

```

#ifndef __msp430x14x
#define __msp430x14x

/*****
* 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 */

#ifndef __IAR_SYSTEMS_ICC /* 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>

#define LPM0          _BIS_SR(LPM0_bits) /* Enter Low Power Mode 0 */
#define LPM0_EXIT    _BIC_SR(LPM0_bits) /* Exit Low Power Mode 0 */
#define LPM1          _BIS_SR(LPM1_bits) /* Enter Low Power Mode 1 */
#define LPM1_EXIT    _BIC_SR(LPM1_bits) /* Exit Low Power Mode 1 */
#define LPM2          _BIS_SR(LPM2_bits) /* Enter Low Power Mode 2 */

```

```

#define LPM2_EXIT _BIC_SR(LPM2_bits) /* Exit Low Power Mode 2 */
#define LPM3      _BIS_SR(LPM3_bits) /* Enter Low Power Mode 3 */
#define LPM3_EXIT _BIC_SR(LPM3_bits) /* Exit Low Power Mode 3 */
#define LPM4      _BIS_SR(LPM4_bits) /* Enter Low Power Mode 4 */
#define LPM4_EXIT _BIC_SR(LPM4_bits) /* Exit Low Power Mode 4 */
#endif /* End #defines for C */

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

/*****
* 特殊功能寄存器地址和控制位
*****/

/*中断使能 1*/
#define IE1_          0x0000
sfrb IE1             = IE1_;
#define WDTIE        0x01      /*看门狗中断使能*/
#define OFIE         0x02      /*外部晶振故障中断使能*/
#define NMIE         0x10      /*非屏蔽中断使能*/
#define ACCVIE       0x20      /*可屏蔽中断使能/flash 写中断错误*/
#define URXIE0       0x40      /*串口 0 接收中断使能*/
#define UTXIE0       0x80      /*串口 0 发送中断使能*/
/*中断标志 1*/
#define IFG1_        0x0002
sfrb IFG1           = IFG1_;
#define WDTIFG       0x01      /*看门狗中断标志*/
#define OFIFG        0x02      /*外部晶振故障中断标志*/
#define NMIFG        0x10      /*非屏蔽中断标志*/
#define URXIFG0      0x40      /*串口 0 接收中断标志*/
#define UTXIFG0      0x80      /*串口 0 发送中断标志*/
/* 中断模式使能 1 */
#define ME1_         0x0004
sfrb ME1            = ME1_;
#define URXE0        0x40      /* 串口 0 接收中断模式使能 */
#define USPIE0       0x40      /* 同步中断模式使能 */
#define UTXE0        0x80      /* 串口 0 发送中断模式使能 */
/* 中断使能 2 */
#define IE2_         0x0001
sfrb IE2            = IE2_;
#define URXIE1       0x10      /* 串口 1 接收中断使能 */
#define UTXIE1       0x20      /* 串口 1 发送中断使能 */
/* 中断标志 2 */
#define IFG2_        0x0003
sfrb IFG2           = IFG2_;
#define URXIFG1      0x10      /* 串口 1 接收中断标志 */
#define UTXIFG1      0x20      /* 串口 1 发送中断标志 */
/* 中断模式使能 2 */
#define ME2_         0x0005
sfrb ME2            = ME2_;
#define URXE1        0x10      /* 串口 1 接收中断模式使能 */
#define USPIE1       0x10      /* 同步中断模式使能 */
#define UTXE1        0x20      /* 串口 1 发送中断模式使能 */

/*****
* 看门狗定时器的寄存器定义
*****/

#define WDTCTL_      0x0120
sfrw WDTCTL         = WDTCTL_;
#define WDTIS0       0x0001      /*选择 WDTCNT 的四个输出端之一*/
#define WDTIS1       0x0002      /*选择 WDTCNT 的四个输出端之一*/
#define WDTSSSEL     0x0004      /*选择 WDTCNT 的时钟源*/

```

```

#define WDCNTCL          0x0008      /*清除 WDCNT 端: 为 1 时 从 0 开始
计数*/
#define WDTTMSSEL       0x0010      /*选择模式 0: 看门狗模式; 1: 定时
器模式*/
#define WDTNMI          0x0020      /*选择 NMI/RST 引脚功能 0:为 RST; 1:
为 NMI*/
#define WDTNMIES        0x0040      /*WDTNMI=1 时.选择触发延 0:为上升
延 1:为下降延*/
#define WDT HOLD        0x0080      /*停止看门狗定时器工作 0:启动;1:停
止*/

#define WDT PW          0x5A00      /* 写密码:高八位*/

/* SMCLK= 1MHz 定时器模式 */
#define WDT_MDLY_32          WDT PW+WDTTMSSEL+WDCNTCL
/* TSMCLK*2POWER15=32ms 复位状态 */
#define WDT_MDLY_8          WDT PW+WDTTMSSEL+WDCNTCL+WDTIS0
/* TSMCLK*2POWER13=8.192ms " */
#define WDT_MDLY_0_5        WDT PW+WDTTMSSEL+WDCNTCL+WDTIS1
/* TSMCLK*2POWER9=0.512ms " */
#define WDT_MDLY_0_064      WDT PW+WDTTMSSEL+WDCNTCL+WDTIS1+WDTIS0
/* TSMCLK*2POWER6=0.512ms " */
/* ACLK=32.768KHz 定时器模式*/
#define WDT_ADLY_1000       WDT PW+WDTTMSSEL+WDCNTCL+WDTSSSEL
/* TACLK*2POWER15=1000ms " */
#define WDT_ADLY_250       WDT PW+WDTTMSSEL+WDCNTCL+WDTSSSEL+WDTIS0 /*
TACLK*2POWER13=250ms " */
#define WDT_ADLY_16        WDT PW+WDTTMSSEL+WDCNTCL+WDTSSSEL+WDTIS1 /*
TACLK*2POWER9=16ms " */
#define WDT_ADLY_1_9        WDT PW+WDTTMSSEL+WDCNTCL+WDTSSSEL+WDTIS1+WDTIS0 /*
TACLK*2POWER6=1.9ms " */
/* SMCLK=1MHz 看门狗模式 */
#define WDT_MRST_32          WDT PW+WDCNTCL
/* TSMCLK*2POWER15=32ms 复位状态 */
#define WDT_MRST_8          WDT PW+WDCNTCL+WDTIS0
/* TSMCLK*2POWER13=8.192ms " */
#define WDT_MRST_0_5        WDT PW+WDCNTCL+WDTIS1
/* TSMCLK*2POWER9=0.512ms " */
#define WDT_MRST_0_064      WDT PW+WDCNTCL+WDTIS1+WDTIS0
/* TSMCLK*2POWER6=0.512ms " */
/* ACLK=32KHz 看门狗模式 */
#define WDT_ARST_1000       WDT PW+WDCNTCL+WDTSSSEL
/* TACLK*2POWER15=1000ms " */
#define WDT_ARST_250        WDT PW+WDCNTCL+WDTSSSEL+WDTIS0
/* TACLK*2POWER13=250ms " */
#define WDT_ARST_16         WDT PW+WDCNTCL+WDTSSSEL+WDTIS1
/* TACLK*2POWER9=16ms " */
#define WDT_ARST_1_9        WDT PW+WDCNTCL+WDTSSSEL+WDTIS1+WDTIS0
/* TACLK*2POWER6=1.9ms " */

/*****
硬件乘法器的寄存器定义
*****/

#define MPY_                0x0130 /* 无符号乘法 */
sfrw MPY
= MPY_;
#define MPYS_                0x0132 /* 有符号乘法*/

