

32 位微控制器

HC32 系列专有代码读保护 PCROP

应用笔记

Rev1.00 2026 年 08 月

适用对象

产品系列	产品型号	产品系列	产品型号	产品系列	产品型号
HC32M413	ALL	HC32M458	ALL	HC32F431	ALL

声 明

- ★ 小华半导体有限公司（以下简称：“XHSC”）保留随时更改、更正、增强、修改小华半导体产品和/或本档的权利，恕不另行通知。用户可在下单前获取最新相关信息。XHSC 产品依据购销基本合同中载明的销售条款和条件进行销售。
- ★ 客户应针对您的应用选择合适的 XHSC 产品，并设计、验证和测试您的应用，以确保您的应用满足相应标准以及任何安全、安保或其它要求。客户应对此独自承担全部责任。
- ★ XHSC 在此确认未以明示或暗示方式授予任何知识产权许可。
- ★ XHSC 产品的转售，若其条款与此处规定不同，XHSC 对此类产品的任何保修承诺无效。
- ★ 任何带有“®”或“™”标识的图形或字样是 XHSC 的商标。所有其他在 XHSC 产品上显示的产品或服务名称均为其各自所有者的财产。
- ★ 本通知中的信息取代并替换先前版本中的信息。

©2026 小华半导体有限公司 保留所有权利

目 录

适用对象	2
声 明	3
目 录	4
1 概述	5
2 专有代码读保护 PCROP	6
2.1 PCROP 保护概述	6
2.2 如何使能 PCROP 保护	6
2.3 如何禁用 PCROP 保护	7
2.4 IP-code 与用户代码	7
3 PCROP 示例	8
3.1 示例概述	8
3.2 示例代码	8
3.2.1 示例代码修改	8
3.2.2 示例代码流程图	9
3.3 生成只执行 IP-Code	9
3.3.1 Keil	10
3.3.2 IAR	11
3.3.3 GCC	12
3.4 应用结果	14
4 注意事项	15
5 总结	16
版本修订记录	17

1 概述

本文主要介绍如何使用 HC32 系列的专有代码读保护 PCROP。

2 专有代码读保护 PCROP

2.1 PCROP 保护概述

XHSC 部分芯片的 FLASH 对专有知识产权代码提供读保护（Proprietary Code Read Out Protection, PCROP）功能，受到 PCROP 功能保护的代码称为 IP-Code。PCROP 功能应用于扇区，保护专有代码不被最终用户代码、调试器工具所读取。PCROP 有以下特点：

- 设定为 PCROP 保护区的数据只能被执行，不能被读取，否则发生读错误和总线错误。
- 使能 PCROP 后，不能执行全擦除，被保护的扇区不能被扇区擦除、编程，否则发生擦写错误和总线错误；被保护扇区的内容只能被执行，不能被读取，否则发生读错误和总线错误。
- PCROP 功能解除会触发主存储器全擦除（ICG 扇区不会被擦除）。
- PCROP 功能可与安全保护功能组合使用。

