

# 上海理工大学

## 单片机原理课程设计报告

### 物联网电子秤



项目组组长（学号） 张路佳（2035062029）

项目组成员（学号） 高浩琦（2035060413）

项目组成员（学号） 魏小柯（2035060303）

指导教师： 侯俊

成绩：

## 目录

1 基本概述.....	3
2 硬件设计.....	3
2.1 功能要求 .....	3
2.2 电路设计框图 .....	3
2.3 数据采集模块 .....	4
2.3.1 压力传感器电路 .....	4
2.3.2 A/D 转换电路 .....	4
2.4 数据处理控制模块 .....	6
2.4.1 复位和 EA 电路 .....	7
2.4.2 晶振电路 .....	7
2.4.3 中断电路 .....	8
2.5 显示模块 .....	8
2.5.1 数码管电路 .....	8
2.5.2 OLED 屏幕电路 .....	9
2.6 串口通讯模块 .....	10
2.6.1 电平转换/虚拟串口电路 .....	10
2.6.2 ESP32 模块电路 .....	11
2.6.3 TCP 协议 .....	11
3 软件设计.....	13
3.1 整体程序结构设计 .....	13
3.2 AD 转换子程序设计 .....	13
3.3 显示子程序设计 .....	13
3.4 串口通讯子程序设计 .....	14
3.5 无线传输模块程序设计 .....	14
4 结论 .....	15
参考文献 .....	15
附录 .....	16
• 附录 I 系统总电路原理图 .....	16
• 附录 II 主程序源代码 .....	16
• 附录 III ESP32 程序源代码 .....	16
• 附录 IV Android 程序源代码 .....	18

# 1 基本概述

随着科技的发展，对物品的称重需求不再满足于显示重量，传统简单机械单一计重秤早已无法满足新要求。数字化、网络化、小型化、多功能集成化成为新的发展趋势，具有精度高、操作简单、体积小、计价与显示等独特优点的电子秤脱颖而出。对此，本文以 89C51 单片机为核心控制芯片，利用 18 位高精度模数转换芯片(MCP3421)和无线技术，完成对具有称重、计价和显示功能，并能通过 WiFi 将数据发送至手机的物联网电子秤的设计。

## 2 硬件设计

### 2.1 功能要求

1. 可以顺利读取传感器和数码转换后的数据；
2. 可以在数码管上显示数据，可显示有符号的小数和整数，小数精确到小数点后四位，整数精确到五位；
3. 在 OLED 屏幕上可以显示该物体的单价和总价；
4. 手机端可接收到相关数据。

### 2.2 电路设计框图

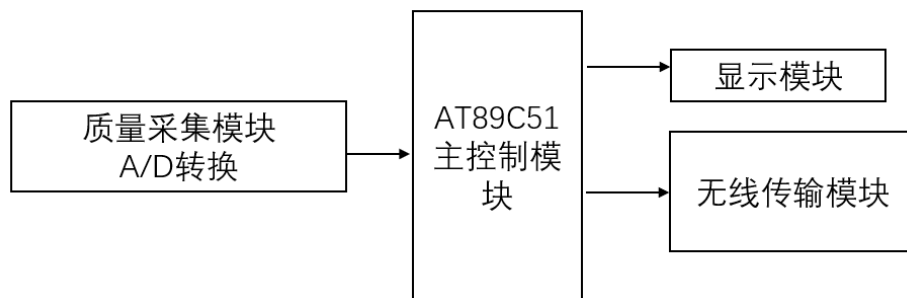


图 2.1：电子秤电路设计框图

本文电子秤系统由四部分组成：质量采集模块、数据处理控制模块、显示模块、无线传输模块。质量采集模块主要用于采集物品质量，并将质量由模拟量转化为数字量，以供单片机处理，该模块由压力传感器及模数转换模块组成；数据处理控制模块用于对采集到的数据进行分析并对整个系统进行控制，其核心为 AT89C51 单片机；显示模块为数码管显示屏输出字符方便用户接收系统传递信息、显示目前状态；无线传输模块使用 ESP32 WIFI 芯片，通过 TCP 协议将收到的来自串口的数据使用 TCP 转发到手机并在手机上显示出来。

硬件的结构和可靠性直接影响着整个系统的可靠性，所以合理安排电路可以在一定程度上提高设计的性能。

## 2.3 数据采集模块

### 2.3.1 压力传感器电路

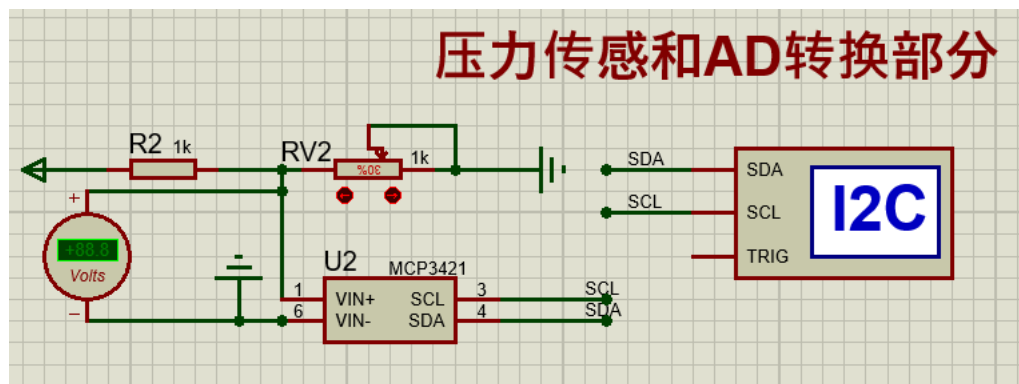


图 2.2: 压力传感器模块电路图

压力计采用应变片式电阻传感器，以应变片为传感器元件，它具有以下优点：精度高、测量范围广；使用寿命长、性能稳定可靠；结构简单、尺寸小、重量轻，因此在测量时，对工件工作状态及应力分布影响小；频率响应特性好，可在低温、高速、高温、强烈振动、强磁场、核辐射和化学腐蚀等恶劣环境条件下工作且应变片种类繁多价格便宜。电阻应变片的工作原理是基于电阻应变效应，即在导体产生机械形变时，它的电阻值相应发生变化，该传感器结构简单、灵敏度高，适用于小压力测量。

