Security 使用指南

概述

为了降低芯片内部程序被窃取的风险，Spintrol 的产品设计之初就考虑到了程序保护的重要性，不仅为用户提供了非常强大而可靠的程序保护措施，还可以让多个合作方安全地共享芯片内部资源，这些措施包括分区保护、Secure Boot、Debug 锁定以及随机码保护。

本手册适用范围：

适用范围	
SPC1125 系列	SPC1125，SPC1128
SPC1168 系列	SPC1155，SPC1156，SPC1158，SPC1168，SPD1148，SPD1178，SPD1188，SPD1163，SPM1173
SPC2168 系列	SPC2168，SPC2165，SPC2166，SPC1198
SPC1169 系列	SPC1169，SPD1179，SPD1176
SPC2188 系列	SPC1185，SPC2188

SPINTROL

目录

SPINTROL

©2024 旋智电子科技（上海）有限公司

图片列表

SPINTROL

版本历史

版本	日期	作者	状态	变更
C/0	2024-08-06	C.Chai	Released	1. 首次发布。

SPINTROL

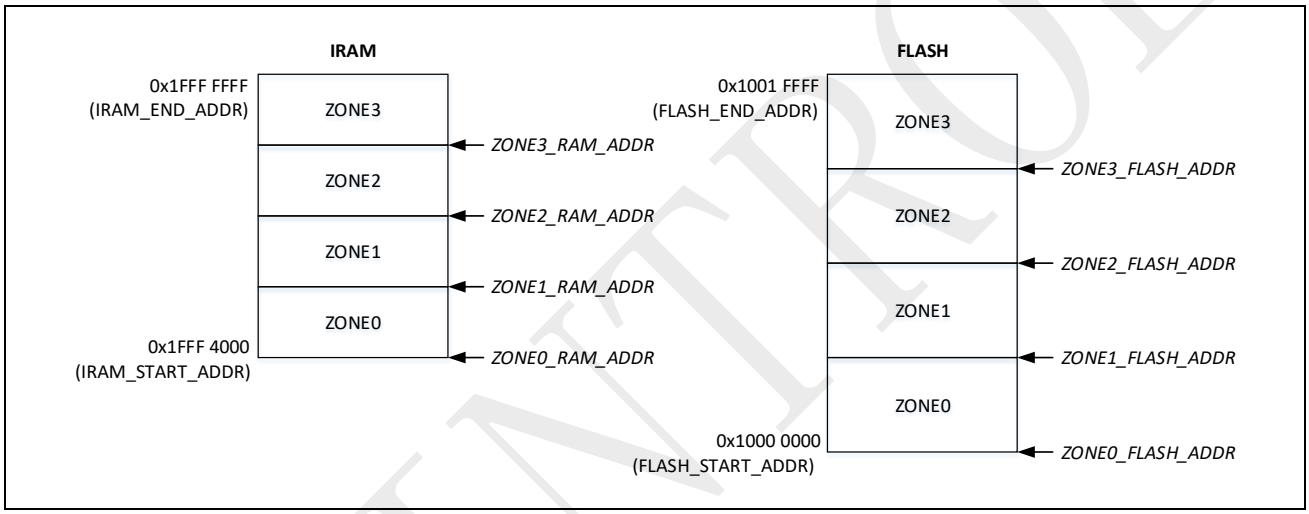
©2024 旋智电子科技（上海）有限公司

1 SPC1168 系列

1.1 分区保护

分区保护（Multi-zone Protect）功能能够保证多个合作方在共享芯片内部资源的同时，不会将程序暴露给任何一个合作方。一旦某个分区的保护被使能，其他分区中的程序就不能够读取或者修改该分区中数据，只能跳转到该分区执行程序。在 SPC1168 系列的产品中，Flash 存储器和 IRAM 最多可以使能 4 个分区的保护功能。用户可以通过设置 Flash 中的 Configuration Words 来使能分区保护。Configuration Words 的具体定义详细请见对应产品的《技术参考手册》。Flash 存储器和 IRAM 的分区示意图如图 1-1 所示。

