Application Note

AN2475/D
3/2003

Generating a PWM Signal Modulated by an Analog Input Using the MC68HC908QY4 Microcontroller

By: Alfredo Olmos
Andre Vilas Boas
Marcus Serra Espindola
Brazil Semiconductor Technology Center
Jaguariuna, S.P., Brazil

Introduction

This document addresses the use of the internal timer and analog-to-digital converter (ADC) modules of the MC68HC908QY/QT Family of microcontrollers (MCUs). In the application, an analog signal is converted to a digital word through the ADC to set the duty cycle of a pulse-width modulated (PWM) signal. The PWM duty cycle varies from 0 to 100% in accordance with the voltage applied at the ADC input. Hence, the PWM signal corresponds to the digital conversion of the analog input. The PWM signal is fed back to build a close-loop system for providing voltage regulation in a simple, very flexible way.

MC68HC908QY/QT Family of Microcontrollers

The MC68HC908QY/QT Family is a member of the low-cost, high-performance M68HC908 Family of 8-bit FLASH MCUs. The M68HC08 Family is a complex instruction set computer (CSIC) with a Von Neumann architecture. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

Features of the MC68HC908QY4 include:

- 4 Kbytes of in-application programmable FLASH and 128 bytes of random-access memory (RAM)
- 2-channel 16-bit timer with selectable input capture, output compare, and PWM
- Easy to use high-performance HC08 CPU
- Trimmable internal oscillator with ±5% accuracy
- 4-channel 8-bit ADC

- Selectable trip point low-voltage inhibit (LVI)
- Computer operating properly (COP) timer
- Flexible high-current input/output (I/O) and keyboard interrupts
  – 13 bidirectional
  – 1 input only
- MC68HC908QY4, MC68HC908QY2, and MC68HC908QY1 are available in these packages:
  – 16-pin plastic dual in-line package (PDIP)
  – 16-pin small outline integrated circuit (SOIC) package
  – 16-pin thin shrink small outline package (TSSOP)
- MC68HC908QT4, MC68HC908QT2, and MC68HC908QT1 are available in these packages:
  – 8-pin PDIP
  – 8-pin SOIC
  – 8-pin dual flat no lead (DFN) package

Analog-to-Digital Converter (ADC)

The 8-bit, 4-channel ADC module is only available on the MC68HC908QY2, MC68HC908QT2, MC68HC908QY4, and MC68HC908QT4. Features include:

- 4 channels with multiplexed input
- Linear successive approximation with monotonicity
- 8-bit resolution
- Single or continuous conversion
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock

Voltage Conversion

When the input voltage to the ADC equals V_{DD}, the ADC converts the signal to $FF (full scale). If the input voltage equals V_{SS}, the ADC converts it to $00. Input voltages between V_{DD} and V_{SS} are a straight-line linear conversion. All other input voltages will result in $FF if greater than V_{DD} and $00 if less than V_{SS}.

Conversion Time

Sixteen internal clock cycles are required to perform one conversion. The ADC starts a conversion on the first rising edge of the ADC internal clock immediately following a write to the ADC status and control register (ADSCR). If the internal clock is selected to run at 1 MHz, then one conversion will take 16 μs to complete. The highest sample rate at this frequency is 62.5 kHz.

Continuous Conversion versus Single Conversion

In the continuous conversion mode, the ADC continuously converts the selected channel filling the ADC data register (ADR) with new data after each conversion. Data from the previous conversion will be overwritten whether that

data has been read or not. Conversions will continue until the ADCO bit is cleared. The COCO bit (ADSCR, $003C) is set after each conversion and can be cleared by writing the ADSCR or reading of the ADR.

In the single conversion mode, once the conversion is completed the ADC puts the converted result in the ADR, sets a flag, and can generate an interrupt.

Please refer to the *MC68HC908QY4, MC68HC908QT4, MC68HC908QY2, MC68HC908QT2, MC68HC908QY1, and MC68HC908QT1 Data Sheet* (Freescale document order number MC68HC908QY4/D) for ADC features and electrical specification details.

Timer Interface Module (TIM)

The timer interface module (TIM) is a 2-channel timer that provides a timing reference with input capture, output compare, and PWM functions. Software can read the TIM counter value at any time without affecting the counting sequence. The two TIM channels are programmable independently as either input capture or output compare.

Features of the TIM include:

- Two input capture/output compare channels:
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
- Buffered and unbuffered PWM signal generation
- Programmable TIM clock input with 7-frequency internal bus clock prescaler selection
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

Pulse-Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM module can generate a PWM signal. The composed value stored in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches that composed value. The time between overflows is the period of the PWM signal.

The composed value of the TIM counter modulo registers and the selected prescaler output determine the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing $00FF (255) to the TIM counter modulo registers produces a PWM period equal to 256 times the internal bus clock period if the prescaler select value is set to 000. Writing $0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50%.

Unbuffered PWM Signal Generation

The PWM pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers. An unsynchronized write instruction to the TIM channel registers for changing the pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

1. When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
2. When changing to a longer pulse width, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in the TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin (TCH1) is available as general-purpose I/O.

NOTE: *In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the currently*

active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered PWM signals.

Since in this application the PWM duty cycle varies from 0 to 100% in accordance with the voltage applied on the ADC, input buffered PWM allows eliminating potentially harmful glitches when the pulse width setting is changed. This is accomplished without requiring a synchronization method as in unbuffered PWM signal generation.

Application Circuit

Figure 1 shows a typical application circuit involving the PWM feature. The MC68HC908QY4 outputs a PWM waveform used to control the PMOS switch and, therefore the effective DC regulated voltage applied on the load. The MCU reads the instantaneous regulated output voltage through the ADC and performs a comparison between the digital word obtained and an internal digital reference previously defined. The load voltage is acquired by the ADC through a feedback path cyclically thus, forcing the voltage regulation. If the load demands more current the regulator output voltage decreases. If the voltage input to the ADC is below the pre-set internal reference value the duty cycle is reduced (decremented), thus increasing the driver current of the converter compensating for the reduced voltage supplied to the load. On the other hand, if the load demands less current than the nominal value the MCU grows the duty cycle reducing the output voltage. When regulation is attained the green light emitting diode (LED) is turned on. Conversely, turning on the red LED indicates lack of regulation. **Figure 2** illustrates the PWM waveform modulated by an analog voltage signal applied on the ADC input whereas, **Figure 3** shows the regulated output voltage versus the load current for the application circuit.

NOTE: *It is possible to achieve a regulated voltage of 2.5 V with less than 30 mV variation driving a current load from 20 mA to 280 mA.*

Regarding the application circuit in **Figure 1** the diode, inductor, and shunt capacitor create an energy storage reservoir to store enough energy to maintain load voltage during the off-time of the transistor. The transistor serves to replace lost energy during its off-time. Operation of the power switch can be broken down into two periods.

- When the power switch is on, load current passes through the inductor to the load and the diode will be reversed biased.
- When the power switch turns off, current must still pass through the inductor. The current path through the off transistor is now open. The diode will be forward biased, thus maintaining a closed current loop through the load.