本项目采取滑动变阻器模拟应变式传感器，将电压值作为模拟量数据进行采集和处理。故将 A/D 转换模块 MCP3421 芯片  $VIN_+$  和  $VIN_-$  端与电压表并联，进行数据采集后的直接处理。

### 2.3.2 A/D 转换电路

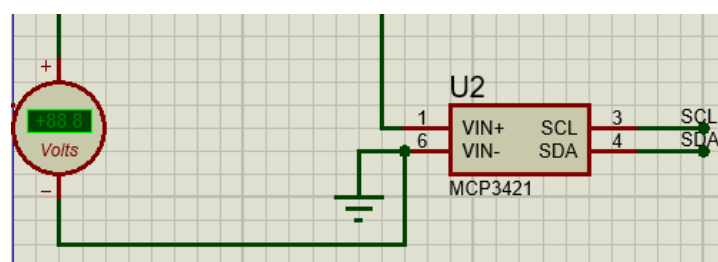


图 2.3: A/D 转换电路图

数模转换模块采取的是 MCP3421，18 位的分辨率器件，其内部采用了 Microchip 专利的差分开关电容转换及数字滤波技术，专为需要高分辨率和低功耗的应用而设计。在这种应用中，空间和低功耗是设计的首要考虑因素。MCP3421 可在 2.7~5.5V 单电源下电压工作，并消耗很低的电流。在  $VDD=3V$ 、单次转换、1sps 条件下，电流消耗仅为  $39\mu A$ （典型值）。

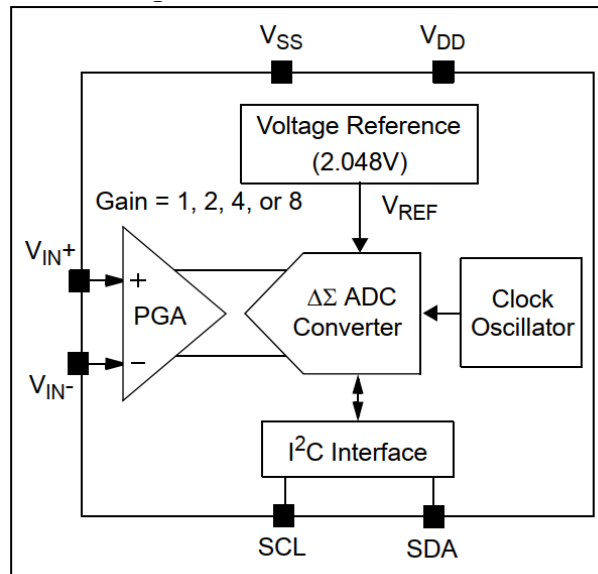


图 2.4: MCP3421 内部结构图

根据 MCP3421 的数据手册可以明确在进行写操作时需要首先由主器件发出启动位，其次在 MCP3421 写入第一个字节，包括器件地址和读写标志位 常规为 0xd0。地址位格式为 1 1 0 1 A2 A1 A0 其中 A2、A1、和 A0 出厂就已经定义号好了，默认为 A2=A1=A0=0。最后一位为读写标志位，1 为读模式，0 为写模式。所以实际写入 MCP3421 的第一个字节为 1101 0000=0xD0，写入后主机等待 MCP3421 发出应答信号，随后向 MCP3421 写入第二个字节（配置字节），进行寄存器的配置，进行后续的调试操作。

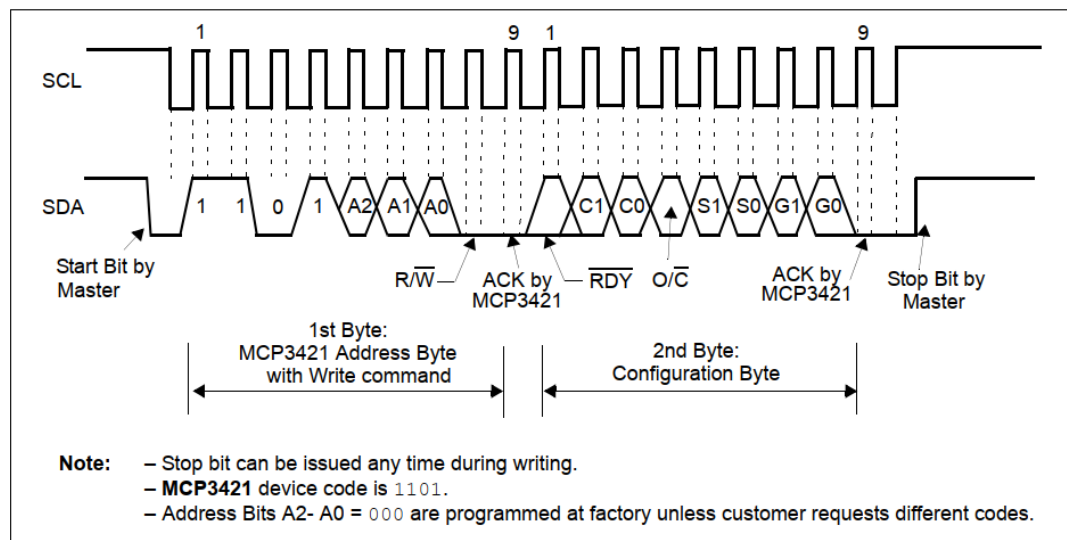


图 2.5: MCP3421 写操作时序图

分析数据转换前，需要了解 MCP3421 数据存储格式，然后才能将采集得到的数据进行正确转换数字输出代码与输入电压和 PGA 设置成正比。输出数据为二进制补码格式。在这种编码方式为：MSB 位可以作为符号位。当 MSB 为逻辑 0，表示为正值；当 MSB 为逻辑 1，表示为负值。输出代码与实际输入电压之间的关系

如下：

输出代码 = (指定采样频率下的最大代码 + 1)\*PGA\*(Vin+ - Vin-)/LSB

其中不同采样速率对应的最大代码如下：

表 1:不同采样率对应的最大代码

分辨率设置	数据率	最小代码	最大代码
12	240SPS	-2048	2047
14	60SPS	-8192	8191
16	15SPS	-32768	32767
18	3.75SPS	-131072	131071

