```
/* This file supports MSP430FR2433 devices. */
/* Version: 1.211 */
/* Default linker script, for normal executables */

OUTPUT_ARCH(msp430)
ENTRY(_start)

MEMORY {
  SFR             : ORIGIN = 0x0000, LENGTH = 0x0010 /* END=0x0010, size 16 */
  BSL             : ORIGIN = 0x1000, LENGTH = 0x0800
  RAM             : ORIGIN = 0x2000, LENGTH = 0x1000 /* END=0x2FFF, size 4096 */
  INFOMEM         : ORIGIN = 0x1800, LENGTH = 0x0200 /* END=0x19FF, size 512 as 1 512-byte segments
*/
  INFOA           : ORIGIN = 0x1800, LENGTH = 0x0200 /* END=0x19FF, size 512 */
  FRAM (rx)       : ORIGIN = 0xC400, LENGTH = 0x3B80 /* END=0xFF7F, size 15232 */
  JTAGSIGNATURE   : ORIGIN = 0xFF80, LENGTH = 0x0004
  BSLSIGNATURE    : ORIGIN = 0xFF84, LENGTH = 0x0004
  VECT1           : ORIGIN = 0xFF88, LENGTH = 0x0002
  VECT2           : ORIGIN = 0xFF8A, LENGTH = 0x0002
  VECT3           : ORIGIN = 0xFF8C, LENGTH = 0x0002
  VECT4           : ORIGIN = 0xFF8E, LENGTH = 0x0002
  VECT5           : ORIGIN = 0xFF90, LENGTH = 0x0002
  VECT6           : ORIGIN = 0xFF92, LENGTH = 0x0002
  VECT7           : ORIGIN = 0xFF94, LENGTH = 0x0002
  VECT8           : ORIGIN = 0xFF96, LENGTH = 0x0002
  VECT9           : ORIGIN = 0xFF98, LENGTH = 0x0002
  VECT10          : ORIGIN = 0xFF9A, LENGTH = 0x0002
  VECT11          : ORIGIN = 0xFF9C, LENGTH = 0x0002
  VECT12          : ORIGIN = 0xFF9E, LENGTH = 0x0002
  VECT13          : ORIGIN = 0xFFA0, LENGTH = 0x0002
  VECT14          : ORIGIN = 0xFFA2, LENGTH = 0x0002
  VECT15          : ORIGIN = 0xFFA4, LENGTH = 0x0002
  VECT16          : ORIGIN = 0xFFA6, LENGTH = 0x0002
  VECT17          : ORIGIN = 0xFFA8, LENGTH = 0x0002
  VECT18          : ORIGIN = 0xFFAA, LENGTH = 0x0002
  VECT19          : ORIGIN = 0xFFAC, LENGTH = 0x0002
  VECT20          : ORIGIN = 0xFFAE, LENGTH = 0x0002
  VECT21          : ORIGIN = 0xFFB0, LENGTH = 0x0002
  VECT22          : ORIGIN = 0xFFB2, LENGTH = 0x0002
  VECT23          : ORIGIN = 0xFFB4, LENGTH = 0x0002
  VECT24          : ORIGIN = 0xFFB6, LENGTH = 0x0002
  VECT25          : ORIGIN = 0xFFB8, LENGTH = 0x0002
  VECT26          : ORIGIN = 0xFFBA, LENGTH = 0x0002
  VECT27          : ORIGIN = 0xFFBC, LENGTH = 0x0002
  VECT28          : ORIGIN = 0xFFBE, LENGTH = 0x0002
  VECT29          : ORIGIN = 0xFFC0, LENGTH = 0x0002
  VECT30          : ORIGIN = 0xFFC2, LENGTH = 0x0002
  VECT31          : ORIGIN = 0xFFC4, LENGTH = 0x0002
  VECT32          : ORIGIN = 0xFFC6, LENGTH = 0x0002
  VECT33          : ORIGIN = 0xFFC8, LENGTH = 0x0002
  VECT34          : ORIGIN = 0xFFCA, LENGTH = 0x0002
  VECT35          : ORIGIN = 0xFFCC, LENGTH = 0x0002
  VECT36          : ORIGIN = 0xFFCE, LENGTH = 0x0002
  VECT37          : ORIGIN = 0xFFD0, LENGTH = 0x0002
  VECT38          : ORIGIN = 0xFFD2, LENGTH = 0x0002
  VECT39          : ORIGIN = 0xFFD4, LENGTH = 0x0002
  VECT40          : ORIGIN = 0xFFD6, LENGTH = 0x0002
  VECT41          : ORIGIN = 0xFFD8, LENGTH = 0x0002
  VECT42          : ORIGIN = 0xFFDA, LENGTH = 0x0002
  VECT43          : ORIGIN = 0xFFDC, LENGTH = 0x0002
  VECT44          : ORIGIN = 0xFFDE, LENGTH = 0x0002
  VECT45          : ORIGIN = 0xFFE0, LENGTH = 0x0002
  VECT46          : ORIGIN = 0xFFE2, LENGTH = 0x0002
  VECT47          : ORIGIN = 0xFFE4, LENGTH = 0x0002
  VECT48          : ORIGIN = 0xFFE6, LENGTH = 0x0002
  VECT49          : ORIGIN = 0xFFE8, LENGTH = 0x0002
  VECT50          : ORIGIN = 0xFFEA, LENGTH = 0x0002
  VECT51          : ORIGIN = 0xFFEC, LENGTH = 0x0002
  VECT52          : ORIGIN = 0xFFEE, LENGTH = 0x0002
  VECT53          : ORIGIN = 0xFFF0, LENGTH = 0x0002
  VECT54          : ORIGIN = 0xFFF2, LENGTH = 0x0002
  VECT55          : ORIGIN = 0xFFF4, LENGTH = 0x0002
```

```
  VECT56           : ORIGIN = 0xFFF6, LENGTH = 0x0002
  VECT57           : ORIGIN = 0xFFF8, LENGTH = 0x0002
  VECT58           : ORIGIN = 0xFFFA, LENGTH = 0x0002
  VECT59           : ORIGIN = 0xFFFC, LENGTH = 0x0002
  RESETVEC         : ORIGIN = 0xFFFE, LENGTH = 0x0002
}

SECTIONS
{
  .jtagsignature      : {} > JTAGSIGNATURE
  .bslsignature       : {} > BSLSIGNATURE
