The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN204381

FM3 Microcontroller Deep Standby Mode Setting/Usage

Associated Part Family: Refer to Section 2

This application note is for users, who are considering using the deep standby mode in the Cypress microcontroller FM3 family.

Contents

1 Introduction....1
2 Target products....1
3 Deep Standby Mode....2
4 Usage Example of the Deep Standby Mode....4
5 Confirmed operation environment....5
6 Circuit Explanation....5
7 Sample Program Explanation....7
8 Cautions for Evaluating the Deep Standby Mode....11
9 Reference Documents....12
Document History....13

1 Introduction

This application note is for users, who are considering using the deep standby mode in the Cypress microcontroller FM3 family.

2 Target products

This application note is described about below products;

(TYPE3)

Series	Product Number (not included Package suffix)
MB9A130L	MB9AF132L,MB9AF131L,MB9AF132K,MB9AF131K
MB9A130LA	MB9AF132LA,MB9AF131LA,MB9AF132KA,MB9AF131KA

(TYPE6)

Series	Product Number (not included Package suffix)
MB9AB40N	MB9AFB41L,MB9AFB42L,MB9AFB44L,MB9AFB41M,MB9AFB42M, MB9AFB44M,MB9AFB41N,MB9AFB42N,MB9AFB44N
MB9AB40NA	MB9AFB41LA,MB9AFB42LA,MB9AFB44LA,MB9AFB41MA,MB9AFB42MA, MB9AFB44MA,MB9AFB41NA,MB9AFB42NA,MB9AFB44NA
MB9AA40N	MB9AFA41L,MB9AFA42L,MB9AFA44L,MB9AFA41M,MB9AFA42M, MB9AFA44M,MB9AFA41N,MB9AFA42N,MB9AFA44N
MB9AA40NA	MB9AFA41LA,MB9AFA42LA,MB9AFA44LA,MB9AFA41MA,MB9AFA42MA, MB9AFA44MA,MB9AFA41NA,MB9AFA42NA,MB9AFA44NA
MB9A340N	MB9AF341L,MB9AF342L,MB9AF344L,MB9AF341M,MB9AF342M, MB9AF344M,MB9AF341N,MB9AF342N,MB9AF344N
MB9A340NA	MB9AF341LA,MB9AF342LA,MB9AF344LA,MB9AF341MA,MB9AF342MA, MB9AF344MA,MB9AF341NA,MB9AF342NA,MB9AF344NA
MB9A140N	MB9AF141L,MB9AF142L,MB9AF144L,MB9AF141M,MB9AF142M,

Series	Product Number (not included Package suffix)
	MB9AF144M,MB9AF141N,MB9AF142N,MB9AF144N
MB9A140NA	MB9AF141LA,MB9AF142LA,MB9AF144LA,MB9AF141MA,MB9AF142MA, MB9AF144MA,MB9AF141NA,MB9AF142NA,MB9AF144NA

(TYPE7)

