

# Debugging MSP430 LPM4.5

Stefan Schauer

MSP430 Product Applications

## ABSTRACT

The Low Power Mode 4.5 (LPM4.5) is used to achieve lowest power consumption in the MSP430F5xx devices. When the device enters LPM4.5, the internal regulator of the Power Management Module (PMM) is disabled, thus switching off the supply voltage that is used to power the device core. The device core includes the JTAG and debug (EEM) logic of the device. To debug code while the device enters or exits the LPM4.5, special considerations and debug techniques are required. This application report discusses the LPM4.5 debugging procedures to be followed for the debugger to synchronized with the device when entering and exiting LPM4.5.

## Contents

1	Introduction .....	1
2	LPM4.5 Debugging Procedure .....	1
3	Application Demo .....	4
4	References .....	5

## List of Figures

1	Enable 'Release JTAG on Go' in CCS .....	2
2	Enable 'Release JTAG on Go' in IAR .....	2

## List of Tables

## 1 Introduction

LPM4.5 is an additional low-power mode feature that is provided in the MSP430F5xx device families in which the regulator of the PMM is completely disabled, providing additional power savings. Because there is no power supplied to device core during LPM5, the CPU and all digital modules including the RAM, the EEM (debug logic), and the JTAG interface are switched off. This forces a communication halt between the debugger and the device; therefore, the debugger loses synchronization to the device. The debugging procedures that are discussed in the following section are necessary to restore debugger synchronization with the device when the device is in LPM4.5.

## 2 LPM4.5 Debugging Procedure

### 2.1 Entering LPM4.5

For details regarding the program flow to enter LPM4.5, see the *MSP430F5xx Family User's Guide* ([SLAU208](#)), *System Resets, Interrupts, and Operating Modes, System Control Module (SYS)* chapter.

## 2.2 Prepare Debugger Before LPM4.5 Entry

Prepare the debugger before losing communication with the device by the selecting the 'Release JTAG on Go' option.

- If using CCS, this option is found under the Run icon; use the drop-down menu by clicking the arrow next to the Run icon (see [Figure 1](#)) and select the 'Run Free' option.

**NOTE:** To switch back to normal mode, select 'Run' option in the same drop-down menu the next time.

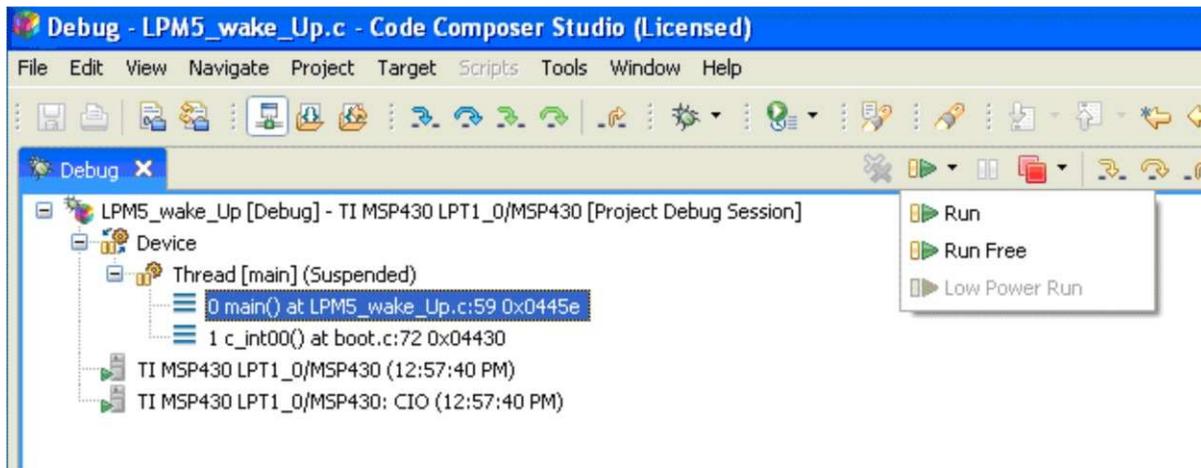


Figure 1. Enable 'Release JTAG on Go' in CCS

- If using IAR, this option is found in the Emulator menu (see [Figure 2](#)).