  __interrupt_vector_1   : { KEEP (*(__interrupt_vector_1 )) } > VECT1
  __interrupt_vector_2   : { KEEP (*(__interrupt_vector_2 )) } > VECT2
  __interrupt_vector_3   : { KEEP (*(__interrupt_vector_3 )) } > VECT3
  __interrupt_vector_4   : { KEEP (*(__interrupt_vector_4 )) } > VECT4
  __interrupt_vector_5   : { KEEP (*(__interrupt_vector_5 )) } > VECT5
  __interrupt_vector_6   : { KEEP (*(__interrupt_vector_6 )) } > VECT6
  __interrupt_vector_7   : { KEEP (*(__interrupt_vector_7 )) } > VECT7
  __interrupt_vector_8   : { KEEP (*(__interrupt_vector_8 )) } > VECT8
  __interrupt_vector_9   : { KEEP (*(__interrupt_vector_9 )) } > VECT9
  __interrupt_vector_10  : { KEEP (*(__interrupt_vector_10)) } > VECT10
  __interrupt_vector_11  : { KEEP (*(__interrupt_vector_11)) } > VECT11
  __interrupt_vector_12  : { KEEP (*(__interrupt_vector_12)) } > VECT12
  __interrupt_vector_13  : { KEEP (*(__interrupt_vector_13)) } > VECT13
  __interrupt_vector_14  : { KEEP (*(__interrupt_vector_14)) } > VECT14
  __interrupt_vector_15  : { KEEP (*(__interrupt_vector_15)) } > VECT15
  __interrupt_vector_16  : { KEEP (*(__interrupt_vector_16)) } > VECT16
  __interrupt_vector_17  : { KEEP (*(__interrupt_vector_17)) } > VECT17
  __interrupt_vector_18  : { KEEP (*(__interrupt_vector_18)) } > VECT18
  __interrupt_vector_19  : { KEEP (*(__interrupt_vector_19)) } > VECT19
  __interrupt_vector_20  : { KEEP (*(__interrupt_vector_20)) } > VECT20
  __interrupt_vector_21  : { KEEP (*(__interrupt_vector_21)) } > VECT21
  __interrupt_vector_22  : { KEEP (*(__interrupt_vector_22)) } > VECT22
  __interrupt_vector_23  : { KEEP (*(__interrupt_vector_23)) } > VECT23
  __interrupt_vector_24  : { KEEP (*(__interrupt_vector_24)) } > VECT24
  __interrupt_vector_25  : { KEEP (*(__interrupt_vector_25)) } > VECT25
  __interrupt_vector_26  : { KEEP (*(__interrupt_vector_26)) } > VECT26
  __interrupt_vector_27  : { KEEP (*(__interrupt_vector_27)) } > VECT27
  __interrupt_vector_28  : { KEEP (*(__interrupt_vector_28)) } > VECT28
  __interrupt_vector_29  : { KEEP (*(__interrupt_vector_29)) } > VECT29
  __interrupt_vector_30  : { KEEP (*(__interrupt_vector_30)) } > VECT30
  __interrupt_vector_31  : { KEEP (*(__interrupt_vector_31)) } > VECT31
  __interrupt_vector_32  : { KEEP (*(__interrupt_vector_32)) } > VECT32
  __interrupt_vector_33  : { KEEP (*(__interrupt_vector_33)) } > VECT33
  __interrupt_vector_34  : { KEEP (*(__interrupt_vector_34)) } > VECT34
  __interrupt_vector_35  : { KEEP (*(__interrupt_vector_35)) } > VECT35
  __interrupt_vector_36  : { KEEP (*(__interrupt_vector_36)) } > VECT36
  __interrupt_vector_37  : { KEEP (*(__interrupt_vector_37)) } > VECT37
  __interrupt_vector_38  : { KEEP (*(__interrupt_vector_38)) } > VECT38
  __interrupt_vector_39  : { KEEP (*(__interrupt_vector_39)) } > VECT39
  __interrupt_vector_40  : { KEEP (*(__interrupt_vector_40)) } > VECT40
  __interrupt_vector_41  : { KEEP (*(__interrupt_vector_41)) } > VECT41
  __interrupt_vector_42  : { KEEP (*(__interrupt_vector_42)) KEEP (*(__interrupt_vector_port2)) } >
VECT42
  __interrupt_vector_43  : { KEEP (*(__interrupt_vector_43)) KEEP (*(__interrupt_vector_port1)) } >
VECT43
  __interrupt_vector_44  : { KEEP (*(__interrupt_vector_44)) KEEP (*(__interrupt_vector_adc)) } >
VECT44
  __interrupt_vector_45  : { KEEP (*(__interrupt_vector_45)) KEEP (*(__interrupt_vector_usci_b0)) } >
VECT45
  __interrupt_vector_46  : { KEEP (*(__interrupt_vector_46)) KEEP (*(__interrupt_vector_usci_a1)) } >
VECT46
  __interrupt_vector_47  : { KEEP (*(__interrupt_vector_47)) KEEP (*(__interrupt_vector_usci_a0)) } >
VECT47
  __interrupt_vector_48  : { KEEP (*(__interrupt_vector_48)) KEEP (*(__interrupt_vector_wdt)) } >
VECT48
  __interrupt_vector_49  : { KEEP (*(__interrupt_vector_49)) KEEP (*(__interrupt_vector_rtc)) } >
VECT49
  __interrupt_vector_50  : { KEEP (*(__interrupt_vector_50)) KEEP (*(__interrupt_vector_timer3_a1)) }
> VECT50
  __interrupt_vector_51  : { KEEP (*(__interrupt_vector_51)) KEEP (*(__interrupt_vector_timer3_a0)) }
> VECT51
```

```
  __interrupt_vector_52  : { KEEP (*(__interrupt_vector_52)) KEEP (*(__interrupt_vector_timer2_a1)) }
> VECT52
  __interrupt_vector_53  : { KEEP (*(__interrupt_vector_53)) KEEP (*(__interrupt_vector_timer2_a0)) }
> VECT53
  __interrupt_vector_54  : { KEEP (*(__interrupt_vector_54)) KEEP (*(__interrupt_vector_timer1_a1)) }
> VECT54
  __interrupt_vector_55  : { KEEP (*(__interrupt_vector_55)) KEEP (*(__interrupt_vector_timer1_a0)) }
> VECT55
  __interrupt_vector_56  : { KEEP (*(__interrupt_vector_56)) KEEP (*(__interrupt_vector_timer0_a1)) }
> VECT56
  __interrupt_vector_57  : { KEEP (*(__interrupt_vector_57)) KEEP (*(__interrupt_vector_timer0_a0)) }
> VECT57
  __interrupt_vector_58  : { KEEP (*(__interrupt_vector_58)) KEEP (*(__interrupt_vector_unmi)) } >
VECT58
  __interrupt_vector_59  : { KEEP (*(__interrupt_vector_59)) KEEP (*(__interrupt_vector_sysnmi)) } >
VECT59
  __reset_vector :
  {
    KEEP (*(__interrupt_vector_60))
    KEEP (*(__interrupt_vector_reset))
    KEEP (*(.resetvec))
  } > RESETVEC

  .rodata :
  {
    . = ALIGN(2);
    *(.plt)
    *(.rodata .rodata.* .gnu.linkonce.r.* .const .const:*)
    *(.rodata1)
    KEEP (*(.gcc_except_table)) *(.gcc_except_table.*)
  } > FRAM

  /* Note: This is a separate .rodata section for sections which are
     read only but which older linkers treat as read-write.
