SPC1068 Flash 使用指南

概述

SPC1068 片内 Flash 通过 QSPI 接口与 APB 总线相连，用于存储用户程序、数据等内容。当芯片上电或者复位后，Boot Loader 负责将 Flash 中的程序装载到片上 SRAM 区域执行。

注意： 本文档主要以 SPC1068 为例进行介绍。

目录

图片列表

表格列表

版本历史

版本	日期	作者	状态	变更
C/0	2024-02-26	周佳莉	Released	首次发布。

术语或缩写

术语或缩写	描述
OTP	One-Time Programmable，仅可编程一次

1 Flash 存储器概述

SPC1068 片内 Flash 通过 QSPI 接口与 APB 总线相连，用于存储用户程序、数据等内容。当芯片上电或者复位后，Boot Loader 负责将 Flash 中的程序装载到片上 SRAM 区域执行。

SPC1068 片内 Flash 有以下特点：

- 容量最多 512KB
- 支持标准 SPI、Dual SPI 以及 Quad SPI 通信方式
- 每个可编程页大小为 256 字节
- 支持大小为 4KB、32KB 以及 64KB 的块擦除
- 支持至少 100000 次擦写
- Flash 内数据至少保存 20 年
- 支持区块保护模式
- 最多 3 个 OTP 存储区（其中一个被 SPINTROL 占用），可供用户使用的容量最多为 512 字节，支持 OTP Lock

2 Flash 存储器映射

SPC1068 片内 Flash 存储器的映射关系如**错误!未找到引用源。 ~ 错误!未找到引用源。**所示。可以看出，Flash 存储器在逻辑上由 Sector（4KB）和 Block（64KB）组成。SPC1068 Flash 存储器包含 8 个 Block，每个 Block 又包含 16 个 Sector。其中，Block 0 是为 Spintrol 保留的区域，Block 2 用于存储用户的程序，其余的 Block 是未使用的区域。用户可以根据自己的需要利用这些未使用的 Block 存储数据。需要强调的是，用户在使用 Flash 存储器时，切勿改动 Block 0 和 Block 2 中的内容。否则，会造成程序运行出错或者芯片工作异常。

SPC1068 片内 Flash 存储器：如果为 Winbond 的 Flash，该 Flash 包含 3 个 OTP Flash 存储块。其中，OTP Flash #1（地址 0x1000~0x10FF）用于存储芯片的标定参数，且芯片出厂后会被 Lock 起来。因此，用户只能使用另外两个 OTP Flash 存储数据；如果为 GigaDevice 的 Flash，该 Flash 包含 2 个 OTP Flash 存储块。其中，OTP Flash #2（地址 0x1000~0x11FF）用于存储芯片的标定参数，且芯片出厂后会被 Lock 起来。因此，用户只能使用另外一个 OTP Flash 存储数据。

SPC1068L 片内 Flash 存储器只有 OTP Flash #1（地址 0x1000~0x11FF），并且被用于存储芯片的标定参数，且芯片出厂后会被 Lock 起来，因此，SPC1068L 没有可供用户使用的 OTP Flash。

