

STM32 UART 的使用过程

1、使用 UART 前必须启动相应的外设时钟，其主要用到固件库的 RCC_APBnPeriphClockCmd 函数。

使能 UART1: 使用 RCC_APB2PeriphClockCmd (RCC_APB2Periph_USART1, ENABLE)

使能 UART2: 使用 RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2 , ENABLE)

2、使用中断进行 UART 操作的需要配置 NVIC，设置中断优先级。如：

```
/* Configure the NVIC Preemption Priority Bits */
NVIC_PriorityGroupConfig(NVIC_PriorityGroup_0);

/* Enable the USART1 Interrupt */
NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQChannel;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
```

3、配置相应的 GPIO 口。

如果系统的 UART 需要进行重映射，需要使用 GPIO_PinRemapConfig 函数进行重映射，如：

```
GPIO_PinRemapConfig(GPIO_Remap_USART2, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;//注意：Rx 为浮空，Tx 为第二功能上拉。
将 Rx 配置为：浮空输入模式，Tx 配置为带上拉的第二功能模式。并用 GPIO_Init() 函数初始化。如：
/* Configure USART2 Rx PA3 input floating */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
/* Configure USART1 Tx (PA.09) as alternate push-pull */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

4、配置 UART

当在 conf 文件中配置正确的外晶振后，在 USART_InitTypeDef 定义的结构体中直接写入 UART 的波特率、通讯长顿、模式、硬件通讯控制，收发模式。再用 USART_Init() 进行初始化。如：

```
USART_InitStructure.USART_BaudRate = 9600;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
USART_InitStructure.USART_StopBits = USART_StopBits_1;
USART_InitStructure.USART_Parity = USART_Parity_No;
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;

/* Configure USART1 */
USART_Init(USART1, &USART_InitStructure);
```

而后使能收发中断。如：

```
/* Enable USART1 Receive and Transmit interrupts */
```

```

USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
USART_ITConfig(USART1, USART_IT_TXE, ENABLE);
// USART_ITConfig(USART1, USART_IT_TXE, DISABLE);

```

注意：一般不将 TXE 中断使能。因为一旦将此中断使能，如果 UART 发送缓冲区空，则会立即进入 UART 中断中，因此可在程序中需要发送数据处，使能 TXE 中断。在 UART 中断种用 USART_SendData（）来发送数据。

完成中断使能后，还需要使能 UART 口：

如：

```

/* Enable the USART1 */
USART_Cmd(USART1, ENABLE);
/* Enable the USART2 */
USART_Cmd(USART2, ENABLE);

```

中断程序（stm32f10x_it.c）可以如下完成发送：注意所有的的串口中断需要在中断服务程序中判断中断源以分别处理。

```

void USART1_IRQHandler(void)
{
    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)
    {
        /* Read one byte from the receive data register */
        RxBuffer1[RxCounter1++] = USART_ReceiveData(USART1);
        if(RxCounter1 == NbrOfDataToRead1)
        {
            USART_ITConfig(USART1, USART_IT_RXNE, DISABLE); //发送完成后，将 RXNE 禁止。
        }
    }

    if(USART_GetITStatus(USART1, USART_IT_TXE) != RESET)
    {
        USART_SendData(USART1, TxBuffer1[TxCounter1++]);
        if(TxCounter1 == NbrOfDataToTransfer1)
        {
            USART_ITConfig(USART1, USART_IT_TXE, DISABLE);
        }
    }
}

```