# Archivační teploměr s využitím mikropočítače řady Motorola HC08

Jakub Reich

Bakalářská práce
2006

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2005/2006

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Jakub REICH** |
| Studijní program: | **B 3902 Inženýrská informatika** |
| Studijní obor: | **Informační technologie** |
| Téma práce: | **Archivační teploměr s využitím mikropočítače řady Motorola HC08** |

Zásady pro vypracování:

**1. Seznamte se s vlastnostmi mikropočítačů řady Motorola HC08 a možnostmi jejich propojení s běžnými senzory teploty.**
**2. Proveďte návrh inteligentního teploměru s použitím zmiňovaného mikropočítače, umožňujícího měření a archivaci teploty pro rozsah venkovního prostředí našeho klimatického pásma.**
**3. Vytvořte softwarové vybavení potřebné pro činnost teploměru**
**4. Proveďte základní ověření funkčnosti zařízení**
**5. Zpracujte technickou dokumentaci navrženého řešení**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:


1. **Váňa V.: Začínáme s mikrokontroléry Motorola HC08 Nitron, BEN 2003**
2. **CPU08 Central Processor Unit Reference manual. Motorola, 2001**
3. **MC68HC908KX8 Data sheet. Freescale semiconductor, 2005**
4. **Hruška F.: Technické prostředky automatizace III : senzory, jejich principy a funkce, Zlin 2002**
5. **Rozehnal Z.: Mikrokontrolery Motorola HC11, BEN 2000**


Vedoucí bakalářské práce: **Ing. Viliam Dolinay**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **14. února 2006**

Termín odevzdání bakalářské práce: **16. června 2006**

Ve Zlíně dne 14. února 2006

prof. Ing. Vladimír Vašek, CSc.
*pověřený děkan*

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

doc. Ing. Ivan Zelinka, Ph.D.
*ředitel ústavu*

## ABSTRAKT

Tato bakalářská práce je inspirována snahou zevrubně přiblížit problematiku vývojové větvě mikrokontrolerů rodiny Motorola HC08 a jednočipových mikropočítačů jako celku. Cílem je vytvoření prototypu archivačního teploměru, jednotky pro samostatné periodické měření a archivaci teplot v klimatických podmínkách mírného podnebního pásu a možnosti jejich zpětného zpracování. Zpřístupněním technického popisu tohoto prototypu se otevírá příležitost jeho uplatnění v praxi nebo úlohách typově příbuzného charakteru.



Klíčová slova:

Mikropočítač, Motorola, HC08, Nitron, teploměr, AD592



## ABSTRACT

This bachelor thesis is inspired by aspiration to approach exact problems of development of microcontrollers Motorola HC08 family branch and singlechip microcontrollers as a whole. Main aim is creation of archiving thermometer, units for separate periodic measurement and archiving of temperatures in climatic conditions of genial climatic zone for the possibility of its back processing. Equalization of the technical documentation of this prototype is opening the opportunity to its usefulness in practices or in tasks of the near type character.



Keywords:

Microcontroller, Motorola, HC08, Nitron, thermometr, AD592

Rád bych na tomto místě poděkoval vedoucímu bakalářské práce Ing. Viliamu Dolinayovi za podněty, rady a odborné vedení mé bakalářské práce. Dále Ing. Petru Dostálkovi za neutuchající ochotu vedení při konstruktivním vypracování návrhu elektroniky plošných spojů a Ing. Josefu Kovářovi za poskytnutí informačních a motivačních prostředků k dokončení práce.

V neposlední řadě bych chtěl poděkovat rodinným členům za projevenou morální a lidskou podporu, která dopomohla vzniku tohoto projektu.

*Motto*

“
Jen dvě věci jsou nekonečné, vesmír a lidská hloupost.
Tím prvním si ovšem nejsem tak jist.
”

ALBERT EINSTEIN (* 1879 – † 1955)

Souhlasím s tím, že s výsledky mé bakalářské práce může být naloženo podle uvážení vedoucího bakalářské práce a vedoucího katedry. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že tuto bakalářskou práci jsem vypracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 14. 06. 2006 .....

Jakub Reich

# OBSAH


**ÚVOD** ..... **8**

**I TEORETICKÁ ČÁST** ..... **9**

**1 MOTOROLA HC08** ..... **10**

1.1 ZNAČENÍ MIKROKONTROLERŮ ŘADY HC08 ..... 11

1.2 NITRON ..... 11

1.2.1 Zapouzdření ..... 12
1.2.2 Mikroprocesor ..... 13
1.2.3 Sběrnice ..... 13
1.2.4 Generátor hodinových kmitů ..... 14
1.2.5 Paměť ..... 14
1.2.6 Čítač/Časovač ..... 17
1.2.7 A/D převodník ..... 18
1.2.8 Reset a přerušení ..... 18
1.2.9 Sériové rozhraní ..... 20
1.2.10 Budiče zobrazovačů LCD ..... 21

1.3 PROGRAMOVÁNÍ MIKROKONTROLERŮ HC08 ..... 21

1.3.1 Monitor ..... 21
1.3.2 Programovací model ..... 22
1.3.3 Jazyk symbolických adres a vyšší jazyky ..... 22
1.3.4 Instrukční soubor ..... 23
1.3.5 Adresovací režimy ..... 24

**2 MĚŘENÍ TEPLOTY** ..... **26**

2.1 POLOVODIČOVÉ SENZORY ..... 27

2.2 AD592 ..... 28

2.2.1 Typické oblasti použití ..... 29
2.2.2 Teorie provozu ..... 30
2.2.3 Faktory ovlivňující přesnost ..... 31
2.2.4 Chyba nastavení, absolutní přesnost a charakteristiky nelinearity ..... 31

**II PRAKTICKÁ ČÁST** ..... **33**

**3 VÝVOJOVÉ PROSTŘEDKY** ..... **34**

3.1 PRINCIP ČINNOSTI ..... 34

3.2 STARTKIT JANUS ..... 35

3.2.1 Napájecí napětí ..... 36
3.2.2 Generátor hodinových kmitů ..... 36
3.2.3 Sériové rozhraní RS232 ..... 37
3.2.4 MINIMON ..... 37

3.3 ICS08 A VIEWCOM ..... 38

**4 TEPLOMĚR SE SENZOREM AD592CN** ..... **40**

**5 SOFTWARE** ..... **42**

5.1 TEPLOMER.ASM ..... 43
**ZÁVĚR ..... 58**
**SEZNAM POUŽITÉ LITERATURY ..... 59**
**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK ..... 60**
**SEZNAM OBRÁZKŮ ..... 61**
**SEZNAM TABULEK ..... 62**
**SEZNAM PŘÍLOH ..... 63**

# ÚVOD

Vznik, vývoj a expanze polovodičových technologií v druhé polovině 20. století otevřely cestu k širokému uplatnění v oblastech výpočetní, měřící a řídící techniky. Vynález mikroprocesoru a posléze i jednočipových mikropočítačů a mikrokontrolerů byl tedy přirozeným vyústěním procesu postupné integrace, zkvalitňování a optimalizace stávajících, i hledání nových konceptů využití těchto technologií. Mikropočítače patří již třetí desetiletí mezi významné mikroelektronické prostředky, s neuvadajícím trendem vývoje a bohatým tržním zázemím.

Aplikačně se objevují při řešení velkého množství charakterově rozmanitých úloh, běžného i profesního života, ať již v multifunkčních měřících systémech, konstrukcích číslicových regulátorů, prodejních automatů nápojů či jízdenek, automobilovém průmyslu, řízení výrobních linek, robotů, bezpečnostních systémech, ale i moderních domácích spotřebičích a hračkách. Upřednostňující výhodou je bezesporu jejich jednoduchost, možnost přímého styku s ovládaným prostředím a začlenění komunikačního rozhraní, nepostradatelného v systémech integrované automatizace.

Takové atributy nabízí i zástupci rodiny osmibitových mikrokontrolerů Motorola 68HC08. Firma Motorola, s bohatou historickou tradicí, zaujímá významnou pozici nejen ve výrobě polovodičových integrovaných obvodů a součástek, ale zejména hraje přední úlohu na trhu automobilového průmyslu, bezdrátových, mobilních a satelitních komunikací a spotřební elektroniky.

Měření teploty je prakticky nepostradatelným prvkem řady průmyslových, zemědělských, vědeckých a lékařských procesů. Často se vyskytne potřeba automatizovaného statistického zpracování měřené veličiny, bez přímého zásahu člověka, v rozmezí několika hodin, dní nebo dokonce i let. Přiměřeně danému problému tedy nasazujeme mikropočítačem řízenou samostatnou archivační jednotku, schopnou uchovávat získaná data (nejlépe bez energetické závislosti) a opětovně je v momentě dotazu zprostředkovat. V případě měření teploty se tato jednotka nazývá archivačním teploměrem.

Cílem mé bakalářské práce je srozumitelnou formou přiblížit problematiku jednočipových mikropočítačů, především větve mikrokontrolerů HC08 a jejich uplatnění v jednoduchých, typově obdobných aplikacích, jakou je archivační teploměr nebo periodické snímaní jiných vstupních signálů, vyjma teploty.

# I. TEORETICKÁ ČÁST

# 1 MOTOROLA HC08

Historie firmy Motorola sahá do roku 1928, kdy bratři Paul a Joseph Galvinové založili v Chicagu, USA malou ruční dílnu na výrobu elektronických součástek radiopřijímačů. Určili tím budoucí primární orientaci firmy na bezdrátovou mobilní komunikaci. V roce 1949 vyčleňuje firma pobočku pro výzkum, vývoj a využití enormního potenciálu tranzistoru a brzy vlastní jednu z předních světových továren na výrobu polovodičových součástek.

Prvním mikroprocesorem, uvedeným na trh v roce 1974, byl osmibitový 6800. Obsahoval 4000 tranzistorů a nejpodstatnějším rozdílem, oproti konkurenčnímu Intelu je mapování vstupně/výstupních periferií do paměti, což umožňuje využít pro práci s nimi standardní instrukce MOV. Mikroprocesor 6800 se stal rodičem dnešních moderních mikropočítačů i evolučním stavebním kamenem PC firmy Apple Macintosh.

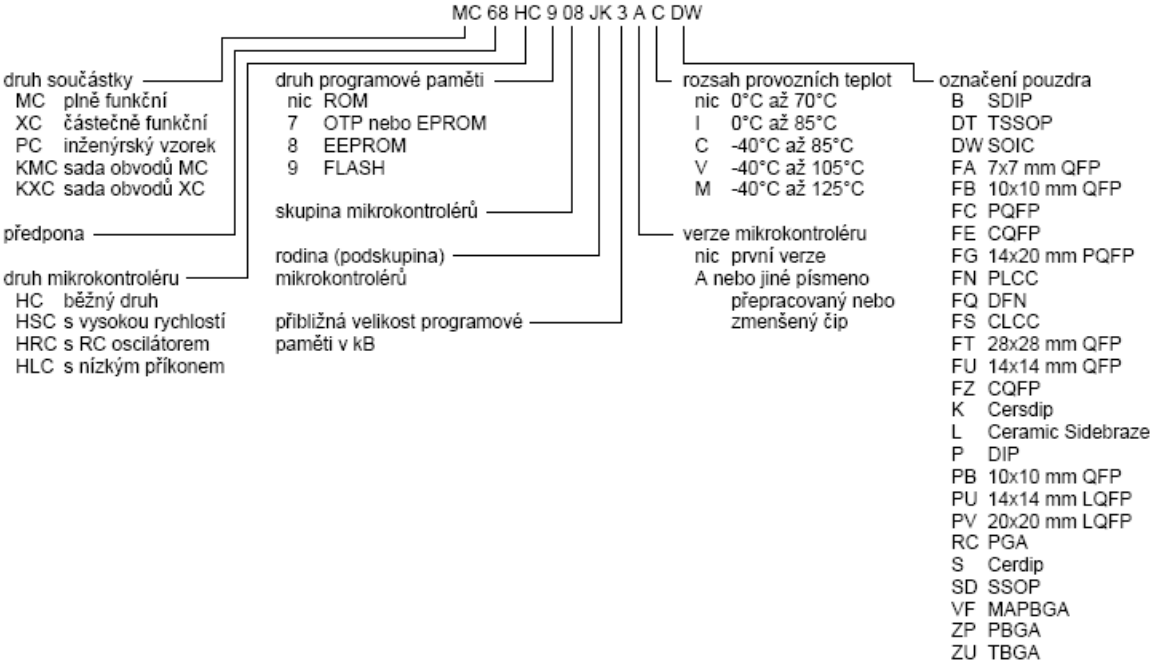
Pojmem mikropočítač či jednočipový mikropočítač označujeme spojení mikroprocesoru, paměti a vstupně/výstupních obvodů do jediného celku s konstrukcí na jediném čipu. Pro nejjednodušší zástupce se vžilo označení mikrořadič nebo mikrokontroler. Mikropočítače jsou programovatelné sekvenční automaty. Tím se rozumí fakt, že zpracovaní dat (signálu) je prováděno v nedělitelných krocích, v nichž se prochází známou posloupností stavů, kde průchod těmito stavy závisí kromě momentálních hodnot datových signálů ještě na jejich předchozích hodnotách.

Mikrokontrolery rodiny HC08 představují typické zástupce své kategorie, navazující na řadu HC05, se kterou jsou zpětně kompatibilní na úrovni zdrojového kódu. Konstrukčně odpovídají osmibitové architektuře s uspořádáním modelu CISC (komplexní instrukční soubor), von Neumannova typu, kdy data i program jsou umístěny ve stejném paměťovém prostoru. Struktura jádra CPU08, vyráběného technologií HCMOS, vychází z historického mikroprocesoru 6800 a staršího jednočipového mikropočítače 6801. Charakteristickou vlastností je minimum pracovních registrů. Odstranění druhého střadače bylo vyváženo doplněním o instrukce pro rychlou a flexibilní práci s pamětí RAM a registry periférií. Jádro je také zdokonaleno lepším řízením smyček, upravenými instrukcemi BCD kódu a předvídáním instrukcí, s úsporou 20 – 30 %.

Další vlastnosti řady HC08 jsou rozebrány na příkladu mikrokontroleru Nitron v kapitole1.2. Nitron.

## 1.1 Značení mikrokontrolerů řady HC08

Jak již bylo dříve uvedeno, tvoří obvody HC08 ucelenou řadu výkonných osmibitových mikrokontrolerů se stejným jádrem CPU08. Jednotlivé typy se liší pouze skladbou periferií a velikostí pamětí. Způsob značení mikrokontrolerů HC08[1] je na obrázku (Obr. 1).

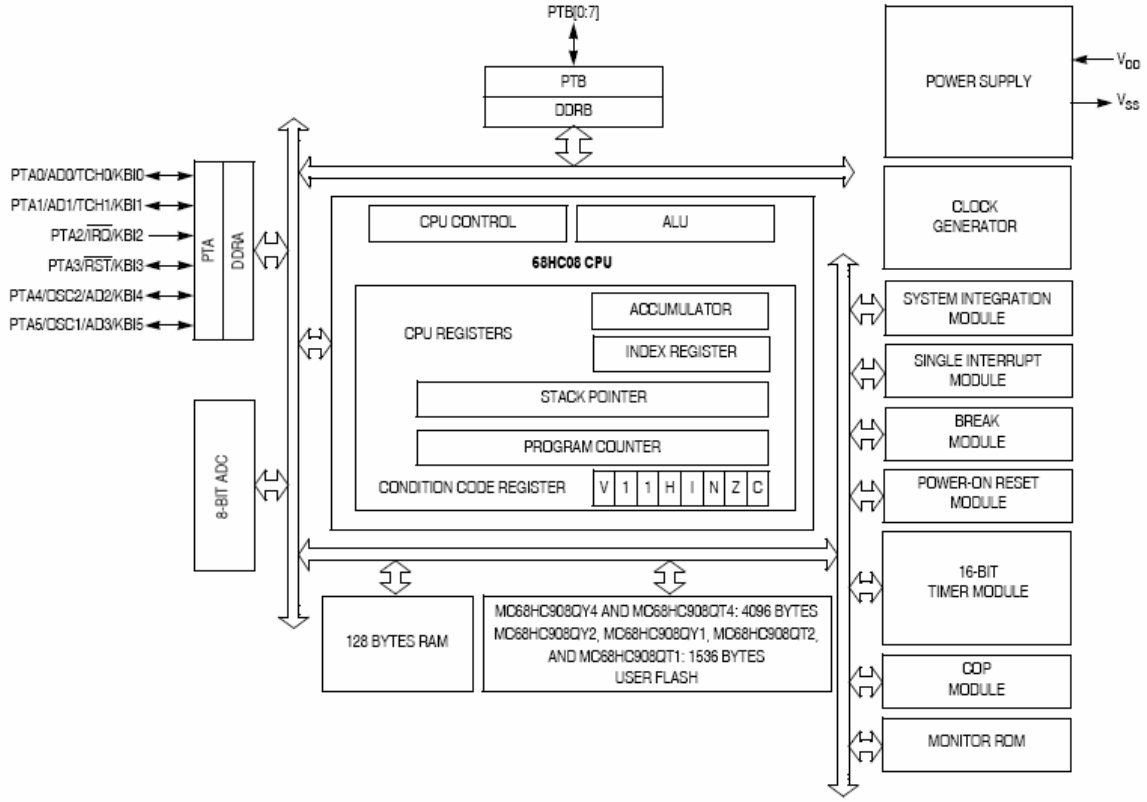



*Obr. 1 Značení mikrokontrolerů HC08*

## 1.2 Nitron

Jedním z nejnovějších a nejmenších zástupců rodiny Motorola HC08 je v současnosti MC68HC908QT/QY s obchodním označením „Nitron“. Nekomplikovanou strukturou, podřízenou záměru dosáhnout značného výkonu při minimálním počtu vývodů a cenovou dostupností nabízí ideální uplatnění v jednoduchých samostatných aplikacích, jakou je právě archivační teploměr.

