



SL2X23

Near Field Communication (NFC) Reader

sl2x23_complete_example 用户快速导入手册 V1.0

NFC makes life better! 

May.2023

Beijing Smartlink Microchip Co.Ltd.

版本历史

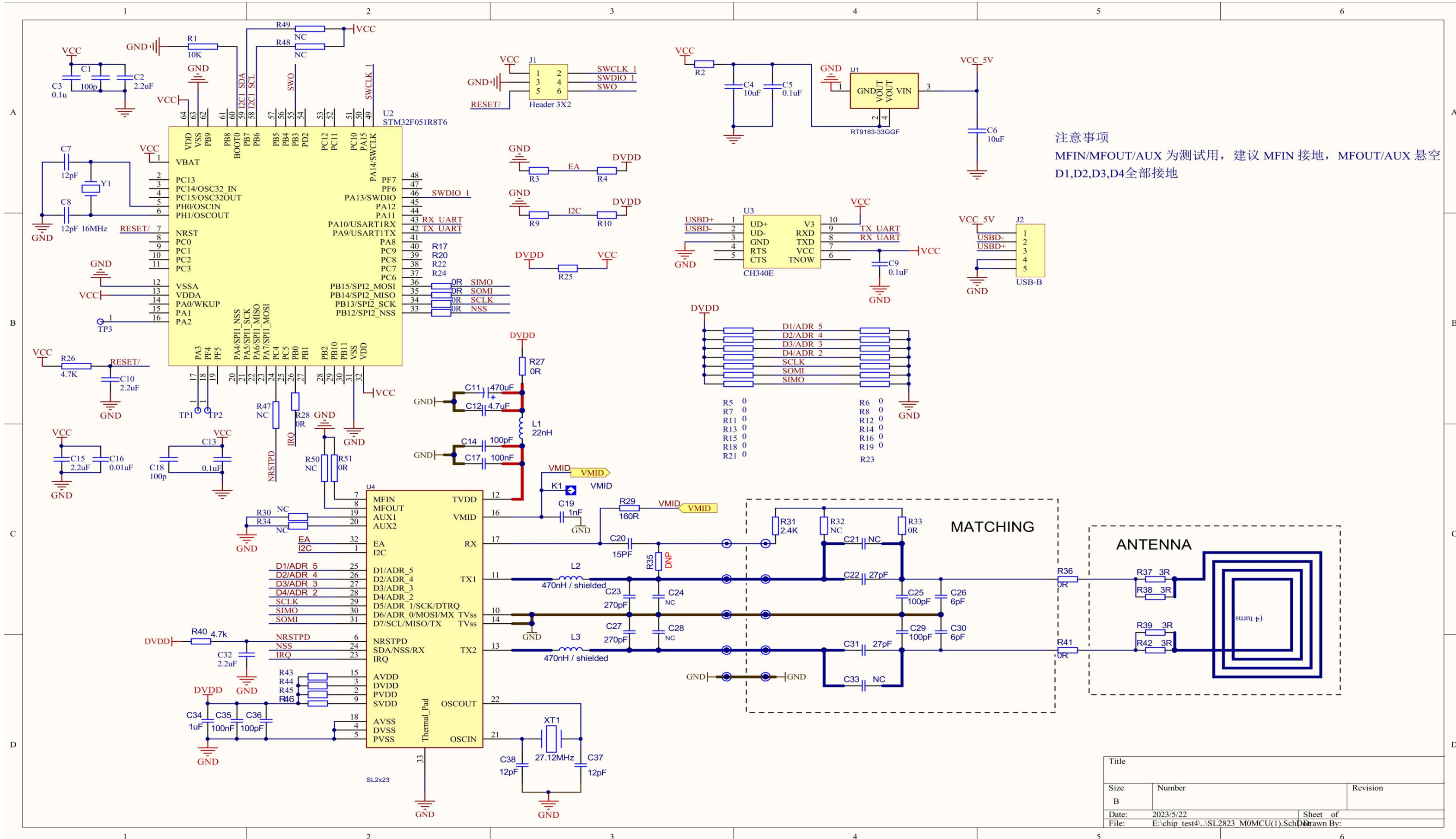
Version	发布时间	发布单位	描述
V1.0	2023.5	北京捷联微芯科技有限公司	初始版本

