**温度监测系统**

**设**

**计**

**报**

**告**

目录

[1.系统功能与指标 2](#_Toc92658497)

[1.1 基础系统功能 2](#_Toc92658498)

[1.2 扩展功能 2](#_Toc92658499)

[1.3 指标 3](#_Toc92658500)

[2. 系统方案 3](#_Toc92658501)

[2.1 系统基本设计思路 3](#_Toc92658502)

[2.2 基于RXTX实现串口通信 4](#_Toc92658503)

[3. 硬件原理与框图 4](#_Toc92658504)

[3.1 初始化蜂鸣器 4](#_Toc92658505)

[3.2 初始化LED灯 5](#_Toc92658506)

[3.3 初始化按键 5](#_Toc92658507)

[4. 软件设计 6](#_Toc92658508)

[4.1 汇编程序设计 6](#_Toc92658509)

[4.2 C程序设计 7](#_Toc92658510)

[4.3 AD转换 7](#_Toc92658511)

[4.4 IRQ处理器 8](#_Toc92658512)

[4.5 基于Java RXTX实现串口通信并进行数据分析 10](#_Toc92658513)

[5. 系统测试 14](#_Toc92658514)

[5.1 HyperTerminal 测试 14](#_Toc92658515)

[5.2 Java串口通信测试 15](#_Toc92658516)

[6. 结论与特色 17](#_Toc92658517)

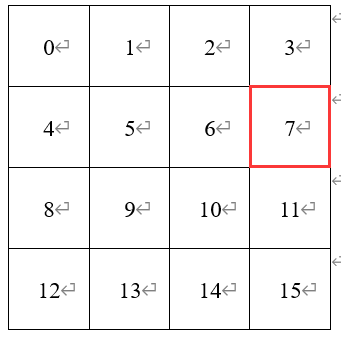
[7. 参考 18](#_Toc92658518)

## 1.系统功能与指标

### 1.1 基础系统功能

监控A8实验机箱上的温控传感器，连续按动5次指定按钮进行检测，每次按动按键，PC机会显示一次温度经AD转换后的值。在第五次按动按钮时，若温度（AD < 2400）则出现蜂鸣器“滴滴”报警声和指示灯闪烁。

注：实验者学号尾数为7，对应键位图如图1，指示灯对应7号灯。实验箱的热敏电阻遇到高温时，AD值变低，实验环境下无法使AD > 3000，因此定义的报警阈值为AD < 2400触发报警。



**图1**

### 1.2 扩展功能

① 增加第五次按动按键报警后的手动解除报警功能。

② 使用基于Java RXTX框架实现串口通信，实现警报弹窗提示功能，将每次按动按键转换的AD值进行数学统计，如计算均值，AD最小值（温度最大值）等，将统计信息保存在本地磁盘，数据保存在Excel中便于对数值进行分析及可视化。

### 1.3 指标

A8实验箱热敏电阻温度值转换好的AD值低于2400，则表示实验环境温度过高，出现一系列报警提示。同时需要满足按键在第五次及以上才实现报警功能。

注：由于是温度监控，所以为了符合高温监控的真实场景，应该在第五次及以上按动按键进行报警，而不仅限于第五次。

## 2. 系统方案

### 2.1 系统基本设计思路

首先在完成一系列硬件的初始化，如：蜂鸣器和LED灯等的硬件初始化之后，轮询等待按键的按动触发事件，在按动指定按键之后触发IRQ中断，然后进行软件延时消抖检测按钮是否真实被按下，维护按动次数的全局变量counter累加一次，紧接着进行AD转换。AD转换过程，需要取出AD转换后的结果，并将结果通过uart.c实现的printf函数将必要的结果值传输到串口中，便于进行A8实验机箱与PC机的串口通信。

对返回的AD转换值与报警阈值2400进行判断，同时也要判断按动次数counter是否大于等于5次，同时满足上述两个条件才触发后续的报警功能。

报警功能实现，首先需要将报警触发的信息送到串口通信，通知PC机报警已触发，接着轮询将蜂鸣器和LED灯的引脚控制值进行修改，在蜂鸣器和灯工作与停止工作中间穿插一小段时间的延时，模拟真实警报的间断性，而不是长鸣和长闪。报警功能的每个周期，添加一段取消功能的逻辑。实验选用的也是7号键位的消抖判断，若触发报警之后，再次按动按动按键，则实现手动关闭蜂鸣器和LED灯，并将日志信息通信PC机。

### 2.2 基于RXTX实现串口通信

由于HyperTerminal终端工具在进入Uboot界面之后，会阻塞式地监听来自A8实验机箱的串口通信信息，只能做简单的展示日志信息，所以需要使用非阻塞式的实现，对串口通信信息进行额外处理。

非阻塞式的实现方式有很多。由于实验者对Java比较熟悉，同时考虑到Java可使用多线程编程实现非阻塞式监听和JDK自带Swing框架实现弹窗警报通知，本实验使用Java实现串口通信的定制化设计。

技术选型为JDK Swing，RXTX[[1]](#RXTX)串口通信框架，Java解析Excel框架EasyExcel[[3]](#EasyExcel)，Hutool IO工具类[[2]](#Hutool)。

Swing的JoptionPane组件可以实现简单的警报弹窗功能，而核心的串口通信功能使用RXTX串口通信框架实现，其原理是建立串口信息绑定后，多线程在后台开启一个监听器，若监测到有串口信息，就将数据流读入并转换为字节数组，后续可对传输的数据，如AD值进行操作。

由于串口通信的数据字符串不止有温度值，还有一些必要的描述日志信息，所以需要在Java端需要截取字符串获取AD值，为了更方便使用此功能，所以在C端使用uart.c实现的printf数据，将AD值使用特殊字符标识出来，然后Java端用正则表达式截取字符串获取AD值。