图 1-1：Flash 存储器和 IRAM 分区示意图



在图 1-1 中，ZONEx_FLASH_ADDR（x = 0，1，2，3）为 Flash 存储器分区 x（FLASH_ZONEx）的起始地址，需要说明的是，ZONE0_FLASH_ADDR 固定为 Flash 存储器的起始地址，即 0x1000 0000；ZONE0_RAM_ADDR 亦固定为 IRAM 的起始地址。

用户可以通过 Configuration Words 中 ZONEx_FLASH_PROT 字段使能分区 FLASH_ZONEx；通过 ZONEx_RAM_PROT 字段使能分区 RAM_ZONEx。在上文中提到，如果某个分区使能后，其他分区中的程序就不能够读取或者修改该分区中的数据。但是有一个例外情形，如果分区 FLASH_ZONEx 和 RAM_ZONEx（x 为同一个值）都被使能，那么这两个分区中的程序是可以相互访问（读/写）对方的。例如，FLASH_ZONE0 和 RAM_ZONE0 都被使能，那么 FLASH_ZONE0 可以访问 RAM_ZONE0 中的内容，同时 RAM_ZONE0 也可以访问 FLASH_ZONE0 中的内容。

Flash 存储器分区 FLASH_ZONEx 的大小由下列因素决定（y = x + 1）：

- FLASH_ZONEx 和 FLASH_ZONEy 都使能了分区保护，则 FLASH_ZONEx 的大小为（ZONEy_FLASH_ADDR - ZONEx_FLASH_ADDR）。例如，FLASH_ZONE0 和 FLASH_ZONE1 都使能了分区保护，则 FLASH_ZONE0 大小为（ZONE1_FLASH_ADDR – ZONE0_FLASH_ADDR）；
- FLASH_ZONEx 使能了分区保护，FLASH_ZONEy 未使能分区保护，则 FLASH_ZONEx 的大小由 FLASH_ZONEz（z > y）的保护状态决定（FLASH_ZONEz 为满足条件 z > y 的任意一个分区）：

