摘 要

基于嵌入式 Linux 的 200MHz 数字存储示波器的设计与实现

硕士研究生 严吉国　导师 赵力 教授

东南大学　信息科学与工程学院

示波器是最常用的电子测量仪器之一，在生产、科研、国防等许多领域有着广泛的应用。数字存储示波器更是以其特有的优势，正在逐渐取代模拟示波器而被广泛使用于各个测试领域。目前，国外的数字存储示波器的技术已经非常成熟，并且它们的产品占领了绝大部分的国内市场份额。而国内的数字存储示波器的研制尚处于起步阶段，因此自主研制数字存储示波器成为必要。

本课题是研制带宽为 200MHz 的数字存储示波器。本文根据数字存储示波器的原理，结合本课题数字存储示波器的设计指标要求，提出硬件采取高速 A/D＋FPGA＋ARM9，软件采用嵌入式 Linux 操作系统的技术方案。本系统采用 Samsung 公司的 ARM9 微处理器 S3C2410 作为控制核心，其强大的控制性能和灵活的接口技术可以大大简化系统的控制和编程。采用 National Semiconductor 公司的最高采样率为每秒 1G 采样点的高速模数转化器 ADC08D500 和 Xilinx 公司的性价比较高的 FPGA 芯片 XC3S500E，实现数据的高速采集和存储。利用嵌入式 Linux 操作系统支持各种设备的特点，充分利用了系统的片上资源，方便实现整个系统的功能扩展和软硬件升级和移植。

本文首先阐述了系统的整体结构和软硬件整体设计。然后重点研究了数据处理模块、键处理模块和显示处理模块的相关原理和软件设计思想。数据处理模块重点讨论并实现了插值算法、幅度类参数测量、时间类参数测量、FFT 等。键处理模块和显示处理模块完成了人机交互的功能。键处理模块主要研究了键盘和旋钮的原理、键扫描与接收和本数字示波器的键功能的编程。显示处理模块重点研究了 LCD 的原理、Linux 下 LCD 驱动和基于 Qt/Embedded 的数字存储示波器的图形用户界面的设计和实现。最后，本文给出了本系统的调试结果。

关键词：数字存储示波器；嵌入式 Linux；插值；参数测量；FFT；图形用户界面；Qt/Embedded

Abstract

Design and Implementation of 200MHz Digital Storage Oscilloscope Based on Embedded Linux

Candidate: Yan Jiguo, Supervisor: Zhao Li

School of Information Science and Engineering, Southeast University, China

Oscilloscope is a kind of instrument which has been widely used in modern electronic measurements. It is widely applied in many fields such as production, scientific research, national defence and so on. Digital Storage Oscilloscope (DSO) has being substituting analog oscilloscope gradually because of its proper characteristics. Now the technology of digital storage oscilloscope is more advanced overseas, and foreign products occupy most of the domestic market share. However, R&D of DSO is immature in domestic, so it is indispensable to develop digital storage oscilloscope independently.

This project is research and development of Digital Storage Oscilloscope with 200MHz bandwidth. The scheme of high-speed A/D+ARM9+FPGA in hardware and Embedded Linux as the operating system in software is presented according to the principle of DSO and requirement details of the project. The microprocessor S3C2410 with ARM9 core produced by Samsung Corporation is chose as control core. The ARM microprocessor makes the control simple as well as the programming because of its strong control ability and flexible connectors. ADC08D500 of National Semiconductor Corporation and cost-effective FPGA chip XC3S500E of Xilinx Corporation are chose to achieve the function of fast acquisition and storage. Because embedded Linux operating system supports a variety of equipment, it makes full use of the system on-chip resources and facilitates upgrades, expansion and transplantation of the system hardware and software.

First, the overall structure of the system and the overall design of the hardware and software are described. Then the data-processing module, the key processing module and display processing module are discussed. Data-processing module focuses on the realization of the interpolation algorithm, range-type parameter measurement, time-type parameter measurement, FFT and so on. Key processing module and display processing module actualize intercommunion between machine and people. Key processing module discusses principles of the knob and keyboard, key scanning and receiving and programming key functions of the DSO. Display processing module focuses on principle of LCD, LCD driver and design and realization GUI of the DSO with Qt / Embedded. Finally, this paper presents the debugging results of the system.

Key words: Digital Storage Oscilloscope(DSO); Embedded Linux; interpolation; parameter measure; FFT; Graphical User Interface(GUI); Qt/Embedded

东南大学学位论文独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得东南大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：平吉国 日 期：2009.02.26

东南大学学位论文使用授权声明

东南大学、中国科学技术信息研究所、国家图书馆有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权东南大学研究生院办理。

研究生签名：平吉国 导师签名：[illegible] 日 期：2009.02.26

第一章 绪论

1.1 示波器概述

测量是揭示客观规律，用数字语言描述周围的世界，进而改造世界的重要手段[1]。人类在认识自然和改造自然的过程中，必定要进行测量活动。电子测量，从广义上来说是利用电子技术进行的测量。电子测量仪器则是采用电子技术测量电量或者非电量的测量仪器。它是电子工业的基础和先行，近年来发展极为迅速，已成为一门独立的学科。可以说，一个国家的电子测量技术水平，在一定程度上反映了该国的电子技术水平[2]。

作为时域测量的经典仪器，示波器是电子行业工程师最为熟悉的工具，它是一种可以用来观察、测量、记录各种瞬时物理现象，并以波形方式显示其与时间关系的电子仪器，它能够直观的显示被测信号的时域信息[3]。传统的模拟示波器仍然在电子行业的各个领域扮演着重要的角色，模拟示波器通常由垂直偏转系统、水平偏转系统和显示电路等部分组成。但是它本身存在一些无法克服的缺陷，例如信号带宽频率受到阴极射线管的限制而不可能太高；不能存储被测波形；无法观测瞬时或单次信号；预触发功能难以实现等等，难以满足需要。为了将各种信号无失真地显示并存储，就必须采用数字技术，这也是数字示波器的基本思想。

随着科学技术的飞速发展，信号的捕捉、测量和研究越来越受到人们的关注和重视，在信息领域、高速计算机、高速数据通信和高速数字集成电路及其系统内，面临着硬件、软件以及由软硬件共同作用而产生的偶发性故障，迫切需要更高速的数字存储示波器（Digital Storage Oscilloscope，DSO）才能得心应手的解决这些难题。数字存储示波器整个系统的调节全部由微处理器在相应的软件支持下自动进行，包括自动设置、自动测量、自动校正、波形存储、计算机 I/O 和打印输出等一系列优点。它的出现和发展根本改变了以往示波器的格局，促进了示波器的发展。此外，数字存储示波器是数据采集、显示与微机管理紧密结合的产物，数据采集采用不同的方法获得信息数据，这便构成了与模拟示波器不同的特征，获得信息数据的方法是对输入高频信号取样，从此意义上讲，所有数字存储示波器可称为取样示波器（但取样示波器并非全是数字存储示波器）。所有这些，为数字存储示波器的快速发展提供了更为广阔的市场。

自从 1998 年 Tektronix 公司推出数字荧光示波器 (Digital Phosphor Oscilloscopes，DPO)以来，其取得了长足的发展。数字荧光示波器是将电信号数字化，并且以三维数据（信号的幅度、时间以及幅度相对于时间的分布）实时的存储、分析和显示波形的仪器。数字荧光示波器被称为第三代示波器，它具有 DSO 的各种传统优点，同时它也具有模拟示波器的明暗显示和实时特性，但价格高昂限制了它的广泛应用。

当前，数字示波器正逐渐取代模拟示波器成为主流的示波器产品。随着大规模集成电路技术、信号分析处理技术的发展，特别是在微型计算机引入到示波器后，示波器无论在设计、性能、功能、使用、操作还是在故障诊断上都取得了巨大的进步。