LSB = 2\*VREF/2^SPS=2\*2.048/2^SPS

所以基于以上表述可以得到输出电压的计算表达式如下：

当 MSB=0 (输入正电压)；输入电压=(输出代码)\*LSB/PGA

MSB=1 (输入负电压)；输入电压=(输出代码的二进制补码)\*LSB/PGA

在 MCP3421 芯片处理后，将其输出端直接接入 51 单片机的 P2.6 口和 P2.7 口进行数据的输入。

2.4 数据处理控制模块

AT89C51 单片机是一种 8 位的单芯片微控制器，具有 4K 的 ROM 和 128 字节的 RAM。 51 单片共有 4 组 I/O 口，分别为 P0 口、P1 口、P2 口和 P3 口，除了做普通 I/O,P0 口还可做数据总线和地址总线，P2 口可做地址总线,P3 口可通过对特殊功能寄存器编程来设置特殊功能。EA/VPP 引脚用于选择控制访问存储器，该引脚为高电平时，单片机先读取内部 ROM，再读取外部 ROM 当为低电平时，单片机直接读取外部 ROM，而不读取内部 ROM。RET 为复位引脚，输入超过两个机器周期的高电平即触发有效。XTAL1 和 XTAL2 引脚用于外接晶振，时钟既可由芯片振荡回

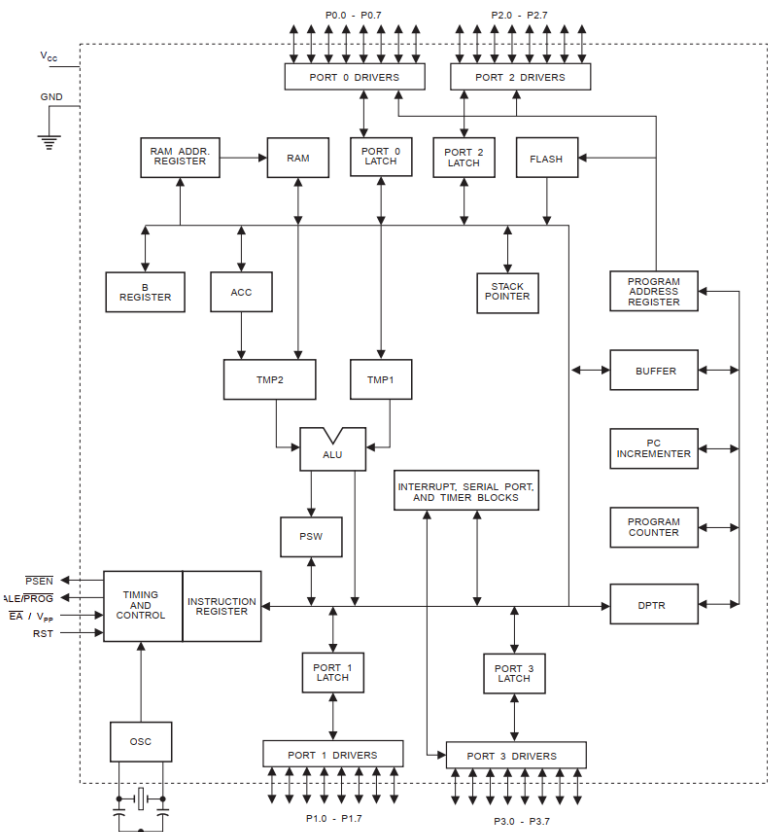


图 2.6: AT89C51 内部结构图

路提供，也可由外部时钟提供，前者需要 XTAL1 和 XTAL2 引脚外接石英晶体和微调电容，后者需要 XTAL1 和 XTAL2 引脚外接时钟脉冲信号。本设计采用使用内部振荡回路，内部振荡回路需要附加外部路形成时钟，即附加外接晶振和电容组成的并联谐振回路。振荡晶体可在 1.2MHz 到 12MHz 内选择，电容可在 20pF 到 100pF 之间选择。51 单片机自带 16 位的定时器/计数器 T0、T1、T2，定时器/计数器有方式 1 和方式 2 两种模式以供选择，模式的选择通过改变特殊功能寄存器 TMOD 来实现，通过指令可以对特殊功能寄存器 TMOD 进行编程。除此之外，定时器/计数器的模式由 TCON 等特殊功能寄存器来控制。无论是定时器还是计数器，当发生溢出时都会向 CPU 发出中断申请，CPU 根据当前中断位的设置判断是否响应。

### 2.4.1 复位和 EA 电路

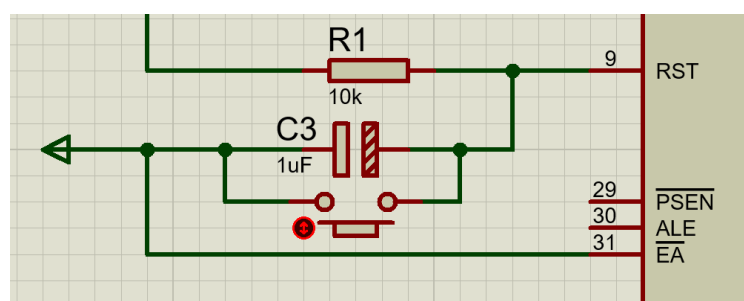


图 2.7: 复位和 EA 电路图

在系统运行的过程中，有时需要对系统进行复位操作，为了避免对硬件系统经常加电和断电造成的损害，本项目采取的为手动的复位电路。如图 2.7 所示，在系统的运行过程中需要复位时，只需使开关闭合，在 RST 端会出现一定时间的高电平信号，从而使单片机实现复位。

AT89C51 芯片 EA 端为低电平有效，所以当 EA 端引脚接高电平时，CPU 只访问片内 Flash ROM 并执行内部程序存储器中的指令；本项目使用内部 ROM 故选择接入高电平。

