**MEMORY**

{

TINYRAM : origin = 0xA, length = 0x16

BSL : origin = 0x1000, length = 0x800

INFOD : origin = 0x1800, length = 0x80

INFOC : origin = 0x1880, length = 0x80

INFOB : origin = 0x1900, length = 0x80

INFOA : origin = 0x1980, length = 0x80

RAM : origin = 0x1C00, length = 0x1000

FRAM : origin = 0x4000, length = 0xBF80

FRAM2 : origin = 0x10000,length = 0x34000

JTAGSIGNATURE : origin = 0xFF80, length = 0x0004, fill = 0xFFFF

BSLSIGNATURE : origin = 0xFF84, length = 0x0004, fill = 0xFFFF

IPESIGNATURE : origin = 0xFF88, length = 0x0008, fill = 0xFFFF

INT00 : origin = 0xFF90, length = 0x0002

INT01 : origin = 0xFF92, length = 0x0002

INT02 : origin = 0xFF94, length = 0x0002

INT03 : origin = 0xFF96, length = 0x0002

INT04 : origin = 0xFF98, length = 0x0002

INT05 : origin = 0xFF9A, length = 0x0002

INT06 : origin = 0xFF9C, length = 0x0002

INT07 : origin = 0xFF9E, length = 0x0002

INT08 : origin = 0xFFA0, length = 0x0002

INT09 : origin = 0xFFA2, length = 0x0002

INT10 : origin = 0xFFA4, length = 0x0002

INT11 : origin = 0xFFA6, length = 0x0002

INT12 : origin = 0xFFA8, length = 0x0002

INT13 : origin = 0xFFAA, length = 0x0002

INT14 : origin = 0xFFAC, length = 0x0002

INT15 : origin = 0xFFAE, length = 0x0002

INT16 : origin = 0xFFB0, length = 0x0002

INT17 : origin = 0xFFB2, length = 0x0002

INT18 : origin = 0xFFB4, length = 0x0002

INT19 : origin = 0xFFB6, length = 0x0002

INT20 : origin = 0xFFB8, length = 0x0002

INT21 : origin = 0xFFBA, length = 0x0002

INT22 : origin = 0xFFBC, length = 0x0002

INT23 : origin = 0xFFBE, length = 0x0002

INT24 : origin = 0xFFC0, length = 0x0002

INT25 : origin = 0xFFC2, length = 0x0002

INT26 : origin = 0xFFC4, length = 0x0002

INT27 : origin = 0xFFC6, length = 0x0002

INT28 : origin = 0xFFC8, length = 0x0002

INT29 : origin = 0xFFCA, length = 0x0002

INT30 : origin = 0xFFCC, length = 0x0002

INT31 : origin = 0xFFCE, length = 0x0002

INT32 : origin = 0xFFD0, length = 0x0002

INT33 : origin = 0xFFD2, length = 0x0002

INT34 : origin = 0xFFD4, length = 0x0002

INT35 : origin = 0xFFD6, length = 0x0002

INT36 : origin = 0xFFD8, length = 0x0002

INT37 : origin = 0xFFDA, length = 0x0002

INT38 : origin = 0xFFDC, length = 0x0002

INT39 : origin = 0xFFDE, length = 0x0002

INT40 : origin = 0xFFE0, length = 0x0002

INT41 : origin = 0xFFE2, length = 0x0002

INT42 : origin = 0xFFE4, length = 0x0002

INT43 : origin = 0xFFE6, length = 0x0002

INT44 : origin = 0xFFE8, length = 0x0002

INT45 : origin = 0xFFEA, length = 0x0002

INT46 : origin = 0xFFEC, length = 0x0002

INT47 : origin = 0xFFEE, length = 0x0002

INT48 : origin = 0xFFF0, length = 0x0002

INT49 : origin = 0xFFF2, length = 0x0002

INT50 : origin = 0xFFF4, length = 0x0002

INT51 : origin = 0xFFF6, length = 0x0002

INT52 : origin = 0xFFF8, length = 0x0002

INT53 : origin = 0xFFFA, length = 0x0002

INT54 : origin = 0xFFFC, length = 0x0002

RESET : origin = 0xFFFE, length = 0x0002

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Specify the LEA memory map \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#define** LEASTACK\_SIZE 0x138