1.2 国内外数字示波器的发展现状

目前，市场上的数字示波器特别是高端产品都是国外产品占主流，如 Tektronix、Fluke、Agilent、LeCroy 等。其中美国 Tektronix 公司的示波器一直处于领先地位，被世界公认为示波器的权威。Tektronix 以前推出的 TDS 系列示波器具有显示所有测量的有效修正，先进的波形处理能力。近来 Tektronix 公司又推出 DPO70000/DSA70000 系列示波器，最高带宽达 20 GHz，在全部四条通道上提供了高达 50GS/s 的实时取样速率。另外，高达 200M 样点的记录长度，支持 MultiView Zoom 功能，导航迅速。同时支持 Pinpoint 触发技术，提供了最灵活、性能最高的触发功能。其中 DPO70804 拥有可至 35 ps 的典型上升时间，并会为关键的抖动测试提供低于 400 fs rms 的典型噪声基底，为业界最低值。每个通道 10M 的内存，这就是说以 40 ps 单次样点间隔进行全速的实时数据采集时，最长时间窗口可达 4

ms，这保证了以最高性能工作时，依然能得到很好的解析度。此外，其 12.1"显示屏则是业内最大的显示器。

Agilent 公司的高性能 80000 系列 Infiniium 示波器的带宽为 8GHz～13GHz，采样频率为单通道 20GSa/s，双通道 40GSa/s，存储深度也达到了 64M，它具有业界最低的本底噪声、最低的抖动测量本底、最低的触发抖动、最平坦的示波器和探头组合频率响应曲线等。卓越的信号完整性，可消除因为示波器或探头系统的噪声、抖动或频率响应曲线不佳而导致的测量精度方面的误差，从而最大限度地提高工程师的设计裕量。

尽管在国际上数字示波器已经有了较为成熟的发展，我国目前在数字示波器生产领域内基本上处于起步阶段。国内示波器主要生产厂家有上海无线电二十一厂、西安红华无线电厂、辽宁无线电二厂、江苏绿扬集团、北京普源精电科技有限公司等。

江苏绿扬电子仪器集团有限公司是我国电子仪器行业的骨干企业，1998 年以来，电子测量仪器的产销总量连续居全国同行第一，约占中国大陆市场销售量的 1/3。但是他们的产品主要是模拟示波器类型的产品。

北京普源精电科技有限公司是一家以自主知识产权生产数字存储示波器的企业，主要产品是台式数字存储示波器和虚拟仪器系列。2004 年 2 月，该公司宣布他们自主开发的 DS5000 系列数字存储示波器，打破了国外产品一统天下的局面，推动了国产仪器的数字化发展。

近十年来，国内数字存储示波器技术研究以及发展也取得了相当的成果。但是我国示波器行业总体水平还是比较落后的，国产示波器的智能化、数字化、集成化还是很低。

1.3 数字存储示波器的原理和特点

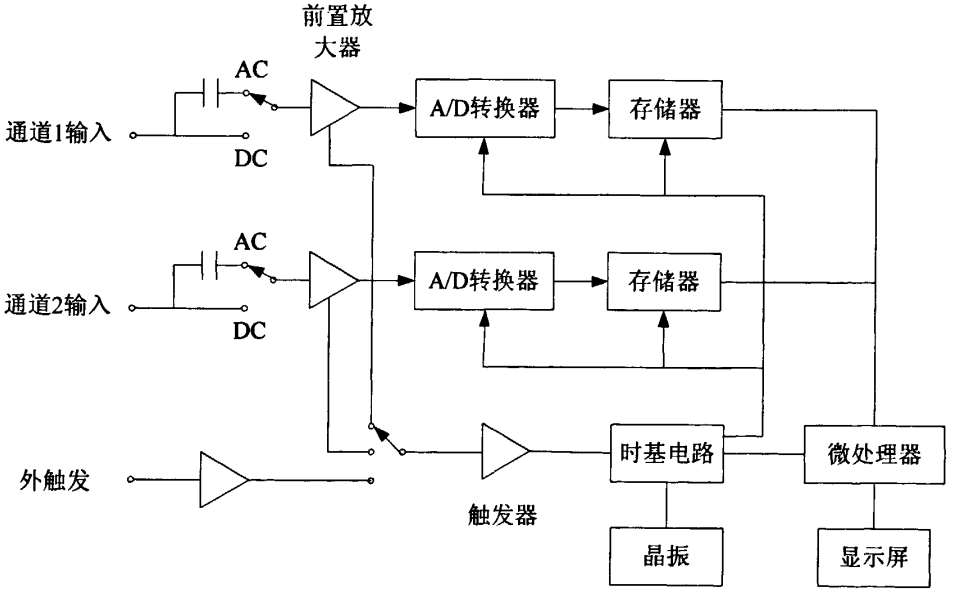


图 1-1 双通道数字存储示波器原理结构框图

一台通用双通道 DSO 的原理[4]结构框图如图 1-1 所示。由图可见，数字存储示波器在输入电路上与模拟示波器相似，采用垂直通道偏转因子和电平控制信号，使信号经过放大、衰减、耦合或加上一定直流偏置，落入 A/D 转换器的电压转换范围之内，其后的数据采集系统中的 A/D 转换器将该输入信号转换成相应的数字值并存入存储器，此过程在采样时基电路的控制下不断地循环进行，而这时仪器的触发电路不断监测输入信号，看是否有触发信号产生，一旦触发条件满足，则采样过程中断，微处理器通过对存储器内采样数据的处理和显示，即可在屏幕上重现信号电压与时间的关系，也就是信

号电压波形。

与传统的模拟示波器相比，数字存储示波器具有许多优点[5][6]，主要表现在以下几个方面：

1. 能够捕获单次信号： DSO 能够捕捉像电源开、关这样的单次瞬态事件和低重复率信号。
2. 很强的数据处理能力：由于 DSO 内部使用一个微处理器，它能够在所获得的波形上完成幅度和时间参数测量以及波形运算等功能，加上选件能够完成更复杂的数学运算，如积分、倒数、指数、对数、平均、数字滤波、极值、FFT 等。
3. 数据存储：DSO 带有非易失性的波形存储器，它们能够提供与 DSO 兼容的存储卡或软盘。示波器也能够容易地与许多绘图仪和打印机相连来进行高质量的硬拷贝。
4. 先进的触发功能：DSO 不仅能显示触发后的信号，而且能显示出发前的信号，并且可以选择超前或滞后的时间。除此之外，DSO 还可以提供边沿触发、组合触发、状态触发、延迟触发等多种形式，来实现多种触发功能。
5. 测量精度高：数字存储示波器由于采用晶振作高稳定的时钟，有很高的测时准确度，采用高分辨率 A/D 转化器也使幅度测量准确度大大提高。
6. 多波形显示：数字存储示波器可存储多个波形，并能在屏幕上显示同一时间或不同时间发生的波形，故可以方便的进行分析和比较。

