

基于STM32F407HAL库串口DMA+空闲中断

核心板：STM32F407

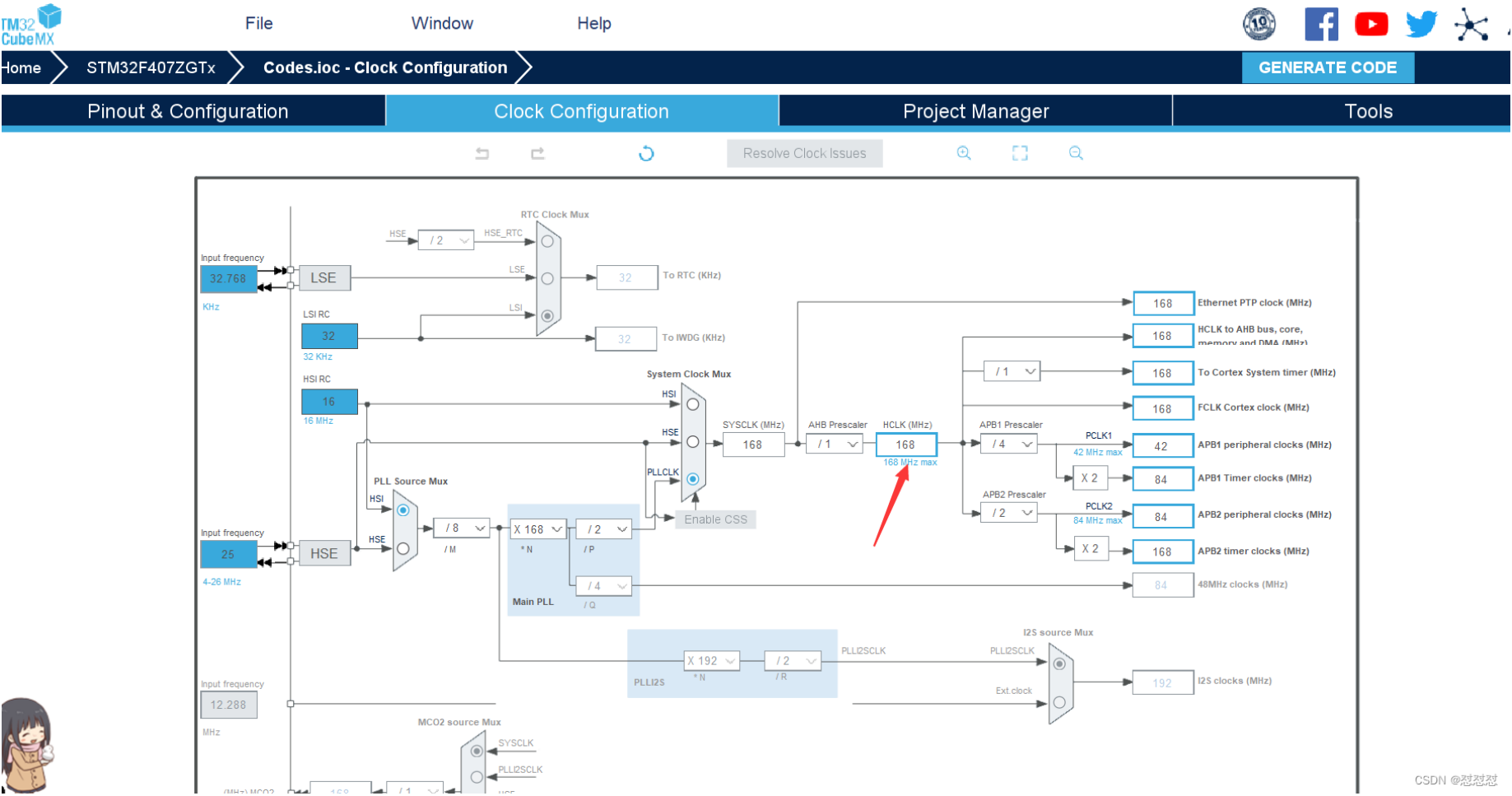