     This prevents older linkers from marking the entire .rodata
     section as read-write.  */
  .rodata2 :
  {
    . = ALIGN(2);
    PROVIDE (__preinit_array_start = .);
    KEEP (*(.preinit_array))
    PROVIDE (__preinit_array_end = .);
    . = ALIGN(2);
    PROVIDE (__init_array_start = .);
    KEEP (*(SORT(.init_array.*)))
    KEEP (*(.init_array))
    PROVIDE (__init_array_end = .);
    . = ALIGN(2);
    PROVIDE (__fini_array_start = .);
    KEEP (*(.fini_array))
    KEEP (*(SORT(.fini_array.*)))
    PROVIDE (__fini_array_end = .);
    . = ALIGN(2);
    *(.eh_frame_hdr)
    KEEP (*(.eh_frame))

    /* gcc uses crtbegin.o to find the start of the constructors, so
       we make sure it is first.  Because this is a wildcard, it
       doesn't matter if the user does not actually link against
       crtbegin.o; the linker won't look for a file to match a
       wildcard.  The wildcard also means that it doesn't matter which
       directory crtbegin.o is in.  */
    KEEP (*crtbegin*.o(.ctors))

    /* We don't want to include the .ctor section from the crtend.o
       file until after the sorted ctors.  The .ctor section from
       the crtend file contains the end of ctors marker and it must
       be last */
    KEEP (*(EXCLUDE_FILE (*crtend*.o ) .ctors))
    KEEP (*(SORT(.ctors.*)))
    KEEP (*(.ctors))
```

```
  KEEP (*crtbegin*.o(.dtors))
  KEEP (*(EXCLUDE_FILE (*crtend*.o ) .dtors))
  KEEP (*(SORT(.dtors.*)))
  KEEP (*(.dtors))
} > FRAM


/* This section contains data that is initialised during load
   but not on application reset.  */
.persistent :
{
  . = ALIGN(2);
  PROVIDE (__persistent_start = .);
  *(.persistent)
  . = ALIGN(2);
  PROVIDE (__persistent_end = .);
} > FRAM

.text :
{
  . = ALIGN(2);
  PROVIDE (_start = .);
  KEEP (*(SORT(.crt_*)))
  *(.lowtext .text .stub .text.* .gnu.linkonce.t.* .text:*)
  KEEP (*(.text.*personality*))
  /* .gnu.warning sections are handled specially by elf32.em.  */
  *(.gnu.warning)
  *(.interp .hash .dynsym .dynstr .gnu.version*)
  PROVIDE (__etext = .);
  PROVIDE (_etext = .);
  PROVIDE (etext = .);
  . = ALIGN(2);
  KEEP (*(.init))
  KEEP (*(.fini))
  KEEP (*(.tm_clone_table))
} > FRAM

.data :
{
  . = ALIGN(2);
  PROVIDE (__datastart = .);

  KEEP (*(.jcr))
  *(.data.rel.ro.local) *(.data.rel.ro*)
  *(.dynamic)

  *(.data .data.* .gnu.linkonce.d.*)
  KEEP (*(.gnu.linkonce.d.*personality*))
  SORT(CONSTRUCTORS)
  *(.data1)
  *(.got.plt) *(.got)

  /* We want the small data sections together, so single-instruction offsets
     can access them all, and initialized data all before uninitialized, so
     we can shorten the on-disk segment size.  */
  . = ALIGN(2);
  *(.sdata .sdata.* .gnu.linkonce.s.* D_2 D_1)

  . = ALIGN(2);

  _edata = .;
  PROVIDE (edata = .);
  PROVIDE (__dataend = .);
} > RAM AT> FRAM

/* Note that crt0 assumes this is a multiple of two; all the
   start/stop symbols are also assumed word-aligned.  */
PROVIDE(__romdatastart = LOADADDR(.data));
PROVIDE (__romdatacopysize = SIZEOF(.data));


.bss :
```

```
{
  . = ALIGN(2);
  PROVIDE (__bssstart = .);
  *(.dynbss)
  *(.sbss .sbss.*)
  *(.bss .bss.* .gnu.linkonce.b.*)
  . = ALIGN(2);
  *(COMMON)
  PROVIDE (__bssend = .);
} > RAM
PROVIDE (__bsssize = SIZEOF(.bss));


/* This section contains data that is not initialised during load
   or application reset.  */
.noinit (NOLOAD) :
{
  . = ALIGN(2);
  PROVIDE (__noinit_start = .);
  *(.noinit)
  . = ALIGN(2);
  PROVIDE (__noinit_end = .);
  end = .;
} > RAM

/* We create this section so that "end" will always be in the
   RAM region (matching .stack below), even if the .bss
   section is empty.  */
.heap (NOLOAD) :
{
  . = ALIGN(2);
  __heap_start__ = .;
  _end = __heap_start__;
  PROVIDE (end = .);
  KEEP (*(.heap))
  _end = .;
  PROVIDE (end = .);
  /* This word is here so that the section is not empty, and thus
     not discarded by the linker.  The actual value does not matter
     and is ignored.  */
  LONG(0);
  __heap_end__ = .;
  __HeapLimit = __heap_end__;
} > RAM
/* WARNING: Do not place anything in RAM here.
