

# **Mikropočítačový systém pro snímání vibrací snímačem ADXL210**

Microcomputer system for vibration sensing by accelerometer  
ADXL210

Ondřej Hladiš

---

Bakalářská práce  
2009



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2008/2009

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej HLADIŠ**

Studijní program: **B 3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Mikropočítačový systém pro snímání vibrací  
snímačem ADXL210**

Zásady pro vypracování:

1. **Prostudujte funkci akcelerometru ADXL210.**
2. **Stanovte hardwarové požadavky na vyhodnocovací mikropočítač a proveďte výběr vhodného typu.**
3. **Proveďte hardwarový návrh mikropočítačové vyhodnocovací jednotky, která bude přenášet naměřené údaje prostřednictvím sériového rozhraní.**
4. **Realizujte prototyp vyhodnocovací jednotky a ověřte její funkci.**
5. **Vytvořte programové vybavení pro příjem a vyhodnocování naměřených údajů v prostředí Matlab.**

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Analog Devices. Low-Cost  $\pm 10$  g Dual-Axis Accelerometer with Duty Cycle ADXL210E., 1999. Dostupný z WWW: <[www.analog.com](http://www.analog.com)>.
2. BURKHARD, Mann. C pro mikrokontroléry. Praha : BEN – technická literatura, 2003. 280 s. ISBN 80-7300-077-6.
3. Freescale Semiconductor. CPU08 Central Processor Unit Reference Manual., 2001. Dostupný z WWW: <[www.freescale.com](http://www.freescale.com)>.
4. Freescale Semiconductor. HCS08 Family Reference Manual, Rev.1., 2003. Dostupný z WWW: <[www.freescale.com](http://www.freescale.com)>.
5. Freescale Semiconductor. MC9S08GB/GT Data Sheet, Rev.2.3., 2004. Dostupný z WWW: <[www.freescale.com](http://www.freescale.com)>.
6. Freescale Semiconductor. MC68HC908KX8 Data Sheet, Rev. 2.1, 2005. Dostupný z WWW: <[www.freescale.com](http://www.freescale.com)>.
7. HRABOVSKÝ, Miroslav, JURÁNEK, Antonín. EAGLE pro začátečníky : Návrhový systém pro plošné spoje. Praha : BEN – technická literatura, 2007. 192 s. ISBN 80-7300-213-2.
8. NOVÁK, Petr. Mobilní roboty – pohony, senzory, řízení. : 1. díl. Praha : BEN – technická literatura, 2005. 256 s. ISBN 80-7300-141-1.
9. VÁŇA, Vladimír. Začínáme s mikrokontroléry Motorola HC08 Nitron. Praha: BEN – technická literatura, 2003. 96 s. ISBN 80-7300-124-1.

Vedoucí bakalářské práce:

**Ing. Petr Dostálek**

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

**20. února 2009**

Termín odevzdání bakalářské práce:

**1. června 2009**

Ve Zlíně dne 13. února 2009

prof. Ing. Vladimír Vašek, CSc.

*děkan*



doc. Ing. Ivan Zelinka, Ph.D.

*ředitel ústavu*

## **ABSTRAKT**

Obsah práce je věnován především způsobu měření akcelerometrem – snímačem zrychlení. Konkrétně je pojednáváno o akcelerometru ADXL210. Tímto senzorem lze měřit nejen změnu rychlosti pohybujícího se tělesa (vibrace, detekce pohybu apod.), ale i vliv gravitace Země (natočení senzoru). Cílem práce bylo navrhnout jednoduchý elektronický obvod s mikropočítačem, který dokáže přijmout a zpracovat signál z akcelerometru a dále jej přenést do počítače přes sériové rozhraní a vykreslit průběh zrychlení do grafu.

Klíčová slova: Mikropočítač, HC08, akcelerometr, ADXL210, MATLAB, RS-232.

## **ABSTRACT**

The content of the work is devoted mainly to the way of measuring by accelerometer - the acceleration sensor. Specifically, it deals with the ADXL210 accelerometer. This sensor can measure not just a change of the speed of moving body (vibration, motion detection, etc.), but also the influence of gravity of the Earth (rotation sensor). The aim of this work was to design and construct a simple electronic circuit with a microcontroller that can receive and process signals from the accelerometer and transfer it to the computer via the serial interface and plot the course of the acceleration in the chart.

Keywords: Microcontroller, HC08, accelerometer, ADXL210, MATLAB, RS-232.

Tímto bych chtěl poděkovat vedoucímu své bakalářské práce, Ing. Petru Dostálkovi, za ochotu, cenné rady a pomoc, kterou mi věnoval při řešení této práce.

*„Zeptáš-li se, budeš pět minut vypadat jako blbec.  
Nezeptáš-li se, budeš blbcem po celý život.“*

Čínské přísloví

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.  
V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 MOTOROLA HC08</b> .....	<b>11</b>
1.1 MC68HC908KX8 .....	12
1.1.1 Zapouzdření.....	13
1.1.2 Paměť .....	14
1.1.3 A/D převodník.....	16
1.1.4 Konfigurační registry .....	16
1.1.5 COP (Watchdog).....	18
1.1.6 Vnitřní generátor hodinového signálu (ICG) .....	18
1.1.7 Externí přerušení (IRQ).....	18
1.1.8 Modul přerušení od klávesnice (KBI).....	19
1.1.9 Modul hlídání napájení (LVI) .....	19
1.1.10 Vstupně/Výstupní porty .....	19
1.1.11 Sériové rozhraní (SCI) .....	21
1.1.12 Systémový modul (SIM) .....	23
1.1.13 Čítač/Časovač (TIM).....	24
1.1.14 Monitor ROM (MON).....	25
<b>2 AKCELEROMETRY</b> .....	<b>27</b>
2.1 POUŽITÍ .....	27
2.2 PRINCIP INTEGROVANÉHO SENZORU ANALOG DEVICES .....	27
2.3 AKCELEROMETR ADXL210.....	29
<b>3 VÝVOJOVÉ PROSTŘEDKY</b> .....	<b>33</b>
3.1 MATLAB .....	33
3.1.1 Základní popis prostředí MATLABu.....	33
3.2 CODEWARRIOR .....	34
3.2.1 Základní popis prostředí.....	34
3.3 EAGLE.....	35
3.3.1 Základní popis prostředí.....	36
<b>II PRAKTICKÁ ČÁST</b> .....	<b>37</b>
<b>4 OBVOD PRO MĚŘENÍ VIBRACÍ</b> .....	<b>38</b>
4.1 NAPÁJENÍ .....	38
4.2 GENERÁTOR HODIN (LADICÍ ADAPTÉR).....	39
4.3 SÉRIOVÉ ROZHŘANÍ A REŽIM MONITOR .....	39
4.4 KONFIGURAČNÍ PROPOJKY .....	40
4.5 SAMOTNÝ MIKROKONTROLÉR .....	41
4.6 OBVOD PRO SNÍMÁNÍ VIBRACÍ .....	41
<b>5 MĚŘENÍ A VÝPOČET DÉLKY IMPULZU</b> .....	<b>43</b>

5.1	POPIS MĚŘENÍ DÉLKY IMPULZU A PERIODY S VYUŽITÍM PŘERUŠENÍ.....	43
<b>6</b>	<b>PROGRAM MIKROPOČÍTAČE .....</b>	<b>45</b>
6.1	KOMUNIKAČNÍ PROTOKOL.....	45
6.2	POPIS PROGRAMU .....	45
<b>7</b>	<b>PROGRAM AKCELEROMETR.....</b>	<b>47</b>
7.1	POPIS PROSTŘEDÍ.....	47
7.2	POPIS PROGRAMU AKCELEROMETR.....	48
<b>8</b>	<b>OVĚŘENÍ ÚDAJŮ .....</b>	<b>53</b>
	<b>ZÁVĚR.....</b>	<b>55</b>
	<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>56</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>57</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>59</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>61</b>
	<b>SEZNAM TABULEK.....</b>	<b>63</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>64</b>



## ÚVOD

Akcelerometry jsou snímače, které měří zrychlení. Toto zrychlení může být buď dynamické, které vzniká změnou rychlosti pohybu tělesa (např. vibrace), nebo statické, které je neustále přítomné z důvodu gravitace Země. Je zřejmé, že možnosti uplatnění akcelerometrů jsou velké. Zřejmě nejčastější využití je v automobilovém průmyslu u airbagů, kde mikro počítač vyhodnocuje při jakém přetížení (a dalších podmínkách) má být airbag vystřelen nebo u zařízení pro řízení jízdní stability vozidla (ESP). Nejen u automobilů pak lze využití najít v zabezpečovacích systémech, kde lze detekovat např. pohyb dveří nebo oken. V průmyslové oblasti je časté měření vibrací. V posledních letech se akcelerometry dostávají i do spotřební elektroniky, zejména pak v mobilních telefonech a fotoaparátech již bývají standardně zabudované a umožňují např. automatické natočení fotografie podle polohy fotoaparátu nebo mobilního telefonu. Moderní akcelerometry Analog Devices na výstupu vysílají signál ve tvaru pulzně – šířkové modulace. Tento signál zvládne zpracovat téměř libovolný levný mikro počítač Freescale řady HC08. Vzhledem k tomu, že má komunikovat s počítačem je vhodné, aby obsahoval zabudované sériové rozhraní (SCI).

Cílem této bakalářské práce je srozumitelnou formou přiblížit problematiku jednočipových mikro počítačů Freescale řady HC08 a akcelerometrů Analog Devices. Praktická funkce akcelerometru ADXL210 je ověřena pomocí jednoduchého elektronického obvodu s mikro počítačem 68HC908KX8, který díky zabudovanému sériovému rozhraní umožňuje jednoduchou komunikaci s počítačem. Mikro počítač zpracuje signály přijaté z akcelerometru a pošle je do počítače, kde je přijme program naprogramovaný v MATLABu, který má za úkol vypočítat výsledné zrychlení a vykreslovat jej do grafu a ukládat do souboru. Snahou bylo zajistit možnost naprogramovat mikro počítač přímo v desce zařízení, aby jej nebylo potřeba stále přemísťovat do vývojového kitu. Díky tomu bude usnadněn případný další vývoj zařízení.

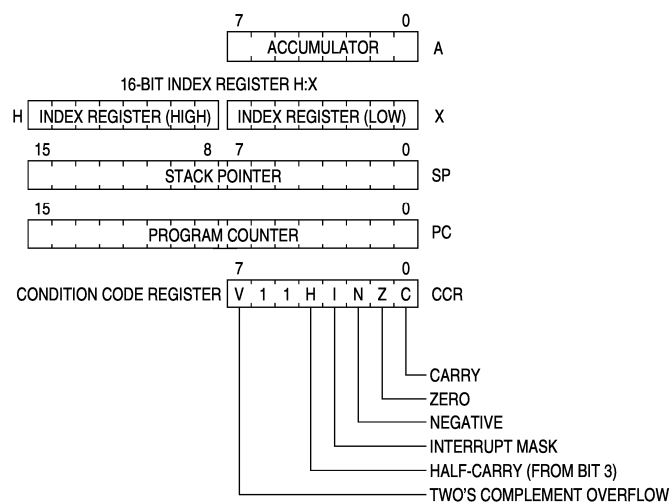
## **I. TEORETICKÁ ČÁST**

## 1 MOTOROLA HC08

Řada HC08 je rodina 8-bitových mikropočítačů firmy Motorola (nyní vyráběny firmou Freescale Semiconductor), která přímo navazuje na řadu HC05, která je dopředně kompatibilní s HC08 na úrovni zdrojového kódu. Typickou vlastností jádra CPU08, které vychází z historického mikroprocesoru 6800 a staršího jednočipového mikropočítače 6801, je minimum pracovních registrů (CPU obsahuje jen akumulátor, indexovací registr, SP, PC a příznakový registr). Díky jednoduššímu jádru bylo možné implementovat instrukce pro rychlý přístup do paměti RAM a k registrům periférií. Podrobně je architektura HC08 popsána ve [2].

Základní registry:

- Akumulátor A (střadač) – základní 8-bitový registr pro uchovávání operandů a mezivýsledků aritmetických a logických operací prováděných CPU.
- Indexový registr H:X – 16-bitový registr, který slouží k adresování paměti (až 64 kB). Je složen ze dvou 8-bitových registrů H a X.
- Ukazatel zásobníku SP – 16-bitový registr, který obsahuje adresu vrcholu zásobníku.
- Programový čítač PC – 16-bitový registr, který obsahuje adresu následující instrukce.
- Příznakový registr CCR – 8-bitový registr indikuje stavy, které jsou nastavovány při aritmetických a logických operacích.



Obr. 1. Registry procesoru HC08.

## 1.1 MC68HC908KX8

MC – značení výrobce – Motorola (dnes Freescale Semiconductor).

68HC – označení základní technologie.

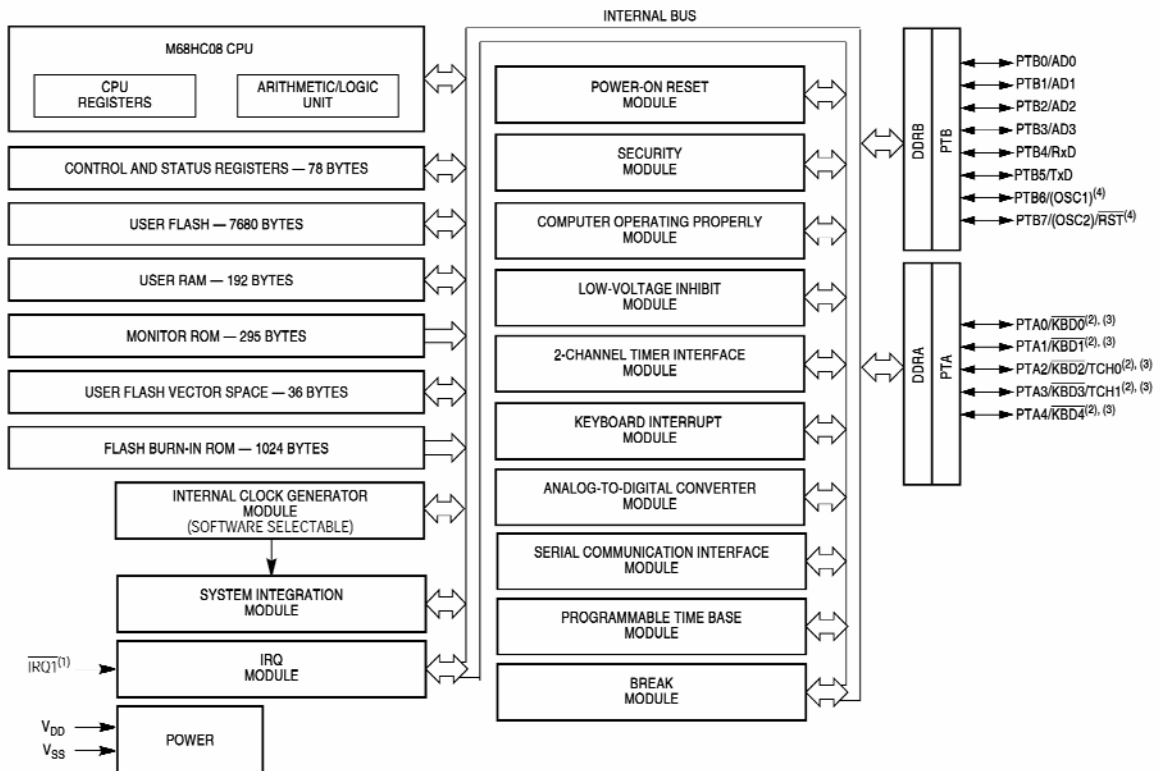
908 – FLASH verze.

KX8 – „rodina“ procesorů.

Základní vlastnosti:

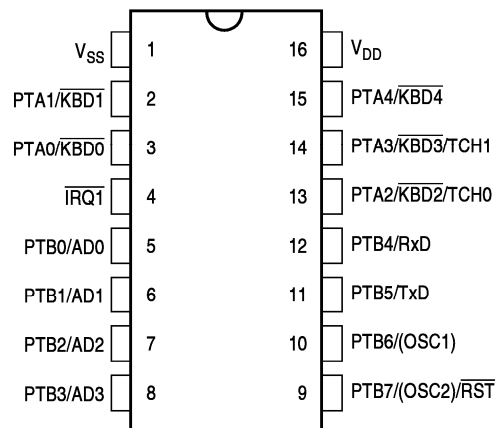
- Von Neumannova architektura, tj. program i data jsou umístěna ve stejném paměťovém prostoru.
- Periferie jsou paměťově mapované.
- Kód kompatibilní s jádrem HC05 (směrem vzhůru).
- 8 kB vnitřní FLASH paměť programovatelná přímo v aplikaci.
- 192 B vnitřní paměti RAM.
- 16-bitový, 2 kanálový čítač/časovač.
- 4 kanálový, 8-bitový A/D převodník.
- Sériové rozhraní SCI.
- 5 vstupů přerušení od klávesnice.
- 13 vstupně/výstupních vývodů.
- Modul hlídání napájení.
- 16 adresovacích režimů.
- Přesuny z paměti do paměti bez použití akumulátoru.
- Plně statická architektura s podporou režimů snížené spotřeby (WAIT a STOP).
- Pokročilé řízení cyklů.
- Optimalizované jádro pro programovací jazyk C.
- Kontrola správné funkce programu, detekce nízkého napájecího napětí, detekce neplatné instrukce, detekce neplatné adresy.
- Instrukce pro práci s čísly v BCD kódu.
- Rychlejší instrukce pro 8-bitové násobení a nová instrukce pro dělení.
- Maximální frekvence vnitřní sběrnice: 8 MHz při 5 V a 4 MHz při 3 V.

