



AVR 单片机 SPI 通信的一种抗干扰方法

西安工程科技学院 郝立中

一般的通信中,硬件抗干扰措施只能尽量减小误码的概率,而不可能绝对消除误码。对于一般个别位的误码,采取适当的辅助措施后,可以不影响实用。然而,如果一次性的干扰使得通信进入不正常状态而无法恢复,那就是严重的问题,不得不特别对待。

在普通单片机的同步串行通信中,从机一方完全依靠主机提供的位同步时钟来工作,没有单独的“群同步”机制。因此一旦时钟信号线上出现干扰,有可能使从机的位计数发生差错。结果是从机一方的字节界限和主机一方发生错位。这种错位会一直持续下去,无法恢复,造成恶性后果。大多数的应用程序中,数据传输中间的空闲时间往往较长,因而在这一段时间中,时钟信号线上受到干扰的可能性也相对较大。还有,如果主机和从机程序不同时开始加电运行,也有可能一开始字节界限就有错位。

本文介绍一种在 AVR 单片机 SPI 主从式通信中较彻底消除字节错位的设计方法。其思想是:通过联络信号实现群同步,而联络信号可以直接利用 AVR 的 SS 引脚。

1 AVR 的 SS 引脚

AVR 单片机 SPI 通信接口有四个引脚:

- MOSI 主机用作数据输出,从机用作数据输入;
- MISO 主机用作数据输入,从机用作数据输出;
- SCK 同步时钟信号;
- SS 从机选择。

作主机用时,SS 引脚可以设成输出方式,这样该引脚可以作为普通并行口的一位(在下文的例中,就是 PB4)而不影响 SPI 的工作。

如果将 SS 引脚设成输入方式,作主机时必须接 1 电位;若输入了 0 电位,就会将该芯片强行改为从机。SPI 的控制寄存器 SPCR 中的 MSTR 位(主从选择位,1 主 0 从)可以由软件设定;但硬件将该芯片强行改为从机时,也会自动修改 MSTR 位。

从机方式时,SS 引脚必须设成输入方式,正常工作必须输入 0 电位。若输入 1 电位,则会封锁从机的 SPI 传输工作。同时,在由 0 变 1 的瞬时,会对 SPI 实行一次

复位。此复位动作不仅会清除控制寄存器 SPCR,同时也会清除位计数。

在封锁期间,如果由软件将 MSTR 位置 1,则会改为主机工作方式;否则,待 SS 引脚输入变为 0 时,可以从头开始按从机方式工作。

2 设计思想

我们将主机和从机的 SS 端相连。主机端将其设为输出,从机端将其设为输入,利用它传送联络信号。(当然,主机不用 SS 端而利用另外一个输出引脚承担这个功能,原则上也可以。)

在 SPI 传输过程中,主机使该引脚保持 0 电位。传输开始前以及数据组间的传输间隙中,主机往该引脚发 1 电位。这样形成的正脉冲,宽度大致覆盖了传输的间隙,其封锁传输的作用可以有效地屏蔽掉大部分干扰信号。而且,即使在传输中已经发生了从机的字节错位,也会因该脉冲使从机的 SPI 复位而消除。

从机方在开始一个数据组的传输前,需要查询 SS 引脚的电位。在该脉冲前沿出现之后,主机开始新的传输之前,重新设置控制寄存器 SPCR,并给 SPDR 写好准备在传输开始时传给主机的数据。

3 实例程序

本例是一个工控设备中的两个模块:主控模块和监视模块。二者的 CPU 都采用 AVR 的 AT90S8515 单片机。

它们之间靠 SPI 通信,MOSI (PB5)、MISO (PB6)、SCK (PB7)、SS (PB4) 四个引脚对应相连。主控模块定时给监视模块发送 8 个字节 1 组的数据。监视模块负责数据显示、与上位机通信等等;同时,将监视模块板上的按钮信息去抖动处理后传给主控模块,与主控送来的每组数据中的第 1 个字节同时传送。

在第一组传输开始前以及各组传输之间,SS 线上都有一个如上所述的正脉冲,作为联络信号。

以下程序中只摘录了与 SPI 有关的段落。

(1) 主控模块(主机) SPI 程序

WORK0 是寄存器区中的一个暂存单元(实际使用



R26), REPORT 是 SRAM 区 8 个字节缓存空间的首地址。利用 Z 指针指示 REPORT 区中的数据。

KEY 单元也是一个寄存器 (实际使用 R2), 存放收到的按钮信息。

每组 8 个字节的第 1 个由定时中断服务程序发送, 其余由 SPI 完成中断处理。注意在撤销 SS 的 1 电位到开始传输前必须留有必要时间间隔 (程序中标 *** 记号处)。

主控 SPI 初始化

```
SER  WORK0          ;输入上拉有效, 输出初始为 1
OUT  PORTB,WORK0
LDI  WORK0,0B10110000  β 口, 除 SS、MOSI、SCK 外
                          ;全为输入
OUT  DDRB,WORK0      β 口
IN   WORK0,SPSR      清 SPI 状态中残留的中断挂号
IN   WORK0,SPDR
LDI  WORK0,0B11010011  开 SPI, 8MHz 主频下 62.5K 波特
OUT  SPCR,WORK0      ;本模块作为主机, SS 已经置 1
.....
```