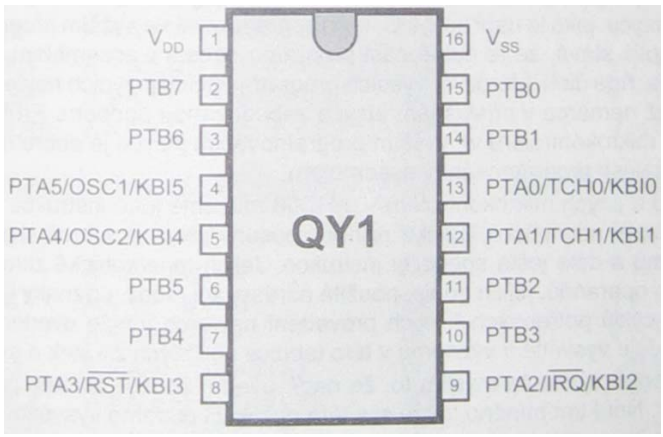
[1] Obvody s malými pouzdry mohou mít popis z prostorových důvodů zkrácený o některé nevýznamné části – např. MC68HC908QT4CP – tedy NITRON v osmi vývodovém provedení má na obvodu označení MC908Q4CP




*Obr. 2 Blokové schéma MC68HC908QT/QY*

### 1.2.1 Zapouzdření

Zapouzdření PDIP či SOIC se vyrábí v provedení s osmi (QT) nebo šestnácti (QY) vývody (šestnácti vývodové také TSSOP). Některé z vývodů mají až tři různé funkce, v daném okamžiku mohou však vykonávat pouze jedinou. Je-li požadavek na několik funkcí téhož vývodu současně, má přednost ta s nejvyšší přiřazenou prioritou, určenou svým pořadím uvedeným u odpovídajícího vývodu, př. (Obr. 3). [1], [7]




*Obr. 3 Zapouzdření MC68HC908QY1*

### 1.2.2 Mikroprocesor

Mikroprocesor nebo také CPU (Central Processor Unit) představuje základní výkonnou jednotkou mikropočítače. Skládá se z řadiče a aritmetickologické jednotky. Řadič se stará o příjem a správné dekódování instrukcí, veškeré časování dějů v mikropočítači a generování řídících signálů pro datové přenosy sběrnice. Aritmetickologická jednotka pak provádí na základě řídících signálů aritmetické a logické operace. Součástí CPU je i sada pracovních registrů. [1]

Jádro Nitronu pracuje s frekvencí vnitřní sběrnice až 8 MHz. Instrukční soubor poskytuje možnost adresovatelného prostoru 64 kB paměti programu/dat v 16 adresovacích módech, přesuny dat z paměti do paměti bez použití střadače a instrukce pro rychlé násobení (8 bitů x 8 bitů) a dělení (16/8 bitů). Modulární architektura dovoluje úpravu vnitřní sběrnice pro adresaci za hranicí 64 kB. Systém nízkonapěťových módů WAIT a STOP umožňuje snadný přechod do úsporných režimů. Schéma pracovních registrů demonstruje (Obr. 4). [2]

*Obr. 4 Pracovní registry Nitronu*

### 1.2.3 Sběrnice

Pojmem sběrnice je označována soustava signálových vodičů, sloužící pro komunikaci vnitřních a vnějších obvodů mikropočítače a řízení této komunikace. Každý mikropočítač má k dispozici alespoň tři druhy sběrnice:

- **Adresová –** jednosměrná (výstupní), slouží k přenosu adresy dat či adresy kódu instrukce.
- **Datová –** obousměrná, slouží k přenosu datových informací.
- **Řídící –** slouží k řízení přenosu dat, adres a synchronizace mikropočítače s okolím.

### 1.2.4 Generátor hodinových kmitů

Mikropočítače pracují téměř výhradně synchronně, tj. ke své korektní činnosti potřebují zdroj hodinových (taktovacích) impulzů o určité frekvenci, kterou využívají vnitřní obvody mikroprocesoru a od které je odvozena délka trvání strojového cyklu a doba vykonávání instrukce. Každá instrukce je složena alespoň z jednoho strojového cyklu a každý strojový cyklus alespoň z jedné fáze.

Strojovým cyklem obvykle označujeme časovou periodu mezi dvěma přístupy k adresové sběrnici. Než se tak stane, musí mikropočítač projít alespoň jednou, zpravidla však výrazně větším počtem fází.

Hodinový signál je signálem s nejvyšším kmitočtem, se kterým mikropočítače pracují. Nemusí být vždy nutně totožný s frekvencí krystalu oscilátoru, který se pro jeho generování používá a zpravidla neposkytuje dostatečnou informativní hodnotu o výpočetním výkonu celého mikropočítače, pro rozdílnou rychlost jádra a sběrnice, která bývá obvykle podílem základní frekvence. Záleží však i na dalších atributech, např. šířce instrukčního souboru. [1]

Interní oscilátor mikrokontrolerů Nitron pracuje při napájecím napětí 2,7 – 5 V standardně s frekvencí 12,8 MHz. Frekvence sběrnice odpovídá 1/4 frekvence jádra (tedy 3,2 MHz). Skutečný kmitočet však kolísá vlivem nedokonalosti výrobních procesů v rozsahu ±25% své nominální hodnoty. Nastavením příslušného registru jej lze doladit k hodnotě ±5%. Variantou je připojení externího oscilátoru, RC oscilátoru (připojuje se pouze odpor, kapacita je integrována na čipu) nebo externího krystalu či keramického rezonátoru. Maximální frekvence jádra při použití externího zdroje hodinových kmitů je 32 MHz (pro napájení do 3 V pouze 16 MHz). [7]

### 1.2.5 Paměť

Pamětí mikropočítače se rozumí polovodičově realizovaný adresovatelný prostor, sloužící k uchování informace (dat) po požadovanou dobu a její zpětné přečtení. Obecně se paměti posuzují dle několika kritérií, především kapacity, přístupové doby (časový interval nutný k přenesení informace z paměti na její výstup), schopnosti zápisu a čtení (RWM, ROM), cyklu paměti (minimální perioda mezi dvěma po sobě následujícími přístupy k paměti), dynamických vlastností (dle nutnosti obnovování uložených dat), destruktivnosti při čtení,

energetické závislosti, metodiky přístupu k jednotlivým paměťovým buňkám (RAM, sekvenční, asociativní), spolehlivosti a ceny za bit.

Adresovatelný prostor rodiny HC08 lze podle účelu, rychlosti a způsobu přístupu k datům logicky rozčlenit na tři části:

- Registry a zápisník
- Zásobník
- Obecná datová paměť

Registry a zápisník slouží k dočasnému uchování zpracovávaných dat. Registry přijímá a předává mikropočítač informace důležité pro chod programu, operandy aritmetických a logických operací, adresy i kódy právě vykonávaných instrukcí, paměťových buněk s nimiž se pracuje, aktuální nastavení a stav periferií, žádosti o přerušení, příznaky přetečení, atd. Některé registry, využívané pouze vnitřně (instrukční registr, adresový registr paměti, …) jsou zpravidla programově přístupné pouze ke čtení. [1]

Zápisník je konstruován jako paměť RAM (Random Access Memory), tedy paměť s libovolným přístupem. Kapacita zápisníku je obvykle velmi omezená, je určen především k ukládání mezivýsledků složitějších výpočtů nebo důležitých proměnných.

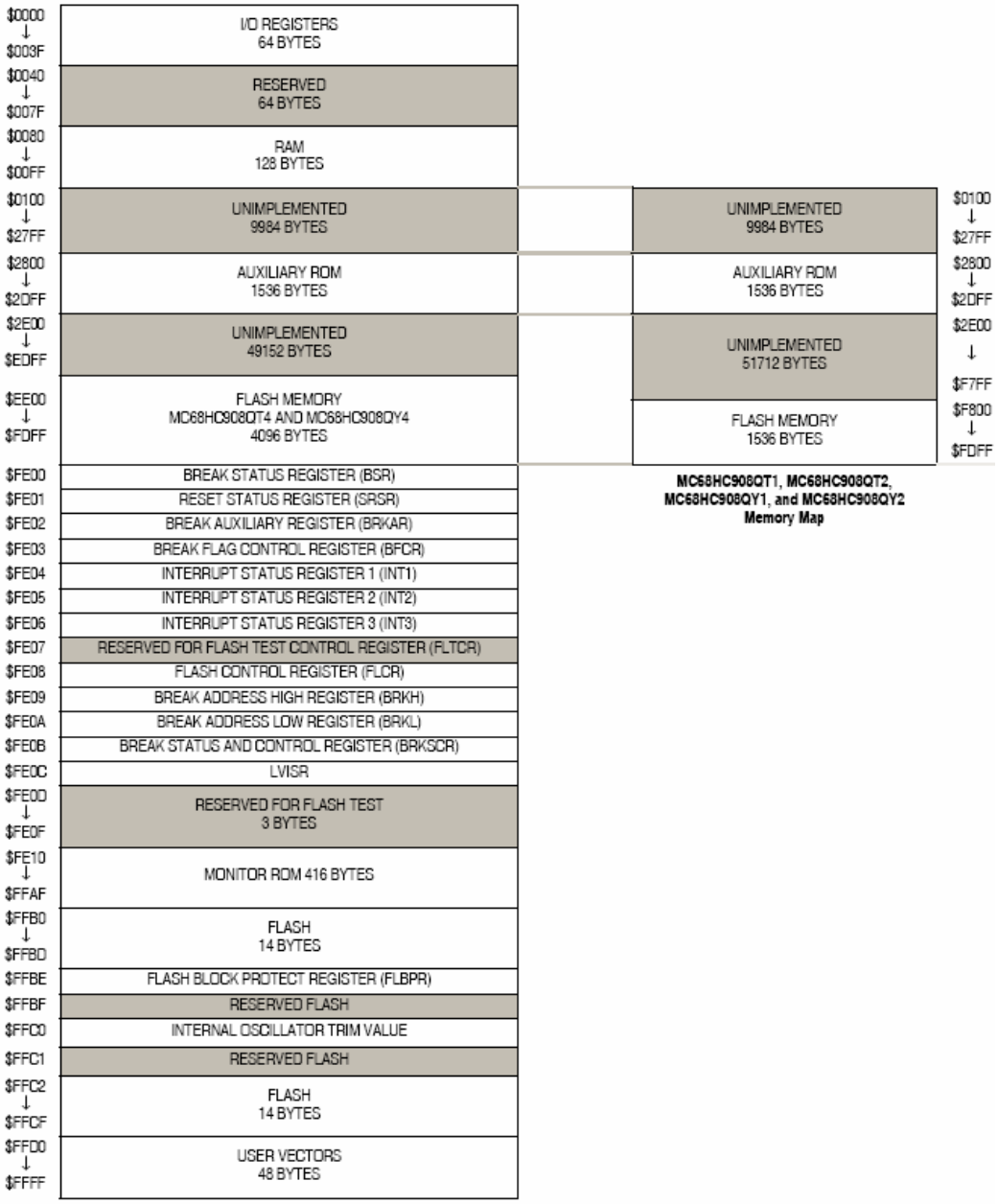
Zásobník je paměť typu LIFO, z anglického označení Last-In First-Out, vyjadřujícího skutečnost, že přístup k uloženým datům je sekvenční, tj. poslední hodnota zapsaná na vrchol zásobníku bude přečtena jako první. Vrcholem zásobníku označujeme aktuální paměťovou buňku, do níž se ukládají nebo odkud se načítají data. Adresu vrcholu uchovává ukazatel zásobníku (Stack Pointer), který je při zápisu dat inkrementován a při čtení dekrementován. Mikropočítače Motorola inkrementují ukazatel zásobníku k nižším adresám a naopak. Zásobník využívá kromě programu i CPU, pro uložení návratové adresy v případě volání podprogramu nebo přerušení. Během přerušení se pak automaticky ukládá na zásobník obsah všech pracovních registrů.

Obecná datová paměť slouží k ukládání dat programu a větších datových struktur. Jedná se o kapacitně nejrozsáhlejší paměť, kterou mikrokontrolery disponují. V řadě HC08 jsou zastoupeny pamětí ROM, EEPROM a FLASH. [2]

Velikost paměti FLASH mikrokontrolerů Nitron, v závislosti na provedení uvádí (Tab. 1). Model paměťové mapy Nitronu představuje (Obr. 5).

| Provedení | Velikost paměti FLASH | A/D převodník | Počet vývodů |
|---|---|---|---|
| MC68HC908QT1 | 1536 B | - | 8 |
| MC68HC908QT2 | 1536 B | 4 kanály, 8 bitů | 8 |
| MC68HC908QT4 | 4096 B | 4 kanály, 8 bitů | 8 |
| MC68HC908QY1 | 1536 B | - | 16 |
| MC68HC908QY2 | 1536 B | 4 kanály, 8 bitů | 16 |
| MC68HC908QY4 | 4096 B | 4 kanály, 8 bitů | 16 |

*Tab. 1 Typy provedení MC68HC908QT/QY*




*Obr. 5 Paměťová mapa mikrokontrolerů Nitron*

### 1.2.6 Čítač/Časovač

Hlavní čítač, jako základní časovací periferie, umožňuje mikropočítačům HC08 generování a měření přesných časových intervalů, generování periodických signálů či pravidelných maskovatelných přerušení. Mimo to představuje zdroj časové základny obvodu COP (Computer Operating Properly). [1], [2]

U Nitronu, je konstruován jako dvoukanálový šestnáctibitový volně běžící čítač signálu, odvozeného od strojových cyklů mikropočítače, s vestavěnou programovatelnou sedmibitovou předděličkou. Podporovanými funkcemi jsou:

- Input capture
- Output compare
- PWM

**COP –** často označovaný termínem „Watchdog“, je nezávislý časovač s přednastavitelnou časovou konstantou, vybavený signálem přetečení. Podstatou funkce obvodu COP je stálá informace o korektním běhu programu. Dojde-li k přetečení, automaticky se vyvolá reset mikropočítače. Aby k resetu nedošlo, musí spuštěný program časovač periodicky nulovat.

**Input capture –** jednotka záchytu hrany je tvořena šestnáctibitovým záchytným registrem připojeným na sběrnici hlavního čítače, obvody řízení záchytu a oddělovacími obvody datové sběrnice. Obvody řízení záchytu detekují zvolený typ hrany (náběžná, sestupné, popř. obojí) vstupního signálu a v okamžiku úspěšné detekce generují zápisový impulz pro záchytný registr. V záchytném registru se tedy uchovává aktuální hodnota hlavního čítače. Jednotku lze využít k měření délky periody, fázového posuvu, apod.

**Output compare –** jednotka výstupního komparátoru je určena ke generování hrany signálu (náběžné, sestupné, obojí, popř. změny úrovně ze stávající na opačnou) v závislosti na aktuální hodnotě hlavního čítače. Jednotku tvoří šestnáctibitový datový registr, šestnáctibitový komparátor a obvody generování výstupní hrany. Hodnota uložená v datovém registru je komparátorem porovnávána s aktuálním stavem hlavního čítače. V okamžiku rovnosti obou hodnot je vytvořen řídící signál pro obvod generování výstupní hrany.

**PWM –** je zkratka anglického označení Pulse Width Modulation, pulsně šířkové modulace signálu. Princip vychází ze signálu s konstantní periodou T, přičemž se mění střída napětí (tj. poměr délky impulzu v logické úrovni 1 k délce mezery v logické úrovni 0, uvažovaný

v jedné periodě). Do úrovně logické 1 přechází signál v momentě přetečení hlavního čítače, změna úrovně je zajištěna jednotkou výstupního komparátoru.

### 1.2.7 A/D převodník

Mikropočítače bývají velmi často nasazovány v aplikacích, ke zpracování analogových signálů. Takový signál nemohou zpracovat bezprostředně, proto jsou vybaveny analogově digitálním převodníkem, obvodem, který periodicky vzorkovaný analogový signál převádí na odpovídající diskrétní hodnoty.

Rodina mikrořadičů HC08 využívá osmibitových (popř. desetibitových) převodníků s postupnou lineární aproximací. Dovolují jednorázový či kontinuální převod s nastavitelnou rychlostí (nejméně však 17 μs) a vyvoláním přerušení po jeho dokončení. Analogový multiplexer zpřístupňuje přepínáním kanálů možnost měření několika analogových jedním převodníkem. Přítomnost A/D převodníku v závislosti na provedení mikrořadiče Nitron uvádí tabulka (Tab. 1) v kapitole 1.2.5 Paměť. [7]

### 1.2.8 Reset a přerušení

Reset a přerušení představují prostředek, kterým lze asynchronně vůči hlavní programové smyčce vyvolat inicializaci mikropočítače (v případě resetu), nebo vykonat obslužný podprogram (v případě přerušení). Uvažujeme-li přitom reset jako nemaskovatelné přerušení s nejvyšší prioritou, odkazující na počáteční adresu operačního kódu instrukce programu, je možné popsat daný mechanizmus metodou nepřímého adresování pomocí tabulky resetu a vektorů přerušení, kterou využívají mikropočítače rodiny HC08. Tuto tabulku tvoří pole šestnáctibitových hodnot. Každá hodnota uložená v tomto poli je adresa, která se načte do programového čítače v okamžiku, kdy se vyvolá přerušení programu nebo reset. [1]

| Priorita | Význam | Označení | Adresa |
|---|---|---|---|
| Nejvyšší | Reset | - | $FFFE-$FFFF |
| ↑ | Instrukce SWI | - | $FFFC-$FFFD |
| | IRQ vstup | IF1 | $FFFA-$FFFB |
| | Časovač, kanál 0 | IF3 | $FFF6-$FFF7 |
| | Časovač, kanál 1 | IF4 | $FFF4-$FFF5 |
| | Časovač, přetečení | IF5 | $FFF2-$FFF3 |
| ↓ | KBI vstup | IF14 | $FFDE-$FFDF |
| Nejnižší | A/D převod dokončen | IF15 | $FFE0-$FFE1 |

*Tab. 2 Tabulka vektorů přerušení mikrokontroleru Nitron*

Programově se liší reset od přerušení pouze inicializací bitů masky příznakového registru CCR a neuložením pracovních registrů CPU do zásobníku. Z hardwarového hlediska pak při resetu dochází k uvedení periférií do jejich základního nastavení. Současně je generován signál reset na resetovacím vývodu mikropočítače.