Blokové schéma



Obr. 2. Blokové schéma mikropočítače MC68HC908KX8.

1.1.1 Zapouzdření



Obr. 3. Rozložení vývodů mikropočítače MC68HC908KX8.

**Napájecí napětí ( $V_{DD}$  a  $V_{SS}$ )**

Napájecí napětí může být 5 V nebo 3 V (v tom případě ale může být maximální frekvence systémové sběrnice 4 MHz). Z důvodu ochrany před případným šumem se doporučuje připojit blokovací kondenzátor mezi vývody (a to co nejbližší)  $V_{DD}$  a  $V_{SS}$ .

**Vývody pro oscilátor (OSC1 a OSC2)**

Tyto vývody mohou být využity pro vnější zdroj hodinového kmitočtu nebo krystalu příp. keramického rezonátoru. V konfiguračním registru procesoru lze pak nastavit, že bude tento zdroj použit.

**Vnější přerušení (IRQ1)**

Vývod IRQ1 umožňuje přerušení mikroprocesoru vnějším signálem.

**Port A – vstupně/výstupní vývody (PTA4/KBD4 – PTA0/KBD0)**

PTA4/KBD4 – PTA0/KBD0 jsou vstupy/výstupy pro příjem/vysílání číslicového signálu. Alternativně mohou fungovat jako vstupy přerušení od klávesnice nebo jako časovač – vývody PTA2 (TCH0) a PTA3 (TCH1).

**Port B – vstupně/výstupní vývody (PTB7/(OSC2)RST – PTB0/AD0)**

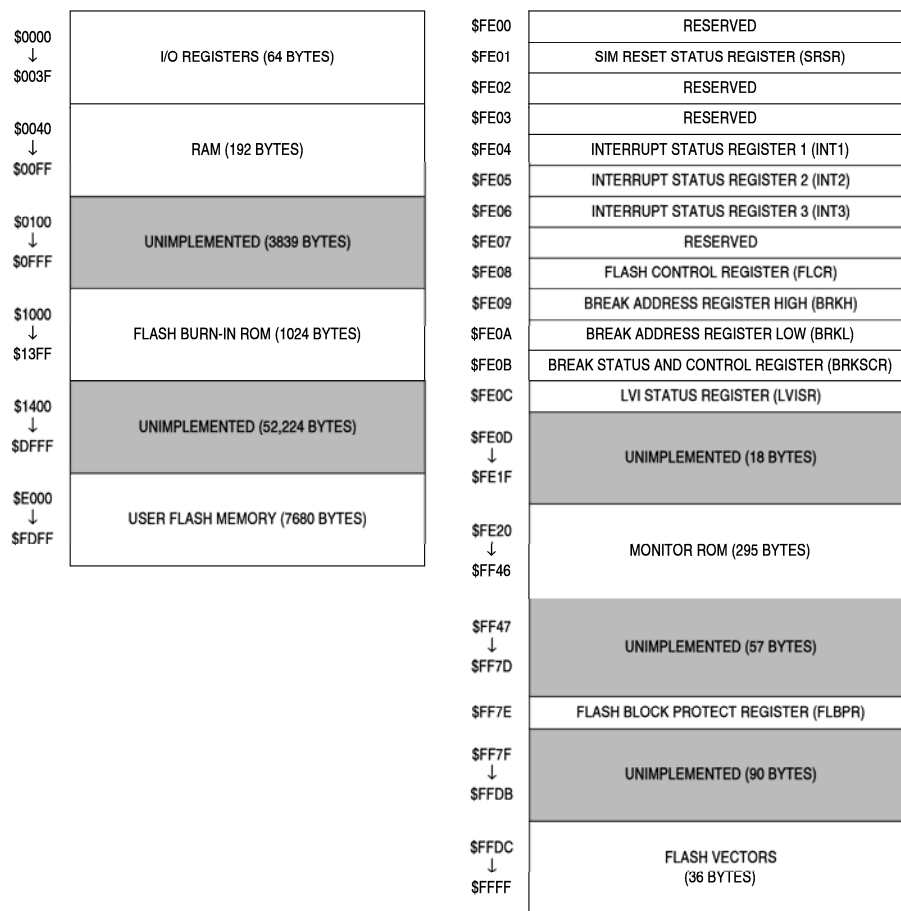
PTB7/(OSC2)RST – PTB0/AD0 jsou vstupy/výstupy pro příjem/vysílání číslicového signálu. PTB7 a PTB6 mohou sloužit jako vstup vnějšího zdroje hodinového kmitočtu. PTB4 a PTB5 mohou sloužit pro sériovou komunikaci přes rozhraní SCI. PTB3 – PTB0 mohou sloužit jako vstup A/D převodníku.

**1.1.2 Paměť**

CPU dokáže díky 16 – bitové adresové sběrnici adresovat 64 kB adresový prostor.

Paměťová mapa obsahuje:

- 7680 B paměti FLASH.
- 192 B paměti RAM.
- 36 B uživatelsky definovaných vektorů.
- 295 B paměti ROM.



Obr. 4. Paměťová mapa mikropočítače MC68HC908KX8.

### I/O registry

Jedná se řídicí, stavové a datové registry mikroprocesoru.

### RAM

V paměti RAM se nachází zápisník, tedy paměť s libovolným přístupem, který slouží pro ukládání mezivýsledků a proměnných. Dále se zde také nachází zásobník, jehož přesné umístění lze nastavit. 16-bitový ukazatel zásobníku teoreticky umožňuje zásobník umístit kamkoli v 64 kB paměťovém prostoru. Prakticky je to samozřejmě možné pouze v paměti RAM.

### Monitor ROM

Slouží k programování a ladění programu přímo v aplikaci a obsahuje instrukce pro funkce Monitoru.

## FLASH paměť

FLASH paměť má kapacitu 7680 B, přídatných 36 B pro uživatelské vektory a jeden 8-bitový registr, který umožňuje chránit paměť před smazáním.

### 1.1.3 A/D převodník

Mikropočítač MC68HC908KX8 obsahuje 8-bitový analogově – digitální převodník s postupnou lineární aproximací. Převodník je 4 kanálový se zabudovaným analogovým multiplexerem pro jejich snadné přepínání. Samotný převod může probíhat jednorázově nebo kontinuálně a po dokončení převodu je výsledek zapsán do datového registru A/D převodníku a je nastaven příznak konce převodu nebo je vyvoláno přerušení. Převodník má nastavitelný zdroj hodinového signálu (vnitřní nebo vnější oscilátor) a pomocí předděličky by frekvence pro něj měla být nastavena přibližně 1 MHz. Převod pak bude trvat 16 – 17  $\mu$ s.

### 1.1.4 Konfigurační registry

Tyto dva systémové registry slouží k inicializaci různých nastavení mikroprocesoru a mohou být zapsány pouze jednou po každém resetu. Všechny bity obou konfiguračních registrů jsou po resetu vynulovány. Je doporučeno, aby byly zapsány okamžitě po resetu. Čteny mohou být kdykoli.

#### Registr CONFIG1

Registr CONFIG1 nabízí nastavení hlídacího obvodu COP (bity COPRS a COPD), modulu hlídání napájení LVI (bity LVISTOP, LVIRSTD, LVIPWRD a LVI5OR3) a povolení instrukce STOP.

Address:	\$001F							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3 <sup>(1)</sup>	SSREC	STOP	COPD
Write:								
Reset:	0	0	0	0	0	0	0	0
Other Resets:	0	0	0	0	U	0	0	0

1. The LVI5OR3 bit is cleared only by a power-on reset (POR).

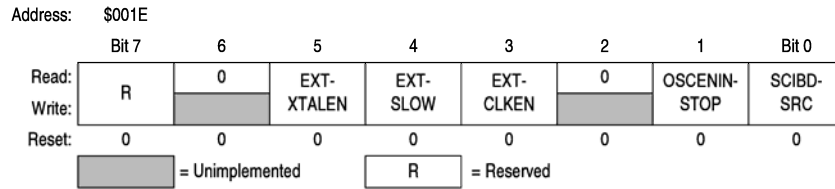
U = Unaffected

Obr. 5. Konfigurační registr CONFIG1.



**Registr CONFIG2**

Registr CONFIG2 slouží k nastavení zdroje hodinového signálu mikropočítače a sériového rozhraní SCI.



Obr. 6. Konfigurační registr CONFIG2.

**EXTCLKEN** (External Clock Enable Bit) a **EXTXTALEN** (External Crystal Enable Bit)

- povolení použití vnějšího zdroje hodinové signálu.

Tab. 1. Nastavení zdroje hodinového signálu.

Nastavení bitů		Funkce vývodu		Popis
EXTCLKEN	EXTXTALEN	PTB6/ (OSC1)	PTB7/ (OSC2)/RST	
0	0	PTB6	PTB7	Vnější oscilátor zakázán – defaultní
0	1	PTB6	PTB7	Vnější oscilátor zakázán dokud není nastaven bit EXTCLKEN
1	0	OSC1	PTB7	Povolen zdroj vnějšího signálu na vývodu OSC1
1	1	OSC1	OSC2	Povolen zdroj vnějšího signálu – krystalu připojeného na vývody OSC1 a OSC2

**OSCENINSTOP** (Oscillator Enable In Stop Mode Bit) – povolení vnitřního oscilátoru v režimu STOP.

- 0 = oscilátor zakázán ve STOP režimu
- 1 = oscilátor povolen ve STOP režimu

**SCIBDSRC** (SCI Baud Rate Clock Source Bit) – nastavení zdroje hodinového signálu pro SCI.

0 = jako hodinový signál pro SCI je použit vnitřní nebo vnější hodinový signál (CGMXCLK)

1 = jako hodinový signál pro SCI je použita vnitřní frekvence sběrnice

### 1.1.5 COP (Watchdog)

COP je nezávislý čítač, který při přetečení vyvolá reset. Pomáhá tak dostat se z nekonečné smyčky či jinak zaseknuté části programu. Je potřeba jej periodicky nulovat, aby nedošlo k resetu bez jakékoliv chyby. COP je možné zakázat v registru CONFIG1.

### 1.1.6 Vnitřní generátor hodinového signálu (ICG)

Slouží jako stabilní zdroj hodinového kmitočtu mikropočítače bez využití jakýchkoli vnějších komponent. ICG je zdroj hodinového signálu pro COP, LVI a další obvody. Generuje také systémový hodinový signál (bus clock) pro systémový modul (SIM).

Vnitřní frekvence sběrnice je rovna jedné čtvrtině frekvence CGMXCLK (výstupní frekvence ICG), která je odvozena buď z vnitřního nebo vnějšího zdroje hodinového signálu, a jedné polovině frekvence CGMOUT (výstupní frekvence ICG – je rovna polovině CGMXCLK).

Vlastnosti:

- Volitelný vnější generátor hodin, buď vnější zdroj hodinového signálu přivedeného na jeden vývod OSC1 nebo krystalový oscilátor připojený na vývody OSC1 a OSC2.
- Vnitřní zdroj hodinového kmitočtu s výstupní frekvencí nastavitelnou v celočíselných násobcích nominální frekvence (307,2 kHz  $\pm$  25 %).
- Rozptyl hodnot frekvence lze zlepšit až na  $\pm$  2 % pomocí registru (ICGR) – nastavení vnitřního kondenzátoru ICG.

### 1.1.7 Externí přerušení (IRQ)

IRQ modul je obvod, který zajišťuje obsluhu maskovatelného přerušení vnějším signálem připojeným na vývod IRQ1. IRQ má vnitřní pull-up rezistor, nastavitelnou citlivost na

(sestupnou) hranu nebo (sestupnou) hranu a (nízkou) úroveň. Obvod také umožňuje automatické potvrzení přerušení.

### 1.1.8 Modul přerušení od klávesnice (KBI)

KBI poskytuje 5 nezávislých, maskovatelných přerušení, určených zejména pro obsluhu klávesnici, ale je možné je využít jako vstup pro jakýkoliv vnější signál. KBI obvod je podobný IRQ, obsahuje stejné možnosti nastavení a navíc umožňuje „probuzení“ z režimů s nízkou spotřebou.

### 1.1.9 Modul hlídání napájení (LVI)

Modul LVI sleduje napětí na vývodu  $V_{DD}$  a umožňuje vyvolat reset procesoru, pokud úroveň napětí klesne pod hodnotu napětí  $V_{TRIPF}$ , která je 3,9 V u 5 V napájení a 2,45 V u 3 V napájení. Pokud je zakázán reset modulem LVI je možné ve stavovém registru LVI číst příznak LVIOUT, který indikuje, že napájecí napětí pokleslo pod hodnotu  $V_{TRIPF}$ .

### 1.1.10 Vstupně/Výstupní porty

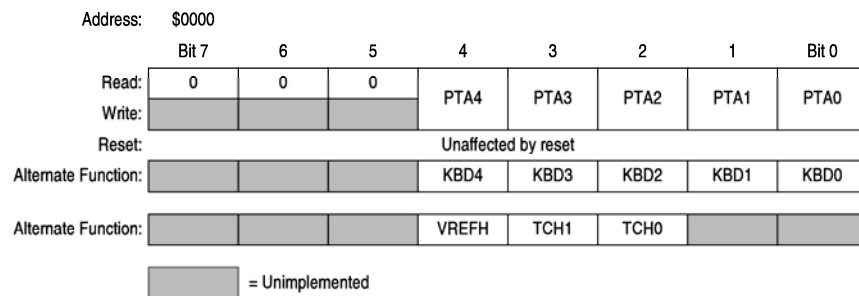
13 obousměrných vstupně/výstupních vývodů. Každý vývod může být nastaven jako vstup nebo výstup a volitelně lze pro Port A použít zabudované pull-up rezistory. Některé vývody mají alternativní funkci. Všechny vývody mohou dodávat nebo spotřebovávat proud až 15 mA.

#### Port A

Port A obsahuje 5 vstupně/výstupních vývodů (PTA0 – PTA4), které mohou být alternativně využity jako vstupy přerušení od klávesnice (KBD0 – KBD4). 2 vývody jsou ještě navíc sdíleny s obvodem časovače (TCH0 a TCH1).

#### Port A Data Register (PTA)

Datový registr portu A. Obsahuje kopii hodnot přijatých na vývodech portu A, pokud jsou nastaveny jako vstupy, nebo hodnoty, které budou na vývody poslány, pokud jsou nastaveny jako výstupy.



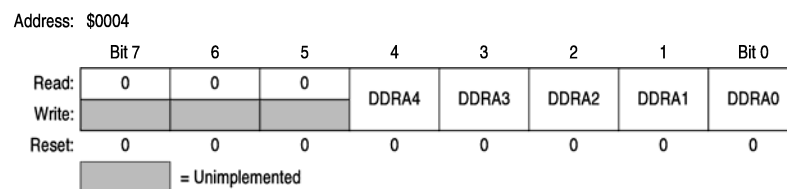
Obr. 7. Port A Data Register (PTA).

### Data Direction Register A (DDRA)

Registr portu A, který určuje, zda se jednotlivé vývody chovají jako vstupy nebo výstupy.

0 = vstup

1 = výstup



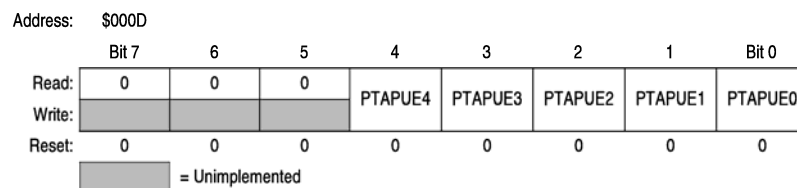
Obr. 8. Data Direction Register A (DDRA).

### Port A Input Pullup Enable Register (PTAPUE)

Povoluje zabudované pull-up rezistory.

0 = pull-up rezistor odpojen

1 = pull-up rezistor připojen



Obr. 9. Port A Input Pullup Enable Register (PTAPUE).

### Port B

Port A obsahuje 8 vstupně/výstupních vývodů (PTB0 – PTB4), které navíc sdílejí svou funkci se vstupy A/D převodníku (AD0 – AD4), vstupem (RxD) a výstupem (TxD) sériového rozhraní SCI a se vstupy vnějšího zdroje hodinového signálu (OSC0 a OSC1).

## Port B Data Register (PTB)

Address:	\$0001							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Write:								
Reset:	Unaffected by reset							
Alternate Function:	OSC2	OSC1	TxD	RxD	AD3	AD2	AD1	AD0

Obr. 10. Port B Data Register (PTB).

### 1.1.11 Sériové rozhraní (SCI)

Rozhraní SCI je sériový asynchronní komunikační systém typu UART, který umožňuje komunikovat na velké vzdálenosti s dalšími mikropočítači nebo jinými systémy vybavenými rozhraním RS-232 (PC, průmyslové počítače).

