Application Report
SLAA395–June 2008

MSP430x5xx Quick Start Guide

Mark Smertneck *MSP430 Applications*

ABSTRACT

This quick start guide is intended to assist existing users of the MSP430 line of products in converting existing code to the new MSP430x5xx modules. While the modules operate in accordance with previous families, the MSP430x5xx modules include a number of enhancements which allow for more effective coding, expansions on current functionality and greater flexibility in the module and system operations.

Accordingly an overview of the MSP430x5xx family should be referenced in order to familiarize yourself with the new functionality in the MSP430x5xx family. This is intended to be a nuts and bolts guide. For a complete description of modules please refer to the appropriate user guide or data sheet section.

Contents

List of Tables

1 Port Pins

When starting the device the default setting for the port pins associated with the crystals is port functionality. Accordingly if using crystals in the application one of the first things which needs to be done programmatically is to switch the port pins to the crystal functionality. Additionally the recommendation for unused pins is to set them as outputs and not connect them on the PC board. Alternatively they may be set as inputs and enabling either the pullup or pulldown resistor. The most important thing is not to leave them in an input condition where the input is floating. This can lead to stray switching currents which affect low power operation.

To support the pullup/pulldown resistors in the MSP430x5xx and the drive strength functionality new registers have been added:

- Port drive strength PxDS
  Bit = 0: Reduced drive strength
  Bit = 1: Full drive strength
- Port resistor enable PxREN
  When a port pin is configured as an input, there is an integrated pullup/pulldown resistor, used in conjunction with PxOUTx to determine if pullup or pulldown functionality is to be used (see Table 1).

Table 1. PxREN Pullup/Pulldown Resistor Configuration

PxDIRx	PxRENx	PxOUTx	I/O Configuration
0	0	x	Input
0	1	0	Input with pulldown resistor
0	1	1	Input with pullup resistor
1	x	x	Output

Port pin interrupts are available on P1 and P2. These are automatically disabled when PxSEL = 1 (see Table 2).

Table 2. Interrupt Edge Select Configuration

PxIESx	PxINx	PxIFGx
0 • 1	0	Unchanged
0 • 1	1	May be set
1 • 0	0	May be set
1 • 0	1	Unchanged

Port P1 now has an interrupt vector, and port pin interrupts are prioritized accordingly, with P1IFG.0 having the highest priority. The port pin interrupts combine to source a single interrupt vector. When accessing the P1IV, the highest priority interrupt is reset. A P1 interrupt handler code example follows:

```
;Interrupt handler for P1IFG                                Cycles
P1_HND                      ; Interrupt latency              6
         ADD     &P1IV,PC   ; Add offset to jump table       3
         RETI               ; Vector 0: No interrupt         5
         JMP     P1_0_HND   ; Vector 2: Port 1 bit 0         2
         JMP     P1_1_HND   ; Vector 4: Port 1 bit 1         2
         JMP     P1_2_HND   ; Vector 6: Port 1 bit 2         2
         JMP     P1_3_HND   ; Vector 8: Port 1 bit 3         2
         JMP     P1_4_HND   ; Vector 10: Port 1 bit 4        2
         JMP     P1_5_HND   ; Vector 12: Port 1 bit 5        2
         JMP     P1_6_HND   ; Vector 14: Port 1 bit 6        2
P1_7_HND                    ; Vector 16: Port 1 bit 7
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_0_HND                    ; Vector 2: Port 1 bit 0
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_1_HND                    ; Vector 4: Port 1 bit 1
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_2_HND                    ; Vector 6: Port 1 bit 2
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_3_HND                    ; Vector 8: Port 1 bit 3
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_4_HND                    ; Vector 10: Port 1 bit 4
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_5_HND                    ; Vector 12: Port 1 bit 5
         ...                ; Task starts here
         RETI               ; Return to main program         5
P1_6_HND                    ; Vector 14: Port 1 bit 6
         ...                ; Task starts here
         RETI               ; Return to main program         5
         RETI               ; Return to main program         5
```

The benefit of using interrupt vectors are linear execution time and reduction in code, both execution and memory usage.

Another important note is that the PxSEL registers are accessible with word format when using the PASEL, PBSEL, PCSEL and so on. Setting PxSEL = 1 does not automatically set the pin direction, other modules may require the PxDIRx bits to be set according the peripheral module function. See the pin schematics in the device-specific data sheet for proper setting.