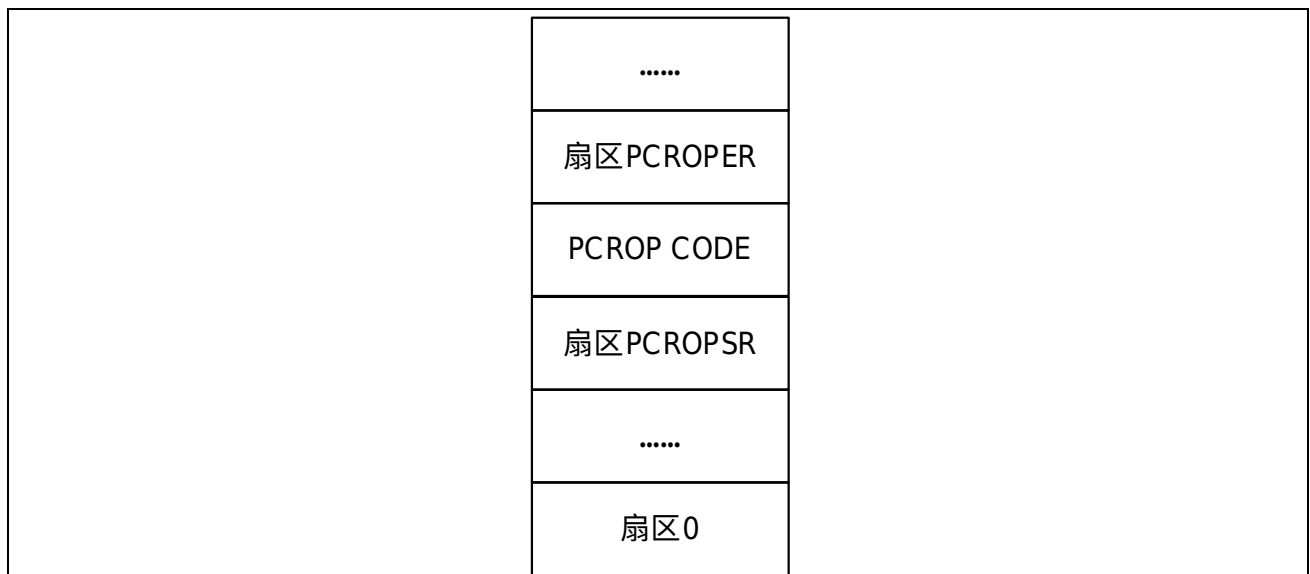


图 2-1 PCROP 启用后内存示意图

2.2 如何使能 PCROP 保护

PCROP 功能启用流程如下：

1. 将目标文件烧入保护区；
2. 按编程流程对 PCROP 使能地址编程 PCROP 功能使能码 0x2916 3505；
3. 按编程流程对 PCROP 保护起始扇区号地址和 PCROP 保护结束扇区号地址分别写入保护起始扇区号和保护结束扇区号；
4. 等待 EFM_FSR.OPTEND=1；
5. 进行系统复位，重新装载保护区设定；
6. 读取 FPCROPFG 寄存器确认保护范围设定正确；

7. 读取 FOCDPRTFG 寄存器确认 PCROP 功能已启用。PCROP 启用的条件是使能码有效且保护区域设定合法。

2.3 如何禁用 PCROP 保护

PCROP 功能解除流程方式一：

1. 对 PCROP 使能地址、PCROP 保护起始扇区号地址、PCROP 保护结束扇区号地址中的任一地址发起扇区擦除；
2. 等待 EFM_FSR.OPTEND=1；
3. 进行系统复位，重新装载保护区设定；
4. 读取 FOCDPRTFG 寄存器确认 PCROP 功能已解除。

PCROP 功能解除流程方式二：

1. 通过 DBGCFMCCTL.PCROPERSREQ 发起 PCROP 解除，该请求在 Flash 安全保护激活时仍可生效；
2. 通过 DBGCFMCCTL.PCROPERSACK 确认 PCROP 已解除成功；
3. 进行系统复位，重新装载保护区设定；
4. 读取 FOCDPRTFG 寄存器确认 PCROP 功能已解除。

以 HC32M413 为例，若需要保护的地址范围是[0x4000, 0x6000)，则 PCROPSR=0x20, PCROPER=0x2F。设置的 PCROPSR、PCROPER 都是包含在保护范围内的，第 PCROPER+1 个扇区才是保护范围之外。PCROPER 不能超过 Flash 主存储器容量，否则 PCROP 功能设置失败。

2.4 IP-code 与用户代码

由于 PCROP 区域只能放置指令，不能放置任何数据，因此用户在编排 PCROP 区域的代码时，须配置编译程序来产生只执行的代码以避免数据产生。

受保护的 IP-Code 可以被用户代码调用，同样的，IP-code 也可以调用用户代码中的函数，在这种情况下，IP-Code 将会包含这些函数的地址，允许 PC（程序计数器）在执行 IP-Code 时跳转到这些函数。一旦 PCROP 被使能，被 PCROP 调用的用户代码函数的地址就不能被改变，即这些函数的地址必须固定下来。因此应将所有与 IP-Code 相关联的函数都一起编译到 PCROP 区域内，或将它们编译到用户代码区的固定地址。

