

应用笔记

Application Note

文档编号: **AN1134**

G32R5xx SDK SFO 库函数应用笔记

版本: **V1.1**

1 引言

本应用笔记描述 SFO 库函数及使用指导。

目录

1	引言	1
2	软件使用	3
3	步骤 1 添加静态库到 IDE.....	4
3.1	Keil MDK	4
3.2	IAR Embedded Workbench.....	5
3.3	Eclipse (Clang)	6
4	步骤 2 添加头文件	7
5	步骤 3 变量声明	8
6	步骤 4 MEP_ScaleFactor 初始化	9
7	步骤 5 应用代码	10
8	版本历史	11

2 软件使用

软件库函数 **SFO** () 用于计算支持 HRPWM 的 PWM 模块的 MEP 比例因子。比例因子是一个取值范围在 1-255 之间的整数，代表一个系统时钟周期内可用的微步边位置数。比例因子的值以一个名为 MEP_ScaleFactor 的整数变量返回。SFO () 软件库函数具体功能参见表格 1。

表格 1 比例因子值

软件库函数	功能描述	更新的变量
SFO ()	返回HRMSTEP寄存器中的MEP比例因子	MEP_ScaleFactor和HRMSTEP寄存器

3 步骤 1 添加静态库到 IDE

SFO 以预编译静态库形式提供。使用前须先按目标器件与工具链，把对应 “.lib” / “.a” 加入工程。库目录：

- G32R501: “libraries/calibration/hrpwm/g32r501/lib/”
- G32R502: “libraries/calibration/hrpwm/g32r502/lib/”

常用文件名（与 SDK 例程 “hrpwm_ex1_duty_sfo” 一致）见表格 2。

表格 2 三平台 SFO 静态库文件名

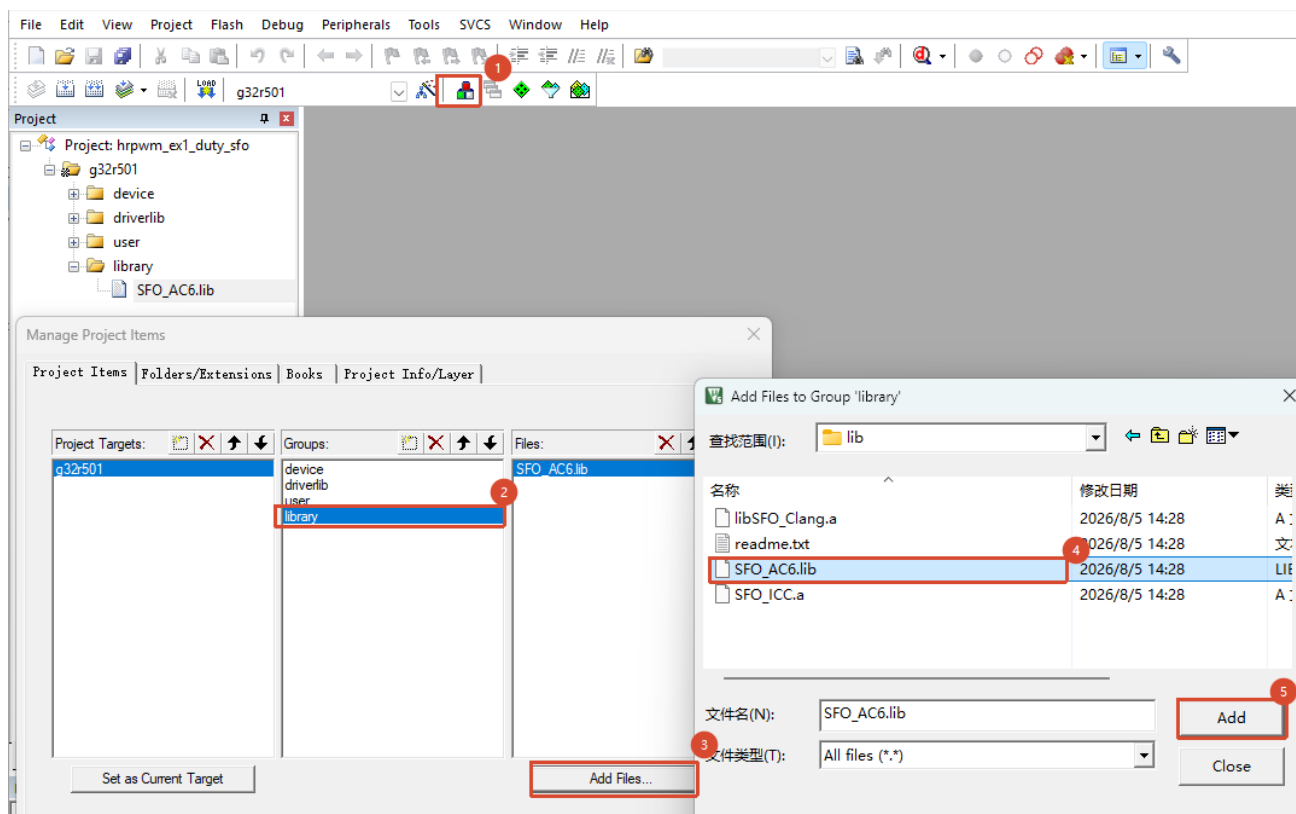
工具链	库文件
Keil MDK (Arm Compiler 6)	“SFO_AC6.lib”
IAR	“SFO_ICC.a”
Eclipse (Clang)	库搜索路径指向上述 “lib” 目录， Libraries 填写 “SFO_Clang”（不加 “lib” 前缀与后缀）

