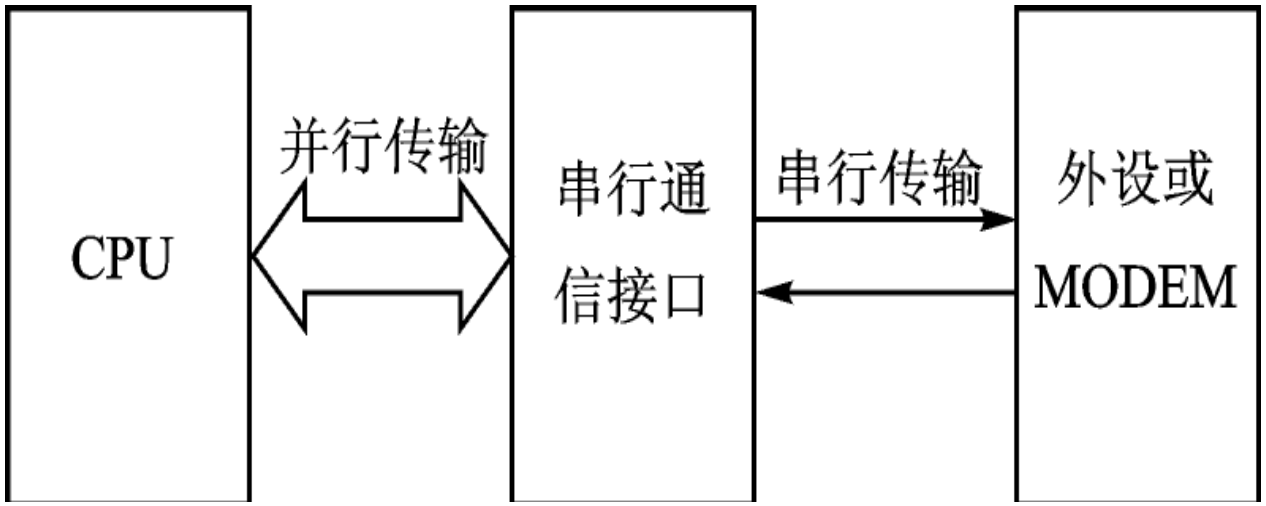
第8章 串行通信接口

8.1 串行通信基础知识

串行通信的特点：

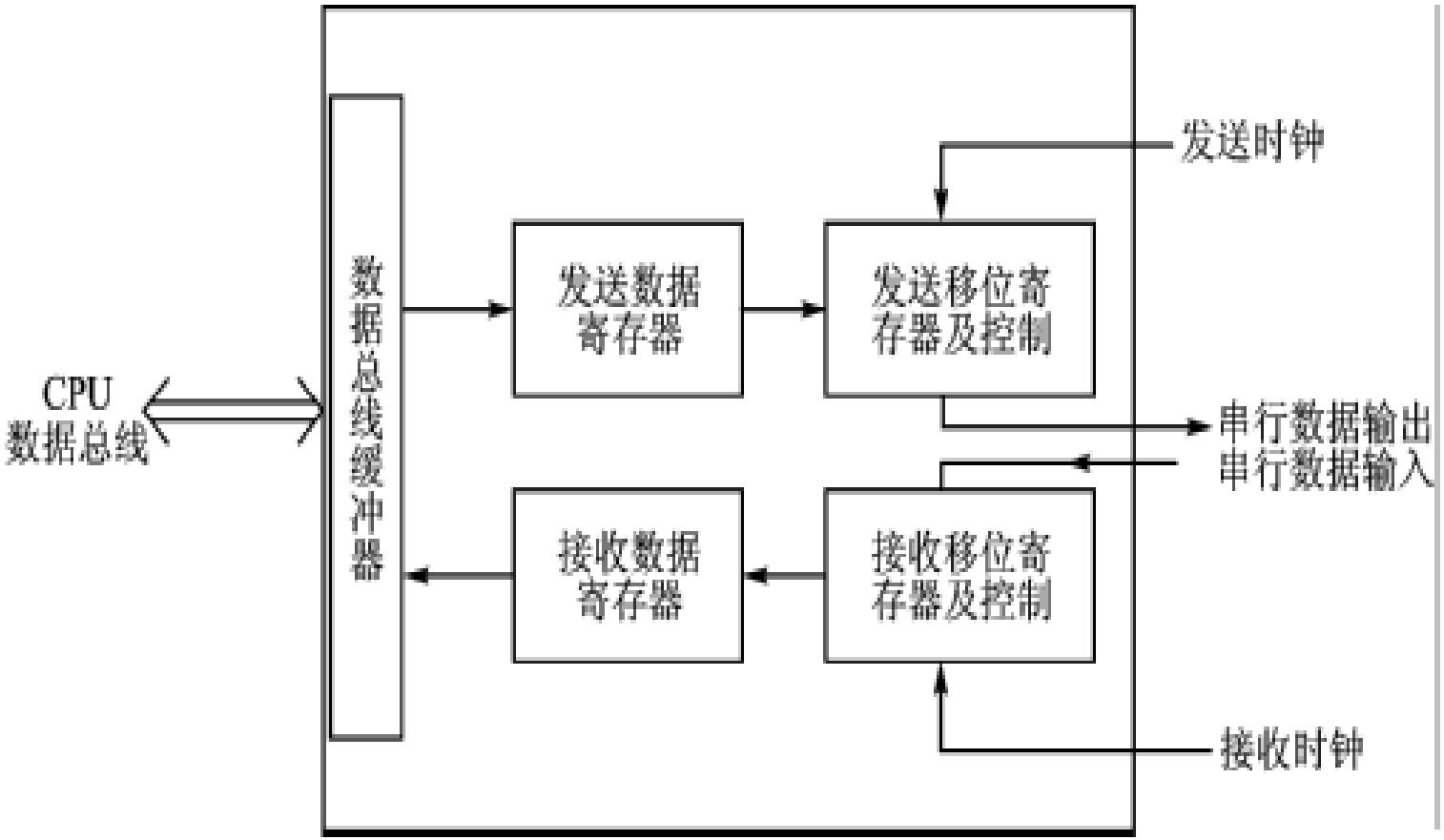
按位传送：数据是按位逐位传送；

通信线路少：一个方向只用一根通信线路传输信息；



异步串行通信基本结构

异步串行通信接口也称为异步接收发送器，简称UART，典型的异步通信接口(UART基本结构)。



8.1.1 串行通信的数据传送方式

串行通信中，数据通常是在两个站点之间进行传送。按照双方数据交换的能力，可分成三种基本传送模式：单工、半双和全双工传送：

- 单工通信：数据仅能从设备A到设备B进行单一方向的传输。
- 半双工通信(Half duplex)：数据可以从设备A到设备B进行传输，也可以从设备B到设备A进行传输，但不能在同一时刻进行双向传输。
- 全双工通信(Full duplex)：数据可以在同一时刻从设备A传输到设备B，或从设备B传输到设备A，即可以同时双向传输。