```

```

sfrw  MPYS                = MPYS_;
#define MAC_                0x0134 /* 无符号乘加 */
sfrw  MAC                  = MAC_;
#define MACS_               0x0136 /* 有符号乘加 */
sfrw  MACS                 = MACS_;
#define OP2_                0x0138 /* 第二乘数 */
sfrw  OP2                  = OP2_;
#define RESLO_              0x013A /* 低 6 位结果寄存器 */
sfrw  RESLO                = RESLO_;
#define RESHI_              0x013C /* 高 6 位结果寄存器 */
sfrw  RESHI                = RESHI_;
#define SUMEXT_             0x013E /* 结果扩展寄存器 */
const sfrw SUMEXT         = SUMEXT_;

/*****
* DIGITAL I/O Port1/2 寄存器定义 有中断功能
*****/

#define P1IN_                0x0020 /* P1 输入寄存器 */
const sfrb P1IN           = P1IN_;
#define P1OUT_              0x0021 /* P1 输出寄存器 */
sfrb  P1OUT                = P1OUT_;
#define P1DIR_              0x0022 /* P1 方向选择寄存器 */
sfrb  P1DIR                = P1DIR_;
#define P1IFG_              0x0023 /* P1 中断标志寄存器 */
sfrb  P1IFG                = P1IFG_;
#define P1IES_              0x0024 /* P1 中断边沿选择寄存器 */
sfrb  P1IES                = P1IES_;
#define P1IE_               0x0025 /* P1 中断使能寄存器 */
sfrb  P1IE                 = P1IE_;
#define P1SEL_              0x0026 /* P1 功能选择寄存器 */
sfrb  P1SEL                = P1SEL_;

#define P2IN_                0x0028 /* P2 输入寄存器 */
const sfrb P2IN           = P2IN_;
#define P2OUT_              0x0029 /* P2 输出寄存器 */
sfrb  P2OUT                = P2OUT_;
#define P2DIR_              0x002A /* P2 方向选择寄存器 */
sfrb  P2DIR                = P2DIR_;
#define P2IFG_              0x002B /* P2 中断标志寄存器 */
sfrb  P2IFG                = P2IFG_;
#define P2IES_              0x002C /* P2 中断边沿选择寄存器 */
sfrb  P2IES                = P2IES_;
#define P2IE_               0x002D /* P2 中断使能寄存器 */
sfrb  P2IE                 = P2IE_;
#define P2SEL_              0x002E /* P2 功能选择寄存器 */
sfrb  P2SEL                = P2SEL_;

/*****
* DIGITAL I/O Port3/4 寄存器定义 无中断功能
*****/

#define P3IN_                0x0018 /* P3 输入寄存器 */
const sfrb P3IN           = P3IN_;
#define P3OUT_              0x0019 /* P3 输出寄存器 */
sfrb  P3OUT                = P3OUT_;
#define P3DIR_              0x001A /* P3 方向选择寄存器 */
sfrb  P3DIR                = P3DIR_;
#define P3SEL_              0x001B /* P3 功能选择寄存器 */
sfrb  P3SEL                = P3SEL_;

#define P4IN_                0x001C /* P4 输入寄存器 */
const sfrb P4IN           = P4IN_;

```

```

#define P4OUT_          0x001D /* P4 输出寄存器 */
sfrb P4OUT            = P4OUT_;
#define P4DIR_         0x001E /* P4 方向选择寄存器 */
sfrb P4DIR            = P4DIR_;
#define P4SEL_         0x001F /* P4 功能选择寄存器 */
sfrb P4SEL            = P4SEL_;

/*****
* DIGITAL I/O Port5/6      I/O 口寄存器定义 PORT5 和 6 无中断功能
*****/

#define P5IN_          0x0030 /* P5 输入寄存器 */
const sfrb P5IN      = P5IN_;
#define P5OUT_         0x0031 /* P5 输出寄存器*/
sfrb P5OUT            = P5OUT_;
#define P5DIR_         0x0032 /* P5 方向选择寄存器*/
sfrb P5DIR            = P5DIR_;
#define P5SEL_         0x0033 /* P5 功能选择寄存器*/
sfrb P5SEL            = P5SEL_;

#define P6IN_          0x0034 /* P6 输入寄存器 */
const sfrb P6IN      = P6IN_;
#define P6OUT_         0x0035 /* P6 输出寄存器*/
sfrb P6OUT            = P6OUT_;
#define P6DIR_         0x0036 /* P6 方向选择寄存器*/
sfrb P6DIR            = P6DIR_;
#define P6SEL_         0x0037 /* P6 功能选择寄存器*/
sfrb P6SEL            = P6SEL_;

/*****
* USART  串口寄存器"UCTL","UTCTL","URCTL"定义的各个位 可串口 1 串口 2 公用
*****/
/* UCTL  串口控制寄存器*/
#define PENA           0x80 /*校验允许位*/
#define PEV            0x40 /*偶校验 为 0 时为奇校验*/
#define SPB            0x20 /*停止位为 2 为 0 时停止位为 1*/
#define CHAR           0x10 /*数据位为 8 位 为 0 时数据位为 7 位*/
#define LISTEN        0x08 /*自环模式(发数据同时在把发的数据接收回
来)*/
#define SYNC           0x04 /*同步模式 为 0 异步模式*/
#define MM             0x02 /*为 1 时地址位多机协议(异步) 主机模式(同
步);为 0 时线路空闲多机协议(异步) 从机模式(同步)*/
#define SWRST         0x01 /*控制位*/

/* UTCTL 串口发送控制寄存器*/
#define CKPH           0x80 /*时钟相位控制位(只同步方式用)为 1 时时钟
UCLK 延时半个周期*/
#define CKPL           0x40 /*时钟极性控制位 为 1 时异步与 UCLK 相反;
同步下降延有效*/
#define SSEL1          0x20 /*时钟源选择位:与 SSEL0 组合为 0,1,2,3 四种
方式*/
#define SSEL0          0x10 /*"0"选择外部时钟,"1"选择辅助时钟,"2","3"选
择系统子时钟 */
#define URXSE         0x08 /*接收触发延控制位(只在异步方式下用)*/
#define TXWAKE        0x04 /*多处理器通信传送控制位(只在异步方式
下用)*/
#define STC            0x02 /*外部引脚 STE 选择位 为 0 时为 4 线模式 为
1 时为 3 线模式*/
#define TXEPT         0x01 /*发送器空标志*/

/* URCTL 串口接收控制寄存器 同步模式下只用两位:FE 和 OE*/
#define FE             0x80 /*帧错标志*/