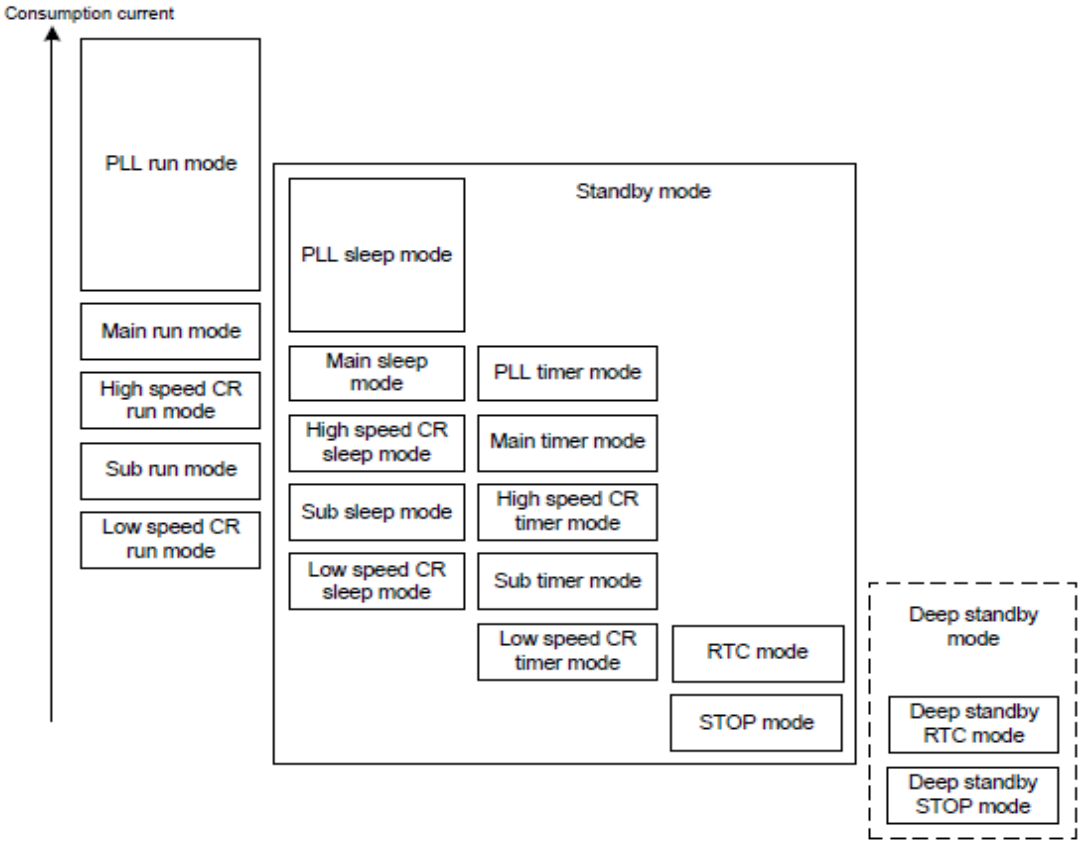
Series	Product Number (not included Package suffix)
MB9AA30N	MB9AFA32L,MB9AFA31L,MB9AFA32M,MB9AFA31M, MB9AFA32N,MB9AFA31N
MB9A130N	MB9AF132M,MB9AF131M,MB9AF132N,MB9AF131N

3 Deep Standby Mode

FM3 has several CPU operation modes.

The relationship between FM3 CPU operation mode and consumption current is indicated in Figure 1

Figure 1. FM3 CPU operation mode



As indicated in Figure 1, the deep standby mode is positioned as a mode with the lowest consumption current.

This application note mainly describes the deep standby RTC mode.

The deep standby mode has 2 modes as indicated in Table 1.

Table 1. Overview of Deep Standby Mode

Mode Name	Overview
Deep standby RTC mode	Deep standby RTC mode is classified as one of the deep standby modes. Deep standby RTC mode stops oscillation other than that of the sub oscillator. All the functions except for the RTC and low voltage detection circuit will be stopped. It turns off following function, - CPU other than RTC and low voltage detect circuit and GPIO - on-chip flash memory - on-chip SRAM - peripheral functions inside the chip.
Deep standby stop mode	Deep standby stop mode is classified as one of the deep standby modes. Deep standby stop mode stops all oscillations. All the functions except for the low voltage detection circuit will be stopped. It turns off following function, - CPU other than RTC and low voltage detect circuit and GPIO - on-chip flash memory - on-chip SRAM - peripheral functions inside the chip.

Some product TYPE can retain the data in on-chip SRAM even in the deep standby modes. In the setting to retain on-chip SRAM data, the power of on-chip SRAM is ON.

For the corresponding product TYPE, see the “FM3 family peripheral manual.”

4 Usage Example of the Deep Standby Mode

The following section explains usage examples of the deep standby mode.

Transition to the deep standby mode is executed by the software control. There are several return sources from the deep standby mode. Those sources are indicated in Table 2 together with the usage examples.