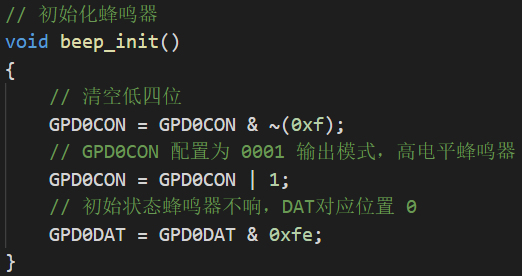
后续需要将求5次及以后的串口通信数据，所以维护一个成员变量List存储温度信息，在保存温度信息的同时，也能获取到AD值的数量，即按键按动的次数，便于进行逻辑判断，是否在第5次及以后进行弹窗报警。

使用Java端对串口数据进行监听还有许多好处，在List保存历史AD值信息的实现下，可以计算温度值的算数平均值，报警前距离报警阈值的温度差值和ADC最小值，即温度最大值，这些数值都具有十分重要的统计意义，因此最好保存在数值统计工具Excel中，便于后续的管理和分析数据。技术使用Alibaba开源的EasyExcel[[3]](#EasyExcel)。

## 3. 硬件原理与框图

### 3.1 初始化蜂鸣器

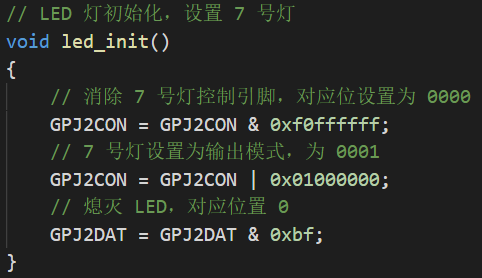
控制蜂鸣器的CON单元GPD0CON的低四位设置为0001，输出模式，高电平蜂鸣器，初始状态关闭蜂鸣器。如图3.1



**图3.1**

### 3.2 初始化LED灯

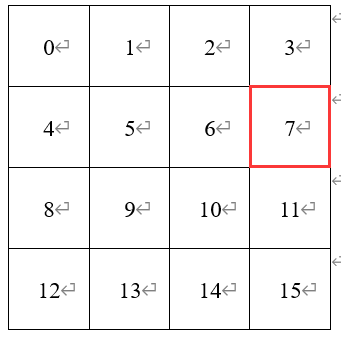
实验者学号尾号为7，按照实验要求初始化7号灯。消除7号灯控制引脚，改为输出模式0001，初始化熄灭状态，置为0。如图3.2所示。



**图3.2**

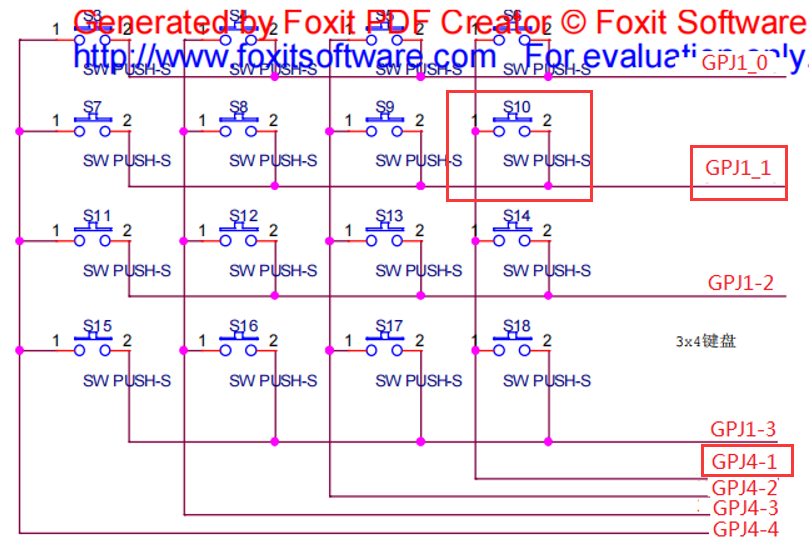
### 3.3 初始化按键

实验者学号尾号为7，按照实验要求初始化7号灯。位置如图3.3.1所示。



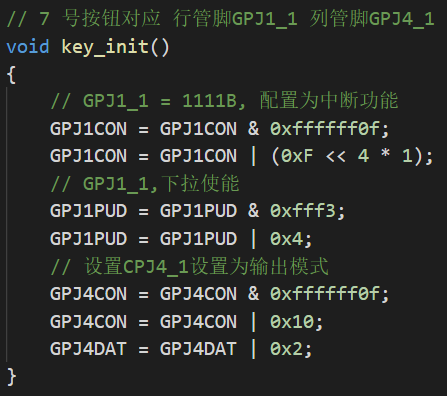
**图3.3.1**

根据按键接口图，可得到7号灯对应行列引脚为GPJ1-1和GPJ4-1，如图3.3.2所示。



**图3.3.2**

给对应的键位行引脚设置中断功能和下拉使能，列引脚设置为输出模式。如图3.3.3所示。

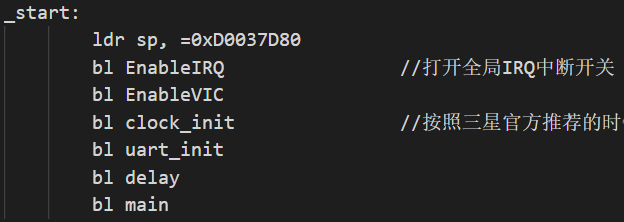


**图3.3.3**

## 4. 软件设计

### 4.1 汇编程序设计

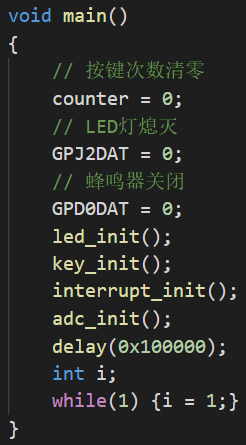
汇编程序入口，开启全局IRQ中断开关，开启VIC，按照芯片手册时钟进行初始化，初始化uart，然后delay延时，最后跳转main进入C程序，逻辑见图4.1。详细代码见附件start.S。



**图4.1**

### 4.2 C程序设计

进入C程序主函数，首先将全局变量counter置零，后续记录按键次数。同时也要将LED灯熄灭和蜂鸣器关闭。然后初始化LED，按键Key，中断和ADC，在短暂延时后进入轮询阻塞，逻辑实现见图4.2。初始化源码见附件adc.c源文件。