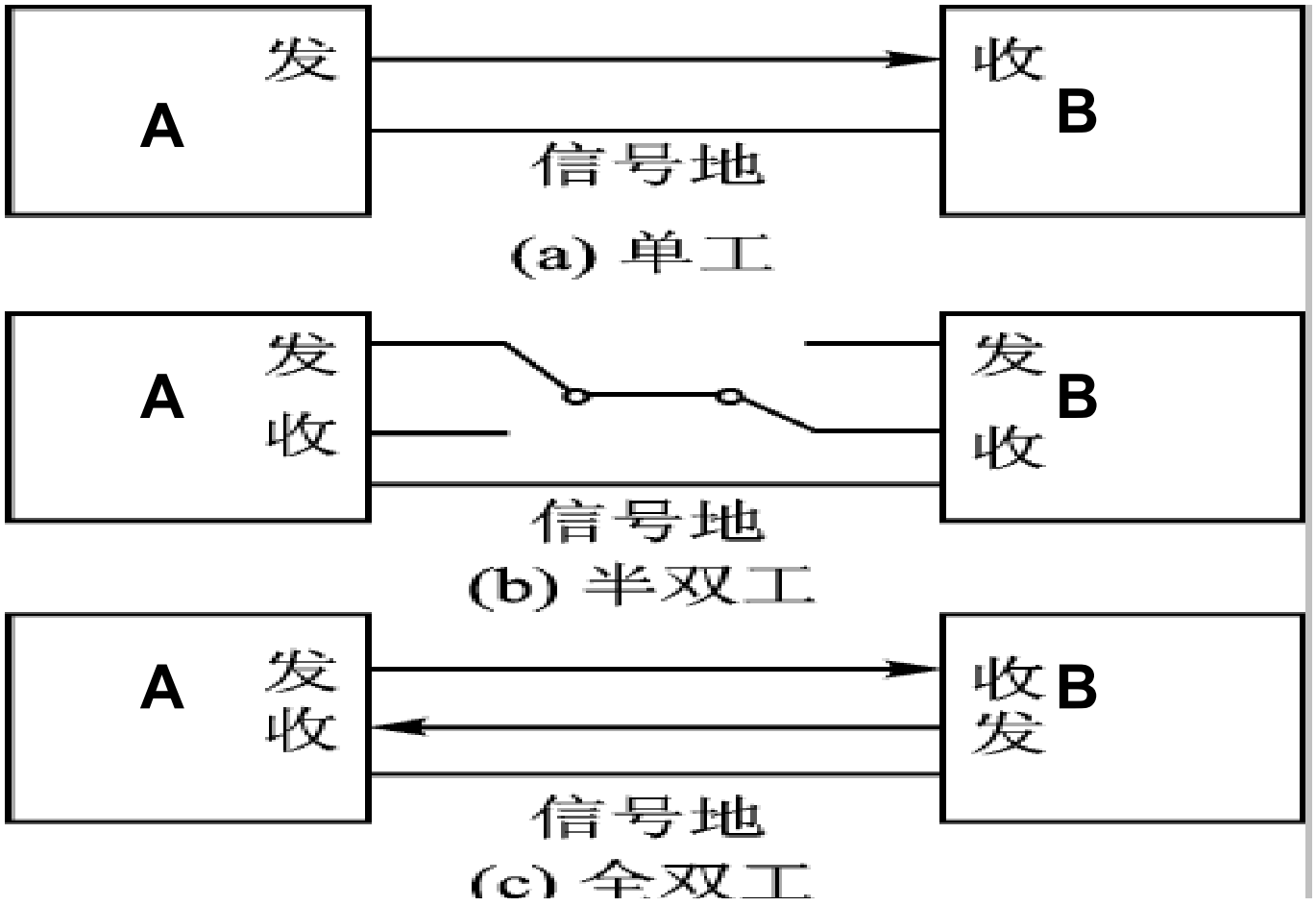


图　数据通信的三种传送模式

8.1.2 串行通信的同步方式

根据同步方式的不同，串行通信可分为异步通信方式和同步通信方式。

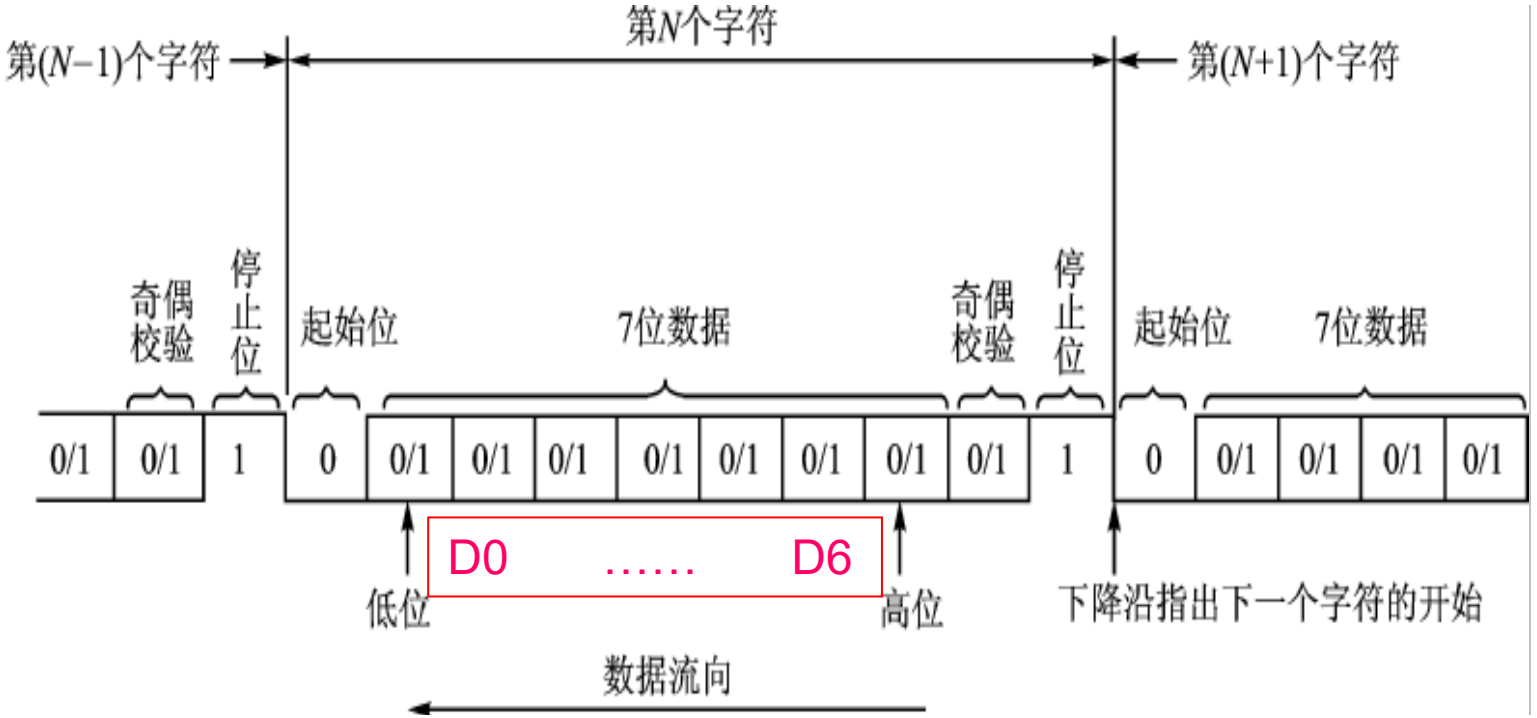
(1)异步通信方式：不需要对通信双方的时钟进行同步，每发送一个数据前先发送起始位，发送完数据后再发送结束位，以此作为双方同步的依据。这种方式对硬件要求较宽松，电路简单，但传输效率不高。

(2)同步通信方式：收发双方的时钟是同步的，数据可按块连续传送。通常通过一根专门的时钟信号线来同步双方的时钟。同步串行通信方式传输效率高，但对硬件要求高，电路结构复杂。

通信协议

通信协议是通信双方对通信内容的约定，有物理层协议、数据/链路层协议、应用层协议等。数据/链路层协议规定了帧结构、波特率和数据校验方式。

异步串行通信采用起止式异步协议：



① 帧结构：起止式异步通信帧结构包含有起始位（0）、数据位、奇偶校验位和停止位（1），每传送一个数据都以起始位来通知对方开始接收数据，以停止位结束数据传送。异步通信对字符的格式、波特率、校验位有确定的要求，一般用在数据速率较低的场合。

② 波特率：每秒传送的符号（数据）速率。常用的波特率9.6、14.4、28.8、56、115.2等。

③ 数据校验：异步串行通信通常采用奇偶校验。

数据校验方式

数据在传输途中，由干扰引起误码是在所难免的。如何发现传输中的错误叫检错；发现错误后，如何消除错误叫纠错。

(1) 奇偶校验

通过校验位控制数据中“1”的个数为奇数或偶数。偶校验就是要使字符加上校验位有偶数个“1”；奇校验就是使字符加上校验位有奇数个“1”。如“00010011”有奇数个1，偶校验时校验位置为“1”，奇校验时校验位置为“0”。

(2) 循环冗余码校验CRC (Cyclic Redundancy Check)

高级的通信协议(如同步串行通信协议)中一般采用循环冗余码CRC检错，具有一定的纠错能力。

调制与解调

进行长距离传输时，需要在发送端将数字信号转换成适合通信线路传输的信号形式，这一过程称为“调制(Modulate)”。在接收端将通信线路上传输的信号还原成原来的数字信号，这一过程称为“解调(Demodulate) ”。

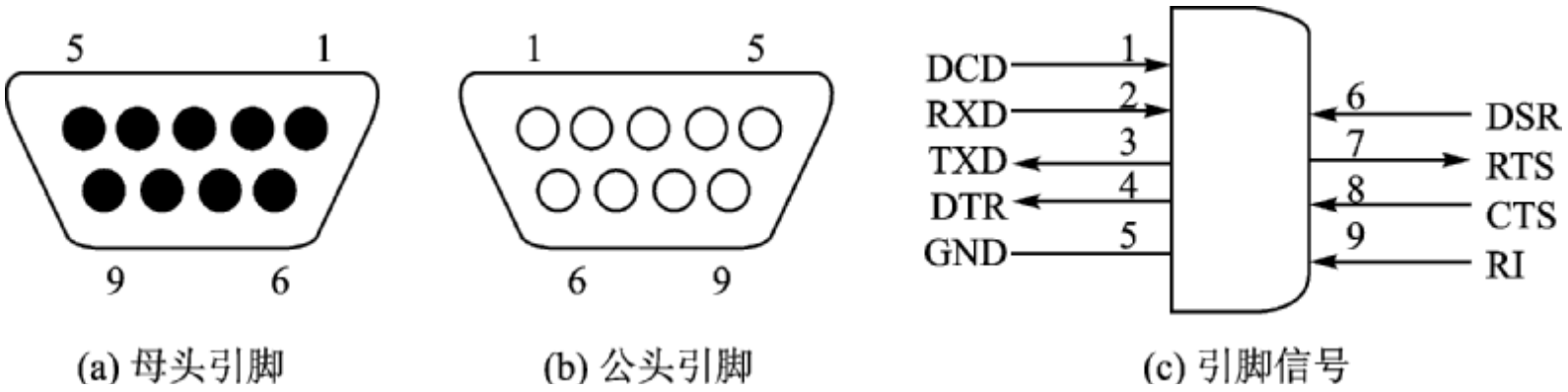
调制解调器MODEM是进行数据通信所需要的设备，通常称之为：数据通信设备(DCE: Data Communication Equipment)或数据装置 (Data Set)。

