

Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

Details	
Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	5
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	8-DIP (0.300", 7.62mm)
Supplier Device Package	8-PDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908qt4ampe">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908qt4ampe</a>

**MC68HC908QY4A**  
**MC68HC908QT4A**  
**MC68HC908QY2A**  
**MC68HC908QT2A**  
**MC68HC908QY1A**  
**MC68HC908QT1A**

Data Sheet

***M68HC08***  
***Microcontrollers***

MC68HC908QY4A  
Rev. 3  
03/2010

[freescale.com](http://freescale.com)

14.3.3	Output Compare . . . . .	126
14.3.3.1	Unbuffered Output Compare . . . . .	127
14.3.3.2	Buffered Output Compare . . . . .	128
14.3.4	Pulse Width Modulation (PWM) . . . . .	128
14.3.4.1	Unbuffered PWM Signal Generation . . . . .	129
14.3.4.2	Buffered PWM Signal Generation . . . . .	129
14.3.4.3	PWM Initialization . . . . .	130
14.4	Interrupts . . . . .	131
14.5	Low-Power Modes . . . . .	131
14.5.1	Wait Mode . . . . .	131
14.5.2	Stop Mode . . . . .	131
14.6	TIM During Break Interrupts . . . . .	131
14.7	I/O Signals . . . . .	132
14.7.1	TIM Channel I/O Pins (TCH1:TCH0) . . . . .	132
14.7.2	TIM Clock Pin (TCLK) . . . . .	132
14.8	Registers . . . . .	132
14.8.1	TIM Status and Control Register . . . . .	132
14.8.2	TIM Counter Registers . . . . .	134
14.8.3	TIM Counter Modulo Registers . . . . .	134
14.8.4	TIM Channel Status and Control Registers . . . . .	135
14.8.5	TIM Channel Registers . . . . .	137

## Chapter 15 Development Support

15.1	Introduction . . . . .	139
15.2	Break Module (BRK) . . . . .	139
15.2.1	Functional Description . . . . .	139
15.2.1.1	Flag Protection During Break Interrupts . . . . .	141
15.2.1.2	TIM During Break Interrupts . . . . .	141
15.2.1.3	COP During Break Interrupts . . . . .	141
15.2.2	Break Module Registers . . . . .	141
15.2.2.1	Break Status and Control Register . . . . .	142
15.2.2.2	Break Address Registers . . . . .	142
15.2.2.3	Break Auxiliary Register . . . . .	143
15.2.2.4	Break Status Register . . . . .	143
15.2.2.5	Break Flag Control Register . . . . .	143
15.2.3	Low-Power Modes . . . . .	144
15.3	Monitor Module (MON) . . . . .	144
15.3.1	Functional Description . . . . .	144
15.3.1.1	Normal Monitor Mode . . . . .	148
15.3.1.2	Forced Monitor Mode . . . . .	148
15.3.1.3	Monitor Vectors . . . . .	149
15.3.1.4	Data Format . . . . .	150
15.3.1.5	Break Signal . . . . .	150
15.3.1.6	Baud Rate . . . . .	150
15.3.1.7	Commands . . . . .	150
15.3.2	Security . . . . .	154

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE0C	LVI Status Register (LVISR) <a href="#">See page 91.</a>	Read:	LVIOUT	0	0	0	0	0	0	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0D ↓ \$FE0F	Reserved									
\$FFBE	FLASH Block Protect Register (FLBPR) <a href="#">See page 34.</a>	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	Unaffected by reset							
\$FFBF	Reserved									
\$FFC0	Internal Oscillator Trim (Factory Programmed VDD = 3.0 V)	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	Unaffected by reset							
\$FFC1	Internal Oscillator Trim (Factory Programmed VDD = 5.0 V)	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	Unaffected by reset							
\$FFFF	COP Control Register (COPCTL) <a href="#">See page 63.</a>	Read:	LOW BYTE OF RESET VECTOR							
		Write:	WRITING CLEARS COP COUNTER (ANY VALUE)							
		Reset:	Unaffected by reset							

= Unimplemented     
  = Reserved     
 U = Unaffected

**Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 5)**

## 2.6 FLASH Memory (FLASH)

The FLASH memory is intended primarily for program storage. In-circuit programming allows the operating program to be loaded into the FLASH memory after final assembly of the application product. It is possible to program the entire array through the single-wire monitor mode interface. Because no special voltages are needed for FLASH erase and programming operations, in-application programming is also possible through other software-controlled communication paths.

