

重庆三峡学院

毕业设计（论文）

设计（论文）题目 便携式智能电子称的设计

院 系 电子与信息工程学院

专 业 电气工程及其自动化

年 级 2014 级电气一班

学生学号 2013240012

学生姓名 刘晓桃

指导教师 聂文亮 职称 助教

重庆三峡学院教务处 制

目录

摘要	I
Abstract.....	II
1 绪论	1
1.1 选题的背景及意义	1
1.2 本次设计需要完成的内容	1
2 总设计原理及方案	3
2.1 智能电子秤的设计要求	3
2.2 智能电子秤的工作原理	3
2.3 智能电子秤原理框图	3
3 硬件电路设计	5
3.1 硬件控制电路模块	5
3.2 单片机模块	5
3.3 模数 (A/D) 转换模块	6
3.4 称重传感器模块	8
3.4.1 电阻应变片的结构及工作原理	8
3.4.2 测量转换电路	9
3.5 LCD 显示模块	10
3.6 超重报警模块	11
3.7 复位电路	11
3.8 按键处理模块	12
4 软件设计	14
4.1 软件开发环境	14
4.2 主程序流程图	14
4.3 显示部分流程图	15
4.4 系统按键部分流程图	15
5 系统调试与调试结果	17
5.1 硬件电路的制作	17
5.2 硬件电路的调试和结果	17
5.2.1 硬件电路的调试	17
5.2.2 硬件电路的调试结果	18
设计总结.....	19
参考文献.....	20
致谢	21
附录	22
附录一 智能电子秤原理图.....	22
附录二 源程序.....	23

便携式智能电子秤设计

刘晓桃

重庆三峡学院电子与信息工程学院电气工程及其自动化专业 2014 级 重庆万州 404100

摘要：秤是我们生活中必不可少的称量物品重量的工具。本文设计了以 STC89C52 单片机为控制核心的便携式智能电子秤，它主要由 AT89C52 单片机、压力传感器、HX711 模数转换器、LCD1602 显示器四大部分组成，利用压力应变片等元件组成的称重传感器将被称量物品的质量转换为可测量的电压信号，电压信号经放大电路放大，利用模数（A/D）转换芯片将模拟信号转换为数字信号传送到单片机中进行处理后，得出当前被称量物品的重量。

关键词：AT89C52，压力传感器，模数转换器，液晶显示

Design of Portable intelligent electronic scale

Liu xiao-tao

Chongqing Three Gorges University School of Electronics and Information Engineering Electrical
Engineering and Automation Professional 2014 Chongqing Wanzhou 404100

Abstract: Scale is an indispensable tool for weighing the weight of things in our life. This article has designed a portable intelligent electronic scale with STC89C52 as its control core. It is mainly composed of AT89C52 MCU, pressure sensor, HX711 analog-to-digital converter and 1602 liquid crystal display. It uses a load cell which made up of elements such as pressure strain gauges to convert the mass of the weighed item into a measurable voltage signal. the voltage signal is amplified by the amplifying circuit, then the analog-to-digital conversion chip is used to simulate the voltage signal. the signal is converted to a digital signal, after that sent digital signal to the MCU for processing. finally, the weight of the currently weighed item is obtained.

Key words: AT89C52 , pressure sensor, analog to digital converter, liquid crystal display

1 绪论

秤是具有悠久历史的量重工具，在我们的日常生活中、实验研究中、产品生产中乃至国际贸易中很多时候都需要用秤来称量物品的重量，可以说称是我们生活中必不可少的称量器具。传统的纯机械称量工具受制作工艺误差和校准误差等因素的影响，称量误差相对较大、耗时较长、功能少、携带不便、显示不够直观、操作相对复杂。近年来，随着社会的不断发展、科学技术的飞速进步，计算机技术、数字技术、电子技术日益广泛地被应用于人们生活中的方方面面。电子秤与机械式称量工具相比具有功能更多、更精确、更智能、更便捷、更可靠、更稳定、显示更直观等优点，因此电子秤受到人们的广泛认可并被快速推广到人们的生活中。本次设计主要研究的是利用 STC89C52 型单片机为核心控制单元设计、制作一台便携式智能电子秤。

1.1 选题的背景及意义

随着社会经济的飞速发展，交易商品的种类及数量与日俱增，人们早已不满足于使用传统的机械式称量工具来称量物品，因而借助科学技术的发展研发出性能更好使用更方便的称量工具。我国的电子秤的研发是从上世纪 60 年代开始的，最初的电子秤是以机电结合的方式实现称量功能的，后来经过数十年的不断研发和改进，伴随着计算机技术和电子技术的快速发展，电子秤朝着数字化、全电子化、智能化、多功能化、高精度的方向发展，电子秤的称量方式逐渐由模拟测量方式发展为数字测量方式。由于电子式称量工具相较于传统称量工具具有巨大优势，因此传统的机械式称量工具逐步快速地被电子型称量工具取代。在过去几十年，我国在电子量重器具的研发方面有着快速地发展，但就目前而言，我国电子衡器产业的发展水平距世界一流水平还有着较大的距离，这其中的差距体现在研发能力较弱、制造技术和工艺相对落后、生产设备和测试仪表老化、产品的工作稳定性较差、功能不够多、环境适应能力较弱、产品的种类不够多、智能化程度不高、测量精度不够高、工作可靠性较差等。随着经济的快速发展，商品贸易的数量剧增、我们对称量工具要求操作方便、易于识别、便于携带、显示直观、称量方便、使用简单、精度较高、智能化程度高。随着电子技术和数字技术的发展，电子型称量工具以其精确度更高、智能化程度高、使用起来更方便快捷、性能可靠、运行稳定、环境适应能力强等优点被广泛地应用于人们的生活中。它不仅极大地方便了人们的生产、生活等方方面面，也极大的提高了人们称量物品的效率和商品交易的效率，同时也促进了了社会经济的发展。

1.2 本次设计需要完成的内容

根据设计要求，此次便携式智能电子秤的设计需要完成的内容有：首先，明确设计要求，根据设计要求在网上和相关书中籍查找、收集相关资料，清楚地了解便携式智能电子秤工作的基本原理，利用所学知识，结合书本和网上收集到的相关资料，理清设计思路，搭建设计框架，设计出智能电子秤的原理图，并利用 Altium Designer 软件绘制出来。然后确定元器件类型及数量，根据便携式智能电子秤实现各部分功能的需求，查询相关资料，确定所需元器件，列出电子元器件清单，在电商网站上购买相关元器件并根据便携式智能电子秤各部

分的功能，利用所学知识，参考相关资料利用 keil uvision4 软件完成系统软件的编写、修正、调试和烧录。最后，根据设计好的原理图，将电子元器件合理排布于覆铜板上并进行焊接，并利用万用表检测各元器件功能是否正常，各线路是否有不符合原理图的短路、开路，对有错误的地方进行分析并改正。完成对电子秤的软、硬件系统检测无误后，利用电池电源对硬件电路进行供电，进行硬件、软件联合调试，在调试的过程中去发现问题、分析问题出现的原因并且解决问题，直至智能电子秤各部分正常工作且符合设计要求。

2 总设计原理及方案

2.1 智能电子秤的设计要求

本次设计是以 STC89C52 单片机为核心控制单元的便携式智能电子秤。根据设计要求，本设计需满足在量程范围和一定精度范围内称量物品重量的功能，整个智能电子秤系统分为单片机部分、称重传感器部分、模数转换部分、数值显示部分、独立按键部分、超重警报部分。本次设计实现的主要功能如下：

- 1) 电子秤的量程范围为 0~10kg；
- 2) 微调功能；
- 3) 去皮功能；
- 4) 测量精度为 0.1%，测量误差为 $\pm 0.010\text{kg}$ ；
- 5) 超出量程最大值 10kg 时声、光报警功能；
- 6) 重量显示功能。

