

学科分类号 0806

怀化学院

本科生毕业论文(设计)

题目 (中文): 网络阻抗测试仪——硬件部分

(英文): The Network Impedance Tester
—Hardware Part

学生姓名: 李 健

学 号: 0910406014

系 别: 物理与信息工程系

专 业: 通信工程

指导教师: 高吉祥 教授

起止日期: 2013. 1-2013. 5

2013年 5月 17日

怀化学院本科毕业论文(设计)诚信声明

作者郑重声明：所呈交的本科毕业论文(设计)，是在指导老师的指导下，独立进行研究所取得的成果，成果不存在知识产权争议。除文中已经注明引用的内容外，论文不含任何其他个人或集体已经发表或撰写过的成果。对论文的研究做出重要贡献的个人和集体均已在文中以明确的方式标明。本声明的法律结果由作者承担。

本科毕业论文（设计）作者签名：

年 月 日

目 录

摘要.....	I
关键词.....	I
Abstract.....	I
Key words.....	II
1 前言.....	1
2 设计方案论证与比较.....	3
2.1 电源模块.....	3
2.2 主控制模块.....	3
2.3 按键显示模块.....	4
2.4 信号源模块.....	5
2.5 测量电路模块.....	6
2.6 相敏检波模块.....	7
2.7 本章小结.....	8
3 系统原理及理论分析.....	8
3.1 自由轴测量法原理.....	8
3.2 相敏检波原理.....	11
3.3 本章小结.....	13
4 硬件设计.....	13
4.1 系统总体设计.....	13
4.2 电源模块.....	14
4.3 主控制模块.....	15
4.4 按键显示模块.....	16
4.5 信号源模块.....	17
4.6 测试模块.....	19
4.7 相敏检波模块.....	20

4.8 本章小结.....	22
5 软件设计.....	23
5.1 系统设计流程图.....	23
5.2 系统子程序说明.....	24
5.2.1 AD9851 转换程序.....	24
5.2.2 CH452 转换程序.....	25
5.2.3 单片机时钟选择程序.....	25
5.2.4 液晶显示程序.....	26
5.3 本章小结.....	27
6 数据测试及分析.....	27
6.1 测试内容与使用仪器.....	27
6.2 测试数据结果.....	27
6.3 数据误差分析.....	28
7 结论.....	29
参考文献.....	29
致 谢.....	31
附录.....	32
附录 A 系统电路图.....	32
附录 B 程序清单.....	35

网络阻抗测试仪——硬件部分

摘要

本系统以 TI 公司 16 位超低功耗单片机 MSP430G2553 为核心，以矩阵键盘和 12864 液晶屏为 人机交换界面。利用一片 AD9851 芯片组成信号发生器，通过程序控制产生频率为 1KHz-100KHz 的正弦波，经过低通滤波器滤除杂波，利用待测阻抗与基准阻抗分压电路和相敏检测器，检测出参考电压相量和阻抗电压向量在参考电压方向上的投影，并通过 MSP430G2553 内部 A/D 进行采样，将采样得到的电压信号在单片机内进行处理。自由轴法的基本思想是：待测阻抗 Z_x 和标准阻抗 Z_s 串联，严格要求被测参数矢量在 X、Y 坐标轴上投影准确正交，然后分别测出待测阻抗、标准阻抗两端的矢量电压 U_x 和 U_s 在直角坐标 X、Y 轴上的分量，最后送入单片机经过四则运算即可求出阻抗网络的阻抗值和阻抗角。

关键词

MSP430G2553 ; AD9851; 自由轴法; 相敏检波

The Network Impedance Tester——Hardware Part

Abstract

This system with the 16-bit ultra low power microcontroller MSP430G2553 TI company as the core, with matrix keyboard and 12864 LCD screen as man-machine exchange interface. Use AD9851 chip of signal generator, frequency generated by program control for sine wave of 1 KHZ to 100 KHZ, passes through A low-pass filter that filters out noise, using impedance under test and benchmark impedance and phase-sensitive detector bleeder circuit, detect the reference voltage vector in the reference voltage and impedance voltage vector in the direction of projection, and through the A/D chip samples, will be sampling the voltage signal into singlechip for

processing. Free axis method the basic idea is: for the impedance model Zs series impedance Zx and standards, strict measured parameter vector projection on X, Y axis orthogonal accurately, and then respectively measured impedance under test, standard impedance at each end of the vector voltage Ux and Us in cartesian coordinates X, Y axis component, finally into the microcontroller through the arithmetic to the impedance values of impedance network and impedance Angle.

Key words

MSP430G2553 ; AD9851; Free method of shaft; Phase sensitive detection

1 前言

随着电子技术的发展，数字控制等先进方法在数字电路应用领域的扩展，现今社会，产品智能化、数字化已成为人们追求的一种趋势，设备的性能，价格，发展空间等备受人们的关注，尤其对电子设备的精密度和稳定度最为关注。机电产品广泛应用于家电、通信、一般工业。无论是日常生活乃至航空航天和军事领域高端科技领域，电子技术的应用均日益深入。掌握必要的电子知识成为我们常生活必要技能。

传统的网络阻抗测量仪体积庞大，功耗大，国内外主要仪器厂家还产生了内含微处理器的各种网络阻抗参数测量仪。这种专用的网络阻抗测量仪具有多功能、多参量、多频率、高速度、高精度、大屏幕、菜单方式显示等优点。如高频导纳电桥，该产品是一种高准确度，宽测试范围的网络阻抗测量仪，有液晶显示屏，全中文菜单。可提供稳定的6位测试分辨率，测试信号电平可进行可编程设置，自动分配选择匹配系数实现自动化，能适用生产现场高速检测等应用。虽然高频导纳电桥具有许多的有点，但价格十分昂贵，不便于社会普及。所以在次基础上，我们要设计出更方便，更便宜的网络阻抗测试仪，在现有技术的基础上，我们加了一些改进，虽然达不到非常高的指标，但是，我们基本还是能够达到了测试的标准。

网络阻抗测量方法有许多种，各自有着不同的优缺点，根据实际需要选择适合的设计方法。比如交流电桥法，运用交流电桥法进行网络阻抗测量时系统要达到平衡，实现平衡必须满足两个条件，必须调

节两个桥臂的参数，才能使电桥完全达到平衡，而且往往需要对这两个参数进行反复地调节，所以交流电桥的平衡调节要比直流电桥的调节困难一些。具有较高的测量精度，被广泛采用，现已派生出许多类型。但电桥法测量需要反复进行平衡调节，测量时间长，很难实现快速的自动测量。还有谐振法，要求较高频率的激励信号，一般不容易满足高精度的要求。由于测试频率不固定，测试速度也很难提高。还有伏安法是最经典的方法，它的测量原理来源于阻抗的定义。即若已知流经被测阻抗的电流相量并测得被测阻抗两端的电压，则通过比率便可得到被测阻抗的相量。采用了基于自由轴伏安测量法的测量原理和四端测量技术。自由轴法可将复数变换成是实数的计算。大大降低了计算的复杂度。

为了方便和高灵敏度地测量网络阻抗，本设计利用 MSP430 单片机作为控制芯片。MSP430 系列单片机是美国德州仪器（TI）1996 年开始推向市场的一种 16 位超低功耗、具有精简指令集（RISC）的混合信号处理器（Mixed Signal Processor）。称之为混合信号处理器，是由于其针对实际应用需求，将多个不同功能的模拟电路、数字电路模块和微处理器集成在一个芯片上，以提供“单片机”解决方案。该系列单片机多应用于需要电池供电的便携式仪器仪表中。

综合之前的知识，我们设计了一种基于自由轴法的智能化的网络阻抗测量系统。该系统由信号源产生电路、测量电路、相敏检波电路、微处理器电路、显示电路、按键电路构成。具有自动换挡测量阻抗的功能。

2 设计方案论证与比较

2.1 电源模块

方案一：用开关稳压电源给整机供电，开关电源工作效率高，能够完成本作品电源的供电，但是开关电源比较复杂，而且体积也比较大，制作不便，而且输出纹波比较大，不适合对电源要求较高的精密仪器。

方案二：单片机控制系统以及外围芯片供电采用 78 系列三端稳压器件，通过全波整流，然后进行滤波稳压^[1]。此方案能满足题目要求，而且简单实用，易于自制。

综合考虑，因三端稳压器具有结构简单、外围器件少、性能优良、调试方便的各个特点，以及优点，所以电源部分采用三端稳压电路，故选用方案二。

2.2 主控制模块

方案一：采用中小规模集成电路构成的控制电路，此方法实现起来电路复杂，元件较多，容易出故障，而且不适合复杂的控制。

方案二：采用可编程逻辑器件（如 FPGA）构成的控制器，该控制器价格比较昂贵，不适合本设计。

方案三：采用以单片机为核心的最小系统方案，以主控制芯片 MSP430G2553 的核心，由于对此芯片比较熟悉，且外围元件不多，容易掌握，可靠性高。

综合考虑，方案三从性价比和性能上都比方案一和方案二好，故选择方案三，原理图框如图 2.1 所示。

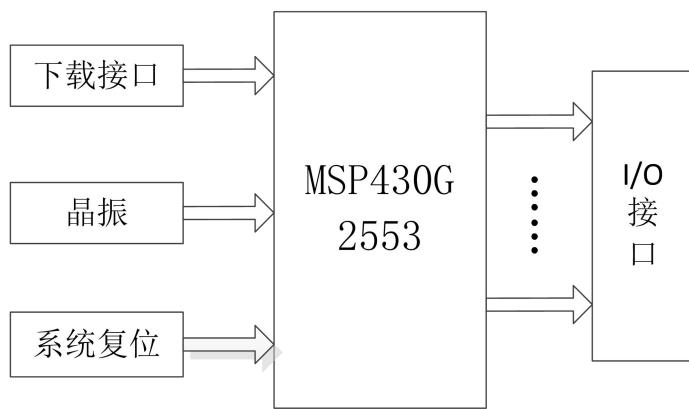


图 2.1 控制模块原理框图

2.3 按键显示模块

方案一：采用独立键盘作为控制，其中每个按键占用一条 I/O 线，当按键数量较多时，I/O 口利用率不高，但程序编制简单。适用于所需按键较少的场合。液晶采用 LCD12864，它能够显示足够的信息量且方便使用。

方案二：采用矩阵键盘作为控制，电路连接复杂，但提高了 I/O 口利用率，软件编程较复杂。适用于需使用大量按键的场合，为了进一步提高 I/O 的利用率，可以使用矩阵按键驱动芯片 CH452，它最少仅需两根线就可以和单片机通信，实现 8*8 矩阵键盘功能。液晶同样采用 LCD12864，它能够显示足够的信息量且方便使用。

综合考虑，方案二比方案一合理，它克服了按键少的缺点，提供丰富的按键资源且仅需两个 I/O 口资源，液晶 LCD12864 作为显示也能满足要求，故选取方案二。

2.4 信号源模块

方案一：采用集成函数发生器产生的波形利用函数发生器如（ICL8038）产生频率可变的正弦波、方波、三角波。此方案实现电路复杂，难于调试，实现波形难度大，且要保证技术指标困难，故采用此方案不理想。

方案二：采用锁相环间接合成（PLL）虽然具有工作频率高、宽带、频谱质量好的优点，但由于锁相环本身是一个惰性环节，锁定时间长。另外由模拟方法合成的正弦波的参数（如幅度频率和相位等）都很难控制，而且要实现 1KHz~10KHz 大范围的频率变化相当困难，不易实现。

方案三：采用直接数字式频率合成（简称 DDS），利用单片机编程控制 AD9851 芯片，按不同的频率要求以频率控制字 K 为步进对相位增量进行累加，以累加相位值作为地址码读取存放在存储器内的波形数据。经 D/A 转换和幅度控制，再滤波就可以得到所需要波形。由于 DDS 具有相对带宽，频率转换时间极短（可小于 20μs），频率分辨率高，全数字化结构便于集成等优点，以及输出相位连续，频率、相位和幅度均可实现程控，因此，可以完全满足本题的要求。

综合考虑，方案三比其他方案合理，克服了其他方案的缺点，故选取方案三，该芯片原理图框如图 2.2 所示。

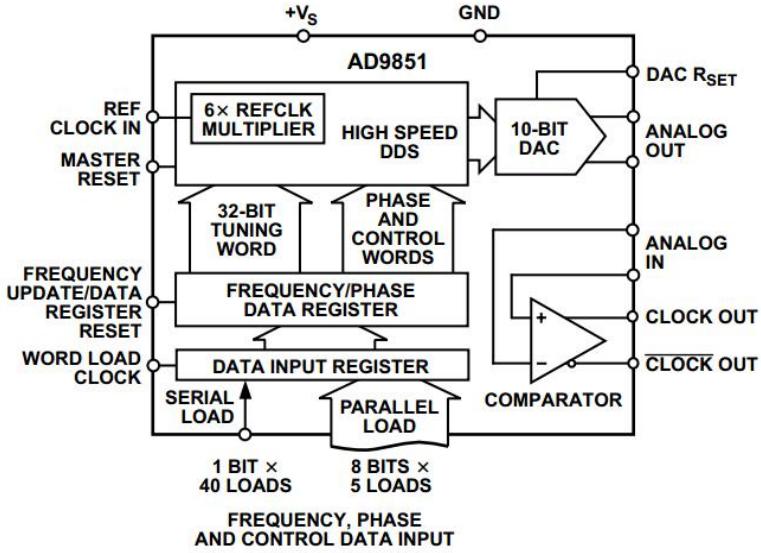


图 2.2 AD9851 芯片原理框图

2.5 测量电路模块

方案一：谐振法要求较高频率的激励信号，一般不容易满足高精度的要求。由于测试频率不固定，测试速度也很难提高。需要调谐到谐振，阻抗测量精度低。

方案二：动平衡电桥法对从低频至高频的宽频率范围，且宽的阻抗测量范围内具有高精度。但其不能适应更高的频率范围要求，而且检测器电路要求比较高，硬件电路设计较为复杂，调试难度大。