This subsection describes the operation of the embedded FLASH memory. The FLASH memory can be read, programmed, and erased from the internal  $V_{DD}$  supply. The program and erase operations are enabled through the use of an internal charge pump.

The minimum size of FLASH memory that can be erased is 64 bytes; and the maximum size of FLASH memory that can be programmed in a program cycle is 32 bytes (a row). Program and erase operations are facilitated through control bits in the FLASH control register (FLCR). Details for these operations appear later in this section.

### NOTE

*An erased bit reads as a 1 and a programmed bit reads as a 0. A security feature prevents viewing of the FLASH contents.<sup>(1)</sup>*

### 2.6.1 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

**Figure 2-3. FLASH Control Register (FLCR)**

#### HVEN — High Voltage Enable Bit

This read/write bit enables high voltage from the charge pump to the memory for either program or erase operation. It can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

- 1 = High voltage enabled to array and charge pump on
- 0 = High voltage disabled to array and charge pump off

#### MASS — Mass Erase Control Bit

This read/write bit configures the memory for mass erase operation.

- 1 = Mass erase operation selected
- 0 = Mass erase operation unselected

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

### 3.3.4 Sources of Error

Several sources of error exist for ADC conversions. These are discussed in the following sections.

#### 3.3.4.1 Sampling Error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy. Given the maximum input resistance of approximately 15 k $\Omega$  and input capacitance of approximately 10 pF, sampling to within

1/4LSB (at 10-bit resolution) can be achieved within the minimum sample window (3.5 cycles / 2 MHz maximum ADCK frequency) provided the resistance of the external analog source ( $R_{AS}$ ) is kept below 10 k $\Omega$ . Higher source resistances or higher-accuracy sampling is possible by setting ADLSMP (to increase the sample window to 23.5 cycles) or decreasing ADCK frequency to increase sample time.

#### 3.3.4.2 Pin Leakage Error

Leakage on the I/O pins can cause conversion error if the external analog source resistance ( $R_{AS}$ ) is high. If this error cannot be tolerated by the application, keep  $R_{AS}$  lower than  $V_{ADV_{IN}} / (4096 * I_{Leak})$  for less than 1/4LSB leakage error (at 10-bit resolution).

#### 3.3.4.3 Noise-Induced Errors

System noise which occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC10 accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 $\mu$ F low-ESR capacitor from  $V_{REFH}$  to  $V_{REFL}$  (if available).
- There is a 0.1 $\mu$ F low-ESR capacitor from  $V_{DDA}$  to  $V_{SSA}$  (if available).
- If inductive isolation is used from the primary supply, an additional 1 $\mu$ F capacitor is placed from  $V_{DDA}$  to  $V_{SSA}$  (if available).
- $V_{SSA}$  and  $V_{REFL}$  (if available) is connected to  $V_{SS}$  at a quiet point in the ground plane.
- The MCU is placed in wait mode immediately after initiating the conversion (next instruction after write to ADSCR).
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive  $V_{DD}$  noise is coupled into the ADC10. In these cases, or when the MCU cannot be placed in wait or I/O activity cannot be halted, the following recommendations may reduce the effect of noise on the accuracy:

- Place a 0.01  $\mu$ F capacitor on the selected input channel to  $V_{REFL}$  or  $V_{SSA}$  (if available). This will improve noise issues but will affect sample rate based on the external analog source resistance.
- Operate the ADC10 in stop mode by setting ACLKEN, selecting the channel in ADSCR, and executing a STOP instruction. This will reduce  $V_{DD}$  noise but will increase effective conversion time due to stop recovery.
- Average the input by converting the output many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock (ACLKEN=1) and averaging. Noise that is synchronous to the ADCK cannot be averaged out.

charging. If externally available, connect the  $V_{REFL}$  pin to the same potential as  $V_{SSA}$  at the single point ground location.