2.2 智能电子秤的工作原理

根据本次设计的要求作如下设计：本次设计以 AT89C52 单片机为整个系统的核心控制单元，按下电源开关，然后通过独立按键设置电子秤的量程（0~10kg），当被称量物品放置于电子秤称量台上时，被称量物品的重量作用于称重传感器内双孔悬臂梁使之发生形变，贴附在悬臂梁的上压力应变片也随之发生形变，从而产生压—电效应。利用这一原理，称重传感器便可将被称量物品的重量转换为可以用相关仪器测量的电压变化量，电压信号进入到放大电路，经过放大、滤波后输入到模数（A/D）转换器，经模数（A/D）转换器将其从模拟信号转换为数字信号并送入单片机进行控制译码，最后再通过液晶显示器显示将被称量物品的重量以数字的形式显示出来。此外，当被称量物品的质量在量程范围（0~10kg）内时，智能电子秤正常工作，超重警报系统不工作，即蜂鸣器不响，发光二极管不闪烁；当被称量物品的重量超过电子秤量程最大值 10kg 时，超重警报系统工作，即蜂鸣器发出“滴”声，随之发光二极管闪烁。

2.3 智能电子秤原理框图

便携式智能电子秤的基本原理是：物品放置于称台上时，称重传感器将物品的重量转换为可以被相关仪器测量的电压或电流的变化量，电压或电流经放大电路放大后，HX711 模数（A/D）将其从模拟量转换为数字量然后输入到单片机进行处理，若物品的重量大于电子秤的最大量程 10kg，单片机便发出指令，使超重报警系统发出警报；若物品的重量在电子秤的量程范围（0~10kg）内，则 LCD 液晶显示器显示物品重量。根据上述智能电子秤的大致原理，结合所学知识并查阅相关资料，设计出如下列框图所示的方案。

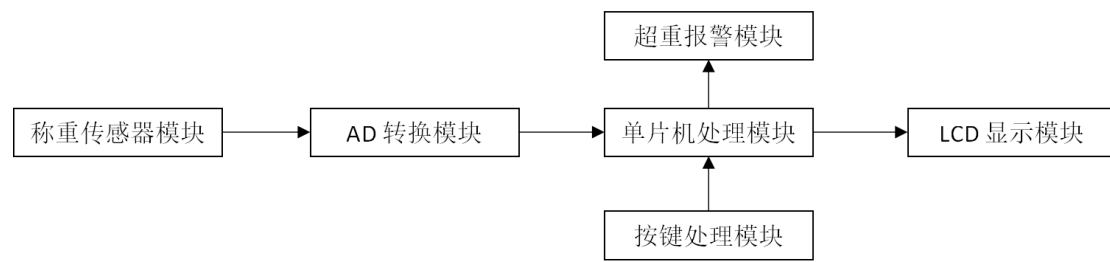


图 2-1 智能电子秤原理图

3 硬件电路设计

3.1 硬件控制电路模块

本次设计的便携式智能电子秤共有称重传感器、AD 转换、单片机处理、超重报警、按键处理、LCD 显示六大模块。

3.2 单片机模块

根据设计需求，选择 AT89C52 单片机作为此次设计的核心控制单元。AT89C52 单片机是美国爱特梅尔公司生产的单片机，在生产该单片机的过程中使用了 CMOS 工艺以及高密度和高速技术，因此，AT89C52 单片机具有功耗较低的特性。AT89C52 单片机具有外部中断、定时器、时钟输出以及计数器等诸多功能。

（一）AT89C52 单片机组成部分：

- 1、8 位中央处理单元。
- 2、512 字节的内部数据存储器
- 3、8k 片内程序存储器。
- 4、3 个 16 位定时/计数器。
- 5、32 个双向输入/输出（I/O）口。
- 6、5 个两级中断结构。
- 7、1 个全双工串行通信口。
- 8、时钟振荡电路。

（二）AT89C52 单片机引脚如图 3-1 所示：

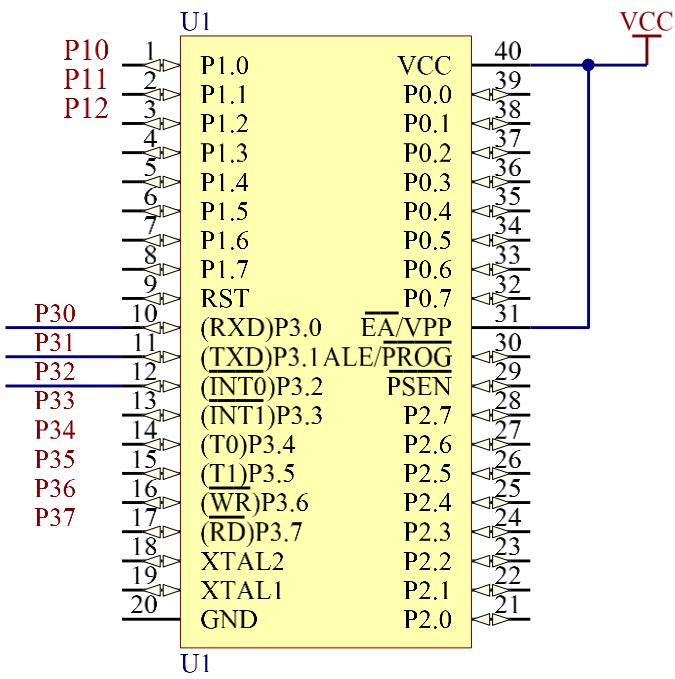


图 3-1 AT89C52 单片机引脚图

表 3-1 AT89C52 单片机引脚功能表

引脚符号	引脚功能说明
VCC	主电源输入引脚，接 + 5V 直流电源
GND	接地端
RST	复位输入引脚，保持两个机器周期的高电平时动作
P0 口	共有 8 位双向 I/O 口，可作为低 8 位地址/数据复用
P1 口	共有 8 位双向 I/O 口。作为输入口使用时，对 pP1 口写入“1”
P2 口	8 位双向 I/O 口，其内部接有上拉电阻。可作为高 8 位地址/数据复用
P3 口	共有 8 位双向 I/O 口，其内部有上拉电阻。此端口可用作特殊功能
XTAL1	反向振荡放大器的输入及内部时钟工作电路的输入端
XTAL2	反向振荡输出端
PSEN	外部程序存储选通信号输出端
EA/VPP	外部程序访问允许端

3.3 模数（A/D）转换模块

A/D 转换模块的作用是把模拟信号转换成数字信号。A/D 转换的性能指标有：分辨率、转换速度、量化误差、偏移误差、满刻度误差、线性度。根据设计需要，本次设计中，电子秤的模数转换模块选用的是 HX711 这款 A/D 转换器。HX711 是一款具有较高精度的 24 位模数转换器芯片，HX711 芯片的组成部分有：稳压电源、外围电路以及片内时钟振荡器等，其具有较高的可靠性、响应速度较快、集成程度较高、抗干扰能力较强、功耗很低、性能稳定性强等特点，而且价格比较便宜。HX711 模数转换芯片使用起来非常方便，且编程简单。工作时控制信号由管脚驱动，由于其 A/D 转换器以及外部传感器可以直接由其内部的稳压电源供电，因此使用时不需要再外接模拟电源。