Přerušení lze rozdělit podle prostředku volání na:

- Hardwarová
- Softwarová (instrukcí SWI)

Podle umístění žadatele na:

- Vnitřní
- Vnější

Podle priority a maskovatelnosti na:

- Reset
- Maskovatelná (IRQ)
- Nemaskovatelná (XIRQ)

Správu i řízení přerušení obsluhuje u mikrokontrolerů HC08 modul systémové integrace (SIM). Podporováno je až 24 různých typů (s možností rozšíření na 128). SIM mimo jiné kontroluje přesné časování CPU, interní sběrnice, zabezpečuje detekci neplatných instrukcí a neplatných adres, signálu externího resetu, inicializaci mikrokontroleru po přivedení napájecího napětí (Power-on Reset Module) a je nadřízeným prvkem modulů COP, LVI, IRQ.

Modul LVI (Low-Voltage Inhibit) zajišťuje monitorování velikosti napájecího napětí, ovládá nulovaní obvod, ovlivňuje spotřebu mikrokontroleru a nastavení STOP módu. Jestliže napájecí napětí poklesne pod stanovenou hodnotu, rozdílnou pro různé napěťové režimy, je nastaven příznak umožňující na tuto skutečnost periodicky upozorňovat, případně samočinně mikropočítač resetovat.

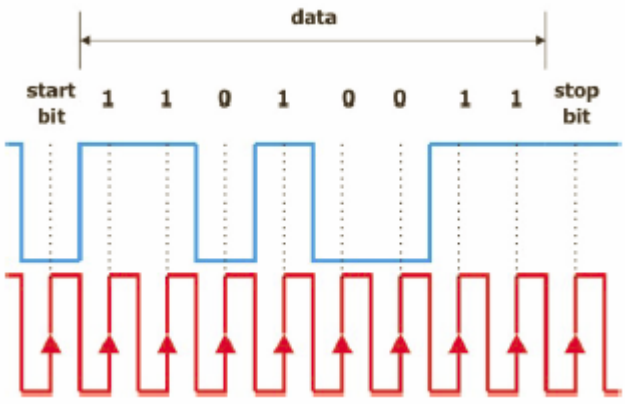
Záchyt žádostí externího maskovatelného přerušení zajišťuje modul IRQ, s nastavitelnou citlivostí na hranu či úroveň signálu, hysterezí a automatickým potvrzováním. Obdobně pracuje i modul KBI, zpřístupňující šest nezávislých, externích maskovatelných přerušení,

určený pro obsluhu vnějších periferií (např. klávesnice) s integrovanou funkcí „probuzení“ z nízkonapěťových režimů. [7]

### 1.2.9 Sériové rozhraní

Při sériové komunikaci probíhá hlavní přenos dat po jednom signálním vodiči, posloupností za sebou řazených bitů. Na úkor výrazně nižší přenosové rychlosti, oproti paralelnímu způsobu, se tím zredukuje počet vodičů a zlepší možnosti galvanického oddělení datové komunikační linky optickými vazebními členy. V zásadě lze rozlišit dva způsoby sériové komunikace:

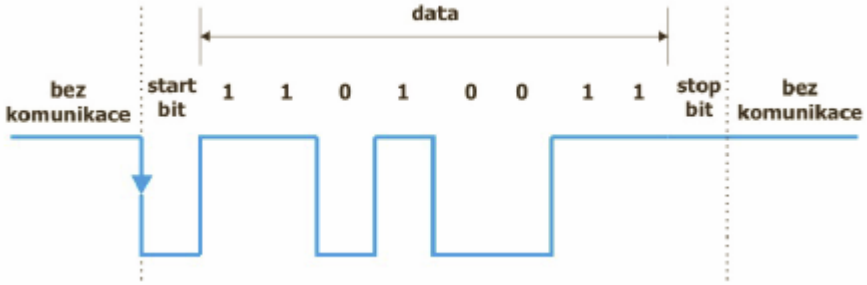
- **Synchronní (SPI)**, kdy se k přenášenému datovému signálu přidává samostatným vodičem nebo vhodnou modulací synchronizační hodinový signál, sloužící k seřízení vysílače a přijímače. Používá se hlavně k přenosu většího množství dat na krátké vzdálenosti, často v rámci periferií jednoho obvodu (např. přenosy budičů LCD, A/D převodníků, atp.) Synchronní kanál bývá kromě vysílacích a přijímacích funkcí vybaven logikou pro detekci kolize a přepínáním směru toku dat. Pracuje v režimu Master/Slave. Jelikož existuje několik standardizovaných formátů synchronního přenosu, je nutné provést shodnou inicializaci vysílače a přijímače.




*Obr. 6 Synchronní přenos*

- **Asynchronní (SCI)**, kdy činnost vysílače a přijímače není synchronizována hodinovým signálem, ale uzavřením dat mezi rámec tzv. start a stop bitů, důležité je dodržení přesné přenosové rychlosti jednoho bitu. V klidovém stavu je signál v logické úrovni 1. Přenos začíná příchodem start bitu (překlopením do logické

úrovně 0). Poté je vyslán určitý počet datových bitů (např. 8, tedy 1 B) a případně paritní bit. Celý rámec bývá uzavřen jedním až dvěma stop bity. Každý bit je vzorkován třemi vzorky uprostřed časového intervalu, přičemž jeho hodnotu určuje stav dvou shodných vzorků. Přenosová rychlost se udává v Baudech, značící počet změn úrovně signálu za jednotku času.




*Obr. 7 Asynchronní přenos*

Nitrony nejsou běžně vybaveny rozhraním pro sériovou komunikaci, její využití musí být ošetřeno hardwarově i programově v dané aplikaci. [2],[3]

### 1.2.10 Budiče zobrazovačů LCD

Některé z modelů rodiny HC08 jsou rovněž vybaveny obvody budičů LCD displejů. Tento modul umožňuje mikropočítači přímo ovládat mnohasegmentový zobrazovač LCD. Všechny napěťové úrovně pro jsou vytvářeny nábojovou pumpou čipu, čímž se omezuje počet vnějších periferií připojených k mikropočítači a tím i cena systému. U tak malých mikrokontrolerů, jako je Nitron, však integrovány nebývají. [10]

## 1.3 Programování mikrokontrolerů HC08

### 1.3.1 Monitor

Monitorem se označuje speciální programový kód instalovaný výrobcem do paměti ROM, sloužící k vytvoření uživatelského rozhraní mezi mikropočítačem a uživatelem. Všechny mikrokontrolery rodiny HC08 jsou vybaveny standardně monitorem MON08. Obsahuje rutinou pro sériovou komunikaci mikropočítače s nadřízeným prvkem (např. PC), rutinu pro práci s programovanou pamětí (EEPROM, FLASH), vykonávání programu v RAM, ale také kompletní testování funkcí mikroprocesoru. [2]

### 1.3.2 Programovací model

Jak je uvedeno v [1], je pojmem programovací model charakterizován soupis vlastností a prostředků, jimiž je mikropočítač vybaven a které může využít při konstrukci programů programátor. Vychází z popisu vnitřní struktury mikropočítače v programátorském smyslu, pracovních registrů CPU, ke kterým má zpravidla programátor nejsnazší přístup a pomocí nichž lze snadno obsloužit i zbývající paměťový prostor (tedy data i program). Z (Obr. 4) v kapitole 1.2.2 je patrné, že základ programovacího modelu mikrokontrolerů HC08 tvoří osmibitový střadač (akumulátor), který se využívá především k aritmetickým a logickým operacím. Dalšími registry jsou šestnáctibitový ukazatel zásobníku SP (Stack Pointer), šestnáctibitový programový čítač PC (Program Counter) a osmibitový stavový registr CCR, využívaný k nastavení maskování a příznaků výsledků aritmetických a logických operací (příznak nuly, přenosu, atd.).

### 1.3.3 Jazyk symbolických adres a vyšší jazyky

Nepřehledný a obtížný zápis programu ve strojovém kódu (pomocí číselného reprezentace instrukcí) byl postupně nahrazen jazykem symbolických adres, kdy každé číselné instrukci je přiřazen její symbolický ekvivalent. Symboly jsou obvykle tvořeny mnemotechnickými zkratkami slov přirozeného jazyka (především angličtiny), zápis programu je tak čitelnější a srozumitelnější.

Překlad programu z jazyka symbolických adres do strojového kódu zajišťuje překladač, tzv. assembler. Transformuje data a adresy různých číselných soustav (např. desítkové, oktalové, hexadecimální) do binární a překládá i znaky do odpovídajících kódů ASCII. Umožňuje programátorovi pomocí tzv. direktiv (příkazy překladače) rezervovat a přidělovat paměť pro dočasné uchování dat, uložit konstanty na určeném místě v paměti, přiřadit symbolická jména paměťovým místům, částem kódu (makrojazyk) a vstupně/výstupním registrům, umí vykonat některé aritmetické operace jako součást procesu překladu, informuje programátora o formálních chybách v programu.

Výhodou je bezprostřední vztah k základní struktuře počítače, což způsobuje, že programy v jazyku symbolických adres mají obvykle menší nároky na rozsah operační paměti než programy zapsané ve vyšších programovacích jazycích. Také jejich optimalizace z hlediska doby potřebné pro výpočet programu je snazší. Nevýhodou jazyka symbolických adres

je nemožnost přenést programy na jiný počítač, pracnost programování a omezené možnosti syntaktických a sémantických kontrol programů při překladu i výpočtu. [11]

Možností je využít k programování některého z vyšších jazyků, zejména mezinárodně uznávaných norem jazyka C (např. ANSI C). Proto byla podpora jazyka C implementována přímo při návrhu jádra mikrokontrolerů rodiny HC08 (nebo HC05).

### 1.3.4 Instrukční soubor

Instrukční soubor typu CISC se vyznačuje vysokým počtem a velkým rozsahem instrukcí, což při jejich efektivním využití vede ke kratšímu výslednému kódu. Instrukční soubor (uvádí [1]) mikropočítačů lze z hlediska určení jednotlivých instrukcí rozdělit na tyto typy:

- **Aritmetické**, operace součtu, rozdílu, součinu, podílu, inkrementace, dekrementace, aritmetických posunů, dvojkového doplňku a porovnávání, jejichž výsledkem je nastavení příznakových bitů (provádí se pomocí operace rozdílu).
- **Logické**, operace logického součtu, součinu, výhradního logického součtu, negace.
- **Rotace a posuny**, operace logických posunů (na rozdíl od aritmetických nerespektují znaménko operandu, provádí prostou záměnu pozic bitů), mohou být prosté nebo s nastavením přetečení.
- **Přesuny**, operace sloužící k přesunu hodnot z paměti do střadače, naopak a mezi jednotlivým paměťovými buňkami.
- **Bitové**, operace pro práci s jednotlivými bity paměťového místa či registrů (nastavení, nulování, podmíněné skoky).
- **Skoky a volání podprogramů**, instrukce podmíněných (relativní adresování) a nepodmíněných (rozšířené a indexové adresování) skoků, volání a návratu z podprogramu, návratu z přerušení.
- **Speciální**, nelze je zařadit do žádné z výše uvedených skupin, nějakým způsobem souvisí s vlastnosti mikropočítače nebo jeho řadiče (např. instrukce WAIT, STOP).

### 1.3.5 Adresovací režimy

Jednou z výhod mikropočítačů HC08, oproti předchůdcům či ostatním vývojovým větvím mikrokontrolerů Motorola (např. HC11) je 16 přístupných adresovacích módů (z [2]):

- **INH (Inherent)**, vlastní adresování. Informace o adrese paměťového místa je obsažena implicitně v kódu instrukce (např. negace hodnoty ve střadači NEGA).
- **IMM (Immediate)**, bezprostřední adresování. Kdy adresa paměťového místa je dána implicitně kódem instrukce a určuje se pouze osmibitový nebo šestnáctibitový operand (např. přičtení hodnoty ke střadači ADD #$01).
- **DIR (Direct)**, přímé adresování. Operand je načítán z adresy uvedené za kódem instrukce. Toto adresování neuvažuje horní byte šestnáctibitových, pracuje se pouze s hodnotami $00-$FF (např. načtení hodnoty z adresy $01 ADD $01). Adresování operující nad oběma byty se nazývá přímé rozšířené.
- **IX (Indexed)**, indexové adresování. Je základním typem nepřímého adresování. Výsledná adresa se generuje z hodnoty obsažené v indexovém registru X, který je součástí instrukce (např. uložení střadače STA ,X).
- **IX1**, indexové s osmibitovým offsetem. Výsledná adresa se generuje z hodnoty obsažené v indexovém registru X a přičtením kladného osmibitového posuvu, jehož hodnota je součástí instrukce (např. STA $12,X).
- **IX2**, indexové registrování s šestnáctibitovým offsetem (např. STA $4345,X).
- **SP1 (Stack Pointer)**, adresování přes SP s osmibitovým offsetem. Obdobně jako u indexového registrování, výsledná adresa se však generuje přičtením offsetu k aktuální hodnotě ukazatele zásobníku (např. STA $01,SP).
- **SP2**, adresování přes SP s šestnáctibitovým offsetem (např. STA $4001,SP).
- **REL (Relative)**, relativní adresování. Používá se pro adresování instrukcí skoků. K programovému čítači se přičítá pevný osmibitový offset, generovaný z šestnáctibitové směrovací adresy, který je považován za znaménkové číslo v rozsahu +127 až – 128.
- **Přenosy z paměti do paměti**
  - **IMD**, bezprostřední adresování / přímé adresování.

  - **DD**, přímé adresování / přímé adresování.
  - **IX+D**, indexové adresování / přímé adresování, kdy hodnota indexového registru je po vykonání instrukce inkrementována.
  - **DIX+**, přímé adresování / indexové adresování, s post inkrementací.
- **IX+**, indexové adresování s post inkrementací.
- **IX1+**, indexové adresování s post inkrementací a osmibitovým offsetem.

# 2 MĚŘENÍ TEPLOTY

Teplota je stavová veličina charakterizující rovnovážný termodynamický stav jakékoliv makroskopické soustavy. Z nultého termodynamického principu plyne, že každému rovnovážnému stavu soustavy lze přiřadit určitou hodnotu teploty a že tato hodnota je táž pro všechny soustavy, které jsou také navzájem v rovnováze. Protože nejrůznější stavové veličiny závisejí na teplotě, tj. mění se při její změně, může být k definici teploty a k jejímu měření použita, obecně vzato, kterákoli z těchto závislostí, a to tak, že se předepíše její tvar u zvolené látky za přesně stanovených podmínek.

Taková definice teploty je však založena na vlastnostech dané látky, tzv. teploměrné látky, popř. skupiny látek (ideálních plynů), a navíc je možná jen pro jistý omezený teplotní interval. Nezávisle na vlastnostech některých látek je definována termodynamická teplota, která je od roku 1960 také jednou ze základních fyzikálních veličin Mezinárodní soustavy jednotek. Pro praktické měření teploty se používá Mezinárodní stupnice z roku 1990 nebo některá z empiricky odvozených teplotních stupnic (Celsiova, Fahrenheitova, Kelvinova).

Termodynamická teplota, definovaná na základě vztahu:

$$T = \frac{Q}{Q_z} T_z$$

platného pro vratný Carnotův cyklus. Q je teplo, které přijme během jednoho cyklu látka, v níž tento vratný děj probíhá, od soustavy s konstantní teplotou T a $Q_z$ je teplo, které během jednoho cyklu tato látka odevzdá další soustavě s konstantní teplotou $T_z$.

Jestliže teplotou $T_z$ je zvolená základní termodynamická teplota, je uvedeným vztahem definována termodynamická teplota T (a to nezávisle na tom, v jaké látce tento cyklus probíhal) a může být také určena měření tepel Q a $Q_z$. Základní teplotou je od roku 1954 stanovena mezinárodně termodynamická teplota trojného bodu vody, jíž se připisuje hodnota 273,16 K. Jednotkou je kelvin, značka K. [6]

Měření je proces, pro jehož udržení je zapotřebí určitý tok energie nebo látkový tok. Je to nejdůležitější, nejpracnější a nejobsáhlejší část experimentální činnosti vedoucí k určení hodnoty žádané veličiny. Měřící proces obvykle prostřednictvím čidla tuto energii více či méně ovlivňuje – spotřebovává. Čidlo (senzor, receptor) pak danou energii transformuje na výslednou informaci o stavu nebo průběhu procesu sledovaného objektu. Nositelem této informace je opět určitá forma energie nebo akumulace látky generované měřícím čidlem.

