/**
*
* Standard register and bit definitions for the Texas Instruments
* MSP430 microcontroller.
*
* This file supports assembler and C development for
* MSP430FR6889 devices.
*

```
* Texas Instruments, Version 1.1
*
* Rev. 1.0, Setup
* Rev. 1.1, ESI: Renamed bit ESIVCC2 to ESIVMIDEN, renamed bit ESIVSS to ESISHTSM
*
*
************************************************************************/

#ifndef __MSP430FR6889
#define __MSP430FR6889

#define __MSP430_HAS_MSP430XV2_CPU__                    /* Definition to show that it has
MSP430XV2 CPU */
#define __MSP430FR5XX_6XX_FAMILY__

#define __MSP430_HEADER_VERSION__ 1203

#ifdef __cplusplus
extern "C" {
#endif


/*----------------------------------------------------------------------------*/
/* PERIPHERAL FILE MAP                                                        */
/*----------------------------------------------------------------------------*/

#ifndef SFR_8BIT
/* External references resolved by a device-specific linker command file */
#define SFR_8BIT(address)      extern volatile unsigned char address
#define SFR_16BIT(address)     extern volatile unsigned int address
//#define SFR_20BIT(address)     extern volatile unsigned int address
typedef void (* __SFR_FARPTR)();
#define SFR_20BIT(address) extern __SFR_FARPTR address
#define SFR_32BIT(address)     extern volatile unsigned long address

#endif


/************************************************************
* STANDARD BITS
************************************************************/

#define BIT0                    (0x0001)
#define BIT1                    (0x0002)
#define BIT2                    (0x0004)
#define BIT3                    (0x0008)
#define BIT4                    (0x0010)
#define BIT5                    (0x0020)
```

```
#define BIT6                        (0x0040)
#define BIT7                        (0x0080)
#define BIT8                        (0x0100)
#define BIT9                        (0x0200)
#define BITA                        (0x0400)
#define BITB                        (0x0800)
#define BITC                        (0x1000)
#define BITD                        (0x2000)
#define BITE                        (0x4000)
#define BITF                        (0x8000)

/************************************************************
* STATUS REGISTER BITS
************************************************************/

#define C                           (0x0001)
#define Z                           (0x0002)
#define N                           (0x0004)
#define V                           (0x0100)
#define GIE                         (0x0008)
#define CPUOFF                      (0x0010)
#define OSCOFF                      (0x0020)
#define SCG0                        (0x0040)
#define SCG1                        (0x0080)

/* Low Power Modes coded with Bits 4-7 in SR */

#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define LPM0                        (CPUOFF)
#define LPM1                        (SCG0+CPUOFF)
#define LPM2                        (SCG1+CPUOFF)
#define LPM3                        (SCG1+SCG0+CPUOFF)
#define LPM4                        (SCG1+SCG0+OSCOFF+CPUOFF)
/* End #defines for assembler */

#else /* Begin #defines for C */
#define LPM0_bits                   (CPUOFF)
#define LPM1_bits                   (SCG0+CPUOFF)
#define LPM2_bits                   (SCG1+CPUOFF)
#define LPM3_bits                   (SCG1+SCG0+CPUOFF)
#define LPM4_bits                   (SCG1+SCG0+OSCOFF+CPUOFF)

#include "in430.h"
#include <intrinsics.h>

#define LPM0          __bis_SR_register(LPM0_bits)          /* Enter Low Power Mode 0 */
#define LPM0_EXIT __bic_SR_register_on_exit(LPM0_bits) /* Exit Low Power Mode 0 */
#define LPM1          __bis_SR_register(LPM1_bits)          /* Enter Low Power Mode 1 */
```

```
#define LPM1_EXIT __bic_SR_register_on_exit(LPM1_bits) /* Exit Low Power Mode 1 */
#define LPM2          __bis_SR_register(LPM2_bits)              /* Enter Low Power Mode 2 */
#define LPM2_EXIT __bic_SR_register_on_exit(LPM2_bits) /* Exit Low Power Mode 2 */
#define LPM3          __bis_SR_register(LPM3_bits)              /* Enter Low Power Mode 3 */
#define LPM3_EXIT __bic_SR_register_on_exit(LPM3_bits) /* Exit Low Power Mode 3 */
#define LPM4          __bis_SR_register(LPM4_bits)              /* Enter Low Power Mode 4 */
#define LPM4_EXIT __bic_SR_register_on_exit(LPM4_bits) /* Exit Low Power Mode 4 */
#endif /* End #defines for C */

/************************************************************
* PERIPHERAL FILE MAP
************************************************************/

/************************************************************
* ADC12_B
************************************************************/
/************************************************************
* Capacitive_Touch_IO 0
************************************************************/
/************************************************************
* Capacitive_Touch_IO 1
************************************************************/
/************************************************************
* Comparator E
************************************************************/

/************************************************************
* CRC Module
************************************************************/
/************************************************************
* CRC Module
************************************************************/

/************************************************************
* CLOCK SYSTEM
************************************************************/
#define __MSP430_HAS_CS__                          /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_CS__ 0x0160
#define CS_BASE                    __MSP430_BASEADDRESS_CS__

SFR_16BIT(CSCTL0);                                 /* CS Control Register 0 */
SFR_8BIT(CSCTL0_L);                                /* CS Control Register 0 */
SFR_8BIT(CSCTL0_H);                                /* CS Control Register 0 */
SFR_16BIT(CSCTL1);                                 /* CS Control Register 1 */
SFR_8BIT(CSCTL1_L);                                /* CS Control Register 1 */
SFR_8BIT(CSCTL1_H);                                /* CS Control Register 1 */
SFR_16BIT(CSCTL2);                                 /* CS Control Register 2 */
```

```
SFR_8BIT(CSCTL2_L);                     /* CS Control Register 2 */
SFR_8BIT(CSCTL2_H);                     /* CS Control Register 2 */
SFR_16BIT(CSCTL3);                      /* CS Control Register 3 */
SFR_8BIT(CSCTL3_L);                     /* CS Control Register 3 */
SFR_8BIT(CSCTL3_H);                     /* CS Control Register 3 */
SFR_16BIT(CSCTL4);                      /* CS Control Register 4 */
SFR_8BIT(CSCTL4_L);                     /* CS Control Register 4 */
SFR_8BIT(CSCTL4_H);                     /* CS Control Register 4 */
SFR_16BIT(CSCTL5);                      /* CS Control Register 5 */
SFR_8BIT(CSCTL5_L);                     /* CS Control Register 5 */
SFR_8BIT(CSCTL5_H);                     /* CS Control Register 5 */
SFR_16BIT(CSCTL6);                      /* CS Control Register 6 */
SFR_8BIT(CSCTL6_L);                     /* CS Control Register 6 */
SFR_8BIT(CSCTL6_H);                     /* CS Control Register 6 */

/* CSCTL0 Control Bits */

#define CSKEY                   (0xA500)        /* CS Password */
#define CSKEY_H                 (0xA5)          /* CS Password for high byte access */

/* CSCTL1 Control Bits */
#define DCOFSEL0                (0x0002)        /* DCO frequency select Bit: 0 */
#define DCOFSEL1                (0x0004)        /* DCO frequency select Bit: 1 */
#define DCOFSEL2                (0x0008)        /* DCO frequency select Bit: 2 */
#define DCORSEL                 (0x0040)        /* DCO range select. */

/* CSCTL1 Control Bits */
#define DCOFSEL0_L              (0x0002)        /* DCO frequency select Bit: 0 */
#define DCOFSEL1_L              (0x0004)        /* DCO frequency select Bit: 1 */
#define DCOFSEL2_L              (0x0008)        /* DCO frequency select Bit: 2 */
#define DCORSEL_L               (0x0040)        /* DCO range select. */

#define DCOFSEL_0               (0x0000)        /* DCO frequency select: 0 */
#define DCOFSEL_1               (0x0002)        /* DCO frequency select: 1 */
#define DCOFSEL_2               (0x0004)        /* DCO frequency select: 2 */
#define DCOFSEL_3               (0x0006)        /* DCO frequency select: 3 */
#define DCOFSEL_4               (0x0008)        /* DCO frequency select: 4 */
#define DCOFSEL_5               (0x000A)        /* DCO frequency select: 5 */
#define DCOFSEL_6               (0x000C)        /* DCO frequency select: 6 */
#define DCOFSEL_7               (0x000E)        /* DCO frequency select: 7 */

/* CSCTL2 Control Bits */
#define SELM0                   (0x0001)        /* MCLK Source Select Bit: 0 */
#define SELM1                   (0x0002)        /* MCLK Source Select Bit: 1 */
#define SELM2                   (0x0004)        /* MCLK Source Select Bit: 2 */
//#define RESERVED              (0x0004)      /* RESERVED */
//#define RESERVED              (0x0008)      /* RESERVED */
#define SELS0                   (0x0010)        /* SMCLK Source Select Bit: 0 */
```

```
#define SELS1                   (0x0020)          /* SMCLK Source Select Bit: 1 */
#define SELS2                   (0x0040)          /* SMCLK Source Select Bit: 2 */
//#define RESERVED              (0x0040)      /* RESERVED */
//#define RESERVED              (0x0080)      /* RESERVED */
#define SELA0                   (0x0100)          /* ACLK Source Select Bit: 0 */
#define SELA1                   (0x0200)          /* ACLK Source Select Bit: 1 */
#define SELA2                   (0x0400)          /* ACLK Source Select Bit: 2 */
//#define RESERVED              (0x0400)      /* RESERVED */
//#define RESERVED              (0x0800)      /* RESERVED */
//#define RESERVED              (0x1000)      /* RESERVED */
//#define RESERVED              (0x2000)      /* RESERVED */
//#define RESERVED              (0x4000)      /* RESERVED */
//#define RESERVED              (0x8000)      /* RESERVED */

/* CSCTL2 Control Bits */
#define SELM0_L                 (0x0001)          /* MCLK Source Select Bit: 0 */
#define SELM1_L                 (0x0002)          /* MCLK Source Select Bit: 1 */
#define SELM2_L                 (0x0004)          /* MCLK Source Select Bit: 2 */
//#define RESERVED              (0x0004)      /* RESERVED */
//#define RESERVED              (0x0008)      /* RESERVED */
#define SELS0_L                 (0x0010)          /* SMCLK Source Select Bit: 0 */
#define SELS1_L                 (0x0020)          /* SMCLK Source Select Bit: 1 */
#define SELS2_L                 (0x0040)          /* SMCLK Source Select Bit: 2 */
//#define RESERVED              (0x0040)      /* RESERVED */
//#define RESERVED              (0x0080)      /* RESERVED */
//#define RESERVED              (0x0400)      /* RESERVED */
//#define RESERVED              (0x0800)      /* RESERVED */
//#define RESERVED              (0x1000)      /* RESERVED */
//#define RESERVED              (0x2000)      /* RESERVED */
//#define RESERVED              (0x4000)      /* RESERVED */
//#define RESERVED              (0x8000)      /* RESERVED */

/* CSCTL2 Control Bits */
//#define RESERVED              (0x0004)      /* RESERVED */
//#define RESERVED              (0x0008)      /* RESERVED */
//#define RESERVED              (0x0040)      /* RESERVED */
//#define RESERVED              (0x0080)      /* RESERVED */
#define SELA0_H                 (0x0001)          /* ACLK Source Select Bit: 0 */
#define SELA1_H                 (0x0002)          /* ACLK Source Select Bit: 1 */
#define SELA2_H                 (0x0004)          /* ACLK Source Select Bit: 2 */
//#define RESERVED              (0x0400)      /* RESERVED */
//#define RESERVED              (0x0800)      /* RESERVED */
//#define RESERVED              (0x1000)      /* RESERVED */
//#define RESERVED              (0x2000)      /* RESERVED */
//#define RESERVED              (0x4000)      /* RESERVED */
//#define RESERVED              (0x8000)      /* RESERVED */

#define SELM_0                  (0x0000)          /* MCLK Source Select 0 */
```

```
#define SELM_1                  (0x0001)        /* MCLK Source Select 1 */
#define SELM_2                  (0x0002)        /* MCLK Source Select 2 */
#define SELM_3                  (0x0003)        /* MCLK Source Select 3 */
#define SELM_4                  (0x0004)        /* MCLK Source Select 4 */
#define SELM_5                  (0x0005)        /* MCLK Source Select 5 */
#define SELM_6                  (0x0006)        /* MCLK Source Select 6 */
#define SELM_7                  (0x0007)        /* MCLK Source Select 7 */
#define SELM__LFXTCLK           (0x0000)        /* MCLK Source Select LFXTCLK */
#define SELM__VLOCLK            (0x0001)        /* MCLK Source Select VLOCLK */
#define SELM__LFMODCLK          (0x0002)        /* MCLK Source Select LFMODOSC */
#define SELM__LFMODOSC          (0x0002)        /* MCLK Source Select LFMODOSC
(legacy) */
#define SELM__DCOCLK            (0x0003)        /* MCLK Source Select DCOCLK */
#define SELM__MODCLK            (0x0004)        /* MCLK Source Select MODOSC */
#define SELM__MODOSC            (0x0004)        /* MCLK Source Select MODOSC (legacy)
*/
#define SELM__HFXTCLK           (0x0005)        /* MCLK Source Select HFXTCLK */

#define SELS_0                  (0x0000)        /* SMCLK Source Select 0 */
#define SELS_1                  (0x0010)        /* SMCLK Source Select 1 */
#define SELS_2                  (0x0020)        /* SMCLK Source Select 2 */
#define SELS_3                  (0x0030)        /* SMCLK Source Select 3 */
#define SELS_4                  (0x0040)        /* SMCLK Source Select 4 */
#define SELS_5                  (0x0050)        /* SMCLK Source Select 5 */
#define SELS_6                  (0x0060)        /* SMCLK Source Select 6 */
#define SELS_7                  (0x0070)        /* SMCLK Source Select 7 */
#define SELS__LFXTCLK           (0x0000)        /* SMCLK Source Select LFXTCLK */
#define SELS__VLOCLK            (0x0010)        /* SMCLK Source Select VLOCLK */
#define SELS__LFMODCLK          (0x0020)        /* SMCLK Source Select LFMODOSC */
#define SELS__LFMODOSC          (0x0020)        /* SMCLK Source Select LFMODOSC
(legacy) */
#define SELS__DCOCLK            (0x0030)        /* SMCLK Source Select DCOCLK */
#define SELS__MODCLK            (0x0040)        /* SMCLK Source Select MODOSC */
#define SELS__MODOSC            (0x0040)        /* SMCLK Source Select MODOSC (legacy)
*/
#define SELS__HFXTCLK           (0x0050)        /* SMCLK Source Select HFXTCLK */

#define SELA_0                  (0x0000)        /* ACLK Source Select 0 */
#define SELA_1                  (0x0100)        /* ACLK Source Select 1 */
#define SELA_2                  (0x0200)        /* ACLK Source Select 2 */
#define SELA_3                  (0x0300)        /* ACLK Source Select 3 */
#define SELA_4                  (0x0400)        /* ACLK Source Select 4 */
#define SELA_5                  (0x0500)        /* ACLK Source Select 5 */
#define SELA_6                  (0x0600)        /* ACLK Source Select 6 */
#define SELA_7                  (0x0700)        /* ACLK Source Select 7 */
#define SELA__LFXTCLK           (0x0000)        /* ACLK Source Select LFXTCLK */
#define SELA__VLOCLK            (0x0100)        /* ACLK Source Select VLOCLK */
#define SELA__LFMODCLK          (0x0200)        /* ACLK Source Select LFMODOSC */
```

```
#define SELA__LFMODOSC              (0x0200)          /* ACLK Source Select LFMODOSC (legacy)
*/

/* CSCTL3 Control Bits */
#define DIVM0                       (0x0001)          /* MCLK Divider Bit: 0 */
#define DIVM1                       (0x0002)          /* MCLK Divider Bit: 1 */
#define DIVM2                       (0x0004)          /* MCLK Divider Bit: 2 */
//#define RESERVED                  (0x0004)      /* RESERVED */
//#define RESERVED                  (0x0008)      /* RESERVED */
#define DIVS0                       (0x0010)          /* SMCLK Divider Bit: 0 */
#define DIVS1                       (0x0020)          /* SMCLK Divider Bit: 1 */
#define DIVS2                       (0x0040)          /* SMCLK Divider Bit: 2 */
//#define RESERVED                  (0x0040)      /* RESERVED */
//#define RESERVED                  (0x0080)      /* RESERVED */
#define DIVA0                       (0x0100)          /* ACLK Divider Bit: 0 */
#define DIVA1                       (0x0200)          /* ACLK Divider Bit: 1 */
#define DIVA2                       (0x0400)          /* ACLK Divider Bit: 2 */
//#define RESERVED                  (0x0400)      /* RESERVED */
//#define RESERVED                  (0x0800)      /* RESERVED */
//#define RESERVED                  (0x1000)      /* RESERVED */
//#define RESERVED                  (0x2000)      /* RESERVED */
//#define RESERVED                  (0x4000)      /* RESERVED */
//#define RESERVED                  (0x8000)      /* RESERVED */

/* CSCTL3 Control Bits */
#define DIVM0_L                     (0x0001)          /* MCLK Divider Bit: 0 */
#define DIVM1_L                     (0x0002)          /* MCLK Divider Bit: 1 */
#define DIVM2_L                     (0x0004)          /* MCLK Divider Bit: 2 */
//#define RESERVED                  (0x0004)      /* RESERVED */
//#define RESERVED                  (0x0008)      /* RESERVED */
#define DIVS0_L                     (0x0010)          /* SMCLK Divider Bit: 0 */
#define DIVS1_L                     (0x0020)          /* SMCLK Divider Bit: 1 */
#define DIVS2_L                     (0x0040)          /* SMCLK Divider Bit: 2 */
//#define RESERVED                  (0x0040)      /* RESERVED */
//#define RESERVED                  (0x0080)      /* RESERVED */
//#define RESERVED                  (0x0400)      /* RESERVED */
//#define RESERVED                  (0x0800)      /* RESERVED */
//#define RESERVED                  (0x1000)      /* RESERVED */
//#define RESERVED                  (0x2000)      /* RESERVED */
//#define RESERVED                  (0x4000)      /* RESERVED */
//#define RESERVED                  (0x8000)      /* RESERVED */

/* CSCTL3 Control Bits */
//#define RESERVED                  (0x0004)      /* RESERVED */
//#define RESERVED                  (0x0008)      /* RESERVED */
//#define RESERVED                  (0x0040)      /* RESERVED */
//#define RESERVED                  (0x0080)      /* RESERVED */
#define DIVA0_H                     (0x0001)          /* ACLK Divider Bit: 0 */
```

```
#define DIVA1_H                 (0x0002)        /* ACLK Divider Bit: 1 */
#define DIVA2_H                 (0x0004)        /* ACLK Divider Bit: 2 */
//#define RESERVED              (0x0400)      /* RESERVED */
//#define RESERVED              (0x0800)      /* RESERVED */
//#define RESERVED              (0x1000)      /* RESERVED */
//#define RESERVED              (0x2000)      /* RESERVED */
//#define RESERVED              (0x4000)      /* RESERVED */
//#define RESERVED              (0x8000)      /* RESERVED */

#define DIVM_0                  (0x0000)        /* MCLK Source Divider 0 */
#define DIVM_1                  (0x0001)        /* MCLK Source Divider 1 */
#define DIVM_2                  (0x0002)        /* MCLK Source Divider 2 */
#define DIVM_3                  (0x0003)        /* MCLK Source Divider 3 */
#define DIVM_4                  (0x0004)        /* MCLK Source Divider 4 */
#define DIVM_5                  (0x0005)        /* MCLK Source Divider 5 */
#define DIVM__1                 (0x0000)        /* MCLK Source Divider f(MCLK)/1 */
#define DIVM__2                 (0x0001)        /* MCLK Source Divider f(MCLK)/2 */
#define DIVM__4                 (0x0002)        /* MCLK Source Divider f(MCLK)/4 */
#define DIVM__8                 (0x0003)        /* MCLK Source Divider f(MCLK)/8 */
#define DIVM__16                (0x0004)        /* MCLK Source Divider f(MCLK)/16 */
#define DIVM__32                (0x0005)        /* MCLK Source Divider f(MCLK)/32 */

#define DIVS_0                  (0x0000)        /* SMCLK Source Divider 0 */
#define DIVS_1                  (0x0010)        /* SMCLK Source Divider 1 */
#define DIVS_2                  (0x0020)        /* SMCLK Source Divider 2 */
#define DIVS_3                  (0x0030)        /* SMCLK Source Divider 3 */
#define DIVS_4                  (0x0040)        /* SMCLK Source Divider 4 */
#define DIVS_5                  (0x0050)        /* SMCLK Source Divider 5 */
#define DIVS__1                 (0x0000)        /* SMCLK Source Divider f(SMCLK)/1 */
#define DIVS__2                 (0x0010)        /* SMCLK Source Divider f(SMCLK)/2 */
#define DIVS__4                 (0x0020)        /* SMCLK Source Divider f(SMCLK)/4 */
#define DIVS__8                 (0x0030)        /* SMCLK Source Divider f(SMCLK)/8 */
