

---

---

## AVR®单片机与串行存储器的接口

---

---

### 特性

---

- EEPROM 器件: Microchip 25xx256 (25AA256/25LC256)
- DataFlash 器件: Microchip SST25VFxxxx (SST25VF010A、SST25VF020、SST25VF040B 和 SST25VF080B)
- SPI DataFlash 驱动程序, 用于 1/2/4/8 Mb SPI 串行闪存和 ATtiny817
- SPI EEPROM 驱动程序, 用于 256K SPI 总线串行 EEPROM 和 ATtiny817
- 完善的串行存储器函数支持
- SST25VFxxxx 特性:
  - 灵活的擦除功能
  - 快速擦除和字节编程
  - 自动地址递增 (Auto Address Increment, AAI) 编程
- 25xx256 特性:
  - 64 字节页
  - 块写保护
  - 内置写保护

### 简介

---

作者: Rupali Honrao, Microchip Technology Inc.

串行接口存储器广泛应用于消费类、汽车、电信、医疗、工业和 PC 相关市场。串行存储器主要用于存储个人偏好数据和配置/设置数据, 是当今使用的最为灵活的非易失性存储器 (Nonvolatile Memory, NVM) 类型。与其他 NVM 解决方案相比, 串行存储器件兼具引脚数更少、封装更小、电压和功耗更低等优势。

大多数 AVR®单片机均提供 SPI 接口, 用于与串行存储器件 (如 EEPROM 25AA256/25LC256) 和 DataFlash 器件 (如 SST25VF010A、SST25VF020、SST25VF040B 和 SST25VF080B) 连接。

为了简化和加速 SPI 串行存储器与 AVR 器件的集成, 我们开发了基本驱动程序以提供高效访问。本应用笔记将介绍这些驱动程序的功能和架构。此外, 本应用笔记还在 Atmel | START 中提供驱动程序源代码。

---

## 目录

---

特性.....	1
简介.....	1
1. 串行存储器.....	3
1.1. 串行存储器与 AVR 硬件的连接.....	3
1.2. SPI 串行存储器访问方法.....	3
1.3. SPI 外设处理.....	9
1.4. 繁忙检测.....	9
2. 从 Atmel   START 获取源代码.....	10
3. 源代码概述.....	11
3.1. SPI 接口的初始化.....	11
3.2. 轮询操作模式函数.....	11
3.3. DataFlash 读和编程函数.....	13
3.4. EEPROM 读和编程函数.....	14
4. 开发环境.....	16
5. 版本历史.....	17
Microchip 网站.....	18
变更通知客户服务.....	18
客户支持.....	18
Microchip 器件代码保护功能.....	18
法律声明.....	18
商标.....	19
质量管理体系.....	19
全球销售及服务网点.....	20

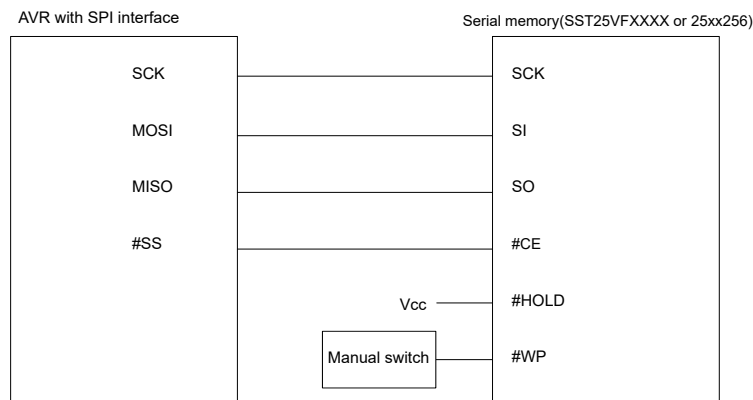
## 1. 串行存储器

以下各节将介绍存储器与 AVR 处理器的连接、存储器访问类型、块写保护位以及繁忙检测。有关更多信息，请参见器件数据手册。

### 1.1 串行存储器与 AVR 硬件的连接

SPI 接口配置为主模式。SCK 和 MOSI 端口为输出，MISO 端口为输入。请参见图 1-1。

图 1-1. 硬件连接



请注意，在本应用笔记中，从选择信号#SS 用于控制串行存储器的芯片使能引脚#CE。

#HOLD 信号可以保持无效高电平状态，因为数据无效时，SPI 接口会始终暂停时钟。

用户可以选择写保护信号#WP 的配置。

注：EEPROM 的工作电压范围为 1.8V 至 5.5V，而 DataFlash 器件的工作电压范围为 2.7V 至 3.6V。

### 1.2 SPI 串行存储器访问方法

为了理解所提供驱动程序的结构，有必要回顾一下不同的串行存储器访问方法。

所有访问都遵循以下序列：

1. 片选线驱动为低电平。
2. 数据线上随 SCK 时钟同步发送或接收大量串行化字节。
3. 片选线驱动为高电平。

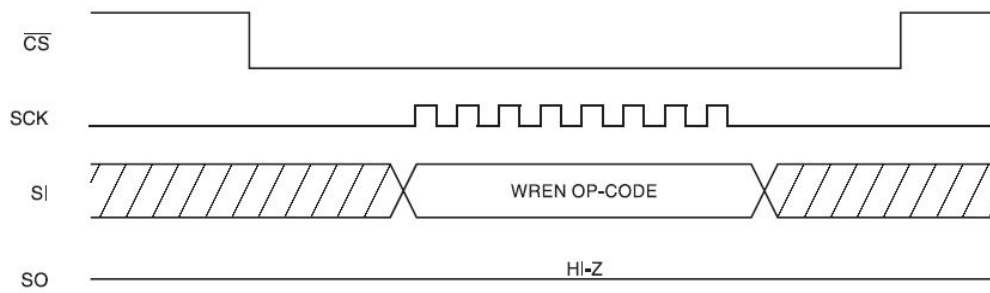
串行化字节的数量和值取决于访问类型。

本应用笔记介绍的 SPI 存储器件均支持 SPI 模式 0 (0,0) 和模式 3 (1,1)。在提到的驱动程序中，选择模式 0 (有关详细信息，请参见串行存储器件数据手册)。