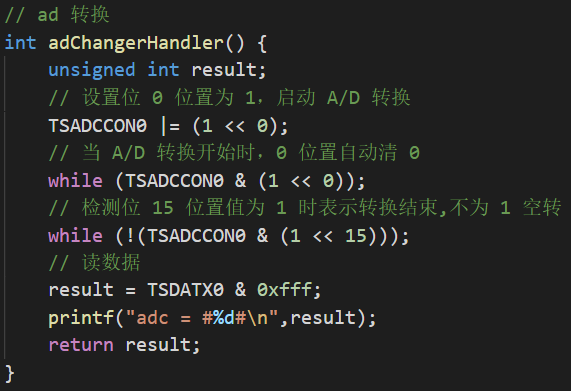


**图4.2**

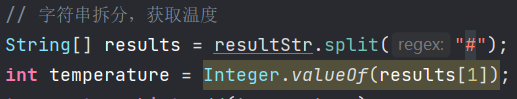
### 4.3 AD转换

AD转换控制引脚对应TSADCCON0寄存器，先将0位置置为1，启动AD转换。根据AD转换原理，当AD转换启动时，0位置会自动清零，所以轮询空转，直到0位置为0，表示AD转换已启动。后续操作需要检测AD转化是否转化完成，对应第15位置是否为1，不为1则空转。转化结果会保存再TSDATX0里面，取出后12位即为AD转换结果。然后将AD转换结果printf串口通信发送到PC机。逻辑见图4.3.1。

由于后续需要使用到Java对串口通信数据进行字符串的截取，为了方便截取AD值，所以手动规定特殊字符将AD值标识，本实验使用‘#’作为分隔符，在Java程序使用正则表达式截取‘#’数据即可。对应Java程序逻辑如图4.3.2。



**图4.3.1**

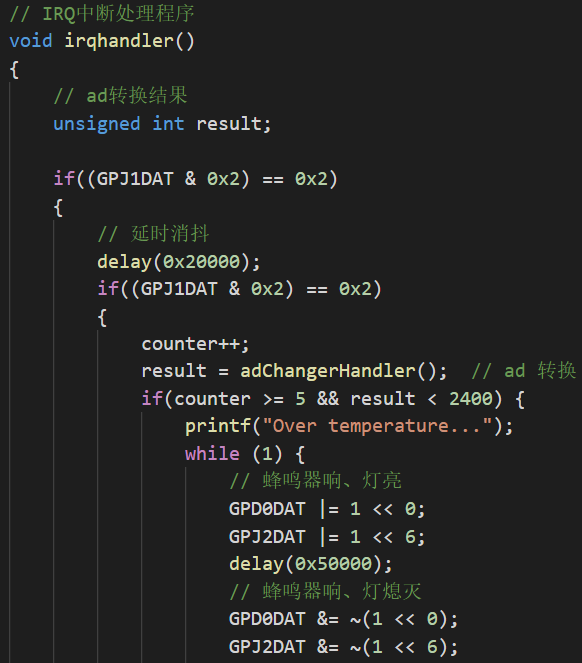


**图4.3.2**

### 4.4 IRQ处理器

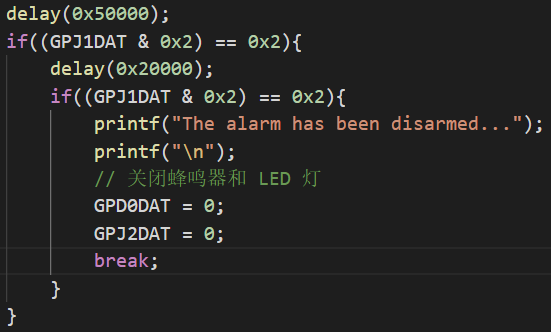
在前序硬件软件的初始化之后，程序在轮询空转，直到按动按键触发IRQ中断信号，接下来跳转到irqhandler执行IRQ中断处理程序。

首先使用软件delay延时实现消抖，检测是否按键真实按下，若为真，则按键次数计数器counter自增一，然后执行AD转换逻辑，如[**4.3**](#_4.3_AD转换)所述。接下来判断本次AD转换值是否小于阈值2400，代表温度超标，同时判断按键次数是否大于等于5次，然后printf串口通信发送温度超标日志给PC机，循环执行蜂鸣器和LED灯的工作和停止工作，在中间穿插一小段时间延时，模拟真实的间断报警情况。如图4.4.1所示。



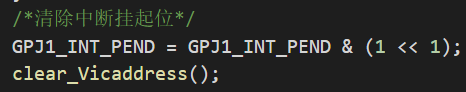
**图4.4.1**

在循环中除了执行蜂鸣器和LED灯的报警程序，穿插了额外的手动关闭报警扩展。其逻辑也是检测按键是否被按下（消抖处理），然后将蜂鸣器和LED灯对应DAT位置零实现手动关闭警报功能。如图4.4.2所示。



**图4.4.1**

最后需要清除中断挂起位，清空VIC，恢复之前状态。如图4.4.3所示。



**图4.4.3**

### 4.5 基于Java RXTX实现串口通信并进行数据分析

RXTX[[1]](#RXTX)是Java客户端的串口通信框架，可以借助它实现HyperTerminal一样的监听功能，同时也能由于Java端开启额外线程在后台进行监听串口数据，所以可以进行非阻塞式的后续操作。

由于本实验时对热敏电阻的AD值进行监控，所以关键数据是温度转化后的AD值，还有相关的数据，如温度报警阈值、均值和差值等，具有重要的统计意义，便于在温度超标事故发生之后，通过日志信息直观看到事故前的温度动态变化。基于上述考虑，在软件设计上，增加数据格式化Excel保存在本地磁盘的设计，技术选型为阿里巴巴开源的Java解析Excel框架EasyExcel[[3]](#EasyExcel)。

**项目结构**：

|  |
| --- |
| \---src  \---main  \---java  \---me  \---hypocrite30  \---serialport  +---easyexcel - EasyExcel 源码  \---serial - RXTX 串口通信源码 |