方案三：基于自由轴法的阻抗测量方法将对阻抗的测量转换成对相量比值的测量，而且自由轴法坐标轴的选择可以是任意的，参考电压可以不与任何一个被测电压的方向同向，由于其精确的正交坐标系主要靠软件产生和保证，硬件电路大大简化，还消除了固定轴法难以克服的同相误差，提高了精确度。被测参数通过 MCU 获得，因而可以方便的计算出阻抗模值|Z_X|、阻抗角、品质因数等。

综合考虑，故采用方案三，原理图框如图 2.3 所示。

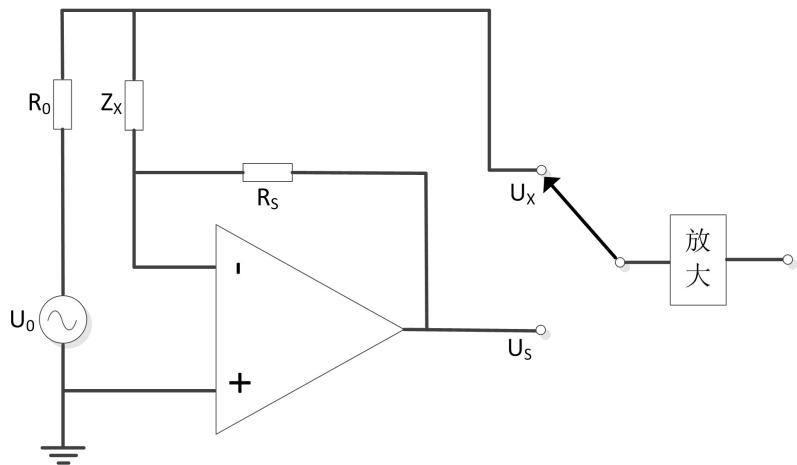


图 2.3 测量电路模块原理框图

2.6 相敏检波模块

方案一：采用专用的数字电表芯片 ICL7135，ICL7135 是采用 CMOS 工艺制作的单片 4.5 位 A/D 转换器，只要附加译码器，数码显示器，驱动器及电阻电容等元件，就可组成一个满量程为 2V 的数字电压表。

方案二：采用差分运放以及加法器等电路组成信号调节电路，最后利用 MSP430G2553 内部自带的 10 位 A/D 转换器采集电压值，该方案原理简单，实现起来不复杂，而且充分利用单片机芯片内部资源，节省外部 A/D 芯片。

综合考虑，由于 MSP430G2553 内部具有集成了 A/D 芯片，而且使用起来方便，所以方案二更合理，故采用方案二，原理图框如图 2.4 所示。

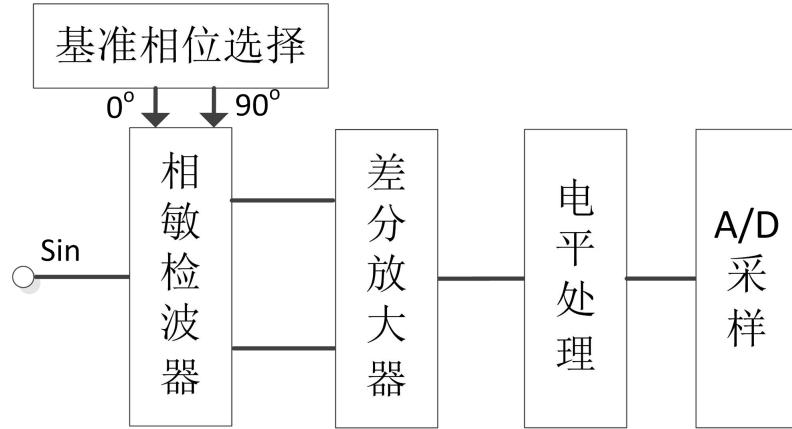


图 2.4 相敏检波模块原理框图

2.7 本章小结

本章主要讨论了系统各个模块的选择，然后通过论证、比较，得到了最佳设计方案。比如电源模块的选择，它的选择决定了系统稳定性，纹波较大的电源会直接影响测量的精度。控制模块的选择，它决定了整个系统在什么样的处理器下工作，处理器是一个系统的控制中心。信号源模块，测量模块，相敏检波模块它们的选择很重要，是整个系统的核心电路，决定了系统能不能达到想要的指标，能不能符合设计的目的。

3 系统原理及理论分析

3.1 自由轴测量法原理

如图 3.1 所示， R_0 为信号源内阻， R_S 是标准电阻， Z_X 为被测阻抗，利用一个集成运算放大器组成反相输入比例运算电路。测量时，开关 S 通过程控可以分别置于 U_X 或 U_S 端。由图可知被测阻抗 Z_X ：

$$Z_X = -\frac{U_X}{U_S} Z_S \quad (3.1)$$

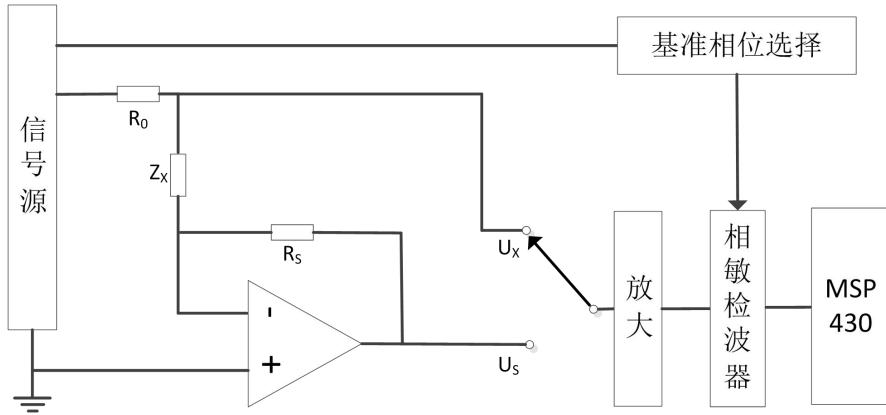


图 3.1 网络测试原理框图

由式(3.1)可知，只要测出 U_x , U_s 在直角坐标系中两坐标轴 X, Y 上的投影分量，经过四则运算，即可求出测量结果。

图 3.2 中，被测信号与相位参考基准信号经过相敏检波器后，输出就是被测信号在坐标轴上的投影分量。相位参考基准代表着坐标轴的方向，为了得到每一被测电压(U_s 或 U_x)在两坐标轴上的投影分量，基准相位发生器需要提供两个相位相差 90° 的相位参考基准信号。需要指出的是在自由轴法中，相位参考基准与 U_s 没有确定关系，可以任意选择，即 X, Y 坐标轴可以任意选择，只需保持两坐标轴准确正交 90° 。 U_x , U_s 和坐标轴的关系如图 3.2 所示。

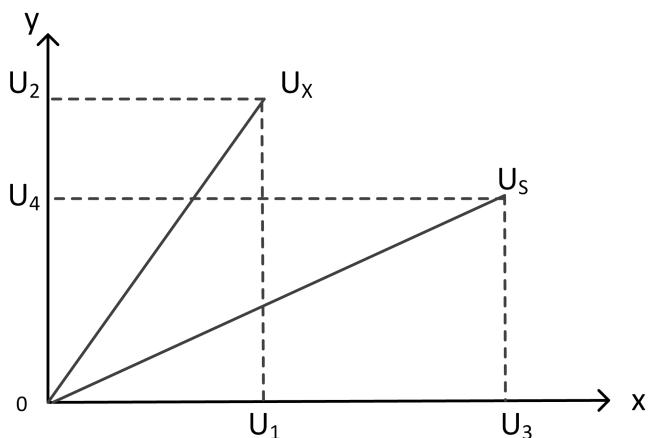


图 3.2 自由轴法矢量图

应用图 3.1 测量时，通过开关 S 选择某一被测量（如 U_x ），基准相位发生器依次送出两个相位相差 90° 的相位参考基准信号，经相敏检波器后分别得到 U_x 在两坐标轴上的投影分量 U_1, U_2 。类似，当开关 S 选择 U_s 时，可分别得到 U_s 在两坐标轴上的投影分量 U_3, U_4 。各投影分量经 A / D 转换器可得对应的数字量，再经微处理器计算便得到被测元件参数值。

若以电阻、电容、电感串联接入，来推导数学模型。由图 3.2 可得到：

$$U_x = U_1 + jU_2 = eN_1 + jeN_2 \quad (3.2)$$

$$U_s = U_3 + jU_4 = eN_3 + jeN_4 \quad (3.3)$$

式中： N_i 为 U_i 对应的数字量， e 为 A / D 转换器的刻度系数，即每个数字所代表的电压值。由式(3.2)，式(3.3)可知：

$$\frac{U_x}{U_s} = \frac{N_1 + jN_2}{N_3 + jN_4} = \frac{N_1N_3 + N_2N_4}{N_3^2 + N_4^2} + j \frac{N_2N_3 - N_1N_4}{N_3^2 + N_4^2} \quad (3.4)$$

直接通过对 $N_1 \sim N_4$ 数值的运算，即可完成矢量除法运算。

由表达式我们可以得到：

$$Z_x = -\frac{U_x}{U_s} Z_s = -\frac{N_1N_3 + N_2N_4}{N_3^2 + N_4^2} Z_s - j \frac{N_2N_3 - N_1N_4}{N_3^2 + N_4^2} Z_s \quad (3.5)$$

所以对于阻抗模 $|A| = \sqrt{a^2 + b^2}$ ，对于阻抗角 $\Phi = \arctan(\frac{b}{a})$

(其中 $a = -\frac{N_1N_3 + N_2N_4}{N_3^2 + N_4^2} Z_s$, $b = -\frac{N_2N_3 - N_1N_4}{N_3^2 + N_4^2} Z_s$)

3.2 相敏检波原理

相敏检波有乘法型 D/A 鉴相和开关鉴相两种方案，由于鉴相前端的直流容易产生漂移，在乘法器中直流也会产生漂移，造成噪声的处理和滤波都很麻烦，所以选择开关鉴相的方案，开关鉴相的等效原理框图如图 3.3 所示。

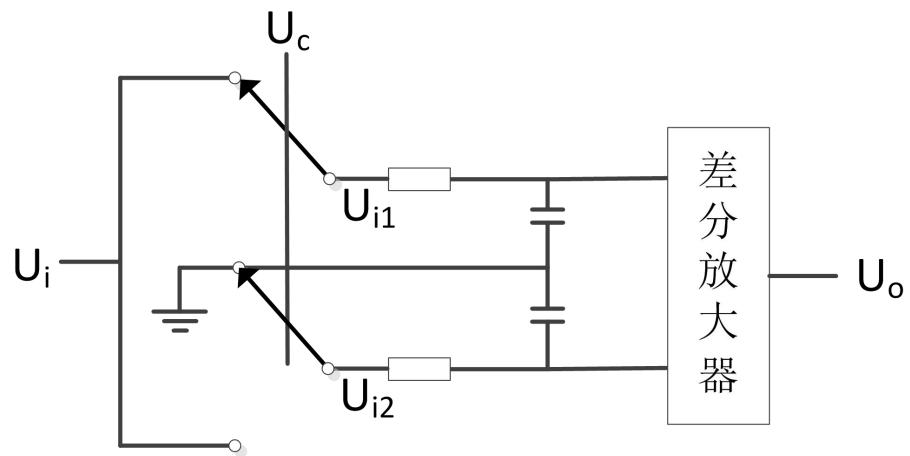


图 3.3 开关鉴相的等效原理框图

在原理框图中， U_i 是输入的待测信号， U_o 是相敏检波后的输出信号， U_c 是单刀双闸开关的控制信号，当 U_c 为高电平时选择上端， U_c 为低电平时选择下端。电路的基本原理是通过单刀双闸开关将待测信号 U_i 分成两路，再分别通过低通滤波器，滤除交流分量，保留直流分量，得到的直流信号通过差分放大器将两路信号合并。其中相位 $\theta=0^\circ$ 的 U_c 控制信号输出波形图，如图 3.4 所示。其中相位 $\theta=90^\circ$ 的 U_c 控制信号波形图，如图 3.5 所示。

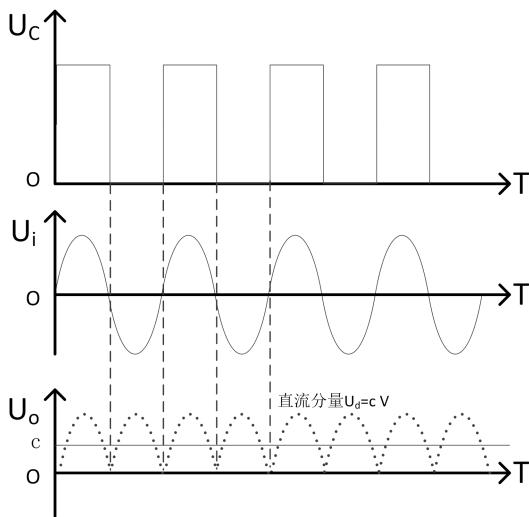


图 3.4 相位 $\theta = 0^\circ$ 的 U_c 控制信号输出波形图

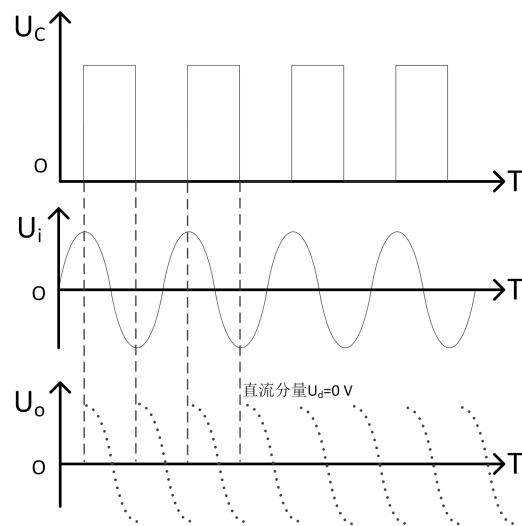


图 3.5 相位 $\theta = 90^\circ$ 的 U_c 控制信号输出波形图

在图 3.4 和图 3.5 中，相位 $\theta=0^\circ$ 的波形图的输出 U_o 代表待测信号在 X 坐标轴上的投影 U_1 ，也就是该信号的实部，相位 $\theta=90^\circ$ 的波形图的输出 U_o 代表待测信号在 Y 坐标轴上的投影 U_2 ，也就是该信号的虚部。可以通过勾股定理计算该待测信号 U_i 的幅度 U_X 。