#### 1.2.1 单字节写命令：WREN、WRDI 和 CHIP ERASE

这些访问的长度为单字节。MOSI 数据线上仅发送指令字节（后面称其为“操作码”）。

图 1-2. WREN 时序示例



### 1.2.2 读/写状态寄存器: RDSR/WRSR

这些访问的长度为双字节: 操作码字节后跟要写入 (WRSR) 或读取 (RDSR) 的字节。

注: 在执行写状态寄存器操作前, 必须先执行 WREN 命令。

图 1-3. WRSR 时序

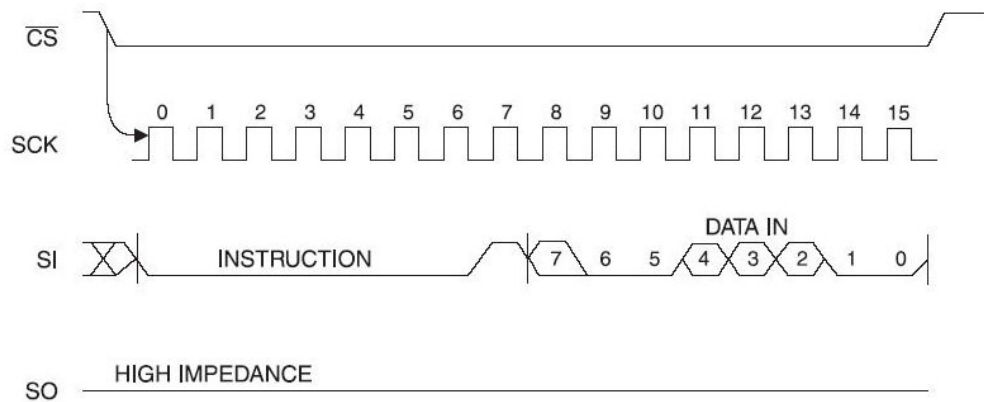
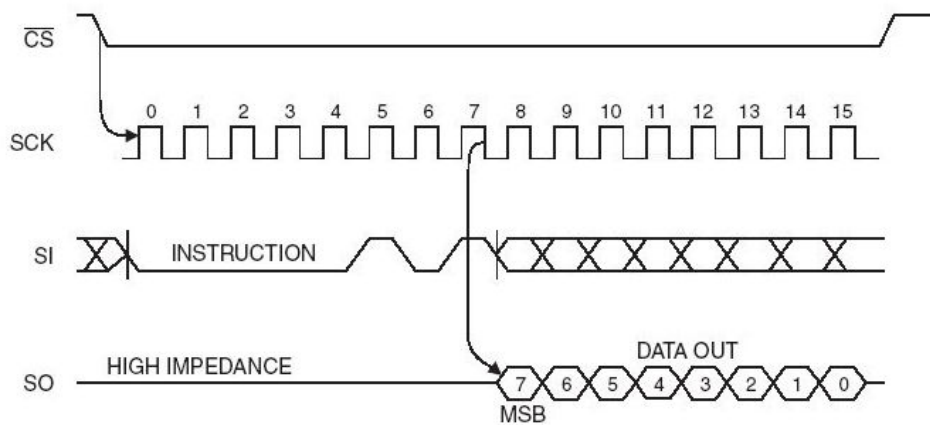


图 1-4. RDSR 时序



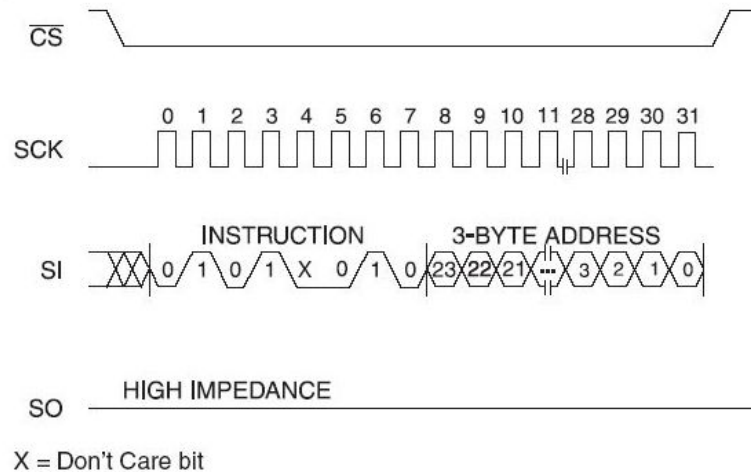
### 1.2.3 扇区/块擦除命令

(仅限 SST25VFxxx 器件)

该访问由一个字节的操作码和三个地址字节组成。图 1-5 给出了 32 KB 的块擦除序列。通过执行 8 位命令 52H (后跟 3 个地址字节) 来启动 32 KB 块擦除指令。

注: 在执行任何写操作之前, 必须先执行写使能 (WREN) 指令。

图 1-5. 32 KB 块擦除时序



### 1.2.4 数据写访问和数据读访问

这两种访问依次由一个字节的操作码、一个地址阶段和一个数据阶段组成。对于 25xx256 器件, 地址阶段是两个字节的地址; 对于 SST25VFxxx 器件, 地址阶段是三个字节的地址。