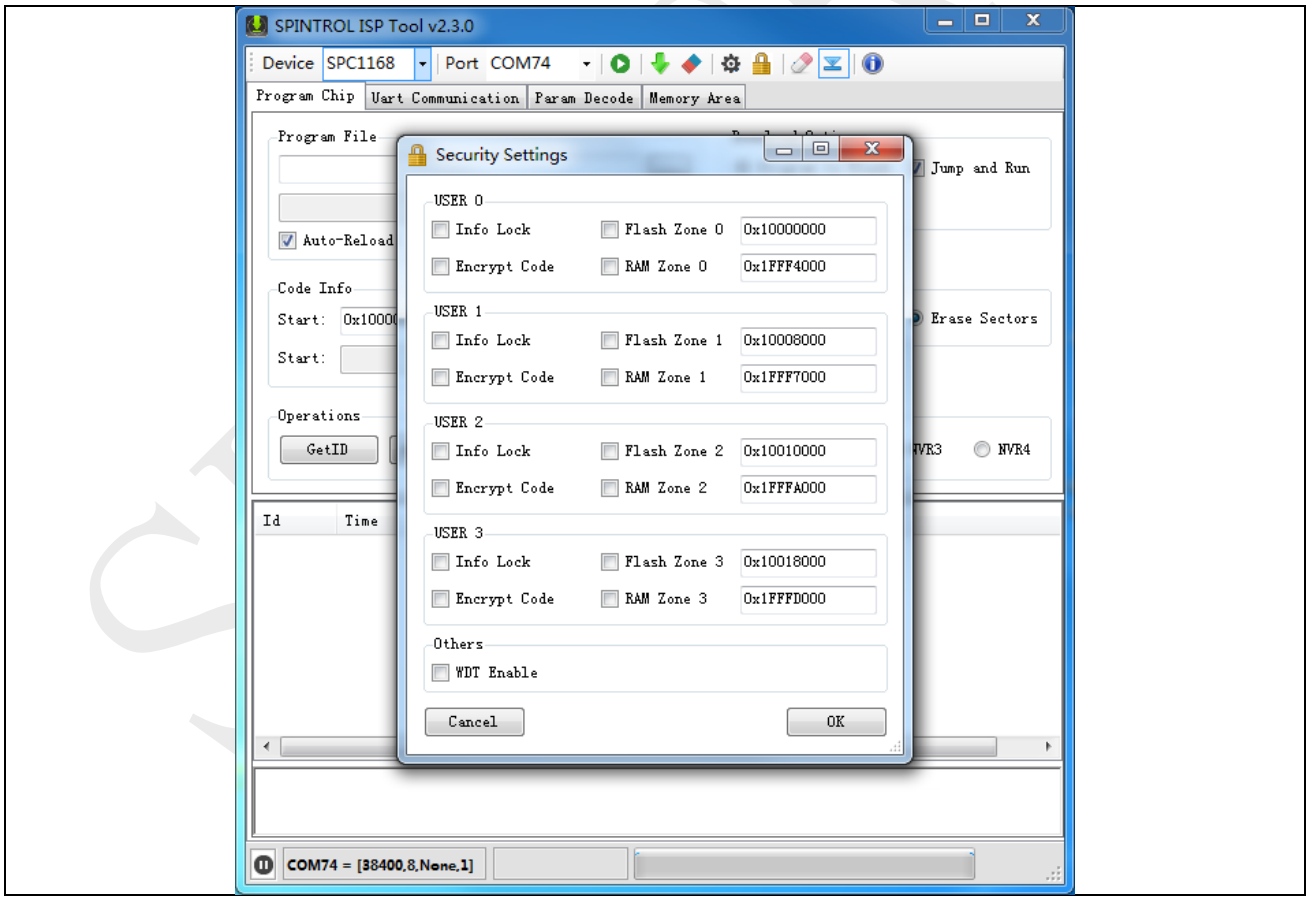
- FLASH_ZONEz 使能了分区保护，则 FLASH_ZONEx 的大小为（ZONEz_FLASH_ADDR - ZONEx_FLASH_ADDR）。此时，ZONEy_FLASH_ADDR = ZONEz_FLASH_ADDR，FLASH_ZONEy 的大小为 0；
- FLASH_ZONEz 未使能分区保护，则 FLASH_ZONEx 的大小为（FLASH_END_ADDR + 1 - ZONEx_FLASH_ADDR）。此时 ZONEy_FLASH_ADDR = ZONEz_FLASH_ADDR = FLASH_END_ADDR，FLASH_ZONEy 和 FLASH_ZONEz 的大小均为 0；

– FLASH_ZONEx 未使能分区保护，FLASH_ZONEy 使能了分区保护，则 FLASH_ZONEx 的大小由 FLASH_ZONEx 是否为 FLASH_ZONE0 决定：

- FLASH_ZONEx 是 FLASH_ZONE0，则 FLASH_ZONEx 的大小为（ZONEy_FLASH_ADDR – ZONE0_FLASH_ADDR）；
- FLASH_ZONEx 不是 FLASH_ZONE0，则 FLASH_ZONEx 的大小为 0。

决定存储器分区 FLASH_ZONEx 的大小的规则，同样适用于 IRAM 的分区 RAM_ZONEx 大小的确定。SPINTROL 提供的 ISP 下载工具可以帮助用户设置 Configuration Words，界面如图 1-2 所示。

