

基于 I2C 接口芯片的单片机 I/O 扩展的实现

王宏伟

(北京邮电大学信息光子学与光通信研究院, 北京 100876)

摘要: I2C 总线是由 Philips 公司开发的一种两线式串行总线。与并行总线相比, 串行总线有其独特的优点: 电路结构简单, 程序编写方便, 易于实现系统软硬件的模块化、标准化等。随着嵌入式系统的不断发展和应用, 其系统复杂度和集成度越来越高, 对 I/O 端口的需求量也越来越大。本文针对嵌入式系统在实际应用中对 I/O 端口扩展的需要, 提出了基于 I2C 总线接口芯片 PCA9532 对单片机 PIC18F452 的 I/O 端口扩展的具体应用。介绍了 I2C 总线的基本原理, 并给出实际应用中的硬件结构和软件实现。

关键词: I2C 总线; 嵌入式系统; PIC18F452; PCA9532

中图分类号: TP23

The Realization of I/O Extension for Singlechip Based on I2C Interface Chip

WANG Hongwei

(Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: I2C bus, invented by Philips Inc, is a kind of serial bus which is composed of two lines. Compared with parallel bus, serial bus has its special advantages: simple circuit construction, convenient program writing and easy realization of the modularization and standardization of software and hardware system, and so on. With the unceasing development of embedded system and its uses, the complexity and the integration of the system are higher and higher, and the need of I/O ports in quantitatively is also increasing. In this paper, a practical application of extending I/O ports of singlechip, using I2C interface chip PCA9532, is proposed, which is for the purpose of extending embedded system's I/O ports in practical application. The basic principle of I2C interface is introduced, and the realization of the hardware and software system in use is also given.

Key words: I2C Bus; Embedded System; PIC18F452; PCA9532

0 引言

单片机在工业控制、仪器仪表等领域的广泛应用和发展, 使得单片机构成的嵌入式系统在功能和成本上的要求越来越高。然而, 实际应用中经常会遇到单片机 I/O 接口资源不够用的情形, 对于复杂的应用系统, 由于被控对象较多, 如控制多个继电器的开关、多个 LED 数码管的显示等, 端口供需矛盾尤其突出。针对该问题, 本文提出了一种基于 I2C 接口芯片 PCA9532 对单片机 I/O 扩展的实现, 给出了相关原理及软、硬件实现。

1 I2C 接口协议简介

1.1 I2C 总线结构及特点

I2C^[1]总线, 是 Philips 公司研制出来的串行扩展总线, 是“Inter Integrated Circuit”的简称。与其它总线接口相比, 其在硬件结构、组网方式、软件编程上都有很大的不同。

I2C 总线控制系统通常可分为三个部分: 主控器、被控器、数据线和时钟线。

(1) 主控器。在 I2C 总线上起控制作用的微处理器, 对总线上信息传送的对象、方向和

作者简介: 王宏伟(1986-), 男, 硕士研究生, 主要研究方向: 光纤传感. E-mail: wanghongwei198633@163.com

传送的终止时间进行控制。总线在进行数据传输时,由它提供时钟来初始化一次数据的传输。它可工作在主控发送或主控接收状态。

(2)被控器。为了通过 I2C 总线与主控器进行通信,在 I2C 总线上挂接的每一个被控集成电路中,都必须设有一个 I2C 总线接口电路,包括可编程地址发生器、地址码型比较器、移位寄存器和锁存器等,使被控电路具有处理数据的能力,以便接收由主控器发出的控制指令和数据。主控器发送地址时,总线上每一个被控器都将起始信号后的地址和自己的地址进行比较,地址相同的被主控器寻址,根据读/写位,可以是被控发送或者被控接收,这完全取决于主控器。被控器的地址由一个固定部分和一个可编程部分组成。固定部分由生产厂家制定,可编程部分决定该器件可以被接入同一 I2C 总线的最大数目。

(3)数据线和时钟线。数据线与时钟线实际上就是两根线,由微处理器处引出,这就决定了其数据传送方式是串行,这种串行总线虽然没有并行总线的输入、输出能力强,但能使电路之间的连接变得简单,还能有效地减少微处理器的控制引脚数。I2C 是一种双向数据传输系统,主控器可以通过 I2C 总线向被控器发送数据,而被控器也可以通过 I2C 总线向主控器传送数据^[2]。

