



第七章 单片机串行通信接口

7.4 通过16C550扩展 串行通信接口





概述

- 随着嵌入式应用越来越复杂，很多通信类项目都有多串口应用的需求，但标准的MCS-51只提供了一个串行口；
- 虽然现在有部分双串口单片机，如台湾华邦(Winbond)的W77E58、Cygnal的C8051F系列单片机中的部分型号等，但昂贵的价格影响了它们的普及。而且在需要使用多于两个串行口的应用场合，双串口单片机也无能为力；
- 16C550为专门的串行口扩展芯片，有多种规格，在一块芯片上可提供1路(16C550)、2路(16C552)或4路(16C554)串行口的扩展，是目前串行口扩展芯片领域事实上的工业标准。很多嵌入式CPU内部集成的串行口都兼容16C550。



7.4.1 16C550简介

- 16C550为TI公司设计生产的串行通信接口芯片，应用非常广泛，目前已成为事实上的工业标准；
- 16C550的主要特性
 - 5V和3.3V的工作电压，工业级温度范围；
 - 最高可支持1Mbps的串行通信速率；
 - 支持完整的硬件流控及Modem控制；
 - 可编程设置多种通信帧格式：
 - 可设置为5~8位数据位，1、1.5、2位停止位；
 - 可设置奇校验、偶校验、Mark、Space、无校验等校验方式；
 - 具有伪起始位检测功能等等...
 - 具有完整的Modem控制信号等等...



7.4.1 16C550简介

- 16C550的主要控制引脚

引脚	功能描述
D0~D7	数据线，和CPU的D0~D7对接
A0~A2	地址线，用于选择16C550的内部寄存器
CS0,CS1, $\overline{CS2}$	片选线，三个信号 都有效 时才选中16C550
\overline{ADS}	地址选通，有效(Low)时方允许片选逻辑
MR	总复位信号，高有效，复位16C550
$\overline{RD1},\overline{RD2}$	读信号， 任一信号有效 均表示CPU读16C550
$\overline{WR1},\overline{WR2}$	写信号， 任一信号有效 均表示CPU写16C550
SIN,SOUT	串行数据输入、输出



7.4.1 16C550简介

- 16C550的主要控制引脚

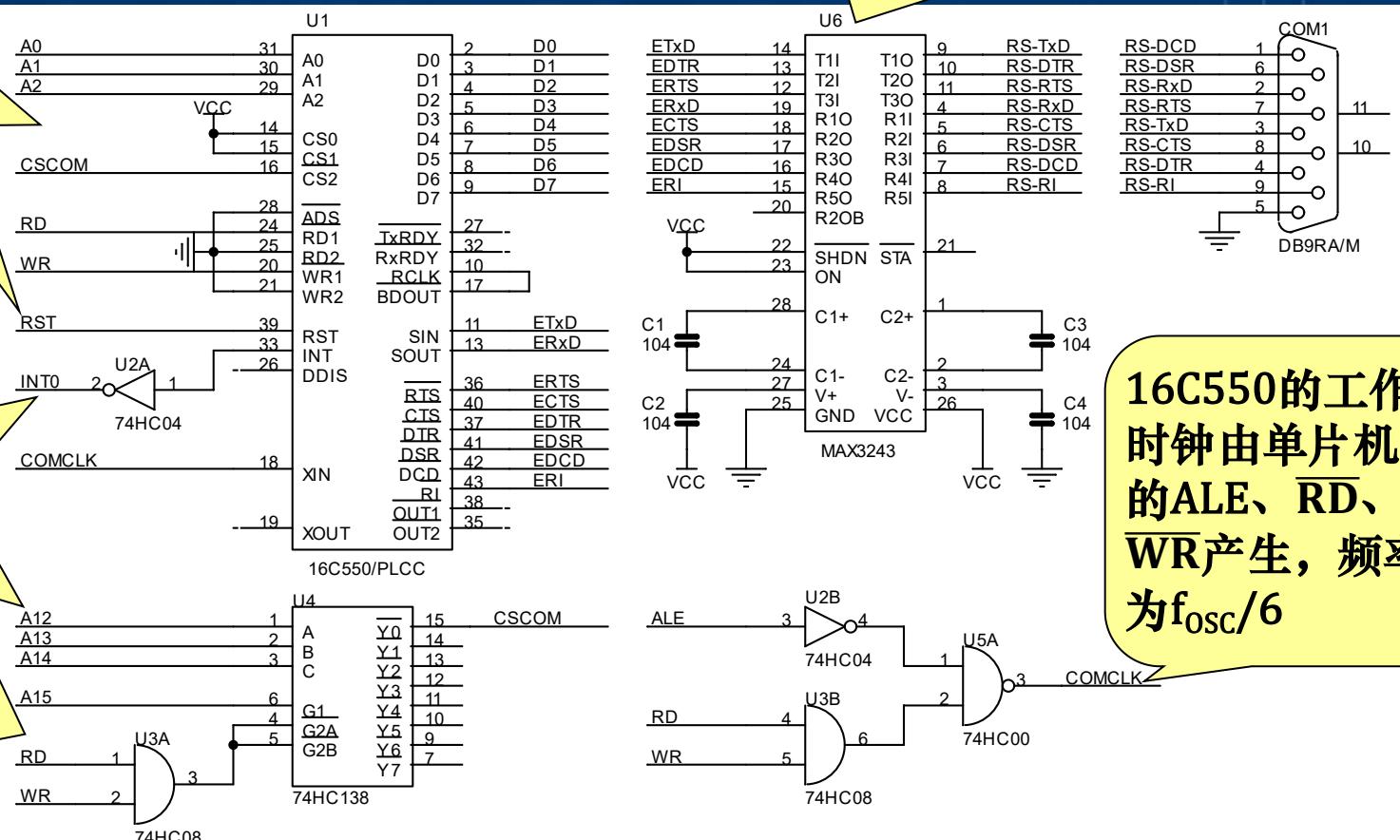
引脚	功能描述
BAUDOUT	16倍波特率时钟输出
RCLK	串行接收时钟，为波特率16倍
RTS、CTS、RI、DTR、DSR、DCD	Modem控制及数据流控信号
INTRRUPT	中断信号，高有效