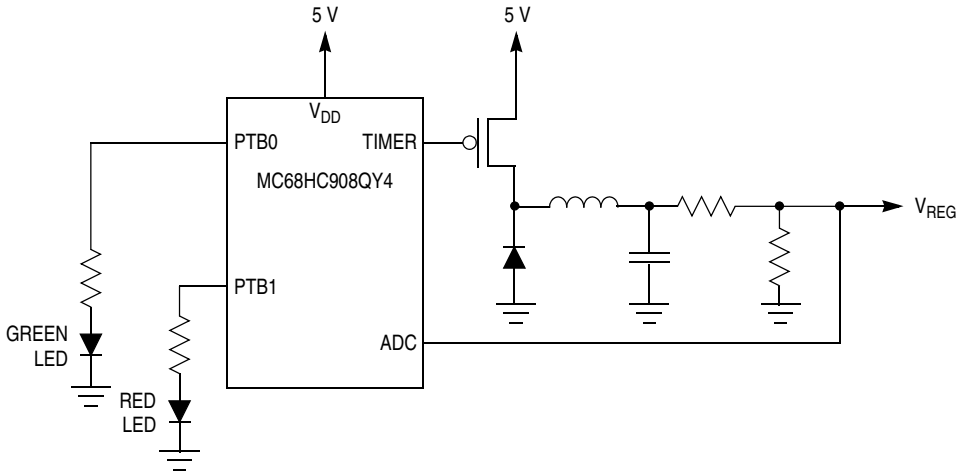

Figure 1. Application Circuit

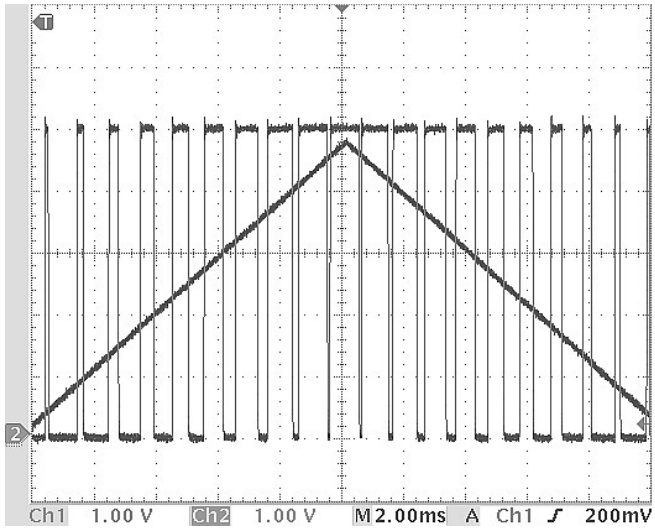

Figure 2. Duty Cycle Modulation versus Analog Input Voltage

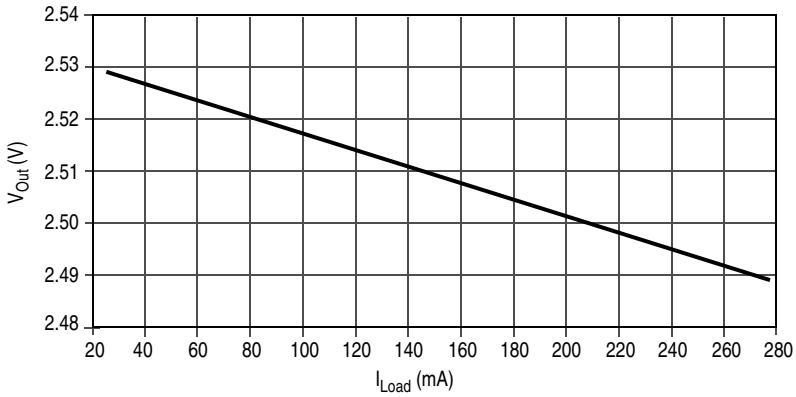

Figure 3. Voltage Regulation versus Load Current

Software Description

In this application the PWM duty cycle varies from 0 to 100% in accordance with the voltage applied on the ADC input. Actually PWM duty cycle varies from approximately 6% to 94% since it is needed to set a PWM resolution limit for proper operation of the close-loop system. User can redefine the PWM resolution limit as explained below. The PWM output of timer channel 0 is proportional to the analog converted value. When the input voltage increases, the PWM duty cycle increases and vice versa.

The ADC is configured for continuous conversion. The timer module is configured for buffered PWM signal generation. The software runs in a closed loop in which the analog input voltage is converted onto a digital data by the ADC and then compared to a target regulation value previously defined by the user. The system reaches regulation if the analog input reading is in between the upper and lower regulation limits also pre-defined by the user. The driver transistor is a PMOS device. If the ADC value approaches the upper target limit, the PWM duty cycle is increased (incremented). When the regulated output voltage is below the target value but above the lower limit the PWM duty cycle will be decremented. In case of the regulated output voltage being either over the upper limit or below the lower limit, the PWM period will be either enlarged or reduced, respectively. Therefore, the software will be seeking a combination of PWM period and duty cycle to provide a regulated voltage satisfying the pre-defined tolerances. PWM period checking range is from 256 to a minimum determined by the system resolution, as discussed below. A software flowchart is shown in **Figure 4**.

Every time the period value is changed, the PWM duty cycle is redefined to 50%. In this application a PWM resolution limit is set to approximately 6% (lower) and 94% (upper) of period. If the regulation is not reached varying PWM duty cycle between those limits, period value is altered again. This sequence is performed until regulation is attained.

The PWM resolution limits are obtained by shifting the period value four times to the right. Due to this shifting, the minimum possible period value is attained to guarantee that resolution is 17 (decimal). However, the user can redefine the PWM resolution limit altering the source by code taking into account the trade-off between system resolution and minimum period attainable.

