

应用笔记

Application Note

文档编号: **AN1133**

G32R5xx 中断应用笔记

版本: **V1.1**

1 引言

本应用笔记介绍 G32R5xx 中断以及应用程序设计。

目录

1	引言	1
2	NVIC	3
2.1	中断向量表与中断优先级	3
2.2	中断服务函数	5
3	EXTI	6
3.1	硬件中断选择	6
3.2	硬件事件选择	6
3.3	软件中断/事件选择	6
4	软件配置流程	7
5	版本历史	9

2 NVIC

NVIC (Nested Vectored Interrupt Controller), 是嵌套向量中断控制器, 控制着整个芯片中断相关的功能, 集成于内核中, G32R5xx 的内核是 Arm® Cortex®-M52。通过编程 NVIC 寄存器, 可以使能中断或者配置中断优先级等。

G32R501 与 G32R502 的外部中断通道数量, 在现行 CMSIS 器件头中均为 227 (IRQ 编号 0~226, 末项如 CTI_INT1_IRQn = 226)。向量表除外部 IRQ 外, 还包含 Cortex-M 系统异常槽位 (Reset、NMI、Fault、SVCall、PendSV、SysTick 等; 头文件中常以 NVIC_RAM_VECTOR_SIZE = 16 + 227 表达“系统槽位 + 外部 IRQ”总长度)。编写中断代码时, 请按目标器件打开对应头文件核对向量名, 不要混用 501/502 外设专有中断名。

2.1 中断向量表与中断优先级

在 G32R5xx 芯片编程时, 若使用到外设中断功能, 则需要将 ROM 中的中断向量表复制到 RAM 中, 且将 RAM 中的中断向量表重新映射到内核 VTOR 寄存器上, 以便于用户重新注册新的中断响应函数。

表格 1 为中断向量表节选:

表格 1 中断向量表 (节选)

异常类型	向量编号	优先级	描述
-	-	-	保留
Reset	-	-15	复位
NMI	-	-14	不可屏蔽中断
HardFault	-	-13	各种硬件故障
MemManage	-	-12	内存故障
BusFault	-	-11	总线故障
UsageFault	-	-10	指令执行故障
SecureFault	-	-9	数据安全故障
SVCall	-	-5	通用 SWIU 指令调用的系统服务
Debug Monitor	-	-4	调试功能异常
PendSV	-	-2	可挂起的系统服务
SysTick	-	-1	系统滴答定时器
...	...	...	...
DCCOMP interrupt	11	可设置	DCCOMP 中断
...	...	...	...
CTI INT0	225	可设置	CTI中断0

异常类型	向量编号	优先级	描述
CTI INT1	226	可设置	CTI中断1

Arm® Cortex®-M52 允许用户对每个异常/中断设置优先级，从而可以更加灵活地管理中斷响应。

2.1.1 抢占优先级与子优先级

可以把“中断优先级”想成两层标签，都由 `Interrupt_setPriorityGroup()` 先规定“多少位给抢占、多少位给子优先级”，再用 `Interrupt_setPriority(中断号, 抢占值, 子优先级值)` 写入具体数值。

- **抢占优先级 (Preempt priority)**：决定能不能打断别人。数值越小，优先级越高。若中断 A 的抢占优先级高于（数值更小）正在运行的中断 B，则 A 可以抢占 B（嵌套）。若两者抢占优先级相同，则 A 不能打断 B。
- **子优先级 (Subpriority)**：只在抢占优先级相同、且多个中断同时挂起时，用来决定谁先被响应。它不能用来打断一个已经在跑、且抢占优先级相同的中断。

举例（便于对照 SDK）：

1. 调用 `Interrupt_setPriorityGroup(INTERRUPT_PRIGROUP_PREEMPT_3_2_SUB_1_0);`：表示用优先级字段中的高位表示抢占、低位表示子优先级（具体位宽见 `interrupt.h` 中该宏注释）。现行 `interrupt_ex1_external` 等例程即采用该分组。
2. 再设置 `Interrupt_setPriority(INT_XINT1, 1, 0);` 与 `Interrupt_setPriority(INT_XINT2, 1, 0);`：两者抢占优先级同为 1、子优先级同为 0，因此谁也不能打断谁；若同时挂起，由硬件仲裁与挂起顺序决定响应先后。
3. 若把 XINT2 改为 `Interrupt_setPriority(INT_XINT2, 0, 0);`（抢占优先级 0 高于 1），则 XINT2 可以在 XINT1 的 ISR 执行过程中抢占进入。

G32R5xx driverlib 在 `interrupt.h` 中提供多种分组宏，例如：

- `INTERRUPT_PRIGROUP_PREEMPT_3_0_SUB_NO`
- `INTERRUPT_PRIGROUP_PREEMPT_3_1_SUB_0`
- `INTERRUPT_PRIGROUP_PREEMPT_3_2_SUB_1_0`（例程常用）
- `INTERRUPT_PRIGROUP_PREEMPT_3_SUB_2_0`
- `INTERRUPT_PRIGROUP_PREEMPT_NO_SUB_3_0`

每种配置定义了抢占优先级和子优先级的分配方式。工程中应先选定一种分组，再统一按该分组填写 `Interrupt_setPriority` 的后两个参数，避免“以为能嵌套、实际同抢占级不能嵌套”的误解。