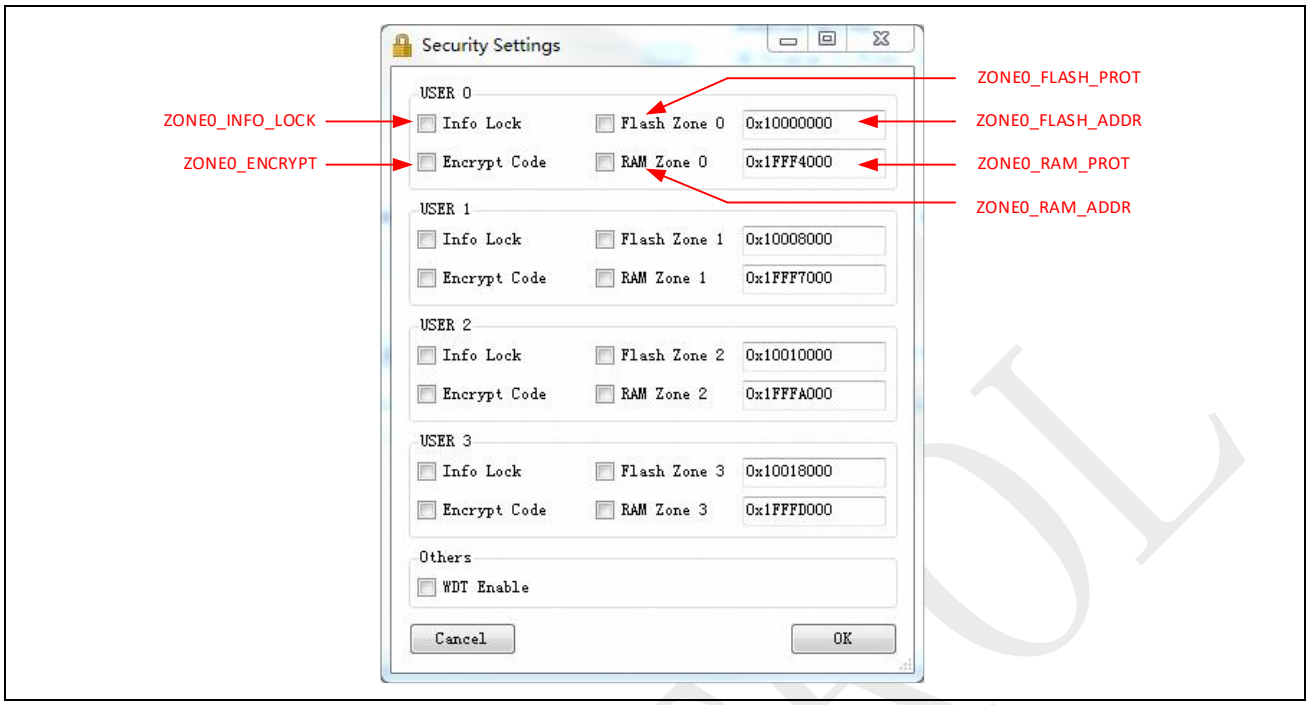
图 1-2：ISP 下载工具 Configuration Words 配置界面



在图 1-3 中，USER x 用于设置分区 x（x = 0，1，2，3）的 Security 功能，图中也标识出了 USER x 各个配置项和 Configuration Words 字段的对应关系（以 USER 0 为例）。

©2024 旋智电子科技（上海）有限公司

图 1-3：ISP 下载工具 Configuration Words 配置项说明



Configuration Words 存放在芯片内部的 Flash 模块中，每个分区相关的字段定义和地址详细请见对应产品的《技术参考手册》。

1.2 Secure Boot

Secure Boot 允许用户的程序以密文的形式写入到芯片内部的 Flash 存储器中。然后，在芯片上电后，bootloader 负责将用户的程序在 Flash 中进行自解密。这样可以避免用户原始程序文件（HEX 文件）在分发过程中泄露程序的风险。Secure Boot 支持 Flash 存储器每个分区的自解密。

在实际使用中，用户在将加密后的程序烧录到芯片 Flash 存储器中后，需要设置 Configuration Words 中 ZONEx_ENCRYPT 字段的值为 0xDEC0DE，表示该分区中的程序被用户加密了。这样，在芯片重新启动后，bootloader 会将用户的程序进行自解密，如果解密成功，则设置 ZONEx_DECRYPT 的值为 0xAA621623；如果解密失败，则设置 ZONEx_DECRYPT 的值为 0x1E55051。

1.3 Debug 锁定

如章节 1.1 所述，当内部的 Flash 存储器或者 IRAM 的任意一个分区保护被使能后，芯片的 Debug 接口就会被锁定。

1.4 随机码保护

随机码保护机制是一种通用型的保护方式，可用于所有产品。

©2024 旋智电子科技（上海）有限公司

有些芯片破解人员也许会具备使用特定设备将芯片打开的能力，将 Flash 中的数据读取出来，然后再将这些数据写到未编程的对应型号的芯片中，实现对用户产品的复制。对于这种情形，Spintrol 的产品具备相应的保护机制。