从图 3.4 和图 3.5 中，我们可以得到其直流分量的理论计算值，图 3.4 中 U_o 的虚线部分代表着该直流分量通过低通滤波器之前所对应的波形，该波形就是周期全波正弦信号，它的傅立叶级数为：

$$f(t) = \frac{2E}{\pi} + \frac{4E}{3\pi} \cos(\omega_1 t) - \frac{4E}{15\pi} \cos(2\omega_1 t) + \frac{4E}{35\pi} \cos(3\omega_1 t) - \dots$$

周期全波正弦信号的直流分量 $c = \frac{2E}{\pi}$ V, 那么该周期全波正弦信号通过截止频率 $\omega_c < \omega_1$ 的 RC 低通滤波器之后就可以得到其直流分量了。图 3.5 中 U_O 的虚线部分代表着该直流分量通过低通滤波器之前所对应的波形, 可以看出该波形是一个奇函数, 根据奇函数的性质, 它的直流分量 $c=0$ V。

3.3 本章小结

本章主要介绍了本系统的两个主要原理, 分别为自由轴测量法原理和相敏检波原理, 从理论上论证了该方案的可行性, 为设计提供了保障, 自由轴测量法的核心是公式的推导, 在软件计算部分还需要利用这些推导的公式来得到计算的结果。相敏检波主要是在硬件电路上实现对信号的一些相关处理, 最终得到需要 A/D 采样的电压值。通过对这两个理论的分析, 就可以很好的理解本设计的各个组成模块和要点。

4 硬件设计

4.1 系统总体设计

本系统是由信号源模块、测试模块、相敏检波模块、键盘显示模块、供电模块以及主控制模块构成, 并以 MSP430G2553 单片机为核 心控制芯片。如图 4.1 所示。

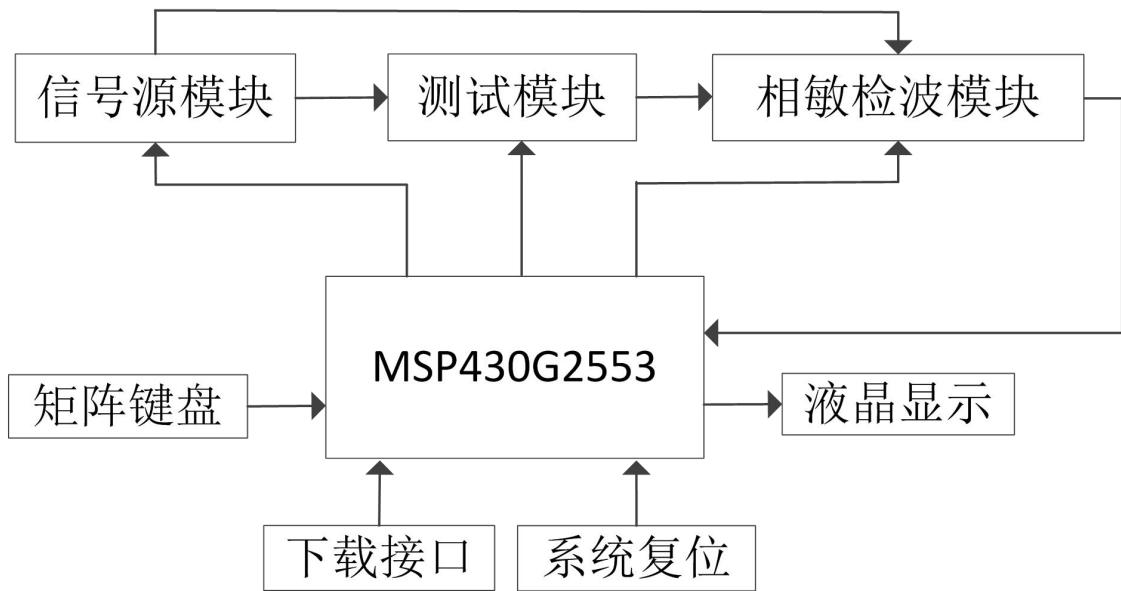


图 4.1 系统总设计原理框图

MSP430G2553 单片机控制信号源模块产生一路正弦波作为测试用信号，两路相互正交的方波作为相敏检波模块的控制信号。单片机还控制测试模块中称电阻的档位选择以及测试模块中放大信号和放大倍数的选择，通过对不同档位的选择可以实现量程的转换，通过对放大信号的选择可以分别测的标称电阻上的信号和待测元件上的信号，通过对信号放大倍数的调整可以使单片机 A/D 采样值始终保持在最佳采样的范围内。在相敏检波模块中，单片机控制选择相位 $\theta=0^\circ$ 和 $\theta=90^\circ$ 的方波，分别测量信号的实部和虚部，最后通过 A/D 采样输出的电压值。矩阵键盘和液晶作为人机交互设备。

4.2 电源模块

本系统供电模块使用三端稳压集成芯片作为电源模块^[2]，能输出可靠，有效的电压，并且有效的减少纹波。由图 4.2 得知，电路由变压器、桥式整流器、滤波电容和稳压芯片、LED 显示灯等部分组成。

电压输出用到集成稳压芯片 LM7805、LM7905、AMS1117-3.3，其中 AMS1117-3.3 提供+3.3V 电压给单片机供电，LM7805 和 LM7905 提供+5V 和-5V 给系统其他各个模块供电。其中为了减少数字电路对模拟电路的影响，我们将数字电源和模拟电路之间加了一个 π 型滤波电路，在数字地和模拟地之间加一个 0 欧姆电阻隔离^[3]，这些元件的加入可以在很大程度上提高电路的稳定性，减少干扰信号。

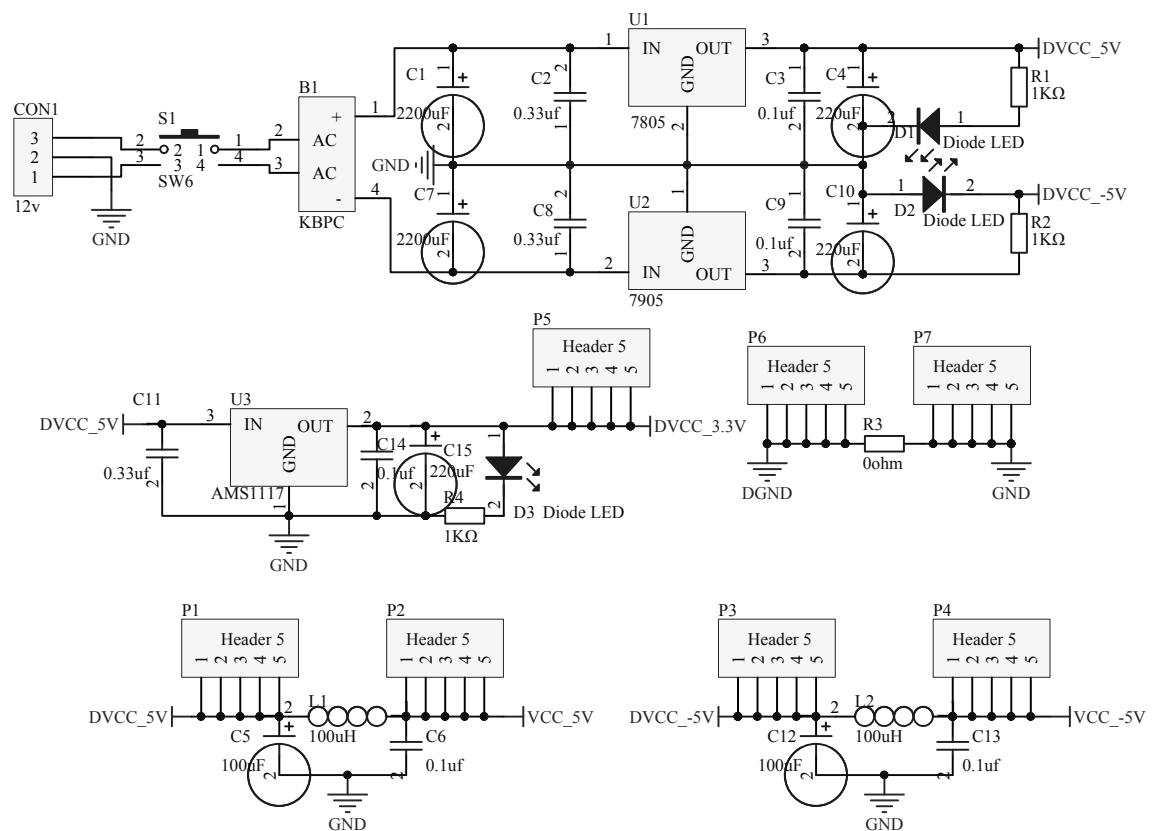


图 4.2 供电模块电路原理图

4.3 主控制模块

采用以 MSP430G2553 单片机为核心的最小系统方案，德州仪器 (TI) MSP430 系列超低功耗微控制器包含多种器件，它们特有面向多种应用的不同外设集。这种架构与 5 种低功耗模式相组合，专为在便携式测量应用中延长电池使用寿命而优化。该器件具有一个强大的

16 位 RISC CPU，16 位寄存器和有助于获得最大编码效率的常数发生器。数字控制振荡器(DCO)可在不到 $1\mu\text{s}$ 的时间里完成从低功耗模式至运行模式的唤醒。MSP430G2553 系列是超低功耗混合信号微控制器，具有内置的 16 位定时器、一个用途模拟比较器以及采用通用串行通信接口的内置通信能力。此外，MSP430G2553 还具有一个 10 位模数(A/D)转换器。

单片机外围组成还包括：一个晶振，两个按键，两个 LED 灯以及一些芯片的 I/O 引脚接口，如图 4.3 所示。

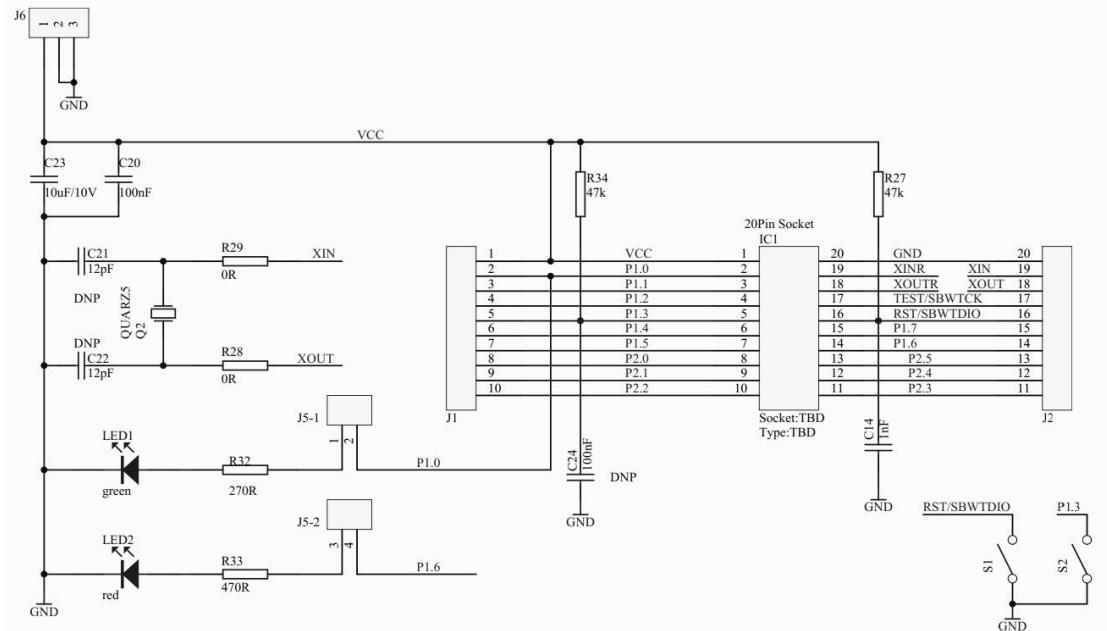


图 4.3 单片机最小系统电路图

4.4 按键显示模块

如下图 4.4 所示，键盘采用 4×4 矩阵键盘，易于实现各种功能的控制，外加一块矩阵按键驱动芯片 CH452，CH452 是数码管显示驱动和键盘扫描控制芯片。CH452 内置时钟振荡电路，可以动态驱动 8 位数码管或者 64 只 LED，具有 BCD 译码、闪烁、移位、段位寻址、光柱译码等功能；同时还可以进行 64 键的键盘扫描；CH452 通过可

以级联的 4 线串行接口或者 2 线串行接口与单片机等交换数据；并且可以对单片机提供上电复位信号。

显示模块采用 LCD12864，液晶显示模块是 128×64 点阵的汉字图形型液晶显示模块，可显示汉字及图形，内置国标 GB2312 码简体中文字库（ 16×16 点阵）、128 个字符（ 8×16 点阵）及 64×256 点阵显示 RAM（GDRAM）。可与单片机串行连接节省 I/O 口资源。

