```
/* Default linker script, for normal executables */
OUTPUT_FORMAT("elf32-msp430")
OUTPUT_ARCH("msp430")

/* INCLUDE memory.x begin */
MEMORY {
  sfr              : ORIGIN = 0x0000, LENGTH = 0x0010 /* END=0x0010, size 16 */
  peripheral_8bit  : ORIGIN = 0x0010, LENGTH = 0x00f0 /* END=0x0100, size 240 */
  peripheral_16bit : ORIGIN = 0x0100, LENGTH = 0x0100 /* END=0x0200, size 256 */
  bsl              : ORIGIN = 0x1000, LENGTH = 0x0400 /* END=0x1400, size 1K as 1 1024-byte segments 
*/
  ram (wx)         : ORIGIN = 0x2000, LENGTH = 0x1000 /* END=0x2400, size 4K */
  rom (rx)         : ORIGIN = 0xc400, LENGTH = 0x3b80 /* END=0xff80, size 15K */
  signature        : ORIGIN = 0xff80, LENGTH = 0x0008 /* END=0xff88, size 8 as 1 8-byte segments */
  vectors          : ORIGIN = 0xff80, LENGTH = 0x0080 /* END=0x10000, size 128 as 64 2-byte segments 
*/
  /* Remaining banks are absent */
  infomem          : ORIGIN = 0x1800, LENGTH = 0x0200
  infoa            : ORIGIN = 0x0000, LENGTH = 0x0000
  infob            : ORIGIN = 0x0000, LENGTH = 0x0000
  infoc            : ORIGIN = 0x0000, LENGTH = 0x0000
  infod            : ORIGIN = 0x0000, LENGTH = 0x0000
  ram2 (wx)        : ORIGIN = 0x0000, LENGTH = 0x0000
  ram_mirror (wx)  : ORIGIN = 0x0000, LENGTH = 0x0000
  tinyram (wx)     : ORIGIN = 0x0000, LENGTH = 0x0000
  usbram (wx)      : ORIGIN = 0x0000, LENGTH = 0x0000
  far_rom          : ORIGIN = 0x00000000, LENGTH = 0x00000000
}
REGION_ALIAS("REGION_TEXT", rom);
REGION_ALIAS("REGION_DATA", ram);
REGION_ALIAS("REGION_FAR_ROM", far_rom); /* Legacy name, no longer used */
REGION_ALIAS("REGION_FAR_TEXT", far_rom);
REGION_ALIAS("REGION_FAR_DATA", ram2);
/* INCLUDE memory.x end */

/*INCLUDE periph.x begin */
/* This file supports MSP430FR2433 devices. */
/* Version: 1.193 */

/************************************************************
* STANDARD BITS
************************************************************/
/************************************************************
* STATUS REGISTER BITS
************************************************************/
/************************************************************
* CPU
************************************************************/
/************************************************************
* PERIPHERAL FILE MAP
************************************************************/
/************************************************************
* ADC
************************************************************/
__ADCCTL0            = 0x0700;
__ADCCTL0_L          = 0x0700;
__ADCCTL0_H          = 0x0701;
__ADCCTL1            = 0x0702;
__ADCCTL1_L          = 0x0702;
__ADCCTL1_H          = 0x0703;
__ADCCTL2            = 0x0704;
__ADCCTL2_L          = 0x0704;
__ADCCTL2_H          = 0x0705;
__ADCLO              = 0x0706;
__ADCLO_L            = 0x0706;
__ADCLO_H            = 0x0707;
__ADCHI              = 0x0708;
__ADCHI_L            = 0x0708;
__ADCHI_H            = 0x0709;
__ADCMCTL0           = 0x070A;
__ADCMCTL0_L         = 0x070A;
__ADCMCTL0_H         = 0x070B;
```

```