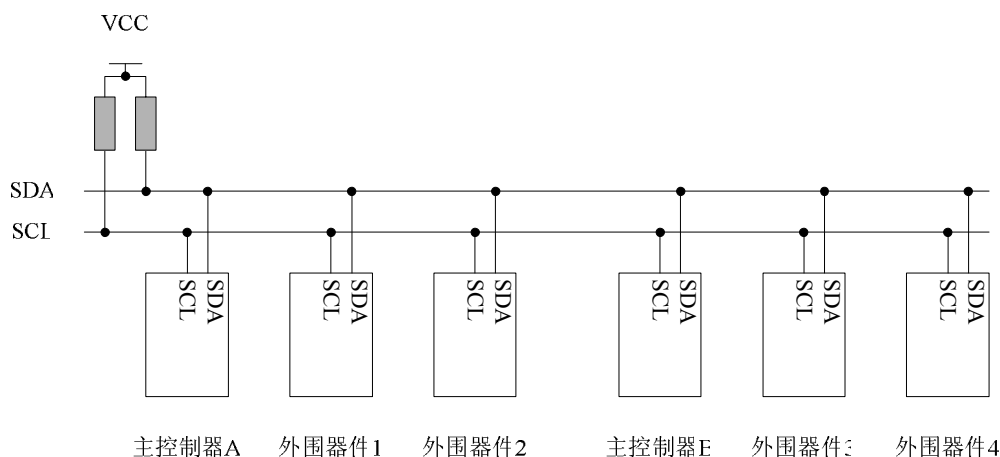


图1 I2C 总线的系统结构

Fig. 1 System Structure of I2C Bus

I2C 总线的其他特点如下:

①I2C 总线所有器件的 SDA、SCL 引脚的输出驱动都为漏极开路或集电极结构,通过外接上拉电阻将总线上所有节点的 SDA、SCL 信号电平实现“线与”的逻辑关系。这不仅可以将多个节点器件按同名端引脚直接挂在 SDA、SCL 线上,还使 I2C 总线具备了“时钟同步”、确保不同工作速度的器件同步工作;

②系统中的所有外围器件都具有一个 7 位的“从器件专用地址码”,其中高 4 位为器件类型地址(由生产厂家制定),低 3 位为器件引脚定义地址(由使用者定义),主控器件通过地址码建立多机通信的机制。因此 I2C 总线省去了外围器件的片选线,这样无论总线上挂接多少器件,其系统仍然为简约的二线结构;

③I2C 总线上的所有器件都具有“自动应答”功能,保证了数据交换的正确性;

④在 I2C 总线系统中可以实现多主控器结构。依靠“总线仲裁”机制确保系统中任何一个主控器都可以掌握总线的控制权。任何一个主控器之间没有优先级,没有中心主机的特权。当多主机竞争总线时,依靠主控器对其 SDA 信号的“线与”逻辑,自动实现“总线仲裁”功能;

⑤I2C 总线系统中的主控器必须是带 CPU 的逻辑模块；而被控器可以是无 CPU 的普通外围器件，也可以是具有 CPU 的逻辑模块。主控器与被控器的区别在于 SCL 的发送权，即对总线的控制权；

⑥I2C 总线的工作速度分为 3 种版本：S（标准模式），速率为 100kb/s；F（快速模式），速率为 400kb/s；Hs（高速模式），速率为 3.4Mb/s。

2 I2C 总线的工作过程

I2C 总线上的所有通信过程都是由主控器引发的。在一次通信过程中，主控器与被控器工作在两个相反的状态，并不断发生互换，即主控器为发送器（主控发送器）时被控器为接收器（被控接收器），主控器为接收器（主控接收器）时被控器为发送器（被控发送器）。

2.1 主控器发送数据过程

主控制器向被控器发送数据的操作过程如下：

①主控器在检测到总线为“空闲状态”（即 SDA、SCL 线均为高电平）时，发送一个启动信号“S”，开始一次通信的开始；

②主控器接着发送一个命令字节。该字节由 7 位的外围器件地址和 1 位读写控制位 R/W 组成（此时 R/W=0）；

③相对应的被控器收到命令字节后向主控器回馈应答信号 ACK（ACK=0）；

④主控器收到被控器的应答信号后开始发送一个字节的的数据；

⑤被控器收到数据后返回一个应答信号 ACK；

⑥主控器收到应答信号后再发送下一个数据……；

⑦当主控器发送最后一个数据字节并收到被控器的 ACK 后，通过向被控器发送一个停止信号 P 结束本次通信并释放总线。被控器收到 P 信号后也退出与主控器之间的通信。

PCA9532 芯片作为数据通信的从属芯片具备一个特点，就是在发送了一个字节地址信息之后，需要再发送一个字节控制信息用来选择后续数据的目的寄存器。因此结合芯片特性及 I2C 接口协议，实际的主控器向被控器发送数据过程如下图所示：