   The heap section must be the last section in RAM and the stack
   section must be placed at the very end of the RAM region.  */

.stack (ORIGIN (RAM) + LENGTH(RAM)) :
{
  PROVIDE (__stack = .);
  *(.stack)
}

.infoA (NOLOAD) : {} > INFOA              /* MSP430 INFO FLASH MEMORY SEGMENTS */

/* The rest are all not normally part of the runtime image.  */

.MSP430.attributes 0 :
{
  KEEP (*(.MSP430.attributes))
  KEEP (*(.gnu.attributes))
  KEEP (*(__TI_build_attributes))
}

/* Stabs debugging sections.  */
.stab          0 : { *(.stab) }
.stabstr       0 : { *(.stabstr) }
.stab.excl     0 : { *(.stab.excl) }
.stab.exclstr  0 : { *(.stab.exclstr) }
.stab.index    0 : { *(.stab.index) }
.stab.indexstr 0 : { *(.stab.indexstr) }
```

```
  .comment       0 : { *(.comment) }
  /* DWARF debug sections.
     Symbols in the DWARF debugging sections are relative to the beginning
     of the section so we begin them at 0.  */
  /* DWARF 1.  */
  .debug          0 : { *(.debug) }
  .line           0 : { *(.line) }
  /* GNU DWARF 1 extensions.  */
  .debug_srcinfo  0 : { *(.debug_srcinfo) }
  .debug_sfnames  0 : { *(.debug_sfnames) }
  /* DWARF 1.1 and DWARF 2.  */
  .debug_aranges  0 : { *(.debug_aranges) }
  .debug_pubnames 0 : { *(.debug_pubnames) }
  /* DWARF 2.  */
  .debug_info     0 : { *(.debug_info .gnu.linkonce.wi.*) }
  .debug_abbrev   0 : { *(.debug_abbrev) }
  .debug_line     0 : { *(.debug_line .debug_line.* .debug_line_end ) }
  .debug_frame    0 : { *(.debug_frame) }
  .debug_str      0 : { *(.debug_str) }
  .debug_loc      0 : { *(.debug_loc) }
  .debug_macinfo  0 : { *(.debug_macinfo) }
  /* SGI/MIPS DWARF 2 extensions.  */
  .debug_weaknames 0 : { *(.debug_weaknames) }
  .debug_funcnames 0 : { *(.debug_funcnames) }
  .debug_typenames 0 : { *(.debug_typenames) }
  .debug_varnames  0 : { *(.debug_varnames) }
  /* DWARF 3 */
  .debug_pubtypes 0 : { *(.debug_pubtypes) }
  .debug_ranges   0 : { *(.debug_ranges) }
  /* DWARF Extension.  */
  .debug_macro    0 : { *(.debug_macro) }

  /DISCARD/ : { *(.note.GNU-stack) }
}


/*****************************************************************************/
/* Include peripherals memory map                                  begin  */
/*****************************************************************************/

/* INCLUDE msp430fr2433_symbols.ld */
/* This file supports MSP430FR2433 devices. */
/* Version: 1.211 */

/************************************************************
* STANDARD BITS
************************************************************/
/************************************************************
* STATUS REGISTER BITS
************************************************************/
/************************************************************
* PERIPHERAL FILE MAP
************************************************************/
/************************************************************
* ADC
************************************************************/
PROVIDE(ADCCTL0           = 0x0700);
PROVIDE(ADCCTL0_L         = 0x0700);
PROVIDE(ADCCTL0_H         = 0x0701);
PROVIDE(ADCCTL1           = 0x0702);
PROVIDE(ADCCTL1_L         = 0x0702);
PROVIDE(ADCCTL1_H         = 0x0703);
PROVIDE(ADCCTL2           = 0x0704);
PROVIDE(ADCCTL2_L         = 0x0704);
PROVIDE(ADCCTL2_H         = 0x0705);
PROVIDE(ADCLO             = 0x0706);
PROVIDE(ADCLO_L           = 0x0706);
PROVIDE(ADCLO_H           = 0x0707);
PROVIDE(ADCHI             = 0x0708);
PROVIDE(ADCHI_L           = 0x0708);
PROVIDE(ADCHI_H           = 0x0709);
PROVIDE(ADCMCTL0          = 0x070A);
```

```
PROVIDE(ADCMCTL0_L         = 0x070A);
PROVIDE(ADCMCTL0_H         = 0x070B);
PROVIDE(ADCMEM0            = 0x0712);
PROVIDE(ADCMEM0_L          = 0x0712);
PROVIDE(ADCMEM0_H          = 0x0713);
PROVIDE(ADCIE              = 0x071A);
PROVIDE(ADCIE_L            = 0x071A);
PROVIDE(ADCIE_H            = 0x071B);
PROVIDE(ADCIFG             = 0x071C);
PROVIDE(ADCIFG_L           = 0x071C);
PROVIDE(ADCIFG_H           = 0x071D);
PROVIDE(ADCIV              = 0x071E);
PROVIDE(ADCIV_L            = 0x071E);
PROVIDE(ADCIV_H            = 0x071F);
/************************************************************
* Backup Memory Module
************************************************************/
PROVIDE(BAKMEM0            = 0x0660);
PROVIDE(BAKMEM0_L          = 0x0660);
PROVIDE(BAKMEM0_H          = 0x0661);
PROVIDE(BAKMEM1            = 0x0662);
PROVIDE(BAKMEM1_L          = 0x0662);
PROVIDE(BAKMEM1_H          = 0x0663);
PROVIDE(BAKMEM2            = 0x0664);
PROVIDE(BAKMEM2_L          = 0x0664);
PROVIDE(BAKMEM2_H          = 0x0665);
PROVIDE(BAKMEM3            = 0x0666);
PROVIDE(BAKMEM3_L          = 0x0666);
PROVIDE(BAKMEM3_H          = 0x0667);
PROVIDE(BAKMEM4            = 0x0668);
PROVIDE(BAKMEM4_L          = 0x0668);
PROVIDE(BAKMEM4_H          = 0x0669);
