

晟矽微电 应用笔记

MS32F031

USART 自动波特率使用

AN22014

V1.0





目 录

1	适用范围	1
2	USART 自动波特率	1
2.1	外设功能	1
2.2	例程运行	1
2.2.1	例程要求	1
2.2.2	例程默认配置运行	1
2.2.3	自动波特率模式 1	2
2.2.4	其它例程配置	3
2.3	主要代码	3
2.3.1	使能自动波特率	3
2.3.2	自动波特率应用逻辑	3
2.4	说明	5
3	修订记录	6
4	免责声明	7



1 适用范围

本文档适用于 MS32F031A6, USART 自动波特率使用参考。

例程基于 MS32F031A6 EV Board V1.1 (2021-11-25)。

注: 本文编写时数据手册 V1.0.2, 用户手册 V1.0.2, 若版本变更请查阅对应章节。

2 USART 自动波特率

2.1 外设功能

USART 自动波特率, 可参阅用户手册 22.4.6 章节。

在打开波特率自动检测之前, USART_BRR 寄存器必须先初始化为一个不为零的波特率值。

将 USART_CR2 寄存器中的 ABREN 位置 1, 打开自动波特率检测功能。然后 USART 就在 RX 线上等待第一个字符过来。自动波特率操作结束后, USART_ISR 寄存器中的 ABRF 标志会被硬件置 1。

注: Mode2, Mode3 模式暂不支持。

2.2 例程运行

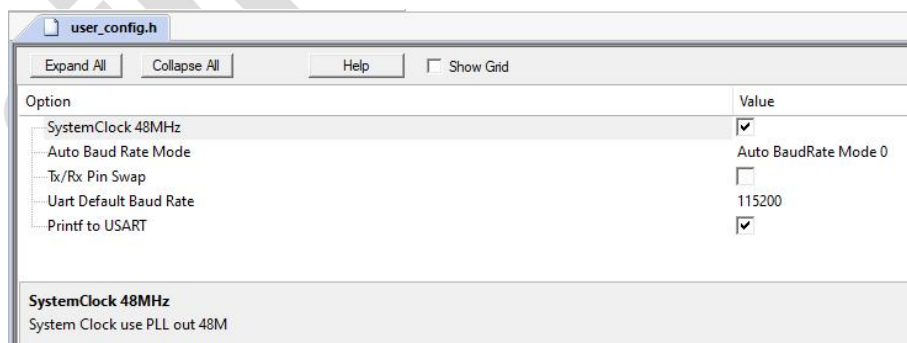
2.2.1 例程要求

附件例程解压后放在 MS32F0x1_Periph_Lib_Example\proj\MS32F031_EV\USART 目录下。



2.2.2 例程默认配置运行

例程默认配置: 系统时钟 48M, 自动波特率模式 0, 收发引脚不交换, 默认波特率 115200, 如下图:

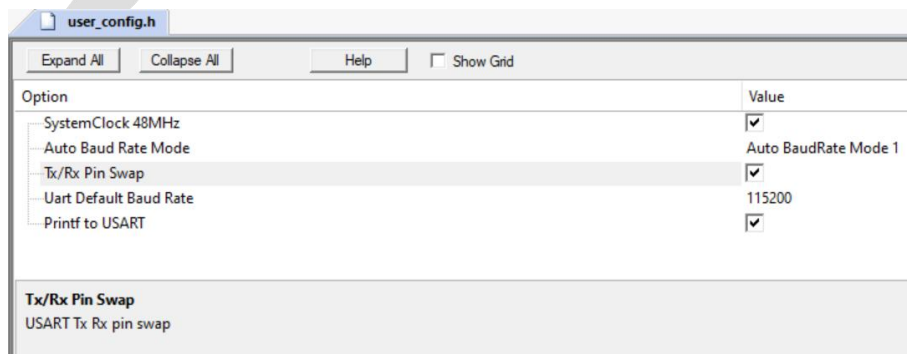


编译运行, 上位机串口波特率设置为 115200 接收 MCU 输出运行信息, LED2 闪烁后, 调整串口波特率为需求波特率(下图以 38400 为例)并发送指定的帧格式(下图#ABCDEF+回车换行), 波特率自动调整为 38400 左右的值, 并进行通信。

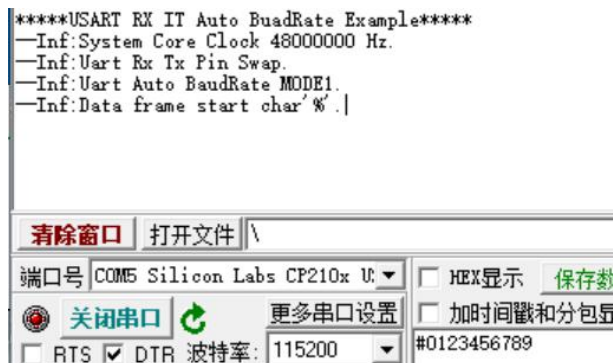


2.2.3 自动波特率模式 1

打开 user_config.h 文件, Configuration Wizard 修改例程配置: 自动波特率模式 1, 收发引脚交换 (可选), 默认波特率 115200, 如下图:



交换 EV 板的 TX, RX 连接关系 (若使能了 SWAP), 编译运行, MCU 发出运行信息如下:

