SMEC98SP

加密芯片开发手册

版本 2.1

2016/9

目录

第 1 章 SMEC98SP 简介

1.1 概述

SMEC98SP 采用增强型 8051 智能卡内核，用户可以把 MCU 中程序一部分关键功能、算法代码下载到 SMEC98SP 中运行。用户采用标准 C 语言编写程序代码，采用 KEIL C 编译器，编译并下载到加密芯片中。在实际运行过程中，通过 I2C 通信，获取加密芯片中运行结果，并以此结果，作为 MCU 程序运行的输入数据。因此 SMEC98SP 成了产品的一部分，而部分关键功能或算法在 SMEC98SP 内部运行，盗版商无法破解，从根本上杜绝了程序被破解的可能。

MCU 程序，分为两部分：一部分是在 MCU 中，另一部分在 SMEC98SP 中，当需要用到 SMEC98SP 中的功能或算法时，MCU 向 SMEC98SP 发送指令，SMEC98SP 根据指令，在内部运行，返回结果给 MCU。

加密原理核心：主控MCU的关键功能或算法代码func21、func22等放入SMEC98SP中运行

图1-1:加密芯片原理示意图

1.2 管脚定义

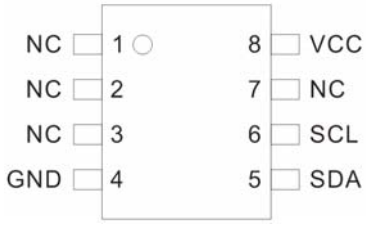


图 1-2：管脚定义

Pin NO.	Obj	Description	Pin NO.	Obj	Description
1	NC	悬空（不接 VCC 和 GND）	8	VCC	VCC
2	NC	悬空（不接 VCC 和 GND）	7	NC	悬空（不接 VCC 和 GND）
3	NC	悬空（不接 VCC 和 GND）	6	SCL	I2C clock
4	GND	地	5	SDA	I2C data

1.3 外形尺寸

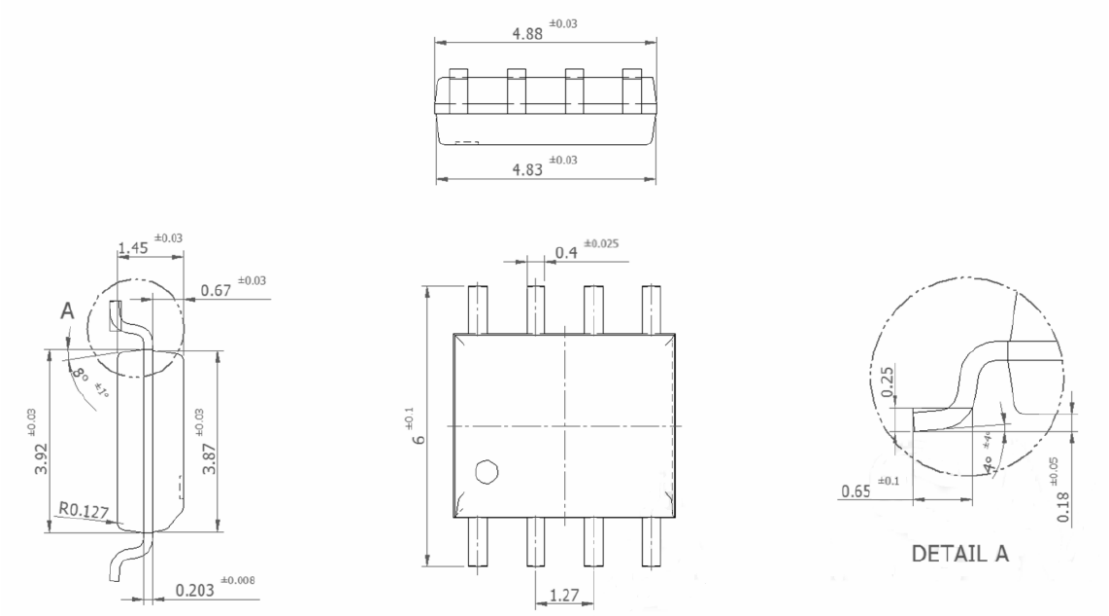


图 1-3：外形尺寸

1.4 产品特性

- 以最高安全等级的智能卡芯片内核为基础，具有极高的软硬件安全性
- 实现客户关键功能或算法代码下载，用户可以灵活实现自有知识产权的保护
- 标准 SOP8 封装形式，器件封装形式小
- 标准 I2C 接口，具有接口线少，控制方式简单，通信速率高等优点

1.4.1 硬件特性

- 采用增强 8051 内核
- 内部 CPU 时钟 30MHz，指令周期 4T
- 硬件 ID 唯一
- 硬件 I2C 接口，最高支持 3.4Mbit/s
- 32 位真随机数发生器
- DES/3DES 处理器

