Freescale Semiconductor
Application Note

AN2900
Rev. 0, 01/2005

Programming the Keyboard Interrupt Module (KBI) on HC(S)08 MCUs

by: **Gabriel Sanchez Barba**
RTAC Americas Mexico

Overview

This document is intended to serve as a quick reference for an embedded engineer to get the keyboard interrupt module up and running for any HC08 MCU. Basic knowledge about the functional description and configuration options will give the user a better understanding on how the KBI works. This application note provides examples which demonstrate one use of the KBI within the HC08 and HCS08 families of microcontrollers. The examples mentioned are intended to be modified to suit the specific needs for any application.

Keyboard Interrupt Module

The keyboard interrupt module (KBI) on HC08s can contain between 4 and 8 interrupt pins depending on the MCU (except for the HC08KH which has 20)—each with separate keyboard interrupt enable bits and one keyboard interrupt mask. These interrupts can be programmed to be either falling-edge-only, or falling-edge/low-level sensitive. The latest versions of the KBI contain rising edge also.

The main Registers found within the KBI are:

1. The keyboard status and control register (KBISCR), which:
    - Flags a keyboard interrupt request (bit 3, KEYF)
    - Helps acknowledge keyboard interrupt requests (bit 2, ACKK)
    - Can prevent the output of the keyboard interrupt mask from generating interrupt requests (bit 1, IMASK)
    - Configures keyboard interrupt triggering sensitivity as falling-edge or falling-edge and low-level; this controls all pins (bit 0, MODEK)
2. The keyboard interrupt enable (KBIE), which:
    - Is capable of enabling and disabling each port pin to operate as a keyboard interrupt (KBIx)

Please refer to the Appendix for the different HC08s families and their capabilities in regards to the KBI.

Hardware Example using the HC68HC908QY

PTA2/KBI2 is configured as a KBI input. Enabling the KBI function on this pin also enables the internal pull-up resistor; this makes the default state for this KBI input a logic ‘1’. A push button is connected as shown in Figure 1. When the button is pressed, the PTA2/KBI2 will read a logic ‘0’. PTA1 is configured as an output by setting the data direction register for PTA1, (DDRA1), and will turn on and off the LED with inverse logic. This means that the LED will turn on with a logic ‘0’ on PTA1, and it will turn off with a logic ‘1’ on the pin. For information on the calculations needed to find the value of R1, refer to application note AN1238: *HC05 MCU LED Drive Techniques Using the MC68HC705J1*.

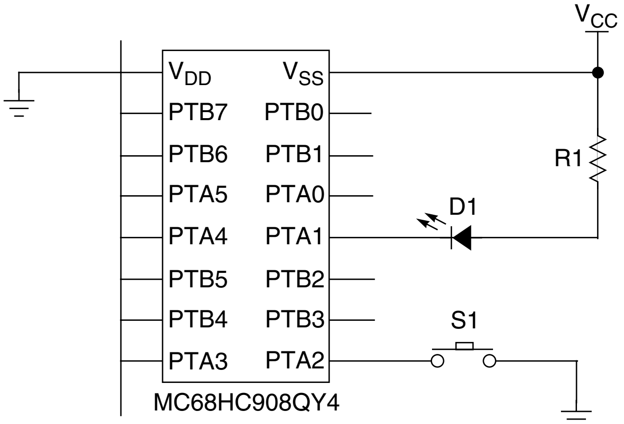

Figure 1. Hardware Example Connections

Code Example

This section contains the initialization code for the KBI using the MC68HC908QY4. Here, only pin 2 of the KBI (PTA2/KBI2) is initialized. During the initialization phase, interrupts are masked because it takes time for the internal pull up (typically 26 kΩ) to reach a logic ‘1’. Next, interrupts that may be pending are cleared, and then the keyboard interrupt is unmasked.

```
void KBI_init(void) {
/*This procedure is to prevent False Interrupts at initialization */
   KBSCR_IMASKK = 1;        /* Mask Keyboard interrupts */
   KBIER_KBIE2 = 1;         /* Enables pin2 of KBI by setting KBIE2(PTA2) */
   KBSCR_ACKK  = 1;         /* Clear any false interrupts */
   KBSCR_IMASKK = 0;        /* Unmask Keyboard interrupts */

/* END Avoidance of False Interrupts */
                    /* Configures KBI Status & Control Register */
   KBSCR = 0;       /* IMASKK=0: Clears KBI Mask Bit (Enable Ints)
                    /* MODEK=0: Interrupt requests on Falling Edge Only */
}
```