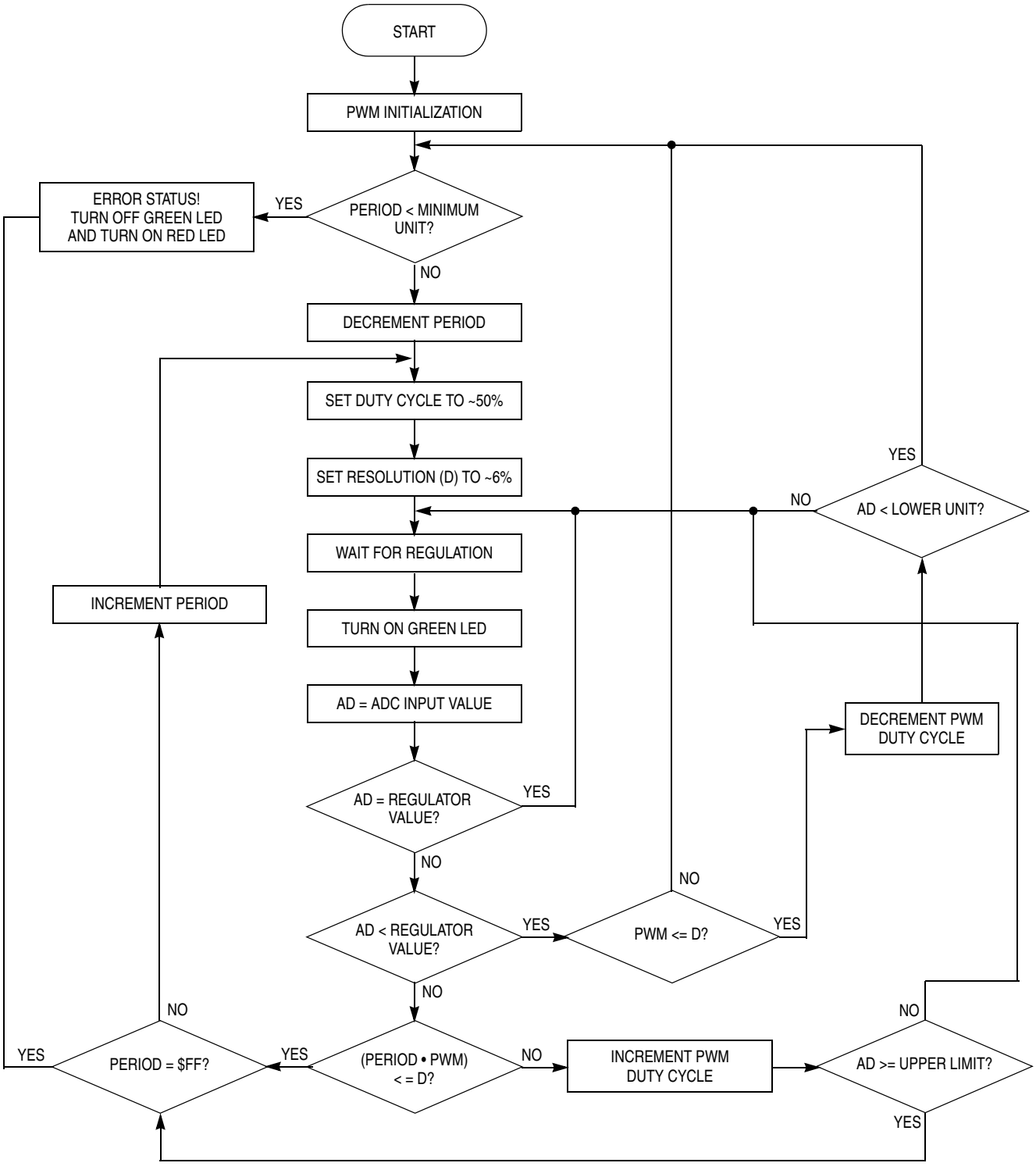

Figure 4. Software Flowchart

Software Listing

```
;*******************************************************************************
;* Title: LoadRegulation.asm                                    Copyright (c)  2002
;*******************************************************************************
;* Author: Alfredo Olmos - Freescale SPS/BSTC
;*         Andre V. Boas - Freescale SPS/BSTC
;*         Marcus Espindola - Freescale SPS/BSTC
;*
;* Description: Generating a PWM Signal Modulated by an Analog Input Using HC908QY4 MCU.
;*
;* Documentation: HC908QY4 Data Sheet (MC68HC908QY4/D) for register and bit explanations
;*
;* Include Files: MC68HC908QT4.equ, LoadRegulation.equ
;*
;* Assembler: Metrowerks CodeWarrior - HIWARE HC08-Assembler
;*
;* Revision History: not yet released
;* Rev #      Date      Who         Comments
;* -----   -----------  ---------   ------------------------------------------
;* 0.4      19-Mar-03   Espindola   Included OSCTRIM setting
;* 0.3      10-Feb-03   Espindola   Included Period variation
;* 0.2      17-Dec-02   Espindola   Cleaned up to match new coding standard
;* 0.1      01-Oct-02   Espindola   Initial data entry
;*******************************************************************************
;*******************************************************************************
;* Freescale reserves the right to make changes without further notice to any product
;* herein to improve reliability, function, or design. Freescale does not assume any
;* liability arising out of the application or use of any product, circuit, or software
;* described herein; neither does it convey any license under its patent rights nor the
;* rights of others. Freescale products are not designed, intended, or authorized for
;* use as components in systems intended for surgical implant into the body, or other
;* applications intended to support life, or for any other application in which the
;* failure of the Freescale product could create a situation where personal injury or
;* death may occur. Should Buyer purchase or use Freescale products for any such
;* intended or unauthorized application, Buyer shall indemnify and hold Freescale and
;* its officers, employees, subsidiaries, affiliates, and distributors harmless against
;* all claims, costs, damages, and expenses, and reasonable attorney fees arising out
;* of, directly or indirectly, any claim of personal injury or death associated with
;* such unintended or unauthorized use, even if such claim alleges that Freescale was
;* negligent regarding the design or manufacture of the part.
;*
;* Freescale is a registered trademark of Freescale, Inc.
;*******************************************************************************

;*******************************************************************************
;* Equates and Data Table Includes
;*******************************************************************************


            include     "MC68HC908QT4.equ"

;*******************************************************************************
;* Constants and Variables for this file
;*******************************************************************************

            include     "LoadRegulation.equ"
```

```
;*******************************************************************************************
;* Code Starts Here
;*******************************************************************************************

            org     FlashStart

Init:       mov     #initCfg1,CONFIG1  ;Set config1 register
                                       ;(LVI and COP disabled)

IntOsc:     lda     #initCfg2
            sta     CONFIG2            ;set MCU to internal oscillator

;* Configure oscillator trim value

            lda     TRIMLOC            ;Load the factory oscillator trim value
            sta     OSCTRIM            ;into the OSCTRIM register

;* Enable ADCH3 continuous convertion

            mov     #initAD,ADSCR      ;Start Convertion, CH3 selected

;* Optionally, select ADC ch 1 and 2 by using the instruction below

;*          mov     #$21,adscr         ;Start Convertion, CH1 selected
;*          mov     #$22,adscr         ;Start Convertion, CH2 selected

            bclr    PTB0,PTB           ;Initial value for PTB0
            bclr    PTB1,PTB           ;Initial value for PTB1
            bset    DDRB0,DDRB         ;PTB0 outputs regulation ok!
            bset    DDRB1,DDRB         ;PTB1 outputs regulation error!

;* Program Timer Interface Module (TIM)
;* Initializing PWM

            mov     #initTim,TSC       ;Reset & Stop Counter
            mov     #periodH,TMODH     ;Set PWM period
            mov     #periodL,TMODL     ;T = periodL*16 int. bus*(1/bus_clk)

            mov     #InitpwmH,TCH0H