3 PCROP 示例

3.1 示例概述

本例展示 EFM 的 PCROP 功能，执行位于 PCROP 内的指令代码、访问 PCROP 内的数据，并通过 LED 灯观察现象，基于 XHSC 官网 HC32M413_DDL_Rev1.0.0 的 efm_pcrop 示例进行修改、调试和验证。

3.2 示例代码

3.2.1 示例代码修改

将 efm_pcrop 示例中按键修改为中断模式，并在主循环 for 中增加蓝灯循环闪烁代码如下：

```
BSP_KEY_Init(BSP_KEY_MD_EXTINT);

for (;;) {
    BSP_LED_Toggle(LED_BLUE);
    Pcrop_DelayMS(DELAY_MS);

    if (SET == BSP_KEY_GetStatus(BSP_KEY_1, BSP_KEY_MD_EXTINT)) {
        /* Enable PCROP */
        EFM_PCROP_Cmd(EFM_PCROP_START_SECTOR, EFM_PCROP_END_SECTOR, ENABLE);
        /* Reset */
        for (; ) {
            DDL_DelayMS(DELAY_MS);
            NVIC_SystemReset();
        }
    }
    if (SET == BSP_KEY_GetStatus(BSP_KEY_2, BSP_KEY_MD_EXTINT)) {
        /* Read the data which at the protected sector, cause bus fault */
        (void)RW_MEM32(u32Addr);
    }
}
```

新建 pcrop.c 文件实现毫秒延时功能代码如下：

```
#if defined (__CC_ARM) /*!< ARM Compiler */
#pragma push
#pragma O0
#endif
__NO_OPTIMIZE void Pcrop_DelayMS(uint32_t u32Count)
{
    __IO uint32_t i;
    const uint32_t u32Cyc = (HCLK_VALUE + 6000UL - 1UL) / 6000UL;
```

```
while (u32Count-- > 0UL) {  
    i = u32Cyc;  
    while (i-- > 0UL) {  
    }  
}  
}  
#if defined (__CC_ARM) /*!< ARM Compiler */  
#pragma pop  
#endif
```