```

```

#define PE                0x40      /*校验错标志位*/
#define OE                0x20      /*溢出标志位*/
#define BRK              0x10      /*打断检测位*/
#define URXEIE           0x08      /*接收出错中断允许位*/
#define URXWIE           0x04      /*接收唤醒中断允许位*/
#define RXWAKE           0x02      /*接收唤醒检测位*/
#define RXERR            0x01      /*接收错误标志位*/

/*****
* USART0                串口0寄存器定义
*****/

#define U0CTL_           0x0070 /* 串口0基本控制寄存器 */
sfrb U0CTL              = U0CTL_;
#define U0TCTL_         0x0071 /* 串口0发送控制寄存器 */
sfrb U0TCTL            = U0TCTL_;
#define U0RCTL_         0x0072 /* 串口0接收控制寄存器 */
sfrb U0RCTL            = U0RCTL_;
#define U0MCTL_         0x0073 /* 波特率调整寄存器 */
sfrb U0MCTL            = U0MCTL_;
#define U0BR0_          0x0074 /* 波特率选择寄存器0 */
sfrb U0BR0             = U0BR0_;
#define U0BR1_          0x0075 /* 波特率选择寄存器1 */
sfrb U0BR1             = U0BR1_;
#define U0RXBUF_        0x0076 /* 接收缓存寄存器 */
const sfrb U0RXBUF     = U0RXBUF_;
#define U0TXBUF_        0x0077 /* 发送缓存寄存器 */
sfrb U0TXBUF           = U0TXBUF_;

/* 改变的寄存器名定义 */

#define UCTL0_           0x0070 /* UART0 Control */
sfrb UCTL0             = UCTL0_;
#define UTCTL0_         0x0071 /* UART0 Transmit Control */
sfrb UTCTL0           = UTCTL0_;
#define URCTL0_         0x0072 /* UART0 Receive Control */
sfrb URCTL0           = URCTL0_;
#define UMCTL0_         0x0073 /* UART0 Modulation Control */
sfrb UMCTL0           = UMCTL0_;
#define UBR00_          0x0074 /* UART0 Baud Rate 0 */
sfrb UBR00            = UBR00_;
#define UBR10_          0x0075 /* UART0 Baud Rate 1 */
sfrb UBR10            = UBR10_;
#define RXBUF0_         0x0076 /* UART0 Receive Buffer */
const sfrb RXBUF0     = RXBUF0_;
#define TXBUF0_         0x0077 /* UART0 Transmit Buffer */
sfrb TXBUF0           = TXBUF0_;

#define UCTL_0_         0x0070 /* UART0 Control */
sfrb UCTL_0           = UCTL_0_;
#define UTCTL_0_        0x0071 /* UART0 Transmit Control */
sfrb UTCTL_0         = UTCTL_0_;
#define URCTL_0_        0x0072 /* UART0 Receive Control */
sfrb URCTL_0         = URCTL_0_;
#define UMCTL_0_        0x0073 /* UART0 Modulation Control */
sfrb UMCTL_0         = UMCTL_0_;
#define UBR0_0_         0x0074 /* UART0 Baud Rate 0 */
sfrb UBR0_0          = UBR0_0_;
#define UBR1_0_         0x0075 /* UART0 Baud Rate 1 */
sfrb UBR1_0          = UBR1_0_;
#define RXBUF_0_        0x0076 /* UART0 Receive Buffer */
const sfrb RXBUF_0   = RXBUF_0_;
#define TXBUF_0_        0x0077 /* UART0 Transmit Buffer */
sfrb TXBUF_0         = TXBUF_0_

```

```

sfrb   TXBUF_0           = TXBUF_0_;

/*****
* USART 1           串口 1 寄存器定义
*****/

#define U1CTL_           0x0078 /* 串口 1 基本控制寄存器 */
sfrb   U1CTL            = U1CTL_;
#define U1TCTL_          0x0079 /* 串口 1 发送控制寄存器 */
sfrb   U1TCTL           = U1TCTL_;
#define U1RCTL_          0x007A /* 串口 1 接收控制寄存器 */
sfrb   U1RCTL           = U1RCTL_;
#define U1MCTL_          0x007B /* 波特率调整控制寄存器 */
sfrb   U1MCTL           = U1MCTL_;
#define U1BR0_           0x007C /* 波特率选择寄存器 0 */
sfrb   U1BR0            = U1BR0_;
#define U1BR1_           0x007D /* 波特率选择寄存器 1 */
sfrb   U1BR1            = U1BR1_;
#define U1RXBUF_         0x007E /* 接收缓存 */
const sfrb U1RXBUF      = U1RXBUF_;
#define U1TXBUF_         0x007F /* 发送缓存 */
sfrb   U1TXBUF          = U1TXBUF_;

/* 改变的寄存器名定义 */

#define UCTL1_           0x0078 /* UART 1 Control */
sfrb   UCTL1            = UCTL1_;
#define UTCTL1_          0x0079 /* UART 1 Transmit Control */
sfrb   UTCTL1           = UTCTL1_;
#define URCTL1_          0x007A /* UART 1 Receive Control */
sfrb   URCTL1           = URCTL1_;
#define UMCTL1_          0x007B /* UART 1 Modulation Control */
sfrb   UMCTL1           = UMCTL1_;
#define UBR01_           0x007C /* UART 1 Baud Rate 0 */
sfrb   UBR01            = UBR01_;
#define UBR11_           0x007D /* UART 1 Baud Rate 1 */
sfrb   UBR11            = UBR11_;
#define RXBUF1_          0x007E /* UART 1 Receive Buffer */
const sfrb RXBUF1      = RXBUF1_;
#define TXBUF1_          0x007F /* UART 1 Transmit Buffer */
sfrb   TXBUF1           = TXBUF1_;

#define UCTL_1_          0x0078 /* UART 1 Control */
sfrb   UCTL_1           = UCTL_1_;
#define UTCTL_1_         0x0079 /* UART 1 Transmit Control */
sfrb   UTCTL_1          = UTCTL_1_;
#define URCTL_1_         0x007A /* UART 1 Receive Control */
sfrb   URCTL_1          = URCTL_1_;
#define UMCTL_1_         0x007B /* UART 1 Modulation Control */
sfrb   UMCTL_1          = UMCTL_1_;
#define UBR0_1_          0x007C /* UART 1 Baud Rate 0 */
sfrb   UBR0_1           = UBR0_1_;
#define UBR1_1_          0x007D /* UART 1 Baud Rate 1 */
sfrb   UBR1_1           = UBR1_1_;
#define RXBUF_1_         0x007E /* UART 1 Receive Buffer */
const sfrb RXBUF_1      = RXBUF_1_;
#define TXBUF_1_         0x007F /* UART 1 Transmit Buffer */
sfrb   TXBUF_1          = TXBUF_1_;

/*****
* Timer A           定时器 A 寄存器定义
*****/

```

```

#define TAIV_          0x012E /* Timer A 中断向量寄存器 */
sfrw  TAIV           = TAIV_;
#define TACTL_        0x0160 /* Timer A 控制寄存器 */
sfrw  TACTL         = TACTL_;
#define TACCTL0_      0x0162 /* Timer A 捕获/比较控制寄存器 0 */
sfrw  TACCTL0       = TACCTL0_;
#define TACCTL1_      0x0164 /* Timer A 捕获/比较控制寄存器 1 */
sfrw  TACCTL1       = TACCTL1_;
#define TACCTL2_      0x0166 /* Timer A 捕获/比较控制寄存器 2 */
sfrw  TACCTL2       = TACCTL2_;
#define TAR_          0x0170 /* Timer A 16 位计数器内容*/
sfrw  TAR            = TAR_;
#define TACCR0_       0x0172 /* Timer A 捕获/比较寄存器 0 */
sfrw  TACCR0        = TACCR0_;
#define TACCR1_       0x0174 /* Timer A 捕获/比较寄存器 1 */
sfrw  TACCR1        = TACCR1_;
#define TACCR2_       0x0176 /* Timer A 捕获/比较寄存器 2 */
sfrw  TACCR2        = TACCR2_;

/* 改变的寄存器名定义 */
#define CCTL0_        0x0162 /* Timer A Capture/Compare Control 0 */
sfrw  CCTL0         = CCTL0_;
#define CCTL1_        0x0164 /* Timer A Capture/Compare Control 1 */
sfrw  CCTL1         = CCTL1_;
#define CCTL2_        0x0166 /* Timer A Capture/Compare Control 2 */
sfrw  CCTL2         = CCTL2_;
#define CCR0_         0x0172 /* Timer A Capture/Compare 0 */
sfrw  CCR0          = CCR0_;
#define CCR1_         0x0174 /* Timer A Capture/Compare 1 */
sfrw  CCR1          = CCR1_;
#define CCR2_         0x0176 /* Timer A Capture/Compare 2 */
sfrw  CCR2          = CCR2_;

/*TACTL 控制寄存器 16 个位寄存器定义*/
#define TASSEL2       0x0400 /* 未用 */
#define TASSEL1       0x0200 /* 时钟输入源控制位 1 */
#define TASSEL0       0x0100 /* 时钟输入源控制位 0 */
#define ID1           0x0080 /* 分频系数选择位 1 */
#define ID0           0x0040 /* 分频系数选择位 0 */
#define MC1           0x0020 /* 计数模式控制位 1 */
#define MC0           0x0010 /* 计数模式控制位 0 */
#define TACLR         0x0004 /* 置 1 位清除定时器 */
#define TAIE          0x0002 /* 定时器中断允许 */
#define TAIFG         0x0001 /* 定时器中断标志 */

#define MC_0          00*0x10 /* 停止模式 */
#define MC_1          01*0x10 /* 增计数模式 */
#define MC_2          02*0x10 /* 连续计数模式 */
#define MC_3          03*0x10 /* 增/减计数模式 */
#define ID_0          00*0x40 /* 直通 */
#define ID_1          01*0x40 /* 2 分频 */
#define ID_2          02*0x40 /* 4 分频 */
#define ID_3          03*0x40 /* 8 分频 */
#define TASSEL_0      00*0x100 /* 时钟源为 TACLK */
#define TASSEL_1      01*0x100 /* 时钟源为 ACLK */
#define TASSEL_2      02*0x100 /* 时钟源为 SMCLK */
#define TASSEL_3      03*0x100 /* 时钟源为 INCLK */

/* Timer A,Timer B 可公用 捕获/比较控制寄存器 X */
#define CM1           0x8000 /* 捕获模式选择位 1 */
#define CM0           0x4000 /* 捕获模式选择位 0 */
#define CCIS1         0x2000 /* 捕获输入信号源选择位 1 */
#define CCIS0         0x1000 /* 捕获输入信号源选择位 0 */

