串口 DMA 接收不定长数据的一种方法

关键字：串口，不定长，RTO

1. 前言

使用串口接收不定长数据时，可以有多种方法，比如最常见的有额外使能一个定时器，在超过定时范围未收到后续的字节时，认为此帧结束；或者利用 IDLE 中断，当数据空闲时，自动产生中断；亦或每接收到一个字节后都通过应用程序进行一次处理。这次我们介绍另外一种方法，在 DMA 方式下利用硬件接收超时中断（Receiver timeout interrupt）实现不定长数据的接收。

2. 实现原理

首先，并非所有的 STM32 系列的 MCU，也并非所有的 USART/UART 外设都支持 Receiver timeout （RTO）特性，具体的支持情况，可以通过对应芯片的参考手册去查询。

对于 MCU 的配置，可以通过 USART_RTOR 寄存器的 RTOEN 位使能接收超时功能，通过 RTO 位域配置超时时间，时间单位为传输一个数据位的时间（即波特率）。

接收超时在停止位设置不同的情况下，计时的起始位置不同：

1. 如果 stop = ' 00 '或 stop = ' 11 '，从停止位的末尾开始。
2. 如果 stop = ' 10 '，则从第二个停止位的末尾开始。
3. 如果 stop = ' 01 '，则从停止位的起始开始。

当计数超过 RTO 位域的设定值时，且使能了 RTOIE 位，就会产生一个错误中断，此时我们就可以认为此帧结束，进而进行处理。

3. 应用示例

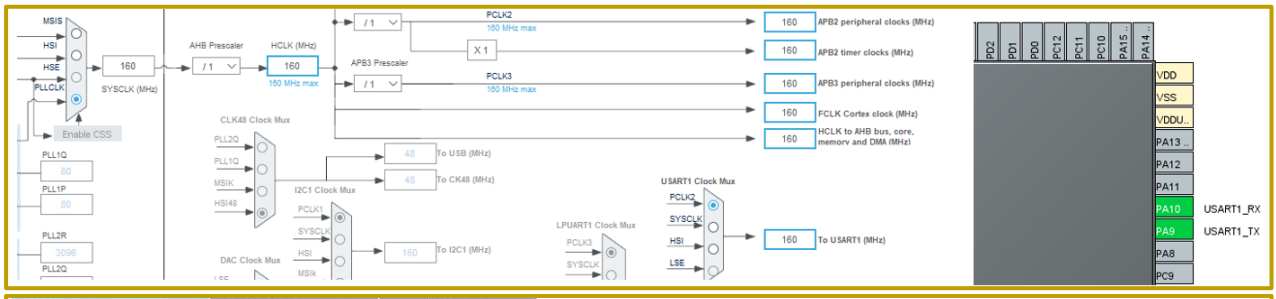
HAL 库已经为我们提供了操作 RTO 相关的 API，应用开发过程，我们直接调用即可。

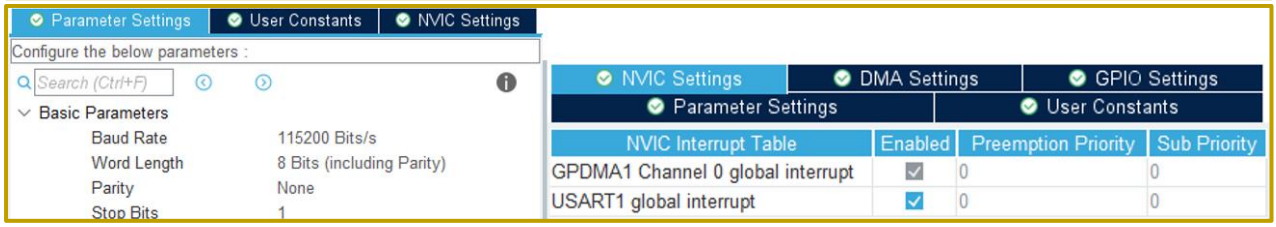
```
void HAL_UART_ReceiverTimeout_Config(UART_HandleTypeDef *huart, uint32_t
TimeoutValue);
HAL_StatusTypeDef HAL_UART_EnableReceiverTimeout(UART_HandleTypeDef *huart);
HAL_StatusTypeDef HAL_UART_DisableReceiverTimeout(UART_HandleTypeDef *huart);
```