（一）HX711 芯片引脚图：

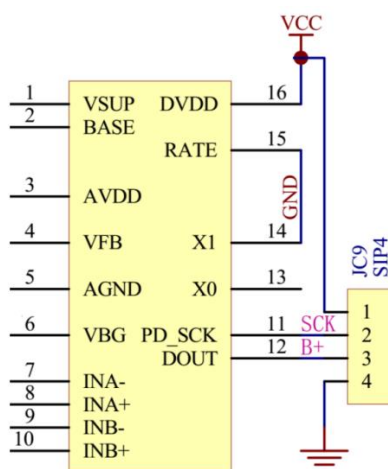


图 3-2 HX711 引脚功能图

（二）HX711 芯片应用电路如下图 3-3 所示。

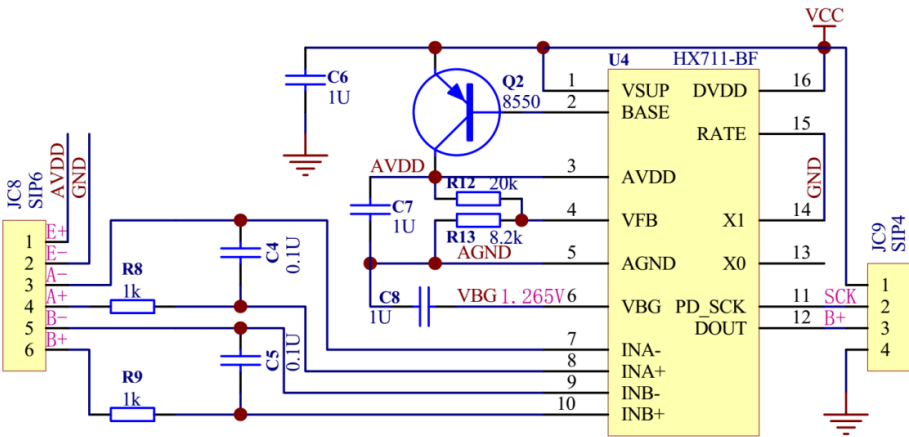


图 3-3 HX711 应用电路图

如上图 3-2 所示，HX711AD 转换芯片的工作原理是：AD 转换芯片的外部电路将从称重传感器直流电桥输出的模拟信号进行滤波，再由 AD 转换芯片的模拟输入通道将模拟信号进行 128 倍的增益，然后采样为 24bit 的数字信号输入到单片机。

表 3-2 HX711AD 转换器引脚功能表

引脚标号	引脚符号	引脚功能说明
1	VSUP	稳压电路供电电源端口 稳压供电电压为 2.6 ~ 5.5V
2	BASE	模拟输出端口 用作稳压电路控制输出
3	AVDD	模拟电源输入端，输入电压为：2.6 ~ 5.5V
4	VFB	模拟输入端 稳压电路控制输入（不用稳压电路时应接地）
5	AGND	模拟地端
6	VBG	模拟输出端 参考电源输出
7	INNA	模拟输入端 通道 A 负输入端
8	INPA	模拟输入端、通道 A 正输入端
9	INNB	模拟输入端、通道 B 负输入端
10	INPB	模拟输入、通道 B 正输入端
11	PD-SCK	数字输入断电控制（高电平有效）和串口时钟输入
12	DOUT	数字输出、 串口数据输出
13	X0	数字输入、输出，晶振输入
14	XI	数字输入、外部时钟或晶振输入端
15	RATE	数字输入、输出数据速率控制端
16	DVDD	电源，供电标准为：2.6 ~ 5.5V

HX711AD 转换芯片输出信号的计算：

本次设计的智能电子秤量程为 0~10kg，灵敏度为 1.0mV/V，称重传感器的供电电压为 4.3V，称重传感器称重恰好为量程最大值 10kg 时的输出电压（满量程电压）为：

$$V_{\text{full-scale}} = V_s * K_s \tag{3-1}$$

如式子 3-1 所示， $V_{\text{full-scale}}$ 为称重传感器的满量程电压， V_s 为称重传感器的供电电压， K_s 为称重传感器的灵敏度系数。满量程电压 $V_{\text{full-scale}}$ ：

$$4.3\text{mV} * 1.0\text{mV} / V = 4.3\text{mV} \quad (3-2)$$

称重传感器输出的电压模拟信号输入到 HX711AD 转换芯片，HX711 内部的增益通道将这一放大，然后采样为 24bit 的数字信号输入到单片机。

HX711 对模拟信号的增益倍数为 128 倍，增益后的电压大小为：

$$4.3\text{mV} * 128 = 550.4\text{mV} \quad (3-3)$$

满量程情况下，HX711 模数转换（采样）后输出的数值为：

$$0.554 * 2^{24} / 4.3 \approx 2147483 \quad (3-4)$$

假设被称量物品的重量为 Wkg，称量该物品时 AD 转换器器输出的电压为 X，则称重传感器输出的电压为：

$$W\text{kg} * 4.3\text{mV} / 10\text{kg} = 0.43W\text{mV} \quad (3-5)$$

经过 AD 转换后输出的数字信号大小为：

$$0.43W\text{mV} * 128 * 2^{24} / 4.3 \approx 21478.65W \quad (3-6)$$

$$X = 21478.65 / 100 = 2147.4865W \quad (3-7)$$

$$W = X / 2147.4865\text{kg} \approx 2.15\text{g} \quad (3-8)$$

3.4 称重传感器模块

称重传感器的作用是将非电量转换成电量，通过这一元件可将被称量物品的重量转换成电压或电流量这种便于单片机或计算机处理、计算、求值的物理量。本设计采用的是电阻应变式称重传感器。此次设计对称重传感器性能的基本要求是：当被称量物品的重量在量程范围内时，称重传感器的输出的电压变化量要与被称量物品的重量保持较好的线性关系，而且能保持较好的灵敏度、较好的环境适应能力、稳定性以及可靠性。

本次设计选择量程为 0~10kg 的称重传感器作为收集被称量物品重量数据的测量单元，其组成部分为：双孔悬臂梁（弹性体）、电阻应变片、电缆线、补偿电路、惠斯通电路、密封材料以及胶粘剂等。

3.4.1 电阻应变片的结构及工作原理

电阻应变片的结构如下图 3-3 所示。

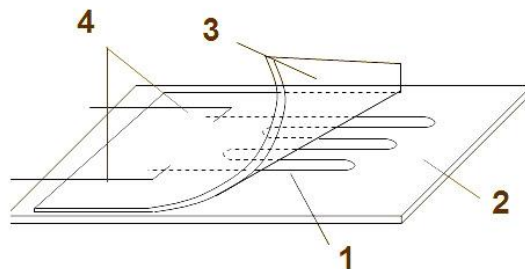


图 3-3 电阻应变片结构图

1—敏感栅：其组成材料是合金金属丝或合金金属箔片。

2—基底：其作用是保持敏感栅的几何形状以及相对位置。

3—盖层：其作用是保护敏感栅，通常由纸、玻璃纤维、胶膜等几种材料制成。

4—引线：其作用是作为导线导出敏感栅产生的电信号，通常用镀锡铜线制成。

电阻应变片的工作原理是压电效应。电阻应变片的测量原理是：将电阻应变片的金属丝紧密地粘贴弹性体上，当弹性体受外力（压力、拉力等）而产生变形时，电阻应变片的金属丝的长度和横截面积也因弹性体形变而发生变化，由此其电阻大小发生变化，经过电阻应变片的电压降也发生变化，通过测量这一电压变化前后的电压差值，在经过相应的计算或仪表测量，便可以得出外力的大小。本次设计选择的电阻应变片式传感器的量程为 $0 \sim 10\text{kg}$ ，测量精度为 0.01% ，在量程范围内的测量误差的范围为 $\pm 0.01\text{kg}$ 。

3.4.2 测量转换电路

电阻应变片将弹性体受外力产生的形变转换成电阻相对变化 $\Delta R/R$ 后，还需要采用一些电路将电阻的变化转换为电压或者电流的变化，这样才可利用测量仪表对变化前后的电压或电流进行测量并得出电压或电流差值。此次设计采用电桥电路将电阻应变片电阻的变化量转换为可测的电压或电流。直流电桥测量电路图，如图 3-4 所示。

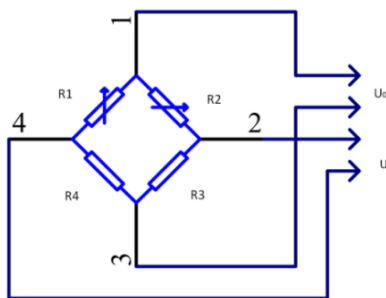


图 3-4 测量转换电路

如图 3-4 所示，电桥初始平衡条件为 $R_2 / R_1 = R_3 / R_4$ ，当 $R_L \rightarrow \infty$ 时，电桥输出电压 U_0 应为：

$$U_0 = U_i \left(\frac{R_1}{R_1 + R_2} - \frac{R_3}{R_3 + R_4} \right) \quad (3-9)$$