Není-li však tato forma energie vhodná pro další zpracování informace používají se převodníky pro transformaci této energie na energii vhodnou pro další zpracování. Měření je realizováno technickými měřícími prostředky, které společně tvoří buď měřící kanál nebo celý měřící systém. [3]

Vlastnosti technických prostředků měření jsou charakterizovány statickými a dynamickými parametry a parametry spolehlivosti [4]. Statické parametry popisují provozně technické vlastnosti přístrojů a jejich jednotlivých prvků v ustáleném stavu. Při sledování časových průběhů proměnných veličin od počátku změny do ustálení odezvy je daný průběh procesu měření charakterizován dynamickými parametry. Příčinou přechodových dějů jsou hmotné, tepelné, elektrické a magnetické kapacity, setrvačnosti, tlumení hmot měřicích prvků, impedance, rychlosti změn fyzikálních a chemických parametrů měřených medií atp. Dynamické parametry jsou definovány časovou konstantou nebo přechodovou funkcí, váhovou funkcí, frekvenční charakteristikou, frekvenční funkcí, dynamickou chybou. Spolehlivost je definována jako schopnost zařízení plnit funkci podle daných požadavků po definovaný časový úsek.

## 2.1 Polovodičové senzory

Pro měření teploty se v současnosti často využívá zejména elektrické (termoelektrické napětí, změna odporu, polovodičový efekt) nebo elektromagnetické (bezdotykové způsoby) závislosti vlastností některých látek a prvků [4]. U polovodičových senzorů teploty se podobně jako u kovových využívá teplotní závislosti elektrického odporu. Tyto senzory lze rozčlenit podle struktury na polykrystalické a monokrystalické.

Polykrystalické polovodičové senzory (termistory):

- **NTC (Negastory)**, mají záporný teplotní součinitel odporu. Se zvyšující se teplotou roste koncentrace nosičů náboje a elektrický odpor klesá. Negastory se vyrábí práškovou technologií (lisováním a slinováním za vysokých teplot) ze směsi oxidů kovů (např. $Fe_20_3$+$Tu0_2$, MnO+CoO, případně $Ti0_2$, CuO, NiO, BaO). Aplikují se běžně v rozsahu teplot od -50 do 150°C (méně často -100 °C až 300 °C, výjimečně do 400 °C) nebo speciálně pro rozsahy od 4 K do 1000 °C. Teplotní součinitel je pětkrát až padesátkrát větší než u kovů. Závislost odporu na teplotě je velmi nelineární.

- **PTC (Pozistory)**, jsou termistory s velmi nelineární závislostí. Mají kladný teplotní součinitel odporu. Jejich odpor s rostoucí teplotou nejprve mírně klesá a po překročení Curieovy teploty strmě roste až k další tepelné hranici, kdy začíná opět klesat. PTC termistory se vyrábí z polykrystalické keramiky, např. z titaničitanu barnatého ($BaTiO_3$), sintrováním přesně řízených práškových směsí obvykle do tvaru malého disku. Referenční teplota PTC termistorů (teplota přechodu) závisí na chemickém složení a obvykle se používá teplota v rozsahu 60 až 180 °C.

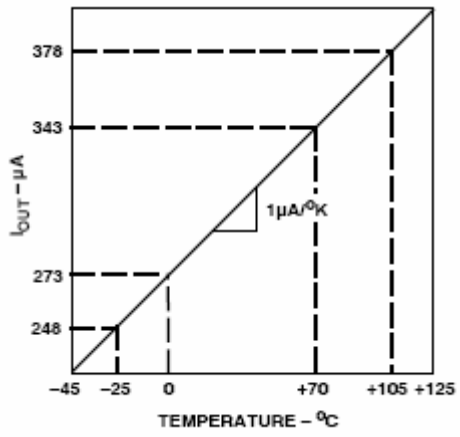
Monokrystalické polovodičové senzory:

- **Bez PN přechodu**, lze vyrobit z křemíku, germania nebo india, v praxi se však setkáváme nejčastěji se senzory křemíkovými. Teplotní rozsah je obvykle -55 až 150 °C, k dostání jsou však běžně i senzory s horní teplotní hranicí 300 °C.
- **S PN přechodem**, jsou založeny na teplotní závislosti změn vodivosti PN přechodu. Polovodičové materiály Si a Ge mají při nízké teplotě velmi malou vodivost. Je to dáno tím, že všechny volné elektrony jsou vázány jako valenční elektrony v krystalové mřížce. Se zvyšováním teploty se zvyšuje jejich energie a překonávají bariéru valenční oblasti a přechází do vodivé oblasti. Jako senzory teploty jsou používány diody, tranzistory, integrované obvody.

## 2.2 AD592

AD592 je dvousvorkový monolitický integrovaný obvod –termoelektrický převodník, který na výstupu poskytuje proud úměrný absolutní teplotě. Díky širokému rozsahu napětí slouží snímač jako vysokoimpedanční, na teplotě závislý zdroj proudu o hodnotě 1 μA/K. Charakteristickými znaky jsou:

- Vysoká předkalibrovaná přesnost: 0,5°C při +25°C.
- Výborná linearita, s odchylkou max 0,15°C (v rozsahu teplot 0°C až 70°C)
- Široký teplotní rozsah, -25°C až +105°C
- Rozsah napájecího napětí 4 – 30 V
- Výborná opakovatelnost a stálost
- Minimální chyba způsobená vlastním ohřevem




*Obr. 8 Závislost výstupního proudu na teplotě*

Oproti předchozím verzím, vylepšený design a laserem vrstvené odpory integrovaného obvodu, dovolují AD592 dosáhnout vysoké úrovně přesnosti a nízké nelinearity chyb dříve nedosažitelné v této cenové hladině.

AD592 může být použit v teplotním rozsahu od –25°C až do +105°C. V tomto teplotním rozsahu jsou obvykle používána teplotní čidla jako např. termistor, RTD, termočlánek, dioda, které jsou při stejných parametrech nesrovnatelně dražší především proto, že je nutné použít pro jejich funkci dalších nákladných prvků jako jsou např. lineární obvody, přesné napěťové odkazy, můstky, odporové měřící obvody, kompenzátory studených konců, atd. To však není potřeba při použití AD592, a právě to zněj dělá cenově velice efektivní teplotní snímač. [3]

### 2.2.1 Typické oblasti použití

AD592 se používají např. jako teplotní senzory, pro automatická měření a řízení teploty, pro systémy monitorující topení, klimatizace a ventilace, pro průmyslové řízení teploty, kompenzace studeného spoje termočlánku. Obzvláště vhodné jsou v zařízeních snímajících dálkově. AD592 je totiž odolný proti náhlému poklesu napětí a napěťovému úbytku na dlouhém vedení a to díky jeho vysokoimpedančnímu napěťovému výstupu.

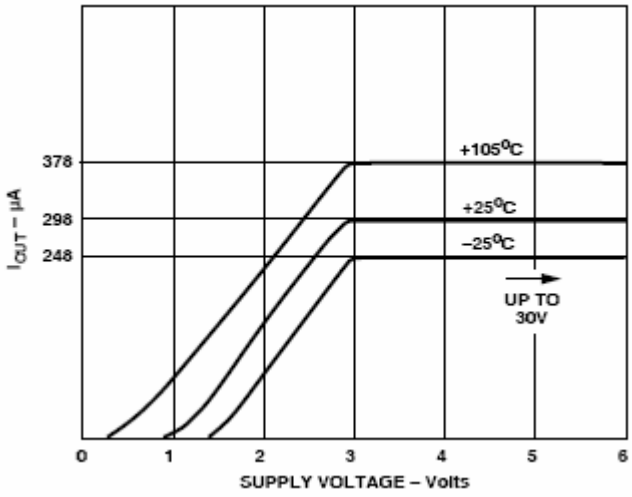
AD592 je uloženo v plastovém obalu TO-92 dimenzovaného pro teploty od -45°C do +125°C. Použití je však v rozsahu od –25°C do +105°C.

### 2.2.2 Teorie provozu

AD592 využívá základních vlastností křemíkových tranzistorů, kterou je schopnost zjistit teplotu úměrně k proudovému výstupu. Pokud dva stejné tranzistory pracují při konstantním poměru kolektorových proudových hustot – r, pak rozdíl základního zdroje napětí je:

$$\Delta U = \frac{kT}{q} \ln r$$

Pokud Boltzmannova konstanta i náboj elektronu q jsou konstantní, výsledné napětí je přímo úměrné absolutní teplotě (PTAT). AD592 převádí toto rozdílné napětí na PTAT proud pomocí nízkoteplotního koeficientu tenké vrstvy odporů. Tento PTAT proud je pak použit k tomu, aby celkový výsledný proud byl přímo úměrný stupňům Kelvina. Výsledkem je zdroj proudu s výstupem, který je roven násobku převodního faktoru a teploty senzoru. (Obr. 9) znázorňuje průběh napětí a proudu v obvodu při 25°C a při extrémních teplotách (-25°C, +105°C).




*Obr. 9 Průběh napětí a proudu v obvodu při 25°C a extrémních teplotách*

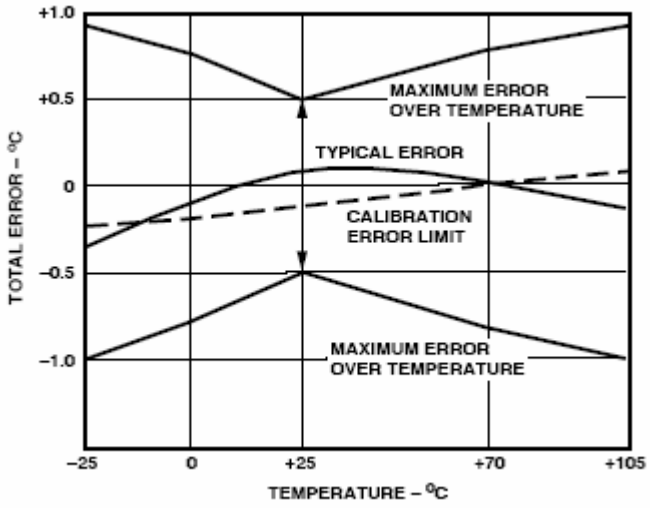
Tovární nastavení převodního faktoru na 1 µA/K je upraveno vrstvenou úrovní tak, aby načítané teploty AD592 korespondovaly s aktuální teplotou. Během laserové úpravy má IC teplotu kolem 25°C a je napájeno napětím o hodnotě 5V. Zařízení je pak baleno a automaticky teplotně testováno.

### 2.2.3 Faktory ovlivňující přesnost

Limity přesnosti usnadňují použití AD592 v různých aplikacích. Pro výpočet celkové chyby v daném systému je důležité správně interpretovat charakteristiky přesnosti, chyby nelinearity, reakce obvodu na změny napětí a vliv okolní teploty. Stejně jako u jiných elektronických provedení bude mít hlavní vliv na přesnost výběr externích komponentů.

### 2.2.4 Chyba nastavení, absolutní přesnost a charakteristiky nelinearity

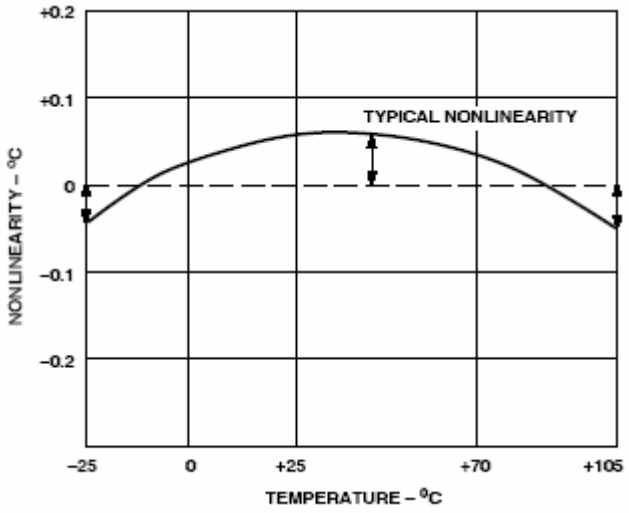
Pro AD592 jsou dány tři základní chybové limity, které umožňují vybrat správný stupeň pro jakoukoliv aplikaci při jakékoliv úrovni přesnosti. Je to nastavitelná přesnost při 25°C, teplotní chyba od 0°C do 70°C a teplotní chyba od –25°C do 105°C. Tyto tři charakteristiky mají vztah k aktuální chybě, kterou uživatel zjistí, pokud proudový výstup AD592 byl přeměněn na napětí přesným odporem. Všimněme si, že maximální chyba při pokojové teplotě, přes komerční IC rozsah teplot nebo rozšířené teplotní pásmo zahrnující bod varu vody, může být vyčtena přímo z tabulky charakteristik.Všechny tři chybové limity jsou kombinací počáteční chyby, změn převodního faktoru a nelineární odchylky od ideálního výstupu 1 µA/K. (Obr. 10) znázorňuje garantovanou přesnost pro AD592 CN.




*Obr. 10 Graf garantované přesnosti AD592CN*

AD592 má vysoce lineární výstup v porovnání se starší technologií senzorů (např. termistory, RTD a termočlánky), proto je chyba nelinearity určována odděleně od absolutní přesnosti, která je dána teplotou. Jako maximální odchylku od přímky tato charakteristika před-

stavuje jedinou chybu, která nemůže být odstraněna. (Obr. 11) zachycuje průběh typické nelinearity AD592 CN v rozsahu celého teplotního pásma.




*Obr. 11 Odchylka od linearity*

# II. PRAKTICKÁ ČÁST

# 3 VÝVOJOVÉ PROSTŘEDKY

Smyslem konstrukce archivačního teploměru je snaha vyvinout prostředek schopný snímat a uchovávat (popř. zpracovávat) teplotu odděleně a samostatně i několik dní, bez potřeby přístupu k získaným datům.

Prioritou pro vývoj konečné podoby bylo vytvoření výhodných finančních podmínek, při zachování vysoké kvality funkce výsledného výrobku. Pro přehlednější řešení lze problém separovat na několik dílčích:

- Nalezení vyhovujícího teplotního senzoru, pro klimatické podmínky mírného podnebního pásu (tedy přibližně v rozmezí teplot -20°C až 50°C). Požadovaným parametrům se svými vlastnostmi blížil polovodičový monolitický integrovaný teplotní senzor AD592, který se oproti alternativním řešením (s termočlánkem, odporovým snímačem) pohybuje v příhodné finanční relaci, při udržení všech klíčových atributů (rozsah teplot, linearita průběhu, odchylky).
- Nalezení vyhovujícího člena z řady mikrokontrolerů rodiny Motorola HC08. Byl zvolen MC68HC908QT4, tedy osmivývodový Nitron s vestavěným A/D převodníkem a 4096 KB paměti FLASH, který ovšem není standardně vybaven žádným komunikačním rozhraním.
- Vývojové zázemí pro vytvoření a test funkce výsledné aplikace.

## 3.1 Princip činnosti teploměru

Funkce mikrokontroleru vychází z prosté myšlenky. Analogově/číslicovým převodníkem, implementovaným na čipu, je v uživatelem přednastavených periodických časových intervalech (nejméně však 1 minuta, maximálně 255 minut) snímána hodnota vstupního signálu zpracovaného teplotním senzorem AD592 a upraveného příslušnou elektronikou. Přesné časování je zajištěno externím hodinovým krystalem o frekvenci 9,8304 MHz. Po každých čtyřech odebraných vzorcích, jsou hodnoty uloženy do paměti FLASH. Komunikačním prostředníkem mezi mikrokontrolerem a vyšším řídícím prostředkem (např. PC) je sériové rozhraní RS232, které musí být samostatně hardwarově zpřístupněno obvodem MAX232. Slouží k přenosu důležitých konstant a řídících signálů směrem k mikrořadiči a naměřených hodnot uložených v paměti FLASH z mikrořadiče.

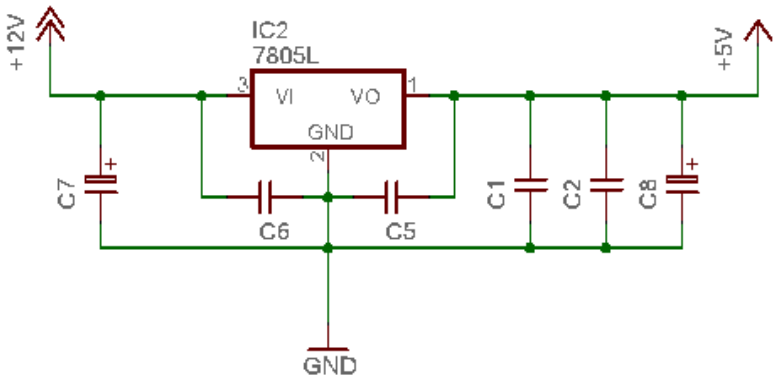
## 3.2 Startkit Janus

Vývojový kit Janus, na němž byla odladěna programová část aplikace, uvedla česká pobočka firmy Motorola na trh v roce 2003. Je koncipován jako velmi jednoduchý a snadno dostupný nástroj pro seznámení se s problematikou mikrokontrolerů Motorola 68HC08. Kit umožňuje vývoj software specificky pro podskupinu mikrokontrolerů HC08 „Nitron“, tedy 68HC908QT a QY. Lze jej velmi dobře využít pro demonstraci možností mikrokontroleru, pro návrh jednoduchých aplikací nebo pro malosériové programování mikrokontrolerů. Pokud je cílová aplikace s mikrokontrolerem 68HC08 vybavená odpovídajícím servisním konektorem (MON08, MINIMON, apod.), lze kit zároveň využít i jako vývojový a programovací adaptér.