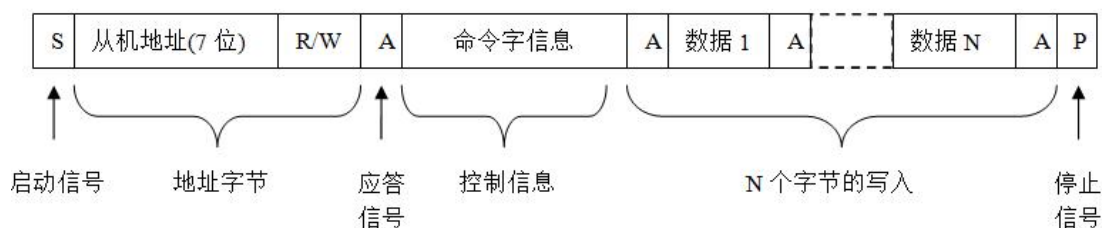


图 2 主控器向被控器发送数据过程

Fig. 2 The Process of Sending Data from Master Controller to Chip Under Control

2.2 主控器接收数据过程

主控器从被控器接收数据的操作过程如下：

①主控器在发送启动信号后，接着发送一个命令字节（此时 R/W=1）；

②对应的被控器收到地址字节后，返回一个应答信号 ACK，并向主控器发送数据；

③主控器收到数据后向被控器回馈一个应答信号 ACK；

④被控器收到应答信号后开始发送下一个字节的数据……；

⑤当主控器完成数据接收后，向被控器发送一个非应答信号 $\overline{\text{ACK}}$ ($\overline{\text{ACK}}=1$)，被控器收到非应答信号后停止发送数据；

⑥主控器发送非应答信号后，发送再发送一个停止信号，释放总线停止通信。

同样结合 PCA9532 芯片特性及 I2C 接口协议，实际的主控器从被控器接收数据过程如下图所示：

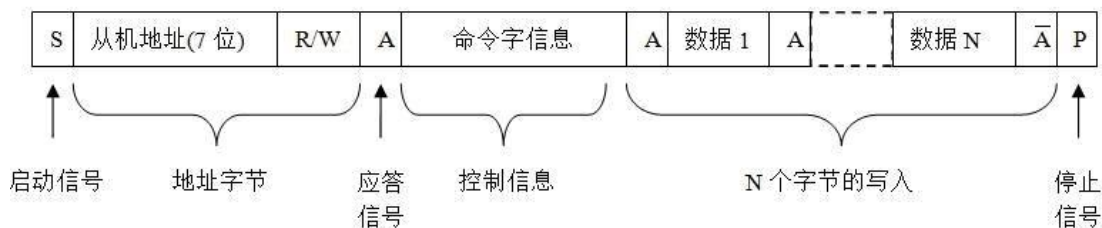


图 3 主控器从被控器接收数据过程

Fig. 3 The Process of Sending Data from Chip Under Control to Master Controller

要说明的是：①主控器通过发送地址码与对应的被控器建立了通信关系，而挂载在总线上的其它被控器虽然同时也收到了地址码，但因为与其自身的地址不相符合，因此提前退出与主控器的通信；②主控器的一次发送通信，其发送的数据数量不受限制。主控器是通过 P 信号通知发送的结束，被控器收到 P 信号后退出本次通信；③主机的每一次发送后都是通过被控器的 ACK 信号了解被控器的接收状况，如果应答错误则重发。

3 I2C 接口模块的实现

3.1 系统硬件连接

本文提出的 I2C 接口的主控器为 PIC18F452^[3]单片机，被控器为 I/O 扩展芯片 PCA9532，用以实现对 LED 指示灯的控制。PCA9532^[4]是 Philips Semiconductors 公司生产的一款 16 位 I2C 总线和 I/O 扩展器。该芯片有三个地址控制端口，16 个可编程控制闪烁率和占空比的 I/O 管脚输出。LED0~LED7 输出管脚的电平高低状态控制是通过向 INPUT0 寄存器写入数据，闪烁周期的控制是通过向 PCS0 寄存器写入数据，占空比的控制是通过向 PWM0 寄存器写入数据。LED8~LED15 输出管脚的电平高低状态控制是通过向 INPUT1 寄存器写入数据，闪烁周期的控制是通过向 PCS1 寄存器写入数据，占空比的控制是通过向 PWM1 寄存器写入数据。芯片硬件连接如图 2 所示^[5]：