PROVIDE(BAKMEM5            = 0x066A);
PROVIDE(BAKMEM5_L          = 0x066A);
PROVIDE(BAKMEM5_H          = 0x066B);
PROVIDE(BAKMEM6            = 0x066C);
PROVIDE(BAKMEM6_L          = 0x066C);
PROVIDE(BAKMEM6_H          = 0x066D);
PROVIDE(BAKMEM7            = 0x066E);
PROVIDE(BAKMEM7_L          = 0x066E);
PROVIDE(BAKMEM7_H          = 0x066F);
PROVIDE(BAKMEM8            = 0x0670);
PROVIDE(BAKMEM8_L          = 0x0670);
PROVIDE(BAKMEM8_H          = 0x0671);
PROVIDE(BAKMEM9            = 0x0672);
PROVIDE(BAKMEM9_L          = 0x0672);
PROVIDE(BAKMEM9_H          = 0x0673);
PROVIDE(BAKMEM10           = 0x0674);
PROVIDE(BAKMEM10_L         = 0x0674);
PROVIDE(BAKMEM10_H         = 0x0675);
PROVIDE(BAKMEM11           = 0x0676);
PROVIDE(BAKMEM11_L         = 0x0676);
PROVIDE(BAKMEM11_H         = 0x0677);
PROVIDE(BAKMEM12           = 0x0678);
PROVIDE(BAKMEM12_L         = 0x0678);
PROVIDE(BAKMEM12_H         = 0x0679);
PROVIDE(BAKMEM13           = 0x067A);
PROVIDE(BAKMEM13_L         = 0x067A);
PROVIDE(BAKMEM13_H         = 0x067B);
PROVIDE(BAKMEM14           = 0x067C);
PROVIDE(BAKMEM14_L         = 0x067C);
PROVIDE(BAKMEM14_H         = 0x067D);
PROVIDE(BAKMEM15           = 0x067E);
PROVIDE(BAKMEM15_L         = 0x067E);
PROVIDE(BAKMEM15_H         = 0x067F);
/************************************************************
* CRC Module
************************************************************/
PROVIDE(CRCDI              = 0x01C0);
PROVIDE(CRCDI_L            = 0x01C0);
PROVIDE(CRCDI_H            = 0x01C1);
PROVIDE(CRCDIRB            = 0x01C2);
```

```
PROVIDE(CRCDIRB_L          = 0x01C2);
PROVIDE(CRCDIRB_H          = 0x01C3);
PROVIDE(CRCINIRES          = 0x01C4);
PROVIDE(CRCINIRES_L        = 0x01C4);
PROVIDE(CRCINIRES_H        = 0x01C5);
PROVIDE(CRCRESR            = 0x01C6);
PROVIDE(CRCRESR_L          = 0x01C6);
PROVIDE(CRCRESR_H          = 0x01C7);
/************************************************************
* CLOCK SYSTEM CONTROL
************************************************************/
PROVIDE(CSCTL0             = 0x0180);
PROVIDE(CSCTL0_L           = 0x0180);
PROVIDE(CSCTL0_H           = 0x0181);
PROVIDE(CSCTL1             = 0x0182);
PROVIDE(CSCTL1_L           = 0x0182);
PROVIDE(CSCTL1_H           = 0x0183);
PROVIDE(CSCTL2             = 0x0184);
PROVIDE(CSCTL2_L           = 0x0184);
PROVIDE(CSCTL2_H           = 0x0185);
PROVIDE(CSCTL3             = 0x0186);
PROVIDE(CSCTL3_L           = 0x0186);
PROVIDE(CSCTL3_H           = 0x0187);
PROVIDE(CSCTL4             = 0x0188);
PROVIDE(CSCTL4_L           = 0x0188);
PROVIDE(CSCTL4_H           = 0x0189);
PROVIDE(CSCTL5             = 0x018A);
PROVIDE(CSCTL5_L           = 0x018A);
PROVIDE(CSCTL5_H           = 0x018B);
PROVIDE(CSCTL6             = 0x018C);
PROVIDE(CSCTL6_L           = 0x018C);
PROVIDE(CSCTL6_H           = 0x018D);
PROVIDE(CSCTL7             = 0x018E);
PROVIDE(CSCTL7_L           = 0x018E);
PROVIDE(CSCTL7_H           = 0x018F);
PROVIDE(CSCTL8             = 0x0190);
PROVIDE(CSCTL8_L           = 0x0190);
PROVIDE(CSCTL8_H           = 0x0191);
/************************************************************
* FRAM Memory
************************************************************/
PROVIDE(FRCTL0             = 0x01A0);
PROVIDE(FRCTL0_L           = 0x01A0);
PROVIDE(FRCTL0_H           = 0x01A1);
PROVIDE(GCCTL0             = 0x01A4);
PROVIDE(GCCTL0_L           = 0x01A4);
PROVIDE(GCCTL0_H           = 0x01A5);
PROVIDE(GCCTL1             = 0x01A6);
PROVIDE(GCCTL1_L           = 0x01A6);
PROVIDE(GCCTL1_H           = 0x01A7);
/************************************************************
* HARDWARE MULTIPLIER 32Bit
************************************************************/
PROVIDE(MPY                = 0x04C0);
PROVIDE(MPY_L              = 0x04C0);
PROVIDE(MPY_H              = 0x04C1);
PROVIDE(MPYS               = 0x04C2);
PROVIDE(MPYS_L             = 0x04C2);
PROVIDE(MPYS_H             = 0x04C3);
PROVIDE(MAC                = 0x04C4);
PROVIDE(MAC_L              = 0x04C4);
PROVIDE(MAC_H              = 0x04C5);
PROVIDE(MACS               = 0x04C6);
PROVIDE(MACS_L             = 0x04C6);
PROVIDE(MACS_H             = 0x04C7);
PROVIDE(OP2                = 0x04C8);
PROVIDE(OP2_L              = 0x04C8);
PROVIDE(OP2_H              = 0x04C9);
PROVIDE(RESLO              = 0x04CA);
PROVIDE(RESLO_L            = 0x04CA);
PROVIDE(RESLO_H            = 0x04CB);
PROVIDE(RESHI              = 0x04CC);
```

```
PROVIDE(RESHI_L           = 0x04CC);
PROVIDE(RESHI_H           = 0x04CD);
PROVIDE(SUMEXT            = 0x04CE);
PROVIDE(SUMEXT_L          = 0x04CE);
PROVIDE(SUMEXT_H          = 0x04CF);
PROVIDE(MPY32L            = 0x04D0);
PROVIDE(MPY32L_L          = 0x04D0);
PROVIDE(MPY32L_H          = 0x04D1);
PROVIDE(MPY32H            = 0x04D2);
PROVIDE(MPY32H_L          = 0x04D2);
PROVIDE(MPY32H_H          = 0x04D3);
PROVIDE(MPYS32L           = 0x04D4);