3.2.2 示例代码流程图

示例代码流程如下图设置：

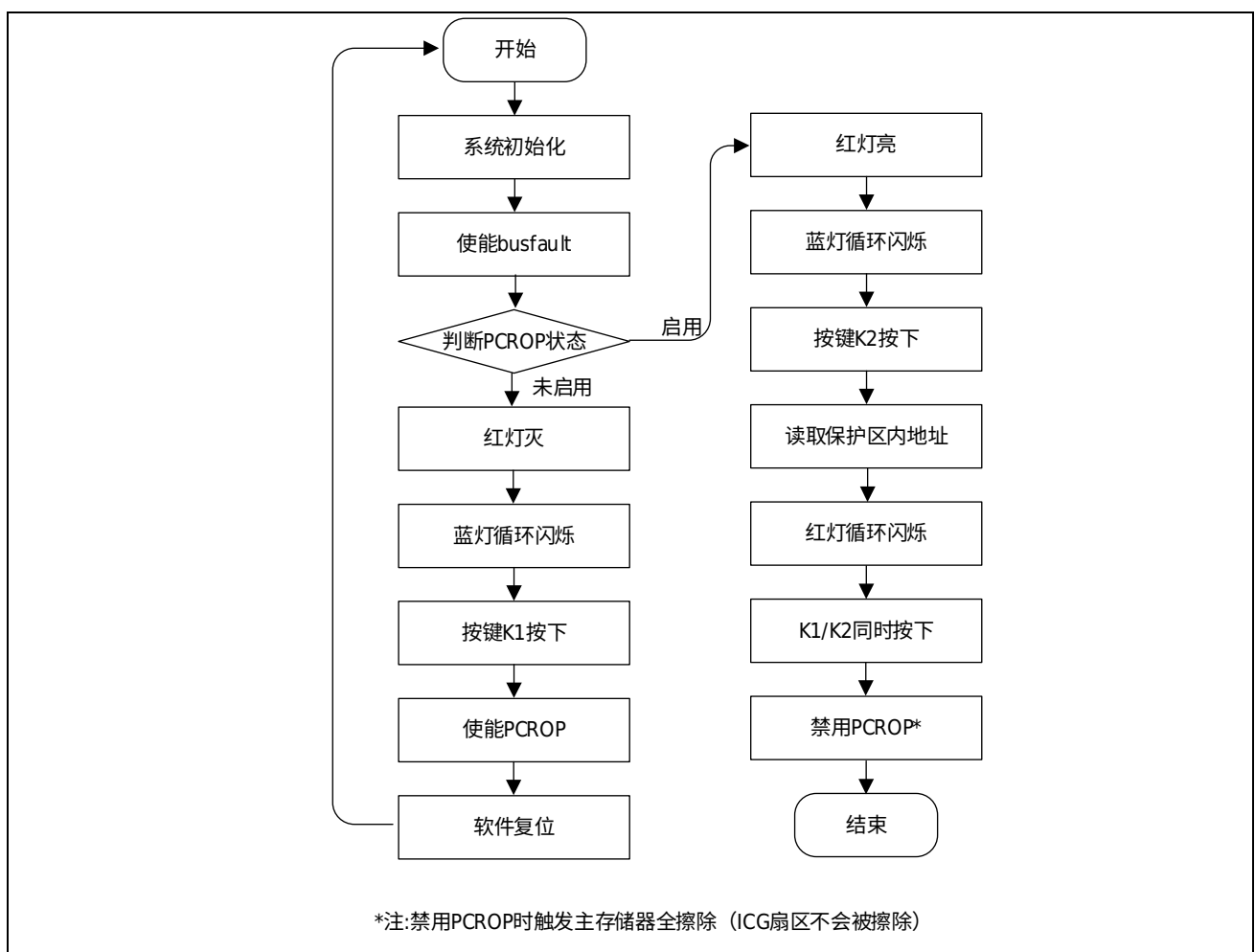


图 3-1 示例代码流程图

3.3 生成只执行 IP-Code

每个工具链都有自己的选项，来防止编译器生成文字池和分支表。例如 Keil 有 Execute-only Code 选项，IAR 有 No data reads in code memory 选项，GCC 有 -mpure-code 编译 flag。另外需要修改分散加载文件将这些代码编译到指定区域。

3.3.1 Keil

对于 Keil，请右键单击文件组，选择 Options for Group ‘pcrop’，如下图所示：

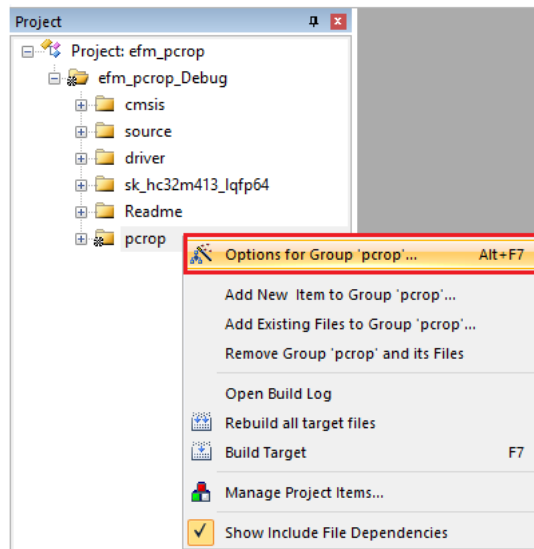


图 3-2 Keil 设置文件组 “options”