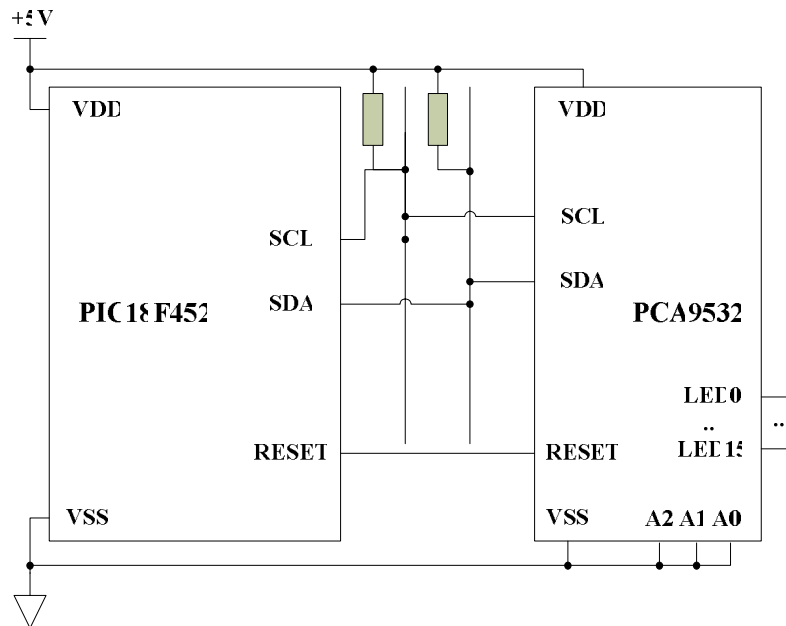


图4 芯片硬件连接

Fig. 4 Hardware Connection of Chips

时钟线 SCL 和数据线 SDA 均接 10K Ω 上拉电阻到 VDD, A2, A1, A0 为 PCA9532 的可编程地址位, LED0~LED15 为漏极开路输出端口, 应用中需接上拉电阻到 VDD。PCA9532 的芯片复位信号接到 PIC 单片机的一个 I/O 输出脚。

3.2 软件实现

基于 Microchip 公司的 MPLAB IDE 和 C18 编译器的开发环境^[6], 主控器 PIC18F452 对 I2C 总线的控制程序如下:

//I2C 读/写操作程序

INT8U i2c_manipulate(INT8U Address_write, INT8U Command, INT8U Data, INT8U Direction)

{

INT8U Address_read = Address_write + 0x01;

I2C_Init(); //初始化操作

i2c_idlecheck(); //总线空闲检测

i2c_start(); //发送起始信号

i2c_send(Address_write); //发送地址信息(最后一位是读/写控制位)

i2c_send(Command); //选择目的寄存器

if(Direction == W) { //写数据

i2c_send(Data); //发送数据

}

else if(Direction == R) { //读数据

SSPCON2bits.RSEN = 1; //重新启动

while(!PIR1bits.SSPIF); //等待重新启动完成

PIR1bits.SSPIF = 0;

i2c_send(Address_read); //选择读地址, 发送读指令

Register = i2c_receive(); //接收被控芯片发来数据

```
}  
i2c_stop();//发送停止信号  
return Register;  
}
```

程序中 Address_write 为被控器地址，默认为主控发送工作方式，即最低位读/写控制位为 0，加 1 后的 Address_read 使工作方式变为主控接收。Command 变量为目的寄存器的地址信息。在实际应用中的数据位只需要一字节，因此将 Data 作为数据字节。Direction 变量用于控制数据传输方向。程序中接收数据的程序如下：

```
INT8U i2c_receive(void)  
{  
    INT8U DATA;  
    SSPCON2bits.RCEN = 1; //使能接收  
    while(!PIR1bits.SSPIF); //等待接收数据  
    PIR1bits.SSPIF = 0;  
    SSPCON2bits.ACKDT = 1; //发送非应答信号  
    SSPCON2bits.ACKEN = 1;  
    while(!PIR1bits.SSPIF);  
    PIR1bits.SSPIF = 0;  
    DATA = SSPBUF; //保存数据  
    while(SSPSTATbits.BF); //等待 SSPBUF 读空  
    return DATA; //返回接收到的数据  
}
```

4 结论

本文给出了使用 I2C 总线接口芯片 PCA9532 扩展单片机 PIC18F452 的 I/O 端口的具体应用。在实际应用中可以挂接多个 PCA9532 芯片，实现对 I/O 端口的高低电平、占空比及闪烁周期进行控制，还可以读出端口的高低电平逻辑值。

[参考文献] (References)

- [1] The I2C Bus Specification (v2.1)[R]. 荷兰：Philips Semiconductors INC, 939839340011, 2000.
- [2] 宋杰, 陈岚, 冯燕. 一种 I2C 总线控制器的接口设计[J]. 信息与电子工程, 2010, 8 (4) : 467~470.
- [3] PIC18F452 Data Sheet[R]. 美国：Microchip Technology INC, DS39564B_CN, 2005.
- [4] PCA9532 Data Sheet[R]. 荷兰：Philips Semiconductors INC, 939775011459, 2003.
- [5] 刘和平, 刘林, 余红欣, 等 (et al). PIC18Fxxx 单片机原理及接口程序设计[M]. 北京：北京航空航天大学出版社, 2004.
- [6] 张明峰. PIC 单片机入门与实战[M]. 北京：北京航空航天大学出版社, 2004.