目录

第一部分 参考设计	4
1.sl2x23 参考原理	4
第二部分 demo 板外观	5
1.sl2x23 demo 板	5
第三部分 sdk 快速移植	6
1.sdk 软件概述	6
2. sdk 快速移植说明	6
3.sdk sl2823 软件目录	6
3.sdk keil 工程目录	6
4.sdk 快速移植步骤（重点）	7
5.sdk 硬件接口函数说明（重点）	7
5.1 irq_flag_io 中断标志	8
5.2 void delay_1ms(uint32_t delay_time) 函数	8
5.3 void write_reg(uint8_t addr, uint8_t RegValue) 函数	8
5.4 uint8_t read_reg(uint8_t addr) 函数	9
6.sdk 演示 demo 调用说明（重点）	9
第四部分 sdk 快速功能验证	10
1. sdk 应用配置（重点）	10
2.学生证 -- 典型应用场景--lpcd 模式	10
3.学生证 -- 典型应用场景--软掉电模式	10
4.学生证 -- 典型应用场景--普通模式	10

第一部分 参考设计

1.sl2x23 参考



第二部分 demo 板外观

1.sl2x23 demo 板

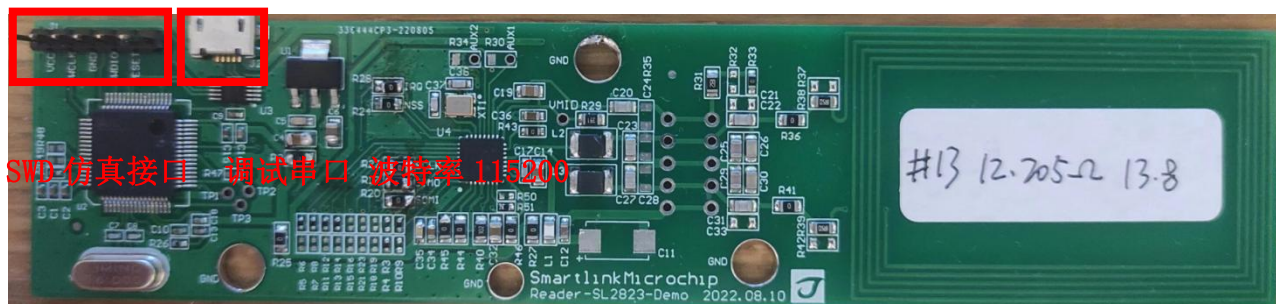


Figure 1. Reader_SL2823_Demo

第三部分 sdk 快速移植

1.sdk 软件概述

本 sdk 是基于 STM32F051R8T6+SL2X23 开发的，SL2X23 与 stm32 之间采用 SPI 进行通信。本 sdk 支持 lpcd、软掉电模式、普通模式三种工作模式。支持 A、B、F、V 全系列卡片读卡。

