

嵌入式系统

北京邮电大学
计算机学院

戴志涛



北京邮电大学

STM32通用同步异步收发器

USART



同步通信和异步通信

➤ 同步通信:

□ 通信双方根据同步信号进行通信的方式

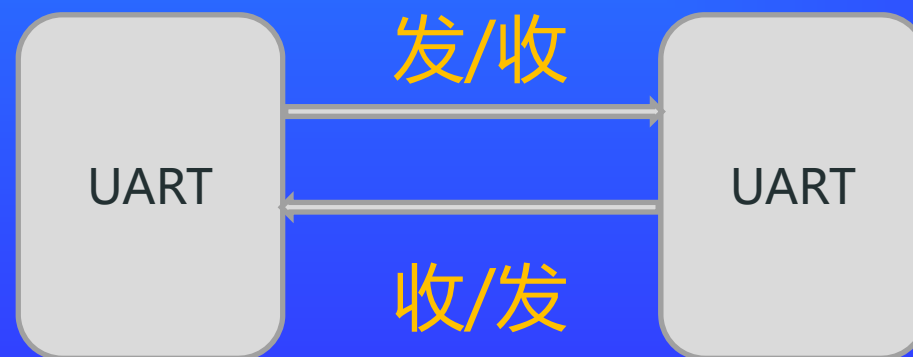
✉ 典型方式：通信双方有一个共同的时钟信号，双方统一规定在时钟信号的上升沿或下降沿对数据线进行采样



同步通信和异步通信

➤ 异步通信:

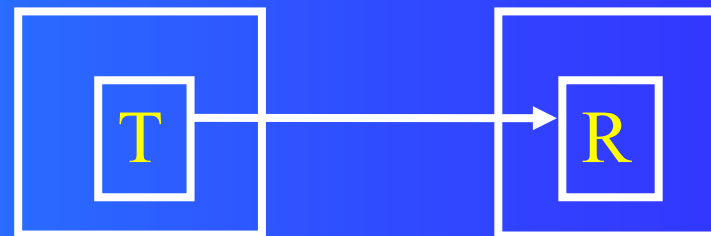
- ❑ 数据传输速度匹配依赖于通信双方自己独立的系统时钟
- ❑ 不需要同步时钟信号，但双方约定好通信的速率，通信的过程中仍需实现同步



单工、半双工、全双工

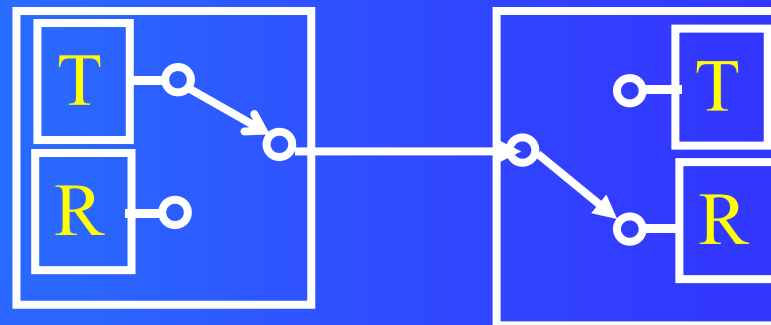
- 单工：单向通信，或收，或发，只能做接收设备或者发送设备

- 例：收音机



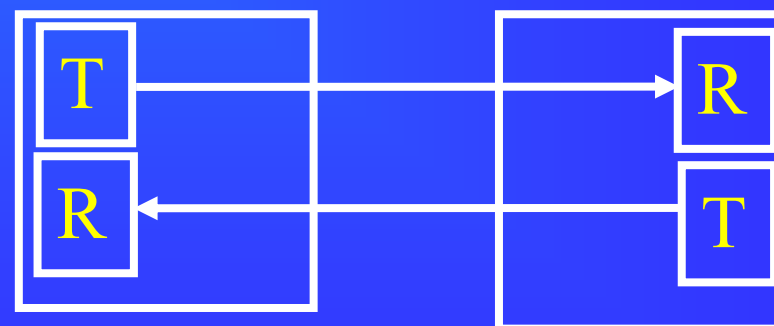
- 半双工：可以收，可以发，但是不能同时收发

- 例：对讲机



- 全双工：可以在同一时刻既接收，又发送

- 例：手机



常见通信协议



通信标准	引脚	通信方式	通信方向
单总线 (1-wire)	DQ: 发送/接收端	异步通信	半双工
SPI	SCK: 同步时钟 MISO: 主机输入/从机输出 MOSI: 主机输出/从机输入	同步通信	全双工
I ² C	SCL: 同步时钟 SDA: 数据输入/输出端	同步通信	半双工
UART (通用异步收发器)	TXD: 发送端 RXD: 接受端 GND: 公共地	异步通信	全双工



UART与USART



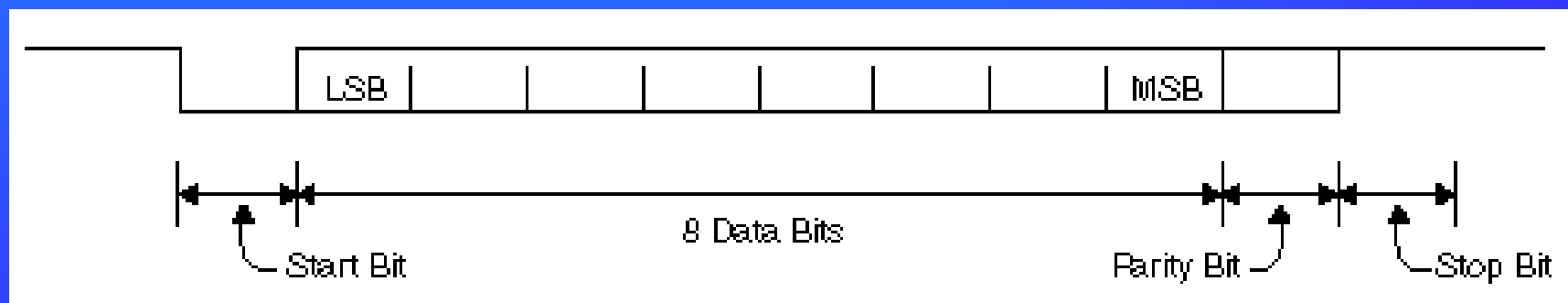
- UART (Universal Asynchronous Receiver/Transmitter)
 - ❑ 通用异步收发器
 - ❑ 一种串行异步通信协议，支持全双工通信
 - ❑ 无需时钟线，速率较低
 - ❑ RS232、RS485等接口协议最常采用的通信格式（经过电平转换器）
- USART (Universal Synchronous Asynchronous Receiver/Transmitter)
 - ❑ 通用同步/异步收发器



UART数据格式

- 起始位 逻辑0 1位
- 数据位 逻辑0或1 5位、6位、7位、8位
- 校验位 逻辑0或1 1位或无
- 停止位 逻辑1 1位、1.5位或2位
- 空闲位 逻辑1 任意数量

- 1 Start Bit, 8 Data Bits, 1 Parity Bit, 1 Stop Bit



字符间异步，字符内部各位间同步



UART协议时序分析

- 空闲位：逻辑“1”状态，表示当前线路上没有数据传送
- 起始位：逻辑“0”的信号，表示传输字符的开始
- 数据位：紧接着起始位之后
 - ❑ 通常为ASCII码
 - ❑ 从最低位开始传输，靠内部时钟定时
- 奇偶校验位：
 - ❑ 数据位加上一位校验位后，“1”的位数应为偶数（偶校验）或奇数（奇校验）
 - ❑ 异步通信的出错率大于同步通信，需要校验位确保数据传输无误
- 停止位：字符结束的标志
 - ❑ 1bit、1.5bit或2bit，高电平



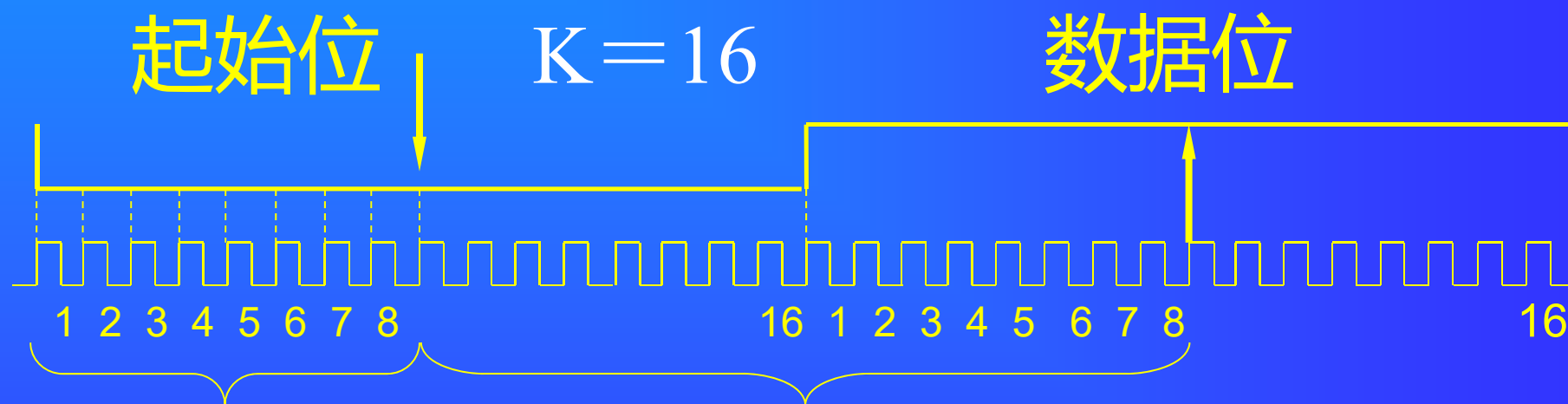
波特率 (Baud rate)

- 波特率：表征数据传输速率的参数
 - ❑ 每秒钟传送的符号数 (symbol)，单位为Baud
- 数据传输速率：单位时间传输的比特数，单位为bit/s或bps (bits per second)
- 如果调制解调器用每个信号变化传输一位，则其数据传输速率和波特率数值相等
- 国际标准波特率系列：110、300、600、1200、1800、2400、4800、9600、115200、14.4Kbps、19.2Kbps.....
- 包括起始位、停止位、校验位等，高于有效数据率
- 在异步通信中，波特率发生器产生的时钟频率不是波特率时钟频率，而是波特率时钟频率的8、16或32倍
 - ❑ 为在接收时进行精确的采样，以提取异步的串行数据