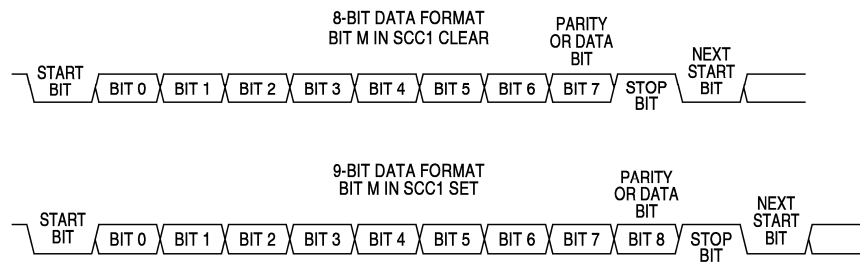
Vlastnosti SCI u HC08:

- Plně duplexní provoz.
- Používá standardní NRZ formát.
- Nastavitelná přenosová rychlost (32 rychlostí).
- Programovatelná délka datového rámce 8 nebo 9 bitů.
- Zcela oddělený přijímač a vysílač.
- Podpora hardwarového generování a kontroly parity.
- Pokročilá detekce chyb na sériové lince (šum na lince, chyba parity, chybný datový rámec, ...).
- Dvojitá vyrovnávací paměť přijímače i vysílače.

Přijímač a vysílač SCI pracují na sobě nezávisle, ačkoli používají stejný zdroj přenosové rychlosti (baud rate). Tato rychlost je určena v konfiguračním registru CONFIG2 bitem SCIBDSCR. Pokud je bit nastaven je zdrojem přenosové rychlosti vnitřní frekvence sběrnice, v opačném případě je zdrojem frekvence CGMXCLK.

#### Formát datového rámce

Datový rámec obsahuje 8 (9) datových bitů, které jsou uzavřeny mezi tzv. Start a Stop bit, sestupná hrana Start bitu data synchronizuje. V klidovém stavu je signál v úrovni logické 1. Přenos začíná Start bitem – překlopením signálu do logické úrovně 0. Pak jsou vyslána data o délce 8 bitů (1 B) a případně parita. Přenosová rychlost je po celou dobu konstantní se udává v baudech (počet změn úrovní signálu za jednotku času).



Obr. 11. Standardní NRZ formát datového rámce.

SCI obsahuje 3 řídicí registry (SCC1 – SCC3), které povolují a řídí komunikaci přes sériové rozhraní, 2 stavové registry (SCS1 a SCS2), které indikují dokončení přenosu, chyby v komunikaci a detekují synchronizační znak a datový registr, který slouží jako vyrovnávací paměť (buffer) pro přijímaná a vysílaná data a registr pro nastavení přenosové rychlosti.

Address: \$0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented     
 R = Reserved

Obr. 12. SCI Baud Rate Register (SCBR).

Přenosová rychlost se vypočítá dle vztahu:

$$BR = \frac{f_{BAUDCLK}}{64 \cdot PD \cdot BD} \quad (1)$$

$f_{BAUDCLK}$  – frekvence zdroje hodinového signálu SCI

$PD$  – předdělička

$BD$  – dělička přenosové rychlosti

Tab. 2. Hodnoty nastavení modulační rychlosti.

SCP [1:0]	Předdělička (PD)	SCR[2:1:0]	Dělička modulační rychlosti (BD)
00	1	000	1
01	3	001	2
10	4	010	4
11	13	011	8
		100	16
		101	32
		110	64
		111	128

### 1.1.12 Systémový modul (SIM)

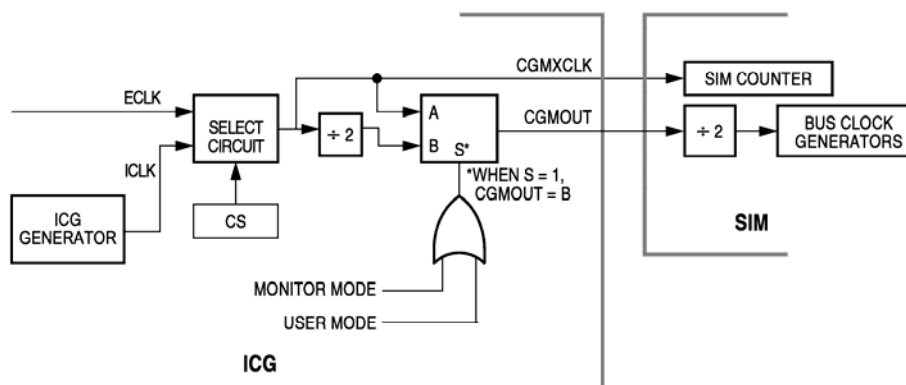
Modul SIM kontroluje CPU a generuje hodinové signály, řídí přerušení (podporuje až 24 typů přerušení), reset a obvod COP, ovládá režimy STOP a WAIT.

Zdroje resetu HC08:

- Obvod Power-on reset (POR).
- Obvod COP.
- Obvod hlídání napájení (LVI).
- Neplatný operační kód (opkód).
- Neplatné adresy.
- Reset při vynuceném vstupu do režimu Monitor (MENRST).

### Generování a řízení vnitřní frekvence sběrnice (bus clock)

Generátor sběrnicové frekvence, který vytváří hodinový signál pro CPU a periferie je odvozen ze vstupujícího hodinového signálu z ICG – CGMOUT. V uživatelském režimu je vnitřní frekvence sběrnice rovna jedné čtvrtině frekvence CGMXCLK a jedné polovině CGMOUT.



Obr. 13. Systémové hodinové signály.

### 1.1.13 Čítač/Časovač (TIM)

Obvod TIM je dvoukanálový čítač, který umožňuje mikropočítači generování a měření přesných časových intervalů – funkce Input Capture, Output Compare, PWM. Je konstruován jako volně běžící 16-bitový čítač signálu odvozeného od strojových cyklů mikropočítače se zabudovanou programovatelnou sedmibitovou předděličkou. Mimo to představuje zdroj časové základny obvodu COP.

#### Input Capture

Jednotka je tvořena 16-bitovým záchytným registrem připojeným na sběrnici hlavního čítače, obvody řízení záchyty a oddělovacími obvody datové sběrnice. Principem této jednotky je detekce zvoleného typu hrany (náběžná, sestupná nebo libovolná) vstupního signálu a v okamžiku úspěšné detekce je generován impulz pro záchytný registr. V záchytném registru se tedy uchovává aktuální hodnota hlavního čítače. Záchytnou jednotku lze tedy využít k měření délky periody, impulzu, fázového posuvu atd.

#### Output Compare

Jednotka výstupního komparátoru slouží k přesnému generování hrany signálu v závislosti na aktuální hodnotě hlavního čítače. Jednotku tvoří 16-bitový datový registr, 16-bitový komparátor a obvody generování výstupní hrany. Datový registr uchovává zadanou hodnotu, kterou komparátor porovnává s aktuální hodnotou hlavního čítače a v okamžiku rovnosti obou hodnot je generován řídicí signál pro obvody generování výstupní hrany.



## PWM

Jednotka výstupního komparátoru umožňuje také generovat pulzně – šířkovou modulaci (PWM). Při přetečení hlavního čítače přechází signál do úrovně log. 1. Periodu signálu určuje hodnota zapsaná v modulo registru časovače (TMODH/L) a šířku impulzu v registru příslušného kanálu časovače TCHxH/L.

Přerušení od časovače:

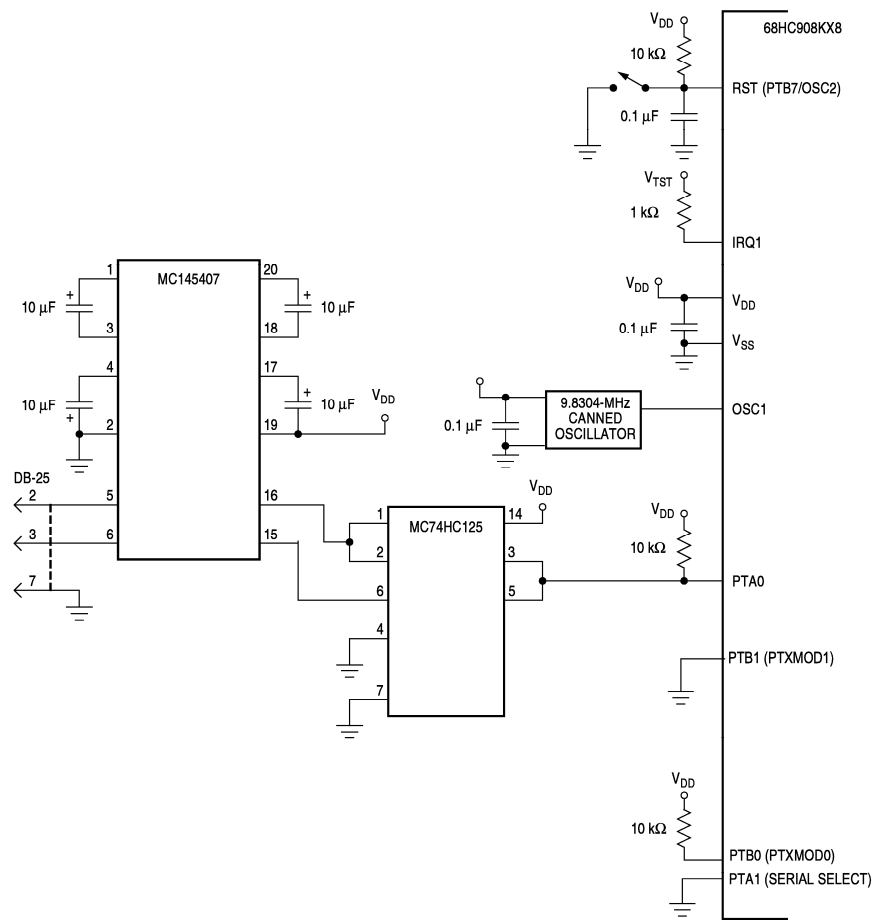
- Příznak přetečení časovače (TOF) – příznak TOF je nastaven, jakmile časovač dosáhne programovatelné hodnoty zapsané v modulo registru. Pokud je nastaven bit TOIE (povolení přerušení od přetečení časovače) registru TSC, nastane přerušení.
- Příznak přerušení od kanálu časovače (CH0F a CH1F) – příznak CHx F je nastaven, pokud nastane událost funkce Input Capture nebo Output Compare. Pokud je nastaven bit CHxIE registru TSCx je přerušení povoleno.

Popis registrů časovače je přiložen v příloze P I.

### 1.1.14 Monitor ROM (MON)

Tento monitor umožňuje kompletní testování mikroprocesoru z externího počítače od kterého přijímá a vykonává příkazy vedené přes jednodrátové rozhraní a dokáže vykonávat program z počítače v paměti RAM nebo FLASH. Obsahuje programovací rutiny a umožňuje zakázat neautorizované přečtení paměti FLASH. Veškerá komunikace mezi mikropočítačem a počítačem probíhá přes vývod PTA0.

Podmínkou pro vstup do režimu Monitor je nutné přivést napětí  $V_{TST}$  ( $V_{DD} + 2,5$  V až  $V_{DD} + 4$  V) na vývod IRQ1 a další nastavení příslušných vývodů. Druhou možností, určenou k programování přímo v aplikaci je vynucený vstup do režimu Monitor bez nutnosti přivést vyšší napětí  $V_{TST}$  na vývod IRQ1, ale pouze pokud je vynulovaný resetovací vektor na adrese \$FFFE a \$FFFF. V obou případech je nutné, aby na vývodu PTA1 byla nízká úroveň napětí (log. 0) po dobu 24 cyklů zdroje hodinové frekvence CGMXCLK po resetu. Podrobné informace o Monitoru a obecně o mikropočítači 68HC908KX8 jsou dostupné v [1].



Obr. 14. Zapojení, při kterém je mikropočítač v režimu Monitor.

Tab. 3. Nastavení pro vstup do režimu Monitor.

$\$FFFE/\$FFFF$	IRQ1	PTB1	PTB0	PTA1	PTA0	CGMOUT	Frekvence sběrnice
X	$V_{TST}$	0	1	0	1	CGMXCLK/2	CGMOUT/2
$\$FF$	$V_{DD}$	X	X	0	1	CGMXCLK/2	CGMOUT/2

## 2 AKCELEROMETRY

Akcelerometry jsou senzory určené pro měření statického zrychlení (síly vzniklé působením gravitace Země) nebo dynamického zrychlení (síly vzniklé změnou rychlosti pohybujícího se předmětu – senzoru) a to jak měření odstředivých a setrvačných sil, tak i určování polohy tělesa, jeho náklon nebo vibrace. Dnešní moderní akcelerometry bývají integrovány v jednom pouzdře s celou mechanickou strukturou. Z principu je statické zrychlení neustále přítomné, proto je nutné jej při měření dynamického zrychlení odfiltrovat.

### 2.1 Použití

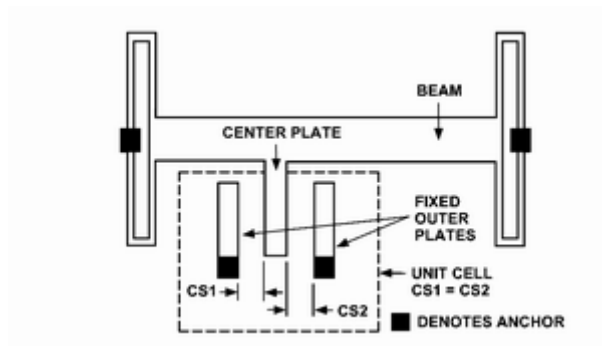
- Automobilový průmysl – senzory pro airbasy, zařízení pro řízení jízdní stability vozidla (ESP).
- Měření vibrací.
- Měření natočení a detekce otáčení.
- Přesné měření náklonu s rychlou odezvou.
- Měření a detekce pohybu a rychlosti.
- Měření a detekce seismické aktivity.
- Měření akcelerace.
- Měření odstředivé síly a zrychlení.
- Navigační systémy.
- Zabezpečovací systémy vozidel.
- Počítačové komponenty.

Prakticky veškerý mechanický pohyb, který lze fyzikálně vztáhnout ke gravitaci lze těmito senzory měřit s dostatečně velkou přesností a hlavně vysokou rychlostí odezvy na změny.

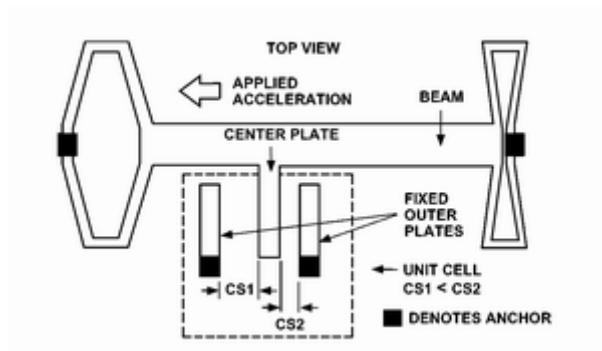
### 2.2 Princip integrovaného senzoru Analog Devices

Jak už bylo řečeno celá mechanická struktura senzoru spolu s vyhodnocovacími obvody je integrována v jednom monolitickém obvodu. Jedná se o technologii iMEMS firmy Analog devices. Samotný senzor je vlastně polykřemíková mikromechanická struktura složená z nosníku (beam), pružin a pevných úchyťů, která je volně pohyblivá na povrchu křemíkového monokrystalu. Princip měření je založen na tom, že vlivem síly vzniklé

zrychlením pouzdra senzoru způsobí deformaci mechanické struktury obvodu a ta je převedena na změnu kapacity proměnlivého diferenciálního kondenzátoru, který se skládá ze dvou pevných desek (fixed outer plates) a prostřední desky (center plate), která je spojena s deformujícím se nosníkem. Na pevných deskách jsou vybudeny dva obdélníkové signály vzájemně posunutě o  $180^\circ$ , přičemž kondenzátor funguje jako dělič těchto dvou signálů. Při posunutí prostřední desky (působením zrychlení) dojde ke změně dělicího poměru kondenzátoru. Na výstupu se objeví obdélníkový signál o amplitudě úměrné hodnotě zrychlení a fázi, která nese informaci o směru pohybu nosníku, tedy o směru působícího zrychlení. Podrobněji je princip popsán v [7].



Obr. 15. Struktura senzoru bez působení zrychlení (0 g).



Obr. 16. Struktura senzoru při působení zrychlení.

### 2.3 Akcelerometr ADXL210

ADXL210 je nízkoodběrový, 2-osý akcelerometr s měřicím rozsahem zrychlení  $\pm 10$  g, který umožňuje měřit jak dynamické zrychlení (např. vibrace) tak i statické zrychlení (např. gravitace).

Výstupem jsou digitální signály, jejichž střída (podíl délky impulzu k periodě) je úměrná ke zrychlení v každé ze 2 měřených os. Výstupy lze snímat přímo čítačem mikropočítače, není potřeba žádný A/D převodník nebo jiné přídavné obvody. Výstupní perioda je nastavitelná v rozsahu 0,5 ms až 10 ms pomocí rezistoru ( $R_{SET}$ ). Vývody  $X_{FILT}$  a  $Y_{FILT}$  jsou určeny k nastavení šířky pásma, ale lze z nich také získat analogový signál.