### 3.7.5 ADC10 Channel Pins (ADn)

The ADC10 has multiple input channels. Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. 0.01  $\mu$ F capacitors with good high-frequency characteristics are sufficient. These capacitors are not necessary in all cases, but when used they must be placed as close as possible to the package pins and be referenced to  $V_{SSA}$ .

## 3.8 Registers

These registers control and monitor operation of the ADC10:

- ADC10 status and control register, ADSCR
- ADC10 data registers, ADRH and ADRL
- ADC10 clock register, ADCLK

### 3.8.1 ADC10 Status and Control Register

This section describes the function of the ADC10 status and control register (ADSCR). Writing ADSCR aborts the current conversion and initiates a new conversion (if the ADCH[4:0] bits are equal to a value other than all 1s).

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:								
Reset:	0	0	0	1	1	1	1	1

**Figure 3-3. ADC10 Status and Control Register (ADSCR)**

#### COCO — Conversion Complete Bit

COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the status and control register is written or whenever the data register (low) is read.

- 1 = Conversion completed
- 0 = Conversion not completed

#### AIEN — ADC10 Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of a conversion. The interrupt signal is cleared when the data register is read or the status/control register is written.

- 1 = ADC10 interrupt enabled
- 0 = ADC10 interrupt disabled

#### ADCO — ADC10 Continuous Conversion Bit

When this bit is set, the ADC10 will begin to convert samples continuously (continuous conversion mode) and update the result registers at the end of each conversion, provided the ADCH[4:0] bits do not decode to all 1s. The ADC10 will continue to convert until the MCU enters reset, the MCU enters stop mode (if ACLKEN is clear), ADCLK is written, or until ADSCR is written again. If stop is entered (with ACLKEN low), continuous conversions will cease and can be restarted only with a write to ADSCR. Any write to ADSCR with ADCO set and the ADCH bits not all 1s will abort the current conversion and begin continuous conversions.

If the bus frequency is less than the ADCK frequency, precise sample time for continuous conversions cannot be guaranteed in short-sample mode (ADLSMP = 0). If the bus frequency is less than 1/11th of the ADCK frequency, precise sample time for continuous conversions cannot be guaranteed in long-sample mode (ADLSMP = 1).

When clear, the ADC10 will perform a single conversion (single conversion mode) each time ADSCR is written (assuming the ADCH[4:0] bits do not decode all 1s).

- 1 = Continuous conversion following a write to ADSCR
- 0 = One conversion following a write to ADSCR

### ADCH[4:0] — Channel Select Bits

The ADCH[4:0] bits form a 5-bit field that is used to select one of the input channels. The input channels are detailed in Table 3-2. The successive approximation converter subsystem is turned off when the channel select bits are all set to 1. This feature allows explicit disabling of the ADC10 and isolation of the input channel from the I/O pad. Terminating continuous conversion mode this way will prevent an additional, single conversion from being performed. It is not necessary to set the channel select bits to all 1s to place the ADC10 in a low-power state, however, because the module is automatically placed in a low-power state when a conversion completes.

**Table 3-2. Input Channel Select**

ADCH4	ADCH3	ADCH2	ADCH1	ADCH0	Input Select <sup>(1)</sup>
0	0	0	0	0	AD0
0	0	0	0	1	AD1
0	0	0	1	0	AD2
0	0	0	1	1	AD3
0	0	1	0	0	AD4
0	0	1	0	1	AD5
0	0	1	1	0	Unused
Continuing through					Unused
1	1	0	0	1	Unused
1	1	0	1	0	BANDGAP REF <sup>(2)</sup>
1	1	0	1	1	Reserved
1	1	1	0	0	Reserved
1	1	1	0	1	V <sub>REFH</sub>
1	1	1	1	0	V <sub>REFL</sub>
1	1	1	1	1	Low-power state

1. If any unused or reserved channels are selected, the resulting conversion will be unknown.
2. Requires LVI to be powered (LVIPWRD = 0, in CONFIG1)

### 3.8.2 ADC10 Result High Register (ADRH)

This register holds the MSBs of the result and is updated each time a conversion completes. All other bits read as 0s. Reading ADRH prevents the ADC10 from transferring subsequent conversion results into the result registers until ADRL is read. If ADRL is not read until the after next conversion is completed, then the intermediate conversion result will be lost. In 8-bit mode, this register contains no interlocking with ADRL.