#define DIVS__16                (0x0040)        /* SMCLK Source Divider f(SMCLK)/16 */
#define DIVS__32                (0x0050)        /* SMCLK Source Divider f(SMCLK)/32 */

#define DIVA_0                  (0x0000)        /* ACLK Source Divider 0 */
#define DIVA_1                  (0x0100)        /* ACLK Source Divider 1 */
#define DIVA_2                  (0x0200)        /* ACLK Source Divider 2 */
#define DIVA_3                  (0x0300)        /* ACLK Source Divider 3 */
#define DIVA_4                  (0x0400)        /* ACLK Source Divider 4 */
#define DIVA_5                  (0x0500)        /* ACLK Source Divider 5 */
#define DIVA__1                 (0x0000)        /* ACLK Source Divider f(ACLK)/1 */
#define DIVA__2                 (0x0100)        /* ACLK Source Divider f(ACLK)/2 */
#define DIVA__4                 (0x0200)        /* ACLK Source Divider f(ACLK)/4 */
#define DIVA__8                 (0x0300)        /* ACLK Source Divider f(ACLK)/8 */
#define DIVA__16                (0x0400)        /* ACLK Source Divider f(ACLK)/16 */
#define DIVA__32                (0x0500)        /* ACLK Source Divider f(ACLK)/32 */
```

```
/* CSCTL4 Control Bits */
#define LFXTOFF                 (0x0001)          /* Low Frequency Oscillator (LFXT) disable
*/
#define SMCLKOFF                (0x0002)          /* SMCLK Off */
#define VLOOFF                  (0x0008)          /* VLO Off */
#define LFXTBYPASS              (0x0010)          /* LFXT bypass mode : 0: internal 1:sourced
from external pin */
#define LFXTDRIVE0              (0x0040)          /* LFXT Drive Level mode Bit 0 */
#define LFXTDRIVE1              (0x0080)          /* LFXT Drive Level mode Bit 1 */
#define HFXTOFF                 (0x0100)          /* High Frequency Oscillator disable */
#define HFFREQ0                 (0x0400)          /* HFXT frequency selection Bit 1 */
#define HFFREQ1                 (0x0800)          /* HFXT frequency selection Bit 0 */
#define HFXTBYPASS              (0x1000)          /* HFXT bypass mode : 0: internal
1:sourced from external pin */
#define HFXTDRIVE0              (0x4000)          /* HFXT Drive Level mode Bit 0 */
#define HFXTDRIVE1              (0x8000)          /* HFXT Drive Level mode Bit 1 */

/* CSCTL4 Control Bits */
#define LFXTOFF_L               (0x0001)          /* Low Frequency Oscillator (LFXT) disable
*/
#define SMCLKOFF_L              (0x0002)          /* SMCLK Off */
#define VLOOFF_L                (0x0008)          /* VLO Off */
#define LFXTBYPASS_L            (0x0010)          /* LFXT bypass mode : 0: internal 1:sourced
from external pin */
#define LFXTDRIVE0_L            (0x0040)          /* LFXT Drive Level mode Bit 0 */
#define LFXTDRIVE1_L            (0x0080)          /* LFXT Drive Level mode Bit 1 */

/* CSCTL4 Control Bits */
#define HFXTOFF_H               (0x0001)          /* High Frequency Oscillator disable */
#define HFFREQ0_H               (0x0004)          /* HFXT frequency selection Bit 1 */
#define HFFREQ1_H               (0x0008)          /* HFXT frequency selection Bit 0 */
#define HFXTBYPASS_H            (0x0010)          /* HFXT bypass mode : 0: internal
1:sourced from external pin */
#define HFXTDRIVE0_H            (0x0040)          /* HFXT Drive Level mode Bit 0 */
#define HFXTDRIVE1_H            (0x0080)          /* HFXT Drive Level mode Bit 1 */

#define LFXTDRIVE_0             (0x0000)          /* LFXT Drive Level mode: 0 */
#define LFXTDRIVE_1             (0x0040)          /* LFXT Drive Level mode: 1 */
#define LFXTDRIVE_2             (0x0080)          /* LFXT Drive Level mode: 2 */
#define LFXTDRIVE_3             (0x00C0)          /* LFXT Drive Level mode: 3 */

#define HFFREQ_0                (0x0000)          /* HFXT frequency selection: 0 */
#define HFFREQ_1                (0x0400)          /* HFXT frequency selection: 1 */
#define HFFREQ_2                (0x0800)          /* HFXT frequency selection: 2 */
#define HFFREQ_3                (0x0C00)          /* HFXT frequency selection: 3 */

#define HFXTDRIVE_0             (0x0000)          /* HFXT Drive Level mode: 0 */
#define HFXTDRIVE_1             (0x4000)          /* HFXT Drive Level mode: 1 */
```

```
#define HFXTDRIVE_2                (0x8000)          /* HFXT Drive Level mode: 2 */
#define HFXTDRIVE_3                (0xC000)          /* HFXT Drive Level mode: 3 */

/* CSCTL5 Control Bits */
#define LFXTOFFG                   (0x0001)          /* LFXT Low Frequency Oscillator Fault Flag
*/
#define HFXTOFFG                   (0x0002)          /* HFXT High Frequency Oscillator Fault
Flag */
#define ENSTFCNT1                  (0x0040)          /* Enable start counter for XT1 */
#define ENSTFCNT2                  (0x0080)          /* Enable start counter for XT2 */

/* CSCTL5 Control Bits */
#define LFXTOFFG_L                 (0x0001)          /* LFXT Low Frequency Oscillator Fault Flag
*/
#define HFXTOFFG_L                 (0x0002)          /* HFXT High Frequency Oscillator Fault
Flag */
#define ENSTFCNT1_L                (0x0040)          /* Enable start counter for XT1 */
#define ENSTFCNT2_L                (0x0080)          /* Enable start counter for XT2 */

/* CSCTL6 Control Bits */
#define ACLKREQEN                  (0x0001)          /* ACLK Clock Request Enable */
#define MCLKREQEN                  (0x0002)          /* MCLK Clock Request Enable */
#define SMCLKREQEN                 (0x0004)          /* SMCLK Clock Request Enable */
#define MODCLKREQEN                (0x0008)          /* MODOSC Clock Request Enable */

/* CSCTL6 Control Bits */
#define ACLKREQEN_L                (0x0001)          /* ACLK Clock Request Enable */
#define MCLKREQEN_L                (0x0002)          /* MCLK Clock Request Enable */
#define SMCLKREQEN_L               (0x0004)          /* SMCLK Clock Request Enable */
#define MODCLKREQEN_L              (0x0008)          /* MODOSC Clock Request Enable */

/************************************************************
* DMA_X
************************************************************/

/************************************************************
* EXTENDED SCAN INTERFACE
************************************************************/
/************************************************************
* EXTENDED SCAN INTERFACE RAM
************************************************************/
/************************************************************
* FRAM Memory
************************************************************/
#define __MSP430_HAS_FRAM__                       /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_FRAM__ 0x0140
#define FRAM_BASE                  __MSP430_BASEADDRESS_FRAM__
```

```
#define __MSP430_HAS_GC__                       /* Definition to show that Module is
available */

SFR_16BIT(FRCTL0);                              /* FRAM Controller Control 0 */
SFR_8BIT(FRCTL0_L);                             /* FRAM Controller Control 0 */
SFR_8BIT(FRCTL0_H);                             /* FRAM Controller Control 0 */
SFR_16BIT(GCCTL0);                              /* General Control 0 */
SFR_8BIT(GCCTL0_L);                             /* General Control 0 */
SFR_8BIT(GCCTL0_H);                             /* General Control 0 */
SFR_16BIT(GCCTL1);                              /* General Control 1 */
SFR_8BIT(GCCTL1_L);                             /* General Control 1 */
SFR_8BIT(GCCTL1_H);                             /* General Control 1 */

#define FRCTLPW                 (0xA500)        /* FRAM password for write */
#define FRPW                    (0x9600)        /* FRAM password returned by read */
#define FWPW                    (0xA500)        /* FRAM password for write */
#define FXPW                    (0x3300)        /* for use with XOR instruction */

/* FRCTL0 Control Bits */
//#define RESERVED              (0x0001)    /* RESERVED */
//#define RESERVED              (0x0002)    /* RESERVED */
//#define RESERVED              (0x0004)    /* RESERVED */
#define NWAITS0                 (0x0010)        /* FRAM Wait state control Bit: 0 */
#define NWAITS1                 (0x0020)        /* FRAM Wait state control Bit: 1 */
#define NWAITS2                 (0x0040)        /* FRAM Wait state control Bit: 2 */
//#define RESERVED              (0x0080)    /* RESERVED */

/* FRCTL0 Control Bits */
//#define RESERVED              (0x0001)    /* RESERVED */
//#define RESERVED              (0x0002)    /* RESERVED */
//#define RESERVED              (0x0004)    /* RESERVED */
#define NWAITS0_L               (0x0010)        /* FRAM Wait state control Bit: 0 */
#define NWAITS1_L               (0x0020)        /* FRAM Wait state control Bit: 1 */
#define NWAITS2_L               (0x0040)        /* FRAM Wait state control Bit: 2 */
//#define RESERVED              (0x0080)    /* RESERVED */

#define NWAITS_0                (0x0000)        /* FRAM Wait state control: 0 */
#define NWAITS_1                (0x0010)        /* FRAM Wait state control: 1 */
#define NWAITS_2                (0x0020)        /* FRAM Wait state control: 2 */
#define NWAITS_3                (0x0030)        /* FRAM Wait state control: 3 */
#define NWAITS_4                (0x0040)        /* FRAM Wait state control: 4 */
#define NWAITS_5                (0x0050)        /* FRAM Wait state control: 5 */
#define NWAITS_6                (0x0060)        /* FRAM Wait state control: 6 */
#define NWAITS_7                (0x0070)        /* FRAM Wait state control: 7 */

/* GCCTL0 Control Bits */
//#define RESERVED              (0x0001)    /* RESERVED */
#define FRLPMPWR                (0x0002)        /* FRAM Enable FRAM auto power up
```

```
after LPM */
#define FRPWR                      (0x0004)          /* FRAM Power Control */
#define ACCTEIE                    (0x0008)          /* RESERVED */
//#define RESERVED                 (0x0010)    /* RESERVED */
#define CBDIE                      (0x0020)          /* Enable NMI event if correctable bit error
detected */
#define UBDIE                      (0x0040)          /* Enable NMI event if uncorrectable bit
error detected */
#define UBDRSTEN                   (0x0080)          /* Enable Power Up Clear (PUC) reset if
FRAM uncorrectable bit error detected */

/* GCCTL0 Control Bits */
//#define RESERVED                 (0x0001)    /* RESERVED */
#define FRLPMPWR_L                 (0x0002)          /* FRAM Enable FRAM auto power up
after LPM */
#define FRPWR_L                    (0x0004)          /* FRAM Power Control */
#define ACCTEIE_L                  (0x0008)          /* RESERVED */
//#define RESERVED                 (0x0010)    /* RESERVED */
#define CBDIE_L                    (0x0020)          /* Enable NMI event if correctable bit error
detected */
#define UBDIE_L                    (0x0040)          /* Enable NMI event if uncorrectable bit
error detected */
#define UBDRSTEN_L                 (0x0080)          /* Enable Power Up Clear (PUC) reset if
FRAM uncorrectable bit error detected */

/* GCCTL1 Control Bits */
//#define RESERVED                 (0x0001)    /* RESERVED */
#define CBDIFG                     (0x0002)          /* FRAM correctable bit error flag */
#define UBDIFG                     (0x0004)          /* FRAM uncorrectable bit error flag */
#define ACCTEIFG                   (0x0008)          /* Access time error flag */

/* GCCTL1 Control Bits */
//#define RESERVED                 (0x0001)    /* RESERVED */
#define CBDIFG_L                   (0x0002)          /* FRAM correctable bit error flag */
#define UBDIFG_L                   (0x0004)          /* FRAM uncorrectable bit error flag */
#define ACCTEIFG_L                 (0x0008)          /* Access time error flag */

/************************************************************
* LCD_C
************************************************************/
#define __MSP430_HAS_LCD_C__                         /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_LCD_C__ 0x0A00
#define LCD_C_BASE                 __MSP430_BASEADDRESS_LCD_C__

SFR_16BIT(LCDCCTL0);                                 /* LCD_C Control Register 0 */
SFR_8BIT(LCDCCTL0_L);                                /* LCD_C Control Register 0 */
SFR_8BIT(LCDCCTL0_H);                                /* LCD_C Control Register 0 */
```

```
SFR_16BIT(LCDCCTL1);                    /* LCD_C Control Register 1 */
SFR_8BIT(LCDCCTL1_L);                   /* LCD_C Control Register 1 */
SFR_8BIT(LCDCCTL1_H);                   /* LCD_C Control Register 1 */
SFR_16BIT(LCDCBLKCTL);                  /* LCD_C blinking control register */
SFR_8BIT(LCDCBLKCTL_L);                 /* LCD_C blinking control register */
SFR_8BIT(LCDCBLKCTL_H);                 /* LCD_C blinking control register */
SFR_16BIT(LCDCMEMCTL);                  /* LCD_C memory control register */
SFR_8BIT(LCDCMEMCTL_L);                 /* LCD_C memory control register */
SFR_8BIT(LCDCMEMCTL_H);                 /* LCD_C memory control register */
SFR_16BIT(LCDCVCTL);                    /* LCD_C Voltage Control Register */
SFR_8BIT(LCDCVCTL_L);                   /* LCD_C Voltage Control Register */
SFR_8BIT(LCDCVCTL_H);                   /* LCD_C Voltage Control Register */
SFR_16BIT(LCDCPCTL0);                   /* LCD_C Port Control Register 0 */
SFR_8BIT(LCDCPCTL0_L);                  /* LCD_C Port Control Register 0 */
SFR_8BIT(LCDCPCTL0_H);                  /* LCD_C Port Control Register 0 */
SFR_16BIT(LCDCPCTL1);                   /* LCD_C Port Control Register 1 */
SFR_8BIT(LCDCPCTL1_L);                  /* LCD_C Port Control Register 1 */
SFR_8BIT(LCDCPCTL1_H);                  /* LCD_C Port Control Register 1 */
SFR_16BIT(LCDCPCTL2);                   /* LCD_C Port Control Register 2 */
SFR_8BIT(LCDCPCTL2_L);                  /* LCD_C Port Control Register 2 */
SFR_8BIT(LCDCPCTL2_H);                  /* LCD_C Port Control Register 2 */
SFR_16BIT(LCDCCPCTL);                   /* LCD_C Charge Pump Control Register 3
*/
SFR_8BIT(LCDCCPCTL_L);                  /* LCD_C Charge Pump Control Register 3
*/
SFR_8BIT(LCDCCPCTL_H);                  /* LCD_C Charge Pump Control Register 3
*/
SFR_16BIT(LCDCIV);                      /* LCD_C Interrupt Vector Register */

// LCDCCTL0
#define LCDON                   (0x0001)        /* LCD_C LCD On */
#define LCDLP                   (0x0002)        /* LCD_C Low Power Waveform */
#define LCDSON                  (0x0004)        /* LCD_C LCD Segments On */
#define LCDMX0                  (0x0008)        /* LCD_C Mux Rate Bit: 0 */
#define LCDMX1                  (0x0010)        /* LCD_C Mux Rate Bit: 1 */
#define LCDMX2                  (0x0020)        /* LCD_C Mux Rate Bit: 2 */
//#define RESERVED              (0x0040)    /* LCD_C RESERVED */
#define LCDSSEL                 (0x0080)        /* LCD_C Clock Select */
#define LCDPRE0                 (0x0100)        /* LCD_C LCD frequency pre-scaler Bit: 0 */
#define LCDPRE1                 (0x0200)        /* LCD_C LCD frequency pre-scaler Bit: 1 */
#define LCDPRE2                 (0x0400)        /* LCD_C LCD frequency pre-scaler Bit: 2 */
#define LCDDIV0                 (0x0800)        /* LCD_C LCD frequency divider Bit: 0 */
#define LCDDIV1                 (0x1000)        /* LCD_C LCD frequency divider Bit: 1 */
#define LCDDIV2                 (0x2000)        /* LCD_C LCD frequency divider Bit: 2 */
#define LCDDIV3                 (0x4000)        /* LCD_C LCD frequency divider Bit: 3 */
#define LCDDIV4                 (0x8000)        /* LCD_C LCD frequency divider Bit: 4 */

// LCDCCTL0
```

```
#define LCDON_L                 (0x0001)          /* LCD_C LCD On */
#define LCDLP_L                 (0x0002)          /* LCD_C Low Power Waveform */
#define LCDSON_L                (0x0004)          /* LCD_C LCD Segments On */
#define LCDMX0_L                (0x0008)          /* LCD_C Mux Rate Bit: 0 */
#define LCDMX1_L                (0x0010)          /* LCD_C Mux Rate Bit: 1 */
#define LCDMX2_L                (0x0020)          /* LCD_C Mux Rate Bit: 2 */
//#define RESERVED              (0x0040)    /* LCD_C RESERVED */
#define LCDSSEL_L               (0x0080)          /* LCD_C Clock Select */

// LCDCCTL0
//#define RESERVED              (0x0040)    /* LCD_C RESERVED */
#define LCDPRE0_H               (0x0001)          /* LCD_C LCD frequency pre-scaler Bit: 0 */
#define LCDPRE1_H               (0x0002)          /* LCD_C LCD frequency pre-scaler Bit: 1 */
#define LCDPRE2_H               (0x0004)          /* LCD_C LCD frequency pre-scaler Bit: 2 */
#define LCDDIV0_H               (0x0008)          /* LCD_C LCD frequency divider Bit: 0 */
#define LCDDIV1_H               (0x0010)          /* LCD_C LCD frequency divider Bit: 1 */
#define LCDDIV2_H               (0x0020)          /* LCD_C LCD frequency divider Bit: 2 */
#define LCDDIV3_H               (0x0040)          /* LCD_C LCD frequency divider Bit: 3 */
#define LCDDIV4_H               (0x0080)          /* LCD_C LCD frequency divider Bit: 4 */

#define LCDPRE_0                (0x0000)          /* LCD_C LCD frequency pre-scaler: /1 */
#define LCDPRE_1                (0x0100)          /* LCD_C LCD frequency pre-scaler: /2 */
#define LCDPRE_2                (0x0200)          /* LCD_C LCD frequency pre-scaler: /4 */
#define LCDPRE_3                (0x0300)          /* LCD_C LCD frequency pre-scaler: /8 */
#define LCDPRE_4                (0x0400)          /* LCD_C LCD frequency pre-scaler: /16 */
#define LCDPRE_5                (0x0500)          /* LCD_C LCD frequency pre-scaler: /32 */
#define LCDPRE__1               (0x0000)          /* LCD_C LCD frequency pre-scaler: /1 */
#define LCDPRE__2               (0x0100)          /* LCD_C LCD frequency pre-scaler: /2 */
#define LCDPRE__4               (0x0200)          /* LCD_C LCD frequency pre-scaler: /4 */
#define LCDPRE__8               (0x0300)          /* LCD_C LCD frequency pre-scaler: /8 */
#define LCDPRE__16              (0x0400)          /* LCD_C LCD frequency pre-scaler: /16 */
#define LCDPRE__32              (0x0500)          /* LCD_C LCD frequency pre-scaler: /32 */

#define LCDDIV_0                (0x0000)          /* LCD_C LCD frequency divider: /1 */
#define LCDDIV_1                (0x0800)          /* LCD_C LCD frequency divider: /2 */
#define LCDDIV_2                (0x1000)          /* LCD_C LCD frequency divider: /3 */
#define LCDDIV_3                (0x1800)          /* LCD_C LCD frequency divider: /4 */
#define LCDDIV_4                (0x2000)          /* LCD_C LCD frequency divider: /5 */
#define LCDDIV_5                (0x2800)          /* LCD_C LCD frequency divider: /6 */
#define LCDDIV_6                (0x3000)          /* LCD_C LCD frequency divider: /7 */
#define LCDDIV_7                (0x3800)          /* LCD_C LCD frequency divider: /8 */
#define LCDDIV_8                (0x4000)          /* LCD_C LCD frequency divider: /9 */
#define LCDDIV_9                (0x4800)          /* LCD_C LCD frequency divider: /10 */
#define LCDDIV_10               (0x5000)          /* LCD_C LCD frequency divider: /11 */
#define LCDDIV_11               (0x5800)          /* LCD_C LCD frequency divider: /12 */
#define LCDDIV_12               (0x6000)          /* LCD_C LCD frequency divider: /13 */
#define LCDDIV_13               (0x6800)          /* LCD_C LCD frequency divider: /14 */
#define LCDDIV_14               (0x7000)          /* LCD_C LCD frequency divider: /15 */
```

```
#define LCDDIV_15       (0x7800)    /* LCD_C LCD frequency divider: /16 */
#define LCDDIV_16       (0x8000)    /* LCD_C LCD frequency divider: /17 */
#define LCDDIV_17       (0x8800)    /* LCD_C LCD frequency divider: /18 */
#define LCDDIV_18       (0x9000)    /* LCD_C LCD frequency divider: /19 */
#define LCDDIV_19       (0x9800)    /* LCD_C LCD frequency divider: /20 */
#define LCDDIV_20       (0xA000)    /* LCD_C LCD frequency divider: /21 */
#define LCDDIV_21       (0xA800)    /* LCD_C LCD frequency divider: /22 */
#define LCDDIV_22       (0xB000)    /* LCD_C LCD frequency divider: /23 */
#define LCDDIV_23       (0xB800)    /* LCD_C LCD frequency divider: /24 */
#define LCDDIV_24       (0xC000)    /* LCD_C LCD frequency divider: /25 */
#define LCDDIV_25       (0xC800)    /* LCD_C LCD frequency divider: /26 */
#define LCDDIV_26       (0xD000)    /* LCD_C LCD frequency divider: /27 */
#define LCDDIV_27       (0xD800)    /* LCD_C LCD frequency divider: /28 */
#define LCDDIV_28       (0xE000)    /* LCD_C LCD frequency divider: /29 */
#define LCDDIV_29       (0xE800)    /* LCD_C LCD frequency divider: /30 */
#define LCDDIV_30       (0xF000)    /* LCD_C LCD frequency divider: /31 */
#define LCDDIV_31       (0xF800)    /* LCD_C LCD frequency divider: /32 */
#define LCDDIV__1       (0x0000)    /* LCD_C LCD frequency divider: /1 */
#define LCDDIV__2       (0x0800)    /* LCD_C LCD frequency divider: /2 */
#define LCDDIV__3       (0x1000)    /* LCD_C LCD frequency divider: /3 */
