AN204379

FM3, Interrupt Factor Vector Relocate Function Usage

This application note explains the usage of the "interrupt factor Vector relocate function" mounted on FM3.

Contents

1 Introduction....1
2 Interrupt factor Vector relocate function....1
3 Correction of file....6
3.1 File to be corrected....6
3.2 Correction of startup routine....6
3.3 Correction of IO definition file....8
4 Setting of interrupt factor Vector relocate function....9
4.1 4.1 Setting of register....9
5 Example of setting (sample program)....10
5.1 Interrupt factor definition file....10
5.2 Sample program....10
6 Precautions....11
7 Additional Information....11
7.1 Target products....11
Document History....12

1 Introduction

This application note explains the usage of the "interrupt factor Vector relocate function" mounted on FM3.

2 Interrupt factor Vector relocate function

The interrupt factor Vector relocate function switches the assigning of the interrupt factor Vector set to FM3. Two types of interrupt factor Vector assigning shown in Table 2-1 can be selected by setting the IRQCMODE bit of the IRQCMODE register. When IRQCMODE is set to 1, the interrupt factors shown in Table 2-2 can be selected for IRQ No.3-10. For the selection of interrupt factor, the INTSELx bit of the RCINTSEL0/1 register is used.

Table 1. Exceptions and Interrupt factor vectors list

Vector No.	IRQ No.	Exceptions and Interrupt factor vectors	
		IRQCMODE=0*	IRQCMODE=1
0	-	Stack pointer initial value	
1	-	Reset	
2	-	Non-Maskable Interrupt (NMI) / Hardware Watchdog Timer	
3	-	Hard Fault	
4	-	Memory Management	
5	-	Bus Fault	
6	-	Usage Fault	
7	-	Reserved	
1	-	SVCall (Supervisor Call)	
1	-	Debug Monitor	
1	-	Reserved	
1	-	PendSV	
1	-	SysTick	

Vector No.	IRQ No.	Exceptions and Interrupt factor vectors	
		IRQCMODE=0*	IRQCMODE=1
1	0	Anomalous Frequency Detection by Clock Supervisor (FCS)	
1	1	Software Watchdog Timer	
1	2	Low Voltage Detector (LVD)	
1	3	MFT unit0, unit1, unit2 Wave Form Generator / DTIF(Motor Emergency Stop)	Selecting the interrupt factor with RCINTSEL0 register
2	4	External Pin Interrupt ch.0 to ch.7	Selecting the interrupt factor with RCINTSEL0 register
2	5	External Pin Interrupt ch.8 to ch.31	Selecting the interrupt factor with RCINTSEL0 register
2	6	Dual Timer / Quad Counter (QPRC) ch.0, ch.1, ch.2	Selecting the interrupt factor with RCINTSEL0 register
2	7	Reception Interrupt of MFS ch.0	Selecting the interrupt factor with RCINTSEL1 register
2	8	Transmission Interrupt and Status Interrupt of MFS ch.0	Selecting the interrupt factor with RCINTSEL1 register
2	9	Reception Interrupt of MFS ch.1	Selecting the interrupt factor with RCINTSEL1 register
2	10	Transmission Interrupt and Status Interrupt of MFS ch.1	Selecting the interrupt factor with RCINTSEL1 register
2	11	Reception Interrupt of MFS ch.2	MFT unit0 Wave Form Generator / DTIF(Motor Emergency Stop)
2	12	Transmission Interrupt and Status Interrupt of MFS ch.2	External pin interrupt ch.0 to ch.7
2	13	Reception Interrupt of MFS ch.3	External pin interrupt ch.8 to ch.31
3	14	Transmission Interrupt and Status Interrupt of MFS ch.3	Dual timer / Quad counter (QPRC) ch.0
3	15	Reception Interrupt of MFS ch.4	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.0
3	16	Transmission Interrupt and Status Interrupt of MFS ch.4	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.1