注意：必须选择与芯片一致的 “g32r501” 或 “g32r502” 子目录，不要混用。

3.1 Keil MDK

1. 打开工程，在 Project 窗口目标下右键，选择 **Add Existing Files to Group...**（或把库拖入某 Source Group）。
2. 浏览到 “libraries/calibration/hrpwm/<device>/lib/SFO_AC6.lib” 并添加。
3. 确认 Options for Target → Linker 未排除该库；重新编译。

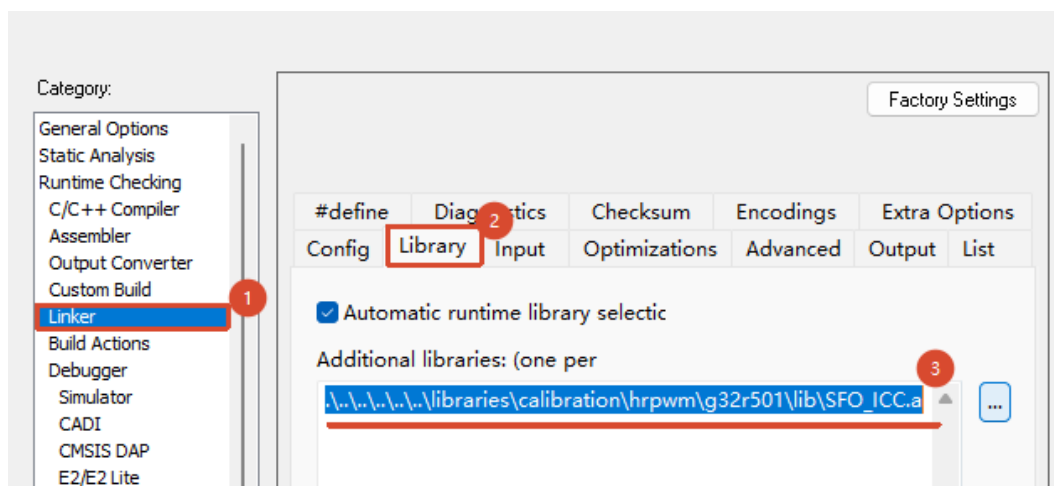
图 1 MDK 添加 SFOAC6.lib



3.2 IAR Embedded Workbench

1. 打开工程，在 Options → **Linker** → **Library**（或 Extra Options）中添加库文件完整路径。
2. 路径示例：
“\$PROJ_DIR\$\..\..\..\..\..\libraries\calibration\hrpwm\g32r502\lib\SFO_ICC.a”（相对层级数以本工程为准；也可使用绝对路径）。