### 4.6.2 Keyboard Status and Control Register

The keyboard status and control register (KBSCR):

- Flags keyboard/auto wakeup interrupt requests
- Acknowledges keyboard/auto wakeup interrupt requests
- Masks keyboard/auto wakeup interrupt requests

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 4-3. Keyboard Status and Control Register (KBSCR)**

#### Bits 7–4 — Not used

These read-only bits always read as 0s.

#### KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending on port A or auto wakeup. Reset clears the KEYF bit.

- 1 = Keyboard/auto wakeup interrupt pending
- 0 = No keyboard/auto wakeup interrupt pending

#### ACKK — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the keyboard/auto wakeup interrupt request on port A and auto wakeup logic. ACKK always reads as 0. Reset clears ACKK.

#### IMASKK— Keyboard Interrupt Mask Bit

Writing a 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests on port A or auto wakeup. Reset clears the IMASKK bit.

- 1 = Keyboard/auto wakeup interrupt requests masked
- 0 = Keyboard/auto wakeup interrupt requests not masked

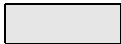
**NOTE**

*MODEK is not used in conjunction with the auto wakeup feature. To see a description of this bit, see [9.8.1 Keyboard Status and Control Register \(KBSCR\)](#).*

### 4.6.3 Keyboard Interrupt Enable Register

The keyboard interrupt enable register (KBIER) enables or disables the auto wakeup to operate as a keyboard/auto wakeup interrupt input.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
Write:								
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 4-4. Keyboard Interrupt Enable Register (KBIER)**



# Chapter 11

## Oscillator (OSC) Module

### 11.1 Introduction

The oscillator (OSC) module is used to provide a stable clock source for the MCU system and bus.

The OSC shares its pins with general-purpose input/output (I/O) port pins. See [Figure 11-1](#) for port location of these shared pins. The OSC2EN bit is located in the port A pull enable register (PTAPUEN) on this MCU. See [Chapter 12 Input/Output Ports \(PORTS\)](#) for information on PTAPUEN register.

### 11.2 Features

The bus clock frequency is one fourth of any of these clock source options:

1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to  $\pm 0.4\%$ . There are three choices for the internal oscillator, 12.8 MHz, 8 MHz, or 4 MHz. The 12.8-MHz internal oscillator is the default option out of reset.
2. External oscillator: An external clock that can be driven directly into OSC1.
3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
4. External crystal: A built-in XTAL oscillator that requires an external crystal or ceramic-resonator. There are three crystal frequency ranges supported, 8–32 MHz, 1–8 MHz, and 32–100 kHz.

### 11.3 Functional Description

The oscillator contains these major subsystems:

- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit

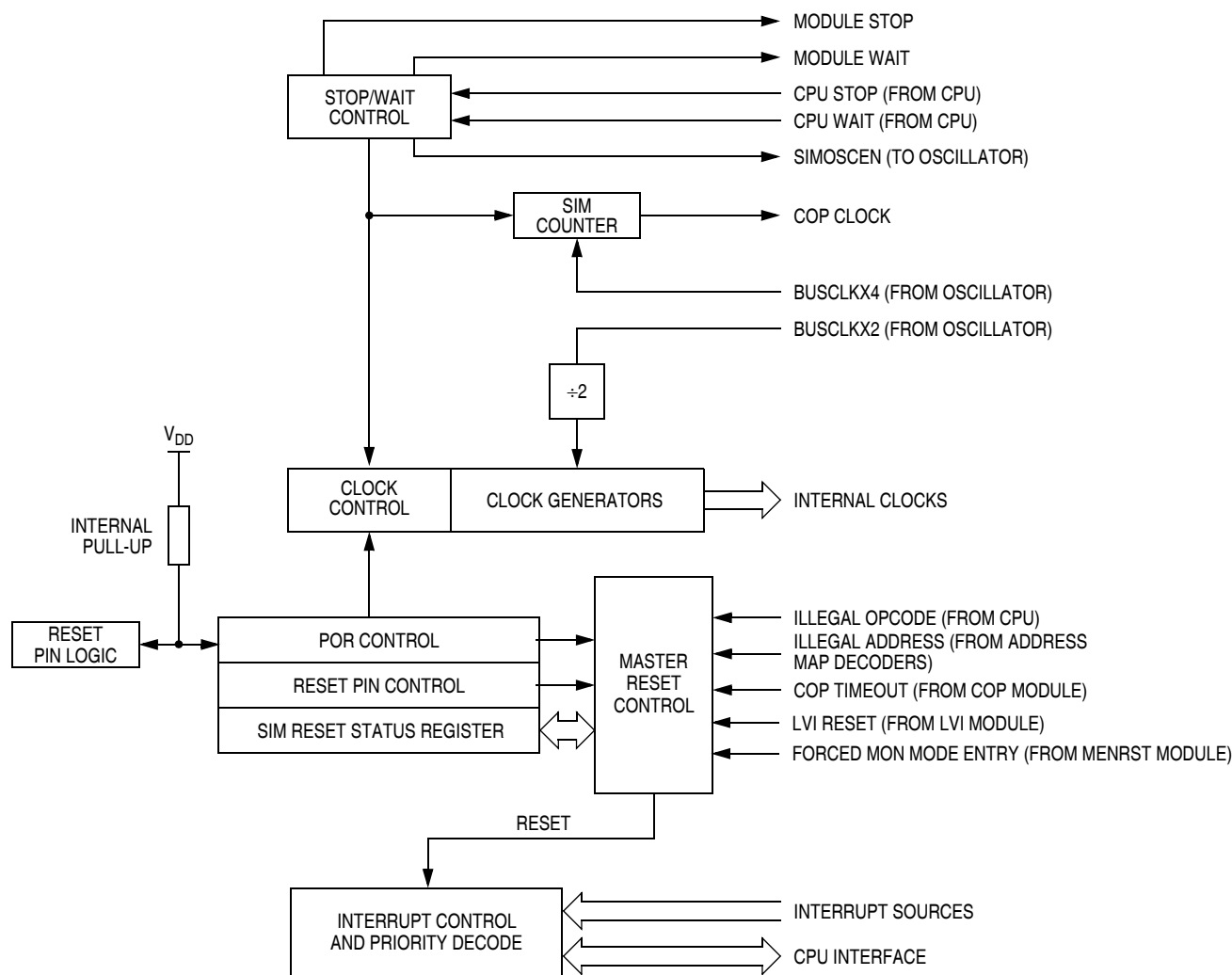


Figure 13-1. SIM Block Diagram

### 13.3 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, BUSCLKX2, as shown in Figure 13-2.

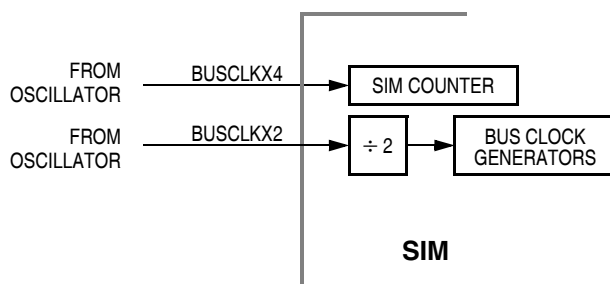


Figure 13-2. SIM Clock Signals

### 13.6.2.1 Interrupt Status Register 1

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

**Figure 13-11. Interrupt Status Register 1 (INT1)**

#### IF1–IF6 — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown in [Table 13-3](#).

1 = Interrupt request present

0 = No interrupt request present

#### Bit 0, 1 — Always read 0

### 13.6.2.2 Interrupt Status Register 2

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

**Figure 13-12. Interrupt Status Register 2 (INT2)**

#### IF7–IF14 — Interrupt Flags

This flag indicates the presence of interrupt requests from the sources shown in [Table 13-3](#).