*Obr. 12 Vývojový kit Janus*

Jelikož je finální produkt teploměru ve své podstatě spojením prvků plošné desky mikrokontroleru Nitron a plošné desky elektronického zázemí integrovaného teplotního senzoru AD592, je plošná deska využitá při konstrukci aplikace do značné míry zjednodušenou obdobou původního vývojového kitu Janus (přinejmenším využívá řadu shodných prvků), lze tedy jediným popisem názorně vysvětlit základní elektronické pochody obou takto soběpříbuzných zapojení. Mezi tři základní shodné části patří zdroj napájení, generátor externího hodinového signálu a rozhraní pro sériovou komunikaci. Vývojový kit Janus navíc obsahuje sadu jednoduchých periférií a konektor MINMON pro ladění softwaru v cílové aplikaci.

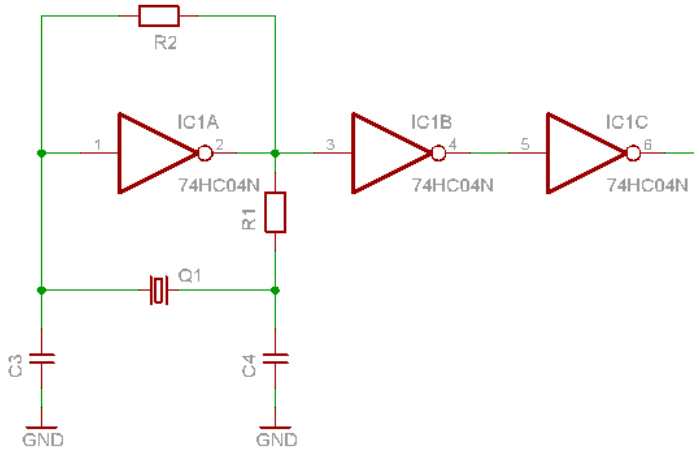
### 3.2.1 Napájecí napětí




*Obr. 13 Schéma stabilizátoru napájení*

Obvod je vnitřně napájen stejnosměrným napětím 5V, které vytváří stabilizátor 7805 ve standardním zapojení s filtračními kondenzátory, proti zkratovému odporu a nežádoucím kmitům z výstupu stabilizátoru. U desky kitu je navíc přívod vybaven ochranou diodou proti náhodnému přepólování a tlačítkem bez aretace pro krátkodobé odpojení mikrokontroleru od napájení.

### 3.2.2 Generátor hodinových kmitů

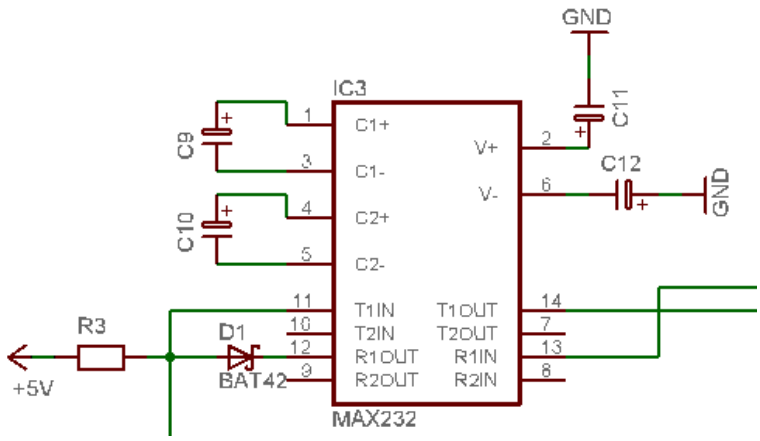



*Obr. 14 Schéma externího krystalového oscilátoru*

Mikrokontrolery Nitron jsou sice standardně vybaveny interním generátorem hodinových kmitů o frekvenci 12,8 MHz, avšak s nedostačující přesností pro práci v módu monitoru, kdy vyžadují lepší externí oscilátor, zejména pro dodržení správný přenosových rychlostí asynchronního sériového kanálu. Kit proto obsahuje jednoduchý krystalový oscilátor (Q1) s kmitočtem 9,8304MHz (v klasickém zapojení s invertorem 74HC04), který je teplomě-

rem využíván nejen při sériové komunikaci, ale také pro přesný odpočet minutové periody, jako základního úseku odebírání vzorků měřené teploty.

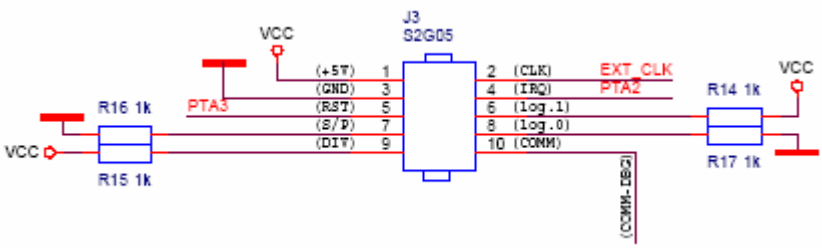
### 3.2.3 Sériové rozhraní RS232




*Obr. 15 Schéma integrovaného obvodu MAX232*

Pro připojení k PC slouží poloduplexní sériové rozhraní RS232 vycházející z obvyklého zapojení obvodu MAX202 (resp. MAX232 pro teploměr). MAX232 je šestnáctipinový obousměrný konvertor RS232/TTL logiky. Obvod má ke své činnosti zabudovaný napěťový měnič a invertor. Kromě napěťových úprav signálů TxD (vysílač), RxD (přijímač) a DTR (pouze v zapojení Janus – slouží k resetování mikrokontroleru z PC) se obvodu využívá ke generování jednoho z nezbytných signálů přepnutí do módu monitoru (zvýšené napětí 10V). Sériový komunikační kanál slouží aplikaci teploměru k přejímaní důležitých dat, řídících signálů a odesílání uchovávaných naměřených hodnot.

### 3.2.4 MINIMON



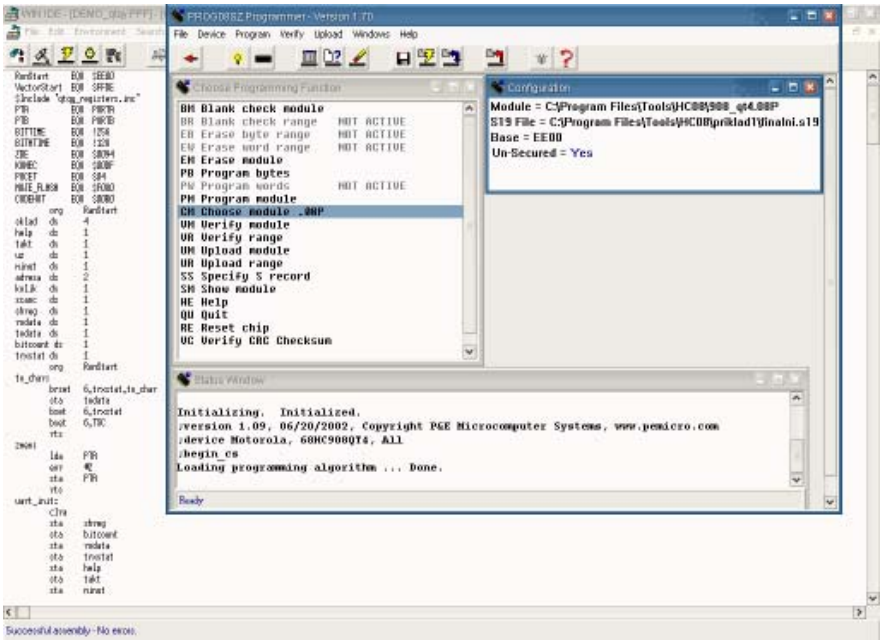

*Obr. 16 Schéma konektoru MINIMON*

V případě vývoje nebo úpravy software v cílové aplikaci lze kit Janus použít též jako programovací/ladicí adaptér mezi aplikaci a počítač. Pro tento případ je kit vybaven konekto-

rem MINIMON, který obsahuje všechny potřebné signály pro vývoj software v aplikaci. Pomocí vhodného kabelu lze připojit aplikaci vybavenou konektorem MON08, MINIMON, nebo uživatelem definovaným, na který jsou vyvedeny signály nezbytné pro ladění.

## 3.3 ICS08 a Viewcom

ICS08 je freeware firmy P&E Micro, obsahující kompletní vývojové prostředí assembleru pro všechny mikrokontrolery řady HC08. Kromě editoru a překladače, se základním upozorněním na chyby v kódu, obsahuje rovněž programátor (pro implementaci programu do paměti EEPROM nebo FLASH), simulátor (pro testování vytvořeného programu na PC), obvodový simulátor (pro simulování běhu programu s využitím obvodů reálného mikrořadiče – „In-Circuit Debugger“) a debugger (pro odlaďování chyb v programu).




*Obr. 17 Vývojové prostředí ICS08*

Testování funkčnosti programu bylo ověřováno za pomoci aplikace Viewcom. Viewcom je freeware firmy IMFsoft, sloužící pro snadný přístup k sériovým portům. Kromě běžných funkcí výběru portu, nastavení přenosové rychlosti, počtu datových bitů, parity a počtu stop bitů, nabízí také možnost komunikace až ve třech různých módech (Char – ASCII znaky, hexadecimální nebo desítková čísla).

### 3.3.1 Ověření základní funkčnosti zařízení

Při vývoji teploměru, musel být brán zřetel na fyzickou nepřítomnost konečného výrobku a tedy poskytnutí alternace ověření základních funkcí tohoto zařízení. Adekvátní pomůckou je kit Janus, který se tak stává nejen prostředkem programování mikrokontroleru, kterým bude osazena deska finálního plošného spoje, ale zároveň lze s výhodou využít přítomnosti jednoduchých periferií, jimiž je start kit vybaven. Během vytváření výsledné podoby se vyskytlo několik zásadních problémů.

Prvním je vysoká nepřesnost interního oscilátoru o frekvenci 12,8 MHz, který ani po doladění pomocí registru OCSTRIM nesplňoval požadované parametry. V důsledku rozsahu kolísání skutečného kmitočtu docházelo k nezanedbatelným odchylkám od požadované měřící periody. Proto je před začátkem činnosti vždy aktivován vyhovující přesnější externí oscilátor o frekvenci 9,8304 MHz.

Přesnost časování byla překontrolována v pro tento účel dostačujících domácích podmínkách, pomocí sledování a porovnávání času v předpokládaných měřících intervalech. Pro zlepšení rekonstrukce celkového výsledku, byla využita možnost sledování přepínání žluté LED diody (rozsvěcení a zhasínání při převodu A/D) pomocí webové kamery MSI StarCam 370i. Ve finální aplikaci je samozřejmostí využití osciloskopu.

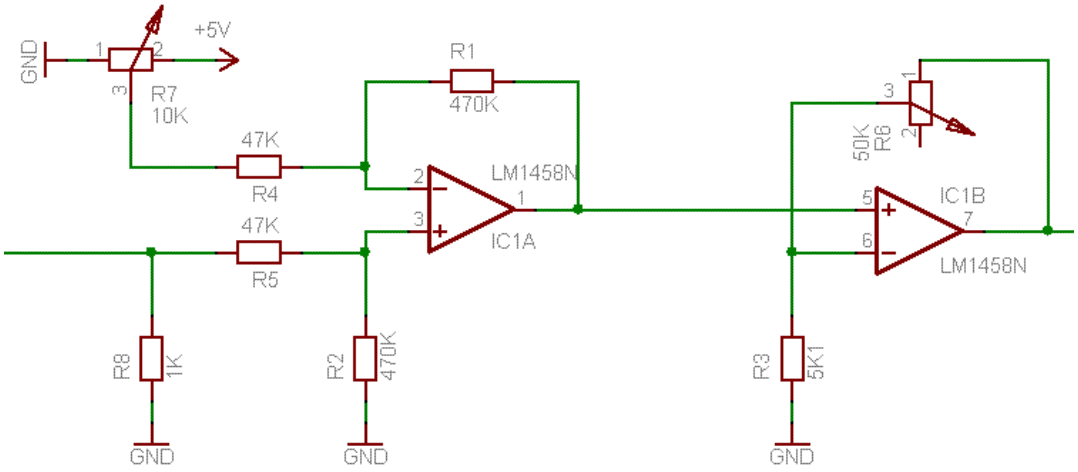
*Obr. 18 MSI StarCam 370i*

Dalším problémem je kontrola funkce A/D převodníku. U kitu Janus nelze zároveň přivést signál externího oscilátoru a napěťového děliče na převodník A/D, jelikož jsou přes jumper propojeny na shodný port. Proto byly vyzkoušeny tři varianty: s externím oscilátorem, jako zdrojem simulovaného měření teplot, s interním oscilátorem a napěťovým děličem, s interním oscilátorem a prototypem elektroniky obvodu teplotního senzoru.

Ke komunikace byly použity programy ViewCom a HyperTerminal OS Windows.

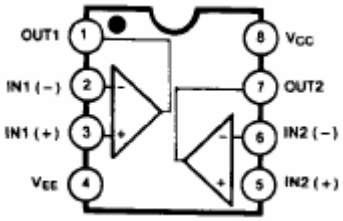
# 4 TEPLOMĚR SE SENZOREM AD592CN

Monolitický teplotní senzor AD592CN má k dispozici nejlepší parametry ze senzorů této série integrovaných obvodů. Chová se jako vysokoimpedanční zdroj proudu, úměrného teplotě s lineární převodní charakteristikou 1 µA/K. Analogově/číslicový převodník mikrokontroleru Nitron je kalibrován na hodnoty vstupního napětí 0 až 5V. Je tedy zjevné, že nelze přímo zpracovávat proudový signál generovaný teplotním čidlem. K přizpůsobení signálu napěťovému vstupu A/D převodníku mikrořadiče je použita samostatná deska s elektronikou na bázi operačních zesilovačů (s integrovaným obvodem LM1458N, se dvěma OZ).




*Obr. 19 Schéma zapojení operačních zesilovačů*

Jak je patrné z obrázku (Obr. 8 Závislost výstupního proudu na teplotě) v kapitole 2.2 AD592, není hodnota výstupního proudu při 0°C nulová, tj. existuje posun měřícího intervalu (0°C odpovídá 248 µA), proto je nutné nejen daný signál zesílit, ale také napěťově přizpůsobit jeho úroveň.

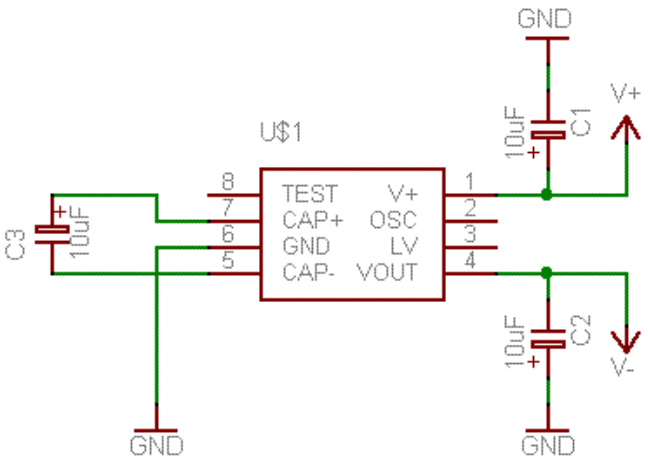



*Obr. 20 Zapojení LM1458N*

Nejprve se generovaný proudový signál teploměru přemění na napěťovém převodníku R8 na napětí. To je přivedeno na vstup operačního zesilovače IC1A v diferenciálním zapojení, jehož cílem je posun spodní limity měřícího intervalu do nuly a zesílení signálu (A = 10). Drobné odchylky způsobené napěťovou nesymetrií vstupů jsou dolaďovány nastavením trimru R7.

Takto upravený signál je potom přiváděn na vstup dalšího operačního zesilovače IC1B v neinvertujícím zapojení. Cílem je zesílit výsledný signál na hodnoty odpovídající napěťovému vstupu A/D převodníku mikrokontroleru Nitron. Napěťovou hladinu lze doladiť kalibračním trimrem R8.

Zdroj nesymetrického napájení operačních zesilovačů ±9V je zajištěn sice finančně nákladnějším, ale spolehlivým integrovaným monolitickým invertorem s nábojovou pumpou ICL7662.




*Obr. 21 Schéma integrovaného invertoru ICL7662*

# 5 SOFTWARE

Cílem programu je od uživatelem zvoleného časového okamžiku ukládat v periodických intervalech hodnoty teploty (resp. napětí) získané z teplotního senzoru AD592CN pomocí integrovaného A/D převodníku a jejich zpětné získání po době nezbytně nutné k samostatnému provozu archivačního teploměru (nejdéle však 60 dní v půlhodinových intervalech). Tato verze byla programována pro vývojový kit Janus, ve finální programu je nutné odstranit některé podprogramy informativního charakteru (práce s LED diodou) a změnit kanál A/D převodníku na port PTA1.

Po inicializaci vyčkává procesor ve smyčce na příchod řídícího znaku od nadřízeného prvku. Význam řídících znaků je následující:

- **$41 („A“)** – spuštění periodického snímaní teploty A/D převodníkem. Přičemž před začátkem měření očekává mikroprocesor přijmutí několik důležitých údajů v tomto pořadí (o velikosti 1B na každou hodnotu):
  - Počet minut na jednu periodu vzorkování
  - Počet minut do odebrání prvního vzorku
  - Aktuální stav hodin
  - Aktuální stav minut
  - Aktuální den
  - Aktuální měsíc

  Tyto údaje by měli být postačující pro zpětnou rekonstrukci data a hodiny získaných hodnot.
- **$42 („B“)** – čtení naměřených hodnoty z paměti FLASH. Data předávaná nadřazenému prvku ke zpracování, jsou hodnoty napětí naměřené A/D převodníkem, úměrné teplotě. Formát předávacího protokolu je ve tvaru:
  - **Hodina, minuty, den, měsíc** (1B na každou hodnotu)
  - **Naměřená data** (Dle rozsahu získaného měření, 1B na každou hodnotu)
  - **$FF**, hodnota prázdné buňky, informace o ukončení přenosu
- **$43 („C“)** – smazání hodnot uložených v paměti FLASH.