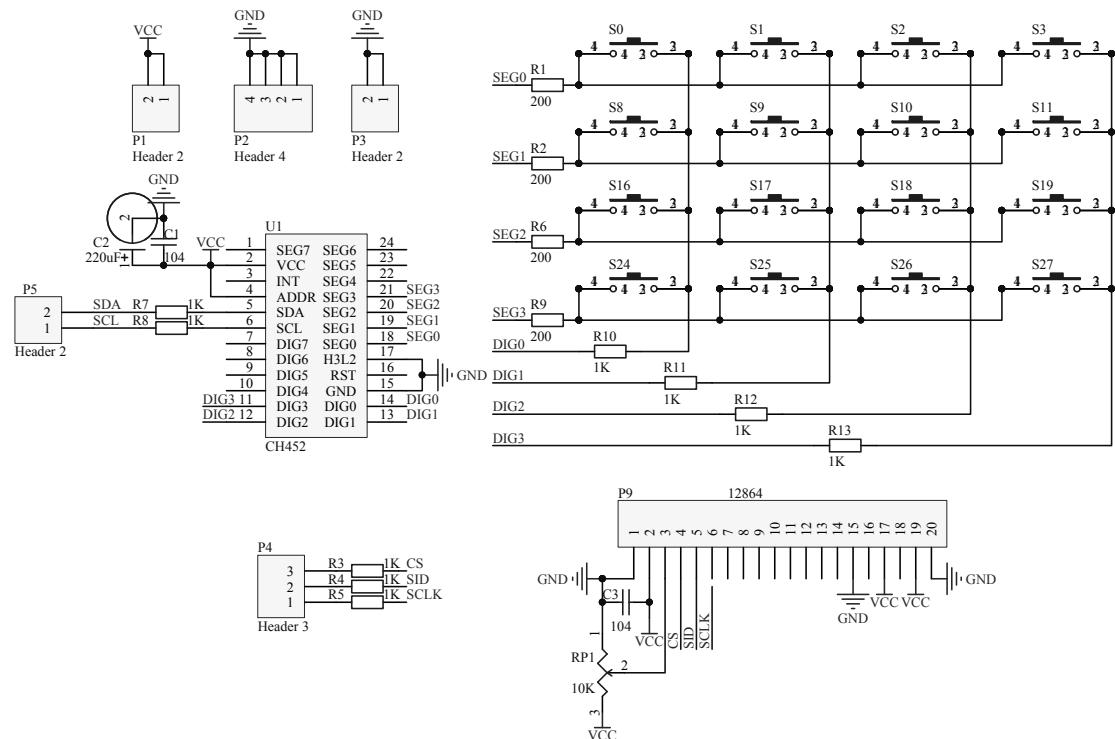


图 4.4 按键和显示模块电路图

4.5 信号源模块

信号源模块采用高集成度 DDS 频率合成器 AD9851 芯片实现波形的产生，AD9851 合成一路正弦波，再经过 LC 组成的 7 阶椭圆滤波器滤除高频干扰^[4]，7 阶椭圆滤波器的截止频率 $f_c=20\text{MHz}$ ，可以滤除晶体振荡器带来的高频干扰，7 阶椭圆滤波器的 Multisim12.0 仿真图，如下图 4.5 所示，再经过一个反相比例放大器实现对波形幅度的调整。

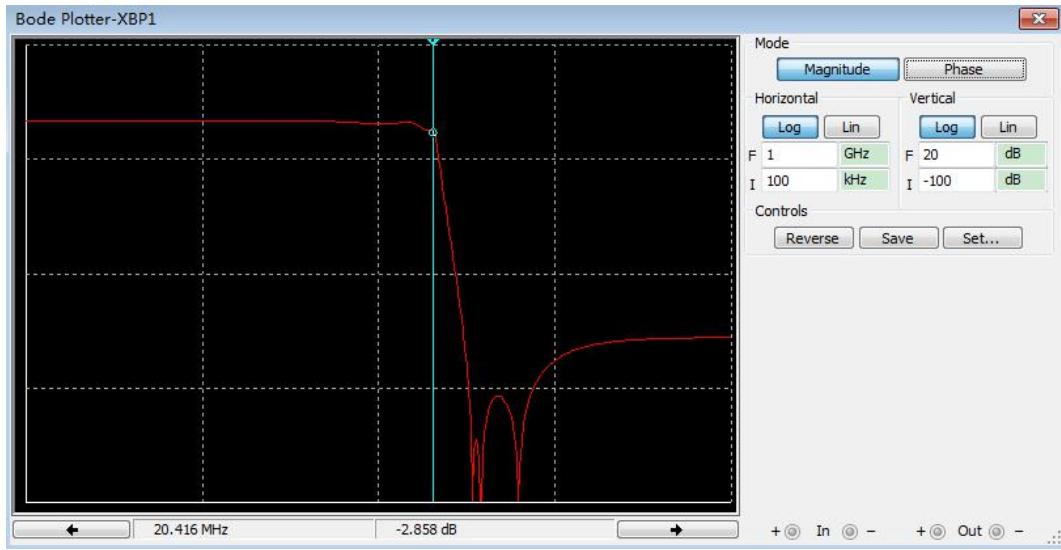


图 4.5 7 阶椭圆滤波器幅频特性图

正弦波再通过 AD9851 内部的比较器输出一个方波信号，该方波通过 CD74HC4046 集成锁相环芯片实现方波的 4 倍频，CD74HC4046 是带 VCO 的高速 CMOS 逻辑锁相环^[5]，用该芯片再加上一个外部的分频器就可以实现 PLL 锁相环倍频。然后再利用移位寄存器芯片 74HC164^[6]实现相互正交的两路方波输出，该相互正交的两路方波与合成的正弦波同频。一路方波产生两路相互正交方波的 Multisim12.0 仿真图如下图 4.6 所示。

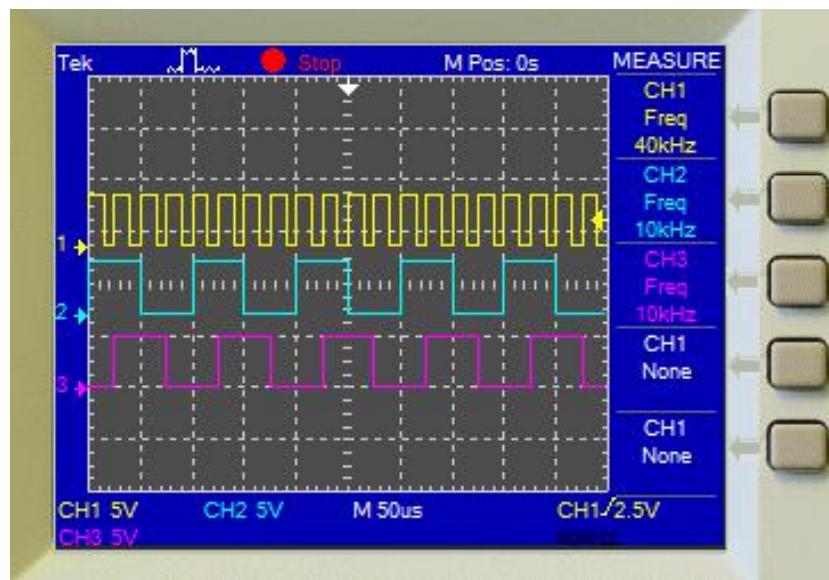


图 4.6 两路相互正交方波图

信号源模块的实际电路原理图 4.7 如下图所示，分别用到了频率合成芯片 AD9851，运算放大器 OPA211，逻辑锁相环芯片 CD74HC4046，双十进制计数器 74LS390，移位寄存器芯片 74HC164。

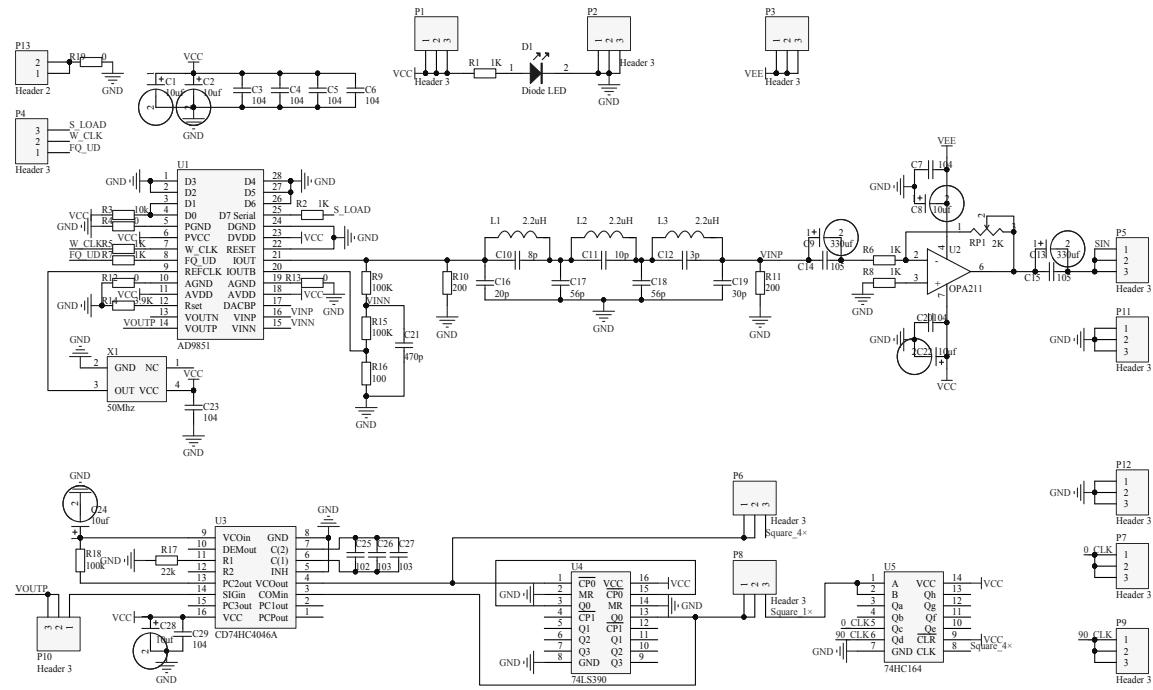


图 4.7 信号源模块电路图

4.6 测试模块

测试模块作用是分别对被测件的电压 U_x 及标称电阻上的电压 U_s 进行处理，以便更适合后级电路的工作，分别做了档位选择和幅度的放大的处理。本设计的测试电路如图 4.8 所示，它由集成运算放大器、模拟开关和仪表放大器组成。

测量时，先通过程控使开关接到测量待测元件 Z_x 上的电压 U_x ，再通过程控使开关接到标称电阻上的电压 U_s 。其中标准电阻 R_9 、 R_{14} 、 R_{15} 、 R_{17} 用来改变量程。

U_x 、 U_s 分别被仪表放大器放大之后，便经程控开关送到输入运

算放大器放大，放大器的增益可以通过程控开关被置为1倍、2倍或者4倍^[7]。

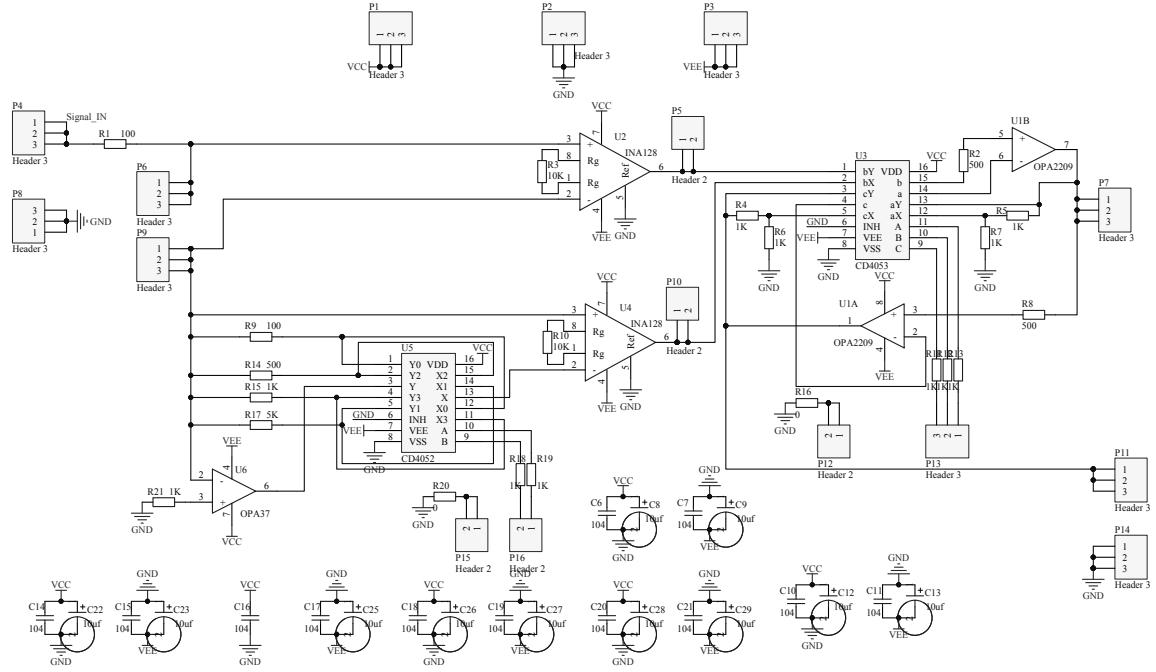


图 4.8 测试模块电路图

4.7 相敏检波模块

相敏检波的作用是将 U_x 、 U_s 用不同参考相位进行鉴相并将其平均值通过 A/D 转换成相应的数字量。相敏检波由模拟开关 CD4053 构成，控制模拟开关通断的信号就是相位 $\theta=0^\circ$ 或 $\theta=90^\circ$ 基准方波信号。相敏检波电路的输出是被测电压信号在坐标轴上的投影，当基准方波信号相位 $\theta=0^\circ$ 时，就是信号在实轴上的投影，当基准方波信号相位 $\theta=90^\circ$ 时，就是信号在虚轴上的投影。相位 $\theta=0^\circ$ 时的 Multisim12.0 仿真图如下图 4.6 所示，示波器第一通道代表 $\theta=0^\circ$ 的方波信号，第二通道代表相敏检波的一路输出，第三通道代表相敏检波的另一路输出。