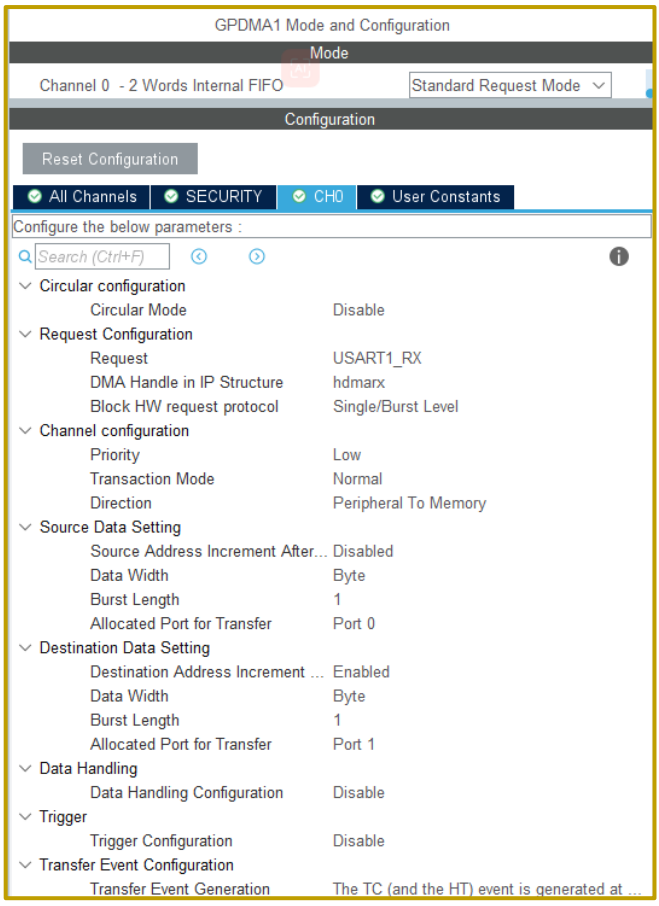
我们以 STM32U575ZIT6 为例，配置一个测试工程。

1. 系统时钟配置为 160MHz

2. 配置 USART1 为 Asynchrones，管脚配置为 PA9，PA10
3. USART1 参数：115200bits/s，8bit，None，1Stop
4. 使能 USART1 中断
5. 配置 USART1_RX GPDMA







生成工程后，配置超时时间，使能接收超时功能，使能串口的 DMA 接收，处理串口的错误回调函数（接收超时后，HAL 库中经 HAL_UART_ErrorCallback()回调）。

```
/* USER CODE BEGIN 2 */
```

```
  HAL_UART_ReceiverTimeout_Config(&huart1, huart1.Init.BaudRate/100); //10ms BaudRate
= 115200
  HAL_UART_EnableReceiverTimeout(&huart1);
  HAL_UART_Receive_DMA(&huart1, Uart_RcvBuf, MAX_UART_RCV_LEN);
  /* USER CODE END 2 */
```

```
#define MAX_UART_RCV_LEN   10 //config the max receive data length
uint8_t Uart_RcvBuf[MAX_UART_RCV_LEN];
uint16_t Uart_RcvLen;


void HAL_UART_ErrorCallback(UART_HandleTypeDef *huart)
{
 if(huart == &huart1)
 {
   /* Check if DMA mode is enabled in UART */
  if (HAL_IS_BIT_SET(huart->Instance->CR3, USART_CR3_DMAR))
  {
    if((huart->ErrorCode & HAL_UART_ERROR_RTO) == HAL_UART_ERROR_RTO)
    {
       uint16_t nb_remaining_rx_data = (uint16_t)
      __HAL_DMA_GET_COUNTER(huart->hdmarx);
       Uart_RcvLen = MAX_UART_RCV_LEN - nb_remaining_rx_data;
    }
    HAL_UART_Receive_DMA(&huart1, Uart_RcvBuf, MAX_UART_RCV_LEN);
  }
 }
}
```

HAL_UART_ErrorCallback 是一个支持所有错误中断的回调函数，在处理 RTO 错误引起的中断响应时，可以通过 ErrorCode 增加相应的判断。

通过调用__HAL_DMA_GET_COUNTER(huart->hdmarx)可以得到 DMA 未搬移完的字节数，从而得到已搬移完成的字节数。

4. 总结

利用 RTO 方式接收不定长串口数据，相比用定时器计时，可以节省一个硬件定时器资源，同时减少了定时器每次进入中断处理的运行时间；相比 IDLE 的方式，可以更灵活的配置超时时间。但需注意的是，所选用的 MCU 以及所使用的 USART/UART 是否支持这个特性要确认好。

参考文献

文件编号	文件标题	版本号	发布日期
RM0456	STM32U5 Series Arm®-based 32-bit MCUs	Rev 4	Feb 2023

文档中所用到的工具及版本

STM32CubeMX 6.8.1、IAR(9.20.4)

LAT 中的附件

USART_ReveiveTimeout.7z

版本历史

日期	版本	变更
2023 年 09 月 20 日	1.0	首版发布

重要通知 – 请仔细阅读

意法半导体公司及其子公司 （“ST”）保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。 ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 *www.st.com/trademarks*。所有其他产品或服务名称均为其 各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码，电路图等）信息，我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。