Table 2. Return Sources and Usage Example of the Deep Standby Mode

No.	Return Source	Return Trigger	Example of Connecting Devices	Example of Applications
1	Low-voltage detection reset	Detection of voltage drop of the power supplied to FM3	Batteries, etc.	Battery-powered devices
2	INITX pin input reset	Pulse signals such as push switches	Push switches, etc.	Devices in general
3	HDMI-CEC/Remote control reception interrupt (*1)	Pulse signal from the remote control	HDMI controllers, etc.	Consumer electronics, AV devices in general
4	WKUP pin input	Detection of approximation of a person and/or an object	Proximity sensor	Parking lot vacancy indication system, restroom vacancy indication system, intrusion detection, vending machine, etc.
		Detection of lighting in the surrounding area	Illuminance sensor	Meeting room vacancy indication system, safe footlight, etc.
		Detection of a touch by a finger	Touch sensor	Remote control device, handy terminal, etc.
		Pushing a button	Push switch	Weight scale, blood-pressure meter, low-frequency therapy equipment, digital camera, etc.
5	Low-voltage detection interrupt	Detection of voltage drop of the power supplied to FM3	Batteries, etc.	Battery-powered devices
6	RTC interrupt (*2)	Timer, time, date setting	Integration to FM3	Rice steamer, vending machine, HDD recorder, etc.

1. HDMI-CEC/Remote control reception interrupt is not compatible with TYPE3 products.
2. The RTC interrupt is compatible only with the deep standby RTC mode

5 Confirmed operation environment

The confirmed operation environment for the contents described in this application note is indicated in Table 3

Table 3. Confirmed Operation Environment

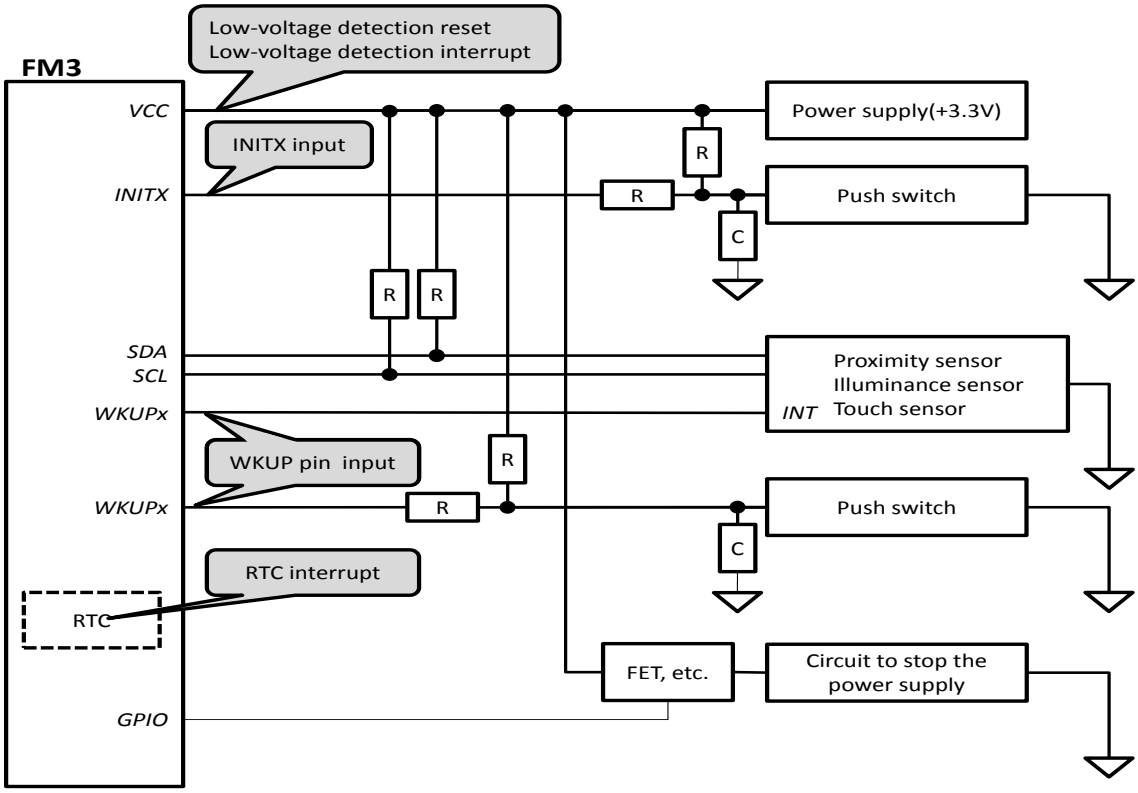
No.	Item	Content
1	Using microcontroller	MB9AF132L
2	Operating frequency	Core : 20 MHz Peripheral : 10 MHz
3	Operating voltage	+3.3V
4	OS	Not used
5	Integrated Development Environment	[IAR] IAR Embedded Workbench for ARM Ver.6.30.4
		[KEIL] MDK-Lite Version 4.22a
6	Compile optimization	None