1.4 数字存储示波器的主要技术指标

数字存储示波器的技术指标[7][8]非常丰富，下面仅介绍最主要的几种。

1. 最高数字化采样率 f_s

采样率，即单位时间内对模拟输入信号的采样次数，它主要取决于 A/D 变换器的转换速率，常用每秒的取样点 Sa/s(sample/second，SPS)来表示。在实际应用中，数字化采样率根据被测信号所设定的扫描时间(t/div)来选择。

$$f_s = \frac{\mathrm{N}}{\mathrm{t/div}} \tag{1.1}$$

其中，N 为每格取样数。

2. 带宽

带宽是指示波器输入不同频率的等幅正弦信号时，屏幕上对应基准频率的显示幅度随频率变化而下跌 3dB 时，其下限到上限频率的范围。这里基准频率是指示波器频率特性平坦部分的某一指定频率。数字存储示波器的带宽分为单次带宽和重复带宽，这取决于它的采样方式是重复采集还是单次采集。对于单次信号和慢速变化的信号，数字存储示波器采用实时取样工作方式，这时带宽又称为实时带宽，其取决于最大取样速率和所采用的显示恢复技术。对于一个满刻度的正弦波来说，单次带宽 B_w 计算公式如下所示：

$$B_w = \text{最大采样速率(MHz)/K} \tag{1.2}$$

式中 K 为一常数。用光点显示时，约等于 25；用矢量显示时，约等于 10；用正弦内插显示时，约等于 2.5。

3. 分辨率

在数字存储示波器中，屏幕上的点不是连续的而是“量化”的。分辨率是反映存储信号波形细节的综合特性，它包括垂直分辨率(电压分辨率)和水平分辨率(时间分辨率)。垂直分辨率一般用示波器中所采 A/D 转换器量化的位数表示。水平分辨率由存储器的容量来决定，常以屏幕每格包含多少取样点或百分数来表示。

4. 通道数

波形测量的输入通道数，主要用于观测若干个信号之间的相互关系。对于通常的经济型故障查询应用来说，流行的是双通道示波器。

5. 存储深度

存储深度又称存储容量，通常定义为数字存储示波器的采样存储器能够连续存入样点的最大字节数。

6. 量程和时基

量程又称垂直灵敏度、垂直偏转因数、幅度档位，是指示波器显示的垂直方向每格所代表的电压幅度值，常以 V/div 或 V/cm 表示。垂直灵敏度参数表明了示波器测量最大和最小信号的能力。时基又称扫描速度、水平偏转因数、扫描时间因数，是指示波器显示的水平方向每格所代表的时间值，常以 s/div 或 s/cm 表示。时基越小，示波器拉宽高频信号的能力越强。

1.5 本论文的设计任务

目前，国外的数字存储示波器产品占领了绝大部分的国内市场。而国内的数字存储示波器的研制尚不成熟，因此自主研制数字存储示波器成为必要。

本课题设计的数字存储示波器主要技术指标如下：

1. 垂直分辨率： 8 bits;
2. 带宽：双通道 200MHz ；
3. 垂直灵敏度范围：1mV/div~10V/div（1-2-5 步进）;
4. 上升/下降时间： ≤1.8ns;
5. 输入阻抗： 1MΩ，13 pF;
6. 最高采样率： 1GSa/s;
7. 存储深度：每通道 10K;
8. 时基范围：2ns/div~50s/div（1-2-5 步进）;
9. 通信接口：USB，RS232 传输波形数据。

本课题设计的数字存储示波器主要功能如下：

1. 自动测量功能：可以自动测量从测量菜单中选择的波形参数（最大值、最小值、峰峰值、平均值、有效值、频率、周期、上升时间、下降时间、正频宽、负频宽等）。
2. 光标测量功能：用光标测量幅度类参数和时间类参数。
3. 波形和设置的存储调出功能：在存储调出菜单下可以选择将屏幕上显示的波形或设置参数存储到某一内存区域；并且可以选择将存储过的波形或设置参数回调出来。
4. 自动设置功能：对任何一种未知的信号，按示波器面板的“自动”键，波形以适当的时基和量程显示在屏幕上。
5. 触发特性：有内部/外部，上升沿和下降沿触发，触发方式有自动，正常和单次，具有预触发功能。
6. ACQUIRE 方式：有采样，峰值检测和平均三种捕捉方式；平均次数可选 4/16/32/64/128 次。
7. 数学运算功能：包括通道 1 和通道 2 加，减，乘和 FFT 功能。
8. 多种语言界面：操作的界面语言可选择中文，英文等。

为了达到这一目标，本课题重点在以下几个方面展开工作:

1. 示波器基本原理及相关技术的研究学习

数字存储示波器的设计应用到许多方面的理论，其中包括采样定理、信号预处理、测频原理、数字信号处理、插值算法、显示技术等。与数字存储示波器相关的技术有嵌入式系统、操作系统、设备驱动和图形用户界面编程等。本论文的初期工作主要是阅读大量的相关书籍，学习相关技术，对项目的整体方案进行论证，解决数字存储示波器设计中的关键技术。

2. 硬件系统设计

硬件设计主要包括前端信号调理电路设计、触发电路设计、数据采集电路设计、显示电路、系统控制电路和可编程器件接口电路设计以及键盘电路等。

3. 软件系统设计

软件编程中设备驱动程序使用C语言编程，应用程序采用Linux下的C++编程，使用QT进行图形用户界面（GUI）编程。设计数字存储示波器的软件整体框架，主要完成采集、数据处理、信号调理电路控制、用户交互界面设计、插值处理程序、各种键功能的设计与实现、频谱分析等。

4. 系统调试

调试重点在信号调理电路、采集和触发电路、各种键功能和LCD显示的刷新速度。信号调理电路要保证在所有档位下都能获得A/D所需的采样信号幅度和足够的信号输入带宽；采集和触发电路要实现正延迟和负延迟触发，自动触发、正常触发和单次触发，触发电平控制等；各种键要正确实现示波器的功能；为了提高LCD显示的刷新速度，必须优化程序。系统调试最终要求完成数字存储示波器的初样。

第二章　系统总体设计

2.1 系统的总体结构

根据数字存储示波器的原理，结合本课题数字存储示波器的设计指标要求，我们将本系统设计成由以下几个子系统组成：信号调理模块、触发控制模块、数据采集和存储模块、系统控制模块、键盘与显示模块、电源模块等。相应的结构框图如图 2-1 所示。

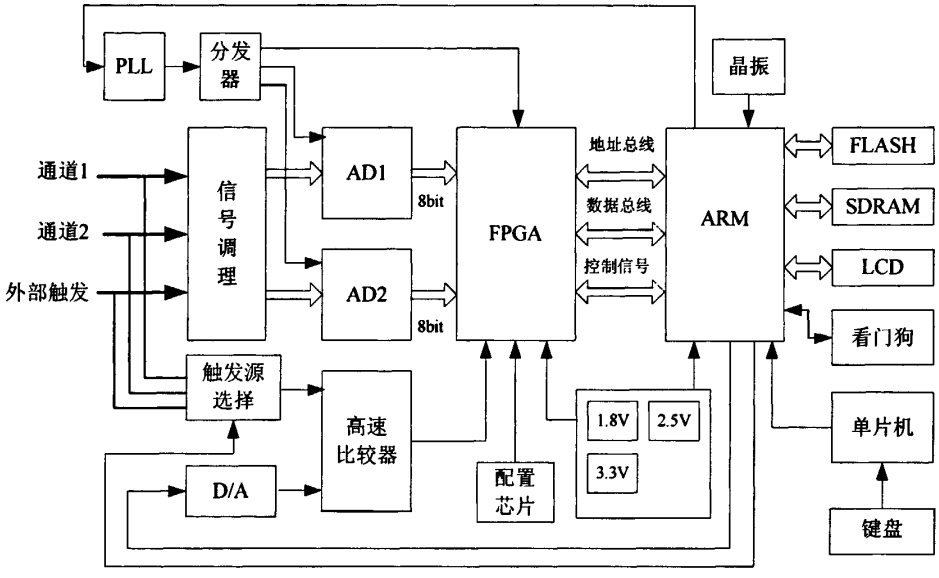


图 2-1 数字存储示波器系统结构框图

各部分的功能描述如下：

1. 信号调理模块包括放大、衰减、通道耦合、触发耦合等。其中，放大、衰减是为了使输入信号的电平范围调节到 300mV 的 ADC 满量程范围内，以便能够适合 ADC 模数转换器的工作范围；通道耦合、触发耦合是控制信号通路是否允许有直流成分的存在。