应变片工作时，其电阻变化为 ΔR ，此时有不平衡电压输出：

$$U_0 = \frac{\frac{R_3}{R_4} \frac{\Delta R_1}{R_1}}{\left(1 + \frac{R_2}{R_1} + \frac{\Delta R_1}{R_1}\right) \left(1 + \frac{R_3}{R_4}\right)} U_i \quad (3-10)$$

设桥臂比 $n = R_2 / R_1$ 电桥初始平衡条件为 $R_2 / R_1 = R_3 / R_4$ ， $\Delta R_1 \ll R_1$ ，将分母中的 $\Delta R_1 / R_1$ 略去，可得式子：

$$U_0 = \frac{n}{(1+n)^2} \frac{\Delta R_1}{R_1} U_i \quad (3-11)$$

电桥灵敏度定义为 k_μ ，上式中 $\frac{n}{(1+n)^2}U_i$ 就是电桥灵敏度的大小。由此可见电源电压越大，

电桥灵敏度系数 k_μ 的值也越大， k_μ 与电源电压的大小成正比。如果电源电压被提高，因为电阻应变片功耗是有一定限制的，此时，电桥灵敏度系数 k_μ 与桥臂比 n 的大小相关。

当 $n=1$ 时，

$$U_o = \frac{U_i}{4} \frac{\Delta R_1}{R_1} \quad (3-12)$$

$$k_\mu = \frac{U_i}{4} \quad (3-13)$$

由此可见，当电源电压 U_i 及电阻相对值恒定时，各桥臂电阻值的大小不会影响到电桥的灵敏度。电桥对称的条件是桥臂比 $n=1$ ，此次设计采用的就是这种形式的电桥。

3.5 LCD 显示模块

液晶显示器是一种利用液晶的扭曲一向列效应制成的显示器，其具有功耗较低、抗干扰能力较强、体积较小、价格低廉等优点。本次设计的智能电子秤的重量显示模块选用的是LCD1602 液晶显示器。LCD1602 液晶显示器是一种专门用来显示符号、数字、字母等的点阵型液晶模块。其组成部分的规格有两种，一种是由若干个 5X11 的等点阵字符位组成，另一种是由若干个 5X7 规格的等点阵字符位组成，每个字符的显示单元都与点阵字符位一一对应，点阵字符位与点阵字符位之间、行与行之间都存在着间隔，这些间隔将液晶显示器中的字符与字符之间以及行与行之间隔离开来。LCD1602 液晶模块引脚功能如下图 3-5 所示。

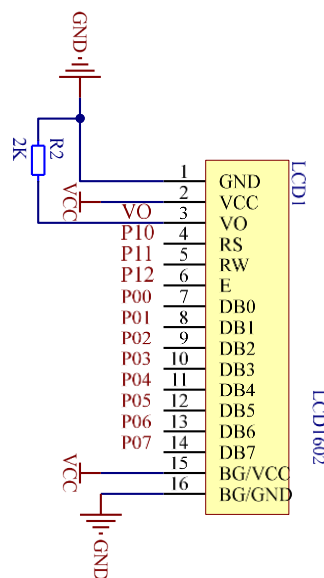


图 3-5 LCD1602 液晶模块引脚功能图

3.6 超重报警模块

根据智能电子秤的设计需要，设计出超重报警模块，其电路如图 3-6 所示。超重报警模块的作用是：当智能电子秤称量物品重量时，如果被称量物品的重量超过智能电子秤量程上限，即大于 10kg 时，发出指令使蜂鸣器发出声音和二极管闪烁来提示使用者，以避免因被称量物品超过量程太多造成电子秤损伤或毁坏。

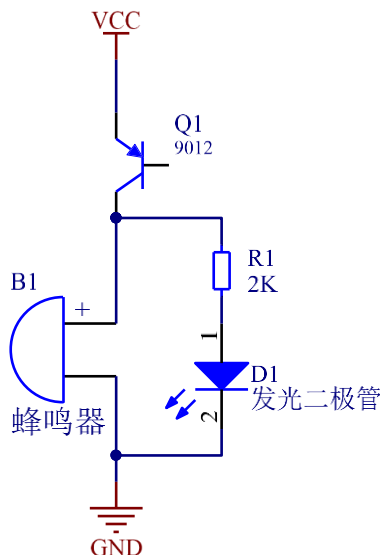


图 3-6 超重报警模块

如上图 3-6 所示，超重报警模块由一个 $2k\Omega$ 的定值电阻、一个发光二极管、一个 9102 三极管、一个蜂鸣器组成，由单片机的 I/O 口 P1.1 口控制其工作情况。其工作原理是：当智能电子秤称量物品重量时，单片机将接收到的当前物品的重量值与设定的最大量程值进行比较，若被称量物品的重量大于电子秤最大量程 10kg，则单片机将调用超重报警子程序，并经 I/O 口 P1.1 发出相关指令，向超重报警模块的三极管的基极输出低电平信号，使得三极管导通，蜂鸣器和发光二极管与电源的正负极形成回路，以发出声音和灯光闪烁的方式向使用者发出超重警报。若电子秤未称量物品或被称量物品的重量小于量程最大值 10kg 时，单片机经 P1.1 I/O 口向三极管基极输出高电平信号，三极管处于截止状态，此时超重报警系统处于开路状态，不发出警报。

3.7 复位电路

智能电子秤在每次使用时，都需要对单片机进行复位（初始化），这样才能使单片机系统的各部件及电子秤的各功能部件从初始状态开始工作。本次设计采用了按键复位的复位方式，如图 3-7 所示。

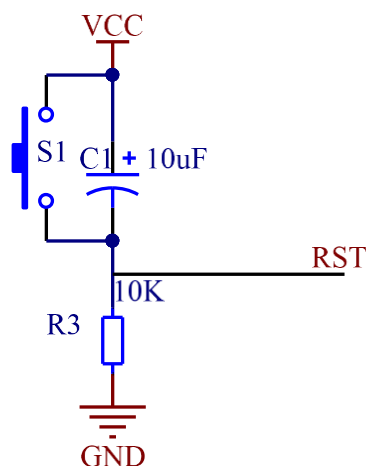


图 3-7 按键复位电路图

如上图 3-7 所示，复位电路由一个独立按键、一个 $10k\Omega$ 电阻、一个 $10\mu F$ 的电容组成。复位电路工作原理时：使用按键复位时，复位电路中的电容会给单片机的 RST 引脚一个短暂的高电平信号，这一信号会随着 V_{CC} 端的 +5V 电源对电容的充电过程而逐渐减弱，由此可见，RST 端的高电平持续时间的长短是由电容的充电时间决定的。因此，RST 端的高电平信号必须要维持足够长的时间（两个机器周期及以上）才能保证系统能够可靠地复位。复位的条件是：一、单片机系统处于正常工作状态且振荡器处于稳定状态。二、复位信号必须是持续两个机器周期及以上的高电平。满足以上两点，单片机就可以响应并且使系统复位。

3.8 按键处理模块

独立按键是智能电子秤中非常重要的部分，当我们在系统程序中编写入想要用独立按键实现的功能的子程序，再配合独立按键，便可更加直观、方便、快捷地对电子产品的参数、功能等进行设定、修改。本次设计使用了三个 SW-PB 独立按键，用来设定智能电子秤的量程。独立按键模块的电路图如下图 3-8 所示。

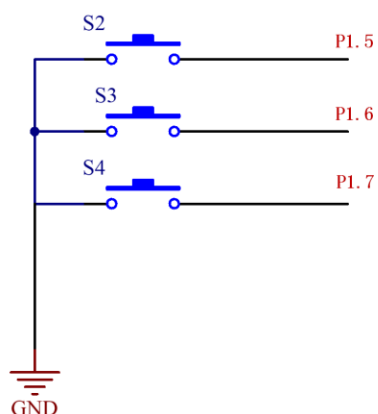


图 3-8 智能电子秤的按键输入模块

智能电子秤量程的设定：独立按键 S2 与 AT89C52 单片机的 P1.5 端口相连接，在设定智能电子秤量程时，独立按键 S2 每按下一次，显示位上的数字便减“1”，当显示位上的数字位“0”时，按下此按键数字便不再变化；独立按键 S3 与单片机的 P1.6 端口相连接，