            mov     #InitpwmL,TCH0L    ;initial value for output compare (50%)
            mov     #tsc0Val,TSC0      ;enable buffered PWM in channel 0

            mov     #TimVal,TSC        ;Disable interruption
                                       ;clk = internal bus / 16

            mov     #ADclkval,ADICLK   ;ADC clock, bus clock/ 16

DecPeriod:  lda     #Permin
            cmp     TMODL              ;Verify if Period is lower than Minimum Period
            bmi     Error              ;Go to Error status if so


            lda     TMODL
            deca
            clrh
            sta     TMODL              ;Decrement period
            ldx     #$02
            div
            tax
            stx     PWMLow             ;Set PWM Duty Cycle to ~50%
            stx     TCH0L
```

```
                bra     RegRes

SetPwmDc:       bset    PTB0,PTB            ;PTB0 = 1, MCU is regulating or trying to do so

                lda     ADR                 ;Read ADC input value
                cmp     #RegVal             ;Compare ADC value with Target Reg. Value
                beq     WaitReg             ;If ADC = RegVal -> wait for regulation
                bmi     DecPwm              ;If ADC < RegVal -> decrement PWM

IncPwm:         ldx     PWMLow
                lda     TMODL
                sub     TCH0L
                cmp     PWMLim              ;Verify if PWM reached its resolution limit
                bmi     IncPeriod           ;Increment Period if so

                incx                        ;Increment PWM value
                stx     PWMLow
                stx     TCH0L

                lda     ADR
                cmp     #RegHLim            ;Verify if ADC value is greater than or
                bge     IncPeriod           ;equal to Reg High Limit -> Inc Period if so

                lda     #$FF

WaitReg:        nsa                         ;Wait ~0.8ms or 2550 cycles to achieve accurate
                nsa                         ;load regulation
                deca
                bne     WaitReg

                bra     SetPwmDc

DecPwm:         ldx     PWMLow
                lda     PWMLow
                cmp     PWMLim              ;Verify if PWM reached its resolution limit
                bmi     DecPeriod           ;Decrement Period if so

                decx                        ;Decrement PWM value
                stx     PWMLow
                stx     TCH0L

                lda     ADR
                cmp     #RegLLim            ;Verify if ADC value is lower than
                bmi     DecPeriod           ;Reg Low Limit -> Decrement Period if so

                lda     #$FF
                bra     WaitReg

IncPeriod:      lda     TMODL
                cmp     #$FF                ;Verify if Period is at high limit
                beq     Error               ;Go to Error Status if so

                lda     TMODL
                inca
                clrh
                sta     TMODL               ;Increment period
                ldx     #$02
                div
                tax                         ;Set PWM Duty Cycle to ~ 50%
                stx     PWMLow
```

```
            stx     TCH0L

RegRes:     lda     TMODL
            Lsra
            lsra
            lsra
            lsra                        ;Shift 4 times tmodl register value to the right
            sta     PWMLim              ;in order to adjusted PWM resolution limit (or D)
                                        ;to ~6%

            lda     #$FF
            bra     WaitReg

Error:      bclr    PTB0,PTB
            bset    PTB1,PTB            ;Set ptb1 and clear ptb0
            bra     Error               ;if MCU is not regulating


;**************************************************************************************
;* Interruptions
;**************************************************************************************

Dummy:
            rti

;**************************************************************************************
;* Start of Vectors Definitions
;**************************************************************************************

            org     $FFDE
            fdb      Dummy              ;ADC conversion complete vector

            org     $FFE0
            fdb      Dummy              ;Keyboard vector

            org     $FFF2
            fdb      Dummy              ;TIM overflow vector

            org     $FFF4
            fdb      Dummy              ;TIM Channel 1 vector

            org     $FFF6
            fdb      Dummy              ;TIM Channel 0 vector

            org     $FFFA
            fdb      Dummy              ;IRQ vector

            org     $FFFC
            fdb      Dummy              ;SWI vector

            org     $FFFE
            fdb      Init               ;Reset vector
```

