

TDC-GP22时间数字芯片关键编程算法研究与实现

陈 玉

(山西大学商务学院信息中心, 山西 太原 030031)

摘 要: TDC-GP22芯片是一款皮秒级高精度时间数字转换芯片, 由于TDC-GP22内部寄存器种类繁多, 所以芯片编程操作相对较复杂。为此根据TDC-GP22的SPI接口特点, 并结合AVR单片机主机优势, 用高级语言编制符合SPI接口协议程序, 实现对TDC-GP22的两种主要操作, 一种是按照SPI协议对芯片的寄存器进行读取数据操作, 另一种是根据SPI协议数据循环特点对芯片寄存器进行写入数据操作。TDC-GP22实验测试结果表明所编制程序运行完全稳定可行。

关键词: 芯片; 高级语言; 单片机; 接口

中图分类号: TP31 **文献标识码:** A

Research and Implementation of Key Programming Algorithm for the TDC-GP22 Time Digital Chip

CHEN Yu

(Information Center, Business College of Shanxi University, Taiyuan 030031, China)

Abstract: The TDC-GP22 chip is a picosecond level high precision time-to-digital conversion chip, the programming of which is relatively complex because of the wide variety of internal registers in TDC-GP22. Therefore, based on the features of TDC-GP22's SPI interface, combined with the advantages of AVR microcontrollers, SPI interface protocol program is compiled with high-level language to implement the two main operations of TDC-GP22, one is to read data operations on the chip registers according to the SPI protocol, the other is to write data to the chip registers according to the data cycle characteristics of the SPI protocol. TDC-GP22 experimental results show that the program is stable and feasible.

Keywords: chip; high-level language; microcontrollers; interface

1 引言(Introduction)

TDC-GP22芯片是德国ACAM公司推出的一款高分辨率时间数字转换芯片, 高分辨率时间差计算使得TDC-GP22可以用于对时间精度要求严格的场合, 如激光测距、电缆长度测量及超声波流量计等领域。此芯片可以通过单片机读写其内部寄存器来完成时间数字转换, 由于该芯片官方文档资料比较零散, 在芯片的开发和使用上带来很多障碍, 软件程序难以调试, 和单片机难以建立通信。同时TDC-GP22芯片是通过SPI(Serial Peripheral Interface)协议接口和主机建立通信, 所以芯片编程上的主机程序需要严格遵循SPI接口协议来对TDC-GP22内部寄存器进行正确读写, 从而完成TDC-GP22的时间数字转换过程。

2 TDC-GP22的SPI接口协议(SPI interface protocol of TDC-GP22)

2.1 SPI通信协议接口简介

SPI的中文名为串行外围接口, 该协议接口是由摩托罗拉公司提出的通信接口技术; SPI是一种高速、全双工、同步串行总线, SPI接口只占用可编程芯片的四个脚位, 这四

个脚位分别为主机输入从机输出(MISO)、主机输出从机输入(MOSI)、从机选择线(SSN)及串行时钟(SCK)。SPI协议接口的优点主要表现为信号连接线少、全双工通信、协议相对简单及数据传输速率较高。SPI协议接口主要应用在一些FLASH型可编程芯片, 如模数转换器、信号处理芯片和各类单片机等。

2.2 SPI通信协议接口工作原理

SPI通信双方为主从方式, 既SPI通信双方一方为主设备另一方为从设备, SPI通信网络中可以有多个从机, 但是主机只能为一个, 主机通过控制从机的SSN片选端来和相应的从机建立数据通信; SPI通信双方中的主机和从机的物理连接只要相应功能引脚对应连接即可, 既主机MISO连接从机MISO, 其他三类引脚也以此互相连接。SPI中的主从机的双方移位寄存器连接成一个环形结构, 主从机的数据通信通过在主机时钟的控制下进行每次两个8bit数据的移出和移入, 数据传输时高位在前低位在后。

2.3 TDC-GP22的SPI

TDC-GP22采用4线制SPI接口, SSN需要强制置高电

平,在每次读写序列之间SSN保持高电平至少要50ns;TDC-GP22只支持如下SPI模式,时钟极性位(Clock Polarity Bit)为0和时钟相位(Clock Phase Bit)为1。

3 TDC-GP22的SPI主机选择(SPI host selection of TDC-GP22)

SPI主机的选择最好选择引脚功能具有SPI功能的各类单片机,当然对于没有SPI引脚功能的单片机也可以使用,只不过这类单片机需要在软件程序中进行额外的SPI功能模拟,相对给开发和使用上带来一些难度。为了简化单片机软件程序和SPI通信中主从机的硬件电路,此处选择ATMEL公司的ATmega128a低功耗单片机^[1];ATmega128a单片机自带SPI接口,该单片机为高性能低功耗8位单片机,工作于16MHz时性能高达16MIPS,ATmega128a自带128K可编程Flash存储,这样方便使用C语言进行大容量单片机编程。ATmega128a可在3.3V电压下稳定工作,TDC-GP22正常工作电压也为3.3V^[2],这样两者SPI引脚就可以直接进行连接。