Vector No.	IRQ No.	Exceptions and Interrupt factor vectors	
		IRQCMODE=0*	IRQCMODE=1
3	17	Reception Interrupt of MFS ch.5	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.2
3	18	Transmission Interrupt and Status Interrupt of MFS ch.5	Reception Interrupt, Transmission Interrupt and Status Interrupt of MFS ch.3
3	19	Reception Interrupt of MFS ch.6	Reception Interrupt of MFS ch.4
3	20	Transmission Interrupt and Status Interrupt of MFS ch.6	Transmission Interrupt and Status Interrupt of MFS ch.4
3	21	Reception Interrupt of MFS ch.7	Reception Interrupt of MFS ch.5
3	22	Transmission Interrupt and Status Interrupt of MFS ch.7	Transmission Interrupt and Status Interrupt of MFS ch.5
3	23	PPG ch.0/2/4/8/10/12/16/18/20	
4	24	External Main OSC / External Sub OSC / Main PLL / PLL for USB/Watch Counter/Real Time Counter	
4	25	A/D Converter unit0	
4	26	A/D Converter unit1	
4	27	A/D Converter unit2 / LCD Controller	
4	28	MFT unit0, unit1, unit2 Free-run Timer	MFT unit0 Free-run Timer, Input Capture, Output Compare
4	29	MFT unit0, unit1, unit2 Input Capture	MFT unit1 Free-run Timer, Input Capture, Output Compare
4	30	MFT unit0, unit1, unit2 Output Compare	MFT unit2 Free-run Timer, Input Capture, Output Compare
4	31	Base Timer ch.0 to ch.7	
4	32	CAN ch.0 / Ethernet ch.0	
4	33	CAN ch.1 / Ethernet ch.1	
5	34	USB ch.0 Function (DRQ of End Point 1 to 5)	

Vector No.	IRQ No.	Exceptions and Interrupt factor vectors	
		IRQCMODE=0*	IRQCMODE=1
5	35	USB ch.0 Function (DRQI of End Point 0, DRQO and each status) /USB ch.0 HOST (each status)	
5	36	USB ch.1 Function (DRQ of End Point 1 to 5) / HDMI-CEC, Remote Control Reception ch.0	
5	37	USB ch.1 Function (DRQI of End Point 0, DRQO and each status) /USB ch.1 HOST (each status) / HDMI-CEC, Remote Control Reception ch.1	
5	38	DMA Controller (DMAC) ch.0	DMA DMA Controller (DMAC) ch.0 to ch.7
5	39	DMA Controller (DMAC) ch.1	MFT unit1 Wave Form Generator / DTIF(Motor Emergency Stop)
5	40	DMA Controller (DMAC) ch.2	MFT unit2 Wave Form Generator / DTIF(Motor Emergency Stop)
5	41	DMA Controller (DMAC) ch.3	Quad counter (QPRC) ch.1
5	42	DMA Controller (DMAC) ch.4	Reception Interrupt of MFS ch.6
5	43	DMA Controller (DMAC) ch.5	Transmission Interrupt and Status Interrupt of MFS ch.6
6	44	DMA Controller (DMAC) ch.6	Reception Interrupt of MFS ch.7
6	45	DMA Controller (DMAC) ch.7	Transmission Interrupt and Status Interrupt of MFS ch.7
6	46	Base Timer ch.8 to ch.15	
6	47	Flash RDY, HANG interrupt	Quad counter (QPRC) ch.2 / Flash RDY, HANG interrupt

*: Compatible with TYPE0 to TYPE2.

Table 2. Selection interrupt factor

Setting value of RCINTSELx:INTSELx	Interrupt Factor
0x00	No interrupt factor is selected
0x01	External interrupt ch.0
0x02	External interrupt ch.1
0x03	External interrupt ch.2
0x04	External interrupt ch.3
0x05	External interrupt ch.4
0x06	External interrupt ch.5
0x07	External interrupt ch.6
0x08	External interrupt ch.7
0x09	External interrupt ch.8
0x0A	External interrupt ch.9
0x0B	External interrupt ch.10
0x0C	External interrupt ch.11
0x0D	IRQ0/IRQ1 of the base timer ch.0
0x0E	IRQ0/IRQ1 of the base timer ch.1
0x0F	IRQ0/IRQ1 of the base timer ch.2
0x10	IRQ0/IRQ1 of the base timer ch.3
0x11	IRQ0/IRQ1 of the base timer ch.4
0x12	IRQ0/IRQ1 of the base timer ch.5
0x13	IRQ0/IRQ1 of the base timer ch.6
0x14	IRQ0/IRQ1 of the base timer ch.7
0x15	Reception interrupt of MFS ch.0
0x16	Reception interrupt of MFS ch.1
0x17	Reception interrupt of MFS ch.2
0x18	Reception interrupt of MFS ch.3
0x19	Zero detection interrupt of MFT unit0 free-run timer ch.0
0x1A	Zero detection interrupt of MFT unit1 free-run timer ch.0
0X1B	Zero detection interrupt of MFT unit2 free-run timer ch.0
0x1C	DMAC ch.0
0x1D	DMAC ch.1
0x1E	DMAC ch.2
0x1F	DMAC ch.3
0x20 to 0xFF	Reserved