**MEMORY**

{

LEARAM : origin = 0x2C00, length = 0x1000 - LEASTACK\_SIZE

LEASTACK : origin = 0x3C00 - LEASTACK\_SIZE, length = LEASTACK\_SIZE

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* SPECIFY THE SECTIONS ALLOCATION INTO MEMORY \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**SECTIONS**

{

**GROUP**(RW\_IPE)

{

**GROUP**(READ\_WRITE\_MEMORY)

{

.TI.persistent : {} /\* For #pragma persistent \*/

**.cio** : {} /\* C I/O Buffer \*/

**.sysmem** : {} /\* Dynamic memory allocation area \*/

} **PALIGN**(0x0400), **RUN\_START**(fram\_rw\_start)

**GROUP**(IPENCAPSULATED\_MEMORY)

{

.ipestruct : {} /\* IPE Data structure \*/

.ipe : {} /\* IPE \*/

.ipe\_const : {} /\* IPE Protected constants \*/

.ipe:\_isr : {} /\* IPE ISRs \*/

} **PALIGN**(0x0400), **RUN\_START**(fram\_ipe\_start) **RUN\_END**(fram\_ipe\_end) **RUN\_END**(fram\_rx\_start)

} > 0x4000

**.cinit** : {} > FRAM /\* Initialization tables \*/

.binit : {} > FRAM /\* Boot-time Initialization tables \*/

**.pinit** : {} > FRAM /\* C++ Constructor tables \*/

.init\_array : {} > FRAM /\* C++ Constructor tables \*/

.mspabi.exidx : {} > FRAM /\* C++ Constructor tables \*/

.mspabi.extab : {} > FRAM /\* C++ Constructor tables \*/

**.text**:\_isr : {} > FRAM /\* Code ISRs \*/

**#ifndef** \_\_LARGE\_DATA\_MODEL\_\_

**.const** : {} > FRAM /\* Constant data \*/

**#else**

**.const** : {} >> FRAM | FRAM2 /\* Constant data \*/

**#endif**

**#ifndef** \_\_LARGE\_CODE\_MODEL\_\_

**.text** : {} > FRAM /\* Code \*/

**#else**

**.text** : {} >> FRAM2 | FRAM /\* Code \*/

**#endif**

**#ifdef** \_\_TI\_COMPILER\_VERSION\_\_

**#if** \_\_TI\_COMPILER\_VERSION\_\_ >= 15009000

**#ifndef** \_\_LARGE\_CODE\_MODEL\_\_

.TI.ramfunc : {} load=FRAM, run=RAM, table(BINIT)

**#else**

.TI.ramfunc : {} load=FRAM | FRAM2, run=RAM, table(BINIT)

