# 实验八 串口打印Manchester解码实验指导书

## 一、实验目的

1.掌握STC8系列单片机Manchester解码的基本使用方法。

2.学习如何通过串口打印解码后的卡号信息。

3.理解并实践如何通过串口调试助手观察解码结果。

## 二、实验平台

1.开发板：SXRFIDB低频ID识别与应用开发板。

2.芯片型号：STC8G1K08/17（微调后可移植至STC8A/F/C/G/H系列单片机）。

3.时钟配置：芯片内部12MHz。

4.串口速率：9600bps。

5.供电方式：通过USB线插入到Uart USB接口给开发板供电。

## 三、实验前准备

1.硬件准备：SXRFIDB开发板、USB线、电脑、STC-ISP编程软件

2.软件准备：安装STC-ISP软件、配置好开发环境（如KeilC51或其他支持STC8系列的编译环境）

## 四、实验步骤

1.硬件连接

1. 使用USB线将开发板的Uart USB接口连接到电脑的USB端口，确保开发板正常供电。
2. 检查开发板上的曼彻斯特码输入端（P3.2）是否连接正常。

2.软件配置

1. 打开STC ISP软件，配置芯片时钟为内部12MHz。
2. 将配置好的时钟参数下载到单片机中，确保时钟配置生效。

3.程序编写与下载

1. 打开实验提供的源代码文件。
2. 将代码编译生成HEX文件。
3. 使用STC-ISP软件将HEX文件下载到开发板上的单片机中。

4.实验操作

1. 上电开发板，确保USB接口连接正常。
2. 打开电脑上的串口调试助手，选择对应串口号（具体串口号可在电脑设备管理器中查询）。
3. 配置波特率为9600bps，选择文本/字符模式接收数据。
4. 在开发板上方5-15cm处放置卡片，观察串口调试助手中是否显示解码后的卡号信息。

## 五、实验原理

1、Manchester编码：Manchester编码是一种自同步编码方法，通过将每个数据位编码为高低电平的跳变来实现数据传输。每个数据位的中间有一个跳变，用于同步。

2、解码过程：通过检测曼彻斯特码的跳变，恢复出原始数据。解码过程中需要处理帧头识别、数据位提取、校验等步骤。

3、串口通信：通过串口将解码后的卡号信息发送到电脑，使用串口调试助手观察结果。

## 六、实验代码解析

1.头文件

1. 包含了主控芯片STC8G的头文件，提供了芯片相关的寄存器定义和功能函数。
2. 包含了标准输入输出库，用于使用`printf`函数。
3. 包含了内联汇编函数库，用于使用`\_nop\_`函数。



