

关于 AN4065 中 STM32F0 IAP 升级后的外部中断不响应问题

前言

客户在使用 STM32F051C8T6 的 IAP 功能时，发现有些端口（比如 PB 端口）的外部中断端口没有反应，但是有些还是可以的（比如 PA 端口），不知是何原因，客户也验证过应用代码在没有使用 IAP 功能时是没有问题的。客户是参考了我们的应用笔记 AN4065 来编写这部分代码。

问题分析

应用代码

通过查找客户的应用代码，发现如下 API 函数的调用：

```
/* Enable the SYSCFG peripheral clock*/
RCC_APB2PeriphResetCmd(RCC_APB2Periph_SYSCFG, ENABLE);
/* Remap SRAM at 0x00000000 */
SYSCFG_MemoryRemapConfig(SYSCFG_MemoryRemap_SRAM);
```

红色标记的 API 函数此处是有问题的，在此处应该将 `RCC_APB2PeriphResetCmd` 函数替换为 `RCC_APB2PeriphClockCmd` 函数。这个在 AN4065 的附件工程中也是这么写的，所以应该是代码的问题。修改后的代码如下，修改之后，程序正常运行了。

```
/* Enable the SYSCFG peripheral clock*/
RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);
/* Remap SRAM at 0x00000000 */
SYSCFG_MemoryRemapConfig(SYSCFG_MemoryRemap_SRAM);
```

深入研究

这个问题其实是这样的：因为 SYSCFG 寄存器的时钟被关闭了，所以才造成一些外部中断没有响应。

首先：我们看看 `RCC_APB2PeriphResetCmd(RCC_APB2Periph_SYSCFG, ENABLE)`；这个函数做了哪些操作，它是应该如何工作的？

在 `APB2PeriphResetCmd()` 函数中得知这个函数是设置 `RCC->APB2RSTR` 的对应位，如果设置了这个寄存器的对应位，那么对应的外设（或者是寄存器）的时钟就被关闭了，那么这个函数本来应该如何工作的呢？通过查看我们的库文件，这个函数只有在 `PPPP_DeInit()` 中才会使用，而且是一对儿一对儿的出现：

```
/* Enable SPI1 reset state */
RCC_APB2PeriphResetCmd(RCC_APB2Periph_SPI1, ENABLE);
/* Release SPI1 from reset state */
RCC_APB2PeriphResetCmd(RCC_APB2Periph_SPI1, DISABLE);
```

```
RCC_APB2PeriphResetCmd(RCC_APB2Periph_TIM1, ENABLE);
RCC_APB2PeriphResetCmd(RCC_APB2Periph_TIM1, DISABLE);
```

```
if (USARTx == USART1)
{
    RCC_APB2PeriphResetCmd(RCC_APB2Periph_USART1, ENABLE);
    RCC_APB2PeriphResetCmd(RCC_APB2Periph_USART1, DISABLE);
}
```

还有很多的例子，就不一一列出了。总之是为了简单的把一个外设或者是寄存器的值恢复到出厂设置，就可以通过调用两次这个函数来做到（比一个一个位清除要来的快），但是如果单独的只是调用了一次，比如客户代码中的情况，那么就是时钟被关闭了（这个可以简单的通过一个 GPIO 来测试，我已经测试过了，只要单独的设置上这个位，那么这个外设就肯定不工作了，无论你是不是开启这个外设的时钟在 RCC_AHBENR 中），所以这个寄存器该起到的效果都没有了。关于 RCC_APB2PeriphClockCmd() 这个函数就很简单了，只是打开对应外设的时钟而已，就不说了。

其次，我们再来看看这个 SYSCFG 寄存器，为啥没有时钟了，它看起来好像也在工作呢？

10 System configuration controller (SYSCFG)

The devices feature a set of configuration registers. The main purposes of the system configuration controller are the following:

- Enabling/disabling I²C Fast Mode Plus on some IO ports
- Remapping some DMA trigger sources to different DMA channels
- Remapping the memory located at the beginning of the code area
- Pending interrupt status registers for each interrupt line on STM32F09x devices
- Managing the external interrupt line connection to the GPIOs
- Managing robustness feature

看看画红线的地方就行了，他是用来重映射内存地址和管理外部中断连接到 GPIO 的。在默认的时候，这个寄存器的时钟是关闭的，所以默认都是从 Flash 的起始地址开始运行，外部中断也都连接到 GPIOA 上，只有需要修改的时候我们才会去开启这个寄存器的时钟，并做对应的修改。这也就是解释了，为什么客户有的 GPIO 是好使的（GPIOA 是 OK 的），有些 GPIO 是不好使的（GPIOB 上的就不好使）。

总结：

根本原因是应用代码中 API 调用错误，一般客户在自己建立工程的时候，又不是很了解如何在开始的时候去设置这个地方造成的。需要以后的工程师在参考 AN4065 中注意一下这个地方。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST 产品的最新信息。ST 产品的销售依照订单确认时的相关ST 销售条款。

买方自行负责对ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和ST 徽标是ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。