PROVIDE(MPYS32L_L         = 0x04D4);
PROVIDE(MPYS32L_H         = 0x04D5);
PROVIDE(MPYS32H           = 0x04D6);
PROVIDE(MPYS32H_L         = 0x04D6);
PROVIDE(MPYS32H_H         = 0x04D7);
PROVIDE(MAC32L            = 0x04D8);
PROVIDE(MAC32L_L          = 0x04D8);
PROVIDE(MAC32L_H          = 0x04D9);
PROVIDE(MAC32H            = 0x04DA);
PROVIDE(MAC32H_L          = 0x04DA);
PROVIDE(MAC32H_H          = 0x04DB);
PROVIDE(MACS32L           = 0x04DC);
PROVIDE(MACS32L_L         = 0x04DC);
PROVIDE(MACS32L_H         = 0x04DD);
PROVIDE(MACS32H           = 0x04DE);
PROVIDE(MACS32H_L         = 0x04DE);
PROVIDE(MACS32H_H         = 0x04DF);
PROVIDE(OP2L              = 0x04E0);
PROVIDE(OP2L_L            = 0x04E0);
PROVIDE(OP2L_H            = 0x04E1);
PROVIDE(OP2H              = 0x04E2);
PROVIDE(OP2H_L            = 0x04E2);
PROVIDE(OP2H_H            = 0x04E3);
PROVIDE(RES0              = 0x04E4);
PROVIDE(RES0_L            = 0x04E4);
PROVIDE(RES0_H            = 0x04E5);
PROVIDE(RES1              = 0x04E6);
PROVIDE(RES1_L            = 0x04E6);
PROVIDE(RES1_H            = 0x04E7);
PROVIDE(RES2              = 0x04E8);
PROVIDE(RES2_L            = 0x04E8);
PROVIDE(RES2_H            = 0x04E9);
PROVIDE(RES3              = 0x04EA);
PROVIDE(RES3_L            = 0x04EA);
PROVIDE(RES3_H            = 0x04EB);
PROVIDE(MPY32CTL0         = 0x04EC);
PROVIDE(MPY32CTL0_L       = 0x04EC);
PROVIDE(MPY32CTL0_H       = 0x04ED);
/************************************************************
* PMM - Power Management System for FR2xx/FR4xx
************************************************************/
PROVIDE(PMMCTL0           = 0x0120);
PROVIDE(PMMCTL0_L         = 0x0120);
PROVIDE(PMMCTL0_H         = 0x0121);
PROVIDE(PMMCTL1           = 0x0122);
PROVIDE(PMMCTL1_L         = 0x0122);
PROVIDE(PMMCTL1_H         = 0x0123);
PROVIDE(PMMCTL2           = 0x0124);
PROVIDE(PMMCTL2_L         = 0x0124);
PROVIDE(PMMCTL2_H         = 0x0125);
PROVIDE(PMMIFG            = 0x012A);
PROVIDE(PMMIFG_L          = 0x012A);
PROVIDE(PMMIFG_H          = 0x012B);
PROVIDE(PMMIE             = 0x012E);
PROVIDE(PMMIE_L           = 0x012E);
PROVIDE(PMMIE_H           = 0x012F);
PROVIDE(PM5CTL0           = 0x0130);
PROVIDE(PM5CTL0_L         = 0x0130);
PROVIDE(PM5CTL0_H         = 0x0131);
/************************************************************
```

```
* DIGITAL I/O Port1/2 Pull up / Pull down Resistors
************************************************************/
PROVIDE(PAIN               = 0x0200);
PROVIDE(PAIN_L             = 0x0200);
PROVIDE(PAIN_H             = 0x0201);
PROVIDE(PAOUT              = 0x0202);
PROVIDE(PAOUT_L            = 0x0202);
PROVIDE(PAOUT_H            = 0x0203);
PROVIDE(PADIR              = 0x0204);
PROVIDE(PADIR_L            = 0x0204);
PROVIDE(PADIR_H            = 0x0205);
PROVIDE(PAREN              = 0x0206);
PROVIDE(PAREN_L            = 0x0206);
PROVIDE(PAREN_H            = 0x0207);
PROVIDE(PASEL0             = 0x020A);
PROVIDE(PASEL0_L           = 0x020A);
PROVIDE(PASEL0_H           = 0x020B);
PROVIDE(PASEL1             = 0x020C);
PROVIDE(PASEL1_L           = 0x020C);
PROVIDE(PASEL1_H           = 0x020D);
PROVIDE(PAIES              = 0x0218);
PROVIDE(PAIES_L            = 0x0218);
PROVIDE(PAIES_H            = 0x0219);
PROVIDE(PAIE               = 0x021A);
PROVIDE(PAIE_L             = 0x021A);
PROVIDE(PAIE_H             = 0x021B);
PROVIDE(PAIFG              = 0x021C);
PROVIDE(PAIFG_L            = 0x021C);
PROVIDE(PAIFG_H            = 0x021D);
PROVIDE(P1IV               = 0x020E);
PROVIDE(P2IV               = 0x021E);
/************************************************************
* DIGITAL I/O Port3 Pull up / Pull down Resistors
************************************************************/
PROVIDE(PBIN               = 0x0220);
PROVIDE(PBIN_L             = 0x0220);
PROVIDE(PBIN_H             = 0x0221);
PROVIDE(PBOUT              = 0x0222);
PROVIDE(PBOUT_L            = 0x0222);
PROVIDE(PBOUT_H            = 0x0223);
PROVIDE(PBDIR              = 0x0224);
PROVIDE(PBDIR_L            = 0x0224);
PROVIDE(PBDIR_H            = 0x0225);
PROVIDE(PBREN              = 0x0226);
PROVIDE(PBREN_L            = 0x0226);
PROVIDE(PBREN_H            = 0x0227);
PROVIDE(PBSEL0             = 0x022A);
PROVIDE(PBSEL0_L           = 0x022A);
PROVIDE(PBSEL0_H           = 0x022B);
PROVIDE(PBSEL1             = 0x022C);
PROVIDE(PBSEL1_L           = 0x022C);
PROVIDE(PBSEL1_H           = 0x022D);
/************************************************************
* Real-Time Clock (RTC) Counter
************************************************************/
PROVIDE(RTCCTL             = 0x0300);
PROVIDE(RTCCTL_L           = 0x0300);
PROVIDE(RTCCTL_H           = 0x0301);
PROVIDE(RTCIV              = 0x0304);
PROVIDE(RTCIV_L            = 0x0304);
PROVIDE(RTCIV_H            = 0x0305);
PROVIDE(RTCMOD             = 0x0308);
PROVIDE(RTCMOD_L           = 0x0308);