2.数据类型定义

定义了常用的数据类型，如`u8`、`u16`、`u32`等。



3.端口/引脚定义

定义了曼彻斯特码输入端为P3.2引脚。



4.用户自定义数据

定义了系统时钟频率、串口波特率计算值和超时阈值。



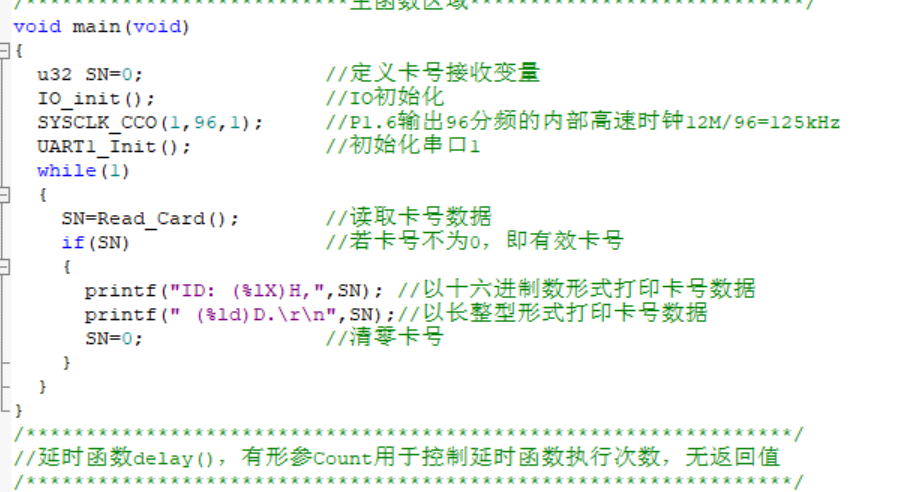
5.函数声明

声明了程序中用到的函数。