**#endif**

**#endif**

**#endif**

.jtagsignature : {} > JTAGSIGNATURE

.bslsignature : {} > BSLSIGNATURE

**GROUP**(SIGNATURE\_SHAREDMEMORY)

{

.ipesignature : {} /\* IPE Signature \*/

.jtagpassword : {} /\* JTAG Password \*/

} > IPESIGNATURE

**.bss** : {} > RAM /\* Global & static vars \*/

**.data** : {} > RAM /\* Global & static vars \*/

.TI.noinit : {} > RAM /\* For #pragma noinit \*/

**.stack** : {} > RAM (HIGH) /\* Software system stack \*/

.tinyram : {} > TINYRAM /\* Tiny RAM \*/

/\* MSP430 INFO memory segments \*/

.infoA : type = **NOINIT**{} > INFOA

.infoB : type = **NOINIT**{} > INFOB

.infoC : type = **NOINIT**{} > INFOC

.infoD : type = **NOINIT**{} > INFOD

.leaRAM : {} > LEARAM /\* LEA RAM \*/

.leaStack : {} > LEASTACK (HIGH) /\* LEA STACK \*/

/\* MSP430 interrupt vectors \*/

.int00 : {} > INT00

.int01 : {} > INT01

.int02 : {} > INT02

.int03 : {} > INT03

.int04 : {} > INT04

.int05 : {} > INT05

.int06 : {} > INT06

.int07 : {} > INT07

.int08 : {} > INT08

.int09 : {} > INT09

.int10 : {} > INT10

.int11 : {} > INT11

.int12 : {} > INT12

.int13 : {} > INT13

.int14 : {} > INT14

.int15 : {} > INT15

.int16 : {} > INT16

.int17 : {} > INT17

LEA : { \* ( .int18 ) } > INT18 type = VECT\_INIT

PORT8 : { \* ( .int19 ) } > INT19 type = VECT\_INIT

PORT7 : { \* ( .int20 ) } > INT20 type = VECT\_INIT

EUSCI\_B3 : { \* ( .int21 ) } > INT21 type = VECT\_INIT

EUSCI\_B2 : { \* ( .int22 ) } > INT22 type = VECT\_INIT

EUSCI\_B1 : { \* ( .int23 ) } > INT23 type = VECT\_INIT

EUSCI\_A3 : { \* ( .int24 ) } > INT24 type = VECT\_INIT

EUSCI\_A2 : { \* ( .int25 ) } > INT25 type = VECT\_INIT

PORT6 : { \* ( .int26 ) } > INT26 type = VECT\_INIT

PORT5 : { \* ( .int27 ) } > INT27 type = VECT\_INIT

TIMER4\_A1 : { \* ( .int28 ) } > INT28 type = VECT\_INIT

TIMER4\_A0 : { \* ( .int29 ) } > INT29 type = VECT\_INIT

AES256 : { \* ( .int30 ) } > INT30 type = VECT\_INIT

RTC\_C : { \* ( .int31 ) } > INT31 type = VECT\_INIT

PORT4 : { \* ( .int32 ) } > INT32 type = VECT\_INIT

PORT3 : { \* ( .int33 ) } > INT33 type = VECT\_INIT

TIMER3\_A1 : { \* ( .int34 ) } > INT34 type = VECT\_INIT

TIMER3\_A0 : { \* ( .int35 ) } > INT35 type = VECT\_INIT

PORT2 : { \* ( .int36 ) } > INT36 type = VECT\_INIT

TIMER2\_A1 : { \* ( .int37 ) } > INT37 type = VECT\_INIT

TIMER2\_A0 : { \* ( .int38 ) } > INT38 type = VECT\_INIT

PORT1 : { \* ( .int39 ) } > INT39 type = VECT\_INIT

TIMER1\_A1 : { \* ( .int40 ) } > INT40 type = VECT\_INIT

TIMER1\_A0 : { \* ( .int41 ) } > INT41 type = VECT\_INIT

DMA : { \* ( .int42 ) } > INT42 type = VECT\_INIT

EUSCI\_A1 : { \* ( .int43 ) } > INT43 type = VECT\_INIT

TIMER0\_A1 : { \* ( .int44 ) } > INT44 type = VECT\_INIT

TIMER0\_A0 : { \* ( .int45 ) } > INT45 type = VECT\_INIT

ADC12\_B : { \* ( .int46 ) } > INT46 type = VECT\_INIT

EUSCI\_B0 : { \* ( .int47 ) } > INT47 type = VECT\_INIT

EUSCI\_A0 : { \* ( .int48 ) } > INT48 type = VECT\_INIT

WDT : { \* ( .int49 ) } > INT49 type = VECT\_INIT

TIMER0\_B1 : { \* ( .int50 ) } > INT50 type = VECT\_INIT

TIMER0\_B0 : { \* ( .int51 ) } > INT51 type = VECT\_INIT

COMP\_E : { \* ( .int52 ) } > INT52 type = VECT\_INIT

UNMI : { \* ( .int53 ) } > INT53 type = VECT\_INIT

SYSNMI : { \* ( .int54 ) } > INT54 type = VECT\_INIT

**.reset** : {} > RESET /\* MSP430 reset vector \*/

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* MPU/IPE SPECIFIC MEMORY SEGMENT DEFINITONS \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**#ifdef** \_IPE\_ENABLE

**#define** IPE\_MPUIPLOCK 0x0080

**#define** IPE\_MPUIPENA 0x0040

**#define** IPE\_MPUIPPUC 0x0020

// Evaluate settings for the control setting of IP Encapsulation

**#if** defined(\_IPE\_ASSERTPUC1)

**#if** defined(\_IPE\_LOCK ) && (\_IPE\_ASSERTPUC1 == 0x08))

fram\_ipe\_enable\_value = (IPE\_MPUIPENA | IPE\_MPUIPPUC |IPE\_MPUIPLOCK);