#define LCDDIV__4       (0x1800)    /* LCD_C LCD frequency divider: /4 */
#define LCDDIV__5       (0x2000)    /* LCD_C LCD frequency divider: /5 */
#define LCDDIV__6       (0x2800)    /* LCD_C LCD frequency divider: /6 */
#define LCDDIV__7       (0x3000)    /* LCD_C LCD frequency divider: /7 */
#define LCDDIV__8       (0x3800)    /* LCD_C LCD frequency divider: /8 */
#define LCDDIV__9       (0x4000)    /* LCD_C LCD frequency divider: /9 */
#define LCDDIV__10      (0x4800)    /* LCD_C LCD frequency divider: /10 */
#define LCDDIV__11      (0x5000)    /* LCD_C LCD frequency divider: /11 */
#define LCDDIV__12      (0x5800)    /* LCD_C LCD frequency divider: /12 */
#define LCDDIV__13      (0x6000)    /* LCD_C LCD frequency divider: /13 */
#define LCDDIV__14      (0x6800)    /* LCD_C LCD frequency divider: /14 */
#define LCDDIV__15      (0x7000)    /* LCD_C LCD frequency divider: /15 */
#define LCDDIV__16      (0x7800)    /* LCD_C LCD frequency divider: /16 */
#define LCDDIV__17      (0x8000)    /* LCD_C LCD frequency divider: /17 */
#define LCDDIV__18      (0x8800)    /* LCD_C LCD frequency divider: /18 */
#define LCDDIV__19      (0x9000)    /* LCD_C LCD frequency divider: /19 */
#define LCDDIV__20      (0x9800)    /* LCD_C LCD frequency divider: /20 */
#define LCDDIV__21      (0xA000)    /* LCD_C LCD frequency divider: /21 */
#define LCDDIV__22      (0xA800)    /* LCD_C LCD frequency divider: /22 */
#define LCDDIV__23      (0xB000)    /* LCD_C LCD frequency divider: /23 */
#define LCDDIV__24      (0xB800)    /* LCD_C LCD frequency divider: /24 */
#define LCDDIV__25      (0xC000)    /* LCD_C LCD frequency divider: /25 */
#define LCDDIV__26      (0xC800)    /* LCD_C LCD frequency divider: /26 */
#define LCDDIV__27      (0xD000)    /* LCD_C LCD frequency divider: /27 */
#define LCDDIV__28      (0xD800)    /* LCD_C LCD frequency divider: /28 */
#define LCDDIV__29      (0xE000)    /* LCD_C LCD frequency divider: /29 */
#define LCDDIV__30      (0xE800)    /* LCD_C LCD frequency divider: /30 */
#define LCDDIV__31      (0xF000)    /* LCD_C LCD frequency divider: /31 */
```

```
#define LCDDIV__32                  (0xF800)          /* LCD_C LCD frequency divider: /32 */

/* Display modes coded with Bits 2-4 */
#define LCDSTATIC                   (LCDSON)
#define LCD2MUX                     (LCDMX0+LCDSON)
#define LCD3MUX                     (LCDMX1+LCDSON)
#define LCD4MUX                     (LCDMX1+LCDMX0+LCDSON)
#define LCD5MUX                     (LCDMX2+LCDSON)
#define LCD6MUX                     (LCDMX2+LCDMX0+LCDSON)
#define LCD7MUX                     (LCDMX2+LCDMX1+LCDSON)
#define LCD8MUX                     (LCDMX2+LCDMX1+LCDMX0+LCDSON)

// LCDCCTL1
#define LCDFRMIFG                   (0x0001)          /* LCD_C LCD frame interrupt flag */
#define LCDBLKOFFIFG                (0x0002)          /* LCD_C LCD blinking off interrupt flag, */
#define LCDBLKONIFG                 (0x0004)          /* LCD_C LCD blinking on interrupt flag, */
#define LCDNOCAPIFG                 (0x0008)          /* LCD_C No cpacitance connected
interrupt flag */
#define LCDFRMIE                    (0x0100)          /* LCD_C LCD frame interrupt enable */
#define LCDBLKOFFIE                 (0x0200)          /* LCD_C LCD blinking off interrupt flag, */
#define LCDBLKONIE                  (0x0400)          /* LCD_C LCD blinking on interrupt flag, */
#define LCDNOCAPIE                  (0x0800)          /* LCD_C No cpacitance connected
interrupt enable */

// LCDCCTL1
#define LCDFRMIFG_L                 (0x0001)          /* LCD_C LCD frame interrupt flag */
#define LCDBLKOFFIFG_L              (0x0002)          /* LCD_C LCD blinking off interrupt flag, */
#define LCDBLKONIFG_L               (0x0004)          /* LCD_C LCD blinking on interrupt flag, */
#define LCDNOCAPIFG_L               (0x0008)          /* LCD_C No cpacitance connected
interrupt flag */

// LCDCCTL1
#define LCDFRMIE_H                  (0x0001)          /* LCD_C LCD frame interrupt enable */
#define LCDBLKOFFIE_H               (0x0002)          /* LCD_C LCD blinking off interrupt flag, */
#define LCDBLKONIE_H                (0x0004)          /* LCD_C LCD blinking on interrupt flag, */
#define LCDNOCAPIE_H                (0x0008)          /* LCD_C No cpacitance connected
interrupt enable */

// LCDCBLKCTL
#define LCDBLKMOD0                  (0x0001)          /* LCD_C Blinking mode Bit: 0 */
#define LCDBLKMOD1                  (0x0002)          /* LCD_C Blinking mode Bit: 1 */
#define LCDBLKPRE0                  (0x0004)          /* LCD_C Clock pre-scaler for blinking
frequency Bit: 0 */
#define LCDBLKPRE1                  (0x0008)          /* LCD_C Clock pre-scaler for blinking
frequency Bit: 1 */
#define LCDBLKPRE2                  (0x0010)          /* LCD_C Clock pre-scaler for blinking
frequency Bit: 2 */
#define LCDBLKDIV0                  (0x0020)          /* LCD_C Clock divider for blinking
```

```
frequency Bit: 0 */
#define LCDBLKDIV1              (0x0040)        /* LCD_C Clock divider for blinking
frequency Bit: 1 */
#define LCDBLKDIV2              (0x0080)        /* LCD_C Clock divider for blinking
frequency Bit: 2 */

// LCDCBLKCTL
#define LCDBLKMOD0_L             (0x0001)        /* LCD_C Blinking mode Bit: 0 */
#define LCDBLKMOD1_L             (0x0002)        /* LCD_C Blinking mode Bit: 1 */
#define LCDBLKPRE0_L            (0x0004)        /* LCD_C Clock pre-scaler for blinking
frequency Bit: 0 */
#define LCDBLKPRE1_L            (0x0008)        /* LCD_C Clock pre-scaler for blinking
frequency Bit: 1 */
#define LCDBLKPRE2_L            (0x0010)        /* LCD_C Clock pre-scaler for blinking
frequency Bit: 2 */
#define LCDBLKDIV0_L            (0x0020)        /* LCD_C Clock divider for blinking
frequency Bit: 0 */
#define LCDBLKDIV1_L            (0x0040)        /* LCD_C Clock divider for blinking
frequency Bit: 1 */
#define LCDBLKDIV2_L            (0x0080)        /* LCD_C Clock divider for blinking
frequency Bit: 2 */

#define LCDBLKMOD_0              (0x0000)        /* LCD_C Blinking mode: Off */
#define LCDBLKMOD_1              (0x0001)        /* LCD_C Blinking mode: Individual */
#define LCDBLKMOD_2              (0x0002)        /* LCD_C Blinking mode: All */
#define LCDBLKMOD_3              (0x0003)        /* LCD_C Blinking mode: Switching */

#define LCDBLKPRE_0             (0x0000)        /* LCD_C Clock pre-scaler for blinking
frequency: 0 */
#define LCDBLKPRE_1             (0x0004)        /* LCD_C Clock pre-scaler for blinking
frequency: 1 */
#define LCDBLKPRE_2             (0x0008)        /* LCD_C Clock pre-scaler for blinking
frequency: 2 */
#define LCDBLKPRE_3             (0x000C)        /* LCD_C Clock pre-scaler for blinking
frequency: 3 */
#define LCDBLKPRE_4             (0x0010)        /* LCD_C Clock pre-scaler for blinking
frequency: 4 */
#define LCDBLKPRE_5             (0x0014)        /* LCD_C Clock pre-scaler for blinking
frequency: 5 */
#define LCDBLKPRE_6             (0x0018)        /* LCD_C Clock pre-scaler for blinking
frequency: 6 */
#define LCDBLKPRE_7             (0x001C)        /* LCD_C Clock pre-scaler for blinking
frequency: 7 */

#define LCDBLKPRE__512          (0x0000)        /* LCD_C Clock pre-scaler for blinking
frequency: 512      */
#define LCDBLKPRE__1024         (0x0004)        /* LCD_C Clock pre-scaler for blinking
frequency: 1024     */
```

```
#define LCDBLKPRE__2048          (0x0008)        /* LCD_C Clock pre-scaler for blinking
frequency: 2048    */
#define LCDBLKPRE__4096          (0x000C)        /* LCD_C Clock pre-scaler for blinking
frequency: 4096    */
#define LCDBLKPRE__8192          (0x0010)        /* LCD_C Clock pre-scaler for blinking
frequency: 8192    */
#define LCDBLKPRE__16384         (0x0014)        /* LCD_C Clock pre-scaler for blinking
frequency: 16384 */
#define LCDBLKPRE__32768         (0x0018)        /* LCD_C Clock pre-scaler for blinking
frequency: 32768 */
#define LCDBLKPRE__65536         (0x001C)        /* LCD_C Clock pre-scaler for blinking
frequency: 65536 */

#define LCDBLKDIV_0              (0x0000)        /* LCD_C Clock divider for blinking
frequency: 0 */
#define LCDBLKDIV_1              (0x0020)        /* LCD_C Clock divider for blinking
frequency: 1 */
#define LCDBLKDIV_2              (0x0040)        /* LCD_C Clock divider for blinking
frequency: 2 */
#define LCDBLKDIV_3              (0x0060)        /* LCD_C Clock divider for blinking
frequency: 3 */
#define LCDBLKDIV_4              (0x0080)        /* LCD_C Clock divider for blinking
frequency: 4 */
#define LCDBLKDIV_5              (0x00A0)        /* LCD_C Clock divider for blinking
frequency: 5 */
#define LCDBLKDIV_6              (0x00C0)        /* LCD_C Clock divider for blinking
frequency: 6 */
#define LCDBLKDIV_7              (0x00E0)        /* LCD_C Clock divider for blinking
frequency: 7 */

#define LCDBLKDIV__1             (0x0000)        /* LCD_C Clock divider for blinking
frequency: /1 */
#define LCDBLKDIV__2             (0x0020)        /* LCD_C Clock divider for blinking
frequency: /2 */
#define LCDBLKDIV__3             (0x0040)        /* LCD_C Clock divider for blinking
frequency: /3 */
#define LCDBLKDIV__4             (0x0060)        /* LCD_C Clock divider for blinking
frequency: /4 */
#define LCDBLKDIV__5             (0x0080)        /* LCD_C Clock divider for blinking
frequency: /5 */
#define LCDBLKDIV__6             (0x00A0)        /* LCD_C Clock divider for blinking
frequency: /6 */
#define LCDBLKDIV__7             (0x00C0)        /* LCD_C Clock divider for blinking
frequency: /7 */
#define LCDBLKDIV__8             (0x00E0)        /* LCD_C Clock divider for blinking
frequency: /8 */

// LCDCMEMCTL
```

```
#define LCDDISP                 (0x0001)        /* LCD_C LCD memory registers for display
*/
#define LCDCLRM                 (0x0002)        /* LCD_C Clear LCD memory */
#define LCDCLRBM                (0x0004)        /* LCD_C Clear LCD blinking memory */

// LCDCMEMCTL
#define LCDDISP_L               (0x0001)        /* LCD_C LCD memory registers for display
*/
#define LCDCLRM_L               (0x0002)        /* LCD_C Clear LCD memory */
#define LCDCLRBM_L              (0x0004)        /* LCD_C Clear LCD blinking memory */

// LCDCVCTL
#define LCD2B                   (0x0001)        /* Selects 1/2 bias. */
#define VLCDREF0                (0x0002)        /* Selects reference voltage for regulated
charge pump: 0 */
#define VLCDREF1                (0x0004)        /* Selects reference voltage for regulated
charge pump: 1 */
#define LCDCPEN                 (0x0008)        /* LCD Voltage Charge Pump Enable. */
#define VLCDEXT                 (0x0010)        /* Select external source for VLCD. */
#define LCDEXTBIAS              (0x0020)        /* V2 - V4 voltage select. */
#define R03EXT                  (0x0040)        /* Selects external connections for LCD mid
voltages. */
#define LCDREXT                 (0x0080)        /* Selects external connection for lowest
LCD voltage. */
#define VLCD0                   (0x0200)        /* VLCD select: 0 */
#define VLCD1                   (0x0400)        /* VLCD select: 1 */
#define VLCD2                   (0x0800)        /* VLCD select: 2 */
#define VLCD3                   (0x1000)        /* VLCD select: 3 */
#define VLCD4                   (0x2000)        /* VLCD select: 4 */
#define VLCD5                   (0x4000)        /* VLCD select: 5 */

// LCDCVCTL
#define LCD2B_L                 (0x0001)        /* Selects 1/2 bias. */
#define VLCDREF0_L              (0x0002)        /* Selects reference voltage for regulated
charge pump: 0 */
#define VLCDREF1_L              (0x0004)        /* Selects reference voltage for regulated
charge pump: 1 */
#define LCDCPEN_L               (0x0008)        /* LCD Voltage Charge Pump Enable. */
#define VLCDEXT_L               (0x0010)        /* Select external source for VLCD. */
#define LCDEXTBIAS_L            (0x0020)        /* V2 - V4 voltage select. */
#define R03EXT_L                (0x0040)        /* Selects external connections for LCD mid
voltages. */
#define LCDREXT_L               (0x0080)        /* Selects external connection for lowest
LCD voltage. */

// LCDCVCTL
#define VLCD0_H                 (0x0002)        /* VLCD select: 0 */
#define VLCD1_H                 (0x0004)        /* VLCD select: 1 */
```

```
#define VLCD2_H                 (0x0008)        /* VLCD select: 2 */
#define VLCD3_H                 (0x0010)        /* VLCD select: 3 */
#define VLCD4_H                 (0x0020)        /* VLCD select: 4 */
#define VLCD5_H                 (0x0040)        /* VLCD select: 5 */

/* Reference voltage source select for the regulated charge pump */
#define VLCDREF_0               (0x0000)        /* Internal */
#define VLCDREF_1               (0x0002)        /* External */
#define VLCDREF_2               (0x0004)        /* Reserved */
#define VLCDREF_3               (0x0006)        /* Reserved */

/* Charge pump voltage selections */
#define VLCD_0                  (0x0000)        /* Charge pump disabled */
#define VLCD_1                  (0x0200)        /* VLCD = 2.60V */
#define VLCD_2                  (0x0400)        /* VLCD = 2.66V */
#define VLCD_3                  (0x0600)        /* VLCD = 2.72V */
#define VLCD_4                  (0x0800)        /* VLCD = 2.78V */
#define VLCD_5                  (0x0A00)        /* VLCD = 2.84V */
#define VLCD_6                  (0x0C00)        /* VLCD = 2.90V */
#define VLCD_7                  (0x0E00)        /* VLCD = 2.96V */
#define VLCD_8                  (0x1000)        /* VLCD = 3.02V */
#define VLCD_9                  (0x1200)        /* VLCD = 3.08V */
#define VLCD_10                 (0x1400)        /* VLCD = 3.14V */
#define VLCD_11                 (0x1600)        /* VLCD = 3.20V */
#define VLCD_12                 (0x1800)        /* VLCD = 3.26V */
#define VLCD_13                 (0x1A00)        /* VLCD = 3.32V */
#define VLCD_14                 (0x1C00)        /* VLCD = 3.38V */
#define VLCD_15                 (0x1E00)        /* VLCD = 3.44V */

#define VLCD_DISABLED           (0x0000)        /* Charge pump disabled */
#define VLCD_2_60               (0x0200)        /* VLCD = 2.60V */
#define VLCD_2_66               (0x0400)        /* VLCD = 2.66V */
#define VLCD_2_72               (0x0600)        /* VLCD = 2.72V */
#define VLCD_2_78               (0x0800)        /* VLCD = 2.78V */
#define VLCD_2_84               (0x0A00)        /* VLCD = 2.84V */
#define VLCD_2_90               (0x0C00)        /* VLCD = 2.90V */
#define VLCD_2_96               (0x0E00)        /* VLCD = 2.96V */
#define VLCD_3_02               (0x1000)        /* VLCD = 3.02V */
#define VLCD_3_08               (0x1200)        /* VLCD = 3.08V */
#define VLCD_3_14               (0x1400)        /* VLCD = 3.14V */
#define VLCD_3_20               (0x1600)        /* VLCD = 3.20V */
#define VLCD_3_26               (0x1800)        /* VLCD = 3.26V */
#define VLCD_3_32               (0x1A00)        /* VLCD = 3.32V */
#define VLCD_3_38               (0x1C00)        /* VLCD = 3.38V */
#define VLCD_3_44               (0x1E00)        /* VLCD = 3.44V */

// LCDCPCTL0
#define LCDS0                   (0x0001)        /* LCD Segment    0 enable. */
```

```
#define LCDS1                   (0x0002)        /* LCD Segment  1 enable. */
#define LCDS2                   (0x0004)        /* LCD Segment  2 enable. */
#define LCDS3                   (0x0008)        /* LCD Segment  3 enable. */
#define LCDS4                   (0x0010)        /* LCD Segment  4 enable. */
#define LCDS5                   (0x0020)        /* LCD Segment  5 enable. */
#define LCDS6                   (0x0040)        /* LCD Segment  6 enable. */
#define LCDS7                   (0x0080)        /* LCD Segment  7 enable. */
#define LCDS8                   (0x0100)        /* LCD Segment  8 enable. */
#define LCDS9                   (0x0200)        /* LCD Segment  9 enable. */
#define LCDS10                  (0x0400)        /* LCD Segment 10 enable. */
#define LCDS11                  (0x0800)        /* LCD Segment 11 enable. */
#define LCDS12                  (0x1000)        /* LCD Segment 12 enable. */
#define LCDS13                  (0x2000)        /* LCD Segment 13 enable. */
#define LCDS14                  (0x4000)        /* LCD Segment 14 enable. */
#define LCDS15                  (0x8000)        /* LCD Segment 15 enable. */

// LCDCPCTL0
#define LCDS0_L                 (0x0001)        /* LCD Segment  0 enable. */
#define LCDS1_L                 (0x0002)        /* LCD Segment  1 enable. */
#define LCDS2_L                 (0x0004)        /* LCD Segment  2 enable. */
#define LCDS3_L                 (0x0008)        /* LCD Segment  3 enable. */
#define LCDS4_L                 (0x0010)        /* LCD Segment  4 enable. */
#define LCDS5_L                 (0x0020)        /* LCD Segment  5 enable. */
#define LCDS6_L                 (0x0040)        /* LCD Segment  6 enable. */
#define LCDS7_L                 (0x0080)        /* LCD Segment  7 enable. */

// LCDCPCTL0
#define LCDS8_H                 (0x0001)        /* LCD Segment  8 enable. */
#define LCDS9_H                 (0x0002)        /* LCD Segment  9 enable. */
#define LCDS10_H                (0x0004)        /* LCD Segment 10 enable. */
#define LCDS11_H                (0x0008)        /* LCD Segment 11 enable. */
#define LCDS12_H                (0x0010)        /* LCD Segment 12 enable. */
#define LCDS13_H                (0x0020)        /* LCD Segment 13 enable. */
#define LCDS14_H                (0x0040)        /* LCD Segment 14 enable. */
#define LCDS15_H                (0x0080)        /* LCD Segment 15 enable. */

// LCDCPCTL1
#define LCDS16                  (0x0001)        /* LCD Segment 16 enable. */
#define LCDS17                  (0x0002)        /* LCD Segment 17 enable. */
#define LCDS18                  (0x0004)        /* LCD Segment 18 enable. */
#define LCDS19                  (0x0008)        /* LCD Segment 19 enable. */
#define LCDS20                  (0x0010)        /* LCD Segment 20 enable. */
#define LCDS21                  (0x0020)        /* LCD Segment 21 enable. */
#define LCDS22                  (0x0040)        /* LCD Segment 22 enable. */
#define LCDS23                  (0x0080)        /* LCD Segment 23 enable. */
#define LCDS24                  (0x0100)        /* LCD Segment 24 enable. */
#define LCDS25                  (0x0200)        /* LCD Segment 25 enable. */
#define LCDS26                  (0x0400)        /* LCD Segment 26 enable. */
```

```
#define LCDS27                  (0x0800)        /* LCD Segment 27 enable. */
#define LCDS28                  (0x1000)        /* LCD Segment 28 enable. */
#define LCDS29                  (0x2000)        /* LCD Segment 29 enable. */
#define LCDS30                  (0x4000)        /* LCD Segment 30 enable. */
#define LCDS31                  (0x8000)        /* LCD Segment 31 enable. */

// LCDCPCTL1
#define LCDS16_L                (0x0001)        /* LCD Segment 16 enable. */
#define LCDS17_L                (0x0002)        /* LCD Segment 17 enable. */
#define LCDS18_L                (0x0004)        /* LCD Segment 18 enable. */
#define LCDS19_L                (0x0008)        /* LCD Segment 19 enable. */
#define LCDS20_L                (0x0010)        /* LCD Segment 20 enable. */
#define LCDS21_L                (0x0020)        /* LCD Segment 21 enable. */
#define LCDS22_L                (0x0040)        /* LCD Segment 22 enable. */
#define LCDS23_L                (0x0080)        /* LCD Segment 23 enable. */

// LCDCPCTL1
#define LCDS24_H                (0x0001)        /* LCD Segment 24 enable. */
#define LCDS25_H                (0x0002)        /* LCD Segment 25 enable. */
#define LCDS26_H                (0x0004)        /* LCD Segment 26 enable. */
#define LCDS27_H                (0x0008)        /* LCD Segment 27 enable. */
#define LCDS28_H                (0x0010)        /* LCD Segment 28 enable. */
#define LCDS29_H                (0x0020)        /* LCD Segment 29 enable. */
#define LCDS30_H                (0x0040)        /* LCD Segment 30 enable. */
#define LCDS31_H                (0x0080)        /* LCD Segment 31 enable. */

// LCDCPCTL2
#define LCDS32                  (0x0001)        /* LCD Segment 32 enable. */
#define LCDS33                  (0x0002)        /* LCD Segment 33 enable. */
#define LCDS34                  (0x0004)        /* LCD Segment 34 enable. */
#define LCDS35                  (0x0008)        /* LCD Segment 35 enable. */
#define LCDS36                  (0x0010)        /* LCD Segment 36 enable. */
#define LCDS37                  (0x0020)        /* LCD Segment 37 enable. */
#define LCDS38                  (0x0040)        /* LCD Segment 38 enable. */
#define LCDS39                  (0x0080)        /* LCD Segment 39 enable. */
#define LCDS40                  (0x0100)        /* LCD Segment 40 enable. */
