

STM32F407VGT6 的 485 通信程序

【 SP3485 芯片 & 中断接收 】

本例程为 STM32F4XX (M4 内核) 的 485 通信程序 , 采用串口 1 发送和接收数据 , 中断接收 , 将接收到的数据重新发送出去。

主函数 main.c 文件如下 :

```
#include "stm32f4xx.h"
```

```
/******\
** 文件名:      mian.c *****
** 库版本:      STM32F4xx_DSP_StdPeriph_Lib_V1.0.1 *****
** 工作环境:   RealView MDK-ARM 4.23 *****
** 作者:       曾有根 *****
** 生成日期:   2012-08-03 *****
** 功能:       RS485 通过串口 1 发送 , 中断接收 ! 将接收到   ***
**             的数据通过再次发送出去   *****
\*****/

extern void uart_init(void);
extern void USART1_SendByte(u8 Data);
extern unsigned char UART1_GetByte(u8 GetData);
extern void delay(unsigned int dl);

void delay(unsigned int dl)
{
    unsigned int i,y;
    for(i = 0; i < 5000; i++)
    {
        for(y = 0; y < dl; y++);
    }
}

static void led_init(void)
{
    GPIO_InitTypeDef  GPIO_InitStructure;

    /* Enable the GPIO_LED Clock */
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOE, ENABLE);

    /* Configure the GPIO_LED pin */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 ;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
}
```

```

    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOE, &GPIO_InitStructure);
}

int main(void)
{
    uart_init();
    led_init();

    while (1)
    {
        USART1_SendByte(0x12);

        GPIO_SetBits(GPIOE, GPIO_Pin_7 );           //LED1 灯闪烁，表示
数据发送完成
        delay(1000);
        GPIO_ResetBits(GPIOE, GPIO_Pin_7 );
        delay(1000);

    }
}

```

串口配置程序， **Uart_Config.c** 文件如下：

```

#include "stm32f4xx.h"

extern void delay(unsigned int dl);

#define TX_485  GPIO_SetBits(GPIOA,GPIO_Pin_8);
#define RX_485  GPIO_ResetBits(GPIOA,GPIO_Pin_8);

void USART1_SendByte(u8 SendData)
{
    TX_485;                                     //打开发
送控制端
    delay(10);                                  //延时，这个
必须加上，不加会导致数据出错
    USART_SendData(USART1,SendData);
    //发送数据
    while(USART_GetFlagStatus(USART1, USART_FLAG_TXE) ==
RESET);    //等待数据发送完成
    delay(10);
    RX_485;                                     //关闭发
送控制端
}

unsigned char UART1_GetByte(u8 MidData)

```

```

{
    RX_485; //打
    开接收控制端
    delay(10); //延时,
    同样必须加上
    MidData = USART_ReceiveData(USART1); //
    接收数据
    delay(10);
    TX_485; //关闭接
    收控制端
    return MidData;
}

void USART1_IRQHandler(u8 GetData)
{
    u8 BackData;

    if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)
        //中断产生
    {
        GetData = UART1_GetByte(BackData);
        //直接读取寄存器的数据也行 GetData = USART1->DR;
        USART1_SendByte(GetData); //
        发送数据

        GPIO_SetBits(GPIOE, GPIO_Pin_8 );
        //LED2 灯闪烁, 表示数据接收成功且发送完成
        delay(1000);
        GPIO_ResetBits(GPIOE, GPIO_Pin_8 );
    }
}

void uart_init(void)
{
    USART_InitTypeDef USART_InitStructure;
    NVIC_InitTypeDef NVIC_InitStructure;
    GPIO_InitTypeDef GPIO_InitStructure;

    /* Enable GPIO clock */
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);

    /* Enable USART clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);

    /* Connect USART pins to A9\10 */
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource9, GPIO_AF_USART1);
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource10, GPIO_AF_USART1);
}

```

```

/* Configure USART Tx and Rx as alternate function push-pull */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;           //输出 TX
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;       //必须
为 AF,OUT 不行
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOA, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;         //输入 RX
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;       //必须
为 AF,与 M3 不同
GPIO_Init(GPIOA, &GPIO_InitStructure);

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 ;         //485 使能端配
置
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);

USART_InitStructure.USART_BaudRate = 115200;
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;
USART_Init(USART1, &USART_InitStructure);

NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);

NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;

NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);

/* Enable USART */
USART_Cmd(USART1, ENABLE);
USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);

USART_ClearFlag(USART1, USART_FLAG_TC);
}

```

说明：已经本人下载至 STM32F4的开发板上成功调试，并且能够正确的收发数据！可供广大奋斗在前线的机油们参考！