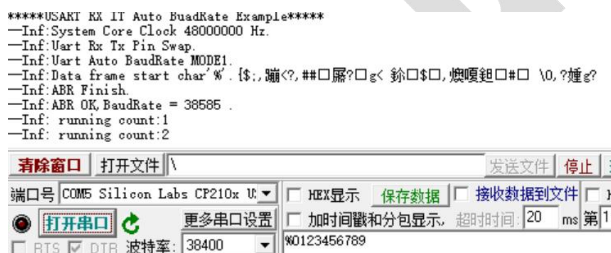


波特率设置为需求的 38400 发送，非要求帧格式，自动波特率不正确，上位机收到乱码；

—Inf:Data frame start char' %\$.;,\$,c??,##□露浼□芳发□□\$□,燠嘎?□□ \0,?□gg



波特率保持为需求的 38400 发送，修改数据符合帧格式后(下图数据: %0123456789 回车换行)，自动波特率 OK，程序输出运行计数信息。



2.2.4 其它例程配置

可借助 Configuration Wizard 修改各种组合测试：系统时钟、默认波特率、是否自动波特率及模式等。

自动波特率字符例程中 Mode0 及 Mode1 分别使用#，%；可参考用户手册要求选择其它满足产品应用需求的字符。

2.3 主要代码

本文例程基于 UART_printf_loop_RX_IT 修改完善，和自动波特率相关的核心代码如下文。

2.3.1 使能自动波特率

选择自动波特率模式及使能自动波特率，主要函数如下（例程 main.c 文件）：

```
MS32_USART_SetAutoBaudRateMode(USART1, MS32_USART_AUTOBAUD_DETECT_ON_STARTBIT);  
MS32_USART_EnableAutoBaudRate(USART1);
```

2.3.2 自动波特率应用逻辑

示例如下（例程 main.c 文件），可依据需求修改：

```
ErrorStatus uart_AutobaduRate(void) {  
    ErrorStatus fun_status_flag= ERROR;  
    FlagStatus ARB_GO_ON_Flag=SET;
```



```
uint8_t ABR_try_times =0,received_char =0;
uint32_t count_LED = 0,led_blink_period = LED_BLINK_SLOWE;
uint32_t usart_clk_freq=0,usart_baudrate=0;

Start_100MS_SF_Timer();
usart_clk_freq =MS32_RCC_GetUSARTClockFreq(MS32_RCC_USART1_CLKSOURCE);

MS32_USART_DisableIT_RXNE(USART1);
while(ARB_GO_ON_Flag ==SET) {
    if(MS32_USART_IsActiveFlag_ABRE(USART1)) {
        printf("\r\n##Err:ABRE Set! Request again.");
        MS32_USART_RequestAutoBaudRate(USART1);    // clear ABRE
    }

    if(MS32_USART_IsActiveFlag_ABR(USART1)) {    // if ABRF set, stop ABR
        printf("\r\n--Inf:ABR Finish.");
        received_char = MS32_USART_ReceiveData8(USART1);
        if( received_char == UART_START_CHAR) {
            usart_baudrate          =MS32_USART_GetBaudRate(USART1,          usart_clk_freq,
MS32_USART_OVERSAMPLING_16);
            printf("\r\n--Inf:ABR OK,BaudRate = %d .",usart_baudrate);
            ARB_GO_ON_Flag=RESET;
            fun_status_flag= SUCCESS;
        } else {
            printf("\r\n!!Warn:ARB again.");
            MS32_USART_RequestAutoBaudRate(USART1);    // clear ABRF ,restart ABR;
            ABR_try_times++;
        }

        if(recv_status == UART_ERROR_ORE) {    //
uart error interrupt occur
            recv_status =UART_ERROR_CLR;
        }

    }

    if(ABR_try_times ==ABR_TRY_MAX_NUM) {
        ARB_GO_ON_Flag=RESET;
        MS32_USART_DisableAutoBaudRate(USART1);
        MS32_USART_Disable(USART1);
        usart1_uart_init(UART_BAUD_RATE_VALUE);
        printf("\r\n!!Warn:ARB try too many times.STOP ABR,BuadRate
default %d",UART_BAUD_RATE_VALUE);
```



```
}  
  
if(Get_State_100MS_SF_Timer() == SF_TIMER_OV) {  
    Start_100MS_SF_Timer();  
  
    count_LED++;  
    if(count_LED >= led_blink_period) {  
        LED2_TOGGLE();  
        count_LED =0;  
    }  
}  
}  
} //end while(ARB_GO_ON_Flag ==SET)  
MS32_USART_EnableIT_RXNE(USART1);  
LED2_OFF();  
return fun_status_flag;  
}
```

2.4 说明

- 例程为易于理解，仅上电后执行自动波特率；
- 例程执行自动波特率时禁止了接收中断，此时会丢失用于波特率同步的数据帧；可更改例程串口接收状态机；如接收数据直接存入接收缓冲区，使用 IDLE 中断判断帧结束，主程序中解析接收缓冲区的合理性等方式实现自动波特时不禁止接收中断。
- 未尽事宜，可依据用户手册进行开发验证或联系我方进行技术讨论。



3 修订记录

版本	修订日期	修订内容
V1.0	2022-10-17	1359, 初版。



4 免责声明

本资料为晟矽微电子（以下简称“我司”）版权所有。

我司将力求资料内容准确无误，同时保留在不通知用户的情况下，对本资料内容的修改权。
如您需要获得最新的资料，请及时联系我司。

Sinomcu.com