图 2-1：SPC1068 Flash（Winbond）存储器映射

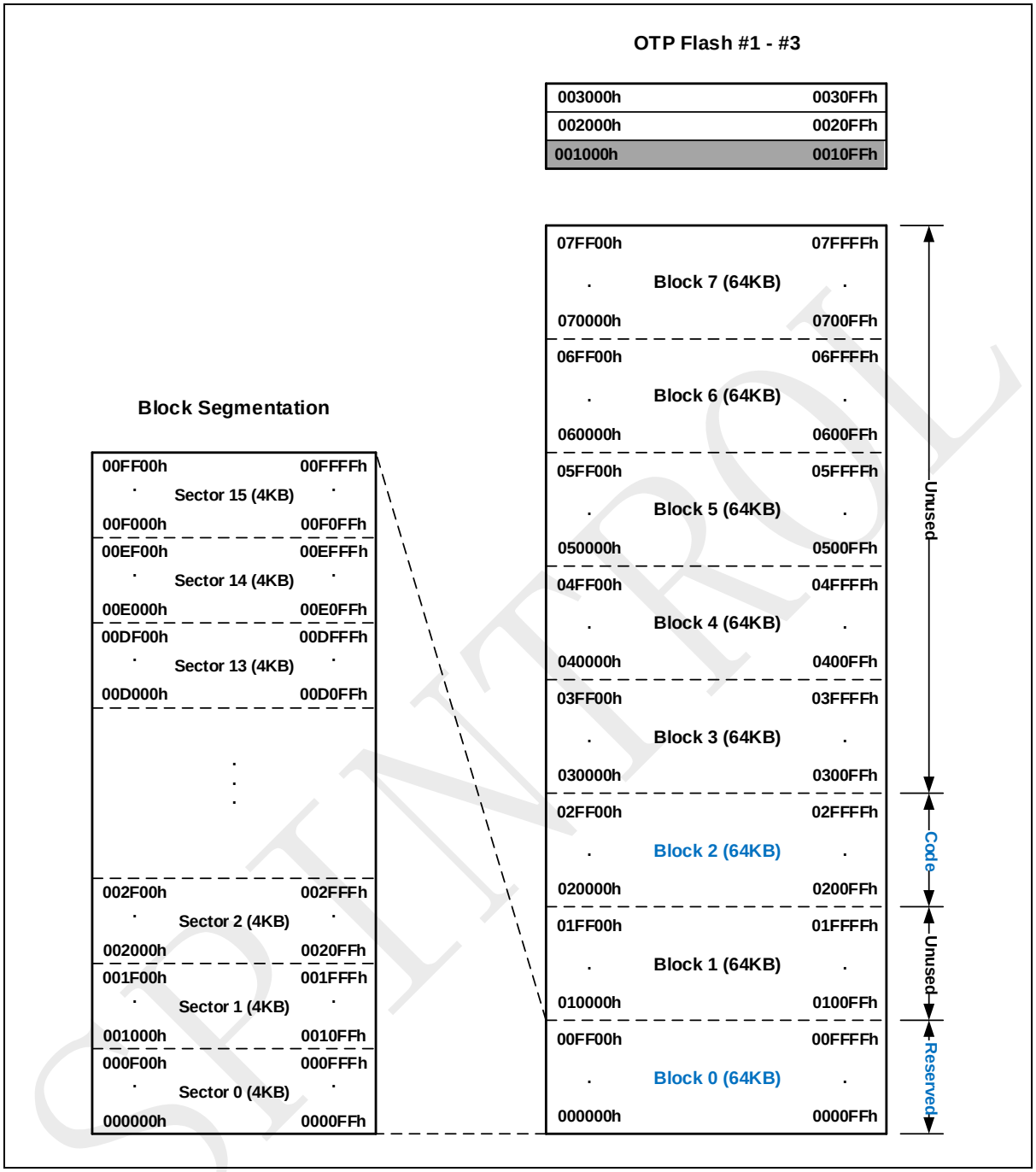


图 2-2：SPC1068 