**目录**

[1 设计题目 1](#_Toc43200190)

[2 设计目的 1](#_Toc43200191)

[3 设计内容及要求 1](#_Toc43200192)

[4 系统总体结构 1](#_Toc43200193)

[5 硬件设计 2](#_Toc43200194)

[5.1 元件选取 2](#_Toc43200195)

[5.2 电路设计 9](#_Toc43200196)

[6 软件设计 12](#_Toc43200197)

[6.1 中断流程图 12](#_Toc43200198)

[6.2 主函数流程图 13](#_Toc43200199)

[6.3 显示函数流程图 15](#_Toc43200200)

[7 硬件调试 16](#_Toc43200201)

[7.1 电桥调零 16](#_Toc43200202)

[7.2 实际电子秤显示砝码质量 16](#_Toc43200203)

[7.3 电压显示程序烧录验证 17](#_Toc43200204)

[7.4 多次曲线拟合和数据记录 18](#_Toc43200205)

[7.5 数据记录 19](#_Toc43200206)

[8 设计小结 22](#_Toc43200207)

[9 参考文献 22](#_Toc43200208)

[附件1 电路原理图 23](#_Toc43200209)

[附件2 PCB 24](#_Toc43200210)

[附件2 源程序代码 25](#_Toc43200211)

**图目录**

[图 4‑1系统框图 2](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200212)

[图 5‑1 MSP430G2553微控制器结构 2](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200213)

[图 5‑2 ADC10总体结构框图 3](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200214)

[图 5‑3 LCD控制流程 4](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200215)

[图 5‑4 LCD驱动模块 4](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200216)

[图 5‑5 基于I2C的IO扩展 4](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200217)

[图 5‑6 全桥结构图 5](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200218)

[图 5‑7 LM324实物俯视图 7](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200219)

[图 5‑8 LM324内部透视图 7](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200220)

[图 5‑9 元件图 7](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200221)

[图 5‑10 TL431的实物 7](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200222)

[图 5‑11 TL431的使用方法 8](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200223)

[图 5‑12 AD620 8](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200224)

[图 5‑13 恒压源、电桥模块 9](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200225)

[图 5‑14 电桥及调零电路 9](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200226)

[图 5‑15 放大电路 10](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200227)

[图 5‑16 按键使用 11](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200228)

[图 5‑17 128段液晶显示 11](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200229)

[图 6‑1 看门狗定时中断 12](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200230)

[图 6‑2 ADC10中断流程图 12](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200231)

[图 6‑3 主函数流程图 13](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200232)

[图 6‑4 通过Grace配置ADC10 14](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200233)

[图 6‑5 ADC10相关时钟配置 14](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200234)

[图 6‑6 显示函数流程图 15](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200235)

[图 7‑1 电桥调零 16](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200236)

[图 7‑2 实际电子秤示数 16](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200237)

[图 7‑3 程序烧录验证 17](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200238)

[图 7‑4 万用表显示与LCD显示值记录 17](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200239)

[图 7‑5 200g砝码 18](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200240)

[图 7‑6 280g砝码 18](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200241)

[图 7‑7 拟合曲线 21](file:///D:\传感器课程设计\电子秤\报告\电子秤.docx#_Toc43200242)

**表目录**

[表格 7‑1 调试数据记录 19](#_Toc43200243)

**摘要**

本文基于MSP430单片机，完成了一款简易电子秤的设计。以MSP430单片机作为控制核心，采用电阻应变片称重传感器采集模拟信号，完成了信号放大与处理电路设计，经单片机自带10位AD进行模数转换，然后进行数据处理后换算成砝码质量[1]。利用MSP430单片机的休眠与唤醒功能，功耗低，经过测试，装置测量精度高，具有较好的应用价值。

**关键字**：电压采集；单片机；MSP430；A/D转换；电阻式应变片

**Abstract**

Based on MSP430 single chip microcomputer, this paper completes the design of a simple electronic scale.

The MSP430 single chip microcomputer is taken as the control core, and the resistance strain gauge weighing sensor is used to collect the analog signal, and the signal amplification and processing circuit design is completed. The single chip microcomputer is equipped with 10-bit AD for analog-to-digital conversion, and then the data is processed and converted into weight mass.

MSP430 single chip microcomputer dormancy and wake-up function, low power consumption, through testing, the device measurement accuracy is high, has a good application value.

**Keywords:** Voltage acquisition; Single chip microcomputer; MSP430; A/D conversion; Resistance strain gauge

**电子秤设计**

# 设计题目

电子秤。

# 设计目的

运用单片机原理及其应用等课程知识,根据题目要求进行软硬件系统的设计和调试，从而加深对传感器的理解，把学过的比较零碎的知识系统化，比较系统地学习开发单片机应用系统的基本步骤和基本方法，使应用知识能力、设计能力、调试能力以及报告撰写能力等有一定的提高。

# 设计内容及要求

设计制作一简易电子秤，测量范围0~1kg，精度±1％,具有显示、去皮等功能，显示分辨率0.1g。

# 系统总体结构

此次简易电子秤包括：应变片电桥模块，恒压源模块，调零电路，显示模块，运放模块，按键模块。

系统上电，由TL431产生恒压源供给电桥模块恒定的电压。电桥模块通过动态平衡输出变化的电压，将输出的差分电压交给由LM324配合电阻构成的运放进行一级差动电压放大，再用集成仪用放大器AD620进行二级放大，单端输出放大后的电压。单片机MSP430G2553采集输出电压，进行ADC转化。转化完成后由扩展板上的128段液晶显示屏显示，同时扩展板上的按键可以控制去皮功能。

