Freescale Semiconductor
Application Note

Document Number: AN3527
Rev. 0, 09/2007

Using the ColdFire Edge Port Module on the MCF521x ColdFire® Microcontroller

by: Paolo Alcántara
RTAC Americas

1 Introduction

This document is a quick reference to get the edge port module configured for the MCF521x microcontroller. Basic knowledge of the functional description and configuration options give you a better understanding of how the EPORT module works. This application note provides an example that demonstrates how to configure the EPORT module for the MCF521x microcontroller. This example can be modified to suit the specific needs of any application.

2 MCF521x EPORT Features

The edge port module (EPORT) has seven external interrupt pins, IRQ7–IRQ1. Each pin can be configured individually as a level-sensitive interrupt pin, an edge-detecting interrupt pin (rising edge, falling edge, or both), or a general-purpose input/output (GPIO) pin.

Contents

© Freescale Semiconductor, Inc., 2007. All rights reserved.

freescale™ semiconductor

2.1 Description

Main features include the following:

- Allows up to seven sources of external interrupts
- Sensitivity can be low-level or both edge detections
- Each pin can be used as interrupt or general I/O
- Schmitt triggered inputs reduce chance of false interrupts
- Operational in all three low-power modes

3 EPORT Module Explanation

The example code shows how the EPORT module works with three buttons for requesting interrupts using the M5213EVB board. The code demonstrates how to configure and use the flash module, allowing programmers to quickly include it in their projects.

The EPORT on the MCF5213 is a collection of seven external interrupt pins, each of them can be individually configured for low-level sensitivity or edge-detecting sensitivity. If edge sensitivity is selected, the edge can be specified as rising edge, falling edge or both edges. Schmitt trigger logic has been incorporated into the port design. If configured for edge sensitivity, false interrupts are avoided on slow rise/fall time transitions.

The EPORT is completely functional in DOZE or WAIT modes in low-power operation. In fact, they can be used to bring the processor out of either mode. The same is true in STOP mode, but because all internal clocks are disabled, the EPORT loses the capability to sync on edge-triggered interrupts; therefore only level-sensitive interrupts can be sensed in this mode of operation.

To use an EPORT pin as an external interrupt request source, it must be configured to the EPORT function employing GPIO pin assignment register and configuring it as input respective bits using the EPORT register. In EPORT mode, GPIO registers for input/output are no longer necessary. In GPIO mode, EPORT input/output registers have no effect.

3.1 Example: Case Study for the EPORT Module

The example code uses the following external hardware in M5213EVB:

- ABORT button
- SW1 button
- SW2 button
- LED[1:4]

You can press any of the three buttons to test the example code in the following way:

ABORT button has the highest interrupt level (level 7).

- Each time IRQ7 interrupt service routine (ISR) is executed, 0x05 is displayed on LEDs for a few moments.

SW2 button has the medium interrupt level (level 5)

- Each time IRQ5 ISR is executed, 0x0C is displayed on LEDs for a few moments

SW1 button has the lowest interrupt level (level 4).

- Each time IRQ4 ISR is executed, 0x03 is displayed on LEDs for a few moments

Table 1 shows some button combinations. If an interrupt is being executed, two cases can occur:

- While the interrupt request with an IRQ level above the current ISR level is executed, the current interrupt is switched by the new interrupt request. This new interrupt finishes and the old interrupt returns to execute at the code line before losing execution.
- While the interrupt request with an IRQ level below the current ISR level is being executed, the current interrupt finishes and only afterwards the pending interrupt begins execution.

Table 1. Example Cases for EPORT Software Demo

Situation	Board will
Interrupt pending: Press SW2 (level 5) and a second later press SW1 (level 4)	First LEDs display 0x0C, then 0x03 and finally LEDs will be off.
Current interrupt switched: Press SW1 (level 4) and a second later press SW2 (level 5)	First LEDs display 0x03, ISR is interrupted and display 0x0C, after that LEDs return to 0x03 and finally LEDs will be off.
Interrupt Pending and other switched: Press SW2 (level 5), a second later press SW1 (level 4) and an instant later press ABORT button (level 7).	First LEDs display 0x0C, ISR is interrupted by ABORT button showing 0x05 value, after that LEDs return to 0x0C, next pending interrupt appear showing 0x03 and finally LEDs will be off.
Idle case: Any button.	At board start or if all IRQ ISRs have finished, LEDs will be off.