Flash（GigaDevice）存储器映射

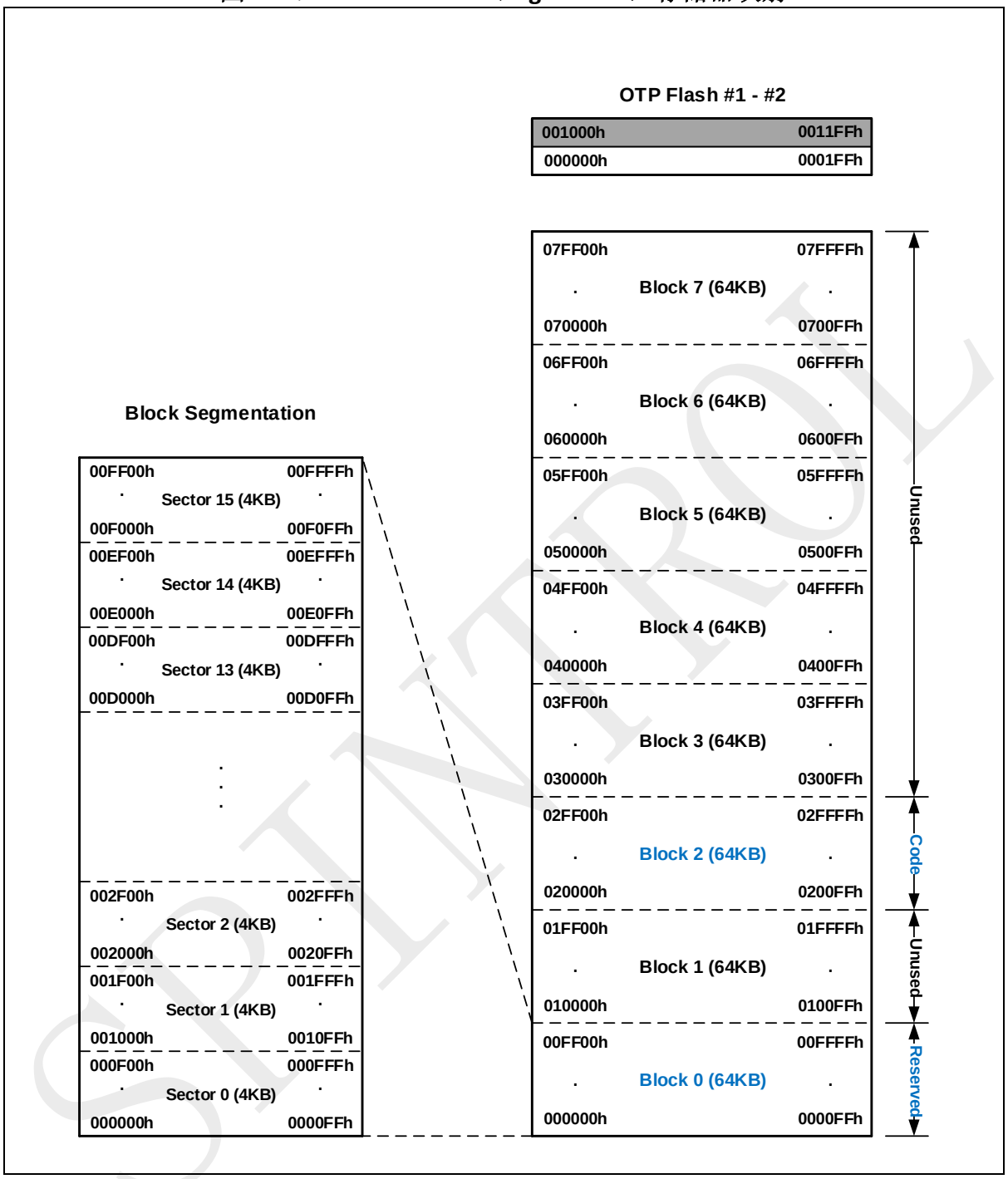
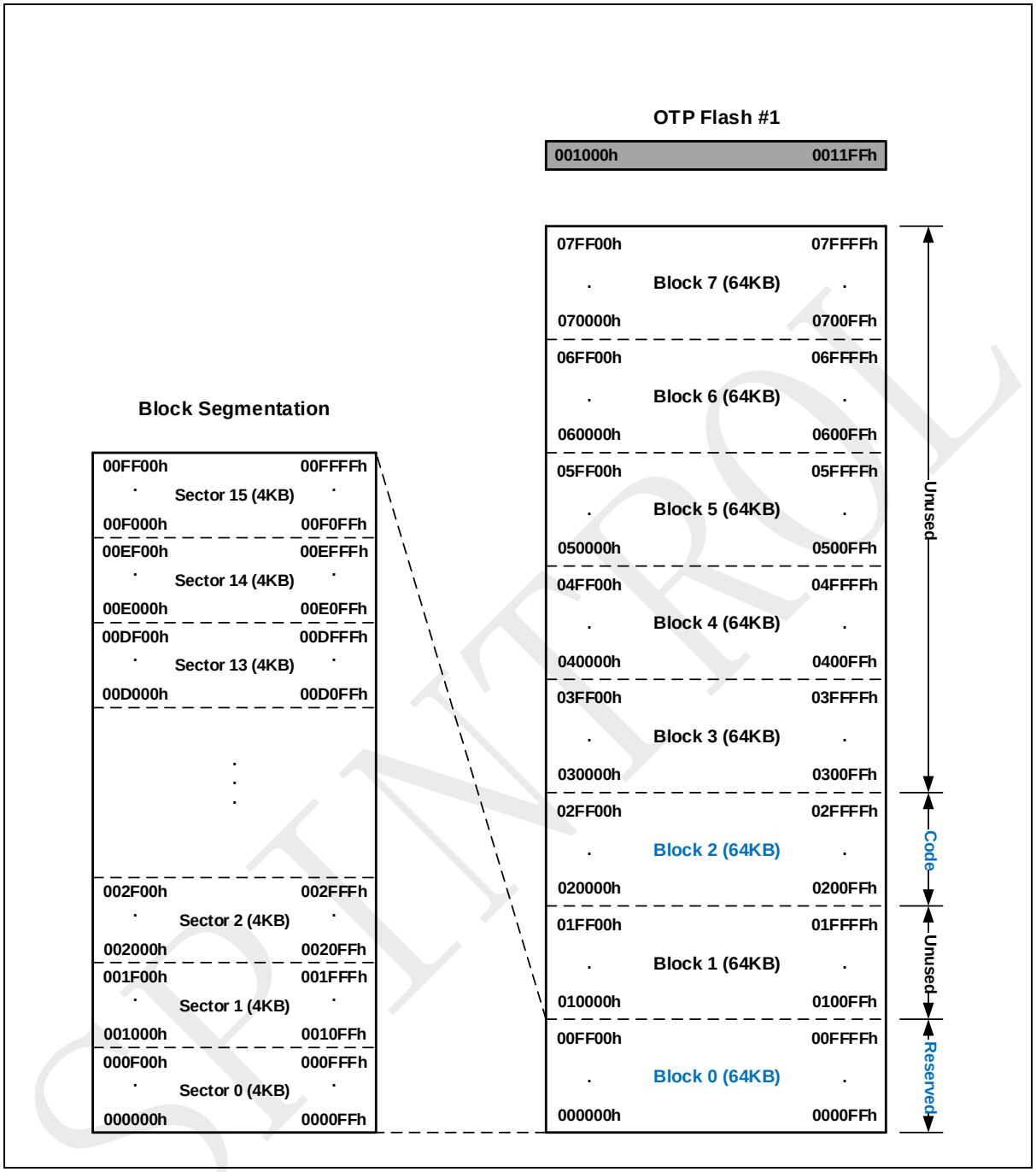