6 Circuit Explanation

The circuit, which operates the sample software described in this application note, is indicated in Figure 2.

Balloons in **Error! Reference source not found.** indicate the return sources of the deep standby mode.

Figure 2. Outline diagram of the circuit



- **Low-voltage dtection reset and low-voltage detection interrupt**

The low-voltage detection reset and the low-voltage detection interrupt are executed by detecting the voltage level applied to the VCC pins. It is used for recovery from the deep standby mode that occurs due to the voltage drop, such as when the power is supplied from the battery to the VCC pin.

- **INITX input**

When L level is input to the INITX pin, recovery from the deep standby mode is executed.

It is used for recovery from the deep standby mode by connecting the push switch and pressing the switch. When L level is input to the INITX pin, INITX pin input reset is also executed.

- **WKUP pin input**

When H level or L level is input to the WKUP pin, recovery from the deep standby mode is executed. The setting of the input level can be executed in the WKUP pin input level register (WILVR). However, WKUP0 always recovers by the L level input.

For all circuits described in this application note, L level input is used.

Connection of a device such as a sensor or switch to the WKUP pin can be used as the recovery source from the deep standby mode.

The number of WKUP pins varies depending on the product. See the datasheet of each product.

- **RTC interrupt**

RTC interrupt executes recovery from the deep standby mode. RTC interrupt can be executed by the alarm setting or the timer setting. For details of interrupt, which can be output by RTC, see the “FM3 family peripheral manual.”

In the sample program described in this application notification, the interrupt by the RTC timer is introduced.

- **Power supply stop to partial circuits**

In the transition to the deep standby mode, the consumption current for the entire system can be reduced by stopping the power supply to circuits not applicable to the FM3 recovery and to circuits that are not used during the deep standby mode, by using FET or other methods.

To enable this process, GPIO can be used.

By setting the standby pin level setting bit of the standby mode control register (STB_CTL), GPIO can retain the contents, which are set before transition to the deep standby mode. One example is the circuit shown in Figure 2, which controls ON/OFF of FET by connecting GPIO to the Pch-FET gate. In standard operations, FET is turned ON by using GPIO as an L output. Before the transition to the deep standby mode, if GPIO is set to the H output and the FET is turned OFF, the circuit in a latter part of FET can retain the power supply stop status.

Table 4. FM3 and Connection Parts (Operation Check Completed)

No.	FM3 Pin (MB9AF132L)		Connection Parts			
	No.	Function Name	Item Name	Model	Maker	Pin
1	1,18, 33,41	VCC, AVCC	Power supply IC	MB39C022G	Cypress Inc.	
2	21	INITX	Push switch	SKRPADE010	ALPS ELECTRIC Co., Ltd.	—
3	60	WKUP3	Illuminance / proximity sensor	BH1772GLC	ROHM Co., Ltd.	INT
4	57	WKUP0	Touch sensor	LDS6204NTGI	IDT	$\overline{\text{INTB}}$
5	35	WKUP1	Push switch	SKRPABE010	ALPS ELECTRIC Co., Ltd.	—

7 Sample Program Explanation

This section explains the sample program, which transitions to the deep standby RTC mode.