#define LCDS41                  (0x0200)        /* LCD Segment 41 enable. */
#define LCDS42                  (0x0400)        /* LCD Segment 42 enable. */
#define LCDS43                  (0x0800)        /* LCD Segment 43 enable. */
#define LCDS44                  (0x1000)        /* LCD Segment 44 enable. */
#define LCDS45                  (0x2000)        /* LCD Segment 45 enable. */
#define LCDS46                  (0x4000)        /* LCD Segment 46 enable. */
#define LCDS47                  (0x8000)        /* LCD Segment 47 enable. */

// LCDCPCTL2
#define LCDS32_L                (0x0001)        /* LCD Segment 32 enable. */
#define LCDS33_L                (0x0002)        /* LCD Segment 33 enable. */
#define LCDS34_L                (0x0004)        /* LCD Segment 34 enable. */
```

```
#define LCDS35_L                (0x0008)        /* LCD Segment 35 enable. */
#define LCDS36_L                (0x0010)        /* LCD Segment 36 enable. */
#define LCDS37_L                (0x0020)        /* LCD Segment 37 enable. */
#define LCDS38_L                (0x0040)        /* LCD Segment 38 enable. */
#define LCDS39_L                (0x0080)        /* LCD Segment 39 enable. */

// LCDCPCTL2
#define LCDS40_H                (0x0001)        /* LCD Segment 40 enable. */
#define LCDS41_H                (0x0002)        /* LCD Segment 41 enable. */
#define LCDS42_H                (0x0004)        /* LCD Segment 42 enable. */
#define LCDS43_H                (0x0008)        /* LCD Segment 43 enable. */
#define LCDS44_H                (0x0010)        /* LCD Segment 44 enable. */
#define LCDS45_H                (0x0020)        /* LCD Segment 45 enable. */
#define LCDS46_H                (0x0040)        /* LCD Segment 46 enable. */
#define LCDS47_H                (0x0080)        /* LCD Segment 47 enable. */

// LCDCCPCTL
#define LCDCPDIS0               (0x0001)        /* LCD charge pump disable */
#define LCDCPDIS1               (0x0002)        /* LCD charge pump disable */
#define LCDCPDIS2               (0x0004)        /* LCD charge pump disable */
#define LCDCPDIS3               (0x0008)        /* LCD charge pump disable */
#define LCDCPDIS4               (0x0010)        /* LCD charge pump disable */
#define LCDCPDIS5               (0x0020)        /* LCD charge pump disable */
#define LCDCPDIS6               (0x0040)        /* LCD charge pump disable */
#define LCDCPDIS7               (0x0080)        /* LCD charge pump disable */
#define LCDCPCLKSYNC            (0x8000)        /* LCD charge pump clock synchronization
*/

// LCDCCPCTL
#define LCDCPDIS0_L             (0x0001)        /* LCD charge pump disable */
#define LCDCPDIS1_L             (0x0002)        /* LCD charge pump disable */
#define LCDCPDIS2_L             (0x0004)        /* LCD charge pump disable */
#define LCDCPDIS3_L             (0x0008)        /* LCD charge pump disable */
#define LCDCPDIS4_L             (0x0010)        /* LCD charge pump disable */
#define LCDCPDIS5_L             (0x0020)        /* LCD charge pump disable */
#define LCDCPDIS6_L             (0x0040)        /* LCD charge pump disable */
#define LCDCPDIS7_L             (0x0080)        /* LCD charge pump disable */

// LCDCCPCTL
#define LCDCPCLKSYNC_H          (0x0080)        /* LCD charge pump clock synchronization
*/

SFR_8BIT(LCDM1);                                /* LCD Memory 1 */
#define LCDMEM_                 LCDM1           /* LCD Memory */
#ifdef __ASM_HEADER__
#define LCDMEM                  LCDM1           /* LCD Memory (for assembler) */
#else
#define LCDMEM                  ((char*)        &LCDM1) /* LCD Memory (for C) */
```

```
#endif
SFR_8BIT(LCDM2);                        /* LCD Memory 2 */
SFR_8BIT(LCDM3);                        /* LCD Memory 3 */
SFR_8BIT(LCDM4);                        /* LCD Memory 4 */
SFR_8BIT(LCDM5);                        /* LCD Memory 5 */
SFR_8BIT(LCDM6);                        /* LCD Memory 6 */
SFR_8BIT(LCDM7);                        /* LCD Memory 7 */
SFR_8BIT(LCDM8);                        /* LCD Memory 8 */
SFR_8BIT(LCDM9);                        /* LCD Memory 9 */
SFR_8BIT(LCDM10);                       /* LCD Memory 10 */
SFR_8BIT(LCDM11);                       /* LCD Memory 11 */
SFR_8BIT(LCDM12);                       /* LCD Memory 12 */
SFR_8BIT(LCDM13);                       /* LCD Memory 13 */
SFR_8BIT(LCDM14);                       /* LCD Memory 14 */
SFR_8BIT(LCDM15);                       /* LCD Memory 15 */
SFR_8BIT(LCDM16);                       /* LCD Memory 16 */
SFR_8BIT(LCDM17);                       /* LCD Memory 17 */
SFR_8BIT(LCDM18);                       /* LCD Memory 18 */
SFR_8BIT(LCDM19);                       /* LCD Memory 19 */
SFR_8BIT(LCDM20);                       /* LCD Memory 20 */
SFR_8BIT(LCDM21);                       /* LCD Memory 21 */
SFR_8BIT(LCDM22);                       /* LCD Memory 22 */
SFR_8BIT(LCDM23);                       /* LCD Memory 23 */
SFR_8BIT(LCDM24);                       /* LCD Memory 24 */
SFR_8BIT(LCDM25);                       /* LCD Memory 25 */
SFR_8BIT(LCDM26);                       /* LCD Memory 26 */
SFR_8BIT(LCDM27);                       /* LCD Memory 27 */
SFR_8BIT(LCDM28);                       /* LCD Memory 28 */
SFR_8BIT(LCDM29);                       /* LCD Memory 29 */
SFR_8BIT(LCDM30);                       /* LCD Memory 30 */
SFR_8BIT(LCDM31);                       /* LCD Memory 31 */
SFR_8BIT(LCDM32);                       /* LCD Memory 32 */
SFR_8BIT(LCDM33);                       /* LCD Memory 33 */
SFR_8BIT(LCDM34);                       /* LCD Memory 34 */
SFR_8BIT(LCDM35);                       /* LCD Memory 35 */
SFR_8BIT(LCDM36);                       /* LCD Memory 36 */
SFR_8BIT(LCDM37);                       /* LCD Memory 37 */
SFR_8BIT(LCDM38);                       /* LCD Memory 38 */
SFR_8BIT(LCDM39);                       /* LCD Memory 39 */
SFR_8BIT(LCDM40);                       /* LCD Memory 40 */
SFR_8BIT(LCDM41);                       /* LCD Memory 41 */
SFR_8BIT(LCDM42);                       /* LCD Memory 42 */
SFR_8BIT(LCDM43);                       /* LCD Memory 43 */

SFR_8BIT(LCDBM1);                       /* LCD Blinking Memory 1 */
#define LCDBMEM_            LCDBM1        /* LCD Blinking Memory */
#ifdef __ASM_HEADER__
#define LCDBMEM             (LCDBM1)      /* LCD Blinking Memory (for assembler)
```

```
*/
#else
#define LCDBMEM                 ((char*)        &LCDBM1) /* LCD Blinking Memory (for C)
*/
#endif
SFR_8BIT(LCDBM2);                               /* LCD Blinking Memory 2 */
SFR_8BIT(LCDBM3);                               /* LCD Blinking Memory 3 */
SFR_8BIT(LCDBM4);                               /* LCD Blinking Memory 4 */
SFR_8BIT(LCDBM5);                               /* LCD Blinking Memory 5 */
SFR_8BIT(LCDBM6);                               /* LCD Blinking Memory 6 */
SFR_8BIT(LCDBM7);                               /* LCD Blinking Memory 7 */
SFR_8BIT(LCDBM8);                               /* LCD Blinking Memory 8 */
SFR_8BIT(LCDBM9);                               /* LCD Blinking Memory 9 */
SFR_8BIT(LCDBM10);                              /* LCD Blinking Memory 10 */
SFR_8BIT(LCDBM11);                              /* LCD Blinking Memory 11 */
SFR_8BIT(LCDBM12);                              /* LCD Blinking Memory 12 */
SFR_8BIT(LCDBM13);                              /* LCD Blinking Memory 13 */
SFR_8BIT(LCDBM14);                              /* LCD Blinking Memory 14 */
SFR_8BIT(LCDBM15);                              /* LCD Blinking Memory 15 */
SFR_8BIT(LCDBM16);                              /* LCD Blinking Memory 16 */
SFR_8BIT(LCDBM17);                              /* LCD Blinking Memory 17 */
SFR_8BIT(LCDBM18);                              /* LCD Blinking Memory 18 */
SFR_8BIT(LCDBM19);                              /* LCD Blinking Memory 19 */
SFR_8BIT(LCDBM20);                              /* LCD Blinking Memory 20 */
SFR_8BIT(LCDBM21);                              /* LCD Blinking Memory 21 */
SFR_8BIT(LCDBM22);                              /* LCD Blinking Memory 22 */

/* LCDCIV Definitions */
#define LCDCIV_NONE                 (0x0000)        /* No Interrupt pending */
#define LCDCIV_LCDNOCAPIFG          (0x0002)        /* No capacitor connected */
#define LCDCIV_LCDCLKOFFIFG         (0x0004)        /* Blink, segments off */
#define LCDCIV_LCDCLKONIFG          (0x0006)        /* Blink, segments on */
#define LCDCIV_LCDFRMIFG            (0x0008)        /* Frame interrupt */

/************************************************************
* Memory Protection Unit
************************************************************/

/************************************************************
* HARDWARE MULTIPLIER 32Bit
************************************************************/
/************************************************************
* PMM - Power Management System for FRAM
************************************************************/


/************************************************************
* DIGITAL I/O Port1/2 Pull up / Pull down Resistors
```

```
************************************************************/
#define __MSP430_HAS_PORT1_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT1_R__ 0x0200
#define P1_BASE                  __MSP430_BASEADDRESS_PORT1_R__
#define __MSP430_HAS_PORT2_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT2_R__ 0x0200
#define P2_BASE                  __MSP430_BASEADDRESS_PORT2_R__
#define __MSP430_HAS_PORTA_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORTA_R__ 0x0200
#define PA_BASE                  __MSP430_BASEADDRESS_PORTA_R__
#define __MSP430_HAS_P1SEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_P2SEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_PASEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_P1SEL1__                     /* Define for DriverLib */
#define __MSP430_HAS_P2SEL1__                     /* Define for DriverLib */
#define __MSP430_HAS_PASEL1__                     /* Define for DriverLib */

SFR_16BIT(PAIN);                                  /* Port A Input */
SFR_8BIT(PAIN_L);                                 /* Port A Input */
SFR_8BIT(PAIN_H);                                 /* Port A Input */
SFR_16BIT(PAOUT);                                 /* Port A Output */
SFR_8BIT(PAOUT_L);                                /* Port A Output */
SFR_8BIT(PAOUT_H);                                /* Port A Output */
SFR_16BIT(PADIR);                                 /* Port A Direction */
SFR_8BIT(PADIR_L);                                /* Port A Direction */
SFR_8BIT(PADIR_H);                                /* Port A Direction */
SFR_16BIT(PAREN);                                 /* Port A Resistor Enable */
SFR_8BIT(PAREN_L);                                /* Port A Resistor Enable */
SFR_8BIT(PAREN_H);                                /* Port A Resistor Enable */
SFR_16BIT(PASEL0);                                /* Port A Selection 0 */
SFR_8BIT(PASEL0_L);                               /* Port A Selection 0 */
SFR_8BIT(PASEL0_H);                               /* Port A Selection 0 */
SFR_16BIT(PASEL1);                                /* Port A Selection 1 */
SFR_8BIT(PASEL1_L);                               /* Port A Selection 1 */
SFR_8BIT(PASEL1_H);                               /* Port A Selection 1 */
SFR_16BIT(PASELC);                                /* Port A Complement Selection */
SFR_8BIT(PASELC_L);                               /* Port A Complement Selection */
SFR_8BIT(PASELC_H);                               /* Port A Complement Selection */
SFR_16BIT(PAIES);                                 /* Port A Interrupt Edge Select */
SFR_8BIT(PAIES_L);                                /* Port A Interrupt Edge Select */
SFR_8BIT(PAIES_H);                                /* Port A Interrupt Edge Select */
SFR_16BIT(PAIE);                                  /* Port A Interrupt Enable */
SFR_8BIT(PAIE_L);                                 /* Port A Interrupt Enable */
SFR_8BIT(PAIE_H);                                 /* Port A Interrupt Enable */
SFR_16BIT(PAIFG);                                 /* Port A Interrupt Flag */
```

```
SFR_8BIT(PAIFG_L);                          /* Port A Interrupt Flag */
SFR_8BIT(PAIFG_H);                          /* Port A Interrupt Flag */


SFR_16BIT(P1IV);                            /* Port 1 Interrupt Vector Word */
SFR_16BIT(P2IV);                            /* Port 2 Interrupt Vector Word */
#define P1IN                    (PAIN_L)          /* Port 1 Input */
#define P1OUT                   (PAOUT_L)         /* Port 1 Output */
#define P1DIR                   (PADIR_L)         /* Port 1 Direction */
#define P1REN                   (PAREN_L)         /* Port 1 Resistor Enable */
#define P1SEL0                  (PASEL0_L)        /* Port 1 Selection 0 */
#define P1SEL1                  (PASEL1_L)        /* Port 1 Selection 1 */
#define P1SELC                  (PASELC_L)        /* Port 1 Complement Selection */
#define P1IES                   (PAIES_L)         /* Port 1 Interrupt Edge Select */
#define P1IE                    (PAIE_L)          /* Port 1 Interrupt Enable */
#define P1IFG                   (PAIFG_L)         /* Port 1 Interrupt Flag */

//Definitions for P1IV
#define P1IV_NONE               (0x0000)          /* No Interrupt pending */
#define P1IV_P1IFG0             (0x0002)          /* P1IV P1IFG.0 */
#define P1IV_P1IFG1             (0x0004)          /* P1IV P1IFG.1 */
#define P1IV_P1IFG2             (0x0006)          /* P1IV P1IFG.2 */
#define P1IV_P1IFG3             (0x0008)          /* P1IV P1IFG.3 */
#define P1IV_P1IFG4             (0x000A)          /* P1IV P1IFG.4 */
#define P1IV_P1IFG5             (0x000C)          /* P1IV P1IFG.5 */
#define P1IV_P1IFG6             (0x000E)          /* P1IV P1IFG.6 */
#define P1IV_P1IFG7             (0x0010)          /* P1IV P1IFG.7 */

#define P2IN                    (PAIN_H)          /* Port 2 Input */
#define P2OUT                   (PAOUT_H)         /* Port 2 Output */
#define P2DIR                   (PADIR_H)         /* Port 2 Direction */
#define P2REN                   (PAREN_H)         /* Port 2 Resistor Enable */
#define P2SEL0                  (PASEL0_H)        /* Port 2 Selection 0 */
#define P2SEL1                  (PASEL1_H)        /* Port 2 Selection 1 */
#define P2SELC                  (PASELC_H)        /* Port 2 Complement Selection */
#define P2IES                   (PAIES_H)         /* Port 2 Interrupt Edge Select */
#define P2IE                    (PAIE_H)          /* Port 2 Interrupt Enable */
#define P2IFG                   (PAIFG_H)         /* Port 2 Interrupt Flag */

//Definitions for P2IV
#define P2IV_NONE               (0x0000)          /* No Interrupt pending */
#define P2IV_P2IFG0             (0x0002)          /* P2IV P2IFG.0 */
#define P2IV_P2IFG1             (0x0004)          /* P2IV P2IFG.1 */
#define P2IV_P2IFG2             (0x0006)          /* P2IV P2IFG.2 */
#define P2IV_P2IFG3             (0x0008)          /* P2IV P2IFG.3 */
#define P2IV_P2IFG4             (0x000A)          /* P2IV P2IFG.4 */
#define P2IV_P2IFG5             (0x000C)          /* P2IV P2IFG.5 */
#define P2IV_P2IFG6             (0x000E)          /* P2IV P2IFG.6 */
```

```
#define P2IV_P2IFG7                 (0x0010)          /* P2IV P2IFG.7 */


/************************************************************
* DIGITAL I/O Port3/4 Pull up / Pull down Resistors
************************************************************/
#define __MSP430_HAS_PORT3_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT3_R__ 0x0220
#define P3_BASE                    __MSP430_BASEADDRESS_PORT3_R__
#define __MSP430_HAS_PORT4_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT4_R__ 0x0220
#define P4_BASE                    __MSP430_BASEADDRESS_PORT4_R__
#define __MSP430_HAS_PORTB_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORTB_R__ 0x0220
#define PB_BASE                    __MSP430_BASEADDRESS_PORTB_R__
#define __MSP430_HAS_P3SEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_P4SEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_PBSEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_P3SEL1__                     /* Define for DriverLib */
#define __MSP430_HAS_P4SEL1__                     /* Define for DriverLib */
#define __MSP430_HAS_PBSEL1__                     /* Define for DriverLib */

SFR_16BIT(PBIN);                                  /* Port B Input */
SFR_8BIT(PBIN_L);                                 /* Port B Input */
SFR_8BIT(PBIN_H);                                 /* Port B Input */
SFR_16BIT(PBOUT);                                 /* Port B Output */
SFR_8BIT(PBOUT_L);                                /* Port B Output */
SFR_8BIT(PBOUT_H);                                /* Port B Output */
SFR_16BIT(PBDIR);                                 /* Port B Direction */
SFR_8BIT(PBDIR_L);                                /* Port B Direction */
SFR_8BIT(PBDIR_H);                                /* Port B Direction */
SFR_16BIT(PBREN);                                 /* Port B Resistor Enable */
SFR_8BIT(PBREN_L);                                /* Port B Resistor Enable */
SFR_8BIT(PBREN_H);                                /* Port B Resistor Enable */
SFR_16BIT(PBSEL0);                                /* Port B Selection 0 */
SFR_8BIT(PBSEL0_L);                               /* Port B Selection 0 */
SFR_8BIT(PBSEL0_H);                               /* Port B Selection 0 */
SFR_16BIT(PBSEL1);                                /* Port B Selection 1 */
SFR_8BIT(PBSEL1_L);                               /* Port B Selection 1 */
SFR_8BIT(PBSEL1_H);                               /* Port B Selection 1 */
SFR_16BIT(PBSELC);                                /* Port B Complement Selection */
SFR_8BIT(PBSELC_L);                               /* Port B Complement Selection */
SFR_8BIT(PBSELC_H);                               /* Port B Complement Selection */
SFR_16BIT(PBIES);                                 /* Port B Interrupt Edge Select */
SFR_8BIT(PBIES_L);                                /* Port B Interrupt Edge Select */
```

```
SFR_8BIT(PBIES_H);                      /* Port B Interrupt Edge Select */
SFR_16BIT(PBIE);                        /* Port B Interrupt Enable */
SFR_8BIT(PBIE_L);                       /* Port B Interrupt Enable */
SFR_8BIT(PBIE_H);                       /* Port B Interrupt Enable */
SFR_16BIT(PBIFG);                       /* Port B Interrupt Flag */
SFR_8BIT(PBIFG_L);                      /* Port B Interrupt Flag */
SFR_8BIT(PBIFG_H);                      /* Port B Interrupt Flag */


SFR_16BIT(P3IV);                        /* Port 3 Interrupt Vector Word */
SFR_16BIT(P4IV);                        /* Port 4 Interrupt Vector Word */
#define P3IN                (PBIN_L)        /* Port 3 Input */
#define P3OUT               (PBOUT_L)       /* Port 3 Output */
#define P3DIR               (PBDIR_L)       /* Port 3 Direction */
#define P3REN               (PBREN_L)       /* Port 3 Resistor Enable */
#define P3SEL0              (PBSEL0_L)      /* Port 3 Selection 0 */
#define P3SEL1              (PBSEL1_L)      /* Port 3 Selection 1 */
#define P3SELC              (PBSELC_L)      /* Port 3 Complement Selection */
#define P3IES               (PBIES_L)       /* Port 3 Interrupt Edge Select */
#define P3IE                (PBIE_L)        /* Port 3 Interrupt Enable */
#define P3IFG               (PBIFG_L)       /* Port 3 Interrupt Flag */

//Definitions for P3IV
#define P3IV_NONE           (0x0000)        /* No Interrupt pending */
#define P3IV_P3IFG0         (0x0002)        /* P3IV P3IFG.0 */
#define P3IV_P3IFG1         (0x0004)        /* P3IV P3IFG.1 */
#define P3IV_P3IFG2         (0x0006)        /* P3IV P3IFG.2 */
#define P3IV_P3IFG3         (0x0008)        /* P3IV P3IFG.3 */
#define P3IV_P3IFG4         (0x000A)        /* P3IV P3IFG.4 */
#define P3IV_P3IFG5         (0x000C)        /* P3IV P3IFG.5 */
#define P3IV_P3IFG6         (0x000E)        /* P3IV P3IFG.6 */
#define P3IV_P3IFG7         (0x0010)        /* P3IV P3IFG.7 */

#define P4IN                (PBIN_H)        /* Port 4 Input */
#define P4OUT               (PBOUT_H)       /* Port 4 Output */
#define P4DIR               (PBDIR_H)       /* Port 4 Direction */
#define P4REN               (PBREN_H)       /* Port 4 Resistor Enable */
#define P4SEL0              (PBSEL0_H)      /* Port 4 Selection 0 */
#define P4SEL1              (PBSEL1_H)      /* Port 4 Selection 1 */
#define P4SELC              (PBSELC_H)      /* Port 4 Complement Selection */
#define P4IES               (PBIES_H)       /* Port 4 Interrupt Edge Select */
#define P4IE                (PBIE_H)        /* Port 4 Interrupt Enable */
#define P4IFG               (PBIFG_H)       /* Port 4 Interrupt Flag */