数据位的检测

- T_d : 数据位的位时间
- T_c : 接收器的采样时钟周期
- $K = T_d / T_c$: 波特率系数 (波特率因子)

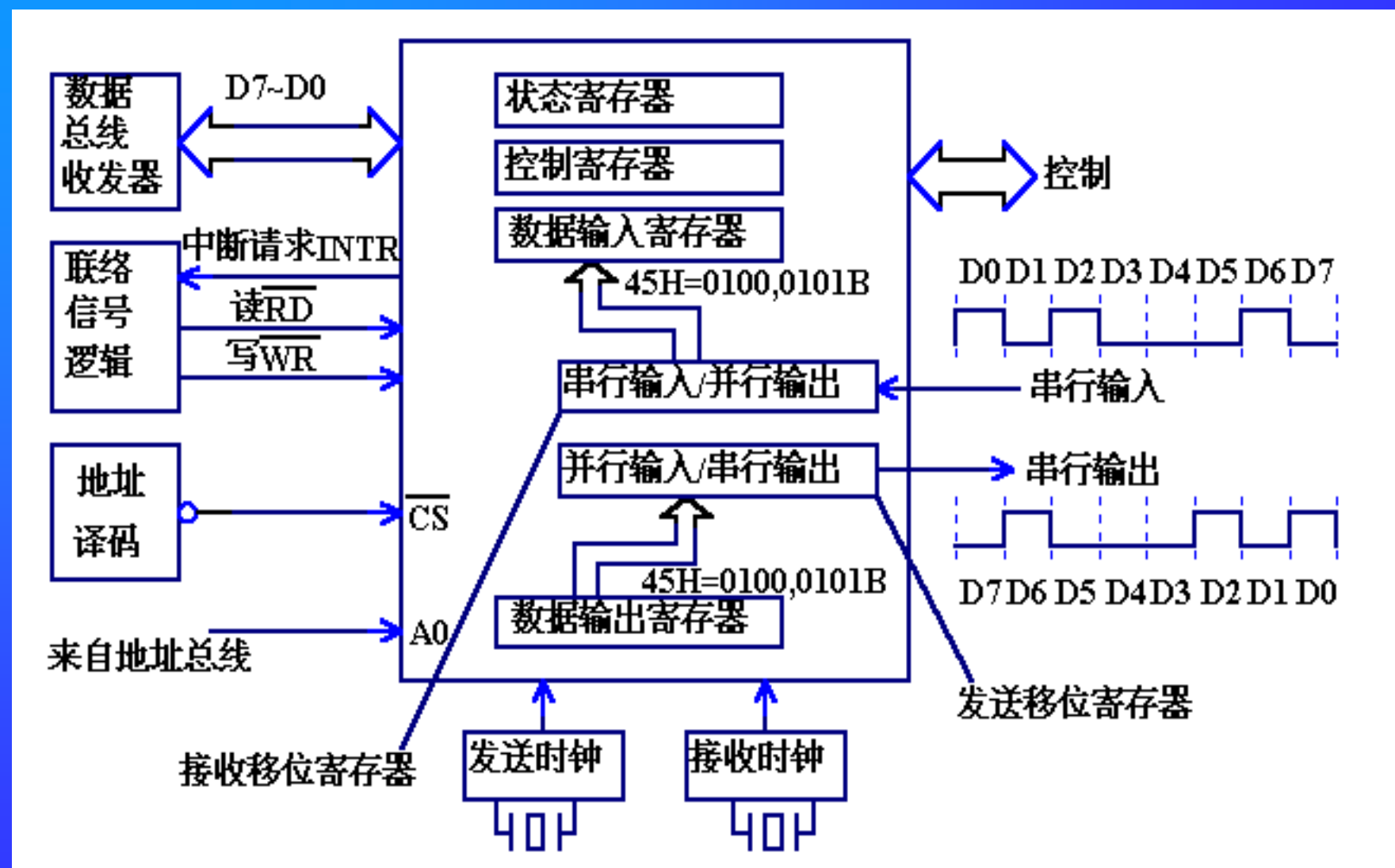


8次采样为0
确认为起始位

以后每隔16个脉冲
采样一次数据位

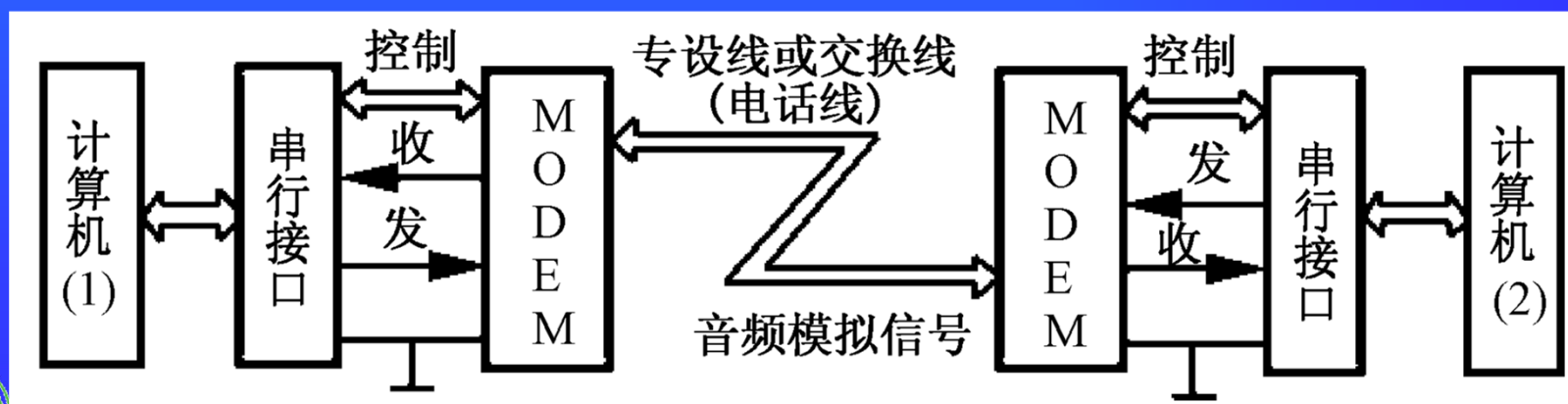


UART控制器简化方框图



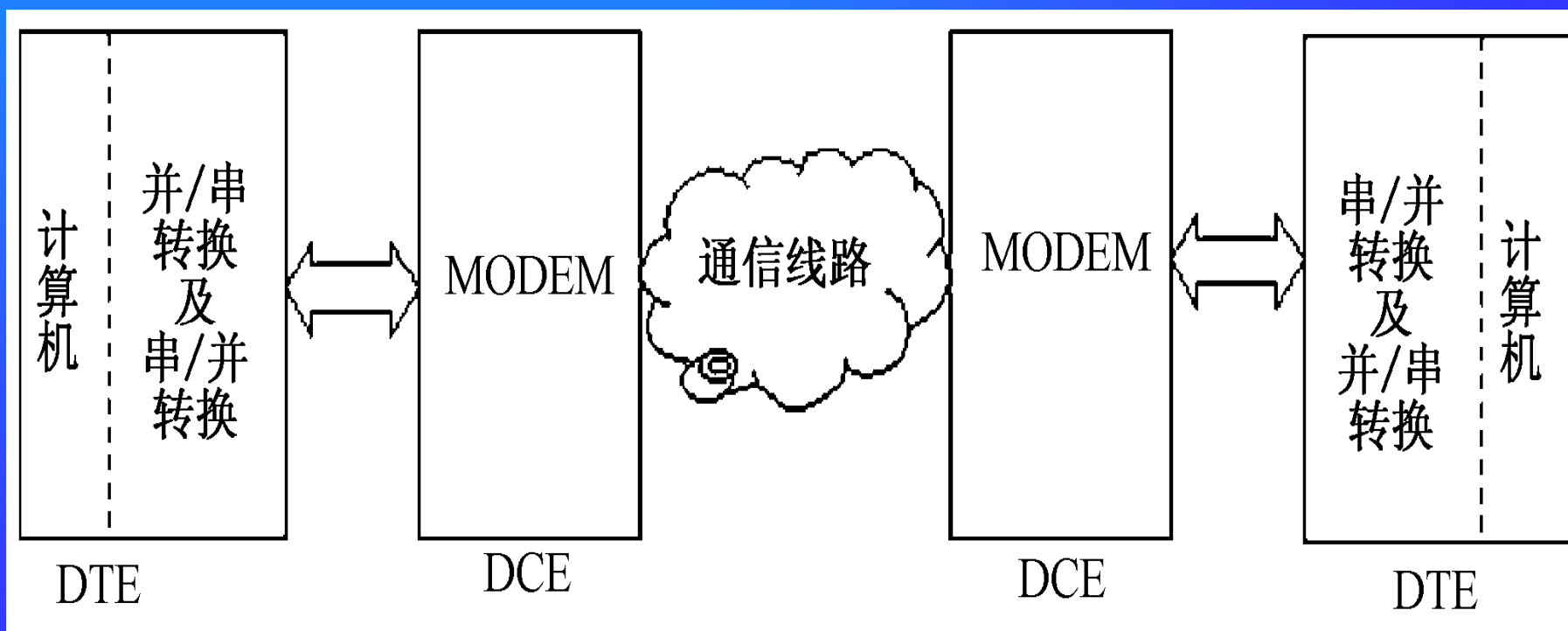
信号调制解调

- 如果数字信号直接在传输线上传送，高次谐波的衰减会很厉害，从而使信号到达接收端后将发生严重畸变和失真
- 解决办法：
 - ❑ 发送方使用调制器（Modulator），把要传送的数字信号调制转换为适合在线路上传输的音频模拟信号
 - ❑ 接收方则使用解调器（Demodulator）从线路上检测出模拟信号，并还原成数字信号



串行通信系统

- 数据终端设备DTE——数据源和目的地
- 数据通信设备DCE——使数据符合线路要求的设备



RS-232接口信号定义



符号	信号方向	信号功能解释
PG	-	电源地 (保护地) Power Ground
TXD	DTE->DCE	发送数据 Transmit Data
RXD	DTE<-DCE	接收数据 Receive Data
RTS	DTE->DCE	请求发送 Require to Send
CTS	DTE<-DCE	清除请求 Clear to Send
DSR	DTE<-DCE	数据装置就绪 Data Set Ready
GND	GND	信号地 Ground
DCD	DTE<-DCE	接收线信号检测 Data Carry Detected
DTR	DTE->DCE	数据终端准备好 Data Terminal Ready
RI	DTE<-DCE	振铃指示 Ring Indicator



