

在 IAR EWARM 中实现 ST STM32 Checksum 计算

介绍

本文介绍了如何设置 IAR ELF Tool，让其使用和 ST 公司 STM32 内置硬件 CRC 模块一样的方式来计算 CRC32 校验。

具体的设置方法将取决于你所使用的 IAR EWARM 版本，本文将介绍 EWARM 6.40 及更高版本的 ELF 工具的设置方法。

IAR ELF 工具设置

在项目名称上单击右键选择“Options..”进入设置，在 Linker->Checksum 标签页进行 CRC32 校验的相关选项设置。

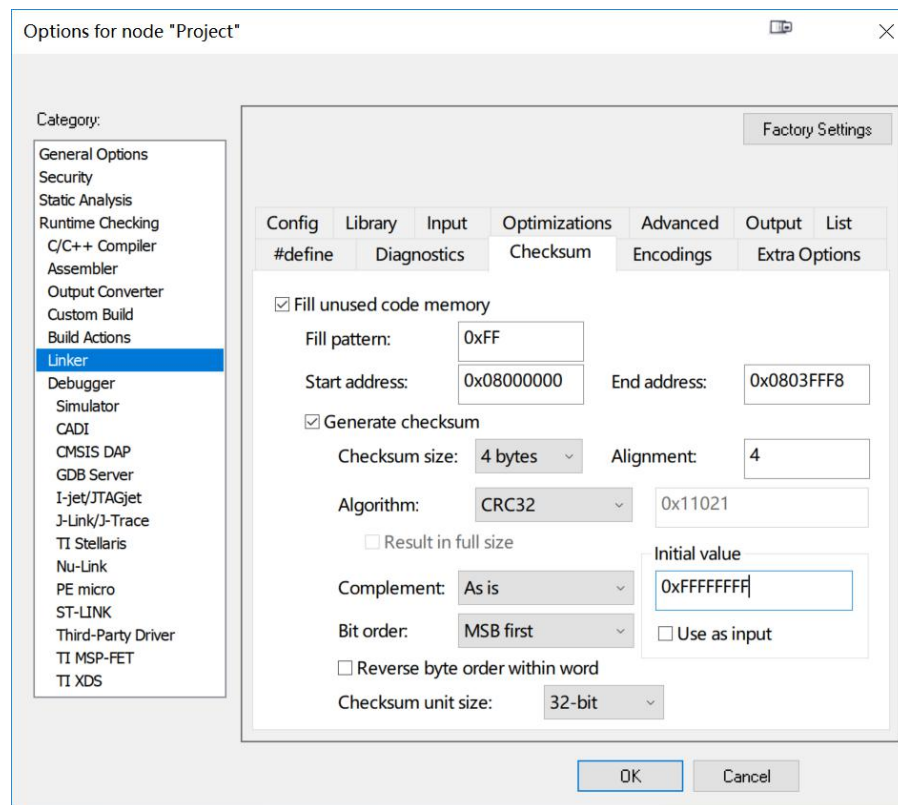


图- 1

校验大小：4 Bytes

对齐方式：4

校验算法：CRC32(0x4C11DB7)

补码：使用默认

字节顺序：高位先行

[取消选择]：在字节内反转字节顺序

校验和初始值：0xFFFFFFFF

[取消选择]：使用输入值作为初始值

校验单元大小：32-bit

示例工程

下面是两个 STM32F10x 的示例工程的下载链接地址。

EWARM 6.40 版本：
https://www.iar.com/contentassets/d8bbf6cb53c44ac39f10798d3c086d67/tn_64424_example_project_v6.40.zip

EWARM 7.80 版本：

https://www.iar.com/contentassets/d8bbf6cb53c44ac39f10798d3c086d67/tn_64424_example_project_v7.80.zip

从示例工程中摘录的源代码

这段源代码显示了如何调用 STM32 内置硬件 CRC32 模块来计算 CRC32 的校验值, 然后将该硬件模块计算的校验和与 IAR ELF Tool 计算的校验和进行比较。

```
1 #include "stm32f10x_crc.h"
2
3 extern uint32_t __checksum;
4
5 uint32_t calcCrc32(uint8_t* data, uint32_t len)
6 {
7     uint32_t* pBuffer = (uint32_t*) data;
8     uint32_t BufferLength = len/4;
9     uint32_t index = 0;
10
11     RCC_AHBPeriphClockCmd(RCC_AHBPeriph_CRC, ENABLE);
12
13     CRC_ResetDR();
14
15     for(index = 0; index < BufferLength; index++)
16     {
17         CRC->DR = pBuffer[index];
18     }
19
20     return CRC->DR;
21 }
22
```

```
23 void main(void)
24 {
25     ...
26
27     uint32_t valCrc32 =
28     calcCrc32((uint8_t*)0x08000000, 0x040000 - 4);
29
30     if (valCrc32 == __checksum)
31     {
32         // TBD
33     }
34     else
35     {
36         // TBD
37     }
38
39     ...
40 }
```