收发数据的设备称为：数据终端设备（DTE：Date Terminal Equipment）

8.1.3 RS－232C串行接口标准

1. RS-232C连接器

最常用的RS－232C连接器是DB－9型连接器，如下图所示。



(a) 母头引脚　　(b) 公头引脚　　(c) 引脚信号

2. RS-232C接口信号定义：

(1) 请求发送RTS(Request To Send)：此信号表示DTE请求DCE发送数据。

(2) 清除(允许)发送CTS(Clear To Send)：此信号表示DCE准备好接收DTE发来的数据。是对请求发送信号的响应信号。

RTS/CTS这对联络信号用于半双工系统中发送方式和接收方式间的切换。在全双工系统中发送方式和接收方式间的切换，因配置双通道，故不需要RTS/CTS联络信号，应该使其接高电平。

(3) 数据装置准备好DSR(Data Set Ready)：此信号有效(ON状态)时表明MODEM处于可以使用的状态。

(4) 数据终端准备好DTR(Data Terminal Ready)：此信号有效（ON状态）时表明数据终端可以使用。

(5) 载波检测DCD(Data Carrier Detection): 此信号用来表示DCE已接通通信信道，即本地MODEM检测到通信链路另一端(远地)的DCE送来的载波信号，通知DTE准备接收数据。

(6)振铃指示RI(Ringing): 当MODEM检测到线路上有振铃呼叫信号时，使该信号有效(ON状态)，通知DTE已被呼叫，是否接听呼叫由DTE决定。

(7)发送数据TxD(Transimitted Data): 通过TxD线计算机将串行数据发送到DCE。

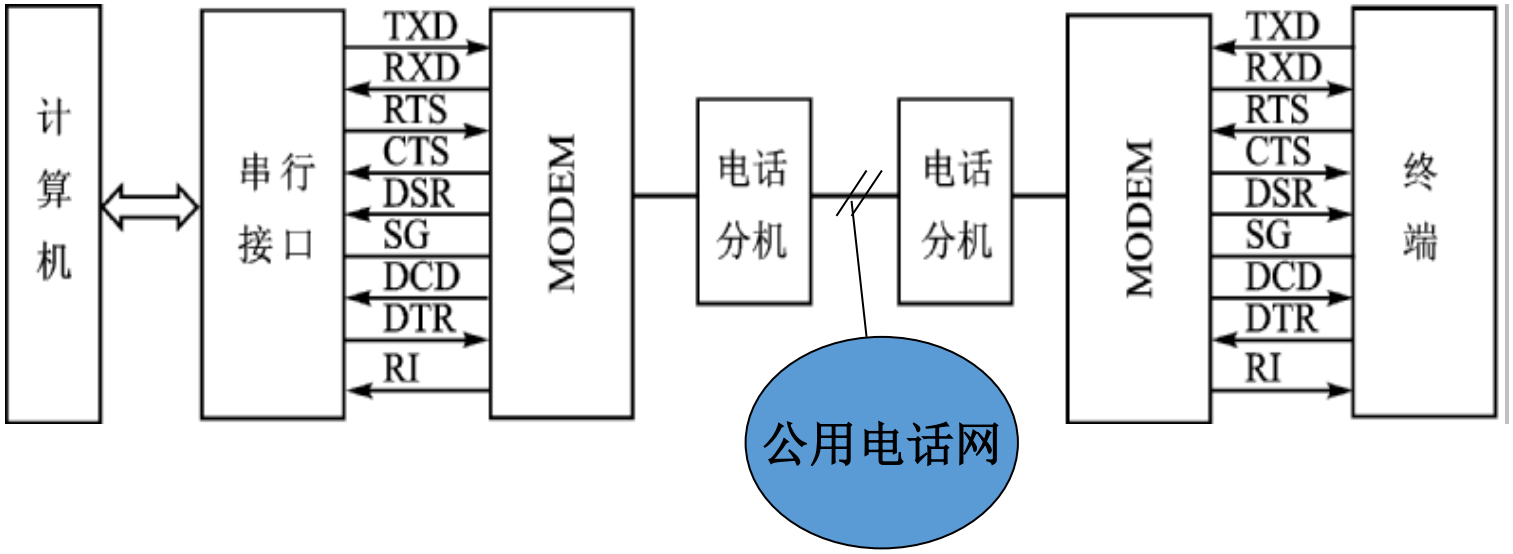
(8)接收数据RxD(Received Data): 通过RxD线计算机接收从DCE送来的串行数据。

(9) SG: 信号地。

3. 信号线的连接

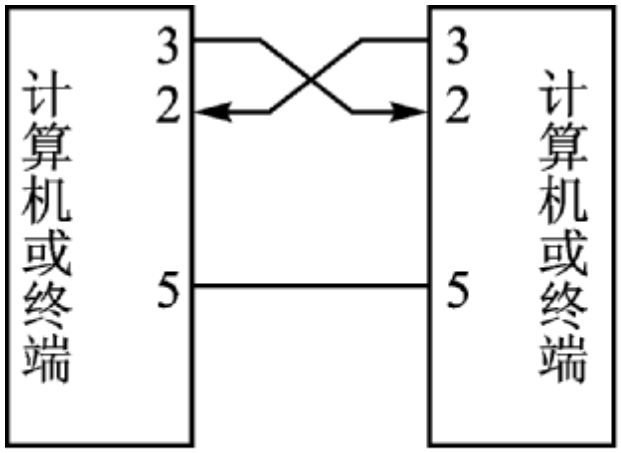
1). 远距离连接

远距离连接需要加调制解调器MODEM，如下图所示。

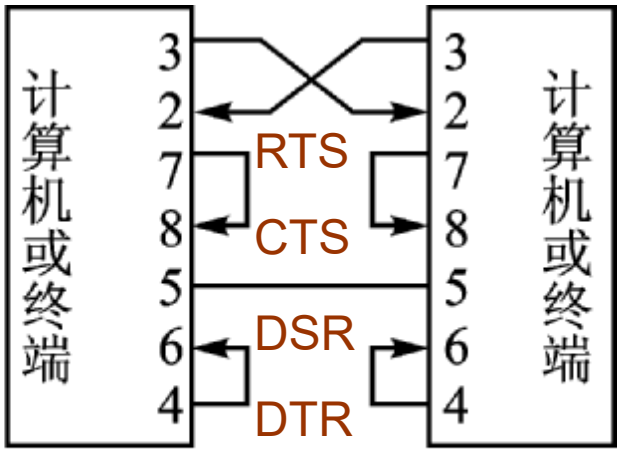


2)． 近距离连接

近距离连接不用MODEM，如下图所示。



(a) 无联络信号



(b) 带联络信号

4. 电气特性

RS－232C对电气特性、逻辑电平和各信号线功能都作了规定。

1). 电平规定

对于数据发送TxD和数据接收RxD线上的信号电平规定为：逻辑1（MARK: 传号）＝－3～－15V，典型值为－12V；逻辑0（SPACE: 空号）＝＋3～＋15V，典型值为＋12V。

对于RTS、CTS、DTR和DCD等控制和状态信号电平规定为：信号有效（接通，ON状态）＝＋3～＋15V，典型值为＋12V；信号无效（断开，OFF状态）＝－3～－15V，典型值为－12V。

2). 电平转换

PC机串口采用RS-232C电平，而ARM中标准逻辑“1”对应的电平是2~3.3v,标准逻辑“0”对应的电平是0~0.4v。显然两者间通信要进行电平转换，通常使用MC1488/MC1489或MAX232等专用IC进行电平转换。

