AN44168

使用外部微控制器（HSSP）对 PSoC® 1 器件进行编程

相关项目：有

相关器件系列：CY8C21x23、CY8C21x34、CY21x45、CY8C22x45、CY8C23x33、CY8C24x23、CY8C24x94、CY8C27x43、CY8C28xxx、CY8C29x66、CY8CTST1xx、CY8CTMG1xx、CY8CTMA120

软件版本：PSoC Designer™ 5.4 SP1

相关文档：请点击此处查看相关文档的完整列表。

AN44168 向您介绍了如何通过 C 代码使用外部微控制器对 PSoC®1 器件进行编程。在该过程中，主机微控制器（又称为主机发起串行编程处理器（HSSP））通过系统内串行编程（ISSP）接口对 PSoC 1 进行编程。本应用笔记使用了 C 代码进行编程，使其移植到其它微控制器的代价尽量小，从而加快 PSoC 1 的 HSSP 应用的开发速度。该代码是按照 PSoC 1 ISSP 编程规范中所介绍的编程流程进行编译的。

目录

1 简介 1
2 HSSP 固件架构....2
2.1 ISSP 协议物理层....2
2.2 ISSP 协议数据包层 2
2.3 HSSP 编程步骤....2
3 用于 PSoC 1 HSSP 编程的硬件连接....4
4 将 HSSP 应用移植到主机编程器....4
4.1 需要移植的文件 4
4.2 移植时所要求的代码更改 5
4.3 属性选择 5
4.4 底层驱动程序修改....6
4.5 向 RAM 缓冲区中加载数据....7
5 修改闪存区块顺序或数量 7
6 使用内置测试点验证闪存写入时间....7
7 UART 调试信息 9
8 限制 9
9 总结 10
10 相关文档 10
A 附录 A：端口位操作函数 11
文档修订记录....12
全球销售和设计支持....13

1 简介

赛普拉斯的可编程片上系统（PSoC）使用方便并且非常灵活，它还具有一系列经济高效的可编程模拟和数字资源。通过这些特性可以实现各种创意设计，例如使用板上主机处理器对 PSoC 进行串行编程。该方法用于现场安装或更新固件，重新编程 PSoC，从而实现其他功能。

HSSP 源代码由赛普拉斯创建，作为系统设计人员创建自己的串行编程软件的基础。设计人员必须最大限度地减少对该代码的修改，以保证它能够与具有该特定的主机编程器兼容。该源代码可应用于多种 PSoC 器件，且为器件带来较高的抽象性。本应用笔记所涉及的器件系列列出在“相关器件系列”中。更多有关 ISSP 的信息，请参考 www.cypress.com/?rID=2907 和 www.cypress.com/?rID=2908 中的 PSoC 1 ISSP 编程规范。有关 CY8C20xx6、CY8CTMG2xx 和 CY8CTST2xx 器件的 HSSP 应用，请参考应用笔记 AN59389。AN59389 实现 ISSP 编程规范 — CY8C20045、CY8C20055、CY8C20065、CY8C20xx6A、CY8C20xx7 中介绍的编程流程。

Bootloader 和 HSSP 之间的区别：

在嵌入式应用中，bootloader 通过一个标准的通信接口更新系统固件。除了支持 bootloader 之外，PSoC 1 还通过使用 HSSP 来实现系统内编程。本节介绍了 HSSP 和 Bootloader 的主要区别。选择最符合您要求的方式。

- 局部固件更新和器件完全编程的区别为：HSSP 仅支持在 PSoC 1 中对闪存区进行完全编程，但不支持局部固件更新。但 Bootloader 仅能更新闪存区中的特定部分（可引导加载区域）。预留了闪存区的一部分（bootloader 区域），用于引导加载操作，并且不能对该部分进行更新。
- 外部主机和 PSoC 1 之间的通信接口：各 Bootloader 通过标准通信接口（包括 USB、I^2C、UART 和 SPI）中的一种来更新 PSoC 1 固件。PSoC 1 中的 HSSP 始终使用 ISSP 协议进行编程。