单片机原理与接口技术教程

7.4.1 16C550简介

- ## ● 16C550和单片机的接口电路

DTE设备专用TTL-232电平变换芯片



16C550的工作时钟由单片机的ALE、 \overline{RD} 、 \overline{WR} 产生，频率为 $f_{osc}/6$



7.4.1 16C550简介

● 16C550内部控制寄存器地址安排

DLAB	A2	A1	A0	寄存器
0	L	L	L	[BASE]接收缓冲寄存器(读,RBR) [BASE]发送保持寄存器(写,THR)
0	L	L	H	[BASE+1]中断使能寄存器IER
X	L	H	L	[BASE+2]中断识别寄存器IIR(只读)
X	L	H	L	[BASE+2]FIFO控制寄存器FCR(写)
X	L	H	H	[BASE+3]线路控制寄存器LCR(写)
X	H	L	L	[BASE+4]Modem控制寄存器MCR(写)
X	H	L	H	[BASE+5]线路状态寄存器LSR(读)
X	H	H	L	[BASE+6]Modem状态寄存器MSR(读)
X	H	H	H	[BASE+7]暂存(Scratch)寄存器
1	L	L	L	[BASE]波特率除数锁存器 LSB
1	L	L	H	[BASE+1]波特率除数锁存器 MSB



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(1) 地址0: 接收数据缓冲寄存器 (只读)

D7	D6	D5	D4	D3	D2	D1	D0
接收的数据							

(2) 地址0: 发送数据暂存寄存器 (只写)

D7	D6	D5	D4	D3	D2	D1	D0
发送的数据							



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(3) 地址1：中断允许寄存器IER（只写）

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	EMSI	ELSI	ETBEI	ERBI

ERBI=1 允许接收数据准备好中断

ETBEI=1 允许发送缓冲器空中断

ELSI=1 允许线路状态变化中断

EMSI=1 允许Modem状态变化中断



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(4) 地址2：中断识别寄存器IIR(只读)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	IID2	IID1	IID0	PEND

PEND=0 表示有中断等待处理
具体内容见下页表



7.4.1 16C550简介

(4) 地址2：中断识别寄存器IIR(只读)

中断识别寄存器				优先级	中断类型	中断源	中断复位方法
b3	b2	b1	b0				
0	0	0	1	无	无	无	无
0	1	1	0	1	接收器线状态	溢出、奇偶、帧错误或断开中断	读LSR
0	1	0	0	2	接收数据可用	接收数据可用或达到FIFO触发门限	读RBR
1	1	0	0	2	字符超时指示	上次收发数据后，在4个字符时间内没有字符收发	读RBR
0	0	1	0	3	THR空	THR数据已发出	读IIR或写THR
0	0	0	0	4	Modem状态变化	CTS,DSR,RI或CD信号有变化	读MSR