PROVIDE(RTCMOD_H           = 0x0309);
PROVIDE(RTCCNT             = 0x030C);
PROVIDE(RTCCNT_L           = 0x030C);
PROVIDE(RTCCNT_H           = 0x030D);
/************************************************************
* SFR - Special Function Register Module
************************************************************/
PROVIDE(SFRIE1             = 0x0100);
PROVIDE(SFRIE1_L           = 0x0100);
```

```
PROVIDE(SFRIE1_H           = 0x0101);
PROVIDE(SFRIFG1            = 0x0102);
PROVIDE(SFRIFG1_L          = 0x0102);
PROVIDE(SFRIFG1_H          = 0x0103);
PROVIDE(SFRRPCR            = 0x0104);
PROVIDE(SFRRPCR_L          = 0x0104);
PROVIDE(SFRRPCR_H          = 0x0105);
/************************************************************
* SYS - System Module
************************************************************/
PROVIDE(SYSCTL             = 0x0140);
PROVIDE(SYSCTL_L           = 0x0140);
PROVIDE(SYSCTL_H           = 0x0141);
PROVIDE(SYSBSLC            = 0x0142);
PROVIDE(SYSBSLC_L          = 0x0142);
PROVIDE(SYSBSLC_H          = 0x0143);
PROVIDE(SYSJMBC            = 0x0146);
PROVIDE(SYSJMBC_L          = 0x0146);
PROVIDE(SYSJMBC_H          = 0x0147);
PROVIDE(SYSJMBI0           = 0x0148);
PROVIDE(SYSJMBI0_L         = 0x0148);
PROVIDE(SYSJMBI0_H         = 0x0149);
PROVIDE(SYSJMBI1           = 0x014A);
PROVIDE(SYSJMBI1_L         = 0x014A);
PROVIDE(SYSJMBI1_H         = 0x014B);
PROVIDE(SYSJMBO0           = 0x014C);
PROVIDE(SYSJMBO0_L         = 0x014C);
PROVIDE(SYSJMBO0_H         = 0x014D);
PROVIDE(SYSJMBO1           = 0x014E);
PROVIDE(SYSJMBO1_L         = 0x014E);
PROVIDE(SYSJMBO1_H         = 0x014F);
PROVIDE(SYSBERRIV          = 0x0158);
PROVIDE(SYSBERRIV_L        = 0x0158);
PROVIDE(SYSBERRIV_H        = 0x0159);
PROVIDE(SYSUNIV            = 0x015A);
PROVIDE(SYSUNIV_L          = 0x015A);
PROVIDE(SYSUNIV_H          = 0x015B);
PROVIDE(SYSSNIV            = 0x015C);
PROVIDE(SYSSNIV_L          = 0x015C);
PROVIDE(SYSSNIV_H          = 0x015D);
PROVIDE(SYSRSTIV           = 0x015E);
PROVIDE(SYSRSTIV_L         = 0x015E);
PROVIDE(SYSRSTIV_H         = 0x015F);
PROVIDE(SYSCFG0            = 0x0160);
PROVIDE(SYSCFG0_L          = 0x0160);
PROVIDE(SYSCFG0_H          = 0x0161);
PROVIDE(SYSCFG1            = 0x0162);
PROVIDE(SYSCFG1_L          = 0x0162);
PROVIDE(SYSCFG1_H          = 0x0163);
PROVIDE(SYSCFG2            = 0x0164);
PROVIDE(SYSCFG2_L          = 0x0164);
PROVIDE(SYSCFG2_H          = 0x0165);
/************************************************************
* Timer0_A3
************************************************************/
PROVIDE(TA0CTL             = 0x0380);
PROVIDE(TA0CCTL0           = 0x0382);
PROVIDE(TA0CCTL1           = 0x0384);
PROVIDE(TA0CCTL2           = 0x0386);
PROVIDE(TA0R               = 0x0390);
PROVIDE(TA0CCR0            = 0x0392);
PROVIDE(TA0CCR1            = 0x0394);
PROVIDE(TA0CCR2            = 0x0396);
PROVIDE(TA0IV              = 0x03AE);
PROVIDE(TA0EX0             = 0x03A0);
/************************************************************
* Timer1_A3
************************************************************/
PROVIDE(TA1CTL             = 0x03C0);
PROVIDE(TA1CCTL0           = 0x03C2);
PROVIDE(TA1CCTL1           = 0x03C4);
PROVIDE(TA1CCTL2           = 0x03C6);
```

```
PROVIDE(TA1R              = 0x03D0);
PROVIDE(TA1CCR0           = 0x03D2);
PROVIDE(TA1CCR1           = 0x03D4);
PROVIDE(TA1CCR2           = 0x03D6);
PROVIDE(TA1IV             = 0x03EE);
PROVIDE(TA1EX0            = 0x03E0);
/************************************************************
* Timer2_A2
************************************************************/
PROVIDE(TA2CTL            = 0x0400);
PROVIDE(TA2CCTL0          = 0x0402);
PROVIDE(TA2CCTL1          = 0x0404);
PROVIDE(TA2R              = 0x0410);
PROVIDE(TA2CCR0           = 0x0412);
PROVIDE(TA2CCR1           = 0x0414);
PROVIDE(TA2IV             = 0x042E);
PROVIDE(TA2EX0            = 0x0420);
/************************************************************
* Timer3_A2
************************************************************/
PROVIDE(TA3CTL            = 0x0440);
PROVIDE(TA3CCTL0          = 0x0442);
PROVIDE(TA3CCTL1          = 0x0444);
PROVIDE(TA3R              = 0x0450);
PROVIDE(TA3CCR0           = 0x0452);
PROVIDE(TA3CCR1           = 0x0454);
PROVIDE(TA3IV             = 0x046E);
PROVIDE(TA3EX0            = 0x0460);
/************************************************************
* USCI A0
************************************************************/
PROVIDE(UCA0CTLW0         = 0x0500);
PROVIDE(UCA0CTLW0_L       = 0x0500);
PROVIDE(UCA0CTLW0_H       = 0x0501);
PROVIDE(UCA0CTLW1         = 0x0502);
PROVIDE(UCA0CTLW1_L       = 0x0502);
PROVIDE(UCA0CTLW1_H       = 0x0503);
PROVIDE(UCA0BRW           = 0x0506);
PROVIDE(UCA0BRW_L         = 0x0506);
PROVIDE(UCA0BRW_H         = 0x0507);
PROVIDE(UCA0MCTLW         = 0x0508);