Using the routine shown above, the KBI is initialized and the program is ready for any keyboard interrupt. Whenever one occurs, the keyboard interrupt is serviced. The following interrupt service routine acknowledges the interrupt and then changes the logic state of PTA1 to blink the LED on and off.

```
interrupt 15 void _KB_Interrupt(void) {
  KBSCR_ACKK   = 1;       /* Acknowledge KB Interrupts */
  PTA_PTA1 = ~PTA_PTA1;   /* Toggles the LED */
}
```

Application Considerations

This example was developed using Metrowerks CodeWarrior IDE version 3.0 for HC(S)08, and was expressly made for the MC68HC908QY4 and for the MC68HC908GP32. Changes in the code may be required to initialize another MCU. It should also be taken into account that not all MCUs have an internal oscillator or an internal clock generator, so it may be necessary to connect and initialize a clock source. It should also be taken into consideration that a delay (20 ms typical) within the software is require to accommodate the mechanical stabilization time of the push button.

References

Refer to the following documents for more information on subjects in this application note.

- MC68HC908QY4 Technical Data Sheet
- AN1238: *HC05 MCU LED Drive Techniques Using the MC68HC705J1*

Appendix

This table provides information for the KBIs of several HC(S)08 MCUs.

	MCU	KBI	No. of KBI Pins	KBSCR Name	KBIER Name
M68HC(9)08 Family	HC(9)08AB	Yes	5	KBSCR	KBIER
	HC908AP	Yes	8	KBSCR	KBIER
	HC(9)08AS	No	—	—	—
	HC(9)08AZ	Yes	5	KBSCR	KBIER
	HC08BD	No	—	—	—
	HC908EY	Yes	5	KBSCR	KBIER
	HC908GP	Yes	8	INTKBSCR	INTKBIER
	HC908GR	Yes	4	INTKBSCR	INTKBIER
	HC908GT	Yes	8	INTKBSCR	INTKBIER
	HC908GZ	Yes	8	INTKBSCR	INTKBIER
	HC08JB	Yes	8	KBSCR	KBIER
	HC908JG	Yes	8	KBSCR	KBIER
	HC908JK	Yes	7	KBSCR	KBIER
	HC(9)08JL	Yes	7	KBSCR	KBIER
	HC08KH	Yes	20	KBxSCR	KBxIER
	HC908LD	Yes	8	KBSCR	KBIER
	HC908LJ	Yes	8	KBSCR	KBIER
	HC908LK	Yes	8	KBSCR	KBIER
	HC908MR	No	—	—	—
	HC908QF	Yes	6	KBSCR	KBIER
	HC908QT	Yes	6	KBSCR	KBIER
	HC908QY	Yes	6	KBSCR	KBIER
	HC908RF	Yes	6	INTKBSCR	INTKBIER
	HC908RK	Yes	6	INTKBSCR	KBIER
	HC908SR	Yes	8	KBSCR	KBIER
HCS08 Family	HC9S08GB	Yes	8	KB1SCR	KB1PE
	HC9S08GT	Yes	8	KB1SCR	KB1PE
	HC9S08RC	Yes	12	KB#SCR	KBI#PE
	HC9S08RD	Yes	12	KB#SCR	KBI#PE
	HC9S08RE	Yes	12	KB#SCR	KBI#PE
	HC9S08RG	Yes	12	KB#SCR	KBI#PE

x = D,E, or F
# = 1 or 2

This page is intentionally blank.

This page is intentionally blank.

This page is intentionally blank.

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

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor 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 or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor 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 Semiconductor was negligent regarding the design or manufacture of the part.