The sample program outlined below is the program that transitions to the deep standby RTC mode. Settings for the transition to the deep standby RTC mode and cautions necessary to keep the consumption current low are explained together in the following section.

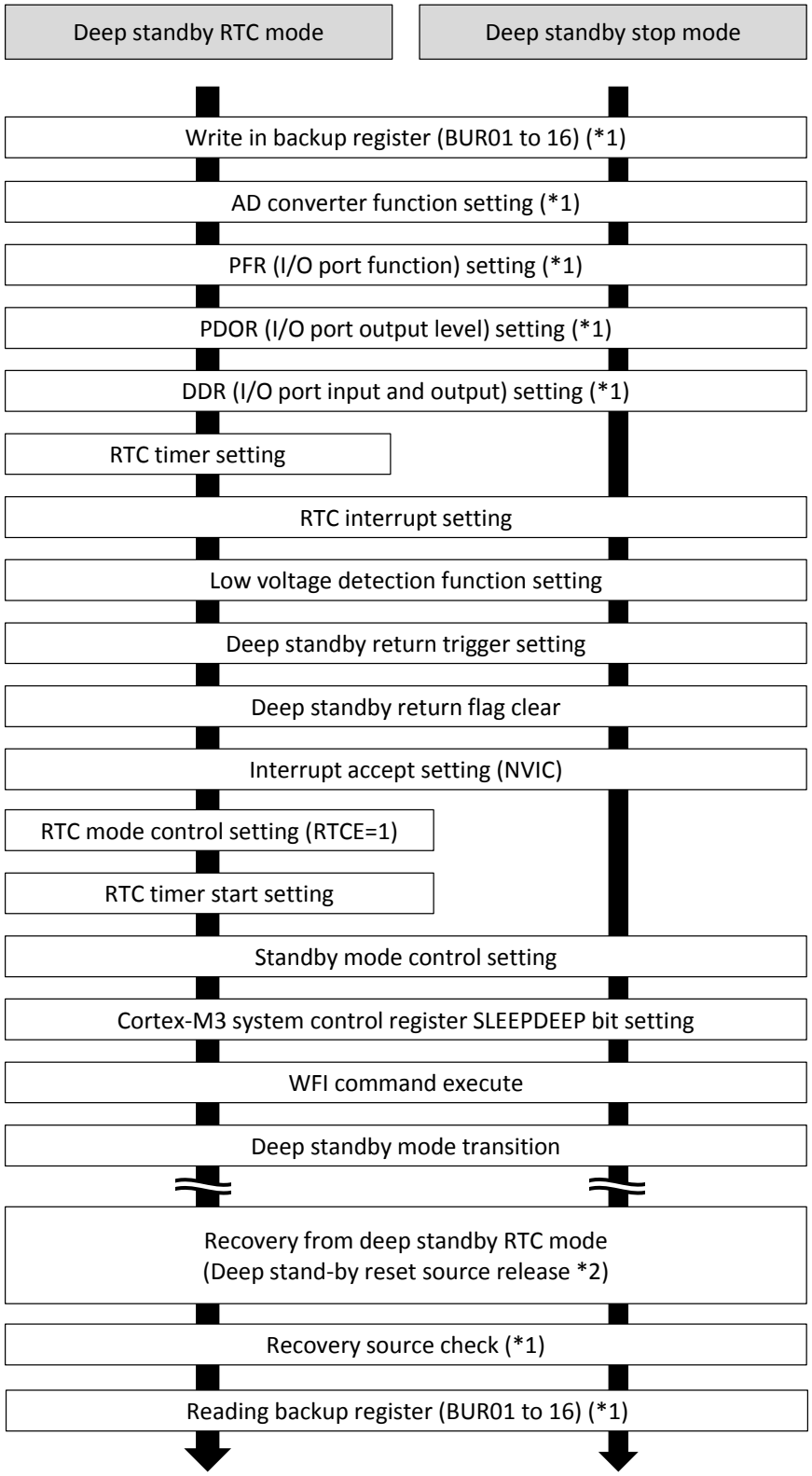
- **Key points for reducing the consumption current**