在尝试写入数据之前, 必须通过发出 WREN 指令将写使能锁存器置 1。

#### 1.2.4.1 25xx256 器件的数据写访问

在启动写周期之前, 最多可将 64 字节的数据发送到器件。惟一的限制是所有字节必须位于同一页中。无论实际写入的字节数如何, 页写操作都仅限于在单个物理页内写入字节。

图 1-6. 字节写序列 (25XX256)

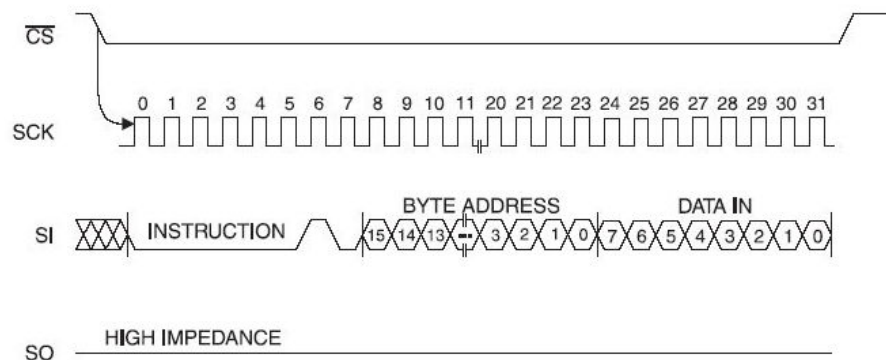
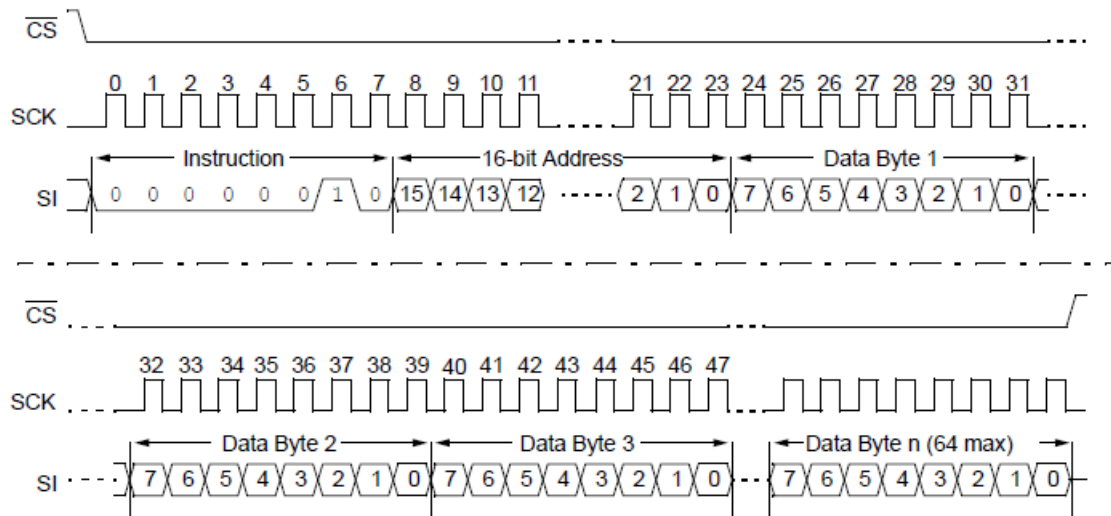


图 1-7. 页写序列 (25XX256)

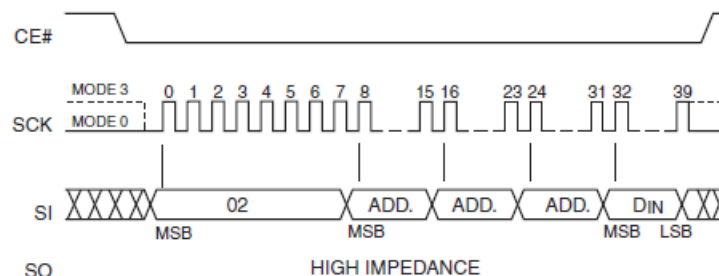


#### 1.2.4.2 SST25VFxxxx 器件的数据写访问

##### 字节编程:

字节编程指令用于将所选字节中的位编程为所需数据, 如图 1-8 所示。

图 1-8. 字节写序列 (SST25VFxxxx)



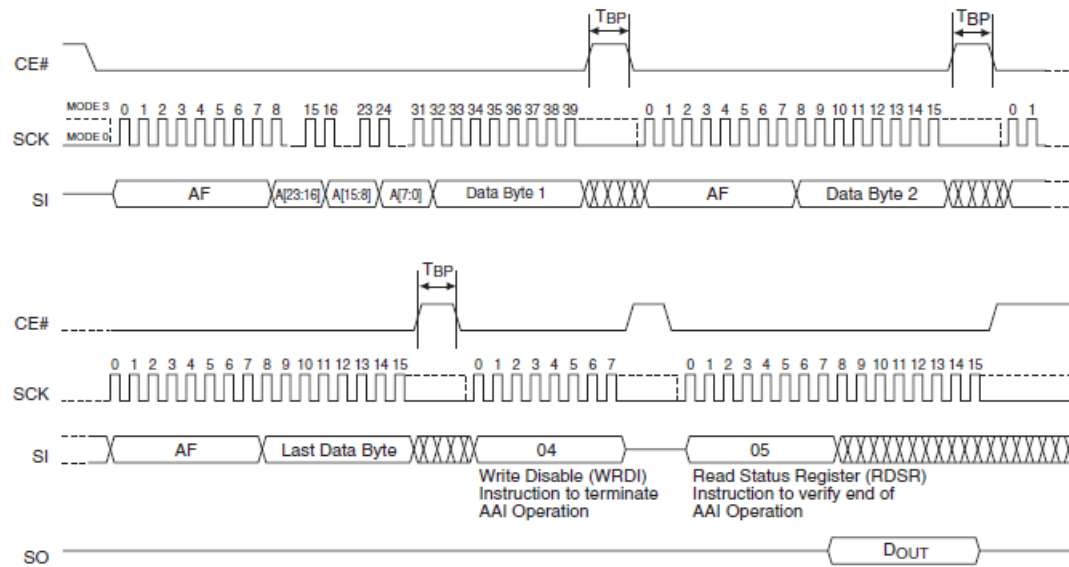
##### 自动地址递增 (AAI) 编程:

AAI 编程指令允许对多个字节的数据进行编程, 无需再次发出下一连续地址单元。当要对整个存储器阵列编程时, 该特性可缩短总编程时间。在执行任何写操作之前, 必须先执行写使能 (WREN) 指令。

通过执行 8 位命令 AFH (后跟地址位[A23-A0]) 来启动 AAI 编程指令。在地址后面, 按照从 MSb (bit 7) 到 LSb (bit 0) 的顺序输入数据。一旦器件完成编程字节, 就可以按顺序编程下一个地址。输入 8 位命令 AFH, 然后输入要编程的数据。当最后一个所需字节完成编程时, 执行写禁止 (WRDI) 指令 04H 以终止 AAI。执行 WRDI 命令后, 用户必须轮询状态寄存器以确认器件已完成编程。

有关 AAI 编程序列的信息, 请参见图 1-9。这适用于器件 SST25VF010A 和 SST25VF020。

图 1-9. 自动地址递增 (AAI) 编程序列 (SST25VF010A/SST25VF020)



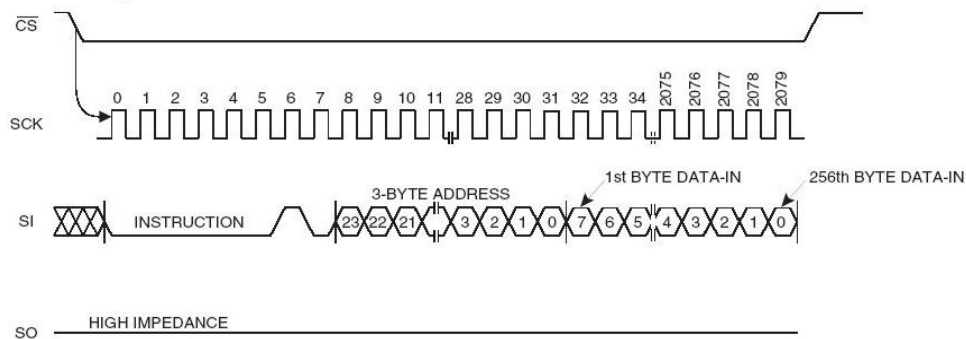
**自动地址递增 (AAI) 字编程:**

自动地址递增 (AAI) 字编程序列与上述自动地址递增 (AAI) 编程序列类似, 不同之处在于:

- AAI 字编程的操作码为 ADH
- 写入两个字节的 数据
- 在硬件写操作结束检测模式下终止 AAI 字编程的序列为: 执行写禁止 (WRDI) 指令 04H, 然后执行 8 位 DBSY 命令 80H。

有关自动地址递增 (AAI) 字编程的信息, 请参见图 1-10。这适用于器件 SST25VF040B/080B。

图 1-10. 自动地址递增 (AAI) 字编程序列 (SST25VF040B/080B)

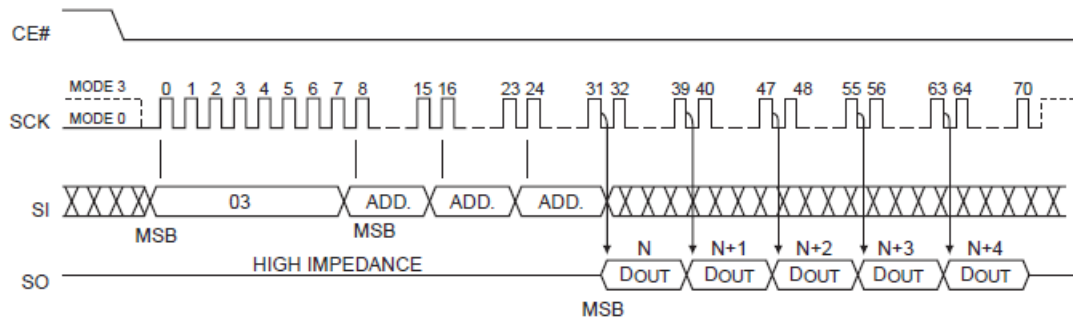


**1.2.4.3 数据读访问**

这种访问依次由一个字节的操作码和一个地址阶段组成。对于 25xx256 器件, 地址阶段是两个字节的数据; 对于 SST25VFxxxx 器件, 地址阶段是三个字节的地址。

通过将 CE 拉为低电平来选择器件。首先发送到器件的是 8 位读指令, 接着是地址。发送正确的读指令和地址后, 存储在存储器中所选地址处的数据将在 SO 引脚上移出。通过继续提供时钟脉冲, 可以按顺序读取存储在存储器中下一地址处的数据。每移出一个数据字节, 内部地址指针便自动递增至下一较高地址。当达到最高地址时, 地址计数器将计满返回到地址 0000h, 这样便可无限期地继续执行读取周期。将 CE 引脚拉为高电平可终止读操作。

图 1-11. 读序列 (SST25VFxxxx)



### 1.2.5 访问类型汇总

下表汇总了各个访问类型及其相应的长度:

表 1-1. 访问类型汇总表

命令	操作码	地址阶段长度 (字节)		数据阶段长度 (字节)	
		25xx256	SST25VFxxxx	MOSI 线	MISO 线
WREN	0000 0110b (06H)	0	0	0	0
WRDI	0000 0100b (04H)	0	0	0	0
RDSR	0000 0101b (05H)	0	0	0	1 <sup>(4)</sup>
WRSR	0000 0001b (01H)	0	0	1	0
READ	0000 0011b (03H)	2	3	0	1 至 ∞ <sup>(4)</sup>
WRITE	0000 0010b (02H)	2	N/A <sup>(1)</sup>	1 至 64	0
字节编程	0000 0010b (02H)	N/A <sup>(1)</sup>	3	1	0
AAI 字编程	1010 1101b (ADH)	N/A <sup>(1)</sup>	3 <sup>(2)</sup>	2	0
AAI 编程	1010 1111b (AFH)	N/A <sup>(1)</sup>	3 <sup>(3)</sup>	1	0
4 KB 扇区擦除	0010 0000b (20H)	N/A <sup>(1)</sup>	3	0	0
32 KB 块擦除	0101 0010b (52H)	N/A <sup>(1)</sup>	3	0	0
64 KB 块擦除	1101 1000b (D8H)	N/A <sup>(1)</sup>	3 <sup>(2)</sup>	0	0
整片擦除	0110 0000b (60H) 或 1100 0111b (C7H)	N/A <sup>(1)</sup>	0	0	0
RDID	1001 0000b (90H) 或 1010 1011b (ABH)	N/A <sup>(1)</sup>	3	0	1 至 ∞ <sup>(4)</sup>

注:

- 命令不适用于此器件。
- 命令适用于 SST25VF040B/080B 器件。
- 命令适用于 SST25VF010A/020 器件。
- SST25VFxxxx 器件的数据阶段长度 (字节) 为 1 至 ∞。在被 SST25VFxxxx 器件 CE# 引脚上低电平到高电平的跳变终止之前, 将一直随着时钟周期连续读取数据。



### 1.3 SPI 外设处理

要访问串行存储器，需要使用同步串行通信接口，即 SPI 接口。

在每个 SPI 时钟周期内，主器件在 MOSI 线上发送一个位，从器件读取该位；从器件在 MISO 线上发送一个位，主器件读取该位。即使仅打算进行单向数据传输，也会保持这种序列。

SPI 按字节操作。发送或接收一个字节后，INTFLAGS 寄存器中的标志 SPIF 置 1。

SPI 外设操作支持以下两种方法：

1. 轮询 INTFLAGS 寄存器以检测 SPIF 是否已置 1，然后再执行下一次 SPI 传输。
2. 允许 SPI 中断并在 SPI 中断处理程序中管理下一次 SPI 传输。

**注：**本应用笔记提供的源代码中使用的是 SPI 外设操作的轮询方法。

### 1.4 繁忙检测

25xx256 串行存储器件通过其状态寄存器中的 WIP（写进行）位指示其当前状态。

SST25VFxxx 串行存储器件通过其状态寄存器中的 BUSY 位指示繁忙状态。当该位置 1 时，表示正在进行写操作；置 0 时，表示未在进行写操作。该位是只读位。

## 2. 从 Atmel | START 获取源代码

示例代码可通过 Atmel | START 获得，Atmel | START 是一种基于 Web 的工具，可通过图形用户界面（Graphical User Interface, GUI）配置应用程序代码。可以通过下面提供的直接示例代码链接或 Atmel | START 首页上的 *BROWSE EXAMPLES*（浏览示例）按钮，为 Atmel Studio 和 IAR Embedded Workbench® 下载代码。

Atmel | START 网页：<http://start.atmel.com/>

### 示例代码

- SPI DataFlash:
  - [http://start.atmel.com/#example/Atmel:spi\\_serial\\_memory:1.0.0::Application:SPI\\_DataFlash:](http://start.atmel.com/#example/Atmel:spi_serial_memory:1.0.0::Application:SPI_DataFlash:)
- SPI EEPROM:
  - [http://start.atmel.com/#example/Atmel:spi\\_serial\\_memory:1.0.0::Application:SPI\\_EEPROM:](http://start.atmel.com/#example/Atmel:spi_serial_memory:1.0.0::Application:SPI_EEPROM:)

有关详细信息和示例项目的相关信息，请按 Atmel | START 中的 *User guide*（用户指南）按钮。*User guide* 按钮可以在该网页中找到，方法是在 Atmel | START 项目配置器中的仪表板视图中单击项目名称。

### Atmel Studio

在 Atmel | START 的网页中单击 *DOWNLOAD SELECTED EXAMPLE*（下载所选示例），为 Atmel Studio 下载 .atzip 文件形式的代码。要从 Atmel | START 下载文件，请单击 *EXPORT PROJECT*（导出项目），然后单击 *DOWNLOAD PACK*（下载包）。

双击下载的 .atzip 文件，项目将导入到 Atmel Studio 7.0。

### IAR Embedded Workbench

有关如何在 IAR Embedded Workbench 中导入项目的信息，请打开 Atmel | START 用户指南，选择 *Using Atmel Start Output in External Tools*（使用外部工具中的 Atmel Start 输出），然后选择 *IAR Embedded Workbench*。单击 Atmel | START 首页右上角的 *About*（关于）或项目配置器中右上角的 *Help And Support*（帮助和支持），均可找到 Atmel | START 用户指南的链接。