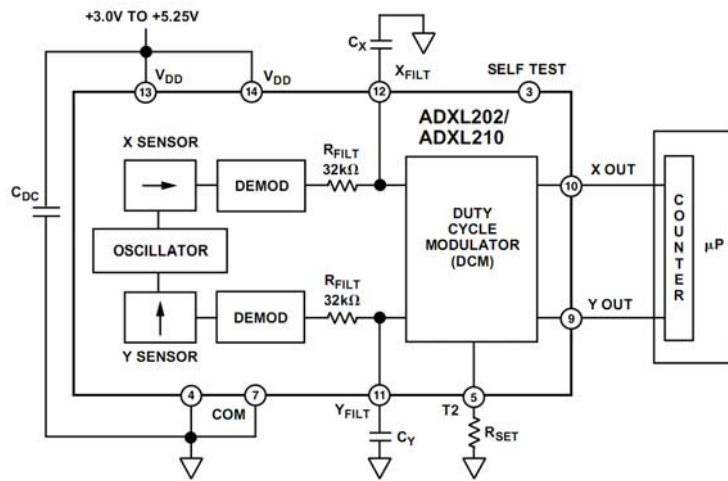
Šířka pásma je nastavitelná v rozsahu 0,01 Hz až 5 kHz pomocí kondenzátorů  $C_X$  a  $C_Y$ . Typická úroveň šumu je  $500 \mu g \sqrt{Hz}$  a s rozlišením až 5 mg ve frekvenčním pásmu 60 Hz.

ADXL210 je dostupný v neprodyšném (hermetickém) 14-pinovém pouzdře, běžný typ umožňuje použití při teplotách 0 °C až 70 °C a průmyslový typ při teplotách -40 °C až 85 °C.

Základní charakteristiky:

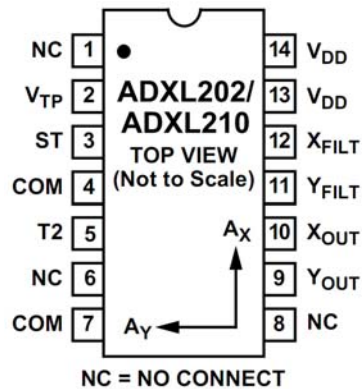
- 2-osý snímač zrychlení umožňující měření statického i dynamického měření.
- Nastavitelná perioda.
- Nízká spotřeba < 0,6 mA.
- Rychlejší odezva než u elektrolytických, rtuťových nebo tepelných snímačů.
- Nastavení šířky pásma pomocí kondenzátoru, rozlišení 5 mg při šířce pásma 60 Hz.
- Napájení 3 V až 5,25 V.
- Maximální impulz 1000 g.

Blokové schéma



Obr. 17. Blokové schéma akcelerometru ADXL210.

Rozložení vývodů

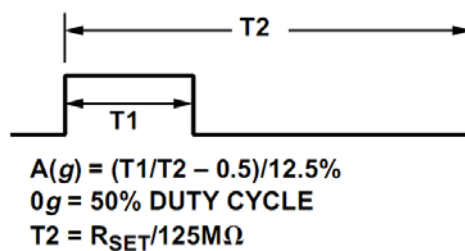


Obr. 18. Rozložení vývodů akcelerometru ADXL210.

Tab. 4. Popis jednotlivých vývodů akcelerometru ADXL210.

Vývod	Název	Popis
1	NC	Nepřipojen
2	V <sub>TP</sub>	Testovací bod
3	ST	Samočinný test
4	COM	Zem
5	T2	Připojení R <sub>SET</sub> pro nastavení periody T2
6	NC	Nepřipojen
7	COM	Zem
8	NC	Nepřipojen
9	Y <sub>OUT</sub>	Výstupní signál osy Y
10	X <sub>OUT</sub>	Výstupní signál osy X
11	Y <sub>FILT</sub>	Připojení filtračního kondenzátoru pro osu Y
12	X <sub>FILT</sub>	Připojení filtračního kondenzátoru pro osu X
13	V <sub>DD</sub>	Napájení 3 V až 5,25 V
14	V <sub>DD</sub>	Napájení 3 V až 5,25 V

Výstupní signály jsou v podobě pulzně – šířkové modulace (PWM) a zrychlení je úměrné poměru délky impulzu T1 k délce periody T2. Perioda T2 je konstantní a je nastavitelná rezistorem R<sub>SET</sub>. Nominální hodnota délky T1 je 50 % periody T2 pro působící zrychlení 0 g.



Obr. 19. Výstupní signál akcelerometru ADXL202 (12,5% platí pro ADXL202, pro 210 – 4%).

Perioda signálu se vypočítá podle vztahu:

$$T2 = \frac{R_{SET}(M\Omega)}{125(M\Omega)} \quad (2)$$

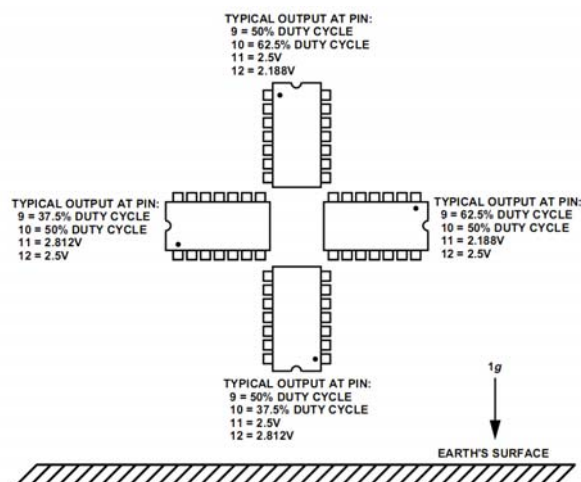
a výsledné zrychlení se vypočítá podle vztahu:

$$A(g) = \frac{\left(\frac{T1}{T2} - 0,5\right)}{4\%} \quad (3)$$

Tab. 5. Nastavení šířky frekvenčního pásma včetně typických úrovní šumu.

Šířka pásma [Hz]	C <sub>X</sub> , C <sub>Y</sub> [μF]	Šum (RMS) [mg]
10	0,47	1,9
50	0,1	4,3
100	0,05	6,1
200	0,027	8,7
500	0,01	13,7

Na Obr. 20 je právě příklad změny výstupního signálu senzoru při pomalém naklonění (otáčení) v gravitačním poli Země a napájení +5V. Více informací o akcelerometru ADXL210 je k dispozici ve [3].



Obr. 20. Parametry výstupního signálu při otáčení senzorem ADXL202/ADXL210.



## 3 VÝVOJOVÉ PROSTŘEDKY

### 3.1 MATLAB

MATLAB je programové prostředí a skriptovací programovací jazyk pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, měření a zpracování signálů, návrhy řídicích a komunikačních systémů.

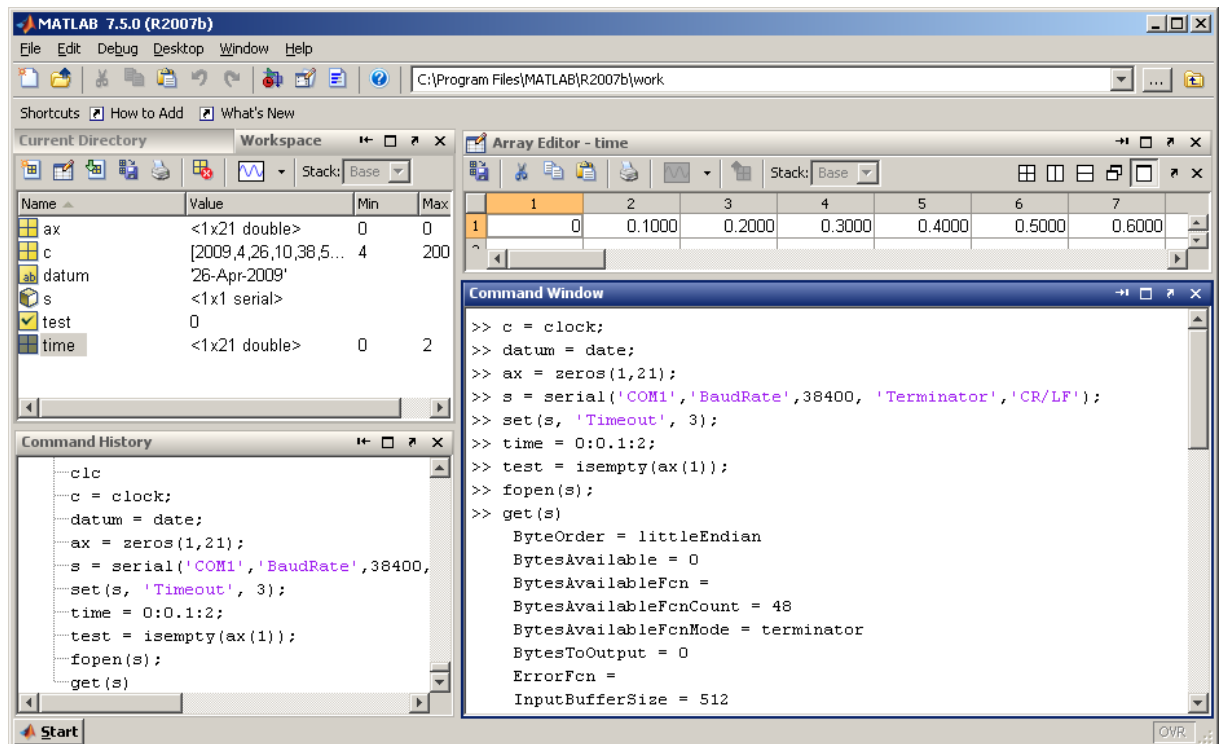
Základní datovou strukturou při výpočtech v MATLABu jsou matice. Samotný název MATLAB vznikl jako zkratka slov MATrix LABoratory („maticová laboratoř“). Vlastní programovací jazyk vychází z jazyka Fortran a umožňuje práci s objekty. MATLAB podporuje operační systémy MS Windows, Unix, Linux, Mac OS X.

Poslední verze MATLABu mají velmi propracovanou grafiku. Umožňuje vytvářet grafy a různě je upravovat (např. stínovat, vybarvovat nebo vkládat do grafu ovládací prvky atd.). Obsahuje interaktivní nástroje pro tvorbu grafického uživatelského rozhraní. Díky propojení s jazykem Java je možné používat knihovny vytvořené pro Javu. V MATLABu lze využívat části kódu vytvořené v jazyce C a naopak do jazyka C exportovat.

Nástavbou MATLABu je Simulink – program pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení především nelineárních diferenciálních rovnic. Otevřená architektura MATLABu vedla ke vzniku knihoven funkcí, nazývaných toolboxy, které rozšiřují použití programu v příslušných vědních a technických oborech. Obsahuje například toolboxy pro oblast řízení a regulace systému (Control System Toolbox), toolbox pro analýzu dat časových řad na finančních trzích (Financial Time Series Toolbox), toolbox pro práci s neuronovými sítěmi (Neural Network Toolbox) a mnoho dalších. Další informace o prostředí MATLAB lze nalézt v [5] a [12].

#### 3.1.1 Základní popis prostředí MATLABu

Do *Command Window* se zadávají příkazy, které se mají provést. Část *Command History* zobrazuje naposled použité příkazy a v poslední lze zobrazit buď *Workspace*, který zobrazuje vytvořené proměnné, jejichž obsah lze prohlížet v *Array Editoru*, nebo *Current Directory*, který zobrazuje obsah pracovního adresáře. Pracovní adresář lze změnit v editačním políčku nahoře.



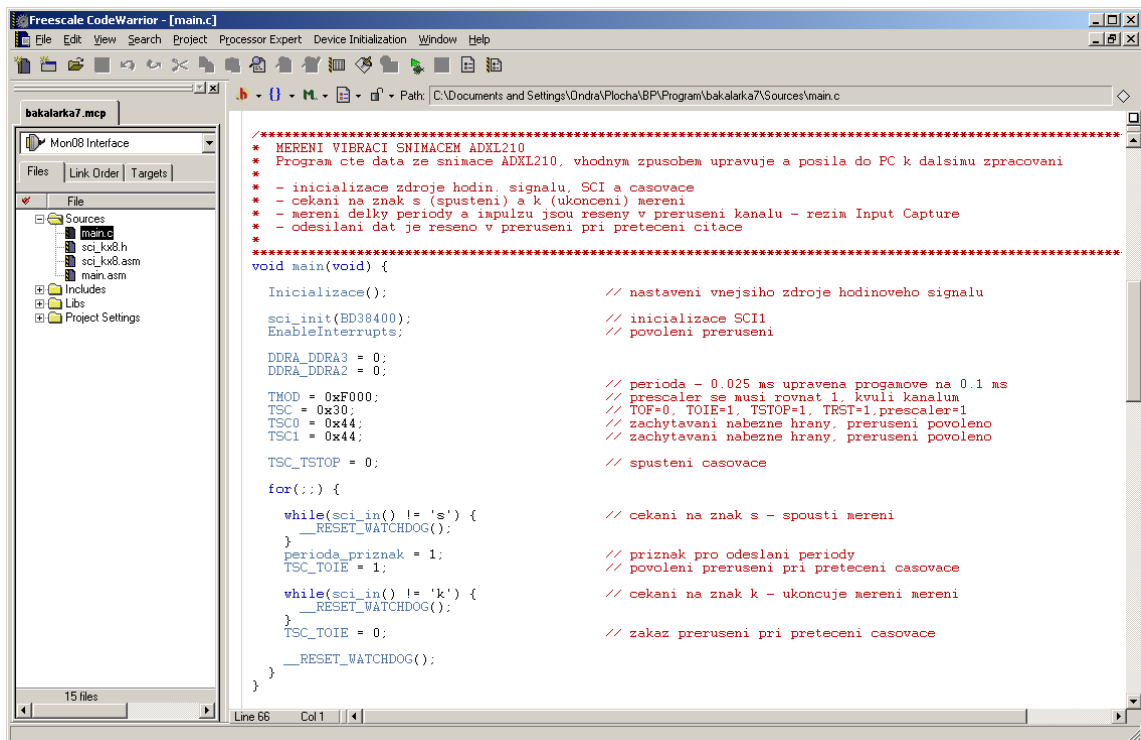
Obr. 21. Prostředí MATLAB.

## 3.2 CodeWarrior

CodeWarrior je integrované vývojové prostředí (IDE), které umožňuje tvořit, sestavovat, kompilovat i ladit kód v jediném IDE. Nabízí intuitivní uživatelské grafické rozhraní, které je pro všechny mikrokontroléry Freescale společné a umožňuje tak snadný přechod k jiným mikrokontrolérům Freescale. Obsahuje vestavěný simulátor a ladící program a poskytuje tak účinné nástroje pro vývoj aplikací. Vyvíjet aplikace lze nejen v jazyce symbolických adres (assembleru), ale i ve vyšších programovacích jazycích (C/C++). Kompilátor CodeWarrior, kompatibilní s platformou ANSI C, byl optimalizován pro zvýšení výkonu a snížení velikosti kódu. Další informace lze nalézt v [10].

### 3.2.1 Základní popis prostředí

V levé části jsou standardně zobrazeny soubory projektu a vpravo editor pro tvorbu zdrojového kódu. Sestavení programu se provádí příkazem **Project -> Make (F7)** a spustit příkazem **Project -> Debug (F7)**, který spouští simulátor a ladící program (debugger). Sestavit a spustit program je také možné pomocí ikon na nástrojové liště.



Obr. 22. Integrované vývojové prostředí CodeWarrior.

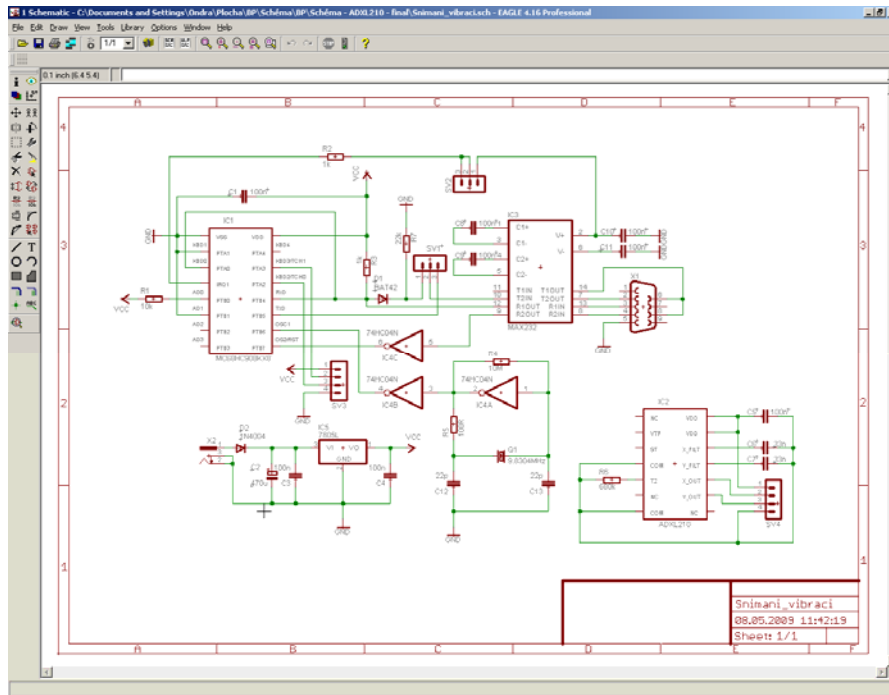
### 3.3 EAGLE

EAGLE je návrhový systém, který slouží k vytváření a editaci elektrotechnických schémat a obvodů. Skládá se ze tří základních modulů:

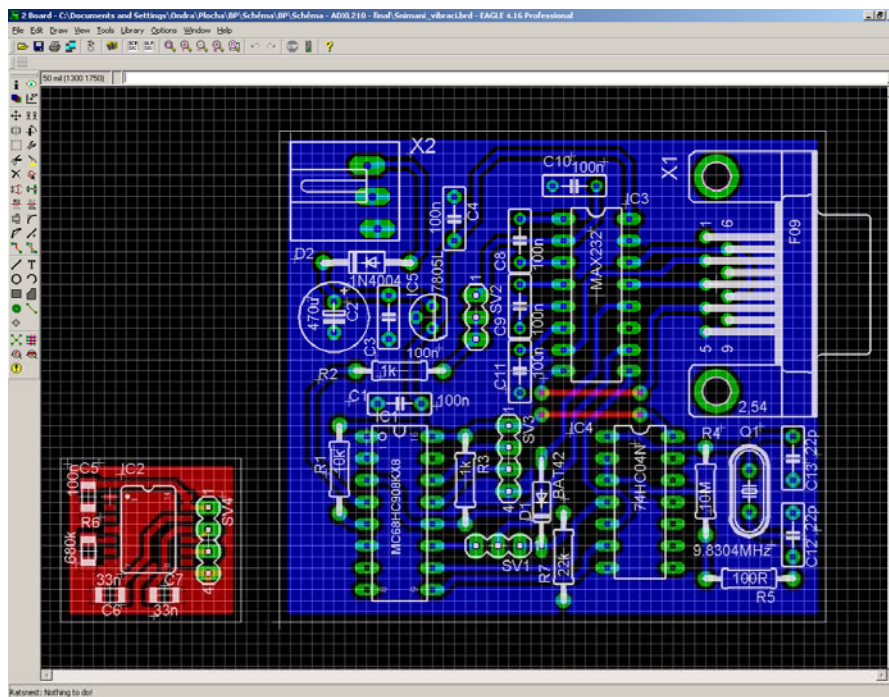
- **Editor schémat** – umožňuje navrhovat elektrotechnická schémata. Může obsahovat až 99 listů jednoho schématu, kontroluje elektrické pravidla zapojení obvodu a umožňuje snadné vytvoření desky plošných spojů ze schématu jediným příkazem.
- **Editor spojů** - umožňuje navrhovat desky plošných spojů (DPS). Obsahuje sadu knihoven součástek a lze také vytvářet vlastní knihovny. Je možné používat klasické i SMD součástky. Obsahuje funkce pro kontrolu pravidel návrhu.
- **Autorouter** – modul, který automaticky rozmístí součástky na desce i jejich propojovací cesty. Pro dosažení lepších výsledků autorouteru je potřeba nastavit pravidla pro propojování.