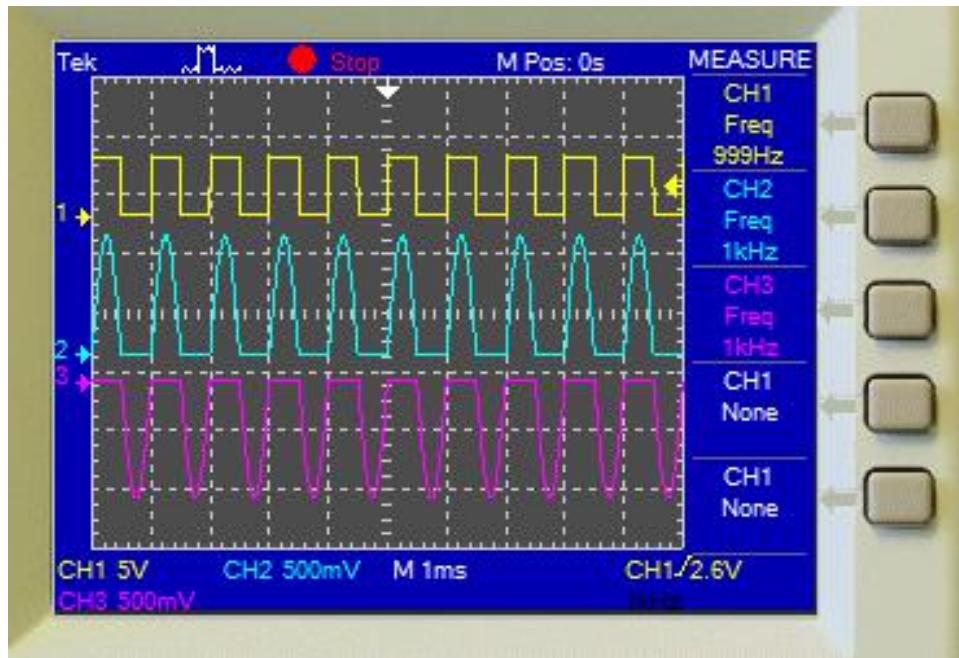


图 4.9 相位 $\theta = 0^\circ$ 时相敏检波输出波形图

这两路输出通过后面的 RC 低通滤波器，输出就得到相应的直流分量，两路直流分量通过差分放大器就能够得到被测电压在坐标轴上的投影值。单片机内部 A/D 对最后所得的信号进行采样，可得到被测阻抗两端电压 U_x 分别在 X 轴和 Y 轴的投影分量以及标称电阻两端电压 U_s 分别在 X 轴和 Y 轴的投影分量。

因为差分放大器输出的电压是有极性的，有可能产生负电压，所以最后加了一级加法器，单片机的有效采样范围是 0~3.3V，而差分放大器的输出电压范围为 -1.2 ~ +1.2V，所以利用加法器将电平抬升 1.5V，那么最后 A/D 采样的范围就是 0.3~2.7V，在这个范围内单片机内部 A/D 芯片能够采样正确。相敏检波模块的硬件电路图，如下图 4.10 所示。

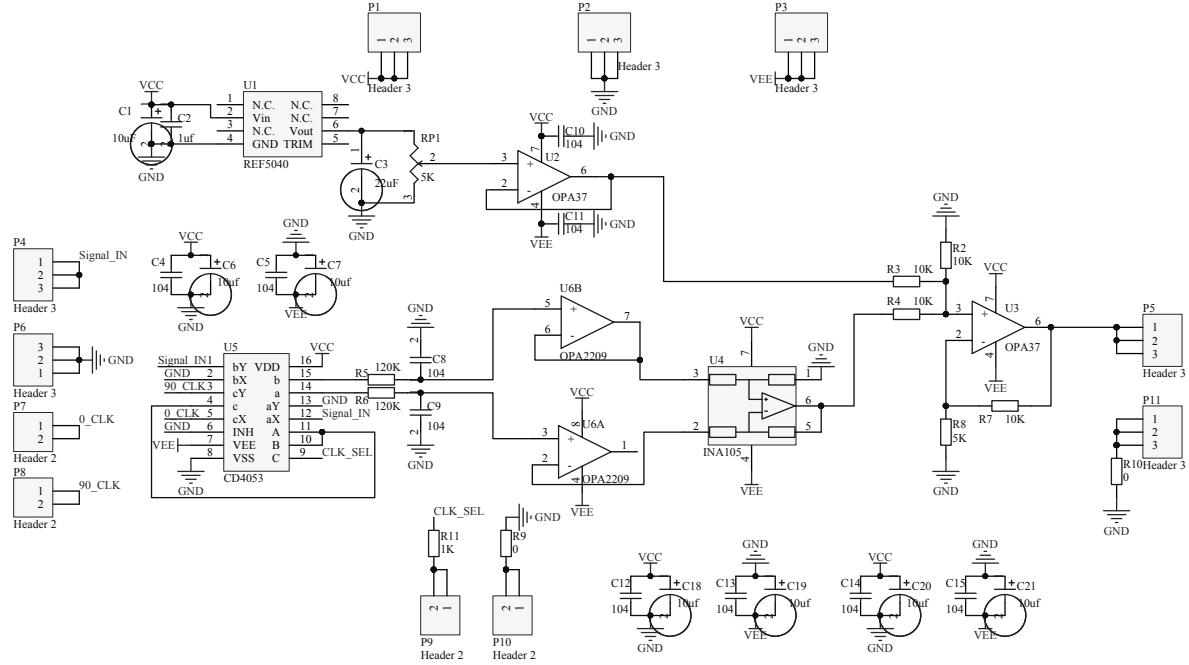


图 4.10 相敏检波模块电路图

4.8 本章小结

本章主要讲解了系统的各个部分硬件电路的设计以及具体的电路原理图，首先讲了电源模块，电源模块是一个系统最重要的部分，同时也决定了系统的稳定性。然后就是单片机最小系统的硬件电路图，采用 TI 公司的 MSP430 系列单片机，该单片机的硬件电路比较简单。还有人机交互的按键和显示模块电路图^[8]，4*4 矩阵键盘和 12864 液晶方便的实现了交互的功能。最后就是本系统用于实现功能的三个重要模块，分别为信号源模块硬件电路、测试模块硬件电路、相敏检波模块硬件电路，还具体讲解了一些关于实现功能的原理仿真，充分利用 Multisim12.0 仿真软件为设计做前期的模拟仿真准备，防止一些理论计算的疏忽而导致的错误。

5 软件设计

5.1 系统设计流程图

软件程序实现整个系统的调控，系统总流程图如图5.1所示：

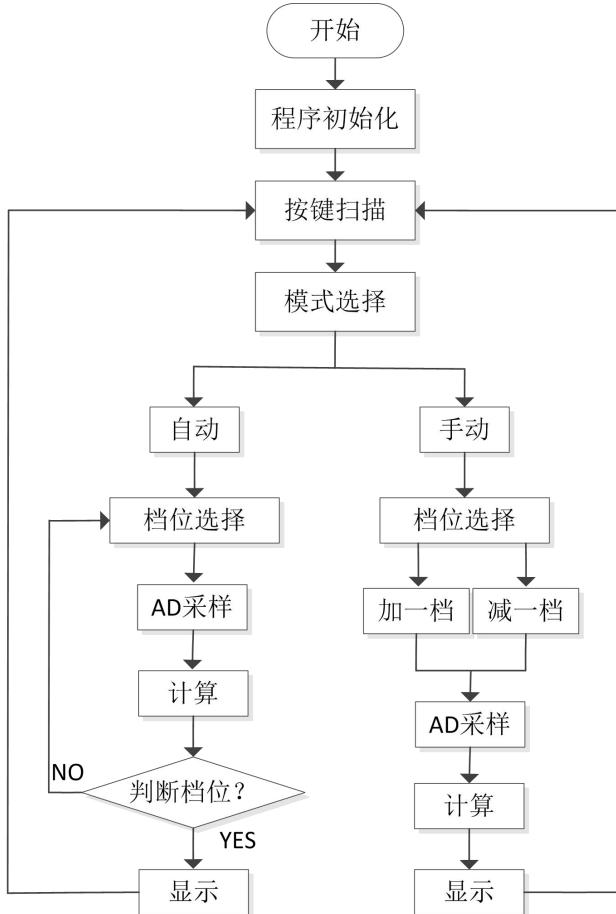


图 5.1 系统软件控制流程图

函数从主函数开始，将程序初始化，比如 MSP430 单片机时钟初始化、液晶显示初始值、按键初始化等等，之后进入按键扫描，如果没有按键值就进入默认设置，有按键值按下，就进行模式的选择，分为自动模式和手动模式。在自动模式中，首先有个默认档位选择，然后是 AD 的采样和计算，如果计算值超过相应的量程，就进行档位切换，直到切换到合适的档位上，如果正确就在液晶上进行显示计算得到的数值，最后程序返回按键扫描进行循环。在手动模式中，档位的

选择是人工控制的，通过按键加一档或者减一档，然后进行 AD 采样计算和显示数值，最后程序返回按键扫描进行循环^[9]。

5.2 系统子程序说明

本程序采用了模块化的思想，除了主程序之外，还分为四个应用程序，分别为 AD9851 转换程序，CH452 转换程序，单片机时钟选择程序，液晶显示程序。

5.2.1 AD9851 转换程序

AD9851 转换程序是根据芯片 AD9851 的时序图进行编写的。如图 5.2 所示，开始变量定义与赋值，再进行串行驱动的初始化，然后写频率控制字、写命令控制一些内部寄存器的操作、写相位控制输出相位。

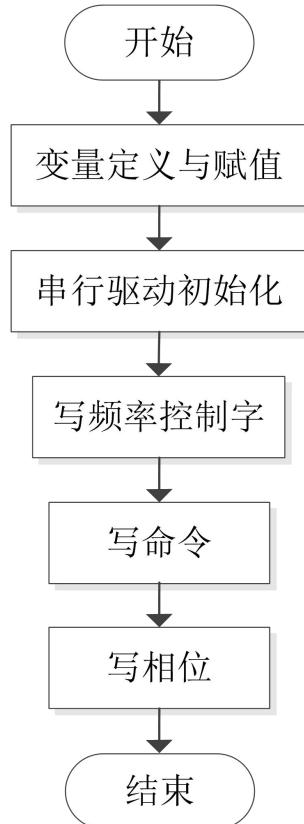


图 5.2 AD9851 转换程序流程图

5.2.2 CH452 转换程序

键盘程序是人机沟通的关键，如图 5.3 所示，矩阵按键驱动芯片 CH452 的驱动程序，开始变量定义与赋值，然后就是启动总线实现通信，接着读取数据，结束总线，最后返回按键值^[10]。

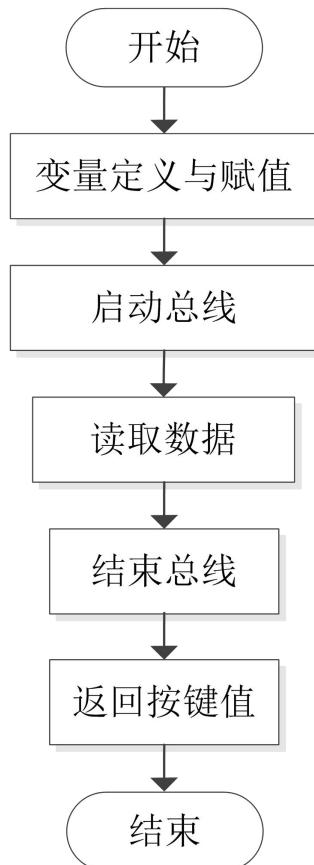


图 5.3 CH452 转换程序流程图

5.2.3 单片机时钟选择程序

如图 5.4 所示，单片机时钟选择程序，主要是配置一些单片机内部寄存器，实现单片机时钟频率的选择，开始变量定义与赋值，然后设定芯片内部振荡器频率，然后就是辅助时钟的设定，它是为外围模块提供低频时钟的信号，再是子时钟的设定，它是为外围模块提供高频时钟的信号，最后是主时钟的设定，为 CPU 和系统提供时钟。

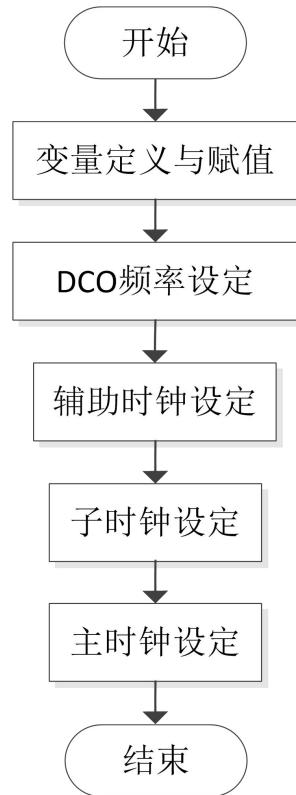


图 5.4 单片机时钟选择程序流程图

5.2.4 液晶显示程序

如图 5.4 所示，显示包涵数据的初始化显示和实时的测量数值显示，开始变量定义与赋值，然后写命令控制，最后写入需要显示的初始化数据^[11]。

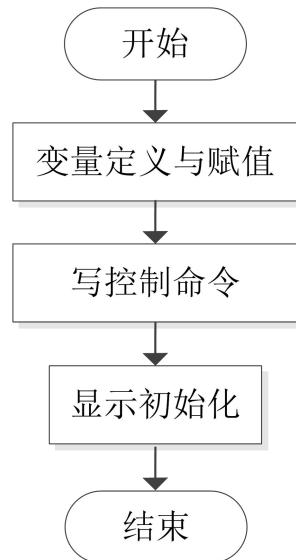


图 5.4 LCD 显示流程图

5.3 本章小结

本章结合系统总设计原理，给出总设计软件流程图。并对此流程图进行步骤解释说明，然后分模块具体介绍各部分流程图，重点介绍了 AD9851 转换程序，CH452 转换程序，单片机时钟选择程序，液晶显示程序，使总体框架更加清晰明了。

6 数据测试及分析

6.1 测试内容与使用仪器

数据测试是反映系统性能的重要指标。因此本测试进行了全面的测试，测试信号源输出频率及幅度，输出待测网络阻抗模及阻抗角，判断网络结构，测量谐振频率点。

本系统测试采用的仪表如下：Tektronix 数字示波器、数字万用表、高频导纳电桥

6.2 测试数据结果

根据题目要求，我们进行了以下测试：

表 6.1 信号源频率和幅值测试表

频率 / KHz	理论值	1	2	5	10	20	50	80	100
	测量值	1.00	2.00	5.00	10.00	20.00	50.01	80.00	100.05
幅值 / V	理论值	1	1	1	1	1	1	1	1
	测量值	1.006	1.000	1.005	1.004	1.001	0.983	1.005	1.000