## 3. 源代码概述

以下各节将介绍外设、SPI 接口的初始化、轮询模式函数以及 DataFlash/EEPROM 读/编程函数。

外设：

- CPU 时钟：（默认）3.33 MHz。
- 使用的外设：
  - SPI
    - PA1: MOSI
    - PA2: MISO
    - PA3: SCK
    - PA4: SS

在 Atmel | START 中配置的项目将生成外设驱动程序函数和文件，以及初始化所有驱动程序的主函数。

- 外设驱动程序的头文件和源文件分别位于 *include* 和 *src* 文件夹中。
- 在 *atmel\_start.c* 文件中，函数 `atmel_start_init()` 初始化项目中的 MCU、外设驱动程序和中间件。
- DataFlash 驱动程序文件：*dataflash.c* 和 *dataflash.h*
- EEPROM 驱动程序文件：*eeprom.c* 和 *eeprom.h*

### 3.1 SPI 接口的初始化

函数 `dataflash_init()`，在 DataFlash 驱动程序中初始化 SPI。

函数 `eeprom_init()`，在 EEPROM 驱动程序中初始化 SPI。

1. 按如下所述配置端口方向：
  - 1.1. 输出：PA3/SCK、PA1/MOSI 和 PA4/SS
  - 1.2. 输入：PA2/MISO。
2. 通过配置 CTRLA 寄存器，将 SPI 接口配置为主模式。
3. 对于 SPI 时钟速度，使用的时钟预分频比为 DIV4，并使用 CTRLA 寄存器中的 CLK2X 位将时钟速度加倍。
4. SPI 传输模式选择为模式 0，通过将 SSD 位配置为高电平禁止从选择。将 SSD 位配置为高电平时，#SS 不会禁止主模式。
5. 将从选择线配置为高电平。

### 3.2 轮询操作模式函数

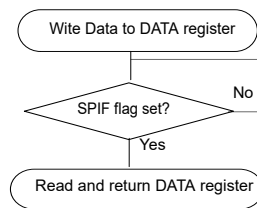
#### SPI 传输基本函数

函数 `spi()` 是在轮询模式下处理 SPI 接口的基本函数。该函数用于发送一个字节并存储收到的字节。请注意，该函数不控制片选线的电平。`spi()` 通过 `tx_spi()` 和 `rx_spi()` 调用。函数 `tx_spi()` 用于通过 SPI 写入一个字节，`rx_spi()` 用于接收字节。

原型：

```
char spi(uint8_t data);
```

图 3-1. SPI 传输函数流程图



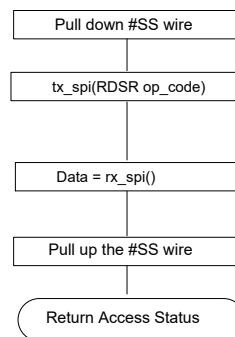
### 读状态寄存器函数

函数 `dataflash_read_status_register()` 和 `EEPROM_read_status_register()` 用于执行 RDSR 访问。其返回值为访问状态值。

原型:

```
uint8_t dataflash_read_status_register(void);
uint8_t EEPROM_read_status_register(void);
```

图 3-2. 读状态寄存器函数流程图



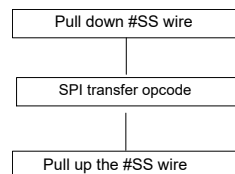
### 单字节写命令

函数 `send_cmd()` 用于发送单字节操作码，例如 `WREN` 和 `WRDI` 等。执行访问前，串行存储器必须就绪。

原型:

```
void send_cmd(uint8_t opcode)
```

图 3-3. 发送命令流程图



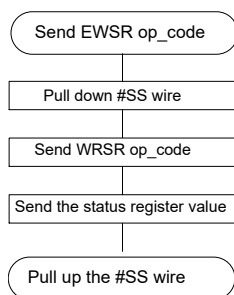
### 写状态寄存器函数

该函数用于执行 RDSR 访问。它发送操作码 `OP_EWSR`（使能写状态寄存器）和 `OP_WRSR`（写状态寄存器），然后将要写入的数据发送到状态寄存器。

原型:

```
void dataflash_write_status_register(uint8_t byte);
```

图 3-4. 写状态寄存器函数流程图



### 3.3 DataFlash 读和编程函数

DataFlash 函数在示例源代码中的 `dataflash.c` 和 `dataflash.h` 中实现和定义。

#### DataFlash 单字节读函数:

函数 `dataflash_read_byte()` 用于从 DataFlash 中的给定地址单元读取单个字节。

原型:

```
uint8_t dataflash_read_byte(uint32_t addr);
```

#### DataFlash 多字节读函数:

以下三个函数用于读取多个字节。这些函数通过示例源代码 `main.c` 中的 `read_dataflash()` 调用。

原型:

```
void dataflash_read_multiple_start(uint32_t addr);
void dataflash_read_multiple_continue(uint8_t *buff, uint8_t buff_length);
void dataflash_read_multiple_stop();
```

`dataflash_read_multiple_start()` 用于下拉#SS 线、发送读操作码以及发送起始地址。

`dataflash_read_multiple_continue()` 用于连续读取数据，直至读取的字节数达到给定的 `buff_length`。数据存储在给定的缓冲区中。

`dataflash_read_multiple_stop()` 用于上拉#SS 线，以停止读过程。

#### DataFlash 单字节编程函数:

函数 `dataflash_program_byte()` 用于将单个字节编程到给定地址。

原型:

```
void dataflash_program_byte(uint32_t addr, uint8_t byte);
```

#### DataFlash 多字节编程函数:

以下三个函数用于编程多个字节。这些函数通过示例源代码 `main.c` 中的 `write_to_dataflash()` 调用。

原型:

```
void dataflash_program_multiple_start(uint32_t addr);  
uint8_t dataflash_program_multiple_continue(uint8_t *buff, uint8_t length);  
void dataflash_program_multiple_stop();
```