芯片在出厂时会被写入一个 8 字节的随机码，这个随机码一旦写入后，将不可再被修改。客户的应用程序在运行时可以从芯片中读取并校验该随机数。如果该随机数与应用程序预设的值不一致，就终止应用程序的运行。由于每个芯片的随机码不相同，即使芯片破解人员获得了 Flash 中的程序数据，然后将这些程序数据写到其它对应型号的芯片中，程序也不会正常工作，以此阻止产品被批量复制。

SPINTROL

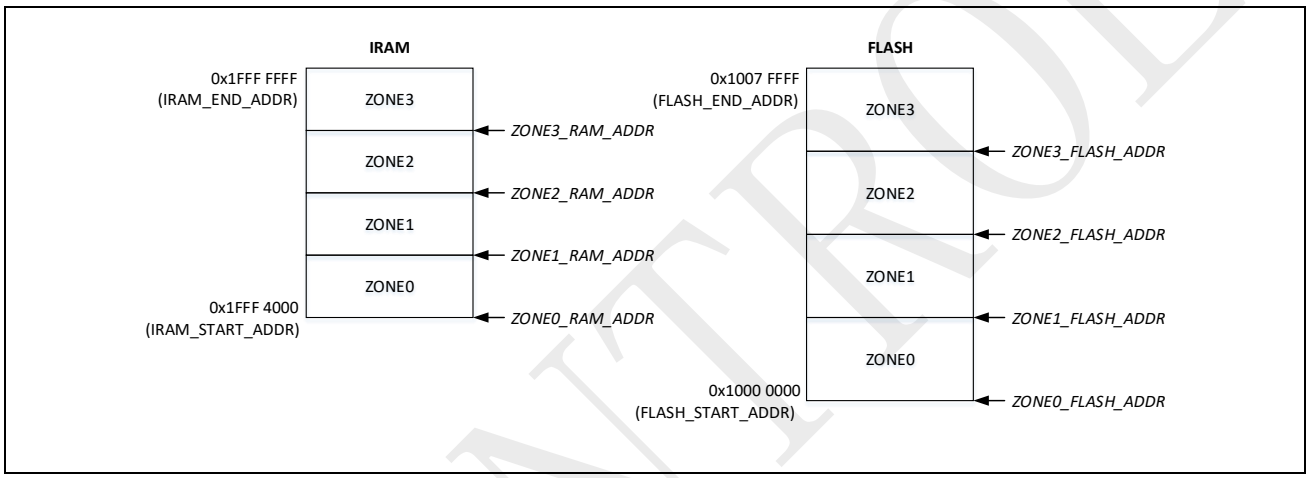
©2024 旋智电子科技（上海）有限公司

2 SPC2168 系列

2.1 分区保护

分区保护（Multi-zone Protect）功能能够保证多个合作方在共享芯片的内部资源的同时，不会将程序暴露给任何一个合作方。一旦某个分区的保护被使能，其他分区中的程序就不能够读取或者修改该分区中数据，只能跳转到该分区执行程序。在 SPC2168 系列的产品中，Flash 存储器和 IRAM 最多可以使能 4 个分区的保护功能。用户可以通过设置 Flash 中的 Configuration Words 来使能分区保护。Configuration Words 的具体定义详细请见对应产品的《技术参考手册》。Flash 存储器和 IRAM 的分区示意图如图 2-1 所示。

图 2-1：Flash 存储器和 IRAM 分区示意图



在图 2-1 中，ZONEx_FLASH_ADDR（x = 0，1，2，3）为 Flash 存储器分区 x（FLASH_ZONEx）的起始地址，需要说明的是，ZONE0_FLASH_ADDR 固定为 Flash 存储器的起始地址，即 0x1000 0000；ZONE0_RAM_ADDR 亦固定为 IRAM 的起始地址。