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(5) 地址2： FIFO控制寄存器FCR(只写)

D7	D6	D5	D4	D3	D2	D1	D0
RTH	RTL	Rsv	Rsv	DMA	TFR	RFR	FEN

FEN=1 表示允许FIFO， 改变此位将清零内部FIFO

RFR=1 将清零内部接收FIFO及其计数器。移位寄存器
不被清零， 清零完成后此位自动复位(0)

TFR=1 将清零内部发送FIFO及其计数器。移位寄存器
不被清零， 清零完成后此位自动复位(0)



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(5) 地址2： FIFO控制寄存器FCR(只写)

D7	D6	D5	D4	D3	D2	D1	D0
RTH	RTL	Rsv	Rsv	DMA	TFR	RFR	FEN

DMA=1 当FEN同时为1时将使能DMA

RTH/RTL 表示接收器FIFO的触发门限， 2位共4种编码
(00, 01, 10, 11)分别表示触发门限为1, 4, 8, 14字节



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(6) 地址3：线路控制寄存器LCR(只写)

D7	D6	D5	D4	D3	D2	D1	D0
DLAB		SP	EPS	PEN	STB	WLS1	WLS0

WLS1,WLS0 数据位长度选择，四种编码对应数据位
长度分别为5, 6, 7, 8bit

STB 停止位长度选择，为0选择停止位为1位，为1停
止位为2位(当数据位为5bit时停止位为1.5bit)

PEN 校验位允许,当PEN=1时允许16C550产生/接收
校验位



7.4.1 16C550简介

(6) 地址3：线路控制寄存器LCR(只写)

D7	D6	D5	D4	D3	D2	D1	D0
DLAB		SP	EPS	PEN	STB	WLS1	WLS0

EPS 偶校验选择。当EPS=1且PEN=1时，16C550对收发的数据进行偶校验，否则进行奇校验。

SP 附加校验位(Stick parity)，即由此位来控制选择Space (bit3,4,5=1)或Mark方式的校验(bit3,5=1, bit4=0)。

Space 模式下校验位总为0，Mark模式下校验位总为1。

DLAB 除数锁存器访问控制。当置DLAB=1后，紧随其后的对16C550地址0, 1的访问都针对除数锁存器，而不是RBR, THR



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(7) 地址4：Modem控制寄存器MCR(只写)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	Loop	$\overline{\text{OUT2}}$	$\overline{\text{OUT1}}$	$\overline{\text{RTS}}$	$\overline{\text{DTR}}$

$\overline{\text{DTR}}$ 控制16C550的 $\overline{\text{RTS}}$ 引脚的输出。当 $\overline{\text{RTS}}$ 位=1时， $\overline{\text{RTS}}$ 引脚输出低电平,否则输出高电平；

$\overline{\text{RTS}}$ 控制16C550的 $\overline{\text{RTS}}$ 引脚的输出。当 $\overline{\text{RTS}}$ 位=1时， $\overline{\text{RTS}}$ 引脚输出低电平,否则输出高电平。

$\overline{\text{OUT1}}, \overline{\text{OUT2}}$ 用于控制16C550的 $\overline{\text{OUT1}}, \overline{\text{OUT2}}$ 引脚，控制方法及逻辑关系同上；

如果系统不控制Modem，上述4个控制位可用作输出。



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(8) 地址5：线路状态寄存器LSR(只读)

D7	D6	D5	D4	D3	D2	D1	D0
RFE	TEMT	THRE	BI	FE	PE	OE	DR

- DR 接收数据完成。当数据接收完成并已传到RBR或FIFO时，DR=1。读RBR或FIFO中的所有数据后该位清零；
- OE 重叠错误，表示RBR中字符被读出之前，已被下一个字符覆盖。此标志在CPU读LSR后即复位；
- PE Parity Error，奇偶校验错。PE=1表示所接收的数据的奇偶校验出错。当CPU读LSR后，PE复位。



7.4.1 16C550简介

(8) 地址5：线路状态寄存器LSR(只读)