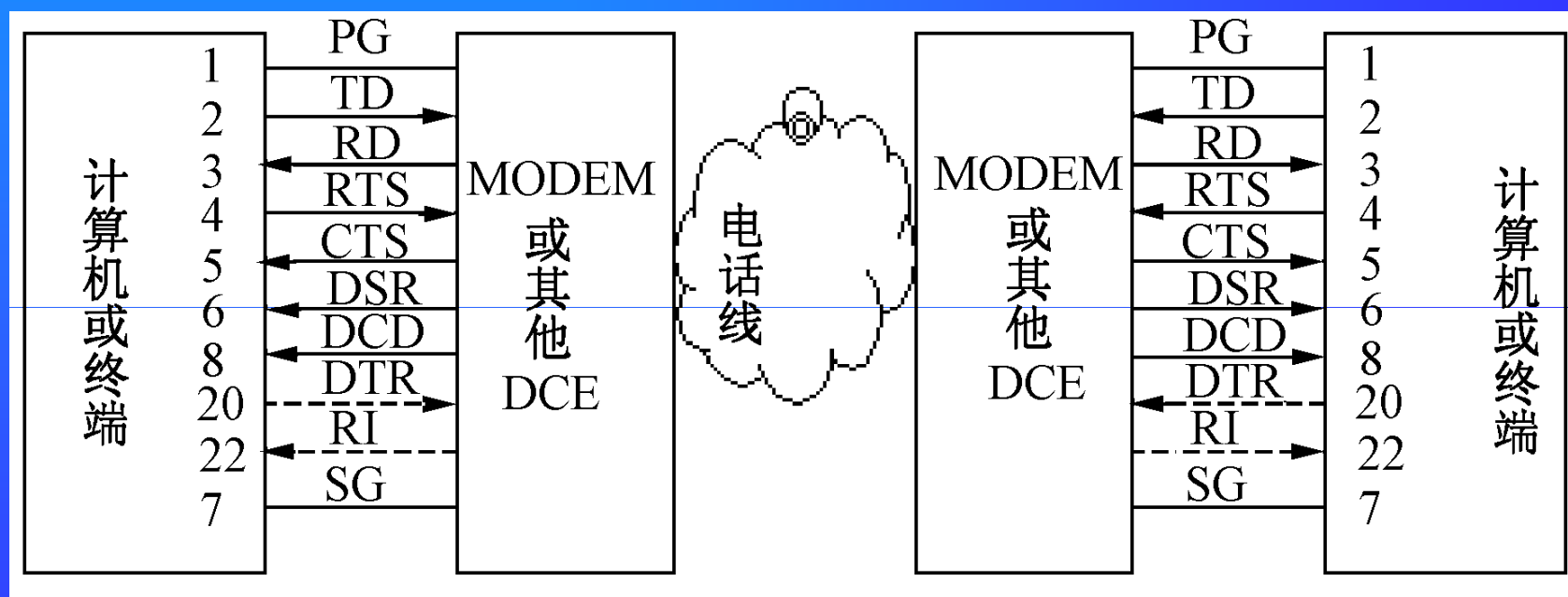
串行通信接口标准RS-232C信号线

引脚	代号	其他表示法	信号名	方向
1	AA (101)	PG	保护地	设备地
2	BA (103)	TxD, SD	发送数据	DTE→DCE
3	BB (104)	RxD	接收数据	DCE→DTE
4	CA (105)	RTS, RS	请求发送	DTE→DCE
5	CB (106)	CTS, CS	允许/清除发送	DCE→DTE
6	CC (108)	DSR, MR	DCE就绪	DCE→DTE
7	AB (102)	SG	信号地	信号公共地
8	CF (109)	RLSD, DCD	接收线路信号检测	DCE→DTE
20	CD (108.2)	DTR	DTE就绪	DTE→DCE
22	CE (125)	RI	振铃指示	DCE→DTE



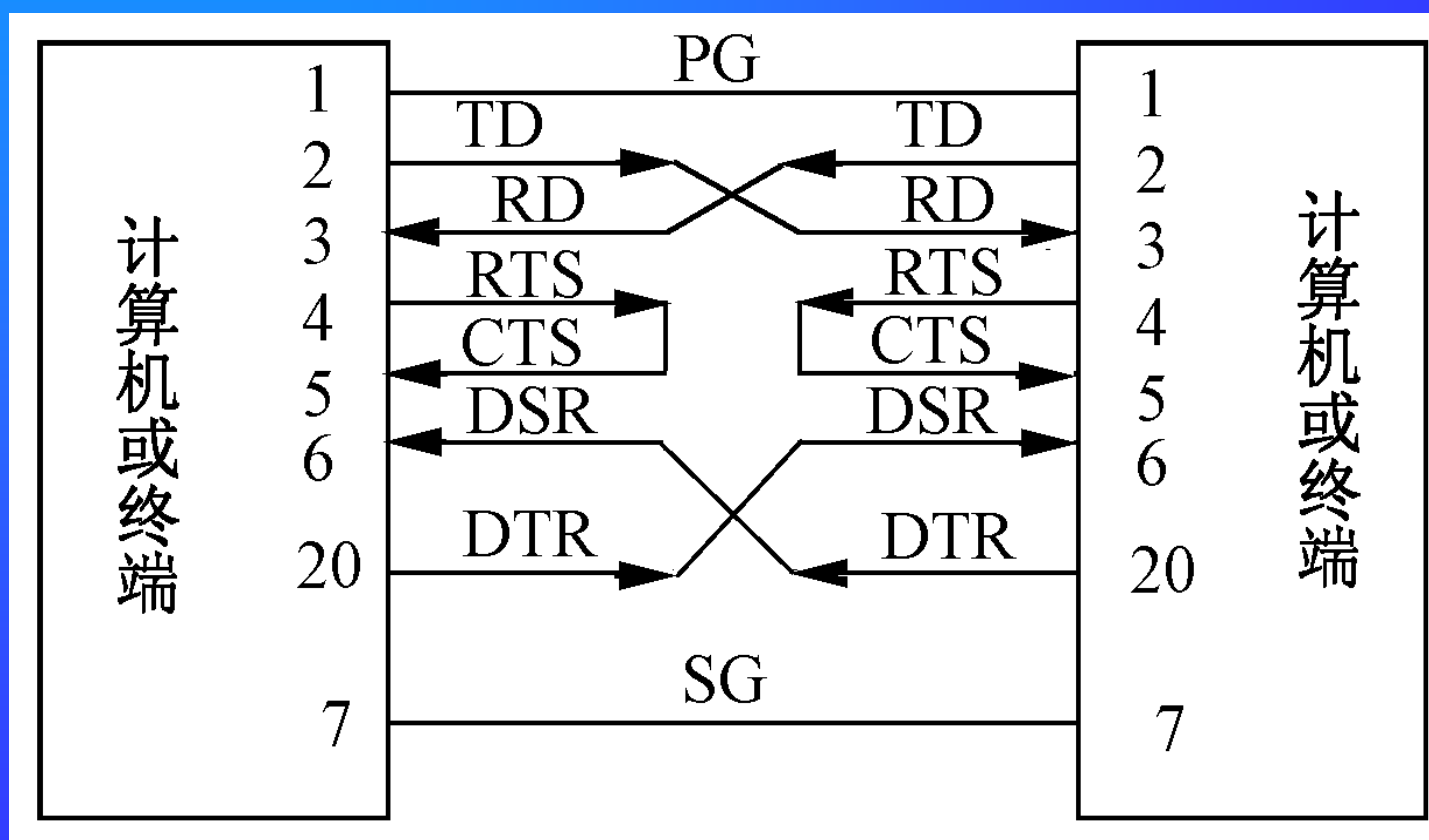
RS-232接口连接方式

➤ 使用MODEM



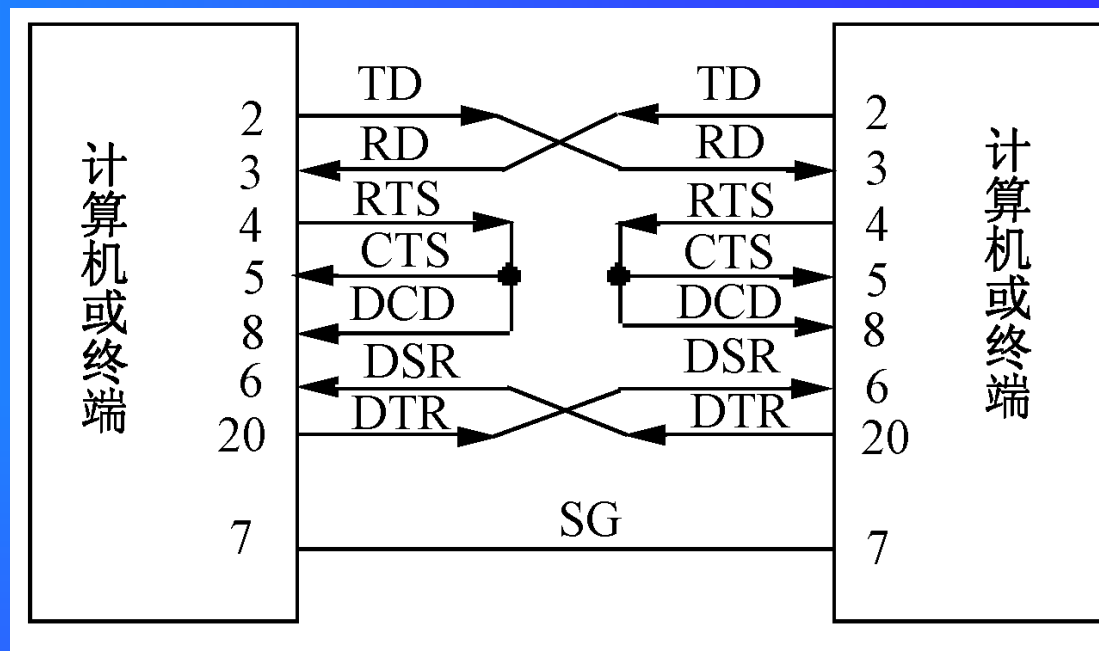
RS-232接口连接方式

➤ 终端直连

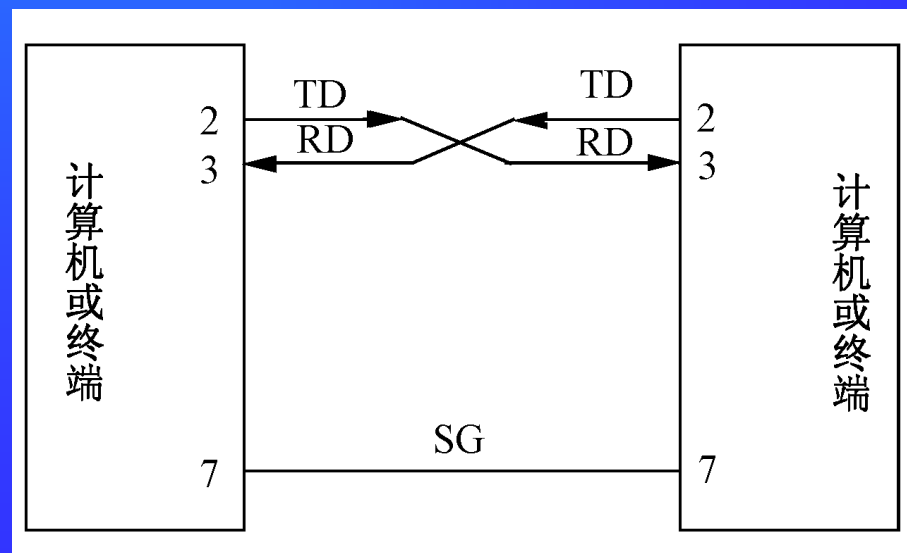


RS-232接口连接方式

➤ 简单连接



➤ 三线连接



RS-232接口电气特性



➤ 电平在 $\pm (3\sim 15)$ V之间

- 负逻辑

- 数据:

 - ☒ 逻辑“1”: 电平低于-3V

 - ☒ 逻辑“0”: 电平高于+3V

- 控制信号:

 - ☒ 接通状态 (ON) (信号有效): 电平高于+3V

 - ☒ 断开状态 (OUT) (信号无效): 电平低于-3V

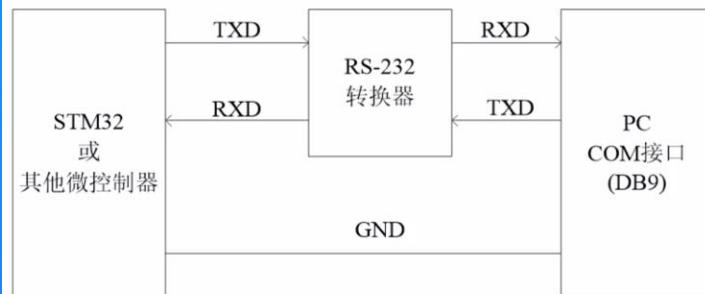
➤ 为实现与TTL/LVTTL电路的连接, 须进行电平转换

- 电平转换芯片实例: MAX232



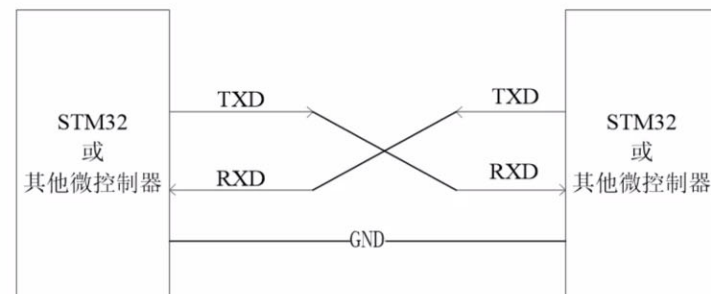
串口电路连接

TTL电平

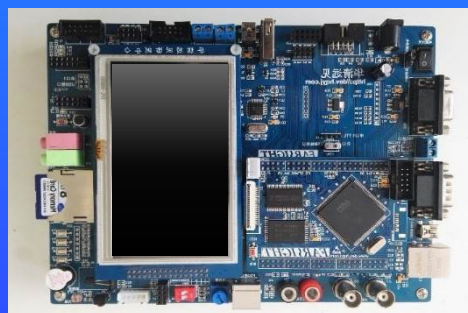


开发板串口与电脑相连

EIA电平



开发板串口与其它开发板串口相连

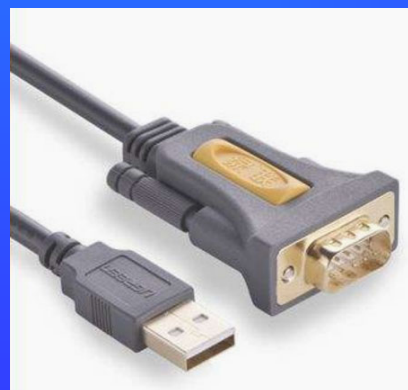
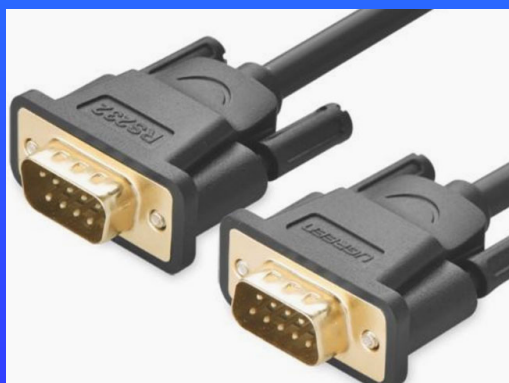
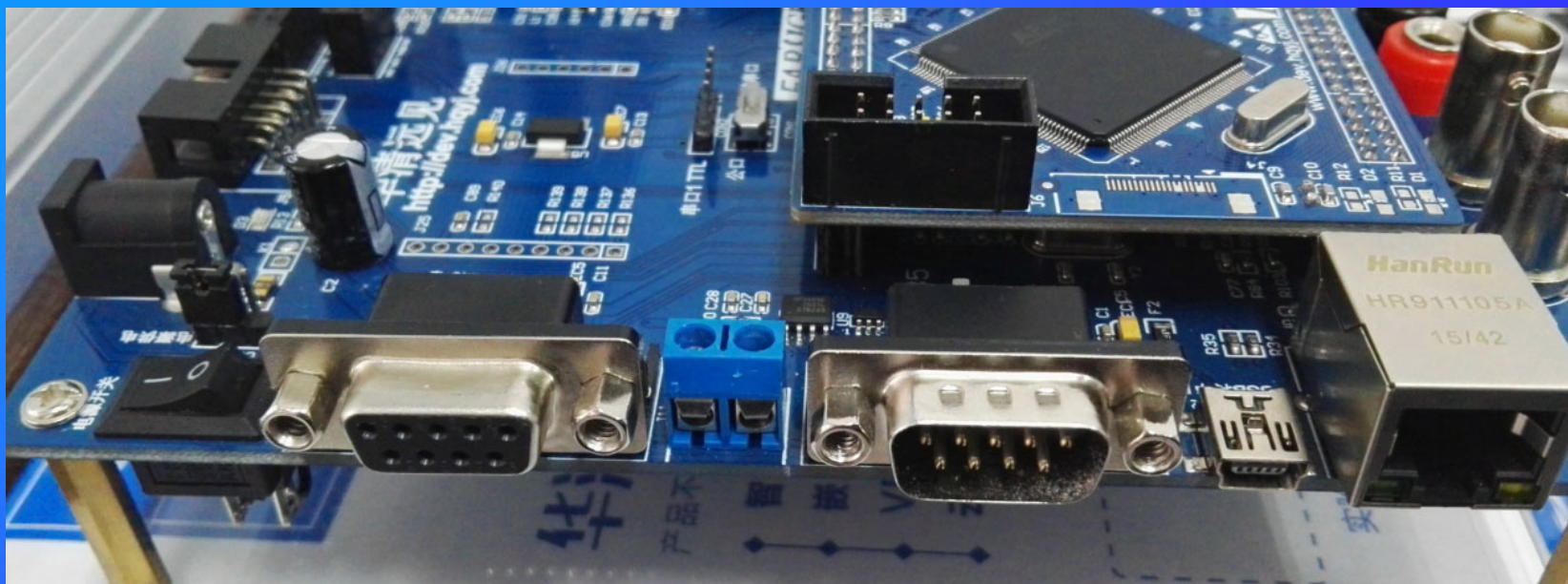


RS232



北京邮电大学

串口硬件连接



STM32 USART Features

- NRZ标准格式 (Mark/Space) 传号/空号, 1/0
- 可编程数据字长 (8或9比特)
- 可配置停止位长度: 0.5, 1, 1.5 or 2 stop bits
- 校验位发生和检测: Even, odd or no-parity
- 可配置的过采样 (oversampling) 方式:
 - ❑ 8倍过采样: 可支持更高速度 (高达 $f_{PCLK}/8$)
 - ⊗ 接收器对时钟容差 (clock tolerance) 的最大容差将会降低
 - ❑ 16倍过采样: 增加接收器时钟容差
 - ⊗ 最大速度限制为最高 $f_{PCLK}/16$
 - ❑ 在速度和时钟容差之间保持平衡



STM32 USART Features

- 分数波特率发生器
- 时钟频率为72MHz且8倍过采样时，通信速率高达9Mbps
- 支持Tx/Rx管脚有效电平翻转 (active level inversion) 及二进制数据翻转
- 支持硬件流控 (CTS and RTS)
- 专用发送和接收标志 (TxE and RxNE)
 - ❑ 支持中断方式
- 支持DMA (Receive DMA request/Transmit DMA request)



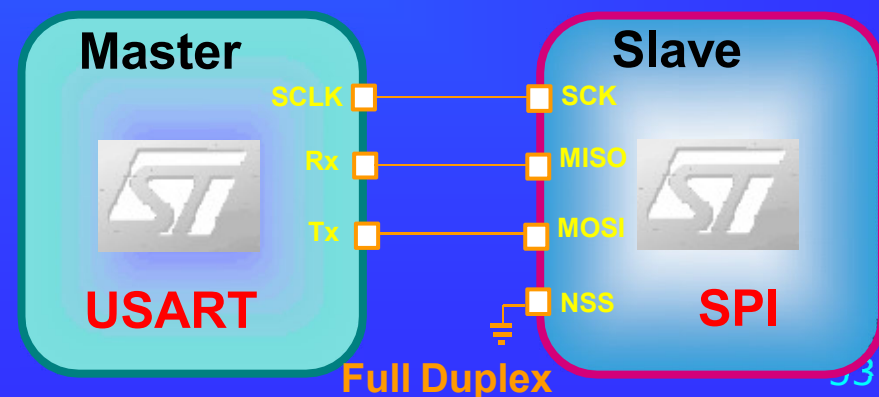
STM32 USART Features

- LIN Master compatible
- Synchronous Mode: Master mode only
- IrDA SIR Encoder Decoder
- Smartcard Capability $T = 0$, $T = 1$
(using the Address/character match, End of block, receiver timeout etc...)
- Single wire Half Duplex Communication