用户可以通过 Configuration Words 中 ZONEx_FLASH_PROT 字段使能分区 FLASH_ZONEx；通过 ZONEx_RAM_PROT 字段使能分区 RAM_ZONEx。在上文中提到，如果某个分区使能后，其他分区中的程序就不能够读取或者修改该分区中的数据。但是有一个例外情形，如果分区 FLASH_ZONEx 和 RAM_ZONEx（x 为同一个值）都被使能，那么这两个分区中的程序是可以相互访问（读/写）对方的。例如，FLASH_ZONE0 和 RAM_ZONE0 都被使能，那么 FLASH_ZONE0 可以访问 RAM_ZONE0 中的内容，同时 RAM_ZONE0 也可以访问 FLASH_ZONE0 中的内容。

Flash 存储器分区 FLASH_ZONEx 的大小由下列因素决定（y = x + 1）：

- FLASH_ZONEx 和 FLASH_ZONEy 都使能了分区保护，则 FLASH_ZONEx 的大小为（ZONEy_FLASH_ADDR - ZONEx_FLASH_ADDR）。例如，FLASH_ZONE0 和 FLASH_ZONE1 都使能了分区保护，则 FLASH_ZONE0 大小为（ZONE1_FLASH_ADDR – ZONE0_FLASH_ADDR）；
- FLASH_ZONEx 使能了分区保护，FLASH_ZONEy 未使能分区保护，则 FLASH_ZONEx 的大小由 FLASH_ZONEz（z > y）的保护状态决定（FLASH_ZONEz 为满足条件 z > y 的任意一个分区）：
  - FLASH_ZONEz 使能了分区保护，则 FLASH_ZONEx 的大小为（ZONEz_FLASH_ADDR - ZONEx_FLASH_ADDR）。此时，ZONEy_FLASH_ADDR = ZONEz_FLASH_ADDR，FLASH_ZONEy 的大小为 0；

©2024 旋智电子科技（上海）有限公司

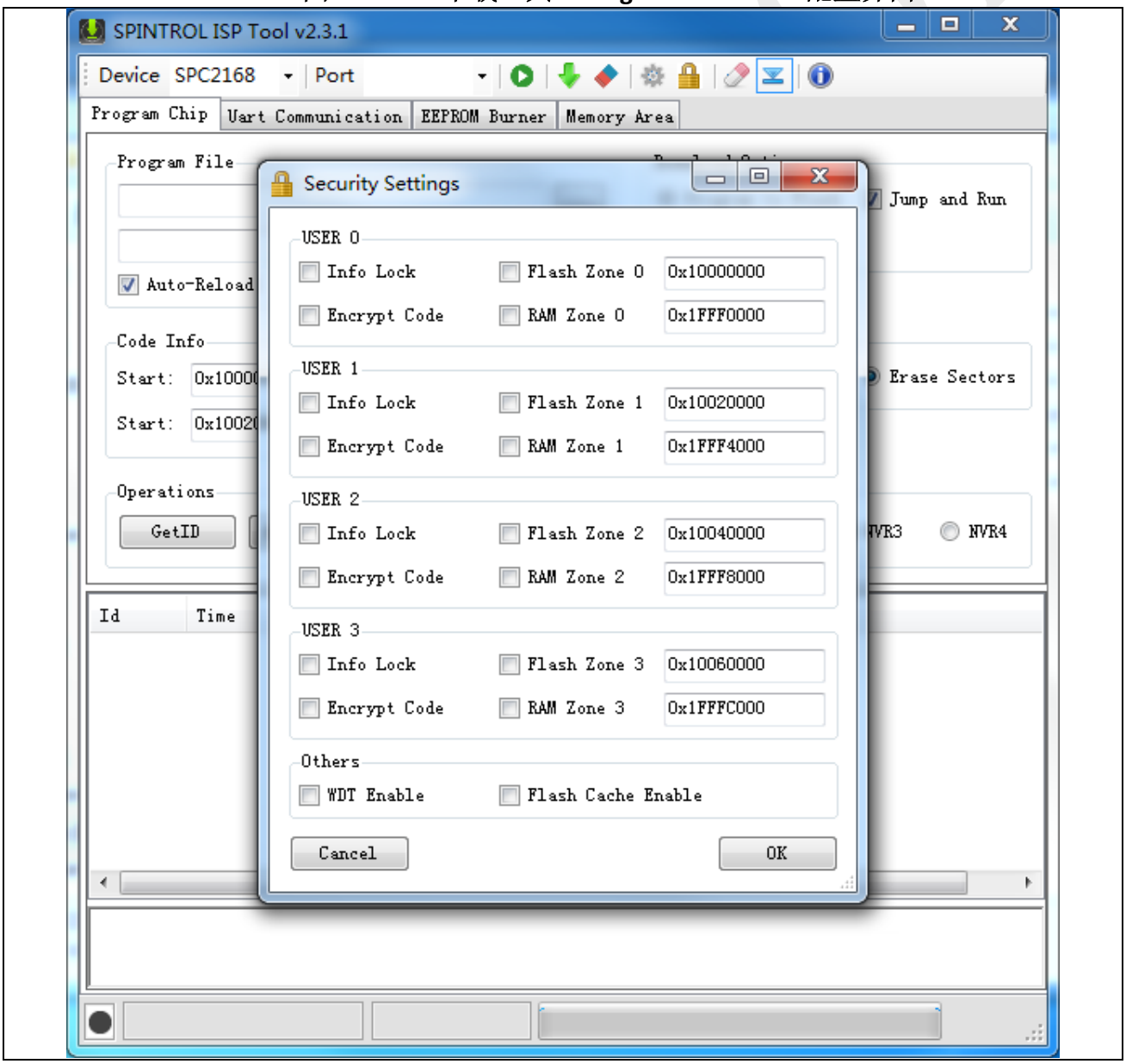
- FLASH_ZONEz 未使能分区保护，则 FLASH_ZONEx 的大小为（FLASH_END_ADDR + 1 - ZONEx_FLASH_ADDR ）。 此 时 ZONEy_FLASH_ADDR = ZONEz_FLASH_ADDR = FLASH_END_ADDR，FLASH_ZONEy 和 FLASH_ZONEz 的大小均为 0；