3 Correction of file

This chapter explains the method for correcting a file required for using the interrupt factor Vector relocate function.

3.1 File to be corrected

If the interrupt factor is changed by setting of the relocate function, it is necessary to correct the following files of the FM3 template project.

- Startup routine (startup_mb9xfxxx.s)
- IO definition file (mb9xxxxx.h)

3.2 Correction of startup routine

In the startup routine, the interrupt handler is defined.

When you use the relocate function, the definition names of the interrupt handlers of "IRQ No.3 to No.22, No.28 to No.30, No.38 to No.45, No.47" of startup_mb9xfxxx.s do not correspond to the interrupt names after relocation. Use it after changing the definition names of the interrupt handlers. For the definition name to be used, an arbitrary name can be given.

An Example for changing the interrupt handler of IRQ No.3 from default "multifunction timer waveform generator/DTIF" to "external interrupt ch.0" is shown in the figures. Figure 1 is for ARM and Figure 2 is for IAR.

Figure 1. Example of definition of interrupt handler (for ARM)

e.g.) Definition of the interrupt handler of IRQ No.3 is changed from "multifunction timer waveform generator/DTIF" to "external interrupt ch.0" (for ARM)

・File name: example¥ARM¥startup_mb9xfxxx.s

<Before change>

```
__Vectors       DCD     __initial_sp              ; Top of Stack
                DCD     Reset_Handler             ; Reset Handler
                DCD     NMI_Handler               ; NMI Handler
                ・・・
                DCD     CSV_Handler               ; 0: Clock Super Visor
                DCD     SWDT_Handler              ; 1: Software Watchdog Timer
                DCD     LVD_Handler               ; 2: Low Voltage Detector
                DCD     MFT_WG_IRQHandler         ; 3: Wave Form Generator / DTIF
・・・
Default_Handler PROC
                ・・・
                EXPORT  CSV_Handler               [WEAK]
                EXPORT  SWDT_Handler              [WEAK]
                EXPORT  LVD_Handler               [WEAK]
                EXPORT  MFT_WG_IRQHandler         [WEAK]
・・・
CSV_Handler
SWDT_Handler
LVD_Handler
MFT_WG_IRQHandler
```

<After change>

```
__Vectors       DCD     __initial_sp              ; Top of Stack
                DCD     Reset_Handler             ; Reset Handler
                DCD     NMI_Handler               ; NMI Handler
                ・・・
                DCD     CSV_Handler               ; 0: Clock Super Visor
                DCD     SWDT_Handler              ; 1: Software Watchdog Timer
                DCD     LVD_Handler               ; 2: Low Voltage Detector
                DCD     INT0_Handler              ; 3: External Interrupt ch.0 (INTSEL0)
・・・
Default_Handler PROC
                ・・・
                EXPORT  CSV_Handler               [WEAK]
                EXPORT  SWDT_Handler              [WEAK]
                EXPORT  LVD_Handler               [WEAK]
                EXPORT  INT0_Handler              [WEAK]
・・・
CSV_Handler
SWDT_Handler
LVD_Handler
INT0_Handler
```

Figure 2. Example of definition of interrupt handler (for IAR)

e.g.) Definition of the interrupt handler of IRQ No.3 is changed from "multifunction timer waveform generator/DTIF" to "external interrupt ch.0" (for IAR)