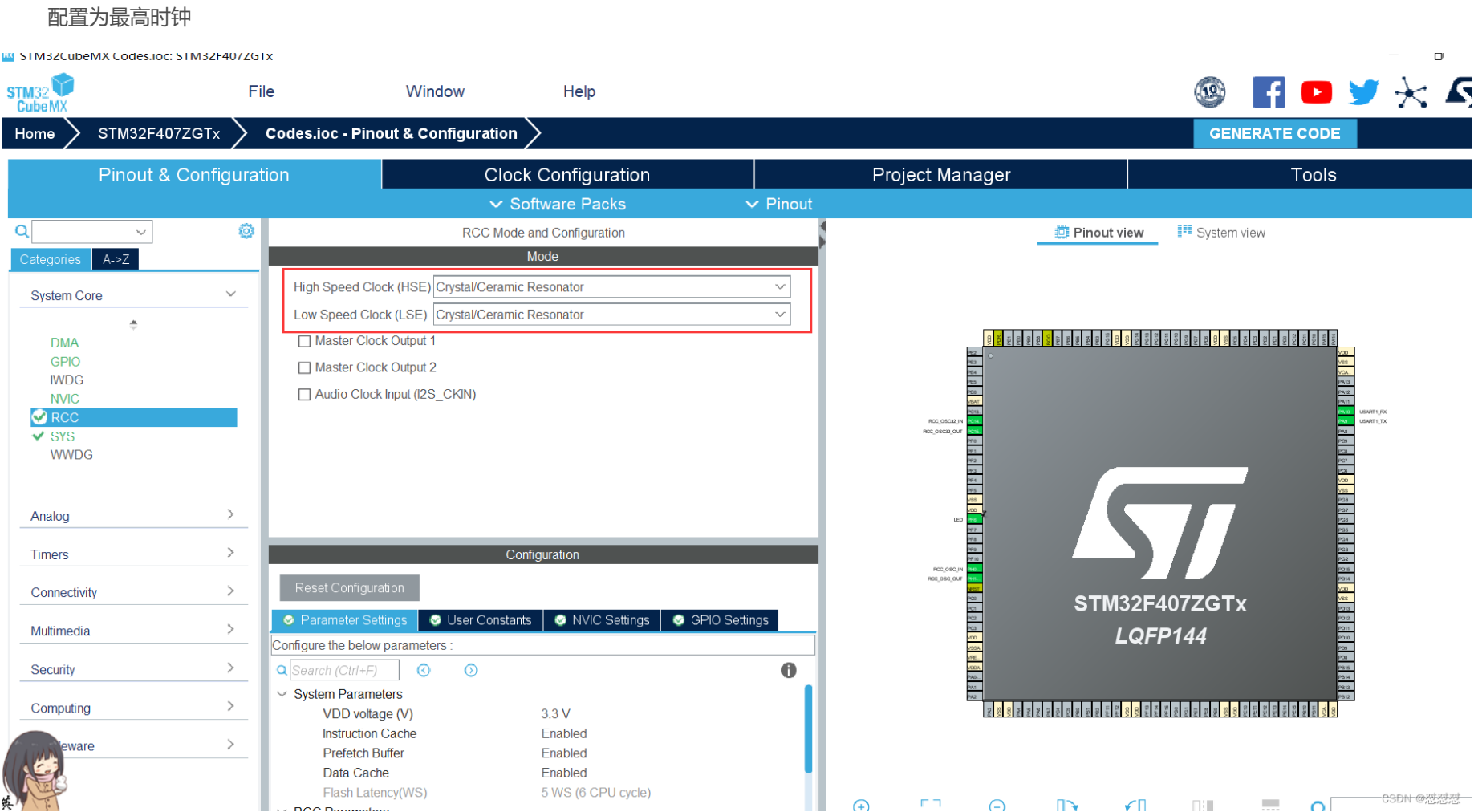
实验目的：通过DMA接收串口发来的数据，并且利用串口空闲中断在将这些数据发送至 串口助手 。