__ADCMEM0            = 0x0712;
__ADCMEM0_L          = 0x0712;
__ADCMEM0_H          = 0x0713;
__ADCIE              = 0x071A;
__ADCIE_L            = 0x071A;
__ADCIE_H            = 0x071B;
__ADCIFG             = 0x071C;
__ADCIFG_L           = 0x071C;
__ADCIFG_H           = 0x071D;
__ADCIV              = 0x071E;
__ADCIV_L            = 0x071E;
__ADCIV_H            = 0x071F;
/*************************************************************
* Backup Memory Module
*************************************************************/
__BAKMEM0            = 0x0660;
__BAKMEM0_L          = 0x0660;
__BAKMEM0_H          = 0x0661;
__BAKMEM1            = 0x0662;
__BAKMEM1_L          = 0x0662;
__BAKMEM1_H          = 0x0663;
__BAKMEM2            = 0x0664;
__BAKMEM2_L          = 0x0664;
__BAKMEM2_H          = 0x0665;
__BAKMEM3            = 0x0666;
__BAKMEM3_L          = 0x0666;
__BAKMEM3_H          = 0x0667;
__BAKMEM4            = 0x0668;
__BAKMEM4_L          = 0x0668;
__BAKMEM4_H          = 0x0669;
__BAKMEM5            = 0x066A;
__BAKMEM5_L          = 0x066A;
__BAKMEM5_H          = 0x066B;
__BAKMEM6            = 0x066C;
__BAKMEM6_L          = 0x066C;
__BAKMEM6_H          = 0x066D;
__BAKMEM7            = 0x066E;
__BAKMEM7_L          = 0x066E;
__BAKMEM7_H          = 0x066F;
__BAKMEM8            = 0x0670;
__BAKMEM8_L          = 0x0670;
__BAKMEM8_H          = 0x0671;
__BAKMEM9            = 0x0672;
__BAKMEM9_L          = 0x0672;
__BAKMEM9_H          = 0x0673;
__BAKMEM10           = 0x0674;
__BAKMEM10_L         = 0x0674;
__BAKMEM10_H         = 0x0675;
__BAKMEM11           = 0x0676;
__BAKMEM11_L         = 0x0676;
__BAKMEM11_H         = 0x0677;
__BAKMEM12           = 0x0678;
__BAKMEM12_L         = 0x0678;
__BAKMEM12_H         = 0x0679;
__BAKMEM13           = 0x067A;
__BAKMEM13_L         = 0x067A;
__BAKMEM13_H         = 0x067B;
__BAKMEM14           = 0x067C;
__BAKMEM14_L         = 0x067C;
__BAKMEM14_H         = 0x067D;
__BAKMEM15           = 0x067E;
__BAKMEM15_L         = 0x067E;
__BAKMEM15_H         = 0x067F;
/*************************************************************
* CRC Module
*************************************************************/
__CRCDI              = 0x01C0;
__CRCDI_L            = 0x01C0;
__CRCDI_H            = 0x01C1;
__CRCDIRB            = 0x01C2;
__CRCDIRB_L          = 0x01C2;
__CRCDIRB_H          = 0x01C3;
```

```
__CRCINIRES          = 0x01C4;
__CRCINIRES_L        = 0x01C4;
__CRCINIRES_H        = 0x01C5;
__CRCRESR            = 0x01C6;
__CRCRESR_L          = 0x01C6;
__CRCRESR_H          = 0x01C7;
/************************************************************
* CLOCK SYSTEM CONTROL
************************************************************/
__CSCTL0             = 0x0180;
__CSCTL0_L           = 0x0180;
__CSCTL0_H           = 0x0181;
__CSCTL1             = 0x0182;
__CSCTL1_L           = 0x0182;
__CSCTL1_H           = 0x0183;
__CSCTL2             = 0x0184;
__CSCTL2_L           = 0x0184;
__CSCTL2_H           = 0x0185;
__CSCTL3             = 0x0186;
__CSCTL3_L           = 0x0186;
__CSCTL3_H           = 0x0187;
__CSCTL4             = 0x0188;
__CSCTL4_L           = 0x0188;
__CSCTL4_H           = 0x0189;
__CSCTL5             = 0x018A;
__CSCTL5_L           = 0x018A;
__CSCTL5_H           = 0x018B;
__CSCTL6             = 0x018C;
__CSCTL6_L           = 0x018C;
__CSCTL6_H           = 0x018D;
__CSCTL7             = 0x018E;
__CSCTL7_L           = 0x018E;
__CSCTL7_H           = 0x018F;
__CSCTL8             = 0x0190;
__CSCTL8_L           = 0x0190;
__CSCTL8_H           = 0x0191;
/*************************************************************