・File name: example¥IAR¥startup_mb9xfxxx.s

<Before change>

```
__vector_table  DCD     sfe(CSTACK)             ; Top of Stack
                DCD     Reset_Handler           ; Reset
                DCD     NMI_Handler             ; NMI
                ・・・
                DCD     CSV_Handler             ; 0: Clock Super Visor
                DCD     SWDT_Handler            ; 1: Software Watchdog Timer
                DCD     LVD_Handler             ; 2: Low Voltage Detector
                DCD     MFT_WG_IRQHandler       ; 3: Wave Form Generator / DTIF
・・・
                PUBWEAK SWDT_Handler
                SECTION .text:CODE:REORDER(1)
SWDT_Handler
                B       SWDT_Handler

                PUBWEAK LVD_Handler
                SECTION .text:CODE:REORDER(1)
LVD_Handler
                B       LVD_Handler

                PUBWEAK MFT_WG_IRQHandler
                SECTION .text:CODE:REORDER(1)
MFT_WG_IRQHandler
                B       MFT_WG_IRQHandler
```

<After change>

```
__vector_table  DCD     sfe(CSTACK)             ; Top of Stack
                DCD     Reset_Handler           ; Reset
                DCD     NMI_Handler             ; NMI
                ・・・
                DCD     CSV_Handler             ; 0: Clock Super Visor
                DCD     SWDT_Handler            ; 1: Software Watchdog Timer
                DCD     LVD_Handler             ; 2: Low Voltage Detector
                DCD     INT0_Handler            ; 3: External Interrupt ch.0 (INTSEL0)
・・・
                PUBWEAK SWDT_Handler
                SECTION .text:CODE:REORDER(1)
SWDT_Handler
                B       SWDT_Handler

                PUBWEAK LVD_Handler
                SECTION .text:CODE:REORDER(1)
LVD_Handler
                B       LVD_Handler

                PUBWEAK INT0_Handler
                SECTION .text:CODE:REORDER(1)
INT0_Handler
                B       INT0_Handler
```

3.3 Correction of IO definition file

In the IO definition file, the interrupt number for the interrupt factor is defined. The section that defines the interrupt number is

"typedef enum IRQn { ・ ・ ・}IRQn_Type;" of "Interrupt Number Definition."

When you use the relocate function, the definitions of interrupt factors of "IRQ No.3 to No.22, No.28 to No.30, No.38 to No.45, No.47" of mb9xxxxx.h do not correspond to the interrupt names after relocation. Use it after changing the definition names of the interrupt factors. For the definition name to be used, an arbitrary name can be given.

Figure 3 shows an example for changing the interrupt factor definition of IRQ No.3 from default "multifunction timer waveform generator/DTIF" to "external interrupt ch.0".

Figure 3. Example of definition of interrupt factor

e.g.) Definition of the interrupt trigger of IRQ No.3 is changed from "multifunction timer waveform generator/DTIF" to "external interrupt ch.0"

・File name: common¥mb9xxxxx.h

<Before change>

```
typedef enum IRQn
{
    NMI_IRQn          = -14, /*  2 Non Maskable      */
    HardFault_IRQn    = -13, /*  3 Hard Fault        */
・・・
    CSV_IRQn          =  0, /* Clock Super Visor                          */
    SWDT_IRQn         =  1, /* Software Watchdog Timer                    */
    LVD_IRQn          =  2, /* Low Voltage Detector                       */
    WFG_IRQn          =  3, /* Wave Form Generator                         */
・・・
} IRQn_Type;
```

<After change>

```
typedef enum IRQn
{
    NMI_IRQn          = -14, /*  2 Non Maskable      */
    HardFault_IRQn    = -13, /*  3 Hard Fault        */
・・・
    CSV_IRQn          =  0, /* Clock Super Visor                          */
    SWDT_IRQn         =  1, /* Software Watchdog Timer                    */
    LVD_IRQn          =  2, /* Low Voltage Detector                       */
    EXINT0_IRQn       =  3, /* External Interrupt ch.0 (INTSEL0 setting)    */
・・・
} IRQn_Type;
```

4 Setting of interrupt factor Vector relocate function

This chapter explains the setting of registers required for using the interrupt factor Vector relocate function.