下面对Java端设计进行源码分析。

#### 4.5.1 环境配置

从RXTX[[1]](#RXTX)官网上下载操作系统适配的运行环境，将两个dll文件移动到Java运行环境JRE的bin目录下，完成串口通信框架的环境搭建。如图4.5.1.1所示。



**图4.5.1.1**

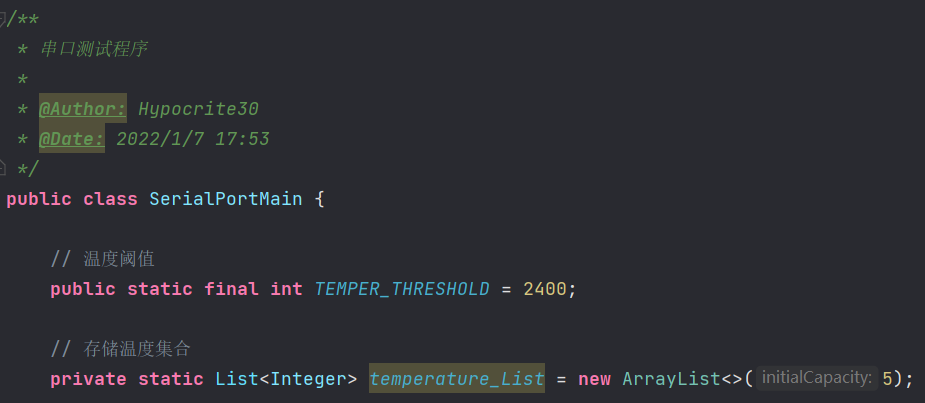
创建Java Maven工程，导入依赖坐标。如图4.5.1.1所示。



**图4.5.1.2**

#### 4.5.2 程序入口及成员变量

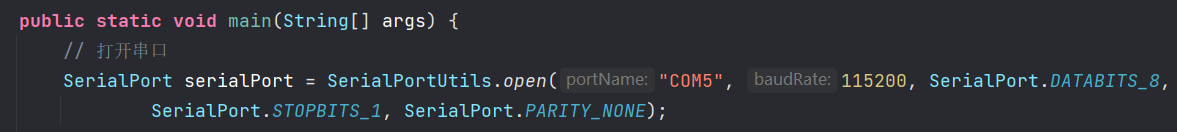
SerialPortMain为程序主类，成员变量有一个常量温度阈值2400和储存串口通信传过来的温度值。见图4.5.2。



**图4.5.2**

#### 4.5.3 RXTX串口通信

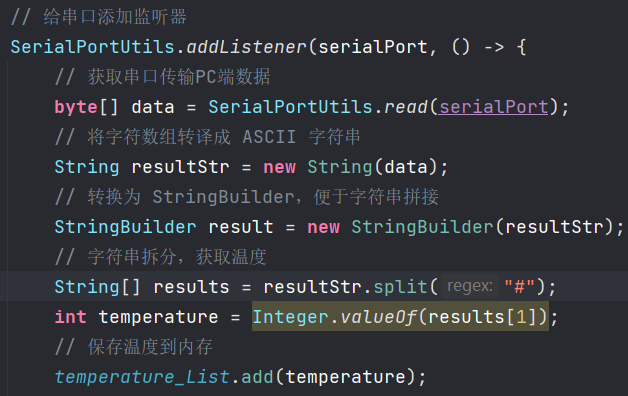
首先也是与HyperTerminal一样，需要打开串口，进行必要设置。信息为：串口名COM5，波特率115200，数据位8，停止位1，无检验位。见下图4.5.3.1。SerialPortUtils为封装的串口通信类，详细见附件。



**图4.5.3.1**

接着需要给串口添加监听器，与HyperTerminal不同的是，Java端添加监听器是单独开一个线程，在后台异步监听串口数据，同时监听器也可以自己编写逻辑，如计算均值，温差等操作，非常灵活。

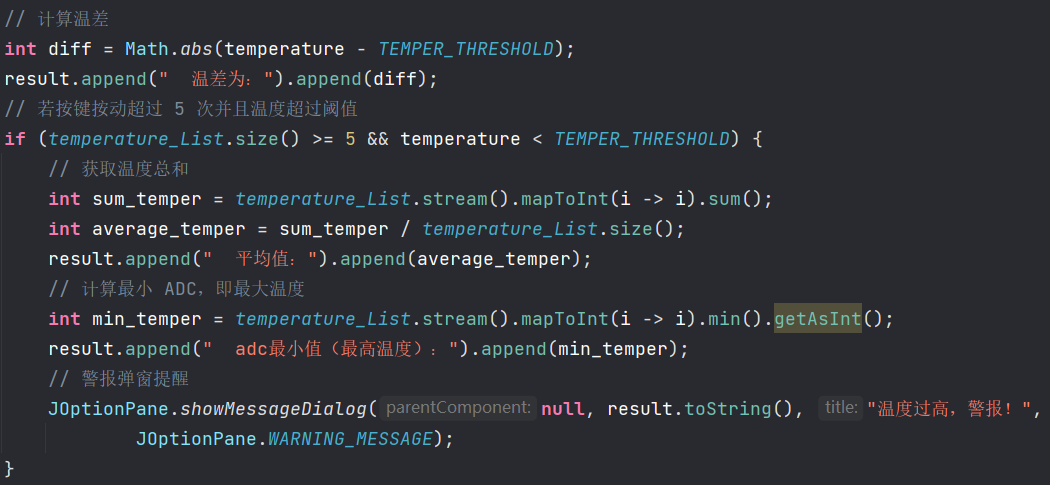
当串口通信有数据传送PC机时，监听器监听到数据，异步read读取串口中的数据，封装成字节数组data返回。由于字节数据是ASCII数据，不便于阅读检验，所以转译为String字符串，而后续需要对结果字符串resultStr进行拼接额外数据统计信息，所以再转化为StringBuilder对象得到result。在前面的C程序[AD转换](#_4.3_AD转换)程序设计提到，为了便于在字符串中获取到准确的温度数据而不是其他字符，所以添加了额外的分割字符‘#’，然后使用正则表达式split拆分字符串，去1下标的数值并转化成Integer整数类型，最终获取串口数据中的AD值temperature，紧接着保存到本地List中，后续数学统计需要用到。逻辑见图4.5.3.2。