### 2.4.2 晶振电路

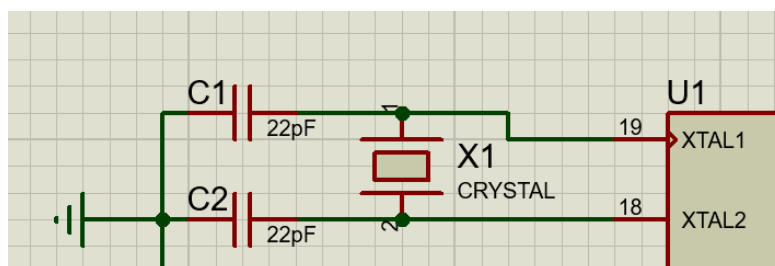


图 2.8: 晶振电路图

单片机必须在时钟的驱动下才能进行工作。51 系列单片机内部都有一个时钟振荡电路，只需外接晶振源，就能产生一定频率的时钟信号送到单片机的内部

的各个单元，从而决定单片机的工作速度。本项目采用外接晶振的方法。电路中的两个电容的作用有两个：一是帮助振荡器起振，C1、C2 的值越大，起振的速度慢，反之则速度快；二是对振荡器的频率起到微调的作用（C1、C2 的值大，频率略有减少，反之，频率略有提高），故本项目 C1 和 C2 的值取用 22pF，晶振频率为 11.0592mHz。

### 2.4.3 中断电路

考虑到计价称重时会有去皮和精度设置的问题，本项目使用外部中断，通过增加按钮来实现先关功能。

其中上侧按钮用于去皮操作，此操作可以触发中断 0，并在中断 0 中实现去皮操作；

下侧按钮用于更换单位操作，同样在串口中断中实现操作。

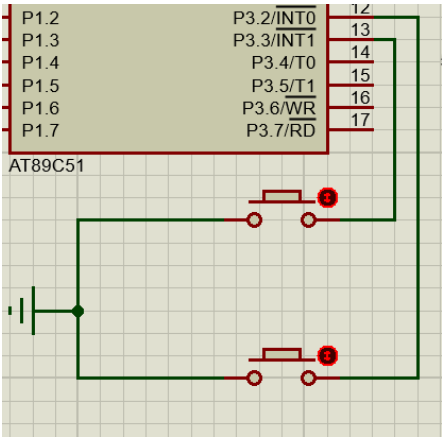


图 2.9：外部中断电路图

## 2.5 显示模块

### 2.5.1 数码管电路

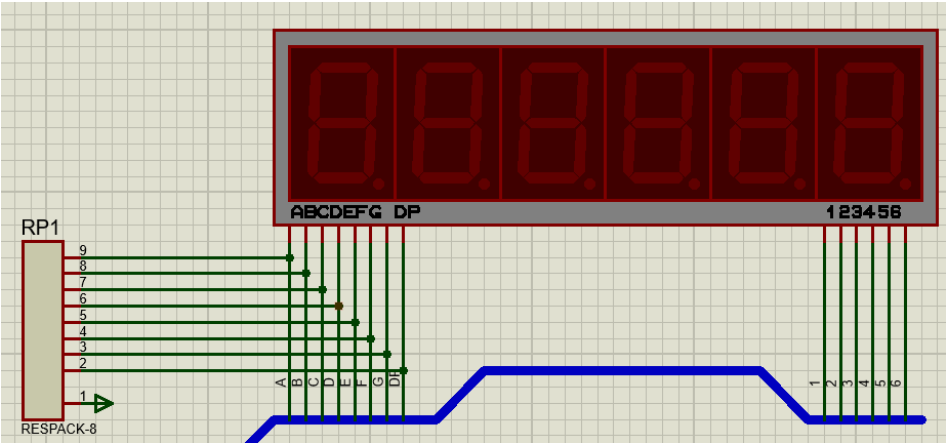


图 2.10：数码管显示电路图

7SEG-MPX6-CA 是 6 位共阳极 7 段式数码显示管。共阳数码管共阳端在接电源前需要接如上拉电阻，保护数码管避免烧坏，且共阳极数码管送高电平时正常工作。



### 2.5.2 OLED 屏幕电路

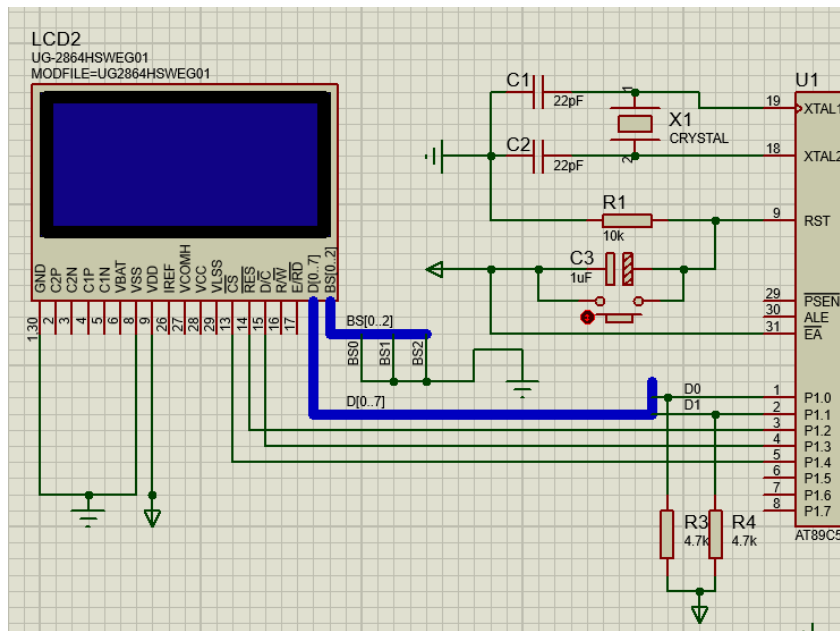


图 2.11: OLED 屏幕电路图

SSD1306 OLED 驱动芯片是一款带控制器的用于 OLED 点阵图形显示系统的单片 CMOS OLED/PLED 驱动器。该芯片专为共阴极 OLED 面板设计。SSD1306 内置对比度控制器、显示 RAM (GDDRAM) 和振荡器，以此减少了外部元件的数量和功耗。