Synchronous Mode

- USART supports Full duplex synchronous communication mode
 - ❑ Full-duplex, three-wire synchronous transfer
 - ❑ USART Master mode only
 - ❑ Programmable clock polarity (CPOL) and phase (CPHA)
 - ❑ Programmable Last Bit Clock Pulse (LBCL) generation
 - ❑ Transmitter Clock output (SCLK)



STM32的USART模块



➤ STM32F407: 四个USART和两个UART

□ USART1和USART6的时钟来源于APB2总线时钟

✉ 最高频率为84MHz

□ USART2、USART3、UART4、UART5的时钟来源于APB1总线时钟

✉ 最高频率为42MHz

	APB2(最高 84MHz)		APB1(最高 42MHz)			
	USART1	USART6	USART2	USART3	UART4	UART5
TX	PA9/PB6	PC6/PG14	PA2/PD5	PB10/PD8 /PC10	PA0/PC10	PC12
RX	PA10/PB7	PC7/PG9	PA3/PD6	PB11/PD9 /PC11	PA1/PC11	PD2
SCLK	PA8	PG7/PC8	PA4/PD7	PB12/PD10 /PC12	-	-
nCTS	PA11	PG13/PG15	PA0/PD3	PB13/PD11	-	-
nRTS	PA12	PG8/PG12	PA1/PD4	PB14/PD12	-	-



STM32的USART引脚



- RX: 接收数据引脚
- TX: 发送数据引脚
- CK (SCLK): 发送时钟输出
 - ❑ 用于控制有移位寄存器的设备
 - ❑ 时钟极性和相位可编程
- nRTS:
 - ❑ 硬件流控引脚, 输出
 - ❑ 低电平时, 表示USART准备好接收数据
- nCTS:
 - ❑ 硬件流控引脚, 输入
 - ❑ 高电平时, 表示当前数据传输结束后应停止数据发送



流控



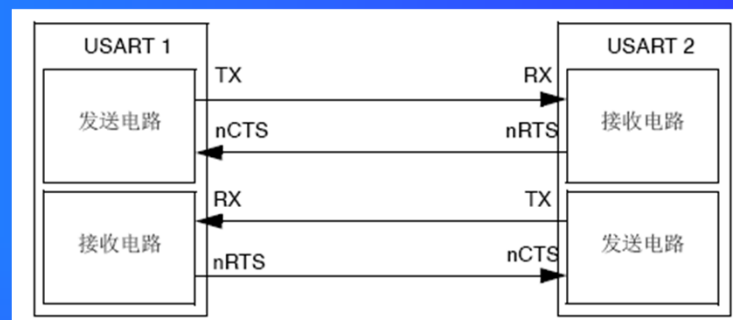
- 当通信数据量较大、速度较快的时候，如果没有流控机制，收方或发方可能会出现数据丢失的现象

- 硬件流控

- ❑ 管脚：CTS和RTS

- ❑ 硬件提供用于流量情况指示的硬件连线

- ❑ 流控本身靠软件实现



- 软件流控

- ❑ 不需要额外的连线，只需要在TxD和Rx D数据通道上发送特殊的流量控制字符



USART mode configuration

USART modes	USART 1	USART 2	USART 3	UART4	UART5	USART 6
Asynchronous mode	X	X	X	X	X	X
Hardware flow control	X	X	X	NA	NA	X
Multibuffer communication (DMA)	X	X	X	X	X	X
Multiprocessor communication	X	X	X	X	X	X
Synchronous	X	X	X	NA	NA	X
Smartcard	X	X	X	NA	NA	X
Half-duplex (single-wire mode)	X	X	X	X	X	X
IrDA	X	X	X	X	X	X
LIN	X	X	X	X	X	X

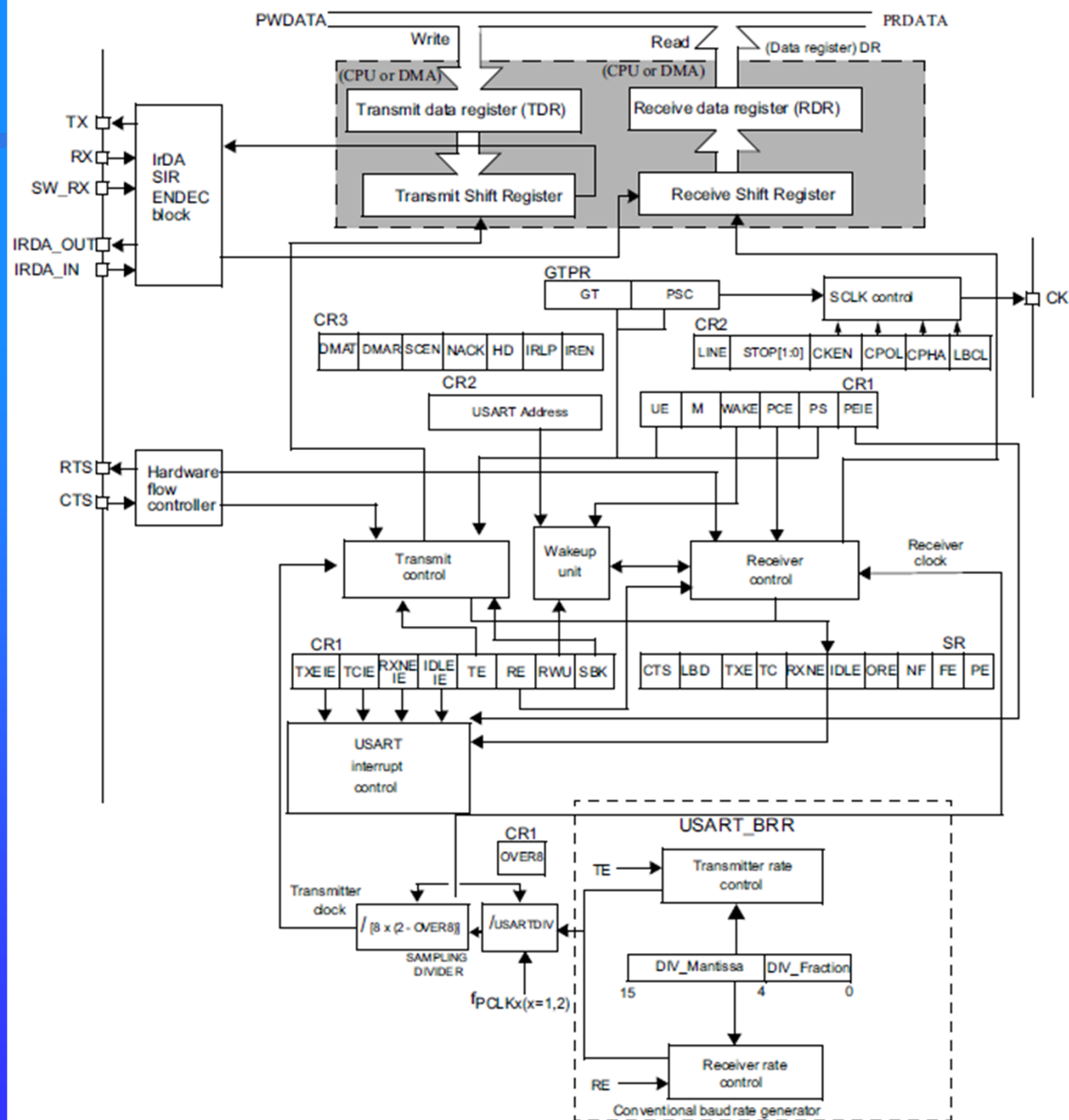
1. X = supported; NA = not applicable.



USART block diagram



北京邮电大学



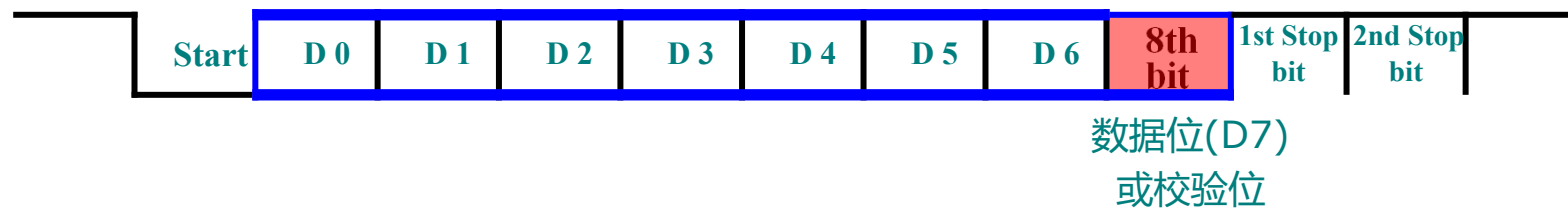
$$\text{USARTDIV} = \text{DIV_Mantissa} + (\text{DIV_Fraction} / 8 \times (2 - \text{OVER8}))$$

STM32 USART数据帧格式

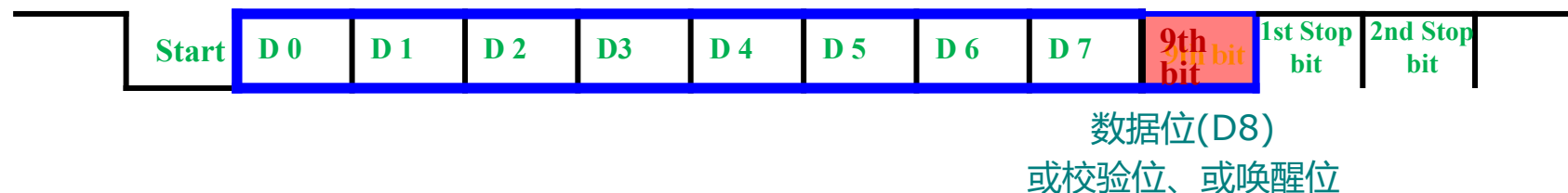
➤ USART数据帧的组成

- ❑ 起始位 (1位)
- ❑ 数据字 (8/9位, M@USART_CR1)
 - ✉ 其中包含了校验位 (0/1位, PCE@USART_CR1)
- ❑ 停止位 (0.5/1/1.5/2位, STOP@USART_CR2)