- CRC16 硬件处理器
- 具有 24K 字节用户程序下载空间(可为用户定制容量)
- 8K 字节数据存储区，256 字节/页，支持页擦字节编程
- 256 字节 data/idata RAM
- 1792 字节 xdata RAM
- 工作电压 1.62V ~ 5.5V
- 工作温度 -25 ℃ 至 +85 ℃
- 工作电流：典型值 2.5mA，最大值 5.0mA
- 上电启动最大时间 Tmax = 10ms
- 无需外部时钟

1.4.2 软件特性

- 支持用户程序下载，关键功能或算法加密芯片内部实现机制
- 支持密码比对方式的身份识别机制
- 支持密钥不出卡，外部认证、内部认证等身份识别机制
- 支持自定义各种安全机制
- 加密方案用户自己定义，每一个客户都可以自定义自己的加密方案，破解者无从了解客户采用的具体加密方案

1.4.3 安全特性

- 最高等级的智能卡芯片为基础，具有处理能力强，安全性高特点。与银行卡、二代身份证同一安全等级
- 具有金属防护层，探测到外部攻击后，内部数据自毁
- 总线和内存加密
- 芯片防篡改设计，序列号唯一
- 硬件错误检测
- 随机数发生器
- 噪音的产生（对边信道攻击）

1.5 应用领域

机顶盒、游戏机、墨盒、控制器、安防监控、汽车电子、平板电脑、路由器、DVR、交换机、仪器仪表等各种电子产品终端。

第 2 章 SMEC98SP 通信说明

SMEC98SP 加密芯片采用标准 I2C 接口，最高支持 3.4Mbit/s 速率。

2.1 I2C 位传输

SCL 为高电平时，SDA 有效。SCL 为低电平时，SDA 才允许变化。如图 2-1。

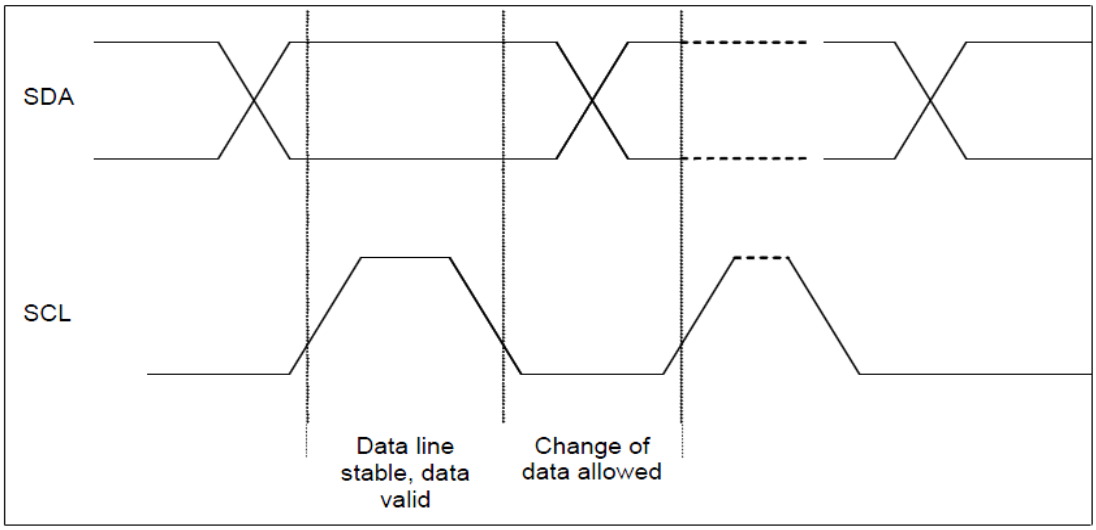


图 2-1：I2C 位传输

2.2 I2C 起始、停止位

起始信号：当 SCL 为高期间，SDA 由高到低的跳变；启动信号是一种电平跳变时序信号，而不是一个电平信号。

停止信号：当 SCL 为高期间，SDA 由低到高的跳变；停止信号也是一种电平跳变时序信号，而不是一个电平信号。如图 2-2。

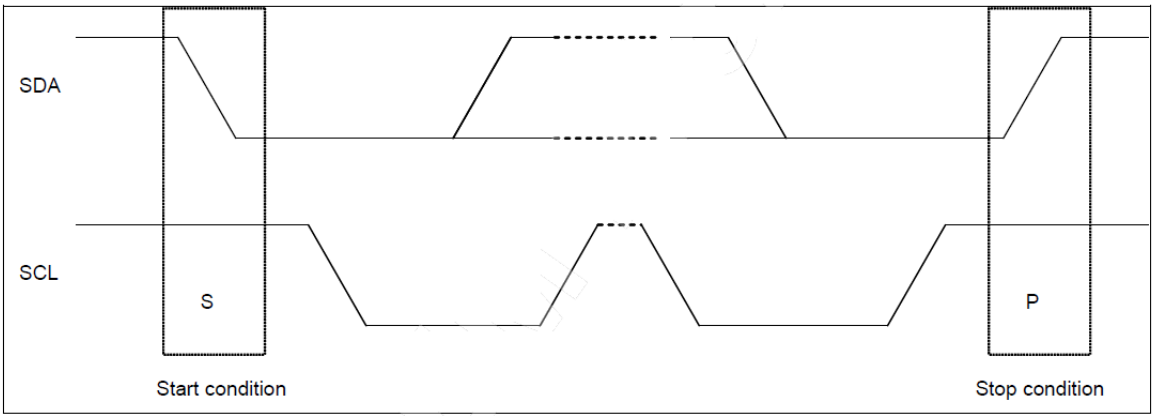


图 2-2：I2C 起始、停止位

2.3 I2C 响应

发送方每发送一个字节，就在时钟脉冲 9 期间释放数据线，由接收方反馈一个应答信号。应答信号为低电平时，规定为有效应答位（ACK 简称应答位），表示接收方已经成功地接收了该字节；应答信号为高电平时，规定为非应答位（NAK），一般表示接收器方收该字节没有成功。