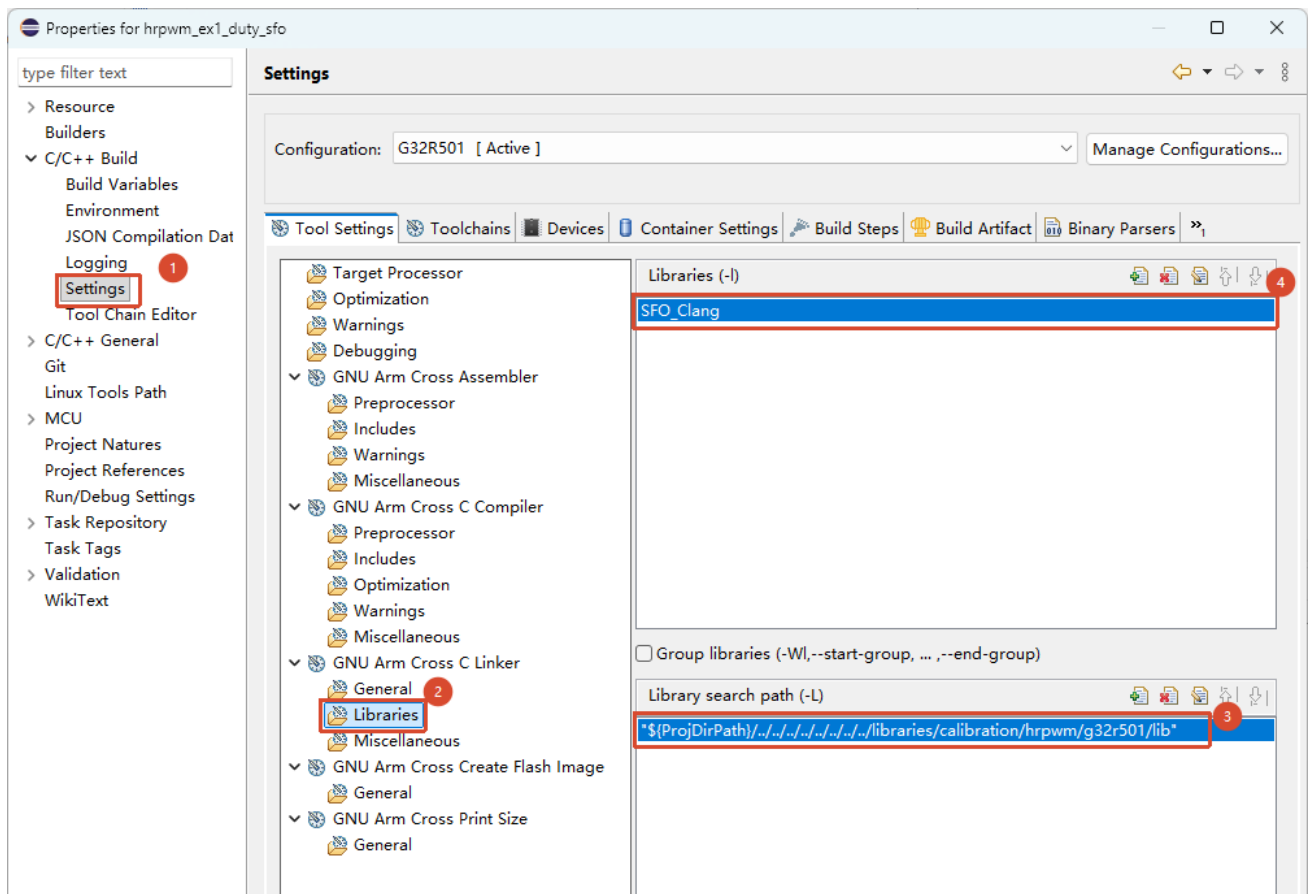
图 2 IAR 添加 SFOICC.a



3.3 Eclipse (Clang)

1. 右键工程 → **Properties** → **C/C++ Build** → **Settings** → **Tool Settings** → **GNU Arm Cross C Linker** → **Libraries**。
2. **Library search path (-L)** : 添加 ".../libraries/calibration/hrpwm/<device>/lib"。
3. **Libraries (-l)** : 添加 "SFO_Clang" (对应文件名一般为 "libSFO_Clang.a" 或工程约定的 "SFO_Clang" 库名, 以该目录实际文件为准)。
4. Apply 后重新 Build。

图 3 Eclipse 添加 SFOClang



完成静态库添加后, 再执行后续步骤 (头文件、变量与调用)。

4 步骤 2 添加头文件

在使用 SFO 库函数时，需包含 SFO_V1.h 文件。

// 示例 2.1 如何添加头文件

```
# include "SFO_V1.h"
```

注意：“SFO_V1.h”按器件分别位于“libraries/calibration/hrpwm/g32r501/include/”与“libraries/calibration/hrpwm/g32r502/include/”。请在 IDE 中把对应器件的“include”目录加入头文件搜索路径。

5 步骤 3 变量声明

声明整型变量 `MEP_ScaleFactor` 作为比例因子的值，如下所示。

```
// 示例 2.2 声明变量
int MEP_ScaleFactor = 0; // Global variable used by the SFO library
// Result can be used for all HRPWM channels
// This variable is also copied to HRMSTEP
// register by SFO() function.
uint16_t status = SFO_INCOMPLETE;
volatile uint32_t PWM[] = {0, PWM1_BASE, PWM2_BASE /* ... */};
```

6 步骤 4 MEP_ScaleFactor 初始化

SFO () 函数不要求初始化 MEP_ScaleFactor 比例因子值。在应用代码中使用 MEP_ScaleFactor 变量之前, 可以调用 SFO () 来驱动 MEP 校准模块进行计算并赋值 MEP_ScaleFactor。

在使用 MEP_ScaleFactor 变量之前, 以下代码作为一次性初始化代码的一部分。

```
// 示例 2.3 初始化比例因子值