### 3.3.1 Základní popis prostředí

Základní nástroje pro tvorbu schémata nebo desky plošných spojů jsou v nástrojové liště na levé straně. Místo těchto ikon lze používat také příkazy, které se zadávají do editačního políčka v horní části prostředí. Další informace lze nalézt v [11].



Obr. 23. EAGLE – Editor schémat.

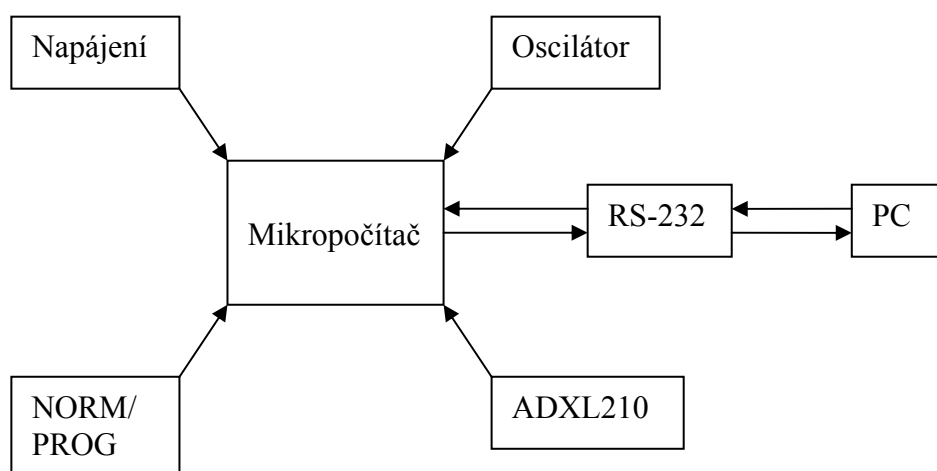


Obr. 24. EAGLE – Editor spojů.

## II. PRAKTICKÁ ČÁST

## 4 OBVOD PRO MĚŘENÍ VIBRACÍ

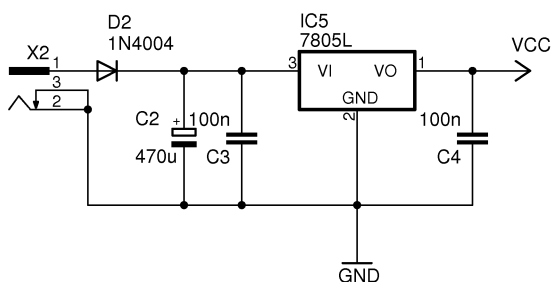
Obvod je navržen tak, aby dokázal zpracovat data z akcelerometru ADXL210 a posílat je do počítače pomocí rozhraní RS-232, proto je vybaven mikropočítačem (mikrokontrolérem) 68HC908KX8. Zařízení také umožňuje naprogramování mikropočítače přímo v obvodu, proto je nutný vnější zdroj hodinového kmitočtu (Oscilátor). K přepínání režimů programování mikropočítače a komunikace s PC slouží konfigurační propojky (NORM/PROG). Celé zařízení je inspirováno vývojovým kitem Janus. Popis kitu Janus (včetně schématu) je dostupný v [8].



Obr. 25. Blokové schéma obvodu.

### 4.1 Napájení

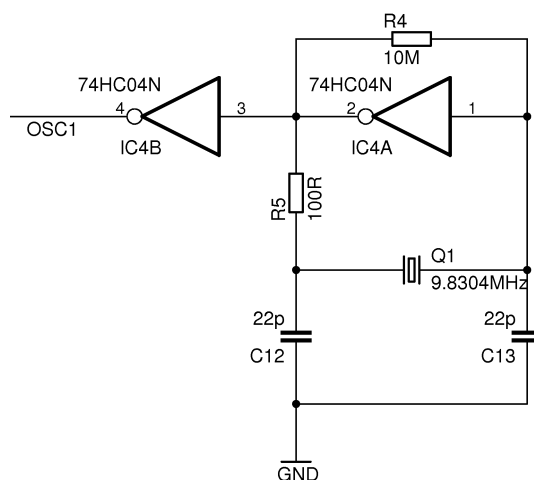
Obvod se napájí usměrněným napětím 9-25 V, maximální spotřeba je 50 mA. Toto napětí je upraveno standardním zdrojem se stabilizátorem 7805 na 5 V a rozvedeno po desce. Proti náhodnému přepólování je obvod chráněn diodou.



Obr. 26. Napájecí část obvodu.

## 4.2 Generátor hodin (ladicí adaptér)

Mikrokontroléry 68HC908KX8 jsou standardně vybaveny vnitřním generátorem hodinové frekvence 307,2 kHz s násobičem frekvence, avšak pro běh v módu Monitoru vyžadují externí generátor hodinového signálu. Obvod proto obsahuje jednoduchý Pierceův krystalový oscilátor s kmitočtem 9,8304MHz v zapojení s invertorem 74HC04. Invertor IC4B slouží k dotvarování signálu. Výstupní signál oscilátoru je připojen na vstup mikropočítače OSC1.



Obr. 27. Generátor hodinového signálu.

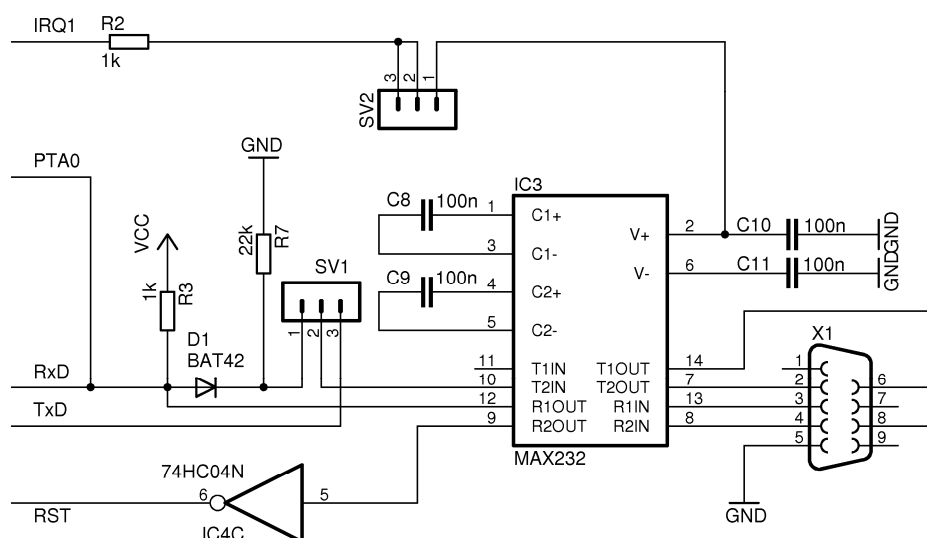
## 4.3 Sériové rozhraní a režim Monitor

Pro komunikaci s mikrokontrolérem s počítačem se využívá sériové rozhraní RS-232 vycházející ze standardního zapojení obvodu MAX232 (IC3) ke kterému je připojen konektor X1 (konektor CAN9) a kapacity C8, C9, C10, C11 pro napěťový měnič.

Mikrokontrolér 68HC908KX8 obsahuje zabudovaný sériový kanál, který umožňuje plně duplexní provoz. Pro komunikaci s počítačem by stačovalo pouze přímé propojení vývodů R1OUT  $\rightarrow$  RxD a TxD  $\rightarrow$  T2IN. Protože je obvod MAX232 sdílen pro komunikaci v režimu Monitor je pro přepínání těchto režimů použita propojka SV1. Na T2IN je také přes diodu D1 připojen vývod PTA0, který slouží ke komunikaci v režimu Monitor. Katoda diody D1 je přes rezistor R7 připojena k zemi z toho důvodu, že nízká úroveň napětí (log. 0) neprojde přes diodu D1 a T2IN by zůstal „viset ve vzduchu“ tzn., že by byl jakoby nepřipojen a hodnota napětí na T2IN by byla nahodilá. Ze spojení s PC se dále využívá signál DTR pro resetování mikrokontroléru z počítače. Aktivací DTR

(otevřením komunikační linky) se RESET mikrokontroléru uvolňuje, deaktivací DTR se mikrokontrolér RESETuje. Signál je veden přes invertor IC4C na RST mikrokontroléru. V době ladění je na signálu DTR log. 0 (užívá se negativní logiky).

Obvodu IC3 zároveň slouží jako zdroj napětí 10 V. Z tohoto napětí se pak vytváří  $V_{TST}$  – zvýšené napětí sloužící jako jeden ze signálů pro přepnutí mikrokontroléru do režimu Monitor. Protože je nutné, aby toto zvýšené napětí bylo k dispozici co nejdříve je to možné, tj. prakticky s napájením, je nezbytné, aby náběh zdvojovače v IC3 byl co nejrychlejší. Proto je doporučeno zapojení RS-232 budiče se 100 nF kondenzátory. Při klasickém zapojení s kondenzátory 1  $\mu$ F nabíhá budič příliš pomalu a napětí  $V_{TST}$  nemusí být na mikrokontrolér přivedeno včas.

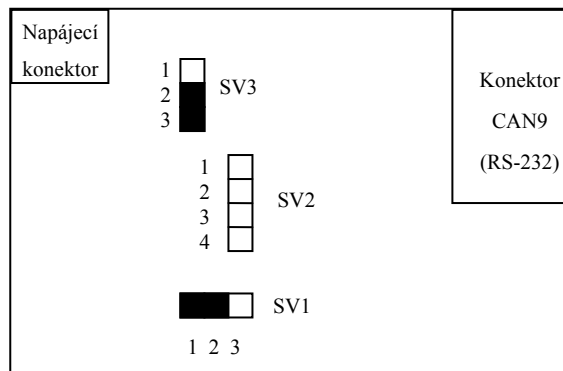


Obr. 28. Sériové rozhraní RS-232.

#### 4.4 Konfigurační propojky

Pomocí propojek (jumper) lze přepínat pracovní módy mikrokontroléru. Normální pracovní režim (komunikace mikropočítače s PC) lze nastavit propojkami SV1 (propojením vývodů 1 a 2) a SV3 (propojením vývodů 2 a 3) tak jak je to zobrazeno na Obr. 29. Přehozením propojek SV1 a SV3 do vedlejších pozic se obvod přepne do programovacího (debug) režimu – Monitor. Tento režim umožňuje naprogramovat mikroprocesor přímo v obvodu a také ladění programu.





Obr. 29. Nastavení režimu obvodu pomocí konfiguračních propojek.

#### 4.5 Samotný mikrokontrolér

Srdcem obvodu je mikrokontrolér 68HC908KX8, který řídí činnost všech částí obvodu. Jeho činnost spočívá v komunikaci s akcelerometrem ADXL210, který posílá dva PWM signály, jejichž střída je úměrná zrychlení akcelerometru. U těchto signálů je vypočítána délka jejich impulzů a perioda. Tyto údaje pak na vyžádání posílá mikrokontrolér do počítače přes rozhraní RS-232. Mikrokontrolér je buzen vnějším krystalovým oscilátorem o frekvenci 9,8304 MHz, která je rovna frekvenci CGMXCLK. Frekvence CGMOUT je rovna 4,9152 MHz a sběrníková frekvence je rovna 2,4576 MHz. Zpracovat údaje z akcelerometru zvládne téměř libovolný levný mikrokontrolér řady HC08 i pomalejší starší mikrokontroléry, protože je perioda akcelerometru nastavitelná. Hlavním kritériem výběru bylo zabudované sériové rozhraní – SCI. Proto byl vybrán mikrokontrolér 68HC908KX8 a ne menší a levnější Nitron. Přestože obsluhu sériového rozhraní lze naprogramovat, tak řešení bez použití časovače, který je používán na obsluhu akcelerometru, by bylo velmi neefektivní.

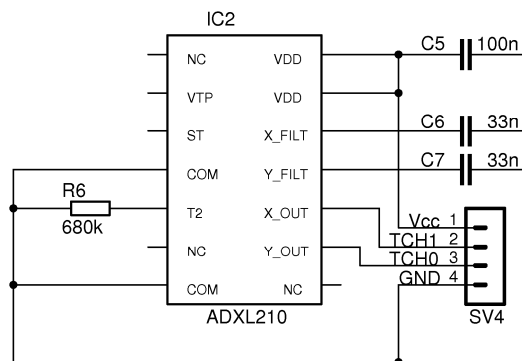
#### 4.6 Obvod pro snímání vibrací

Tato část obvodu je umístěna na samostatné desce plošných spojů z důvodu snadné manipulace se snímačem (naklánění, umístění na vibrující se těleso atd.). Základem je snímač zrychlení (akcelerometr) ADXL210, který umožňuje měřit zrychlení až  $\pm 10$  g ve dvou osách. Deska je spojena s hlavním obvodem čtyřžilovým plochým kabelem, kde 2 vodiče slouží jako napájení a 2 vodiče pro přenos zrychlení pro obě osy pomocí pulzně – šířkové modulace (PWM). Rezistor R6 ( $R_{SET}$ ) slouží pro nastavení periody PWM signálu,

který je poslán na vývodech  $X_{OUT}$  ( $X\_OUT$ ) a  $Y_{OUT}$  ( $Y\_OUT$ ). Perioda lze vypočítat pomocí vztahu (2).

$$T2 = \frac{680}{125} = 5,44 \text{ ms}$$

Kondenzátory  $C6$  ( $C_X$ ) a  $C7$  ( $C_Y$ ) slouží jako filtrační kondenzátory a určují šířku přeneseného pásma s tím, že čím větší je šířka přeneseného pásma, tím větší vzniká šum. Podle Tab. 5 jejich kapacitě 33 n odpovídá šířka pásma necelých 200 Hz. Kondenzátor  $C5$  je blokovací kondenzátor. Konektor  $SV4$  slouží pro připojení k hlavnímu obvodu s mikropočítačem. Kvůli nízkým rozměrům desky byly, s výjimkou konektoru  $SV4$ , použity součástky SMD.



Obr. 30. Obvod pro snímání vibrací.

## 5 MĚŘENÍ A VÝPOČET DÉLKY IMPULZU

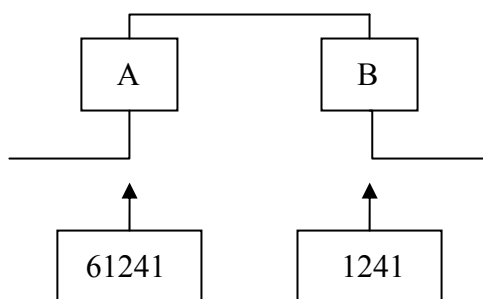
Pro výpočet zrychlení je nutné nejprve změřit délku impulzu a periody signálů z výstupů snímače ADXL210  $X_{OUT}$  a  $Y_{OUT}$ , které jsou připojeny na jednotku časovače mikropočítače (vývody TCH0 a TCH1). K tomu se využívá záchytná jednotka časovače (Input Capture). Tato jednotka detekuje zvolený typ hrany a uloží čas do registru záchytné jednotky časovače TCHx.

### 5.1 Popis měření délky impulzu a periody s využitím přerušení

Jednotku záchyty inicializujeme do režimu záchyty náběžné hrany a povolíme přerušení od příslušného kanálu. Základní princip spočívá v tom, že při prvním přerušení je detekována náběžná hrana a hodnota hlavního čítače je uložena do pomocné proměnné čas A. V obsluze přerušení je záchytná jednotka nastavena na detekci sestupné hrany. V dalším přerušení je tedy detekována sestupná hrana a hodnota hlavního čítače je uložena do proměnné čas B. Pokud od hodnoty čas B odečteme hodnotu čas A získáme počet cyklů hlavního čítače během trvání impulzu. Pro získání délky impulzu je nutné tento údaj vynásobit převrácenou hodnotou frekvence hlavního čítače.