应答位：SCL 高电平时，SDA 低电平，ACK 应答。

SCL 高电平时，SDA 低电平，NAK 应答。

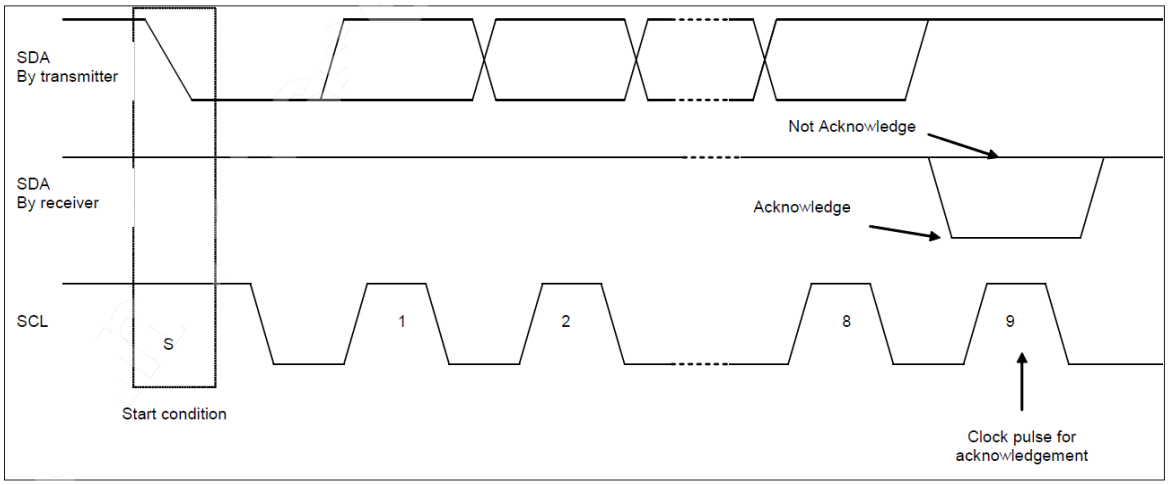


图 2-3：I2C 响应

2.4 I2C 写

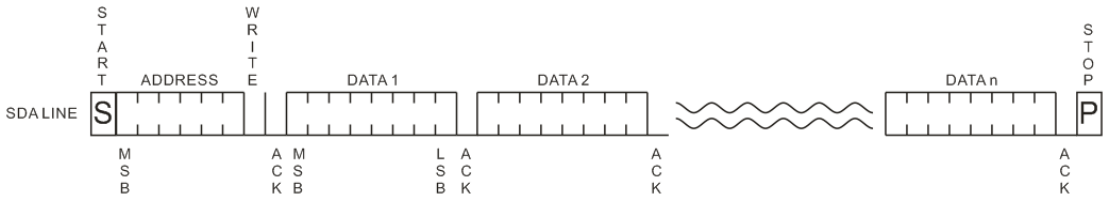


图 2-4：I2C 写

2.5 I2C 读

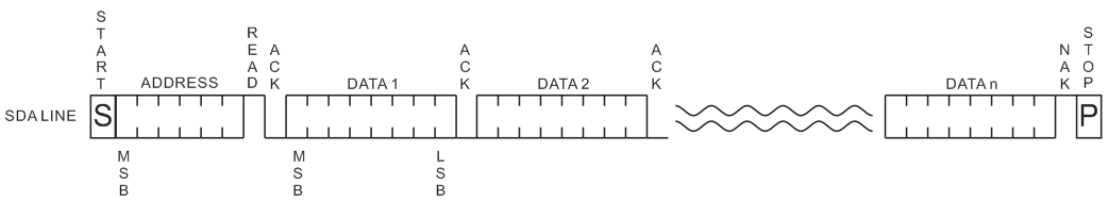


图 2-5：I2C 读

第 3 章 SMEC98SP 开发板

SMEC98SP 开发板提供了联机下载、脱机下载、联机调试等功能。

1. 开发板使用 USB 供电，或者 5V 电源直接供电。