D7	D6	D5	D4	D3	D2	D1	D0
RFE	TEMT	THRE	BI	FE	PE	OE	DR

- FE Framing Error, 帧错误, 表示接收的字符没有有效的停止位。
当CPU读LSR后, 此位复位;
- BI Break Interrupt, 断开中断, 为1时表示线路上有超过一个完整字符时间的持续低电平。当CPU读LSR后, 此位复位;
- THRE 发送保持缓冲器空, 为1时表示THR空, 可以接收下一个待发数据。若允许THRE中断, 16C550会向CPU申请中断。当CPU向THR写入数据或THR中数据送到TSR时, 此位复位。



7.4.1 16C550简介

(8) 地址5：线路状态寄存器LSR(只读)

D7	D6	D5	D4	D3	D2	D1	D0
RFE	TEMT	THRE	BI	FE	PE	OE	DR

TEMT 发送器空指示位，表示THR和TSR均为空。当THR或TSR任何一个中有数据的时候，此位复位；

RFE Receive FIFO Error，接收FIFO错误指示，表示在接收FIFO中至少有一个奇偶校验错、帧或断开错。当CPU读LSR或FIFO中没有后续错误时，RFE复位。



7.4.1 16C550简介

- 16C550内部寄存器功能描述

(9) 地址6: Modem状态寄存器MSR(只读)

D7	D6	D5	D4	D3	D2	D1	D0
DCD	RI	DSR	CTS	dDCD	TERI	dDSR	dCTS

dCTS,dDSR,dDCD

表示自上次CPU读MSR后， 对应的Modem状态线发生过变化。如果允许Modem状态变化中断，则将引发此中断；

TERI 表示已检测到RI脉冲的后沿，说明RI信号已有效。如果允许Modem状态变化中断，则将引发此中断。



7.4.1 16C550简介

(9) 地址6：Modem状态寄存器MSR(只读)

D7	D6	D5	D4	D3	D2	D1	D0
DCD	RI	DSR	CTS	dDCD	TERI	dDSR	dCTS

CTS,DSR,RI,DCD

对应16C550的Modem状态信号线当前状态的补码（即负逻辑）。如果系统不控制Modem，这四个引脚可用作输入，输入状态可通过MSR的高4 bit读入（负逻辑）。



7.4.1 16C550简介

● 16C550程序设计实例：宏定义

```
#ifndef __16C550__
#define __16C550__

#include <absacc.h>

// 定义16C550基地址，该地址可根据具体电路连接情况改变
#define COMBASE 0x8000
// 定义单片机系统晶振频率及XIN输入时钟频率
#define OSC      22118400
#define XIN      OSC/6
// 定义16C550内部寄存器读写地址
#define ComTHR  XBYTE[COMBASE+0x00]      // 发送数据寄存器(只写)
#define ComRHR  XBYTE[COMBASE+0x00]      // 接收数据寄存器(只读)
#define ComIER  XBYTE[COMBASE+0x01]      // 中断使能寄存器(只写)
#define ComIIR  XBYTE[COMBASE+0x02]      // 中断识别寄存器(只读)
#define ComFCR  XBYTE[COMBASE+0x02]      // FIFO控制寄存器(只写)
#define ComLCR  XBYTE[COMBASE+0x03]      // 线控制寄存器(只写)
#define ComMCR  XBYTE[COMBASE+0x04]      // 调制解调器控制寄存器(只写)
#define ComLSR  XBYTE[COMBASE+0x05]      // 线路状态寄存器(只读)
#define ComMSR  XBYTE[COMBASE+0x06]      // 调制解调器状态寄存器(只读)
#define ComSCR  XBYTE[COMBASE+0x07]      // 暂存寄存器
#define ComDLL  XBYTE[COMBASE+0x00]      // 波特率除数锁存器(低)
#define ComDLM  XBYTE[COMBASE+0x01]      // 波特率除数锁存器(高)

void Init16C550(unsigned long);

#endif
```



7.4.1 16C550简介

- 16C550程序设计实例：初始化

```
/*****************************************************************************  
函数 :  Init16C550  
功能 :  初始化16C550 , 波特率、数据位、中断等参数均自己定义  
参数 :  波特率  
*****/  
void Init16C550(unsigned long Baud)  
{  
    ComLCR = 0x80;                      // DLAB=1 , 准备写DLL/DLM  
    ComDLL = (XIN/16/Baud)%256;          // 设置波特率除数低8位(DLL)  
    ComDLM = (XIN/16/Baud)/256;          // 设置波特率除数高8位(DLM)  
    ComLCR = 0x03;                      // 帧格式:8位数据位,1位停止位,无校验  
    ComFCR = 0x00;                      // 不使用FIFO  
    ComMCR = 0x0b;                      // RTS,DTR有效(低),OUT2为低,OUT1为高  
    ComIER = 0x00;                      // 禁用中断  
}
```