```



```

#define SCS                0x0800 /* 信号同步位 0:异步捕获;1:同步捕获 */
#define SCCI                0x0400 /* 锁存输入信号 */
#define CAP                 0x0100 /* 模式选择: 0:比较模式;1:捕获模式 */
#define OUTMOD2            0x0080 /* 输出模式选择位 2 */
#define OUTMOD1            0x0040 /* 输出模式选择位 1 */
#define OUTMOD0            0x0020 /* 输出模式选择位 0 */
#define CCIE                0x0010 /* 中断允许位 */
#define CCI                 0x0008 /* 读出输入信号源位 ccis0\1 */
#define OUT                 0x0004 /* 输出信号(选择输出模式 0) */
#define COV                 0x0002 /* 捕获溢出标志 */
#define CCIFG               0x0001 /* 中断标志 */

#define OUTMOD_0            0*0x20 /* 输出模式 */
#define OUTMOD_1            1*0x20 /* 置位模式 */
#define OUTMOD_2            2*0x20 /* 翻转/复位模式 */
#define OUTMOD_3            3*0x20 /* 置位/复位模式 */
#define OUTMOD_4            4*0x20 /* 翻转模式 */
#define OUTMOD_5            5*0x20 /* 复位模式 */
#define OUTMOD_6            6*0x20 /* 翻转/置位模式 */
#define OUTMOD_7            7*0x20 /* 复位/置位模式 */
#define CCIS_0              0*0x1000 /* 选择 CCIXA 为捕获事件的输入信号源 */
#define CCIS_1              1*0x1000 /* 选择 CCIXB 为捕获事件的输入信号源 */
#define CCIS_2              2*0x1000 /* 选择 GND 为捕获事件的输入信号源 */
#define CCIS_3              3*0x1000 /* 选择 VCC 为捕获事件的输入信号源 */
#define CM_0                0*0x4000 /* 禁止捕获模式 */
#define CM_1                1*0x4000 /* 上升沿捕获模式 */
#define CM_2                2*0x4000 /* 下降沿捕获模式 */
#define CM_3                3*0x4000 /* 上升沿和下降沿都捕获模式 */

/*****
* Timer B                定时器 B 寄存器定义
*****/

#define TBIV_                0x011E /* 中断向量寄存器:BIT1-BIT3 有效 */
sfrw TBIV_                  = TBIV_;
#define TBCTL_               0x0180 /* 定时器 B 控制寄存器:全部控制都集中在这 */
sfrw TBCTL_                 = TBCTL_;
#define TBCCTL0_             0x0182 /* 定时器 B 捕获/比较控制寄存器 0 */
sfrw TBCCTL0_              = TBCCTL0_;
#define TBCCTL1_             0x0184 /* 定时器 B 捕获/比较控制寄存器 1 */
sfrw TBCCTL1_              = TBCCTL1_;
#define TBCCTL2_             0x0186 /* 定时器 B 捕获/比较控制寄存器 2 */
sfrw TBCCTL2_              = TBCCTL2_;
#define TBCCTL3_             0x0188 /* 定时器 B 捕获/比较控制寄存器 3 */
sfrw TBCCTL3_              = TBCCTL3_;
#define TBCCTL4_             0x018A /* 定时器 B 捕获/比较控制寄存器 4 */
sfrw TBCCTL4_              = TBCCTL4_;
#define TBCCTL5_             0x018C /* 定时器 B 捕获/比较控制寄存器 5 */
sfrw TBCCTL5_              = TBCCTL5_;
#define TBCCTL6_             0x018E /* 定时器 B 捕获/比较控制寄存器 6 */
sfrw TBCCTL6_              = TBCCTL6_;
#define TBR_                 0x0190 /* 计数器 */
sfrw TBR_                   = TBR_;
#define TBCCR0_              0x0192 /* 定时器 B 捕获/比较寄存器 0 */
sfrw TBCCR0_                = TBCCR0_;
#define TBCCR1_              0x0194 /* 定时器 B 捕获/比较寄存器 1 */
sfrw TBCCR1_                = TBCCR1_;
#define TBCCR2_              0x0196 /* 定时器 B 捕获/比较寄存器 2 */
sfrw TBCCR2_                = TBCCR2_;
#define TBCCR3_              0x0198 /* 定时器 B 捕获/比较寄存器 3 */
sfrw TBCCR3_                = TBCCR3_;
#define TBCCR4_              0x019A /* 定时器 B 捕获/比较寄存器 4 */
sfrw TBCCR4_                = TBCCR4_;