➤ 8位数据帧



➤ 9位数据帧



USART两种特殊帧

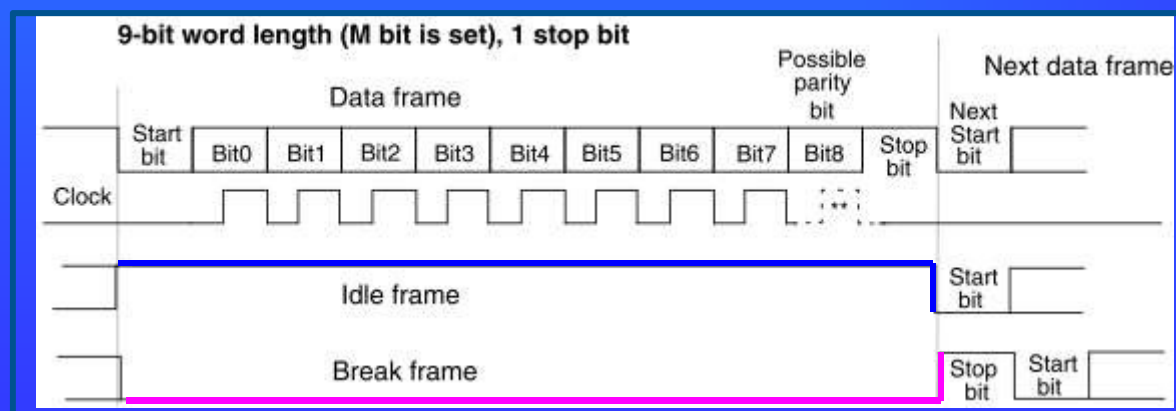
➤ 空闲帧 (Idle frame)

- ❑ 从起始位到停止位全部是高电平
- ❑ 之后跟随一个有效的数据帧的起始位

➤ 断缺 (休息) 帧 (Break frame)

- ❑ 从起始位到停止位全部是低电平
- ❑ 之后再发送1位或2位的停止位

➤ 例：9位字符长度、1位停止位配置下的特殊帧



校验控制



➤ 使能控制: PCE@USART_CR1

❑ 发送方产生校验位

☒ 一旦使能, 发送数据的msb时会附加校验位

❑ 接收方核对校验位

☒ 若核对失败则置位PE@USART_SR, 并可产生中断请求

➤ 奇偶校验: PS@USART_CR1

❑ 偶校验: 使得数据和校验位中“1”的个数为偶数

❑ 奇校验: 使得数据和校验位中“1”的个数为奇数

➤ 校验位包含在字长度(8或9位)中

字长控制M@CR1	校验使能控制PCE@CR1	USART帧格式
0 (8位数据帧)	0	起始位+8位数据+停止位
0 (8位数据帧)	1	起始位+7位数据+校验位+停止位
1 (9位数据帧)	0	起始位+9位数据+停止位
1 (9位数据帧)	1	起始位+8位数据+校验位+停止位



时钟波特率控制



➤ 收、发模块使用相同的波特率发生器

□ 收、发有各自的使能控制

✉ TE & RE@USART_CR1

$$\text{波特率} = \frac{f_{PLCK}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

□ 波特率 = $f_{CK} / [8 \times (2 - \text{OVER8}) \times \text{USARTDIV}]$

✉ f_{CK} : USART模块所在的外设总线时钟

✉ OVER8@USART_CR1: 8倍/16倍过采样

□ OVER8=1, 8倍过采样

➤ 波特率 = $f_{CK} / [8 \times \text{USARTDIV}]$

□ OVER8=0, 16倍过采样

➤ 波特率 = $f_{CK} / [16 \times \text{USARTDIV}]$



时钟波特率控制

➤ 收、发模块使用相同的波特率发生器

□ 波特率 = $f_{CK} / [8 * (2 - OVER8) * USARTDIV]$

✉ USARTDIV: USART_BRR寄存器中的无符号定点数

□ DIV_Mantissa[11:0]: USARTDIV的整数部分

□ DIV_Fraction[3:0]: USARTDIV的小数部分

16倍过采样

➤ OVER8=0: 小数部分被编码为4比特 (DIV_fraction[3:0] bits in the USART_BRR register), 小数部分为DIV_fraction[3:0]/16

8倍过采样

➤ OVER8=1: 小数部分被编码为3比特 (DIV_fraction[2:0] bits in the USART_BRR register), DIV_fraction[3]必须清零, 小数部分为DIV_fraction[2:0]/8

Baud rate register (USART_BRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

波特率配置实例

➤ 配置USART2的波特率为9600，计算BRR的值

□ 设PCLK=42MHz，16倍过采样

□ 波特率 = $f_{ck} / (8 \times (2 - OVER8) \times USARTDIV)$

□ $USARTDIV = 42M / (16 \times 9600) = 273.4375$

□ $DIV_Fraction = 16 \times 0.4375 = 0d07 = 0x7$

□ $DIV_Mantissa = \text{取整}(0d273.4375) = 0d273 = 0x111$

□ 故 $USART_BRR = 0x1117$

Baud rate register (USART_BRR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w



USARTDIV计算举例



➤ OVER8=1 (8倍过采样)

□ 例1: USART_BRR=0x1B6

$$6/8 = 0d0.75$$

⊗ DIV_Mantissa = 0d27, DIV_Fraction[2:0] = 0d6

⊗ 故USARTDIV=27.75

□ 例2: 设置USARTDIV=25.62

⊗ DIV_Fraction = $8 * 0d0.62 = 0d4.96$, 取整至0d5 = 0x5

⊗ DIV_Mantissa = mantissa (0d25.620) = 0d25 = 0x19

⊗ 故USART_BRR = 0x195 => USARTDIV = 0d25.625

□ 例3: 设置USARTDIV=50.99

⊗ DIV_Fraction = $8 * 0d0.99 = 0d7.92$, 取整至0d8 = 0x8

□ DIV_frac[2:0]溢出, 向整数部分进位

⊗ DIV_Mantissa = mantissa (0d50.990 + carry) = 0d51 = 0x33

⊗ 故USART_BRR = 0x0330 => USARTDIV = 0d51.000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIV_Mantissa[11:0]												DIV_Fraction[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



STM32 USART寄存器



- 状态寄存器——USART_SR
- 数据寄存器——USART_DR
- 波特率寄存器——USART_BRR
- 控制寄存器——USART_CR1、2、3
- 保护时间和预分频器寄存器——USART_GTPR



串口寄存器-USART_CR1

26.6.4 控制寄存器 1 (USART_CR1)

Control register 1

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- 位31:16 保留: 必须保持复位值
- 位15 **OVER8**: 过采样模式 (Oversampling mode)
 - ❑ 0: 16 倍过采样
 - ❑ 1: 8 倍过采样
 - ❑ 8 倍过采样在智能卡、IrDA和LIN模式下不可用
- 位14 保留: 必须保持复位值



串口寄存器-USART_CR1

26.6.4 控制寄存器 1 (USART_CR1)

Control register 1

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

➤ 位13 **UE**: USART使能 (USART enable)

- ❑ 由软件置1和清0
- ❑ 1: 使能 USART
- ❑ 0: 禁止USART预分频器和输出

☒ 该位清零后, USART预分频器和输出将停止, 并会结束当前字节传输以降低功耗



串口寄存器-USART_CR1

26.6.4 控制寄存器 1 (USART_CR1)

Control register 1

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

➤ 位12 M: 字长 (Word length)

- ☐ 由软件置1或清零
- ☐ 0: 1 起始位, 8 数据位, n 停止位
- ☐ 1: 1 起始位, 9 数据位, n 停止位
- ☐ 在数据传输 (发送和接收) 期间不得更改M位



串口寄存器-USART_CR1

26.6.4 控制寄存器 1 (USART_CR1)

Control register 1

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

➤ 位 10 **PCE**: 奇偶校验控制使能 (Parity control enable)

- ☐ 由软件置 1 和清0, 选择硬件奇偶校验控制 (生成和检测)
- ☐ 0: 禁止奇偶校验控制
- ☐ 1: 使能奇偶校验控制

✉ 计算出的奇偶校验位被插入到msb位置 (如果M=1, 则为第9位; 如果M=0, 则为第8位), 并对接收到的数据检查奇偶校验位



串口寄存器-USART_CR1

26.6.4 控制寄存器 1 (USART_CR1)

Control register 1

偏移地址: 0x0C

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Reserved	UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK
rw	Res.	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

➤ 位 9 **PS**: 奇偶校验选择 (Parity selection)

- ☐ 由软件置1和清0, 用于在使能奇偶校验生成/检测 (PCE 位置1) 时选择奇校验或偶校验
- ☐ 0: 偶校验
- ☐ 1: 奇校验



串口寄存器-USART_CR1



- 位8 **PEIE**: PE (Parity Error) 中断使能 (PE interrupt enable)
 - ❑ 由软件置1和清0
 - ❑ 0: 禁止中断
 - ❑ 1: 当USART_SR寄存器中PE=1时, 产生USART中断
- 位7 **TXEIE**: TXE中断使能 (TXE interrupt enable)
 - ❑ 由软件置1和清0
 - ❑ 0: 禁止中断
 - ❑ 1: 当USART_SR寄存器中 TXE=1时, 产生USART中断
- 位6 **TCIE**: 传送完成中断使能 (Transmission complete interrupt enable)
 - ❑ 由软件置1和清0
 - ❑ 0: 禁止中断
 - ❑ 1: 当 USART_SR寄存器中TC=1时, 产生USART中断



串口寄存器-USART_CR1



- 位 5 **RXNEIE**: RXNE (Received Data Ready to be Read) 中断使能 (RXNE interrupt enable)
 - ❑ 由软件置1和清0
 - ❑ 0: 禁止中断
 - ❑ 1: 当USART_SR寄存器中ORE=1或RXNE=1时, 产生USART中断
- 位 4 **IDLEIE**: IDLE (Idle Line Detected) 中断使能 (IDLE interrupt enable)
 - ❑ 由软件置1和清0
 - ❑ 0: 禁止中断
 - ❑ 1: 当USART_SR寄存器中IDLE=1时, 产生USART中断



串口寄存器-USART_CR1



- 位3 **TE**: 发送器使能 (Transmitter enable)
 - ☐ 由软件置1和清0
 - ☐ 0: 禁止发送器
 - ☐ 1: 使能发送器
- 位2 **RE**: 接收器使能 (Receiver enable)
 - ☐ 由软件置1和清0
 - ☐ 0: 禁止接收器
 - ☐ 1: 使能接收器并开始搜索起始位



串口寄存器-USART_CR2

26.6.5 控制寄存器 2 (USART_CR2)

Control register 2

偏移地址: 0x10

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	Res.	ADD[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

➤ 位 13:12 STOP: 停止位 (STOP bit)

- ☐ 00: 1 个停止位
- ☐ 01: 0.5 个停止位 (不适用于UART4和UART5)
- ☐ 10: 2 个停止位
- ☐ 11: 1.5 个停止位 (不适用于UART4和UART5)