系统总设计框图如下：

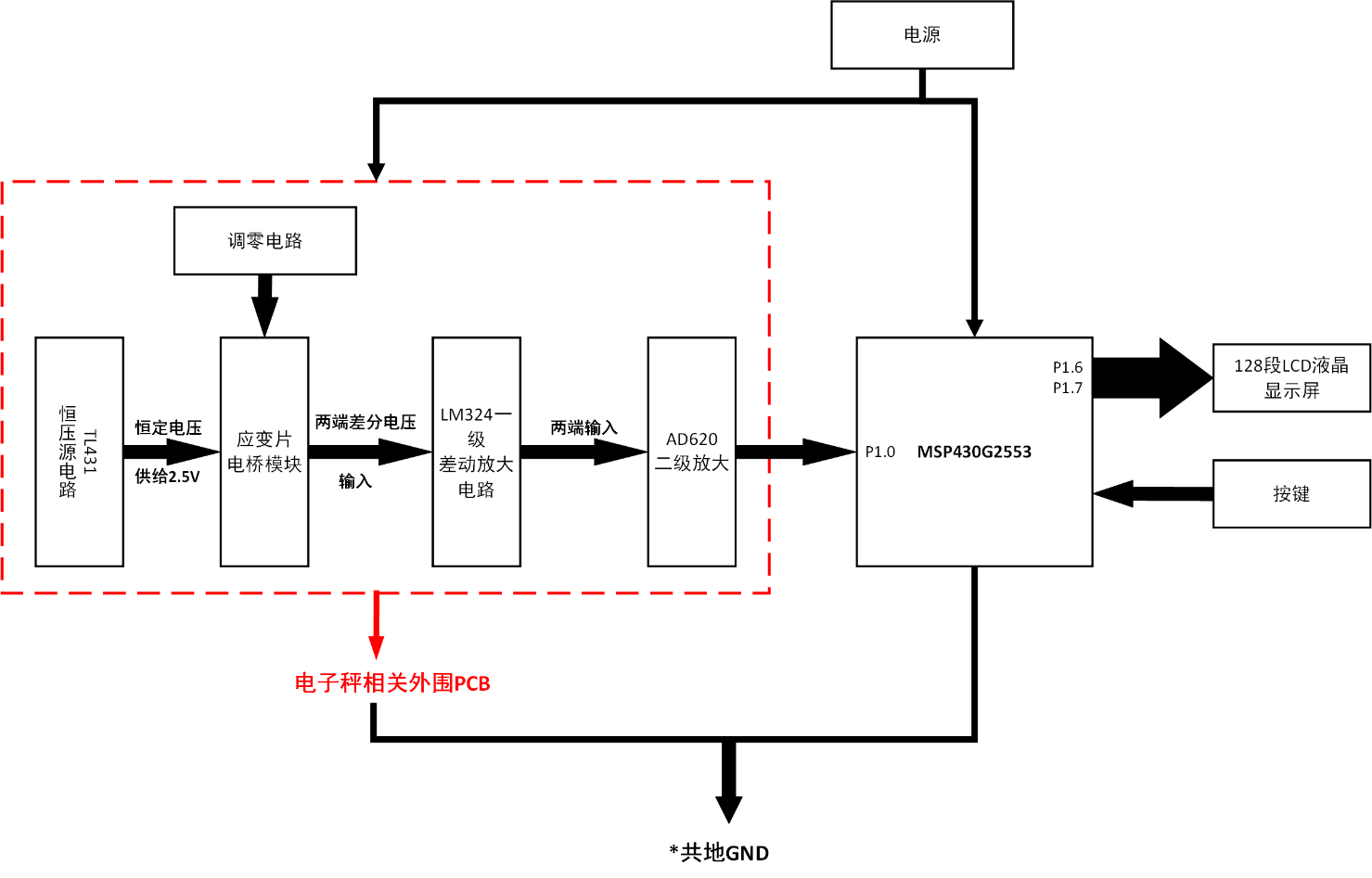


图 4‑1系统框图

# 硬件设计

## 元件选取

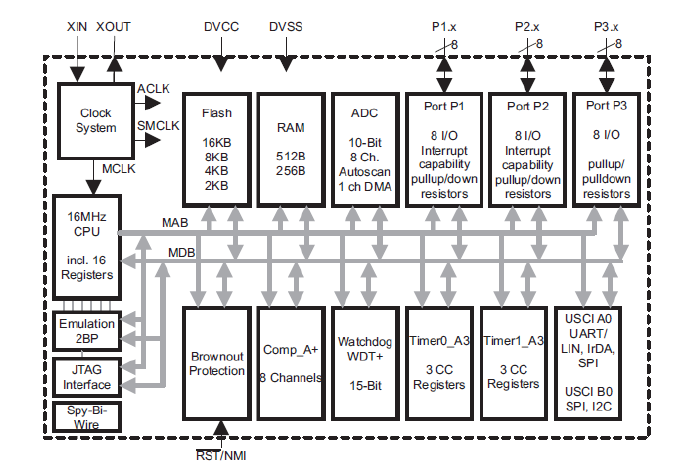
1. **MSP430G2553**

图 5‑1 MSP430G2553微控制器结构

如图 5‑1所示，此次课程设计主要用到了MSP430G2553内部的ADC功能。

ADC10是MSP430单片机的片上模数转换器，其转换位数为10bit，该模块内部是一个SAR型的AD内核，可以在片内产生参考电压，并且具有数据传输控制器。数据传输控制器能够在CPU不参与的情况下，完成AD数据向内存任意位置的传输。

ADC10模块工作的核心是ADC10的核，即图中的10-bitSAR。ADC10的核将模拟量转换成10位数字量并储存在ADC10MEM寄存器里。这个核使用VR+和VR-来决定转换模拟值的门限。模拟电压Vin的输入范围：VR\_-≤Vin≤VR\_+。当输入Vin≥VR+，NADC=1023。当输入Vin≤VR-，NADC=0。采样值的计算公式为：

反推模拟量的大小Vin

本次被测电阻两端的电压可由Vin得到，以下是ADC10总体结构框图。

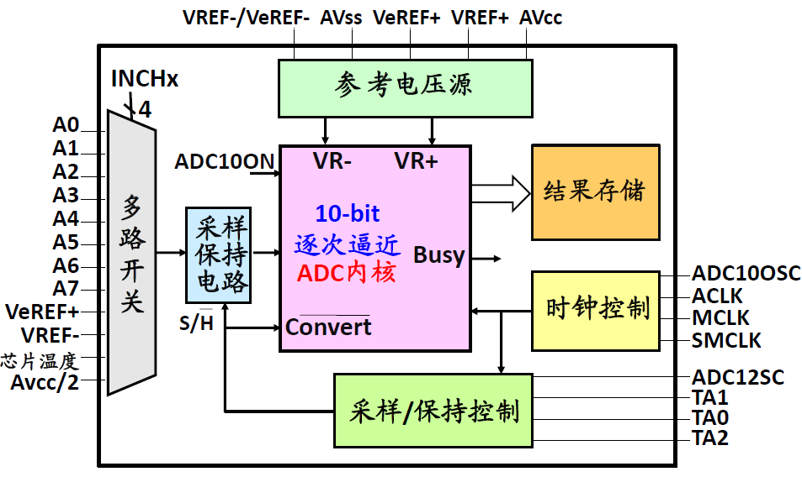


图 5‑2 ADC10总体结构框图

本次课程设计采样引脚为P1.0，对应的通道为A0，内部采取的参考电压为2.5V，所以单片机实际采样得到的电压计算公式为

1. **128段LCD液晶屏**

MSP430G2553通过I2C协议SCL、SDA，对应为P1.6和P1.7去控制扩展版上的TCA6416A芯片输出4个信号CS、WR、RD、DATA，对应的引脚分别为P1.4、P1.6、P1.5、P1.7，控制LCD驱动芯片HT1621，来实现128段LCD的显示。下和为LCD的控制流程和LCD原理图

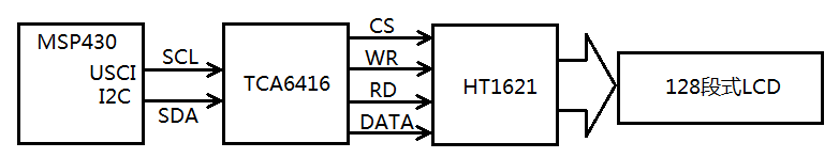


图 5‑3 LCD控制流程

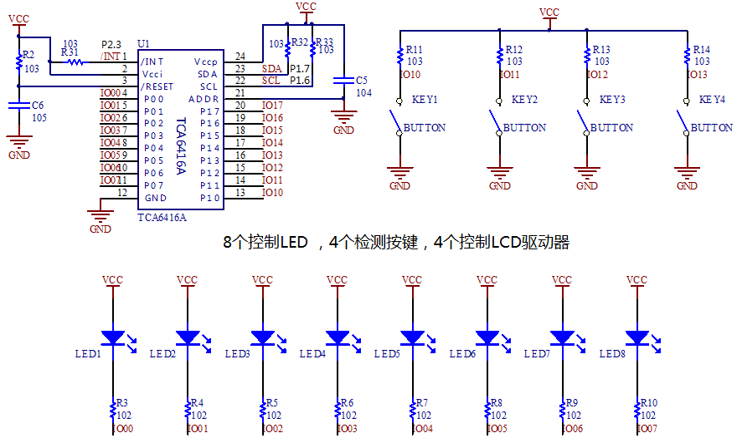
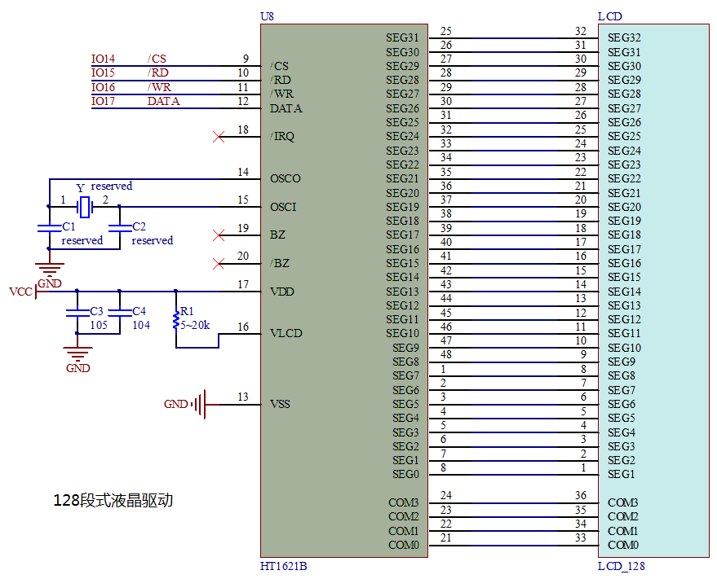


图 5‑4 LCD驱动模块

图 5‑5 基于I2C的IO扩展

1. **传感器选取**

压电传感器是一种典型的有源传感器，又称自发电式传感器。其工作原理是基于某些材料受力后在其相应的特定表面产生电荷的压电效应。

压电传感器体积小、重量轻、结构简单、工作可靠，适用于动态力学量的测量，不适合测频率太低的被测量，更不能测静态量。目前多用于加速度和动态力或压力的测量。压电器件的弱点：高内阻、小功率。功率小，输出的能量微弱，电缆的分布电容及噪声干扰影响输出特性，这对外接电路要求很高。

电阻应变式传感器是一种利用电阻应变效应，将各种力学量转换为电信号的结构型传感器。电阻应变片式电阻应变式传感器的核心元件，其工作原理是基于材料的电阻应变效应，电阻应变片即可单独作为传感器使用，又能作为敏感元件结合弹性元件构成力学量传感器。

导体的电阻随着机械变形而发生变化的现象叫做电阻应变效应。电阻应变片把机械应变信号转换为△R/R后，由于应变量及相应电阻变化一般都很微小，难以直接精确测量，且不便处理。因此，要采用转换电路把应变片的△R/R变化转换成电压或电流变化。其转换电路常用测量电桥。