//Definitions for P4IV
#define P4IV_NONE           (0x0000)        /* No Interrupt pending */
#define P4IV_P4IFG0         (0x0002)        /* P4IV P4IFG.0 */
#define P4IV_P4IFG1         (0x0004)        /* P4IV P4IFG.1 */
```

```
#define P4IV_P4IFG2                (0x0006)          /* P4IV P4IFG.2 */
#define P4IV_P4IFG3                (0x0008)          /* P4IV P4IFG.3 */
#define P4IV_P4IFG4                (0x000A)          /* P4IV P4IFG.4 */
#define P4IV_P4IFG5                (0x000C)          /* P4IV P4IFG.5 */
#define P4IV_P4IFG6                (0x000E)          /* P4IV P4IFG.6 */
#define P4IV_P4IFG7                (0x0010)          /* P4IV P4IFG.7 */


/************************************************************
* DIGITAL I/O Port5/6 Pull up / Pull down Resistors
************************************************************/
#define __MSP430_HAS_PORT5_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT5_R__ 0x0240
#define P5_BASE                    __MSP430_BASEADDRESS_PORT5_R__
#define __MSP430_HAS_PORT6_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT6_R__ 0x0240
#define P6_BASE                    __MSP430_BASEADDRESS_PORT6_R__
#define __MSP430_HAS_PORTC_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORTC_R__ 0x0240
#define PC_BASE                    __MSP430_BASEADDRESS_PORTC_R__
#define __MSP430_HAS_P5SEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_P6SEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_PCSEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_P5SEL1__                     /* Define for DriverLib */
#define __MSP430_HAS_P6SEL1__                     /* Define for DriverLib */
#define __MSP430_HAS_PCSEL1__                     /* Define for DriverLib */

SFR_16BIT(PCIN);                                  /* Port C Input */
SFR_8BIT(PCIN_L);                                 /* Port C Input */
SFR_8BIT(PCIN_H);                                 /* Port C Input */
SFR_16BIT(PCOUT);                                 /* Port C Output */
SFR_8BIT(PCOUT_L);                                /* Port C Output */
SFR_8BIT(PCOUT_H);                                /* Port C Output */
SFR_16BIT(PCDIR);                                 /* Port C Direction */
SFR_8BIT(PCDIR_L);                                /* Port C Direction */
SFR_8BIT(PCDIR_H);                                /* Port C Direction */
SFR_16BIT(PCREN);                                 /* Port C Resistor Enable */
SFR_8BIT(PCREN_L);                                /* Port C Resistor Enable */
SFR_8BIT(PCREN_H);                                /* Port C Resistor Enable */
SFR_16BIT(PCSEL0);                                /* Port C Selection 0 */
SFR_8BIT(PCSEL0_L);                               /* Port C Selection 0 */
SFR_8BIT(PCSEL0_H);                               /* Port C Selection 0 */
SFR_16BIT(PCSEL1);                                /* Port C Selection 1 */
SFR_8BIT(PCSEL1_L);                               /* Port C Selection 1 */
SFR_8BIT(PCSEL1_H);                               /* Port C Selection 1 */
```

```
SFR_16BIT(PCSELC);                      /* Port C Complement Selection */
SFR_8BIT(PCSELC_L);                     /* Port C Complement Selection */
SFR_8BIT(PCSELC_H);                     /* Port C Complement Selection */


#define P5IN                    (PCIN_L)        /* Port 5 Input */
#define P5OUT                   (PCOUT_L)       /* Port 5 Output */
#define P5DIR                   (PCDIR_L)       /* Port 5 Direction */
#define P5REN                   (PCREN_L)       /* Port 5 Resistor Enable */
#define P5SEL0                  (PCSEL0_L)      /* Port 5 Selection 0 */
#define P5SEL1                  (PCSEL1_L)      /* Port 5 Selection 1 */
#define P5SELC                  (PCSELC_L)      /* Port 5 Complement Selection */

#define P6IN                    (PCIN_H)        /* Port 6 Input */
#define P6OUT                   (PCOUT_H)       /* Port 6 Output */
#define P6DIR                   (PCDIR_H)       /* Port 6 Direction */
#define P6REN                   (PCREN_H)       /* Port 6 Resistor Enable */
#define P6SEL0                  (PCSEL0_H)      /* Port 6 Selection 0 */
#define P6SEL1                  (PCSEL1_H)      /* Port 6 Selection 1 */
#define P6SELC                  (PCSELC_H)      /* Port 6 Complement Selection */


/************************************************************
* DIGITAL I/O Port7/8 Pull up / Pull down Resistors
************************************************************/
#define __MSP430_HAS_PORT7_R__                  /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT7_R__ 0x0260
#define P7_BASE                 __MSP430_BASEADDRESS_PORT7_R__
#define __MSP430_HAS_PORT8_R__                  /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT8_R__ 0x0260
#define P8_BASE                 __MSP430_BASEADDRESS_PORT8_R__
#define __MSP430_HAS_PORTD_R__                  /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORTD_R__ 0x0260
#define PD_BASE                 __MSP430_BASEADDRESS_PORTD_R__
#define __MSP430_HAS_P7SEL0__                   /* Define for DriverLib */
#define __MSP430_HAS_P8SEL0__                   /* Define for DriverLib */
#define __MSP430_HAS_PDSEL0__                   /* Define for DriverLib */
#define __MSP430_HAS_P7SEL1__                   /* Define for DriverLib */
#define __MSP430_HAS_P8SEL1__                   /* Define for DriverLib */
#define __MSP430_HAS_PDSEL1__                   /* Define for DriverLib */

SFR_16BIT(PDIN);                        /* Port D Input */
SFR_8BIT(PDIN_L);                       /* Port D Input */
SFR_8BIT(PDIN_H);                       /* Port D Input */
SFR_16BIT(PDOUT);                       /* Port D Output */
```

```
SFR_8BIT(PDOUT_L);                      /* Port D Output */
SFR_8BIT(PDOUT_H);                      /* Port D Output */
SFR_16BIT(PDDIR);                       /* Port D Direction */
SFR_8BIT(PDDIR_L);                      /* Port D Direction */
SFR_8BIT(PDDIR_H);                      /* Port D Direction */
SFR_16BIT(PDREN);                       /* Port D Resistor Enable */
SFR_8BIT(PDREN_L);                      /* Port D Resistor Enable */
SFR_8BIT(PDREN_H);                      /* Port D Resistor Enable */
SFR_16BIT(PDSEL0);                      /* Port D Selection 0 */
SFR_8BIT(PDSEL0_L);                     /* Port D Selection 0 */
SFR_8BIT(PDSEL0_H);                     /* Port D Selection 0 */
SFR_16BIT(PDSEL1);                      /* Port D Selection 1 */
SFR_8BIT(PDSEL1_L);                     /* Port D Selection 1 */
SFR_8BIT(PDSEL1_H);                     /* Port D Selection 1 */
SFR_16BIT(PDSELC);                      /* Port D Complement Selection */
SFR_8BIT(PDSELC_L);                     /* Port D Complement Selection */
SFR_8BIT(PDSELC_H);                     /* Port D Complement Selection */


#define P7IN                    (PDIN_L)        /* Port 7 Input */
#define P7OUT                   (PDOUT_L)       /* Port 7 Output */
#define P7DIR                   (PDDIR_L)       /* Port 7 Direction */
#define P7REN                   (PDREN_L)       /* Port 7 Resistor Enable */
#define P7SEL0                  (PDSEL0_L)      /* Port 7 Selection 0 */
#define P7SEL1                  (PDSEL1_L)      /* Port 7 Selection 1 */
#define P7SELC                  (PDSELC_L)      /* Port 7 Complement Selection */

#define P8IN                    (PDIN_H)        /* Port 8 Input */
#define P8OUT                   (PDOUT_H)       /* Port 8 Output */
#define P8DIR                   (PDDIR_H)       /* Port 8 Direction */
#define P8REN                   (PDREN_H)       /* Port 8 Resistor Enable */
#define P8SEL0                  (PDSEL0_H)      /* Port 8 Selection 0 */
#define P8SEL1                  (PDSEL1_H)      /* Port 8 Selection 1 */
#define P8SELC                  (PDSELC_H)      /* Port 8 Complement Selection */


/************************************************************
* DIGITAL I/O Port9/10 Pull up / Pull down Resistors
************************************************************/
#define __MSP430_HAS_PORT9_R__                  /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT9_R__ 0x0280
#define P9_BASE                 __MSP430_BASEADDRESS_PORT9_R__
#define __MSP430_HAS_PORT10_R__                 /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORT10_R__ 0x0280
#define P10_BASE                __MSP430_BASEADDRESS_PORT10_R__
#define __MSP430_HAS_PORTE_R__                  /* Definition to show that Module is
```

```
available */
#define __MSP430_BASEADDRESS_PORTE_R__ 0x0280
#define PE_BASE                        __MSP430_BASEADDRESS_PORTE_R__
#define __MSP430_HAS_P9SEL0__                    /* Define for DriverLib */
#define __MSP430_HAS_P10SEL0__                   /* Define for DriverLib */
#define __MSP430_HAS_PESEL0__                    /* Define for DriverLib */
#define __MSP430_HAS_P9SEL1__                    /* Define for DriverLib */
#define __MSP430_HAS_P10SEL1__                   /* Define for DriverLib */
#define __MSP430_HAS_PESEL1__                    /* Define for DriverLib */

SFR_16BIT(PEIN);                                 /* Port E Input */
SFR_8BIT(PEIN_L);                                /* Port E Input */
SFR_8BIT(PEIN_H);                                /* Port E Input */
SFR_16BIT(PEOUT);                                /* Port E Output */
SFR_8BIT(PEOUT_L);                               /* Port E Output */
SFR_8BIT(PEOUT_H);                               /* Port E Output */
SFR_16BIT(PEDIR);                                /* Port E Direction */
SFR_8BIT(PEDIR_L);                               /* Port E Direction */
SFR_8BIT(PEDIR_H);                               /* Port E Direction */
SFR_16BIT(PEREN);                                /* Port E Resistor Enable */
SFR_8BIT(PEREN_L);                               /* Port E Resistor Enable */
SFR_8BIT(PEREN_H);                               /* Port E Resistor Enable */
SFR_16BIT(PESEL0);                               /* Port E Selection 0 */
SFR_8BIT(PESEL0_L);                              /* Port E Selection 0 */
SFR_8BIT(PESEL0_H);                              /* Port E Selection 0 */
SFR_16BIT(PESEL1);                               /* Port E Selection 1 */
SFR_8BIT(PESEL1_L);                              /* Port E Selection 1 */
SFR_8BIT(PESEL1_H);                              /* Port E Selection 1 */
SFR_16BIT(PESELC);                               /* Port E Complement Selection */
SFR_8BIT(PESELC_L);                              /* Port E Complement Selection */
SFR_8BIT(PESELC_H);                              /* Port E Complement Selection */


#define P9IN                           (PEIN_L)          /* Port 9 Input */
#define P9OUT                           (PEOUT_L)        /* Port 9 Output */
#define P9DIR                          (PEDIR_L)         /* Port 9 Direction */
#define P9REN                           (PEREN_L)        /* Port 9 Resistor Enable */
#define P9SEL0                         (PESEL0_L)        /* Port 9 Selection 0 */
#define P9SEL1                         (PESEL1_L)        /* Port 9 Selection 1 */
#define P9SELC                         (PESELC_L)        /* Port 9 Complement Selection */

#define P10IN                          (PEIN_H)          /* Port 10 Input */
#define P10OUT                          (PEOUT_H)        /* Port 10 Output */
#define P10DIR                         (PEDIR_H)         /* Port 10 Direction */
#define P10REN                          (PEREN_H)        /* Port 10 Resistor Enable */
#define P10SEL0                        (PESEL0_H)        /* Port 10 Selection 0 */
#define P10SEL1                        (PESEL1_H)        /* Port 10 Selection 1 */
#define P10SELC                        (PESELC_H)        /* Port 10 Complement Selection */
```

```
/************************************************************
* DIGITAL I/O PortJ Pull up / Pull down Resistors
************************************************************/
#define __MSP430_HAS_PORTJ_R__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_PORTJ_R__ 0x0320
#define PJ_BASE                         __MSP430_BASEADDRESS_PORTJ_R__
#define __MSP430_HAS_PJSEL0__                     /* Define for DriverLib */
#define __MSP430_HAS_PJSEL1__                     /* Define for DriverLib */

SFR_16BIT(PJIN);                                  /* Port J Input */
SFR_8BIT(PJIN_L);                                 /* Port J Input */
SFR_8BIT(PJIN_H);                                 /* Port J Input */
SFR_16BIT(PJOUT);                                 /* Port J Output */
SFR_8BIT(PJOUT_L);                                /* Port J Output */
SFR_8BIT(PJOUT_H);                                /* Port J Output */
SFR_16BIT(PJDIR);                                 /* Port J Direction */
SFR_8BIT(PJDIR_L);                                /* Port J Direction */
SFR_8BIT(PJDIR_H);                                /* Port J Direction */
SFR_16BIT(PJREN);                                 /* Port J Resistor Enable */
SFR_8BIT(PJREN_L);                                /* Port J Resistor Enable */
SFR_8BIT(PJREN_H);                                /* Port J Resistor Enable */
SFR_16BIT(PJSEL0);                                /* Port J Selection 0 */
SFR_8BIT(PJSEL0_L);                               /* Port J Selection 0 */
SFR_8BIT(PJSEL0_H);                               /* Port J Selection 0 */
SFR_16BIT(PJSEL1);                                /* Port J Selection 1 */
SFR_8BIT(PJSEL1_L);                               /* Port J Selection 1 */
SFR_8BIT(PJSEL1_H);                               /* Port J Selection 1 */
SFR_16BIT(PJSELC);                                /* Port J Complement Selection */
SFR_8BIT(PJSELC_L);                               /* Port J Complement Selection */
SFR_8BIT(PJSELC_H);                               /* Port J Complement Selection */

/************************************************************
* RAM Control Module for FRAM
************************************************************/
/************************************************************
* Shared Reference
************************************************************/
/************************************************************
* Real Time Clock
************************************************************/
#define __MSP430_HAS_RTC_C__                      /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_RTC_C__ 0x04A0
#define RTC_C_BASE                      __MSP430_BASEADDRESS_RTC_C__
```

```
SFR_16BIT(RTCCTL0);                     /* Real Timer Clock Control 0/Key */
SFR_8BIT(RTCCTL0_L);                    /* Real Timer Clock Control 0/Key */
SFR_8BIT(RTCCTL0_H);                    /* Real Timer Clock Control 0/Key */
#define RTCPWD                  RTCCTL0_H
SFR_16BIT(RTCCTL13);                    /* Real Timer Clock Control 1/3 */
SFR_8BIT(RTCCTL13_L);                   /* Real Timer Clock Control 1/3 */
SFR_8BIT(RTCCTL13_H);                   /* Real Timer Clock Control 1/3 */
#define RTCCTL1                 RTCCTL13_L
#define RTCCTL3                 RTCCTL13_H
SFR_16BIT(RTCOCAL);                     /* Real Timer Clock Offset Calibartion */
SFR_8BIT(RTCOCAL_L);                    /* Real Timer Clock Offset Calibartion */
SFR_8BIT(RTCOCAL_H);                    /* Real Timer Clock Offset Calibartion */
SFR_16BIT(RTCTCMP);                     /* Real Timer Temperature
Compensation */
SFR_8BIT(RTCTCMP_L);                    /* Real Timer Temperature
Compensation */
SFR_8BIT(RTCTCMP_H);                    /* Real Timer Temperature
Compensation */
SFR_16BIT(RTCPS0CTL);                   /* Real Timer Prescale Timer 0 Control */
SFR_8BIT(RTCPS0CTL_L);                  /* Real Timer Prescale Timer 0 Control */
SFR_8BIT(RTCPS0CTL_H);                  /* Real Timer Prescale Timer 0 Control */
SFR_16BIT(RTCPS1CTL);                   /* Real Timer Prescale Timer 1 Control */
SFR_8BIT(RTCPS1CTL_L);                  /* Real Timer Prescale Timer 1 Control */
SFR_8BIT(RTCPS1CTL_H);                  /* Real Timer Prescale Timer 1 Control */
SFR_16BIT(RTCPS);                       /* Real Timer Prescale Timer Control */
SFR_8BIT(RTCPS_L);                      /* Real Timer Prescale Timer Control */
SFR_8BIT(RTCPS_H);                      /* Real Timer Prescale Timer Control */
SFR_16BIT(RTCIV);                       /* Real Time Clock Interrupt Vector */
SFR_16BIT(RTCTIM0);                     /* Real Time Clock Time 0 */
SFR_8BIT(RTCTIM0_L);                    /* Real Time Clock Time 0 */
SFR_8BIT(RTCTIM0_H);                    /* Real Time Clock Time 0 */
SFR_16BIT(RTCTIM1);                     /* Real Time Clock Time 1 */
SFR_8BIT(RTCTIM1_L);                    /* Real Time Clock Time 1 */
SFR_8BIT(RTCTIM1_H);                    /* Real Time Clock Time 1 */
SFR_16BIT(RTCDATE);                     /* Real Time Clock Date */
SFR_8BIT(RTCDATE_L);                    /* Real Time Clock Date */
SFR_8BIT(RTCDATE_H);                    /* Real Time Clock Date */
SFR_16BIT(RTCYEAR);                     /* Real Time Clock Year */
SFR_8BIT(RTCYEAR_L);                    /* Real Time Clock Year */
SFR_8BIT(RTCYEAR_H);                    /* Real Time Clock Year */
SFR_16BIT(RTCAMINHR);                   /* Real Time Clock Alarm Min/Hour */
SFR_8BIT(RTCAMINHR_L);                  /* Real Time Clock Alarm Min/Hour */
SFR_8BIT(RTCAMINHR_H);                  /* Real Time Clock Alarm Min/Hour */
SFR_16BIT(RTCADOWDAY);                  /* Real Time Clock Alarm day of
week/day */
SFR_8BIT(RTCADOWDAY_L);                 /* Real Time Clock Alarm day of
week/day */
SFR_8BIT(RTCADOWDAY_H);                 /* Real Time Clock Alarm day of
```

```
week/day */
SFR_16BIT(BIN2BCD);                         /* Real Time Binary-to-BCD conversion
register */
SFR_16BIT(BCD2BIN);                         /* Real Time BCD-to-binary conversion
register */

#define RTCSEC                  RTCTIM0_L
#define RTCMIN                  RTCTIM0_H
#define RTCHOUR                 RTCTIM1_L
#define RTCDOW                  RTCTIM1_H
#define RTCDAY                  RTCDATE_L
#define RTCMON                  RTCDATE_H
#define RTCYEARL                RTCYEAR_L
#define RT0PS                   RTCPS_L
#define RT1PS                   RTCPS_H
#define RTCAMIN                 RTCAMINHR_L     /* Real Time Clock Alarm Min */
#define RTCAHOUR                RTCAMINHR_H     /* Real Time Clock Alarm Hour */
#define RTCADOW                 RTCADOWDAY_L    /* Real Time Clock Alarm day of week
*/
#define RTCADAY                 RTCADOWDAY_H    /* Real Time Clock Alarm day */

/* RTCCTL0 Control Bits */
#define RTCOFIE                 (0x0080)        /* RTC 32kHz cyrstal oscillator fault
interrupt enable */
#define RTCTEVIE                (0x0040)        /* RTC Time Event Interrupt Enable Flag */
#define RTCAIE                  (0x0020)        /* RTC Alarm Interrupt Enable Flag */
#define RTCRDYIE                (0x0010)        /* RTC Ready Interrupt Enable Flag */
#define RTCOFIFG                (0x0008)        /* RTC 32kHz cyrstal oscillator fault
interrupt flag */
#define RTCTEVIFG               (0x0004)        /* RTC Time Event Interrupt Flag */
#define RTCAIFG                 (0x0002)        /* RTC Alarm Interrupt Flag */
#define RTCRDYIFG               (0x0001)        /* RTC Ready Interrupt Flag */

/* RTCCTL0 Control Bits */
#define RTCOFIE_L               (0x0080)        /* RTC 32kHz cyrstal oscillator fault
interrupt enable */
#define RTCTEVIE_L              (0x0040)        /* RTC Time Event Interrupt Enable Flag */
#define RTCAIE_L                (0x0020)        /* RTC Alarm Interrupt Enable Flag */
#define RTCRDYIE_L              (0x0010)        /* RTC Ready Interrupt Enable Flag */
#define RTCOFIFG_L              (0x0008)        /* RTC 32kHz cyrstal oscillator fault
interrupt flag */
#define RTCTEVIFG_L             (0x0004)        /* RTC Time Event Interrupt Flag */
#define RTCAIFG_L               (0x0002)        /* RTC Alarm Interrupt Flag */
#define RTCRDYIFG_L             (0x0001)        /* RTC Ready Interrupt Flag */

#define RTCKEY                  (0xA500)        /* RTC Key for RTC write access */
#define RTCKEY_H                (0xA5)          /* RTC Key for RTC write access (high
word) */
```

```
/* RTCCTL13 Control Bits */
#define RTCCALF1                (0x0200)        /* RTC Calibration Frequency Bit 1 */
#define RTCCALF0                (0x0100)        /* RTC Calibration Frequency Bit 0 */
#define RTCBCD                  (0x0080)        /* RTC BCD    0:Binary / 1:BCD */
#define RTCHOLD                 (0x0040)        /* RTC Hold */
#define RTCMODE                 (0x0020)        /* RTC Mode 0:Counter / 1: Calendar */
#define RTCRDY                  (0x0010)        /* RTC Ready */
#define RTCSSEL1                (0x0008)        /* RTC Source Select 1 */