* FRAM Memory
*************************************************************/
__FRCTL0             = 0x01A0;
__FRCTL0_L           = 0x01A0;
__FRCTL0_H           = 0x01A1;
__GCCTL0             = 0x01A4;
__GCCTL0_L           = 0x01A4;
__GCCTL0_H           = 0x01A5;
__GCCTL1             = 0x01A6;
__GCCTL1_L           = 0x01A6;
__GCCTL1_H           = 0x01A7;
/************************************************************
* HARDWARE MULTIPLIER 32Bit
************************************************************/
__MPY                = 0x04C0;
__MPY_L              = 0x04C0;
__MPY_H              = 0x04C1;
__MPYS               = 0x04C2;
__MPYS_L             = 0x04C2;
__MPYS_H             = 0x04C3;
__MAC                = 0x04C4;
__MAC_L              = 0x04C4;
__MAC_H              = 0x04C5;
__MACS               = 0x04C6;
__MACS_L             = 0x04C6;
__MACS_H             = 0x04C7;
__OP2                = 0x04C8;
__OP2_L              = 0x04C8;
__OP2_H              = 0x04C9;
__RESLO              = 0x04CA;
__RESLO_L            = 0x04CA;
__RESLO_H            = 0x04CB;
__RESHI              = 0x04CC;
__RESHI_L            = 0x04CC;
__RESHI_H            = 0x04CD;
```

```
__SUMEXT             = 0x04CE;
__SUMEXT_L           = 0x04CE;
__SUMEXT_H           = 0x04CF;
__MPY32L             = 0x04D0;
__MPY32L_L           = 0x04D0;
__MPY32L_H           = 0x04D1;
__MPY32H             = 0x04D2;
__MPY32H_L           = 0x04D2;
__MPY32H_H           = 0x04D3;
__MPYS32L            = 0x04D4;
__MPYS32L_L          = 0x04D4;
__MPYS32L_H          = 0x04D5;
__MPYS32H            = 0x04D6;
__MPYS32H_L          = 0x04D6;
__MPYS32H_H          = 0x04D7;
__MAC32L             = 0x04D8;
__MAC32L_L           = 0x04D8;
__MAC32L_H           = 0x04D9;
__MAC32H             = 0x04DA;
__MAC32H_L           = 0x04DA;
__MAC32H_H           = 0x04DB;
__MACS32L            = 0x04DC;
__MACS32L_L          = 0x04DC;
__MACS32L_H          = 0x04DD;
__MACS32H            = 0x04DE;
__MACS32H_L          = 0x04DE;
__MACS32H_H          = 0x04DF;
__OP2L               = 0x04E0;
__OP2L_L             = 0x04E0;
__OP2L_H             = 0x04E1;
__OP2H               = 0x04E2;
__OP2H_L             = 0x04E2;
__OP2H_H             = 0x04E3;
__RES0               = 0x04E4;
__RES0_L             = 0x04E4;
__RES0_H             = 0x04E5;
__RES1               = 0x04E6;
__RES1_L             = 0x04E6;
__RES1_H             = 0x04E7;
__RES2               = 0x04E8;
__RES2_L             = 0x04E8;
__RES2_H             = 0x04E9;
__RES3               = 0x04EA;
__RES3_L             = 0x04EA;
__RES3_H             = 0x04EB;
__MPY32CTL0          = 0x04EC;
__MPY32CTL0_L        = 0x04EC;
__MPY32CTL0_H        = 0x04ED;
/************************************************************
* PMM - Power Management System for FR2xx/FR4xx