7.4.1 16C550简介

- 16C550程序设计实例：查询方式数据发送

```
*****
函数： ComWriteChar
功能： 通过查询方式向16C550发送一个字节的数据
参数： 要发送的字符
返回： 通信是否成功，正常返回0，出错返回非0
*****
char ComWriteChar(unsigned char c)
{
    int t = 0;
    while((ComLSR&0x20) == 0)           // 如果THRE无效，则循环判断是否超时
    {
        t++;                         // 执行t+1操作，累计等待时间
        if(t>5000) return COMTIMEOUT; // 大于超时门限则返回错误指示。门限值应根据
                                       // 波特率设置，波特率越高，超时循环门限越小。
    }

    ComTHR = c;                      // 如果THRE有效，则将待发数据写入 THR
    return 0;                         // 函数返回。数据由16C550硬件自动发送
}
```



7.4.1 16C550简介

- 16C550程序设计实例：查询方式数据接收

```
/**************************************************************************  
函数： ComReadChar  
功能： 通过查询方式读入16C550接收到的数据（1个字节）  
参数： 无  
返回： 读入的字符  
*****/  
unsigned char ComReadChar(void)  
{  
    while((ComLSR&0x01) == 0);           // 如果DR无效，则循环等待  
    return (ComRHR);                   // DR有效，返回读入数据  
}
```



第七章 单片机串行通信接口



7.5 通过RS-485总线
实现单片机的多机通信



概述

● RS-485总线通信方式的特点

- 速度较高 —— 250Kbps~1Mbps；
- 通信距离远 —— 无中继通信距离达4000英尺/1200米；
- 信号电平兼容 —— 线路采用差分方式驱动；
- 支持多点通信—— 同一总线下可挂接多达128个节点；

● 目前常用的RS-485总线驱动芯片

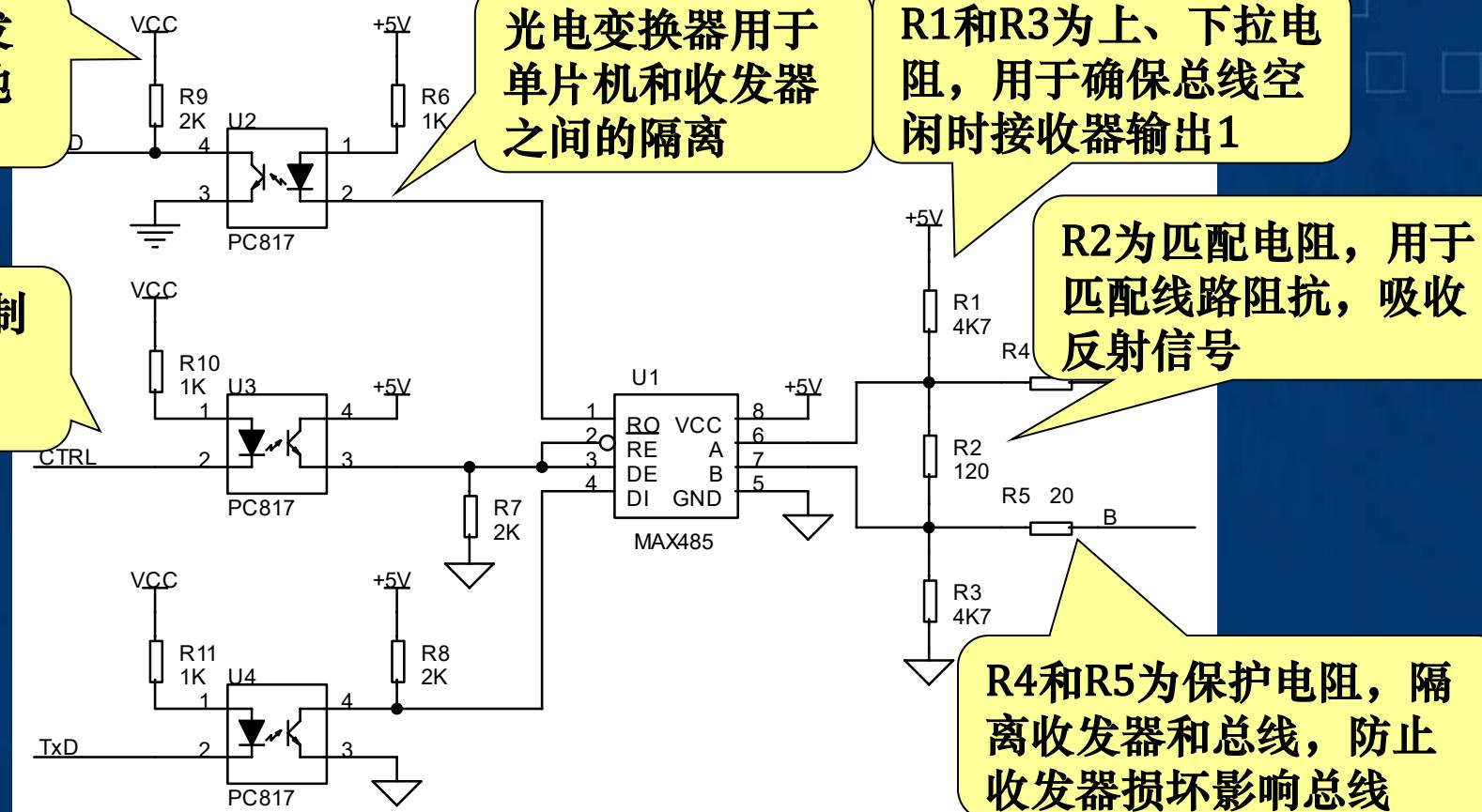
- MAX48x/148x系列、75176、SP48x等；
- 通信方式有半双工和全双工；
- 芯片等级分为商业级 (0~70°C) 、工业级 (-40~85 °C) 和汽车/军事级 (-55~125 °C) ；
- 同一总线下最多可挂接32~128个节点。