直流电桥的特点是信号不会受各元件和导线的分布电感及电容的影响，抗干扰能力强，但因机械应变的输出信号小，要求用高增益和高稳定性的放大器放大。

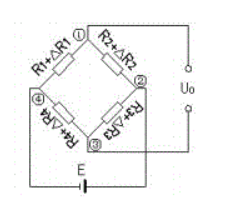
下图 5‑6为一直流供电的平衡电阻电桥，②④接直流电源E，①③输出变化的电压：

图 5‑6 全桥结构图

当电桥输出端接无穷大负载电阻时，可视输出端为开路，此时直流电桥称为电压桥，即只有电压输出。当满足条件

U0=0，即电桥平衡。公式5-4称平衡条件。

应变片测量电桥在测量前使电桥平衡，从而使测量时电桥输出电压只与应变片感受的应变所引起的电阻变化有关。

若差动工作，即R1=R-△R，R2=R+△R，R3=R-△R，R4=R+△R

则电桥输出为

其中为电阻丝相对变化，K为应变灵敏系数，为电阻丝长度相对变化。

应变片式传感器有如下特点：

（1）应用和测量范围广，应变片可制成各种机械量传感器。

（2）分辨力和灵敏度高，精度较高。

（3）结构轻小，对试件影响小，对复杂环境适应性强，可在高温、高压、强磁场等特殊环境中使用，频率响应好。

课题要求称重范围0～1Kg，精度±1％，考虑到秤台自重、振动和冲击分量，还要避免超重损坏传感器，所以传感器量程必须大于额定称重1Kg。我们选择的是电阻应变片压力传感器，量程为5Kg，精度为0.01%，满足本系统的精度要求。

模块的参数手册上给出满量程输出电压=激励电压\*灵敏度1.0mV/V。举例如：供电电压是5V，乘以灵敏度1.0mV/V=满量程5mV。相当于有5Kg重力产生时候产生5mV的电压。因为本次供电电压为2.5V，所以最大产生电压为2.5mV。单片机内部设置的采样参考电压为2.5V，所以设计后级放大电路时需要扩大1000倍。

1. **LM324**

LM324系列是低成本的四路运算放大器，14脚双列直插塑料封装，具有真正的差分输入外形如下图所示。单双电源均可工作。共模输入范围包括负电源。因此在众多应用中无需外部偏置元器件。输出电压范围也包括负电源电压。

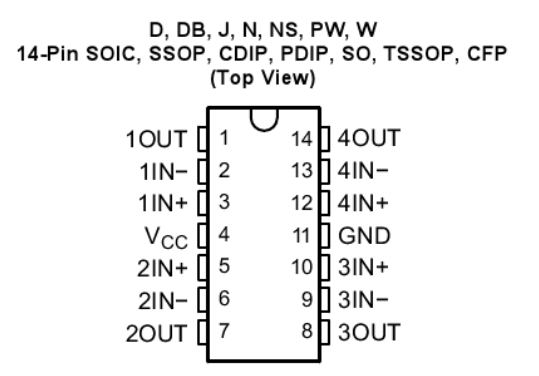
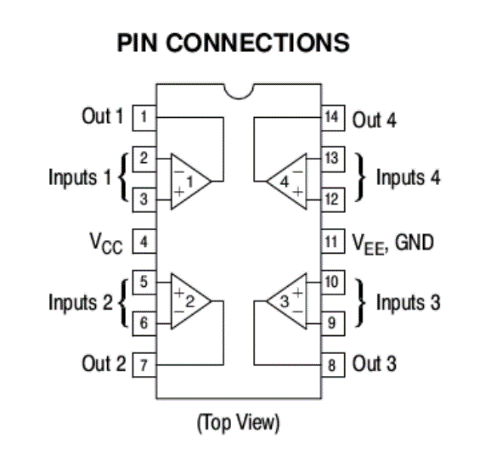


图 5‑7 LM324实物俯视图

图 5‑8 LM324内部透视图

1. **TL431**

TL431是可控精密稳压源。它的输出电压用两个电阻就可以任意的设置到从Vref（2.5V）到36V范围内的任何值。该器件的典型动态阻抗为0.2，在很多应用中用它代替稳压二极管，例如，数字电压表，运放电路，可调压电源，开关电源等。

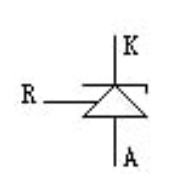
以下是分别是TL431的实物草图、软件中的元件图。

图 5‑9 元件图

图 5‑10 TL431的实物

TL431可等效为一只稳压二极管，其基本连接方法如下图所示。

下图a可作2.5V基准源，下图b作可调基准源，电阻R2和R3与输出电压的关系为U0=(1+R2/R3)2.5V

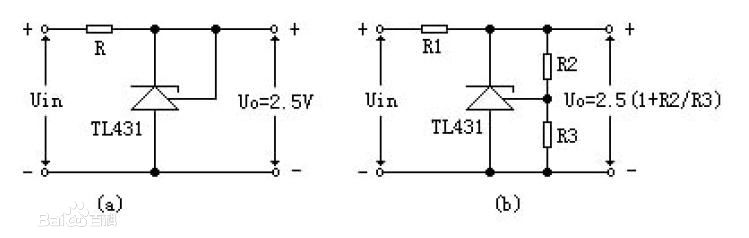


图 5‑11 TL431的使用方法

1. **AD620**

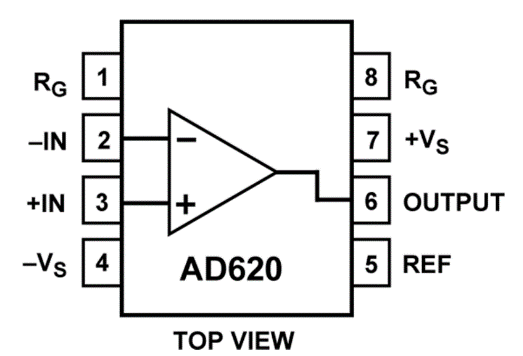


图 5‑12 AD620

如图 5‑12，AD620是一款低成本、高精度仪表放大器，内部为三运放集成的仪表放大器结构，仅需要一个外部电阻来设置增益，增益范围为1至10000。AD620具有高精度（最大非线性度40 ppm）、低失调电压（最大50 µV）和低失调漂移（最大0.6 µV/°C）特性，是电子秤和传感器接口等精密数据采集系统的理想之选。它还具有低噪声、低输入偏置电流和低功耗特性，使之非常适合ECG和无创血压监测仪等医疗应用。

AD620 的两个内部增益电阻为24. 7 kΩ , 因而增益方程式为

其中为引脚1、8的外接电阻，使用时只需要在1、8引脚之间焊接一枚滑动变阻器用来调节放大倍数即可。

## 电路设计

1. **恒压源**

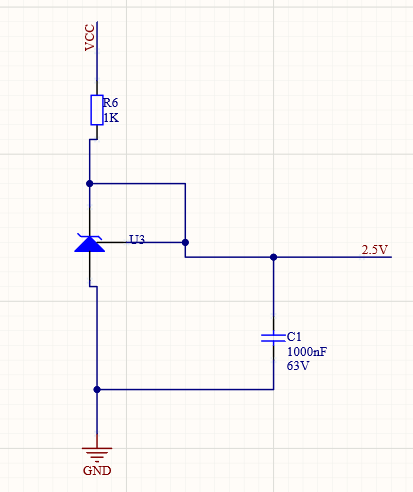


图 5‑13 恒压源、电桥模块

TL431和 R6提供恒压源Vref=2.5V, 电容C1起到稳压作用。

1. **调零电路**

电阻应变片电桥模块已有现成，无需自己制作，电路原理图只是起到示意的作用。

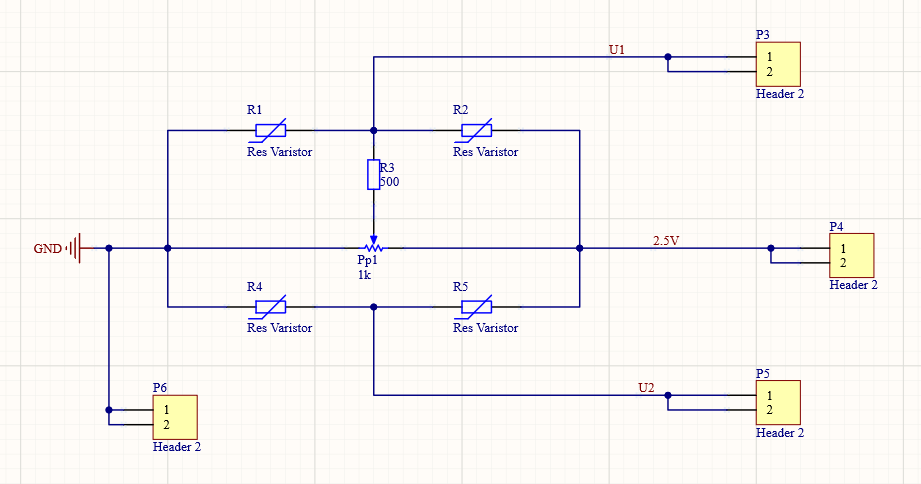


图 5‑14 电桥及调零电路

如所示，R1R2R4R5是电阻式应变片，模块内部已经集成，但是调零电路需要自己添加，电阻R3是限流电阻，起到减小调节范围的作用。下方的1KΩ滑动变阻器是真正的调零电位器，即当称重模块空载时，需要手动将输出电压调零。