表 6.2 $|Z|$ 和 ϕ 的测试测试表

(1) R、L 串联 $R=1K$ $L=10mH$

频率 / KHz		1	10	50	100	200
$ Z /\Omega$	理论值	1002	1181	3296	6362	12606
	测量值	1000	1175	3300	6358	12596
ϕ	理论值	3.89	7.16	72.33	80.95	85.45
	测量值	5.00	8.20	82.00	90.00	90.00

(2) R、C 串联 $R=500\Omega$ $C=20nF$

频率 / KHz		1	10	50	100	200
Z / Ω	理论值	7973	939	524	506	501
	测量值	7969	946	530	512	511
Φ	理论值	-86.40	-78.68	-17.67	-9.05	-4.55
	测量值	-95.00	-88.88	-19.99	-1.33	-1.55

(3) L、C 串联 $L=0.5mH$ $C=10nF$

频率 / KHz		1	10	50	100	200
Z / Ω	理论值	15912	1560	161	155	549
	测量值	15922	1569	169	165	558
Φ	理论值	-90.00	-90.00	-90.00	-90.00	90.00
	测量值	-80.00	-83.66	-79.99	-80.99	85.02

表 6.3 网络结构的测试

		网络 1	网络 2	网络 3	网络 4	网络 5	网络 6
R	理论值	500 Ω	1K	/	500 Ω	1K	/
	测量值	478 Ω	990 Ω	/	489 Ω	991 Ω	/
L	理论值	/	10mH	0.5mH	/	20mH	10mH
	测量值	/	8mH	0.4mH	/	18mH	9mH
C	理论值	20nF	10nF	/	5nF	/	1nF
	测量值	18nF	9nF	/	4nF	/	1nF
串、并联	理论值	串	串	串	并	并	并
	测量值	串	串	串	并	并	并

6.3 数据误差分析

正弦波信号发生器的误差: AD9851 的工作频率比较高, 有噪声影响产生的信号。

幅值测量的误差: 电路本身有转换误差, 而且 A/D 采集后数据处理也有误差。

放大及滤波电路的误差: 放大器不是理想的运放, 滤波电路不可能滤去所有频段的干扰。

测量结果的误差: 由于器件本身的固定误差以及环境因素导致了结果产生的误差。

7 结论

在设计制作网络阻抗测试仪的过程中，我学到了许许多多的知识和以前没有接触到的东西。我深切体会到，实践是理论运用的最好检验。本次设计是对我们四年所学知识的一次综合性检测和考验，无论是动手能力还是理论知识运用能力都得到了提高，同时加深了我们对网络资源认识，大大提高了查阅资料的效率，使我们有充足的时间投入到电路设计当中。本系统的研制主要应用到了模拟电子技术、单片机控制技术、仪器仪表相关的工作原理、电子工艺等多方面的知识，所设计的基于单片机程序控制的网络阻抗测试仪，基本达到了题目要求。但是在过程中我也遇到了许许多多的问题，其中包括芯片的保护，PCB 的布线排版原则，电源方案的选择等等。在数据测试和调试方面，由于仪表存在误差和电路器件因工作时间过长而产生的误差，使得测量数据不是很精确，本系统通过多次测定和修正，最终达到了预期的目标。

参考文献

- [1] 全国大学生电子设计竞赛组委会. 第六届全国大学生电子设计竞赛作品选编. 北京：电子工业出版社，2007.
- [2] 高吉祥. 模拟电子技术（第二版）. 北京：电子工业出版社，2007.
- [3] 华成英，童诗白. 模拟电子技术基础. 北京：高等教育出版社，2006. 12.
- [4] 高吉祥. 模拟电子技术学习辅导及习题详解. 北京：电子工业出版社，2005. 1.
- [5] 高吉祥. 电子技术基础实验与课程设计（第二版）. 北京：电子工业出版社，2005. 2
- [6] 高吉祥. 数字电子技术（第二版）. 北京：电子工业出版社，2008. 1.
- [7] 高吉祥，唐朝京. 全国大学生电子设计竞赛培训系列教程—模拟电子线路设计. 北京：电子工业出版社，2007. 8
- [8] 潘永雄，沙 河. 电子线路 CAD 实用教程(第三版). 西安：西安电子科技大学出版

社. 2009. 7.

- [9] 谭浩强. C 程序设计 (第三版) . 北京: 清华大学出版社, 2007.
- [10] 谭浩强. C 程序设计题解与上机指导 (第三版) . 北京: 清华大学出版社, 2007.
- [11] 郭天祥. 新概念 51 单片机 C 语言教程. 北京: 电子工业出版社, 2009. 1.

致 谢

时光匆匆如流水，转眼便是大学毕业时节，春梦秋云，聚散真容易。离校日期已日趋临近，毕业论文的完成也随之进入了尾声。从开始进入课题到论文的顺利完成，一直都离不开老师、同学、朋友给我热情的帮助，在这里请接受我诚挚的谢意！

经过近几个月的毕业设计工作，我的论文和设计终于完成了。在设计网络阻抗测试仪的过程中，高吉祥教授对本设计的理论运用、报告撰写和数据测量等方面给予了诸多宝贵的意见，在此特别感谢我的导师高吉祥教授给予我的指导和帮助。老师在渊博的学识，学术上严谨求实，兢兢业业、一丝不苟的工作作风都是我将来走出校门学习的榜样。我可以感受到一个学者的严谨和务实，这些都让我获益匪浅，并且将终生受用无穷。毕竟“经师易得，人师难求”，希望借此机会向高教授表示最衷心的感谢！同时非常感谢期间帮助过我的同学们，在他们的帮助下，我们的设计才得以顺利完成。

在此，我再一次衷心的感谢所有帮助，关心和支持我本次设计的老师和同学们，你们不仅培养了我对本专业浓厚的兴趣，而且你们循循善诱的教导和不拘一格的思路给予我无尽的启迪，是你们让我懂得了什么叫“一份耕耘，一份收获”。在未来的日子里，我会更加努力的学习和工作，不辜负你们对我的殷切期望！