2. 触发控制模块由外部的高速模拟比较器和 D/A 组成，由 ARM 控制 D/A 产生预置比较信号，与用户选定的触发输入信号进行比较，产生触发信号送入 FPGA 内，形成触发。

3. 数据采集和存储模块主要器件为高速 ADC 和 FPGA。ADC 负责将调理过的信号采集下来，FPGA 根据触发信号对采集下来的波形值进行接收和存储，并将存储的波形数据传输给 ARM 主控模块进行后续的处理和显示工作。考虑到系统对采样速率的高要求，我们采用了并行采样技术，利用两个 500M 的 ADC 分别在时钟信号的上升沿和下降沿采样来实现最高 1GSPS 的采样速率。这样不但可以降低数据的传输数率以便后端的处理，也减少了系统成本。

4. 系统控制模块由 ARM 完成对系统其它模块的控制，并接收 FPGA 传输来的波形数据，根据键盘和旋钮的动作控制波形和菜单进行实时的显示与更新。它还完成系统与上位机的通信。

5. 键盘与显示模块主要完成系统的人机交互工作以及波形和菜单的实时显示和更新。本系统采用一个单片机单独管理键盘，一个 TFT 型液晶显示器作为系统的显示屏幕。当操作者每一次按键或旋动旋钮时，单片机都通过串口将键值发送给 ARM，ARM 对键值进行分析，根据操作者的要求实时地更新菜单和波形。

6. 电源模块完成对系统的供电功能。

2.2 本方案的优点

本方案在硬件上采取高速 A/D+FPGA+ARM9，软件上选择嵌入式 Linux 操作系统。使用本方案来实现 DSO 具有以下几个优点：

1. 采用 ARM 处理器，利用其强大的控制性能，实现系统控制的优化；
2. 利用 ARM 灵活的接口技术使系统的存储器扩展、键盘控制、显示控制等方面的工作大大简化，利用其内置的 USB 控制器和 UART 控制器可实现和上位机的高速通信；
3. 利用 FPGA 的高速 I/O 口实现高速数据的缓冲，任务转化为并行处理低速数据，从而减轻了后端 ARM 的负担；
4. 利用 FPGA 丰富的硬件资源来实现一些外围芯片的功能，提高了系统的集成度；
5. 引入了嵌入式 Linux 操作系统，利用 Linux 支持各种硬件的特点，使系统的片上资源能够得到最大限度的利用。并能在不作大规模软件修改的情况下，实现整个系统的功能扩展和软硬件升级和移植。

2.3 系统硬件设计

下面主要介绍信号调理模块、数据采集和存储模块、系统控制模块，键盘与显示模块放在后面章节介绍。

2.3.1 信号调理模块

信号调理模块包括放大衰减电路、耦合控制电路、直流偏置等。

放大衰减电路调整输入波形的幅度和范围。必须把不同幅度的信号进行变换以适应屏幕的显示范围，这样就可以按照标尺刻度对波形进行测量，为此就要求对大信号进行衰减、对小信号进行放大。

耦合控制电路决定输入信号从示波器面板上的输入端到该通道垂直放大器其它部分的方式。耦合控制可以有两种设置方式，即 DC 耦合和 AC 耦合。DC 耦合方式为信号提供直接的连接通路，因此信号的所有分量（AC 和 DC）都会影响示波器的波形显示。AC 耦合方式则在输入端和衰减器之间串联一个电容，这样，信号的 DC 分量就被阻断，而信号的低频 AC 分量也将受阻或大为衰减。

直流偏置的作用是控制信号的直流偏移。

图 2-2 是前端模拟信号调理电路结构框图。

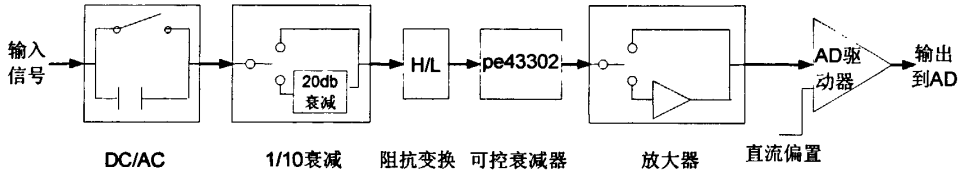


图 2-2 前端模拟信号调理电路结构框图

2.3.2 数据采集和存储模块

如前所述，本系统的设计要求是：双通道，实时最高采样率为 1GSPS，带宽达到 200MHz，垂直分辨率为 8bit，每个通道的存储深度为 10k，并且要求低功耗低成本。由于最高采样率达到 1GSPS，所以需要采取一种方案来满足这种高速的数据采集、存储、处理和显示。

高速数字存储示波器的一种方案是采用高速 A/D 转换器+硬件 FIFO+MCU+LCD 控制器的硬件结构来实现示波器从采样，数据处理再到显示的一系列的过程，其中包括 MCU 对 A/D 转换器的采样控制电路，MCU 对 FIFO 控制电路，MCU 内部数据处理，触发控制电路，LCD 控制器的电路等多个复杂部分，由于涉及到的中低集成度的电子元器件较多，电路复杂，整个系统的集成度较低，对一个高速的系统来说，稳定性和可靠性都大大降低。

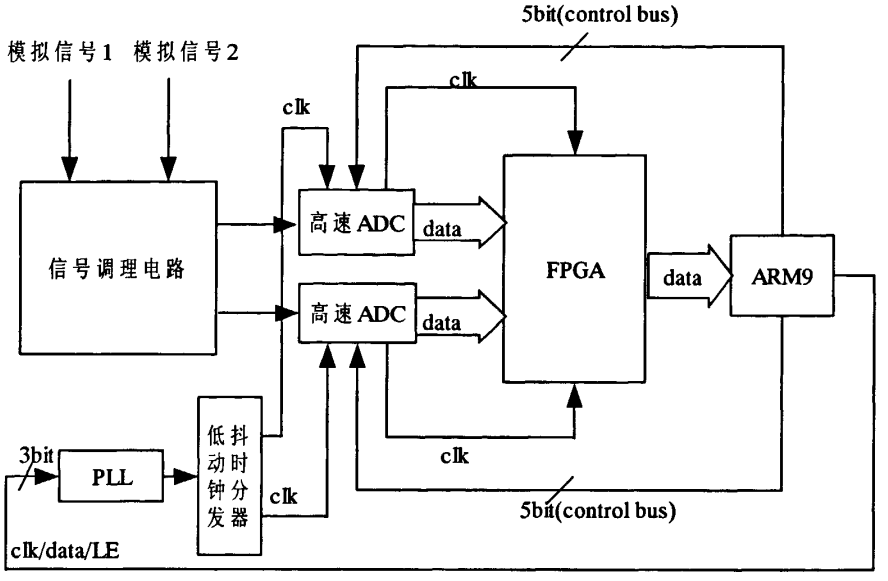


图2-3 数据采集与存储模块示意图

基于上述原因，本课题采用了高速 A/D 转换器＋FPGA＋ARM9 的硬件结构，如图 2-3 所示，前端利用 FPGA 电路设计灵活，硬件资源丰富的特点，用 FPGA 作为高速数据的缓存，并产生触发控制信号，后端利用 ARM9 内部丰富的资源来对整机进行控制，包括从 FPGA 中读取采样数据，进行数值计算，对 LCD 的控制，菜单和波形的显示，外扩 USB 接口和 RS232 串口的控制等。整个系统为 3 个主要的芯片组成，大大简化了硬件电路，由于 FPGA 的加入，一些数据的计算既可以在 FPGA 内也可以在 ARM 内完成，使得整个设计更加灵活，提高了该硬件结构的通用性、有效性和可靠性。