方法一：利用以下函数

```
1 /* 在DMA模式下接收一定数量的数据，直到接收到预期数量的数据或发生空闲事件 */
2 HAL_UARTEx_ReceiveToIdle_DMA(UART_HandleTypeDef *huart, uint8_t *pData, uint16_t Size);
3 /* 以DMA模式发送大量数据 */
4 HAL_UART_Transmit_DMA(UART_HandleTypeDef *huart, const uint8_t *pData, uint16_t Size);
5 /* 接待事件回调(使用高级接待服务后调用的Rx事件通知) */
6 void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size);
```

方法二：自己手动打开IDLE空闲中断，开启DMA接收模式，编写空闲中断中的逻辑代码。

首先利用STM32CubeMX 进行配置



配置串口DMA模式接收与发送

STM32CubeMX CODES.IOC - STM32F407ZGTx

File Window Help

Home STM32F407ZGTx Codes.ioc - Pinout & Configuration GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

Categories A->Z

System Core Analog Timers Connectivity

CAN1 CAN2 ETH FSMC I2C1 I2C2 I2C3 SDIO SPI1 SPI2 SPI3 UART4 UART5 **USART1** USART2 USART3 USART6 USB_OTG_FS USB_OTG_HS

USART1 Mode and Configuration

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Reset Configuration

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

Baud Rate 115200 Bits/s

Word Length 8 Bits (including Parity)

Parity None

Stop Bits 1

Advanced Parameters

Data Direction Receive and Transmit

Over Sampling 16 Samples

Pinout view System view

STM32F407ZGTx LQFP144

CSDN @赵想忍

开启串口全局中断

STM32CubeMX CODES.IOC - STM32F407ZGTx

File Window Help

Home STM32F407ZGTx Codes.ioc - Pinout & Configuration GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

Categories A->Z

System Core Analog Timers Connectivity

CAN1 CAN2 ETH FSMC I2C1 I2C2 I2C3 SDIO SPI1 SPI2 SPI3 UART4 UART5 **USART1** USART2 USART3 USART6 USB_OTG_FS USB_OTG_HS

USART1 Mode and Configuration

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Reset Configuration

NVIC Settings DMA Settings GPIO Settings

Parameter Settings User Constants

NVIC Interrupt Table

	Enabled	Preemption Priority	Sub Priority
USART1 global interrupt	<input checked="" type="checkbox"/>	0	0
DMA2 stream2 global interrupt	<input checked="" type="checkbox"/>	0	0
DMA2 stream7 global interrupt	<input checked="" type="checkbox"/>	0	0

Pinout view System view

STM32F407ZGTx LQFP144

CSDN @赵想忍

配置DMA，点击ADD进行添加，接收为循环模式，发送为正常模式

STM32CubeMX CODES.IOC - STM32F407ZGTx

File Window Help

Home STM32F407ZGTx Codes.ioc - Pinout & Configuration GENERATE CODE

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

Categories A->Z

System Core Analog Timers Connectivity

CAN1 CAN2 ETH FSMC I2C1 I2C2 I2C3 SDIO SPI1 SPI2 SPI3 UART4 UART5 **USART1** USART2 USART3 USART6 USB_OTG_FS USB_OTG_HS

USART1 Mode and Configuration

Mode

Mode Asynchronous

Hardware Flow Control (RS232) Disable

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings **DMA Settings** GPIO Settings

DMA Request	Stream	Direction	Priority
USART1_RX	DMA2 Stream 2	Peripheral To Memory	Low
USART1_TX	DMA2 Stream 7	Memory To Peripheral	Low