附录

附录 A 系统电路图

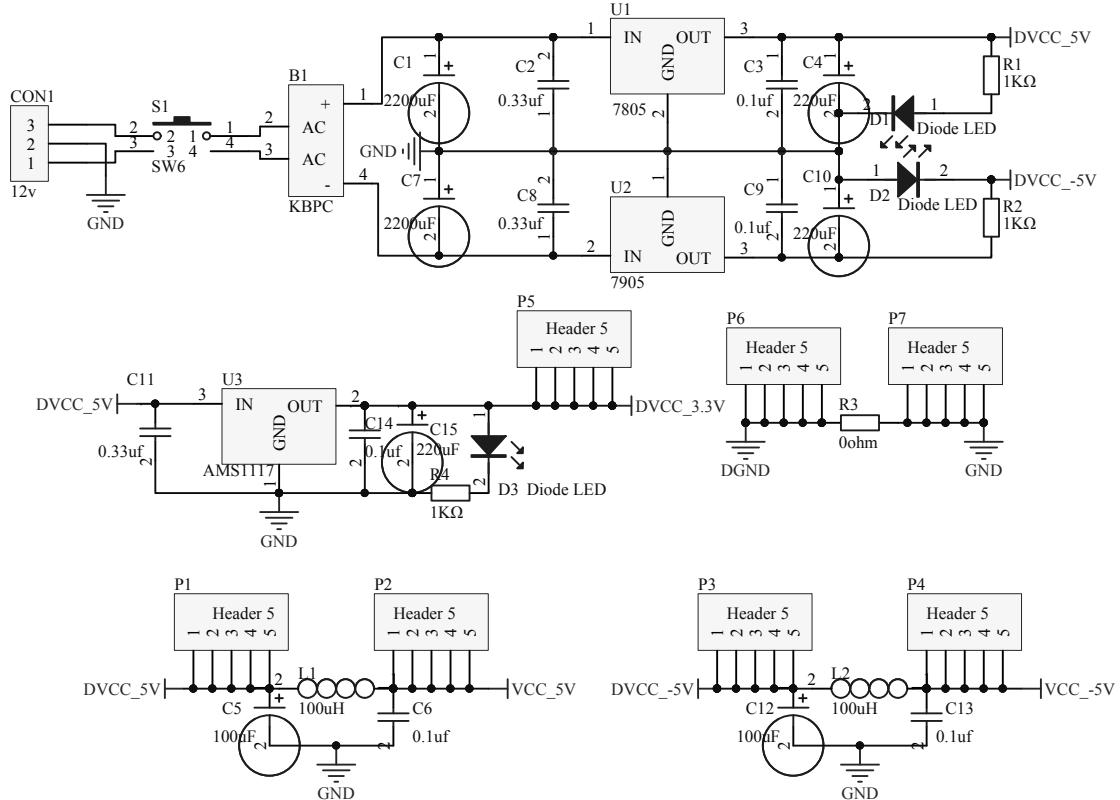


图 A1 供电模块电路图

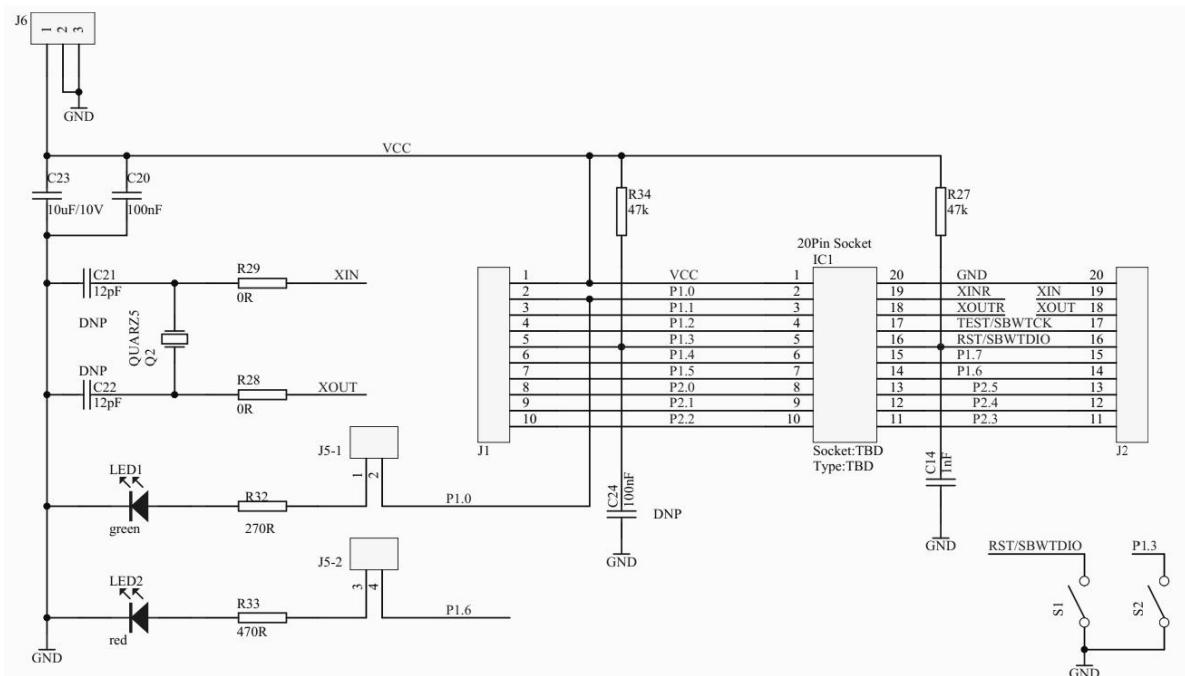


图 A2 控制模块电路图

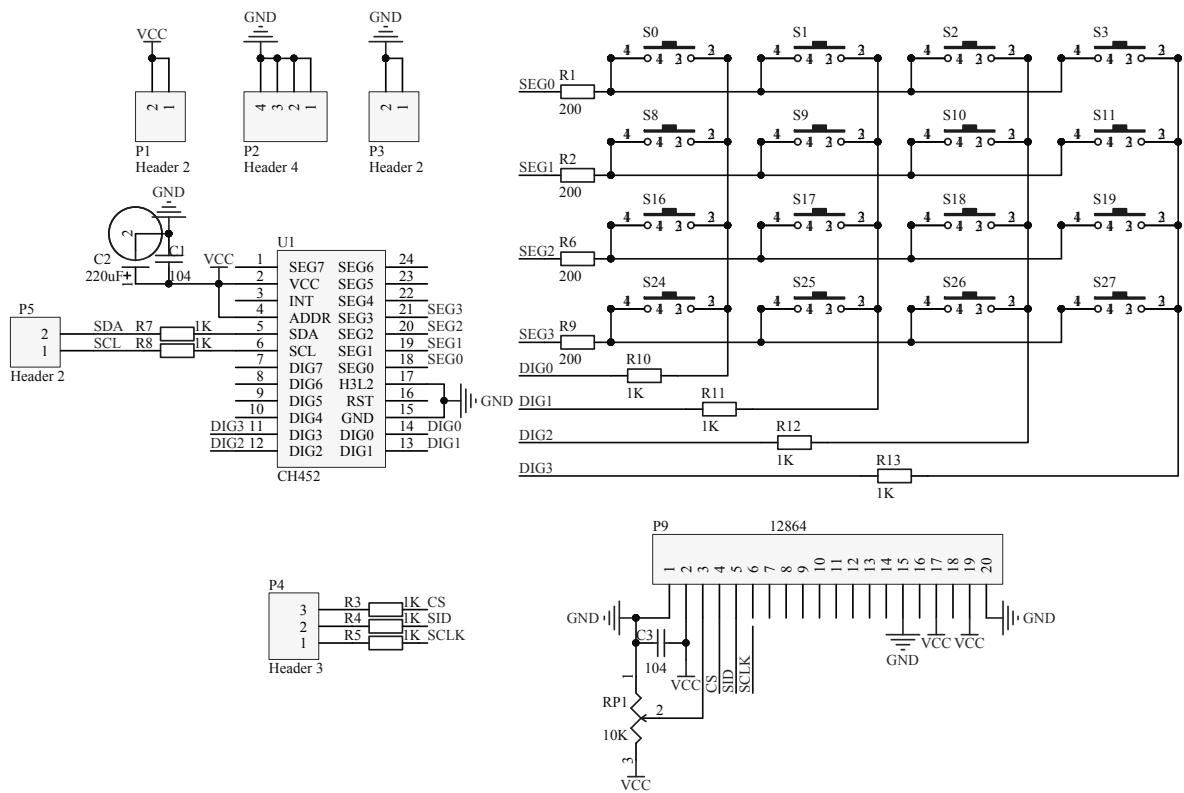


图 A3 按键和显示电路图

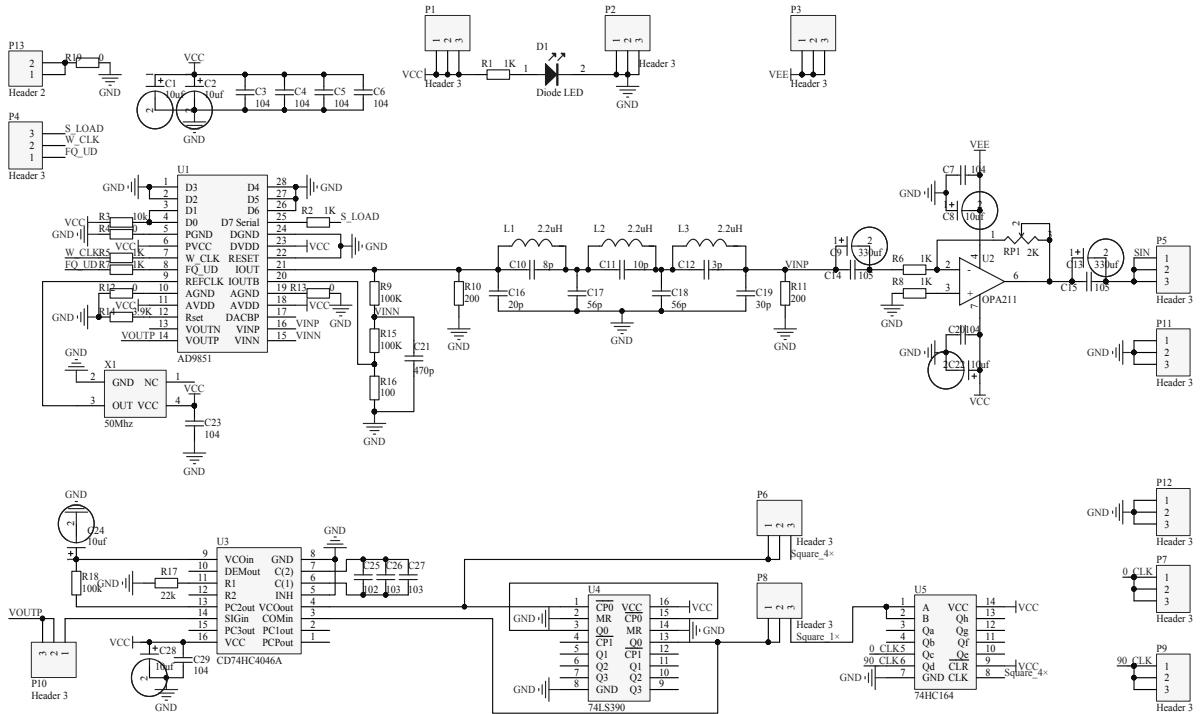


图 A4 信号源模块电路图

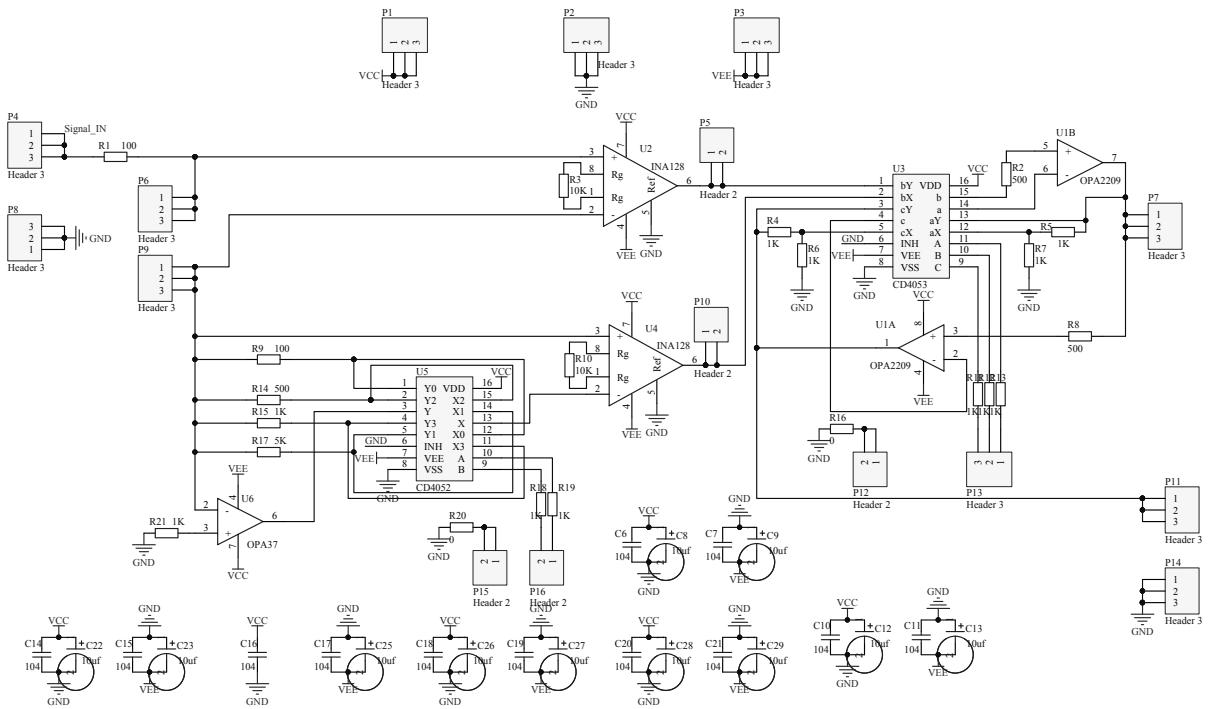


图 A5 测试模块电路图

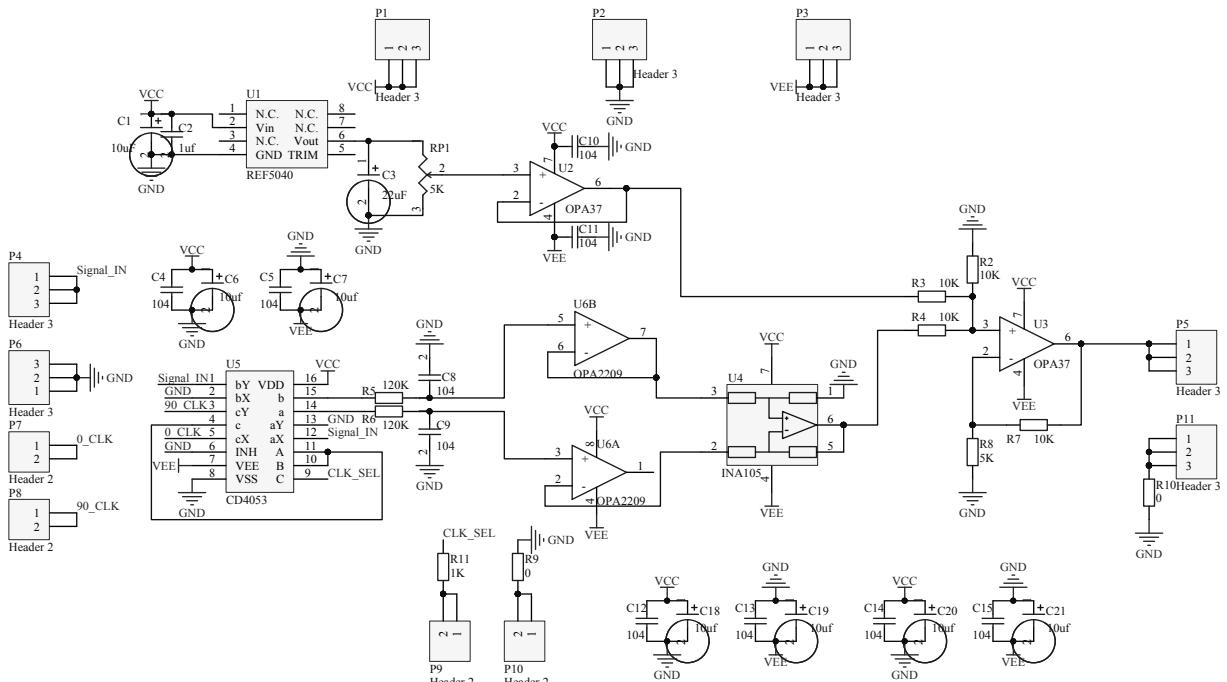


图 A6 相敏检波模块电路图

附录 B 程序清单

```
#include<msp430g2553.h>
#include<math.h>
#include"ClkInit.h"
#include"AD9851_Serial.h"
#include"show_12864.h"
#include"CH452.h"

#define KeyInvalid 16

#define Rs_Sel_A      BIT6
#define Rs_Sel_B      BIT7
#define Rs_Sel_A_1    P2OUT|=Rs_Sel_A
#define Rs_Sel_A_0    P2OUT&=~Rs_Sel_A
#define Rs_Sel_B_1    P2OUT|=Rs_Sel_B
#define Rs_Sel_B_0    P2OUT&=~Rs_Sel_B
#define Rs_Sel_100   Rs_Sel_A_0;Rs_Sel_B_0      //1
#define Rs_Sel_500   Rs_Sel_A_0;Rs_Sel_B_1      //2
#define Rs_Sel_1K    Rs_Sel_A_1;Rs_Sel_B_1      //3
#define Rs_Sel_5K    Rs_Sel_A_1;Rs_Sel_B_0      //4

#define Gain_Sel_A    BIT0
#define Gain_Sel_B    BIT1
#define Gain_Sel_C    BIT2
#define Gain_Sel_A_1  P2OUT|=Gain_Sel_A
#define Gain_Sel_A_0  P2OUT&=~Gain_Sel_A
#define Gain_Sel_B_1  P2OUT|=Gain_Sel_B
#define Gain_Sel_B_0  P2OUT&=~Gain_Sel_B
#define Gain_Sel_C_1  P2OUT|=Gain_Sel_C
#define Gain_Sel_C_0  P2OUT&=~Gain_Sel_C
#define Gain_Sel_Us   Gain_Sel_B_0
#define Gain_Sel_Ux   Gain_Sel_B_1
#define Gain_Sel_1X   Gain_Sel_A_1;Gain_Sel_C_1  //5
#define Gain_Sel_2X   Gain_Sel_A_1;Gain_Sel_C_0  //6
#define Gain_Sel_4X   Gain_Sel_A_0;Gain_Sel_C_0  //7

//测实部为 0 度， 测虚部为 90 度
#define R_I_Sel      BIT5
#define R_I_Sel_1    P2OUT|=R_I_Sel
#define R_I_Sel_0    P2OUT&=~R_I_Sel
#define R_I_Sel_R    R_I_Sel_0
#define R_I_Sel_I    R_I_Sel_1
```

```

#define Test_Ux_R      Gain_Sel_Ux;R_I_Sel_R
#define Test_Ux_I      Gain_Sel_Ux;R_I_Sel_I
#define Test_Us_R      Gain_Sel_Us;R_I_Sel_R
#define Test_Us_I      Gain_Sel_Us;R_I_Sel_I

unsigned char volatile KeyWord=16;

void LCDDisplayAuto()
{
    _DINT();
    show_12864_xy(2,0);
    printf("F:          Auto");
    show_12864_xy(3,0);
    printf("A:          Ω");
    show_12864_xy(3,5);
    printf("Φ:+00");
    show_12864_xy(4,0);
    printf("R:          X:          ");
    _EINT();
}

void LCDDisplayHand()
{
    _DINT();
    show_12864_xy(2,0);
    printf("F:          Hand");
    show_12864_xy(3,0);
    printf("A:          Ω");
    show_12864_xy(3,5);
    printf("Φ:+00");
    show_12864_xy(4,0);
    printf("R:          X:          ");
    _EINT();
}

void ADC10_Init()
{
    ADC10CTL0  =  SREF_0  +  ADC10SHT_2  +  ADC10ON;//Ref=VCC,times sample
(16×ADC10CLKs)
    ADC10CTL1 = INCH_7;                      // input A7 , P1.7
    ADC10AE0 |= BIT7;                       // PA.7 ADC option select
}

void TestInit()
{

```

```

    Rs_Sel_1K;
    Gain_Sel_1X;
    Test_Ux_R;
}

void main(void)
{
    unsigned long int Freq=1000;
    unsigned char LCDMode = 0, n=0,Freq_temp=0,Range=0;
    unsigned long int number[6]={0,0,2,0,0,0};

    double Ux_R_N=0,Ux_I_N=0,Us_R_N=0,Us_I_N=0,R=0,X=0,Imp_A=0,Imp_O=0;
    float ADC10_Val;
    unsigned int Timer_num=0;
    float Vout_Pre=0,Vout_Cur=0;
    unsigned char N_num=0;

    WDTCTL = WDTPW +WDTHOLD;           // Stop Watchdog Timer
    P2SEL &=~BIT6+~BIT7;
    P1DIR = 0xFF;
    P2DIR = 0xFF;

    Init_Clk();
    show_12864_init();
    InitKey();
    TestInit();
    AD9851_set_serial(); //串行写入模式
    WriteData(85899,0); //默认 1Khz
    ADC10_Init();
    while(1)
    {
        if(KeyWord == 15)           //模式切换!
        {
            LCDMode=~LCDMode;
            if(LCDMode)LCDDisplayAuto(); //LCDMode==1 自动
            else LCDDisplayHand(); //LCDMode==0 手动

            KeyWord = KeyInvalid;
        }

        if(LCDMode==0 && KeyWord<=14)//"10" 代表小数点; "11"代表清除;
        {
            if(KeyWord!=12 && (n==0||KeyWord==11) && KeyWord!=10)//显示清零
            {