************************************************************/
__PMMCTL0            = 0x0120;
__PMMCTL0_L          = 0x0120;
__PMMCTL0_H          = 0x0121;
__PMMCTL1            = 0x0122;
__PMMCTL1_L          = 0x0122;
__PMMCTL1_H          = 0x0123;
__PMMCTL2            = 0x0124;
__PMMCTL2_L          = 0x0124;
__PMMCTL2_H          = 0x0125;
__PMMIFG             = 0x012A;
__PMMIFG_L           = 0x012A;
__PMMIFG_H           = 0x012B;
__PMMIE              = 0x012E;
__PMMIE_L            = 0x012E;
__PMMIE_H            = 0x012F;
__PM5CTL0            = 0x0130;
__PM5CTL0_L          = 0x0130;
__PM5CTL0_H          = 0x0131;
/************************************************************
* DIGITAL I/O Port1/2 Pull up / Pull down Resistors
************************************************************/
```

```
__PAIN               = 0x0200;
__PAIN_L             = 0x0200;
__PAIN_H             = 0x0201;
__PAOUT              = 0x0202;
__PAOUT_L            = 0x0202;
__PAOUT_H            = 0x0203;
__PADIR              = 0x0204;
__PADIR_L            = 0x0204;
__PADIR_H            = 0x0205;
__PAREN              = 0x0206;
__PAREN_L            = 0x0206;
__PAREN_H            = 0x0207;
__PASEL0             = 0x020A;
__PASEL0_L           = 0x020A;
__PASEL0_H           = 0x020B;
__PASEL1             = 0x020C;
__PASEL1_L           = 0x020C;
__PASEL1_H           = 0x020D;
__PAIES              = 0x0218;
__PAIES_L            = 0x0218;
__PAIES_H            = 0x0219;
__PAIE               = 0x021A;
__PAIE_L             = 0x021A;
__PAIE_H             = 0x021B;
__PAIFG              = 0x021C;
__PAIFG_L            = 0x021C;
__PAIFG_H            = 0x021D;
__P1IV               = 0x020E;
__P2IV               = 0x021E;
/************************************************************
* DIGITAL I/O Port3 Pull up / Pull down Resistors
************************************************************/
__PBIN               = 0x0220;
__PBIN_L             = 0x0220;
__PBIN_H             = 0x0221;
__PBOUT              = 0x0222;
__PBOUT_L            = 0x0222;
__PBOUT_H            = 0x0223;
__PBDIR              = 0x0224;
__PBDIR_L            = 0x0224;
__PBDIR_H            = 0x0225;
__PBREN              = 0x0226;
__PBREN_L            = 0x0226;
__PBREN_H            = 0x0227;
__PBSEL0             = 0x022A;
__PBSEL0_L           = 0x022A;
__PBSEL0_H           = 0x022B;
__PBSEL1             = 0x022C;
__PBSEL1_L           = 0x022C;
__PBSEL1_H           = 0x022D;
/************************************************************
* Real-Time Clock (RTC) Counter
************************************************************/
__RTCCTL             = 0x0300;
__RTCCTL_L           = 0x0300;
__RTCCTL_H           = 0x0301;
__RTCIV              = 0x0304;
__RTCIV_L            = 0x0304;
__RTCIV_H            = 0x0305;
__RTCMOD             = 0x0308;
__RTCMOD_L           = 0x0308;
__RTCMOD_H           = 0x0309;
__RTCCNT             = 0x030C;
__RTCCNT_L           = 0x030C;
__RTCCNT_H           = 0x030D;
/************************************************************
* SFR - Special Function Register Module
************************************************************/
__SFRIE1             = 0x0100;
__SFRIE1_L           = 0x0100;
__SFRIE1_H           = 0x0101;
__SFRIFG1            = 0x0102;
```

```
__SFRIFG1_L          = 0x0102;
__SFRIFG1_H          = 0x0103;
__SFRRPCR            = 0x0104;
__SFRRPCR_L          = 0x0104;
__SFRRPCR_H          = 0x0105;
/************************************************************