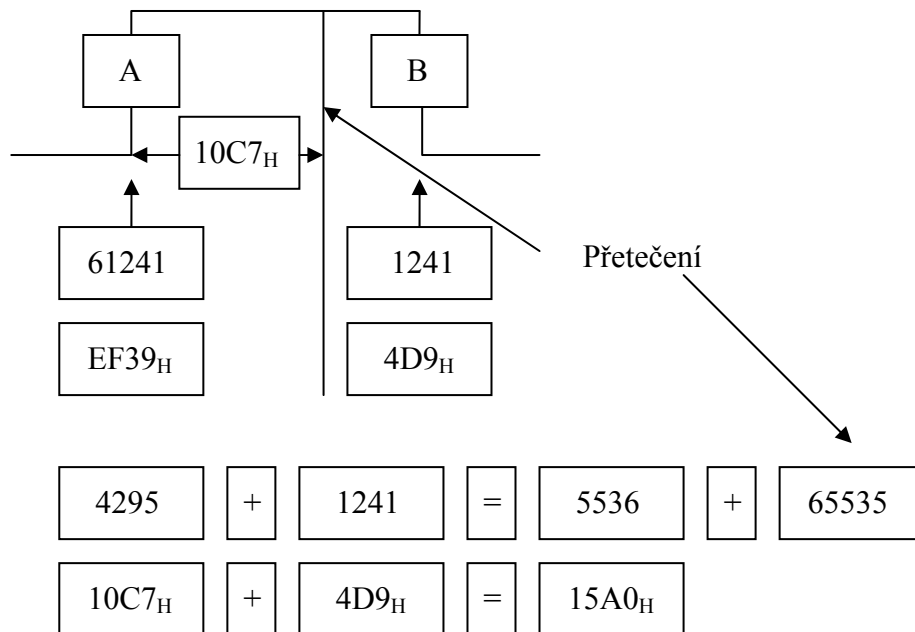
$$D_{IMP} = K(B - A) \quad (4)$$

Tento způsob je problémový v tom, že pokud dojde přetečení hlavního čítače mezi uložením času A a času B, tj. odečítáme větší číslo od menšího, získáme nesmyslné údaje. Pokud bude například čas A roven 61241 a čas B 1241, tak délka impulzu se bude rovnat  $D_{IMP} = K(1241 - 61241) = K(-5536)$  a tedy záporná hodnota. Je zřejmé, že tato hodnota nemůže být záporná, proto je nutné s přetečením počítat.



Obr. 31. Měření impulzu při přetečení časovače.

Jiné řešení je založeno na použití vícebajtové aritmetiky s tím, že současně zjišťujeme, zda došlo k přetečení. Podle potřeby pak budeme toto přerušení ukládat do nejvyššího bajtu hodnot použitých pro výpočet. Lepším způsobem je řešení výpočtu, které je přímo založeno na přetečení, které v minulých případech vadilo.



Obr. 32. Měření impulsu při přetečení časovače.

Postup řešení vychází z reálného chování hlavního čítače. Nejprve určíme počet cyklů hlavního čítače zbývajících do přetečení. Pro správnost výpočtu je nutné používat neznaménkové operace podle vztahu:

$$Z_B = 0 - A \quad (5)$$

kde  $Z_B$  je hodnota udávající zbytek do přetečení,  $A$  je hodnota hlavního čítače při zachycení náběžné hrany. Díky použití neznaménkových operací přeteče šestnáctibitová aritmetika při rozdílu obdobně jako čítač, získáme vlastně jako výsledek správnou šestnáctibitovou kladnou hodnotu. V uvedeném příkladě to bude 4295. K této hodnotě pak přičteme hodnotu hlavního čítače naměřenou v bodě B a máme výsledek udávající délku impulsu v počtu cyklů čítače. Pro převod na čas tuto hodnotu vynásobíme převrácenou hodnotou kmitočtu signálu, který čítá hlavní čítač.

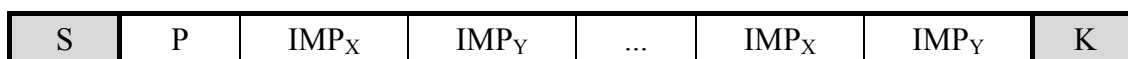
$$D_{IMP} = K(Z_B + B) \quad (6)$$

V Obr. 32 je vidět, že tento výpočet funguje i pokud k přetečení čítače nedošlo. Podrobněji je princip vysvětlen v [6].

## 6 PROGRAM MIKROPOČÍTAČE

### 6.1 Komunikační protokol

Pro komunikaci mezi mikropočítačem a PC byl vytvořen jednoduchý komunikační protokol. Začátek i konec komunikace určuje PC jako nadřazený systém. Začátek komunikace nastane odesláním znaku *s* z PC do mikropočítače. Mikropočítač pak periodicky odesílá do PC blok dat obsahující impuls signálu osy *x* ( $IMP_X$ ) a impuls osy *y* ( $IMP_Y$ ) a v prvním bloku dat také periodu signálu, která je konstantní a stejná pro obě osy, takže není nutné ji odesílat vícekrát. Komunikace se ukončuje odesláním znaku *k* z PC do mikropočítače. Komunikační protokol je znázorněn na Obr. 33, kde jsou znaky, které odesílá PC zvýrazněny šedě.



Obr. 33. Komunikační protokol mezi mikropočítačem a PC.

### 6.2 Popis programu

Nejprve je nutné definovat hlavičkové soubory, které slouží jako odkazy na externí knihovny. K automaticky vygenerovaným je přidán příkazem `#include` soubor `sci_kx8.h`, který zpřístupňuje podprogramy externí knihovny `sci_kx8.asm`, který slouží ke komunikaci mezi mikropočítačem a PC (příp. jiným zařízením) přes sériové rozhraní SCI. Jedná se o upravený ovladač pro mikropočítač MC9S08GB60 dostupný v [13]. Popis práce s ovladačem je dostupný v příloze P II. Následují prototypy funkcí a inicializace proměnných.

Na začátku hlavního programu se volá funkce *Inicializace*, která nastavuje vnější zdroj hodinového kmitočtu procesoru a SCI (2,4576 MHz) v registru CONFIG2.

```
void Inicializace(void) {
    CONFIG2 = 9;          // povolení vnějšího zdroje hodinového kmitočtu
                        //SCI clock = BUS clock = 2,4576 MHz
    ICGCR_ECGON = 1;     // povolíme vnější generator hod. kmitočtu
    while (ICGCR_ECGS!=1) { // počkáme na stabilizaci hodinového
        __RESET_WATCHDOG(); // kmitočtu
    }
    ICGCR_CS = 1; // prepne ICG na externí zdroj hodinového kmitočtu
}
```

Po inicializaci zdroje hodinového kmitočtu mikropočítače se inicializuje sériové rozhraní SCI na rychlost 38400 Bd. Následuje inicializace časovače. Přerušení při přetečení časovače je prozatím zakázáno. Povoleny jsou přerušení obou kanálů v režimu Input Capture pro záchyt náběžné hrany. Perioda je nastavena pomocí modulo registru na 25 ms.

Ve smyčce hlavního programu jsou pouze 2 cykly, které slouží pro spuštění a ukončení měření a komunikace s PC. První cyklus čeká na příchod znaku *s*, což je příkaz pro spuštění měření. Jakmile je tento příkaz přijat, je cyklus ukončen a je povoleno přerušení při přetečení časovače. Zároveň je také nastavena proměnná *perioda\_priznak* na hodnotu 1, což znamená, že se má poslat perioda signálu. Druhý cyklus obdobně jako první cyklus čeká na příchod znaku *k*, což je příkaz pro ukončení měření a je zakázáno přerušení při přetečení časovače.

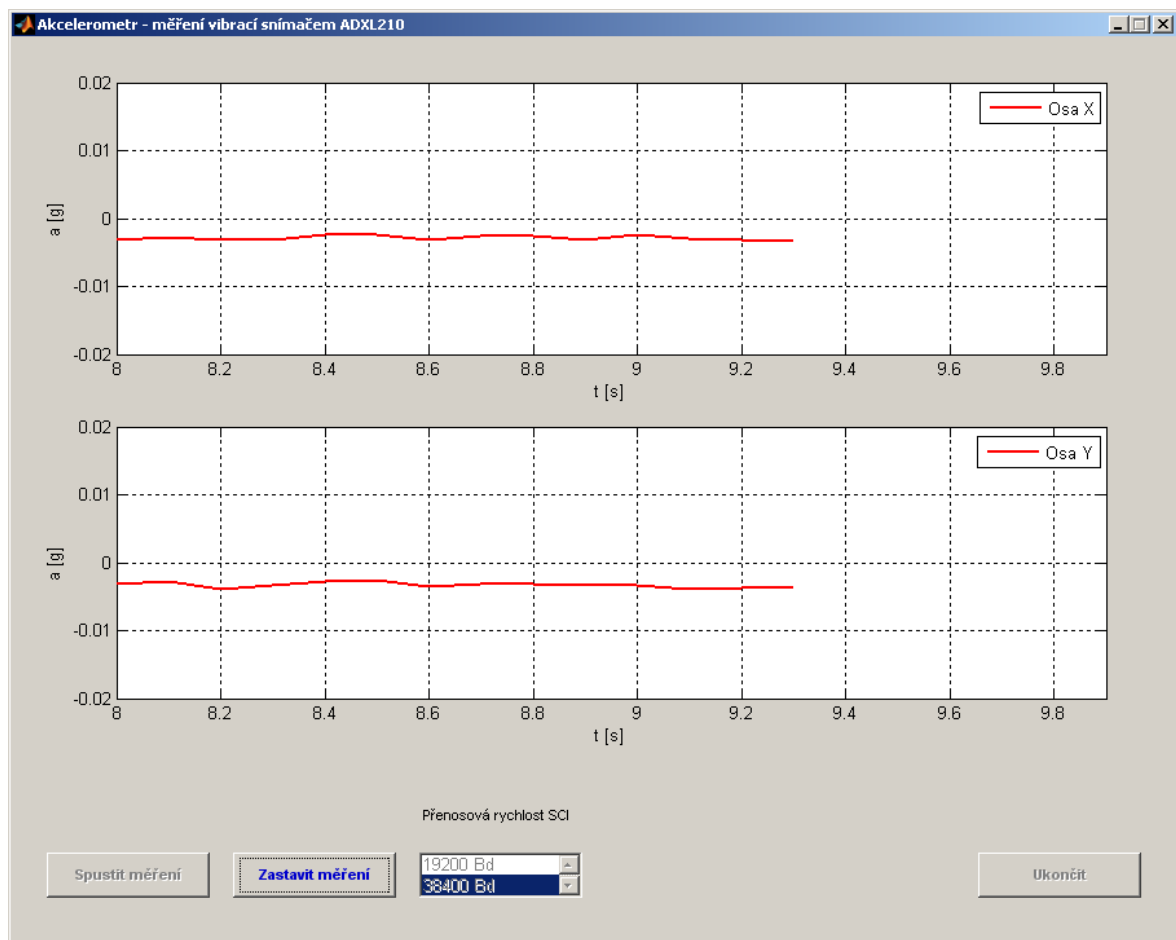
### Měření délky periody a impulzů

Pro změření délky periody a impulzů jsou využity 2 přerušení časovače v režimu Input Capture pro každou osu zvlášť. Režim Input Capture je nastaven na přerušení při detekci náběžné hrany. V prvním přerušení je vypočítána hodnota zbytku do přetečení podle vztahu (5) a uložena do proměnné *zbytek\_y* a nastavení záchytné jednoty (Input Capture) je změněno na detekci sestupné hrany. K rozlišení náběžné a sestupné hrany v přerušení slouží příznak *priznak\_y*. Pokud je *priznak\_y* nulový jedná se o náběžnou hranu jinak o sestupnou hranu. V druhém přerušení je uložena hodnota náměru sestupné hrany do proměnné *cas\_y2* a je vypočítána délka impulzu sečtením proměnných *zbytek\_y* a *cas\_y2* podle vztahu (6). Výsledná hodnota je uložena do proměnné *impulz\_y*. Nastavení záchytné jednoty je opět změněno na detekci náběžné hrany a dále se přerušení provádí stejným způsobem. Přerušení pro výpočet délky impulzu signálu pro osu *x* je řešeno obdobným způsobem jako pro osu *y* s tím rozdílem, že v tomto přerušení je pouze při detekci druhé náběžné hrany vypočítána perioda signálu, proto bylo nutné použít další příznak *priznak\_p*. Perioda je spočítána pouze jednou, protože je konstantní a stejná pro obě osy.

## 7 PROGRAM AKCELEROMETR

### 7.1 Popis prostředí

Program **Akcelerometr** naprogramovaný v MATLABu slouží přijetí dat z mikropočítače (délka periody a impulzu) a výpočet výsledného zrychlení a to z toho důvodu, aby mikropočítač nemusel vykonávat operace dělení s reálnými čísly. Program obsahuje 3 tlačítka a jeho ovládání je velmi jednoduché. Prvním tlačítkem „Spustit měření“ zahájíme přenos dat mezi mikropočítačem a PC, vypočítané zrychlení je pak průběžně vykreslováno do grafu a zároveň ukládáno do souboru *data.csv*. Měření trvá tak dlouho, dokud není ukončeno tlačítkem „Zastavit měření“. V seznamu lze nastavit přenosovou rychlost, která je standardně nastavena na hodnotu 38400 Bd. Nastavovat přenosovou rychlost je možné, pouze pokud není měření spuštěno.



Obr. 34. Program Akcelerometr pro výpočet a vizualizaci zrychlení.

## 7.2 Popis programu Akcelerometr

### Hlavní okno

Základem programu je skript *Akcelerometr.m*, který vytváří grafické uživatelské prostředí (GUI) a definuje funkce, které jsou vyvolány při obsluze událostí (stisknutí tlačítka, výběr přenosové rychlosti ze seznamu atd.). V tomto skriptu jsou také definovány některé globální proměnné – *soubor* a proměnná *abort*, která slouží k ukončení měření. Nedílnou součástí je *serial port objekt*. Jedná se o objekt, který umožňuje jednoduchou a efektivní práci se sériovým rozhraním.

### Serial port objekt

*Serial port objekt* se vytváří příkazem `serial`. Jediným povinným parametrem je název portu, který se má otevřít. Dalšími parametry mohou být vlastnosti objektu, které lze samozřejmě nastavit i později.

```
s = serial('nazev_portu', 'vlastnost', 'hodnota')
s = serial('COM1', 'BaudRate', '38400')
```

V tomto případě bude otevřen port COM1 a přenosová rychlost bude nastavena 38400 Bd.

Další vlastnosti objektu lze nastavovat a měnit pomocí příkazu `set`.

```
set(objekt, 'vlastnost', 'hodnota')
set(s, 'Terminator', 'CR/LF')
```

V tomto případě bude `Terminator` nastaven na hodnotu CR/LF (odřádkovací sekvence – návrat vozíku a posun na další řádek).

Nejdůležitějšími vlastnostmi *serial port objektu* jsou (v závorce jsou i konkrétní nastavené hodnoty):

```
Status (open|closed)
Timeout (maximální povolená doba trvání operace - 3 s)
BaudRate (přenosová rychlost v baudech - 38400 Bd)
Port (port - COM1)
Terminator (ukončovací znak - CR/LF)
```

Hodnoty vlastností objektu lze získat pomocí příkazu `get`.

```
get(objekt)
get(objekt, 'vlastnost', 'hodnota')
get(objekt, 'Status')
```

Tento příklad vrací stav objektu – (open|closed).

Pokud není v parametru uvedena žádná vlastnost, budou vypsané hodnoty všech vlastností.

Před použitím je nutné vytvořený *serial port objekt* otevřít příkazem `fopen`.

```
fopen(objekt)
```



```
fopen(s)
```

Tento příklad otevře *serial port objekt* – *s*.

Posílat data na port lze příkazy `fwrite` (pro binární data) a `fprintf` (pro textová data)

```
fwrite(objekt, binarni_data)
fprintf(objekt, 'retezec')
fprintf(objekt, 's')
```

Tento příklad pošle na port znak *s*.

Číst data přijatá na portu lze příkazy `fread` (pro binární data), `fscanf` (pro textová data) a `fgets` (pro čtení textového řádku).

```
[data, pocet, zprava] = fread(objekt)
[data, pocet, zprava] = fgets(objekt)
[data, pocet, zprava] = fscanf(objekt)
[rxdata, count, msg] = fscanf(s)
```

Tento příklad načte přijatá data do proměnné *rxdata*, počet přijatých znaků do proměnné *count* a případné chybové hlášení je uloženo do proměnné *msg*. Pokud byla data přečtena v pořádku je v *msg* prázdný řetězec.

Před ukončením programu je nutné port uzavřít a smazat *serial port objekt*. K tomu slouží příkazy `fclose` a `delete`.

```
fclose(objekt)
delete(objekt)
fclose(s)
delete(s)
```

Tento příklad provede uzavření portu a smaže *serial port objekt*.

### Měření a komunikace

Měření se spouští tlačítkem „Spustit měření“. Vyvolaná událost spouští funkci `fce_disable_buttons`, funkci `fce_start_mereni` a nastavuje proměnnou `abort = 0`.