编程多个字节时使用 **AAI** 编程指令，该指令允许对多个字节的数据进行编程，无需再次发出下一连续地址单元。

`dataflash_program_multiple_start()` 调用 `AAI_program_start()` 函数。该函数会发送 **EBSY** 和 **WREN** 操作码、下拉 **#SS** 线并发送 **OP\_AAI\_WORD\_PROG** 操作码，然后发送开始读取数据的起始地址。

`dataflash_program_multiple_continue()` 调用 `AAI_program_continue()` 来写入字或字节，具体取决于所选的器件。对于器件 **SST25VF040B/SST25VF080B**，编程一个字；而对于器件 **SST25VF010A/SST25VF020**，则编程一个字节。在写入的字节数达到给定长度之前，将一直调用函数 `AAI_program_continue()`。最后一个字节完成编程后，该函数返回 **0**。

`dataflash_program_multiple_stop()` 函数调用 `AAI_program_stop()`。该函数会发送停止序列：发送 **OP\_WRD1** 和 **OP\_DBSY** 操作码。

**注：** 有关操作码的说明，请参见 `dataflash.h` 或器件数据手册。

#### DataFlash 读器件 ID 和制造商 ID 函数:

原型:

```
uint8_t dataflash_read_id(uint32_t addr); /*使用地址 ADD_DEV_ID (0x01) 读取器件 ID */  
uint32_t dataflash_jedec_id_read(); /*从地址 0x00 处读取制造商 ID*/
```

#### DataFlash 擦除函数:

原型:

```
void dataflash_erase_chip(void); /*擦除整个芯片*/  
void dataflash_erase_sector_4k(uint32_t addr); /*擦除 4K 扇区*/  
void dataflash_erase_block_32k(uint32_t addr); /*擦除 32K 扇区*/  
void dataflash_erase_block_64k(uint32_t addr); /*擦除 64K 扇区*/
```

## 3.4 EEPROM 读和编程函数

EEPROM 读和编程函数在 SPI EEPROM 示例源代码中的 `eeprom.c` 和 `eeprom.h` 中实现和定义。

#### EEPROM 单字节读函数:

函数 `eeprom_read_byte()` 用于从 EEPROM 中的给定地址单元读取单个字节。

原型:

```
uint8_t eeprom_read_byte(uint16_t addr);
```

#### EEPROM 多字节读函数:

函数 `eeprom_program_multiple_read()` 用于从给定地址读取多个字节（最多等于 `buff_length`）并将相应字节存储到给定缓冲区中。

原型:

```
void eeprom_program_multiple_read(uint16_t addr, uint8_t *buff, uint8_t buff_length);
```

该函数中的序列为：下拉#SS 线、发送读操作码以及发送起始地址。然后连续读取数据，直至读取的字节数达到给定的 `buff_length`。数据存储在给定缓冲区中，随后上拉#SS 线以停止读过程。

### EEPROM 单字节编程函数：

函数 `eeprom_program_byte` 用于将单格字节编程到给定地址。

原型：

```
void eeprom_program_byte(uint16_t addr, uint8_t byte);
```

### EEPROM 多字节编程函数：

该函数用于从给定数组写入多个字节（最多等于 `buff_length`）。该函数从给定的起始地址开始写入。在该函数中，将对地址和 `buff_length` 进行验证。起始地址应为页地址的开头，`buff_length` 应在限制范围内。如果验证失败，则该函数将返回而不写入任何内容。

如果地址和 `buff_length` 有效，则计算需要写入的总页数并调用 `eeprom_page_write()` 函数，该函数用于向页（最大为 `PAGE_SIZE`，即 64 字节）写入数据。

原型：

```
void eeprom_program_multiple_write(uint16_t addr, uint8_t *buff, uint8_t buff_length);  
void eeprom_page_write(uint16_t addr, uint8_t *buff, uint8_t buff_length);
```

### EEPROM 擦除函数：

原型：

```
void eeprom_erase_chip(); /*擦除整个芯片*/  
void eeprom_erase_page(uint16_t addr); /*擦除页，地址应为页地址的开头*/
```

---

---

## 4. 开发环境

随附的驱动程序代码使用以下硬件和软件进行开发：

### 硬件

- ATtiny817 Xplained pro 板
- 实验电路板上的 8 引脚 SOIC 插座
- 焊接的 SOIC 封装 AT25AA256/AT25LC256
- 焊接的 SOIC 封装 SST25VF010A
- 焊接的 SOIC 封装 SST25VF020
- 焊接的 SOIC 封装 SST25VF040B
- 焊接的 SOIC 封装 SST25VF080B

### 软件

- Atmel Studio 7



## 5. 版本历史

文档版本	日期	备注
DS00002665A	3/2018	本文档使用 Microchip 的 DS 编号取代 Atmel 2595C
2595C	11/2017	更新为使用串行存储器: AT25xx256/SST25VFxxxx
2595B	07/2016	新模板
2595A	03/2005	文档初始版本

---

## Microchip 网站

---

Microchip 网站 <http://www.microchip.com/> 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问，网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题（FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

---

## 变更通知客户服务

---

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 <http://www.microchip.com/>。在“支持”（Support）下，点击“变更通知客户”（Customer Change Notification）服务后按照注册说明完成注册。

---

## 客户支持

---

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师（FAE）
- 技术支持

客户应联系其代理商、代表或应用工程师（FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过以下网站获得技术支持：<http://www.microchip.com/support>

---

## Microchip 器件代码保护功能

---

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿意与关心代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

---

## 法律声明

---

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，否则在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzzer、PackerTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、PrecisionEdge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQL、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2019, Microchip Technology Incorporated 版权所有。