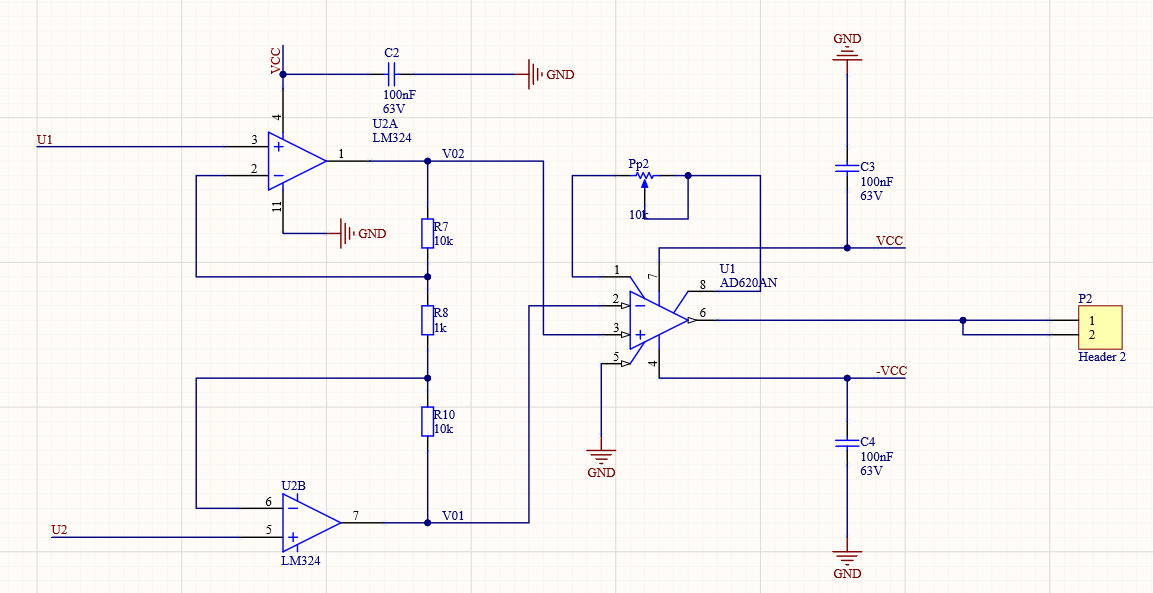
1. **放大电路**

图 5‑15 放大电路

由图 5‑15可知，R8上的电压是U1-U2，流过R7、R5的电流等于流过滑动变阻器R8的电流。

一级差动放大电路放大倍数：

因为事先约定放大倍数为1000倍，则二级仪用放大器的放大倍数：

又因为由公式5-6可知，所以1、8引脚需要接入的电阻值Rg

1. **显示、按键模块**

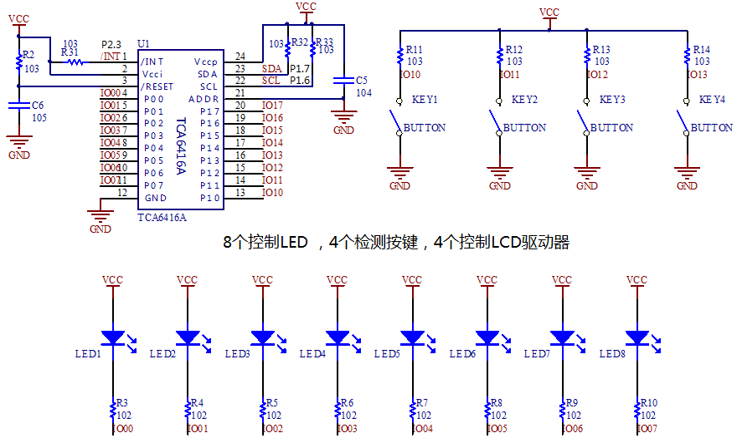


图 5‑16 按键使用

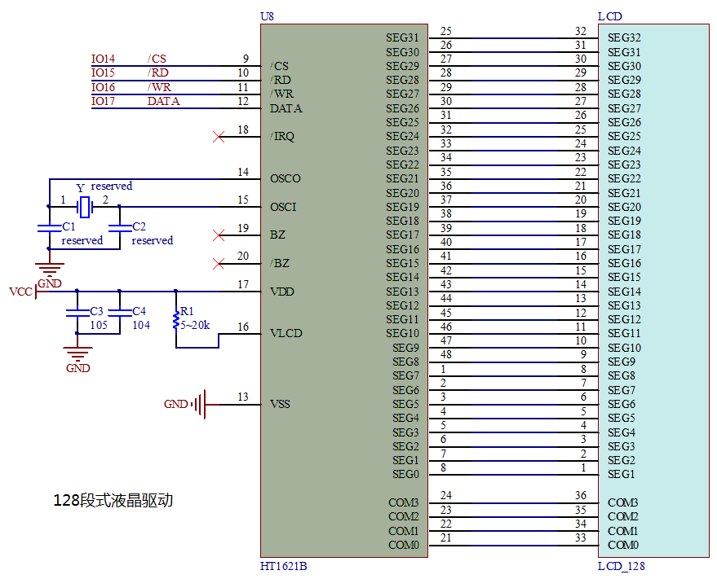


图 5‑17 128段液晶显示

此次课题用到了KEY1和KEY3，KEY1按键作为去皮功能的按键，KEY3作为取消去皮功能的按键。

在没按下去皮功能键KEY1之前，电子秤正常称重。按下KEY1键之后，开始进入去皮模式显示。按下KEY3退出去皮模式后，回到最初的称重模式。

# 软件设计

## 中断流程图





图 6‑1 看门狗定时中断

图 6‑2 ADC10中断流程图

中断子程序有ADC10中断程序和看门狗中断程序。ADC10中断就是当ADC转换完成之后，使得CPU退出低功耗模式，继续执行主函数里低功耗语句后未完成的程序。看门狗中断就是用来检查扩展板上的按键是否按下。变量key用来保存按下的按键号。

## 主函数流程图

图 6‑3 主函数流程图

如图 6‑3所示，为此次主程序的流程图，P1.0复用作为外部模拟电压的输入端口。系统初始化完成后，首先显示个人的信息。之后开始使能ADC中断，开始ADC转化，CPU进入LPM0模式。一旦ADC转化完成后，将转化后的值保存在ADC10MEN寄存器，同时CPU退出LPM0模式。接下来考虑到电压采样的不稳定，采取采样满170次求平均作为当前的电压。同时需要判断扩展版上的按键是否按下，进行相应的语句操作，最后将电压值转化为克数显示在LCD屏幕上。ADC的初始化借助Grace来进行配置。

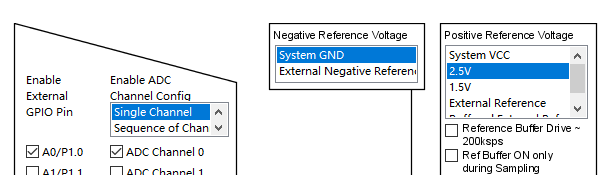


图 6‑4 通过Grace配置ADC10

ADC一共有8个外部输入，且全部与IO复用，所以首先需要使能IO的复用功能。如图 6‑4所示，我将P1.0复用作为外部模拟电压的输入，选择单通道采样，通道选择ADC Channel 0。由于外部输入电压的最大值为2.5V，在参考电压配置处选择内部基准电压2.5V。

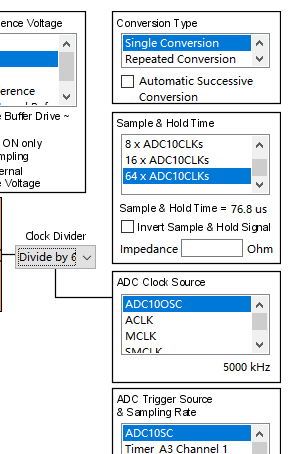


图 6‑5 ADC10相关时钟配置

选择转换模式为单次转换，“吱一声” 动一下。人工设定采样保持时间为64倍 ADC10CLK。ADC的时钟选择5MHz专用振荡器，8分频。采样起始信号选择软件写ADC10SC位，通过写ADC10SC位来开始一次AD转换。

至此，ADC10\_init()配置到此结束，写程序时只需找到将Grace工程目录下的ADC10\_init.c，复制其中的语句即可。

## 显示函数流程图



图 6‑6 显示函数流程图

主函数调用显示函数之前，已经对重量的整数部分和小数部分完成处理，将整数部分保存在变量zhenshu中，将小数部分保存在变量xiaoshu里。调用显示函数时，首先判断整数部分的数值区间，然后再执行不同的取位函数，最后将整数和小数一同显示在LCD1602上。