1 = Interrupt request present

0 = No interrupt request present

### 13.6.2.3 Interrupt Status Register 3

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF22	IF21	IF20	IF19	IF18	IF17	IF16	IF15
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

**Figure 13-13. Interrupt Status Register 3 (INT3)**

#### IF15–IF22 — Interrupt Flags

These flags indicate the presence of interrupt requests from the sources shown in [Table 13-3](#).

1 = Interrupt request present

0 = No interrupt request present

If another TIM overflow occurs before the clearing sequence is complete, then writing 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Writing a 1 to TOF has no effect.

- 1 = Counter has reached modulo value
- 0 = Counter has not reached modulo value

**TOIE — TIM Overflow Interrupt Enable Bit**

This read/write bit enables TIM overflow interrupts when the TOF bit becomes set.

- 1 = TIM overflow interrupts enabled
- 0 = TIM overflow interrupts disabled

**TSTOP — TIM Stop Bit**

This read/write bit stops the counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the counter until software clears the TSTOP bit.

- 1 = Counter stopped
- 0 = Counter active

**NOTE**

*Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode. Also, when the TSTOP bit is set and the timer is configured for input capture operation, input captures are inhibited until the TSTOP bit is cleared.*

**TRST — TIM Reset Bit**

Setting this write-only bit resets the counter and the TIM prescaler. Setting TRST has no effect on any other timer registers. Counting resumes from \$0000. TRST is cleared automatically after the counter is reset and always reads as 0.

- 1 = Prescaler and counter cleared
- 0 = No effect

**NOTE**

*Setting the TSTOP and TRST bits simultaneously stops the counter at a value of \$0000.PS[2:0] — Prescaler Select Bits*

These read/write bits select one of the seven prescaler outputs as the input to the counter as [Table 14-1](#) shows.

**Table 14-1. Prescaler Selection**

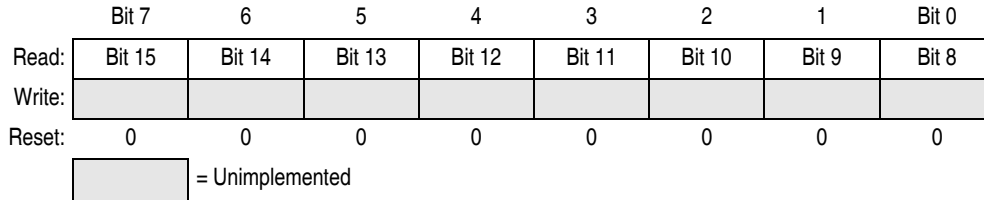
PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	TCLK (if available)

### 14.8.2 TIM Counter Registers

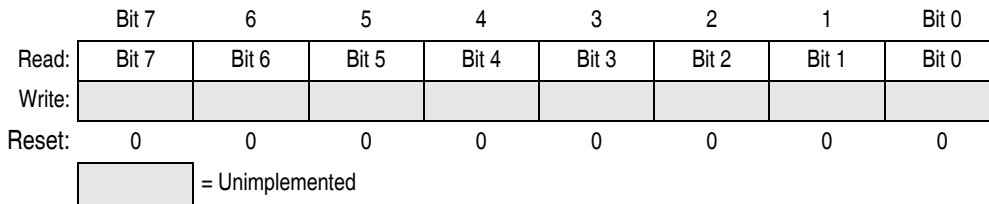
The two read-only TIM counter registers contain the high and low bytes of the value in the counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

**NOTE**

*If you read TCNTH during a break interrupt, be sure to unlatch TCNTL by reading TCNTL before exiting the break interrupt. Otherwise, TCNTL retains the value latched during the break.*