7.5.1 单片机和RS-485总线收发器的接口电路设计

- 带有总线保护及光电隔离的RS-485接口电路

单片机和收发器的供电及地都需要隔离



光电变换器用于单片机和收发器之间的隔离

R1和R3为上、下拉电阻，用于确保总线空闲时接收器输出1

R2为匹配电阻，用于匹配线路阻抗，吸收反射信号

R4和R5为保护电阻，隔离收发器和总线，防止收发器损坏影响总线



7.5.2 单片机主从式多机通信的原理

1、多机通信控制位

- 当串行口工作在方式2或3（多机通信模式）接收数据时，SCON中的SM2为多机通信控制位：
 - 多机通信模式下，单片机串行口发送和接收数据均增加1位附加位，分别记为TB8和RB8；
 - 单片机串行口发送一字节的数据时，TB8将紧接在该字节后发出；
 - TB8和RB8位于SCON中，可位寻址。



7.5.2 单片机主从式多机通信的原理

1、多机通信控制位

- 当串行口工作在方式2或3（多机通信模式）接收数据时，SCON中的SM2为多机通信控制位：
 - SM2=1时，单片机串行口接收一个字节后，只有当紧接该字节的第9位数据RB8也为1时，该字节才会被装入SBUF，并置RI为1。否则单片机认为该数据无效，不置位RI，CPU也不会做任何处理；
 - SM2=0时，单片机串行口接收一个字节后，不管紧随其后的RB8是0还是1，均会将该字节装入SBUF，并置RI为1；
 - 利用单片机的上述特性，可实现主从式半双工多机通信。

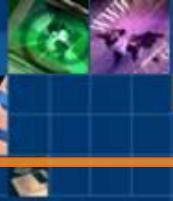


7.5.2 单片机主从式多机通信的原理

2、主从式半双工多机通信系统的构成及原理

● 主从式半双工多机通信系统的构成

- 采用RS-485总线；
- 采用一主机+多从机的结构；
- 单片机使用串口方式2或3进行多机通信；
- 主机采用轮询的方式访问各从机；
- 各从机之间的数据交换只能通过主机进行转发；
- 或者各节点轮流成为主机（需要协议支持）；
- 因为任何时候通信都只能是单向的（主机→从机或从机→主机），所以构成的是半双工的通信系统。