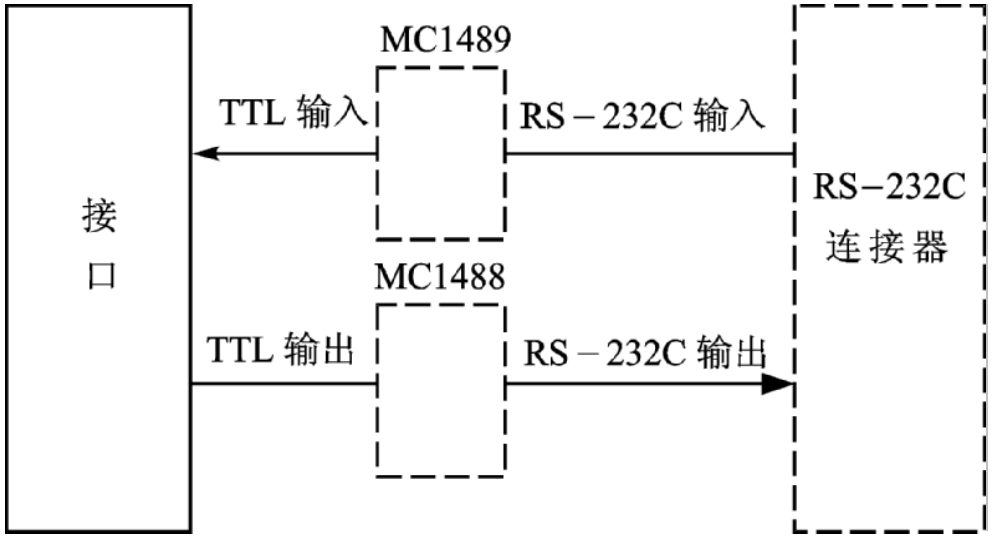


图　EIA-RS-232C与TTL电路电平转换（1）

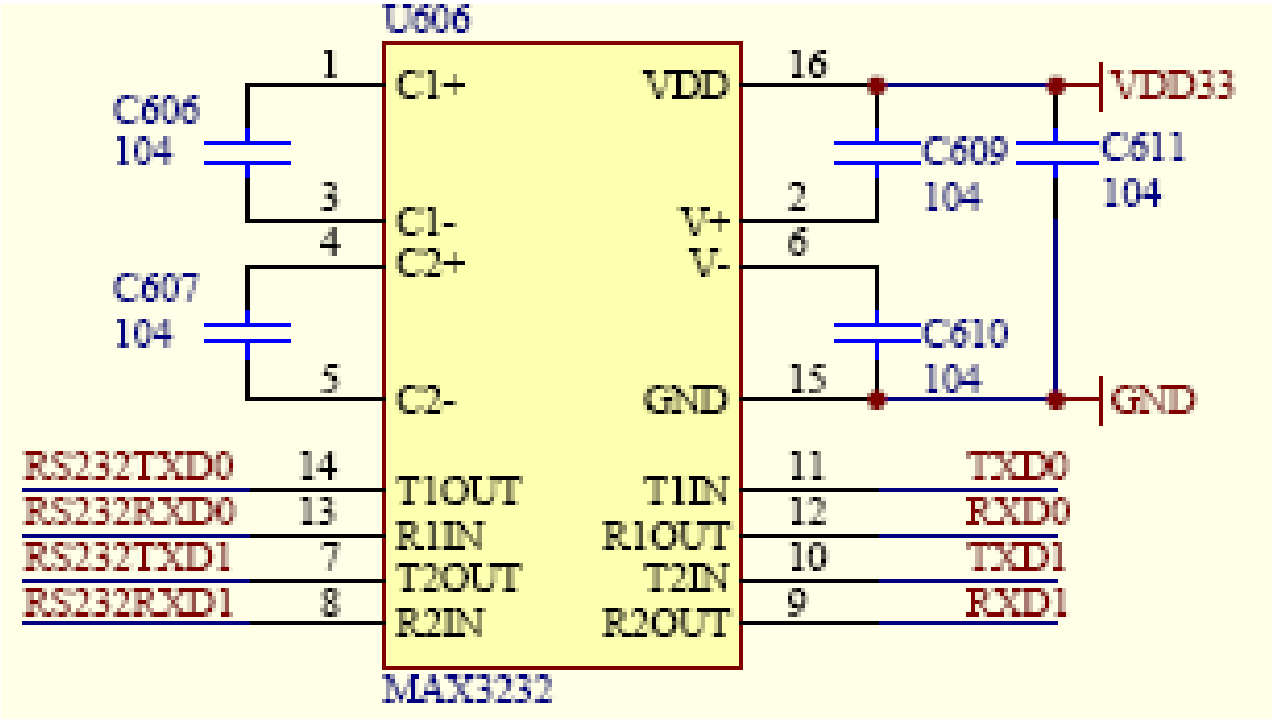


图　EIA-RS-232C与TTL电路电平转换（2）

3)．传输距离及通信速率

EIA颁布的RS-232C标准规定：DTE和DCE之间最大传输距离为15m，传输数据速率不能高于20Kbps。

RS-232-C接口标准出现较早，难免有不足之处，主要有以下四点：

（1）接口的信号电平值较高，易损坏接口电路的芯片，又因为与TTL 电平不兼容故需使用电平转换电路方能与TTL电路连接。

（2）传输速率较低，在异步传输时，波特率为20Kbps。

（3）抗噪声干扰性能弱，接口使用一根信号线和一根信号返回线而构成共地的传输形式， 这种共地传输容易产生共模干扰，所以。

（4）传输距离有限，最大传输距离标准值为50英尺，实际上也只能 用在50米左右。

针对RS-232-C的不足，不断出现了一些新的接口标准，RS-485就是其中之一，它具有以下特点：

1）电气特性：逻辑“1”以两线间的电压差为+2~+6v表示；逻辑“0”以两线间的电压差为-2~-6v表示。

2）收发器采用平衡发送和差分接收，即在发送端，驱动器将TTL电平信号转换成差分信号输出；在接收端，接收器将差分信号变成TTL电平，因此具有抑制共模干扰的能力。接收器能够检测低达200mV的电压，具有高的灵敏度，故数据传输距离可达千米以上。

3）RS-485总线支持多节点：一般最大支持32个节点，如果使用特制的485芯片，可以达到128个或者256个节点，最大的可以支持到400个节点。

因RS-485接口具有良好的抗噪声干扰性，长的传输距离和多站能力等优点使其在嵌入式系统中获得大量运用。

RS485接口只需两根连线组成半双工通信网络，所以RS485接口均可直接采用屏蔽双绞线传输。

RS485接口连接器采用DB-9的9芯插头座，与智能终端RS485接口采用DB-9(孔)，与键盘连接的键盘接口RS485采用DB-9(针)。

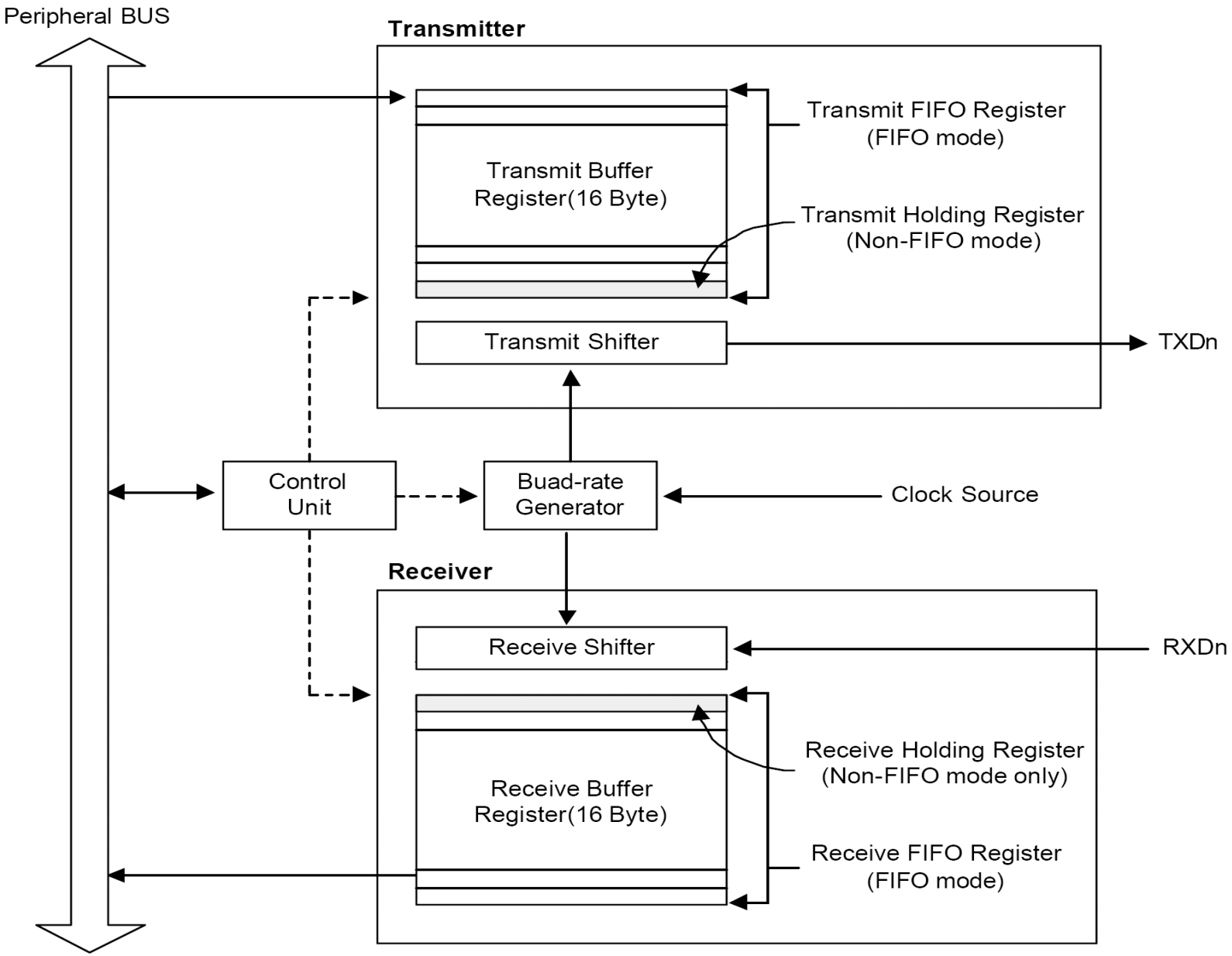
8.2 S3C2410串行接口

S3C2410有3个独立的UART（Universal Asynchronous Receiver-Transmitter: 通用异步串行口）：UART0、UART1、UART2，每个串口都可rf以在中断和DMA两种模式下进行收发。UART支持的最高波特率达230.4kbps。

每个UART包含：波特率发生器、接收单元、发送单元和控制单元。波特率发生器以PCLK或UCLK为时钟源。发送器和接收器各包含1个16 字节的FIFO寄存器和移位寄存器。

