

# 基于 RS485 总线的评分系统实验报告

班级：计科 2002 班 学号：202004061409 姓名：杨鹏宇

## 一.实验目的

通过本案例加深理解 RS485 通信方式，实现上位机的主控制器与所有的下位机进行通信。

## 二.实验过程

### 1.实验目标

通过 RS485 实现两单片机的通信，从而实现一个评分系统。评分系统的设备包括主机和从机，从机可以进行打分，并设置从机编号，评分系统程序向主机发送数据包，主机根据功能通过 RS485 向从机发送功能对应的数据包，从机接收数据包后将回复数据包，主机将数据包发到评分系统，评分系统对数据包进行解析，实现获取评分数据和设备检测等功能。

评分系统需要实现的主要为三个功能：

检测从机是否正常；获取从机评分；评分结束后对从机进行复位

### 2.实验知识点

#### (1)RS485

RS485 属于半双工通信，数据可以在一个信号载体的两个方向上传输，但是不能同时进行传输。电平转换采用差分电路方式，A、B 两线的电压差大于 0.2 认为是逻辑“1”，小于-0.2 认为是逻辑“0”，只有通信双方一方处于发送，一方处于接收时，通信才能正常进行。使用 RS485 进行单片机的串口通信需要考虑数据冲突的问题。

#### (2)通信协议

##### 主机向从机发送：

设备检测：数据包头 + 从机地址 + 检测功能码 (Fun\_CheckSlave) + 自定义内容 (Check\_Content) + 校验字节

获取从机评分：数据包头 + 检测正常从机地址+读下位机功能码 (Fun\_ReadInfo) + 从机地址 + 校验字节

复位：数据包头+广播地址+复位功能码 (Fun\_Reset) + 0x00 + 校验字节

### 从机回复:

回应查询: 数据包头 + 从机地址 + 检测功能码 (Fun\_CheckSlave) + 自定义内容 (主机 Check\_Content) + 校验字节

回应错误: 数据包头 + 从机地址 + 检测功能码 (Fun\_CheckSlave) + 错误码 (ErrorInfo) + 校验字节

结果返回: 数据包头 + 从机地址 + 读下位机功能码 (Fun\_ReadInfo) + 从机返回的分数值 + 校验字节

## 3.程序设计

按照评分系统需要实现的功能, 即主机需要发送的数据进行程序设计。程序使用 c++编写。

### (1)串口读写

串口读写参考串口实验中老师给出的参考代码改写, 为了方便读写, 定义了 serial 类, 包含文件句柄和 epoll 事件。并定义相关函数。在构造函数中对事件监听进行设置, 读写函数中, 将读取和写入的数据都保存在 vector 中。

写入数据时直接调用 write 进行写入, count 计数保证数据全部写入。读取数据时, 等待事件后进行读取, 为了保证读取足够字节的数据, 循环进行读取。在前面的串口实验中, 单片机连续重复发出数据, 因此可以通过多次读取保证读取一定字节的数据, 在本次实验中, 由于单片机只发送一次数据, 因此 while 循环没有实际意义。

```
std::vector<unsigned char> serial::myRead(size_t n) const {
    size_t count = 0;
    std::vector<unsigned char> buffer(n);
    while (count < n) {
        epoll_event event1 = event;
        // 等待数据, 指定超时值, 避免无限期阻塞等待, 没有等到则不读取
        if(epoll_wait(epfd, &event1, 1, 5000) == 0) continue;
        // 读取数据, 然后根据读取到的数据数量决定是否需要继续读取
        count += ::read(board, &buffer[count], n-count);
    }
    return buffer;
}

void serial::myWrite(const std::vector<unsigned char> &data) const {
    size_t count = 0;
    while (count < data.size()) {
        // 向串口写入数据
        count += ::write(board, &data[count], data.size() - count);
    }
}
```

## (2)评分功能实现

评分功能实现需要完成发送与接收数据相关的函数，其中关于数据包中功能码的定义如下：

功能码	读下位机功能码	0X03
	检测功能码	0X08
	地址错误功能码	0X10
	复位功能码	0X01
附加数据	错误码	0X6F
	包头	0X5A
	广播地址	0X00
	自定义内容	0X13

