



机械与能源工程系

SUSTech

Department of  
Mechanical and Energy  
Engineering

# 实 验 报 告

课程名称： 机电一体化

课程编号： ME3333

实验题目： DMA 实验

学 号： 12313215

姓 名： 梁家源

专 业： 机器人工程

指导教师： 柯文德 曾千里

实验成绩：

实验日期： 2025 年 5 月 14 日

# DMA 实验

## 一、实验目的

本实验旨在掌握 STM32F7 开发板上 DMA 的配置与应用，结合红外遥控器实现字符串的动态生成、缓存管理、LCD 实时显示及 DMA 传输至串口助手。

## 二、实验仪器和用具

主要仪器设备：

1. 计算机
2. Keil uVision 软件

## 三、实验方法与步骤

### 1. 实验原理

DMA 允许外设和内存之间直接交换数据，无需 CPU 干预，从而减少中断开销并提高整体系统性能。STM32F767 系列提供两个 DMA 控制器(DMA1 和 DMA2)，每个控制器最多有 8 个数据流和 8 个通道，可以实现多通道并发数据流传输。本实验采用 DMA2\_Stream7\_Channel4 实现 USART1 的发送通道配置，结合遥控器控制数据生成与触发发送行为。

### 2. 遥控防抖机制

确保每次只响应一次按键

```
if (key != 0 && key != last_key)
{
    delay_ms(20);
    if (key == Remote_Scan())
    {
        last_key = key; // 更新状态，只处理一次
    }
}
if (key == 0) last_key = 0;
```

### 3. Up 键生成随机字符串

编写 generate\_random\_string() 函数以生成随机长度随机内容的字符串

```
void generate_random_string(char *dest, int max_len) {
    const char charset[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
    int len = 10 + rand() % (max_len - 10); // 随机长度: 10 ~ max_len-1
    int i;
    for (i = 0; i < len; ++i) {
        int key = rand() % (int)(sizeof(charset) - 1);
        dest[i] = charset[key];
    }
    dest[len] = '\0';
}
```

```

case 98: // UP
    generate_random_string(str_buf, MAX_STR_LEN);
    LCD_Fill(30, 150, 470, 166, WHITE);
    LCD_ShowString(30, 150, 240, 16, 16, str_buf);
    break;

if(ir_val && ir_val != last_ir_val)
{
    last_ir_val = ir_val; // 更新锁
    if(ir_val == 98)
    {
        adc_enabled = 1;
    }
    else if(ir_val == 168)
    {
        adc_enabled = 0;
    }
}
else if(ir_val == 0)
{
    last_ir_val = 0; // 清空上次记录，等待新按键
}

```

#### 4. Down 键保存字符串到缓冲区，并追加 \r\n

增加换行符使得在串口助手能够换行显示每条字符串

```

case 168: // DOWN
{
    int len = strlen(str_buf);
    if (buf_offset + len + 2 < SEND_BUF_SIZE)
    {
        // 拷贝字符串
        memcpy(&SendBuff[buf_offset], str_buf, len);
        buf_offset += len;

        // 添加换行符
        SendBuff[buf_offset++] = '\r';
        SendBuff[buf_offset++] = '\n';

        str_count++;
        LCD_Fill(30, 170, 240, 186, WHITE);
        LCD_ShowString(30, 170, 240, 16, 16, "Saved:");
        LCD_ShowNum(90, 170, str_count, 4, 16);
    }
    else
    {
        LCD_ShowString(30, 190, 240, 16, 16, "Buffer Full!");
    }
}
break;

```

#### 5. Left 键启动 DMA 传输字符串内容

```

case 34: // LEFT
if (buf_offset > 0) {
    LCD_ShowString(30, 190, 240, 16, 16, "Sending...");
    HAL_UART_Transmit_DMA(&UART1_Handler, SendBuff, buf_offset);
    while ( __HAL_DMA_GET_FLAG(&UART1TxDMA_Handler, DMA_FLAG_TCIF3_7) == RESET);
    __HAL_DMA_CLEAR_FLAG(&UART1TxDMA_Handler, DMA_FLAG_TCIF3_7);
    HAL_UART_DMAStop(&UART1_Handler);
    LCD_ShowString(30, 190, 240, 16, 16, "Sent Successfully ");
}
break;

```

## 6.Right 键清空缓冲区和计数器

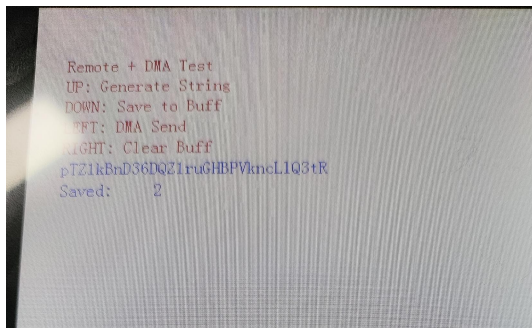
```
case 194: // RIGHT
    memset(SendBuff, 0, SEND_BUF_SIZE);
    buf_offset = 0;
    str_count = 0;
    LCD_Fill(30, 170, 240, 206, WHITE);
    LCD_ShowString(30, 170, 240, 16, 16, "Cleared. Count: 0");
    break;
```

# 四、实验分析及结论

## 1. 实验结果

LCD 显示当前生成字符串内容、已存条数；串口助手分行接收多个字符串。

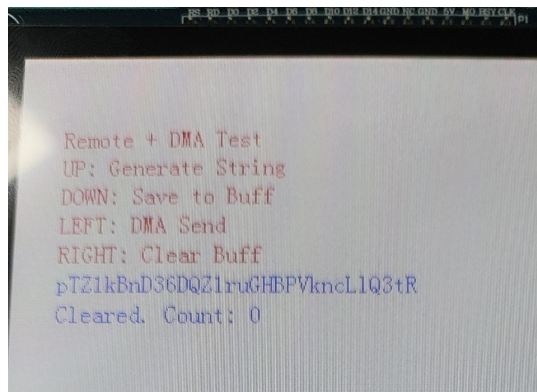
按 Up 和 Down 后存储字符串：



按 Left 后传输至串口助手：



按 Down 后清空：



## 2. 实验总结

通过本实验，掌握了 DMA 配置流程、USART 数据发送与 LCD 显示的集成应用。红外遥控器输入结合动态字符串处理逻辑，是实际嵌入式系统中常见的人机交互模式。同时，通过加入防抖和按键识别机制，提高了系统的稳定性。DMA 的使用明显减轻了 CPU 负担，使系统具有良好的实时性和可扩展性。