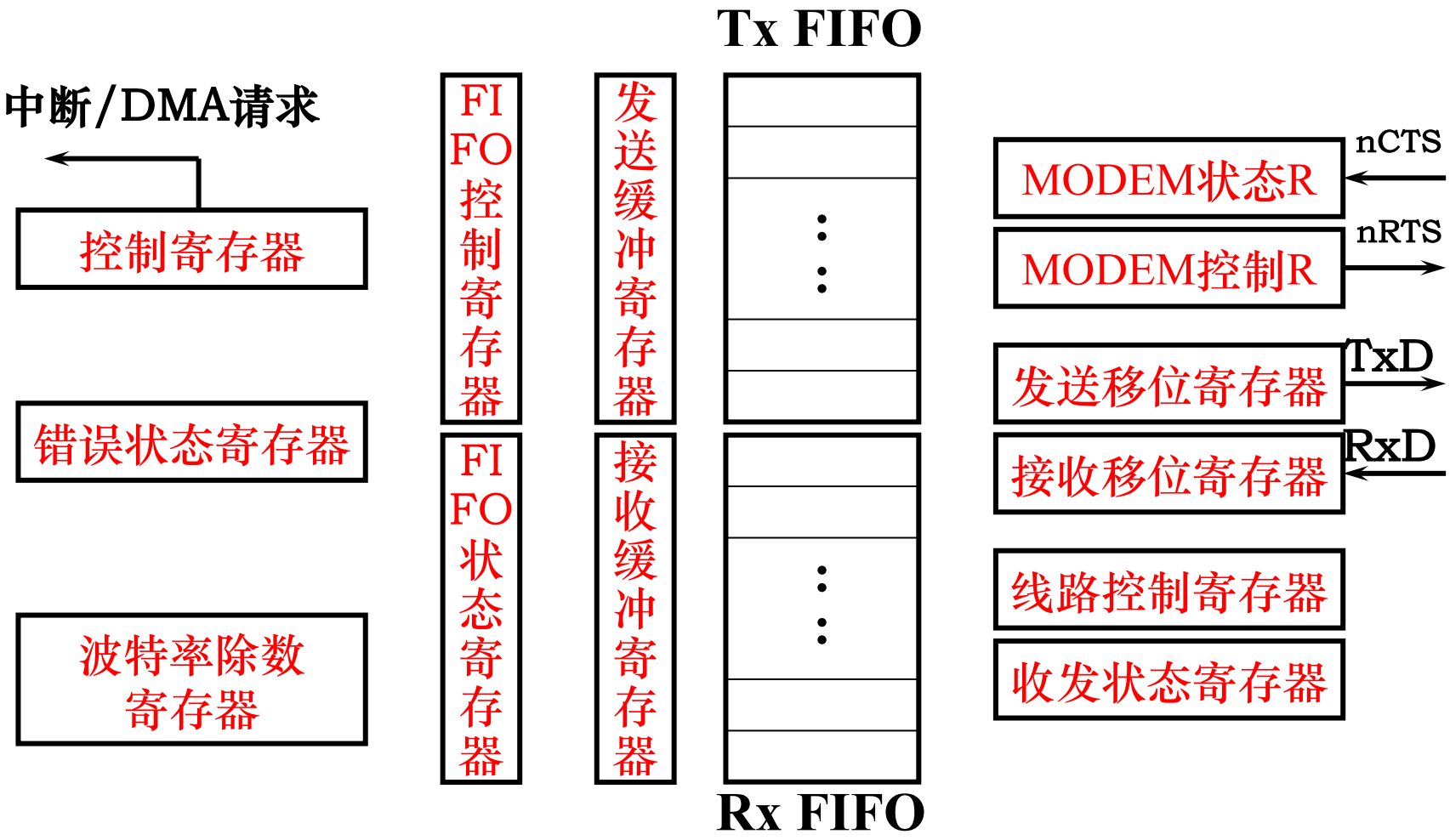
发送数据：数据先写到FIFO，然后拷贝到发送移位寄存器，再从数据输出端口(TxDn)依次被移位输出。

接收数据：从接收端口(RxDn)移位输入到移位寄存器，然后拷贝到FIFO中。



In FIFO mode, all 16 Byte of Buffer register are used as FIFO register.
In non-FIFO mode, only 1 Byte of Buffer register is used as Holding register.

8.2.1 S3C2410 UART接收器/发送器结构



8.2.2 S3C2410工作原理

（1）串行口的操作

数据帧格式：可编程，包含1个开始位、5 到8 个数据位、1个可选的奇偶校验位、1个或2个停止位，使用通过线路控制器(ULCONn)来设置。发送器还可以产生中止(暂停)状态，在一帧发送期间连续输出逻辑0 。

发送暂停 (中止)信号：迫使串口输出逻辑0 ，这种状态保持一个传输帧的时间长度。通常在一帧传输数据完整地传输完成之后，再通过这个全0 状态将中止信号发送给对方。中止信号发出后，继续发送数据到Tx FIFO (在不使用FIFO模式下，将被放到输出保持寄存器)。

8.2.2 S3C2410工作原理

（1）串行口的操作

接收器具有错误检测功能：可以检测出溢出错误，奇偶校验错误，帧错误和暂停错误，每种情况下都会将一个错误标志在接收状态寄存器置位。

注：暂停错误是指RxDn维持逻辑0的时间超过了一个帧时间长度。

（2）串行口的波特率发生器

每个UART 的波特率发生器为传输提供了串行移位时钟。波特率发生器的时钟源可以从S3C2410的内部系统时钟PCLK或UCLK中来选择。波特率数值决定于波特率除数寄存器(UBRDIVn)的值，波特率数与UBRDIVn 的关系为：

$$UBRDIVn=(int)(CLK/(f_B*16))-1$$

其中CLK为所选择的时钟频率，f_B为波特率。

$$f_B= CLK/16/(UBRDIVn + 1)$$

例如，如果波特率为115200bps且PCLK或UCLK 为40MHz,则UBRDIVn 为：

$$UBRDIVn = (int)(40000000)/(115200*16) - 1$$

$$= (int)(21.7) - 1$$

$$= 21 - 1 = 20$$

（3）串行口波特率误差极限

在应用中，实际波特率往往与理想波特率有差别，其误差不能超过一定的范围，其极限为：UART传输10bit数据的时间误差应该小于1.87％（3/160）。

t_true = (UBRDIVn + 1)×16×10 / PCLK

实际的传输10bit所需时间

t_ideal = 10 / baud_rate

理想情况下传输10位需要的时间

UART error＝(t_true – t_ideal) / t_ideal ×100%

（4）自动流控制功能(AFC：Auto Flow Control)

流控制是指数据流收发握手控制。当数据在两个串口这间传输时，常常会由于接收端数据处理跟不上，造成接收缓冲区满，此时发送端如果继续发送数据，接收端就会丢失数据。使用流控制可以解决该问题。

UART0和UART1不仅有完整的握手信号，而且有自动流控制功能，在MODEM控制寄存器UMCONn中设置实现。

S3C2410的UART0和UART1使用nRTS和nCTS信号支持自动流控制，在这种情况下，它可以连接另一个UART设备。如果用户希望将UART连接到MODEM，则需要通过软件来禁止UMCONn寄存器中的自动流控制位并控制nRTS信号。S3C2410的UART2 不支持自动流控制功能。

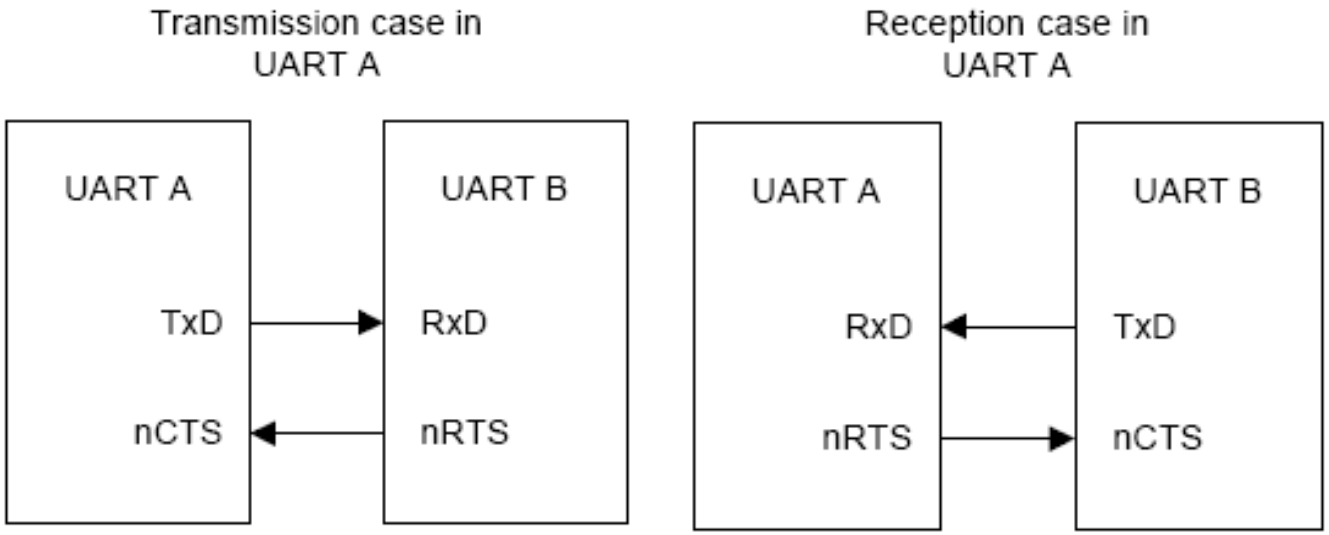


图 **UART AFC**接口

在AFC状态，nRTS根据接收器的状态和nCTS信号控制发送器的操作。

UART AFC发送侧：UART发送器仅当nCTS信号是有效时(在AFC状态，nCTS有效表示其他UART的FIFO已经准备好接收数据)，UART的发送器才发送在FIFO中的数据。