# 硬件调试

## 电桥调零

首先将电路板上电，调节电桥之间的滑动变阻器，使得两端的差分电压为0，如下图 7‑1所示。

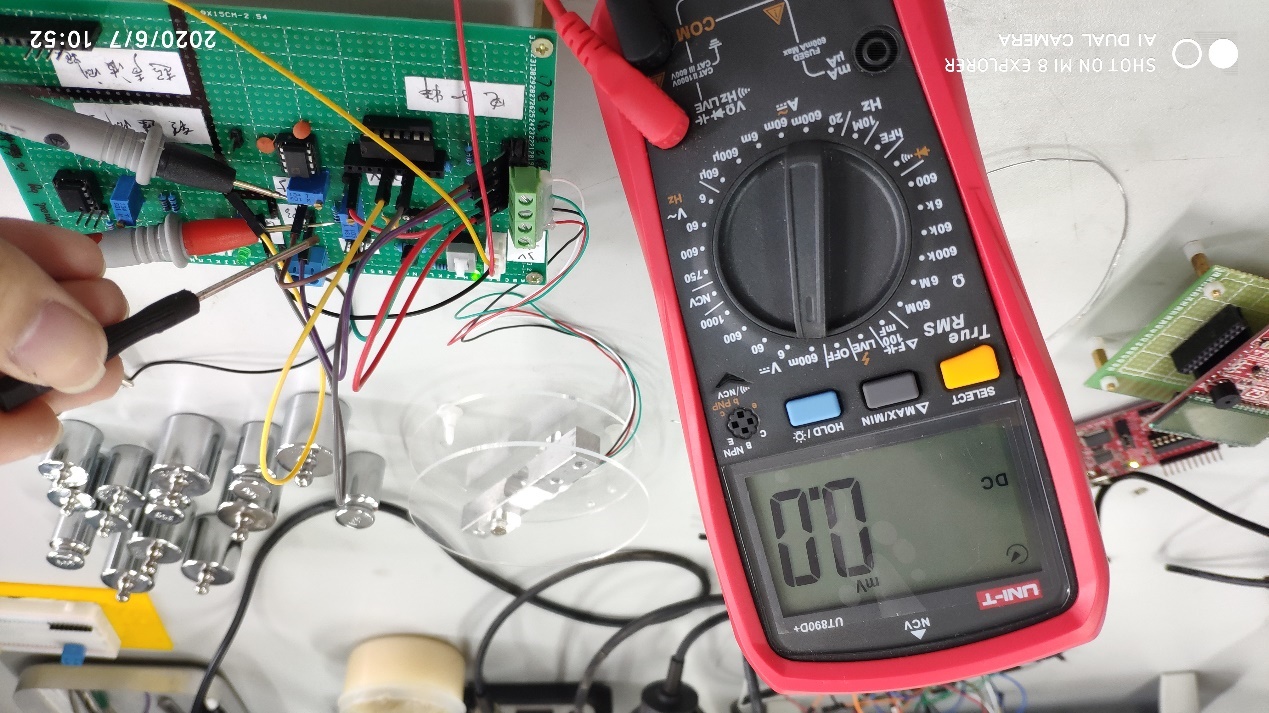


图 7‑1 电桥调零

## 实际电子秤显示砝码质量

放入不同克数的砝码，记录实际电子秤显示的质量。如图 7‑2所示

图 7‑2 实际电子秤示数

## 电压显示程序烧录验证

将外部模拟信号由精密滑动变阻器分压输入，观察LCD显示与万用表测量值是否有较大的偏差。

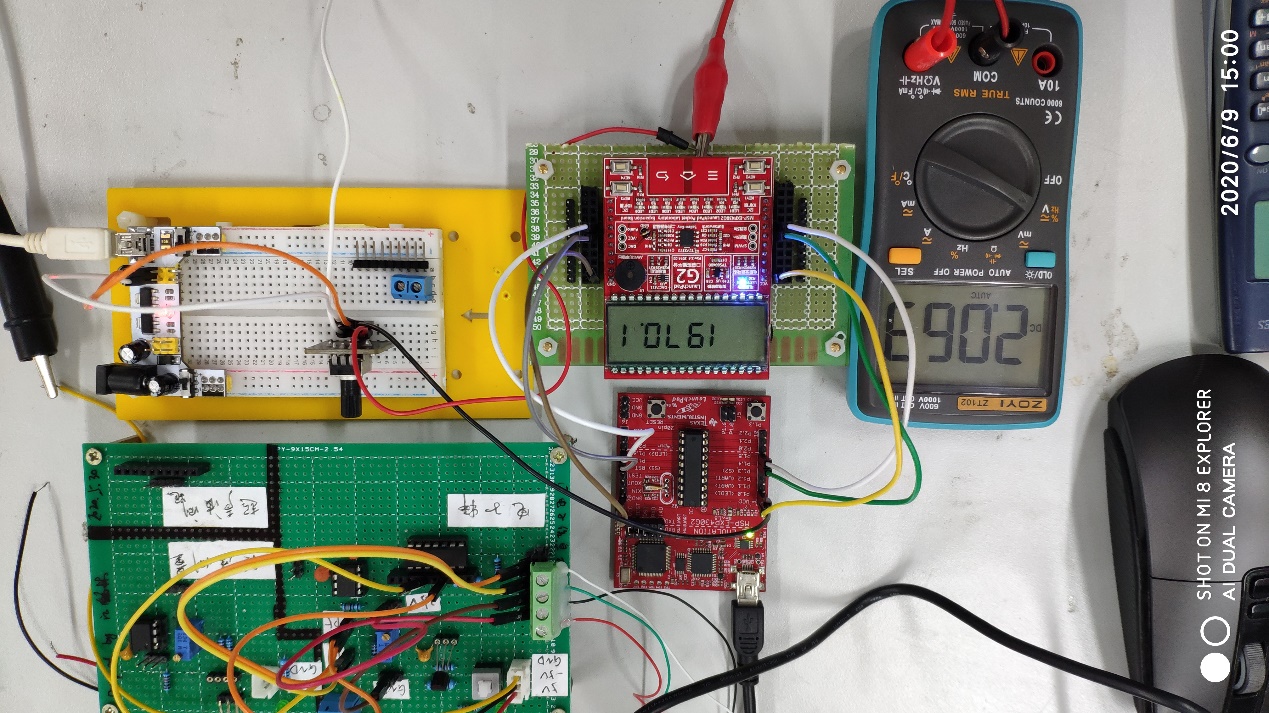


图 7‑3 程序烧录验证

放上不同的砝码克数，记录PCB输出的万用表电压值和LCD显示电压值。



图 7‑4 万用表显示与LCD显示值记录

## 多次曲线拟合和数据记录

将电压与克数转化语句写入程序后，烧录单片机记录数据，发现误差偏大的部分进行分段拟合，直至误差在要求范围内。

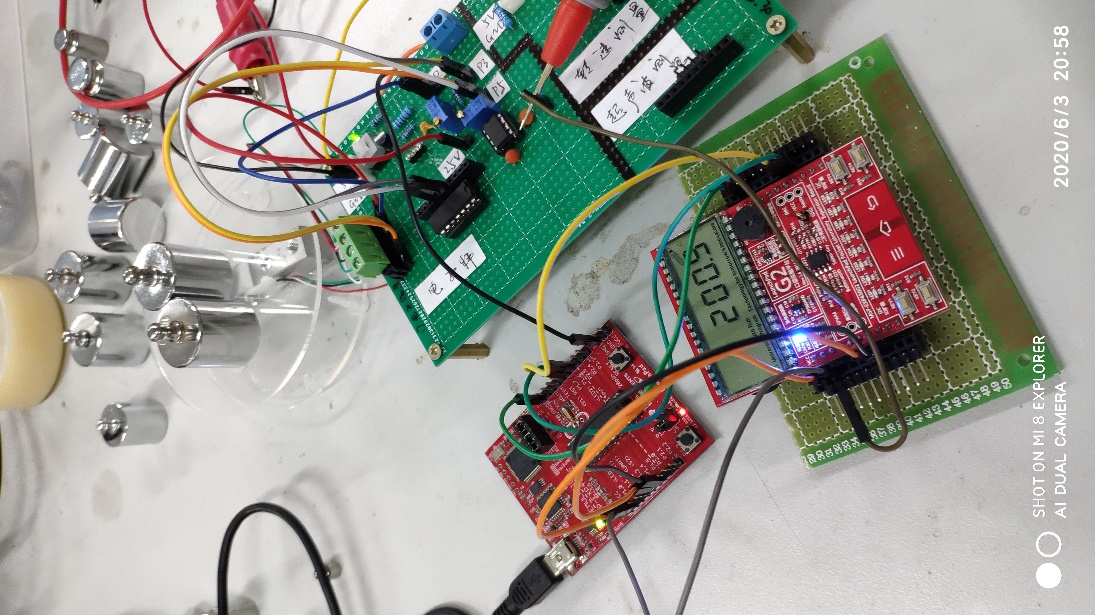


图 7‑5 200g砝码

如上图 7‑5所示，放上200g砝码，LCD显示值为200.9g。

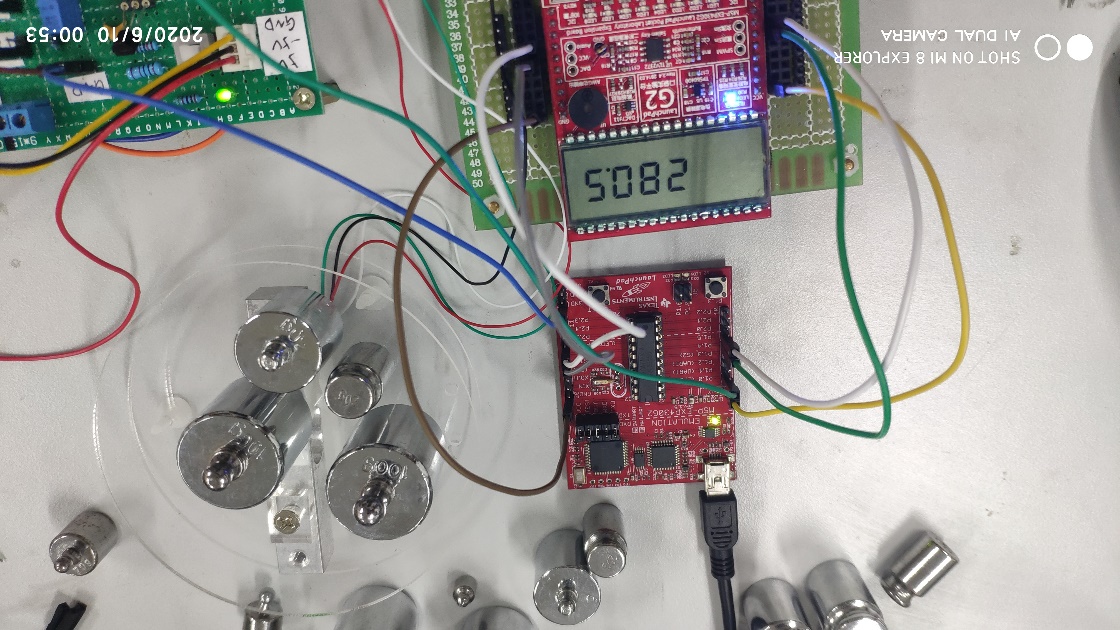


图 7‑6 280g砝码

如上图 7‑6所示，放上280g砝码，LCD显示值为280.5g。

## 数据记录

表格 7‑1 调试数据记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 砝码标称值  单位g | 电子称测量第1次  单位g | LCD128显示  单位mv | 万用表显示  单位mv | 拟合后LCD128显示  单位g | 拟合后误差 |
| 2 | 2.0 | 627.0 | 635 | 2.7 | **35.00%** |
| 4 | 4.1 | 628.2 | 636 | 4.5 | **12.50%** |
| 7 | 7.1 | 629.2 | 637 | 8.7 | **24.29%** |
| 9 | 9.1 | 629.8 | 638 | 9.2 | **2.22%** |
| 10 | 9.9 | 629.9 | 638 | 10.7 | **7.00%** |
| 12 | 12.0 | 630.5 | 639 | 12.8 | **6.67%** |
| 14 | 14.1 | 630.8 | 639 | 13.0 | **-7.14%** |
| 17 | 17.1 | 632.0 | 640 | 18.4 | **8.24%** |
| 20 | 20.0 | 632.9 | 641 | 22.6 | **13.00%** |
| 40 | 40.0 | 638.7 | 647 | 39.8 | -0.50% |
| 60 | 59.9 | 645.9 | 654 | 58.5 | -2.50% |
| 80 | 79.7 | 651.9 | 660 | 78.8 | -1.50% |
| 100 | 100.0 | 656.7 | 666 | 100.5 | 0.50% |
| 120 | 120.1 | 663.0 | 672 | 119.8 | -0.17% |
| 140 | 140.1 | 669.0 | 678 | 138.4 | -1.14% |
| 160 | 160.1 | 675.2 | 684 | 160.5 | 0.31% |
| 180 | 179.9 | 680.8 | 690 | 180.9 | 0.50% |
| 200 | 200.2 | 687.1 | 696 | 199.8 | -0.10% |
| 220 | 219.9 | 693.5 | 702 | 220.4 | 0.18% |
| 240 | 240.0 | 700.3 | 708 | 239.4 | -0.25% |
| 260 | 260.0 | 705.0 | 714 | 260.7 | 0.27% |