1. To set the I/O port input/output and output according to a component connected to FM3, determine the output level.
   To set the input, handle the pin such as executing a pull-up, in order to prevent an unstable input voltage level due to the noise.
2. The initial value of JTAG-related pins is set to the JTAG function. When transitioning to the deep standby mode, set the I/O port according to the component connected to FM3.
3. The initial value of the AD converter-related pin is set to the AD converter function. When transitioning to the deep standby mode, set the I/O port according to the component connected to FM3.
4. The low-voltage detection circuit integrated in FM3 operates also during the deep standby mode, depending on the setting. When the voltage detection circuit is not used in the FM3 mounted system, set the low-voltage detection circuit to non-active. By doing so, the consumption current can be kept low.
5. The GPIO setting, which is set when transitioning to the deep standby mode, maintains its status even during the deep standby mode. Therefore, if the power supply is stopped to the partial circuit by FET as described in the “Circuit Explanation” section, set GPIO to control and stop the power during the deep standby mode.

- **Using the backup register**

The backup register is a general-purpose register that maintains a value during the deep standby mode. As an example of use, the following section explains the case in which a FM3-mounted system has several modes as applications.

1. Decide a register that stores the value, which indicates each mode of the system.
2. Write the value, which indicates the current mode, in this backup register.
3. Transition to the deep standby mode.
4. After recovery from the deep standby mode, read the value, which indicates the mode written in the backup register.
5. Transition to the mode indicated by value of back up registor and start a normal mode operation



1. Execute this whenever necessary.
2. For releasing the deep standby reset source, see the “FM3 family peripheral manual.”

Sample program

```
/* Preconditions
Transition from the sub-run mode, SPL=0, RTC initialization completed, and RTC timer interrupt used (5 sec)
*/
/* Defined as the wake-up flag clear*/
uint8_t u8wkup_clr;
/* AD converter common pin setting*/
FM3_GPIO->ADE = 0x00000000;
/* Main clock pin is set to GPIO */
FM3_GPIO->SPSR = 0x00000001;
/* I/O port pin PFR (port function) setting*/
FM3_GPIO->PFR0 = 0x00000000;
FM3_GPIO->PFR1 = 0x00000000;
FM3_GPIO->PFR2 = 0x00000000;
FM3_GPIO->PFR3 = 0x00000000;
FM3_GPIO->PFR4 = 0x00000000;
FM3_GPIO->PFR5 = 0x00000000;
FM3_GPIO->PFR6 = 0x00000000;
FM3_GPIO->PFR8 = 0x00000000;
FM3_GPIO->PFRE = 0x00000000;
/* GPIO PDOR (output level) setting*/
FM3_GPIO->PDOR0 = 0x00000000;
FM3_GPIO->PDOR1 = 0x00000000;
FM3_GPIO->PDOR2 = 0x00000000;
FM3_GPIO->PDOR3 = 0x00000000;
FM3_GPIO->PDOR4 = 0x00000000;
FM3_GPIO->PDOR5 = 0x00000000;
FM3_GPIO->PDOR6 = 0x00000000;
FM3_GPIO->PDOR8 = 0x00000000;
FM3_GPIO->PDORE = 0x00000000;
```

[AD converter common pin setting]

In the AD converter common pin, the AD converter is valid as an initial value. Execute the setting appropriate to the system while each AD converter common pin is in the deep standby mode.

[I/O port pin PFR (port function) setting]

The I/O port pin status in the deep standby mode affects the consumption current.

All I/O port pins are set to GPIO in this example. However, in order to reduce the consumption current of the entire system, not only FM3, consider the input/output status and the voltage level of devices connected to FM3. In addition, it is necessary to transition to the deep standby mode in the optimal condition.