* SYS - System Module
************************************************************/
__SYSCTL             = 0x0140;
__SYSCTL_L           = 0x0140;
__SYSCTL_H           = 0x0141;
__SYSBSLC            = 0x0142;
__SYSBSLC_L          = 0x0142;
__SYSBSLC_H          = 0x0143;
__SYSJMBC            = 0x0146;
__SYSJMBC_L          = 0x0146;
__SYSJMBC_H          = 0x0147;
__SYSJMBI0           = 0x0148;
__SYSJMBI0_L         = 0x0148;
__SYSJMBI0_H         = 0x0149;
__SYSJMBI1           = 0x014A;
__SYSJMBI1_L         = 0x014A;
__SYSJMBI1_H         = 0x014B;
__SYSJMBO0           = 0x014C;
__SYSJMBO0_L         = 0x014C;
__SYSJMBO0_H         = 0x014D;
__SYSJMBO1           = 0x014E;
__SYSJMBO1_L         = 0x014E;
__SYSJMBO1_H         = 0x014F;
__SYSBERRIV          = 0x0158;
__SYSBERRIV_L        = 0x0158;
__SYSBERRIV_H        = 0x0159;
__SYSUNIV            = 0x015A;
__SYSUNIV_L          = 0x015A;
__SYSUNIV_H          = 0x015B;
__SYSSNIV            = 0x015C;
__SYSSNIV_L          = 0x015C;
__SYSSNIV_H          = 0x015D;
__SYSRSTIV           = 0x015E;
__SYSRSTIV_L         = 0x015E;
__SYSRSTIV_H         = 0x015F;
__SYSCFG0            = 0x0160;
__SYSCFG0_L          = 0x0160;
__SYSCFG0_H          = 0x0161;
__SYSCFG1            = 0x0162;
__SYSCFG1_L          = 0x0162;
__SYSCFG1_H          = 0x0163;
__SYSCFG2            = 0x0164;
__SYSCFG2_L          = 0x0164;
__SYSCFG2_H          = 0x0165;
/************************************************************
* Timer0_A3
************************************************************/
__TA0CTL             = 0x0380;
__TA0CCTL0           = 0x0382;
__TA0CCTL1           = 0x0384;
__TA0CCTL2           = 0x0386;
__TA0R               = 0x0390;
__TA0CCR0            = 0x0392;
__TA0CCR1            = 0x0394;
__TA0CCR2            = 0x0396;
__TA0IV              = 0x03AE;
__TA0EX0             = 0x03A0;
/************************************************************
* Timer1_A3
************************************************************/
__TA1CTL             = 0x03C0;
__TA1CCTL0           = 0x03C2;
__TA1CCTL1           = 0x03C4;
__TA1CCTL2           = 0x03C6;
__TA1R               = 0x03D0;
__TA1CCR0            = 0x03D2;
```

```
__TA1CCR1            = 0x03D4;
__TA1CCR2            = 0x03D6;
__TA1IV              = 0x03EE;
__TA1EX0             = 0x03E0;
/************************************************************
* Timer2_A2
************************************************************/
__TA2CTL             = 0x0400;
__TA2CCTL0           = 0x0402;
__TA2CCTL1           = 0x0404;
__TA2R               = 0x0410;
__TA2CCR0            = 0x0412;
__TA2CCR1            = 0x0414;
__TA2IV              = 0x042E;
__TA2EX0             = 0x0420;
/************************************************************
* Timer3_A2
************************************************************/
__TA3CTL             = 0x0440;
__TA3CCTL0           = 0x0442;
__TA3CCTL1           = 0x0444;
__TA3R               = 0x0450;
__TA3CCR0            = 0x0452;
__TA3CCR1            = 0x0454;
__TA3IV              = 0x046E;
__TA3EX0             = 0x0460;
/************************************************************