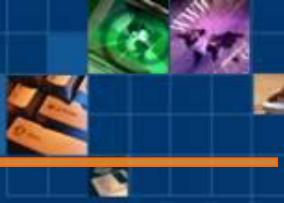


7.5.2 单片机主从式多机通信的原理

2、主从式半双工多机通信系统的构成及原理

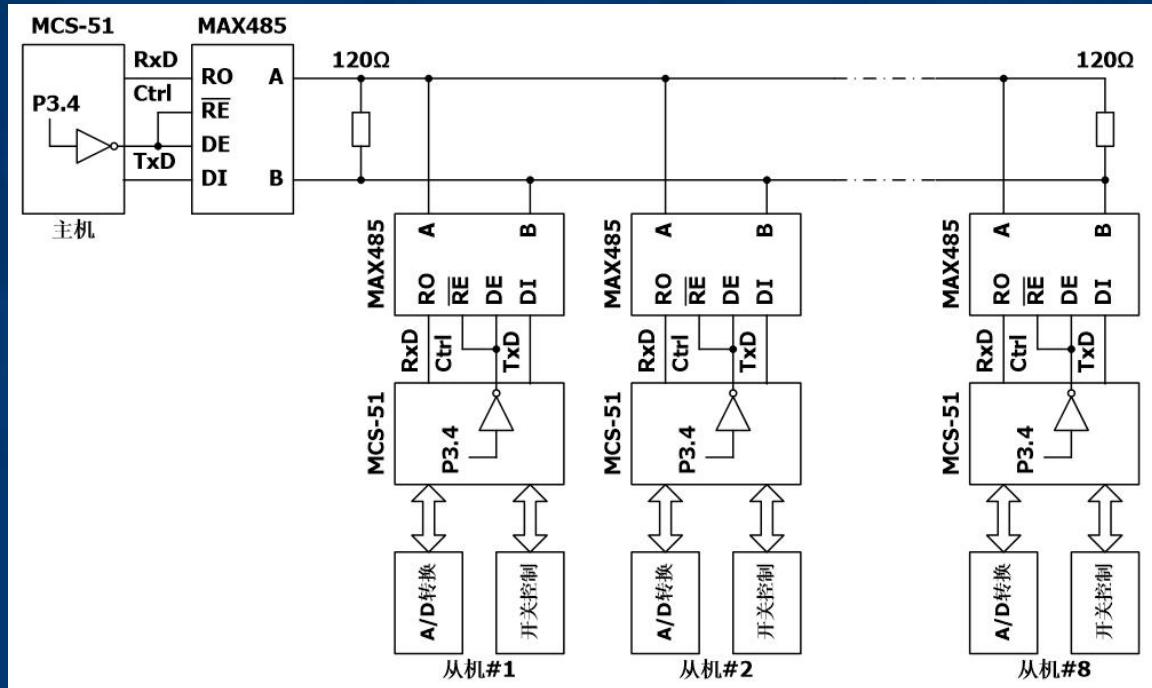
● 多机通信的实现

- 使所有从机SM2=1，等待主机发送地址；
- 主机置TB8=1表示发送的是地址，并发送地址字节；
- 所有从机都可收到此字节，各自比较地址码是否和主机发来的数据相符，地址码符合的从机置SM2=0；
- 主机给被寻址的从机发送数据，发送数据期间置TB8为0，此时只有SM2=0的从机可收到该数据，其它所有的SM2=1的从机不会接收TB8=0的数据；
- 主机和被寻址从机间完成数据通信，主机置TB8=1、从机置SM2=1，准备开始下一轮循环。