根据以上技术要求，本课题数据采集使用的芯片是 ADC08D500。ADC08D500 是 2005 年美国国家半导体（National Semiconductor）公司推出的差分输入，取样率达 500MSPS 的 8 位双通道模拟/数字转换器，这款 A/D 转换器的特点在于其高性能和低功耗，在过去的一些 A/D 转换器中，若采样速率超过 1GSPS，而同时又要确保动态性能保持在极高的水平，那么其功耗一定很高。而这款 A/D 转换器的功耗低至 1.4w 以下，因此在设计电路时无需在为该 A/D 转换器加设散热器，不仅节省了电路板的板面空间，而且还可降低系统成本以及提高系统的可靠性。其具体性能[9]如下：

1. 8bits 采样精度；
2. 10^{-18} 误码率；
3. 有效位数（ENOB）达 7.5Bits（输入信号 250MHz 时）；
4. 微分非线性误差（DNL 为±0.15LSB）；
5. 可调的模拟输入范围；
6. 500MSPS 实时采样工作时的功耗 1.4W，掉电状态功耗为 3.5mW；
7. 支持单边数据率和双边数据率的输出时钟；
8. 1.9V 供电（1.8V－2.0V）。

ADC08D500 除了单通道能实现最高采样达 500MSPS 的速率外，最大的特点是能够进行双边沿采样（DES）。在 DES 模式下，只有一路信号被采样，另一路信号接空，被接入信号在一个转换器内以采样时钟的上升沿采样，而在另一个转换器内以采样时钟的下降沿采样，这样在一个采样时钟周期内对该信号进行了两次采样，使 500MHz 采样信号能够进行 1GHz 的采样。

在 DES 模式下，输出是一个 1:4 的信号分离器，即信号分 4 路并行输出。在采样速率为 500MHz 时（在 DES 模式下进行的是 1GHz 的采样），每个输出端口的数据速率是 250MHz。

输出的模式分为单边数据率模式和双边数据率模式，在单边数据率模式下，输出数据和输出时

钟的频率相同，外部处理器可以在输出时钟的上升沿采样，而在双边数据率模式下，输出时钟的频率是输出数据速率的一半，也即外部处理器必须在输出时钟的上升和下降边沿都采样，才能采集完整。采用双边数据率模式可以降低输出的时钟频率，增加系统稳定性。

利用 ADC08D500 双边采样的这种特性，在不增加任何硬件成本的情况下使系统的单次带宽指标提高了一倍。当采样速率为 500MHz，输出时钟在双边数据率输出模式下，变为输出数据速率的一半，即 125MHz，降低了后端 FPGA 的接收频率。同时，采用了较低速度的器件实现高速数据的采样，减小了电路实现的难度，也提高了系统的可靠性。

本系统中，前端高速 A/D 转换器的输出信号通道较多（8 个通道），如果采用分离的存储器来搜集采集过来的数据，所用的存储器数量将达到 8 片，那么电路板上的数据线和地址线将大大增加，面积也相应增加了，并且要增加控制电路对存储器进行有序的读取，不仅降低了整个系统的集成度而且增加了设计和控制的难度，所以对于本系统的存储器，最终选择了集成度高，控制灵活的 FPGA 芯片。

高速 FPGA 用来接收前端高速 A/D 转换器采集的多通道数据，用作数据的缓存。由于 FPGA 的 I/O 端口读写速度要求与 A/D 转换器输出的时钟相匹配，并且按照设计的要求，该 DSO 需要每个通道 10K 的存储容量，故对 FPGA 的存储空间也有一定的要求，在 FPGA 内需要作数字预处理，也需要高速的乘法器来帮助完成，综合成本考虑，最终选择了 Xilinx 公司的性价比较高的 Spartan-3E 系列的 FPGA 芯片 XC3S500E。

Spartan-3E 是 Xilinx 公司 2005 年推出的 FPGA 系列，其第四个采用先进的 90nm 制造工艺技术生产的系列产品。该系列的器件密度为 10 万门到 160 万门，其单位逻辑单元的成本是 FPGA 行业中最低的，并且优化了针对以逻辑为中心的设计，并且能够内嵌微型处理器，微型控制器和信号处理器，这些使 Spartan-3E 系列的芯片性价比大大提高，降低了器件的成本和总体系统的成本。

XC3S500E 主要内部资源和性能特点[10]为：

1. 500K个系统门；
2. 10476个逻辑单元（LE）；
3. 360KBits的RAM 存储块（Block RAM）；
4. 73KBits 可分配 RAM 单元（Distributes RAM）；
5. 4 个数字时钟管理单元（DCM）；
6. 20 个 18×18 位的硬件乘法器（Multipliers）；
7. 最大单端 I/O 数为 232 个；
8. 最大差分 I/O 数为 92 个；
9. 支持 SPI 接口或者字节宽并行接口 flash 对其配置；
10. 简易实现 DDR 存储器接口；
11. 支持增强行 DDR 接口；
12. 支持 18 种通用 I/O 标准，包括 LVCOMS，LVTTL，HSTL 和 SSTL 单端信号标准，以及 LVDS、RSDS、mini-LVDS 等差分信号标准；
13. 最大 I/O 口数据传输率达 622MHz；
14. ISE（集成软件环境）工具提供对 Spartan-3E 系列芯片进行仿真调试下载。

2.3.2 系统控制模块

系统控制模块是数字存储示波器的重要组成模块，它作为前端采集与存储模块和后端键盘和显示模块的衔接，在系统中起着重要的作用。其主要任务有：将前端采集存储的数据以波形的形式显示在 LCD 上；通过串口接受键盘的键值，根据键值改变 LCD 上波形和菜单的显示；给前端的采集模块发控制信号，控制数据采集的速度；系统控制模块中的 USB 以及 UART 接口还负责和上位机通信。

本系统是嵌入式系统，所以选择嵌入式处理器作为系统的控制核心。从应用的角度来划分，嵌

入式处理器分为嵌入式微处理器、嵌入式微控制器和嵌入式 DSP。嵌入式微控制器又称单片机，它的突出优点是控制能力强，但数据处理速度较低。嵌入式 DSP 的数据处理能力非常强，但控制能力较弱。最终我们选择了嵌入式微处理器以满足本系统的高速数据处理和多种控制功能。经过论证，我们使用 Samsung 公司以 ARM920T 为内核的 S3C2410[11]嵌入式微处理器作为数字存储示波器的主控制芯片。图 2-4 是系统控制模块结构框图。Samsung 公司推出的 32 位 RISC 处理器 S3C2410，最高处理速度达到 203MHz，并兼有低功耗、高性能和高集成度等特性，是业内应用最广的嵌入式微处理器之一。

S3C2410 的显著特性是它的 CPU 核心，是一个由 Advanced RISC Machines（ARM）有限公司设计的 16/32 位 ARM920T RISC 处理器。ARM920T 采用 5 级流水线，实现了 MMU，AMBA BUS 和 Harvard 高速缓冲体系结构。这一结构具有独立的 16KB 指令 Cache 和 16KB 数据 Cache，每个都是由 8 字长的行（line）构成。