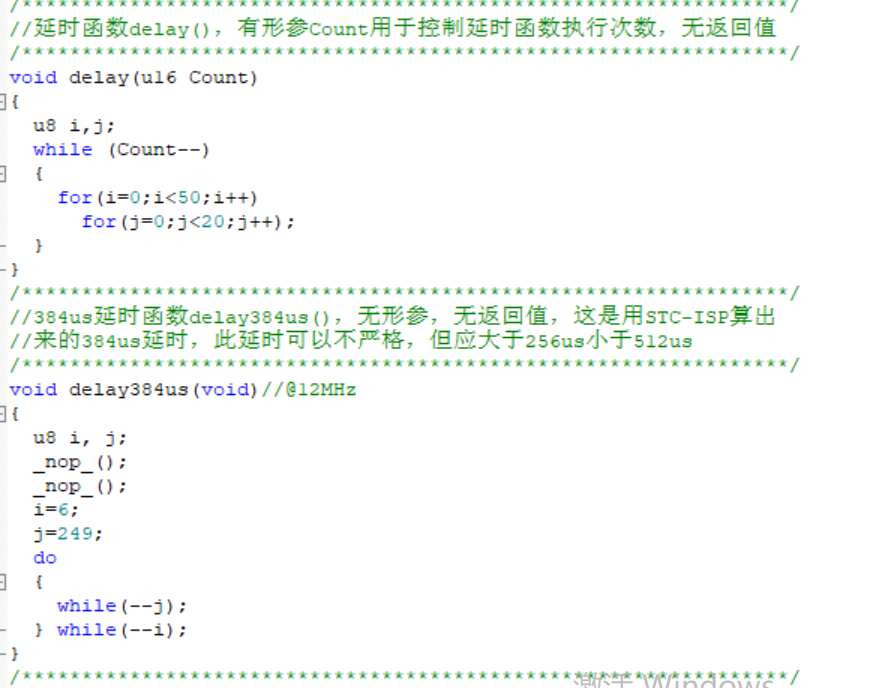


6.主函数

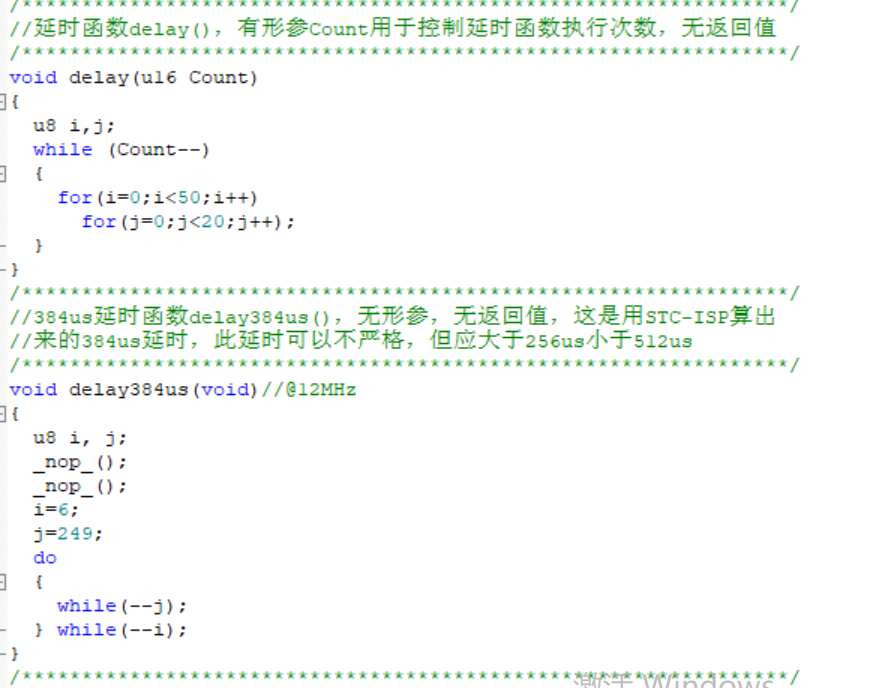
1. 初始化IO端口和串口。
2. 进入无限循环，读取卡号并打印。



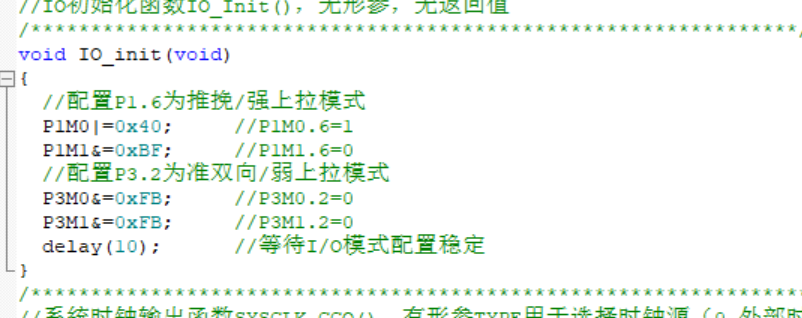
7.延时函数：用于产生延时，确保硬件配置的稳定性。