4.1 4.1 Setting of register

To use the relocate function, it is necessary to set the following registers.

- Interrupt factor Vector relocate setting register (IRQCMODE)
- Interrupt factor selection register 0 (RCINTSEL0)
- Interrupt factor selection register 1 (RCINTSEL1)

With the RCINTSEL0/1 register, set the interrupt factor to be allocated to IRQ No.3 to No.10 (Vectors No.19 to No.26). The settable interrupt factors are shown in "Table 2." The RCINTSEL0 register corresponds to IRQ No.3 to No.6 and the RCINTSEL1 register corresponds to IRQ No.7 to No.10. The initial value of the RCINTSEL0/1 register is set to "no interrupt factor is selected."

The IRQCMODE register performs the switching of the relocate function. The interrupt factor Vector is switched as shown in "Table 4.1 List of exceptions and interrupt factor vectors" by setting the IRQCMODE bit to ”1.”

Figure 4 shows an example of the register setting.

Figure 4. Setting of Register

```
e.g.) Setting of register

  FM3_INTREQ->RCINTSEL0 |= 0x00000001;   /* bit7-0:INTSEL0=0x01 Select Ext-Int ch.0 */
  FM3_INTREQ->RCINTSEL0 |= 0x00000200;   /* bit15-8:INTSEL1=0x02 Select Ext-Int ch.1 */
  FM3_INTREQ->RCINTSEL0 |= 0x00030000;   /* bit23-16:INTSEL2=0x03 Select Ext-Int ch.2 */
  FM3_INTREQ->RCINTSEL0 |= 0x04000000;   /* bit31-24:INTSEL3=0x04 Select Ext-Int ch.3 */

  FM3_INTREQ->RCINTSEL1 |= 0x0000000D;   /* bit7-0:INTSEL4=0x0D Select BT ch.0 IRQ0/1 */
  FM3_INTREQ->RCINTSEL1 |= 0x00001500;   /* bit15-8:INTSEL5=0x15 Select MFS ch.0 RX */
  FM3_INTREQ->RCINTSEL1 |= 0x00190000;   /* bit23-16:INTSEL6=0x19 Select MFTO FRT ch.0 */
  FM3_INTREQ->RCINTSEL1 |= 0x1C000000;   /* bit31-24:INTSEL7=0x1C Select DMAC ch.0 */

  FM3_INTREQ->IRQCMODE = 0x00000001;    /* bit0:IRQCMODE=0b1 Interrupt Vector Relocate */
```

5 Example of setting (sample program)

This chapter shows an example of the sample program using the interrupt factor Vector relocate function.

5.1 Interrupt factor definition file

The startup routine (startup_mb9xfxxx.s) and the IO definition file (mb9xxxxx.h) in which the interrupt factors are defined use the definition file corrected in "3 Correction of file."

5.2 Sample program

The sample program of the Vector relocate function is shown in Figure 5-1. The setting of the sample program is shown below.

- External interrupt ch.0 is set for IRQ No.3
- For IRQ No.4 to No.10, an interrupt factor is not selected
- Interrupt factor Vector relocate function ON
- Terminal of external interrupt ch.0 uses INT00_0
- External interrupt ch.0 detects a falling edge