独立按键 S3 每按下一次，数字显示位上的数字便加“1”，但显示为上的数字为“9”时，按下此按键该数字显示位上的数字不再变化；独立按键 S4 与单片机的 P1.7 口相连接，其主要作用是选择需要设定数值的数字显示位，独立按键 S4 每按下一次，光标便会向右移动一个数字显示位，当光标移动到最右边的数字显示位时，此时若在按一次按键 S4，光标便会移到最左边的数字显示位，如此从左往右循环移动。

微调和去皮：连续 5 次按下按键 S3，当液晶显示屏的左上角第一个数字显示位出现一个“M”符号时，便可根据校准需要利用独立按键 S2 和独立按键 S4 对当前值进行微调，每按下一次按键 S2，便会在当前重量值的基础上加 0.001，每按下一次按键 S4，便会在当前值的基础上减去 0.001，此时若按下复位电路（图 3-7）中的按键 S1，便可将当前值设定为智能电子秤去皮的重量值。

4 软件设计

4.1 软件开发环境

本次设计的程序编写设计部分采用的是 Keil μ Visio4 程序开发软件。Keil C51 是德国 Keil Software 公司研发的 51 系列兼容单片机 C 语言软件开发系统。可以用来完成连接、编辑、仿真、编译、调试等整个开发流程。本次程序设计使用的 Keil μ Visio4 程序编译软件是 Keil 公司于 2009 年 2 月发布的，其具有十分强大的界面功能，主要由项目管理器、源程序文件编辑器、源程序调试器等几部分组成。

C 语言与汇编语言相比具有结构更简单、编写更容易、可读性更好、功能更多样、可维护性强等优点，而且 C 语言比汇编语言学起来更加容易，用起来也更方便。所以，此次设计的程序设计部分选用 C 语言进行设计、编写。

4.2 主程序流程图

主程序的主要作用是对系统各功能模块进行初始化以及调用各功能模块的子程序，本次设计的智能电子秤的主程序如下图 4-1 所示。

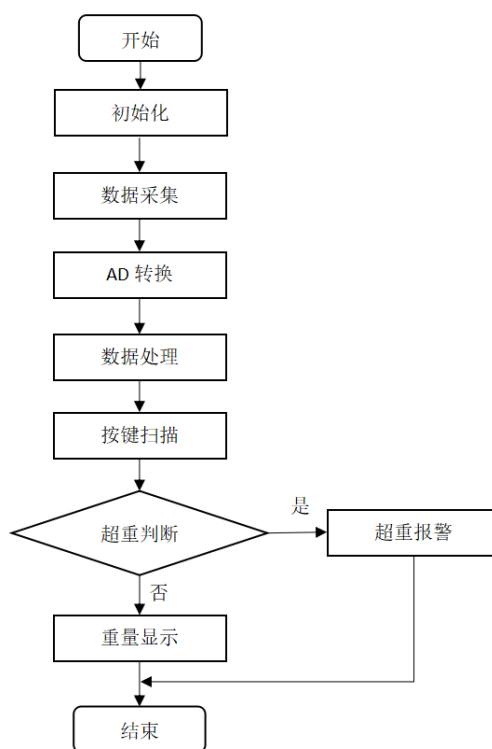


图 4-1 主程序流程图

如上图 4-1 所示，本次设计的智能电子秤的主程序调用的有以下几组子程序：

- 1、AT89C51 单片机的初始化程序；
- 2、LCD1602 液晶显示模块的初始化程序；
- 3、独立按键子程序；

- 4、超重报警模块子程序；
- 5、称重传感器子程序；
- 6、HX711 模数（A/D）转换模块子程序。

主程序运行的大致流程是：初次使用电子秤时，单片机及各功能模块、各 I/O 口的初始化程序开始运行。称量物品时，主程序调用称重传感器的子程序使称重传感器对物品的重量数据进行采集，获取重量数据的模拟量后，主程序调用 AD 转换模块子程序使 AD 转换模块对称重传感器输出的模拟量进行模数转换，得出物品重量的数字量，然后单片机对数字量进行计算、处理，并与设定的量程的最大值（10kg）进行比较，若当前物品的重量大于智能电子秤量程的最大值，则主程序调用超重报警模块的子程序使其向使用者发出声、光报警；若物品的重量没有超出电子秤量程的最大值，则主程序调用重量显示模块的子程序，对当前物品的重量显示在液晶显示屏上。在以上整个过程中，主程序会一直调用按键处理模块的子程序，对各个独立按键的动作情况进行扫描，若有按键按下，则优先调用按键处理模块的子程序和显示模块的子程序，对当前按键的功能进行显示，并执行按键操作的相应功能。

4.3 显示部分流程图

系统显示部分程序运行的大致流程是：在智能电子秤的运行过程中，单片机根据独立按键是否按下以及独立按键功能的不同来调用相应的子程序，使液晶显示器显示当前按键的功能或显示被称量物品的重量。系统显示部分的程序框图如下图 4-2 所示。

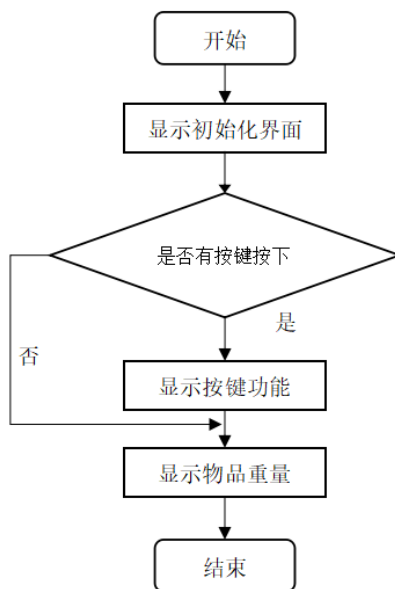


图 4-2 显示部分流程图

4.4 系统按键部分流程图

根据设计需求，本次设计的按键部分为 4 个独立按键，其各自的功能为：按键 S1 用于数据清零，另外三个按键用于设定便携式智能电子秤的量程，在这三个按键中，按键 S2 用

于选择需要设置的数字显示位，按键 S3、S4 用于根据需要在选定的数字显示位上设定所需的数值。此外，连续 5 次按下按键 S3，可以进行微调、去皮参数的设置。系统按键部分流程图如下图 4-3 所示。

