## 定义在RAM中的数组

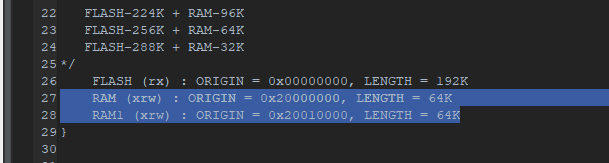
由于DMA基地址寄存器的高16位不会随着基地址自增而进位，为保证数组较大时每个数组元素的低16位地址都不会出现0XFFFF，故在LD文件与启动文件中将RAM分为两块（如下图分为RAM、RAM1），当需要的数组较大时，需将数组分成两个数组分开存储，第一块中正常定义数组，因为只有64K，如有其它静态变量，则可定义的数组大小要适当调小；第二块中定义需要特殊操作，具体如下：

1. Link.ld中需更改处：

1）将128KRAM分两块各64K：

RAM (xrw) : ORIGIN = 0x20000000, LENGTH = 64K

RAM1 (xrw) : ORIGIN = 0x20010000, LENGTH = 64K



2）如图此处添加：

.dalign1 :

{

. = ALIGN(4);

PROVIDE(\_data1\_vma = .);

} >RAM1 AT>FLASH

.dlalign1 :

{

. = ALIGN(4);

PROVIDE(\_data1\_lma = .);

} >FLASH AT>FLASH

.data1 :

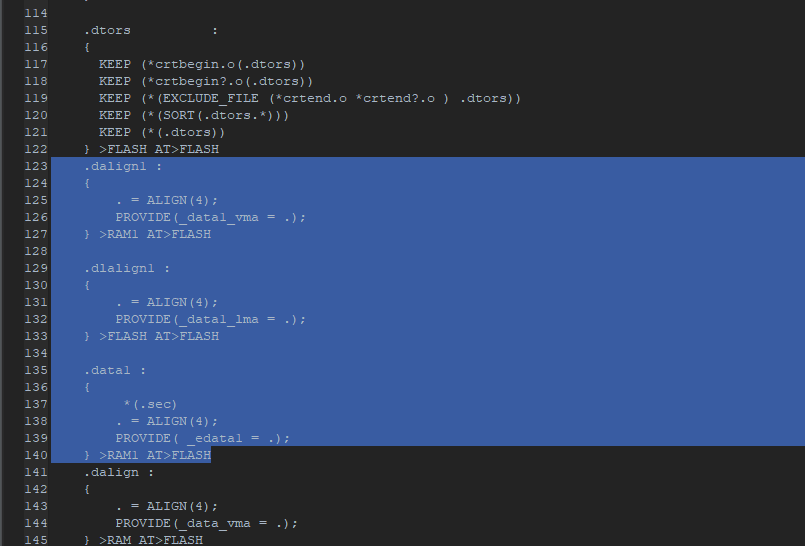
{

\*(.sec)

. = ALIGN(4);

PROVIDE( \_edata1 = .);

} >RAM1 AT>FLASH



1. startup\_ch32v30x\_D8C.S启动文件需要更改：

2:

/\* Load data section from flash to RAM1 \*/

la a0, \_data1\_lma

la a1, \_data1\_vma

la a2, \_edata1

bgeu a1, a2, 2f

1:

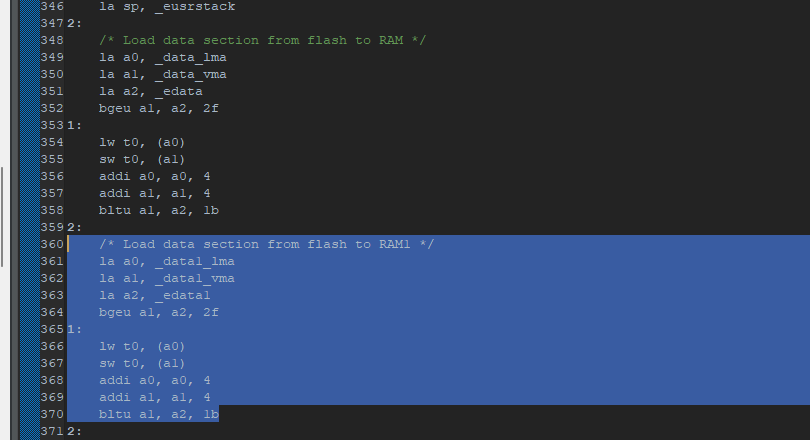
lw t0, (a0)