选择 C/C++ 选项组，勾选 execute-only code，如下图所示：

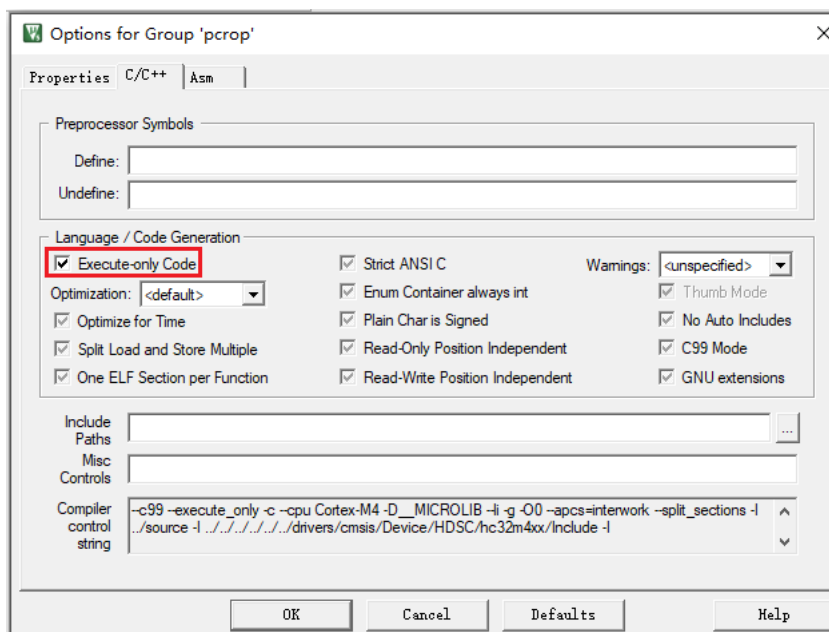


图 3-3 设置 Keil 选项 “Keil execute-only code”

修改 sct 文件，如下所示：

修改 ROM1 范围：

```
LR_IROM1 0x00000000 0x0001FE00
ER_IROM1 0x00000000 0x0001FE00
```

增加 PRCOP 段：

```
LR_PRCOP 0x1FE00 0x00000200{  
    ER_PRCOP 0x1FE00 0x00000200{  
        pcrop.o(+X0)  
    }  
}
```

3.3.2 IAR

对于 IAR，请右键单击文件组，选择 Options，如下图所示：

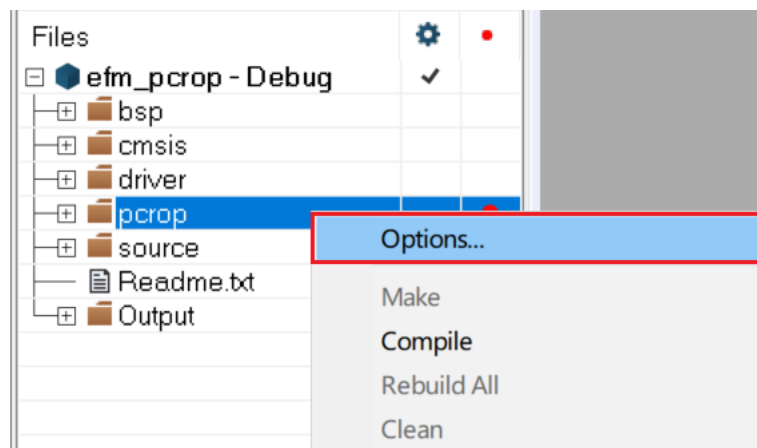


图 3-4 IAR 设置文件组 “options”

选择 C/C++ compiler 选项组勾选 override inherited settings, 并勾选 No data reads in code memory, 如下图所示：