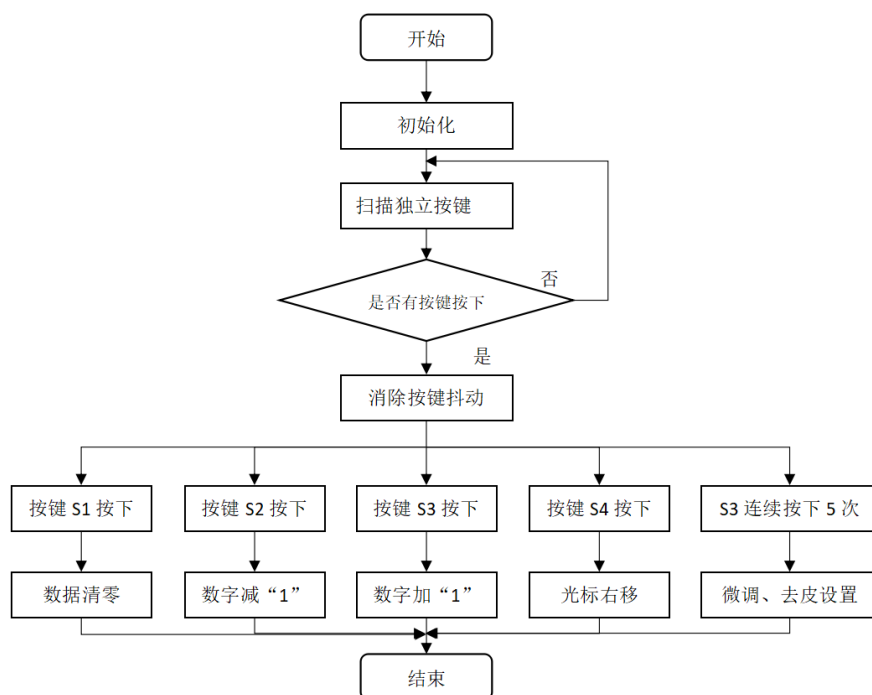


图 4-3 按键部分流程图

4.5 超重报警部分流程图

智能电子秤超重报警部分流程图如下图 4-4 所示

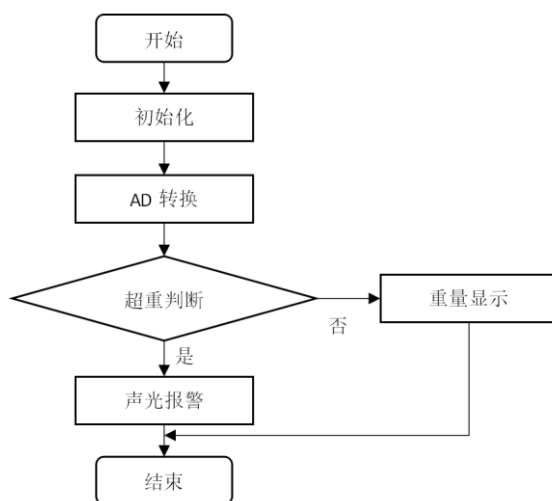


图 4-4 超重报警部分流程图

超重报警部分大致的工作方式是：称重传感器将物品重量转换为可测量的电压变化量，然后 AD 转换器将这一模拟量转换为数字量送入单片机，单片机将物品重量值与设定的量程进行比较，如果物品重量超出智能电子秤量程的最大值，则单片机会调用超重报警子程序并发出指令，使超重报警器发出超重警报。

5 系统调试与调试结果

5.1 硬件电路的制作

根据设计的电路图，将各元器件合理地排布于覆铜板上，再根据电路图进行手工焊接。实物图如图 5-1 所示。

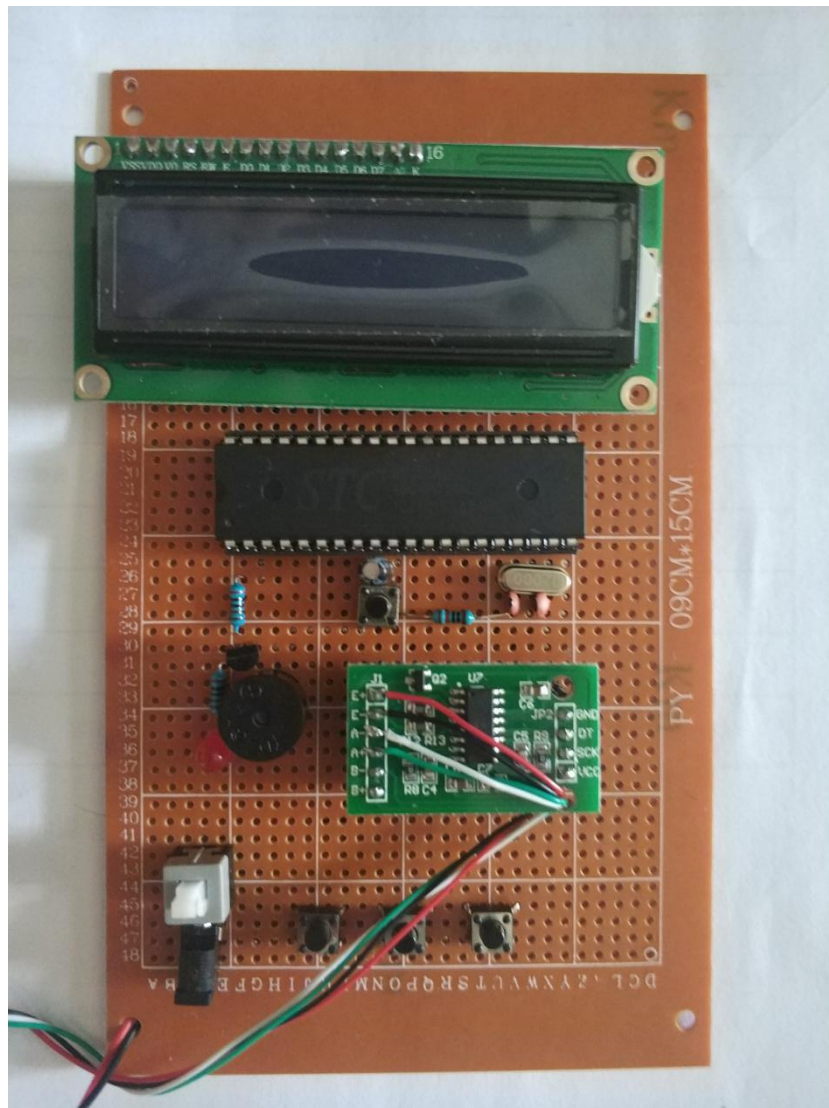


图 5-2 系统电路板实物图

5.2 硬件电路的调试和结果

5.2.1 硬件电路的调试

1、硬件部分各元器件焊接完毕后，用万用表对各元器件及各段电路进行分别测量，检测是否有元器件功能不正常以及电路是否有不符合设计要求的短路、开路等情况，更换不能

正常工作的元器件并修改不正确的电路。

2、对编写的程序进行检查、修改，最终检查无误后，将 STC89C52 单片机装入开发板，并且将程序烧录进单片机，然后将单片机取下插入电子秤的单片机座上。

3、万用表检测电路完全无误后，将三节 1.5V 的 5 号干电池正确装入电池槽，然后打开电源开关，蜂鸣器发出一声“滴”声并伴随着发光二极管闪烁一下后，LCD1602 液晶显示器显示的各项参数均正常。

4、操作三个独立按键，将电子秤量程设定为 0~10kg，然后将砝码放置于电子秤称台上，此时电子秤显示器示值与砝码本身标定的重量值一致，取下砝码，然后按下复位按键，电子秤显示器重量示值归“0”。将估计重量值大于 10kg 的物品慢慢放置于秤台上，当被称量物品施加于电子秤的重量刚刚超过 10kg 时，电子秤蜂鸣器发出报警声，电子秤超重报警系统工作正常。

5.2.2 硬件电路的调试结果

在对硬件电路进行调试时，使用智能电子秤对以下物品进行测量，得出的结果如下表 5-1 所示。

表 5-1 物品重量称量结果表

物品名称	标准重量（g）	智能电子秤示值（kg）
5g 砝码	5	0.005
10g 砝码	10	0.010
20g 砝码	20	0.020
50g 砝码	50	0.050
一元硬币	6.05	0.006
5kg 杠铃片	5000	4.998
10kg 杠铃片	10000	9.995
20kg 杠铃片	20000	超重报警

如表 5-1 所示，根据称量物品重量的数据可知，便携式智能电子秤称量物品的测量精度、量程以及测量误差均符合设计要求，电子秤各功能模块均能正常工作，智能电子秤的设计达到预期效果。

设计总结

通过本次便携式智能电子秤的设计我学会了如何将书本中、课堂上所学到的知识应用到实践中去，提高了发现问题、分析问题、解决问题的能力，提高了动手能力和自学能力，巩固了我在大学四年中学到的部分专业知识。了解了一件简单的电子产品的设计与制作有统硬件电路的设计、系统软件的编程和仿真以及软、硬件电路的联合调试这几个阶段。根据设计的要求，通过网络、图书馆、询问指导老师等方法查询收集相关资料，再结合自己所学的知识，从明确设计需求，到设计硬件电路，再到软件系统程序的编写，最后进行软、硬件联合调试以及系统软、硬件故障的检测与排除。