2. 联机操作时，与 PC 接口为 HID 接口，无需装载驱动。可直接与 PC 端开发工具相连。

3.1 开发板

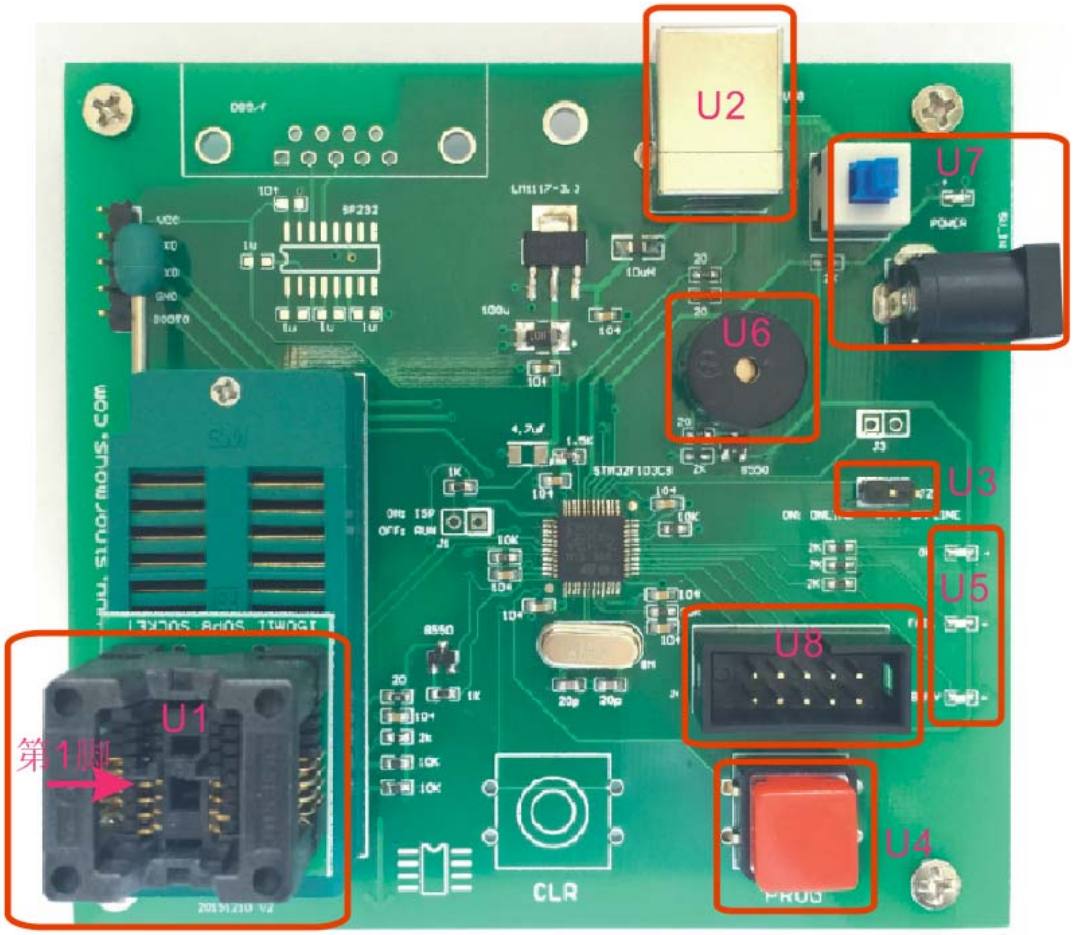


图 3-1：开发板

U1：SOP8 芯片座
U2：USB 接口
U3：联机或脱机跳线：连接 ON－－联机下载；断开 OFF－－脱机下载
U4：编程按钮
U5：指示灯
U6：蜂鸣器
U7：脱机电源及开关
U8：自动烧录机控制线

3.2 自动烧录机接口

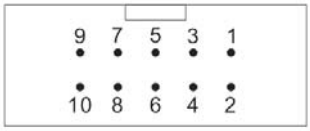
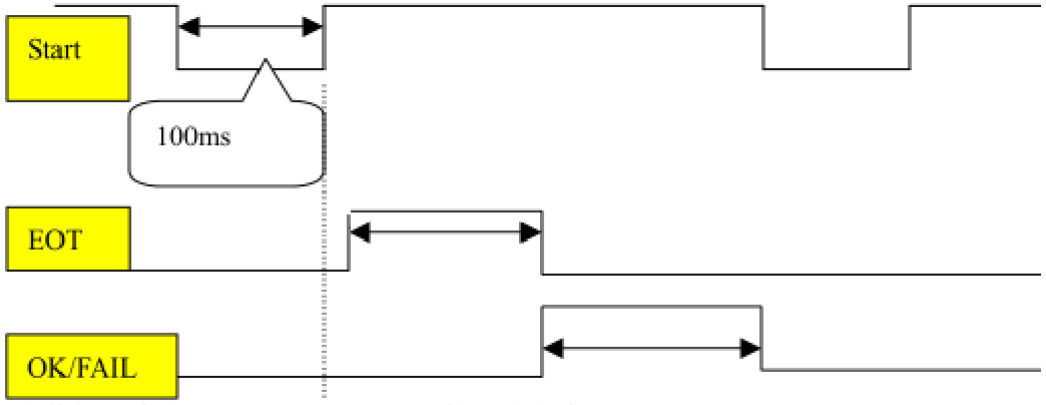