2 HSSP 固件架构

PSoC 1 的 HSSP 用 C 代码分为多个层次来实现。这些层次包括：

1. ISSP 协议物理层
2. ISSP 协议数据包层
3. HSSP 编程步骤层

请参见图 1，了解这些层次之间的控制流程。

2.1 ISSP 协议物理层

下面表描述了组成 ISSP 协议物理层的文件。

源文件	说明
ISSP_PHYSICALLAYER (.C)	该文件所包含的各种宏和函数用于操作编程引脚（SDA 和 SCL），并控制目标器件的复位和电源引脚。这些引脚上的信号是通过 Bit bang 方式生成的。本文件中的各个函数由数据包层的更高位函数调用。

该文件中的固件是针对 PSoC 1 CY8C29466 作为主机微控制器编写。要想将该文件移植到其他主机微控制器内，您应该正确修改所有函数和宏。

2.2 ISSP 协议数据包层

下面表描述了组成 ISSP 协议数据包层的文件：

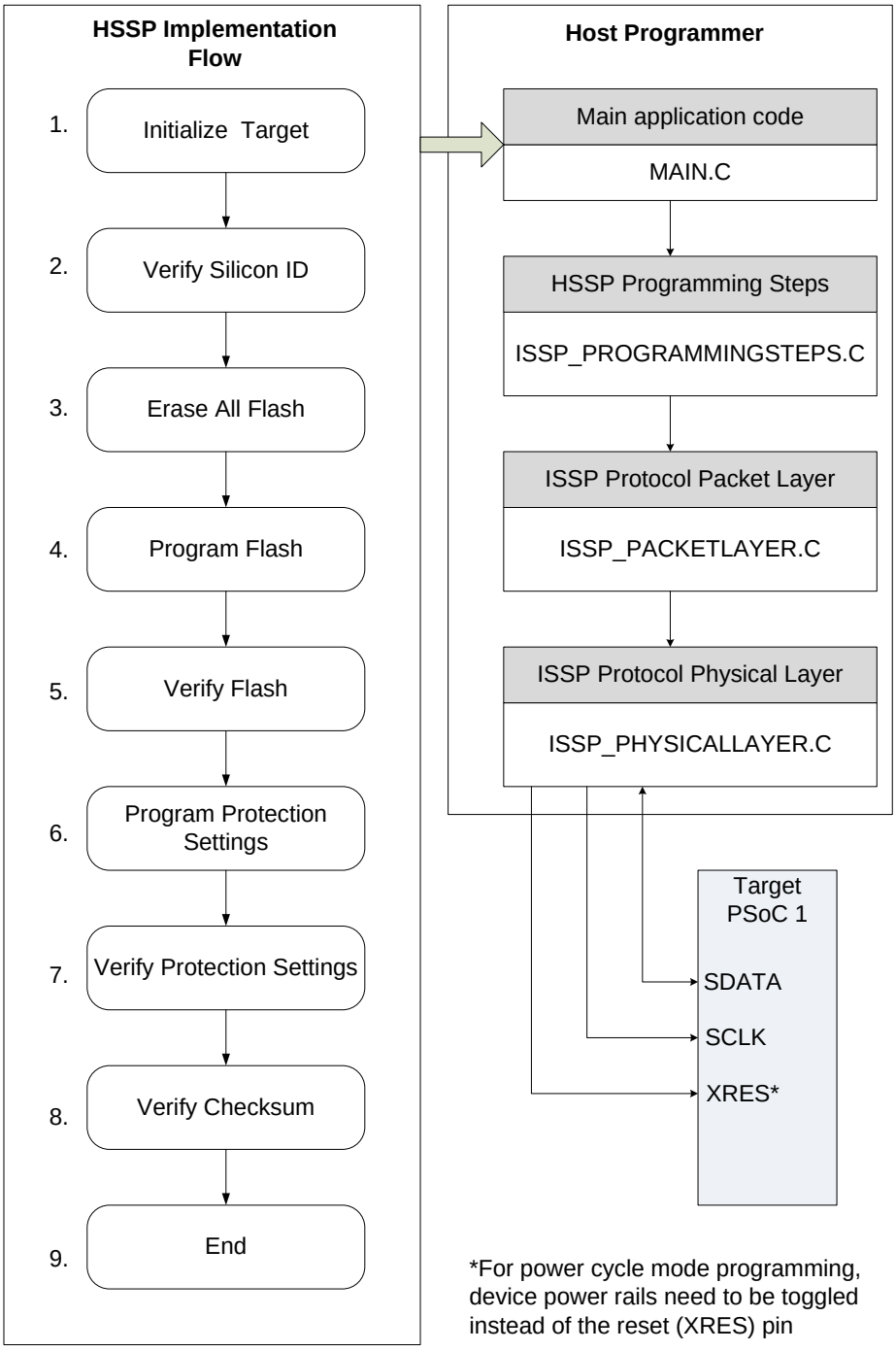
源文件	说明
ISSP_PACKETLAYER (.C)	该文件使用了 ISSP_PhysicalLayer 文件中所定义的各个函数进行发送 ISSP 向量。ISSP 向量只是位的序列，它代表一组指令。主机处理器将这些指令发送给目标器件执行。这些向量会因编程阶段的不同而有所改变。更多有关 ISSP 向量的信息，请参考 PSoC 1 ISSP 编程规范。该文件中的函数用于读取数据，并将向量数据包写入到目标器件中。ISSP_PROGRAMMINGSTEPS.C 文件中的函数直接调用本层所定义的各函数。