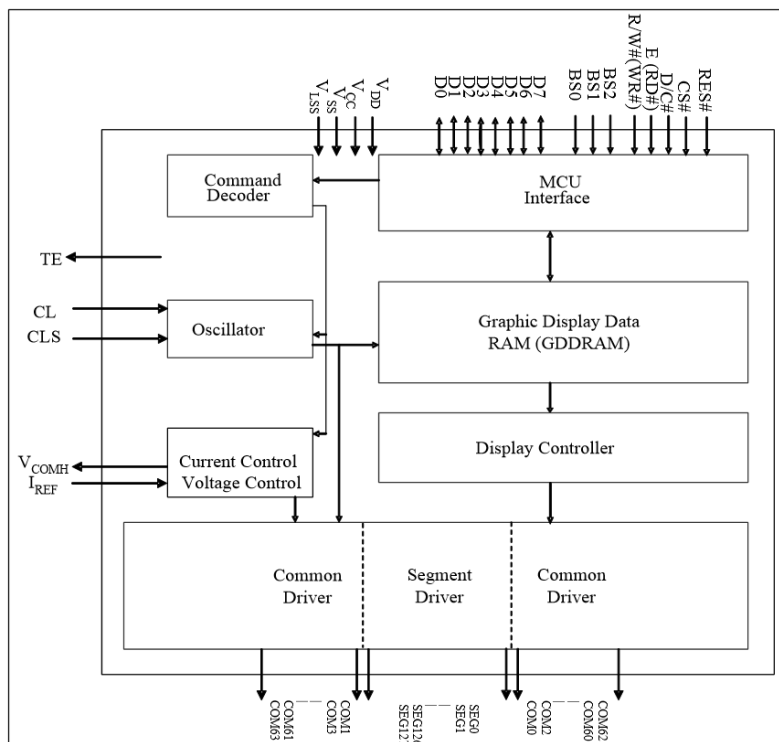


图 2.12: SSD1306 OLED 芯片内部结构图

本项目选择的接口类型为 4 线 SPI，接口引脚包括：串行时钟 (SCLK)、串

行数据(SDIN)、数据/命令控制(D/C#)、片选(CS#)。在 4 线 SPI 模式下,D0 用作 SCLK,D1 用作 SDIN。对于未使用的数据管脚,D2 应保持打开状态。从 D3 到 D7, E 和 R/W#(WR#)的引脚连接到外部地(拉低)。

表 2：线控制引脚表

Function	E	R/W#	CS#	D/C#
Write command	Tie LOW	Tie LOW	L	L
Write data	Tie LOW	Tie LOW	L	h

在 SCLK 每个上升沿, SDIN 上的数据按 D7, D6, ..., D0 的顺序(高位在前)移位到一个 8 位移位寄存器中。每八个时钟对 D/C#进行一次采样, 移位寄存器中的 8 位数据根据 D/C#的采样结果决定写入到图形显示数据 RAM (GDDRAM) 或命令寄存器。4 线 SPI 的写时序如下所示：

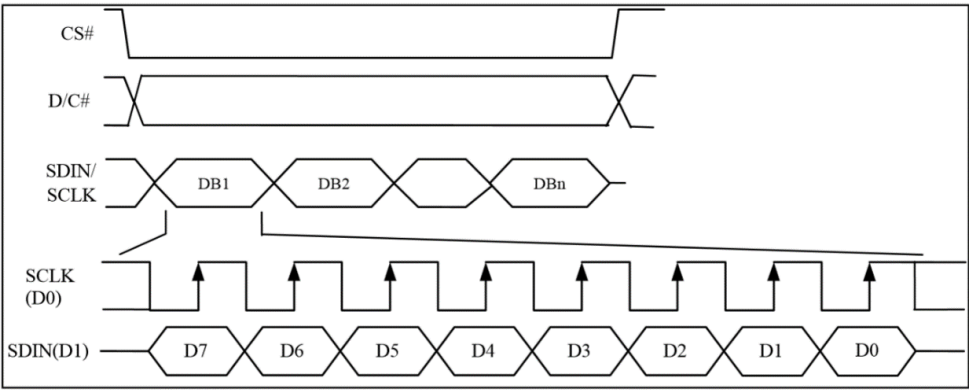


图 2.13:4 线 SPI 时序图

2.6 串口通讯模块

2.6.1 电平转换/虚拟串口电路

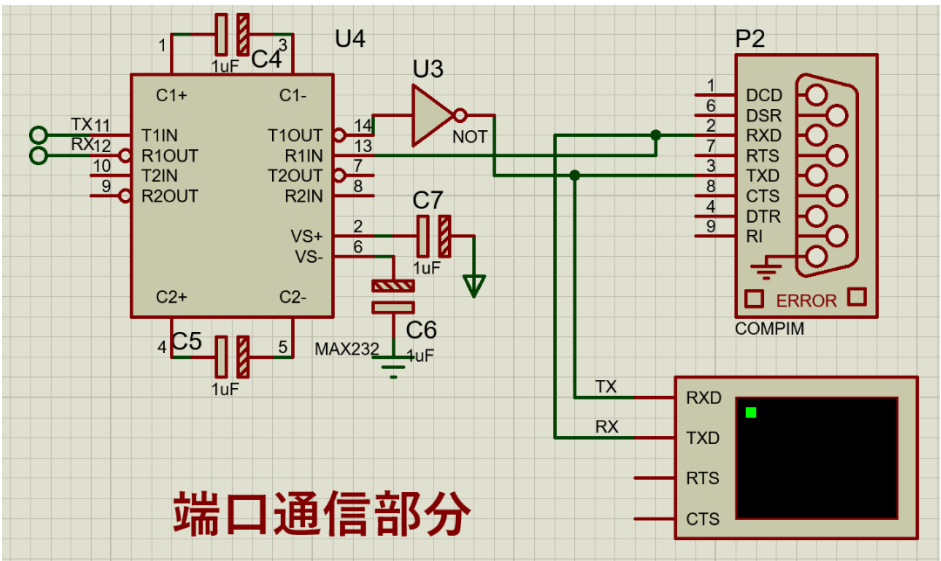


图 2.14：端口电路图

当单片机和其他外设通过串口进行通信时，尽管单片机有串行通信的功能，但单片机提供的信号电平和部分外设的标准不一样，因此要通过进行电平转换。本项目选取 MAX232 芯片，MAX232 是一种双组驱动器/接收器，片内含有一个电容性电压发生器以便在单 5V 电源供电时提供 EIA/TIA-232-E 电平。

电荷泵电路。由 1、2、3、4、5、6 脚和 4 只电容构成，以产生+12V 和-12V 两个电源，提供给串口电平的需要。且 7、8、9、10、11、12、13、14 脚构成两个数据通道。而本项目采用 13 脚（R1IN）、12 脚（R1OUT）、11 脚（T1IN）、14 脚（T1OUT）第一数据通道进行数据传输。