在完成便携式智能电子秤设计的过程中，为了满足设计要求，查询相关资料是非常重要的。首先，要对核心部分即 STC89C52 单片机的内部结构，及各部分功能有详细的了解，在软件编写的过程中，利用画流程图的方式理清编写思路，软件编写完成后，检查程序中所存在的错误并改正，对控制不同部分、实现不同功能的各部分软件加以注释，使人观之一目了然。

参考文献

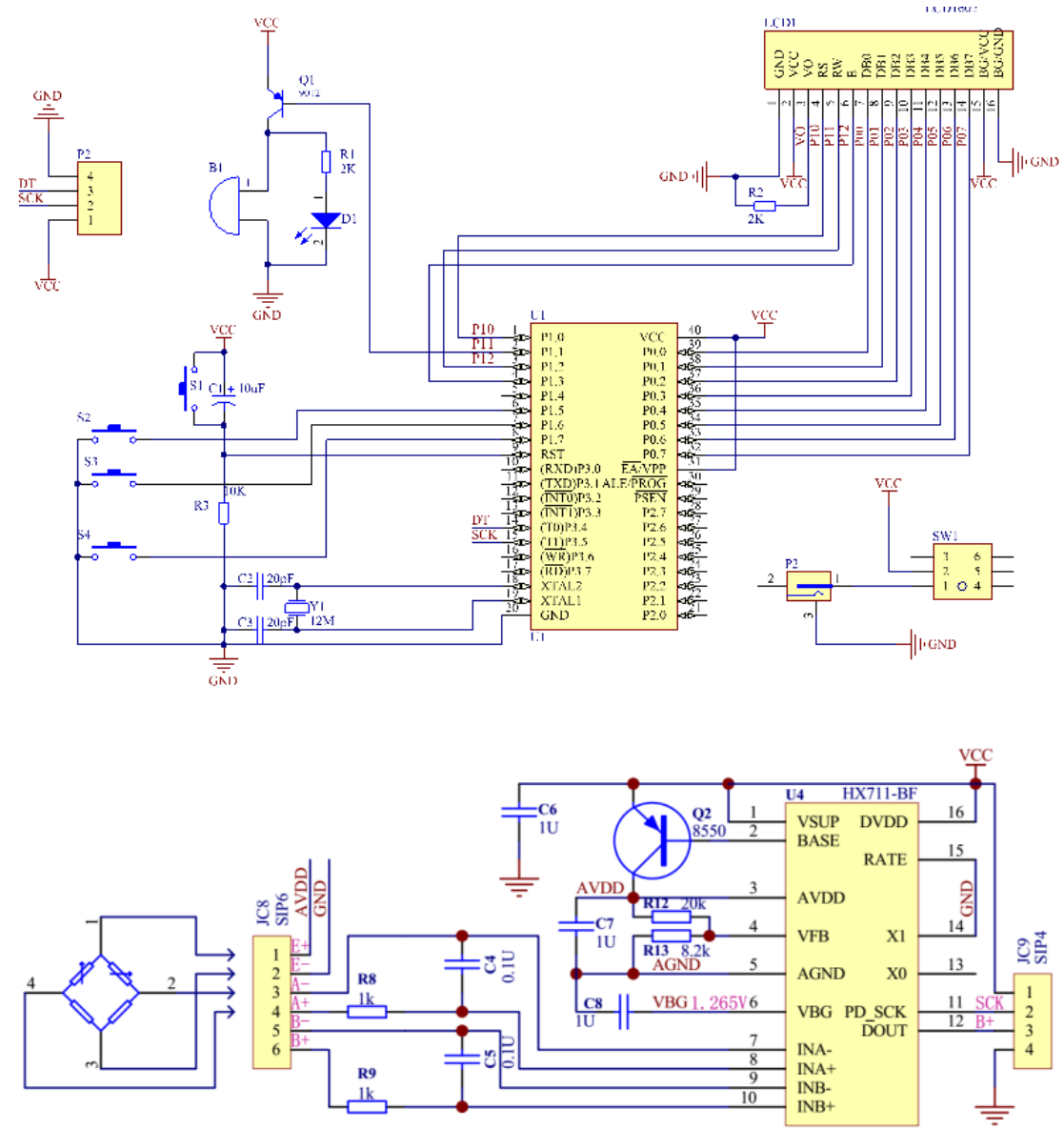
- [1]胡向东. 传感器与检测技术[M]. 机械工业出版社, 2013. 9.
- [2]谢维成. 单片机原理及 C51 程序设计[M]. 清华大学出版社, 2014. 1.
- [3]王祁. 智能仪器设计基础[M]. 机械工业出版社, 2015. 2.
- [4]谭浩强. C 语言设计教程[M]. 清华大学出版社, 2013. 8.
- [5]朱巍. 微机原理及接口技术[M]. 人民邮电出版社, 2016. 1.
- [6]马岚. 数字集成电路[M]. 电子工业出版社, 2017. 1.
- [7]李阳. 电子与自动化类毕业设计指导[M]. 中国电力出版社, 2016. 5.
- [8]韦建英. 徐安静 模拟电子技术[M]. 华中科技大学出版社, 2013. 2.
- [9]刘理云. 嵌入式单片机开发与应用[M]. 北京理工大学出版社, 2016. 1.
- [10]魏芬. 基于 proteus 的单片机实验与课程设计[M]. 清华大学出版社, 215. 3.

致谢

毕业设计是对大学所学部分知识的一次综合检验与应用,也是对四年的大学学习所交出的一份答卷与总结,毕业设计完成的情况体现的不仅是我们大学期间对相关知识的学习情况、掌握情况,也体现了我们将所学知识用于实际应用的能力。感谢指导老师聂文亮在本次设计的过程中不辞辛劳地予以帮助,导师的循循善诱、孜孜教导使学生受益匪浅并感佩于心。在此次设计过程当中,每当遇到有不懂的问题时,老师和同学都会在百忙之中抽出时间为我悉心讲解,和我一起探讨、分析并解决问题,指出我设计上的误区和不足,帮我改进设计中的欠缺点,使我能及时地发现问题、改正错误,把设计顺利的进行下去,在此也表示最衷心的感谢

附录

附录一 智能电子秤原理图



附录二 源程序

```
#include <reg52.h>           //调用单片机头文件
#define uchar unsigned char  //无符号字符型 宏定义      变量范围 0~255
#define uint  unsigned int   //无符号整型 宏定义      变量范围 0~65535

#define ulong unsigned long

#include<intrins.h>

uchar code table_num[]="0123456789abcdefg";

sbit rs=P1^0;    //寄存器选择信号 H:数据寄存器      L:指令寄存器
sbit rw=P1^2;    //寄存器选择信号 H:数据寄存器      L:指令寄存器
sbit e =P1^3;    //片选信号    下降沿触发

sbit hx711_dout = P3^5;
sbit hx711_sck  = P3^4;

sbit beep = P1^1; //蜂鸣器 I0 口定义

long weight;

unsigned long HX711_Buffer = 0;
unsigned long Weight_Maopi = 0;

uint temp,qi_weight;
bit chaozhong_flag;
bit lei_ji_flag;
uchar flag_200ms,key_d ;
bit flag_bufang ;    //布防标志位

float  GapValue = 1120;

bit flag_alarm;
```

```

uchar flag_p;

uchar a_a, i, menu_1, value;

uint t_high = 80; // 设置值

/*****
删除键
去皮
价格清零
*****/

/*****1ms 延时函数***12M 晶振*****/
void delay_1ms(uint q)
{
    uint i, j;
    for(i=0; i<q; i++)
        for(j=0; j<120; j++);
}

/*****
* 名称 : delay_uint()
* 功能 : 小延时。
* 输入 : 无
* 输出 : 无
*****/
void delay_uint(uint q)
{
    while(q--);
}

/*****
* 名称 : write_com(uchar com)
* 功能 : 1602 命令函数
* 输入 : 输入的命令值
* 输出 : 无
*****/