| 280 | 280.0 | 710.4 | 720 | 279.0 | -0.36% |
| 300 | 300.3 | 716.5 | 726 | 299.1 | -0.30% |
| 320 | 320.3 | 721.9 | 731 | 316.0 | -1.25% |
| 340 | 340.1 | 727.7 | 737 | 338.5 | -0.44% |
| 360 | 360.1 | 735.3 | 744 | 360.3 | 0.08% |
| 380 | 380.2 | 740.6 | 750 | 376.6 | -0.89% |
| 400 | 400.5 | 747.0 | 756 | 399.6 | -0.10% |
| 420 | 420.5 | 753.9 | 762 | 419.4 | -0.14% |
| 440 | 440.3 | 760.2 | 769 | 439.1 | -0.20% |
| 460 | 460.3 | 765.0 | 775 | 458.6 | -0.30% |
| 480 | 480.3 | 771.3 | 781 | 477.4 | -0.54% |
| 500 | 500.6 | 778.6 | 787 | 497.9 | -0.42% |
| 520 | 520.7 | 782.9 | 793 | 520.1 | 0.02% |
| 540 | 540.4 | 789.8 | 800 | 538.6 | -0.26% |
| 560 | 560.5 | 795.8 | 806 | 558.6 | -0.25% |
| 580 | 580.5 | 803.4 | 812 | 579.6 | -0.07% |
| 600 | 600.7 | 807.2 | 818 | 599.8 | -0.03% |
| 620 | 620.8 | 813.8 | 824 | 619.7 | -0.05% |
| 640 | 640.8 | 822.3 | 831 | 637.7 | -0.36% |
| 660 | 660.9 | 826.9 | 836 | 658.4 | -0.24% |
| 680 | 680.6 | 832.6 | 842 | 680.1 | 0.01% |
| 700 | 700.9 | 839.1 | 849 | 700.0 | 0.00% |
| 720 | 721.0 | 843.7 | 855 | 718.7 | -0.18% |
| 740 | 740.8 | 849.2 | 861 | 739.4 | -0.08% |
| 760 | 760.8 | 856.1 | 867 | 758.9 | -0.14% |
| 780 | 780.8 | 862.4 | 872 | 779.7 | -0.04% |
| 800 | 801.0 | 867.9 | 879 | 798.9 | -0.14% |
| 820 | 821.1 | 874.5 | 885 | 819.0 | -0.12% |
| 840 | 841.1 | 879.2 | 891 | 835.6 | -0.52% |
| 860 | 860.9 | 886.1 | 897 | 855.7 | -0.50% |
| 880 | 881.0 | 891.9 | 902 | 871.2 | -1.00% |
| 900 | 901.2 | 898.2 | 909 | 895.1 | -0.54% |
| 920 | 921.3 | 903.4 | 915 | 915.0 | -0.54% |
| 940 | 941.3 | 909.0 | 921 | 934.0 | -0.64% |
| 960 | 961.3 | 917.4 | 927 | 958.3 | -0.18% |
| 980 | 981.0 | 919.0 | 928 | 976.2 | -0.39% |
| 1000 | 1001.3 | 927.8 | 940 | 991.5 | -0.85% |