**图4.5.3.2**

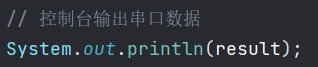
在获取温度值后，即可开始进行数学统计。计算温差算法是当前温度与报警阈值的差值取绝对值，然后字符串拼接，将计算结果信息拼接到result。判断报警的逻辑是按动按键（储存温度List的size）大于等于5次，且温度高于阈值（AD小于阈值），则触发报警。

报警逻辑：先计算前面几次的温度总和，求平均，字符串拼接，计算ADC最小值（温度最大值），字符串拼接，最后使用JDK Swing的JoptionPane组件进行弹窗报警。见图4.5.3.3。



**图4.5.3.3**

每次监听都会将结果值打印到控制台中展示信息，如图图4.5.3.3。

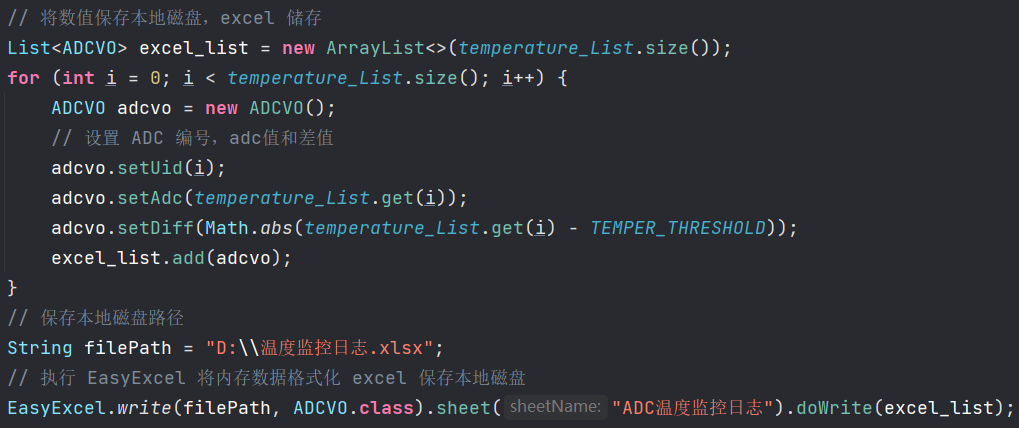


**图4.5.3.4**

#### 4.5.4 EasyExcel解析Excel储存并统计

EasyExcel[[3]](#EasyExcel)是阿里巴巴优秀的开源框架，可将内存中的数据序列化以Excel形式储存到本地磁盘中，同时制作可视化表格便于分析。

ADCVO对象是Excel中的表头属性，有ADC编号，温度差值和ADC值。取出前面存储的温度列表，分别封装成VO对象，然后指定本地磁盘路径，最后执行write方法写入磁盘，可以设置Excel的sheet表单名称。逻辑见图4.5.4。