```

```
*****/
void write_com(uchar com)
{
    e=0;
    rs=0;
    rw=0;
    P0=com;
    delay_uint(3);
    e=1;
    delay_uint(25);
    e=0;
}

/*****
* 名称 : write_data(uchar dat)
* 功能 : 1602 写数据函数
* 输入 : 需要写入 1602 的数据
* 输出 : 无
*****/
void write_data(uchar dat)
{
    e=0;
    rs=1;
    rw=0;
    P0=dat;
    delay_uint(3);
    e=1;
    delay_uint(25);
    e=0;
}

/*****lcd1602 上显示这字符函数*****/
void write_string(uchar hang,uchar add,uchar *p)
{
    if(hang==1)
        write_com(0x80+add);
    else
```

```

        write_com(0x80+0x40+add);
    while(1)

    {
        if(*p == '\0') break;
        write_data(*p);
        p++;
    }
}

/*****压力显示程序*****/
void write_1602_y1(uchar hang,uchar add,long date)
{
    if(hang==1)
        write_com(0x80+add);
    else
        write_com(0x80+0x40+add);
    write_data(0x30+date/10000%10);
    write_data(0x30+date/1000%10);
    write_data('.') ;
    write_data(0x30+date/100%10);
    write_data(0x30+date/10%10);
    write_data(0x30+date%10);
    write_data('K');
    write_data('g');
}

/*****lcd1602 初始化设置*****/
void init_1602()          //lcd1602 初始化设置
{
    write_com(0x38);      //
    write_com(0x0c);
    write_com(0x06);
    delay_uint(1000);
    write_string(1,0,"  W:00.00kg  ");
    write_string(2,0," Set:00.00kg  ");
    write_1602_y1(2,5,t_high ); //显示数

```



```
}

void Delay__hx711_us(void)
{
    _nop_();
    _nop_();
}

unsigned long HX711_Read(void) //增益 128
{
    unsigned long count;
    unsigned char i;
    hx711_dout=1;
    Delay__hx711_us();
    hx711_sck=0;
    count=0;
    // EA = 1;
    while(hx711_dout);
    // EA = 0;
    for(i=0;i<24;i++)
    {
        hx711_sck=1;
        count=count<<1;
        hx711_sck=0;
        if(hx711_dout)
            count++;
    }
    hx711_sck=1;
    count=count^0x800000;//第 25 个脉冲下降沿来时，转换数据
    Delay__hx711_us();
    hx711_sck=0;
    return(count);
}

void get_pizhong() //获取皮重，秤盘重量
{
    Weight_Maopi = HX711_Read();
}
```

```

}

void get_weight()          //获取被测物体重量
{
    weight = HX711_Read();
    weight = weight - Weight_Maopi;          //获取净重
    if(weight > 0)
    {
        weight = (unsigned int)((float)weight/GapValue);          //计算
实物的实际重量

    }
    else
    {
        weight = 0;
    }
}

/*****独立按键程序*****/
uchar key_can;    //按键值

void key()          //独立按键程序
{
    static uchar key_new;
    key_can = 20;          //按键值还原
    P1 |= 0xf0;
    if((P1 & 0xf0) != 0xf0)          //按键按下
    {
        delay_1ms(1);          //按键消抖动
        if(((P1 & 0xf0) != 0xf0) && (key_new == 1))
        {
            //确认是按键按
下

            key_new = 0;
            switch(P1 & 0xf0)
            {
                case 0xd0: key_can = 3; break;          //得到 k1 键值
                case 0xb0: key_can = 2; break;          //得到 K2 键值
                case 0x70: key_can = 1; break;          //得到 k3 键值
            }
        }
    }
}

```

```
        }
        beep = 0;           //打开蜂鸣器
        delay_1ms(50);
        beep = 1;           //关闭蜂鸣器
    }
}
else
    key_new = 1;
}

/*****按键处理显示函数*****/
void key_with()
{
    if(menu_1 == 0)
    {
        if(key_can == 3)    //去皮
        {
            key_d = 0;
            get_pizhong();   //获取皮重，秤盘重量
        }
    }
    if(key_can == 1)        //设置键
    {
        key_d = 0;
        menu_1 ++;
        if(menu_1 > 5)
        {
            menu_1 = 0;
            init_1602();    //lcd1602 初始化
        }
    }
    if(menu_1 == 1)         //设置数
    {
        if(key_can == 2)
        {
            if(t_high / 10000 < 5)
                t_high += 10000 ;           // 设置数加
        }
    }
}
```

```

        if(key_can == 3)
        {
            if(t_high / 10000 != 0)
                t_high -=10000 ;           // 设置数减
        }
        write_1602_y1(2,5,t_high ); //显示数
        write_com(0x80+0x40+5);          //将光标移动到秒个位
        write_com(0x0f);                  //显示光标并且闪烁
    }
    if(menu_1 == 2)                        //设置数
    {
        if(key_can == 2)
        {
            if(t_high / 1000 % 10 != 9)
                t_high +=1000 ;           // 设置数加
        }
        if(key_can == 3)
        {
            if(t_high / 1000 % 10 != 0)
                t_high -=1000 ;           // 设置数减
        }
        write_1602_y1(2,5,t_high ); //显示数
        write_com(0x80+0x40+6);          //将光标移动到秒个位
        write_com(0x0f);                  //显示光标并且闪烁
    }
    if(menu_1 == 3)                        //设置数
    {
        if(key_can == 2)
        {
            if(t_high / 100 % 10!= 9)
                t_high +=100 ;           // 设置数加
        }
        if(key_can == 3)
        {
            if(t_high / 100 % 10 != 0)
                t_high -=100 ;           // 设置数减
        }
        write_1602_y1(2,5,t_high ); //显示数
    }

```

```
        write_com(0x80+0x40+8);           //将光标移动到秒个位
        write_com(0x0f);                   //显示光标并且闪烁
    }
    if(menu_1 == 4)                         //设置数
    {
        if(key_can == 2)
        {
            if(t_high / 10 % 10 != 9)
                t_high +=10 ;               // 设置数加
        }
        if(key_can == 3)
        {
            if(t_high / 10 % 10 != 0)
                t_high -=10 ;               // 设置数减
        }
        write_1602_y1(2,5,t_high ); //显示数
        write_com(0x80+0x40+9);           //将光标移动到秒个位
        write_com(0x0f);                   //显示光标并且闪烁
    }
    if(menu_1 == 5)                         //设置数
    {
        if(key_can == 2)
        {
            if(t_high % 10 != 9)
                t_high +=1 ;               // 设置数加
        }
        if(key_can == 3)
        {
            if(t_high % 10 != 0)
                t_high -=1 ;               // 设置数减
        }
        write_1602_y1(2,5,t_high ); //显示数
        write_com(0x80+0x40+10);           //将光标移动到秒个位
        write_com(0x0f);                   //显示光标并且闪烁
    }
}
```

```

/*****报警函数*****/
void alarm_dis()
{
    if(weight > t_high)
    {
        beep = ~beep;          //蜂鸣器叫 3 声
    }
    else
    {
        beep = 1;              //关闭蜂鸣器
    }
}

/*****主函数*****/
void main()
{
    beep = 0;                  //蜂鸣器叫一声
    delay_1ms(100);
    P0 = P1 = P2 = P3 = 0xff;   //单片机 IO 口初始化为 1
    init_1602();               //lcd1602 初始化
    while(1)
    {
        key(); //独立按键程序
        if(key_can < 20)
        {
            key_with();
        }
        flag_200ms ++;
        if(flag_200ms >= 200)
        {
            flag_200ms = 0;
            get_weight(); //获取被测物体重量
            if(weight <= 5)
                weight = 0;
            if(menu_1 == 0)
            {

```

```
        write_1602_y1(1,5,weight);           //显示压力
    }
    alarm_dis();    //报警函数

}
delay_1ms(1);
}
}
```