**#elif** defined(\_IPE\_LOCK )

fram\_ipe\_enable\_value = (IPE\_MPUIPENA | IPE\_MPUIPLOCK);

**#elif** (\_IPE\_ASSERTPUC1 == 0x08)

fram\_ipe\_enable\_value = (IPE\_MPUIPENA | IPE\_MPUIPPUC);

**#else**

fram\_ipe\_enable\_value = (IPE\_MPUIPENA);

**#endif**

**#else**

**#if** defined(\_IPE\_LOCK )

fram\_ipe\_enable\_value = (IPE\_MPUIPENA | IPE\_MPUIPLOCK);

**#else**

fram\_ipe\_enable\_value = (IPE\_MPUIPENA);

**#endif**

**#endif**

// Segment definitions

**#ifdef** \_IPE\_MANUAL // For custom sizes selected in the GUI

fram\_ipe\_border1 = (\_IPE\_SEGB1>>4);

fram\_ipe\_border2 = (\_IPE\_SEGB2>>4);

**#else** // Automated sizes generated by the Linker

fram\_ipe\_border2 = fram\_ipe\_end >> 4;

fram\_ipe\_border1 = fram\_ipe\_start >> 4;

**#endif**

fram\_ipe\_settings\_struct\_address = Ipe\_settingsStruct >> 4;

fram\_ipe\_checksum = ~((fram\_ipe\_enable\_value & fram\_ipe\_border2 & fram\_ipe\_border1) | (fram\_ipe\_enable\_value & ~fram\_ipe\_border2 & ~fram\_ipe\_border1) | (~fram\_ipe\_enable\_value & fram\_ipe\_border2 & ~fram\_ipe\_border1) | (~fram\_ipe\_enable\_value & ~fram\_ipe\_border2 & fram\_ipe\_border1));

**#endif**

**#ifdef** \_MPU\_ENABLE

**#define** MPUPW (0xA500) /\* MPU Access Password \*/

**#define** MPUENA (0x0001) /\* MPU Enable \*/

**#define** MPULOCK (0x0002) /\* MPU Lock \*/

**#define** MPUSEGIE (0x0010) /\* MPU Enable NMI on Segment violation \*/

\_\_mpu\_enable = 1;

// Segment definitions

**#ifdef** \_MPU\_MANUAL // For custom sizes selected in the GUI

mpu\_segment\_border1 = \_MPU\_SEGB1 >> 4;

mpu\_segment\_border2 = \_MPU\_SEGB2 >> 4;

mpu\_sam\_value = (\_MPU\_SAM0 << 12) | (\_MPU\_SAM3 << 8) | (\_MPU\_SAM2 << 4) | \_MPU\_SAM1;

**#else** // Automated sizes generated by Linker

**#ifdef** \_IPE\_ENABLE //if IPE is used in project too

//seg1 = any read + write persistent variables

//seg2 = ipe = read + write + execute access

//seg3 = code, read + execute only

mpu\_segment\_border1 = fram\_ipe\_start >> 4;

mpu\_segment\_border2 = fram\_rx\_start >> 4;

mpu\_sam\_value = 0x1573; // Info R, Seg3 RX, Seg2 RWX, Seg1 RW

**#else**

mpu\_segment\_border1 = fram\_rx\_start >> 4;

mpu\_segment\_border2 = fram\_rx\_start >> 4;

mpu\_sam\_value = 0x1513; // Info R, Seg3 RX, Seg2 R, Seg1 RW

**#endif**

**#endif**

**#ifdef** \_MPU\_LOCK

**#ifdef** \_MPU\_ENABLE\_NMI

mpu\_ctl0\_value = MPUPW | MPUENA | MPULOCK | MPUSEGIE;

**#else**

mpu\_ctl0\_value = MPUPW | MPUENA | MPULOCK;

**#endif**

**#else**

**#ifdef** \_MPU\_ENABLE\_NMI

mpu\_ctl0\_value = MPUPW | MPUENA | MPUSEGIE;

**#else**

mpu\_ctl0\_value = MPUPW | MPUENA;

**#endif**

**#endif**

**#endif**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* INCLUDE PERIPHERALS MEMORY MAP \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

-**l** msp430fr5994.cmd