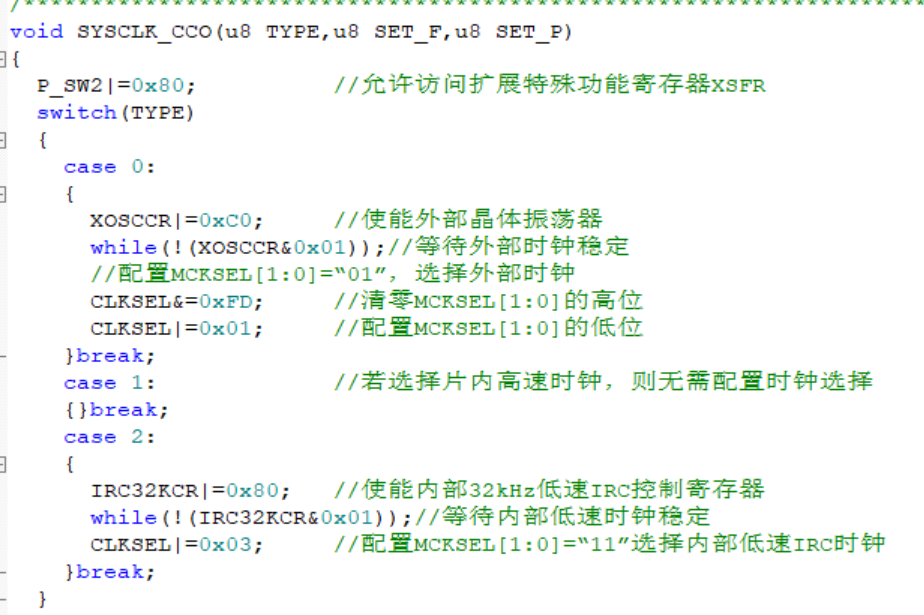
8.384us延时函数：用于产生384us的延时，确保曼彻斯特码的正确解码。

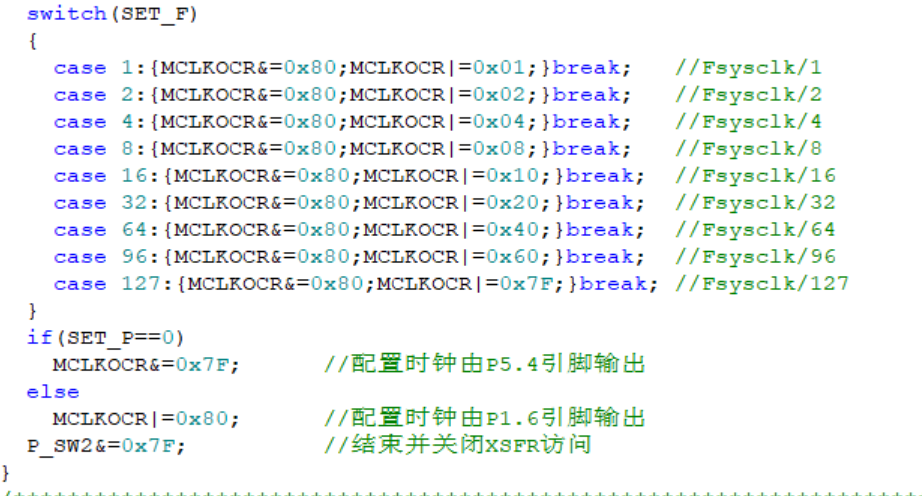


9.IO初始化函数：配置IO引脚模式，确保信号输入输出正常。

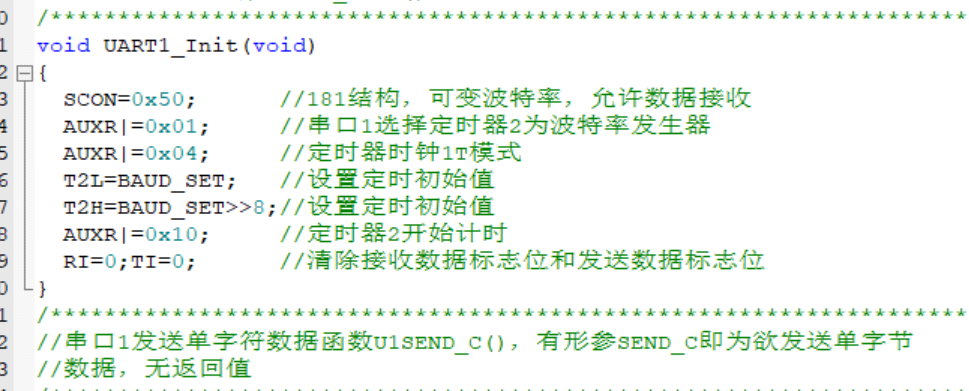


