



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： CAN 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 21 日

CAN 实验

一、实验目的

掌握 STM32F767 内置 CAN 控制器的初始化及发送、接收流程。了解 CAN 协议物理层结构及显性/隐性电平机制。实现两个 STM32 开发板之间通过 CAN 通信传输数据。掌握 RS485 串口通信接口及双板数据回传实现方式。综合运用 CAN + RS485，实现多总线协同通信流程。

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

CAN (Controller Area Network) 是 ISO11898 标准串行通信协议。采用两根线差分通信 (CAN_H 与 CAN_L)，电平分为显性 (逻辑 0) 与隐性 (逻辑 1)。支持多主通信、优先级仲裁、错误检测和自动恢复功能。支持标准帧 (11 位 ID) 与扩展帧 (29 位 ID)。

2. 板号自动识别：

读取 PA0 电平作为判断依据；若为低电平 (接地)，认为是板 1 (主板)；若为高电平 (悬空或上拉)，则为板 2 (从板)；此机制允许将相同固件烧录进两块开发板。

```
u8 Read_Board_ID(void)
{
    return HAL_GPIO_ReadPin(BOARD_ID_GPIO, BOARD_ID_PIN);
}
```

3. 串口接收中断

串口接收使用中断方式；通过 USART_RX_STA 标志位记录接收长度；接收到回车符 (\r) 即认为数据接收完成；缓冲区上限为 64 字节。

```

void USART1_IRQHandler(void)
{
    u8 res;
    if (__HAL_UART_GET_FLAG(&UART1_Handler, UART_FLAG_RXNE) != RESET)
    {
        HAL_UART_CLEAR_FLAG(&UART1_Handler, UART_FLAG_RXNE);
        res = (uint8_t)(READ_REG(UART1_Handler.Instance->RDR) & 0xFF);
        if ((USART_RX_STA & 0x8000) == 0)
        {
            if (USART_RX_STA < 64)
            {
                if (res == '\r') USART_RX_STA |= 0x8000;
                else usart_rx_buf[USART_RX_STA++] = res;
            }
            else USART_RX_STA |= 0x8000;
        }
    }
}

```

4. 按键触发 CAN 发送（主板）

用户通过串口输入数据后，按下 KEY0；最多发送 8 字节至 CAN；发送完成后清除接收缓存；LCD 显示提示信息“CAN Sent”。

```

if (KEY_Scan(0) == KEY0_PRES)
{
    for (i = 0; i < 8 && i < USART_RX_STA; i++)
        can_rx_buf[i] = usart_rx_buf[i];

    CAN1_Send_Msg(can_rx_buf, i);
    LCD_ShowString(30, 140, 200, 16, 16, "CAN Sent.");
    USART_RX_STA = 0;
}

```

5. 从板接收 CAN 并回传（RS485）

从 CAN 控制器读取接收到的数据；显示在 LCD 屏幕；通过 RS485 发送回主板；使用 RS485_Send_Data 实现自动方向控制。

```

else // 板2: CAN → RS485
{
    len = CAN1_Receive_Msg(can_rx_buf);
    if (len > 0)
    {
        LCD_ShowString(30, 100, 200, 16, 16, "CAN Received:");
        LCD_ShowString(30, 120, 200, 16, 16, (u8*)can_rx_buf);
        RS485_Send_Data(can_rx_buf, len);
    }
}

```

6. 主板接收 RS485 数据并回显串口

轮询检测 RS485 是否接收到回传数据；显示至 LCD；并逐字节写入串口发送寄存器，实现串口助手回显。

```

RS485_Receive_Data(rs485_rx_buf, &rs485_rx_cnt);
if (rs485_rx_cnt > 0)
{
    LCD_ShowString(30, 170, 200, 16, 16, "RS485 Received:");
    LCD_ShowString(30, 190, 200, 16, 16, (u8*)rs485_rx_buf);
    for (i = 0; i < rs485_rx_cnt; i++)
        WRITE_REG(UART1_Handler.Instance->TDR, rs485_rx_buf[i]);
    rs485_rx_cnt = 0;
}

```

四、实验分析及结论

1. 实验结果

板 1 连接串口助手，输入“123ABC\r”；按下 KEY0；LCD 提示“CAN Sent”；板 2 接收数据，通过 RS485 发送回；板 1 串口助手显示返回内容“123ABC”，LCD 也显示“RS485 Received”。

结果：

数据成功在两板之间传输；LCD 与串口反馈一致；按键控制生效，通信逻辑正确

2. 实验总结

通过本次实验，掌握了 STM32F767 中 CAN 控制器的配置与使用流程，进一步结合 RS485 串口通信，实现了一个完整的多总线、双板协同通信系统。关键编程技能包括中断接收、CAN 帧构造与发送、RS485 方向控制、LCD 信息反馈及 GPIO 输入识别等。