unsigned short vystup1;
    unsigned short vystup2;
    char sgn=1;
    unsigned short x;
    unsigned short fi;
    DacCS_ClrVal();// nulování chip-selectu
    /*zápis hodnoty sinu do registru B*/
    setReg16(SPI_DTR,Breg + (i>>9));
    setReg16(SPI_DTR,i & 511);
    /**/
    while (!(getReg16(SPI_SCR) & 1)) {;}//čekání, dokud na posledním bitu daného
registru není nula
    PWM1_SetRatio16(i);//nastavení nového poměru dob, kdy bude na PWM low a kdy
high
    DacCS_SetVal();//nastavení chip-selectu na úroveň high
    phase += deltaphase; //phase je nasobena 80, 400=5 deg.
    /*využití periodičnosti sinu ve výpočtu*/
    if (phase>=28800)
    {
        phase -= 28800;
    }
    else
    {
        phase -= 0;
    }
    /**/
}

```



```

if (newamp)
{
    DacCS_ClrVal();
    /*nastavení amplitudy v příslušném registru*/
    setReg16(SPI_DTR,Areg + (Ampl>>9));
    setReg16(SPI_DTR,Ampl & 511);
    /**/
    while (!(getReg16(SPI_SCR) & 1)) {;}
    dum=0; //"zdržovací příkaz"
    /*přizpůsobení amplitudy podle požadavků z terminálu*/
    if (Ampl1>Ampl0) Ampl++; else Ampl--;
    if (Ampl1>Ampl0 && Ampl1<=Ampl){
        newamp=0;
        Ampl0=Ampl1;
    }
    if (Ampl1<=Ampl0 && Ampl1>=Ampl){
        newamp=0;
        Ampl0=Ampl1;
    }
    /**/
    DacCS_SetVal();
}
if (newf) /*přizpůsobení frekvence požadavkům z terminálu*/
{
    if (freq1>freq0) freqx++; else freqx--;
    deltaphase=freqx; //576*freqx;
    if (freq1>freq0 && freq1<=freqx){
        newf=0;
        freq0=freq1;
    }
    if (freq1<=freq0 && freq1>=freqx){
        newf=0;
        freq0=freq1;
    }
}
fi= phase;
/*výpočet sinu pomocí interpolace*/
if (fi>14400) {
    sgn=0;
    fi-=14400;
}
if (fi>7200) fi=14400-fi;
vstup1=(unsigned short) (fi >> 3);

```

```

vystup1=pole[vstup1];
vystup2=pole[vstup1+1];
if (fi>s) x= (((fi-(vstup1 << 3))*(vystup2-vystup1))>>3)+vystup1;
else x=0;
i= (unsigned short) ( sgn ? 32768+x : 32768-x);
/**/
/*synchronizovaný vstup dat do vlastních DA převodníků*/
DacLD_ClrVal();
DacLD_SetVal();
/**/
}
/*
** #####
**
** This file was created by UNIS Processor Expert 2.98 [03.79]
** for the Freescale 56800 series of microcontrollers.
**
** #####
** */

```

## Literatura

- [1] Žilavý P.: *Nabíjení prachových částic*, disertační práce, MFF UK, Praha 2001
- [2] Čermák I.: *Laboruntersuchung elektrischer Aufladung kleiner Staubteilchen*, PhD thesis, Ruprecht-Karls-Universität, Heidelberg 1994
- [3] Beránek M., Jeřáb M., Pavlů J., Šafránková J., Němeček Z.: *Proposal of the electrodynamic trap for photoemission measurements of dust grains*, zasláno do Proceedings of WDS2007, ed. J. Šafránková, MFF UK, Praha 2007
- [4] Beránek M.: *Systém pro levitaci prachových zrn*, diplomová práce, MFF UK, Praha 2007
- [5] *56F8014 Data Sheet, Preliminary Technical Data*, MC56F8014, Rev. 2, Freescale Semiconductor, 2005
- [6] *DAC8814, Quad, Serial Input 16 Bit Multiplying Digital-To-Analog Converter*, SBAS338D, Texas Instruments, 2005
- [7] *56F8014, Demonstration Board User Guide*, MC56F8014DBUG, Rev. 0, Freescale Semiconductor, 2004
- [8] *DAC8803/14 EVM, User's Guide*, SLAU184, Texas Instruments, 2006
- [9] <http://www.processorexpert.com>
- [10] <http://www.freescale.com>
- [11] <http://www.ti.com>