实验测量数据如上两表所示，数据拟合以LCD显示的电压为横轴，实际电子秤示数为纵轴，绘制二阶多项式曲线，如下**错误!未找到引用源。**所示。

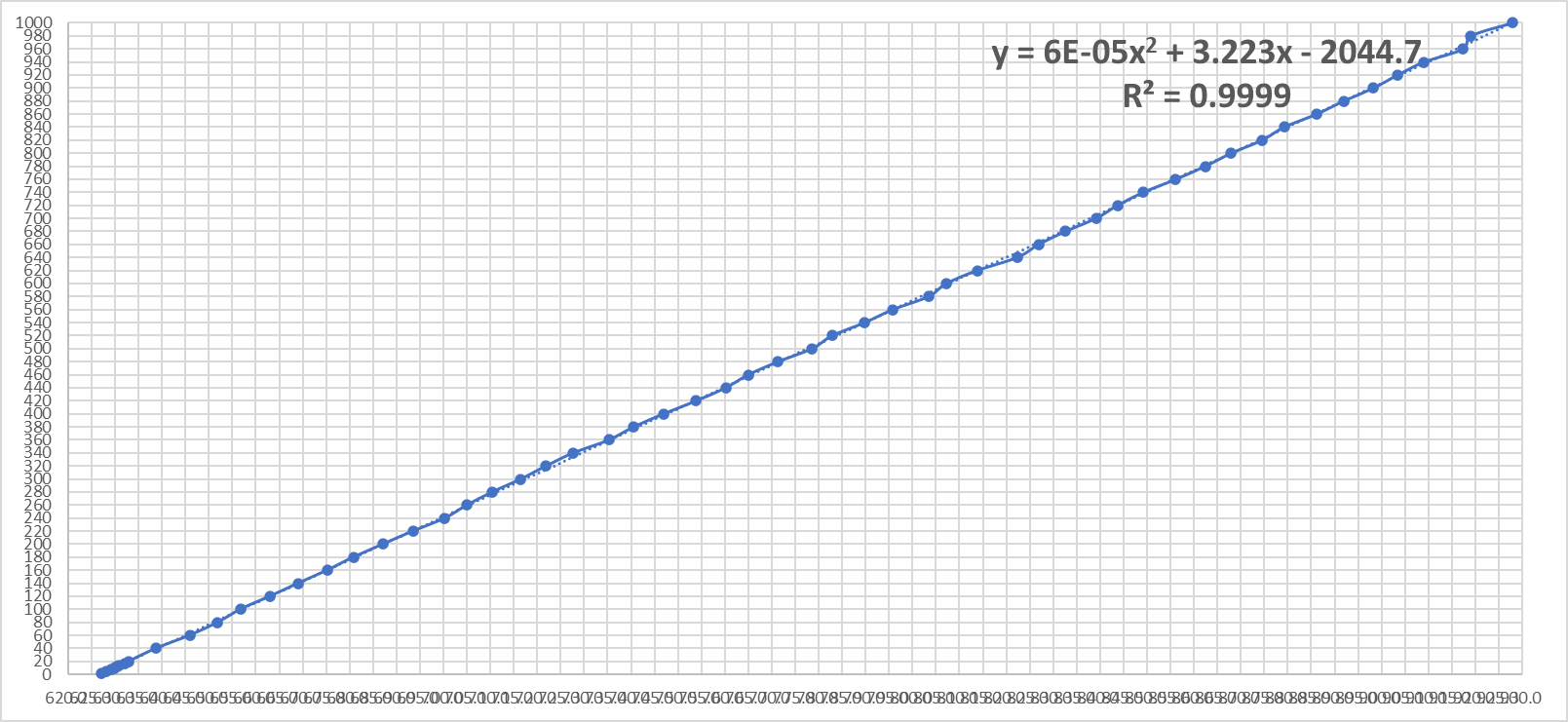


图 7‑7 拟合曲线

将数据拟合的曲线公式：

将公式7-1带入软件重新编译运行，记录第2组数据。由上表格 7‑1可以观察到，40g~1000g的误差控制在1%左右，满足要求，0g~20g的误差还是偏大。改进方案将会在之后给出。

# 设计小结

此次的电子秤的设计让我对电阻应变式传感器有了更加深刻的认识。设计之初传感器的选型很重要，需要符合设计的要求，另外还要考虑预算。我们还需要设计放大电路将应变片电桥产生的微弱电压放大，供单片机采样。

单片机采样显示的电压和实际电压之间一定会有误差，我们需要做的就是将显示值和实际电压记录下来进行拟合处理，将拟合后得到的曲线公式带入程序中进行再一次测试和数据记录。

此次的电子秤课题我只进行了一次电压拟合，结果是20g以下的显示误差特别大，我的改进方案是将0~20g的单独再进行一次曲线拟合，得到的误差应该会比之前小很多。

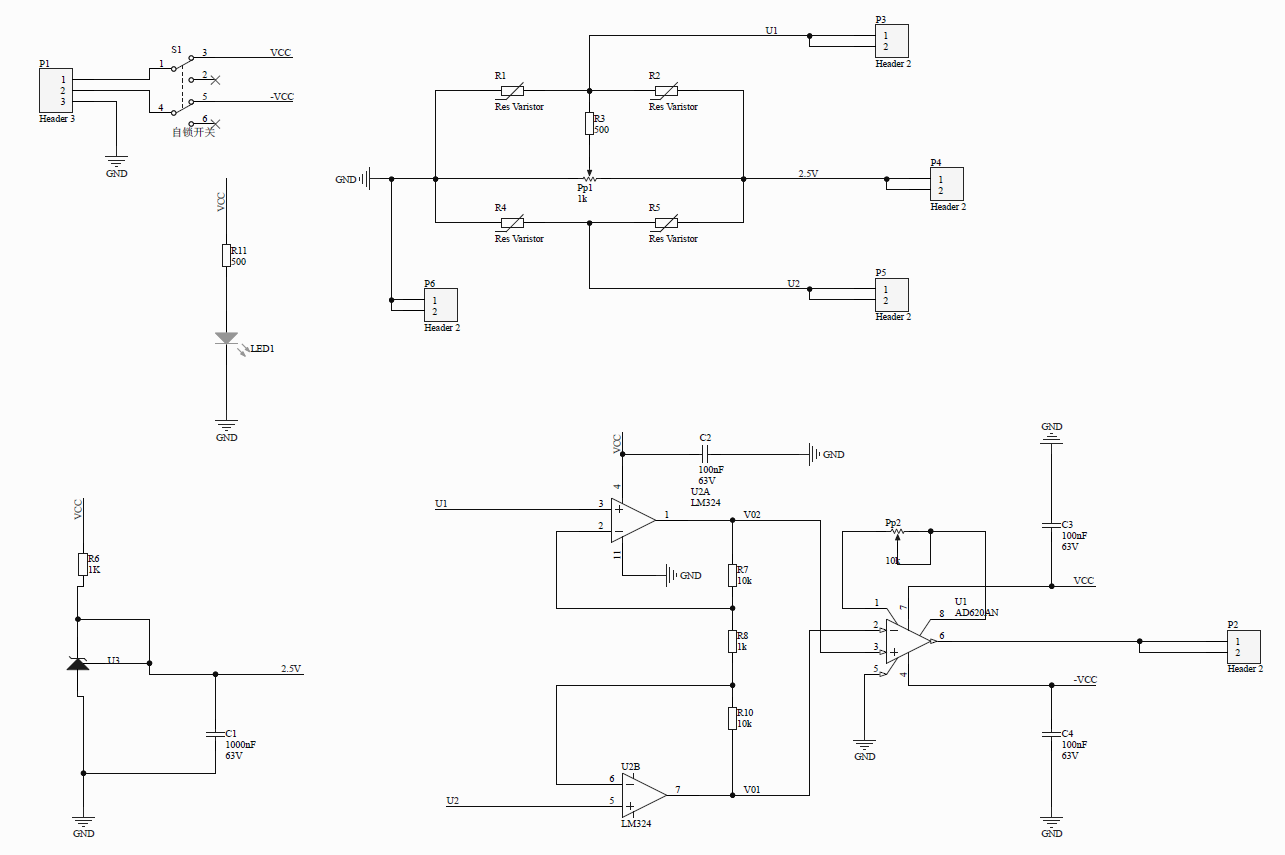
另外，由于每次采样得到的电压值有细微的波动，我采取多次采样求平均值的方法来使电压稳定。仔细想了一下，电压光取平均值还不行，我的改进思路是将多次采样得到的电压平均值用数组进行排序，剔除最大和最小值后再取平均，这样能够更加稳定。

此次的传感器的课程设计，感谢李老师和季老师的指导与帮助，不管是在硬件电路设计上还是程序设计上他们都给了我特别大的启发。

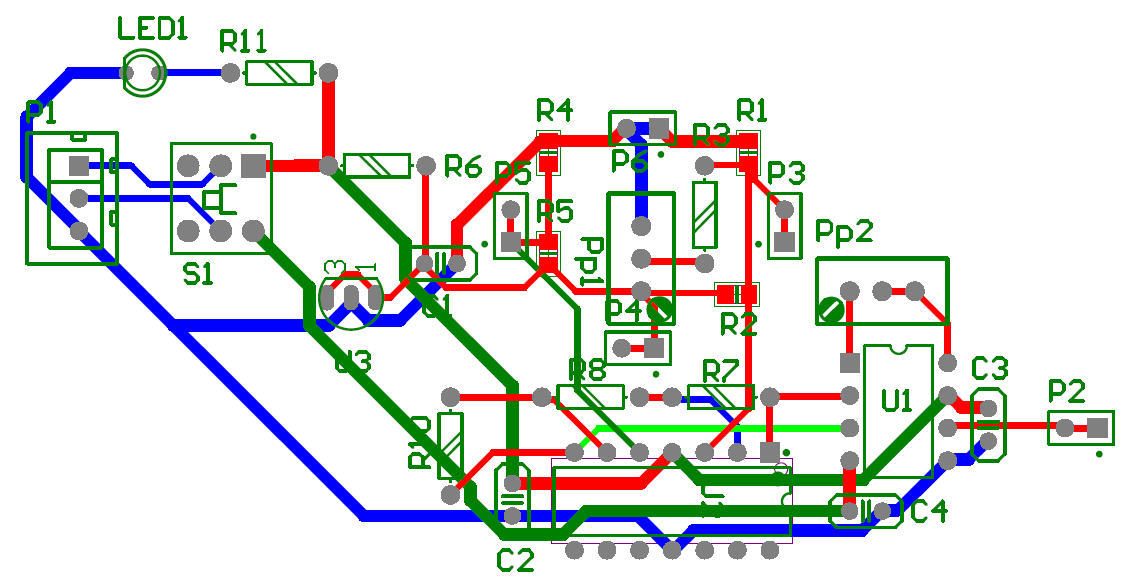
# 参考文献

[1]张彦宇.基于MSP430的高精密电子秤设计与研究[J].自动化技术与应用,2019,38(11):26-28+36.