图 3-2：自动烧录机控制线接口

Pin	Obj	Description	Pin	Obj	Description
1	EOT	忙信号(表示正在烧录的过程)，是输出信号	6	NC	悬空
2	NC	悬空	7	START	输入信号，是由自动机给烧录器的开始烧录信号
3	FAIL	烧录失败信号，是输出信号	8	VCC	烧录器对外提供 5V 电源
4	NC	悬空	9	GND	接地
5	OK	烧录成功信号，是输出信号	10	NC	悬空



注：光藕的限流电阻 470Ω，8mA 以上的驱动电流需要

图 3-3：自动烧录机控制时序图

信号描述：

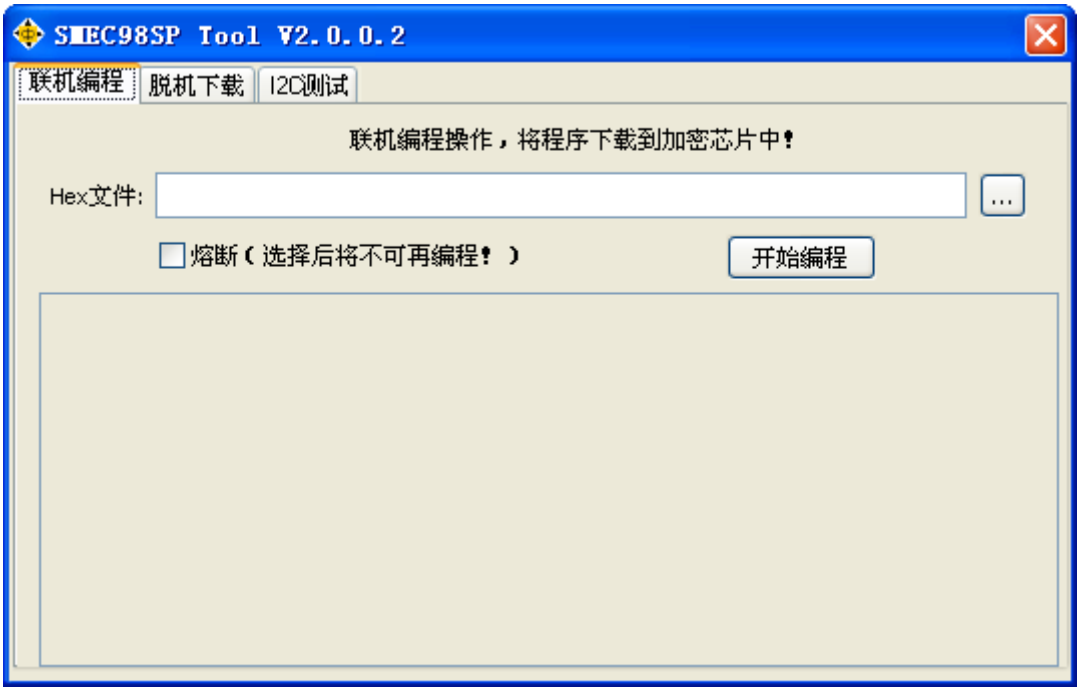
- START：开始烧录信号，常态为高，由高变到低电平，保持至少 100 毫秒，再升到高电平，这时烧录器开始烧录
- EOT：忙信号，常态为低电平，烧录器开始烧录时，将其拉到高电平，烧录完成时再降到低电平
- OK ：成功信号,烧录完成并烧录成功时，EOT 信号降到低电平，同时将 OK 信号拉高，并保持高电平，直到检测到下一个开始烧录信号时回到低电平
- FAIL：失败信号,烧录完成并烧录失败时，EOT 信号降到低电平，同时将 FAIL 信号拉高，并保持高电平，直到检测到下一个开始烧录信号时回到低电平