Register and Bit Definitions for MC68HC908QY4

```
;*******************************************************************************
;* Title: HC908QY4.equ                                      Copyright (c)  2002
;*******************************************************************************
;* Author:  Kazue Kikuchi
;*
;* Description:  Register and bit name definitions for MC68HC908QY4 and MC68HC908QT4
;*
;* Documentation:  MC68HC908QT4/D  Advance Information
;* Include Files:  none
;*
;* Assembler:  P&E Microcomputer Systems - CASM08Z  (v3.16)
;*
;* Revision History:
;* Rev #      Date       Who     Comments
;* ------  -----------   ------  ------------------------------------------------
;* ES 1.0   29-Apr-02    KK      Initial release
;* ES 1.1   07-Jun-02    KK      Fixed OSCTRIM address and typos
;*
;*******************************************************************************
;*******************************************************************************
;*
;*   Freescale reserves the right to make changes without further notice to any product
;*   herein to improve reliability, function, or design.  Freescale does not assume any
;*   liability arising out of the application or use of any product, circuit, or software
;*   described herein; neither does it convey any license under its patent rights nor the
;*   rights of others.  Freescale products are not designed, intended, or authorized for
;*   use as components in systems intended for surgical implant into the body, or other
;*   applications intended to support life, or for any other application in which the
;*   failure of the Freescale product could create a situation where personal injury or
;*   death may occur.  Should Buyer purchase or use Freescale products for any such
;*   intended or unauthorized application, Buyer shall indemnify and hold Freescale and
;*   its officers, employees, subsidiaries, affiliates, and distributors harmless against
;*   all claims, costs, damages, and expenses, and reasonable attorney fees arising out
;*   of, directly or indirectly, any claim of personal injury or death associated with
;*   such unintended or unauthorized use, even if such claim alleges that Freescale was
;*   negligent regarding the design or manufacture of the part.
;*
;*   Freescale and the Freescale logo are registered trademarks of Freescale, Inc.
;*******************************************************************************

;****  Memory Map and Interrupt Vectors  ******************************************
;*
RamStart:    equ    $0080         ;Start of RAM
RamLast:     equ    $00FF         ;Last RAM location
FlashStart:  equ    $EE00         ;Start of Flash
FlashLast:   equ    $FDFF         ;Last Flash location
;
Vadc:        equ    $FFDE         ;ADC vector
Vkbd:        equ    $FFE0         ;Keyboard vector
Vtimov:      equ    $FFF2         ;Timer overflow vector
Vtimch1:     equ    $FFF4         ;Timer channel 1 vector
Vtimch0:     equ    $FFF6         ;Timer channel 0 vector
Virq:        equ    $FFFA         ;IRQ vector
Vswi:        equ    $FFFC         ;SWI vector
Vreset:      equ    $FFFE         ;Reset vector
```

```
;****  Input/Output (I/O) Ports  ********************************************************
;*
PTA:          equ    $00           ;Port A data register
; bit numbers for use in BLCR, BSET, BRCLR, and BRSET
AWUL          equ    6             ;Auto wake-up latch data
PTA5:         equ    5             ;Port A data bit 5
PTA4:         equ    4             ;Port A data bit 4
PTA3:         equ    3             ;Port A data bit 3
PTA2:         equ    2             ;Port A data bit 2
PTA1:         equ    1             ;Port A data bit 1
PTA0:         equ    0             ;Port A data bit 0
; bit position masks
mAWUL:        equ    %01000000     ;Auto wake-up latch data
mPTA5:        equ    %00100000     ;Port A data bit 5
mPTA4:        equ    %00010000     ;Port A data bit 4
mPTA3:        equ    %00001000     ;Port A data bit 3
mPTA2:        equ    %00000100     ;Port A data bit 2
mPTA1:        equ    %00000010     ;Port A data bit 1
mPTA0:        equ    %00000001     ;Port A data bit 0

PTB:          equ    $01           ;Port B data register
; bit numbers for use in BLCR, BSET, BRCLR, and BRSET
PTB7:         equ    7             ;Port B data bit 7
PTB6:         equ    6             ;Port B data bit 6
PTB5:         equ    5             ;Port B data bit 5
PTB4:         equ    4             ;Port B data bit 4
PTB3:         equ    3             ;Port B data bit 3
PTB2:         equ    2             ;Port B data bit 2
PTB1:         equ    1             ;Port B data bit 1
PTB0:         equ    0             ;Port B data bit 0
; bit position masks
mPTB7:        equ    %10000000     ;Port B data bit 7
mPTB6:        equ    %01000000     ;Port B data bit 6
mPTB5:        equ    %00100000     ;Port B data bit 5
mPTB4:        equ    %00010000     ;Port B data bit 4
mPTB3:        equ    %00001000     ;Port B data bit 3
mPTB2:        equ    %00000100     ;Port B data bit 2
mPTB1:        equ    %00000010     ;Port B data bit 1
mPTB0:        equ    %00000001     ;Port B data bit 0

DDRA:         equ    $04           ;Port A data direction register
; bit numbers for use in BLCR, BSET, BRCLR, and BRSET
DDRA5:        equ    5             ;Port A data direction bit 5
DDRA4:        equ    4             ;Port A data direction bit 4
DDRA3:        equ    3             ;Port A data direction bit 3
DDRA1:        equ    1             ;Port A data direction bit 1
DDRA0:        equ    0             ;Port A data direction bit 0
; bit position masks
mDDRA5:       equ    %00100000     ;Port A data direction bit 5
mDDRA4:       equ    %00010000     ;Port A data direction bit 4
mDDRA3:       equ    %00001000     ;Port A data direction bit 3
mDDRA1:       equ    %00000010     ;Port A data direction bit 1
mDDRA0:       equ    %00000001     ;Port A data direction bit 0

DDRB:         equ    $05           ;Port B data direction register
; bit numbers for use in BLCR, BSET, BRCLR, and BRSET
DDRB7:        equ    7             ;Port B data direction bit 7
DDRB6:        equ    6             ;Port B data direction bit 6
DDRB5:        equ    5             ;Port B data direction bit 5
DDRB4:        equ    4             ;Port B data direction bit 4
```

```
DDRB3:       equ    3             ;Port B data direction bit 3
DDRB2:       equ    2             ;Port B data direction bit 2
DDRB1:       equ    1             ;Port B data direction bit 1
DDRB0:       equ    0             ;Port B data direction bit 0
; bit position masks
mDDRB7:      equ    %10000000     ;Port B data direction bit 7
mDDRB6:      equ    %01000000     ;Port B data direction bit 6
mDDRB5:      equ    %00100000     ;Port B data direction bit 5
mDDRB4:      equ    %00010000     ;Port B data direction bit 4
mDDRB3:      equ    %00001000     ;Port B data direction bit 3
mDDRB2:      equ    %00000100     ;Port B data direction bit 2
mDDRB1:      equ    %00000010     ;Port B data direction bit 1
mDDRB0:      equ    %00000001     ;Port B data direction bit 0

PTAPUE:      equ    $0B           ;Port A input pullup enable register
; bit numbers for use in BLCR, BSET, BRCLR, and BRSET
OSC2EN:      equ    7             ;OSC2 pin enable
PTAPUE5:     equ    5             ;Port A input pull up enable bit 5
PTAPUE4:     equ    4             ;Port A input pull up enable bit 4
PTAPUE3:     equ    3             ;Port A input pull up enable bit 3
PTAPUE2:     equ    2             ;Port A input pull up enable bit 2
PTAPUE1:     equ    1             ;Port A input pull up enable bit 1
PTAPUE0:     equ    0             ;Port A input pull up enable bit 0
; bit position masks
mOSC2EN:     equ    %10000000     ;OSC2 pin enable
mPTAPUE5:    equ    %00100000     ;Port A input pull up enable bit 5
mPTAPUE4:    equ    %00010000     ;Port A input pull up enable bit 4
mPTAPUE3:    equ    %00001000     ;Port A input pull up enable bit 3
mPTAPUE2:    equ    %00000100     ;Port A input pull up enable bit 2
mPTAPUE1:    equ    %00000010     ;Port A input pull up enable bit 1
mPTAPUE0:    equ    %00000001     ;Port A input pull up enable bit 0

PTBPUE:      equ    $0C           ;Port B input pullup enable register
; bit numbers for use in BLCR, BSET, BRCLR, and BRSET
PTBPUE7:     equ    7             ;Port B input pull up enable bit 7
PTBPUE6:     equ    6             ;Port B input pull up enable bit 6