#### **Funkce `fce_enable_buttons(bStart, bStop, bUkoncit, lRychlost)`**

Tato funkce zakáže komponenty uvedené v parametrech (tj. tlačítka a seznam), aby je nebylo možné používat, protože je nežádoucí, aby uživatel měnil přenosovou rychlost, ukončoval program nebo se snažil znovu spustit měření, pokud je měření spuštěno.

**Funkce** `fce_start_mereni(s, aGraf_x, aGraf_y, bStart, bStop, bUkoncit, lRychlost, fpath)`

Nejprve je zjištěn stav komunikačního portu a poté je otevřen (pokud nebyl otevřen již předtím). Dále jsou inicializovány proměnné. Časový rozsah grafu je nastaven na 2 s, perioda posílání dat do PC je 25 ms a počet kroků je 80. Tolikrát posílá mikropočítač data do PC. Pomocí těchto údajů je inicializována proměnná *time* a vektory zrychlení *ax* a *ay*. Perioda vykreslování dat do grafu je z důvodu vyššího zatížení procesoru snížena na 100 ms. Proto jsou definovány proměnné *delitel*, *axgraf*, *aygraf* a *timegraf*. Do souboru jsou ukládány všechny hodnoty.

Příkaz `fprintf(s, 's')` dává pokyn mikropočítači a tím zahajuje měření. Mikropočítač začne posílat data, nejprve jednou periodu signálu, která je konstantní a pak proměnlivé impulzy. Pokud nejsou žádná data přijata, do proměnné *msg* je uloženo chybové hlášení, nejpravděpodobněji vypršení maximální povolené doby operace („A timeout occurred before the Terminator was reached“) a to nejčastěji z důvodu chybně nastavené přenosové rychlosti nebo pokud není zařízení připojeno. Pokud však zařízení není připojeno, pak nastane při příkazu `fprintf(s, 's')` echo tzn., že je přečten znak *s* (+ Terminator CR/LF), který byl poslán. Vzhledem k tomu, že data byla přečtena a nebyla tedy zjištěna žádná chyba, je do proměnné *msg* uložen prázdný řetězec. Chyba lze poznat tak, že byly přijaty pouze 3 znaky (*s* + CR/LF). V těchto případech je vyvoláno chybové hlášení a měření není spuštěno. Pokud je vše v pořádku je přijat první údaj – perioda signálu, která je uvozena znakem *p*. Znak *p* je odstraněn, zbytek řetězce je převeden na číslo a uložen do proměnné *perioda*.

Příkaz `ph_x=plot(aGraf_x,0)` vytvoří tzv. handle na objekt *aGraf\_x*, který je potřeba k tomu, aby bylo možné graf měnit příkazem `set` (nastavení dat, legenda, popisy os, atd.). Stejný handle je vytvořen i pro objekt *aGraf\_y*.

Samotné měření a komunikace s mikropočítačem probíhá v nekonečném cyklu, který je možné přerušit stisknutím tlačítka „Zastavit měření“. Uvnitř běží další cyklus, který běží vždy přibližně 2 s a po tuto dobu je přijato 80 impulzů každé osy. Impulzy jsou tedy posílány každých 25 ms. Samotné impulzy jsou načítány příkazem `fscanf(s)`. Každý impulz je uvozen znakem *x* (pro osu *x*) a *y* (pro osu *y*). Tyto znaky je nutné odstranit, zbytky řetězce převést na čísla a uložit do vektorů *impulz\_x* a *impulz\_y*. Konečný výpočet zrychlení pro osu *x* se pak provede příkazem  $ax(i+1) = (impulz_x(i+1)/perioda -$

0.5)/4 a obdobným příkazem pro osu  $y$ . Vykreslování dat do grafů probíhá tak, že vektory  $ax$  a  $ay$  se postupně zvětšují o nová data, a proto jsou data vykreslována postupně. Každý čtvrtý prvek vektorů  $ax$ ,  $ay$  a  $time$  je překopírován do vektorů  $axgraf$ ,  $aygraf$  a  $timegraf$ . Tímto se zajistí změna periody vykreslování do grafu z 25 ms na 100 ms. Pokud by byly vykreslovány celé vektory, pak byla vykreslena předchozí data a novými by byla postupně překreslována. Jiným řešením by bylo vykreslování až po naplnění vektorů, tedy každé dvě sekundy. Takové vykreslování zatěžuje procesor počítače podstatně méně, ale i při nižším intervalu je nepřirozené. Změna hodnot v grafu se provede příkazem `set` v každém cyklu a okamžité vykreslení se zajistí příkazem `drawnow`.

### Výpis do souboru

Výpis dat se provádí do textového souboru s příponou CSV, ve kterém jsou data oddělena středníkem. Tento typ souboru lze otevřít i v MS Excelu nebo do Excelu jednoduše importovat. Příkazem `fpath = 'data.csv'` je definován soubor do které se budou data zapisovat. Soubor lze otevřít příkazem `fopen`.

```
soubor = fopen(fpath)
soubor = fopen(fpath, 'opraveni')
soubor = fopen(fpath, 'w')
```

Tento příklad otevře soubor pro zápis (původní soubor bude přepsán) nebo vytvoří nový. Bez nastavení oprávnění nebo s parametrem oprávnění  $r$  se otevře soubor jen pro čtení. Parametr oprávnění  $a$  slouží k otevření souboru pro připsání dat na konec souboru. Zápis do souboru se provádí příkazem `fprintf`.

```
fprintf(soubor, format, data)
fprintf(soubor, '%s; %s\n', 'Datum:', date)
```

Tento příklad zapíše do souboru „Datum:;“ a aktuální datum ve tvaru DD-MMM-YYYY (např. 20-Apr-2009).

Čtení ze souboru se provede příkazem `fscanf`.

```
fscanf(soubor, format)
```

Před ukončení programu je nutné soubor uzavřít příkazem `fclose`.

```
fclose(soubor)
```

Zápis do souboru probíhá tak, že se nejprve otevře soubor pro zápis a do souboru se zapíše aktuální datum a čas a názvy sloupců. Soubor se pak znovu otevře pro připsání dat. Data se zapisují průběžně v každém cyklu. Soubor je uzavřen při ukončení programu.

Měření se ukončuje tlačítkem „Zastavit měření“. Vyvolaná událost spouští funkci *fce\_enable\_buttons*, funkci *fce\_konec\_mereni* a nastavuje proměnnou *abort = 0*.

**Funkce *fce\_enable\_buttons(bStart, bStop, bUkoncit, lRychlost)***

Tato funkce povolí prvky zakázané funkcí *fce\_disable\_buttons*.

**Funkce *fce\_konec\_mereni(s)***

Příkaz `fprintf(s, 'k')` dává pokyn mikropočítači k ukončení měření a komunikace.

Následuje uzavření *serial port objektu*.

**Funkce *fce\_listbox(s, lRychlost)***

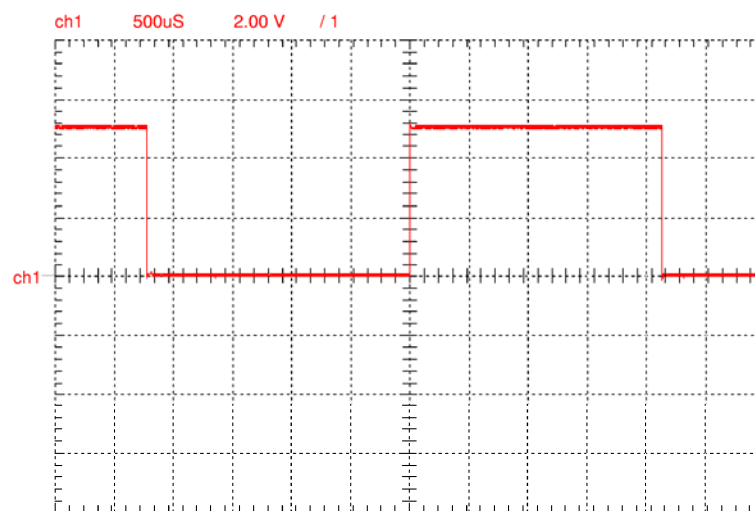
Nastavuje přenosovou rychlost *serial port objektu* vybranou ze seznamu.

**Funkce *fce\_konec(s)***

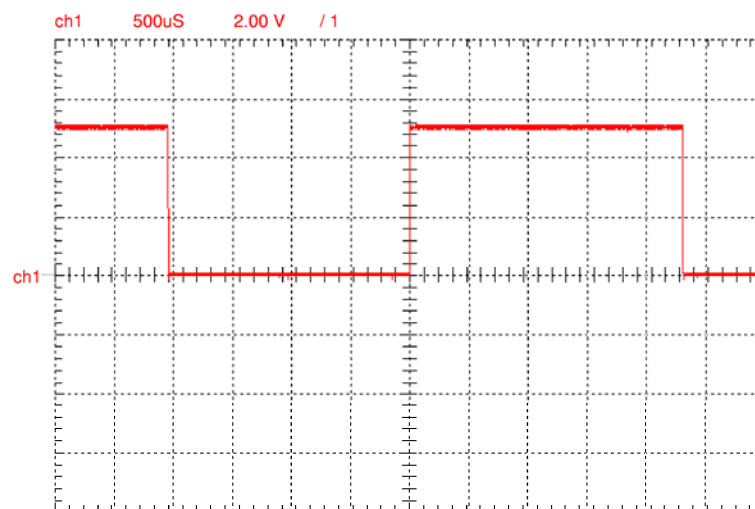
Pokud je program ukončován křížkem v průběhu měření je vyvolán dialog, jestli má být program ukončen nebo ne. Pokud ano, zavolá se funkce *fce\_konec\_mereni*, zavře se soubor a je smazán *serial port objekt*. Příkaz `closereq` ukončí program.

## 8 OVĚŘENÍ ÚDAJŮ

Na osciloskopu byly pro ověření správné funkce změřeny průběhy signálů získané z obou os akcelerometru. Tyto údaje lze také spočítat z hodnot naměřených mikropočítačem. Na Obr. 35 je vidět průběh signálu bez natočení akcelerometru a na Obr. 36 je průběh při natočení akcelerometru o  $90^\circ$  v ose x. Z grafů lze vyčíst, že hodnota periody se rovná přibližně 4,45 ms. Délka impulzu je v prvním případě 2,2 ms a ve druhém 2,3 ms. Hodnoty výsledného zrychlení jsou tedy po výpočtu podle vztahu (3)  $a_x = a_y = -0,0014 \text{ g}$  pro první případ a  $a_x = 0,0169 \text{ g}$  pro druhý případ.

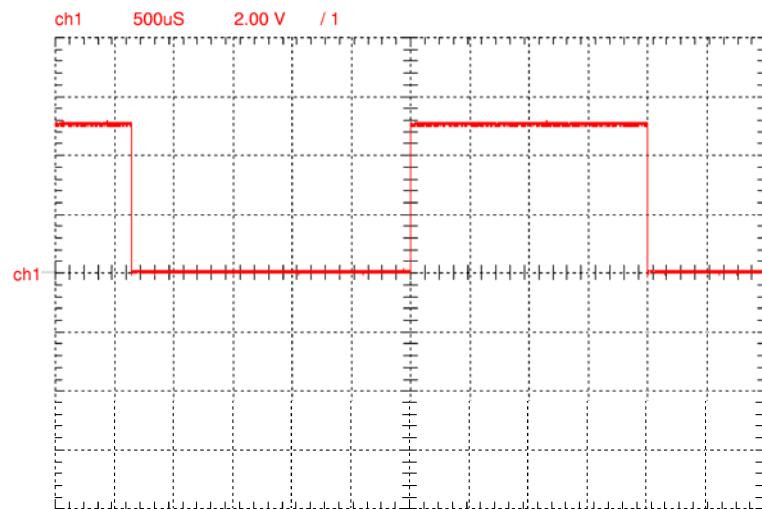


Obr. 35. Signál z akcelerometru bez natočení.



Obr. 36. Signál z akcelerometru pro natočení  $90^\circ$  v ose x.

Na Obr. 37 je průběh signálu při natočení akcelerometru o  $90^\circ$  v ose  $y$ . Perioda je přibližně 4,35 ms a délka impulzu 2 ms. Po výpočtu získáme hodnotu zrychlení  $a_y = -0,0101$  g. Tato hodnota je přibližně rovna hodnotě při natočení o  $90^\circ$  v ose  $x$  s tím, že akcelerometr byl natočen v opačném směru, proto je hodnota záporná. Všechny naměřené výsledky jsou pouze orientační, protože nebyl použit žádný nástroj, který by zajistil přesné natočení o  $90^\circ$  a navíc byl obvod přidržován rukou, což vedlo k přenosu vibrací a ovlivnění přesnosti měření.



Obr. 37. Signál z akcelerometru pro natočení  $90^\circ$  v ose  $y$ .

## ZÁVĚR

Cílem této práce bylo navrhnout a vytvořit jednoduché elektronické zařízení, které by dokázalo vyhodnocovat signály přijaté z akcelerometru – snímače zrychlení a následně je posílat do počítače. Dalším požadavkem byla schopnost naprogramovat mikropočítač přímo v desce zařízení, aby jej nebylo nutné stále přemísťovat do vývojového kitu. Na straně PC pak měl vzniknout program, který dokáže vizualizovat výsledné zrychlení.

Výsledkem je zařízení, jehož „srdcem“ je mikropočítač od firmy Freescale 68HC908KX8, který byl volen jako levný zástupce mikropočítačů řady HC08 se zabudovaným sériovým rozhraním. Druhým základním kamenem je akcelerometr ADXL210, který je umístěn na samostatné desce pro jeho snadnou manipulaci.

Pro vizualizaci zrychlení byl vytvořen program *Akcelerometr* naprogramovaný v prostředí MATLAB, které nabízí přímou podporu pro komunikaci přes sériové rozhraní a grafickou vizualizaci dat. Program *Akcelerometr* umožňuje v reálném čase vykreslovat průběh zrychlení akcelerometru a zároveň ukládat do souboru. Kvůli vysokým nárokům na procesor však nejsou do grafu vykreslována všechna data.

Toto zařízení má nepochybně své využití ať už třeba v průmyslu při měření vibrací, náklonu, jako detektor pohybu v zabezpečovacích systémech nebo jako výukový přípravek.

Výhodou desky s akcelerometrem jsou díky použitým SMD součástkám velmi malé rozměry. Do jisté míry je to i nevýhoda, protože tyto součástky jsou určeny zejména pro automatizovanou montáž. Ruční montáží může vzniknout méně kvalitní deska, což se případně projeví zašuměním při vyšších frekvencích posílaných dat.

## ZÁVĚR V ANGLIČTINĚ

The aim of this work was to design and construct a simple electronic device, capable to evaluate signals received from the accelerometer – the acceleration sensor and then send it to the computer. Another requirement was the ability to program a microcontroller inside the evaluation unit without the necessity of transferring it into a development kit. A new program should be developed for the PC that can visualize the measured acceleration.

The result is a device whose "heart" is a 68HC908KX8 microcontroller from Freescale, which was selected as a representative of inexpensive microcontrollers HC08 series with built-in serial interface. The second cornerstone is the ADXL210 accelerometer, which is located on a separate board for easy handling.

A program developed for the visualization of acceleration called *Akcelerometr* was programmed in MATLAB environment, which offers direct support for communication via the serial interface and graphical visualization of data.

The *Akcelerometr* program allows real-time plotting of the course of acceleration of the accelerometer and simultaneously saving to the file. Nevertheless, due to high demands on the processor all the data are not plotted to the chart.

This device has wide utilization in the industry in the measurement of vibrations and tilt, as a motion detector in the security systems or as an educational aid.

The advantage of the board with the accelerometer is compact dimensions due to SMD usage. But this is also the disadvantage, because these devices are especially designed for automated assembly. Manual assembly without high-quality equipment and the necessary skills can result in a lower quality of board. A noise may occur at higher frequencies as well.



**SEZNAM POUŽITÉ LITERATURY**

- [1] Freescale Semiconductor. *MC68HC908KX8 Data Sheet*, Rev. 2.1, 2005. Dostupný z WWW: <[www.freescale.com](http://www.freescale.com)>.
- [2] Freescale Semiconductor. *HCS08 Family Reference Manual*, Rev. 1., 2003. Dostupný z WWW: <[www.freescale.com](http://www.freescale.com)>
- [3] Analog Devices. *Low-Cost  $\pm 10$  g Dual-Axis Accelerometer with Duty Cycle ADXL210E*, 1999. Dostupný z WWW: <[www.analog.com](http://www.analog.com)>.
- [4] VÁŇA, Vladimír. *Začínáme s mikrokontroléry Motorola HC08 Nitron*. Praha: BEN – technická literatura, 2003. 96 s. ISBN 80-7300-124-1.
- [5] PERŮTKA, Karel. *MATLAB – Základy pro studenty automatizace a informačních technologií*, 2005. 303 s. ISBN 80-7318-355-2.
- [6] ROZEHNAL, Zdeněk. *Mikrokontroléry Motorola HC11*. 1. vyd. Praha: BEN – technická literatura, 2001. 192 s. ISBN 80-86056-77-5.
- [7] VOJÁČEK, Antonín. *Akcelerometry - integrované snímače od AD*. [cit. 2009-05-01]. Dostupný z WWW: <<http://automatizace.hw.cz/mereni-a-regulace/ART143-akcelerometry--integrovane-snimace-od-ad.html>>.
- [8] *Vývojový kit JANUS konstrukční manuál*. [cit. 2009-05-01]. Motorola, 2003. Dostupný z WWW: <[http://measure.feld.cvut.cz/groups/micro/HC08/soubory/qt4/janus-konstrukcni\\_manual.pdf](http://measure.feld.cvut.cz/groups/micro/HC08/soubory/qt4/janus-konstrukcni_manual.pdf)>.
- [9] DOLÍVKA, Lukáš. *Semestrální práce do předmětu Speciální číslicové systémy: Motorola HC08*. [cit. 2009-05-01]. Praha: ČVUT, 2003. Dostupná z WWW: <<http://noel.feld.cvut.cz/vyu/scs/prezentace2004/Motorola-HC08/hc08.pdf>>.
- [10] MOTOROLA. *Motorola usnadňuje vývoj 8bitových mikrokontrolerů*. [cit. 2009-05-01]. Dostupný z WWW: <<http://pr.motorola.cz/index.php?key=article&id=296>>.
- [11] Eagle Online. *Informace o programu EAGLE*. [cit. 2009-05-01]. Dostupný z WWW: <<http://www.elcad.cz/eagle/info.htm>>.
- [12] HUMUSOFT. *MATLAB*. [cit. 2009-05-01]. Dostupný z WWW: <<http://www.humusoft.cz/produkty/matlab/matlab/index.php?lang=cz&p1=1&p2=1&p3=1>>.