3.2 Example: Configuring the CFM Module

Table 2 shows the EPORT registers.

Table 2. EPORT Registers

EPORT register	Function
EPPAR	Configure EPORT pins for level detection, rising, and/or falling edge detection.
EPDDR	Configure EPORT pins as input or output.
EPIER	Enable EPORT interrupt request per pin.
EPDR	Writes in this register are reflected in EPORT pins if configured as output.
EPPDR	This register reflects EPORT pin state if configured as input.
EPFR	Only set if EPORT pin detects an edge event.

In the example code, each of the three pins are configured to detect an edge rising event and changes the M5213EVB LEDs according to the IRQ pin source.

3.3 Register Configuration

In this section the registers configuration that affects the entire EPORT module and the code lines used are described.

3.4 Port NQ Pin Assignment Register

The following code lines are used to configure the PNQPAR to assign EPORT functionality, see Figure 1.

```
  /* Enable IRQ signals on the port */
MCF_GPIO_PNQPAR = 0
    | MCF_GPIO_PNQPAR_IRQ4_IRQ4
    | MCF_GPIO_PNQPAR_IRQ5_IRQ5
    | MCF_GPIO_PNQPAR_IRQ7_IRQ7;
```

IPSBAR Access: User read/write
Offset: 0x10_0050 (PNQPAR)

	7	6	5	4	3	2	1	0
R W	PNQPAR7	PNQPAR6	PNQPAR5	PNQPAR4	PNQPAR3	PNQPAR2	PNQPAR1	
Reset:	1	1	1	1	1	1	0	1

Figure 1. Port NQ Pin Assignment Register (PNQPAR)

3.4.1 EPORT Pin Assignment Register

To configure the EPPAR, the next code lines are used. The changes performed on the register are shown in Figure 2.

```
/* Set EPORT to look for rising edges */
MCF_EPORT_EPPAR = 0
    | MCF_EPORT_EPPAR_EPPA4_RISING
    | MCF_EPORT_EPPAR_EPPA5_RISING
    | MCF_EPORT_EPPAR_EPPA7_RISING;
```

IPSBAR Access: Supervisor read/write
Offset: 0x13_0000 (EPPAR)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EPPA7		EPPA6		EPPA5		EPPA4		EPPA3		EPPA2		EPPA1		0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2. EPORT Pin Assignment Register

3.4.2 EPORT Interrupt Enable Register:

To write the values shown in Figure 3, the program uses these lines in the register EPIER:

```
/* Clear any currently triggered events on the EPORT */
MCF_EPORT_EPIER = 0
    | MCF_EPORT_EPIER_EPIE4
    | MCF_EPORT_EPIER_EPIE5
    | MCF_EPORT_EPIER_EPIE7;
```

IPSBAR 0x13_0003 (EPIER) Access: Supervisor read/write
Offset:

	7	6	5	4	3	2	1	0
R	EPIE7	0EPIE6	EPIE5	EPIE4	EPIE3	EPIE2	EPIE1	0
W								
Reset:	0	0	0	0	0	0	0	0

Figure 3. EPORT Port Interrupt Enable Register

3.4.3 Interrupt Mask Register:

To write the values shown in Figure 4 on the IMRL, the program uses this code:

```
/* Enable interrupts in the interrupt controller */
MCF_INTC_IMRL &= ~(0
    | MCF_INTC_IMRL_MASK4
    | MCF_INTC_IMRL_MASK5
    | MCF_INTC_IMRL_MASK7
    | MCF_INTC_IMRL_MASKALL);
```

IPSBAR Access: Read/write
Offset: 0x0C08 (IMRH*n*)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INT_MASK[63:48]															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT_MASK[47:32]															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 4. Interrupt Mask Register High