PTBPUE5:     equ    5             ;Port B input pull up enable bit 5
PTBPUE4:     equ    4             ;Port B input pull up enable bit 4
PTBPUE3:     equ    3             ;Port B input pull up enable bit 3
PTBPUE2:     equ    2             ;Port B input pull up enable bit 2
PTBPUE1:     equ    1             ;Port B input pull up enable bit 1
PTBPUE0:     equ    0             ;Port B input pull up enable bit 0
; bit position masks
mPTBPUE7:    equ    %10000000     ;Port B input pull up enable bit 7
mPTBPUE6:    equ    %01000000     ;Port B input pull up enable bit 6
mPTBPUE5:    equ    %00100000     ;Port B input pull up enable bit 5
mPTBPUE4:    equ    %00010000     ;Port B input pull up enable bit 4
mPTBPUE3:    equ    %00001000     ;Port B input pull up enable bit 3
mPTBPUE2:    equ    %00000100     ;Port B input pull up enable bit 2
mPTBPUE1:    equ    %00000010     ;Port B input pull up enable bit 1
mPTBPUE0:    equ    %00000001     ;Port B input pull up enable bit 0

;****  Keyboard Interrupt Module (KBI)  ************************************************
;*
KBSCR:       equ    $1A           ;Keyboard status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
KEYF:        equ    3             ;Keyboard flag
ACKK:        equ    2             ;Keyboard acknowledge
IMASKK:      equ    1             ;Keyboard interrupt mask
MODEK:       equ    0             ;Keyboard triggering sesitivity
```

```
; bit position masks
mKEYF:        equ    %00001000     ;Keyboard flag
mACKK:        equ    %00000100     ;Keyboard acknowledge
mIMASKK:      equ    %00000010     ;Keyboard interrupt mask
mMODEK:       equ    %00000001     ;Keyboard triggering sesitivity

KBIER:        equ    $1B           ;Keyboard interrupt enable register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
AWUIE:        equ    6             ;Auto wake-up interrupt enable
KBIE5:        equ    5             ;Port A keyboard interrupt enable bit 5
KBIE4:        equ    4             ;Port A keyboard interrupt enable bit 4
KBIE3:        equ    3             ;Port A keyboard interrupt enable bit 3
KBIE2:        equ    2             ;Port A keyboard interrupt enable bit 2
KBIE1:        equ    1             ;Port A keyboard interrupt enable bit 1
KBIE0:        equ    0             ;Port A keyboard interrupt enable bit 0
; bit position masks
mAWUIE        equ    %01000000     ;Auto wake-up interrupt enable
mKBIE5:       equ    %00100000     ;Port A keyboard interrupt enable bit 5
mKBIE4:       equ    %00010000     ;Port A keyboard interrupt enable bit 4
mKBIE3:       equ    %00001000     ;Port A keyboard interrupt enable bit 3
mKBIE2:       equ    %00000100     ;Port A keyboard interrupt enable bit 2
mKBIE1:       equ    %00000010     ;Port A keyboard interrupt enable bit 1
mKBIE0:       equ    %00000001     ;Port A keyboard interrupt enable bit 0

;****  External Interrupt (IRQ)  ********************************************************
;*
INTSCR:       equ    $1D           ;IRQ status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
IRQF1:        equ    3             ;IRQ flag
ACK1:         equ    2             ;IRQ interrupt request acknowledge
IMASK1:       equ    1             ;IRQ interrupt mask
MODE1:        equ    0             ;IRQ edge/level select
; bit position masks
mIRQF1:       equ    %00001000     ;IRQ flag
mACK1:        equ    %00000100     ;IRQ interrupt request acknowledge
mIMASK1:      equ    %00000010     ;IRQ interrupt mask
mMODE1:       equ    %00000001     ;IRQ edge/level select

;**** Configuration Registers (CONFIG)  ************************************************
;*
CONFIG2:      equ    $1E           ;Configuration register 2
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
IRQPUD:       equ    7             ;IRQ pin pullup control
IRQEN:        equ    6             ;IRQ pin function selection
OSCOPT1:      equ    4             ;Selection bit 1 for oscillator option
OSCOPT0:      equ    3             ;Selection bit 0 for oscillator option
RSTEN:        equ    0             ;RST pin function selection
; bit position masks
mIRQPUD:      equ    %10000000     ;IRQ pin pullup control
mIRQEN:       equ    %01000000     ;IRQ pin function selection
mOSCOPT1:     equ    %00010000     ;Selection bit 1 for oscillator option
mOSCOPT0:     equ    %00001000     ;Selection bit 0 for oscillator option
mRSTEN:       equ    %00000001     ;RST pin function selection

CONFIG1:      equ    $1F           ;Configuration register 1
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
COPRS:        equ    7             ;COP reset period selection
LVISTOP:      equ    6             ;LVI enable in stop mode
LVIRSTD:      equ    5             ;LVI reset disable
LVIPWRD:      equ    4             ;LVI power disable
LVI5OR3:      equ    3             ;LVI 5V or 3V operating mode
```

```
SSREC:      equ    2            ;Short stop recovery
STOP:       equ    1            ;STOP instruction enable
COPD:       equ    0            ;COP disable
; bit position masks
mCOPRS:     equ    %10000000    ;COP reset period selection
mLVISTOP:   equ    %01000000    ;LVI enable in stop mode
mLVIRSTD:   equ    %00100000    ;LVI reset disable
mLVIPWRD:   equ    %00010000    ;LVI power disable
mLVI5OR3:   equ    %00001000    ;LVI 5V or 3V operating mode
mSSREC:     equ    %00000100    ;Short stop recovery
mSTOP:      equ    %00000010    ;STOP instruction enable
mCOPD:      equ    %00000001    ;COP disable

;****  Timer Interface module (TIM)  ************************************************
;*
TSC:        equ    $20          ;Timer status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
TOF:        equ    7            ;TIM overflow flag
TOIE:       equ    6            ;TIM overflow interrupt enable
TSTOP:      equ    5            ;TIM Stop bit
TRST:       equ    4            ;TIM Reset bit
PS2:        equ    2            ;Prescaler select bit 2
PS1:        equ    1            ;Prescaler select bit 1
PS0:        equ    0            ;Prescaler select bit 0
; bit position masks
mTOF:       equ    %10000000    ;TIM overflow flag
mTOIE:      equ    %01000000    ;TIM overflow interrupt enable
mTSTOP:     equ    %00100000    ;TIM Stop bit
mTRST:      equ    %00010000    ;TIM Reset bit
mPS2:       equ    %00000100    ;Prescaler select bit 2
mPS1:       equ    %00000010    ;Prescaler select bit 1
mPS0:       equ    %00000001    ;Prescaler select bit 0

TSC0:       equ    $25          ;Timer channel 0 status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
CH0F:       equ    7            ;Channel 0 flag
CH0IE:      equ    6            ;Channel 0 interrupt enable
MS0B:       equ    5            ;Mode select bit B
MS0A:       equ    4            ;Mode select bit A
ELS0B:      equ    3            ;Edge/level select bit B
ELS0A:      equ    2            ;Edge/level select bit A
TOV0        equ    1            ;Toggle on overflow
CH0MAX      equ    0            ;Channel 0 maximum duty cycle
; bit position masks
mCH0F:      equ    %10000000    ;Channel 0 flag
mCH0IE:     equ    %01000000    ;Channel 0 interrupt enable
mMS0B:      equ    %00100000    ;Mode select bit B
mMS0A:      equ    %00010000    ;Mode select bit A
mELS0B:     equ    %00001000    ;Edge/level select bit B
mELS0A:     equ    %00000100    ;Edge/level select bit A
mTOV0       equ    %00000010    ;Toggle on overflow
mCH0MAX     equ    %00000001    ;Channel 0 maximum duty cycle

TSC1:       equ    $28          ;Timer channel 1 status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
CH1F:       equ    7            ;Channel 1 flag
CH1IE:      equ    6            ;Channel 1 interrupt enable
MS1B:       equ    5            ;Mode select bit B
MS1A:       equ    4            ;Mode select bit A
ELS1B:      equ    3            ;Edge/level select bit B
ELS1A:      equ    2            ;Edge/level select bit A
```

```
TOV1          equ    1              ;Toggle on overflow
CH1MAX        equ    0              ;Channel 1 maximum duty cycle
; bit position masks
mCH1F:        equ    %10000000      ;Channel 1 flag
mCH1IE:       equ    %01000000      ;Channel 1 interrupt enable
mMS1B:        equ    %00100000      ;Mode select bit B
mMS1A:        equ    %00010000      ;Mode select bit A
mELS1B:       equ    %00001000      ;Edge/level select bit B
mELS1A:       equ    %00000100      ;Edge/level select bit A
mTOV1         equ    %00000010      ;Toggle on overflow