➤ 位 11 CLKEN: 时钟使能 (Clock enable)

- ☐ 0: 禁止 SCLK 引脚
- ☐ 1: 使能 SCLK 引脚 (不适用于UART4和UART5)



串口寄存器-USART_CR3

26.6.6 控制寄存器 3 (USART_CR3)

Control register 3

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- 位 10 CTSIE: CTS 中断使能 (CTS interrupt enable)
- 位 9 CTSE: CTS 使能 (CTS enable)
- 位 8 RTSE: RTS 使能 (RTS enable)
- 位 7 DMAT: 发送器DMA使能(DMA enable transmitter)
- 位 6 DMAR: 接收器DMA使能(DMA enable receiver)



串口寄存器-USART_CR3

26.6.6 控制寄存器 3 (USART_CR3)

Control register 3

偏移地址: 0x14

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- 位 3 HDSEL: 半双工选择 (Half-duplex selection)
- 位 0 EIE: 错误中断使能 (Error interrupt enable)



串口寄存器-USART_SR

26.6.1 状态寄存器 (USART_SR)

Status register

偏移地址: 0x00

复位值: 0x00C0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Reserved							CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
							rc_w0	rc_w0	r	rc_w0	rc_w0	r	r	r	r	r

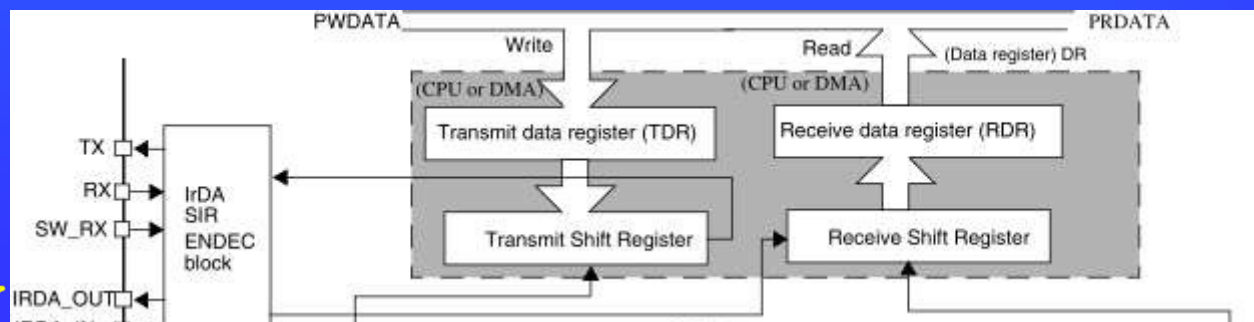
- 位 7 TXE: 发送数据寄存器为空 (Transmit data register empty)
 - ❑ 0: 数据未传输到移位寄存器
 - ❑ 1: 数据传输到移位寄存器
- 位 6 TC: 发送完成 (Transmission complete)
 - ❑ 0: 传送未完成
 - ❑ 1: 传送已完成
- 位 5 RXNE: 读取数据寄存器不为空 (Read data register not empty)
 - ❑ 0: 未接收到数据
 - ❑ 1: 已准备好读取接收到的数据



串口寄存器-USART_DR

➤ 数据寄存器 (USART_DR) Data register

- ❑ 偏移地址: 0x04
- ❑ 复位值: 0xFFFF XXXX
- ❑ 位 31:9 保留, 必须保持复位值
- ❑ 位 8:0 DR[8:0]: 数据值
- ❑ 包含接收到的数据字符或将要发送的数据字符, 具体取决于所执行的操作是“读取”操作还是“写入”操作
- ❑ 数据寄存器包含两个寄存器, 一个用于发送 (TDR), 一个用于接收 (RDR)
 - ☒ TDR寄存器在内部总线和输出移位寄存器之间提供了并行接口
 - ☒ RDR寄存器在输入移位寄存器和内部总线之间提供了并行接口



与发送相关的标志位

➤ TxE@USART_SR (Transmit data register empty)

❑ 置位时表明：发送数据寄存器为空

✉ 当前没有数据在发送：发送缓冲为空

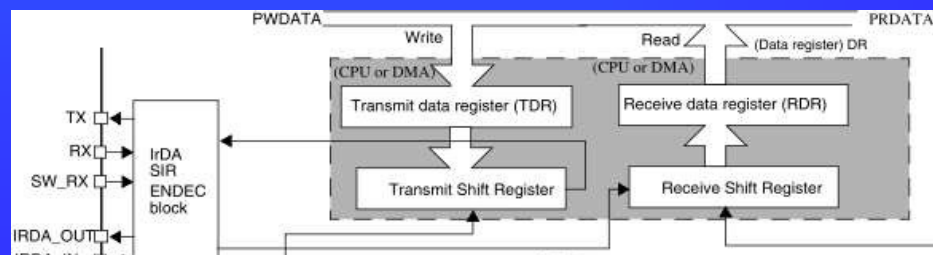
❑ 写TDR会把数据直接放到移位寄存器中，并立即开始发送，TxE也会马上置位

✉ 当前有数据正在发送：发送缓冲的数据已经移到移位寄存器中发送

❑ 写TDR会把数据先存在发送缓冲器中，直到当前数据发送结束时再将数据拷贝至移位寄存器

✉ 若TxEIE置位，会产生“发送中断”

❑ 写TDR则复位TxE标志



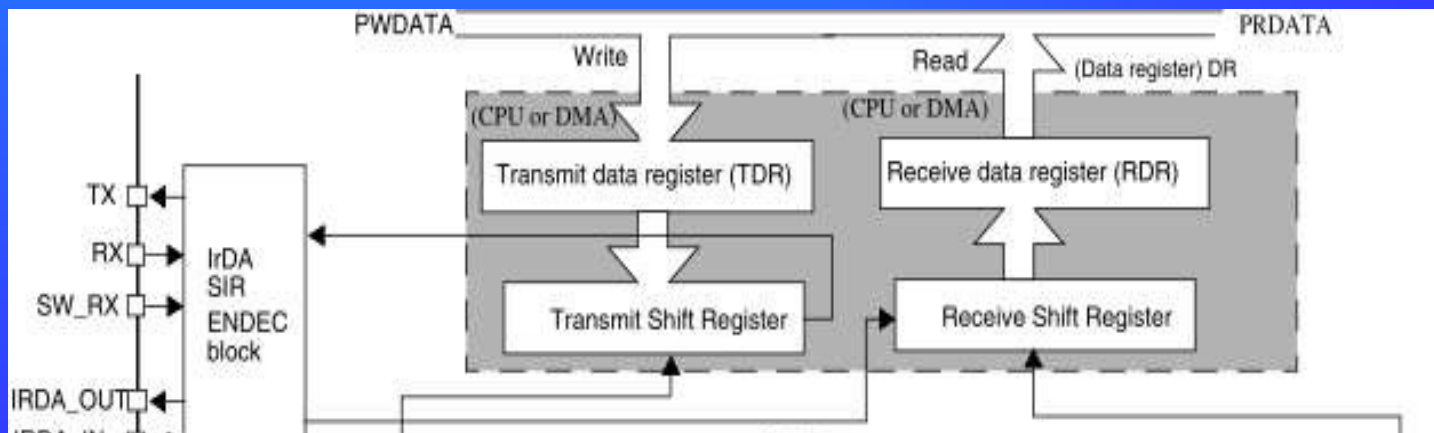
与发送相关的标志位

➤ TC@USART_SR (Transmission complete)

□ 置位时表明：发送完成

- ✉ 一帧数据的停止位已经发送完，且TxE保持置位
- ✉ 如果TCIE置位，会产生“发送完成中断”
- ✉ 必须等到TC置位后才能关闭USART或者进入低功耗

□ 读取SR再写TDR，则复位TC



发送操作流程



➤ 普通数据帧发送序列

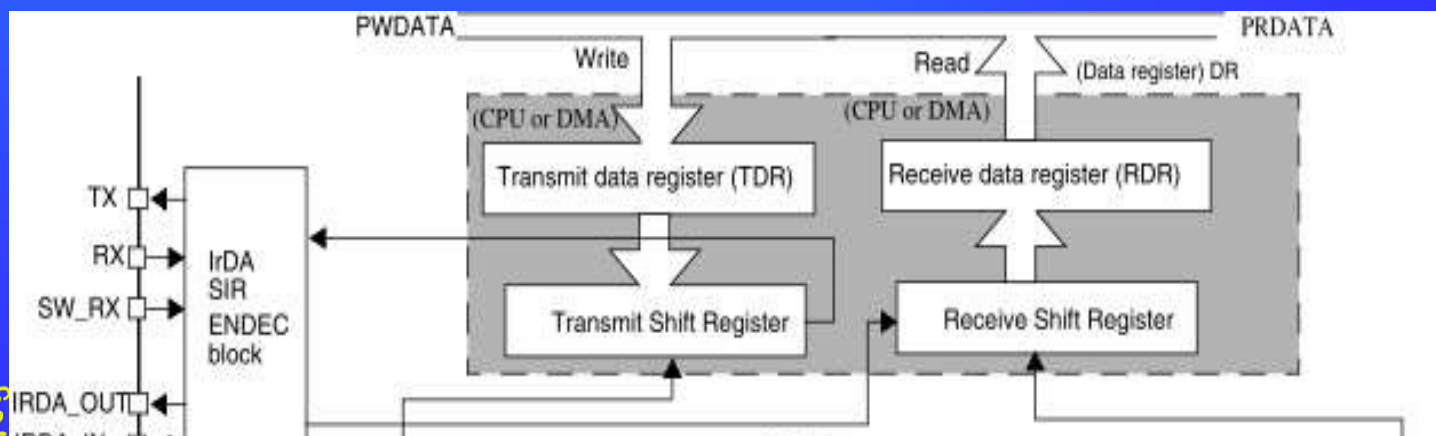
- ❑ 使能USART (置位UE@USART_CR1)
- ❑ 定义数据长度 (设置M@USART_CR1)
- ❑ 定义停止位数 (设置STOP@USART_CR2)
- ❑ 设置波特率 (设置USART_BRR)
- ❑ 先发一个空闲帧 (置位TE@USART_CR1) (发送器使能)
- ❑ 循环以下步骤:
 - ✉ 若TxE置位, 则发送数据 (写USART_DR)
- ❑ 写完最后一个数据, 等待TC置位后才能关闭USART或进入低功耗模式