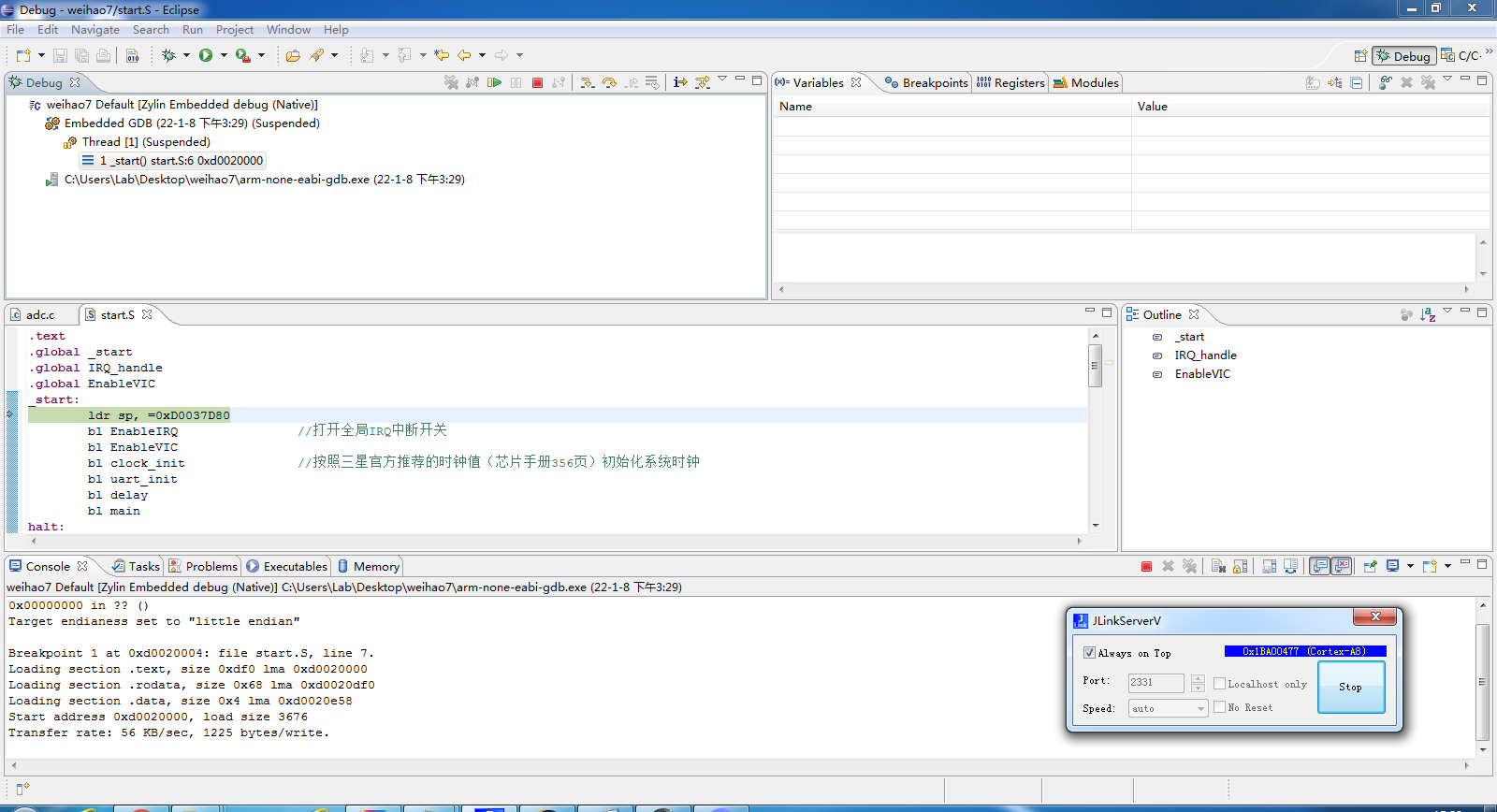
**图4.5.4**

## 5. 系统测试

### 5.1 HyperTerminal 测试

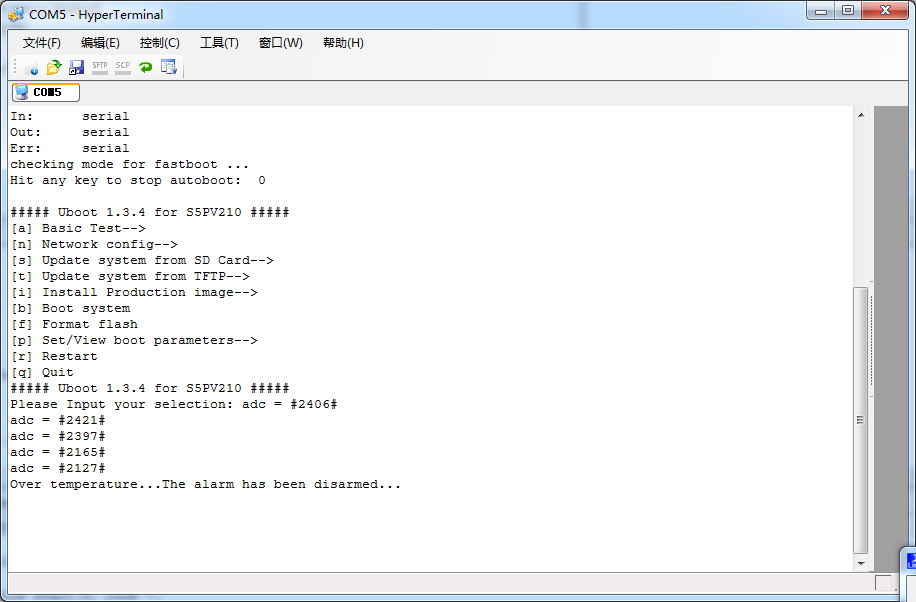
在 HyperTerminal 中，进行如下配置：连接COM5串口，设置波特率为115200，数据位为8，停止位为1，无校验位。

打开JlinkServerV，然后在Eclipse IDE中编译汇编和C程序，之后进入Debug调试启动，进入如下图5.1.1界面。



**图5.1.1**

每按动一次按键后，A8实验机箱串口通信发送PC端ADC数据，HyperTerminal阻塞监听到发送的数据，在第五次按下按键，且温度过高造成AD值小于阈值2400之后，发出警报，实验机箱出现间断鸣笛和闪光，同时PC端接收到「Over temperature」温度超标日志信息，再一次按动按键，警报解除，收到「The alarm has been disarmed」警报解除日志信息。效果如图5.1.2所示。