Add Delete

DMA Request Settings

Mode **Circular**

Increment Address ☐

Memory ☒

Use Fifo ☐ Threshold

Data Width Byte

Byte

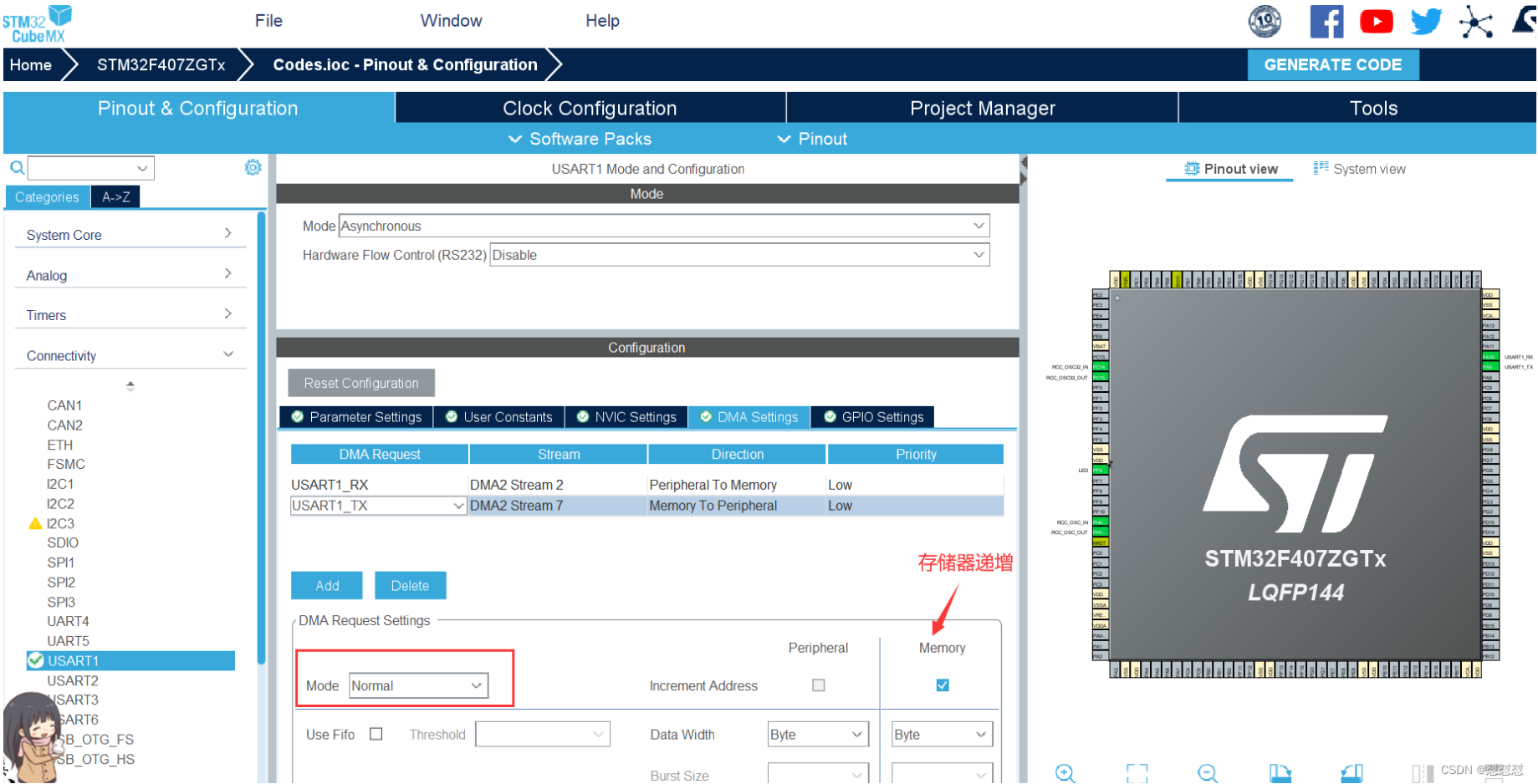
Burst Size

Byte

Pinout view System view

STM32F407ZGTx LQFP144

CSDN @赵想忍



生成代码。

方法一具体代码及步骤

打开main.c，在主函数外定义相关变量并且编写回调函数

```
1 char pData[255];
2
3 void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size)
4 { //Size为接收到的数据大小
5     if(huart->Instance == USART1)
6     {
7         HAL_UART_DMASStop(&huart1); //关闭是为了重新设置发送多少数据，不关闭会造成数据错误
8
9         HAL_UART_Transmit_DMA(&huart1, (uint8_t *)pData, Size); //设置DMA发送多少数据
10
11         HAL_UARTEx_ReceiveToIdle_DMA(&huart1, (uint8_t *)pData, 255); //继续开启空闲中断DMA发送
12     }
13 }
```