mCH1MAX       equ    %00000001      ;Channel 1 maximum duty cycle

TCNTH:        equ    $21            ;Timer counter register high
TCNTL:        equ    $22            ;Timer counter register Low
TMODH:        equ    $23            ;Timer counter modulo register high
TMODL:        equ    $24            ;Timer counter modulo register low
TCH0H:        equ    $26            ;Timer channel 0 register high
TCH0L:        equ    $27            ;Timer channel 0 register low
TCH1H:        equ    $29            ;Timer channel 1 register high
TCH1L:        equ    $2A            ;Timer channel 1 register low

;****  Oscillator Module (OSC)  ************************************************
;*
OSCSTAT:      equ    $36            ;Oscillator status register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
ECGON:        equ    1              ;External clock generator on
ECGST:        equ    0              ;External clock status
; bit position masks
mECGON:       equ    %00000010      ;External clock generator on
mECGST:       equ    %00000001      ;External clock status

OSCTRIM:      equ    $38            ;Oscillator trim register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
TRIM7:        equ    7              ;Internal oscillator trim factor bit 7
TRIM6:        equ    6              ;Internal oscillator trim factor bit 6
TRIM5:        equ    5              ;Internal oscillator trim factor bit 5
TRIM4:        equ    4              ;Internal oscillator trim factor bit 4
TRIM3:        equ    3              ;Internal oscillator trim factor bit 3
TRIM2:        equ    2              ;Internal oscillator trim factor bit 2
TRIM1:        equ    1              ;Internal oscillator trim factor bit 1
TRIM0:        equ    0              ;Internal oscillator trim factor bit 0
; bit position masks
mTRIM7:       equ    %10000000      ;Internal oscillator trim factor bit 7
mTRIM6:       equ    %01000000      ;Internal oscillator trim factor bit 6
mTRIM5:       equ    %00100000      ;Internal oscillator trim factor bit 5
mTRIM4:       equ    %00010000      ;Internal oscillator trim factor bit 4
mTRIM3:       equ    %00001000      ;Internal oscillator trim factor bit 3
mTRIM2:       equ    %00000100      ;Internal oscillator trim factor bit 2
mTRIM1:       equ    %00000010      ;Internal oscillator trim factor bit 1
mTRIM0:       equ    %00000001      ;Internal oscillator trim factor bit 0

TRIMLOC:      equ    $FFC0          ;Internal oscillator trim value


;****  Analog-to-Digital Converter (ADC)  ***************************************
;*
ADSCR:        equ    $3C            ;ADC status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
COCO:         equ    7              ;Conversions complete
AIEN:         equ    6              ;ADC interrupt enable bit
ADCO:         equ    5              ;ADC continuous conversion
```

```
CH4:          equ    4              ;ADC channel select bit 4
CH3:          equ    3              ;ADC channel select bit 3
CH2:          equ    2              ;ADC channel select bit 2
CH1:          equ    1              ;ADC channel select bit 1
CH0:          equ    0              ;ADC channel select bit 0
; bit position masks
mCOCO:        equ    %10000000      ;Conversions complete
mAIEN:        equ    %01000000      ;ADC interrupt enable bit
mADCO:        equ    %00100000      ;ADC continuous conversion
mCH4:         equ    %00010000      ;ADC channel select bit 4
mCH3:         equ    %00001000      ;ADC channel select bit 3
mCH2:         equ    %00000100      ;ADC channel select bit 2
mCH1:         equ    %00000010      ;ADC channel select bit 1
mCH0:         equ    %00000001      ;ADC channel select bit 0

ADR:          equ    $3E            ;ADC data register

ADICLK:       equ    $3F            ;ADC input clock register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
ADIV2:        equ    7              ;ADC clock prescaler bit 2
ADIV1:        equ    6              ;ADC clock prescaler bit 1
ADIV0:        equ    5              ;ADC clock prescaler bit 0
; bit position masks
mADIV2:       equ    %10000000      ;ADC clock prescaler bit 2
mADIV1:       equ    %01000000      ;ADC clock prescaler bit 1
mADIV0:       equ    %00100000      ;ADC clock prescaler bit 0

;****  System Integration Module (SIM)  ************************************************
;*
BSR:          equ    $FE00          ;SIM break status register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
SBSW          equ    1              ;SIM break stop/wait
; bit position masks
mSBSW:        equ    %00000010      ;SIM break stop/wait

SRSR:         equ    $FE01          ;SIM reset status register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
POR:          equ    7              ;Power-on reset
PIN:          equ    6              ;External reset
COP:          equ    5              ;COP reset
ILOP:         equ    4              ;Illegal opcode reset
ILAD:         equ    3              ;Illegal address reset
MODRST:       equ    2              ;Monitor mode entry module reset
LVI:          equ    1              ;LVI reset
; bit position masks
mPOR:         equ    %10000000      ;Power-on reset
mPIN:         equ    %01000000      ;External reset
mCOP:         equ    %00100000      ;COP reset
mILOP:        equ    %00010000      ;Illegal opcode reset
mILAD:        equ    %00001000      ;Illegal address reset
mMODRST:      equ    %00000100      ;Monitor mode entry module reset
mLVI:         equ    %00000010      ;LVI reset

BRKAR:        equ    $FE02          ;Break auxiliary register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
BDCOP:        equ    0              ;Break disable COP
; bit position masks
mBDCOP:       equ    %00000001      ;Break disable COP

BFCR:         equ    $FE03          ;Break flag control register
```

```
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
BCFE:       equ    7              ;Break clear flag enable
; bit position masks
mBCFE:      equ    %10000000      ;Break clear flag enable

INT1:       equ    $FE04          ;Interrupt status register 1
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
IF5:        equ    6              ;Interrupt flag 5
IF4:        equ    5              ;Interrupt flag 4
IF3:        equ    4              ;Interrupt flag 3
IF1:        equ    2              ;Interrupt flag 1

INT2:       equ    $FE05          ;Interrupt status register 2
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
IF14:       equ    7              ;Interrupt flag 14

INT3:       equ    $FE06          ;Interrupt status register 3
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
IF15:       equ    0              ;Interrupt flag 15

;****  Flash Memory  ************************************************************
;*
FLCR:       equ    $FE08          ;Flash control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
HVEN:       equ    3              ;High-voltage enable bit mask
MASS:       equ    2              ;Mass erase control bit mask
ERASE:      equ    1              ;Erase control bit mask
PGM:        equ    0              ;Program control bit mask
; bit position masks
mHVEN:      equ    %00001000      ;High-voltage enable bit mask
mMASS:      equ    %00000100      ;Mass erase control bit mask
mERASE:     equ    %00000010      ;Erase control bit mask
mPGM:       equ    %00000001      ;Program control bit mask

FLBPR:      equ    $FFBE          ;Flash block protect register

;****  Breakpoint Module (BRK)  *************************************************
;*
BRKH:       equ    $FE09          ;Break address register high
BRKL:       equ    $FE0A          ;Break address register low

BRKSCR:     equ    $FE0B          ;Break status and control register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
BRKE:       equ    7              ;Break enable
BRKA:       equ    6              ;Break active
; bit position masks
mBRKE:      equ    %10000000      ;Break enable
mBRKA:      equ    %01000000      ;Break active

;****  Low-Voltage Inhibit (LVI)  ***********************************************
;*
LVISR:      equ    $FE0C          ;LVI status register
; bit numbers for use in BCLR, BSET, BRCLR, and BRSET
LVIOUT:     equ    7              ;LVI output bit
; bit position masks
mLVIOUT:    equ    %10000000      ;LVI output bit

;****  Computer Operating Properly (COP)  ***************************************
;*
COPCTL:     equ    $FFFF          ;COP control register
```