Důkladnější rozbor jednotlivých částí programu uvádí následující kapitola.

## 5.1 Teplomer.asm

```
1  RAMStart     EQU  $0080
2  RomStart     EQU  $EE00
3  VectorStart  EQU  $FFDE
4  $Include     'qtqy_registers.inc'
5  PTA          EQU  PORTA
6  PTB          EQU  PORTB
7  BITTIME      EQU  !256
8  BITHTIME     EQU  !128
9  ZDE          EQU  $0094
10 KONEC        EQU  $00DF
11 POCET        EQU  $04
12 MOJE_FLASH   EQU  $F080
13 CODEHOT      EQU  $00B0
```

Definice konstant, začátku paměťových bloků zápisníku, programu a důležitých registrů s využitím externího souboru qtqy_reigistres.inc, který je součástí elektronické přílohy.

```
14         org     RamStart
15 sklad   ds      4
16 help    ds      1
17 takt    ds      1
18 uz      ds      1
19 minut   ds      1
20 adresa  ds      2
21 kolik   ds      1
22 scanc   ds      1
23 shreg   ds      1
24 rxdata  ds      1
25 txdata  ds      1
26 bitcount ds     1
27 trxstat ds      1
```

Definice proměnných v zápisníku od adresy RamStart. Primární význam jednotlivých proměnných je následující:

- **sklad**, je paměťové místo pro ukládání mezivýsledků jednotlivých měření teplot (jednou za čtyři periody jsou hodnoty uloženy do paměti FLASH).

- **help**, je pomocný příznakový registr, přepínání režimu sériové komunikace a periodického snímání teploty.
- **takt**, pomocná proměnná, sloužící k odladění času jedné minuty.
- **uz**, určuje kolik minut zbývá do dalšího převodu
- **minut**, počet minut na jednu periodu snímání teploty
- **adresa**, dvoubytová proměnná, k ukládání aktuální adresa paměti FLASH, od níž se budou ukládat naměřené hodnoty
- **kolik**, počet bytů, které mají být uloženy do paměti FLASH
- **scanc**, pomocný buffer sériového kanálu
- **shreg**, posuvný registr pro příjem/vysílání dat
- **rxdata**, přijatá data
- **txdata**, data určená k odvysílání
- **bitcount**, rx/tx bitový čítač odvysílaného rámce (1 startbit, 2-9 data, 10 stopbit)
- **trxstat**, stavový registr sériového kanálu (bit 7 – probíhá vysílání TXA, bit 6 - txdata je naplněn, bit 4 – přijímač je zahlcen daty, bit 3 – přetečení příjmu, předchozí data přijata nebo se ztratila RO, bit 2 – přijímač aktivní RXA, bit 1 – chyba přenosu, framing error FA)

```
28          org     RomStart
29  tx_char:
30          brset   6,trxstat,tx_char
31          sta     txdata
32          bset    6,trxstat
33          bset    6,TSC
34          Rts
```

Podprogram tx_char, zabezpečující odesílání znaku. Před začátkem vysílání je položen dotaz na připravenost vysílače. Je-li vytížen, čeká se tak dlouho, dokud není k dispozici. Poté je naplněn registr dat k odvysílání a nastaven příznak vysílání.

```
35  zmen:
36          lda     PTA
37          eor     #2
38          sta     PTA
```

```
39          Rts
```

Podprogram zmen má smysl pouze v odlaďovací verzi. Používá se k invertování stavu žluté diody vývojového kitu Janus, připojené na vývod PTA1. Rozsvěcení a zhasínání diody má informativní charakter.

```
40 uart_init:
41          clra
42          sta     shreg
43          sta     bitcount
44          sta     rxdata
45          sta     trxstat
46          sta     help
47          sta     takt
48          sta     minut
49          sta     uz
50          mov     #$30,TSC
51          ldhx    #BITTIME
52          sthx    TMODH
53          bclr    7,TSC
54          mov     #1,KBIER
55          bset    0,PTAPUE
56          clr     KBSCR
57          bclr    0,DDRA
58          Rts
```

Inicializace sériového rozhraní UART. Nastavení časovače na hodnotu délky jednoho bitu, aby byla dodržena přenosová rychlost 9600 Bd. Časovač je vynulován a pozastaven. Aktivováno vnější přerušení KBI na portu PTA0.

```
59 casuj_init:
60          mov     #$76,TSC
61          mov     #$63,TMODH
62          mov     #$EE,TMODL
63          bclr    5,TSC
64          Rts
```

Podprogram inicializace časovače pro odpočet periody snímání měřené teploty. Základní frekvence sběrnice, odvozená z externího krystalu 2,4576 MHz je dělena přednastavenou předděličkou, hodnotou 64 a časovač inicializován hodnotou $63EE. Časovač je odladěn pro přesné časování intervalu jedné minuty (s ohledem na vykonávané instrukce během přerušení).

```
65 main:
66          rsp
67          clrx
68          clra
69          mov     #$31,CONFIG1
70          mov     #$8,CONFIG2
```

Nastavení konfiguračních registrů. COP a LVI je vypnuto, bude se pracovat s externím oscilátorem 9,8304 MHz. [7]

```
71          ldx     #$10
72 cekej:
73          dbnza   cekej
74          decx
75          cpx     #$0
76          bne     cekej
77          bset    1,OSCSTAT
78 cekej2:
79          brclr   0,OSCSTAT,cekej2
```

Čekací smyčka slouží k ustálení externího oscilátoru. Poté je nastavením příznakového bitu registru OSCSTAT aktivován a vnitřní generátor hodin naopak vypnut. Čeká se do té doby, není-li externí oscilátor v provozu.[7].

```
80 sem:
81          lda     #$FF
82          sta     PTA
83          sta     PTB
84          mov     #$02,DDRA
85          mov     #0,DDRB
86          mov     #$1C,PTAPUE
87          bsr     uart_init
88          cli
89 main_loop:
90          bsr     znak
91          cbeqa   #$41,casuj
92          cbeqa   #$42,cti
93          cbeqa   #$43,smaz
94          bsr     zmen
95          bra     main_loop
```

Čekací smyčka příchodu řídícího signálu. CPU pracuje v režimu WAIT do okamžiku vnějšího přerušení od sériové komunikační linky. Podle příchozího znaku se rozhoduje o ope-

raci, jež bude vykonávána. $41 – start periodického snímání teploty. $42 – načtení naměřených hodnot, uložených v paměti FLASH. $43 – vymazání uchovávaných dat z paměti FLASH.

```
 96 casuj:
 97         jsr     znak
 98         sta     minut
 99         jsr     znak
100         sta     uz
101         jsr     znak
102         sta     sklad
103         jsr     znak
104         sta     sklad+1
105         jsr     znak
106         sta     sklad+2
107         jsr     znak
108         sta     sklad+3
```

Předání důležitých parametrů mikrokontroleru, předává se: počet minut na jednu periodu snímání (maximálně 255), počet minut do odebrání prvního vzorku (maximálně 255) a nastavení aktuálního času ve formátu hodina, minuty, den, měsíc, které jsou zároveň prvními uloženými daty v paměti FLASH. Předpokládá se, že tyto hodnoty předá mikrořadiči nadřízený prvek (např. PC).

```
109         ldhx    #MOJE_FLASH
110         sthx    adresa
111         ldhx    #fwrite
112         jsr     kopiruj
113         mov     #POCET,kolik
114         jsr     ZDE
115         bset    1,help
116         bsr     casuj_init
```

Do proměnné adresa je načtena počáteční adresa paměti FLASH, od níž se začnou ukládat naměřená data. Jelikož nutná rutina pro obsluhu paměti FLASH musí být vykonávána z paměti RAM (nelze zároveň zapisovat do FLASH a číst program), převezme podprogram kopíruj přes indexový registr počáteční adresu podprogramu ukládání dat do paměti a pak celý podprogram překopíruje do RAM od adresy $94 a zavolá pro uložení dat získaných přenosem ze sériové linky (datum a čas). Následně zapne je zapnut časovač pro odpočet periody snímání teploty.

```
117 smycka:
118         wait
119         brset   2,trxstat,zpet
120         bra     smycka
```

Zde se očekává přerušení od časovače, sériového kanálu (návrat do hlavní čekací smyčky s volbou podprogramu) nebo A/D převodníku.

```
121 zpet:
122         bclr    1,help
123         bra     sem
124 znak:
125         wait
126         brclr   4,trxstat,znak
127         lda     rxdata
128         bclr    4,trxstat
129         Rts
```

Podprogram znak, přepíná jádro do režimu WAIT. Čeká se do příchodu znaku po sériové lince.

```
130 cti:
131         ldhx     #MOJE_FLASH
132 jeste:
133         mov      x+,kolik
134         lda      kolik
135         jsr      tx_char
136         cbeqa    #$FF,main_loop
137         bra      ještě
```

Podprogram cti má na starosti rutinu načítání a odesílání naměřených hodnot z paměti FLASH. Na počátku je načtena první adresa uložených dat, vybírá se tak dlouho, není-li načtená hodnota $FF, tedy hodnota prázdné paměťové buňky. Touto hodnotou je ukončen sériový přenos.

```
138 smaz:
139         ldhx    #ferase
140         jsr     kopiruj
141         ldhx    #MOJE_FLASH
142         sthx    adresa
143         mov     #$46,kolik
144 zvysuj:
145         jsr     ZDE
146         aix     #64
```

```
147         dec     kolik
148         lda     kolik
149         cmp     #$0
150         bne     zvysuj
151         Rts
```

Podprogram smaz, maže po 64 B blocích pevný úsek paměti FLASH od počáteční ukládací adresy, předáváním parametru v proměnné adresa a voláním procedury ferase, která obsluhuje příslušnou rutinou FLASH.

```
152 kopiruj:
153         mov     #ZDE,kolik
154 more:
155         mov     x+,help
156         lda     help
157         pshx
158         pshh
159         clrh
160         ldx     kolik
161         sta     ,x
162         pulh
163         pulx
164         inc     kolik
165         lda     kolik
166         cbeqa   #KONEC,dal
167         bra     more
168 dal:
169         clr     kolik
170         clr     help
171         Rts
```

Podprogram sloužící k překopírování části programového kódu do paměti RAM. Počáteční adresa se předává v indexovém registru X. Kopíruje se do pevně zvoleného úseku $94 až $DF.

```
172 fwrite:
173         ldhx    #FLCR
174         lda     ,X
175         ora     #1
176         sta     ,X
177         lda     FLBPR
178         sta     MOJE_FLASH
```

Povolení programování paměti FLASH a zapsání libovolné hodnoty na počáteční adresu.

```
179         lda     #10
180 DEL_1:
181         dbnza   DEL_1
182         lda     ,X
183         ora     #8
184         sta     ,X
185         lda     #5
```

Nastavení příznaku vysokého příkonu pro programování FLASH.

```
186 DEL_2:
187         dbnza   DEL_2
188         ldhx    adresa
189 DALSI_B:
190         mov     sklad,X+
191         inc     CODEHOT
192         lda     #31
```

Data z proměnné sklad se kopírují do paměti FLASH, od adresy v indexovém registru. Je využito dynamické změny programu za chodu, kdy je inkrementována aktuální adresa proměnné sklad, odkud se bude vybírat.

```
193 DEL_3:
194         dbnza   DEL_3
195         sthx    adresa
196         dec     kolik
197         lda     kolik
198         cmpa    #$0
```

V proměnné kolik se předává počet ukládaných bitů.

```
199         bne     DALSI_B
200         mov     #sklad,CODEHOT
201         ldhx    #FLCR
202         lda     ,X
203         and     #$FE
204         sta     ,X
205         lda     #5
206 DEL_4:
207         dbnza   DEL_4
208         lda     ,X
209         and     #$F7
210         sta     ,X
211         Rts
```

Nezbytné obnovení výchozích hodnot dynamického registru CODEHOT a vyčištění příznakových bitů.

```
212 ferase:
213         ldhx    #FLCR
214         lda     ,X
215         ora     #2
216         and     #$FB
217         sta     ,X
218         ldhx    adresa
219         lda     FLBPR
220         sta     ,x
```

Obdobně jako u podprogramu fwrite, nastavení příznakového bitu mazání paměti FLASH, ale hodnota registru FLBPR musí být zapsána na první pozici mazaného 64 B bloku. Adresa mazaného bloku se předává parametrem uloženým v proměnné adresa.

```
221         ldhx    #FLCR
222         lda     #10
223 DEL_8:
224         dbnza   DEL_8
225         lda     ,X
226         ora     #8
227         sta     ,X
228         lda     #4
```

A povolení příznaku vysokého příkonu pro vymazání FLASH.

```
229 DEL_9:
230         dbnza   DEL_9
231         ldhx    #FLCR
232         lda     ,X
233         and     #$FD
234         sta     ,X
235         lda     #5
236 DEL_10:
237         dbnza   DEL_10
238         lda     ,X
239         and     #$F7
240         sta     ,X
241         Rts
```

Uvedení příznakových bitů pro práci s FLASH do původního stavu.

```
242 adc_isr:
```

```
243          jsr     zmen
244          lda     kolik
245          ldhx    #sklad
246          cbeqa   #$0,hop
247 uloz:
248          incx
249          deca
250          cmp     #$0
251          bne     uloz
```

Do indexového registru se vloží počáteční adresa odkládacího bloku proměnné sklad a inkrementuje se tak dlouho, dokud není ukazatel nad aktuálním bytem, určeným pro uložení hodnoty získané měření převodníku A/D.

```
252 hop:
253          lda     ADR
254          sta     ,x
255          inc     kolik
256          lda     kolik
257          cbeqa   #POCET,spust
258          Rti
```

Jestliže počet naplněných bytu je čtyři zavolej rutinu ukládání do paměti FLASH. Jinak proběhne návrat z přerušení.

```
259 spust:
260          bset    5,TSC
261          ldhx    adresa
262          cphx    #$FBE0
263          bhi     tu
264          jsr     ZDE
```

Zabezpečení, aby volání podprogramu fwrite, uloženého v paměti RAM mohlo uskutečnit pouze nebyla-li překročena limitní adresa a vypnutí časovače po dobu nutnou k vykonání této rutiny.

```
265 tu:
266          bclr    5,TSC
267          Rti
```

Návrat z obslužného přerušení dokončení převodu A/D převodníku. Přerušení je voláno jednou za periody snímání měřené teploty.

```
268 bit_isr:
269          mov     PTA,scanc
```

```
270         bclr    7,TSC
271         brclr   1,help,přes
```

Kontrola pomocného příznakového bitu proměnné help. Jestliže je příznak vynulován, jde o přenos sériového kanálu a pokračuje se dále. V opačném případě se vykonává obsluha periodického snímání teploty.

```
272         inc     takt
273         lda     uz
274         cbeqa   #$0,preved
275         lda     takt
276         cbeqa   #$59,pricti
277         Rti
```

Přesný odpočet jedné minuty (závislý na správném nastavení časovače). Proměnná takt se inkrementuje tak dlouho, neuběhla-li právě jedna minuta (v úvahu se bere také délka vykonávání instrukcí přerušení). Zároveň se provádí kontrola proměnné uz, signalizující periodu snímání měřené teploty.

```
278 pricti:
279         dec     uz
280         clr     takt
281         Rti
```

Jestliže uběhla právě jedna minuta, dekrementuje se proměnná uz. Je-li hodnota proměnné uz nulová je aktivováno spuštění A/D převodníku.

```
282 preved:
283         mov     #$70,ADICLK
284         mov     #$43,ADSCR
285         clr      takt
286         mov      minut,uz
287         Rti
```

Nastavení obslužných registrů A/D převodníku. Inicializován je třetí kanál na portu PTA5 (ve finální verzi první kanál na portu PTA1). Proběhne jeden nekontinuální převod 1/16 frekvence interní sběrnice, s voláním přerušení po dokončení (podrobnosti uvádí [7], s.151).

```
288 pres:
```

Začátek obslužné rutiny sériového kanálu.

```
289         lda     bitcount
290         tsta
291         brset   2,trxstat,bit_isr_rcv
```

Jestliže je nastaven příznakový bit přijímače, pak přejdi do přijímací části.

```
292         beq     bit_isrtx_startbit
293         cmpa    #$0A
294         beq     bit_isrtx_stoptrans
```

Jestliže bit je start bit nebo stop bit, zavolej příslušnou rutinu.

```
295         brset   0,shreg,bit_isrtx_log1
296         bclr    0,PTA
297         bset    0,DDRA
298         bclr    0,PTAPUE
299         bra     bit_isrtx_fin
300 bit_isrtx_log1:
301         bclr    0,DDRA
302         bset    0,PTAPUE
```

Aktivuje vysílací port a rozhodne, jestli je odesílaný bit nulový nebo jedničkový.

```
303 bit_isrtx_fin:
304         sec
305         ror     shreg
306         inc     bitcount
307         Rti
```

Rotuje odesílaná data o jeden bit dále (na přenos dalšího bitu).

```
308 bit_isrtx_startbit:
309         clr     KBIER
310         bclr    0,PTA
311         bset    0,DDRA
312         pshh
313         ldhx    #BITTIME
314         sthx    TMODH
315         pulh
316         mov     txdata,shreg
317         bset    7,trxstat
318         bclr    6,trxstat
319         mov     #1,bitcount
320         rti
```