10.系统时钟输出函数：配置系统时钟输出，选择时钟源、分频系数和输出引脚。

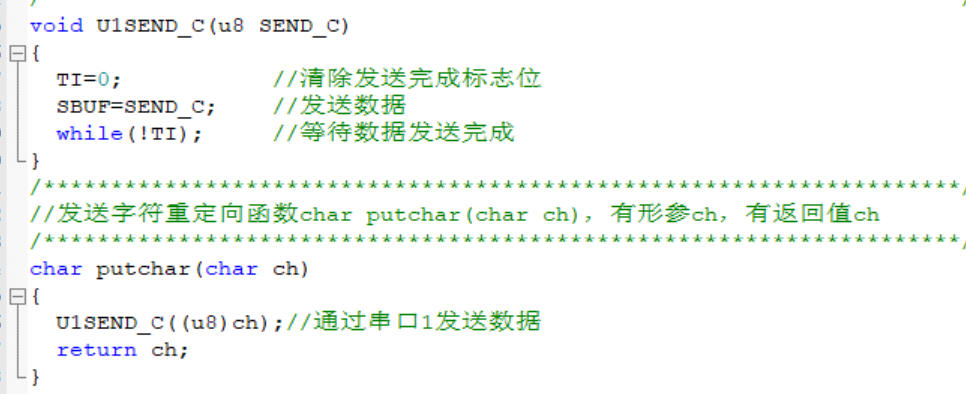




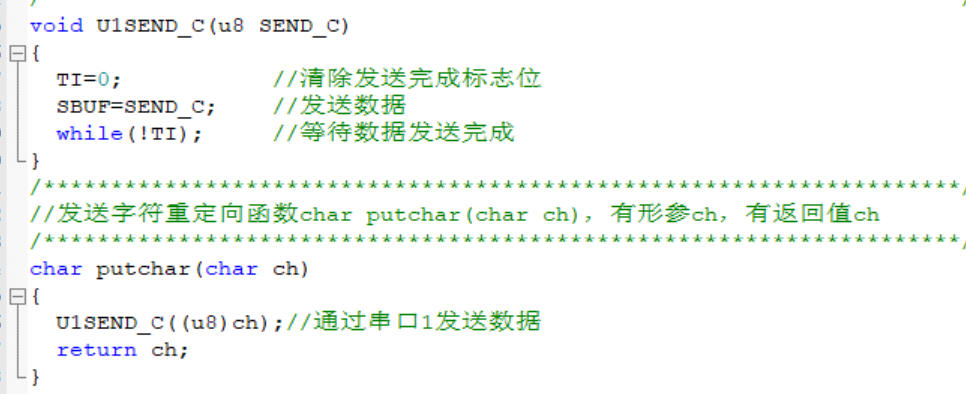
11.串口1初始化函数：配置串口通信参数，包波特率、定时器等。