Constants and variables definitions

```
;*******************************************************************************
;* Title: LoadRegulation.equ                                    Copyright (c) ************************
;* Author: Alfredo Olmos - Freescale SPS/BSTC
;*         Andre V. Boas - Freescale SPS/BSTC
;*         Marcus Espindola - Freescale SPS/BSTC
;*
;* Description: Constants and variables definitions for MC68HC908QY4 and MC68HC908QT4.
;*
;* Documentation: HC908QY4 Data Sheet (MC68HC908QY4/D) for register and bit explanations
;*
;* Include Files:
;*
;* Assembler: P&E Microcomputer Systems - CASM for HC08
;*
;* Revision History: not yet released
;* Rev #      Date      Who         Comments
;* -----   -----------  ---------   ------------------------------------------
;* 0.1      19-Mar-03   Espindola   Initial data entry
;*******************************************************************************
;*******************************************************************************
;* Freescale reserves the right to make changes without further notice to any product
;* herein to improve reliability, function, or design. Freescale does not assume any
;* liability arising out of the application or use of any product, circuit, or software
;* described herein; neither does it convey any license under its patent rights nor the
;* rights of others. Freescale products are not designed, intended, or authorized for
;* use as components in systems intended for surgical implant into the body, or other
;* applications intended to support life, or for any other application in which the
;* failure of the Freescale product could create a situation where personal injury or
;* death may occur. Should Buyer purchase or use Freescale products for any such
;* intended or unauthorized application, Buyer shall indemnify and hold Freescale and
;* its officers, employees, subsidiaries, affiliates, and distributors harmless against
;* all claims, costs, damages, and expenses, and reasonable attorney fees arising out
;* of, directly or indirectly, any claim of personal injury or death associated with
;* such unintended or unauthorized use, even if such claim alleges that Freescale was
;* negligent regarding the design or manufacture of the part.
;*
;* Freescale is a registered trademark of Freescale, Inc.
;*******************************************************************************

;*******************************************************************************
;* Constants and Variables for this file
;*******************************************************************************

            org    RamStart

initCfg1: equ   12
;initCfg1: equ   %00010001     ;Config1 Register value
;                ||||||||     CONFIG1 is a write once register
;                |||||||+-COPD    - 1 disable COP Watchdog
;                ||||||+--STOP    - 0 disable STOP instruction
;                |||||+---SSREC   - 0 4096 cycle STOP recovery
;                ||||+----LVI5OR3 - 0 set LVI for 3V system
;                |||+-----LVIPWRD - 1 disable power to LVI system
;                ||+------LVIRSTD - 0 enable reset on LVI trip
;                |+-------LVISTOP - 0 disable LVI in STOP mode
;                +--------COPRS   - 0 long COP timeout
```

```
initCfg2: equ   %00000000     ;Config2 Register value
;                |||||||      CONFIG2 is a write once register
;                |||||||+-RSTEN   - 0 Reset function inactive in pin
;                ||||||+--R       - 0 Reserved bit
;                |||||+---R       - 0 Reserved bit
;                ||||+----OSCOPT0 - 0 Set oscillator option as internal
;                |||+-----OSCOPT1 - 0 Set oscillator option as internal
;                ||+------R       - 0 Reserved bit
;                |+-------IRQEN   - 0 disable IRQ function
;                +--------IRQPUD  - 0 Internal pullup conecte IRQ and VDD


initAD: equ     %00100011     ;AD configuration value
;                |||||||      ADC Status and Control Register
;                |||||||+-CH0     - 1 Mux to select ADC channel
;                ||||||+--CH1     - 1 Mux to select ADC channel
;                |||||+---CH2     - 0 Mux to select ADC channel
;                ||||+----CH3     - 0 Mux to select ADC channel
;                |||+-----CH4     - 0 Channel 3 selected
;                ||+------ADCO    - 1 Set ADC as continuous conversion
;                |+-------AIEN    - 0 disable ADC interrupt
;                +--------COCO    - 0 Conversions Complete Bit


initTim: equ    %00110000     ;Timer Status and control Reg. value
;                |||||||      TIM Status and Control Register
;                |||||||+-PS0     - 0 Prescaler select bit
;                ||||||+--PS1     - 0 Prescaler select bit
;                |||||+---PS2     - 0 Tim clock source int. bus / 8
;                ||||+----0       - 0
;                |||+-----TRST    - 1 TIM reset bit
;                ||+------TSTOP   - 1 TIM counter stopped
;                |+-------TOIE    - 0 disable TIM overflow interrupts
;                +--------TOF     - 0 TIM overflow flag bit


periodH: equ    %00000000     ;TIM Counter Modulo Registers
                              ;Set period high register value
periodL: equ    %11111111     ;Initial Period value
                              ;Set period to 256

InitpwmH: equ   %00000000     ;Initial PWM duty cycle value
InitpwmL: equ   %10000000     ;Set PWM duty cycle to 50%

ADclkval: equ   %10000000     ;AD clock configuration
                              ;ADC Clock prescaler bit

tsc0Val: equ    %00101010     ;TSC0 Initial Value
;                |||||||      TIM Channel Status and Control Register
;                |||||||+-CH0MAX  - 0 Channel 0 Maximum Duty Cycle bit
;                ||||||+--TOV0    - 1 Channel 0 toggle on overflow
;                |||||+---ELS0A   - 0 |
;                ||||+----ELS0B   - 1 |
;                |||+-----MS0A    - 0 |
;                ||+------MS0B    - 1 |-> Config TIM as buffered PWM
;                |+-------CH0IE   - 0 disable Channel 0 CPU interrupt
;                +--------CH0F    - 0 No input capture or output compare
```

```
TimVal: equ     %00000100     ;Timer Register configuration value
;                |||||||      TIM Status and Control Register
;                |||||||+-PS0     - 0 Prescaler select bit
;                ||||||+--PS1     - 0 Prescaler select bit
;                |||||+---PS2     - 1 Tim clock source int. bus / 16
;                ||||+----0       - 0
;                |||+-----TRST    - 0 TIM reset bit
;                ||+------TSTOP   - 0 TIM counter active
;                |+-------TOIE    - 0 disable TIM overflow interrupts
;                +--------TOF     - 0 TIM overflow flag bit


RegHLim: equ    %10000101     ;Upper Limit Period value set
                              ;by user

RegLLim: equ    %01111011     ;Lower Limit Period value set
                              ;by user


RegVal: equ     %10000000     ;Target Reg. Value set by user

Permin: equ     %00010001     ;Minimum Period Value set by user

PWMLim: rmb     1             ;PWM Duty Cycle resolution limit (or D)

PWMLow: rmb     1             ;PWM Duty Cycle
```

Conclusion

In this application a PWM modulator is implemented using the MC68HC908QY4 microcontroller. Typical PWM applications include motor speed control, battery chargers, and switching voltage regulators, all of them easily implemented using the 68HC08-based MCU's. This low-cost, low pin count family of microcontrollers brings to designers flexibility in the system development process helping to reduce overall system costs and speed up time-to-market. In addition, a variety of features and benefits make MC68HC908QYx microcontrollers versatile for a wide range of systems like home applications, computer peripherals, and automotive electronics.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com