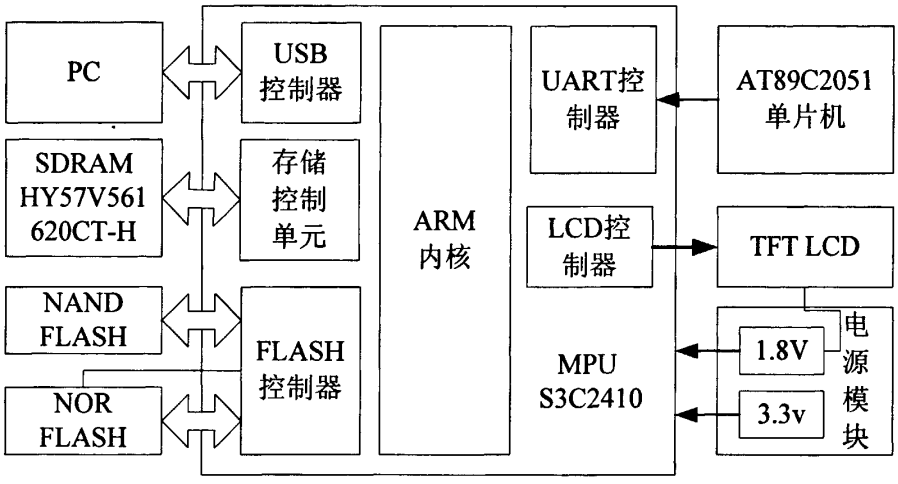


图 2-4 系统控制模块结构框图

S3C2410 主要集成以下片上功能：

1. 1.8V/2.0V 内核供电，3.3V 存储器供电，3.3V 外部 I/O 供电；
2. 具备 16KB 的 I-Cache 和 16KB 的 D-Cache/MMU；
3. 外部存储控制器（SDRAM 控制和片选逻辑）；
4. LCD 控制器（最大支持 4K 色 STN 和 256K 色 TFT）并带有 1 个 通道 LCD 专用 DMA；
5. 4 通道 DMA 并有外部请求引脚；
6. 3 通道 UART(IrDA1.0,16 字节 Tx FIFO，和 16 字节 Rx FIFO)/2 通道 SPI；
7. 1 通道多主 IIC-BUS/1 通道 IIS-BUS 控制器；
8. 兼容 SD 主接口协议 1.0 版和 MMC 卡协议 2.11 兼容版；
9. 2 端口 USB 主机/1 端口 USB 设备（1.1 版）；
10. 4 通道 PWM 定时器和 1 通道内部定时器；
11. 看门狗定时器；
12. 117 个通用 I/O 口和 24 通道外部中断源；
13. 功耗控制模式：具有普通，慢速，空闲和掉电模式；
14. 8 通道 10 比特 ADC 和触摸屏接口；
15. 具有日历功能的 RTC；
16. 具有 PLL 片上时钟发生器。

可以看出，在本课题中需要的各种控制器（LCD 控制器、UART 控制器、外部存储控制器等）在 S3C2410 上都已经集成，所以不需要另外的电路，提高了系统的集成度。另外，S3C2410 中的 117

个通用 I/O 口和其它各种接口为它控制本系统的其它硬件提供了方便。

2.4 系统软件设计

2.4.1 嵌入式系统

如今，嵌入式系统已成为当今最为热门的领域之一，它迅猛的发展势头引起了社会各方面人士的关注。如家用电器、手持通信设备、信息终端、仪器仪表、汽车、航天航空、军事装备、制造工业、过程控制等。

根据 IEEE（国际电气和电子工程师协会）的定义，嵌入式系统是“控制、监视或者辅助设备、机器和车间运行的装置”（原文为 devices used to control, monitor, or assist the operation of equipment, machinery or plants.）[12]。这个定义主要是从嵌入式系统的用途方面来进行定义的，可以看到，单个嵌入式系统的功能较为单一，是专为某一具体的用途而设计的。这与通用计算机功能的“大而全”形成了鲜明的对比。

嵌入式系统更加常用的定义为：嵌入式系统是指以应用为中心，以计算机技术为基础，软硬件可裁剪，适用应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统[13]。

这个定义较为具体的指明了嵌入式系统的三大基本要素：嵌入性、专用性、计算机系统。可以说，“嵌入性”是它的特征，“专用性”是它的灵魂，“计算机系统”则是它的实质。

嵌入式系统主要由嵌入式处理器、外围硬件设备、嵌入式操作系统以及用户应用软件等部分组成，其体系结构如图 2-5 所示。

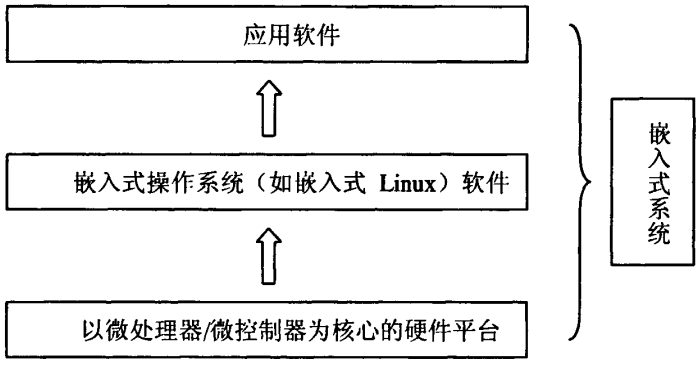


图 2-5 嵌入式系统体系结构图

从该图中可以清楚的看到嵌入式系统体系结构上下层之间的关系。其中，硬件平台包括嵌入式处理器和外围设备，它们位于嵌入式系统结构中的最底层；嵌入式操作系统与通用操作系统的功能类似，为用户屏蔽硬件底层的具体细节，提供一个透明的操作空间；应用软件是位于嵌入式操作系统之上的，当然，用户也可以直接在嵌入式操作系统之上进行开发。

如前所述，本系统的硬件平台是以 S3C2410 嵌入式微处理器为核心，通过它来控制系统的采集、存储、显示等硬件模块。选择好嵌入式处理器以后就要考虑选择什么嵌入式操作系统，下面介绍嵌入式操作系统的选择。

2.4.2 嵌入式操作系统的选择和依据

2.4.2.1 典型的嵌入式操作系统

从 20 世纪 80 年代开始，市场上出现各种各样的商用嵌入式操作系统系，这些操作系统大部分都是为专有系统开发的，从而逐步演化成了现在多种形式的商用嵌入式操作系统百家争鸣的局面[14]。嵌入式操作系统有：μC/OS，VxWorks，pSOS，QNX，Linux 和 Windows CE 等。下面分别介绍几

个具有代表性的嵌入式操作系统。

(1) μC/OS

μC/OS 是一个典型的实时操作系统。该系统从 1992 年开始发展，目前流行的是第二个版本，即μC/OS-II。它的特点可以概括为以下几个方面：公开源代码，代码结构清晰明了，注释详细，组织有条理，可移植性好，可裁剪，可固化。内核属于抢占式，最多可以管理 60 个任务。该系统短小精悍，是研究和学习实时操作系统的首选。缺点是缺乏便利的开发环境。

(2) Windows CE

Windows CE 是微软公司的产品，但不是削弱的 Windows 版本，它是从整体上为有限资源的平台设计的多线程、完整优先权、多任务的操作系统。Windows CE 采用模块化设计，并允许它对于从掌上电脑到专用的工控电子设备进行定制。操作系统的基本内核至少需要 200KB 的 ROM。从 SEGA 的 DreamCast 游戏机到现在大部分的高价掌上电脑很多都采用了 Windows CE 作为操作系统。其缺点是价格过高，使得整个产品的成本急剧上升。

(3) VxWorks

VxWorks 是 WindRiver 公司专门为实时嵌入式操作系统设计开发的操作系统软件，为程序员提供了高效的实时任务调度，中断管理，实时的系统资源以及实时的任务间通信。应用程序员可以将尽可能多的精力放在应用程序本身，而不必再去关心系统资源的管理。该系统主要应用在单板机，数据网络（以太网交换机，路由器）和通信等多方面。VxWorks 是一个非常优秀的实时系统，但是其昂贵的价格使很多厂商望而却步。

(4) 嵌入式 Linux

在所有的操作系统中，Linux 是发展最快、应用最广泛的。Linux 本身的种种特性使它成为嵌入式开发的首选。嵌入式 Linux（Embedded Linux）是指对 Linux 经过裁剪小型化后，可固化在存储器或单片机中，应用于特定嵌入式场合的专用 Linux 操作系统。为了更好的适合嵌入式领域的开发，嵌入式 Linux 在 Linux 基础上做了部分改进，如改善了内核结构和提高了系统实时性等。在进入市场的前两年中，嵌入式 Linux 的设计通过广泛应用而获得巨大成功。随着嵌入式 Linux 技术的成熟，定制需要的尺寸尤为方便，同时支持更多平台，并从早期的试用阶段迈进到成为嵌入式市场的主流。根据 IDC 的报告，Linux 已经成为全球第二大操作系统。预计在服务器市场上，Linux 在未来几年将以每年 25%的速度增长，中国的 Linux 市场更是保持每年 40%左右的增长速度。