附件1 电路原理图



附件2 PCB



附件2 源程序代码

#include "msp430g2553.h"

#include "LCD\_128.h"

#include "HT1621.h"

#include "TCA6416A.h"

int zhenshu = 0, xiaoshu = 0;

unsigned char key = 0; //去皮按键标志位

unsigned int count = 1; //去皮数据保护位

void ADC10\_init(void);

void LCD\_Init();

void LCD\_Display();

void WDT\_init();

void I2C\_IODect();

void display\_qupi();

void display\_my\_info();

int main()

{

WDTCTL = WDTPW + WDTHOLD; //关看门狗

float Vin = 0, sum = 0,mv=0,g=0;

int qupi = 0;

int i = 0;

ADC10\_init();

LCD\_Init();

WDT\_init();

display\_my\_info();

\_EINT();

while (1)

{

ADC10CTL0 |= ENC + ADC10SC; //在主循环中开启 ADC10 转换

LPM0; //休眠开总中断，等待 ADC10 转换完成后，进入中断运行唤醒 CPU 代码

//-----ADC转换完成中断唤醒CPU后才执行以下代码-----

Vin = (ADC10MEM \* 2.5) / 1023;

sum = Vin + sum;

i++;

if (i == 170)

{

Vin = sum / 170.0;

mv = Vin \* 1000;

if (key == 1) //去皮功能语句块

{

if (count == 1)

{

display\_qupi();

\_\_delay\_cycles(10000);

qupi = 0.00006\*mv\*mv+3.223\*mv-2044.7; //记录产品包装的质量

count = 0; //去皮质量第一次记录后就保护起来的标志

}

g = 0.00006\*mv\*mv+3.223\*mv-2044.7-qupi;

}

else

g = 0.00006\*mv\*mv+3.223\*mv-2044.7;

zhenshu = (int)g;

xiaoshu = (g - zhenshu) \* 10;

LCD\_Display();

i = 0;

sum = 0;

}

}

}

void display\_my\_info()

{

LCD\_DisplaySeg(4);

LCD\_DisplaySeg(9);

LCD\_DisplaySeg(10); //显示J

LCD\_DisplayDigit(8, 2);

LCD\_ClearSeg(12);

LCD\_ClearSeg(19); //显示H

LCD\_DisplayDigit(0, 3);

LCD\_ClearSeg(25);

LCD\_ClearSeg(26);

LCD\_ClearSeg(27); //显示L

LCD\_DisplayDigit(1, 4);

LCD\_DisplayDigit(0, 5);

LCD\_DisplayDigit(7, 6);

HT1621\_Reflash(LCD\_Buffer);

\_\_delay\_cycles(1000000);

LCD\_Clear();

LCD\_DisplayDigit(0, 5);

LCD\_DisplayDigit(0, 6);

LCD\_DisplaySeg(\_LCD\_DOT4);

HT1621\_Reflash(LCD\_Buffer);

}

#pragma vector = WDT\_VECTOR

\_\_interrupt void watchdog\_timer(void)

{

PinIN();

I2C\_IODect();

LPM0\_EXIT;

}

void I2C\_IODect() //检测事件确实发生了

{

static unsigned char keyEY\_Past = 0, keyEY\_Now = 0;

keyEY\_Past = keyEY\_Now;

//----判断I2C\_IO10所连的keyEY1按键是否被按下------

if ((TCA6416A\_InputBuffer & BIT8) == BIT8)

keyEY\_Now |= BIT0;

else

keyEY\_Now &= ~BIT0;

if (((keyEY\_Past & BIT0) == BIT0) && (keyEY\_Now & BIT0) != BIT0)

{

PinOUT(0, 0); //LED1亮，LED234灭

PinOUT(1, 0);

PinOUT(2, 0);

PinOUT(3, 0);

PinOUT(4, 0);

PinOUT(5, 0);

PinOUT(6, 0);

PinOUT(7, 0);

PinOUT(8, 0);

key = 1;

}

//----判断I2C\_IO12所连的keyEY3按键是否被按下------

if ((TCA6416A\_InputBuffer & BITA) == BITA)

keyEY\_Now |= BIT2;

else

keyEY\_Now &= ~BIT2;

if (((keyEY\_Past & BIT2) == BIT2) && (keyEY\_Now & BIT2) == 0)

{

PinOUT(0, 1);

PinOUT(1, 1);

PinOUT(2, 1);

PinOUT(3, 1);

PinOUT(4, 1);

PinOUT(5, 1);

PinOUT(6, 1);

PinOUT(7, 1);

PinOUT(8, 1);

count = 1;

key = 3;

}

}

void display\_qupi()

{

LCD\_Clear();

LCD\_DisplayDigit(9, 3);

LCD\_ClearSeg(20);

LCD\_DisplayDigit(0, 4);

LCD\_ClearSeg(55);

LCD\_DisplayDigit(8, 5);

LCD\_ClearSeg(56);

LCD\_ClearSeg(61);

LCD\_DisplayDigit(1, 6);

HT1621\_Reflash(LCD\_Buffer);

}

void WDT\_init()

{

//-----设定WDT为-----------

WDTCTL = WDT\_ADLY\_16;

//-----WDT中断使能----------------------

IE1 |= WDTIE;

}

void LCD\_Init()

{

TCA6416A\_Init();

HT1621\_init();

//相关硬件的初始化，其中 I2C 模块的初始化由 TCA6416A 初始化函数在内部完成了， LCD\_128 库函数由 HT1621 初始化函数在内部引用了

//-----显示固定不变的LCD段-----

}

void LCD\_Display()

{

LCD\_DisplaySeg(\_LCD\_DOT4);

LCD\_DisplayDigit(LCD\_DIGIT\_CLEAR, 2);

LCD\_DisplayDigit(LCD\_DIGIT\_CLEAR, 3);

LCD\_DisplayDigit(LCD\_DIGIT\_CLEAR, 4);

LCD\_DisplayDigit(LCD\_DIGIT\_CLEAR, 5);

LCD\_DisplayDigit(LCD\_DIGIT\_CLEAR, 6);

//-----根据zhenshu拆分并显示数字-----

if (zhenshu > 999)

{

LCD\_DisplayDigit(zhenshu / 1000, 2);

LCD\_DisplayDigit((zhenshu / 100) % 10, 3);

LCD\_DisplayDigit((zhenshu / 10) % 10, 4);

LCD\_DisplayDigit(zhenshu % 10, 5);

LCD\_DisplayDigit(xiaoshu, 6);

}

else if (zhenshu > 99) //100~999（3位）

{

LCD\_DisplayDigit(zhenshu / 100, 3);

LCD\_DisplayDigit((zhenshu / 10) % 10, 4);

LCD\_DisplayDigit(zhenshu % 10, 5);

LCD\_DisplayDigit(xiaoshu, 6);

}

else if (zhenshu > 9) //2位

{

LCD\_DisplayDigit(zhenshu / 10, 4);

LCD\_DisplayDigit(zhenshu % 10, 5);

LCD\_DisplayDigit(xiaoshu, 6);

}

else if (zhenshu > 1) //1位

{

LCD\_DisplayDigit(zhenshu, 5);

LCD\_DisplayDigit(xiaoshu, 6);

}

else

{

LCD\_DisplayDigit(0, 5);

LCD\_DisplayDigit(xiaoshu, 6);

}

HT1621\_Reflash(LCD\_Buffer); //-----更新缓存，真正显示-----

}

void ADC10\_init(void)

{

ADC10CTL0 &= ~ENC;

ADC10CTL0 = ADC10IE | ADC10ON | REFON | REF2\_5V | ADC10SHT\_0 | SREF\_1;

//启用内部2.5V参考电压，采样保持时间为64 x ADC10CLKs

ADC10CTL1 = CONSEQ\_0 | ADC10SSEL\_0 | ADC10DIV\_3 | SHS\_0 | INCH\_0;

//ADC时钟分频为7分频

ADC10AE0 = 0x1; //P1.0为ADC输入端

\_\_delay\_cycles(20000);

//ADC10CTL0 |= ENC;

}

#pragma vector = ADC10\_VECTOR

\_\_interrupt void ADC10\_ISR(void)

{

LPM0\_EXIT;

}