```

```

#define TBCCR5_          0x019C /* 定时器 B 捕获/比较寄存器 5 */
sfrw TBCCR5            = TBCCR5_;
#define TBCCR6_          0x019E /* 定时器 B 捕获/比较寄存器 6 */
sfrw TBCCR6            = TBCCR6_;
/* 定时器 B 控制寄存器:全部控制都集中在这 */
#define SHR1             0x4000 /* 装载比较锁存器控制位 1 :受 TBCCTLx 中的
CCLDx 位控制 */
#define SHR0             0x2000 /* 装载比较锁存器控制位 0 :受 TBCCTLx 中的
CCLDx 位控制*/
#define TBCLGRP1         0x4000 /* 装载比较锁存器控制位 1 :受 TBCCTLx 中的
CCLDx 位控制*/
#define TBCLGRP0         0x2000 /* 装载比较锁存器控制位 0 :受 TBCCTLx 中的
CCLDx 位控制*/
#define CNTL1            0x1000 /* 定时器位数长度控制位 1 */
#define CNTL0            0x0800 /* 定时器位数长度控制位 0 */
#define TBSSEL2         0x0400 /* 未用 */
#define TBSSEL1         0x0200 /* 时钟输入源控制位 1 */
#define TBSSEL0         0x0100 /* 时钟输入源控制位 0 */
#define TBCLR           0x0004 /* 置 1 清除定时器 */
#define TBIE            0x0002 /* 中断允许 */
#define TBIFG           0x0001 /* 中断标志 */

#define TBSSEL_0        0*0x0100 /* 时钟源为:TBCLK */
#define TBSSEL_1        1*0x0100 /* 时钟源为:ACLK */
#define TBSSEL_2        2*0x0100 /* 时钟源为:SMCLK */
#define TBSSEL_3        3*0x0100 /* 时钟源为:INCLK */
#define CNTL_0          0*0x0800 /* 16 位计数模式*/
#define CNTL_1          1*0x0800 /* 12 位计数模式*/
#define CNTL_2          2*0x0800 /* 10 位计数模式 */
#define CNTL_3          3*0x0800 /* 8 位计数模式 */
#define SHR_0           0*0x2000 /* 单独装载(初始值) */
#define SHR_1           1*0x2000 /* 分三组装载: 1 - 3 groups (1-2, 3-4, 5-6) */
#define SHR_2           2*0x2000 /* 分二组装载: 2 - 2 groups (1-3, 4-6)*/
#define SHR_3           3*0x2000 /* 不分组装载: 3 - 1 group (all) */
#define TBCLGRP_0      0*0x2000 /* 单独装载(初始值) */
#define TBCLGRP_1      1*0x2000 /* 分三组装载: 1 - 3 groups (1-2, 3-4, 5-6) */
#define TBCLGRP_2      2*0x2000 /* 分二组装载: 2 - 2 groups (1-3, 4-6)*/
#define TBCLGRP_3      3*0x2000 /* 不分组装载: 3 - 1 group (all) */

/* Additional Timer B Control Register bits are defined in Timer A */

#define SLSHR1          0x0400 /* Compare latch load source 1 */
#define SLSHR0          0x0200 /* Compare latch load source 0 */
#define CLLD1           0x0400 /* 定义比较锁存器 TBCLx 的装载方式控制位 1 */
#define CLLD0           0x0200 /* 定义比较锁存器 TBCLx 的装载方式控制位 0 */

#define SLSHR_0         0*0x0200 /* 立即装载 */
#define SLSHR_1         1*0x0200 /* TBR 计数到 0 时装载 */
#define SLSHR_2         2*0x0200 /* 在增减模式下,计数到 TBCLx 或 0 时装载;
在连续计数模式下,计数到 0 时装载 */
#define SLSHR_3         3*0x0200 /* 当计数到 TBCL0 时装载*/

#define CLLD_0          0*0x0200 /* 立即装载 */
#define CLLD_1          1*0x0200 /* TBR 计数到 0 时装载 */
#define CLLD_2          2*0x0200 /* 在增减模式下,计数到 TBCLx 或 0 时装载;
在连续计数模式下,计数到 0 时装载 */
#define CLLD_3          3*0x0200 /* 当计数到 TBCL0 时装载 */

/*****
* Basic Clock Module
*****/

#define DCOCTL_         0x0056 /* DCO 时钟频率控制寄存器 :复位后的值位

```

```

060h*/
sfrb DCOCTL = DCOCTL_;
#define BCSCTL1_ 0x0057 /* 系统时钟控制寄存器 1:复位后的值位 084h*/
sfrb BCSCTL1 = BCSCTL1_;
#define BCSCTL2_ 0x0058 /* 系统时钟控制寄存器 2:复位后的值位 000h*/
sfrb BCSCTL2 = BCSCTL2_;
/* DCO 时钟频率控制寄存器 */
#define MOD0 0x01 /* DCO 插入周期控制位 0 */
#define MOD1 0x02 /* DCO 插入周期控制位 1 */
#define MOD2 0x04 /* DCO 插入周期控制位 2 */
#define MOD3 0x08 /* DCO 插入周期控制位 3 */
#define MOD4 0x10 /* DCO 插入周期控制位 4 */
#define DCO0 0x20 /* 8 种频率控制位 0 */
#define DCO1 0x40 /* 8 种频率控制位 1 */
#define DCO2 0x80 /* 8 种频率控制位 2 */
/* 系统时钟控制寄存器 1:复位后的值位 084h*/
#define RSEL0 0x01 /* 选择内部电阻控制位 0 */
#define RSEL1 0x02 /* 选择内部电阻控制位 1 */
#define RSEL2 0x04 /* 选择内部电阻控制位 2 */
#define XT5V 0x08 /* 必须为 0 */
#define DIVA0 0x10 /* ACLK 分频系数控制位 0 */
#define DIVA1 0x20 /* ACLK 分频系数控制位 1 */
#define XTS 0x40 /* LFXT1 工作模式控制位 0:低频模式. / 1: 高频模式. */
#define XT2OFF 0x80 /* XT2CLK 使能控制位 0:开启; 1:关闭 */

#define DIVA_0 0x00 /* ACLK 分频系数为: 1 */
#define DIVA_1 0x10 /* ACLK 分频系数为: 2 */
#define DIVA_2 0x20 /* ACLK 分频系数为: 4 */
#define DIVA_3 0x30 /* ACLK 分频系数为: 8 */
/* 系统时钟控制寄存器 2:复位后的值位 000h*/
#define DCOR 0x01 /* 内外电阻选择控制位 */
#define DIVS0 0x02 /* SMCLK 分频控制位 0 */
#define DIVS1 0x04 /* SMCLK 分频控制位 1 */
#define SELS 0x08 /* SMCLK 时钟源选择位 t 0:DCOCLK /
1:XT2CLK/LFXTCLK */
#define DIVM0 0x10 /* MCLK 分频控制位 0 */
#define DIVM1 0x20 /* MCLK 分频控制位 1 */
#define SELM0 0x40 /* MCLK 时钟输入源选择位 0 */
#define SELM1 0x80 /* MCLK 时钟输入源选择位 1 */

#define DIVS_0 0x00 /* SMCLK 分频系数为: 1 */
#define DIVS_1 0x02 /* SMCLK 分频系数为: 2 */
#define DIVS_2 0x04 /* SMCLK 分频系数为: 4 */
#define DIVS_3 0x06 /* SMCLK 分频系数为: 8 */

#define DIVM_0 0x00 /* MCLK 分频系数为: 1 */
#define DIVM_1 0x10 /* MCLK 分频系数为: 2 */
#define DIVM_2 0x20 /* MCLK 分频系数为: 4 */
#define DIVM_3 0x30 /* MCLK 分频系数为: 8 */

#define SELM_0 0x00 /* MCLK 时钟输入源: DCOCLK */
#define SELM_1 0x40 /* MCLK 时钟输入源: DCOCLK */
#define SELM_2 0x80 /* MCLK 时钟输入源: XT2CLK/LFXTCLK */
#define SELM_3 0xC0 /* MCLK 时钟输入源: LFXTCLK */

/*****
* Flash Memory FLASH 操作寄存器定义
*****/

#define FCTL1_ 0x0128 /* FLASH 控制寄存器 1:控制编程、擦除 */
sfrw FCTL1 = FCTL1_;
#define FCTL2_ 0x012A /* FLASH 控制寄存器 2 : 控制时钟分频*/