2.3 HSSP 编程步骤

执行 HSSP 编程步骤的文件包括：

源文件	说明
ISSP_PROGRAMMINGSTEPS (.C)	该文件包含 HSSP 应用的顶层函数。要想了解每一步的信息，请参考 PSoC 1 ISSP 编程规范。

图 1. PSoC 1 HSSP 固件构架



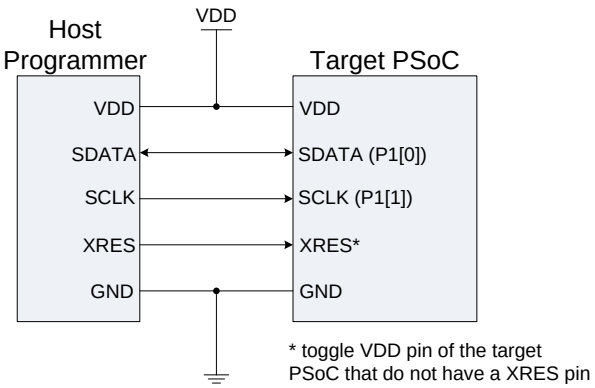
3 用于 PSoC 1 HSSP 编程的硬件连接

图 2 显示了主机编程器和目标 PSoC 4 器件之间所需要的基本硬件连接。可以使用两种不同的模式进行编程 PSoC 器件，即：复位和电源重启。当 PSoC 器件存在 XRES 引脚时，将使用复位模式（首选的编程模式） 。对于不带有 XRES 引脚的器件，必须将其编程为电源重启模式。在这种情况下，HSSP 微控制器将在器件的“ON”和“OFF”状态间切换。在这两种编程模式下，主机均需要三个 I/O 引脚。具体情况为：复位模式中的串行数据（SDATA）、串行时钟（SCLK）和外部复位（XRES）引脚，以及电源重启模式中的 SDATA、SCLK 和 PSoC 电源（PWR）控制引脚。这些引脚通过固件控制。

主机处理器上的 SDATA 引脚必须为双向引脚。主机必须拥有修改该引脚属性的能力；使之变成强驱动（写入时） 或高阻态模式（读取时）。

有关编程模式、编程接口连接和编程规范的更多信息，请查看 www.cypress.com/?rID=2907 和 www.cypress.com/?rID=2908 网址上的 PSoC 1 ISSP 编程规范。

图 2. 主机编程器和目标器件之间的基本连接



4 将 HSSP 应用移植到主机编程器

本应用笔记所提供的项目将 PSoC 1 CY8C29466 作为主机编程器使用。在 HSSP 应用中，主机编程器可以是任何其他微控制器。本节说明了将 HSSP 应用代码移植到特定主机以编程目标器件时所需要的更改。

4.1 需要移植的文件

表 1 显示的是需要移植到系统主机设备内的各个文件。

表 1. 需要移植的文件

需要移植的头文件	需要移植的源文件
ISSP_DEFS.H	ISSP_PHYSICALLAYER.C
ISSP_DELAYS.H	ISSP_PACKETLAYER.C
ISSP_DIRECTIVES.H	ISSP_PROGRAMMINGSTEPS.C
ISSP_ERRORS.H	
ISSP_REVISION.H	
ISSP_VECTORS.H	