### 设备检测功能

设备检测需要先发送设备检测数据包，然后接受从机数据包，进行数据分析。设备检测功能通过 check 函数实现。发送的数据为协议定义的 5a + 从机地址 + 检测功能码(08) + 自定义内容 check\_content(13) + 校验码。其中地址为用户输入并作为参数传入该函数，校验码使用累加和，将传入的地址加上其他位的值得到。然后将地址和校验码插入 vector 中，写入数据。发送数据后睡眠 1s 等待从机作出反应，然后读取数据。首先检验校验码是否正确，错误则直接返回宏定义的 check\_failed。如果校验正确，判断是否得到正确的查询回应，即与主机发送的数据是否相同，如果相同返回 1，不相同则判断回应查询是否得到了错误回应(0x6f)，并返回 0。

```
#define error -1
#define check_failed -2
//设备正常检测: 5a + 从机地址 + 检测功能码(08) + check_content(13) + 校验码
int check(serial &serial1,int addr){
    int check_code = 117 + addr, ret; //校验码为累加和
    vector<uchar> code = {0x5a, 0x08, 0x13};
    vector<uchar> rec;
    code.insert(code.begin()+1, (uchar)addr); //插入从机地址
    code.push_back((uchar)check_code); //插入校验码
    /* 写入并接收回应数据包 */
    serial1.myWrite(code);
    sleep(1);
    rec = serial1.myRead(5);
    if(rec[0] + rec[1] + rec[2] + rec[3] != rec[4]) return check_failed; //校验不通过
    if(format(rec) == format(code)) //检验接收数据包是否与发送相同，相同则从机地址正确
        ret = 1;
    else if(rec[3] == 0x6f) ret = error; //回应错误
    return ret;
}
```

## 获取分数功能

获取分数功能定义 `get_score` 函数实现。与检测设备类似，先发送查询分数的数据包，然后获取从机回应的数据包进行分析。具体实现与设备检测类似，通过传入的从机地址计算校验码，然后将数据写入串口。写入后再读取串口数据，检验校验码，检查是否错误，如果没有异常，将分数值转换为 10 进制数字的整型返回。

```
//获取从机分数: 5a + 00 + 读取功能码03 + 从机地址 + 校验字节
int get_score(serial &serial1, int addr){
    int check_code = 93 + addr, ret;           //校验码为累加和
    vector<uchar> code = {0x5a, 0x00, 0x03};
    vector<uchar> rec;
    code.insert(code.begin()+3, (uchar)addr);  //插入从机地址
    code.push_back((uchar)check_code);         //插入校验码
    /* 写入并接收回应数据包 */
    serial1.myWrite(code);
    sleep(1);
    rec = serial1.myRead(5);
    if(rec[0] + rec[1] + rec[2] + rec[3] != rec[4]) return check_failed; //校验不通过
    if(rec[3] == 0x6f) ret = error;           //回应错误
    else
        ret = (int)rec[3];                   //整型返回
    return ret;
}
```

## 从机复位

复位功能定义 `reset` 函数实现。复位只需要发送数据，不需要读取数据，且数据是固定的 5a0001005b，因此每次只需要向串口写入该数据。经过测试，从机不一定能收到主机发送的数据，并成功复位，因此通过多次发送复位数据包的方式，确保从机能够复位。其实在实现设备检测和获取分数时，也存在从机无法成功接收数据而无相应的情况。但是由于设备检测和获取分数需要读取从机发送的数据，如果重复向从机发送数据，从机回复数据时就会产生总线冲突，导致数据错误，因此前两个功能没有重复向串口写入数据。

```
//从机复位: 5a + 广播地址00 + 复位功能码01 + 00 + 校验字节
void reset(serial &serial1){
    vector<uchar> code = {0x5a, 0x00, 0x01, 0x00, 0x5b};
    /* 发送数据包 */
    for(int i=0; i<500; i++) {
        serial1.myWrite(code);
    }
    cout<<"从机已复位"<<endl;
    return;
}
```