```

```

sfrw   FCTL2           = FCTL2_;
#define FCTL3_         0x012C /* FLASH 控制寄存器 3: 状态标志 */
sfrw   FCTL3           = FCTL3_;

#define FRKEY          0x9600 /* 读 FLASH 密码 */
#define FWKEY          0xA500 /* 写 FLASH 密码 */
#define FXKEY          0x3300 /* for use with XOR instruction */
/* FLASH 控制寄存器 1:控制编程、擦除 */
#define ERASE          0x0002 /* 擦除段使能 */
#define MERAS          0x0004 /* 主存擦除使能 */
#define WRT            0x0040 /* 编程使能 */
#define BLKWRT         0x0080 /* 段编程使能 */
/* FLASH 控制寄存器 2 : 控制时钟分频*/
#define FN_0           0x0000 /*直通 */
#define FN_1           0x0001 /*2 分频 */
#define FN_2           0x0002 /*3 分频*/
#define FN_3           0x0003 /*4 分频 */
#define FN_4           0x0004 /*5 分频 */
#define FN_5           0x0005 /*6 分频*/
#define FN_6           0x0006 /*7 分频 */
#define FN_7           0x0007 /*8 分频*/
#define FN_8           0x0008 /*9 分频 */
#define FN_9           0x0009 /*10 分频 */
#define FN_10          0x000A /*11 分频*/
#define FN_11          0x000B /*12 分频 */
#define FN_12          0x000C /*13 分频*/
#define FN_13          0x000D /*14 分频 */
#define FN_14          0x000E /*15 分频 */
#define FN_15          0x000F /*16 分频*/
#define FN_16          0x0010 /*17 分频 */
#define FN_17          0x0011 /*18 分频*/
#define FN_18          0x0012 /*19 分频 */
#define FN_19          0x0013 /*20 分频 */
#define FN_20          0x0014 /*21 分频*/
#define FN_21          0x0015 /*22 分频 */
#define FN_22          0x0016 /*23 分频*/
#define FN_23          0x0017 /*24 分频 */
#define FN_24          0x0018 /*25 分频 */
#define FN_25          0x0019 /*26 分频*/
#define FN_26          0x001A /*27 分频 */
#define FN_27          0x001B /*28 分频*/
#define FN_28          0x001C /*29 分频 */
#define FN_29          0x001D /*30 分频 */
#define FN_30          0x001E /*31 分频*/
#define FN_31          0x001F /*32 分频 */
#define FN_32          0x0020 /*33 分频*/
#define FN_33          0x0021 /*34 分频 */
#define FN_34          0x0022 /*35 分频 */
#define FN_35          0x0023 /*36 分频*/
#define FN_36          0x0024 /*37 分频 */
#define FN_37          0x0025 /*38 分频*/
#define FN_38          0x0026 /*39 分频 */
#define FN_39          0x0027 /*40 分频 */
#define FN_40          0x0028 /*41 分频*/
#define FN_41          0x0029 /*42 分频 */
#define FN_42          0x002A /*43 分频*/
#define FN_43          0x002B /*44 分频 */
#define FN_44          0x002C /*45 分频 */
#define FN_45          0x002D /*46 分频*/
#define FN_46          0x002E /*47 分频 */
#define FN_47          0x002F /*48 分频*/
#define FN_48          0x0030 /*49 分频 */
#define FN_49          0x0031 /*50 分频 */

```

```

#define FN_50          0x0032 /*51 分频*/
#define FN_51          0x0033 /*52 分频 */
#define FN_52          0x0034 /*53 分频*/
#define FN_53          0x0035 /*54 分频 */
#define FN_54          0x0036 /*55 分频 */
#define FN_55          0x0037 /*56 分频*/
#define FN_56          0x0038 /*57 分频 */
#define FN_57          0x0039 /*58 分频*/
#define FN_58          0x003A /*59 分频 */
#define FN_59          0x003B /*60 分频 */
#define FN_60          0x003C /*61 分频*/
#define FN_61          0x003D /*62 分频 */
#define FN_62          0x003E /*63 分频*/
#define FN_63          0x003F /*64 分频 */
#define FSSEL_0        0x0000 /* Flash 时钟选择: ACLK */
#define FSSEL_1        0x0040 /* Flash 时钟选择: MCLK */
#define FSSEL_2        0x0080 /* Flash 时钟选择: SMCLK */
#define FSSEL_3        0x00C0 /* Flash 时钟选择: SMCLK */
/* FLASH 控制寄存器 3: 状态标志 */
#define BUSY           0x0001 /* Flash 忙标志*/
#define KEYV           0x0002 /* Flash 安全键值出错标志 */
#define ACCVIFG        0x0004 /* Flash 非法访问中断标志*/
#define WAIT           0x0008 /* 等待指示信号位*/
#define LOCK           0x0010 /* 锁定位 */
#define EMEX           0x0020 /* 紧急退出位 */

/*****
* Comparator A          比较器 A 寄存器定义
*****/

#define CACTL1_        0x0059 /* 比较器 A 控制寄存器 1 */
sfrb CACTL1          = CACTL1_;
#define CACTL2_        0x005A /* 比较器 A 控制寄存器 2 */
sfrb CACTL2          = CACTL2_;
#define CAPD_          0x005B /*比较器 A 端口禁止寄存器*/
sfrb CAPD            = CAPD_;
/* 比较器 A 控制寄存器 1 */
#define CAIFG          0x01 /*比较器 A 中断标志*/
#define CAIE           0x02 /* 比较器 A 中断使能 */
#define CAIES          0x04 /* 比较器 A 中断边沿触发选择 0:上升延 1:下降延 */
#define CAON           0x08 /* 比较器电源开关*/
#define CAREF0         0x10 /* 选择参考源位 0 */
#define CAREF1         0x20 /* 选择参考源位 1 */
#define CARSEL         0x40 /* 选择内部参考源加到比较器的正端或负端 */
#define CAEX           0x80 /* 交换比较器的输入端 */

#define CAREF_0        0x00 /* 选择参考源 0 : Off 使用外部参考源*/
#define CAREF_1        0x10 /* 选择参考源 1 : 0.25*Vcc 为参考源 */
#define CAREF_2        0x20 /* 选择参考源 2 : 0.5*Vcc 为参考源 */
#define CAREF_3        0x30 /* 选择参考源 3 : Vt*/
/* 比较器 A 控制寄存器 2 */
#define CAOUT          0x01 /* 比较器输出 */
#define CAF            0x02 /* 选择比较器是否经过 RC 低通滤波器 */
#define P2CA0          0x04 /* 外部引脚信号连接到比较器 A 的 CA0 */
#define P2CA1          0x08 /* 外部引脚信号连接到比较器 A 的 CA1 */
#define CACTL24        0x10
#define CACTL25        0x20
#define CACTL26        0x40
#define CACTL27        0x80

#define CAPD0          0x01 /* Comp. A Disable Input Buffer of Port Register .0 */
*/

```

```

#define CAPD1          0x02   /* Comp. A Disable Input Buffer of Port Register .1
*/
#define CAPD2          0x04   /* Comp. A Disable Input Buffer of Port Register .2
*/
#define CAPD3          0x08   /* Comp. A Disable Input Buffer of Port Register .3
*/
#define CAPD4          0x10   /* Comp. A Disable Input Buffer of Port Register .4
*/
#define CAPD5          0x20   /* Comp. A Disable Input Buffer of Port Register .5
*/
#define CAPD6          0x40   /* Comp. A Disable Input Buffer of Port Register .6
*/
#define CAPD7          0x80   /* Comp. A Disable Input Buffer of Port Register .7
*/

/*****
* ADC12   A/D 采样寄存器定义
*****/
/*ADC12 转换控制类寄存器*/
#define ADC12CTL0_    0x0;' /* ADC12 Control 0 */
sfrw   ADC12CTL0    = ADC12CTL0_;
#define ADC12CTL1_    0x01A2 /* ADC12 Control 1 */
sfrw   ADC12CTL1    = ADC12CTL1_;

/*ADC12 中断控制类寄存器*/
#define ADC12IFG_    0x01A4 /* ADC12 Interrupt Flag */
sfrw   ADC12IFG    = ADC12IFG_;
#define ADC12IE_    0x01A6 /* ADC12 Interrupt Enable */
sfrw   ADC12IE    = ADC12IE_;
#define ADC12IV_    0x01A8 /* ADC12 Interrupt Vector Word */
sfrw   ADC12IV    = ADC12IV_;

/*ADC12 存储器类寄存器*/
#define ADC12MEM_    0x0140 /* ADC12 Conversion Memory */
#ifndef __IAR_SYSTEMS_ICC
#define ADC12MEM    ADC12MEM_ /* ADC12 Conversion Memory (for
assembler) */
#else
#define ADC12MEM    ((int*) ADC12MEM_) /* ADC12 Conversion Memory (for C)
*/
#endif
#define ADC12MEM0_    ADC12MEM_ /* ADC12 Conversion Memory 0 */
sfrw   ADC12MEM0    = ADC12MEM0_;
#define ADC12MEM1_    0x0142 /* ADC12 Conversion Memory 1 */
sfrw   ADC12MEM1    = ADC12MEM1_;
#define ADC12MEM2_    0x0144 /* ADC12 Conversion Memory 2 */
sfrw   ADC12MEM2    = ADC12MEM2_;
#define ADC12MEM3_    0x0146 /* ADC12 Conversion Memory 3 */
sfrw   ADC12MEM3    = ADC12MEM3_;
#define ADC12MEM4_    0x0148 /* ADC12 Conversion Memory 4 */
sfrw   ADC12MEM4    = ADC12MEM4_;
#define ADC12MEM5_    0x014A /* ADC12 Conversion Memory 5 */
sfrw   ADC12MEM5    = ADC12MEM5_;
#define ADC12MEM6_    0x014C /* ADC12 Conversion Memory 6 */
sfrw   ADC12MEM6    = ADC12MEM6_;
#define ADC12MEM7_    0x014E /* ADC12 Conversion Memory 7 */
sfrw   ADC12MEM7    = ADC12MEM7_;
#define ADC12MEM8_    0x0150 /* ADC12 Conversion Memory 8 */
sfrw   ADC12MEM8    = ADC12MEM8_;
#define ADC12MEM9_    0x0152 /* ADC12 Conversion Memory 9 */
sfrw   ADC12MEM9    = ADC12MEM9_;
#define ADC12MEM10_    0x0154 /* ADC12 Conversion Memory 10 */
sfrw   ADC12MEM10    = ADC12MEM10_;