The internal pull-up resistor of the debug pin is set to "connect" in the initial value. As the pull-up resistor affects the consumption current, review thoroughly before executing the setting. In addition, when the corresponding pin is in the output setting, the pull-up resistor is turned off.

[GPIO PDOR (output level) setting]

Depending on the connecting part, there is a case where GPIO is set to input. However, in this sample program, all GPIOs are set to L output. By using the connecting part, execute the optimal input/output setting.

(continue to the next page)

```
/* GPIO DDR (input/output) setting*/
FM3_GPIO->DDR0 = 0x0000FFFF;
FM3_GPIO->DDR1 = 0x0000FFFF;
FM3_GPIO->DDR2 = 0x0000FFFF;
FM3_GPIO->DDR3 = 0x0000FFFF;
FM3_GPIO->DDR4 = 0x0000FFFF;
FM3_GPIO->DDR5 = 0x0000FFFF;
FM3_GPIO->DDR6 = 0x0000FFFF;
FM3_GPIO->DDR8 = 0x0000FFFF;
FM3_GPIO->DDRE = 0x0000FFFF;
/* RTC timer setting */
/* Timer setting */
FM3_RTC->WTTR = 0x09; /* 5sec */
/* Set Timer Counter Control */
bFM3_RTC_WTCR2_TMEN = 0;
/* RTC interrupt setting */
/* Clear Timer interrupt flag */
bFM3_RTC_WTCR1_INTTMI = 0;
/* Enable the timer interrupt */
bFM3_RTC_WTCR1_INTTMIE = 1;
/* Low-voltage detection interrupt setting */
/* Release the protection of low-voltage detection
voltage setting register */
FM3_LVD->LVD_RLR = 0x1ACCE553;
/* Set low-voltage detection voltage setting register */
FM3_LVD->LVD_CTL = 0x00000400;
/* Enable low-voltage detection interrupt */
bFM3_LVD_LVD_CTL_LVDIE = 1;
/* Set low-voltage detection function low power */
bFM3_LVD_LVD_CTL_LVDIM = 1;
/* Set the protection for low-voltage detection
 voltage setting register */
FM3_LVD->LVD_RLR = 0x00000000;
/* deep standby recovery trigger setting */
FM3_DS->WIER = 0x3B;
```

(continue to the next page)

[GPIO DDR (input/output) setting]

GPIO input/output direction is set. It is necessary to set the optimal input/output by the connection device. However, when setting the output, it is recommended to set DDR after PDOR in order to prevent outputting the unintended level.
When setting the input, fix the level by executing the pull-up, and set H or L to the input voltage. Pay close attention to prevent an unstable condition. There is a case where the consumption current becomes unstable.

[RTC timer setting]

Set the RTC timer when the timer counter is at stop.
As an example, the timer interrupt is set only once after 5 seconds from the start of the timer.

[Low-voltage detection interrupt setting]

To set the low-voltage detection function, release the protection of the low-voltage detection voltage setting register and execute the setting. After the setting, validate the protection again.
To keep the consumption current low, execute the low power setting when necessary. When the low power setting is used, the low-voltage detection reset function stops. Review thoroughly before setting.

[Deep standby recovery trigger setting]

It sets the trigger that accepts from WKUP [3:1] pin, LVD interrupt, and RTC interrupt. The WKUP0 pin is always allowed, and prohibition setting is not possible.

```
/* Deep standby recovery flag clear */
u8wkup_clr = FM3_DS->WIFSR;
u8wkup_clr = FM3_DS->WRFSR;
```

[Deep standby recovery flag clear]
It is cleared by reading the deep standby return source registers 1 and 2.

```
/* Interrupt accept setting */
/* Enables NVIC */
NVIC_EnableIRQ(OSC_PLL_RTC_IRQn);
```

[Interrupt accept setting]
For xxx of NVIC_EnableIRQ(xxx), use a name defined in the IO definition file.
For NVIC, see the “Cortex-M3 technical reference manual.”

```
/* RTC mode control setting */
bFM3_DS_PMD_CTL_RTCE = 1;
/* Standby mode control register setting */
FM3_CRG->STB_CTL = 0x1ACC0006;

/* Timer start */
bFM3_RTC_WTCR2_TMST = 1;

/* SLEEPDEEP bit setting Cortex-M3 system control register */
SCB->SCR |= (0x00000004);
/* WFI command execute */
__WFI();
```

8 Cautions for Evaluating the Deep Standby Mode

This section explains the cautions necessary for evaluating the transition and recovery to/from the deep standby mode.