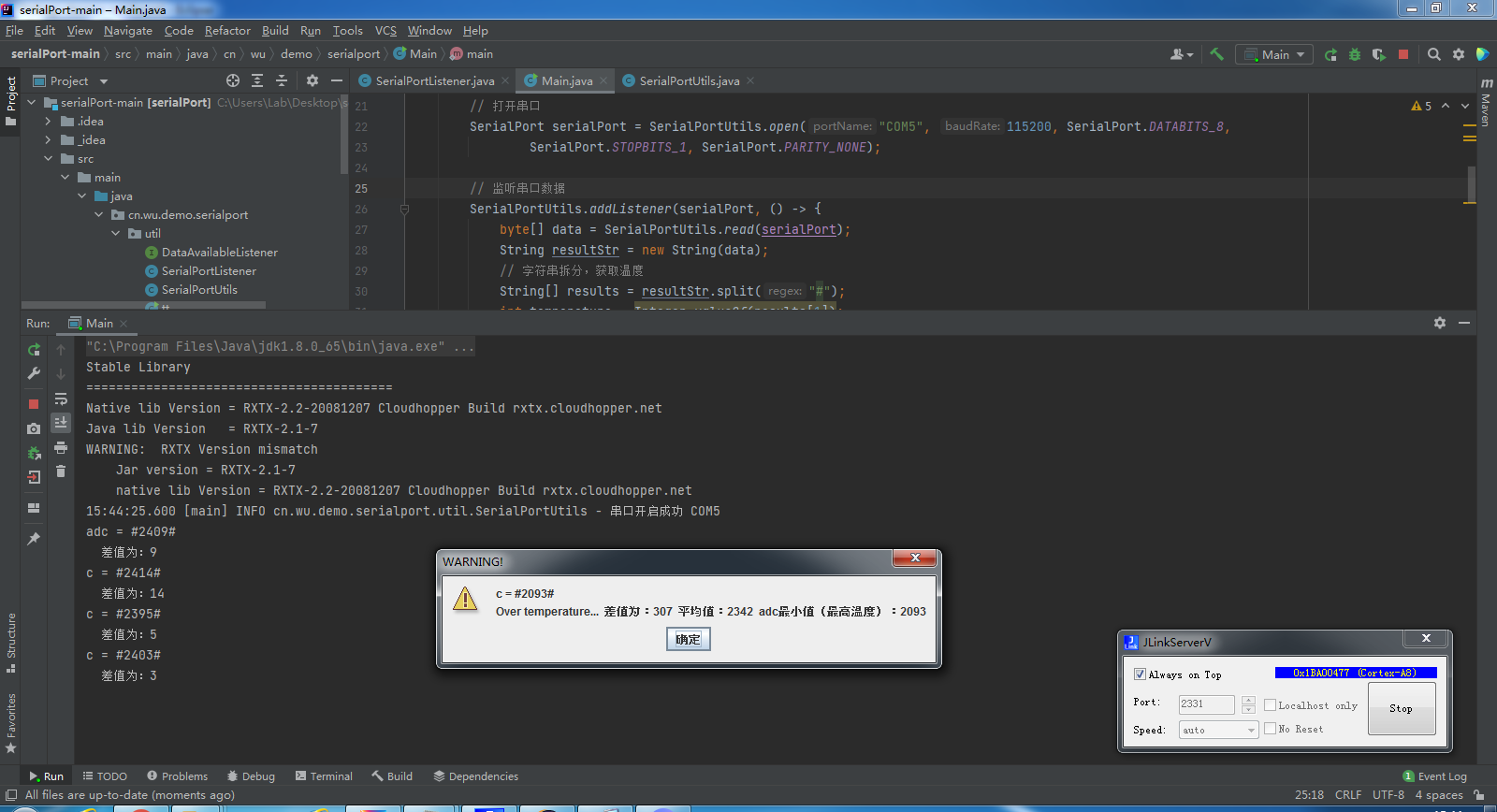


**图5.1.2**

### 5.2 Java串口通信测试

选用的IDE为Jetbrains公司的Intellj IDEA 2021.2 Community版本，将事先编辑好的代码进行Build编译，然后Debug执行，监听器在后台多线程监听串口通信数据。最终效果如图5.2.1所示。

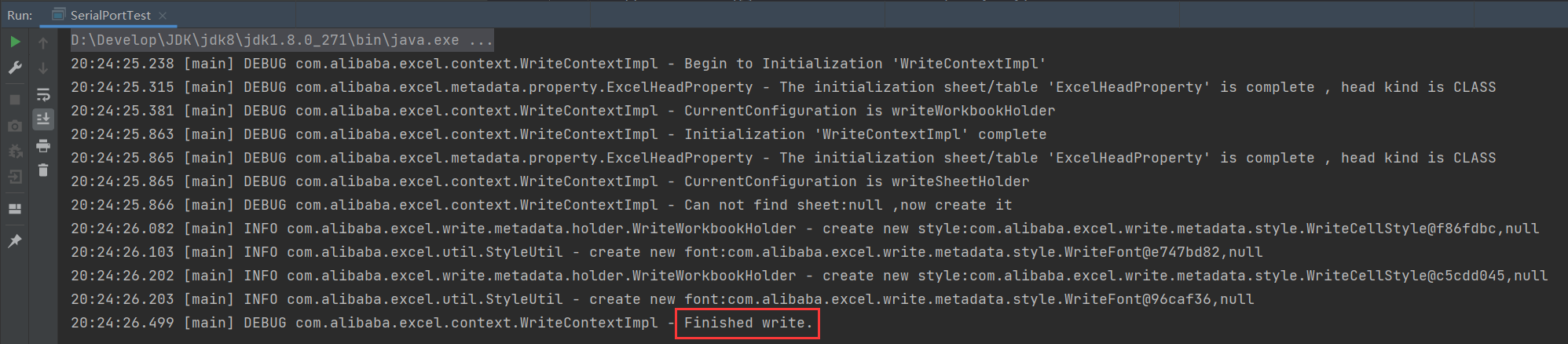
在控制台可以清楚看到每一次的AD转换结果值，同时可以看到与阈值2400的差值，即距离警报的温度差值。在第五次按下按键，同时温度超标的情况下，会出现弹窗警报，此时AD值为2093，小于2400，超出307，前五次的温度均值2342，最高温度为2093。