```

```

#define ADC12MEM11_      0x0156 /* ADC12 Conversion Memory 11 */
sfrw  ADC12MEM11      = ADC12MEM11_;
#define ADC12MEM12_      0x0158 /* ADC12 Conversion Memory 12 */
sfrw  ADC12MEM12      = ADC12MEM12_;
#define ADC12MEM13_      0x015A /* ADC12 Conversion Memory 13 */
sfrw  ADC12MEM13      = ADC12MEM13_;
#define ADC12MEM14_      0x015C /* ADC12 Conversion Memory 14 */
sfrw  ADC12MEM14      = ADC12MEM14_;
#define ADC12MEM15_      0x015E /* ADC12 Conversion Memory 15 */
sfrw  ADC12MEM15      = ADC12MEM15_;

/*ADC12 存贮控制类寄存器*/
#define ADC12MCTL_      0x0080 /* ADC12 Memory Control */
#ifndef __IAR_SYSTEMS_ICC
#define ADC12MCTL      ADC12MCTL_ /* ADC12 Memory Control (for assembler)
*/
#else
#define ADC12MCTL      ((char*) ADC12MCTL_) /* ADC12 Memory Control (for C)
*/
#endif
#define ADC12MCTL0_      ADC12MCTL_ /* ADC12 Memory Control 0 */
sfrb  ADC12MCTL0      = ADC12MCTL0_;
#define ADC12MCTL1_      0x0081 /* ADC12 Memory Control 1 */
sfrb  ADC12MCTL1      = ADC12MCTL1_;
#define ADC12MCTL2_      0x0082 /* ADC12 Memory Control 2 */
sfrb  ADC12MCTL2      = ADC12MCTL2_;
#define ADC12MCTL3_      0x0083 /* ADC12 Memory Control 3 */
sfrb  ADC12MCTL3      = ADC12MCTL3_;
#define ADC12MCTL4_      0x0084 /* ADC12 Memory Control 4 */
sfrb  ADC12MCTL4      = ADC12MCTL4_;
#define ADC12MCTL5_      0x0085 /* ADC12 Memory Control 5 */
sfrb  ADC12MCTL5      = ADC12MCTL5_;
#define ADC12MCTL6_      0x0086 /* ADC12 Memory Control 6 */
sfrb  ADC12MCTL6      = ADC12MCTL6_;
#define ADC12MCTL7_      0x0087 /* ADC12 Memory Control 7 */
sfrb  ADC12MCTL7      = ADC12MCTL7_;
#define ADC12MCTL8_      0x0088 /* ADC12 Memory Control 8 */
sfrb  ADC12MCTL8      = ADC12MCTL8_;
#define ADC12MCTL9_      0x0089 /* ADC12 Memory Control 9 */
sfrb  ADC12MCTL9      = ADC12MCTL9_;
#define ADC12MCTL10_     0x008A /* ADC12 Memory Control 10 */
sfrb  ADC12MCTL10     = ADC12MCTL10_;
#define ADC12MCTL11_     0x008B /* ADC12 Memory Control 11 */
sfrb  ADC12MCTL11     = ADC12MCTL11_;
#define ADC12MCTL12_     0x008C /* ADC12 Memory Control 12 */
sfrb  ADC12MCTL12     = ADC12MCTL12_;
#define ADC12MCTL13_     0x008D /* ADC12 Memory Control 13 */
sfrb  ADC12MCTL13     = ADC12MCTL13_;
#define ADC12MCTL14_     0x008E /* ADC12 Memory Control 14 */
sfrb  ADC12MCTL14     = ADC12MCTL14_;
#define ADC12MCTL15_     0x008F /* ADC12 Memory Control 15 */
sfrb  ADC12MCTL15     = ADC12MCTL15_;

/* ADC12CTL0 内 16 位控制寄存器位*/
#define ADC12SC          0x001 /*采样/转换控制位*/
#define ENC              0x002 /* 转换允许位*/
#define ADC12TOVIE      0x004 /*转换时间溢出中断允许位*/
#define ADC12OVIE       0x008 /*溢出中断允许位*/
#define ADC12ON          0x010 /*ADC12 内核控制位*/
#define REFON           0x020 /*参考电压控制位*/
#define REF2_5V         0x040 /*内部参考电压的电压值选择位 '0'为 1.5V;
'1'为 2.5V*/
#define MSH              0x080 /*多次采样/转换位*/