UART AFC接收侧：在UART接收数据之前，当其接收FIFO具有多余2字节的空闲空间时，nRTS有效；如果其接收FIFO的空闲空间少于1字节，则nRTS无效(在AFC状态，nRTS指示它自己的接收FIFO已经准备好接收数据)。

（5）使用FIFO进行收发

通过对FIFO状态寄存器UFSTATn的查询，确定进行收发。

使用FIFO进行发送：

① 选择发送模式（中断或者DMA模式）

② 查询对方是否有请求发送要求，由MODEM状态寄存器UMSTATn[0]给出，该位为1，则有请求，再查询FIFO状态寄存器UFSTATn的数据满状态位是否为1，如果不是1，可以向发送缓冲寄存器UTXHn写入发送的数据。上面二者有一个或者两个都不满足，则不发送数据。

使用FIFO进行接收（请求发送）：

① 选择接收模式（中断或者DMA模式）

② 请求发送。先要查询FIFO状态寄存器UFSTATn的数据满状态位是否为1，如果不是1，则可以向对方发出“请求发送信号”，对MODEM控制寄存器UMCONn中的请求发送信号产生位置1，使UARTn发出nRTS信号；如果UFSTATn的数据满状态位是1，则不能够请求发送数据。

（6）不使用FIFO进行收发

通过对收/发状态寄存器UTRSTATn的查询，确定进行收发。

数据发送：

① 选择发送模式(中断或者DMA模式)

② 查询对方是否有请求发送要求，由MODEM状态寄存器UMSTATn[0]给出，该位为1，则有请求，再查询发送/接收状态寄存器UTRSTATn[1]的“发送缓冲器空”状态位是否为1，如果是1，可以向发送缓冲寄存器UTXHn写入发送的数据。

数据接收(请求发送)：

① 选择接收模式（中断或者DMA模式）

② 请求发送。先要查询发送/接收状态寄存器UTRSTATn[0]的接收缓冲器“数据就绪状态位”是否为1，如果是1，需要先读取数据，然后再请求对方发送数据，方法是对MODEM控制寄存器UMCONn[0]的请求发送信号产生位置1，使UARTn发出nRTS信号。如果UFSTATn的数据满状态位是1，则不能够请求发送数据。

（7）中断或DMA请求

中断或DMA请求分为3类：接收中断请求、发送中断请求、错误中断请求，共7种事件：溢出错误、奇偶校验错误、帧格式错误、通信中断信号、接收缓冲器数据就绪、发送缓冲器空、发送移位器空。

① 接收中断：

•非FIFO模式：当接收缓冲寄存器收到数据后，产生中断请求。

•FIFO模式：Rx FIFO中数据的数目达到了触发中断的水平，或者超时（在三帧时间内未收到任何数据），均产生中断请求。

② 发送中断：

•非FIFO模式：当发送缓冲器空时，产生中断请求。

•FIFO模式：Tx FIFO中数据的数目达到了触发中断的水平，产生中断请求。

③错误中断：

有4种错误中断：溢出错误、奇偶检验错误、帧格式错误、通信中断信号错误。

•非FIFO模式：只要有任何一个错误出现，就会产生中断请求。

•FIFO模式：Rx FIFO中数据溢出，或者出现了帧格式错误、奇偶校验错误、传输中断信号错误，都会产生中断请求。

说明：

1)奇偶校验错误、帧格式错误、通信中断信号错误，在数据接收时就产生了，在读出错误数据时才出现中断请求。

2)如果设置的是DMA模式，以上所出现的中断请求，应该替换为DMA请求。

3)通信中断的含义：RxDn输入线上保持逻辑“0”状态的持续时间超过一个数据帧传输的时间。

（8）回环模式(Loopback Mode)

S3C2410的每个UART都提供有检测功能，它是一种数据循环流动的自发、自收方式，数据从发送缓冲器传送到TxD，数据不经过引脚输出，在内部将数据传到接收引脚RxD，再传输到接收缓冲器。该模式通过设置UART控制寄存器(UCONn)的回送位来进行选择。

（9）红外模式（Infra-Red （IR） Mode）

S3C2410的UART模块支持红外发送和接收，该模式可以通过设置UART线路控制寄存器（ULCONn）中的红外模式位来选择。IR发送时，数据中的逻辑‘1’不发送脉冲，逻辑‘0’发送脉冲，发送的脉冲宽度是通常串口发送数据位的3/16。IR接收时，接收器检测到通常串口数据位3/16的脉冲宽度作为逻辑‘0’值，检测到没有脉冲作为逻辑‘1’值。

8.2.3 UART专用寄存器

每个UART有11个专用寄存器，标*号的只有UART0和UART1才有

Register	Address	R/W	Description	Reset Value
ULCONn	0x5000x000	R/W	线路控制寄存器	0x00
UCONn	0x5000x004	R/W	控制寄存器	0x00
UFCONn	0x5000x008	R/W	FIFO控制寄存器	0x00
UMCONn	0x5000x00C	R/W	MODEM控制寄存器*	0x00
UTRSTATn	0x5000x010	R	发送/接收状态寄存器	0x6
UERSTATn	0x5000x014	R	Rx错误状态寄存器	0x0
UFSTATn	0x5000x018	R	FIFO状态寄存器	0x00
UMSTATn	0x5000x01C	R	MODEM状态寄存器*	0x0
UTXHn	0x5000x020/23	W	发送缓冲寄存器	-
URXHn	0x5000x024/27	R	接收缓冲寄存器	-
UBRDIVn	0x5000x028	R/W	波特率除数寄存器	-

1. 线路控制寄存器（ULCON）

Register	Address	R/W	Description	Reset Value
ULCON0	0x50000000	R/W	UART0线路控制寄存器	0x00
ULCON1	0x50004000	R/W	UART1线路控制寄存器	0x00
ULCON2	0x50008000	R/W	UART2线路控制寄存器	0x00

字段名	位	意　　义	初值
-	7	保 留	0
Infra-Red-Mode	6	红外模式设置位。0:正常模式；1:红外	0
Parity Mode	5:3	奇偶校验类型。　0xx：不校验； 100：奇校验；　101：偶校验； 110：强制为1；　111：强制为0	000
Num of stop bit	2	停止位个数。　0：1个；1：2个	0
Word Length	1:0	数据位数目。　00：5位； 01：6位；　10：7位；　11：8位	00

2. 控制寄存器（UCON）

Register	Address	R/W	Description	Reset Value
UCON0	0x50000004	R/W	UART0控制寄存器	0x00
UCON1	0x50004004	R/W	UART1控制寄存器	0x00
UCON2	0x50008004	R/W	UART2控制寄存器	0x00

2. 控制寄存器（UCON）

字段名	位	意　　义	初值
Clock Selection	10	波特率时钟源选择。0：PCLK；1：UCLK	0
Tx Int Type	9	发送中断请求类型。0 ：脉冲型；1：电平	0
Rx Int Type	8	接收中断请求类型。0 ：脉冲型；1：电平	0
Rx Time OV Ena	7	接收超时中断控制。0：禁止；1：允许	0
Rx ERR Int Ena	6	接收错误中断控制。0：禁止；1：允许	0
Loopback Mode	5	回环模式控制。0 = 正常操作；1 = 回环模式	0
Send Break Signal	4	发送暂停（中止）信号控制。 0 = 正常传输；1 = 发送暂停信号（全为0）	0
Transmit Mode	3:2	发送/接收模式控制。00：禁止发送/接收；01：中断或查询模式；10：UART0、2用DMA0、DMA3；11：UART1用DMA1	00
Receive Mode	1:0		00

3. FIFO控制寄存器（UFCON）

Register	Address	R/W	Description	Reset Value
UFCON0	0x50000008	R/W	UART0 FIFO控制寄存器	0x00
UFCON1	0x50004008	R/W	UART1 FIFO控制寄存器	0x00
UFCON2	0x50008008	R/W	UART2 FIFO控制寄存器	0x00

3. FIFO控制寄存器(UFCON)

字段名	位	意　　义	初值
Tx FIFO Tri Level	7:6	Tx FIFO的触发条件设置。　　00：空； 01：4字节；10：8字节；11：12字节	00
Rx FIFO Tri Level	5:4	Rx FIFO的触发条件设置。 00：4字节；　01：8字节； 10：12字节；11 ：16字节	00
reserved	3	保　　留	0
Tx FIFO Reset	2	Tx FIFO清除控制。0：正常；1：清零	0
Rx FIFO Reset	1	Rx FIFO清除控制。0：正常；1：清零	0
FIFO Enable	0	FIFO应用控制。 0：禁止；1：使能	0

4. MODEM控制寄存器（UMCON）

Register	Address	R/W	Description	Reset Value
UMCON0	0x5000000C	R/W	UART0 FIFO控制寄存器	0x00
UMCON1	0x5000400C	R/W	UART1 FIFO控制寄存器	0x00
reserved	0x5000800C	-	保留	-

字段名	位	意　义	初值
reserved	7:5	保留(为0)	000
Auto Flow Control (AFC)	4	自动流控制。 0：一般方式；1：自动流控制	0
reserved	3:1	保留(为0)	000
Request to Send	0	nRTS引脚信号控制。　0：nRTS 为高电平；1：nRTS为低电平，有效。	0