```

```

show_12864_xy(2,1);printf("      ");show_12864_xy(2,1);
number[0]=0;number[1]=0;number[2]=0;number[3]=0;number[4]=0;number[5]=0;
n=0;
}

if(KeyWord<10)                                //频率输入
{
    number[n] = KeyWord;
    show_12864_com(1,KeyWord+48);
    n++;
    if(n==6)n=0;
    Freq_temp++;
    if(Freq_temp==7)Freq_temp=1;
}

if(KeyWord==12)                                //确定
{
    switch(Freq_temp)
    {
        case 0:
        case 1:
        case 2:
        case 3:break;
        case 4:
        {
            Freq=number[0]*1000+number[1]*100+number[2]*10+number[3];
            break;
        }
        case 5:
        {
            Freq=number[0]*10000+number[1]*1000+number[2]*100+number[3]*10+number[4];
            break;
        }
        case 6:
        {
            Freq=number[0]*100000+number[1]*10000+number[2]*1000+number[3]*100+number[4]*10+n
umber[5];
            break;
        }
        default:break;
    }
    if(Freq>=1000&&Freq<=100000)WriteData((unsigned long int)(Freq*85.89934592),0);
}

```

```

n=0;
Freq_temp=0;
}

if(KeyWord==10)//加減档位
{
    Range++;
    if(Range==4)Range=0;
}
if(KeyWord==13)//减
{
    if(Range==0)Freq = Freq - 1000;
    else if(Range==1)Freq = Freq - 100;
    else if(Range==2)Freq = Freq - 10;
    else Freq = Freq - 1;
    show_12864_xy(2,1);
    if(Freq<1000)Freq=1000;
    if(Freq>100000)Freq=100000;
    show_12864_6int(Freq);
    if(Freq>=1000&&Freq<=100000)WriteData((unsigned
int)(Freq*85.89934592),0);                                long
}
if(KeyWord==14)//加
{
    if(Range==0)Freq = Freq + 1000;
    else if(Range==1)Freq = Freq + 100;
    else if(Range==2)Freq = Freq + 10;
    else Freq = Freq + 1;
    show_12864_xy(2,1);
    if(Freq<1000)Freq=1000;
    if(Freq>100000)Freq=100000;
    show_12864_6int(Freq);
    if(Freq>=1000&&Freq<=100000)WriteData((unsigned
int)(Freq*85.89934592),0);                                long
}
KeyWord = KeyInvalid;
}

Timer_num++;
ADC10CTL0 |= ENC + ADC10SC;                                // Sampling and conversion start
while (!(ADC10IFG & ADC10CTL0));                          // First conversion?
if(ADC10MEM<0x3ff&&ADC10MEM>0)ADC10_Val=(float)ADC10MEM;
Vout_Pre = Vout_Cur;                                       //第一次为 0

```

```

Vout_Cur = Vout_Pre*0.98 + ADC10_Val*0.02;
if(Timer_num == 2000) //显示
{
    Timer_num=0;

    switch(N_num)
    {
        case 0: Ux_R_N = (double)(Vout_Cur-430);break; //抬高了 1.5V, N=430
        case 1: Ux_I_N = (double)(Vout_Cur-430);break;
        case 2: Us_R_N = (double)(Vout_Cur-430);break;
        case 3: Us_I_N = (double)(Vout_Cur-430);break;
        default:break;
    }
    if(N_num==3) //代表四个值都读取完毕！ 计算.....。
    {
        R =
        ((Ux_R_N*Us_R_N+Ux_I_N*Us_I_N)/(Us_R_N*Us_R_N+Us_I_N*Us_I_N))*1000;//电阻
        X =
        ((Ux_I_N*Us_R_N-Ux_R_N*Us_I_N)/(Us_R_N*Us_R_N+Us_I_N*Us_I_N))*1000;//电抗
        show_12864_xy(4,1);
        show_12864_4int((unsigned int)R);
        show_12864_xy(4,5);
        show_12864_4int((unsigned int)X);
        Imp_A = sqrt(R*R+X*X);
        show_12864_xy(3,1);
        show_12864_4int((unsigned int)Imp_A);
        Imp_O = atan2(X,R);
        Imp_O = Imp_O*180/3.1415;
        show_12864_xy(3,5);
        if(Imp_O>=0)
        {
            printf("Φ:+");
            show_12864_char((unsigned char)fabs(Imp_O));
        }
        else
        {
            printf("Φ:-");
            show_12864_char((unsigned char)fabs(Imp_O));
        }
    }

    N_num++;
    if(N_num==4)N_num=0;
}

```

```

        }
        switch(N_num)          //初始 N_num=0
        {
            case 0: Test_Ux_R;break;
            case 1: Test_Ux_I;break;
            case 2: Test_Us_R;break;
            case 3: Test_Us_I;break;
            default:break;
        }
    }
}

#pragma vector=PORT2_VECTOR
__interrupt void Port_2(void)           // Port 2 interrupt service routine
{
    if(P2IFG & BIT4)
    {
        P2IFG &= ~BIT4;      // P2.4 IFG cleared
        KeyWord=CH452_Read_Key(0); //0 反会的是数字 1 返回的事 ACCSII 码
        //        show_12864_xy(3,3);
        //        show_12864_char(KeyWord);
    }
}

```

```

#include <msp430g2553.h>

*****
* 静态函数声明
*****
static void DcoClkSet(unsigned char x,unsigned char y); //msp430g2553datasheet P30
static void MClkSet(unsigned char Div);
static void SMClkSet(unsigned char Div);
static void AClkSet(unsigned char Div);

*****
* 函数名 : DcoClkSet
* 函数功能 : 对时钟 DCOCLK 进行配置
* 函数形参 : 传入的形参为 x 和 y, 其值参考 2553datasheet 第 28 页中 DCO 频率表
* 函数返回值 : 无
*****
```

```

void DcoClkSet(unsigned char x,unsigned char y) // msp430g2553datasheet P30
{
    DCOCTL &= 0x00;
    BCSCTL1 &= 0x80;
    unsigned char temp=(x<<4)+y;
    switch(temp)
    {
        case 0x00:
        {
            DCOCTL &=~( DCO0 + DCO1 + DCO2);
            BCSCTL1 &=~( RSEL0 + RSEL1 + RSEL2 + RSEL3);
            break;
        }
        case 0x03:
        {
            DCOCTL |= ( DCO0 + DCO1 );
            BCSCTL1 &=~( RSEL0 + RSEL1 + RSEL2 + RSEL3);
            break;
        }
        case 0x13:
        {
            DCOCTL |= ( DCO0 + DCO1 );
            BCSCTL1 |= ( RSEL0 );
            break;
        }
        case 0x23:
        {
            DCOCTL |= ( DCO0 + DCO1 );
            BCSCTL1 |= ( RSEL1 );
            break;
        }
        case 0x33:
        {
            DCOCTL |= ( DCO0 + DCO1 );
            BCSCTL1 |= ( RSEL0 + RSEL1 );
            break;
        }
        case 0x43:
        {
            DCOCTL |= ( DCO0 + DCO1 );
            BCSCTL1 |= ( RSEL2);
            break;
        }
        case 0x53:

```

```

{
    DCOCTL |= ( DCO0 + DCO1 );
    BCSCTL1 |= ( RSEL0 + RSEL2 );
    break;
}
case 0x63:
{
    DCOCTL |= ( DCO0 + DCO1 );
    BCSCTL1 |= ( RSEL1 + RSEL2 );
    break;
}
case 0x73:
{
    DCOCTL |= ( DCO0 + DCO1 );
    BCSCTL1 |= ( RSEL0 + RSEL1 + RSEL2 );
    break;
}
case 0x83:
{
    DCOCTL |= ( DCO0 + DCO1 );
    BCSCTL1 |= ( RSEL3 );
    break;
}
case 0x93: //2M
{
    DCOCTL |= ( DCO0 + DCO1 );
    BCSCTL1 |= ( RSEL0+ RSEL3 );
    break;
}
case 0xA3:
{
    DCOCTL |= ( DCO0 + DCO1 );
    BCSCTL1 |= ( RSEL1 + RSEL3 );
    break;
}
case 0xB3:
{
    DCOCTL |= ( DCO0 + DCO1 );
    DCOCTL |= ( MOD4 + MOD3 + MOD2 + MOD1 + MOD0 );//微调 DCOCLK
    BCSCTL1 |= ( RSEL0 + RSEL1 + RSEL3 );
    break;
}
case 0xC3:
{

```

```

DCOCTL |= ( DCO0 + DCO1 );
BCSCTL1 |= ( RSEL2 + RSEL3);
break;
}
case 0xD3:
{
DCOCTL |= ( DCO0 + DCO1 );
DCOCTL |= ( MOD4 + MOD3 + MOD2 + MOD1 + MOD0 );//微调 DCOCLK
BCSCTL1 |= ( RSEL0 + RSEL2 + RSEL3);
break;
}
case 0xE3:
{
DCOCTL |= ( DCO0 + DCO1 );
BCSCTL1 |= ( RSEL1 + RSEL2 + RSEL3);
break;
}
case 0xF3:
{
DCOCTL |= ( DCO0 + DCO1 );
BCSCTL1 |= ( RSEL0 + RSEL1 + RSEL2 + RSEL3);
break;
}
case 0xF7:
{
DCOCTL |= ( DCO0 + DCO1 + DCO2 );
BCSCTL1 |= ( RSEL0 + RSEL1 + RSEL2 + RSEL3);
break;
}
default:
{
DCOCTL |= ( DCO0 + DCO1 + DCO2 );
BCSCTL1 |= ( RSEL0 + RSEL1 + RSEL2 + RSEL3);
}
}
}

*****
* 函数名 : MClkSet
* 函数功能 : 对时钟 MCLK 进行配置
* 函数形参 : 传入的形参为 Div, 对时钟源 DCOCLK 进行 Div 分频
* 函数返回值 : 无
****

void MClkSet(unsigned char Div)
{

```

```

BCSCTL2 &= ~(SELM1+SELM0); //select DCOCLK for MCLK
switch(Div)
{
    case 0x01:
        { //1 分频
            BCSCTL2 &= ~(DIVM1 + DIVM0);
            break;
        }
    case 0x02:
        { //2 分频
            BCSCTL2 &= ~(DIVM1 + DIVM0); //清数
            BCSCTL2 |= DIVM0;
            break;
        }
    case 0x04:
        { //4 分频
            BCSCTL2 &= ~(DIVM1 + DIVM0); //清数
            BCSCTL2 |= DIVM1;
            break;
        }
    case 0x08:
        { //8 分频
            BCSCTL2 |= (DIVM1 + DIVM0);
            break;
        }
    default :
        { //默认不分频
            BCSCTL2 &= ~(DIVM1 + DIVM0);
        }
    }
}
//*********************************************************************
* 函数名 : SMClkSet
* 函数功能 : 对时钟 MCLK 进行配置
* 函数形参 : 传入的形参为 Div, 对时钟源 DCOCLK 进行 Div 分频
* 函数返回值 : 无
//********************************************************************/
void SMClkSet(unsigned char Div)
{
    BCSCTL2 &= ~(SELS); //select DCOCLK for SMCLK
    switch(Div)
    {
        case 0x01:
            { //1 分频

```

```

BCSCTL2 &= ~(DIVS_3);
break;
}
case 0x02:
{ //2 分频
    BCSCTL2 &= ~(DIVS_3); //清数
    BCSCTL2 |= (DIVS_1);
    break;
}
case 0x04:
{ //4 分频
    BCSCTL2 &= ~(DIVS_3); //清数
    BCSCTL2 |= (DIVS_2);
    break;
}
case 0x08:
{ //8 分频
    BCSCTL2 |= (DIVS_3);
    break;
}
default :
{ //默认不分频
    BCSCTL2 &= ~(DIVS_3);
}
}
}

/*****************/
* 函数名 : AClkSet
* 函数功能 : 对时钟 MCLK 进行配置
* 函数形参 : 传入的形参为 Div, 对时钟源 LFXT1CLK 进行 Div 分频
* 函数返回值 : 无
/*****************/
void AClkSet(unsigned char Div)
{
    BCSCTL1 &= ~(XTS); //low-frequency mode
    switch(Div)
    {
        case 0x01:
        { //1 分频
            BCSCTL1 &= ~(DIVA_3);
            break;
        }
        case 0x02:
        { //2 分频

```

```

BCSCTL1 &= ~(DIVA_3);
BCSCTL1 |= (DIVA_1);
break;
}
case 0x04:
{ //4 分频
BCSCTL1 &= ~(DIVA_3);
BCSCTL1 |= (DIVA_2);
break;
}
case 0x08:
{ //8 分频
BCSCTL1 |= (DIVA_3);
break;
}
default :
{ //默认不分频
BCSCTL1 &= ~(DIVA_3);
}
}

//BCSCTL3 |= XT2S0 + LFXT1S0 + XCAP_3; //配置 ACLK 匹配电容
BCSCTL3 |= LFXT1S_2; //使用内部 VLO 作为时钟源!
}

```

```

*****
* 名称 : Init_Clk()
* 功能 : MSP430 时钟系统初始化程序
* 输入 : 无
* 返回值 : 无
*****
void Init_Clk() //时钟系统设置
{
    DcoClkSet(11,3); //4M
    AClkSet(0x01); //1 分频 VLOCLK 12K
    SMClkSet(0x01); //1 分频 SMCLK = 4M/1 =4M
    MClkSet(0x01); //1 分频 MCLK = 4M/1 =4M
}
***** end of file *****/

```