主函数

```
1 int main(void)
2 {
3     /* USER CODE BEGIN 1 */
4
5     /* USER CODE END 1 */
6
7     /* MCU Configuration-----*/
8
9     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
10    HAL_Init();
11
12    /* USER CODE BEGIN Init */
13
14    /* USER CODE END Init */
15
16    /* Configure the system clock */
17
18    SystemClock_Config();
19
20    /* USER CODE BEGIN SysInit */
21
22    /* USER CODE END SysInit */
23
24    /* Initialize all configured peripherals */
25    MX_GPIO_Init();
26    MX_DMA_Init();
27    MX_USART1_UART_Init();
28    /* USER CODE BEGIN 2 */
29
30    HAL_UARTEx_ReceiveToIdle_DMA(&huart1, (uint8_t *)pData, 255); //开启串口空闲中断DMA接收数据
31
32    /* USER CODE END 2 */
33
34    /* Infinite Loop */
35    /* USER CODE BEGIN WHILE */
36    while (1)
37    {
```

```
38 |      /* USER CODE END WHILE */39 |
39 |
40 |      /* USER CODE BEGIN 3 */
41 |  }
42 |  /* USER CODE END 3 */
43 | }
```

方法二具体代码及步骤

在main.c中声明变量并在main.h声明该变量为全局变量

```
1 | /* main.c中 */
2 | char pData[255];
3 |
4 | /* main.h中 */
5 | extern char pData[255];
```

在usart.c中编写空闲中断中的逻辑代码并在usart.h中声明

```
1 | void USER_UART_RxCpltCallback(UART_HandleTypeDef *huart)
2 | {
3 |     if(huart->Instance == USART1)
4 |     {
5 |         if(__HAL_UART_GET_FLAG(&huart1, UART_FLAG_IDLE) != RESET)
6 |         {
7 |             __HAL_UART_CLEAR_IDLEFLAG(&huart1); //清除IDLE标志
8 |
9 |             uint8_t Len = 255 - __HAL_DMA_GET_COUNTER(&hdma_usart1_rx);
10 |
11 |             HAL_UART_DMAStop(&huart1); //停止DMA，为了重新设置DMA发送多少数据
12 |
13 |             HAL_UART_Transmit_DMA(&huart1, (uint8_t *)pData, Len);
14 |
15 |             HAL_UART_Receive_DMA(&huart1, (uint8_t *)pData, 255);
16 |         }
17 |     }
18 | }
```

在stm32f4xx_it.c中包含usart.h，并在void USART1_IRQHandler()中调用USER_UART_RxCpltCallback(&huart1);