5. 发送/接收状态寄存器（UTRSTAT）

Register	Address	R/W	Description	Reset Value
UTRSTAT0	0x50000010	R	UART0状态寄存器	0x06
UTRSTAT1	0x50004010	R	UART1状态寄存器	0x06
UTRSTAT2	0x50008010	R	UART2状态寄存器	0x06

字段名	位	意　义	初值
Transmitter empty	2	发送器空状态位。　0：发送器非空 1：发送器(发送缓冲器和移位器)空。	1
Transmit buffer empty	1	发送缓冲器空状态位。 0：非空；1：空 在非FIFO模式，激发中断或DMA请求	1
Receive buffer data ready	0	接收缓冲器状态位。 0：空；1：有数据 在非FIFO模式，激发中断或DMA请求	0

6. Rx错误状态寄存器（UERSTAT）

Register	Address	R/W	Description	Reset Value
UERSTAT0	0x50000014	R	UART0Rx错误状态寄存器	0x0
UERSTAT1	0x50004014	R	UART1Rx错误状态寄存器	0x0
UERSTAT2	0x50008014	R	UART2Rx错误状态寄存器	0x0

字段名	位	意　义	初值
Break Detect	3	暂停信号状态。　0：无暂停信号； 1：收到暂停信号(产生中断请求)	0
Frame Error	2	帧错误状态位。　0：无帧错误； 1：有帧错误(产生中断请求)	0
Parity Error	1	奇偶校验错误状态。0：无奇偶校验错 1：有奇偶校验错误(产生中断请求)	0
Overrun Error	0	溢出错误状态位。　0：无溢出错误； 1：溢出错误(产生中断请求)	0

7. FIFO状态寄存器（UFSTAT）

Register	Address	R/W	Description	Reset Value
UFSTAT0	0x50000018	R	UART0 FIFO状态寄存器	0x00
UFSTAT1	0x50004018	R	UART1 FIFO状态寄存器	0x00
UFSTAT2	0x50008018	R	UART2 FIFO状态寄存器	0x00

字段名	位	意　义	初值
Reserved	15:10	保留（为0）	0
Tx FIFO Full	9	发送FIFO满状态。 0：未满； 1：满	0
Rx FIFO Full	8	接收FIFO状态位。 0：未满； 1：满	0
Tx FIFO Count	7:4	发送FIFO中数据的数目，字节单位。	0
Rx FIFO Count	3:0	接收FIFO中数据的数目，字节单位。	0

8. MODEM状态寄存器（UMSTAT）

Register	Address	R/W	Description	Reset Value
UMSTAT0	0x5000001C	R	UART0 Modem状态寄存器	0x0
UMSTAT1	0x5000401C	R	UART1 Modem状态寄存器	0x0
Reserved	0x5000801C	R	保留	-

字段名	位	意　义	初值
Reserved	3	保留（为0）	0
Delta CTS	2	nCTS引脚信号自上次CPU读后变化状态。 0：未改变；1：已改变。	0
Reserved	1	保留（为0）	0
Clear to Send	0	nCTS引脚信号状态。0：nCTS为高电平；1：nCTS引脚为低电平，有效。	0

9. 发送缓冲寄存器（UTxH）

Register	Address	R/W	Description	Reset Value
UTxH0	0x50000020(L) 0x50000023(B)	W (byte)	UART0 发送缓冲寄存器	-
UTxH1	0x50004020(L) 0x50004023(B)	W (byte)	UART1发送缓冲寄存器	-
UTxH2	0x50008020(L) 0x50008023(B)	W (byte)	UART2发送缓冲寄存器	-

字段名	位	意　义	初值
Tx DATAn	7:0	UARTn发送的一个字节数据	-

10. 接收缓冲寄存器（URxH）

Register	Address	R/W	Description	Reset Value
URxH0	0x50000024(L) 0x50000027(B)	R (byte)	UART0 接收缓冲寄存器	0x00
URxH1	0x50004024(L) 0x50004027(B)	R (byte)	UART1接收缓冲寄存器	0x00
URxH2	0x50008024(L) 0x50008027(B)	R (byte)	UART2接收缓冲寄存器	0x00

字段名	位	意　义	初值
Rx DATAn	7:0	UARTn接收的一个字节数据。	-

11. 波特率除数寄存器（UBRDIV）

Register	Address	R/W	Description	Reset Value
UBRDIV0	0x50000028	R/W	UART0 波特率除数寄存器	-
UBRDIV1	0x50004028	R/W	UART1 波特率除数寄存器	-
UBRDIV2	0x50008028	R/W	UART2 波特率除数寄存器	-

字段名	位	意　义	初值
UBRDIV	15:0	波特率除数值。UBRDIVn >0	-

8.2.3 S3C2410 UART使用举例

【例】编写一程序，使用S3C2410的UART2进行串行数据收发，要求用脉冲请求中断的方式、使用收/发FIFO，8个数据位、1个停止位、不校验，波特率为125kbps。设PCLK为50MHz。（提示：主程序对UART2初始化、引脚配置、中断初始化等，并进行一次发送；中断服务程序进行数据收发，并且清除中断请求标志和中断服务标志）

解：

（1）计算波特率除数：

由公式：UBRDIVn=(int)（CLK/（ f_B*16））— 1

这里： CLK=PCLK=50MHz， f_B = 125kb/s

计算得：UBRDIVn=25 -1=24

（2）UART2控制寄存器：

线路控制寄存器： ULCON2=0b 0 000 0 11=0x03

含义：非红外、不校验、 1个停止位、 8个数据位

控制寄存器：UCON2=0b 0 0 0 0 1 0 0 01 01=0x05

含义：选PCLK、发/收中断脉冲请求、关闭接收超时中断、允许接收错误中断、不回环、不发送暂停信号、发/收用中断方式。

FIFO控制寄存器：UFCON2=0b 10 01 0 0 0 1=0x91

含义：发/收FIFO选8字节触发、保留位为0、不复位发/收FIFO、使能FIFO。

（3）引脚配置

TxD2、RxD2对应GPH6、GPH7，在GPH配置寄存器GPHCON中的位置为：

0b 1 0 1 0 xx xx xx xx xx xx

方法：GPHCON= GPHCON&~(0xF<<12)|(0xA<<12)

（4）中断寄存器设置

中断模式寄存器：INTMOD&=~(1<<15)

INT_UART2位于第15位，将UART2设置为IRQ中断

中断屏蔽寄存器：INTMSK&=~(1<<15)

中断优先级寄存器PRIORITY：不设置，使用固定优先级。

子中断屏蔽寄存器：INTSUBMSK&=~(7<<6)

INT_ERR2、INT_TXD2、INT_RXD2位于子中断屏蔽寄存器中的8、7、6位。

（5）在中断服务程序中对寄存器的操作

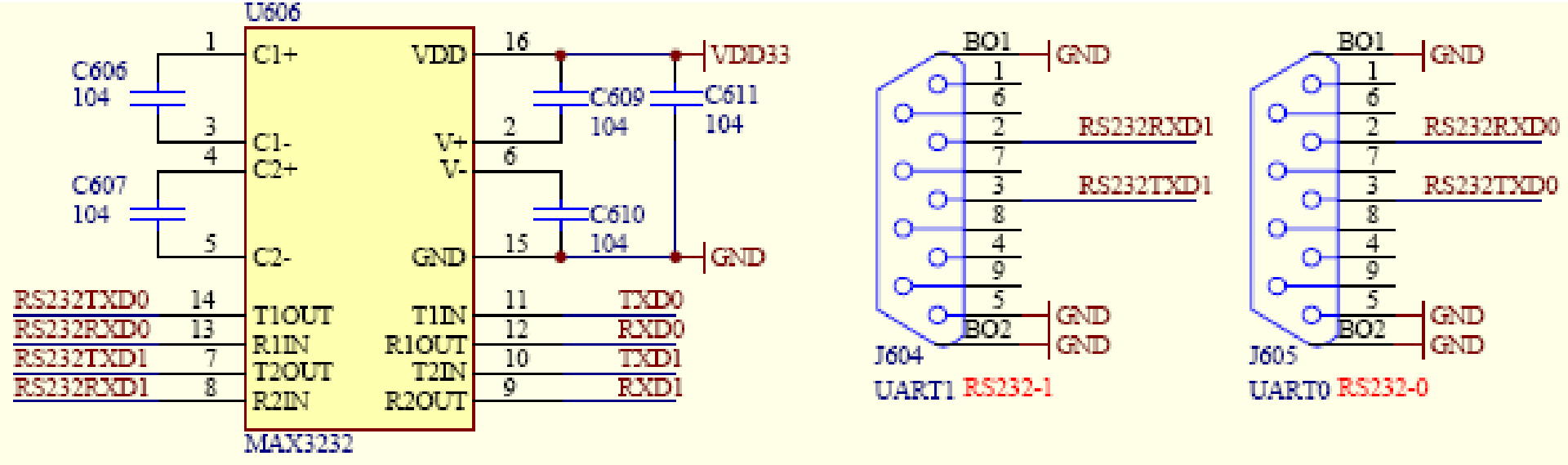
清除中断请求寄存器相应位：SRCPND&=~(1<<15)

清除中断服务寄存器相应位：INTPND&=~(1<<15)

