Chapter 4:
USART: communicate with PC

Prof. Li Wen Zheyuan Gong

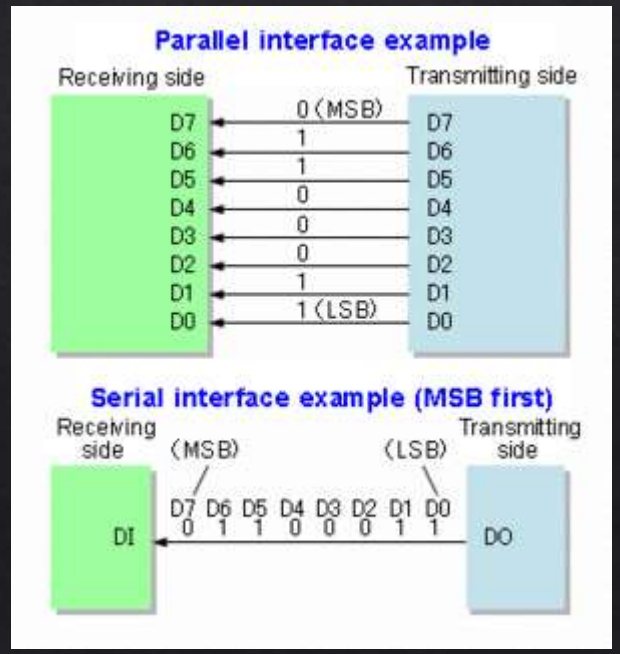
zheyuangong@163.com

USART: communicate with PC

- Aims:
  - Understand the principles of serial communication
  - Understand how to use USART

USART: Serial communication on STM32

- information transfers in or out one bit a time
- in contrast to a parallel port



USART: communicate with PC

- Coding Steps:
  - 1. enable CLK of USART & GPIO
  - 2. reset USART
  - 3. configure GPIO
  - 4. initialize USART
  - 5. enable interrupt & NVIC
  - 6. enable USART
  - 7. code interrupt service function

USART: communicate with PC

- 1. enable CLK of USART & GPIO (PA.9, PA.10)

  ```
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1|RCC_APB2Periph_GPIOA, ENABLE);
  ```

- 2. reset USART

  ```
  USART_DeInit(USART1);
  ```

- 3. configure GPIO
  - PA.9: Alternate function push-pull (GPIO_Mode_AF_PP)
  - PA.10: Input floating (GPIO_Mode_IN_FLOATING)

USART: communicate with PC

- 4. initialize USART

```
USART_InitStructure.USART_BaudRate = bound;//一般设置为9600;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;//字长为8位数据格式
USART_InitStructure.USART_StopBits = USART_StopBits_1;//一个停止位
USART_InitStructure.USART_Parity = USART_Parity_No;//无奇偶校验位
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;//无硬件数据流控制
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;  //收发模式

USART_Init(USART1, &USART_InitStructure); //初始化串口
USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);//开启中断
USART_Cmd(USART1, ENABLE);                    //使能串口
```

- 5. enable interrupt & NVIC

```
NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority=3 ;//抢占优先级3
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 3;    //子优先级3
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;      //IRQ通道使能
NVIC_Init(&NVIC_InitStructure); //根据指定的参数初始化VIC寄存器
```

- 6. enable USART

  USART_Cmd(USART1, ENABLE);

USART: communicate with PC

- 8. registers - **Data register**

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							DR[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:9 Reserved, forced by hardware to 0.

Bits 8:0 **DR[8:0]**: Data value

Contains the Received or Transmitted data character, depending on whether it is read from or written to.

The Data register performs a double function (read and write) since it is composed of two registers, one for transmission (TDR) and one for reception (RDR)

The TDR register provides the parallel interface between the internal bus and the output shift register (see Figure 1).

The RDR register provides the parallel interface between the input shift register and the internal bus.

When transmitting with the parity enabled (PCE bit set to 1 in the USART_CR1 register), the value written in the MSB (bit 7 or bit 8 depending on the data length) has no effect because it is replaced by the parity.

When receiving with the parity enabled, the value read in the MSB bit is the received parity bit.

Thanks

2016.5.