ISBN: 978-1-5224-4786-3

## 质量管理体系

有关 Microchip 质量管理体系的更多信息，请访问 [www.microchip.com/quality](http://www.microchip.com/quality)。

## 全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
<b>公司总部</b> 2355 West Chandler Blvd. 钱德勒, 亚利桑那州 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: <a href="http://www.microchip.com/support">http://www.microchip.com/support</a> 网址: <a href="http://www.microchip.com">www.microchip.com</a>	<b>澳大利亚 - 悉尼</b> 电话: 61-2-9868-6733 <b>中国 - 北京</b> 电话: 86-10-8569-7000 <b>中国 - 成都</b> 电话: 86-28-8665-5511 <b>中国 - 重庆</b> 电话: 86-23-8980-9588 <b>中国 - 东莞</b> 电话: 86-769-8702-9880 <b>中国 - 广州</b> 电话: 86-20-8755-8029 <b>中国 - 杭州</b> 电话: 86-571-8792-8115 <b>中国 - 香港特别行政区</b> 电话: 852-2943-5100 <b>中国 - 南京</b> 电话: 86-25-8473-2460 <b>中国 - 青岛</b> 电话: 86-532-8502-7355 <b>中国 - 上海</b> 电话: 86-21-3326-8000 <b>中国 - 沈阳</b> 电话: 86-24-2334-2829 <b>中国 - 深圳</b> 电话: 86-755-8864-2200 <b>中国 - 苏州</b> 电话: 86-186-6233-1526 <b>中国 - 武汉</b> 电话: 86-27-5980-5300 <b>中国 - 西安</b> 电话: 86-29-8833-7252 <b>中国 - 厦门</b> 电话: 86-592-2388138 <b>中国 - 珠海</b> 电话: 86-756-3210040	<b>印度 - 班加罗尔</b> 电话: 91-80-3090-4444 <b>印度 - 新德里</b> 电话: 91-11-4160-8631 <b>印度 - 浦那</b> 电话: 91-20-4121-0141 <b>日本 - 大阪</b> 电话: 81-6-6152-7160 <b>日本 - 东京</b> 电话: 81-3-6880-3770 <b>韩国 - 大邱</b> 电话: 82-53-744-4301 <b>韩国 - 首尔</b> 电话: 82-2-554-7200 <b>马来西亚 - 吉隆坡</b> 电话: 60-3-7651-7906 <b>马来西亚 - 槟榔屿</b> 电话: 60-4-227-8870 <b>菲律宾 - 马尼拉</b> 电话: 63-2-634-9065 <b>新加坡</b> 电话: 65-6334-8870 <b>台湾地区 - 新竹</b> 电话: 886-3-577-8366 <b>台湾地区 - 高雄</b> 电话: 886-7-213-7830 <b>台湾地区 - 台北</b> 电话: 886-2-2508-8600 <b>泰国 - 曼谷</b> 电话: 66-2-694-1351 <b>越南 - 胡志明市</b> 电话: 84-28-5448-2100	<b>奥地利 - 韦尔斯</b> 电话: 43-7242-2244-39 传真: 43-7242-2244-393 <b>丹麦 - 哥本哈根</b> 电话: 45-4450-2828 传真: 45-4485-2829 <b>芬兰 - 埃斯波</b> 电话: 358-9-4520-820 <b>法国 - 巴黎</b> 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 <b>德国 - 加兴</b> 电话: 49-8931-9700 <b>德国 - 哈恩</b> 电话: 49-2129-3766400 <b>德国 - 海尔布隆</b> 电话: 49-7131-72400 <b>德国 - 卡尔斯鲁厄</b> 电话: 49-721-625370 <b>德国 - 慕尼黑</b> 电话: 49-89-627-144-0 传真: 49-89-627-144-44 <b>德国 - 罗森海姆</b> 电话: 49-8031-354-560 <b>以色列 - 若那那市</b> 电话: 972-9-744-7705 <b>意大利 - 米兰</b> 电话: 39-0331-742611 传真: 39-0331-466781 <b>意大利 - 帕多瓦</b> 电话: 39-049-7625286 <b>荷兰 - 德卢内市</b> 电话: 31-416-690399 传真: 31-416-690340 <b>挪威 - 特隆赫姆</b> 电话: 47-72884388 <b>波兰 - 华沙</b> 电话: 48-22-3325737 <b>罗马尼亚 - 布加勒斯特</b> 电话: 40-21-407-87-50 <b>西班牙 - 马德里</b> 电话: 34-91-708-08-90 传真: 34-91-708-08-91 <b>瑞典 - 哥德堡</b> 电话: 46-31-704-60-40 <b>瑞典 - 斯德哥尔摩</b> 电话: 46-8-5090-4654 <b>英国 - 沃金厄姆</b> 电话: 44-118-921-5800 传真: 44-118-921-5820
<b>亚特兰大</b> 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 <b>奥斯汀, 德克萨斯州</b> 电话: 512-257-3370 <b>波士顿</b> 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 <b>芝加哥</b> 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 <b>达拉斯</b> 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 <b>底特律</b> 诺维, 密歇根州 电话: 248-848-4000 <b>休斯顿, 德克萨斯州</b> 电话: 281-894-5983 <b>印第安纳波利斯</b> 诺布尔斯维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 <b>洛杉矶</b> 米镇维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 <b>罗利, 北卡罗来纳州</b> 电话: 919-844-7510 <b>纽约, 纽约州</b> 电话: 631-435-6000 <b>圣何塞, 加利福尼亚州</b> 电话: 408-735-9110 电话: 408-436-4270 <b>加拿大 - 多伦多</b> 电话: 905-695-1980 传真: 905-695-2078			