主控定时中断服务程序

开始数据组传送, 输出第 1 个字节。

```
TRNS0: .....          ;保护现场
      CBI  PORTB,4  往 SS 送 0, 撤销联络信号
      .....          ;(***)准备 REPORT 输出报告数据;
      LDI  ZL,LOW(REPORT);令 Z 指针  REPORT;
      CLR  ZH          ;本例中 SRAM 区较小, 故 ZH 固定为 0
      LD   WORK0,Z+
      OUT  SPDR,WORK0;Z 指针所指  SPI 输出;Z 指针增 1;
      .....          ;恢复现场
      RETI
```

主控 SPI 完成中断服务程序

接收传来的按钮信息, 输出后续各字节。

```
TRNS1:.....          ;保护现场
      ;IF Z 指针  REPORT+8 THEN [封锁从机, 中断返回];
      CPI  ZL,LOW(REPORT+8)
      BRCC TRNSZ
      ;IF Z 指针 =REPORT+1 THEN 读 SPI 输入  KEY;
      CPI  ZL,LOW(REPORT+1)
      BRNE TRNSY
      IN  KEY,SPDR
TRNSY: LD  WORK0,Z+
      OUT SPDR,WORK0 ;Z 指针所指  SPI 输出;
                          ;Z 指针增 1;
      RJMP TRNSX
      ;封锁从机 ( 监视模块 ) SPI
TRNSZ: SBIP  ORTB,4      开始联络信号, 封锁从机
TRNSX: .....          ;恢复现场
      RETI
```

(2) 监视模块 (从机) SPI 程序

以下程序中 TEMP0 是寄存器区中的一个暂存单元 (实际使用 R18), REPORT 是 SRAM 区 8 个字节缓存空间的首地址。利用 Y 指针指示 REPORT 区中的数据。

KEY 单元也是一个寄存器 (实际使用 R1), 存放已经去抖动处理过准备发送的按钮信息。

每个字节均由 SPI 完成中断来接收。在接收完一组开始准备下一组时, 将按钮信息送入 SPDR, 以备发送。

从机 SPI 初始化

```
SER  TEMP0          ;输入上拉有效输出初始为 1
OUT  PORTB,TEMP0
LDI  TEMP0,0B01000000  β 口,除 MISO 外全为输入
OUT  DDRB,TEMP0      β 口
LDI  YL,REPORT      预备第 1 次的接收
CLR  YH
IN   TEMP0,SPSR      清 SPI 状态中的中断挂号
IN   TEMP0,SPDR
HERE: SBIS  PINB,4      查询联络信号
      RJMP  HERE
      LDI  TEMP0,0B11000011  开 SPI,可中断,从机
      OUT  SPCR,TEMP0
      OUT  SPDR,KEY      预备第 1 次的发送
      .....
```

从机 SPI 完成中断服务程序

```
TRANS: .....          ;保护现场
      SEI          ;为更高优先的操作而开中断
      IN  TEMP0,SPDR  接收数据
      ST  Y+,TEMP0
      CPI  YL,REPORT+8 ;一组完了吗?
      BRCS TRNSX
      .....          ;处理 REPORT 中收到的数据组
                          ;(&&&)
TRNS1: SBIC  PINB,4      查询联络信号
      RJMP  TRNSY
      SEI
      NOP
      CLI
      RJMP  TRNS1
TRNSY: LDI  TEM P0,0B11000011 开 SPI,可中断从机
      OUT  SPCR,TEMP0
      OUT  SPDR,KEY      准备下一次输出
                          ;(&&&)
                          ;重新初始化 Y 指针
      LDI  YL,REPORT
TRNSX: .....          ;恢复现场
      RETI
```

注意程序中两个标 &&& 记号处之间的一段处理。假如没有更高优先级的操作, 这一段本可以和



初始化程序中一样简单。但因为系统中还有优先级更高的操作，因而需要开中断。而在查询到SS高电位后，直到重新设置控制寄存器SPCR并给SPDR写好数据，持续时间又不能太长，必须小于上文标***处留出的时间（否则，极端情况下，如果查询到高电位时已经接近后沿，就可能发生数据冲突）。因此这一段采用了较复杂的开关中断设计。

4 对比试验结果

原来的程序没有采用上述方法（从机SS直接接0）。

已经调试成功以后，人为加大干扰进行试验：将SPI传输线的屏蔽去掉，有意在附近制造强电的火花等。结果出现运行混乱，由监视模块传来的按钮信息发生了字节错位。改用现方法后，同样的强干扰试验，再没有出现程序运行混乱。

参考文献

1 8-bit AVR Microcontroller with 4K/8K Bytes In-System Programmable Flash AT90S4414/8515. www.atmel.com

2 AVR Instruction set. www.atmel.com

（收稿日期：2003-08-06）