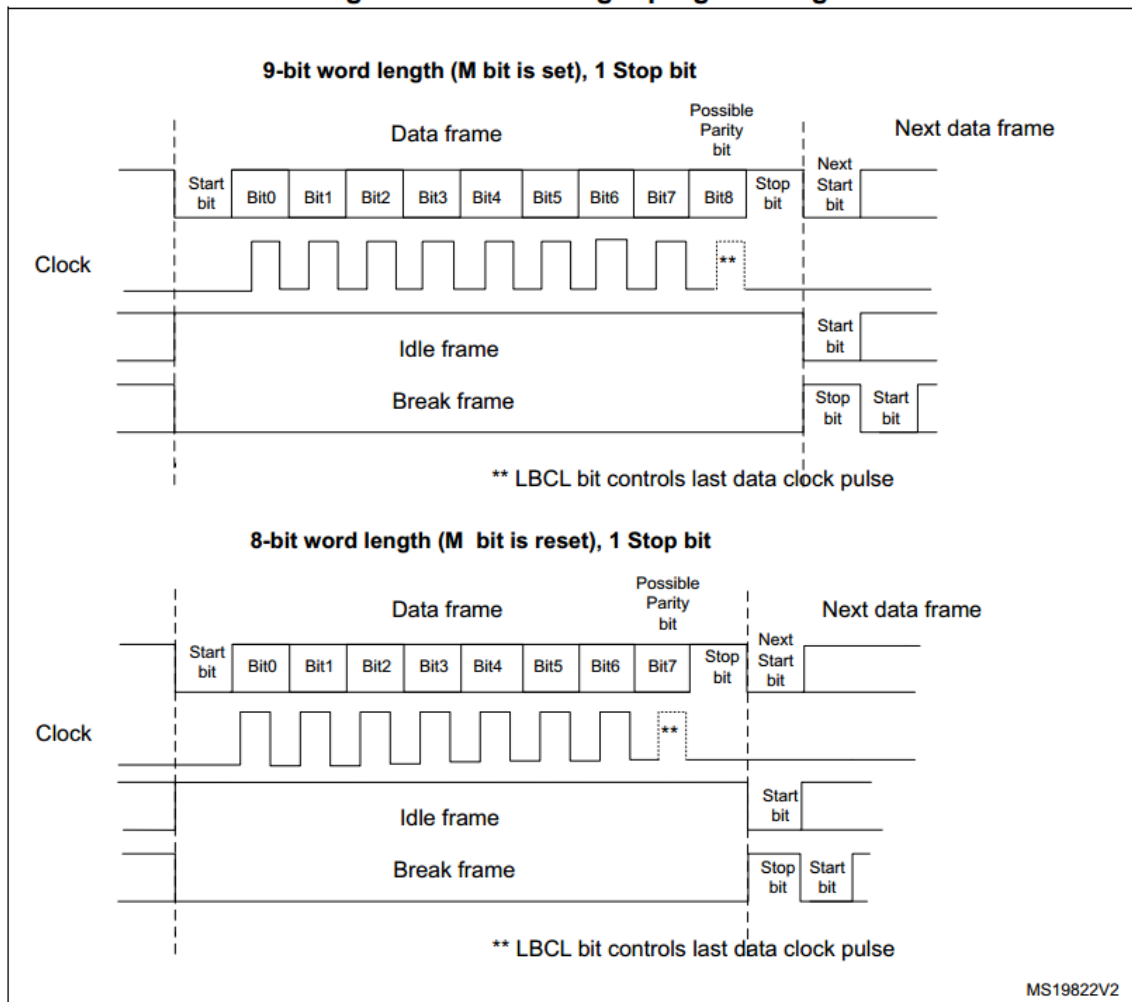


# 串口uart

## stm32参考手册的一些笔记

- 串口是相关协议

Figure 297. Word length programming



- 起始位（低电平）+数据位+奇偶校验位+停止位（高电平0.5或者1或者1.5或者2个时钟周期）
- 串口波特率计算公式

### Equation 1: Baud rate for standard USART (SPI mode included)

$$\text{Tx/Rx baud} = \frac{f_{\text{CK}}}{8 \times (2 - \text{OVER8}) \times \text{USARTDIV}}$$

过采样 ← 时钟频率 ← 寄存器的值

### Equation 2: Baud rate in Smartcard, LIN and IrDA modes

$$\text{Tx/Rx baud} = \frac{f_{\text{CK}}}{16 \times \text{USARTDIV}}$$

上图的时钟频率为外设时钟频率APBx

- 接收完成和发送完成中断分别是在中断方式发送完成时产生，在中断方式接收缓存满了时产生。
- 使用串口DMA发送数据的时候，DMA发送完所有数据之后，一个串口发送完成中断会产生。为了确保数据能够完整地发送出去，需要在监测到TC=1也就是发送完成中断标志位置1的时候，再去把串口关闭或者进入Stop mode。
- **硬件流控**：RTS拉低的时候，接收器准备就绪接收新的数据。接收寄存器满的时候，RTS被拉高，说明需要在传输完这个帧（一个字节）之后停止传输。CTS被拉低的时候，下一个数据将会传输。在CTS被拉高之后，则不能传输数据。在CTS引脚跳变的时候，CTSIF状态位会自动的被硬件置位，在CTSIF被置位的时候，会产生一个中断。

## 串口使用中断方式接收数据的相关问题

- 在串口使用中断方式接收数据的时候，需要在初始化完成之后才使能接收中断，也就是说，在MX\_USART1\_UART\_Init()之后才使用HAL\_UART\_Receive\_IT(&huart8, \_\_g\_rev\_char, 10)
- 一般来说，在使用HAL\_UART\_Receive\_IT(&huart8, \_\_g\_rev\_char, 10)语句之后，就将接收中断使能并且设置接收缓存的大小为10个字节。
- 接收到一个字节之后就中断一次，然后，在HAL\_UART\_IRQHandler(&huart8)函数中将DR寄存器中的数据拷贝到相应的缓存区中，在还没有接收够10个字节之前，接收器的状态都处于忙的状态（也就是可以从RX线上接收数据）。
- 在接收完10个字节的数据之后，将会调用一个接收完成回调函数（STM32的HAL库）。如果没有，则需要观察接收寄存器。
- 在调试过程中，如果在接收中断里面打断点，会造成数据丢失的情况发生（一个字节中断一次）。
- 在使用stm32的HAL库的时候，应该在**接收完成的回调函数**里面去重新打开串口接收中断使能。