2 Unified Clock System (UCS)

The UCS has all the previous functionality of the MSP430 families clock modules including the very-low-power low-frequency oscillator (VLO) of the MSP430x2xx family and the frequency locked loop (FLL) of the MSP430x4xx family. It is, however, more flexible and has a greater range of configuration options, as well as new clock sources such as the REFO, a trimmed internal reference oscillator, and the MODOSC, which provides a no-configuration clock source for the flash control module. With respect to the register set, the register names have been unified under the register names UCSCTL0 through UCSCTL8. You will find the controls for the clock system you are familiar with in addition to the new controls (see Table 3).

Table 3. UCS Register Map

Register	Function
UCSCTL0	DCO tap selection and MOD bits
UCSCTL1	DCORSEL DCO range select and DISMOD disable modulation
UCSCTL2	FLL loop divider and FLL multiplier bits
UCSCTL3	SELREF bits for FLL and FLLREFDIV bits
UCSCTL4	Clock source selection for ACLK, SMCLK, and MCLK
UCSCTL5	Clock source divider for ACLK, SMCLK, and MCLK
UCSCTL6	Crystal control bits for XT2 and XT1 (drive, bypass, off, cap, mode) SMCLK off bit
UCSCTL7	FLL Interrupt enable and flags, XT2, XT1 and DCO fault flags
UCSCTL8	Fail safe controls for ACLK, SMCLK, CLK request enables for peripheral modules to use ACLK, SMCLK, and MCLK when in low-power modes

Code examples comparing MSP430x2xx and MSP430x4xx to MSP430x5xx clocks

MSP430x2xx code example

```
mov.b &CALBC1_1MHZ, &BCSCTL1    ;set range
mov.b &CALBC1_1MHZ, &DCOCTL     ;set DCO step = modulation
```

MSP430x4xx code example

```
mov.b #(32-1), &SCFQCTL         ;set MCLK = 32 * ACLK, DCOPLUS = 0
mov.b #FN_2, &SCFIO             ;select DCO Range
```

MSP430x5xx code example equivalent

```
mov.w   #SELREF1, &UCSCTL3      ; Set FLLREF to REFO
mov.w   #FLLN5, &UCSCTL2        ; FLL multiplier N = 32
```

As you can see the MSP430x5xx family is closest to the MSP430x4xx family in basic operation due to the use of the FLL. There are a number of enhancements to the clock system beyond the scope of this document, for more information consult the *MSP430x5xx Family User's Guide* Universal Clock System chapter.

Note: Before changing MCLK to higher frequencies, you must modify the VCORE to the appropriate level. In the PMM section of this guide, there is a code example showing how to raise it to the highest level, which allows the MSP430x5xx to operate at maximum MCLK frequency. Intermediate levels are suggested for lower ranges and operate at lower current consumption. It is recommended that you review the PMM section in detail to utilize the appropriate V_{CORE} to optimize the active current consumption in each application.

3 Memory Map

The MSP430x5xx family has a 1-MB unified memory map with expanded peripheral space over previous families (see Table 4).

- Peripheral space is eight times the size available in the MSP430x2xx/MSP430x4xx families.
- Info A is two times the size available in the MSP430x2xx/MSP430x4xx families.
- Interrupt vector space is two times the size available in the MSP430x2xx/MSP430x4xx families.

Table 4. Example Memory Map (MSP430F5438)

Segment		Address
Memory	Size	256KB
Main: Interrupt vector	Flash	00FFFFh to 00FF80h
Main: Code memory	Flash	045BFFh to 005C00h
RAM	Size	16KB
		005BFFh to 001C00h
Information Memory	Size	512B
	Flash	0019FFh to 001800h
Boot Memory	Size	2KB
	Flash	0017FFh to 001000h
Peripherals	Size	4KB
	Flash	000FFFh to 000000h