- [13] UTB, FAI. *Soubory ovladače sériového rozhraní (SCI)*, Dostupné z WWW:  
<[www.vyuka.fai.utb.cz](http://www.vyuka.fai.utb.cz)>, kurz Mikropočítače.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

A/D	Analogově/Digitální
BCD	Binárně dekadický kód
BD	Dělička přenosové rychlost SCI
COP	Hlídací obvod Watchdog
CCR	Příznakový registr
CGMXCLK	Výstupní frekvence ICG
CGMOUT	Výstupní frekvence ICG, polovina CGMXCLK
CPU	Centrální procesorová jednotka
CR/LF	Návrat vozíku/Posun na další řádek
FLASH	FLASH paměť
GUI	Grafické uživatelské prostředí
ICG	Generátor hodinového signálu
IRQ	Externí přerušení
KBI	Přerušení od klávesnice
LVI	Modul hlídání napájení
NRZ	NRZ kódování (Non Return to Zero)
PC	Osobní počítač
PC	Programový čítač
PD	Předdělička
POR	Reset vyvolaný při náběhu napájení
PWM	Pulzně – šířková modulace
RAM	Paměť s libovolným přístupem
ROM	Paměť pouze pro čtení
SCI	Sériové komunikační rozhraní

SIM	Systemový modul
SMD	Součástky pro povrchovou montáž
SP	Ukazatel zásobníku
TIM	Modul časovače

**SEZNAM OBRÁZKŮ**

Obr. 1. Registry procesoru HC08. ....	11
Obr. 2. Blokové schéma mikropočítače MC68HC908KX8. ....	13
Obr. 3. Rozložení vývodů mikropočítače MC68HC908KX8. ....	13
Obr. 4. Paměťová mapa mikropočítače MC68HC908KX8. ....	15
Obr. 5. Konfigurační registr CONFIG1. ....	16
Obr. 6. Konfigurační registr CONFIG2. ....	17
Obr. 7. Port A Data Register (PTA). ....	20
Obr. 8. Data Direction Register A (DDRA). ....	20
Obr. 9. Port A Input Pullup Enable Register (PTAPUE). ....	20
Obr. 10. Port B Data Register (PTB). ....	21
Obr. 11. Standardní NRZ formát datového rámce. ....	22
Obr. 12. SCI Baud Rate Register (SCBR). ....	22
Obr. 13. Systémové hodinové signály. ....	24
Obr. 14. Zapojení, při kterém je mikropočítač v režimu Monitor. ....	26
Obr. 15. Struktura senzoru bez působení zrychlení (0 g). ....	28
Obr. 16. Struktura senzoru při působení zrychlení. ....	28
Obr. 17. Blokové schéma akcelerometru ADXL210. ....	30
Obr. 18. Rozložení vývodů akcelerometru ADXL210. ....	30
Obr. 19. Výstupní signál akcelerometru ADXL202 (12,5% platí pro ADXL202, pro 210 – 4%). ....	31
Obr. 20. Parametry výstupního signálu při otáčení senzorem ADXL202/ADXL210. ....	32
Obr. 21. Prostředí MATLAB. ....	34
Obr. 22. Integrované vývojové prostředí CodeWarrior. ....	35
Obr. 23. EAGLE – Editor schémat. ....	36
Obr. 24. EAGLE – Editor spojů. ....	36
Obr. 25. Blokové schéma obvodu. ....	38
Obr. 26. Napájecí část obvodu. ....	38
Obr. 27. Generátor hodinového signálu. ....	39
Obr. 28. Sériové rozhraní RS-232. ....	40
Obr. 29. Nastavení režimu obvodu pomocí konfiguračních propojek. ....	41
Obr. 30. Obvod pro snímání vibrací. ....	42

---

Obr. 31. Měření impulzu při přetečení časovače.....	43
Obr. 32. Měření impulzu při přetečení časovače.....	44
Obr. 33. Komunikační protokol mezi mikropočítačem a PC. ....	45
Obr. 34. Program Akcelerometr pro výpočet a vizualizaci zrychlení. ....	47
Obr. 35. Signál z akcelerometru bez natočení. ....	53
Obr. 36. Signál z akcelerometru pro natočení 90° v ose x. ....	53
Obr. 37. Signál z akcelerometru pro natočení 90° v ose y. ....	54

**SEZNAM TABULEK**

Tab. 1. Nastavení zdroje hodinového signálu.....	17
Tab. 2. Hodnoty nastavení modulační rychlosti.....	23
Tab. 3. Nastavení pro vstup do režimu Monitor.....	26
Tab. 4. Popis jednotlivých vývodů akcelerometru ADXL210.....	31
Tab. 5. Nastavení šířky frekvenčního pásma včetně typických úrovní šumu.....	32

## **SEZNAM PŘÍLOH**

PŘÍLOHA P I: REGISTRY ČASOVAČE TIM

PŘÍLOHA P II: PRÁCE S SCI OVLADAČEM

PŘÍLOHA P III: SCHÉMA ZAŘÍZENÍ

PŘÍLOHA P IV: DESKY PLOŠNÝCH SPOJŮ

PŘÍLOHA P V: ELEKTRONICKÉ PŘÍLOHY NA CD




## PŘÍLOHA P I: REGISTRY ČASOVAČE TIM

### TIM Status and Control Register (TSC)

Address: \$0020

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0

 = Unimplemented

**TOF** (TIM Overflow Flag Bit) – příznak přetečení časovače.

0 = časovač nedosáhl hodnoty modulo registru

1 = časovač dosáhl hodnoty modulo registru

**TOIE** (TIM Overflow Interrupt Enable Bit) – povolení přerušení při přetečení časovače.

0 = přerušení zakázáno

1 = přerušení povoleno

**TSTOP** (TIM Stop Bit) – zastavení časovače.

0 = časovač je zastaven

1 = časovač je aktivní

**TRST** (TIM Reset Bit) – reset časovače a předděličky.

0 = žádná změna

1 = časovač a předdělička jsou resetovány

**PS2–PS0** (Prescaler Select Bits) – nastavení předděličky.

PS2 – PS0	Hodnota dělení
000	1
001	2
010	4
011	8
100	16
101	32
110	64

**TCNTH a TCNTL** (TIM Counter Registers) – 16-bitový registr, složený ze dvou 8-bitových, uchovává obsah čítače TIM. Tento registr slouží pouze ke čtení.


**TMODH a TMODL** (TIM Counter Modulo Registers) – obsahuje modulo hodnotu čítače TIM. Pokud čítač dosáhne této hodnoty nastane přerušení, pokud je povoleno.

**TSC0 a TSC1** (TIM Channel Status and Control Registers)

Register name and address: TSC0 — \$0025								
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CHOIE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Register name and address: TSC1 — \$0028								
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**CHF** (Channel x Flag Bit) – příznak události vyvolané jednotkou Input Capture nebo Output Compare

0 = žádná událost Input Capture nebo Output jednotky

1 = detekována událost Input Capture nebo Output jednotky

**CHxIE** (Channel x Interrupt Enable Bit) – povoluje přerušení od kanálu x.

0 = přerušení od kanálu x zakázáno

1 = přerušení od kanálu x povoleno

**MS0B, MSxA, ELSxB a ELSxA** – nastavení funkce kanálu x (Input Capture, Output Compare, PWM)

**TCHxH a TCHxL** (TIM Channel Registers) – slouží jako datové registry jednotek Input Capture a Output Compare.

## **PŘÍLOHA P II: PRÁCE S SCI OVLADAČEM**

`char sci_init (char)` – provede inicializaci SCI rozhraní na požadovanou přenosovou rychlost. Vrací 1 v případě úspěšné inicializace, jinak 0.

```
stav=sci_init(BD38400);
```

`char sci_in (void)` – vrací přijatý znak z SCI rozhraní. V případě, že nebyl žádný znak přijat, čeká na jeho příjem.

```
znak=sci_in();
```

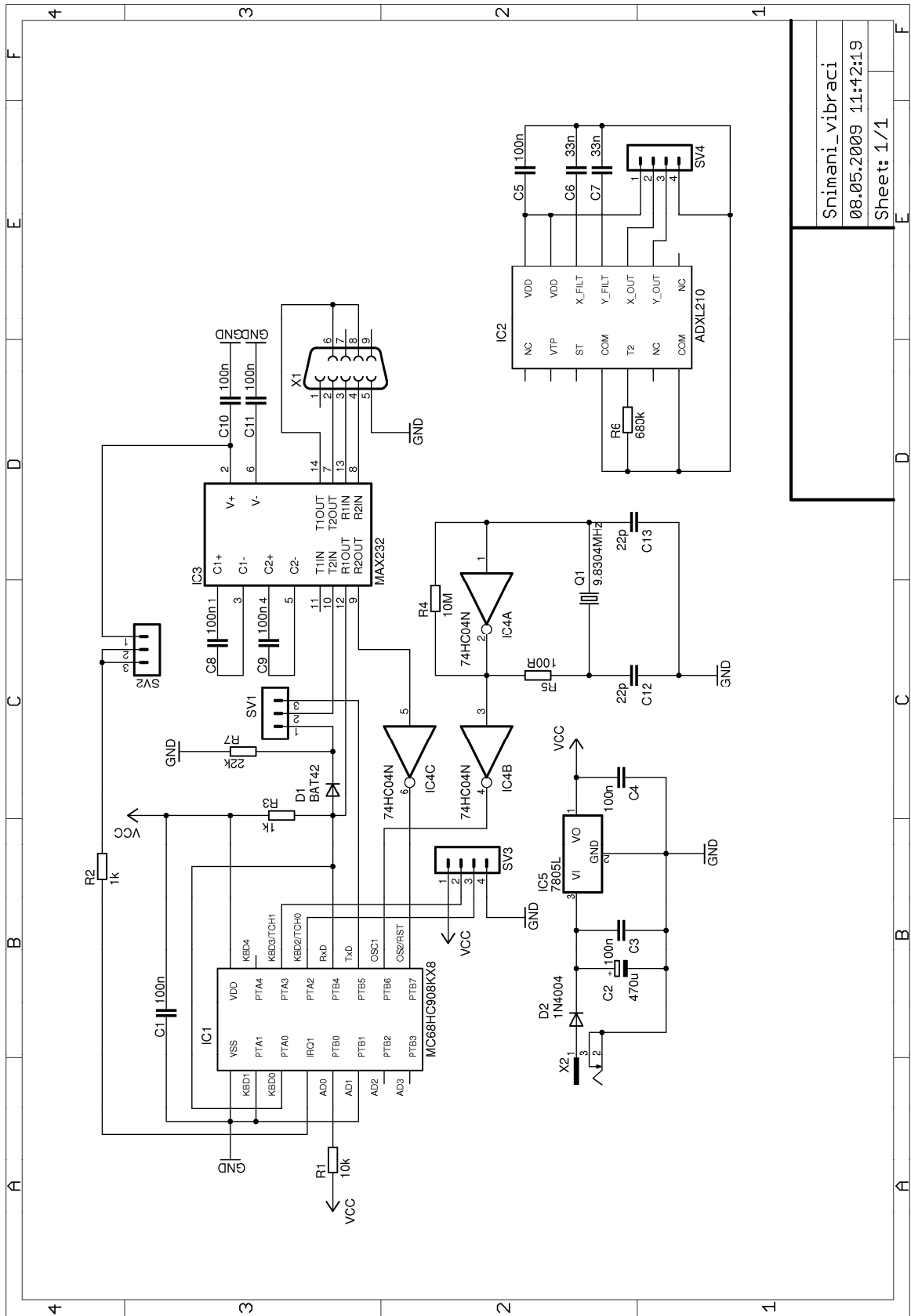
`void sci_out (char znak)` – odešle znak na sériové rozhraní.

```
sci_out('C');
```

`void sci_str_out (char* string)` – výstup textového řetězce na SCI.

```
sci_str_out("odesilany retezec");
```

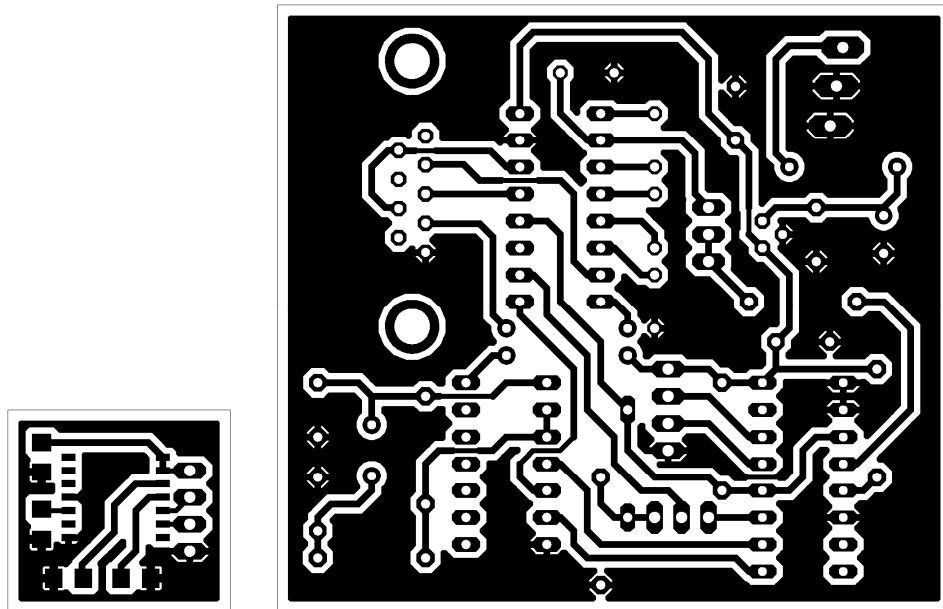
# PŘÍLOHA P III: SCHÉMA ZAŘÍZENÍ



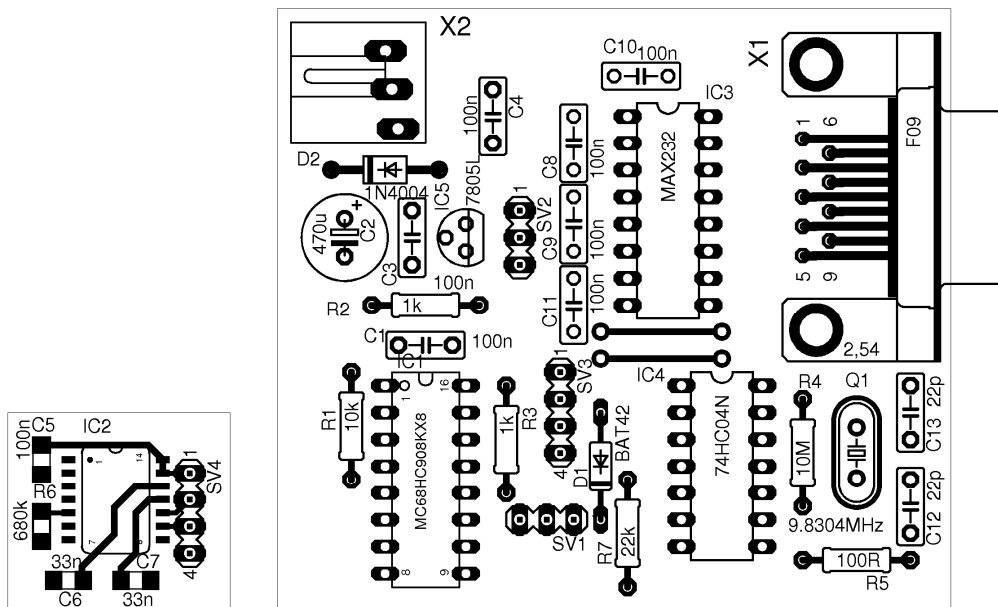
Snimani\_vibraci  
 08.05.2009 11:42:19  
 Sheet: 1/1

# PŘÍLOHA P IV: DESKY PLOŠNÝCH SPOJŮ

## DPS STRANA SPOJŮ



## OSAZOVACÍ PLÁN



Pozn.: Malý obvod s akcelerometrem je zobrazen ze strany spojů, protože s výjimkou připojovacího konektoru jsou všechny součástky SMD osazené na straně spojů.