Inicializuje přenos nového znaku, znepřístupní přerušení KBI externího zdroje a přivede na výstupu portu 0, tj. začátek vysílání (start bit). Poté proběhne nastavení rychlosti přenosu jednoho bitu, naplnění vysílacího registru daty a oznámení start bitu.

```
321 bit_isrtx_stoptrans:
322         brset   6,trxstat,bit_isrtx_startbit
```

```
323         bclr    7,trxstat
324         bra     bit_isrrx_recvend
```

Dokončení přenosu. Jestliže nejsou žádná další data k dispozici, ukončí vysílání.

```
325 bit_isr_rcv:
326         beq     bit_isrrx_startbit
327         cmpa    #$09
328         beq     bit_isrrx_stopbit
329         brset   0,scanc,bit_isrrx_bit
330 bit_isrrx_bit:
331         ror     shreg
332         inc     bitcount
333         Rti
```

Zjistí jestli se jedná o start nebo stop bit a zavolá příslušnou rutinu. V opačném případě rotuje datový registr o jeden bit dále.

```
334 bit_isrrx_stopbit:
335         brclr   0,scanc,bit_isrrx_nostop
336         mov     shreg,rxdata
337         brclr   4,trxstat,bit_isrrx_noover
338         bset    3,trxstat
339 bit_isrrx_noover:
340         bset    4,trxstat
341         bra     bit_isrrx_recvend
342 bit_isrrx_nostop:
343         bset    2,trxstat
344         bra     bit_isrrx_recvend
```

Jsou přijatá data skutečně stop bitem? Pokud ano, inicializuje přijímač k dalšímu přenosu. Jestliže ne, nastaví příznak chyby a připraví přijímač k dalšímu přenosu.

```
345 bit_isrrx_startbit:
346         brset   0,scanc,bit_isrrx_recvend
347         pshh
348         ldhx    #BITTIME-1
349         sthx    TMODH
350         pulh
351         mov     #1,bitcount
352         Rti
```

Když je přijímaný znak start bitem, provede nastavení rychlosti přenosu jednoho bitu.

```
353 bit_isrrx_recvend:
354         bclr    2,trxstat
```

```
355         brset   6,trxstat,bit_isr_idle
356         lda     TSC
357         and     #$3F
358         sta     TSC
```

Kontrola přítomnosti dalších přijímaných dat. Pokud ne, vypne časovač.

```
359 bit_isr_idle:
360         bset    0,KBIER
361         bset    2,KBSCR
362         bclr    0,DDRA
363         clr     bitcount
364         Rti
```

Obnovení externího přerušení KBI a vynulování čítače bitů.

```
365 rxstart_isr:
366         mov     #$50,TSC
367         bclr    0,KBIER
368         pshh
369         ldhx    #BITHTIME-30
370         sthx    TMODH
371         pulh
372         bclr    7,TSC
373         clr     bitcount
374         clr     shreg
375         bset    2,trxstat
376         Rti
```

Začátek příjmu sériové linky. Byla.li detekována sestupná hrana, přerušením KBI, je inicializována rychlost přenosu čítače, reset a nastaven příznak příchozího znaku.

```
377 dummy_isr:
378         stop
```

Chybné přerušení.

```
379         org     VectorStart
380         dw      adc_isr         ; FFDE - ADC Conversion Complete
381         dw      rxstart_isr     ; FFE0 - Keyboard Vector
382         dw      dummy_isr       ; FFE2 - Reserved
383         dw      dummy_isr       ; FFE4 - Reserved
384         dw      dummy_isr       ; FFE6 - Reserved
385         dw      dummy_isr       ; FFE8 - Reserved
386         dw      dummy_isr       ; FFEA - Reserved
387         dw      dummy_isr       ; FFEC - Reserved
```

```
388         dw      dummy_isr       ; FFEE - Reserved
389         dw      dummy_isr       ; FFF0 - Reserved
390         dw      bit_isr         ; FFF2 - TIM Overflow Vector
391         dw      dummy_isr       ; FFF4 - TIM Channel 1 Vector
392         dw      dummy_isr       ; FFF6 - TIM Channel 0 Vector
393         dw      dummy_isr       ; FFF8 - Reserved
394         dw      dummy_isr       ; FFFA - IRQ1 Vector
395         dw      dummy_isr       ; FFFC - SWI Vector
396         dw      main            ; FFFE - Reset Vector
```

Tabulka vektorů přerušení, pro přehlednost jsou zachovány původní poznámky.

# ZÁVĚR

Nosnou myšlenkou této práce bylo vytvoření provozuschopného prototypu archivačního teploměru, elektronického obvodu s řídící jednotkou a teplotním senzorem, upotřebitelného prakticky k pravidelnému sledování změn teploty v klimatických podmínkách mírného podnebního pásu a popisu jeho technického zázemí. Výsledky také spontánně vyústily k existenci hodnotného průvodce základní problematikou mikrokontrolerů rodiny Motorola HC08 a tématiky jednočipových mikropočítačů obecně.

Finální verzi teploměru tvoří dva oddělené plošné spoje. Prvním je elektronika upravující proudový výstupní signál polovodičového monolitického integrovaného obvodu teplotního senzoru AD592CN na vstupní napěťovou úroveň A/D převodníku mikrokontroleru, kterým je osazená druhá deska. Ta kromě patice mikrořadiče obsahuje sadu periférií tvořících funkční zázemí aplikace (přesný externí krystal pro udržení časových intervalů snímání teplot a přenosových rychlostí sériového rozhraní a obvod logiky sériové linky MAX232).

Navrhované řešení archivačního teploměru, vycházející z uplatnění nejmenšího zástupce osmibitových mikropočítačů Motorola HC08 - Nitron, zcela jistě není jediným možným volitelným postupem a nevyčerpává plně potenciál této vývojové větve. Důraz při vývoji byl kladen především na vytvoření příznivé finanční hladiny a energeticky šetrné spotřeby výsledného modulu (tak, aby mohl pracovat s akumulátorovým zdrojem i stovky hodin), nicméně při zachování vysoké úrovně správného plnění své funkce. Nespornou výhodou je tedy kompaktnost celého systému.

Ovšem nekomplikovanost takové konstrukce musí být nepochybně vykoupena nějakým nedostatkem. Největší nevýhodou navržené jednotky je její přílišná závislost na řídících signálech nadřízeného prvku (např. PC). Což v praxi znamená, že nedokáže zahájit svou činnost samostatnou inicializací z vlastních periférií, ale čeká na příslušný pokyn (příchozí znak). Patrně by lépe požadavkům soběstačného modulu vyhovovalo použít některého z robustnějších kolegů série HC08, vybaveného budičem LCD displeje, jednotkou hodin reálného času a tlačítky pro nastavení. Komunikační kanál by tak nesloužil jako prostředek oživení funkce teploměru, ale pouze jako prostředník přejímání naměřených dat a zároveň by existovala možnost jejich přímé vizualizace na implementovaném LCD panelu.

Takto vybavený obvod ale může být podnětem k další samostatné práci.

# SEZNAM POUŽITÉ LITERATURY

[1] VÁŇA, Vladimír. *Začínáme pracovat s mikrokontroléry Motorola HC08 Nitron*. 1. vyd. Praha: BEN, 2003. 96 s. ISBN 80-7300-124-1

[2] ROZEHNAL, Zdeněk. *Mikrokontroléry Motorola HC11*. 1. vyd. Praha: BEN, 2001. 192 s. ISBN 80-86056-77-5

[3] VALNÍČEK, Tomáš. *Měřící systém pro dům s ekologickými prvky*. Brno: Vysoké učení technické v Brně, 2001. 74 s. VUT-EU-ODDI-3302-21-01

[4] HRUŠKA, František. *Technické prostředky automatizace II*. 2. vyd. Zlín: Univerzita Tomáše Bati, Fakulta aplikované informatiky, 2006. 112 s. ISBN 80-7318-397-8

[5] HRUŠKA, František. *Technické prostředky automatizace III: (senzory, jejich principy a funkce)*. 2. vyd. Zlín: Univerzita Tomáše Bati, Fakulta aplikované informatiky, 2005. 118 s. ISBN 80-7318-315-3

[6] MECHLOVÁ, Erika. *Výkladový slovník fyziky pro základní vysokoškolský kurz fyziky*. 1. vyd. Praha: Prométheus, 2001. 588 s. ISBN 80-7196-151-5

[7] MC68HC908QT4 Data sheet. Motorola, 2002.

[8] CPU08 Central Processor Unit Reference Manual. Motorola, 2001.

[9] Vývojový kit Janus uživatelská manuál. Motorola, 2003.

[10] DOLÍVKA, Lukáš. *Semestrální práce do předmětu Speciální číslicové systémy: Motorola HC08.* [cit. 2006-05-20]. Praha: ČVUT, 2003. Dostupná z WWW:

<http://noel.feld.cvut.cz/vyu/scs/prezentace2004/Motorola-HC08/hc08.pdf>

[11] Programování mikropočítače 68HC11 v assembleru. [cit. 2006-05-20]. Dostupný z WWW:

<http://vyuka.ft.utb.cz/file.php?file=/53/Assembler_registry.pdf>

# SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|---|---|
| BCD | Binary Coded Decimal |
| CISC | Complex Instruction Set Computer |
| COP | Computer Operating Properly |
| CPU | Central Procesor Unit |
| EEPROM | Electrically Eraseable Programable Read Only Memory |
| HCMOS | High Speed Complementary Metal Oxid Semiconductur |
| LCD | Liquid Crystal Display |
| LIFO | Last-In First-Out |
| LVI | Low-Voltage Inhibit |
| NTC | Negative Temperature Coefficient |
| PC | Personal Computer |
| PDIP | Plastic Dual In-line Package |
| PTC | Positive Temperature Coefficient |
| PWM | Pulse Width Modulation |
| RAM | Random Access Memory |
| ROM | Read Only Memory |
| RWM | Read Write Memory |
| SCI | Serial Comunication Interface |
| SIM | System Integration Module |
| SOIC | Small Outline Integrated Circuit |
| SPI | Serial Peripheral Interface |
| TSSOP | Thin Shrink Small Outline Package |
| UART | Universal Asynchronnous Receiver Transmiter |

# SEZNAM OBRÁZKŮ

Obr. 1 Značení mikrokontrolerů HC08 ..... 11
Obr. 2 Blokové schéma MC68HC908QT/QY ..... 12
Obr. 3 Zapouzdření MC68HC908QY1 ..... 12
Obr. 4 Pracovní registry Nitronu ..... 13
Obr. 5 Paměťová mapa mikrokontrolerů Nitron ..... 16
Obr. 6 Synchronní přenos ..... 20
Obr. 7 Asynchronní přenos ..... 21
Obr. 8 Závislost výstupního proudu na teplotě ..... 29
Obr. 9 Průběh napětí a proudu v obvodu při 25°C a extrémních teplotách ..... 30
Obr. 10 Graf garantované přesnosti AD592CN ..... 31
Obr. 11 Odchylka od linearity ..... 32
Obr. 12 Vývojový kit Janus ..... 35
Obr. 13 Schéma stabilizátoru napájení ..... 36
Obr. 14 Schéma externího krystalového oscilátoru ..... 36
Obr. 15 Schéma integrovaného obvodu MAX232 ..... 37
Obr. 16 Schéma konektoru MINIMON ..... 37
Obr. 17 Vývojové prostředí ICS08 ..... 38
Obr. 18 MSI StarCam 370i ..... 39
Obr. 19 Schéma zapojení operačních zesilovačů ..... 40
Obr. 20 Zapojení LM1458N ..... 40
Obr. 21 Schéma integrovaného invertoru ICL7662 ..... 41

# SEZNAM TABULEK

Tab. 1 Typy provedení MC68HC908QT/QY ..... 16
Tab. 2 Tabulka vektorů přerušení mikrokontroleru Nitron ..... 18