图 2-3：SPC1068L Flash 存储器映射



3 Flash 驱动函数

SPC1068 提供了 Flash 模块驱动函数，如**错误!未找到引用源。**所示。

表 3-1：Flash 驱动函数

函数名称	描述
FlagStatus FLASH_GetBusyStatus（void）	读取 Flash Busy 状态位
ErrorStatus FLASH_SetProtectionMode（ FLASH_ProtectionEnum eProtectMode）	设置 Flash 存储器区块保护
ErrorStatus FLASH_EraseAll（void）	擦除整个 Flash 存储器
ErrorStatus FLASH_SectorErase（uint32_t u32SectorNum）	擦除指定的 Sector
ErrorStatus FLASH_Block32KErase（uint32_t u32BlockNum）	擦除指定的 32KB Block
ErrorStatus FLASH_Block64KErase（uint32_t u32BlockNum）	擦除指定的 64KB Block
ErrorStatus FLASH_Erase（uint32_t u32StartAddr, uint32_t u32EndAddr）	擦除指定地址区间的存储单元
uint32_t FLASH_Read（uint8_t *pu8Buf, uint32_t u32Addr, uint32_t u32NumBytes）	从 Flash 指定地址读取数据
ErrorStatus FLASH_Write（ uint8_t *pu8Buf, uint32_t u32Addr, uint32_t u32NumBytes）	向 Flash 指定地址写入数据
ErrorStatus FLASH_EraseOTP（uint32_t u32Addr）	擦除指定的 OTP Flash 存储区
uint32_t FLASH_ReadOTP（ uint8_t *pu8Buf, uint32_t u32Addr, uint32_t u32NumBytes）	从指定的 OTP Flash 存储区读取数据
ErrorStatus FLASH_WriteOTP（ uint8_t *pu8Buf, uint32_t u32Addr, uint32_t u32NumBytes）	向指定的 OTP Flash 存储区写入数据
ErrorStatus FLASH_LockOTP（uint16_t u16LockBitsVal）	Lock 指定的 OTP Flash 存储区

注意： 用户如果对 Flash 存储器的 Block2 区域设置了区块保护，那么芯片的 Boot Loader 就无法对 Block2 区域进行重新编程了。因此，用户需要在其程序中