while (status == SFO_INCOMPLETE)
{
    status = SFO();
    if (status == SFO_ERROR)
    {
        // SFO returns 2 if MEP steps/coarse step exceeds maximum of 255.
        while (1)
        {
        }
    }
}
```

7 步骤 5 应用代码

在应用程序运行时，可以预期设备温度和供电电压的波动。为了确保每个 PWM 模块都使用良好的比例因子，SFO 函数可以定期作为较慢的后台循环的一部分重复运行。示例代码如下：

```
// 示例 2.4 SFO 函数调用
void main(void)
{
    while (1)
    {
        status = SFO();
        if (status == SFO_ERROR)
        {
            // SFO returns 2 if MEP steps/coarse step exceeds maximum of 255.
            while (1)
            {
            }
        }
    }
}
```

注意：SFO 相关源码位于 “libraries/calibration/hrpwm/”（按 SDK 内 “g32r501”/“g32r502” 子目录选用）；例程请在 “driverlib/g32r501/examples/” 或 “driverlib/g32r502/examples/” 下查找 HRPWM / SFO 相关工程并打开源文件按步骤运行。

8

版本历史

表格 3 文件版本历史

日期	版本	变更历史
2025.1	1.0	● 新建
2026.8	1.1	● 适用产品列为 G32R501、G32R502 ● libraries 路径含 g32r501 与 g32r502 ● 新增步骤 1: MDK/IAR/Eclipse 三平台添加 SFO 静态库说明。

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为

准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现校正笔误或因测试环境差异所导致的误差，因此用户应当理解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中

所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿 responsibility，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

©2026 珠海极海半导体有限公司 – 保留所有权利