7.26

1.

问题：载入到板子上后发现亮灯时间不对。

原因：hal\_delay(n)实际上是延时n+1ms。

解决方法：后面的视频有讲。

7.30

1.问题：输入"off"后输入"on"无反应，但输入"on"后输入"off"有反应

原因：针对您提到的问题（输入"off"后输入"on"无反应，但输入"on"后输入"off"有反应），可能的原因是buf中残留了上一次接收的数据。为了解决这个问题，您可以在重新启动DMA接收之前清除buf：

c

void HAL\_UARTEx\_RxEventCallback(UART\_HandleTypeDef \*huart, uint16\_t Size)

{

if (huart == &huart1)

{

// 在处理接收到的数据之前，先清零buf，以避免旧数据的干扰

memset(buf, 0, sizeof(buf)); // 假设buf是一个字符数组

// 您的数据处理逻辑...

HAL\_UART\_Transmit(&huart1, "successfully receive\r\n", sizeof("successfully receive\r\n") - 1, 100);

// 重新启动DMA接收

HAL\_UARTEx\_ReceiveToIdle\_DMA(&huart1, buf, 100);

}

}

请注意，memset(buf, 0, sizeof(buf))用于清除buf中的内容，以确保下一次接收时缓冲区是干净的。同时，在HAL\_UART\_Transmit中，我减去了1，以确保不会发送字符串结尾的null字符。

此外，确保buf是一个足够大的字符数组，以存储接收到的数据和null终止符，例如：

c

char buf[101]; // 假设最大接收100个字符，加上1个字符用于null终止符

最后，请确保在您的主程序或初始化代码中正确配置了UART和DMA，并且已经启动了初始的DMA接收操作。

解决方法：发送代码和问题给文心一言。

2.问题：使用haldelay后呼吸灯无法正常运行

原因：避免使用HAL\_Delay()：

在中断服务程序（ISR）中，不推荐使用HAL\_Delay()函数，因为它会阻塞CPU，导致中断服务程序不能及时返回，从而影响系统的实时性。HAL\_Delay()是基于SysTick定时器的，它会阻塞当前线程直到达到指定的延时时间，这在中断服务程序中是不可接受的。  
解决方法：询问文心一言。参考：[STM32 HAL库 中断处理中使用延时函数（HAL\_Delay）\_stm32f103 hal 延时 死循环-CSDN博客](https://blog.csdn.net/Dir_x/article/details/128899621)