设计解除 Block2 区块保护的方法，否则的话，就无法通过 Boot Loader 向芯片重新写入用户程序。

4 Flash 操作实例

4.1 基本操作实例

该例子用于实现 Flash 的 Sector 擦除、数据写入以及数据读取等操作。具体代码如下：

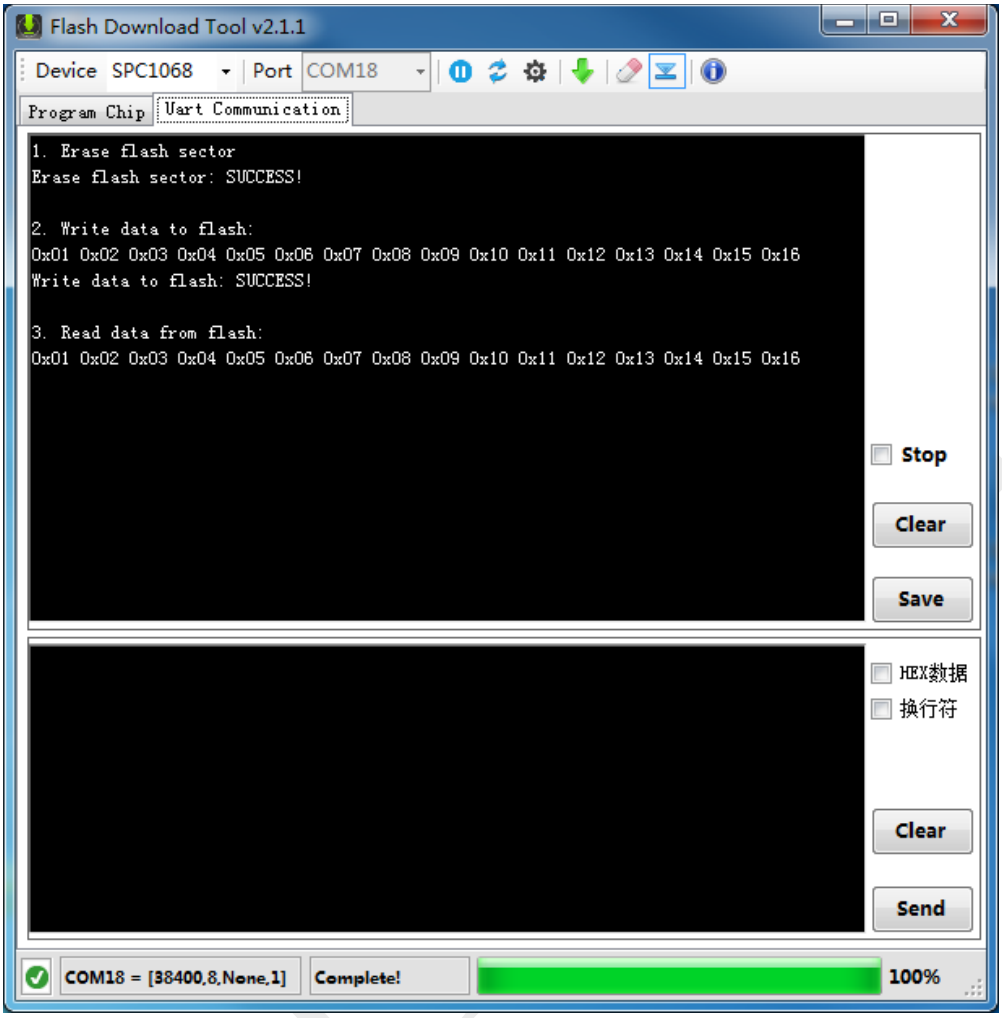
\SPC1068\Project\SDK examples\Flash_operate

```
#include "spc1068.h"
#include <stdio.h>

/* Buffer stores data to be written to flash */
uint8_t au8WriteData[16] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
0x07, 0x08,0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };

/* Buffer stores data read from flash */
uint8_t au8ReadData[16] = {0};


int main()
{
  ErrorStatus status;

  uint32_t i;

  /* System Init */
  Sys_Init();

  /* Clock Init */
  CLOCK_InitWithRCO(CLOCK_HCLK_24MHZ);

  /* Delay Init */
  Delay_Init();


  /* Set GPIO function as UART  */
  GPIO_SetPinChannel(GPIO_34, GPIO34_UART_TXD);
  GPIO_SetPinChannel(GPIO_35, GPIO35_UART_RXD);

  /* Enable UART Clock */
  CLOCK_EnableModule(UART_MODULE);
  /* UART Init */
  UART_Init(UART, 38400);


  /* Erase flash sector */
  printf("1. Erase flash sector\n");
  status = FLASH_SectorErase(FLASH_SECTOR_NUM(0x10000));
  if(status == ERROR)
  {
    return 1;
  }
```

```
  printf("Erase flash sector: SUCCESS!\n\n");


  /* Write data to flash */
  printf("2. Write data to flash:\n");
  for(i = 0; i < 16; i++)
  {
    printf("0x%02X ", au8WriteData[i]);
  }
  printf("\n");

  status = FLASH_Write(au8WriteData, 0x10000, 16);
  if(status == ERROR)
  {
    return 1;
  }
  printf("Write data to flash: SUCCESS!\n\n");


  /* Read data from flash */
  printf("3. Read data from flash:\n");
  FLASH_Read(au8ReadData, 0x10000, 16);
  for(i = 0; i < 16; i++)
  {
    printf("0x%02X ", au8ReadData[i]);
  }

  printf("\n");
}
```