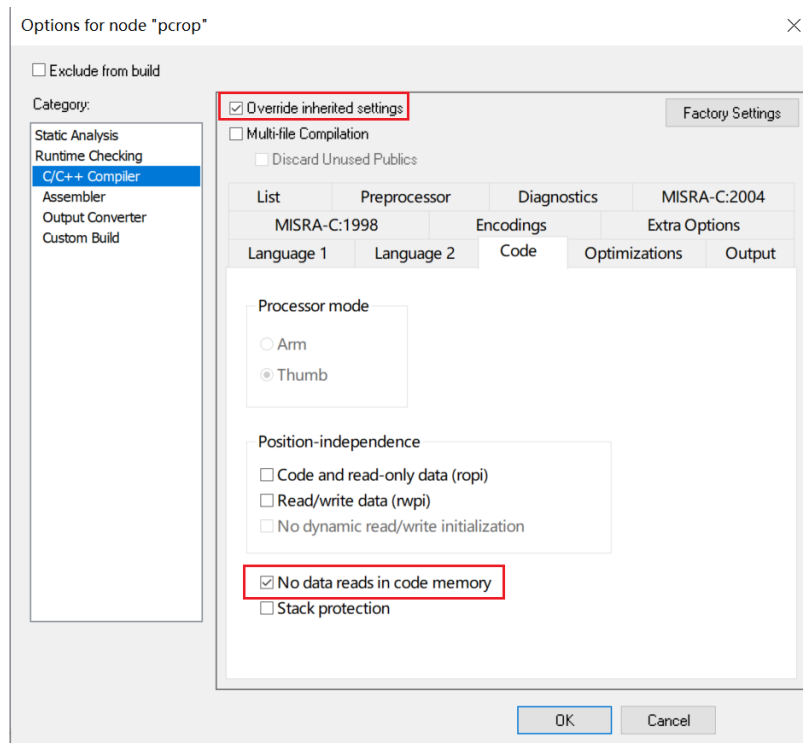


图 3-5 设置 IAR 选项组 “No data reads in code memory”

修改 icf 文件，如下所示：

```
define symbol __ICFEDIT_region_PCROP_start__ = 0x1FE00;  
define symbol __ICFEDIT_region_PCROP_end__ = 0x1FFFF;  
define region PCROP_region = mem:[from __ICFEDIT_region_PCROP_start__ to __ICFEDIT_region_PCROP_end__];  
place in PCROP_region { ro object pcrop.o };
```

3.3.3 GCC

对于 GCC，请右键单击文件组，并选择 properties，如下图所示：

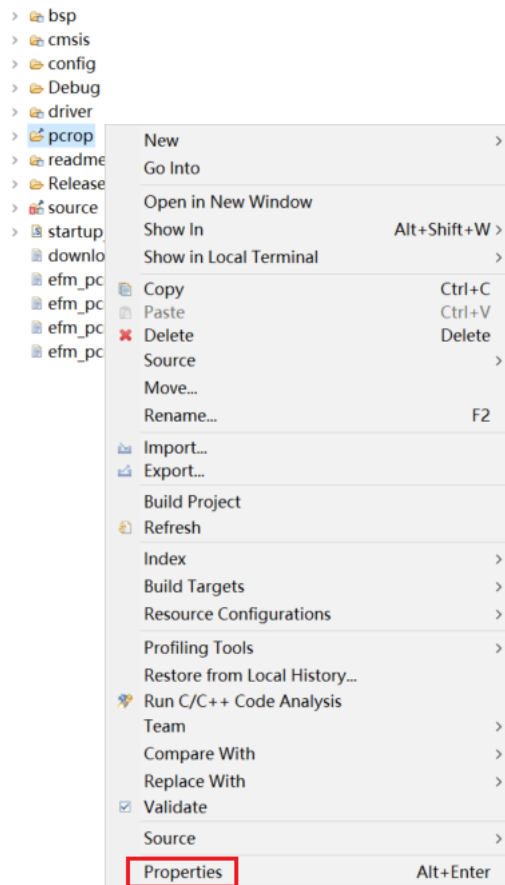


图 3-6 GCC 设置文件组 “Properties”

选择 C/C++ Build 选项组，并在 GNU ARM Cross Compiler 的 miscellaneous 中 Other compiler flag 选项下添加 “-mpure-code”，如下图所示：