4 Power Management Module (PMM)

The PMM is a new module to the MSP430 families. The main purpose of the PMM is the LDO which reduces the core CPU voltage. The PMM contains a programmable LDO which supplies the CPU and a limited section of other peripherals. At the default minimum setting of 1.35 V, the CPU can operate up to 12 MHz. To operate at maximum CPU frequency, the V_{CORE} needs to be set to maximum. The details of the PMM can be found in the *MSP430x5xx Family User's Guide* (SLAU208). To increase the V_{CORE} to maximum, use the following code:

```
PMMCTL0_H = 0xA5;                       // Open PMM module registers for write access
PMMCTL0 = 0xA500 + PMMCOREV_2;          // Set VCore to 1.75 V
SVSMLCTL = SVMLE + SVSMLRRL_6;          // Set SVM new Level
while ((PMMIFG & SVSMLDLYIFG) == 0);    // Wait till SVM is settled (Delay)
PMMIFG &= ~(SVMLVLRIFG + SVMLIFG);      // Clear already set flags
if ((PMMIFG & SVMLIFG))
  while ((PMMIFG & SVMLVLRIFG) == 0);   // Wait till level is reached
SVSMLCTL &= ~SVMLE;                     // Disable Low side SVM
PMMCTL0_H = 0x00;                       // Lock PMM module registers for write access
```

The PMM includes two monitoring elements, Supply Voltage Supervisor (SVS) and Supply Voltage Monitor (SVM). In the previous code example, they are set in a sequence to prevent the SVS and SVM from providing an interrupt as the V_{CORE} is changed. It is important that the sequence be followed to prevent unintended resets. Additional information regarding the SVS and SVM is available in the *MSP430x5xx Family User's Guide* (SLAU208) in the PMM chapter.

5 Timer A

Timer A is essentially the same as previous families. There are two differences between Timer A in the MSP430x5xx family and in the previous MSP430 families.

The first is the addition of a second clock divider, which allows the chosen clock source to be further divided down from the existing IDx setting. The additional divider is set up in a new register (TAEX0), where the bits IDEX(2:0) set the second divide down.

The other difference is that the overflow vector has been moved to 0Eh, as opposed to previous families, which had the vector at 0Ah.

Table 5. TAIV

TAIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	–	Highest
02h	Capture/compare 1	TACCR1 CCIFG	
04h	Capture/compare 2	TACCR2 CCIFG	
06h	Capture/compare 3	TACCR3 CCIFG	
08h	Capture/compare 4	TACCR4 CCIFG	
0Ah	Reserved	–	
0Ch	Reserved	–	
0Eh	Timer overflow	TAIFG	Lowest

6 Timer B

Timer B is essentially the same as previous families. There are two differences between Timer B in MSP430x5xx family and in the previous MSP430 families.

The first is the addition of a second clock divider, which allows the chosen clock source to be further divided down from the existing IDx setting. The additional divider is setup in a new register (TBEX0), where the bits IDEX(2:0) set the second divide down.

The other difference is that the overflow vector has been moved to 0Eh, as opposed to previous families, which had the vectors at 0Ah.

Table 6. TBIV

TBIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	–	Highest
02h	Capture/compare 1	TBCCR1 CCIFG	
04h	Capture/compare 2	TBCCR2 CCIFG	
06h	Capture/compare 3	TBCCR3 CCIFG	
08h	Capture/compare 4	TBCCR4 CCIFG	
0Ah	Capture/compare 5	TBCCR5 CCIFG	
0Ch	Capture/compare 6	TBCCR6 CCIFG	
0Eh	Timer overflow	TBIFG	Lowest

7 ADC12

The differences between the ADC12 in MSP430x5xx devices and the ADC12 in MSP430x4xx devices are as follows:

- ADCCTL2 register
  - Additional divide-down stage for the ADC12CLK that is controlled by the ADC12PDIV bit
  - ADC12 resolution may be set to 8-bit, 10-bit, or 12-bit resolution using the ADC12RES bits.
  - ADC12 results may be stored in either unsigned binary or 2s complement format using the ADC12DF bit.
  - ADC12 sampling rate may be set to approximately 200 ksps or 50 ksps using the ADC12SR bit. The lower sampling rate is a lower-power version.
  - ADC12 reference may be output using the ADCREFOUT bit.
  - ADC12 reference buffer can be set to remain on continuously or to turn on only during sample and convert operations using the ADC12REFBURST bit.
- When in 2s complement format, the ADC12MEM conversion registers hold the conversion result in the upper 12 bits in a left justified format. In the normal binary mode, the format is as in previous ADC12 families.
- The standard definition names for some of the bits have been changed to reflect their source; e.g., MSC is now ADC12MSC and REF2_5 is now ADC12REF2_5

The differences between the ADC12 in MSP430x5xx devices and the ADC10 in the MSP430x2xx devices are as follows:

- No ADC-based DTC mechanism. Instead of the DTC, the MSP430x5xx family uses DMA to facilitate transfers from ADC12MEM.
- ADCCTL2 register
  - Additional divide-down stage for the ADC12CLK that is controlled by the ADC12PDIV bit.
  - ADC12 resolution may be set to 8-bit, 10-bit, or 12-bit resolution using the ADC12RES bits.
  - ADC12 results may be stored in either unsigned binary or 2s complement format using the ADC12DF bit.
  - ADC12 reference may be output using the ADCREFOUT bit.