2.4.2.2 选择嵌入式 Linux 操作系统的依据

在嵌入式系统的前期设计中，首先必须决定采用哪一种嵌入式操作系统。这是一个很重要的决定，因为这将影响到工程后期的发布以及软件的维护。抛去时限要求，可以把嵌入式操作系统分为商用型和免费型两种。商用型系统功能稳定，可靠，有完善的技术支持和售后服务，但往往价格昂贵。免费的操作系统在价格方面具有优势。但是不管选用什么样的系统，都应该考虑操作系统对硬件的支持。其次要考虑开发调试的工具，特别是对技术水平不强的企业来说开发工具往往起决定作用。第三要考虑该系统是否满足应用要求。如果一个操作系统提供的 API 很少，那么应用层很难进行二次开发。综上所述，我们选择嵌入式 Linux 操作系统，依据如下：

(1) 可应用于多种硬件平台。Linux 已经被移植到多种硬件平台，这对受开销、时间限制的研究与开发项目是很有吸引力的。原型可以在标准平台上开发然后移植到具体的硬件上，加快了软件与硬件的开发过程；

(2) Linux 可以随意地配置不需要任何的许可证或商家的合作关系。唯一的限制是开发者必须做出对 Linux 社区有益的改动；

(3) 它是免费的，源代码可以得到。这是最吸引人的。毫无疑问，这会节省大量的开发费用；

(4) 优异的网络支持。内核直接提供网络支持，而很多其他操作系统需外挂 TCP/IP 协议包；

(5) Linux 的高度模块化使添加部件非常容易，模块可以动态添加；

(6) Linux 在台式机上的成功，也保证了 Linux 在嵌入式系统中的辉煌前景；

(7) Linux 具备一整套工具链，容易自行建立嵌入式系统的开发环境和交叉运行环境，可以跨越嵌入式系统开发中仿真工具的障碍；

(8) Linux 是一个自由开放的世界，在 Linux（无论 PC 还是嵌入式系统）上进行软件开发都可以在广袤的网络资源中获取帮助。

2.4.3 基于嵌入式 Linux 操作系统的软件设计

本系统的软件又分为底层设备驱动程序和上层应用程序。硬件，设备驱动程序，操作系统和应用程序的关系[15]如图 2-6 所示。

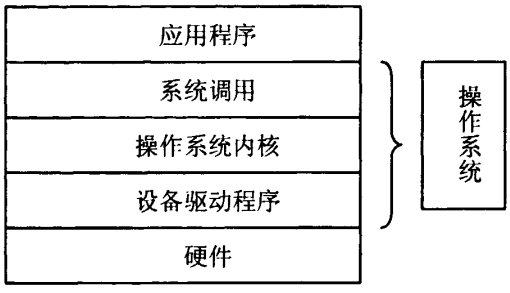


图 2-6 硬件、设备驱动程序、操作系统和应用程序的关系

底层设备驱动程序是硬件和操作系统内核的接口，驱动程序直接操作硬件，并且为操作系统所调用。

当应用程序要对硬件操作时，它使用 Linux 系统调用接口，而 Linux 调用驱动程序，驱动程序实现对硬件的操作。

2.4.4 底层设备驱动程序

Linux 系统将设备分成三种类型[16]：字符设备、块设备、网络设备。字符设备的读写以字节为单位，存取时没有缓存；块设备读写以块为单位，存取时有缓存支持以提高效率；网络设备用于通信，在 Linux 里做专门的处理。

Linux 是一种类 Unix 系统，Unix 的一个基本特点是“一切皆为文件”[17]。它将所有的硬件设备都像普通文件一样看待，也就是说硬件可以像普通文件一样来打开、读写和关闭。Linux 系统中的设备都用一个特殊文件表示，叫做设备文件。相应的，设备文件分为字符设备文件、块设备文件、网络设备文件。

Linux 设备驱动程序完成以下的功能：

1. 设备初始化、释放；
2. 把数据从内核传送到硬件和从硬件读取数据；
3. 读取应用程序传送给设备文件的数据和回送应用程序请求的数据；
4. 检测和处理设备工作过程中出现的错误。

为了使移植了嵌入式 Linux 操作系统的数字存储示波器充分利用和操作硬件，必须编写硬件设备驱动程序。在本系统中主要包括以下几类设备驱动程序：GPIO 驱动程序、SPI 驱动程序、FPGA 驱动程序和 LCD 驱动程序等。

GPIO 驱动是指利用 S3C2410 的通用 I/O 口控制硬件的一类驱动。GPIO 驱动程序包括前端衰减器驱动、前端通道控制驱动和触发控制驱动。前端衰减器驱动的作用是控制通道的电压幅度，这通过控制可控衰减芯片 PE4302 完成。前端通道控制驱动是控制通道的 1/10 衰减和直流交流耦合的功能，以及触发通道的选择。触发控制驱动主要完成触发源的选择和触发方式的选择等。

SPI 驱动是指利用 SPI（Serial Peripheral Interface，串行外围接口）控制硬件的一类驱动。SPI

驱动程序包括锁相环 PLL 及时钟控制驱动，AD 转换器驱动和板上 DA 电平控制驱动。锁相环 PLL 及时钟控制驱动通过控制芯片 ADF4360-7DA 来为系统提供时钟功能。AD 转换器驱动通过控制芯片 ADC08D500 来完成 AD 的初始化。电平控制驱动是用来控制每一个通道的直流偏移位置，以及使用电平触发方式时的触发电平。

FPGA 驱动程序采用内存映射的方式，将 FPGA 的寄存器映射到 ARM 的地址空间里去，ARM 从 FPGA 中读取数据。

LCD 驱动程序利用 Linux 为显示设备提供的帧缓冲接口将显示缓冲区抽象，屏蔽图像底层硬件的底层差异，允许上层应用程序在图形模式下直接对显示缓冲区进行读写操作。用户不必关心物理显示缓冲区的具体位置及存放方式，这些都由帧缓冲设备驱动本身来完成。

上面的数字存储示波器的设备驱动都属于字符设备驱动。

Linux 内核中采用可加载的模块化设计（LKMs，Loadable Kernel Modules）[18]，Linux 设备驱动属于内核的一部分，Linux 内核的一个模块可以以两种方式被编译和加载：

（1）直接编译进 Linux 内核，随同 Linux 启动时加载；

（2）编译成一个可加载和删除的模块，使用 insmod 加载（modprobe 和 insmod 命令类似，但依赖于相关的配置文件），rmmod 删除。这种方式控制了内核的大小，而模块一旦被插入内核，它就和内核其他部分一样。

本课题中，对于每一个设备首先编写对应的设备驱动程序和测试程序，以模块的方式加载到内核中，用测试程序对设备驱动程序进行测试，查看硬件是否如预期的工作，如果不正确，调试驱动程序，直到调试成功，这时把这些驱动直接编译到内核（如 LCD 驱动）或是在 Linux 的启动脚本中加入加载这些驱动模块的命令，这样就会在示波器的每次启动时加载它们。

2.4.5 应用程序

应用程序的总体流程如图 2-7 所示。数字存储示波器整个系统的应用程序包括系统初始化模块，按键扫描模块，键消息处理模块，系统控制模块，数据采集模块，数据处理模块，显示处理模块等七个部分。