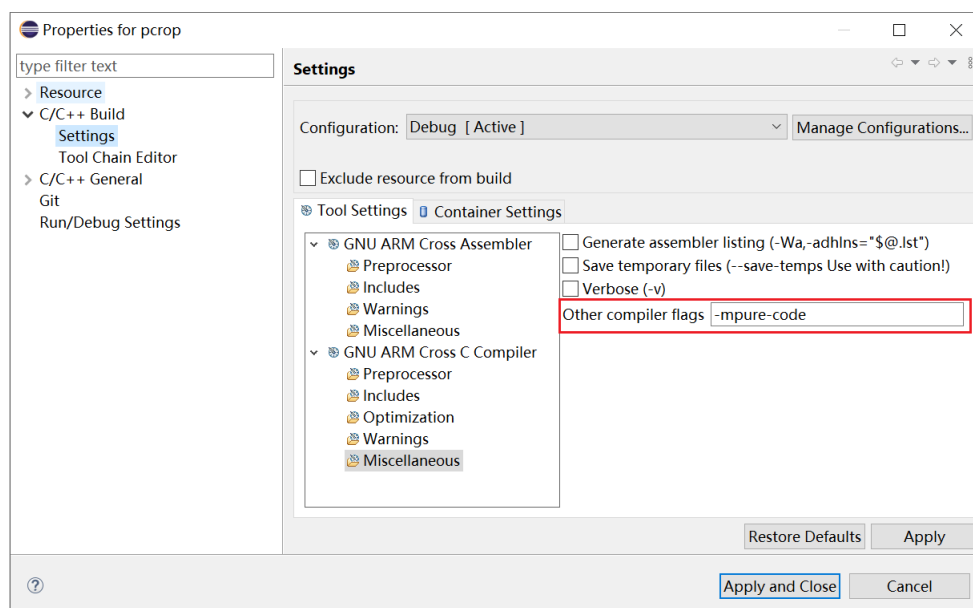


图 3-7 添加 GCC 编译 flag “-mpure-code”

修改 ld 文件，如下所示：

修改 MEMORY 定义：

```
MEMORY
{
    FLASH      (rx): ORIGIN = 0x00000000, LENGTH = 128K - 512
    PCROP      (rx): ORIGIN = 0x0001FE00, LENGTH = 512
    ICG        (rx): ORIGIN = 0x00020000, LENGTH = 20
    RAM        (rwx): ORIGIN = 0x1FFFE000, LENGTH = 16K
}
```

SECTIONS 增加 pcrop 段：

```
.pcrop 0x0001FE00 :
{
    . = ALIGN(4);
    __pcrop_start = .;
    KEEP(*pcrop.o(.text .text.*.rodata .rodata.*))
    . = ALIGN(4);
    __pcrop_end = .;
} >PCROP
ASSERT(__pcrop_end <= ORIGIN(PCROP) + LENGTH(PCROP), "PCROP region overflowed")
```

3.4 应用结果

该示例的执行如下所示：

1. 下载代码后，默认 PCROP 未生效，蓝灯亮；若 PCROP 已生效，则红灯亮；
2. 代码循环执行 PCROP 区域的指令，蓝灯闪烁；
3. 按下按键 K1 使能 PCROP 功能，PCROP 生效，红灯亮；
4. 在红灯亮的状态下按下按键 K2，读取受保护区域数据，触发总线错误，红灯闪烁；
5. 在红灯闪烁的状态下同时按下按键 K1 和 K2，解除 PCROP 功能，芯片全擦除，按下复位按键，所有灯熄灭。

如果编译器没有设置生成只执行的代码，则按下按键 K1 使能 PCROP 后直接触发总线错误，红灯闪烁。

4 注意事项

1. 所有 IP-Code 常量（例如整数、浮点数或字符串）必须放在 PCROP 保护的区域之外，因为它们不能被 D-Code 总线进行访问。
2. IP-Code 和用户区代码都有调用到的 C 语言标准库函数（如 `memset()` 及 `memcpy()` 等），可以将其编译到 PCROP 区域，或在 PCROP 区域内重新实现该函数。
3. 如果被 IP-Code 调用的函数位于 PCROP 保护区域之外（比如浮点运算库函数等因包含文字池无法编译到 PCROP 区域的函数），则被调用的函数应被编译到固定地址。
4. 不可将扇区 0 设置为保护区域，不可将包含中断向量表的扇区设置为保护区域。

5 总结

本文介绍了如何使用 HC32 系列的专有代码读保护 PCROP，并分别进行演示，给客户实际使用提供了参考。

版本修订记录

版本号	修订日期	修订内容
Rev1.00	2026/08/05	初版发布。