```

```

#define MSC 0x080 /*多次采样/转换位*/
/*SHT0 采样保持定时器 0 控制 ADC12 的结果存贮器 MEM0~MEM7 的采样周期*/
#define SHT0_0 0*0x100 /*采样周期=TADC12CLK*4 */
#define SHT0_1 1*0x100 /*采样周期=TADC12CLK*8 */
#define SHT0_2 2*0x100 /*采样周期=TADC12CLK*16 */
#define SHT0_3 3*0x100 /*采样周期=TADC12CLK*32 */
#define SHT0_4 4*0x100 /*采样周期=TADC12CLK*64 */
#define SHT0_5 5*0x100 /*采样周期=TADC12CLK*96 */
#define SHT0_6 6*0x100 /*采样周期=TADC12CLK*128 */
#define SHT0_7 7*0x100 /*采样周期=TADC12CLK*192 */
#define SHT0_8 8*0x100 /*采样周期=TADC12CLK*256 */
#define SHT0_9 9*0x100 /*采样周期=TADC12CLK*384 */
#define SHT0_10 10*0x100 /*采样周期=TADC12CLK*512 */
#define SHT0_11 11*0x100 /*采样周期=TADC12CLK*768 */
#define SHT0_12 12*0x100 /*采样周期=TADC12CLK*1024 */
#define SHT0_13 13*0x100 /*采样周期=TADC12CLK*1024 */
#define SHT0_14 14*0x100 /*采样周期=TADC12CLK*1024 */
#define SHT0_15 15*0x100 /*采样周期=TADC12CLK*1024 */
/*SHT1 采样保持定时器 1 控制 ADC12 的结果存贮器 MEM8~MEM15 的采样周期*/
#define SHT1_0 0*0x100 /*采样周期=TADC12CLK*4 */
#define SHT1_1 1*0x100 /*采样周期=TADC12CLK*8 */
#define SHT1_2 2*0x100 /*采样周期=TADC12CLK*16 */
#define SHT1_3 3*0x100 /*采样周期=TADC12CLK*32 */
#define SHT1_4 4*0x100 /*采样周期=TADC12CLK*64 */
#define SHT1_5 5*0x100 /*采样周期=TADC12CLK*96 */
#define SHT1_6 6*0x100 /*采样周期=TADC12CLK*128 */
#define SHT1_7 7*0x100 /*采样周期=TADC12CLK*192 */
#define SHT1_8 8*0x100 /*采样周期=TADC12CLK*256 */
#define SHT1_9 9*0x100 /*采样周期=TADC12CLK*384 */
#define SHT1_10 10*0x100 /*采样周期=TADC12CLK*512 */
#define SHT1_11 11*0x100 /*采样周期=TADC12CLK*768 */
#define SHT1_12 12*0x100 /*采样周期=TADC12CLK*1024 */
#define SHT1_13 13*0x100 /*采样周期=TADC12CLK*1024 */
#define SHT1_14 14*0x100 /*采样周期=TADC12CLK*1024 */
#define SHT1_15 15*0x100 /*采样周期=TADC12CLK*1024 */

/* ADC12CTL1 内 16 位控制寄存器位*/
#define ADC12BUSY 0x0001 /*ADC12 忙标志位*/
#define CONSEQ_0 0*2 /*单通道单次转换*/
#define CONSEQ_1 1*2 /*序列通道单次转换*/
#define CONSEQ_2 2*2 /*单通道多次转换*/
#define CONSEQ_3 3*2 /*序列通道多次转换*/
#define ADC12SSEL_0 0*8 /*ADC12 内部时钟源*/
#define ADC12SSEL_1 1*8 /*ACLK*/
#define ADC12SSEL_2 2*8 /*MCLK*/
#define ADC12SSEL_3 3*8 /*SCLK*/
#define ADC12DIV_0 0*0x20 /*1 分频*/
#define ADC12DIV_1 1*0x20 /*2 分频*/
#define ADC12DIV_2 2*0x20 /*3 分频*/
#define ADC12DIV_3 3*0x20 /*4 分频*/
#define ADC12DIV_4 4*0x20 /*5 分频*/
#define ADC12DIV_5 5*0x20 /*6 分频*/
#define ADC12DIV_6 6*0x20 /*7 分频*/
#define ADC12DIV_7 7*0x20 /*8 分频*/
#define ISSH 0x0100 /*采样输入信号反向与否控制位*/
#define SHP 0x0200 /*采样信号(SAMPCON)选择控制位*/
#define SHS_0 0*0x400 /*采样信号输入源选择控制位 ADC12SC*/
#define SHS_1 1*0x400 /*采样信号输入源选择控制位
TIMER_A.OUT1*/
#define SHS_2 2*0x400 /*采样信号输入源选择控制位
TIMER_B.OUT0*/
#define SHS_3 3*0x400 /*采样信号输入源选择控制位

```



```

TIMER_B.OUT1*/
/*转换存储器地址定义位*/
#define CSTARTADD_0      0*0x1000    /*选择 MEM0 首地址*/
#define CSTARTADD_1      1*0x1000    /*选择 MEM1 首地址*/
#define CSTARTADD_2      2*0x1000    /*选择 MEM2 首地址*/
#define CSTARTADD_3      3*0x1000    /*选择 MEM3 首地址*/
#define CSTARTADD_4      4*0x1000    /*选择 MEM4 首地址*/
#define CSTARTADD_5      5*0x1000    /*选择 MEM5 首地址*/
#define CSTARTADD_6      6*0x1000    /*选择 MEM6 首地址*/
#define CSTARTADD_7      7*0x1000    /*选择 MEM7 首地址*/
#define CSTARTADD_8      8*0x1000    /*选择 MEM8 首地址*/
#define CSTARTADD_9      9*0x1000    /*选择 MEM9 首地址*/
#define CSTARTADD_10     10*0x1000   /*选择 MEM10 首地址*/
#define CSTARTADD_11     11*0x1000   /*选择 MEM11 首地址*/
#define CSTARTADD_12     12*0x1000   /*选择 MEM12 首地址*/
#define CSTARTADD_13     13*0x1000   /*选择 MEM13 首地址*/
#define CSTARTADD_14     14*0x1000   /*选择 MEM14 首地址*/
#define CSTARTADD_15     15*0x1000   /*选择 MEM15 首地址*/

/* ADC12MCTLx */
#define INCH_0            0            /*选择模拟量通道 0 A0 */
#define INCH_1            1            /*选择模拟量通道 0 A1*/
#define INCH_2            2            /*选择模拟量通道 0 A2*/
#define INCH_3            3            /*选择模拟量通道 0 A3*/
#define INCH_4            4            /*选择模拟量通道 0 A4*/
#define INCH_5            5            /*选择模拟量通道 0 A5*/
#define INCH_6            6            /*选择模拟量通道 0 A6*/
#define INCH_7            7            /*选择模拟量通道 0 A7*/
#define INCH_8            8            /*VEREF+*/
#define INCH_9            9            /*VEREF-*/
#define INCH_10           10           /*片内温度传感器的输出*/
#define INCH_11           11           /*(AVCC-AVSS)/2*/
#define INCH_12           12           /*(AVCC-AVSS)/2*/
#define INCH_13           13           /*(AVCC-AVSS)/2*/
#define INCH_14           14           /*(AVCC-AVSS)/2*/
#define INCH_15           15           /*(AVCC-AVSS)/2*/
/*参考电压源选择位*/
#define SREF_0            0*0x10      /*VR+ = AVCC; VR- = AVSS*/
#define SREF_1            1*0x10      /*VR+ = VREF+; VR- = AVSS*/
#define SREF_2            2*0x10      /*VR+ = VREF+; VR- = AVSS*/
#define SREF_3            3*0x10      /*VR+ = VREF+; VR- = AVSS*/
#define SREF_4            4*0x10      /*VR+ = AVCC; VR- = VREF-*/
#define SREF_5            5*0x10      /*VR+ = VREF+; VR- = VREF-*/
#define SREF_6            6*0x10      /*VR+ = VREF+; VR- = VREF-*/
#define SREF_7            7*0x10      /*VR+ = VREF+; VR- = VREF-*/
#define EOS                0x80       /*序列结束选择位*/

/*****
* Interrupt Vectors (offset from 0xFFE0) 16 个中断矢量定义
*****/

//#define BASICTIMER_VECTOR 0 * 2 /* 0xFFE0 Basic Timer MSP430F149 没有*/
#define PORT2_VECTOR      1 * 2 /* 0xFFE2 Port 2 */
#define UART1TX_VECTOR    2 * 2 /* 0xFFE4 UART 1 Transmit */
#define UART1RX_VECTOR    3 * 2 /* 0xFFE6 UART 1 Receive */
#define PORT1_VECTOR      4 * 2 /* 0xFFE8 Port 1 */
#define TIMERA1_VECTOR    5 * 2 /* 0xFFEA Timer A CC1-2, TA */
#define TIMERA0_VECTOR    6 * 2 /* 0xFFEC Timer A CC0 */
#define ADC_VECTOR        7 * 2 /* 0xFFEE ADC */
#define UART0TX_VECTOR    8 * 2 /* 0xFFF0 UART 0 Transmit *//2010-08-18
#define UART0RX_VECTOR    9 * 2 /* 0xFFF2 UART 0 Receive */
#define WDT_VECTOR        10 * 2 /* 0xFFF4 Watchdog Timer */
#define COMPARATORA_VECTOR 11 * 2 /* 0xFFF6 Comparator A */

```

```
#define TIMERB1_VECTOR    12 * 2 /* 0xFFF8 Timer B 1-6 */
#define TIMERB0_VECTOR    13 * 2 /* 0xFFFA Timer B 0 */
#define NMI_VECTOR        14 * 2 /* 0xFFFC Non-maskable */
#define RESET_VECTOR      15 * 2 /* 0xFFFE Reset [Highest Priority] */

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

#endif /* #ifndef __msp430x44x */
```

第九单片机论坛  
www.9mcu.com