4.2 移植时所要求的代码更改

表 2 显示的是将附件的 HSSP 应用移植到除 PSoC 1 外任何主机编程器时源文件需要更新的项目。

表 2. 代码变更

文件	说明	需要更改
ISSP_DEFS.H	该文件包含了同所有 PSoC 1 器件有关的闪存组数量、每一组中的模块以及安全字节数量的信息。	无需更改。
ISSP_DELAYS.H	该文件包含了用于目标器件的等待和轮询事件、XRES 触发和电源重启的定时器设置内容。	根据主机处理器的运行速度调整延迟计数。
ISSP_DIRECTIVES.H	该文件为用户提供了各个选项：选择目标 PSoC 1 器件、使能 / 禁用调试代码（UART、测试点）、选择目标电压和编程模式 — 复位或电源重启。	请参考属性选择一节的内容。
ISSP_ERRORS.H	该文件包含了编程过程中可能发生的不同错误的#define 语句。	无需更改。
ISSP_EXTERN.H	该文件包含函数和变量原型。	无需更改。
ISSP_REVISION.H	该文件用于版本控制。	无需更改。
ISSP_VECTORS.H	该文件包含所有 ISSP 指令的向量。	无需更改。
ISSP_PHYSICALLAYER.C	该文件包含低层驱动程序，用于与目标器件进行通讯。	请参考底层驱动程序修改一节的内容。
ISSP_PACKETLAYER.C	该文件包含了各种函数，用于为不同编程步骤创建向量数据包。	需要更新代码才能接收编程数据。请参考向 RAM 缓冲区中加载数据一节的内容。
ISSP_PROGRAMMINGSTEPS.C	该文件包含了用于为编程流程中不同步骤调用个别函数的代码。	无需更改。

4.3 属性选择

设计人员必须设置以下三个属性：属性、标签和说明。要想实现该设置，可以对 `ISSP_DIRECTIVES.H` 文件中的某些 `#define` 语句添加或撤销注释。这些`#define` 语句都标注了‘User Attention Required’（用户需要注意），非常明显，并且容易找到。您还可以进行页面搜索，查找单个标签。下面部分是每个属性及其标签的说明。

属性： UART 调试
标签： `USE_TX`
说明： 注释该`#define` 语句，从而禁止将信息发送在 UART 上的代码； 撤销注释以使能 UART 函数的执行。

属性： 测试点
标签： `USE_TP`
说明： 注释该`#define` 语句，以禁止代码的执行（其位拆裂（bit bang）测试引脚）。该引脚用于调试目的。在终端设计中不需要该引脚。

属性： 目标供电电压
标签： `TARGET SUPPLY VOLTAGE`
说明： 如果目标电压为 3.3 V，将添加注释`#define` 语句；如果目标电压为 5 V，将撤销注释该语句。

属性：编程模式
标签：PROGRAMMING MODE
说明：如果您使用电源重启模式，需要添加注释#define 语句。如果撤销注释#define，将采用复位模式对目标进行编程。

属性：目标 PSoC 器件
标签： TARGET PSOC
说明：在此部分中选择目标 PSoC。在所有给定情况中，都只能使能一个器件，所有其他的器件都带有注释。如果列表中没有该器件，请与赛普拉斯技术支持联系。

4.4 底层驱动程序修改

设计人员需要提供主机指定代码，以控制编程目标 PSoC 时所涉及到的引脚。这些 API 可以在 ISSP_PHYSICALLAYER.C 中找到，并标为“Processor Specific”（特定于处理器）和‘User Attention Required’（用户需要注意）。

- **端口位掩码（Port Bit Masks）：**对于所使用的特定主机处理器，必须调整四个端口位掩码。请注意，虽然要设置四个位，但根据所使用的编程方法，编程过程中只需要使用以下三位：复位模式下的 SDATA、SCLK 和 XRES；电源重启模式下的 SDATA、SCLK 和 PWR。
- **Delay(n)函数：**调整该函数，使“while”周期的循环时间至少为 2 µs。循环时间通常没有上限。但是，循环时间越长，对目标的编程时间就越长。例如，如果主机微处理器也是一个 PSoC，那么每次小循环大概需要 2 µs，且主机微处理器的再需要 7 µs 的时间进行处理。因此，该函数将产生 2n+7 µs 的延迟，其中 n 是传递给该函数的参数。要想调整主机处理器的延迟时间，请修改 ISSP_DELAYS.H 中的#define 语句。
- **端口位操作函数：**这些函数将控制主机的各引脚，以产生 PSoC 编程所需的信号。这些函数用于将引脚驱动为高电平或低电平，以及将引脚释放至高阻状态。以下列出了这些函数。大多函数的用途都非常明显，但在代码中仍有记录。

```
fSDATACheck()
SCLKHigh()
SCLKLow()
SetSCLKStrong()
SetSDATAHigh()
SetSDATALow()
SetSDATAHiZ()
SetSDATAStrong()
SetXRESStrong()
AssertXRES()
DeassertXRES()
SetSCLKHiZ()
SetTargetVDDStrong()
ApplyTargetVDD()
RemoveTargetVDD()
```