由于本项目在 Proteus 中进行仿真，故为了模拟发出串口的功能，选择连接 COMPIM（虚拟串口），同时在电脑上安装 Virtual Serial Port Driver 软件进行仿真，且通过该软件将虚拟串口内容转发到电脑物理串口输出连接到 ESP32 WIFI 芯片转发数据。

### 2.6.2 ESP32 模块电路

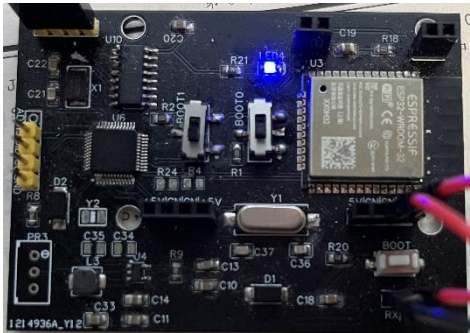


图 2.15: ESP32 物理板

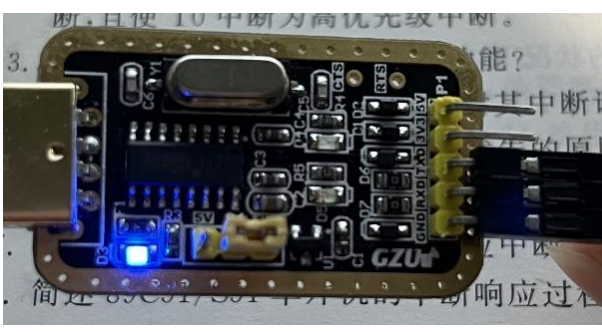


图 2.16: CH340 usb 转串口

ESP32 采用两个哈佛结构 Xtensa LX6 CPU 构成双核系统。所有的片上存储器、片外存储器以及外设都分布在两个 CPU 的数据总线和 / 或指令总线上，两个 CPU 的地址映射呈对称结构，即使用相同的地址访问同一目标。系统中多个外设能够通过 DMA 访问片上存储器。两个 CPU 的名称分别是“PRO\_CPU”和“APP\_CPU”，正常情况下两个 CPU 的功能是相同的。

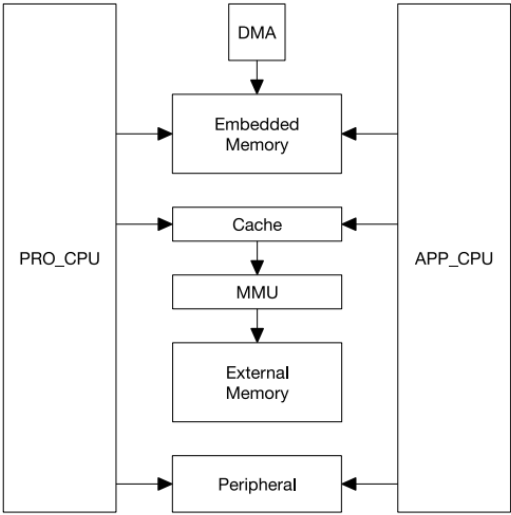


图 2.17: ESP32 系统结构图

### 2.6.3 TCP 协议

传输控制协议（TCP，Transmission Control Protocol）是为了在不可靠的互联网络上提供可靠的端到端字节流而专门设计的一个传输协议。

TCP 是因特网中的传输层协议，使用三次握手协议建立连接。当主动方发出 SYN 连接请求后，等待对方回答 SYN+ACK，并最终对对方的 SYN 执行 ACK 确认。这种建立连接的方法可以防止产生错误的连接，TCP 使用的流量控制协议是可变大小的滑动窗口协议。