#define RTCSSEL0                (0x0004)        /* RTC Source Select 0 */
#define RTCTEV1                 (0x0002)        /* RTC Time Event 1 */
#define RTCTEV0                 (0x0001)        /* RTC Time Event 0 */

/* RTCCTL13 Control Bits */
#define RTCBCD_L                (0x0080)        /* RTC BCD    0:Binary / 1:BCD */
#define RTCHOLD_L               (0x0040)        /* RTC Hold */
#define RTCMODE_L               (0x0020)        /* RTC Mode 0:Counter / 1: Calendar */
#define RTCRDY_L                (0x0010)        /* RTC Ready */
#define RTCSSEL1_L              (0x0008)        /* RTC Source Select 1 */
#define RTCSSEL0_L              (0x0004)        /* RTC Source Select 0 */
#define RTCTEV1_L               (0x0002)        /* RTC Time Event 1 */
#define RTCTEV0_L               (0x0001)        /* RTC Time Event 0 */

/* RTCCTL13 Control Bits */
#define RTCCALF1_H              (0x0002)        /* RTC Calibration Frequency Bit 1 */
#define RTCCALF0_H              (0x0001)        /* RTC Calibration Frequency Bit 0 */

#define RTCSSEL_0               (0x0000)        /* RTC Source Select ACLK */
#define RTCSSEL_1               (0x0004)        /* RTC Source Select SMCLK */
#define RTCSSEL_2               (0x0008)        /* RTC Source Select RT1PS */
#define RTCSSEL_3               (0x000C)        /* RTC Source Select RT1PS */
#define RTCSSEL__LFXT           (0x0000)        /* RTC Source Select LFXT */
#define RTCSSEL__RT1PS          (0x0008)        /* RTC Source Select RT1PS */

#define RTCSSEL__ACLK           (0x0000)        /* Legacy define */

#define RTCTEV_0                (0x0000)        /* RTC Time Event: 0 (Min. changed) */
#define RTCTEV_1                (0x0001)        /* RTC Time Event: 1 (Hour changed) */
#define RTCTEV_2                (0x0002)        /* RTC Time Event: 2 (12:00 changed) */
#define RTCTEV_3                (0x0003)        /* RTC Time Event: 3 (00:00 changed) */
#define RTCTEV__MIN             (0x0000)        /* RTC Time Event: 0 (Min. changed) */
#define RTCTEV__HOUR            (0x0001)        /* RTC Time Event: 1 (Hour changed) */
#define RTCTEV__0000            (0x0002)        /* RTC Time Event: 2 (00:00 changed) */
#define RTCTEV__1200            (0x0003)        /* RTC Time Event: 3 (12:00 changed) */

#define RTCCALF_0               (0x0000)        /* RTC Calibration Frequency: No Output */
#define RTCCALF_1               (0x0100)        /* RTC Calibration Frequency: 512 Hz */
#define RTCCALF_2               (0x0200)        /* RTC Calibration Frequency: 256 Hz */
```

```
#define RTCCALF_3                   (0x0300)        /* RTC Calibration Frequency: 1 Hz */

/* RTCOCAL Control Bits */
#define RTCOCALS                    (0x8000)        /* RTC Offset Calibration Sign */
#define RTCOCAL7                    (0x0080)        /* RTC Offset Calibration Bit 7 */
#define RTCOCAL6                    (0x0040)        /* RTC Offset Calibration Bit 6 */
#define RTCOCAL5                    (0x0020)        /* RTC Offset Calibration Bit 5 */
#define RTCOCAL4                    (0x0010)        /* RTC Offset Calibration Bit 4 */
#define RTCOCAL3                    (0x0008)        /* RTC Offset Calibration Bit 3 */
#define RTCOCAL2                    (0x0004)        /* RTC Offset Calibration Bit 2 */
#define RTCOCAL1                    (0x0002)        /* RTC Offset Calibration Bit 1 */
#define RTCOCAL0                    (0x0001)        /* RTC Offset Calibration Bit 0 */

/* RTCOCAL Control Bits */
#define RTCOCAL7_L                  (0x0080)        /* RTC Offset Calibration Bit 7 */
#define RTCOCAL6_L                  (0x0040)        /* RTC Offset Calibration Bit 6 */
#define RTCOCAL5_L                  (0x0020)        /* RTC Offset Calibration Bit 5 */
#define RTCOCAL4_L                  (0x0010)        /* RTC Offset Calibration Bit 4 */
#define RTCOCAL3_L                  (0x0008)        /* RTC Offset Calibration Bit 3 */
#define RTCOCAL2_L                  (0x0004)        /* RTC Offset Calibration Bit 2 */
#define RTCOCAL1_L                  (0x0002)        /* RTC Offset Calibration Bit 1 */
#define RTCOCAL0_L                  (0x0001)        /* RTC Offset Calibration Bit 0 */

/* RTCOCAL Control Bits */
#define RTCOCALS_H                  (0x0080)        /* RTC Offset Calibration Sign */

/* RTCTCMP Control Bits */
#define RTCTCMPS                    (0x8000)        /* RTC Temperature Compensation Sign */
#define RTCTCRDY                    (0x4000)        /* RTC Temperature compensation ready
*/
#define RTCTCOK                     (0x2000)        /* RTC Temperature compensation write
OK */
#define RTCTCMP7                    (0x0080)        /* RTC Temperature Compensation Bit 7
*/
#define RTCTCMP6                    (0x0040)        /* RTC Temperature Compensation Bit 6
*/
#define RTCTCMP5                    (0x0020)        /* RTC Temperature Compensation Bit 5
*/
#define RTCTCMP4                    (0x0010)        /* RTC Temperature Compensation Bit 4
*/
#define RTCTCMP3                    (0x0008)        /* RTC Temperature Compensation Bit 3
*/
#define RTCTCMP2                    (0x0004)        /* RTC Temperature Compensation Bit 2
*/
#define RTCTCMP1                    (0x0002)        /* RTC Temperature Compensation Bit 1
*/
#define RTCTCMP0                    (0x0001)        /* RTC Temperature Compensation Bit 0
*/
```

```
/* RTCTCMP Control Bits */
#define RTCTCMP7_L              (0x0080)        /* RTC Temperature Compensation Bit 7
*/
#define RTCTCMP6_L              (0x0040)        /* RTC Temperature Compensation Bit 6
*/
#define RTCTCMP5_L              (0x0020)        /* RTC Temperature Compensation Bit 5
*/
#define RTCTCMP4_L              (0x0010)        /* RTC Temperature Compensation Bit 4
*/
#define RTCTCMP3_L              (0x0008)        /* RTC Temperature Compensation Bit 3
*/
#define RTCTCMP2_L              (0x0004)        /* RTC Temperature Compensation Bit 2
*/
#define RTCTCMP1_L              (0x0002)        /* RTC Temperature Compensation Bit 1
*/
#define RTCTCMP0_L              (0x0001)        /* RTC Temperature Compensation Bit 0
*/

/* RTCTCMP Control Bits */
#define RTCTCMPS_H              (0x0080)        /* RTC Temperature Compensation Sign
*/
#define RTCTCRDY_H              (0x0040)        /* RTC Temperature compensation ready
*/
#define RTCTCOK_H               (0x0020)        /* RTC Temperature compensation write
OK */

#define RTCAE                   (0x80)          /* Real Time Clock Alarm enable */

/* RTCPS0CTL Control Bits */
//#define Reserved              (0x8000)
//#define Reserved              (0x4000)
#define RT0PSDIV2               (0x2000)        /* RTC Prescale Timer 0 Clock Divide Bit: 2
*/
#define RT0PSDIV1               (0x1000)        /* RTC Prescale Timer 0 Clock Divide Bit: 1
*/
#define RT0PSDIV0               (0x0800)        /* RTC Prescale Timer 0 Clock Divide Bit: 0
*/
//#define Reserved              (0x0400)
//#define Reserved              (0x0200)
#define RT0PSHOLD               (0x0100)        /* RTC Prescale Timer 0 Hold */
//#define Reserved              (0x0080)
//#define Reserved              (0x0040)
//#define Reserved              (0x0020)
#define RT0IP2                  (0x0010)        /* RTC Prescale Timer 0 Interrupt Interval
Bit: 2 */
#define RT0IP1                  (0x0008)        /* RTC Prescale Timer 0 Interrupt Interval
Bit: 1 */
```

```
#define RT0IP0                  (0x0004)        /* RTC Prescale Timer 0 Interrupt Interval
Bit: 0 */
#define RT0PSIE                 (0x0002)        /* RTC Prescale Timer 0 Interrupt Enable
Flag */
#define RT0PSIFG                (0x0001)        /* RTC Prescale Timer 0 Interrupt Flag */

/* RTCPS0CTL Control Bits */
//#define Reserved              (0x8000)
//#define Reserved              (0x4000)
//#define Reserved              (0x0400)
//#define Reserved              (0x0200)
//#define Reserved              (0x0080)
//#define Reserved              (0x0040)
//#define Reserved              (0x0020)
#define RT0IP2_L                (0x0010)        /* RTC Prescale Timer 0 Interrupt Interval
Bit: 2 */
#define RT0IP1_L                (0x0008)        /* RTC Prescale Timer 0 Interrupt Interval
Bit: 1 */
#define RT0IP0_L                (0x0004)        /* RTC Prescale Timer 0 Interrupt Interval
Bit: 0 */
#define RT0PSIE_L               (0x0002)       /* RTC Prescale Timer 0 Interrupt Enable
Flag */
#define RT0PSIFG_L              (0x0001)        /* RTC Prescale Timer 0 Interrupt Flag */

/* RTCPS0CTL Control Bits */
//#define Reserved              (0x8000)
//#define Reserved              (0x4000)
#define RT0PSDIV2_H             (0x0020)        /* RTC Prescale Timer 0 Clock Divide Bit: 2
*/
#define RT0PSDIV1_H             (0x0010)        /* RTC Prescale Timer 0 Clock Divide Bit: 1
*/
#define RT0PSDIV0_H             (0x0008)        /* RTC Prescale Timer 0 Clock Divide Bit: 0
*/
//#define Reserved              (0x0400)
//#define Reserved              (0x0200)
#define RT0PSHOLD_H             (0x0001)        /* RTC Prescale Timer 0 Hold */
//#define Reserved              (0x0080)
//#define Reserved              (0x0040)
//#define Reserved              (0x0020)

#define RT0IP_0                 (0x0000)        /* RTC Prescale Timer 0 Interrupt Interval /2
*/
#define RT0IP_1                 (0x0004)        /* RTC Prescale Timer 0 Interrupt Interval /4
*/
#define RT0IP_2                 (0x0008)        /* RTC Prescale Timer 0 Interrupt Interval /8
*/
#define RT0IP_3                 (0x000C)        /* RTC Prescale Timer 0 Interrupt Interval
/16 */
```

```
#define RT0IP_4                 (0x0010)        /* RTC Prescale Timer 0 Interrupt Interval
/32 */
#define RT0IP_5                 (0x0014)        /* RTC Prescale Timer 0 Interrupt Interval
/64 */
#define RT0IP_6                 (0x0018)        /* RTC Prescale Timer 0 Interrupt Interval
/128 */
#define RT0IP_7                 (0x001C)        /* RTC Prescale Timer 0 Interrupt Interval
/256 */

#define RT0PSDIV_0              (0x0000)        /* RTC Prescale Timer 0 Clock Divide: /2 */
#define RT0PSDIV_1              (0x0800)        /* RTC Prescale Timer 0 Clock Divide: /4 */
#define RT0PSDIV_2              (0x1000)        /* RTC Prescale Timer 0 Clock Divide: /8 */
#define RT0PSDIV_3              (0x1800)        /* RTC Prescale Timer 0 Clock Divide: /16 */
#define RT0PSDIV_4              (0x2000)        /* RTC Prescale Timer 0 Clock Divide: /32 */
#define RT0PSDIV_5              (0x2800)        /* RTC Prescale Timer 0 Clock Divide: /64 */
#define RT0PSDIV_6              (0x3000)        /* RTC Prescale Timer 0 Clock Divide: /128
*/
#define RT0PSDIV_7              (0x3800)        /* RTC Prescale Timer 0 Clock Divide: /256
*/

/* RTCPS1CTL Control Bits */
#define RT1SSEL1                (0x8000)        /* RTC Prescale Timer 1 Source Select Bit: 1
*/
#define RT1SSEL0                (0x4000)        /* RTC Prescale Timer 1 Source Select Bit: 0
*/
#define RT1PSDIV2               (0x2000)        /* RTC Prescale Timer 1 Clock Divide Bit: 2
*/
#define RT1PSDIV1               (0x1000)        /* RTC Prescale Timer 1 Clock Divide Bit: 1
*/
#define RT1PSDIV0               (0x0800)        /* RTC Prescale Timer 1 Clock Divide Bit: 0
*/
//#define Reserved              (0x0400)
//#define Reserved              (0x0200)
#define RT1PSHOLD                (0x0100)        /* RTC Prescale Timer 1 Hold */
//#define Reserved              (0x0080)
//#define Reserved              (0x0040)
//#define Reserved              (0x0020)
#define RT1IP2                  (0x0010)        /* RTC Prescale Timer 1 Interrupt Interval
Bit: 2 */
#define RT1IP1                  (0x0008)        /* RTC Prescale Timer 1 Interrupt Interval
Bit: 1 */
#define RT1IP0                  (0x0004)        /* RTC Prescale Timer 1 Interrupt Interval
Bit: 0 */
#define RT1PSIE                 (0x0002)        /* RTC Prescale Timer 1 Interrupt Enable
Flag */
#define RT1PSIFG                (0x0001)        /* RTC Prescale Timer 1 Interrupt Flag */

/* RTCPS1CTL Control Bits */
```

```
//#define Reserved              (0x0400)
//#define Reserved              (0x0200)
//#define Reserved              (0x0080)
//#define Reserved              (0x0040)
//#define Reserved              (0x0020)
#define RT1IP2_L                    (0x0010)        /* RTC Prescale Timer 1 Interrupt Interval
Bit: 2 */
#define RT1IP1_L                    (0x0008)        /* RTC Prescale Timer 1 Interrupt Interval
Bit: 1 */
#define RT1IP0_L                    (0x0004)        /* RTC Prescale Timer 1 Interrupt Interval
Bit: 0 */
#define RT1PSIE_L                   (0x0002)        /* RTC Prescale Timer 1 Interrupt Enable
Flag */
#define RT1PSIFG_L                  (0x0001)        /* RTC Prescale Timer 1 Interrupt Flag */

/* RTCPS1CTL Control Bits */
#define RT1SSEL1_H                  (0x0080)        /* RTC Prescale Timer 1 Source Select Bit: 1
*/
#define RT1SSEL0_H                  (0x0040)        /* RTC Prescale Timer 1 Source Select Bit: 0
*/
#define RT1PSDIV2_H                 (0x0020)        /* RTC Prescale Timer 1 Clock Divide Bit: 2
*/
#define RT1PSDIV1_H                 (0x0010)        /* RTC Prescale Timer 1 Clock Divide Bit: 1
*/
#define RT1PSDIV0_H                 (0x0008)        /* RTC Prescale Timer 1 Clock Divide Bit: 0
*/
//#define Reserved              (0x0400)
//#define Reserved              (0x0200)
#define RT1PSHOLD_H                 (0x0001)        /* RTC Prescale Timer 1 Hold */
//#define Reserved              (0x0080)
//#define Reserved              (0x0040)
//#define Reserved              (0x0020)

#define RT1IP_0                     (0x0000)        /* RTC Prescale Timer 1 Interrupt Interval /2
*/
#define RT1IP_1                     (0x0004)        /* RTC Prescale Timer 1 Interrupt Interval /4
*/
#define RT1IP_2                     (0x0008)        /* RTC Prescale Timer 1 Interrupt Interval /8
*/
#define RT1IP_3                     (0x000C)        /* RTC Prescale Timer 1 Interrupt Interval
/16 */
#define RT1IP_4                     (0x0010)        /* RTC Prescale Timer 1 Interrupt Interval
/32 */
#define RT1IP_5                     (0x0014)        /* RTC Prescale Timer 1 Interrupt Interval
/64 */
#define RT1IP_6                     (0x0018)        /* RTC Prescale Timer 1 Interrupt Interval
/128 */
#define RT1IP_7                     (0x001C)        /* RTC Prescale Timer 1 Interrupt Interval
```

```
/256 */

#define RT1PSDIV_0                (0x0000)        /* RTC Prescale Timer 1 Clock Divide: /2 */
#define RT1PSDIV_1                (0x0800)        /* RTC Prescale Timer 1 Clock Divide: /4 */
#define RT1PSDIV_2                (0x1000)        /* RTC Prescale Timer 1 Clock Divide: /8 */
#define RT1PSDIV_3                (0x1800)        /* RTC Prescale Timer 1 Clock Divide: /16 */
#define RT1PSDIV_4                (0x2000)        /* RTC Prescale Timer 1 Clock Divide: /32 */
#define RT1PSDIV_5                (0x2800)        /* RTC Prescale Timer 1 Clock Divide: /64 */
#define RT1PSDIV_6                (0x3000)        /* RTC Prescale Timer 1 Clock Divide: /128
*/
#define RT1PSDIV_7                (0x3800)        /* RTC Prescale Timer 1 Clock Divide: /256
*/

#define RT1SSEL_0                 (0x0000)        /* RTC Prescale Timer 1 Source Select: 0 */
#define RT1SSEL_1                 (0x4000)        /* RTC Prescale Timer 1 Source Select: 1 */
#define RT1SSEL_2                 (0x8000)        /* RTC Prescale Timer 1 Source Select: 2 */
#define RT1SSEL_3                 (0xC000)        /* RTC Prescale Timer 1 Source Select: 3 */

/* RTC Definitions */
#define RTCIV_NONE                 (0x0000)        /* No Interrupt pending */
#define RTCIV_RTCOFIFG            (0x0002)        /* RTC Osc fault: RTCOFIFG */
#define RTCIV_RTCRDYIFG           (0x0004)        /* RTC ready: RTCRDYIFG */
#define RTCIV_RTCTEVIFG           (0x0006)        /* RTC interval timer: RTCTEVIFG */
#define RTCIV_RTCAIFG             (0x0008)        /* RTC user alarm: RTCAIFG */
#define RTCIV_RT0PSIFG            (0x000A)        /* RTC prescaler 0: RT0PSIFG */
#define RTCIV_RT1PSIFG            (0x000C)        /* RTC prescaler 1: RT1PSIFG */

/* Legacy Definitions */
#define RTC_NONE                   (0x0000)        /* No Interrupt pending */
#define RTC_RTCOFIFG              (0x0002)        /* RTC Osc fault: RTCOFIFG */
#define RTC_RTCRDYIFG             (0x0004)        /* RTC ready: RTCRDYIFG */
#define RTC_RTCTEVIFG             (0x0006)        /* RTC interval timer: RTCTEVIFG */
#define RTC_RTCAIFG               (0x0008)        /* RTC user alarm: RTCAIFG */
#define RTC_RT0PSIFG              (0x000A)        /* RTC prescaler 0: RT0PSIFG */
#define RTC_RT1PSIFG              (0x000C)        /* RTC prescaler 1: RT1PSIFG */

/************************************************************
* SFR - Special Function Register Module
************************************************************/
#define __MSP430_HAS_SFR__                      /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_SFR__ 0x0100
#define SFR_BASE                  __MSP430_BASEADDRESS_SFR__

SFR_16BIT(SFRIE1);                              /* Interrupt Enable 1 */
SFR_8BIT(SFRIE1_L);                             /* Interrupt Enable 1 */
SFR_8BIT(SFRIE1_H);                             /* Interrupt Enable 1 */
```

```
/* SFRIE1 Control Bits */
#define WDTIE                  (0x0001)  /* WDT Interrupt Enable */
#define OFIE                   (0x0002)  /* Osc Fault Enable */
//#define Reserved             (0x0004)
#define VMAIE                  (0x0008)  /* Vacant Memory Interrupt Enable */
#define NMIIE                  (0x0010)  /* NMI Interrupt Enable */
#define JMBINIE                (0x0040)  /* JTAG Mail Box input Interrupt Enable */
#define JMBOUTIE               (0x0080)  /* JTAG Mail Box output Interrupt Enable
*/

#define WDTIE_L                (0x0001)  /* WDT Interrupt Enable */
#define OFIE_L                 (0x0002)  /* Osc Fault Enable */
//#define Reserved             (0x0004)
#define VMAIE_L                (0x0008)  /* Vacant Memory Interrupt Enable */
#define NMIIE_L                (0x0010)  /* NMI Interrupt Enable */
#define JMBINIE_L              (0x0040)  /* JTAG Mail Box input Interrupt Enable */
#define JMBOUTIE_L             (0x0080)  /* JTAG Mail Box output Interrupt Enable
*/

SFR_16BIT(SFRIFG1);                      /* Interrupt Flag 1 */
SFR_8BIT(SFRIFG1_L);                     /* Interrupt Flag 1 */
SFR_8BIT(SFRIFG1_H);                     /* Interrupt Flag 1 */
/* SFRIFG1 Control Bits */
#define WDTIFG                 (0x0001)  /* WDT Interrupt Flag */
#define OFIFG                  (0x0002)  /* Osc Fault Flag */
//#define Reserved             (0x0004)
#define VMAIFG                 (0x0008)  /* Vacant Memory Interrupt Flag */
#define NMIIFG                 (0x0010)  /* NMI Interrupt Flag */
//#define Reserved             (0x0020)
#define JMBINIFG               (0x0040)  /* JTAG Mail Box input Interrupt Flag */
#define JMBOUTIFG              (0x0080)  /* JTAG Mail Box output Interrupt Flag */

#define WDTIFG_L               (0x0001)  /* WDT Interrupt Flag */
#define OFIFG_L                (0x0002)  /* Osc Fault Flag */
//#define Reserved             (0x0004)
#define VMAIFG_L               (0x0008)  /* Vacant Memory Interrupt Flag */
#define NMIIFG_L               (0x0010)  /* NMI Interrupt Flag */
//#define Reserved             (0x0020)
#define JMBINIFG_L             (0x0040)  /* JTAG Mail Box input Interrupt Flag */
#define JMBOUTIFG_L            (0x0080)  /* JTAG Mail Box output Interrupt Flag */

SFR_16BIT(SFRRPCR);                      /* RESET Pin Control Register */
SFR_8BIT(SFRRPCR_L);                     /* RESET Pin Control Register */
SFR_8BIT(SFRRPCR_H);                     /* RESET Pin Control Register */
/* SFRRPCR Control Bits */
#define SYSNMI                 (0x0001)  /* NMI select */
#define SYSNMIIES              (0x0002)  /* NMI edge select */
#define SYSRSTUP               (0x0004)  /* RESET Pin pull down/up select */
```

```
#define SYSRSTRE                (0x0008)          /* RESET Pin Resistor enable */

#define SYSNMI_L                (0x0001)          /* NMI select */
#define SYSNMIIES_L             (0x0002)          /* NMI edge select */
#define SYSRSTUP_L              (0x0004)          /* RESET Pin pull down/up select */
#define SYSRSTRE_L              (0x0008)          /* RESET Pin Resistor enable */

/************************************************************
* SYS - System Module
************************************************************/
/************************************************************
* Timer0_A3
************************************************************/
#define __MSP430_HAS_T0A3__                       /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_T0A3__ 0x0340
#define TIMER_A0_BASE               __MSP430_BASEADDRESS_T0A3__

SFR_16BIT(TA0CTL);                                /* Timer0_A3 Control */