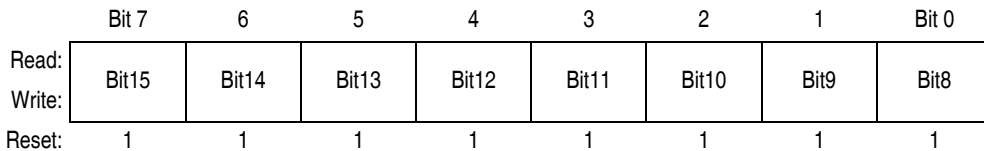
**Figure 14-5. TIM Counter High Register (TCNTH)**



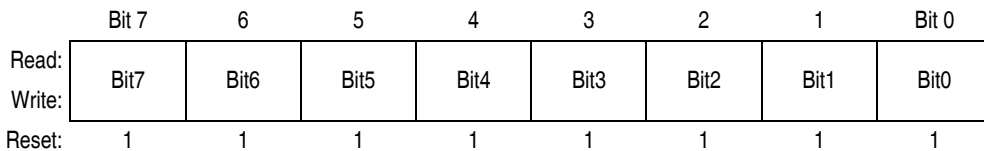
**Figure 14-6. TIM Counter Low Register (TCNTL)**

### 14.8.3 TIM Counter Modulo Registers

The read/write TIM modulo registers contain the modulo value for the counter. When the counter reaches the modulo value, the overflow flag (TOF) becomes set, and the counter resumes counting from \$0000 at the next timer clock. Writing to the high byte (TMODH) inhibits the TOF bit and overflow interrupts until the low byte (TMODL) is written. Reset sets the TIM counter modulo registers.