第 4 章 PC 端开发工具

PC 端开发工具，配合 SMEC98SP 开发板，可实现加密芯片的联机下载，脱机下载，联机调试等功能。

4.1 联机下载

联机下载，将 Keil C 开发的 hex 文件，下载到加密芯片中，用于前期开发调试用。



熔断选项说明：不选中“熔断”时，加密芯片可以被反复编程，
“熔断”选中时，加密芯片将被锁死，不允许再次编程。

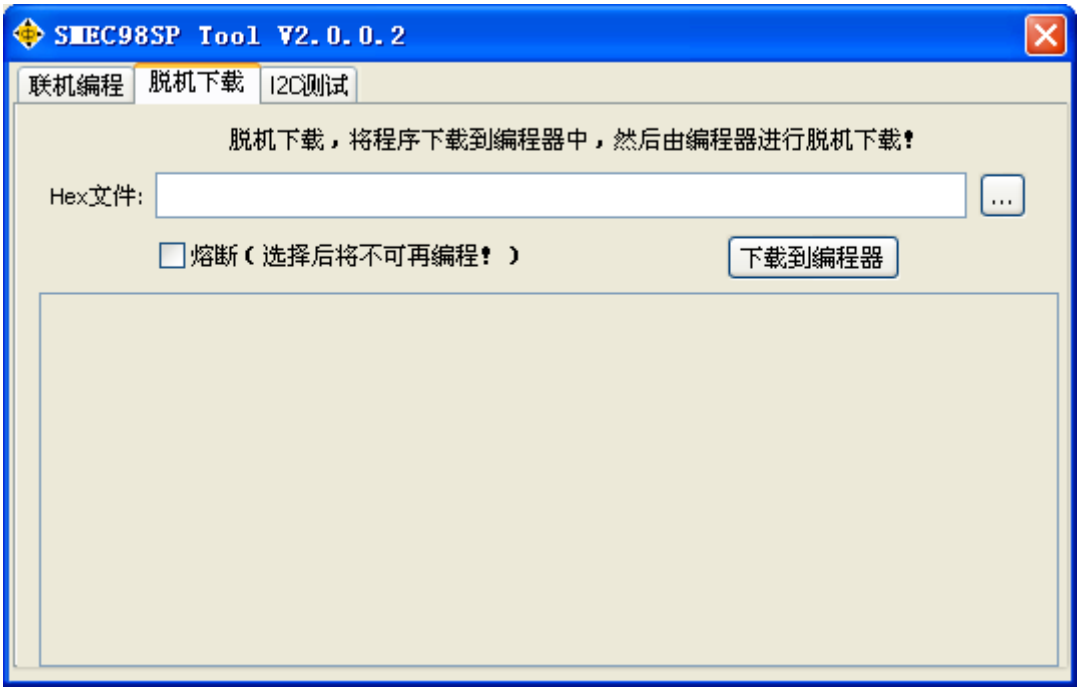
操作步骤：

1. 确定开发板 U3 跳线为“ON”状态。
2. 用 USB 线将开发板链接到电脑。
3. 选择 Keil C 编译的 hex 文件。
4. 根据需要，确定是否需要选择“熔断”选项。
5. 将加密芯片放入开发板的 SOP8 芯片座（U1）中，注意第一脚方向。
6. 点击“开始编程” 按钮。

成功下载后，开发板中指示灯绿灯（OK）亮起，蜂鸣器发出滴的一声。出现错误，红灯（FAIL）亮起，蜂鸣器发出长鸣。

4.2 脱机下载

脱机下载，将 Keil C 开发的 hex 文件，下载到开发板中，然后由开发板，批量下载加密芯片程序。可以单颗芯片，手工下载，也可以连接批量的下载设备。



熔断选项说明：不选中“熔断”时，加密芯片可以被反复编程，

“熔断”选中时，加密芯片将被锁死，不允许再次编程。

操作步骤：

1. 确定开发板 U3 跳线为“ON”状态。
2. 用 USB 线将开发板链接到电脑。
3. 选择 Keil C 编译的 hex 文件。
4. 根据需要，确定是否需要选择“熔断”选项。
5. 点击“下载到编程器” 按钮，将程序脱机下载到开发板中。
6. 将开发板 U3 跳线跳至“OFF”状态，启用开发板脱机下载模式。
7. USB 加电（或直接用 5V 电源供电）。
8. 将加密芯片放入开发板的 SOP8 芯片座（U1）中，注意第一脚方向。
9. 按下开发板 U4 编程按钮开关。

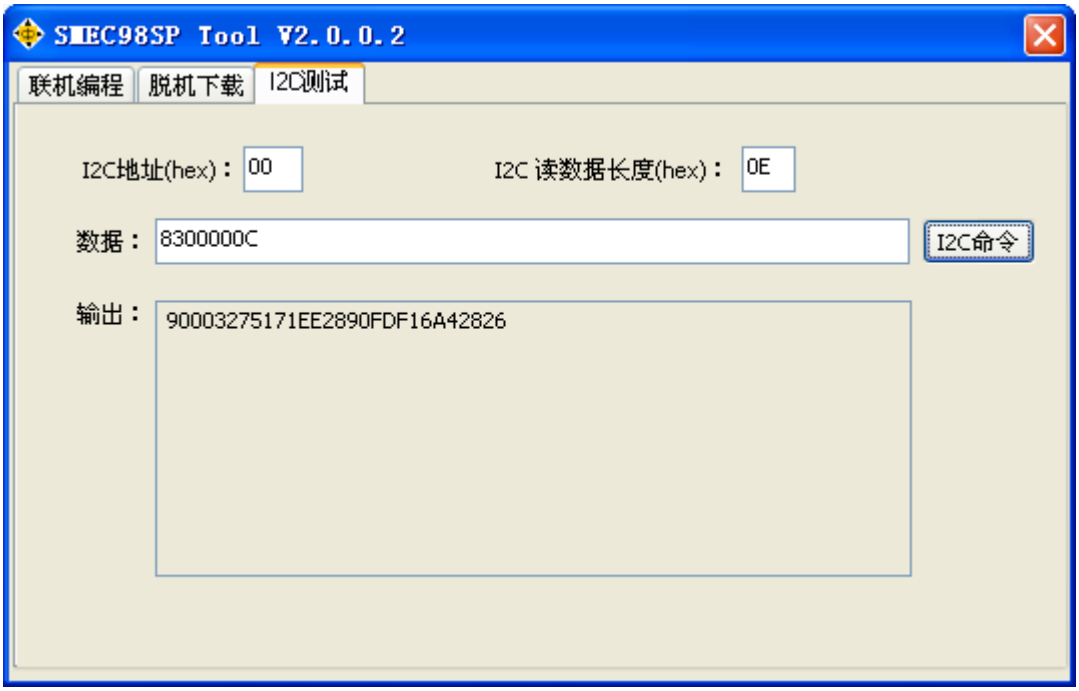
成功下载后，开发板中指示灯绿灯（OK）亮起，蜂鸣器发出滴的一声。出现错误，红灯（FAIL）亮起，蜂鸣器发出长鸣。