* USCI A0
************************************************************/
__UCA0CTLW0          = 0x0500;
__UCA0CTLW0_L        = 0x0500;
__UCA0CTLW0_H        = 0x0501;
__UCA0CTLW1          = 0x0502;
__UCA0CTLW1_L        = 0x0502;
__UCA0CTLW1_H        = 0x0503;
__UCA0BRW            = 0x0506;
__UCA0BRW_L          = 0x0506;
__UCA0BRW_H          = 0x0507;
__UCA0MCTLW          = 0x0508;
__UCA0MCTLW_L        = 0x0508;
__UCA0MCTLW_H        = 0x0509;
__UCA0STATW          = 0x050A;
__UCA0RXBUF          = 0x050C;
__UCA0RXBUF_L        = 0x050C;
__UCA0RXBUF_H        = 0x050D;
__UCA0TXBUF          = 0x050E;
__UCA0TXBUF_L        = 0x050E;
__UCA0TXBUF_H        = 0x050F;
__UCA0ABCTL          = 0x0510;
__UCA0IRCTL          = 0x0512;
__UCA0IRCTL_L        = 0x0512;
__UCA0IRCTL_H        = 0x0513;
__UCA0IE             = 0x051A;
__UCA0IE_L           = 0x051A;
__UCA0IE_H           = 0x051B;
__UCA0IFG            = 0x051C;
__UCA0IFG_L          = 0x051C;
__UCA0IFG_H          = 0x051D;
__UCA0IV             = 0x051E;
/************************************************************
* USCI A1
************************************************************/
__UCA1CTLW0          = 0x0520;
__UCA1CTLW0_L        = 0x0520;
__UCA1CTLW0_H        = 0x0521;
__UCA1CTLW1          = 0x0522;
__UCA1CTLW1_L        = 0x0522;
__UCA1CTLW1_H        = 0x0523;
__UCA1BRW            = 0x0526;
__UCA1BRW_L          = 0x0526;
__UCA1BRW_H          = 0x0527;
__UCA1MCTLW          = 0x0528;
```

```
__UCA1MCTLW_L        = 0x0528;
__UCA1MCTLW_H        = 0x0529;
__UCA1STATW          = 0x052A;
__UCA1RXBUF          = 0x052C;
__UCA1RXBUF_L        = 0x052C;
__UCA1RXBUF_H        = 0x052D;
__UCA1TXBUF          = 0x052E;
__UCA1TXBUF_L        = 0x052E;
__UCA1TXBUF_H        = 0x052F;
__UCA1ABCTL          = 0x0530;
__UCA1IRCTL          = 0x0532;
__UCA1IRCTL_L        = 0x0532;
__UCA1IRCTL_H        = 0x0533;
__UCA1IE             = 0x053A;
__UCA1IE_L           = 0x053A;
__UCA1IE_H           = 0x053B;
__UCA1IFG            = 0x053C;
__UCA1IFG_L          = 0x053C;
__UCA1IFG_H          = 0x053D;
__UCA1IV             = 0x053E;
/************************************************************
* USCI B0
************************************************************/
__UCB0CTLW0          = 0x0540;
__UCB0CTLW0_L        = 0x0540;
__UCB0CTLW0_H        = 0x0541;
__UCB0CTLW1          = 0x0542;
__UCB0CTLW1_L        = 0x0542;
__UCB0CTLW1_H        = 0x0543;
__UCB0BRW            = 0x0546;
__UCB0BRW_L          = 0x0546;
__UCB0BRW_H          = 0x0547;
__UCB0STATW          = 0x0548;
__UCB0STATW_L        = 0x0548;
__UCB0STATW_H        = 0x0549;
__UCB0TBCNT          = 0x054A;
__UCB0TBCNT_L        = 0x054A;
__UCB0TBCNT_H        = 0x054B;
__UCB0RXBUF          = 0x054C;
__UCB0RXBUF_L        = 0x054C;
__UCB0RXBUF_H        = 0x054D;
__UCB0TXBUF          = 0x054E;
__UCB0TXBUF_L        = 0x054E;
__UCB0TXBUF_H        = 0x054F;
__UCB0I2COA0         = 0x0554;
__UCB0I2COA0_L       = 0x0554;
__UCB0I2COA0_H       = 0x0555;
__UCB0I2COA1         = 0x0556;
__UCB0I2COA1_L       = 0x0556;
__UCB0I2COA1_H       = 0x0557;
__UCB0I2COA2         = 0x0558;
__UCB0I2COA2_L       = 0x0558;
__UCB0I2COA2_H       = 0x0559;
__UCB0I2COA3         = 0x055A;
__UCB0I2COA3_L       = 0x055A;
__UCB0I2COA3_H       = 0x055B;
__UCB0ADDRX          = 0x055C;
__UCB0ADDRX_L        = 0x055C;
__UCB0ADDRX_H        = 0x055D;
__UCB0ADDMASK        = 0x055E;
__UCB0ADDMASK_L      = 0x055E;
__UCB0ADDMASK_H      = 0x055F;
__UCB0I2CSA          = 0x0560;
__UCB0I2CSA_L        = 0x0560;
__UCB0I2CSA_H        = 0x0561;
__UCB0IE             = 0x056A;
__UCB0IE_L           = 0x056A;
__UCB0IE_H           = 0x056B;
__UCB0IFG            = 0x056C;
__UCB0IFG_L          = 0x056C;
__UCB0IFG_H          = 0x056D;
__UCB0IV             = 0x056E;
```

```
/************************************************************
* WATCHDOG TIMER A
************************************************************/
__WDTCTL             = 0x01CC;
__WDTCTL_L           = 0x01CC;
__WDTCTL_H           = 0x01CD;
/************************************************************
* TLV Descriptors
************************************************************/
/************************************************************
* Interrupt Vectors (offset from 0xFF80 + 0x10 for Password)
************************************************************/
/************************************************************
* End of Modules
************************************************************/
/*INCLUDE periph.x end */

SECTIONS
{
  /* Read-only sections, merged into text segment.  */
  .hash            : { *(.hash)          }
  .dynsym          : { *(.dynsym)        }
  .dynstr          : { *(.dynstr)        }
  .gnu.version     : { *(.gnu.version)   }
  .gnu.version_d   : { *(.gnu.version_d) }
  .gnu.version_r   : { *(.gnu.version_r) }
  .rel.init      : { *(.rel.init)  }
  .rela.init     : { *(.rela.init) }
  .rel.fini      : { *(.rel.fini)  }
  .rela.fini     : { *(.rela.fini) }
  .rel.text      : { *(.rel.text .rel.text.* .rel.gnu.linkonce.t.*)        }
  .rela.text     : { *(.rela.text .rela.text.* .rela.gnu.linkonce.t.*)     }
  .rel.rodata    : { *(.rel.rodata .rel.rodata.* .rel.gnu.linkonce.r.*)    }
  .rela.rodata   : { *(.rela.rodata .rela.rodata.* .rela.gnu.linkonce.r.*) }
  .rel.data      : { *(.rel.data .rel.data.* .rel.gnu.linkonce.d.*)        }
  .rela.data     : { *(.rela.data .rela.data.* .rela.gnu.linkonce.d.*)     }
  .rel.bss       : { *(.rel.bss .rel.bss.* .rel.gnu.linkonce.b.*)          }
  .rela.bss      : { *(.rela.bss .rela.bss.* .rela.gnu.linkonce.b.*)       }
  .rel.ctors     : { *(.rel.ctors)  }
  .rela.ctors    : { *(.rela.ctors) }
  .rel.dtors     : { *(.rel.dtors)  }
  .rela.dtors    : { *(.rela.dtors) }
  .rel.got       : { *(.rel.got)    }
  .rela.got      : { *(.rela.got)   }
  .rel.plt       : { *(.rel.plt)    }
  .rela.plt      : { *(.rela.plt)   }
  .text :
  {
     . = ALIGN(2);
     KEEP(*(.init .init.*))
     KEEP(*(.init0))  /* Start here after reset.               */
     KEEP(*(.init1))  /* User definable.                       */
     KEEP(*(.init2))  /* Initialize stack.                     */
     KEEP(*(.init3))  /* Initialize hardware, user definable.  */
     KEEP(*(.init4))  /* Copy data to .data, clear bss.        */
     KEEP(*(.init5))  /* User definable.                       */
     KEEP(*(.init6))  /* C++ constructors.                     */