2.2 中断服务函数

中断的四种状态: Active、Pending、Inactive、Active and pending。

- **Active:**

- 正在被处理。
- 尚未结束处理; 响应程序被更高优先级异常/中断的响应程序抢占。

- **Pending:** 已经生成了一个异常/中断, 但尚未激活。

- **Inactive:** 异常/中断尚未生成。

- **Active and pending:**

- 异常/中断的一个实例是活动状态, 异常/中断的第二个实例是挂起状态。
- 只有异步异常/中断可以同时处于活动和挂起状态。同步异常/中断要么是无效的, 要么是挂起的, 要么是活动的。

当异常/中断处于 **Active** 时, 此时程序会响应中断服务函数, 处理器根据中断向量表找到对应的中断服务函数, 并跳转到该函数执行。

在中断服务函数中, 程序会执行特定的代码来处理相关事件, 在处理完成后, 程序通常需要清除对应的中断标志, 以通知系统该中断已被处理, 准备接受下一次中断。

中断服务函数执行完毕后, 系统会恢复之前保存的上下文, 继续执行被中断的程序代码。

3 EXTI

使用 EXTI 中断线，应配置好并使能中断线才能产生中断。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写“1”使能中断请求。若使能了中断，当 EXTI 中断线上出现选定信号沿时，便会产生中断请求，对应的中断标志位也会置 1。在挂起寄存器的对应位写“1”，将清除该中断标志。

要产生事件，必须先配置好并使能事件线。根据需要的边沿检测设置 2 个触发寄存器，同时在事件屏蔽寄存器的相应位写“1”允许事件请求。当事件线上出现选定信号沿时，便会产生事件脉冲，对应的挂起位不会置 1。

所有 EXTI 线，可以通过在软件中对软件中断/事件寄存器写“1”，也可以产生软件中断/事件请求。

3.1 硬件中断选择

要将一根线配置为中断源，请执行以下步骤：

1. 配置中断线的屏蔽位。
2. 配置中断线的触发选择位。
3. 配置用于控制映射到扩展中断控制器（EXTI）的 NVIC IRQ 通道的使能位和屏蔽位，从而使任一中断线中的请求可以被正确地响应。

3.2 硬件事件选择

要将一根线配置为事件源，请执行以下步骤：

1. 配置事件线的屏蔽位。
2. 配置事件线的触发选择位。

3.3 软件中断/事件选择

可将任一可配置线配置为软件中断/事件线。要产生软件中断，必须执行以下步骤：

1. 配置中断/事件线的屏蔽位。
2. 在软件中断寄存器清除相应的请求位。
3. 在软件中断寄存器设置相应的请求位。
4. 在软件中断寄存器清除相应的请求位。

4 软件配置流程

本节以 I2CA FIFO 中断为例，说明 G32R5xx 中断软件配置流程：

1. “Interrupt_initModule();” 禁用全局中断，并失能所有中断。
2. “Interrupt_initVectorTable();” 初始化 NVIC 向量表。
3. “Interrupt_setPriorityGroup(INTERRUPT_PRIGROUP_PREEMPT_3_2_SUB_1_0);” 设置中断优先级组（与现行 SDK 例程 “interrupt_ex1_external” 等一致；旧文档中的 “INTERRUPT_PRIGROUP_PREEMPT_7_6_SUB_5_0” 已不再使用）。
4. “Interrupt_register(INT_I2CA_FIFO, &INT_I2CA_FIFO_IRQHandler);” 根据中断向量号，写入相关的中断服务函数入口地址至 RAM 区向量表，完成中断服务函数的注册。两个参数含义如下：
 - “INT_I2CA_FIFO”： “INT_I2CA_FIFO” 中断向量序号，即其在中断向量表对应的位置。
 - “INT_I2CA_FIFO_IRQHandler”： I2CA FIFO 中断服务函数，即把 “INT_I2CA_FIFO_IRQHandler” 函数入口地址写到 RAM 区向量表对应位置。
5. “Interrupt_setPriority(INT_I2CA_FIFO, 1, 0);” 设置中断优先级（第二个参数为抢占优先级，第三个参数为子优先级；含义见上一节）。
6. “Interrupt_enable(INT_I2CA_FIFO);” 使能 NVIC 中断。

示例代码如下（优先级分组宏已与现行 SDK 对齐）：

```
//
// Initialize NVIC and clears NVIC registers. Disables CPU interrupts.
//
Interrupt_initModule();
//
// Initialize the NVIC vector table with pointers to the shell Interrupt
// Service Routines (ISR).
//
Interrupt_initVectorTable();
//
// Interrupts that are used in this example are re-mapped to ISR functions
// found within this file.
//
Interrupt_register(INT_I2CA_FIFO, &INT_I2CA_FIFO_IRQHandler);
//
// Set the priority group to indicate the PREEMPT and SUB priority bits.
//
Interrupt_setPriorityGroup(INTERRUPT_PRIGROUP_PREEMPT_3_2_SUB_1_0);
```

```
//  
// Set the global and group priority to allow CPU interrupts  
// with higher priority  
//  
Interrupt_setPriority(INT_I2CA_FIFO,1,0);  
//  
// Enable interrupts required for this example  
//  
Interrupt_enable(INT_I2CA_FIFO);  
//  
// Enable Global Interrupt (INTM) and realtime interrupt (DBGM)  
//  
EINT;  
ERTM;
```

5 版本历史

表格 2 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 ● 分器件说明外部中断数量 ● 补充抢占/子优先级概念与举例 ● 优先级分组宏对齐现行 interrupt_ex1_external。

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