SFR_16BIT(TA0CCTL0);                              /* Timer0_A3 Capture/Compare Control 0
*/
SFR_16BIT(TA0CCTL1);                              /* Timer0_A3 Capture/Compare Control 1
*/
SFR_16BIT(TA0CCTL2);                              /* Timer0_A3 Capture/Compare Control 2
*/
SFR_16BIT(TA0R);                                  /* Timer0_A3 */
SFR_16BIT(TA0CCR0);                               /* Timer0_A3 Capture/Compare 0 */
SFR_16BIT(TA0CCR1);                               /* Timer0_A3 Capture/Compare 1 */
SFR_16BIT(TA0CCR2);                               /* Timer0_A3 Capture/Compare 2 */
SFR_16BIT(TA0IV);                                 /* Timer0_A3 Interrupt Vector Word */
SFR_16BIT(TA0EX0);                                /* Timer0_A3 Expansion Register 0 */

/* TAxCTL Control Bits */
#define TASSEL1                 (0x0200)          /* Timer A clock source select 1 */
#define TASSEL0                 (0x0100)          /* Timer A clock source select 0 */
#define ID1                     (0x0080)          /* Timer A clock input divider 1 */
#define ID0                     (0x0040)          /* Timer A clock input divider 0 */
#define MC1                     (0x0020)          /* Timer A mode control 1 */
#define MC0                     (0x0010)          /* Timer A mode control 0 */
#define TACLR                   (0x0004)          /* Timer A counter clear */
#define TAIE                    (0x0002)          /* Timer A counter interrupt enable */
#define TAIFG                   (0x0001)          /* Timer A counter interrupt flag */

#define MC_0                    (0*0x10u)         /* Timer A mode control: 0 - Stop */
#define MC_1                    (1*0x10u)         /* Timer A mode control: 1 - Up to CCR0
*/
#define MC_2                    (2*0x10u)         /* Timer A mode control: 2 - Continuous
up */
```

```
#define MC_3                    (3*0x10u)       /* Timer A mode control: 3 - Up/Down */
#define ID_0                    (0*0x40u)       /* Timer A input divider: 0 - /1 */
#define ID_1                    (1*0x40u)       /* Timer A input divider: 1 - /2 */
#define ID_2                    (2*0x40u)       /* Timer A input divider: 2 - /4 */
#define ID_3                    (3*0x40u)       /* Timer A input divider: 3 - /8 */
#define TASSEL_0                (0*0x100u)      /* Timer A clock source select: 0 - TACLK */
#define TASSEL_1                (1*0x100u)      /* Timer A clock source select: 1 - ACLK  */
#define TASSEL_2                (2*0x100u)      /* Timer A clock source select: 2 - SMCLK */
#define TASSEL_3                (3*0x100u)      /* Timer A clock source select: 3 - INCLK */
#define MC__STOP                (0*0x10u)       /* Timer A mode control: 0 - Stop */
#define MC__UP                  (1*0x10u)       /* Timer A mode control: 1 - Up to CCR0
*/
#define MC__CONTINUOUS          (2*0x10u)       /* Timer A mode control: 2 - Continuous
up */
#define MC__CONTINOUS           (2*0x10u)       /* Legacy define */
#define MC__UPDOWN              (3*0x10u)       /* Timer A mode control: 3 - Up/Down
*/
#define ID__1                   (0*0x40u)       /* Timer A input divider: 0 - /1 */
#define ID__2                   (1*0x40u)       /* Timer A input divider: 1 - /2 */
#define ID__4                   (2*0x40u)       /* Timer A input divider: 2 - /4 */
#define ID__8                   (3*0x40u)       /* Timer A input divider: 3 - /8 */
#define TASSEL__TACLK           (0*0x100u)      /* Timer A clock source select: 0 - TACLK */
#define TASSEL__ACLK            (1*0x100u)      /* Timer A clock source select: 1 - ACLK  */
#define TASSEL__SMCLK           (2*0x100u)      /* Timer A clock source select: 2 - SMCLK */
#define TASSEL__INCLK           (3*0x100u)      /* Timer A clock source select: 3 - INCLK */

/* TAxCCTLx Control Bits */
#define CM1                     (0x8000)        /* Capture mode 1 */
#define CM0                     (0x4000)        /* Capture mode 0 */
#define CCIS1                   (0x2000)        /* Capture input select 1 */
#define CCIS0                   (0x1000)        /* Capture input select 0 */
#define SCS                     (0x0800)        /* Capture sychronize */
#define SCCI                    (0x0400)        /* Latched capture signal (read) */
#define CAP                     (0x0100)        /* Capture mode: 1 /Compare mode : 0 */
#define OUTMOD2                 (0x0080)        /* Output mode 2 */
#define OUTMOD1                 (0x0040)        /* Output mode 1 */
#define OUTMOD0                 (0x0020)        /* Output mode 0 */
#define CCIE                    (0x0010)        /* Capture/compare interrupt enable */
#define CCI                     (0x0008)        /* Capture input signal (read) */
#define OUT                     (0x0004)        /* PWM Output signal if output mode 0 */
#define COV                     (0x0002)        /* Capture/compare overflow flag */
#define CCIFG                   (0x0001)        /* Capture/compare interrupt flag */

#define OUTMOD_0                (0*0x20u)       /* PWM output mode: 0 - output only */
#define OUTMOD_1                (1*0x20u)       /* PWM output mode: 1 - set */
#define OUTMOD_2                (2*0x20u)       /* PWM output mode: 2 - PWM
toggle/reset */
#define OUTMOD_3                (3*0x20u)       /* PWM output mode: 3 - PWM set/reset
```

```
*/
#define OUTMOD_4                  (4*0x20u)          /* PWM output mode: 4 - toggle */
#define OUTMOD_5                  (5*0x20u)          /* PWM output mode: 5 - Reset */
#define OUTMOD_6                  (6*0x20u)          /* PWM output mode: 6 - PWM
toggle/set */
#define OUTMOD_7                  (7*0x20u)          /* PWM output mode: 7 - PWM reset/set
*/
#define CCIS_0                   (0*0x1000u)       /* Capture input select: 0 - CCIxA */
#define CCIS_1                   (1*0x1000u)       /* Capture input select: 1 - CCIxB */
#define CCIS_2                   (2*0x1000u)       /* Capture input select: 2 - GND */
#define CCIS_3                   (3*0x1000u)       /* Capture input select: 3 - Vcc */
#define CM_0                      (0*0x4000u)        /* Capture mode: 0 - disabled */
#define CM_1                      (1*0x4000u)        /* Capture mode: 1 - pos. edge */
#define CM_2                      (2*0x4000u)        /* Capture mode: 1 - neg. edge */
#define CM_3                      (3*0x4000u)        /* Capture mode: 1 - both edges */

/* TAxEX0 Control Bits */
#define TAIDEX0                  (0x0001)           /* Timer A Input divider expansion Bit: 0 */
#define TAIDEX1                  (0x0002)           /* Timer A Input divider expansion Bit: 1 */
#define TAIDEX2                  (0x0004)           /* Timer A Input divider expansion Bit: 2 */

#define TAIDEX_0                 (0*0x0001u)       /* Timer A Input divider expansion : /1 */
#define TAIDEX_1                 (1*0x0001u)       /* Timer A Input divider expansion : /2 */
#define TAIDEX_2                 (2*0x0001u)       /* Timer A Input divider expansion : /3 */
#define TAIDEX_3                 (3*0x0001u)       /* Timer A Input divider expansion : /4 */
#define TAIDEX_4                 (4*0x0001u)       /* Timer A Input divider expansion : /5 */
#define TAIDEX_5                 (5*0x0001u)       /* Timer A Input divider expansion : /6 */
#define TAIDEX_6                 (6*0x0001u)       /* Timer A Input divider expansion : /7 */
#define TAIDEX_7                 (7*0x0001u)       /* Timer A Input divider expansion : /8 */

/* T0A3IV Definitions */
#define TA0IV_NONE                (0x0000)           /* No Interrupt pending */
#define TA0IV_TACCR1             (0x0002)           /* TA0CCR1_CCIFG */
#define TA0IV_TACCR2             (0x0004)           /* TA0CCR2_CCIFG */
#define TA0IV_3                  (0x0006)           /* Reserved */
#define TA0IV_4                  (0x0008)           /* Reserved */
#define TA0IV_5                  (0x000A)           /* Reserved */
#define TA0IV_6                  (0x000C)           /* Reserved */
#define TA0IV_TAIFG              (0x000E)          /* TA0IFG */

/* Legacy Defines */
#define TA0IV_TA0CCR1            (0x0002)           /* TA0CCR1_CCIFG */
#define TA0IV_TA0CCR2            (0x0004)           /* TA0CCR2_CCIFG */
#define TA0IV_TA0IFG             (0x000E)          /* TA0IFG */

/************************************************************
* Timer1_A3
************************************************************/
```

```
#define __MSP430_HAS_T1A3__                       /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_T1A3__ 0x0380
#define TIMER_A1_BASE              __MSP430_BASEADDRESS_T1A3__

SFR_16BIT(TA1CTL);                                /* Timer1_A3 Control */
SFR_16BIT(TA1CCTL0);                              /* Timer1_A3 Capture/Compare Control 0
*/
SFR_16BIT(TA1CCTL1);                              /* Timer1_A3 Capture/Compare Control 1
*/
SFR_16BIT(TA1CCTL2);                              /* Timer1_A3 Capture/Compare Control 2
*/
SFR_16BIT(TA1R);                                  /* Timer1_A3 */
SFR_16BIT(TA1CCR0);                               /* Timer1_A3 Capture/Compare 0 */
SFR_16BIT(TA1CCR1);                               /* Timer1_A3 Capture/Compare 1 */
SFR_16BIT(TA1CCR2);                               /* Timer1_A3 Capture/Compare 2 */
SFR_16BIT(TA1IV);                                 /* Timer1_A3 Interrupt Vector Word */
SFR_16BIT(TA1EX0);                                /* Timer1_A3 Expansion Register 0 */

/* Bits are already defined within the Timer0_Ax */

/* TA1IV Definitions */
#define TA1IV_NONE                 (0x0000)       /* No Interrupt pending */
#define TA1IV_TACCR1               (0x0002)       /* TA1CCR1_CCIFG */
#define TA1IV_TACCR2               (0x0004)       /* TA1CCR2_CCIFG */
#define TA1IV_3                    (0x0006)       /* Reserved */
#define TA1IV_4                    (0x0008)       /* Reserved */
#define TA1IV_5                    (0x000A)       /* Reserved */
#define TA1IV_6                    (0x000C)       /* Reserved */
#define TA1IV_TAIFG                (0x000E)       /* TA1IFG */

/* Legacy Defines */
#define TA1IV_TA1CCR1              (0x0002)       /* TA1CCR1_CCIFG */
#define TA1IV_TA1CCR2              (0x0004)       /* TA1CCR2_CCIFG */
#define TA1IV_TA1IFG               (0x000E)       /* TA1IFG */

/************************************************************
* Timer2_A2
************************************************************/
#define __MSP430_HAS_T2A2__                       /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_T2A2__ 0x0400
#define TIMER_A2_BASE              __MSP430_BASEADDRESS_T2A2__

SFR_16BIT(TA2CTL);                                /* Timer2_A2 Control */
SFR_16BIT(TA2CCTL0);                              /* Timer2_A2 Capture/Compare Control 0
*/
SFR_16BIT(TA2CCTL1);                              /* Timer2_A2 Capture/Compare Control 1
```

```
*/
SFR_16BIT(TA2R);                                    /* Timer2_A2 */
SFR_16BIT(TA2CCR0);                                 /* Timer2_A2 Capture/Compare 0 */
SFR_16BIT(TA2CCR1);                                 /* Timer2_A2 Capture/Compare 1 */
SFR_16BIT(TA2IV);                                   /* Timer2_A2 Interrupt Vector Word */
SFR_16BIT(TA2EX0);                                  /* Timer2_A2 Expansion Register 0 */

/* Bits are already defined within the Timer0_Ax */

/* TA2IV Definitions */
#define TA2IV_NONE                  (0x0000)        /* No Interrupt pending */
#define TA2IV_TACCR1                (0x0002)        /* TA2CCR1_CCIFG */
#define TA2IV_3                     (0x0006)        /* Reserved */
#define TA2IV_4                     (0x0008)        /* Reserved */
#define TA2IV_5                     (0x000A)        /* Reserved */
#define TA2IV_6                     (0x000C)        /* Reserved */
#define TA2IV_TAIFG                 (0x000E)        /* TA2IFG */

/* Legacy Defines */
#define TA2IV_TA2CCR1               (0x0002)        /* TA2CCR1_CCIFG */
#define TA2IV_TA2IFG                (0x000E)        /* TA2IFG */

/************************************************************
* Timer3_A5
************************************************************/
#define __MSP430_HAS_T3A5__                         /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_T3A5__ 0x0440
#define TIMER_A3_BASE               __MSP430_BASEADDRESS_T3A5__

SFR_16BIT(TA3CTL);                                  /* Timer3_A5 Control */
SFR_16BIT(TA3CCTL0);                                /* Timer3_A5 Capture/Compare Control 0
*/
SFR_16BIT(TA3CCTL1);                                /* Timer3_A5 Capture/Compare Control 1
*/
SFR_16BIT(TA3CCTL2);                                /* Timer3_A5 Capture/Compare Control 2
*/
SFR_16BIT(TA3CCTL3);                                /* Timer3_A5 Capture/Compare Control 3
*/
SFR_16BIT(TA3CCTL4);                                /* Timer3_A5 Capture/Compare Control 4
*/
SFR_16BIT(TA3R);                                    /* Timer3_A5 */
SFR_16BIT(TA3CCR0);                                 /* Timer3_A5 Capture/Compare 0 */
SFR_16BIT(TA3CCR1);                                 /* Timer3_A5 Capture/Compare 1 */
SFR_16BIT(TA3CCR2);                                 /* Timer3_A5 Capture/Compare 2 */
SFR_16BIT(TA3CCR3);                                 /* Timer3_A5 Capture/Compare 3 */
SFR_16BIT(TA3CCR4);                                 /* Timer3_A5 Capture/Compare 4 */
SFR_16BIT(TA3IV);                                   /* Timer3_A5 Interrupt Vector Word */
```

```
SFR_16BIT(TA3EX0);                                  /* Timer3_A5 Expansion Register 0 */

/* Bits are already defined within the Timer0_Ax */

/* TA3IV Definitions */
#define TA3IV_NONE                  (0x0000)        /* No Interrupt pending */
#define TA3IV_TACCR1                (0x0002)        /* TA3CCR1_CCIFG */
#define TA3IV_TACCR2                (0x0004)        /* TA3CCR2_CCIFG */
#define TA3IV_TACCR3                (0x0006)        /* TA3CCR3_CCIFG */
#define TA3IV_TACCR4                (0x0008)        /* TA3CCR4_CCIFG */
#define TA3IV_TAIFG                 (0x000E)        /* TA3IFG */

/* Legacy Defines */
#define TA3IV_TA3CCR1               (0x0002)        /* TA3CCR1_CCIFG */
#define TA3IV_TA3CCR2               (0x0004)        /* TA3CCR2_CCIFG */
#define TA3IV_TA3CCR3               (0x0006)        /* TA3CCR3_CCIFG */
#define TA3IV_TA3CCR4               (0x0008)        /* TA3CCR4_CCIFG */
#define TA3IV_TA3IFG                (0x000E)        /* TA3IFG */

/************************************************************
* Timer0_B7
************************************************************/
#define __MSP430_HAS_T0B7__                         /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_T0B7__ 0x03C0
#define TIMER_B0_BASE               __MSP430_BASEADDRESS_T0B7__

SFR_16BIT(TB0CTL);                                  /* Timer0_B7 Control */
SFR_16BIT(TB0CCTL0);                                /* Timer0_B7 Capture/Compare Control 0
*/
SFR_16BIT(TB0CCTL1);                                /* Timer0_B7 Capture/Compare Control 1
*/
SFR_16BIT(TB0CCTL2);                                /* Timer0_B7 Capture/Compare Control 2
*/
SFR_16BIT(TB0CCTL3);                                /* Timer0_B7 Capture/Compare Control 3
*/
SFR_16BIT(TB0CCTL4);                                /* Timer0_B7 Capture/Compare Control 4
*/
SFR_16BIT(TB0CCTL5);                                /* Timer0_B7 Capture/Compare Control 5
*/
SFR_16BIT(TB0CCTL6);                                /* Timer0_B7 Capture/Compare Control 6
*/
SFR_16BIT(TB0R);                                    /* Timer0_B7 */
SFR_16BIT(TB0CCR0);                                 /* Timer0_B7 Capture/Compare 0 */
SFR_16BIT(TB0CCR1);                                 /* Timer0_B7 Capture/Compare 1 */
SFR_16BIT(TB0CCR2);                                 /* Timer0_B7 Capture/Compare 2 */
SFR_16BIT(TB0CCR3);                                 /* Timer0_B7 Capture/Compare 3 */
SFR_16BIT(TB0CCR4);                                 /* Timer0_B7 Capture/Compare 4 */
```

```
SFR_16BIT(TB0CCR5);                          /* Timer0_B7 Capture/Compare 5 */
SFR_16BIT(TB0CCR6);                          /* Timer0_B7 Capture/Compare 6 */
SFR_16BIT(TB0EX0);                           /* Timer0_B7 Expansion Register 0 */
SFR_16BIT(TB0IV);                            /* Timer0_B7 Interrupt Vector Word */

/* Legacy Type Definitions for TimerB */
#define TBCTL                   TB0CTL       /* Timer0_B7 Control */
#define TBCCTL0                 TB0CCTL0     /* Timer0_B7 Capture/Compare Control 0
*/
#define TBCCTL1                 TB0CCTL1     /* Timer0_B7 Capture/Compare Control 1
*/
#define TBCCTL2                 TB0CCTL2     /* Timer0_B7 Capture/Compare Control 2
*/
#define TBCCTL3                 TB0CCTL3     /* Timer0_B7 Capture/Compare Control 3
*/
#define TBCCTL4                 TB0CCTL4     /* Timer0_B7 Capture/Compare Control 4
*/
#define TBCCTL5                 TB0CCTL5     /* Timer0_B7 Capture/Compare Control 5
*/
#define TBCCTL6                 TB0CCTL6     /* Timer0_B7 Capture/Compare Control 6
*/
#define TBR                      TB0R        /* Timer0_B7 */
#define TBCCR0                   TB0CCR0     /* Timer0_B7 Capture/Compare 0 */
#define TBCCR1                   TB0CCR1     /* Timer0_B7 Capture/Compare 1 */
#define TBCCR2                   TB0CCR2     /* Timer0_B7 Capture/Compare 2 */
#define TBCCR3                   TB0CCR3     /* Timer0_B7 Capture/Compare 3 */
#define TBCCR4                   TB0CCR4     /* Timer0_B7 Capture/Compare 4 */
#define TBCCR5                   TB0CCR5     /* Timer0_B7 Capture/Compare 5 */
#define TBCCR6                   TB0CCR6     /* Timer0_B7 Capture/Compare 6 */
#define TBEX0                    TB0EX0      /* Timer0_B7 Expansion Register 0 */
#define TBIV                     TB0IV       /* Timer0_B7 Interrupt Vector Word */
#define TIMERB1_VECTOR          TIMER0_B1_VECTOR /* Timer0_B7 CC1-6, TB */
#define TIMERB0_VECTOR          TIMER0_B0_VECTOR /* Timer0_B7 CC0 */

/* TBxCTL Control Bits */
#define TBCLGRP1                 (0x4000)    /* Timer0_B7 Compare latch load group 1
*/
#define TBCLGRP0                 (0x2000)    /* Timer0_B7 Compare latch load group 0
*/
#define CNTL1                    (0x1000)    /* Counter lenght 1 */
#define CNTL0                    (0x0800)    /* Counter lenght 0 */
#define TBSSEL1                  (0x0200)    /* Clock source 1 */
#define TBSSEL0                  (0x0100)    /* Clock source 0 */
#define TBCLR                    (0x0004)    /* Timer0_B7 counter clear */
#define TBIE                     (0x0002)    /* Timer0_B7 interrupt enable */
#define TBIFG                    (0x0001)    /* Timer0_B7 interrupt flag */

#define SHR1                     (0x4000)    /* Timer0_B7 Compare latch load group 1
```

```
*/
#define SHR0                      (0x2000)          /* Timer0_B7 Compare latch load group 0
*/

#define TBSSEL_0                  (0*0x0100u)       /* Clock Source: TBCLK */
#define TBSSEL_1                  (1*0x0100u)       /* Clock Source: ACLK    */
#define TBSSEL_2                  (2*0x0100u)       /* Clock Source: SMCLK */
#define TBSSEL_3                  (3*0x0100u)       /* Clock Source: INCLK */
#define CNTL_0                    (0*0x0800u)       /* Counter lenght: 16 bit */
#define CNTL_1                    (1*0x0800u)       /* Counter lenght: 12 bit */
#define CNTL_2                    (2*0x0800u)       /* Counter lenght: 10 bit */
#define CNTL_3                    (3*0x0800u)       /* Counter lenght:    8 bit */
#define SHR_0                     (0*0x2000u)       /* Timer0_B7 Group: 0 - individually */
#define SHR_1                     (1*0x2000u)       /* Timer0_B7 Group: 1 - 3 groups (1-2, 3-4,
5-6) */
#define SHR_2                     (2*0x2000u)       /* Timer0_B7 Group: 2 - 2 groups (1-3,
4-6)*/
#define SHR_3                     (3*0x2000u)       /* Timer0_B7 Group: 3 - 1 group (all) */
#define TBCLGRP_0                 (0*0x2000u)       /* Timer0_B7 Group: 0 - individually */
#define TBCLGRP_1                 (1*0x2000u)       /* Timer0_B7 Group: 1 - 3 groups (1-2, 3-4,
5-6) */
#define TBCLGRP_2                 (2*0x2000u)       /* Timer0_B7 Group: 2 - 2 groups (1-3,
4-6)*/
#define TBCLGRP_3                 (3*0x2000u)       /* Timer0_B7 Group: 3 - 1 group (all) */
#define TBSSEL__TBCLK             (0*0x100u)        /* Timer0_B7 clock source select: 0 - TBCLK
*/
#define TBSSEL__TACLK             (0*0x100u)        /* Timer0_B7 clock source select: 0 - TBCLK
(legacy) */
#define TBSSEL__ACLK              (1*0x100u)        /* Timer0_B7 clock source select: 1 - ACLK
*/
#define TBSSEL__SMCLK             (2*0x100u)        /* Timer0_B7 clock source select: 2 -
SMCLK */
#define TBSSEL__INCLK             (3*0x100u)        /* Timer0_B7 clock source select: 3 - INCLK
*/