12.串口1发送单字符数据函数：实现串口发送单字节数据的功能。

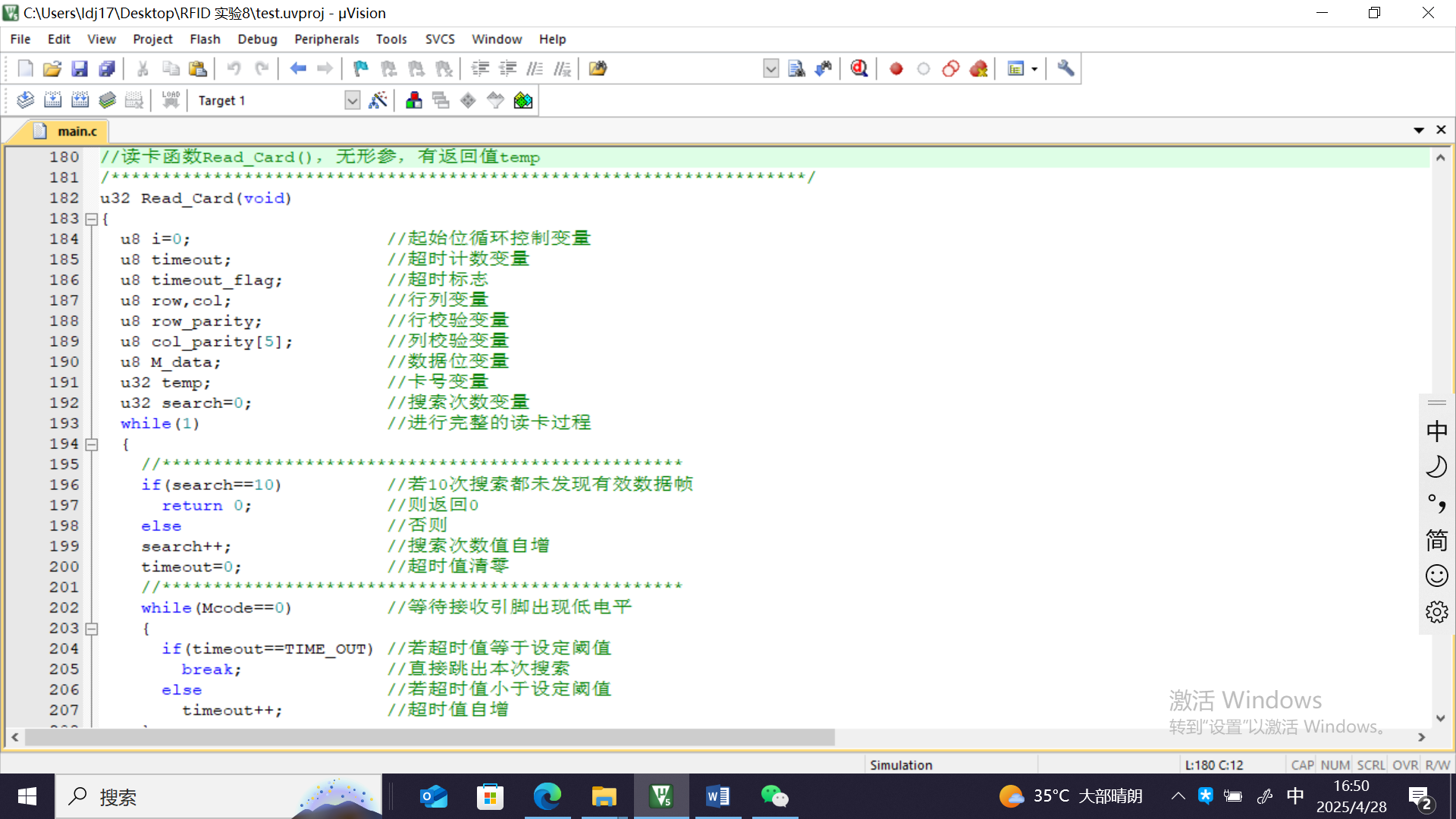


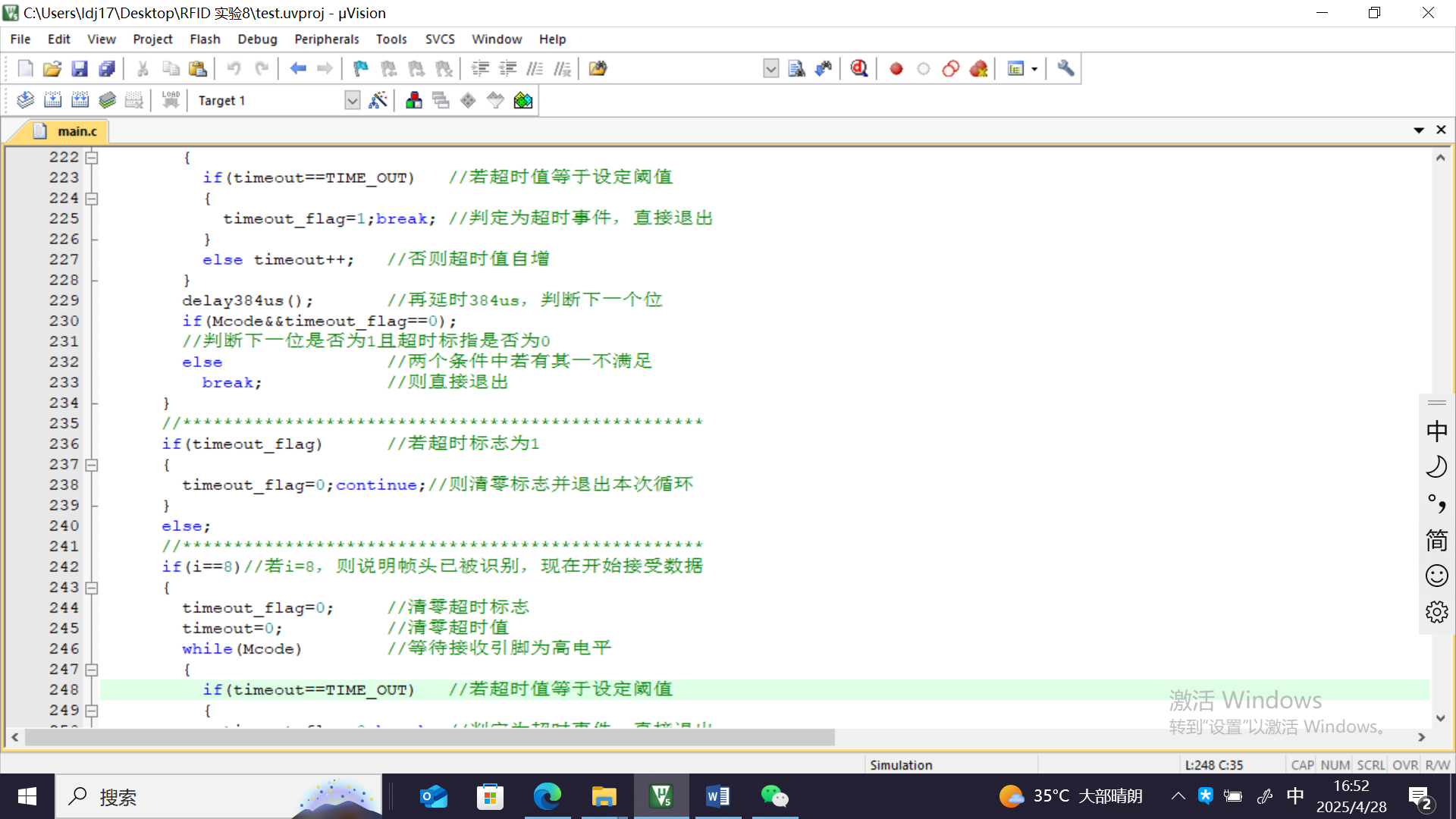
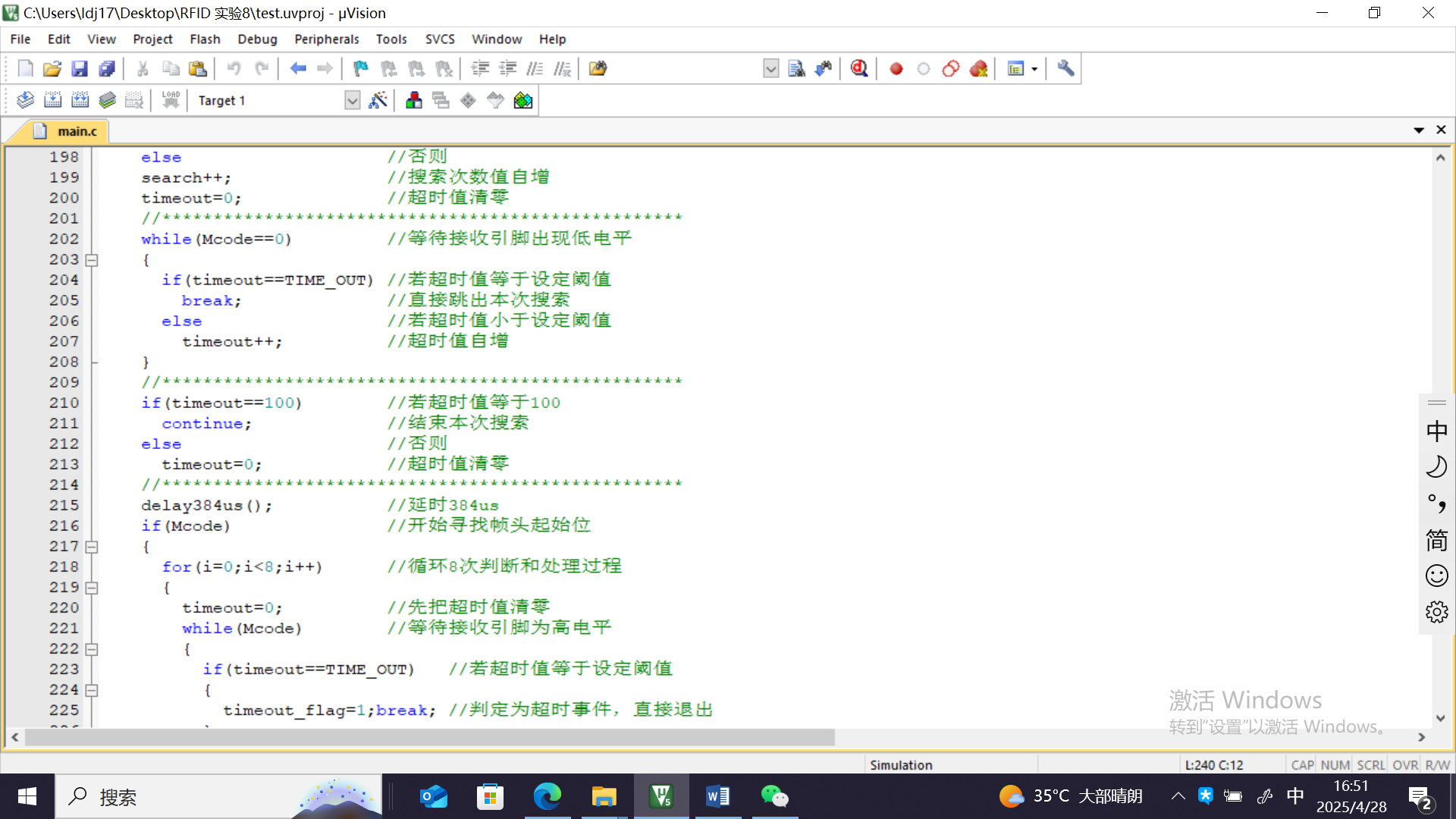
13.发送字符重定向函数：重定向`printf`函数的输出到串口。