- **UART 函数：**这些函数用于调试目的。将会通过 UART 发送信息，以通知编程流程的进度。用户应根据主机处理器修改下面函数。TX8 用户模块被放置在设计中以用于 UART 发送器。

```
InitTx()
SendDebugMessage()
PrintReceivedSiliconID()
DisplaySecurityValues()
```

- **测试点动能：**这些测试点也用于调试目的。修改下面函数以执行主机处理器操作：

```
InitTP()
SetTPHigh()
SetTPLow()
ToggleTP()
```

4.5 向 RAM 缓冲区中加载数据

HSSP 代码将按顺序依次从 64 字节缓冲区取出数据，用以编程各 PSoC 闪存模块。该过程从最低模块地址开始。对第一个模块进行编程后，将使用同一个缓冲区对后续闪存区块进行编程。

设计人员必须根据数据源（USB、RS-232、SD 卡等）编写代码，填充该缓冲区。对于所使用的特定主机处理器，应编写以下两个函数： – `LoadProgramData()`和 `LoadSecurityData()`。这些函数可以在 `ISSP_PACKETLAYER.C` 中找到，并标记为‘Processor Specific（特定于处理器）’和‘User Attention Required（用户需要注意）’。一般，这些函数会将递增值加载到缓冲区内。

5 修改闪存区块顺序或数量

在某些情况下，必须对闪存中的特定区域进行编程。例如，专用于存储特征、校准或固件现场升级的区域。这些特性通常通过 EEPROM 用户模块实现。但在某些情形下，如果代码空间受限，则可以在 PSoC 中直接编写这些特性，从而节省代码空间。

您可以更改目标区块的起始地址和各摸块的编程顺序。上述操作不会产生任何问题，因为每个编程顺序都包含了模块地址。但是，请记住以下几点：

- 对于每个模块写操作，只能对闪存组的序号进行一次设置。由于其他系列只有一个闪存组，一次这种情况仅适用于 CY8C29x66、CY8C28xxx、CY8C24x94、CY8CTST120、CY8CTMG120 和 CY8CTMA120 产品系列。有关各闪存组的更多信息，请参考 PSoC 技术参考手册（TRM）。通过 `FLS_PR1` 寄存器可以确定受编程调用影响的闪存模块。
- 如果修改了编程循环，则必须对验证循环进行相同的修改，否则验证将会失败。
- 代码在运行过程中会累加校验和，并根据截至当时的整个闪存检查校验和。如果您只是对闪存的某个区域进行了编程，那么应该相应设置变量 `iChecksumData`。

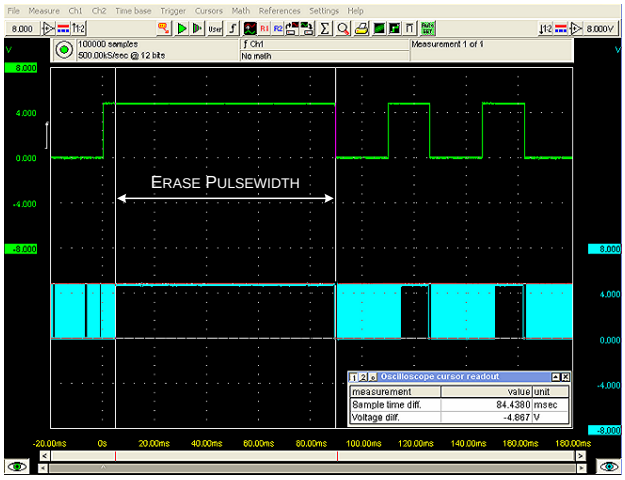
6 使用内置测试点验证闪存写入时间

成功实现主机源编程最关键的一个因素是采用正确的擦除和写入脉冲宽度。为了帮助您完成该过程，在程序的重要位置执行了几个测试点（TP）调用。要想使能该调试模式，请在 `ISSP_DIRECTIVES.H` 中对 `USE_TP #define` 撤销注释。有几个函数与调试模式相关，它们与本应用笔记前面提及的引脚操作函数相似。系统设计人员必须提供主机指定的代码，以驱动引脚输出高电平、低电平或切换引脚。