#define CNTL__16                  (0*0x0800u)       /* Counter lenght: 16 bit */
#define CNTL__12                  (1*0x0800u)       /* Counter lenght: 12 bit */
#define CNTL__10                  (2*0x0800u)       /* Counter lenght: 10 bit */
#define CNTL__8                   (3*0x0800u)       /* Counter lenght:    8 bit */

/* Additional Timer B Control Register bits are defined in Timer A */
/* TBxCCTLx Control Bits */
#define CLLD1                     (0x0400)          /* Compare latch load source 1 */
#define CLLD0                     (0x0200)          /* Compare latch load source 0 */

#define SLSHR1                    (0x0400)          /* Compare latch load source 1 */
#define SLSHR0                    (0x0200)          /* Compare latch load source 0 */

#define SLSHR_0                   (0*0x0200u)       /* Compare latch load sourec : 0 -
```

```
immediate */
#define SLSHR_1                 (1*0x0200u)     /* Compare latch load sourec : 1 - TBR
counts to 0 */
#define SLSHR_2                 (2*0x0200u)     /* Compare latch load sourec : 2 - up/down
*/
#define SLSHR_3                 (3*0x0200u)     /* Compare latch load sourec : 3 - TBR
counts to TBCTL0 */

#define CLLD_0                  (0*0x0200u)     /* Compare latch load sourec : 0 -
immediate */
#define CLLD_1                  (1*0x0200u)     /* Compare latch load sourec : 1 - TBR
counts to 0 */
#define CLLD_2                  (2*0x0200u)     /* Compare latch load sourec : 2 - up/down
*/
#define CLLD_3                  (3*0x0200u)     /* Compare latch load sourec : 3 - TBR
counts to TBCTL0 */

/* TBxEX0 Control Bits */
#define TBIDEX0                 (0x0001)        /* Timer0_B7 Input divider expansion Bit: 0
*/
#define TBIDEX1                 (0x0002)        /* Timer0_B7 Input divider expansion Bit: 1
*/
#define TBIDEX2                 (0x0004)        /* Timer0_B7 Input divider expansion Bit: 2
*/

#define TBIDEX_0                (0*0x0001u)     /* Timer0_B7 Input divider expansion : /1
*/
#define TBIDEX_1                (1*0x0001u)     /* Timer0_B7 Input divider expansion : /2
*/
#define TBIDEX_2                (2*0x0001u)     /* Timer0_B7 Input divider expansion : /3
*/
#define TBIDEX_3                (3*0x0001u)     /* Timer0_B7 Input divider expansion : /4
*/
#define TBIDEX_4                (4*0x0001u)     /* Timer0_B7 Input divider expansion : /5
*/
#define TBIDEX_5                (5*0x0001u)     /* Timer0_B7 Input divider expansion : /6
*/
#define TBIDEX_6                (6*0x0001u)     /* Timer0_B7 Input divider expansion : /7
*/
#define TBIDEX_7                (7*0x0001u)     /* Timer0_B7 Input divider expansion : /8
*/
#define TBIDEX__1               (0*0x0001u)     /* Timer0_B7 Input divider expansion : /1
*/
#define TBIDEX__2               (1*0x0001u)     /* Timer0_B7 Input divider expansion : /2
*/
#define TBIDEX__3               (2*0x0001u)     /* Timer0_B7 Input divider expansion : /3
*/
#define TBIDEX__4               (3*0x0001u)     /* Timer0_B7 Input divider expansion : /4
```

```
*/
#define TBIDEX__5                   (4*0x0001u)      /* Timer0_B7 Input divider expansion : /5
*/
#define TBIDEX__6                   (5*0x0001u)      /* Timer0_B7 Input divider expansion : /6
*/
#define TBIDEX__7                   (6*0x0001u)      /* Timer0_B7 Input divider expansion : /7
*/
#define TBIDEX__8                   (7*0x0001u)      /* Timer0_B7 Input divider expansion : /8
*/

/* TB0IV Definitions */
#define TB0IV_NONE                   (0x0000)         /* No Interrupt pending */
#define TB0IV_TBCCR1                (0x0002)         /* TB0CCR1_CCIFG */
#define TB0IV_TBCCR2                (0x0004)         /* TB0CCR2_CCIFG */
#define TB0IV_TBCCR3                (0x0006)         /* TB0CCR3_CCIFG */
#define TB0IV_TBCCR4                (0x0008)         /* TB0CCR4_CCIFG */
#define TB0IV_TBCCR5                (0x000A)         /* TB0CCR5_CCIFG */
#define TB0IV_TBCCR6                (0x000C)         /* TB0CCR6_CCIFG */
#define TB0IV_TBIFG                 (0x000E)         /* TB0IFG */

/* Legacy Defines */
#define TB0IV_TB0CCR1               (0x0002)         /* TB0CCR1_CCIFG */
#define TB0IV_TB0CCR2               (0x0004)         /* TB0CCR2_CCIFG */
#define TB0IV_TB0CCR3               (0x0006)         /* TB0CCR3_CCIFG */
#define TB0IV_TB0CCR4               (0x0008)         /* TB0CCR4_CCIFG */
#define TB0IV_TB0CCR5               (0x000A)         /* TB0CCR5_CCIFG */
#define TB0IV_TB0CCR6               (0x000C)         /* TB0CCR6_CCIFG */
#define TB0IV_TB0IFG                (0x000E)         /* TB0IFG */


/************************************************************
* USCI A0
************************************************************/

/************************************************************
* USCI A1
************************************************************/

/************************************************************
* USCI B0
************************************************************/

/************************************************************
* USCI B1
************************************************************/
/************************************************************
* WATCHDOG TIMER A
************************************************************/
```

```
#define __MSP430_HAS_WDT_A__                    /* Definition to show that Module is
available */
#define __MSP430_BASEADDRESS_WDT_A__ 0x0150
#define WDT_A_BASE                  __MSP430_BASEADDRESS_WDT_A__

SFR_16BIT(WDTCTL);                              /* Watchdog Timer Control */
SFR_8BIT(WDTCTL_L);                             /* Watchdog Timer Control */
SFR_8BIT(WDTCTL_H);                             /* Watchdog Timer Control */
/* The bit names have been prefixed with "WDT" */
/* WDTCTL Control Bits */
#define WDTIS0                  (0x0001)        /* WDT - Timer Interval Select 0 */
#define WDTIS1                  (0x0002)        /* WDT - Timer Interval Select 1 */
#define WDTIS2                  (0x0004)        /* WDT - Timer Interval Select 2 */
#define WDTCNTCL                (0x0008)        /* WDT - Timer Clear */
#define WDTTMSEL                (0x0010)        /* WDT - Timer Mode Select */
#define WDTSSEL0                (0x0020)        /* WDT - Timer Clock Source Select 0 */
#define WDTSSEL1                (0x0040)        /* WDT - Timer Clock Source Select 1 */
#define WDTHOLD                 (0x0080)        /* WDT - Timer hold */

/* WDTCTL Control Bits */
#define WDTIS0_L                (0x0001)        /* WDT - Timer Interval Select 0 */
#define WDTIS1_L                (0x0002)        /* WDT - Timer Interval Select 1 */
#define WDTIS2_L                (0x0004)        /* WDT - Timer Interval Select 2 */
#define WDTCNTCL_L              (0x0008)        /* WDT - Timer Clear */
#define WDTTMSEL_L              (0x0010)        /* WDT - Timer Mode Select */
#define WDTSSEL0_L              (0x0020)        /* WDT - Timer Clock Source Select 0 */
#define WDTSSEL1_L              (0x0040)        /* WDT - Timer Clock Source Select 1 */
#define WDTHOLD_L               (0x0080)        /* WDT - Timer hold */

#define WDTPW                   (0x5A00)

#define WDTIS_0                 (0*0x0001u)     /* WDT - Timer Interval Select: /2G */
#define WDTIS_1                 (1*0x0001u)     /* WDT - Timer Interval Select: /128M */
#define WDTIS_2                 (2*0x0001u)     /* WDT - Timer Interval Select: /8192k */
#define WDTIS_3                 (3*0x0001u)     /* WDT - Timer Interval Select: /512k */
#define WDTIS_4                 (4*0x0001u)     /* WDT - Timer Interval Select: /32k */
#define WDTIS_5                 (5*0x0001u)     /* WDT - Timer Interval Select: /8192 */
#define WDTIS_6                 (6*0x0001u)     /* WDT - Timer Interval Select: /512 */
#define WDTIS_7                 (7*0x0001u)     /* WDT - Timer Interval Select: /64 */
#define WDTIS__2G               (0*0x0001u)     /* WDT - Timer Interval Select: /2G */
#define WDTIS__128M             (1*0x0001u)     /* WDT - Timer Interval Select: /128M */
#define WDTIS__8192K            (2*0x0001u)     /* WDT - Timer Interval Select: /8192k */
#define WDTIS__512K             (3*0x0001u)     /* WDT - Timer Interval Select: /512k */
#define WDTIS__32K              (4*0x0001u)     /* WDT - Timer Interval Select: /32k */
#define WDTIS__8192             (5*0x0001u)     /* WDT - Timer Interval Select: /8192 */
#define WDTIS__512              (6*0x0001u)     /* WDT - Timer Interval Select: /512 */
#define WDTIS__64               (7*0x0001u)     /* WDT - Timer Interval Select: /64 */
```

```
#define WDTSSEL_0                   (0*0x0020u)       /* WDT - Timer Clock Source Select:
SMCLK */
#define WDTSSEL_1                   (1*0x0020u)       /* WDT - Timer Clock Source Select: ACLK
*/
#define WDTSSEL_2                   (2*0x0020u)       /* WDT - Timer Clock Source Select:
VLO_CLK */
#define WDTSSEL_3                   (3*0x0020u)       /* WDT - Timer Clock Source Select:
reserved */
#define WDTSSEL__SMCLK              (0*0x0020u)       /* WDT - Timer Clock Source Select:
SMCLK */
#define WDTSSEL__ACLK               (1*0x0020u)       /* WDT - Timer Clock Source Select: ACLK
*/
#define WDTSSEL__VLO                (2*0x0020u)       /* WDT - Timer Clock Source Select:
VLO_CLK */

/* WDT-interval times [1ms] coded with Bits 0-2 */
/* WDT is clocked by fSMCLK (assumed 1MHz) */
#define WDT_MDLY_32                 (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2)
/* 32ms interval (default) */
#define WDT_MDLY_8                  (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTIS0)
/* 8ms         " */
#define WDT_MDLY_0_5                (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTIS1)
/* 0.5ms      " */
#define WDT_MDLY_0_064              (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTIS1+WDTIS0)
/* 0.064ms " */
/* WDT is clocked by fACLK (assumed 32KHz) */
#define WDT_ADLY_1000               (WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTSSEL0)
/* 1000ms    " */
#define WDT_ADLY_250
(WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTSSEL0+WDTIS0)                /* 250ms      " */
#define WDT_ADLY_16
(WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTSSEL0+WDTIS1)                /* 16ms       " */
#define WDT_ADLY_1_9
(WDTPW+WDTTMSEL+WDTCNTCL+WDTIS2+WDTSSEL0+WDTIS1+WDTIS0)    /* 1.9ms      " */
/* Watchdog mode -> reset after expired time */
/* WDT is clocked by fSMCLK (assumed 1MHz) */
#define WDT_MRST_32                 (WDTPW+WDTCNTCL+WDTIS2)
/* 32ms interval (default) */
#define WDT_MRST_8                  (WDTPW+WDTCNTCL+WDTIS2+WDTIS0)
/* 8ms         " */
#define WDT_MRST_0_5                (WDTPW+WDTCNTCL+WDTIS2+WDTIS1)
/* 0.5ms      " */
#define WDT_MRST_0_064              (WDTPW+WDTCNTCL+WDTIS2+WDTIS1+WDTIS0)
/* 0.064ms " */
/* WDT is clocked by fACLK (assumed 32KHz) */
#define WDT_ARST_1000               (WDTPW+WDTCNTCL+WDTSSEL0+WDTIS2)
/* 1000ms    " */
#define WDT_ARST_250                (WDTPW+WDTCNTCL+WDTSSEL0+WDTIS2+WDTIS0)
```

```
/* 250ms      " */
#define WDT_ARST_16             (WDTPW+WDTCNTCL+WDTSSEL0+WDTIS2+WDTIS1)
/* 16ms       " */
#define WDT_ARST_1_9            (WDTPW+WDTCNTCL+WDTSSEL0+WDTIS2+WDTIS1+WDTIS0)
/* 1.9ms      " */


/************************************************************
* TLV Descriptors
************************************************************/
#define __MSP430_HAS_TLV__                          /* Definition to show that Module is
available */
#define TLV_BASE                    __MSP430_BASEADDRESS_TLV__

#define TLV_CRC_LENGTH              (0x1A01)        /* CRC length of the TLV structure */
#define TLV_CRC_VALUE               (0x1A02)        /* CRC value of the TLV structure */
#define TLV_START                   (0x1A08)        /* Start Address of the TLV structure */
#define TLV_END                     (0x1AFF)        /* End Address of the TLV structure */

#define TLV_LDTAG                   (0x01)          /*    Legacy descriptor (1xx, 2xx, 4xx
families) */
#define TLV_PDTAG                   (0x02)          /*    Peripheral discovery descriptor */
#define TLV_Reserved3               (0x03)          /*    Future usage */
#define TLV_Reserved4               (0x04)          /*    Future usage */
#define TLV_BLANK                   (0x05)          /*    Blank descriptor */
#define TLV_Reserved6               (0x06)          /*    Future usage */
#define TLV_Reserved7               (0x07)          /*    Serial Number */
#define TLV_DIERECORD               (0x08)          /*    Die Record    */
#define TLV_ADCCAL                  (0x11)          /*    ADC12 calibration */
#define TLV_ADC12CAL                (0x11)          /*    ADC12 calibration */
#define TLV_ADC10CAL                (0x13)          /*    ADC10 calibration */
#define TLV_REFCAL                  (0x12)          /*    REF calibration */
#define TLV_TAGEXT                  (0xFE)          /*    Tag extender */
#define TLV_TAGEND                  (0xFF)          //    Tag End of Table

/************************************************************
* Interrupt Vectors (offset from 0xFF80 + 0x10 for Password)
************************************************************/

#pragma diag_suppress 1107
#define VECTOR_NAME(name)               name##_ptr
#define EMIT_PRAGMA(x)                  _Pragma(#x)
#define CREATE_VECTOR(name)             void * const VECTOR_NAME(name) = (void
*)(long)&name
#define PLACE_VECTOR(vector,section)    EMIT_PRAGMA(DATA_SECTION(vector,section))
#define PLACE_INTERRUPT(func)           EMIT_PRAGMA(CODE_SECTION(func,".text:_isr"))
#define ISR_VECTOR(func,offset)         CREATE_VECTOR(func); \
                                        PLACE_VECTOR(VECTOR_NAME(func), offset) \
```

PLACE_INTERRUPT(func)

```
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define RTC_VECTOR                  ".int28"                    /* 0xFFC8 RTC */
#else
#define RTC_VECTOR                  (28 * 1u)                   /* 0xFFC8 RTC */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define LCD_C_VECTOR                ".int29"                    /* 0xFFCA LCD C */
#else
#define LCD_C_VECTOR                (29 * 1u)                   /* 0xFFCA LCD C */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define PORT4_VECTOR                ".int30"                    /* 0xFFCC Port 4 */
#else
#define PORT4_VECTOR                (30 * 1u)                   /* 0xFFCC Port 4 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define PORT3_VECTOR                ".int31"                    /* 0xFFCE Port 3 */
#else
#define PORT3_VECTOR                (31 * 1u)                   /* 0xFFCE Port 3 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER3_A1_VECTOR            ".int32"                    /* 0xFFD0 Timer3_A2
CC1, TA */
#else
#define TIMER3_A1_VECTOR            (32 * 1u)                   /* 0xFFD0 Timer3_A2
CC1, TA */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER3_A0_VECTOR            ".int33"                    /* 0xFFD2 Timer3_A2
CC0 */
#else
#define TIMER3_A0_VECTOR            (33 * 1u)                   /* 0xFFD2 Timer3_A2
CC0 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define PORT2_VECTOR                ".int34"                    /* 0xFFD4 Port 2 */
#else
#define PORT2_VECTOR                (34 * 1u)                   /* 0xFFD4 Port 2 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER2_A1_VECTOR            ".int35"                    /* 0xFFD6 Timer2_A3
CC1, TA */
#else
#define TIMER2_A1_VECTOR            (35 * 1u)                   /* 0xFFD6 Timer2_A3
CC1, TA */
```

```
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER2_A0_VECTOR              ".int36"                        /* 0xFFD8 Timer2_A3
CC0 */
#else
#define TIMER2_A0_VECTOR              (36 * 1u)                       /* 0xFFD8 Timer2_A3
CC0 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define PORT1_VECTOR                  ".int37"                        /* 0xFFDA Port 1 */
#else
#define PORT1_VECTOR                  (37 * 1u)                       /* 0xFFDA Port 1 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER1_A1_VECTOR              ".int38"                        /* 0xFFDC Timer1_A3
CC1-2, TA1 */
#else
#define TIMER1_A1_VECTOR              (38 * 1u)                       /* 0xFFDC Timer1_A3
CC1-2, TA1 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER1_A0_VECTOR              ".int39"                        /* 0xFFDE Timer1_A3
CC0 */
#else
#define TIMER1_A0_VECTOR              (39 * 1u)                       /* 0xFFDE Timer1_A3
CC0 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define DMA_VECTOR                    ".int40"                        /* 0xFFE0 DMA */
#else
#define DMA_VECTOR                    (40 * 1u)                       /* 0xFFE0 DMA */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define USCI_B1_VECTOR                ".int41"                        /* 0xFFE2 USCI B1
Receive/Transmit */
#else
#define USCI_B1_VECTOR                (41 * 1u)                       /* 0xFFE2 USCI B1
Receive/Transmit */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define USCI_A1_VECTOR                ".int42"                        /* 0xFFE4 USCI A1
Receive/Transmit */
#else
#define USCI_A1_VECTOR                (42 * 1u)                       /* 0xFFE4 USCI A1
Receive/Transmit */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER0_A1_VECTOR              ".int43"                        /* 0xFFE6 Timer0_A5
```

```
CC1-4, TA */
#else
#define TIMER0_A1_VECTOR                (43 * 1u)                       /* 0xFFE6 Timer0_A5
CC1-4, TA */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER0_A0_VECTOR                ".int44"                        /* 0xFFE8 Timer0_A5
CC0 */
#else
#define TIMER0_A0_VECTOR                (44 * 1u)                       /* 0xFFE8 Timer0_A5
CC0 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define ADC12_VECTOR                    ".int45"                        /* 0xFFEA ADC */
#else
#define ADC12_VECTOR                    (45 * 1u)                       /* 0xFFEA ADC */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define USCI_B0_VECTOR                  ".int46"                        /* 0xFFEC USCI B0
Receive/Transmit */
#else
#define USCI_B0_VECTOR                  (46 * 1u)                       /* 0xFFEC USCI B0
Receive/Transmit */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define USCI_A0_VECTOR                  ".int47"                        /* 0xFFEE USCI A0
Receive/Transmit */
#else
#define USCI_A0_VECTOR                  (47 * 1u)                       /* 0xFFEE USCI A0
Receive/Transmit */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define ESCAN_IF_VECTOR                 ".int48"                        /* 0xFFF0 Extended Scan
IF */
#else
#define ESCAN_IF_VECTOR                 (48 * 1u)                       /* 0xFFF0 Extended Scan
IF */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define WDT_VECTOR                      ".int49"                        /* 0xFFF2 Watchdog
Timer */
#else
#define WDT_VECTOR                      (49 * 1u)                       /* 0xFFF2 Watchdog
Timer */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER0_B1_VECTOR                ".int50"                        /* 0xFFF4 Timer0_B3
CC1-2, TB */
```

```
#else
#define TIMER0_B1_VECTOR            (50 * 1u)                   /* 0xFFF4 Timer0_B3
CC1-2, TB */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define TIMER0_B0_VECTOR            ".int51"                    /* 0xFFF6 Timer0_B3
CC0 */
#else
#define TIMER0_B0_VECTOR            (51 * 1u)                   /* 0xFFF6 Timer0_B3
CC0 */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define COMP_E_VECTOR               ".int52"                     /* 0xFFF8 Comparator
E */
#else
#define COMP_E_VECTOR               (52 * 1u)                    /* 0xFFF8 Comparator E
*/
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define UNMI_VECTOR                 ".int53"                     /* 0xFFFA User
Non-maskable */
#else
#define UNMI_VECTOR                 (53 * 1u)                    /* 0xFFFA User
Non-maskable */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define SYSNMI_VECTOR               ".int54"                    /* 0xFFFC System
Non-maskable */
#else
#define SYSNMI_VECTOR               (54 * 1u)                   /* 0xFFFC System
Non-maskable */
#endif
#ifdef __ASM_HEADER__ /* Begin #defines for assembler */
#define RESET_VECTOR                ".reset"                    /* 0xFFFE Reset [Highest
Priority] */
#else
#define RESET_VECTOR                (55 * 1u)                   /* 0xFFFE Reset [Highest
Priority] */
#endif

/************************************************************
* End of Modules
************************************************************/

#ifdef __cplusplus
}
#endif /* extern "C" */
```

```
#endif /* #ifndef __MSP430FR6889 */
```