PROVIDE(UCA0MCTLW_L       = 0x0508);
PROVIDE(UCA0MCTLW_H       = 0x0509);
PROVIDE(UCA0STATW         = 0x050A);
PROVIDE(UCA0RXBUF         = 0x050C);
PROVIDE(UCA0RXBUF_L       = 0x050C);
PROVIDE(UCA0RXBUF_H       = 0x050D);
PROVIDE(UCA0TXBUF         = 0x050E);
PROVIDE(UCA0TXBUF_L       = 0x050E);
PROVIDE(UCA0TXBUF_H       = 0x050F);
PROVIDE(UCA0ABCTL         = 0x0510);
PROVIDE(UCA0IRCTL         = 0x0512);
PROVIDE(UCA0IRCTL_L       = 0x0512);
PROVIDE(UCA0IRCTL_H       = 0x0513);
PROVIDE(UCA0IE            = 0x051A);
PROVIDE(UCA0IE_L          = 0x051A);
PROVIDE(UCA0IE_H          = 0x051B);
PROVIDE(UCA0IFG           = 0x051C);
PROVIDE(UCA0IFG_L         = 0x051C);
PROVIDE(UCA0IFG_H         = 0x051D);
PROVIDE(UCA0IV            = 0x051E);
/************************************************************
* USCI A1
************************************************************/
PROVIDE(UCA1CTLW0         = 0x0520);
PROVIDE(UCA1CTLW0_L       = 0x0520);
PROVIDE(UCA1CTLW0_H       = 0x0521);
PROVIDE(UCA1CTLW1         = 0x0522);
PROVIDE(UCA1CTLW1_L       = 0x0522);
PROVIDE(UCA1CTLW1_H       = 0x0523);
PROVIDE(UCA1BRW           = 0x0526);
PROVIDE(UCA1BRW_L         = 0x0526);
```

```
PROVIDE(UCA1BRW_H          = 0x0527);
PROVIDE(UCA1MCTLW          = 0x0528);
PROVIDE(UCA1MCTLW_L        = 0x0528);
PROVIDE(UCA1MCTLW_H        = 0x0529);
PROVIDE(UCA1STATW          = 0x052A);
PROVIDE(UCA1RXBUF          = 0x052C);
PROVIDE(UCA1RXBUF_L        = 0x052C);
PROVIDE(UCA1RXBUF_H        = 0x052D);
PROVIDE(UCA1TXBUF          = 0x052E);
PROVIDE(UCA1TXBUF_L        = 0x052E);
PROVIDE(UCA1TXBUF_H        = 0x052F);
PROVIDE(UCA1ABCTL          = 0x0530);
PROVIDE(UCA1IRCTL          = 0x0532);
PROVIDE(UCA1IRCTL_L        = 0x0532);
PROVIDE(UCA1IRCTL_H        = 0x0533);
PROVIDE(UCA1IE             = 0x053A);
PROVIDE(UCA1IE_L           = 0x053A);
PROVIDE(UCA1IE_H           = 0x053B);
PROVIDE(UCA1IFG            = 0x053C);
PROVIDE(UCA1IFG_L          = 0x053C);
PROVIDE(UCA1IFG_H          = 0x053D);
PROVIDE(UCA1IV             = 0x053E);
/************************************************************
* USCI B0
************************************************************/
PROVIDE(UCB0CTLW0          = 0x0540);
PROVIDE(UCB0CTLW0_L        = 0x0540);
PROVIDE(UCB0CTLW0_H        = 0x0541);
PROVIDE(UCB0CTLW1          = 0x0542);
PROVIDE(UCB0CTLW1_L        = 0x0542);
PROVIDE(UCB0CTLW1_H        = 0x0543);
PROVIDE(UCB0BRW            = 0x0546);
PROVIDE(UCB0BRW_L          = 0x0546);
PROVIDE(UCB0BRW_H          = 0x0547);
PROVIDE(UCB0STATW          = 0x0548);
PROVIDE(UCB0STATW_L        = 0x0548);
PROVIDE(UCB0STATW_H        = 0x0549);
PROVIDE(UCB0TBCNT          = 0x054A);
PROVIDE(UCB0TBCNT_L        = 0x054A);
PROVIDE(UCB0TBCNT_H        = 0x054B);
PROVIDE(UCB0RXBUF          = 0x054C);
PROVIDE(UCB0RXBUF_L        = 0x054C);
PROVIDE(UCB0RXBUF_H        = 0x054D);
PROVIDE(UCB0TXBUF          = 0x054E);
PROVIDE(UCB0TXBUF_L        = 0x054E);
PROVIDE(UCB0TXBUF_H        = 0x054F);
PROVIDE(UCB0I2COA0         = 0x0554);
PROVIDE(UCB0I2COA0_L       = 0x0554);
PROVIDE(UCB0I2COA0_H       = 0x0555);
PROVIDE(UCB0I2COA1         = 0x0556);
PROVIDE(UCB0I2COA1_L       = 0x0556);
PROVIDE(UCB0I2COA1_H       = 0x0557);
PROVIDE(UCB0I2COA2         = 0x0558);
PROVIDE(UCB0I2COA2_L       = 0x0558);
PROVIDE(UCB0I2COA2_H       = 0x0559);
PROVIDE(UCB0I2COA3         = 0x055A);
PROVIDE(UCB0I2COA3_L       = 0x055A);
PROVIDE(UCB0I2COA3_H       = 0x055B);
PROVIDE(UCB0ADDRX          = 0x055C);
PROVIDE(UCB0ADDRX_L        = 0x055C);
PROVIDE(UCB0ADDRX_H        = 0x055D);
PROVIDE(UCB0ADDMASK        = 0x055E);
PROVIDE(UCB0ADDMASK_L      = 0x055E);
PROVIDE(UCB0ADDMASK_H      = 0x055F);
PROVIDE(UCB0I2CSA          = 0x0560);
PROVIDE(UCB0I2CSA_L        = 0x0560);
PROVIDE(UCB0I2CSA_H        = 0x0561);
PROVIDE(UCB0IE             = 0x056A);
PROVIDE(UCB0IE_L           = 0x056A);
PROVIDE(UCB0IE_H           = 0x056B);
PROVIDE(UCB0IFG            = 0x056C);
PROVIDE(UCB0IFG_L          = 0x056C);
```

```
PROVIDE(UCB0IFG_H          = 0x056D);
PROVIDE(UCB0IV             = 0x056E);
/************************************************************
* WATCHDOG TIMER A
************************************************************/
PROVIDE(WDTCTL             = 0x01CC);
PROVIDE(WDTCTL_L           = 0x01CC);
PROVIDE(WDTCTL_H           = 0x01CD);
/************************************************************
* TLV Descriptors
************************************************************/
/************************************************************
* Interrupt Vectors (offset from 0xFF80 + 0x10 for Password)
************************************************************/
/************************************************************
* End of Modules
************************************************************/
/* Include peripherals memory map                                    end  */
```