1. JTAG ICE (Integrated Development Environment) connection during the deep standby mode

During the deep standby mode, the power of the debug function turns off. Therefore, the connection of the JTAG ICE enters into the disconnected condition.

Then, if the recovery from the deep standby mode is executed with JTAG ICE being connected, part of the JTAG ICE functions cannot be operated as same status before transition to the deep standby mode.

2. Setting the recovery condition from the deep standby

Before the transition to the deep standby mode, set the deep standby recovery enable register (WIER) and the WKUP pin input level register.

When the set value is not appropriate for the system (circuit), the recovery occurs immediately after transition to the deep standby mode.

For example, if transition to the deep standby mode is executed in the following conditions, recovery from the deep standby mode occurs immediately after the transition.

- Accept a recovery by the WKUP1 pin (Set with the WIER1 register)
- Set the recovery enable level of the WKUP1 pin to L level (Set with the WILVR register)

- The WKUP1 pin is pulled down on the circuit, and in the deep standby mode, the signal line connected to WKUP1 is in L level.

3. Conditions where the ICE connection becomes disabled continuously and the recovery method

In the following cases, connection of JTAG ICE becomes disabled. The recovery method in such case is also explained here.

- Condition where ICE cannot be connected

  For example, when the software used for a transition to the deep standby mode is written in FM3, and if this software transitions to the deep standby mode without any condition and without using a trigger such as a switch, connection to JTAG ICE becomes impossible.

  This happens because the debug function stops by the transition to the deep standby mode, and the power turns off.

  Even if the recovery is executed by using the WKUP pin or by reset operation, transition to the deep standby mode re-occurs unconditionally. Therefore, connection to JTAG ICE becomes impossible.

- Recovery method

  If the connection to JTAG ICE becomes impossible in the above condition, try the following methods.

  - Set the FM3 MD0 pin to the H level in the condition where JTAG ICE is connected and execute the reset operation (by turning the power on or by L input to the INITX pin). Set the MD0 pin to the H level and execute the reset operation. When the FM3 transitions to the serial writer mode, the written software does not start. Therefore, connection of JTAG ICE becomes possible. When the connection is executed, rewrite to the software, which does not transition to the deep standby mode unconditionally.
  - Use the FLASH MCU Programmer for FM3 or the FLASH USB DIRECT Programmer.

    Set the FM3 MD0 pin to the H level and reset (by turning on the power or by L input to the INITX pin). By setting the MD0 pin to the H level and resetting, FM3 transitions to the serial writer mode. Rewrite to the software, which does not transition to the deep standby mode unconditionally, by using the FLASH MCU Programmer for FM3 or the FLASH USB DIRECT Programmer.

    After the rewrite operation, connection to JTAG ICE is possible as transition to the deep standby mode is not executed.

9 Reference Documents

1. .FM3 Family PERIPHERAL MANUAL (MN706-00002-5v0-E)

(Please refer to the latest document.)

Document History

Document Title: AN204381 - FM3 Microcontroller Deep Standby Mode Setting/Usage

Document Number: 002-04381

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	YUIS	07/12/2012	Rev1.0 Initial Release
			01/31/2014	Rev1.1 Company name and layout design change
*A	5032802	YUIS	12/03/2015	Migrate Spansion Application Note from AN706-00050-1v1-E to Cypress Format
*B	5872542	AESATP12	09/06/2017	Updated logo and copyright.

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 | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

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

© 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.