脱机下载，也可以连接自动烧录机下载，具体连线方式，请参照第 3 章第 2 节自动烧录机接口。

4.3 I2C 测试

I2C 测试，通过开发板，建立 SMEC98SP 加密芯片与 PC 端的链接，然后通过 PC 端软件，

操作加密芯片。实现在 PC 端测试加密芯片功能。



I2C 地址(hex)：加密芯片编程完成后的新的 I2C 地址，16 进制。
I2C 读取数据长度(hex)：每次操作，需要在 I2C 总线上读取数据的长度，16 进制。
数据：需要通过 I2C 写操作，发送给加密芯片的数据。
输出：返回操作结果。

操作步骤：

1. 确定开发板 U3 跳线为“ON”状态。
2. 用 USB 线将开发板链接到电脑。
3. 输入编程后的加密芯片的 I2C 地址。
4. 在“数据”域，填入要通过 I2C 写操作发送给加密芯片的命令。如：“8300000C”
5. 确定要从 I2C 读取的数据长度。如“0E”
6. 点击 I2C 命令。

第 5 章 SMEC98SP 开发流程

本章简单描述了基于 SMEC98SP 加密芯片的开发过程，开发者可以根据自己的实际情况，确定开发步骤。

5.1 编写加密芯片程序

建议在 SMEC98SP_Demo 的样例工程上进行修改。

1. 设定加密芯片的工作频率。
2. 设定加密芯片的地址(I2CADR)
3. 查询方式接收 I2C 总线上通信数据。(调用 i2c_xfer 函数)
4. 根据接收到的 I2C 命令，判断其合法性，并处理。将处理结果放入全局变量 I2C_Buf，并设定 I2C_send_bytes，指定数据长度。再调用 i2c_xfer 函数，等待 I2C 总线 read 操作。