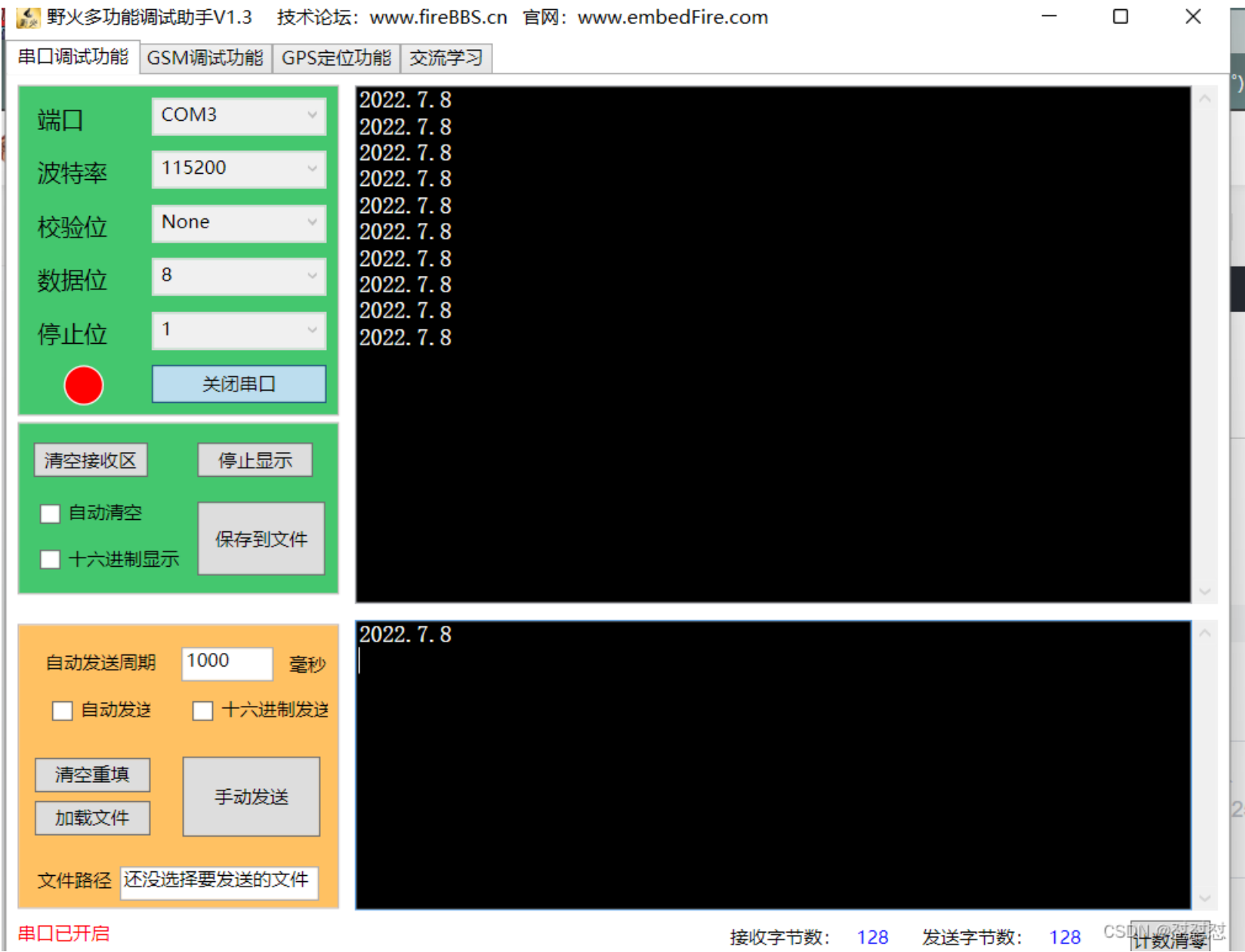
```
1 | void USART1_IRQHandler(void)
2 | {
3 |     /* USER CODE BEGIN USART1_IRQn 0 */
4 |
5 |     USER_UART_RxCpltCallback(&huart1);
6 |
7 |     /* USER CODE END USART1_IRQn 0 */
8 |     HAL_UART_IRQHandler(&huart1);
9 |     /* USER CODE BEGIN USART1_IRQn 1 */
10 |
11 |     /* USER CODE END USART1_IRQn 1 */
12 | }
```

在主函数中开启空闲中断，开启DMA接收

```
1 | int main(void)
2 | {
3 |     /* USER CODE BEGIN 1 */
4 |
5 |     /* USER CODE END 1 */
6 |
7 |     /* MCU Configuration-----*/
8 |
9 |     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
10 | HAL_Init();
11 |
12 |     /* USER CODE BEGIN Init */
13 |
14 |     /* USER CODE END Init */
15 |
16 |     /* Configure the system clock */
17 | SystemClock_Config();
18 |
19 |     /* USER CODE BEGIN SysInit */
20 |
21 |     /* USER CODE END SysInit */
22 |
23 |     /* Initialize all configured peripherals */
24 | MX_GPIO_Init();
25 | MX_DMA_Init();
26 | MX_USART1_UART_Init();
```

```
27 | /* USER CODE BEGIN 2 */28 |
29 | __HAL_UART_ENABLE_IT(&huart1, UART_IT_IDLE); //使能串UART1 IDLE中断
30 |
31 | HAL_UART_Receive_DMA(&huart1, (uint8_t *)pData, 255); //开启DMA接收模式
32 |
33 | /* USER CODE END 2 */
34 |
35 | /* Infinite Loop */
36 | /* USER CODE BEGIN WHILE */
37 | while (1)
38 | {
39 |     /* USER CODE END WHILE */
40 |
41 |     /* USER CODE BEGIN 3 */
42 | }
43 | /* USER CODE END 3 */
44 | }
```

实验结果：（两个都一样）



对于不太清除为什么DMA要不断使能失能可以看一下下面这个文章

浅谈使用DMA时需要的不断使能失能的原因

以上仅供自己与大家学习积累，欢迎各位大佬批评与指正！