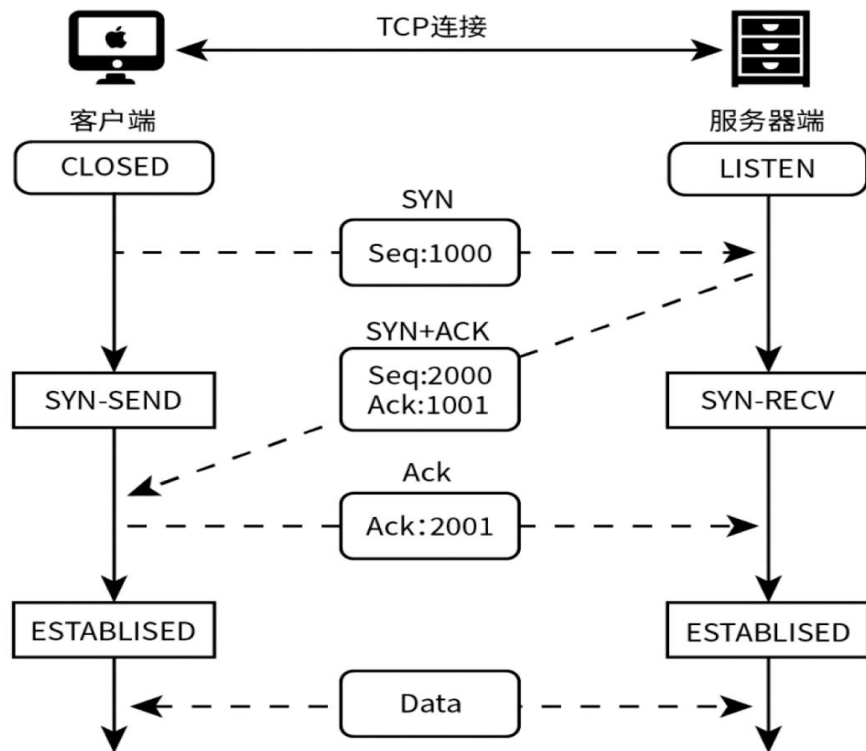


图 2.18: TCP 协议原理图

## 3 软件设计

### 3.1 整体程序结构设计

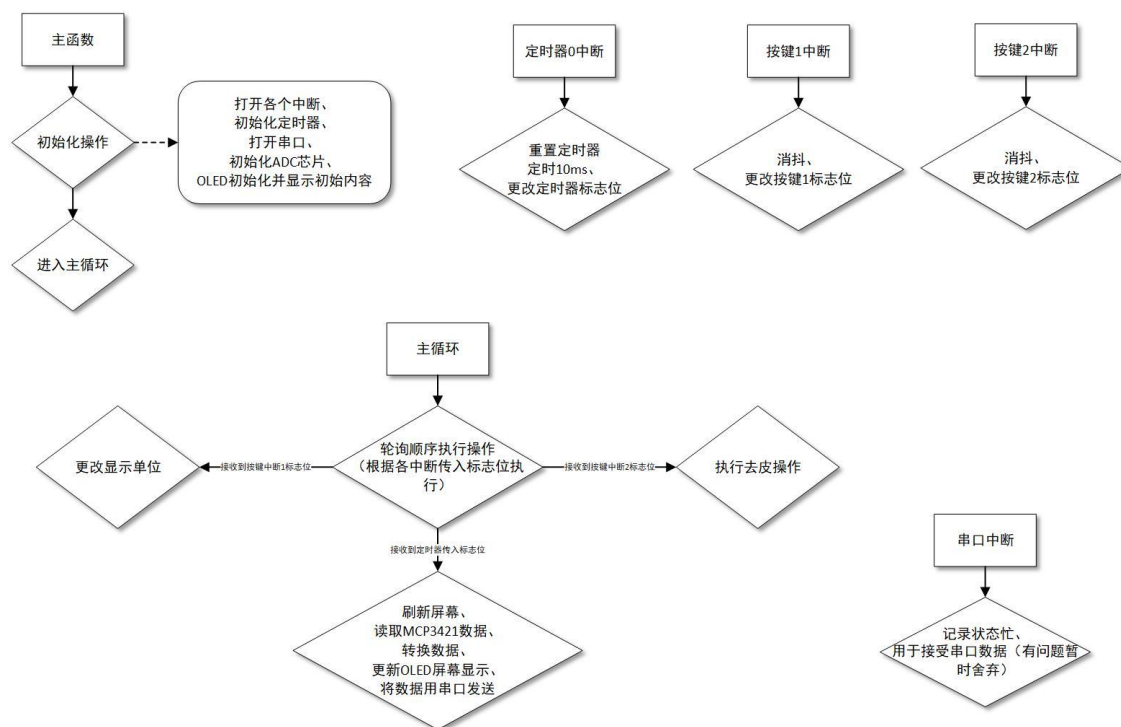


图 3.1: 整体程序结构设计

### 3.2 AD 转换子程序设计

位于 mcp3421.h 和 mcp3421.c 文件中，在其中定义了函数：

`Write_MCP3421(unsigned char WR_Data)`: 用于向 MCP3421 发送指令、

`float READ_MCP3421()`: 用于读取 MCP3421 传回的数据并直接输出转换结果。

### 3.3 显示子程序设计

位于 oled.h、oled.c 文件中（数码管部分位于 mp6.h、mp6.c 文件）

其中 OLED 显示部分，定义了一系列函数用于 OLED 屏幕的驱动（见附件代码部分），并通过取字模软件截取必要中文字符存储在单片机内存中；

数码管显示部分，定义了函数：

`void add_point(u8 pos)`: 用于向指定位置添加小数点

`void show(float kg)`: 用于显示浮点数于数码管，支持正负且自由精度（如输入 1.11 则显示 1.11000，输入 123.45678 则显示 123.456，输入 -123.45678 则显示 -123.45）

`void screen_refresh(int)`: 用于数码管刷新，该函数将被放置在定时器中断用于刷新数码管（\*注\*因为 89C51 稀缺的资源以及仿真的卡顿，该部分暂时被从代码中被删除，单独的实现效果如下）

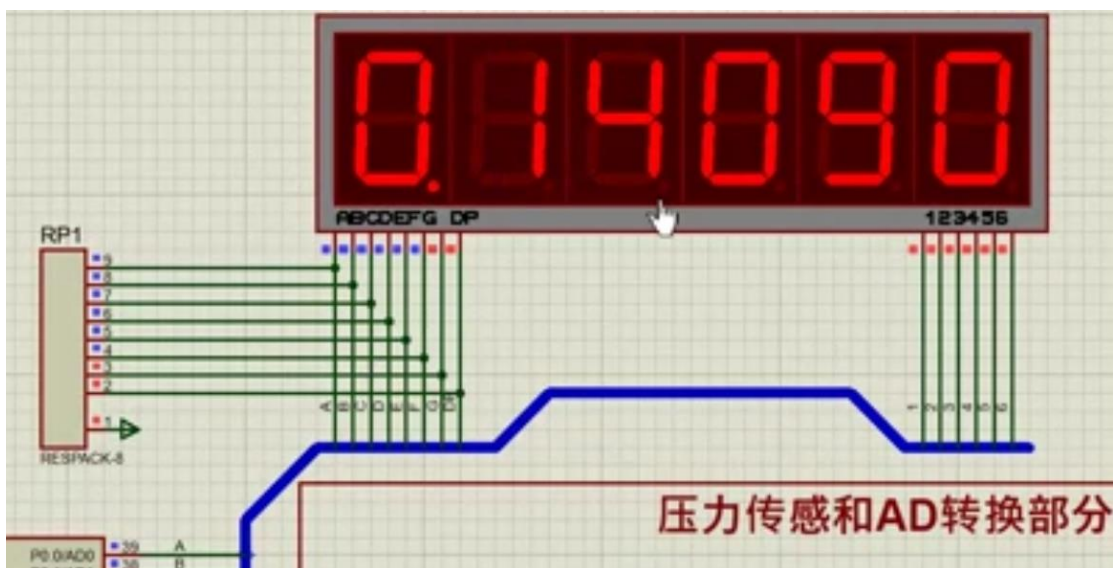


图 3.2：数码管刷新函数实现效果图

### 3.4 串口通讯子程序设计

位于 uart.h、uart.c 文件中，定义函数：

`void SendData(BYTE dat)`: 用于发送单个字节数据；

`void SendString(char *s)`: 用于发送一个字符串。

### 3.5 无线传输模块程序设计

ESP32 程序及 Android 程序见附件。

Android 程序界面简单设计：



图 3.3:Android 程序界面图

## 4 结论

随着电子技术的不断发展,未来电子秤的功能注定得到不断的改进。本文电子秤运用 MCP3421、AT89C51 以及 WiFi 蓝牙技术,能对 2kg 以内物品进行重量称量并计价,重量误差在 0.001kg 以内,精度高,体积小,能够显示物品质量,操作简单,携带方便,体现了一定的实用价值和创新性。