5.2 联机下载程序到加密芯片中

按照第 4 章第 1 节联机下载的方法，将 keil C 编译的 hex 文件，烧录到加密芯片中。

5.3 用 SMEC98SP Tool 工具测试

按照第 4 章第 3 节所描述的 I2C 测试方法，进行功能上的测试。测试完全 OK 后，进行后续操作。

5.4 根据原理图画 PCB 板

SMEC98SP 加密芯片，只有 4 个脚与外界连接，分别是：VDD，GND，SCL，SDA。其他未用到的管脚请悬空。

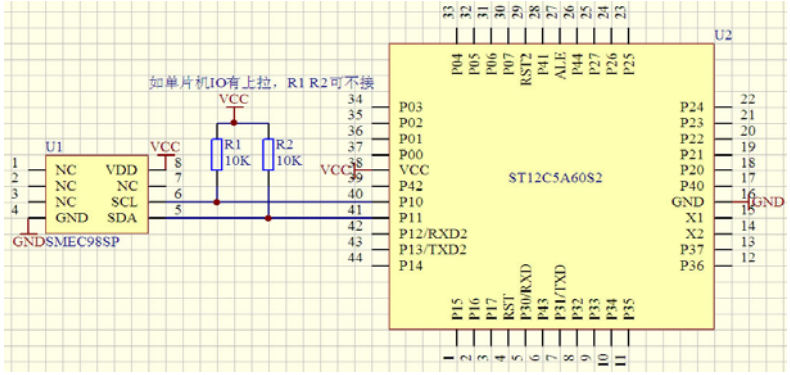


图 5-1：SMEC98SP 典型电路

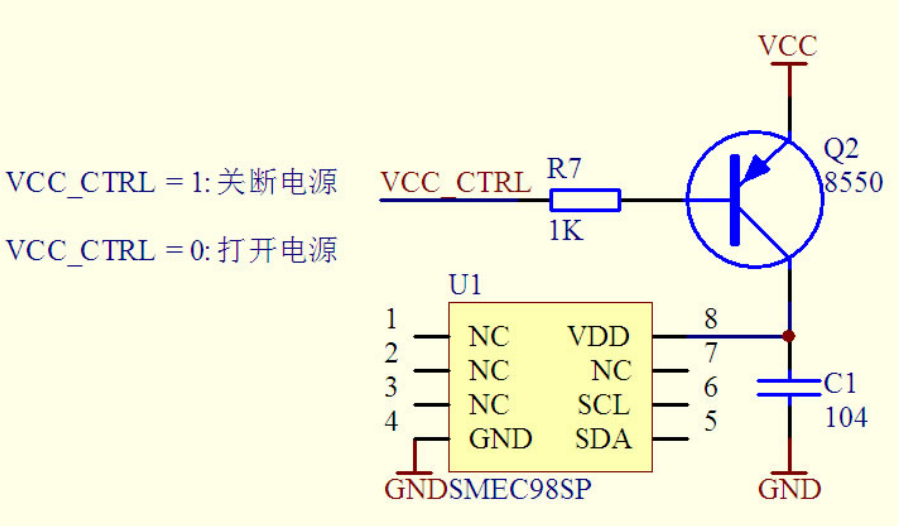


图 5-2：SMEC98SP 带电源控制参考电路图

5.5 在 MCU 中调试

根据自己的实际应用，加入 MCU 中调试，从而启动加密芯片的保护。

5.6 脱机下载加密芯片程序

一切调试 OK 后，可由开发板脱机下载加密芯片程序，操作过程请参见第 4 章第 2 节脱机下载。

第 6 章 SMEC98SP 典型设计

本章列举了几种常见的安全设计机制，由于 SMEC98SP 加密芯片为可编程芯片，开发者可以自行组合这几种安全机制，也可另行开发出适合自己需求的安全机制。

6.1 基于 PIN 认证的安全设计

PIN 认证，也叫口令认证或密码认证。是用一组特定长度的数据，来确定使用者的合法身份。实际过程描述如下：

1. 预先在 MCU 和加密芯片中分别存储 8 字节密码：55 66 77 88 99 AA BB CC
2. 由 MCU 向加密芯片中发 PIN 认证操作：70 00 00 08 55 66 77 88 99 AA BB CC
3. 加密芯片收到 PIN 认证命令后，将判断 PIN 是否正确。如果正确，则返回正确响应码，并允许做后序操作。否则返回错误码，并且不允许后续相关操作。

优点：操作简单

缺点：PIN 在 I2C 线路上传输，容易被截获。

6.2 基于对称密钥的安全设计

SMEC98SP 支持硬件 DES，3DES 的对称算法，计算一组 DES 时间为 7.2us(时钟频率 20M 时)。

对称算法用于安全设计原理为：数据发信方将明文（原始数据）和加密密钥一起经过特殊加密算法处理后，使其变成复杂的加密密文发送出去。收信方收到密文后，若想解读原文，则需要使用加密用过的密钥及相同算法的逆算法对密文进行解密，才能使其恢复成可读明文。在对称加密算法中，使用的密钥只有一个。

我们例程中采用了银行卡的外部认证方法，写了一个简单示例，具体实现如下：

1. 预先在 MCU 和加密芯片中分别存储 8 字节密钥：88 77 66 55 44 33 22 11
2. 由 MCU 发送取随机数指令：84 00 00 08
3. 加密芯片返回随机数：23 ED FE 2A DF 42 A0 F6
4. MCU 用预存的 8 字节密钥，对随机数做 DES 运算，结果为：CD C2 F8 93 2A D0 17 8A
5. MCU 发送外部认证命令：82 00 00 08 CD C2 F8 93 2A D0 17 8A
6. 加密芯片用预先存储的密钥解密，所收到的数据，并与之前生成的随机数做比对，一致则返回正确响应码，并允许做后序操作。否则返回错误码，并且不允许后续相关操作。

优点：密钥不出卡，即使在 I2C 线路上监听，也无法破解。

缺点：DES/3DES 算法对 MCU 运算速度有一定的要求。

密钥还是需要在 MCU 中存储，如果 MCU 被破解后，并被反汇编。侵入者理解了整个汇编代码，还是有可能被破解。

6.3 基于 Hash 的安全设计

把任意长度的输入，通过哈希算法，变换成固定长度的输出，该输出就是哈希值。这种转换是一种压缩映射。主要 Hash 算法有 MD5, SHA1 等。

可以预先在 MCU 和加密芯片中分别存储一组相同长度的数据，作为 Hash 算法的初始输入，然后由 MCU 另外加输入数据，向加密芯片发起 Hash 运算。后将加密芯片返回的 Hash 值与 MCU 自己算的 Hash 值做比对，若一致则 MCU 继续工作，否则停止工作。

优点：即使在 I2C 线路上监听，也无法破解。

缺点：初始输入还是需要在 MCU 中存储，如果 MCU 被破解后，并被反汇编。侵入者理解了整个汇编代码，还是有可能被破解。

6.4 基于算法嵌入的安全设计

将 MCU 中的一部分关键代码，放入加密芯片中运行，当需要用到 SMEC98SP 中的算法时，由 MCU 向 SMEC98SP 发送指令，SMEC98SP 根据指令，在内部运行，返回结果给 MCU。

我们例程中写了一个算圆周长的简单示例，具体实现如下：

1. 加密芯片中存储算圆周长关键算法(周长 C = 2 * π * R)
2. 由 MCU 发送算圆周长指令：72 00 00 01 03 （R = 03）
3. 加密芯片根据 R 值，利用周长公式，算出周长 0x12，返回给 MCU。

优点：关键算法在加密芯片中，即使 MCU 被破解，并被理解反汇编代码，也无济于事。

缺点：暂无