sw t0, (a1)

addi a0, a0, 4

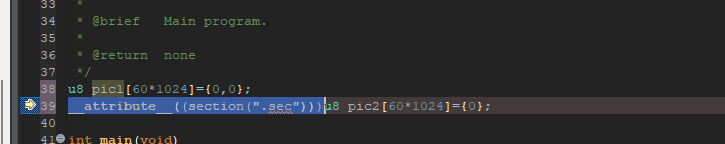
addi a1, a1, 4

bltu a1, a2, 1b



1. 定义数组.C或.h文件中需要添加：

\_\_attribute\_\_((section(".sec")))需要定义的数组前加此声明。

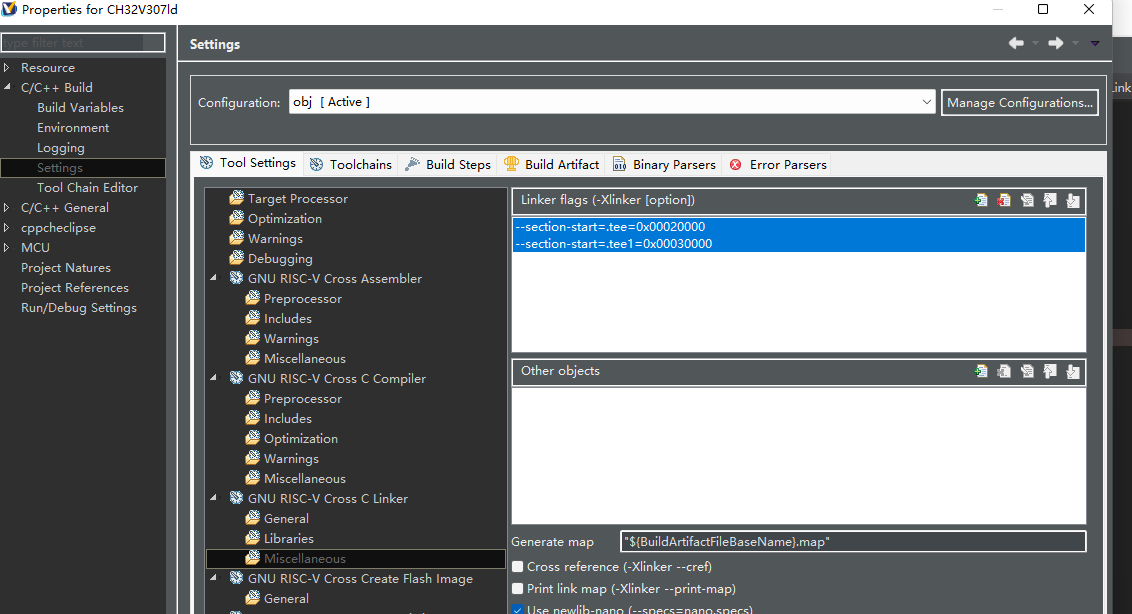


## 定义在FLASH中的数组：

同理定义在FLASH中的数组，为保证数组较大时每个数组元素的低16位地址都不会出现0XFFFF，可以通过如下操作指定数组的首地址：

1. 此处添加两个段：--section-start=.tee=0x00020000

--section-start=.tee1=0x00030000代码其他部分不能超过64K（0x080000000-0x080010000）。



二、定义数组.C或.h文件中需要添加：\_\_attribute\_\_((section(".tee")))（注意添加位置）

