

## WinCE 下精简 ISA 总线硬件中断的使用方法

英创公司

2008 年 1 月

英创公司的 EM9000 系列产品是基于 ARM9 内核、预装了 WinCE 操作系统的高性能嵌入式模块。模块具有丰富的接口资源，其中包括有精简 ISA 总线和外部中断，以方便客户做专用电路的扩展。在 WinCE 下对于中断的处理是一个相对比较复杂的过程，当一个硬件中断发生时，它是被发送到内核来处理的，内核再调用中断服务例程来处理这个中断，在具体实现中断处理的过程中，需要调用系统提供的中断相关的函数，就涉及到内核函数的调用。而用户在使用 eVC 等工具软件进行应用程序开发时，是不能直接调用这些内核函数的。针对这一情况，为了方便客户对于外部中断的使用，英创公司设计完成了在内核中直接加载外部中断的驱动程序，一旦中断事件发生，驱动程序的中断线程将响应硬件中断同时产生一个事件，以通知上层的用户模式线程。按照这种方式导出一个定制的用户接口（共 4 个函数），用户只需要在应用软件中直接调用这些定制的用户接口函数即可方便地实现对于中断的应用处理，下面就介绍这些相关的接口函数。

### (1) HANDLE InstallExternIRQ( DWORD dwIRQNo );

功能描述：安装启动 ISA 总线外部中断。

输入参数 dwIRQNo：需要启动安装的外部中断号，目前对于 EM9000 和 EM9260 该中断号均为 1。

返回值 = NULL：安装 ISA 总线外部中断失败。

!= NULL：启动 ISA 总线外部中断返回的句柄。

### (2) HANDLE GetExternIRQEvent( HANDLE hIRQ );

功能描述：获取 ISA 总线外部中断的中断事件。

输入参数 hIRQ：调用函数 InstallExternIRQ( )所返回的外部中断句柄。

返回值 = NULL：操作失败。

!= NULL：ISA 总线外部中断的中断事件。

### (3) BOOL EnableExternIRQ( HANDLE hIRQ );

功能描述：使能 ISA 总线外部中断，允许下一次的中断。

输入参数 **hIRQ**: 函数 `InstallExternIRQ( )` 所返回的外部中断句柄。

返回值 = `TRUE`: 操作成功。

= `FASLE`: 操作失败。

#### (4) `BOOL UninstallExternIRQ( HANDLE hIRQ );`

功能描述: 卸载关闭 ISA 总线外部中断。

输入参数 **hIRQ**: 函数 `InstallExternIRQ( )` 所返回的外部中断句柄。

返回值 = `TRUE`: 操作成功。

= `FASLE`: 操作失败。

这四个函数定义在 `IRQ_API.h` 文件下, 相应的 `IRQ_API.LIB` 已经直接打包在 `SDK` 中, 用户直接安装 `ARM9` 模块的 `SDK` 文件即可。

作为应用程序来说, 可以通过函数 `InstallExternIRQ( ... )` 来启动外部中断, 并通过 `GetExternIRQEvent( ... )` 获取中断事件的句柄。用户自己定义一个处理线程, 该线程等待中断事件来触发, 一般用 `WaitForSingleObject( )` 来等待事件被触发, 当硬件中断发生时, 用户处理线程就可以完成必要的 `I/O` 操作来采集数据或处理数据了, 当中断处理完成时, 再调用函数 `EnableExternIRQ( )` 再次开启硬件中断。