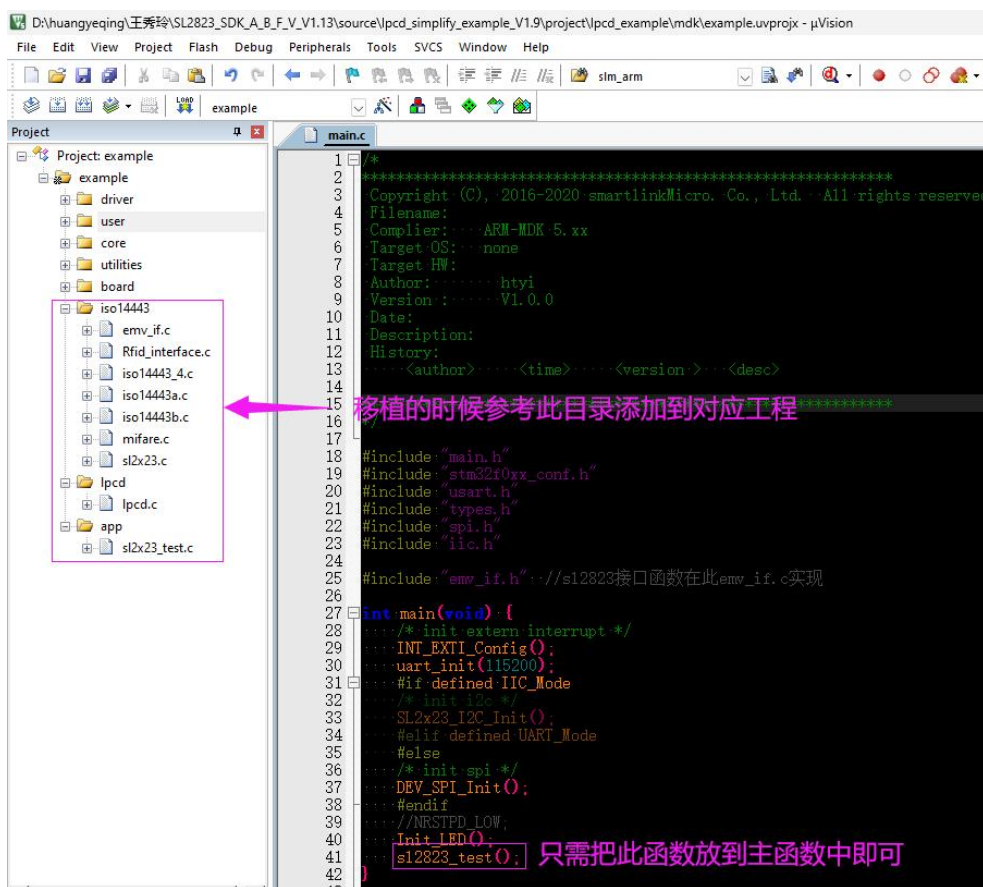
2. sdk 快速移植说明

本 sdk 为了方便客户快速移植，软件上已经完全解耦合。用户只需要适配 emv_if.c 文件里面的 3 个接口文件，Nfc 功能就可以正常运行起来。

3.sdk sl2823 软件目录

名称	修改日期	类型	大小
app	2023/5/21 15:38	文件夹	
iso14443	2023/5/21 15:38	文件夹	
lpcd	2023/5/21 15:38	文件夹	

3.sdk keil 工程目录



4.sdk 快速移植步骤（重点）

第一步：把 sl2823 目录下面的 3 个文件夹源码，拷贝到对应的软件平台代码中。

第二步：参考 keil 软件工程目录添加源代码和头文件。

第三步：硬件接口实现（接口函数在 emv_if.h 文件里面）。

第四步：演示程序调用（把 sl2823_test() 函数放到主函数中即可）

5.sdk 硬件接口函数说明（重点）

sdk 一共只有 3 个硬件接口函数和一个中断变量需要实现：

```

26 volatile uint8_t irq_flag_io;
27 void delay_lms(uint32_t delay_time) {
28     #ifdef USE_STDPERIPH_DRIVER
29         delay_ms(delay_time);
30     #else
31     #endif
32 }
33
34 void write_reg(uint8_t addr, uint8_t RegValue) {
35     #if defined IIC_Mode
36         i2c_write_word(SL2x23_DEV_ADDRESS << 1, addr, RegValue);
37     #elif defined UART_Mode
38         addr = addr & 0x3f; //code the first byte
39         uart1_wrreg(addr, RegValue);
40     #else
41         addr = ((addr & 0x3f) << 1) & 0x7f; //code the first byte
42     #endif USE_STDPERIPH_DRIVER
43     DEV_WriteSR(addr, RegValue);
44     #else
45     //接口函数
46     #endif
47     #endif
48 }
49
50 uint8_t read_reg(uint8_t addr) {
51     #if defined IIC_Mode
52         uint8_t reg_value;
53         i2c_read_word(SL2x23_DEV_ADDRESS << 1, addr, &reg_value);
54         return reg_value;
55     #elif defined UART_Mode
56         addr = (addr & 0x3f) | 0x80; //code the first byte
57         uart1_rdreg(addr);
58     #else
59         addr = ((addr & 0x3f) << 1) | 0x80; //code the first byte
60     #endif USE_STDPERIPH_DRIVER
61     return DEV_ReadSR(addr);
62     #endif
63     //接口函数
64     #endif
65 }

```

5.1 irq_flag_io 中断标志

只需要把此中断标志放到 irq 中断函数中，并进行置位即可。

```

363 void EXTI0_1_IRQHandler(void)
364 {
365     /* wakeup interrupt */
366     if (EXTI_GetITStatus(DEV_EXTI_Line_1) == SET)
367     {
368         /* user handle sudo */
369         Irq1_Delay_Cnt = 6; // 4*2 = 8ms
370         EXTI_ClearITPendingBit(DEV_EXTI_Line_1); // clean irq flag
371     }
372     /* data interrupt */
373     if (EXTI_GetITStatus(DEV_EXTI_Line_2) == SET)
374     {
375         irq_flag_io = 1;
376         EXTI_ClearITPendingBit(DEV_EXTI_Line_2); // clean irq flag
377     }
378 }
  