同时在此次的项目实践的过程中,从设计目标的制定,到电路原理图的搭建,程序代码的调试,逐渐梳理清楚如何完整地完一个单片机项目,如何进行功能的调试。并且把长学期中学习到理论知识应用到实际的项目中,从实际的需求中发现问题并努力解决问题,像单片机的内部程序空间如何有效地调用,每个端口到底如何连接,总线如何设计,通讯协议如何真正建立等。

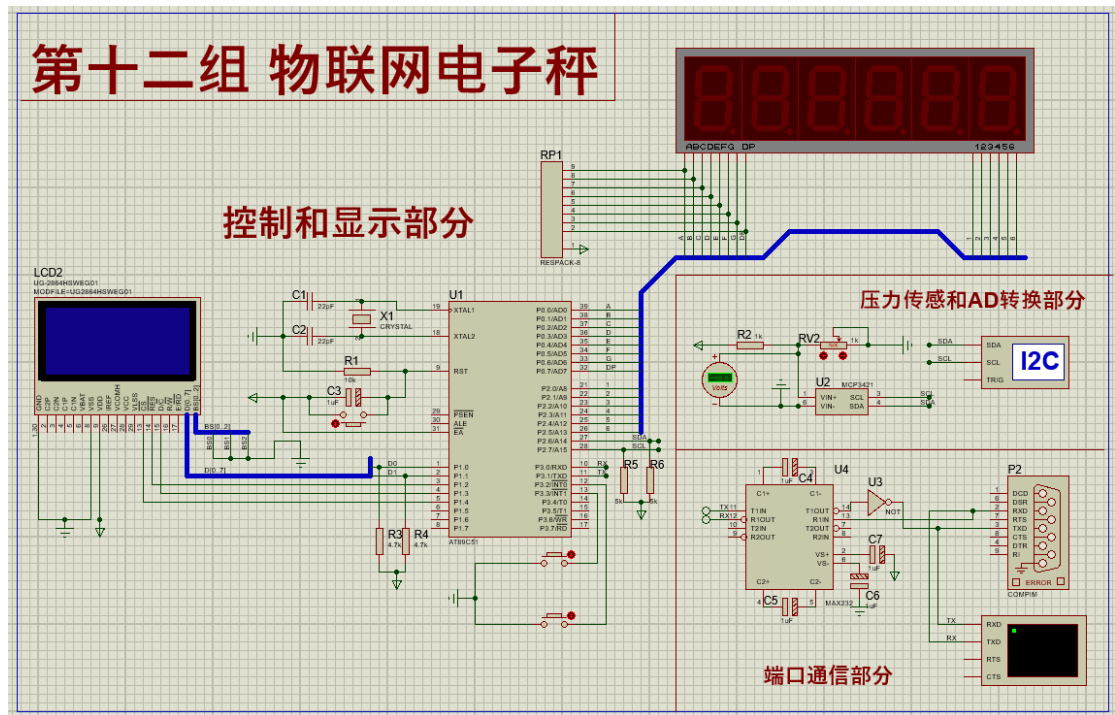
## 参考文献

- [1]AT89C51 原版数据手册
- [2]MCP3421 数据手册
- [3]SSD1306 OLED 数据手册
- [4]ESP32 数据手册
- [5]吴莹妃. 基于单片机与蓝牙传输的电子秤设计[J]. 现代信息科技, 2022(6): 181-185
- [6]谢健翔. 基于 51 单片机为核心的智能压力传感器[J]. 机电产品开发与创新, 2011(24): 127-128.
- [7]张祥瑞, 鲁西坤. 基于 51 单片机的电子秤设计[J]. 设计研发, 2019(11): 32-31
- [8]徐婷, 钱松荣. 基于单片机压力监测系统设计与仿真[J]. 现代电子技术, 2019(42): 468.
- [9]吴玉林, 方鹏斌, 严黎华. 简易数字电子秤的设计[J]. 咸宁学院学报, 2010(30): 11-12
- [10]沈梁, 王鑫荣, 陈文召. 蓝牙技术原理与技术综述[J]. 无线电工程, 2016, 31(9): 17-20
- [11]彭茗, 王凌志, 时慧慧. 基于 STC89C52 单片机的智能电子称设[P]. 南京工程学院 2021



# 附录

- 附录 I 系统总电路原理图



附录一：完整版电路原理图

- 附录 II 主程序源代码

见附件压缩包 escales\_c51.zip

- 附录 III ESP32 程序源代码

```
#include <WiFi.h>
WiFiServer server(2333); //声明服务器对象
uint64_t chipid=ESP.getEfuseMac(); //获取芯片唯一 MAC 地址
unsigned long long1 = (unsigned long)((chipid & 0xFFFF0000) >> 16 );
unsigned long long2 = (unsigned long)((chipid & 0x0000FFFF));
String hex = String(long1, HEX) + String(long2, HEX); // six octets
void setup()
{
  Serial.begin(921600); //串口初始化
  WiFi.softAP(hex.c_str(),"12345678");
  Serial.println("Connected");
  server.begin(); //服务器启动, 端口号 2333
}
long int flag = 1;
```



```

int sum = 0, i = 0;
String readserial;
String readBuff="";    //读取信息暂存
void loop()
{
    WiFiClient client = server.available(); //尝试建立客户对象
    if (client) //如果当前客户可用
    {
        Serial.println("[存在客户端连接]");
        while (client.connected()) //如果客户端处于连接状态
        {
            while (Serial.available() > 0)
            {
                readserial += char(Serial.read()); //复制数据到发送缓存
                sum++;
                if (sum == 500)
                { sum = 0;
                  client.print(readserial);
                  readserial = "";
                }
            }
            if (sum != 0) {
                flag++;
            }
            else if (flag != 0) {
                flag = 0;
            }
            if (sum != 0 && flag > 10000) {
                client.print(readserial);
                readserial = "";
                sum = 0;
                flag = 0;
            }
        }
        while(client.available()) //如果有可读数据
        {
            char c = client.read(); //读取一个字节
            readBuff += c;
        }
        if (readBuff != "") {
            Serial.println(readBuff); //从串口打印
            readBuff = "";
        }
    }
}

```

}

- 附录IV Android 程序源代码

见附件 `escales_android.zip`