     KEEP(*(.init7))  /* User definable.                       */
     KEEP(*(.init8))  /* User definable.                       */
     KEEP(*(.init9))  /* Call main().                          */
     KEEP(*(.fini9))  /* Falls into here after main(). User definable.  */
     KEEP(*(.fini8))  /* User definable.                           */
     KEEP(*(.fini7))  /* User definable.                           */
     KEEP(*(.fini6))  /* C++ destructors.                          */
     KEEP(*(.fini5))  /* User definable.                           */
     KEEP(*(.fini4))  /* User definable.                           */
     KEEP(*(.fini3))  /* User definable.                           */
     KEEP(*(.fini2))  /* User definable.                           */
     KEEP(*(.fini1))  /* User definable.                           */
     KEEP(*(.fini0))  /* Infinite loop after program termination.  */
     KEEP(*(.fini .fini.*))
     . = ALIGN(2);
```

```
   __ctors_start = . ;
   KEEP(*(.ctors))
   __ctors_end = . ;
   __dtors_start = . ;
   KEEP(*(.dtors))
   __dtors_end = . ;
   . = ALIGN(2);
  *(.text .text.* .gnu.linkonce.t.*)
   . = ALIGN(2);
}  > REGION_TEXT
.rodata   :
{
   . = ALIGN(2);
  *(.rodata .rodata.* .gnu.linkonce.r.*)
   . = ALIGN(2);
}  > REGION_TEXT
 _etext = .; /* Past last read-only (loadable) segment */
.data   :
{
   . = ALIGN(2);
   PROVIDE (__data_start = .) ;
  *(.data .data.* .gnu.linkonce.d.*)
   . = ALIGN(2);
   _edata = . ;  /* Past last read-write (loadable) segment */
}  > REGION_DATA AT > REGION_TEXT
 PROVIDE (__data_load_start = LOADADDR(.data) );
 PROVIDE (__data_size = SIZEOF(.data) );
.bss   :
{
   PROVIDE (__bss_start = .) ;
  *(.bss .bss.*)
  *(COMMON)
   . = ALIGN(2);
   PROVIDE (__bss_end = .) ;
}  > REGION_DATA
 PROVIDE (__bss_size = SIZEOF(.bss) );
.noinit   :
{
   PROVIDE (__noinit_start = .) ;
  *(.noinit .noinit.*)
   . = ALIGN(2);
   PROVIDE (__noinit_end = .) ;
}  > REGION_DATA
 . = ALIGN(2);
 _end = . ;   /* Past last write (loadable) segment */
.infomem   :
{
  *(.infomem)
   . = ALIGN(2);
  *(.infomem.*)
}  > infomem
.infomemnobits   :
{
  *(.infomemnobits)
   . = ALIGN(2);
  *(.infomemnobits.*)
}  > infomem
.infoa   :
{
  *(.infoa .infoa.*)
}  > infoa
.infob   :
{
  *(.infob .infob.*)
}  > infob
.infoc   :
{
  *(.infoc .infoc.*)
}  > infoc
.infod   :
{
  *(.infod .infod.*)
  }  > infod
  .vectors  :
  {
     PROVIDE (__vectors_start = .) ;
    KEEP(*(.vectors*))
     _vectors_end = . ;
  }  > vectors
  .fartext :
  {
     . = ALIGN(2);
    *(.fartext)
     . = ALIGN(2);
    *(.fartext.*)
     _efartext = .;
  }  > REGION_FAR_ROM
  /* Stabs for profiling information*/
  .profiler 0 : { *(.profiler) }
  /* 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_frame    0 : { *(.debug_frame) }
  .debug_str      0 : { *(.debug_str) }
  .debug_loc      0 : { *(.debug_loc) }
  .debug_macinfo  0 : { *(.debug_macinfo) }
  /* DWARF 3 */
  .debug_pubtypes 0 : { *(.debug_pubtypes) }
  .debug_ranges   0 : { *(.debug_ranges) }
   PROVIDE (__stack = ORIGIN(ram) + LENGTH(ram));
   PROVIDE (__data_start_rom = _etext);
   PROVIDE (__data_end_rom   = _etext + SIZEOF (.data));
}
```