正确的调试过程需要监控 TP 和 SDATA 线，且必须测量擦除和编程脉冲。实现该目的的最佳方式是使用双通道示波器，并在 TP 通道的上升沿上进行触发，使其进入单序列模式。

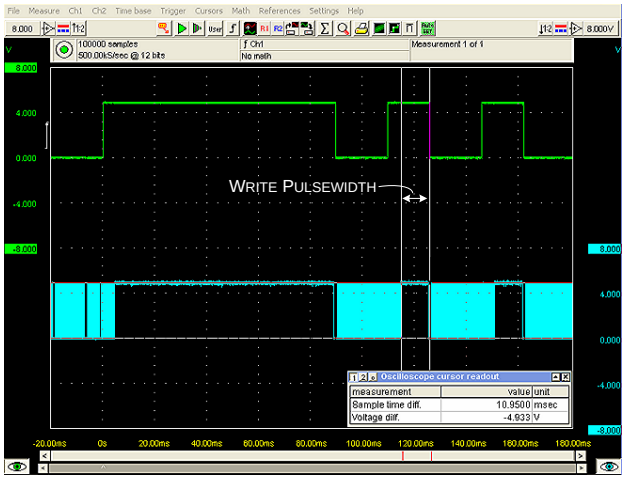
擦除脉冲宽度从数据突发的末端开始测量，到 TP 下降沿结束，如图 3 所示。请注意，TP 上升沿与数据突发末端不在一条线上。由于 TP 引脚被驱动为高的时间与主机开始发出数据的时间间存在延迟，因此要预防发生这种情况。

图 3. 测量擦除脉冲宽度



编程脉冲宽度亦是从数据突发的末端开始测量，到 TP 下降沿结束。图 4 显示的是该测量。与擦除脉冲宽度一样，TP 信号的上升沿与数据突发的末端也不在一条线上。

图 4. 测量写入脉冲宽度



各种 PSoC 器件的理想擦除和写脉冲宽度，请参考相应器件数据手册中的“交流编程规范”表。测量值必须在-3%到+15%的理想值范围内。如果无法满足该要求将导致错误编程，并产生不利影响，例如比指定规范更短的闪存数据保持时间[1]，以及比预期更少的闪存擦除和写入循环[2]。

[1] 在器件数据手册的“直流电编程规范”部分对该情况进行了规定，并使用符号 $Flash_{DR}$ 标出。

[2] 在器件数据手册“直流电编程规范”部分予以规定，并以符号 $Flash_{ENPB}$ 和 $Flash_{ENT}$ 标出。

7 UART 调试信息

可以使用现有的代码（使能了 UART）发送调试信息。图 5 显示的是编程过程中被发出的状态信息示例。

图 5. 状态信息



8 限制

系统设计人员要仔细研究 *main.c* 开头部分的注释所包含的有用和重要信息。这些注释对 HSSP 代码的某些限制进行了说明。下面是简短摘要。

- 串行编程只能在 5 °C 和 50 °C 的温度范围内进行。
- HSSP 程序不支持 3.0 V 以下的电压。
- 编程过程仅在一个电压范围内完成。如果器件初始电压为 5.0 V，则整个过程必须在 5.0 V 范围内、向量为 5.0 V 完成。

- 当前不支持 CY8C20x34 和 PSoC 过期版本。有关该应用笔记不支持的器件列表，请参考随附项目中 *main.c* 的头部分。
- 器件数据手册的“交流编程规范”部分指定了 SCLK 频率的上限，以符号 F_{SCLK} 标出。
- 赛普拉斯不推荐 CY8C20x36/46A/66A/96A/ CY8CTMG2xx/ CY8CTST2xx 器件与其他 PSoC 器件共同使用 ISSP 总线。但在 CY8C20x36/46A/66A/96A/ CY8CTMG2xx/ CY8CTST2xx 器件与其他 PSoC 器件共同使用 ISSP 总线时，应避免 CY8C20x36/46A/66A/96A/ CY8CTMG2xx/ CY8CTST2xx 器件检查到‘AC52’键处于复位状态。更多有关信息，请参考知识库文章，CY8C20X36A/46A/66A/96A：共享 ISSP 总线的问题。