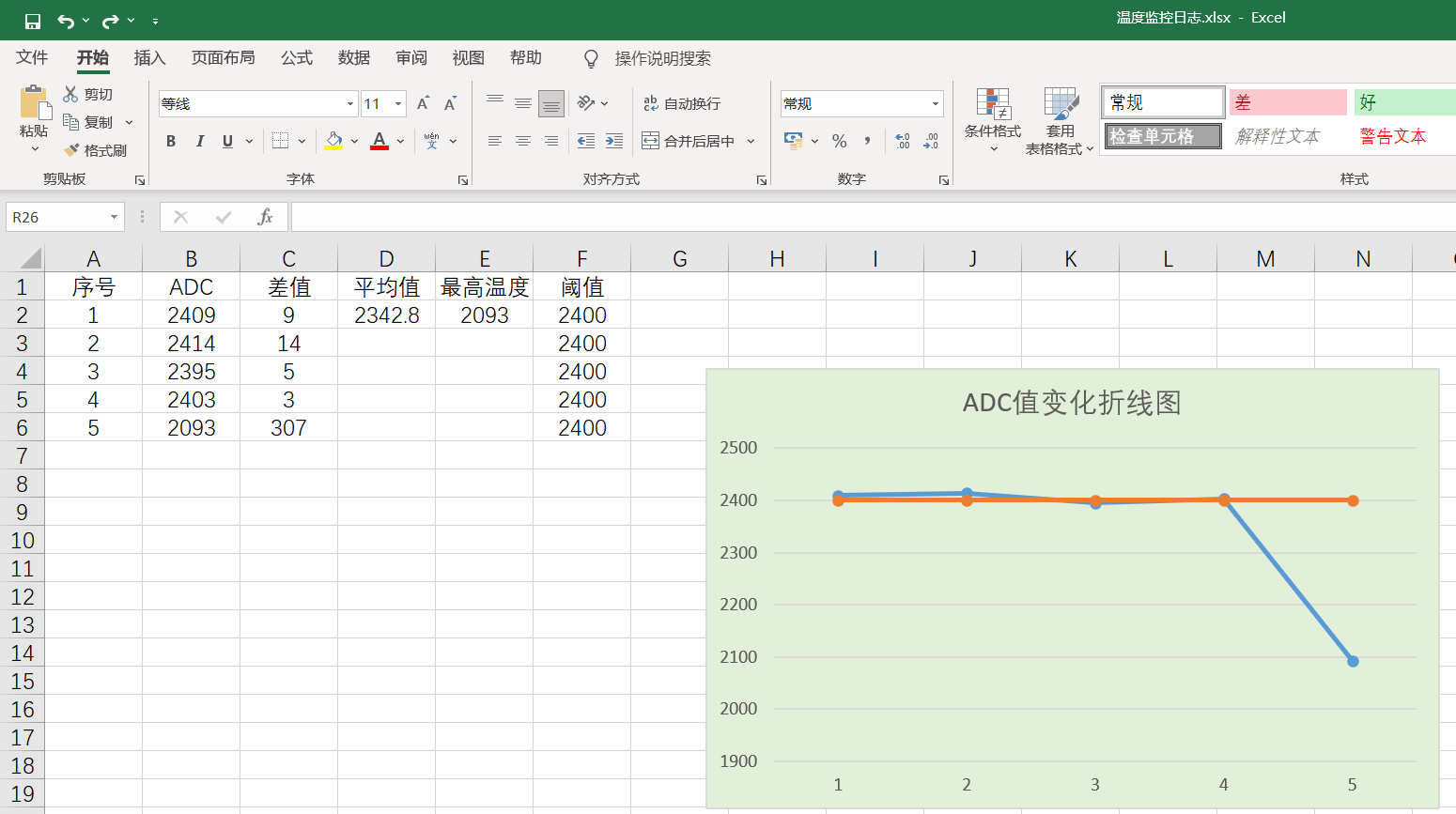




**图5.2.1**

最后将数据通过Hutool IO工具类将数据保存在本地磁盘，使用EasyExcel对数据转化成Excel存储形式，并自动生成折线图，温度走向直观明了。见下图5.2.2所示。





**图5.2.2**

## 6. 结论与特色

本次实验是结合硬件和软件设计一个温控检测系统，热敏电阻、蜂鸣器和LED灯都在A8实验机箱上，当按下按键时，会将当前的温度值转化为AD值，然后将AD值通过串口传输到PC机上，在串口通信工具上显示AD值。

基础功能点需要让指定按键赋能，第五次按下按键判断温度值，若高于阈值，则发出报警信息，报警信息是让蜂鸣器间断鸣叫和灯光闪烁。这部分信息使用汇编和C语言即可实现所有的逻辑，包括硬件初始化，AD转换，温度判断，串口传输。

在实验过程中我发现这套方案存在几个漏洞。① 题目要求的AD值达到3000，在当前实验环境无法达到，同时热敏电阻的特性是温度越高，转化的AD值越低，实验环境的室温在3-10℃，AD值恒定在2500左右；真实的报警场景不应该只有第五次才会报警，在后续按下报警器，若仍然超过温度阈值，依然需要报警。基于上述考虑将报警的逻辑判断修改为按键按动5次及以上且AD值小于2400，触发报警。② 报警事件发生后，警报需要解除，为模拟真实警卫处人工手动按下按键关闭警报，所以在基础上需添加手动解除报警功能。 ③ 使用HyperTerminal终端助手进入Uboot界面，可以实现PC机监听串口，但这是在阻塞环境下的监听，只能对串口通信数据做简单的打印展示，无法进行其他操作，如数学统计。温度超标后，除了事件发生时，立刻触发报警，还应该在实践结束后，对温度超标前的数据进行数学分析，这样普通的温度值信息才具有实际意义。

基于上述②③点的考虑，本实验进行了两点扩展。手动解除警报的功能很好实现，也是防抖检测按键是否按下，然后将蜂鸣器和LED灯的DAT置0。为实现异步监听串口数据，同时将数据进行数学分析，技术选型使用Java及其配套框架实现，具有天然的高级语言优势，可以进行非常复杂的逻辑实现。

本实验一大亮点，也是一大难点就在于Java环境如何替代Terminal进行串口通信，怎么从串口传输过来的字节数据中获取准确的AD值，对数值需要进行哪些有意义的数学统计，统计的结果怎么更好地保存和展示给管理员。

首先在查阅大量资料后，选用Java成熟的串口通信框架RXTX[[1]](#RXTX)；在C程序进行AD转换后获取到的AD值，串口通信发送的数据手动添加数据标识符，这样在Java端使用正则表达式即可非常方便地进行字符串切割获取准确温度值；具有数学统计意义的计算首先是AD值的统计，储存在List中，计算温差，即距离温度超标的“距离”还有多近，计算均值，即报警前均值在怎样的范围，计算AD最小值，即温度的最大值，可直观感受温度超标的程度；警报需要非常醒目，所以使用到JDK Swing弹窗提醒；为了后续更好保存展示日志信息和数学统计，使用EasyExcel[[3]](#EasyExcel)将内存中的数据以Excel形式格式化并写入磁盘中，而Excel本身就是非常强大的数学统计工具，可以生成图标折线图等，这些操作都可以在Java程序里面实现，全程自动化，异步执行，并且程序的可扩展性非常强，甚至可以将程序部署到云端，配合网页前端做到更强大的数据可视化，大数据分析等。

当然，本实验也存在不少缺陷。如：没有实现远程关闭警报，即PC机串口通信通知远程实验机箱关闭警报；串口通信总是会因为电路问题，丢失部分数据，有时丢失严重会影响程序执行，这部分尚待解决。

本次实验总体而言比较成功，实现基础功能点情况下，也对实际应用场景进行思考，在查阅大量资料，结合前面实验的铺垫，实现系统的定制化设计，较全面分析了运行情形，在这一过程使我受益匪浅，得到很大的提升。

本项目已开源Github[[4]](#Github)，所有源码在附件或Github中。

## 7. 参考

[1]Java串口通信框架 <http://fizzed.com/oss/rxtx-for-java>

[2] Java国人开发的优秀工具类 [https://hutool.cn/docs/#/](https://hutool.cn/docs/%23/)

[3]阿里巴巴开源Java解析Excel框架<https://github.com/alibaba/easyexcel>

[4]项目源码 <https://github.com/hypocrite30/SerialPort-RXTX>