4 TDC-GP22和AVR单片机的SPI通信算法(SPI communication algorithm between TDC-GP22 and AVR)

AVR单片机和TDC-GP22时间数字芯片之间以SPI方式通信,其中AVR单片机为主机,TDC-GP22为从机。

4.1 TDC-GP22算法前导

TDC-GP22有七个32位的配置寄存器,每个寄存器的24位是用作配置,是只可以写入的,这七个寄存器用于配置TDC-GP22的相应操作。同时TDC-GP22还有相应可读状态的结果寄存器和状态寄存器^[3,4],TDC-GP22的测量结果和测量状态都可以通过发送操作码0xBX从可读寄存器中读取。

4.2 TDC-GP22算法详解

TDC-GP22在正常运行前,必须先对其七个配置寄存器进行相应的模式配置,七个配置寄存器的配置是相互联系制约的,在寄存器参数的配置上不能有任何冲突,否则TDC-GP22会出现运行异常的状况。下面代码主要是七个配置寄存器的配置示例。

```
gp22_wr_config_reg(0x80,0x00242000); //最大测量
时间2xTref=2 μs
gp22_wr_config_reg(0x81,0x19490000); //在测量模
式1下Stop CH2 Stop CH1
gp22_wr_config_reg(0x82,0xE0000000); //通过
Timeout给出中断
gp22_wr_config_reg(0x83,0x00000000);
gp22_wr_config_reg(0x84,0x20000000);
gp22_wr_config_reg(0x85,0x10000000); //EN_
STARTNOISE=1,switch on
gp22_wr_config_reg(0x86,0x00000000);
gp22_send_1byte(0x70); //初始化
PORTF|=(1<<2); PORTF&=~(1<<2);
//给start引脚一个虚拟开始信号
```

上述代码中为了测试stop2-stop1引脚时间差,所以先给start引脚一个虚拟开始信号,然后TDC-GP22的stop2和stop1引脚才开始工作。

下面示例函数gp22_send_1byte(unsigned char gp22_opcode_byte)可以按照SPI协议向TDC-GP22寄存器写入相应数据。

```
void gp22_send_1byte(unsigned char gp22_opcode_
byte)
{
PORTB&=~(1<<0); //TDC-GP22的SS位置低
SPDR=gp22_opcode_byte;
while(!(SPSR&(1<<SPIF))){} //等待传输结束
delay_us(8);
PORTB|=(1<<0); //将SS位置高
}
```

上述函数中主机往SPDR写入数据以移位到从机中去,串行发送结束后,SPIF置位,若此时寄存器SPCR的SPIE和全局中断使能位置位,SPI中断即产生。

下面示例函数gp22_read_1_bytes(unsigned char read_opcode,unsigned char read_addr)为读取TDC-gp22寄存器中八位数据算法;函数中SPDR=0xFF是每次虚写一次寄存器,只有虚写一次才能保证单片机和TDC-gp22二者的寄存器数据循环移位^[5]。

```
unsigned char gp22_read_1_bytes(unsigned char
read_opcode,unsigned char read_addr)
{
unsigned char read_opcode_addr=read_opcode|read_
addr;
PORTB&=~(1<<0); //ss置低
SPDR=read_opcode_addr;
while(!(SPSR&(1<<SPIF))){} //SPIF等于1时发送或
接收数据完成
delay_us(8);
SPDR=0xFF; //虚写一次,推出数据
while(!(SPSR&(1<<SPIF))){}
delay_us(8);
return SPDR; //主机读取数据
PORTB|=(1<<0); //将ss位置高
}
```

如下函数gp22_read_status_bytes()为读取TDC-GP22中状态寄存器算法,算法中用到了循环结构,这样可以保证读取TDC-GP22状态寄存器的多个字节数据^[6]。

```
unsigned int gp22_read_status_bytes()
{
int n;
unsigned int Result_read=0;
unsigned char n_bytes=2;
unsigned char read_opcode_addr=0xB0|0x04;
PORTB&=~(1<<0); //ss位置低
SPDR=read_opcode_addr;
while(!(SPSR&(1<<SPIF))){} //等待传输结束
delay_us(8);
SPDR=0xFF; //虚写一次,推出数据
```

```

while(!(SPSR&(1<<SPIF))){}
delay_us(8);
Result_read=SPDR; //主机读取数据
for(n=1;n<n_bytes;n++){
    SPDR=0xFF; //虚写一次, 推出数据
while(!(SPSR&(1<<SPIF))){}
    delay_us(8);
    Result_read=Result_read<<8;
    Result_read |=SPDR; }
PORTB|=(1<<0); //将ss位置高
return Result_read;
}