**Figure 14-7. TIM Counter Modulo High Register (TMODH)**

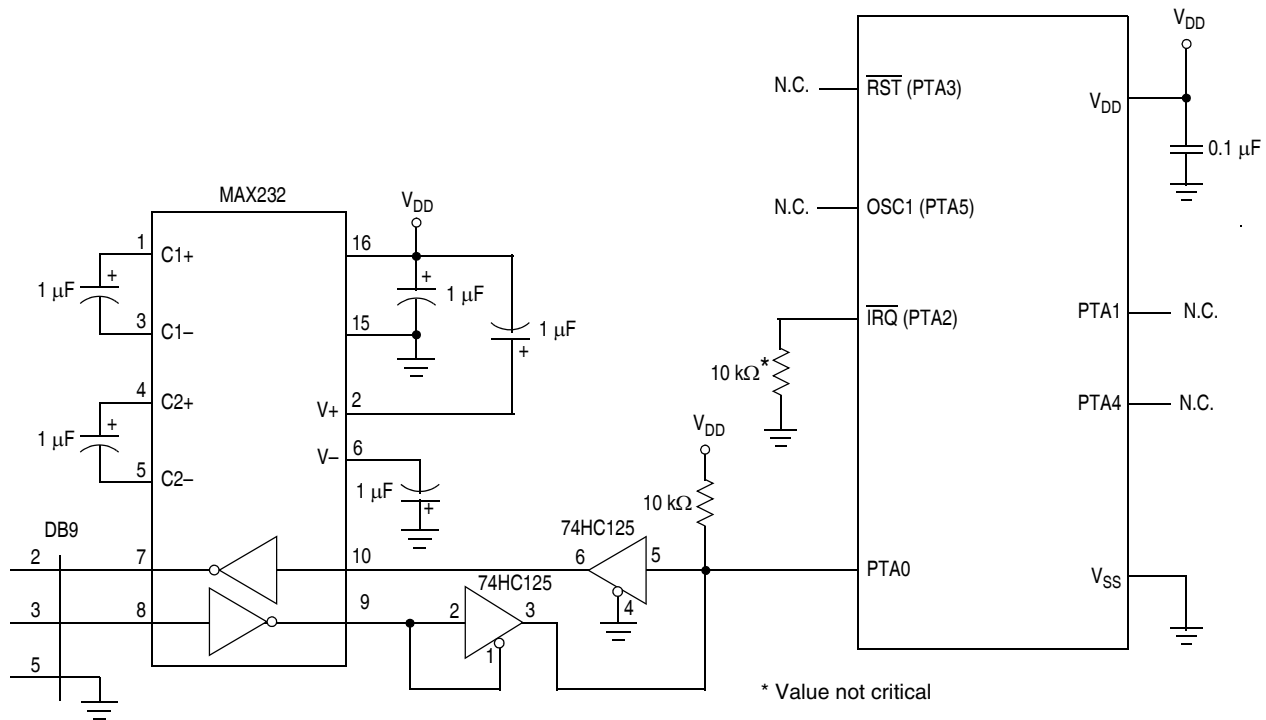


**Figure 14-8. TIM Counter Modulo Low Register (TMODL)**

**NOTE**

*Reset the counter before writing to the TIM counter modulo registers.*





**Figure 15-12. Monitor Mode Circuit (Internal Clock, No High Voltage)**

The monitor code has been updated from previous versions of the monitor code to allow enabling the internal oscillator to generate the internal clock. This addition, which is enabled when  $\overline{\text{IRQ}}$  is held low out of reset, is intended to support serial communication/programming at 9600 baud in monitor mode by using the internal oscillator, and the internal oscillator user trim value  $\text{OSCTRIM}$  (FLASH location \$FFC0, if programmed) to generate the desired internal frequency (3.2 MHz). Since this feature is enabled only when  $\overline{\text{IRQ}}$  is held low out of reset, it cannot be used when the reset vector is programmed (i.e., the value is not \$FFFF) because entry into monitor mode in this case requires  $V_{\text{TST}}$  on  $\overline{\text{IRQ}}$ . The  $\overline{\text{IRQ}}$  pin must remain low during this monitor session in order to maintain communication.

Table 15-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If \$FFFE and \$FFFF do not contain \$FF (programmed state):
  - The external clock is 9.8304 MHz
  - $\overline{\text{IRQ}} = V_{\text{TST}}$
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - The external clock is 9.8304 MHz
  - $\overline{\text{IRQ}} = V_{\text{DD}}$  (this can be implemented through the internal  $\overline{\text{IRQ}}$  pullup)
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - $\overline{\text{IRQ}} = V_{\text{SS}}$  (internal oscillator is selected, no external clock required)

The rising edge of the internal  $\overline{\text{RST}}$  signal latches the monitor mode. Once monitor mode is latched, the values on PTA1 and PTA4 pins can be changed.

Once out of reset, the MCU waits for the host to send eight security bytes (see 15.3.2 Security). After the security bytes, the MCU sends a break signal (10 consecutive 0s) to the host, indicating that it is ready to receive a command.



**Ordering Information and Mechanical Specifications**

Case 626 page 2 of 3



# Appendix A

## 908QTA/QYxA Conversion Guidelines

### A.1 Introduction

This engineering bulletin describes the 908QTA/QYxA. The 908QTA/QYxA is an enhanced device intended to replace the 908QT/QYx series of devices (referred to as the QY Classic in this document). Customer requests have led to the advanced design of the QYxA that has added adaptability, new features, and contains lead-free packaging.

This document:

- Provides information needed to convert from QY Classic to the enhanced QYxA
- Highlights the benefits of making this change

Sections:

- [A.2 Benefits of the Enhanced QYxA](#)
- [A.3 Conversion Considerations](#)
- [A.4 Code Changes Checklist](#)
- [A.5 Development Tools](#)
- [A.6 Differences in Packaging](#)

### A.2 Benefits of the Enhanced QYxA

The QYxA contains new and enhanced modules that add more flexibility and new features to the QY Classic. These benefits can improve the operation of an application or lead to new features for an application. For more information regarding these features refer to the QYxA data sheet (Freescale document order number MC68HC908QYxA).

#### A.2.1 New Analog-to-Digital Converter Module (ADC)

The QYxA contains a 10-bit ADC which replaces the 8-bit ADC on the QY Classic. This module allows both 10-bit and 8-bit conversion modes. The increased precision for ADC readings can be very useful in many applications.

Features of the ADC new 10-bit module include:

- There are two new ADC channels that have been placed on PTB0 and PTB1 allowing added flexibility especially when debugging in Monitor Mode.
  - A limitation of QY Classic debugging is that access to the ADC channels is limited because many of the QY Classic pins are multiplexed. Having extra ADC channels on the PTB pins resolves this limitation.