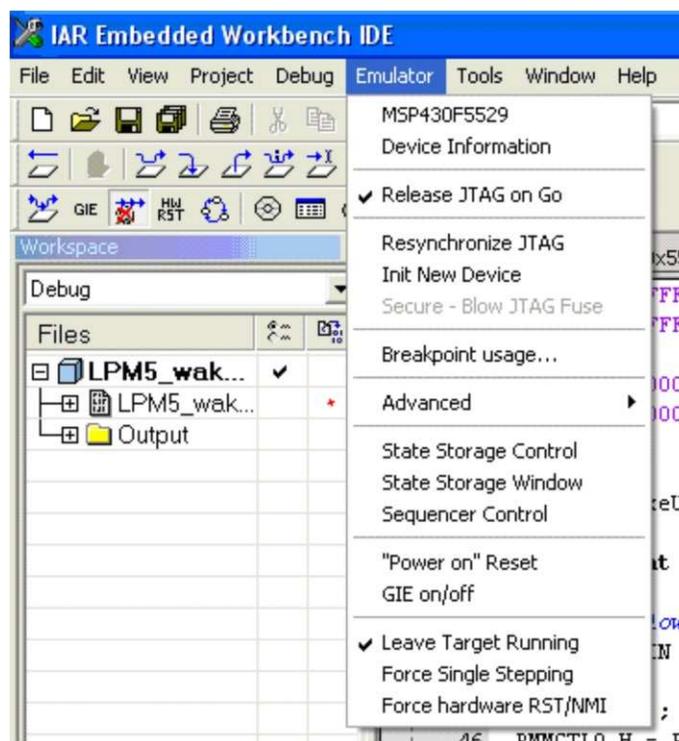


Figure 2. Enable 'Release JTAG on Go' in IAR

Selecting the 'Release JTAG on Go' option releases the driver on the JTAG lines and makes sure the debugger does not lose communication with the device. However, with this option selected, the debugger cannot identify breakpoints.

Once the 'Release JTAG on Go' option is enabled, execute the 'RUN' command.

- If using CCS, click Target → Run.
- If using IAR, click Debug → Go.

---

**NOTE:** During debug process, the user should not enter the LPM4.5 mode by single stepping. This could cause the debugger to lose control and force a reset. Therefore, the final step to enter the LPM4.5 should be the execution of the 'RUN' Command.

---

## 2.3 Wake From LPM4.5

Wake-up from LPM4.5 is possible with an RST event, a power cycle, or any interrupt-capable port pin (this feature is available in new 5xx devices only; verify in the device-specific data sheet). For more details regarding exiting LPM4.5, see the *MSP430F5xx Family User's Guide (SLAU208)*, *System Resets, Interrupts, and Operating Modes, System Control Module (SYS)* chapter.

Exit from LPM4.5 causes a BOR event and immediately follows the normal program execution that is located at the system reset vector location. With the wake-up from LPM4.5, all the prior debug logic configurations are lost. To synchronize with the debugger after wake-up from LPM4.5, a software debug trap needs to be included in the application code. This section of software trap code is required only for debugging purposes and can be removed from the final application code.

```
// LPM4.5 - SW Trap in C
Void main(void)
[
    WDTCTL = WDTPW + WDTHOLD;           // Stop the Watchdog
    If (wake up from LPM4.5)
    [
        stop_LPM4_5_WakeUp =1;         // Set Flag for SW Trap
        while(stop_LPM5_WakeUp);      // SW Trap if wake-up from LPM5
    ]
    ...
]
```

```
; LPM4.5 - SW Trap in ASM
mov.w    #WDTPW+WDTHOLD,&WDTCTL      ; Stop the Watchdog
bit.w    #wake_up_from_LPM4.5,status ; check a for LPM4.5 wake up
jnz     $                               ; SW Trap if wake-up from LPM5
```

Reconnecting the debugger back to the device is done by using the 'HALT' or 'BREAK' button.

- If using CCS, click Target → Halt to halt the debugger.
- If using IAR, click Debug → Break to halt the debugger.