```

以下函数gp22_wr_config_reg(unsigned char opcode_address,unsigned long config_reg_data)为写状态寄存器的示例算法, 该算法中按照SPI协议规范, 先写高位字节, 然后再写低位字节。

```

void gp22_wr_config_reg(unsigned char opcode_address,
    unsigned long config_reg_data)
{
    unsigned char Data_Byte_Lo=config_reg_data;
//自动截取低8位进行赋值
    unsigned char Data_Byte_Mid1=config_reg_data>>8;
    unsigned char Data_Byte_Mid2=config_reg_data>>16;
    unsigned char Data_Byte_Hi=config_reg_data>>24;
    PORTB&=~(1<<0); //ss位置低
    SPDR=opcode_address;
    while(!(SPSR&(1<<SPIF))){}
    delay_us(8);
    SPDR=Data_Byte_Hi;
    while(!(SPSR&(1<<SPIF))){};
    delay_us(8);
    SPDR=Data_Byte_Mid2;
    while(!(SPSR&(1<<SPIF))){}
    delay_us(8);
    SPDR=Data_Byte_Mid1;
    while(!(SPSR&(1<<SPIF))){}
    delay_us(8);
    SPDR=Data_Byte_Lo;
    while(!(SPSR&(1<<SPIF))){}
    delay_us(8);
    PORTB|=(1<<0); //将ss位置高
}

```

5 实验测试过程(Experimental testing process)

为了验证上述算法的准确性, 专门用TDC-GP22进行了双绞线长度测试实验; 由于双绞线可看作传输线, 所以测量时向被测双绞线注入一定宽度的入射矩形脉冲^[7], 如果被测双

绞线终端开路就会产生矩形脉冲的正反射, 同时将入射矩形脉冲和反射脉冲分别送入TDC-GP22的stop1、stop2引脚, 通过计算stop2-stop1的时间差并按如下公式即可计算出开路双绞线的长度。

$$L = \frac{1}{2} \times V \times \Delta t$$

其中, L 为被测电缆长度, V 为脉冲信号在电缆中的传播速度, 一般 V 为0.66—0.85倍光速^[8], 约为200m/ μ s, Δt 为入射脉冲与反射脉冲的时间差, 具体测试数据见表1。

表1 测试结果

Tab.1 Test results

实长	时间差	测试长度	测试状态
5m	0.048 μ s	4.8m	开路
10m	0.1 μ s	10m	开路
20m	0.195 μ s	19.5m	开路
35m	0.346 μ s	34.6m	开路
100m	0.97 μ s	97m	开路

表1测试结果显示双绞线测试长度的误差在10%以内, 该误差范围符合实际工程需求, 测试结果也验证了TDC-GP22算法准确稳定。

6 结论(Conclusion)

TDC-GP22的SPI协议接口编程必须严格按照SPI接口数据环状传输的特点进行编程控制, 否则主机和从机难以建立有效的通信。TDC-GP22主机的选择有多种方案, 这里选用了AVR系列单片机, 并用C语言做了编程示范, 当然也可以选用其他厂家其他型号的单片机作为控制主机, 并结合相应的程序语言进行SPI接口编程。同时用测量双绞线长度的实验进一步验证了算法的准确性。

参考文献(References)

- [1] Microchip Technology.ATmega128A datasheet complete[EB/OL].<https://www.microchip.com/wwwproducts/en/ATmega128A>,2016-12-10.
- [2] Acam-messelectronic gmbh.TDC-GP22 ultrasonic flow converter datasheet[EB/OL].<https://ams.com/tdc-gp22>,2014-03-13.
- [3] 莫朗,朱建良,薄煜明.基于TDC-GP22的室内定位系统的设计及实现[J].电子设计工程,2017,25(18):168-177.
- [4] 刘宇杰,李斌,刘振凯,等.基于TDC-GP22的超声波流量计设计[J].工业控制计算机,2015,28(3):166-170.
- [5] 崔永俊,刘阳,杨兵.基于铷原子钟和双TDC-GP22的高精度时间基准测量系统的设计[J].电子器件,2017,40(5):1072-1077.
- [6] 韦乃棋,陈自力,朱安石,等.基于TDC的无人机数据链抗干扰技术研究[J].测控技术,2013,32(3):124-134.
- [7] 王红亮,杨晓.AVR单片机与串行A/D的SPI接口设计[J].电测与仪表,2008,45(9):46-48.
- [8] 尹文芹,施韶华,刘音华.基于FPGA实现TDC的布局布线优化方法研究[J].时间频率学报,2018,41(1):27-36.

作者简介:

陈玉(1975-),男,硕士,实验师.研究领域:计算机应用技术,嵌入式开发.