（6）程序（略）

8.3 串行通信举例

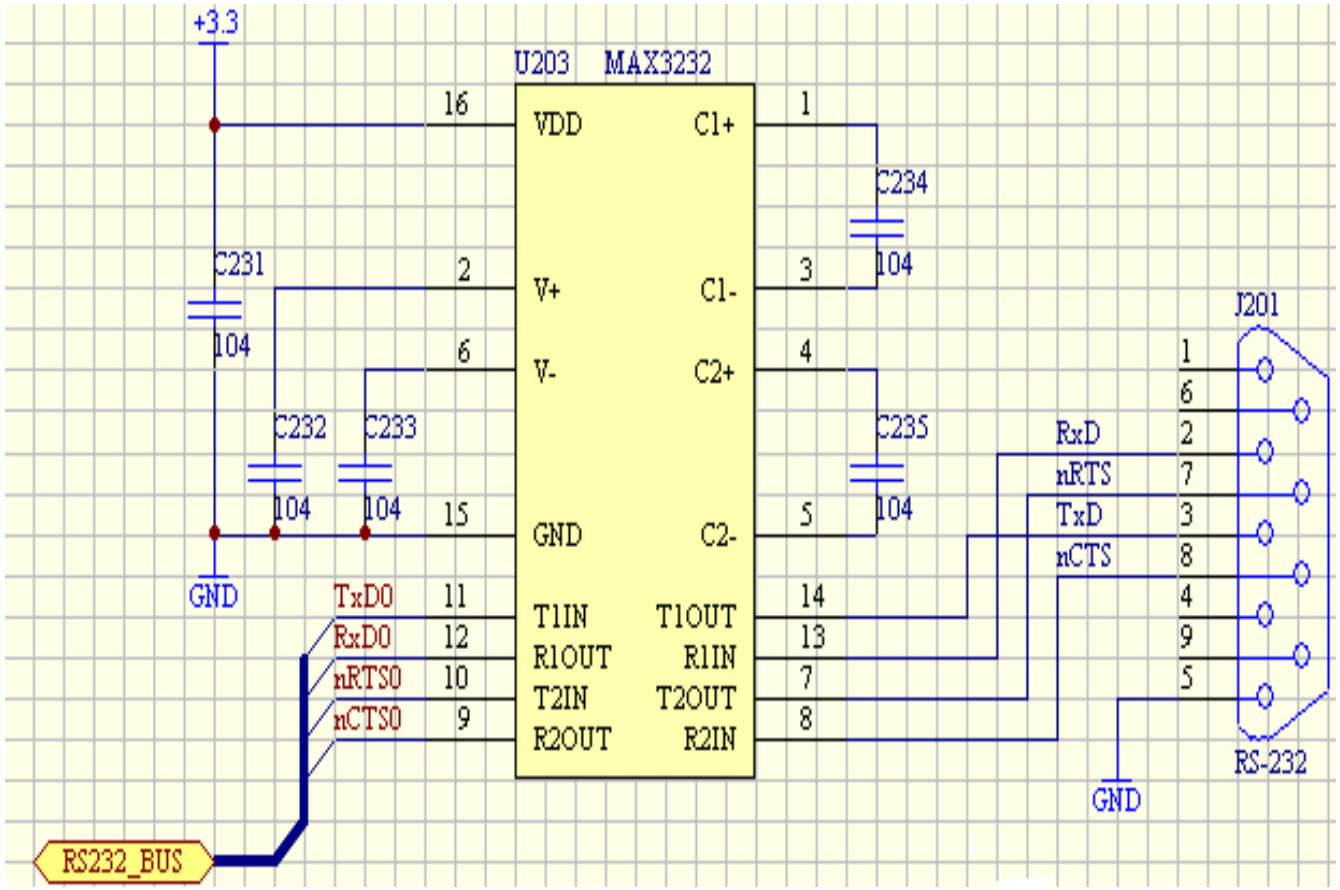
- RS-232C接口设计
- MAX3232实现2路3线串口电平转换



T1IN/T2IN：TTL/CMOS INPUT；　**R1OUT/R2OUT**：TTL/CMOS OUTPUT

R1IN/R2IN：RS-232 INPUT；　**T1OUT/T2OUT**：RS-232 OUTPUT

- RS-232C接口设计
- MAX3232实现1路5线串口电平转换



上图是一个1路5线的RS-232接口电路。电路中所采用的电平转换电路芯片为MAX3232，S3C2410芯片的UART0相关引脚（即：TxD0、RxD0、nRTS0、nCTS0）经过MAX3232电平转换后连接到DB9型的插座上。这样就可以使用S3C2410芯片内部的UART0部件来控制符合RS-232标准的串行通信。

- RS-232C接口设计--串口初始化

【例8.1】以UART0为例，说明串口初始化函数。该函数虽然设计成通用的串口初始化函数，即对UART0、UART1、UART2均可以初始化，但本例子中只给出UART0和UART1初始化对应的代码，对于UART2的初始化代码未给出。与UART1初始化代码类似，只不过涉及的寄存器不同而已。

```
/*********************************************
    功能：初始化串口(不使用FIFO，不产生接收错误中断)
    参数：
Uartnum：选择UART0/UART1/UART2(0/1/2)
          parity: 选择奇偶校验方式(0：无校验，4：奇校验，5：偶校验)
          stop:   选择停止位(0：1位停止位，1：2位停止位)
          data:   选择数据位(0：5位，1：6位，2：7位，3：8位)
          baud: 波特率
*********************************************/
```

```
void rs232_Init(INT8U com ,INT32U parity ,INT32U stop ,INT32U data ,int baud)
{
 if(Uartnum ==0)                          //初始化UART0
 {
   rGPHCON=(rGPHCON&0xffffff00)|0xaa;    //设置GPH端口为UART
   rUFCON0=0x0;                          //不使用FIFO
   rUMCON0=0x0;                           //禁止自动流控制
   rULCON0=(parity<<3)|(stop<<2)|(data);
    //8位数据位，1位停止位，奇偶校验位，不采用红外线传输模式
   rUCON0=0x205; //当Tx缓冲为空时，以电平信号触发发送中断请求
         //当Rx缓冲有数据时，以脉冲信号触发接收中断请求
         //禁止超时中断，禁止产生接收出错状态的中断请求
         //禁止回环模式，禁止发送暂停信号，发送数据操作按中断方式
         //接收数据操作按中断方式
   rUBRDIV0=（int）（PCLK/（baud*16））-1 ; //设置波特率
   rINTMSK=rINTMSK&(~(BIT_GLOBAL|BIT_URXD0));  //开中断
   pISR_URXD0=(int)rxCharDone_0;              //设置中断入口
}
```

```
else if (Uartnum ==1)            //UART1的初始化
     {
      rUFCON1=0x0;
      rUMCON1=0x0;
      rULCON1=(parity<<3)|(stop<<2)|(data);
      rUCON1=0x205;
      rUBRDIV0=（int）（PCLK/（baud*16））-1 ;
     }
    else
     {
         // UART2的初始化与UART1类似，略
     }
}
```

发送/接收程序举例

例8.2 发送和接收程序

```
/*****************************************
    功能：字符发送
    参数：
     Uartnum：选择UART0/UART1/UART2(0/1/2)
          data：要发送的字符
 *****************************************/
/*字符发送程序Uart_SendByte*/
#define WrUTXH0 (ch) ( * (volatile unsigned char * ) 0x50000020)=(unsigned char)(ch)
#define WrUTXH1 (ch) ( * (volatile unsigned char *) 0x50004020)=(unsigned char)(ch)
#define WrUTXH2 (ch) ( * (volatile unsigned char *) 0x50008020)=(unsigned char)(ch)
```

```
void Uart_SendByte (INT8U Uartnum, INT8U data)
{
 if (Uartnum ==0)                    //UART0
 {
  while ((rUTRSTAT0 & 0x2));
        //当发送数据缓冲区不空，执行下一条语句
  Delay(2);
  WrUTXH0 (data);
 }
 else if (Uartnum ==1)
 {
  while ((rUTRSTAT1 & 0x2));                  //UART1
  Delay(2);
  WrUTXH1 (data);
 }
```

```
else if (Uartnum ==2)
 {
   while ((rUTRSTAT2 & 0x2))；                    //UART2
   Delay(2)；
   WrUTXH2 (data)；
 }
}
```

```
/*********************************************************
    功能：字符接收
    参数：
Uartnum：选择UART0/UART1/UART2(0/1/2)
        *Revdata：接收字符
Uart_GetByte：正确接收
*********************************************************/
/*字符接收程序Uart_GetByte*/
#define RdURXH0 ( ) (*(volatile unsigned char *) 0x50000024)
#define RdURXH1 ( ) (*(volatile unsigned char *) 0x50004024)
#define RdURXH2 ( ) (*(volatile unsigned char *) 0x50008024)
```

```
char Uart_GetByte(char* Revdata，int Uartnum)
{
 int i=0;
 if (Uartnum ==0)                    //UART0
  {
   while((rUTRSTAT &0x1))；           //读接收数据
   *Revdata=RdURXH0( )；
   return TRUE；
  }
  else (Uartnum ==1)                 //UART1
  {
   while((rUTRSTAT1 & 0x1))；
   *Revdate=RdURXH1( )；
   return TRUE；
  }
```

```
else (Uartnum ==2)                    //UART2
 {
   while((rUTRSTAT2 & 0x1));
   *Revdate=RdURXH2( );
   return TRUE;
 }
}
```

课外作业

- 教材第2题/第3题：设fpclk为40MHz，将S3C2410的UART0初始化为波特率115200bps，8位数据位，1位停止位，1为奇校验，不采用流控制，请给出各相应的寄存器设置值，并写出初始化程序。