– FLASH_ZONEx 未使能分区保护，FLASH_ZONEy 使能了分区保护，则 FLASH_ZONEx 的大小由 FLASH_ZONEx 是否为 FLASH_ZONE0 决定：

- FLASH_ZONEx 是 FLASH_ZONE0，则 FLASH_ZONEx 的大小为（ZONEy_FLASH_ADDR – ZONE0_FLASH_ADDR）；
- FLASH_ZONEx 不是 FLASH_ZONE0，则 FLASH_ZONEx 的大小为 0。

决定存储器分区 FLASH_ZONEx 的大小的规则，同样适用于 IRAM 的分区 RAM_ZONEx 大小的确定。SPINTROL 提供的 ISP 下载工具可以帮助用户设置 Configuration Words，界面如图 2-2 所示。

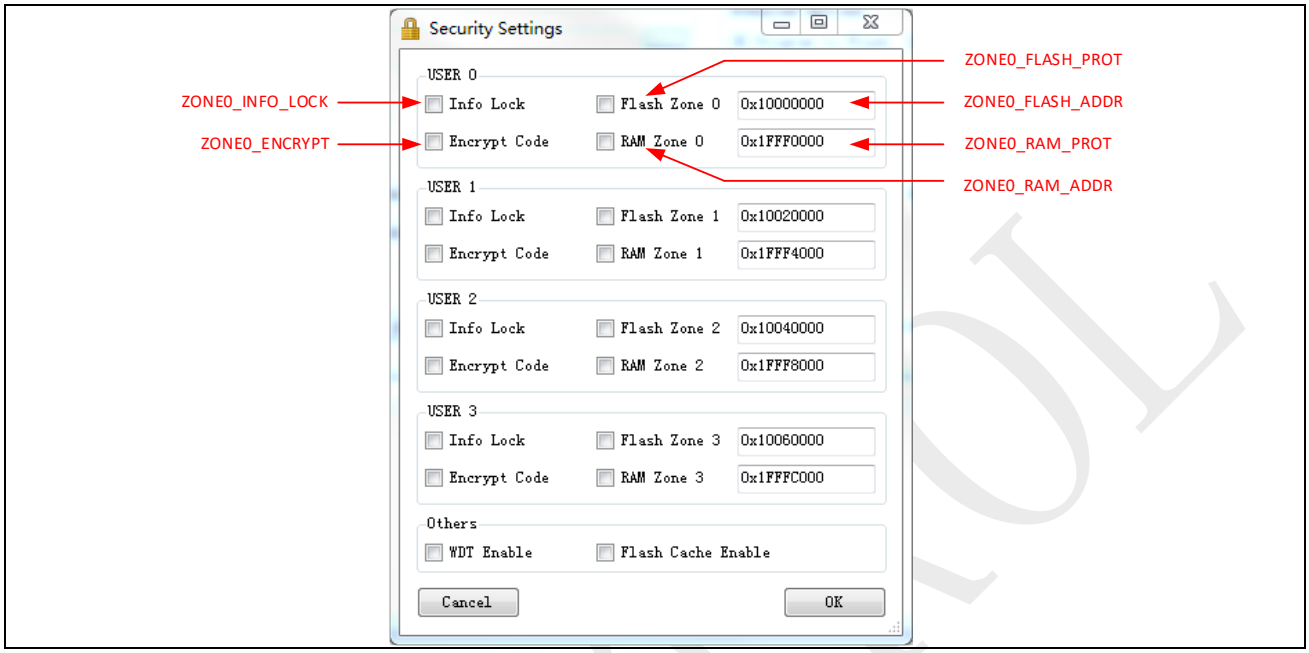
图 2-2：ISP 下载工具 Configuration Words 配置界面



©2024 旋智电子科技（上海）有限公司

在图 2-3 中，USER x 用于设置分区 x（x = 0，1，2，3）的 Security 功能，图中也标识出了 USER x 各个配置项和 Configuration Words 字段的对应关系（以 USER 0 为例）。

图 2-3：ISP 下载工具 Configuration Words 配置项说明



2.2 Secure Boot

Secure Boot 允许用户的程序以密文的形式写入到芯片内部的 Flash 存储器中。然后，在芯片上电后，bootloader 负责将用户的程序在 Flash 中进行自解密。这样可以避免用户原始程序文件（HEX 文件）在分发过程中泄露程序的风险。Secure Boot 支持 Flash 存储器每个分区的自解密。