8 Flash

The MSP430x5xx family flash can operate on DV_{CC} supplied over the complete device range of 1.8 V to 3.6 V. Additionally, the MSP430x5xx flash has the following differences:

- Flash controller uses MODOSC for write and erase. MODOSC is a dedicated module oscillator that requires no setup and is provided automatically.
- Marginal read mode operation is supported.
- Banks can be erased while program execution is occurring in a different flash bank.
- Long word and long word block write is supported.

Specific differences between the MSP430x5xx and MSP430x2xx flash are:

- FCTL1
  - BLKWRT and WRT now operate in conjunction to define three modes – byte/word write, long word write, and long word block write.
  - Smart write allows for faster program time using the SWRT bit.
  - Emergency interrupt exit allows an emergency interrupt to stop a flash operation using the EEIEX bit.
  - Segment erase may be interrupted during an interrupt request using the EEI bit. Erase continues after the interrupt is serviced.
  - MERAS and GMERAS no longer allow the erasure of the protected memory segment. Only segment erase, mass erase of a single bank, and erase all except protected memory is supported.
- FCTL2 has been eliminated as it was the clock control for the flash, which is now implemented using the MODOSC and requires no setup.
- FCTL3
  - There is a new bit (FAIL) to indicate that the flash operation failed.
  - The information A segment can be specifically locked using the LOCKA bit.
- FCTL4 is a new register that contains the controls for the marginal read modes, lock for information memory, and VPE error bit.
  - LOCKINFO, when set, locks the information memory and does not allow it to be erased in segment mode or written.
  - Marginal read mode is controlled with MRG1 and MRG0, which enable marginal read mode 1 or 0, respectively. See the *MSP430x5xx Family User's Guide* (SLAU208) for additional information on using marginal modes.
  - Voltage changed during program error bit is set by hardware if DV_{CC} changes significantly during programming and indicates that the programming may be corrupt.

9 DMA

The DMA on MSP430x5xx devices now supports up to eight channels of DMA. The associated changes from the MSP430x4xx family of devices are:

- DMACTL0
  DMA trigger select up to 32 different triggers, device dependant. Accordingly, DMA1TSELx and DMA0TSELx are now five bits and occupy the lower five bits of each byte.
- DMACTL1
  Contains the trigger selects for DMA2 and DMA3
- DMACTL2
  Contains the trigger selects for DMA4 and DMA5 (if available in device)
- DMACTL3
  Contains the trigger selects for DMA6 and DMA7 (if available in device)
- DMACTL4
  DMAONFETCH no longer exists but is replaced by DMARMWDIS.
- DMAxCTL, DMA Channel X control is unchanged.
- DMAxSA, DMA Source Address Register is unchanged.
- DMAxDA, DMA Destination Address Register is unchanged.
- DMAxSZ, DMA Size Address Register is unchanged.
- DMAIV now supports up to eight channels of DMA (if available on the device).

10 Real-Time Clock and Basic Timer

The RTC_A module has a number of differences from the RTC module on the MSP430x4xx family of device. Essentially, it encompasses the Basic Timer on MSP430x4xx devices and adds an alarm mode functionality that requires no CPU intervention and can generate an interrupt. Also, there is calibration logic available to correct for time offset in RTC mode. Specifically, the differences and the associated registers are:

- New register RTCCTL0
  - Real-time clock time event interrupt enable controlled by the RTCTEVIE bit. This allows an interrupt to be generated every minute, hour, every day at noon, or every day at midnight, selectable with the RTCEV bits. The RTCTEVIFG flag is set when the event occurs
  - Real-time clock alarm event interrupt enable controlled by the RTCAIE bit. This allows the alarm mode to generate an interrupt when the programmed time in the real-time clock alarm registers is reached. The RTCAIFG flag is set when the alarm is reached.
  - A flag that determines if the RTC can be read safely. This can be checked with the RTCRDYIFG flag.
- The RTCCTL register is now the RTCCTL1 register with the following changes:
  - The RTC modes have been consolidated so RTCMODE is now a single bit. RTCMODE no longer selects the clock source. Setting the RTCMODE bit puts the RTC in calendar mode.
  - The RTCRDY bit is a read-only bit that, when is zero, signifies that the RTC time values are in transition.
  - The RTCIE and RTCFG bits have been moved from this register to RTCCTL0.
- New registers (RTCCTL2 and RTCCTL3) that support the RTC calibration for offset. See the user's guide chapter for use of this functionality.
- RTCNT1 to RTCNT4 have no changes from the MSP430x4xx family of devices.
- RTCSEC, RTCMIN, RTCHOUR, RTCDOW, RTCDOM, RTCMON, RTCYEARL, and RTCYEARH are unchanged.
- New registers are added to support calendar mode alarm functionality – RTCAMIN, RTCAHOUR, RTCADOW, and RTCADAY.
- New registers to set the real-time prescalers used for the Basic Timer control, which has been integrated into the RTC.
  - RTCPS0CTL
    - RT0SSEL selects the prescaler clock source, either ACLK or SMCLK. In calendar mode, these are do not care, and RT0PS is set to ACLK.
    - RT0PSDIV, these three bits select the divide for the prescaler from 2 to 256.
    - RT0PSHOLD stops the RT0PS when in counter mode if set, it is do not care in calendar mode
    - RT0IP, these three bits select the divide down from the prescaler timer 0 interval.
    - RT0IE and RT0IFG enable the prescaler interrupt and the associated flag.
  - RTCPS1CTL
    - RT1SSEL selects the clock from ACLK, SMCLK, or the output from the RT0PS. In calendar mode, the RT0PS output is automatically selected.
    - RT1PSDIV, these three bits select the divide for the prescaler from 2 to 256.
    - RT1PSHOLD stops the RT0PS when in counter mode if set, and it is do not care in calendar mode.
    - RT1IP, these three bits select the divide down from the prescaler timer 0 interval.
    - RT1IE and RT1IFG enable the prescaler interrupt and the associated flag.
    - RTCPS0 and RTCPS1 are counter registers for the prescalers.
- RTCIV, RTC interrupt vector register, four bits that, when an RTC interrupt occurs, may be added to the program counter to service the proper interrupt event. This replaces the IE2 and IFG2 registers in the MSP430x4xx family of devices.

11 32-Bit Hardware Multiplier (MPY32)

The 32-bit hardware multiplier has a number of new features apart from 32-bit operations. It supports fractional modes based on the Q14 and Q15 standards and utilizes saturation to prevent overflow and underflow of signed operations. The support for operations under the previous 16-bit hardware multiplier is unchanged. The registers used exist in their original formats and, unless the MPY32CTL0 register is modified from its default state, the hardware multiplier functions as a 16-bit hardware multiplier. Information on using the functionality of the 32-bit hardware multiplier is found in the MPY32 chapter of the *MSP430x5xx Family User's Guide* (SLAU208).

12 Universal Serial Communications Interface (USCI) I2C Mode

The most significant difference in the USCI module for the I2C mode is unifying the interrupt vectors into a single vector. This required the move of the UCNACKIFG, UCSTPIFG, UCSTTIFG, and UCALIFG, which were bits 3:0 in the UCBxSTAT, USCI_Bx Status Register to the USBxIFG, USCI_Bx Interrupt Flag Register. The IFG2 register has been replaced by the USBxIFG register and still contains the UCBxTXIFG and UCBxRXIFG flags which have been renamed UCTXIFG and UCRXIFG. The reason for the change is twofold: first, collecting the flags into a single register simplifies testing them and, second, to support Interrupt Vector Generator, which is a more efficient, linear methodology for servicing I2C ISRs. An example of the new UCBxIV servicing follows. The new bit definitions are in the table below.

Table 7. USCI I2C IFG/IE Bit Changes

IFG Name	MSP430x4xx Register/Bit Position	MSP430x5xx Register/Bit Position
UCNACKIFG/IE	UCBxSTAT/UCBxIE, bit 3	UCBxIFG, bit 5
UCALIFG/IE	UCBxSTAT, bit 0	UCBxIFG, bit 4
UCSTPIFG/IE	UCBxSTAT, bit 2	UCBxIFG, bit 3
UCSTTIFG/IE	UCBxSTAT, bit 1	UCBxIFG, bit 2
UCTXIFG/IE	was UCBxTXIFG	UCBxIFG, bit 1
UCRXIFG/IE	was UCBxTXIFG	UCBxIFG, bit 0