3.4.4 Status Register:

In the next code lines, the status register is configured according to Figure 5. To modify SR[I], SR[S] must be set.

```
/*
 * Modify 5213 interrupt level to zero
 * to allow all interrupt-level-request above zero
 */
asm(move.w #0x2000,SR);
```

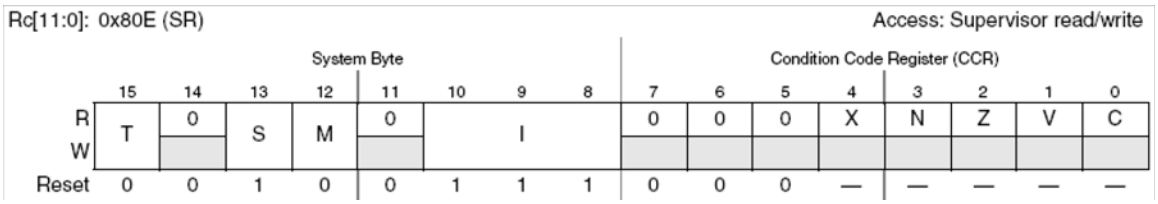

Figure 5. Status Register

3.5 Configuration Summary

The following steps are needed to configure IRQ pins to allow external events using edge detect.

1. Modify vectors.s file providing ISR (interrupt service routine) address. For details check vector.s in EPORT SoftwareDemo software
2. Configure EPORT pins as the primary functions in the assignment register
3. Writing in the EPPAR register sets the edge that is detected (rising, falling or both) by each EPORT pin
4. Clear any currently triggered event in the EPORT using the EPIER register
5. Enable interrupts in the interrupt controller using the IMRL register
6. Modify 5213 interrupt level to zero to allow all interrupt-level-requests above zero. By default the interrupt level is set to 7
7. For each ISR:

   Read IRQ pin state until it is deasserted using EPPDR register

   Execute a specific action according to the respective ISR

   Clear the interrupt event before leaving ISR by setting the respective bit in the EPFR register

4 Configuration Notes

The following details are considered important when configuring and using the EPORT:

- If using EPORT functionality, configure pin direction (input or output). The EPORT feature only works with input direction. Setting as output disables EPORT capability.
- The IRQ interrupt level associated can not be changed
- The only way to exit from STOP mode using IRQ pins is configuring selected EPORT pins to detect level changes, because in this power consumption level there is no clock to detect edges
- To guarantee that a level-sensitive interrupt request is acknowledged, the interrupt source must keep the signal asserted until acknowledge by the software

5 Conclusion

The EPORT module is a useful part in MCF521x ColdFire processors. To prioritize these events, each pin can be used to communicate an action to be executed. Each pin has a different fixed level. For example: The IRQ7 is always executed every time an event is recognized, even if a lower IRQ level is in execution

(unmaskable capability). The other IRQs can be programmed and configured to inhibit other IRQs. While an event is being executed and there is an interrupt, the new event may go into pending or be executed immediately leaving the prior event in pending. All these features allow the programmer to build robust applications to handle external events without losing control over other modules.

5.1 Considerations and References

Find the newest software updates and configuration files for the MCF521x on the Freescale Semiconductor home page: www.freescale.com

- This application note considers MCF5211, MCF5212, and MCF5213 devices.
- M5213EVB development board employed EPORT software demo.
- For more information on edge port module refer to MCF5213 ColdFire Integrated Microcontroller Reference Manual, Rev 2.0, at www.freescale.com
- The EPORT SoftwareDemo software was developed in CodeWarrior for ColdFire v6.3.
- Download the source files for EPORTSoftwareDemo software (EPORTSoftwareDemo.zip) from www.freescale.com.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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

Document Number: AN3527
Rev. 0
09/2007

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.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale’s Environmental Products program, go to http://www.freescale.com/epp.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.
© Freescale Semiconductor, Inc. 2007. All rights reserved.

freescale™
semiconductor