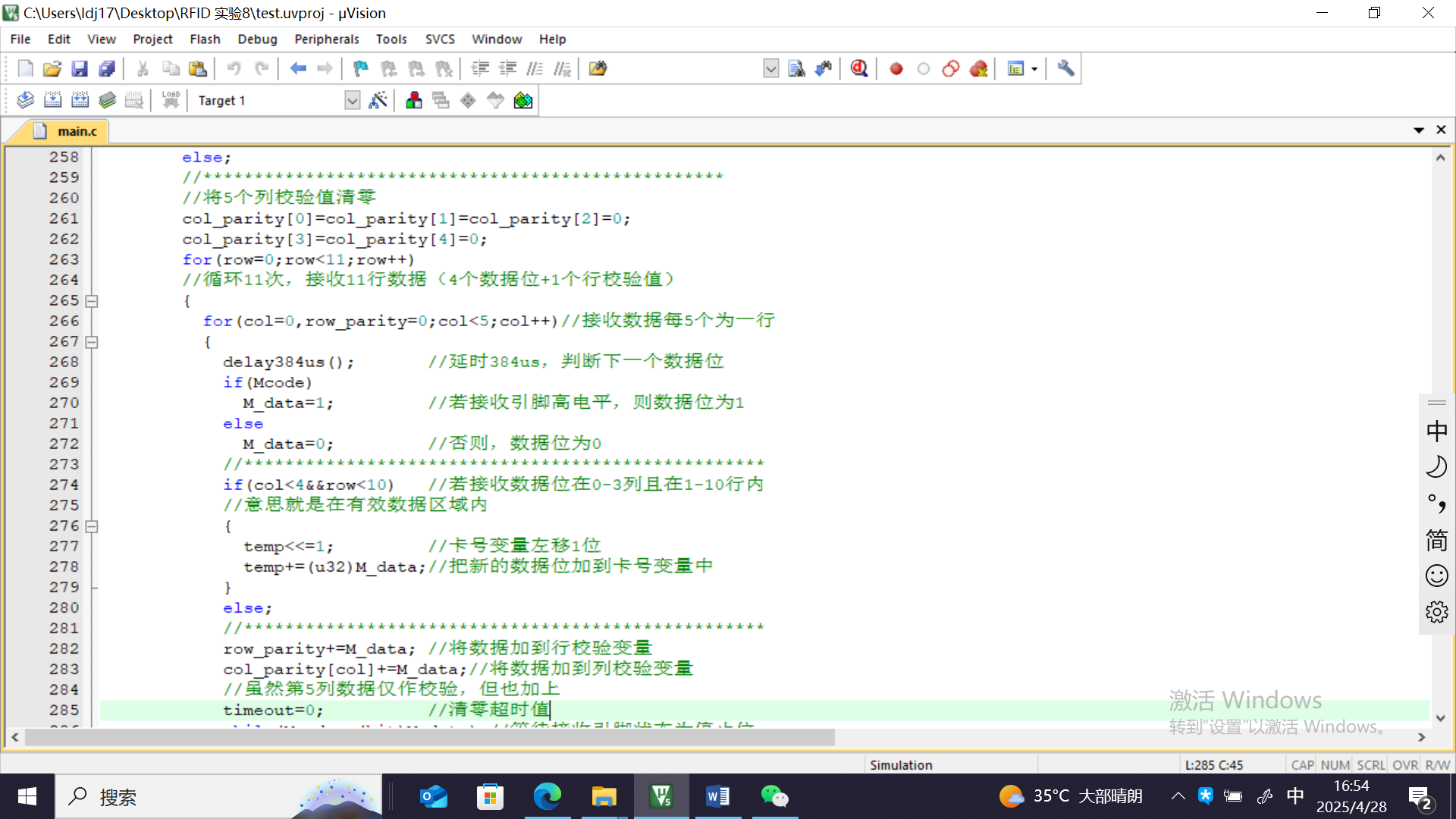
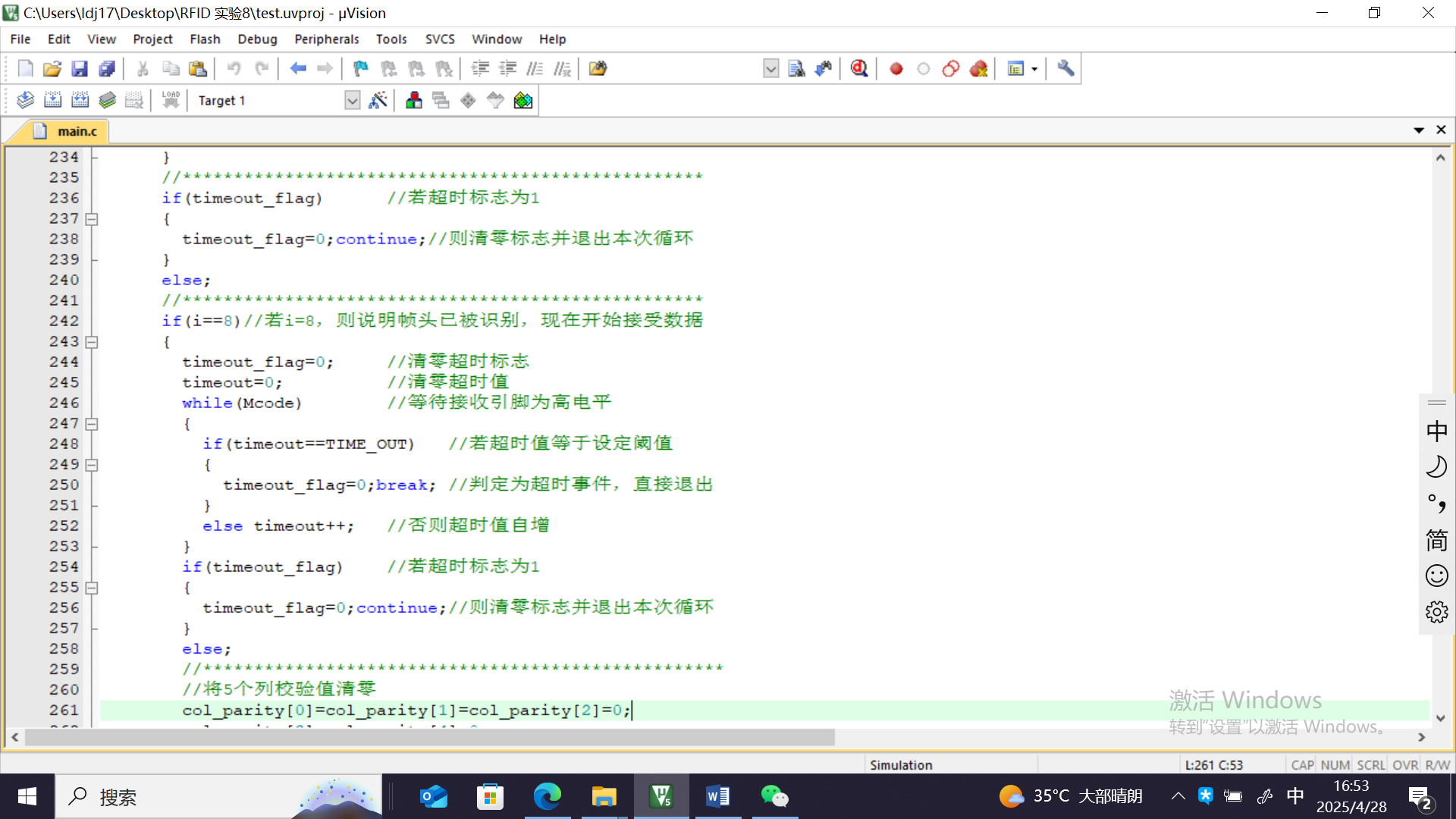