用 ISP 工具将上述程序下载到 SPC1068 芯片中，执行结果如**错误!未找到引用源。**所示。

图 4-1：实例 1 执行结果



4.2 基本操作实例

该例子用于实现 OTP Flash 的擦除、数据写入、数据读取以及 OTP Lock 等操作。具体代码如下：

\SPC1068\Project\SDK examples\Flash_otp

```
#include "spc1068.h"
#include <stdio.h>

/* Buffer stores data to be written to flash security register */
uint8_t au8WriteData[16] = { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
0x07, 0x08, 0x09, 0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16 };

/* Buffer stores data read from flash security register */
uint8_t au8ReadData[16] = {0};


int main()
{
```

```
ErrorStatus status;

uint32_t i;

/* System Init */
Sys_Init();

/* Clock Init */
CLOCK_InitWithRCO(CLOCK_HCLK_24MHZ);

/* Delay Init */
Delay_Init();


/* Set GPIO function as UART  */
GPIO_SetPinChannel(GPIO_34, GPIO34_UART_TXD);
GPIO_SetPinChannel(GPIO_35, GPIO35_UART_RXD);

/* Enable UART Clock */
CLOCK_EnableModule(UART_MODULE);
/* UART Init */
UART_Init(UART, 38400);


/* Erase flash sector */
printf("1. Erase flash security register 2\n");
status = FLASH_EraseOTP(FLASH_OTP2_ADDR);
if(status == ERROR)
{
  return 1;
}
printf("Erase flash security register 2: SUCCESS!\n\n");


/* Read data from flash */
printf("2. Read data from flash security register 2:\n");
FLASH_ReadOTP(au8ReadData, FLASH_OTP2_ADDR, 16);
for(i = 0; i < 16; i++)
{
  printf("0x%02X ", au8ReadData[i]);
}
printf("\n\n");


/* Write data to flash */
printf("3. Write data to flash security register 2:\n");
for(i = 0; i < 16; i++)
{
  printf("0x%02X ", au8WriteData[i]);
}
printf("\n");
```

```
  status = FLASH_WriteOTP(au8WriteData, FLASH_OTP2_ADDR, 16);
  if(status == ERROR)
  {
    return 1;
  }
  printf("Write data to flash security register 2: SUCCESS!\n\n");


  /* Read data from flash */
  printf("4. Read data from flash security register 2:\n");
  FLASH_ReadOTP(au8ReadData, FLASH_OTP2_ADDR, 16);
  for(i = 0; i < 16; i++)
  {
    printf("0x%02X ", au8ReadData[i]);
  }
  printf("\n\n");


  /* Lock flash security register 2 */
/*Be careful:If locked,data in security register 2 can not be
changed */
  printf("5. Lock flash security register 2:\n");
  status = FLASH_LockOTP(FLASH_OTP2_LOCK_BIT_Val);
  if(status == ERROR)
  {
    return 1;
  }
  printf("Lock flash security register 2: SUCCESS!\n");
}
```

用 ISP 工具将上述程序下载到 SPC1068 芯片中，执行结果如**错误!未找到引用源。**所示。

图 4-2：实例 2 执行结果