9 总结

本应用笔记提供了一些可用的示例 HSSP 代码，设计人员可以灵活地创建自己的串行编程软件。另外，它还为工程师提供良好的参考，以实现 PSoC 1 主机编程解决方案，并通过提供便携的 C 源代码降低开发解决方案的时间。

10 相关文档

- PSoC® 1 ISSP 编程规范 — CY8C21x23、CY8C21x34、CY8C23x33、CY8C24x23A、CY8C27x43、CY8CTMG110、CY8CTST110
- PSoC® 1 ISSP 编程规范 — CY8C21x45、CY8C22x45、CY8C24x94、CY8C28xxx、CY8C29x66、CY8CTST120、CY8CTMA120、CY8CTMG120、CY7C64215
- ISSP 编程规范
- AN59389 — CY8C20xx6A 的主机源串行编程

A 附录 A：端口位操作函数

表 3. 端口位操作函数

函数名称	说明
SetSCLKStrong()	将 SCLK 引脚设置为输出（强驱动模式）。
SetSCLKHiZ()	将 SCLK 引脚释放至高阻状态。
SetSDATAHigh()	将 SDATA 引脚置为高电平。
SetSDATALow()	将 SDATA 引脚置为低电平。
SetSDATAStrong()	将 SDATA 引脚置为输出（强驱动模式）。
SetSDATAHiZ()	将 SDATA 引脚释放至高阻状态（被目标驱动）。
AssertXRES()	将 XRES 引脚置为高电平。
DeassertXRES()	将 XRES 引脚置为低电平。
SetXRESStrong()	将 XRES 引脚置为输出（强驱动模式）。
ApplyTargetVDD()	给目标 PSoC 供电。
RemoveTargetVDD()	将目标 PSoC 断电。
SetTargetVDDStrong()	将 PWR 引脚设置为输出（强驱动模式）。

文档修订记录

文档标题： AN44168 — 使用外部微控制器（HSSP）对 PSoC® 1 器件进行编程

文档编号： 001-78742

版本	ECN	变更者	提交日期	变更说明
**	3601083	VLX	04/26/2012	本文档版本号为 Rev**，译自英文版 001-44168 Rev*E。
*A	4719977	HENG	04/10/2015	本文档版本号为 Rev*A，译自英文版 001-44168 Rev*G。
*B	5043824	HENG	12/10/2015	本文档版本号为 Rev*B，译自英文版 001-44168 Rev*H。
*C	6070734	XITO	02/15/2018	本文档版本号为 Rev*C，译自英文版 001-44168 Rev*I。

全球销售和设计支持

赛普拉斯公司具有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。要想查找离您最近的办事处，请访问赛普拉斯所在地。

产品

Arm® Cortex® 微控制器	cypress.com/arm
汽车级产品	cypress.com/automotive
时钟与缓冲器	cypress.com/clocks
接口	cypress.com/interface
物联网	cypress.com/iot
存储器	cypress.com/memory
微控制器	cypress.com/mcu
PSoC	cypress.com/psoc
电源管理 IC	cypress.com/pmic
触摸感应	cypress.com/touch
USB 控制器	cypress.com/usb
无线连接	cypress.com/wireless

PSoC®解决方案

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

赛普拉斯开发者社区

社区 | 项目 | 视频 | 博客 | 培训 | 组件

技术支持

cypress.com/support

此处引用的所有其他商标或注册商标归其各自所有者所有。

赛普拉斯半导体公司
198 Champion Court
San Jose，CA 95134-1709

© 赛普拉斯半导体公司，2008-2018 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码形式的向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。

在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。没有任何电子设备是绝对安全的。因此，尽管赛普拉斯在其硬件和软件产品中采取了必要的安全措施，但是赛普拉斯并不承担任何由于使用赛普拉斯产品而引起的安全问题及安全漏洞的责任，例如未经授权的访问或使用赛普拉斯产品。此外，本材料中所介绍的赛普拉斯产品有可能存在设计缺陷或设计错误，从而导致产品的性能与公布的规 格不一致。（如果发现此类问题，赛普拉斯会提供勘误表）赛普拉斯保留更改本文件的权利，届时将不另行通知。在适用法律允许的限度内，赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为供参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何主张、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的主张，并使之免受损失。

赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，WICED，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。