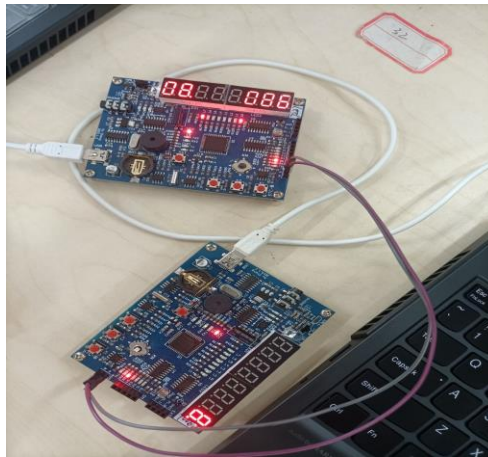
### (3)主函数

main 函数中接收用户输入，主要接收从机地址，复位命令和退出程序的命令。并调用上述实现的函数进行与从机的通信。在调用函数后对返回值进行检查，判断是否发生了错误，并对不同的错误给出输出。具体实现见工程文件。

## 4.程序测试及结果分析

### 功能测试

将单片机程序下载到单片机，将主机与从机分别连接到计算机，在主机连接的计算机上运行程序。从机首先设置一个地址和分数，设置地址为 8，分数为 86：



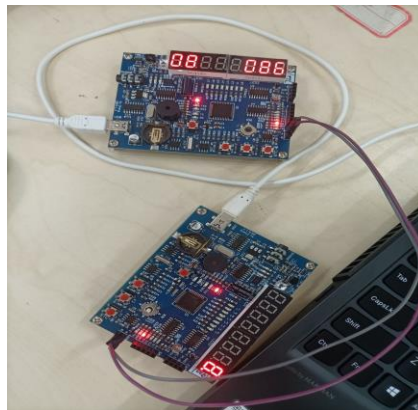
在主机连接的计算机上运行程序，输入从机地址，尝试获取分数，获取到了正确的分数。

```
y@ubuntu:~/STC1/1516$ sudo ./a.out
-----基于RS485的总线评分系统-----
请输入：
从机地址:查询分数
-2:将从机复位
-1:退出程序
8
设备检测正常
分数: 86
-----基于RS485的总线评分系统-----
请输入：
从机地址:查询分数
-2:将从机复位
-1:退出程序
```

输入复位命令-2，将从机复位：

```
y@ubuntu: ~/STC1/1536$ sudo ./a.out
-----基于RS485的总线评分系统-----
请输入:
从机地址: 查询分数
-2: 将从机复位
-1: 退出程序
8
设备检测正常
分数: 86
-----基于RS485的总线评分系统-----
请输入:
从机地址: 查询分数
-2: 将从机复位
-1: 退出程序
-2
从机已复位
-----基于RS485的总线评分系统-----
请输入:
从机地址: 查询分数
-2: 将从机复位
-1: 退出程序
```

从机复位成功，第 1 位和第 8 位 LED 灯熄灭。



## 结果分析

通过测试，程序可以完成评分功能。但发送数据后，从机不一定能够接收数据并发出回应，因此需要多次尝试检测设备，获取分数才能获取数据。推测这与单片机程序有关，单片机程序不能及时接收到主机发送的数据，也只发送一次数据给主机，导致通信存在困难。程序的设计应该是正确的。

## 三.实验总结

通过本次实验，复习并加深了对 RS485 总线通信的理解，也进一步熟悉了使用 c++ 进行串口的数据读写，以及时间监听等机制。成功实现了评分系统。在实现的过程中与同学配合，将串口及事件监听作为类封装参考了同学的设计，数据的收发及解析主要结合实验资料中的定义完成。最后的多级评分尝试在原程序的基础上进行实现，但是由于单片机不能每次都准确的接收数据并进行响应，没有测试成功。通过串口相关的几次实验，我对单片机与计算机的串口通信以及基于 RS485 的通信都有了深入的了解。为后续相关知识的学习建立了基础。