```

5.2 void delay_lms(uint32_t delay_time) 函数

此函数非常重要，单位是 lms。请把对应平台的延时函数放到此处即可。

```

27 void delay_lms(uint32_t delay_time) {
28     #ifdef USE_STDPERIPH_DRIVER
29         delay_ms(delay_time);
30     #else
31     #endif
32 }
  
```

5.3 void write_reg(uint8_t addr, uint8_t RegValue) 函数

此函数非常重要，写寄存器函数。请参考 demo 例程，自行实现。

```

34 void write_reg(uint8_t addr, uint8_t RegValue) {
35     #if defined IIC_Mode
36         i2c_write_word(SL2x23_DEV_ADDRESS << 1, addr, RegValue);
37     #elif defined UART_Mode
38         addr = addr & 0x3f; // code the first byte
39         uart1_wrreg(addr, RegValue);
40     #else
41         addr = ((addr & 0x3f) << 1) & 0x7F; // code the first byte
42     #endif
43     #ifdef USE_STDPERIPH_DRIVER
44         DEV_WriteSR(addr, RegValue);
45     #else
46         // 接口函数
47     #endif
48 }
  
```


5.4 uint8_t read_reg(uint8_t addr) 函数

此函数非常重要，读寄存器函数。请参考 demo 例程，自行实现。

```
50 uint8_t read_reg(uint8_t addr) {
51     #if defined IIC_Mode
52     uint8_t reg_value;
53     i2c_read_word(SL2x23_DEV_ADDRESS << 1, addr, &reg_value);
54     return reg_value;
55     #elif defined UART_Mode
56     addr = (addr & 0x3f) | 0x80; //code the first byte
57     uart1_rddreg(addr);
58     #else
59     addr = ((addr & 0x3f) << 1) | 0x80; //code the first byte
60     #ifdef USE_STDPERIPH_DRIVER
61     return DEV_ReadSR(addr);
62     #endif
63     //接口函数
64     #endif
65 }
```

注释：到此所有的硬件接口已经全部移植完毕

6.sdk 演示 demo 调用说明（重点）

如果上面的接口函数已经全部完成，只需 spi_wr_test() 此函数测试通过，说明移植工作已经完成。串口会看到“spi test pass!”打印，串口波特率 115200。

```
smartlinkMicro Co., Ltd. V1.0
spi test pass!
```

spi_wr_test() 测试通过后，只需要把 sl2823_test() 函数，放到 main() 函数中，就可以开始 nfc 功能调试。

```
27 int main(void) {
28     /* init extern interrupt */
29     INT_EXTI_Config();
30     uart_init(115200);
31     #if defined IIC_Mode
32     /* init i2c */
33     SL2x23_I2C_Init();
34     #elif defined UART_Mode
35     #else
36     /* init spi */
37     DEV_SPI_Init();
38     #endif
39     //NRSTPD_LOW;
40     Init_LED();
41     sl2823_test();
42 }
```

第四部分 sdk 快速功能验证

1. sdk 应用配置（重点）

Nfc 所有的应用配置，在 sl2x23.h 头文件里面，只需要打开相对应的宏就可以开启对应的功能。

```

17  // #define NOT_IRQ
18  #define IRQ_MODE 0 // 0下降沿触发 1上升沿触发
19  // #define IIC_Mode
20
21
22  #define LPCD_MODE // Lpcd功能
23  // #define SPD_MODE // 软掉电工程 SPD_MODE和LPCD_MODE不能同时开启
24
25  // #define NFC_DEBUG 1
26  // #define LPCD_DEBUG
27  // #define AGC_CALIB_EN // 每次读卡都做AGC, 不要打开
28

```

2. 典型应用场景--lpcd 模式

```

#define IRQ_MODE 0 //下降沿触发
#define LPCD_MODE //Lpcd 功能

```

3. 典型应用场景--软掉电模式

```

#define IRQ_MODE 0 //下降沿触发
#define SPD_MODE //软掉电工程

```

4. 典型应用场景--普通模式

```

#define IRQ_MODE 0 //下降沿触发

```

注释：其它调试宏，调试的时候打开。正式使用的时候不用打开。