# SEZNAM PŘÍLOH

P I Schéma zapojení senzoru AD592CN

P II Schéma plošného spoje senzoru AD592CN

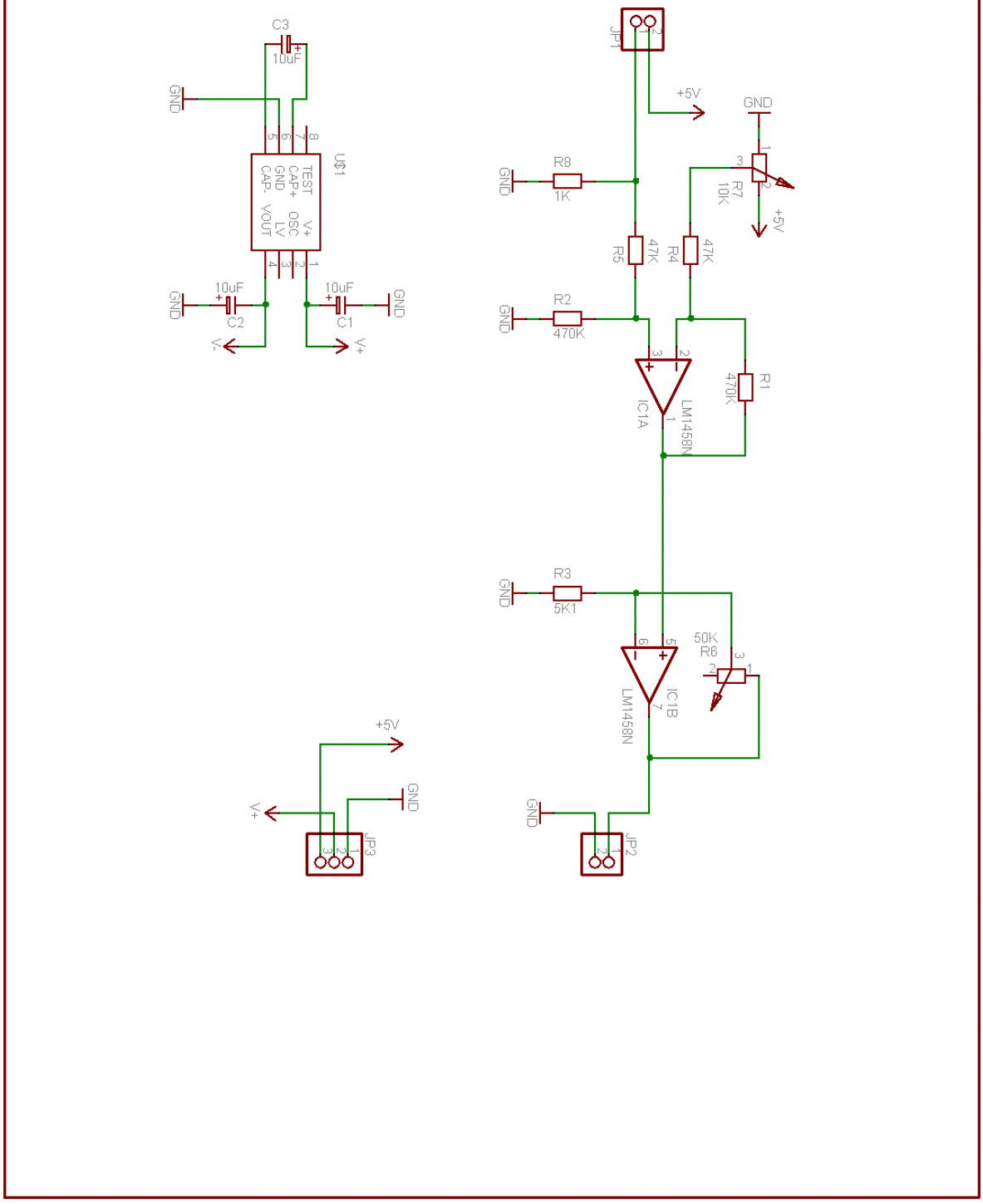
P III Schéma zapojení MCU MC68HC908QT4

P IV Schéma plošného spoje MCU MC68HC908QT4

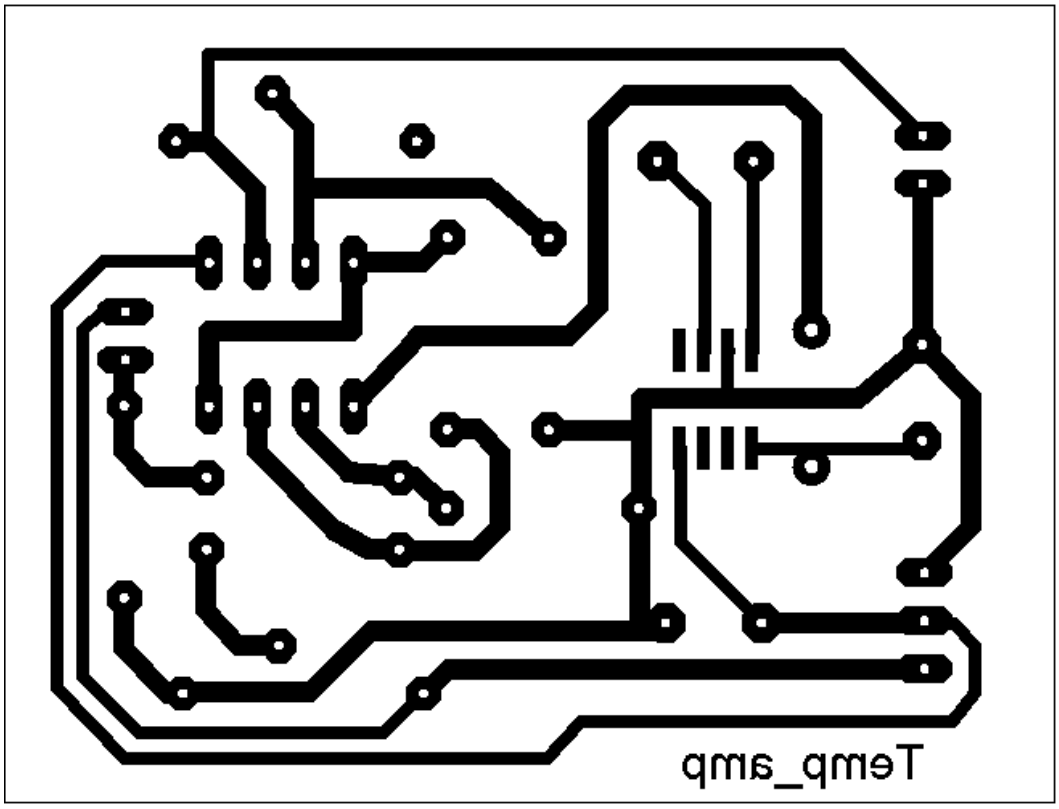
P V Teplotní specifikace senzoru AD592

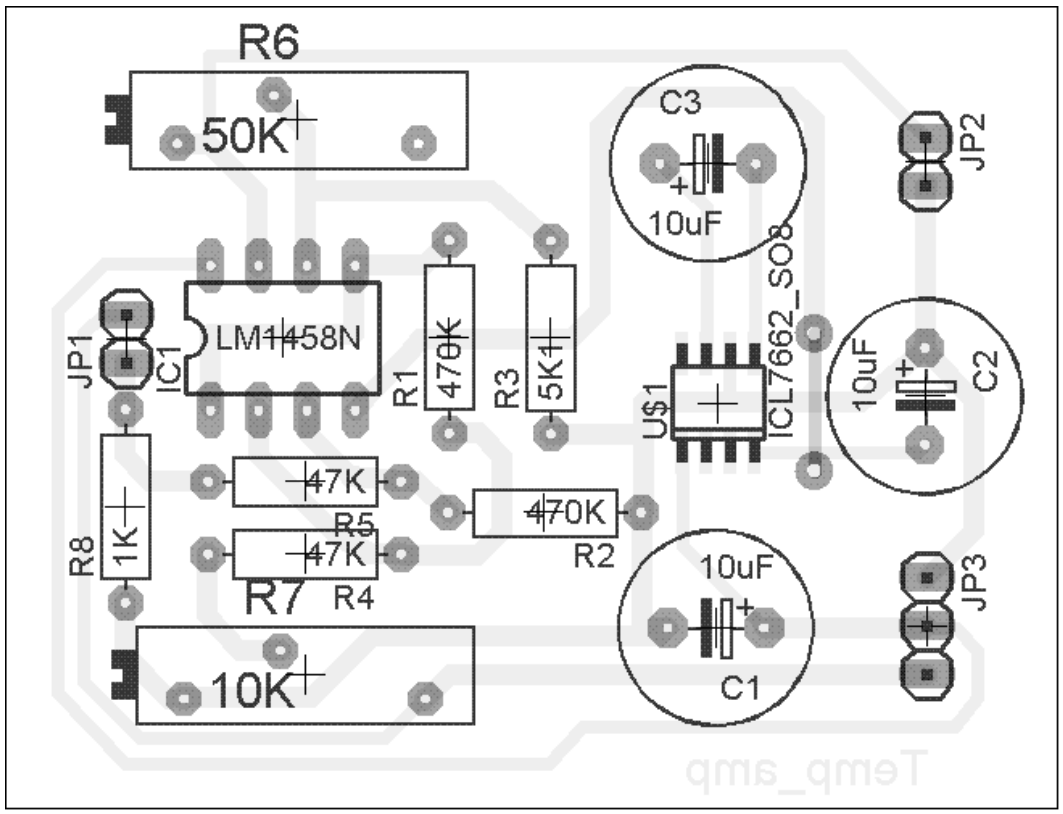
P VI Elektronické přílohy a verze bakalářské práce na CD

# PŘÍLOHA P I: SCHÉMA ZAPOJENÍ SENZORU AD592CN

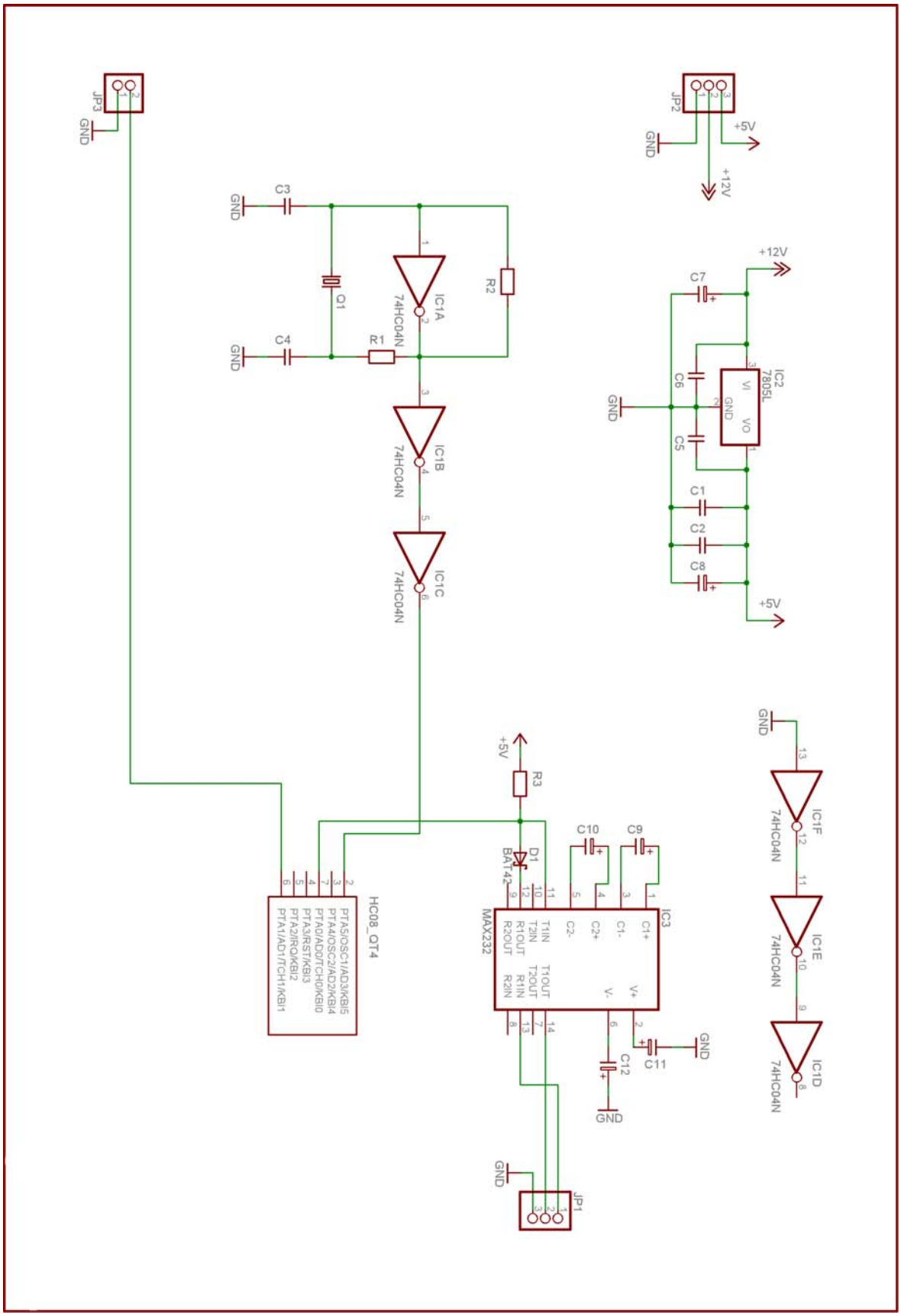


# PŘÍLOHA P II: SCHÉMA PLOŠNÉHO SPOJE SENZORU AD592CN

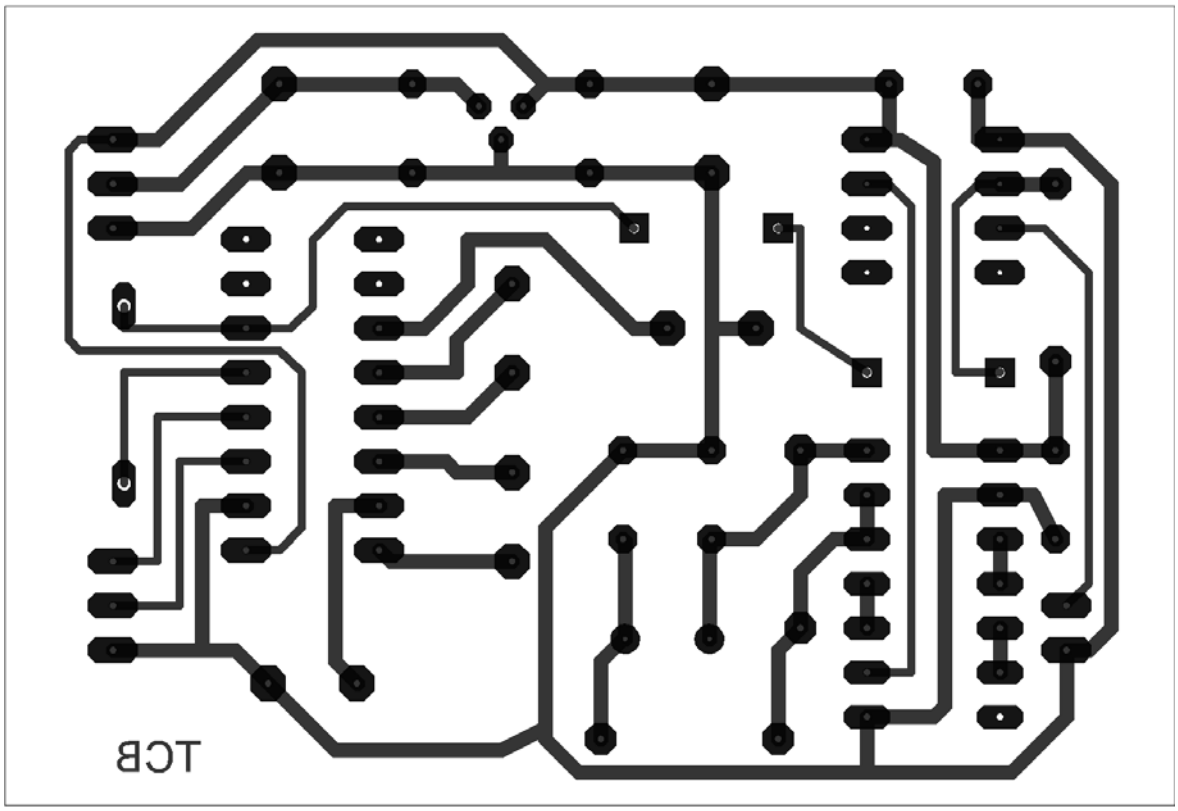


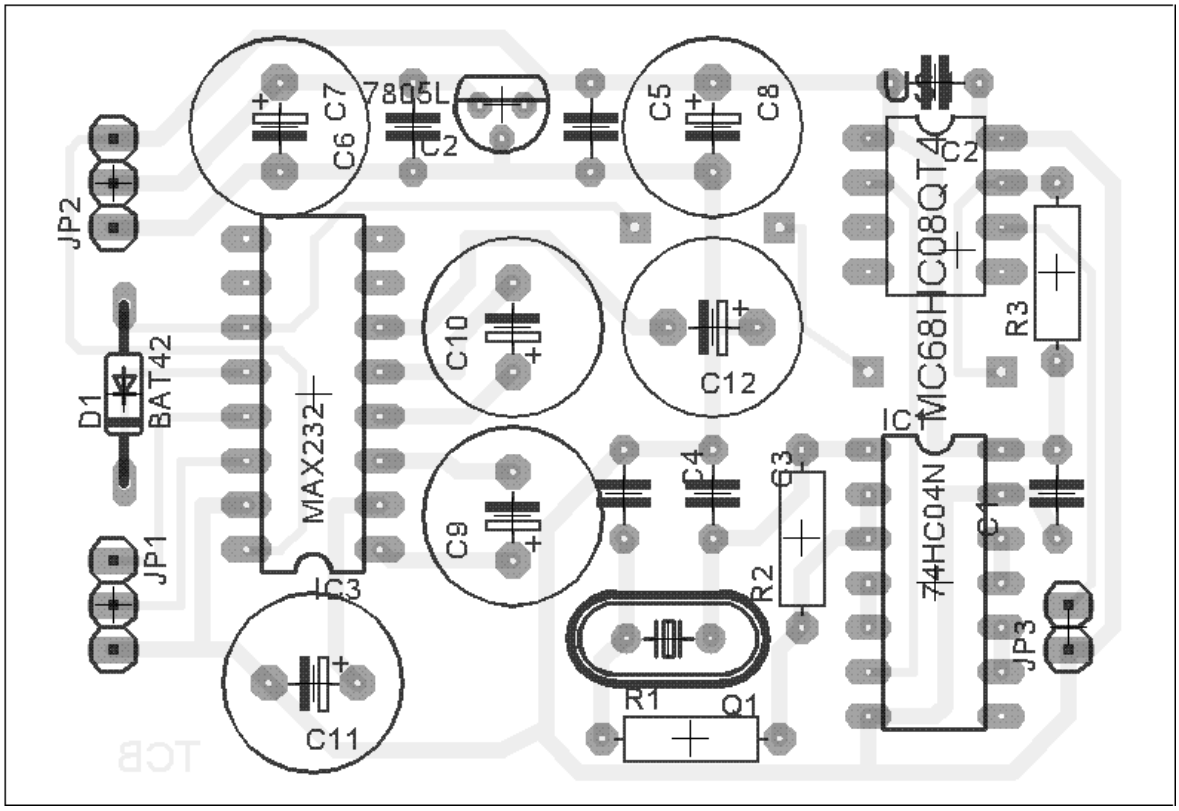



# PŘÍLOHA P III: SCHÉMA ZAPOJENÍ MCU MC68HC908QT4



# PŘÍLOHA P IV: SCHÉMA PLOŠNÉHO SPOJE MCU MC68HC908QT4





# PŘÍLOHA P V: TEPLOTNÍ SPECIFIKACE SENZORU AD592

## AD592–SPECIFICATIONS (typical @ $T_A$ = +25°C, $V_S$ = +5 V, unless otherwise noted)

| Model | AD592AN Min | AD592AN Typ | AD592AN Max | AD592BN Min | AD592BN Typ | AD592BN Max | AD592CN Min | AD592CN Typ | AD592CN Max | Units |
|---|---|---|---|---|---|---|---|---|---|---|
| ACCURACY | | | | | | | | | | |
| Calibration Error @ +25°C[1] | | 1.5 | 2.5 | | 0.7 | 1.0 | | 0.3 | 0.5 | °C |
| $T_A$ = 0°C to +70°C | | | | | | | | | | |
| Error over Temperature | | 1.8 | 3.0 | | 0.8 | 1.5 | | 0.4 | 0.8 | °C |
| Nonlinearity[2] | | 0.15 | 0.35 | | 0.1 | 0.25 | | 0.05 | 0.15 | °C |
| $T_A$ = –25°C to +105°C | | | | | | | | | | |
| Error over Temperature[3] | | 2.0 | 3.5 | | 0.9 | 2.0 | | 0.5 | 1.0 | °C |
| Nonlinearity[2] | | 0.25 | 0.5 | | 0.2 | 0.4 | | 0.1 | 0.35 | °C |
| OUTPUT CHARACTERISTICS | | | | | | | | | | |
| Nominal Current Output @ +25°C (298.2K) | | 298.2 | | | 298.2 | | | 298.2 | | µA |
| Temperature Coefficient | | 1 | | | 1 | | | 1 | | µA/°C |
| Repeatability[4] | | | 0.1 | | | 0.1 | | | 0.1 | °C |
| Long Term Stability[5] | | | 0.1 | | | 0.1 | | | 0.1 | °C/month |
| ABSOLUTE MAXIMUM RATINGS | | | | | | | | | | |
| Operating Temperature | –25 | | +105 | –25 | | +105 | –25 | | +105 | °C |
| Package Temperature[6] | –45 | | +125 | –45 | | +125 | –45 | | +125 | °C |
| Forward Voltage (+ to –) | | | 44 | | | 44 | | | 44 | V |
| Reverse Voltage (– to +) | | | 20 | | | 20 | | | 20 | V |
| Lead Temperature (Soldering 10 sec) | | | 300 | | | 300 | | | 300 | °C |
| POWER SUPPLY | | | | | | | | | | |
| Operating Voltage Range | 4 | | 30 | 4 | | 30 | 4 | | 30 | V |
| Power Supply Rejection | | | | | | | | | | |
| +4 V < $V_S$ < +5 V | | | 0.5 | | | 0.5 | | | 0.5 | °C/V |
| +5 V < $V_S$ < +15 V | | | 0.2 | | | 0.2 | | | 0.2 | °C/V |
| +15 V < $V_S$ < +30 V | | | 0.1 | | | 0.1 | | | 0.1 | °C/V |

NOTES
[1]An external calibration trim can be used to zero the error @ +25°C.
[2]Defined as the maximum deviation from a mathematically best fit line.
[3]Parameter tested on all production units at +105°C only. C grade at –25°C also.
[4]Maximum deviation between +25°C readings after a temperature cycle between –45°C and +125°C. Errors of this type are noncumulative.
[5]Operation @ +125°C, error over time is noncumulative.
[6]Although performance is not specified beyond the operating temperature range, temperature excursions within the package temperature range will not damage the device.
Specifications subject to change without notice.
Specifications shown in boldface are tested on all production units at final electrical test. Results from those tests are used to calculate outgoing quality levels. All min and max specifications are guaranteed, although only those shown in boldface are tested on all production units.