在实际使用中，用户在将加密后的程序烧录到芯片 Flash 存储器中后，需要设置 Configuration Words 中 ZONEx_ENCRYPT 字段的值为 0xDEC0DE，表示该分区中的程序被用户加密了。这样，在芯片重新启动后，bootloader 会将用户的程序进行自解密，如果解密成功，则设置 ZONEx_DECRYPT 的值为 0xAA621623；如果解密失败，则设置 ZONEx_DECRYPT 的值为 0x1E55051。

2.3 Debug 锁定

如章节 2.1 所述，当内部的 Flash 存储器或者 IRAM 的任意一个分区保护被使能后，芯片的 Debug 接口就会被锁定。

2.4 随机码保护

详细请见章节 1.4。

SPINTROL

3 SPC1169 系列

3.1 Debug 锁定

当产品内部的 Flash 存储器地址 0x1001FFFC 写入非 0xFFFFFFFF 时，芯片的 Debug 接口就会被锁定，详细细节请见芯片的《技术参考手册》。

3.2 随机码保护

详细请见章节 1.4。

SPINTROL

©2024 旋智电子科技（上海）有限公司

4 SPC2188 系列

4.1 Debug 锁定

当产品使能内部的 Flash ECC 功能时，往 Flash 存储器地址 0x1003FFFC 写入非 0xFFFFFFFF 时，芯片的 Debug 接口就会被锁定。

当产品失能内部的 Flash ECC 功能时，往 Flash 存储器地址 0x1007FFFC 写入非 0xFFFFFFFF 时，芯片的 Debug 接口就会被锁定。

4.2 随机码保护

详细请见章节 1.4。

SPINTROL

5 SPC1125 系列

5.1 Debug 锁定

当产品失能内部的 Flash 存储器地址 0x1100060C 以及 0x11000614 均写入非 0xFFFFFFFF 时，芯片的 Debug 接口就会被锁定。

5.2 随机码保护

详细请见章节 1.4。

SPINTROL

©2024 旋智电子科技（上海）有限公司