7.5.3 单片机主从式多机通信的实例

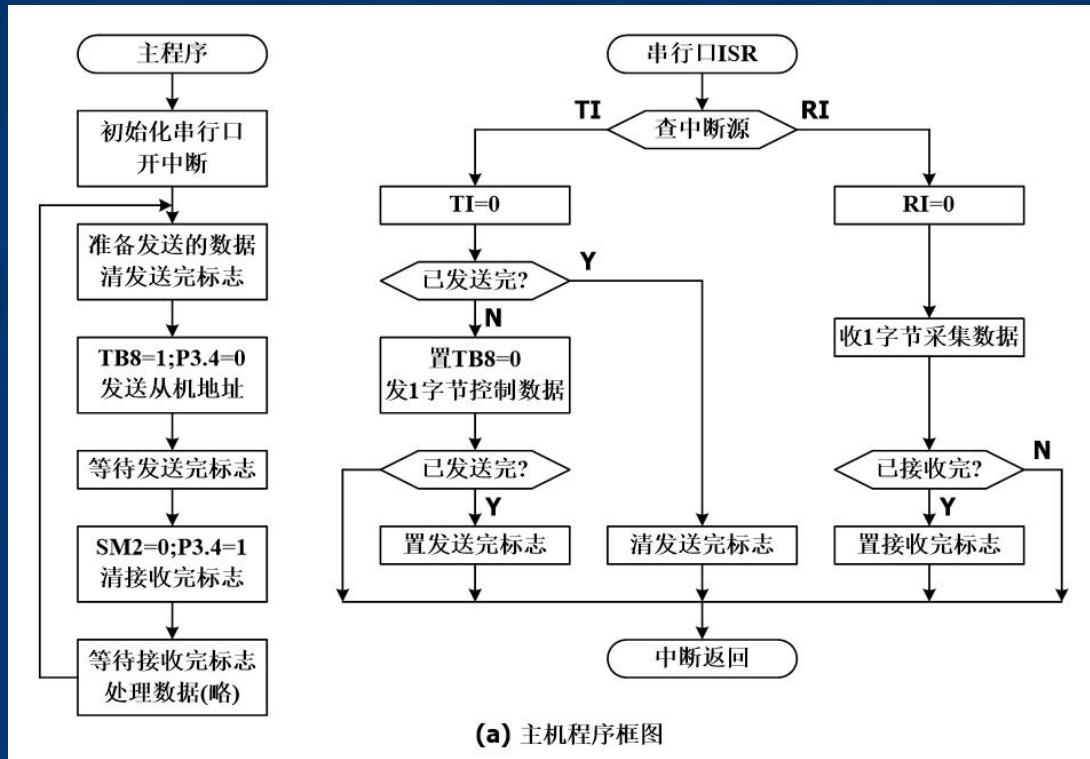
● 系统电路



单片机原理与接口技术教程

7.5.3 单片机主从式多机通信的实例

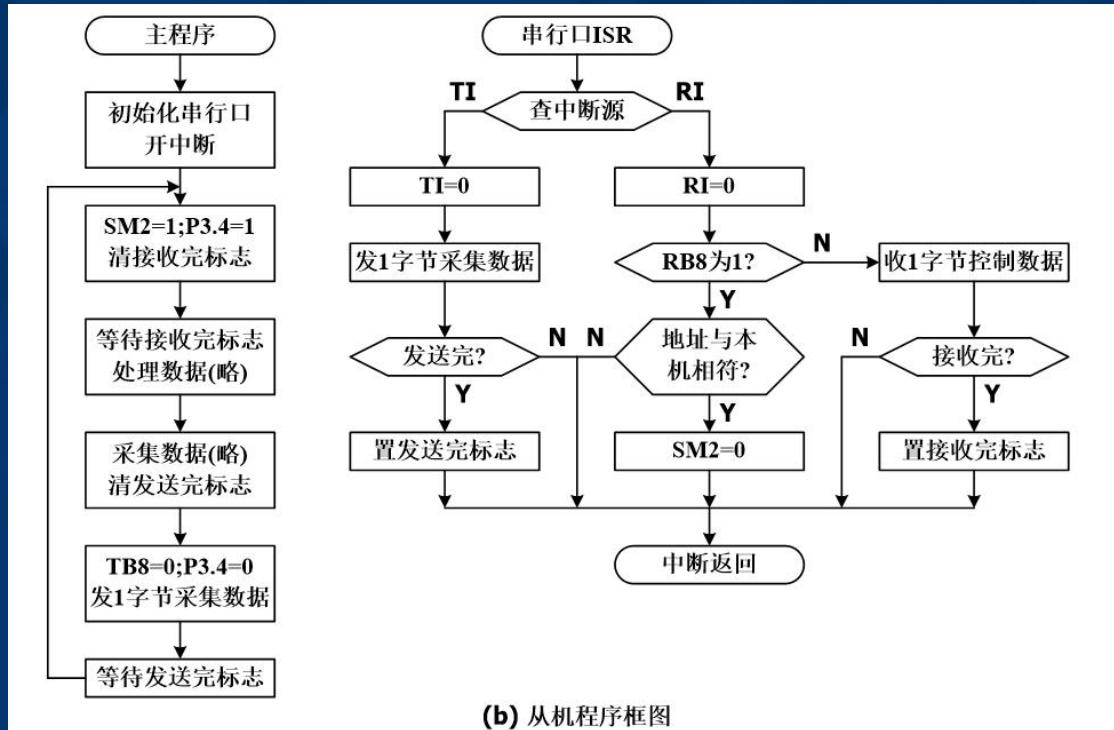
● 主机程序流程图





7.5.3 单片机主从式多机通信的实例

● 从机程序流程图





7.5.3 单片机主从式多机通信的实例

- 代码 (略)