Table 8. USCI I2C IV Table

UCBxIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	–	
02h	Arbitration lost	UCALIFG	Highest
04h	Not acknowledgement	UNACKIFG	
06h	Start condition received	UCSTTIFG	
08h	Stop condition received	UCSTPIFG	
0Ah	Data received	UCRXIFG	
0Ch	Transmit buffer empty	UCTXIFG	Lowest

```
;Interrupt handler for USCI_I2C_ISR                      Cycles
P1_HND                          ; Interrupt latency         6
         ADD    &UCB0IV,PC      ; Add offset to jump table  3
         RETI                   ; Vector 0: No interrupt    5
         JMP    ALIFG_ISR       ; Vector 2: ALIFG           2
         JMP    NACKIFG_ISR     ; Vector 4: NACKIFG         2
         JMP    STTIFG_ISR      ; Vector 6: STTIFG          2
         JMP    STPIFG_ISR      ; Vector 8: STPIFG          2
         JMP    RXIFG_ISR       ; Vector 10: RXIFG          2
         JMP    TXIFG_ISR       ; Vector 12: TXIFG          2
ALIFG_ISR                       ; Vector 2: ALIFG
         ...                    ; Task starts here
         RETI                   ; Return to main program    5
NACKIFG_ISR                     ; Vector 4: NACKIFG
         ...                    ; Task starts here
         RETI                   ; Return to main program    5
STTIFG_ISR                      ; Vector 6: STTIFG
         ...                    ; Task starts here
         RETI                   ; Return to main program    5
STPIFG_ISR                      ; Vector 8: STPIFG
         ...                    ; Task starts here
         RETI                   ; Return to main program    5
RXIFG_ISR                       ; Vector 10: RXIFG
         ...                    ; Task starts here
         RETI                   ; Return to main program    5
TXIFG_ISR                       ; Vector 12: TXIFG
         ...                    ; Task starts here
         RETI                   ; Return to main program    5
```

13 Universal Serial Communications Interface (USCI) SPI Mode

The most significant difference in the USCI module for the SPI mode is the separation of RX and TX interrupt enables into UCAxIE and UCBxIE and the interrupt flags into UCAxIFG and UCBxIFG registers. The other change is the use of UCIVx, the interrupt vector register, which has the definitions shown in Table 10.

Table 9. USCI SPI IV Table

UCBxIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	–	
02h	Data received	UCRXIFG	Highest
04h	Transmit buffer empty	UCTXIFG	Lowest

```
;Interrupt handler for USCI_SPI_ISR                    Cycles
P1_HND                      ; Interrupt latency          6
        ADD    &UCB0IV,PC   ; Add offset to jump table    3
        RETI                ; Vector 0: No interrupt      5
        JMP    RXIFG_ISR    ; Vector 2: RXIFG             2
        JMP    TXIFG_ISR    ; Vector 4: TXIFG             2
RXIFG_ISR                   ; Vector 2: RXIFG
        ...                 ; Task starts here
        RETI                ; Return to main program      5
TXIFG_ISR                   ; Vector 4: TXIFG
        ...                 ; Task starts here
        RETI                ; Return to main program      5
```

14 Universal Serial Communications Interface (USCI) UART Mode

The most significant difference in the USCI module for the UART mode is the separation of RX and TX interrupts enables into UCAxIE and the interrupt flags into UCAxIFG and UCBxIFG registers. The other change is the use of UCIVx the Interrupt vector register which has the definitions shown in Table 9.

Table 10. USCI UART IV Table

UCBxIV Contents	Interrupt Source	Interrupt Flag	Interrupt Priority
00h	No interrupt pending	–	
02h	Data received	UCRXIFG	Highest
04h	Transmit buffer empty	UCTXIFG	Lowest

```
;Interrupt handler for USCI_UART_ISR                   Cycles
P1_HND                      ; Interrupt latency          6
        ADD    &UCB0IV,PC   ; Add offset to jump table   3
        RETI                ; Vector 0: No interrupt     5
        JMP    RXIFG_ISR    ; Vector 2: RXIFG            2
        JMP    TXIFG_ISR    ; Vector 4: TXIFG            2
RXIFG_ISR                   ; Vector 2: RXIFG
        ...                 ; Task starts here
        RETI                ; Return to main program     5
TXIFG_ISR                   ; Vector 4: TXIFG
        ...                 ; Task starts here
        RETI                ; Return to main program     5
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265