

AN1503 应用说明

变量定义到MCU内部FLASH的方法

2018.02

Author: Zheng Pan



Revision History

NO	日期	描述	作者
1	2018-2	新建	郑攀

1. 概要

有一些应用中需要将数据常量放入 Flash 区域，用户在程序中通过查表操作直接访问 Flash 区域中的常量数据表。下面的部分详细介绍如何在 CDK 环境中实现此功能。

2. 定义变量到flash区

变量定义有以下两种方式。

一、变量地址由编译器分配的方式

CDK 中全局量加 `const` 关键字时会自动分配存储到 flash 中。此时该常量在 Flash 中的存储地址由编译器决定。在访问该常量时，可以通过获取此常量的地址指针的方式操作其内容。

参考如下代码：

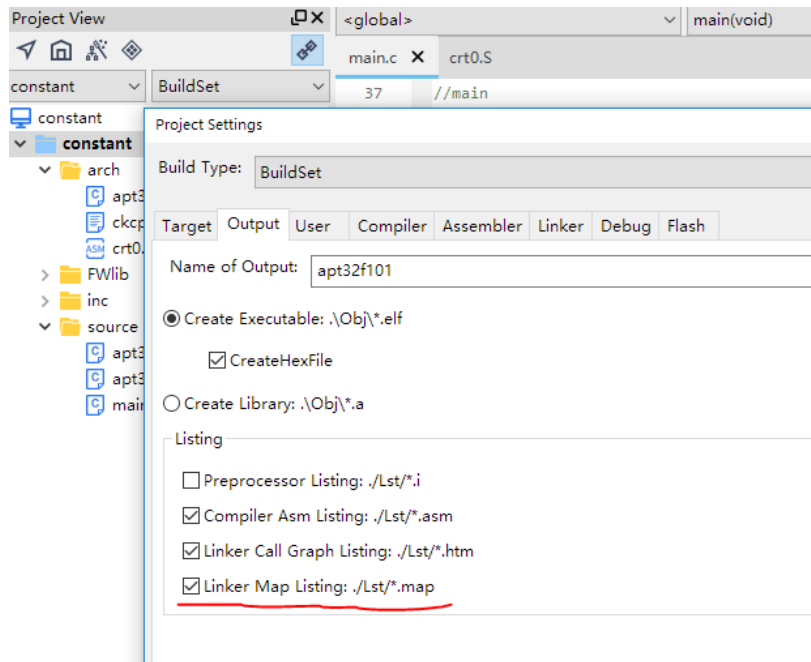
```
const uint32_t table[4] = {1,2,3,4};
/*****
//main
*****/
int main(void)
{
    APT32F101_init();
    volatile uint32_t *pTr = table;

    volatile uint32_t result, i;

    while(i++<4)
    {
        result = *pTr++ * 3;
    }
}
```

在代码中，我们首先定义了一个全局常量类型的数组，该数组共有 4 个元素。在编译后，该数组被存入 Flash 的 0x00002E60 地址，我们可以通过查看编译后生成的*.map 文件得到该信息。

*.map 文件存放于项目文件夹下的 Lst 目录中，若无法找到该文件，请确认项目的编译选项中已经开启了产生 map 文件选项。



Map 文件中存放了所有编译过程中，解析到的 label 所对应的 Flash 存储信息。打开 map 文件，搜索 table 变量名，我们可以看到如下结果：

```

__memset_fast      0x00002dbc   w F    164  .text
memset            0x00002dbc   w F    164  .text
table             0x00002e60   0      16   .rodata
_end_rodata       0x00002f2c   0      0    .rodata
_CHAR_            0x20000000   0      10   .data
    
```

文件结果表明 table 这个名字的 symbol 被定义在 Flash 内的只读字段，地址为 0x00002e60

通过上面的方式，我们已经很容易的将一个数据表存入 Flash 中。下面介绍如果在主程序中，调用该数据表。

在主程序中，我们可以声明一个指针型变量 pTr，将该变量初始化为数据表的起始地址。通过该地址变量的加减，可以快速的直接访问到 Flash 内存存储的常量表中单个元素的内容。具体可以参考前面的例程。

二、指定存放地址的方式

1、在变量定义的地方按照如下格式定义变量，即在变量定义的前面对变量的存储段属性进行声明。

```
int VinFlash[10] __attribute__((section(".textcsky.VinFlash")));
```

```
int VinFlash[10] = {0,1,2,3,4,5,6,7,8,9};
```

2、链接文件 cpu.ld 里面添加相应的段说明内容，一个完整的 cpu.ld 文件示例如下：

```

MEMORY
{
    ROM(RX)  : ORIGIN = 0x00000000, LENGTH = 32K
    
```

```
CUSTDATA(RWX) : ORIGIN = 0x00007C00, LENGTH = 1k
```

```
RAM(RWX) : ORIGIN = 0x20000000, LENGTH = 2K
```

```
}
```

```
__kernel_stack = ORIGIN(RAM) + LENGTH(RAM) -8 ;
```

```
ENTRY(__start)
```

```
REGION_ALIAS("REGION_CUSTOM", CUSTDATA);
```

```
SECTIONS {
```

```
    .text :
```

```
    {
```

```
        . = ALIGN(0x4) ;
```

```
        *crt0.o (.text)
```

```
        *(.text)
```

```
    } >ROM
```

```
    .rodata :
```

```
    {
```

```
        . = ALIGN(0x4) ;
```

```
        *(.rodata)
```

```
        *(.rodata.*)
```

```
        . = ALIGN(0x4) ;
```

```
        _end_rodata = .;
```

```
    } >ROM
```

```
    .data : AT(_end_rodata)
```

```
    {
```

```
        . = ALIGN(0x4) ;
```

```
    _start_data = .;
```

```
*(.data);  
. = ALIGN(0x4);  
_end_data = .;  
}>RAM  
  
.bss :  
{  
    . = ALIGN(0x4);  
    _bss_start = .;  
    *(.sbss)  
    *(.sbss.*)  
    *(.scommon)  
    *(.bss)  
    *(.bss.*)  
    *(COMMON)  
    . = ALIGN(0x4);  
    _ebss = .;  
    _end = .;  
    end = .;  
}>RAM
```

```
.text.custom : {  
    . = ALIGN(0x4);  
    *(.textcsky.*)  
}> REGION_CUSTOM  
}
```