与接收相关的标志位

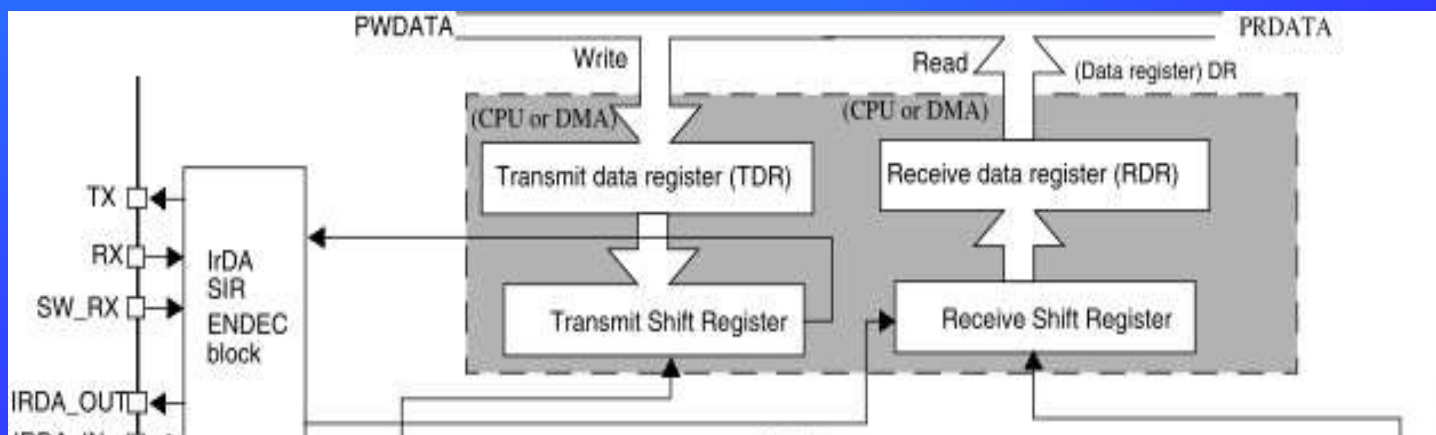
➤ RxNE@USART_SR

- ❑ 读数据寄存器不空 (Read data register not empty)
- ❑ 移位寄存器中内容已经被送入接收寄存器中
 - ✉ 可以读取数据
 - ✉ 也可以读取错误标志
 - ❑ 接收过程中可能会检测到帧错误、噪声或溢出错误
- ❑ 如果RxNEIE已被置位，则可产生中断请求
- ❑ 非DMA模式下，软件读取RDR清零该标志
 - ✉ 必须在下一个数据到来之前清零，否则产生溢出错误
- ❑ DMA模式下，DMA读取RDR并清零该标志



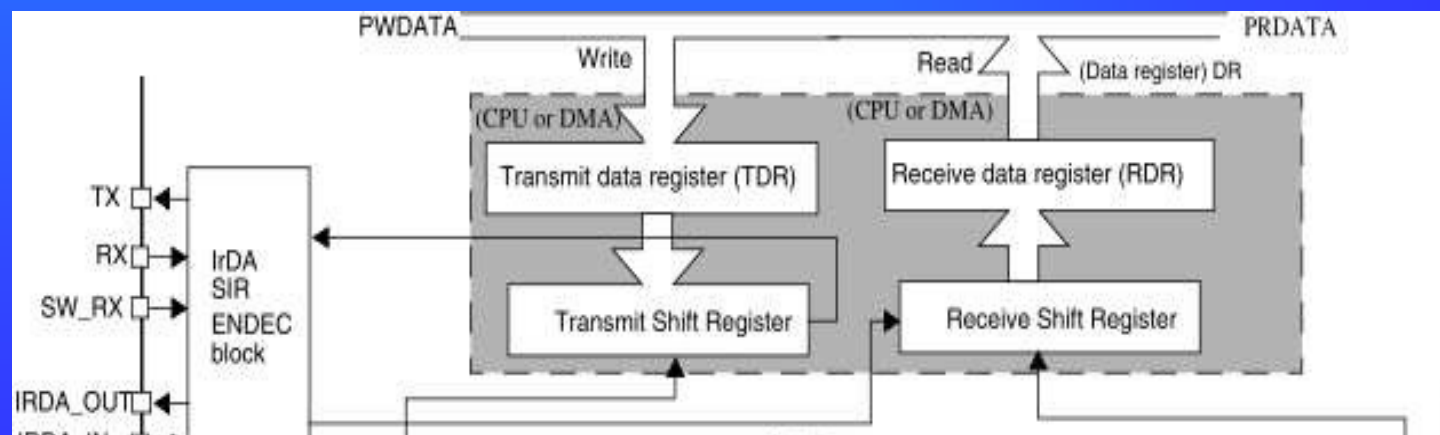
接收时的错误标志

- ORE: 溢出错误 (Overrun error), 在RxNE置位时收到新的数据
 - ❑ 接收缓冲器中仍保留先前未读取的数据, 而移位寄存器被溢出的数据不断覆盖
 - ❑ 使能后, ORE置位时可以产生中断
 - ❑ 读USART_SR和USART_DR可以复位ORE标志



接收时的错误标志

- FE: 帧错误 (Framing error), 未识别出停止位
 - ❑ FE和RxNE同时置位, 表示数据已送到USART_DR
 - ❑ 非DMA模式下, FE没有自己的中断
 - ❑ DMA模式下, 若EIE置位会产生中断
 - ❑ 读USART_SR和USART_DR复位该标志位
- PE: 奇偶校验错误 (Parity error)
 - ❑ 置位PE@USART_SR, 如果PEIE置位将产生中断
 - ❑ 读SR, 接着读或写DR, 可以清零该标志



接收操作流程



➤ 普通数据帧接收序列

- ☐ 使能USART (置位UE@USART_CR1)
- ☐ 定义数据长度 (设置M@USART_CR1)
- ☐ 定义停止位数 (设置STOP@USART_CR2)
- ☐ 设置波特率 (设置USART_BRR)
- ☐ 使能接收并开始搜寻起始位 (置位RE@USART_CR1)
 - ☐ 检测起始位...
- ☐ 循环以下步骤:
 - ☐ 等待RxNE置位, 接收数据 (读取USART_DR)
 - ☐ 此时也可读取相应错误标志(噪声、溢出、帧错误)



USART interrupt requests

Table 11-1 USART interrupt requests

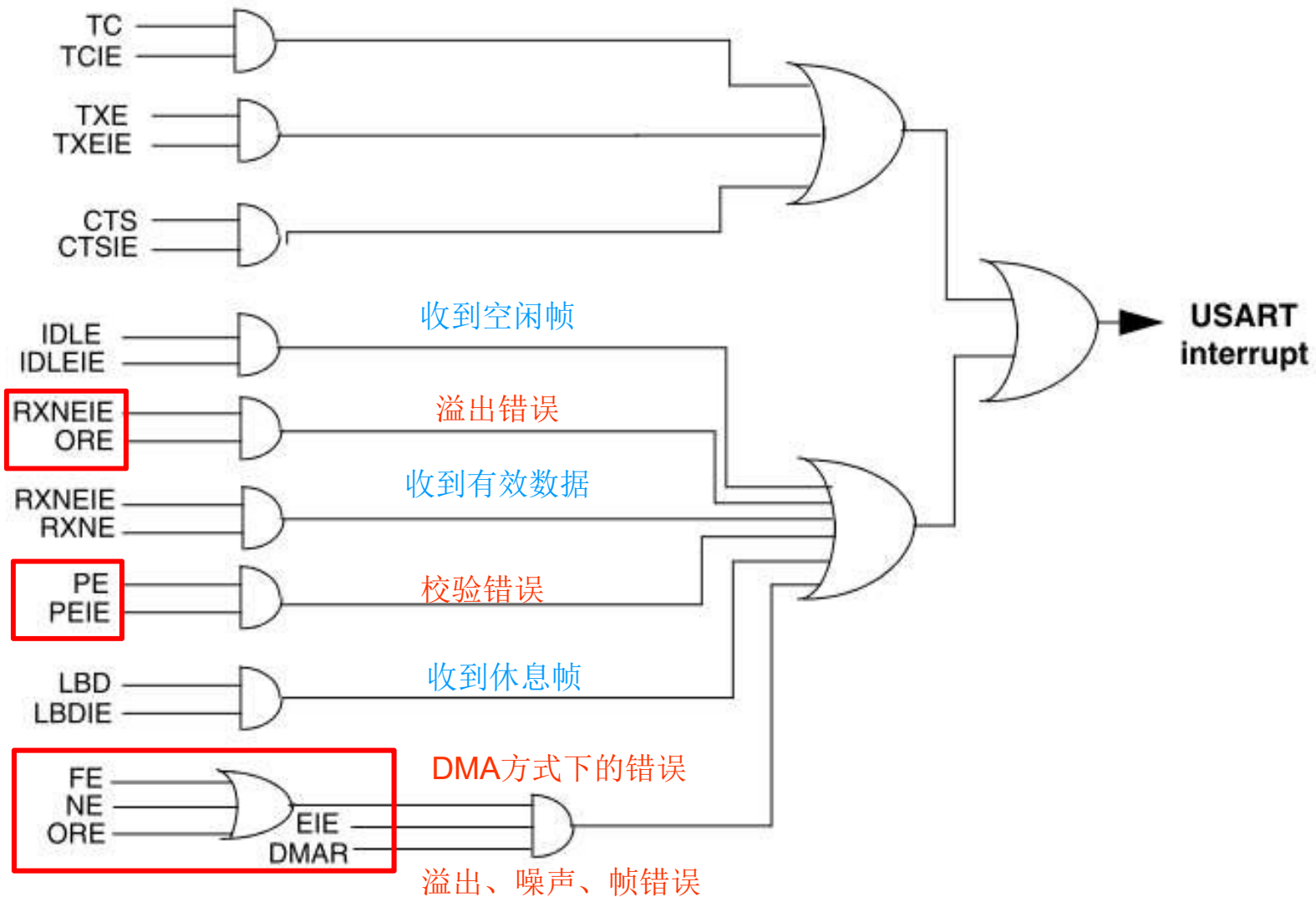
Interrupt event	Event flag	Enable control bit
Transmit Data Register Empty	TXE	TXEIE
CTS flag	CTS	CTSIE
Transmission Complete	TC	TCIE
Received Data Ready to be Read	RXNE	RXNEIE
Overrun Error Detected	ORE	
Idle Line Detected	IDLE	IDLEIE
Parity Error	PE	PEIE
Break Flag	LBD	LBDIE
Noise Flag, Overrun error and Framing Error in multibuffer communication	NF or ORE or FE	EIE



USART 中断管理

发送
中断

接收
中断



本章结束