Once the debugger is halted, it synchronizes with the device again. The program counter would be set to the next code statement and the code that follows can be debugged.

The same can be achieved by checking the input state of the interrupt capable port I/O that is used for waking the device up from LPM4.5. In the example below the input PA.7 or P1.7 is used for the wakeup. This would avoid a complete lock up if the debug code has not been removed.

```

while ((PAIN & 0x0080) == 0);      // SW trap for LPM4.5
                                   // -- execution will stop here as long as
                                   //   input is low
                                   // This could be used to synch up the debugger
                                   // and continue to debug from here after LPM4.5

```

```

; SW trap for LPM4.5
lpm5_wait
bit.w  #0x0080,&PAIN                ; -- execution will stop here as long as
jz     lpm5_wait                    ;   input is low
                                   ; This could be used to synch up the debugger
                                   ; and continue to debug from here after LPM4.5

```

### 3 Application Demo

This application report also includes a demo with source code that shows how to enter and exit LPM4.5 using the debugging techniques discussed before.

In the demo code, the register SYSRSTIV is checked to identify the "wake-up from LPM4.5" startup condition. The Reset Source Stack in the SYSRSTIV register is evaluated until all Reset events are cleared. If the wake-up from LPM4.5 event is detected, the function 'LPM4\_5\_WakeUP' is called and a flag is set. This function also initializes the port pins.

Entering or exiting the LPM4.5 in the application demo is externally controlled by the following:

- Falling edge on P1.6 to enter LPM4.5
- Falling edge on P1.7 to exit LPM4.5

P1.0 indicates when the device enters/exits LPM4.5:

- When the device enters LPM4.5, P1.0 is set (LED at P1.0 is ON)
- When the device exits LPM4.5, P1.0 is reset (LED at P1.0 is OFF)

PortB is used to indicate the following application states:

- 0x01: BOR event detected
- 0x02: Toggle in active mode
- 0x04: Set before entering LPM4.5
- 0x08: Wakeup form LPM4.5 event detected
- 0x10: Set after init from LPM4.5 has been done

After an LPM4.5 wakeup event, the ports must be re-initialized and then the LOCKIO bit must be cleared to activate the control for the port pins again. This is done in the LPM4\_5\_WakeUP function.

```

PM5CTL0 &= ~LOCKIO;                // Clear LOCKIO and enable ports

```

---

**NOTE:** If there are set Interrupt Flags from the Port Logic, they needs to be cleared after the LOCKIO bits have been cleared. Access to these port interrupt registers is not available before the LOCKIO bit is cleared.

---

### 3.1 Steps to Run the LPM4.5 Debugger Demo

1. Download demo code provided along with the application report. See [Section 2.2](#) to start the debug process.
2. On running the code execution, the device initializes; PBOUT = 0x0001 (indicates BOR event detected). Device enters LPM4 and waits for external input on P1.6.
3. To enter LPM4.5, press button S1 (falling edge on P1.6).
4. Device enters LPM4.5; P1.0 is set (LED at P1.0 is ON); PBOUT = 0x0004 (indicates device in LPM4.5).
5. Device stays in LPM4.5 with all port pins kept in the state it was last initialized.
6. To exit LPM4.5, press button S2 (falling edge on P1.7).
7. Device will exit LPM4.5; P1.0 is reset (LED at P1.0 is OFF); PBOUT = 0x0008 (indicates wakeup from LPM4.5 event detected).
8. Follow steps in [Section 2.3](#) to regain device-debugger synchronization.
9. On halting the debugger (as discussed in [Section 2.3](#)), device will be in active mode waiting for falling edge on P1.6 to re-enter LPM4.5; PBOUT = 0x0002 toggles (indicates device in active mode).

## 4 References

1. *MSP430x5xx Family User's Guide* ([SLAU208](#))
2. *IAR Embedded Workbench Version 3+ for MSP430 User's Guide* ([SLAU138](#))
3. *Code Composer Studio v4 for MSP430 User's Guide* ([SLAU157](#))

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (<https://www.ti.com/legal/termsofsale.html>) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2021, Texas Instruments Incorporated