Figure 5. Sample Program

```
File name: main.c
                                    External interrupt ch.0 is selected for IRQ No.3
                                    For IRQ No.4 to 10, the interrupt trigger is not selected
int32_t main( void )                (default)
{
  /* setting Interrupt Factor Vector Relocate */
  FM3_INTREQ->RCINTSEL0 = 0x00000001;   /* bit7-0:INTSEL0=0x01 Select Ext-Int ch.0 */

  FM3_INTREQ->RCINTSEL1 = 0x00000000;   /* Interrupt request no select */

  FM3_INTREQ->IRQCMODE = 0x00000001;    /* bit0:IRQCMODE=0b1 Interrupt Vector Relocate */
                                        << Interrupt trigger Vector relocate function ON
  /* setting INT00_0 */
  FM3_GPIO->EPFR06 = 0x00000000;        /* bit5,4:EINT00S=0b00 EINT-ch0 use INT00_0 */

  FM3_GPIO->PFR5 |= 0x0001;             /* bit2:PFR5_0=0b1 P50 use peripheral port */

  /* setting External Interrupt ch.0 */
  FM3_EXTI->ENIR  = 0x00000000;         /* INT interrupt disable */

  FM3_EXTI->ELVR  = 0x00000003;         /* bit1,0:LB0,LA0=0b11 INT0 low level edge */

  FM3_EXTI->EICL  = 0x00000000;         /* bit2:ECL=0b0 INT0 interrupt request clear */

  FM3_EXTI->ENIR  = 0x00000001;         /* bit2:EN0=0b1 enable INT0 */

  /* interrupt priority setting */
  NVIC_SetPriority( EXINT0_IRQn , 0x00);  << Name defined by the IO definition file is used

  /* interrupt enable set bit setting for INT00_0 interrupt */
  NVIC_EnableIRQ( EXINT0_IRQn );
                                        << Name defined by the IO definition file is used
  /* main loop */
  while(1){;}

}

/*****************************
 ** Interrupt Handler
 *****************************/
void INT0_Handler (void)   << Name defined by the startup routine is used
{
  FM3_EXTI->EICL &= 0xFFFFFFFE;                 /* clear interrupt request */
}
```

6 Precautions

- An interrupt factor selected by the RCINTSEL0/1 register is masked by IRQ11 to IRQ47. (The relevant bits of IRQ11MON to IRQ47MON registers are also masked.)
- Set the interrupt factor to be selected by INTSEL0 to INTSEL7 bits without duplication.
- The bit field definition and the bit band definition of the IRQxxMON register in the IO definition file are definitions before relocation. Note that the definition after relocation is different. When needed to use the IRQxxMON register after relocation, use the register definition.

e.g.) Read access to IRQ12MON using register definition

```
Unsigned int tmp;
tmp = FM3_INTREQ->IRQ12MON;
```

7 Additional Information

7.1 Target products

This application note is described about below products;

(TYPE4)

Series	Product Number (not included Package suffix)
MB9B110R	MB9BF112N, MB9BF112R, MB9BF114N, MB9BF114R, MB9BF115N, MB9BF115R, MB9BF116N, MB9BF116R
MB9B310R	MB9BF312N, MB9BF312R, MB9BF314N, MB9BF314R, MB9BF315N, MB9BF315R, MB9BF316N, MB9BF316R
MB9B410R	MB9BF412N, MB9BF412R, MB9BF414N, MB9BF414R, MB9BF415N, MB9BF415R, MB9BF416N, MB9BF416R
MB9B510R	MB9BF512N, MB9BF512R, MB9BF514N, MB9BF514R, MB9BF515N, MB9BF515R, MB9BF516N, MB9BF516R

(TYPE5)

Series	Product Number (not included Package suffix)
MB9A110K	MB9AF111K, MB9AF112K
MB9A310K	MB9AF311K, MB9AF312K

(TYPE6)

Series	Product Number (not included Package suffix)
MB9A140N	MB9AF141L, MB9AF141M, MB9AF141N, MB9AF142L, MB9AF142M, MB9AF142N, MB9AF144L, MB9AF144M, MB9AF144N
MB9A340N	MB9AF341L, MB9AF341M, MB9AF341N, MB9AF342L, MB9AF342M, MB9AF342N, MB9AF344L, MB9AF344M, MB9AF344N
MB9AA40N	MB9AFA41L, MB9AFA41M, MB9AFA41N, MB9AFA42L, MB9AFA42M, MB9AFA42N, MB9AFA44L, MB9AFA44M, MB9AFA44N
MB9AB40N	MB9AFB41L, MB9AFB41M, MB9AFB41N, MB9AFB42L, MB9AFB42M, MB9AFB42N, MB9AFB44L, MB9AFB44M, MB9AFB44N

Document History

Document Title: AN204379 - FM3, Interrupt Factor Vector Relocate Function Usage

Document Number:002-04379

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NNAK	08/30/2012	First Edition.
*A	5640803	NNAK	02/23/2017	Converted Fujitsu Application Note “AN706-00049-1v0-E” to Cypress format

Sales, Solutions, and Legal Information

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | WICED IoT Forums | Projects | Video | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2012-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.