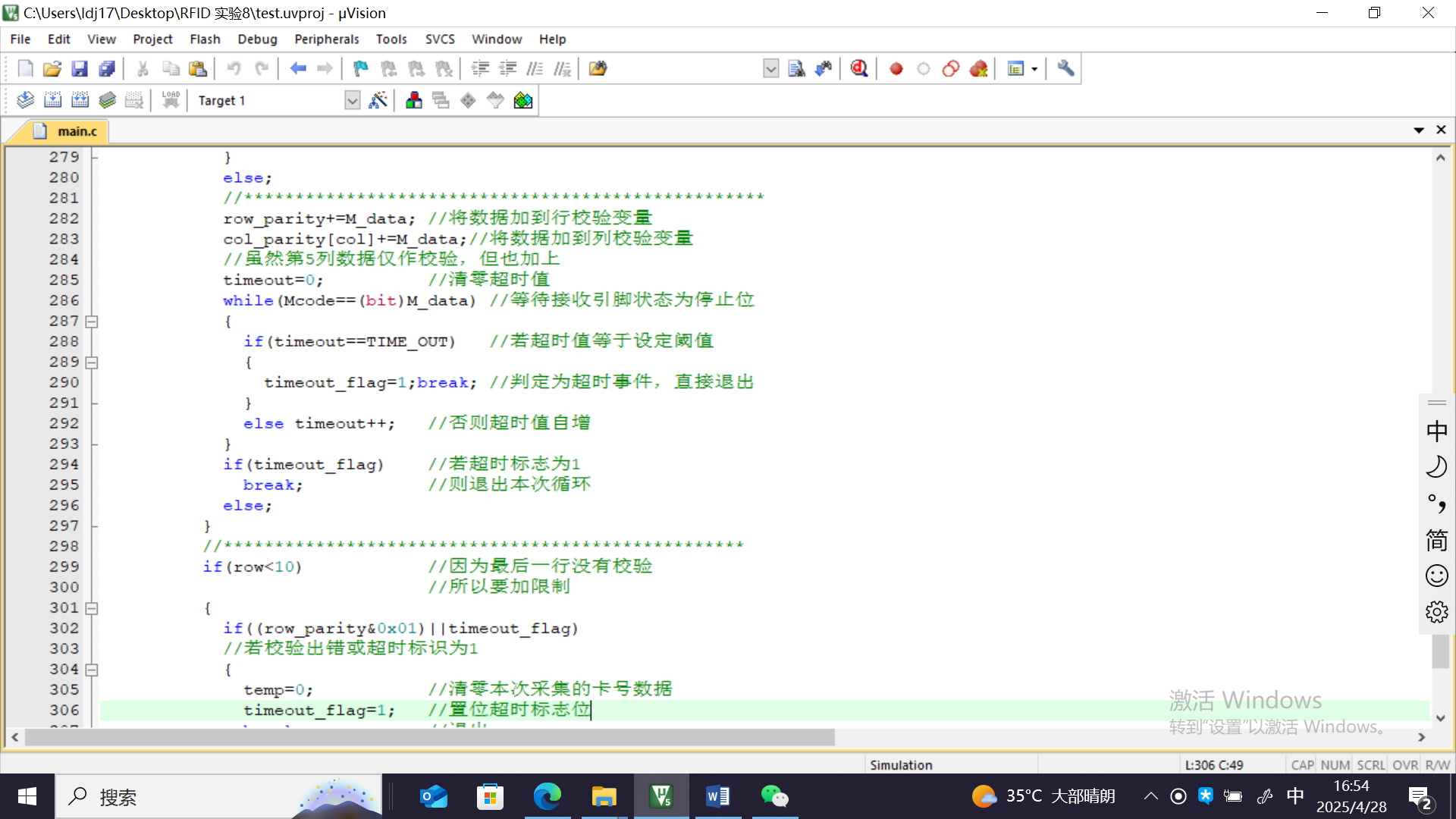
14.读卡函数：实现曼彻斯特码的解码，包括帧头识别、数据位提取、校验等步骤。

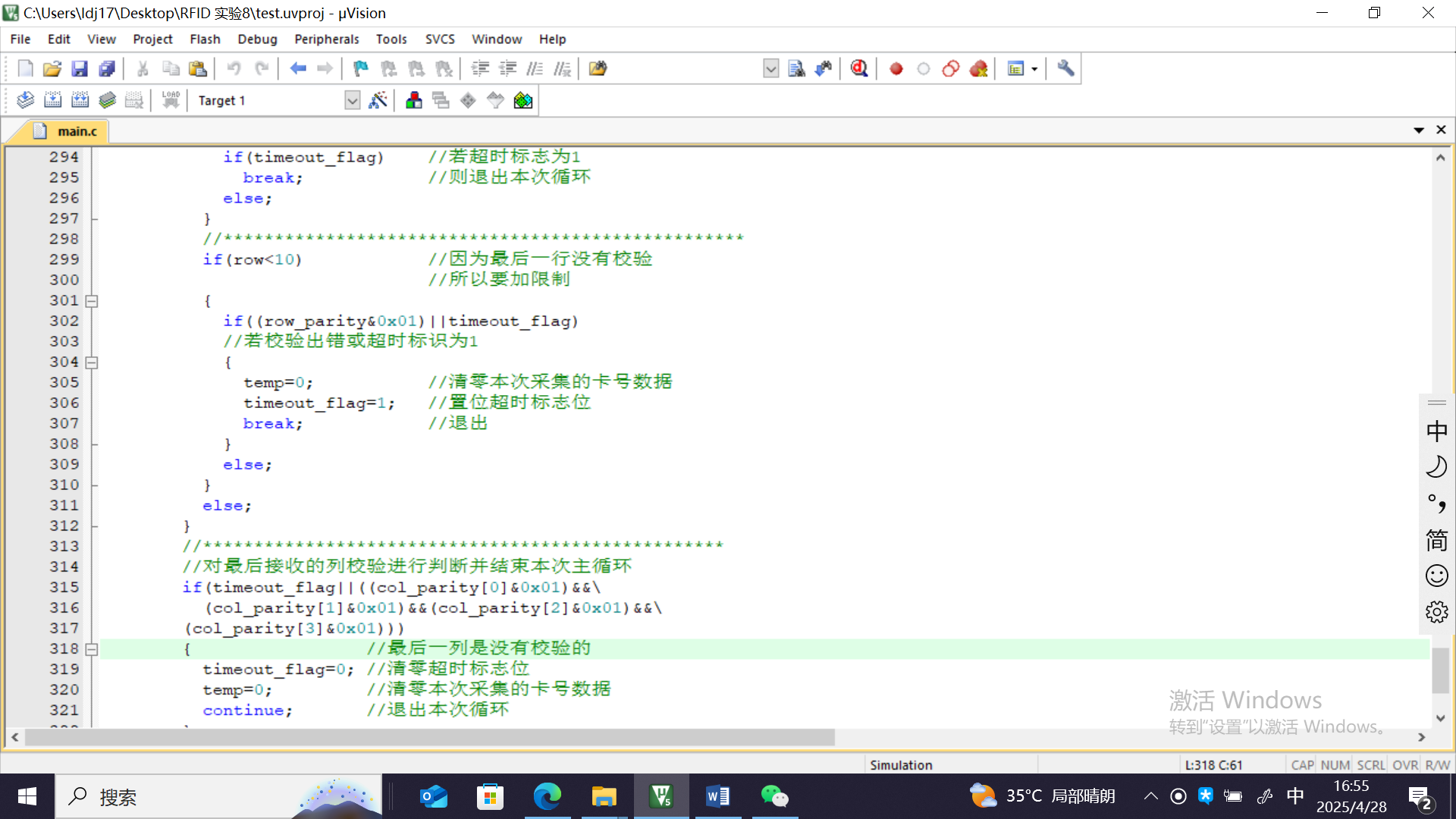












## 七、实验注意事项

1、在进行硬件连接时，务必确保连接正确，避免短路或误操作导致开发板损坏。

2、在下载程序时，确保STC-ISP软件配置正确，时钟参数与芯片型号匹配。

3、在实验过程中，注意观察串口调试助手的输出，确保解码结果正确。

## 八、实验总结

通过本次实验，掌握了STC8系列单片机Manchester解码的基本使用方法，学会了如何通过串口打印解码后的卡号信息，并通过串口调试助手验证了解码结果。实验过程中，成功实现了卡号信息的解码和打印，为后续学习更复杂的单片机应用奠定了基础。