系统初始化模块对整个系统进行配置，主要是打开数字存储示波器各种硬件驱动设备，设置各种硬件的初始状态，初始化各种参数：时基档位、幅度档位、菜单选择、采样方式、显示方式、耦合方式、水平位置、探头衰减、触发源、触发方式、触发电平、语言选择等。

按键扫描模块的主要功能是接收单片机通过串口发送过来的键盘或旋钮的键值。详细的介绍和实现见第四章。

键消息处理模块主要的功能是根据按键扫描模块得到的用户按下的键或是旋钮的键值来改变示波器对应的系统参数变量的值。

系统控制模块根据系统参数变量的值的改变执行相应的控制，包括硬件状态的改变，系统状态的改变等。

数据采集模块根据采集模式（采样、峰值检测、平均）的不同和是否是 SCAN 方式做不同的采集处理。

数据处理模块包括波形重构，幅度类和时间类测量，波形运算和频域分析与处理等，详细的内容见第三章。

显示处理模块将要显示的波形和其它信息显示到 LCD 上，在第五章作详细的阐述。

程序的流程是上电后首先执行系统初始化，配置好系统运行环境，接着进行按键扫描，如果有按键按下，则进行键消息处理，设置对应的系统参数的值，系统控制模块根据系统参数的值的改变来进行相应的控制，接着进行数据采集模块、数据处理模块和显示处理模块；如果没有按键按下，则直接进入数据处理模块、数据处理模块和显示处理模块。

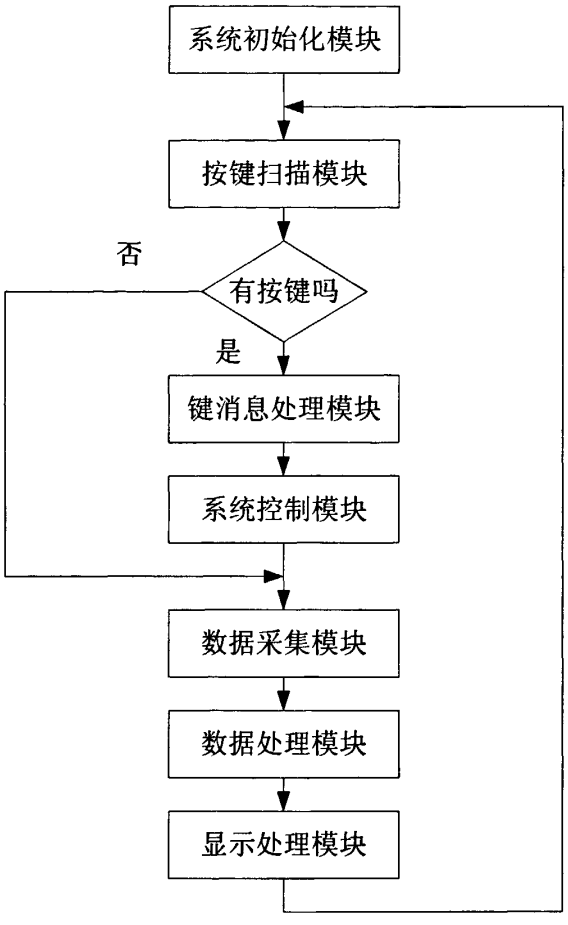


图 2-7 应用程序的总体流程图

第三章　数据处理模块

数字存储示波器的一大特点便是可以对采集的数据进行必要的处理[19]，通常的模拟示波器只能将波形显示在显示屏上，而对于被测信号的量化分析只能靠人工进行计算。而数字示波器可以通过软件编写好的算法对于测量的结果进行多种运算，从而得到所需要的被测信号的详细特征。

本章的数据处理模块内容包括了波形重构、幅度类和时间类参数测量、波形运算、窗函数和 FFT。高速宽带数字示波器对高频信号进行高速采样时，仅能采集信号的少数样点，为了看清信号的全貌和精确测量其参数，必须在每两个样点之间进行插值以便重构波形。幅度类和时间类测量的测量参数包括有平均值、峰峰值、周期、频率、有效值、最大值、最小值、上升时间、下降时间、正频宽、负频宽等。离散傅里叶变换（DFT）是分析信号频域特征的重要工具，FFT 是它的快速算法，能量泄漏和栅栏效应是 FFT 算法引起的两种固有误差，对原始数据进行加窗函数处理可以减少泄漏误差。在下面的内容分别进行介绍。

3.1 波形重构

示波器是时域测量仪器，要求它不失真地显示波形。数字存储示波器工作时，将按一定的时间间隔对信号电压进行采样，用模数变换器(ADC)对这些瞬时采样点进行变换而生成代表每个采样点电压的二进制字，然后利用这些数据在屏幕上重建波形。波形重构的方法与采样方式有关，下面首先介绍数字存储示波器的几种采样方式。

3.1.1 采样方式

采样方式[20][21]分为实时采样和非实时采样（也称等效采样），非实时采样又分为顺序采样和随机采样。

3.1.1.1 实时采样

实时采样是一种最基本的采样方式。如果在信号实际经历的时间内完成了全部采样，称为实时采样。实时取样的原则是所有的采样点都是按照一个固定的次序来采集的。波形采样的次序和采样点在示波器屏幕上出现的次序是相同的。只要一个触发时间就可以启动全部采集动作，如图 3-1 所示。

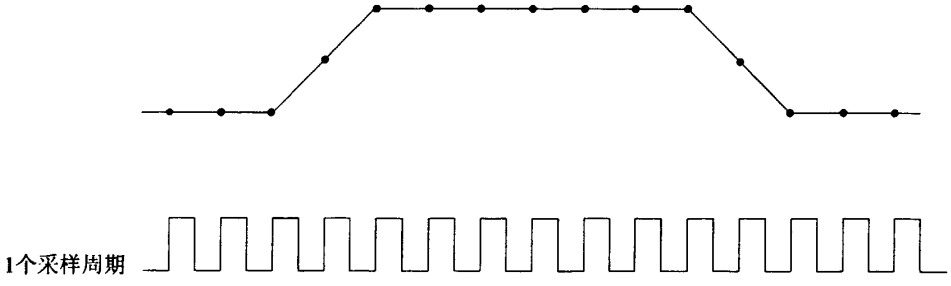


图 3-1 实时采样

采用实时采样的示波器测量重复信号和测量单次信号具有相同的带宽，也称实时带宽(Real-Time BW)。为了提高实时带宽，必须提高采样速率。根据采样定理，采样频率至少必须两倍于被测信号的上限频率。但是由于实际被测信号有可能存在高频分量，为了不违背采样定理，避免混淆现象发生，实际中实时采样 DSO 的采样频率一般规定为带宽的 4-5 倍，同时还必须采用适当的内插算法，如果不用内插算法，则一般规定采样速率为实时带宽的 10 倍。

实时采样的优点是适用于所有信号，是对单次信号测量的唯一选择。缺点是带宽直接与采样率有关；采样率和存储器的容量限制了时间间隔测量的分辨率和预触发范围；在慢扫描速率时对假波敏感。

本课题所设计的数字存储示波器采用实时采样，最高采样率为 1GSa/s。时基范围为 2ns/div～50s/div，当以最快的采样速率 1GSa/s 进行采样时，1ns 采样一个点，而每一格 25 个点，所以在时基为 2ns /div、 5ns /div、 10ns /div 、20ns /div 时需要插值产生足够的数据点以供液晶显示。

3.1.1.2 顺序采样

顺序采样通常对周期为 T 的信号每经大约 m 个周期采集一点（m 为正整数），但每次采样都比前次在波形的相对位置上滞后△t，也就是每经（mT+△t）采集一个点，如图 3-2 所示。这种采样方式对采样速度的要求大大降低了，或者说可以用不是非常高的采样速率，“等效”极高的采样速率，进而使示波器的通频带做的很宽。这种采样方式常用于取样示波器中。

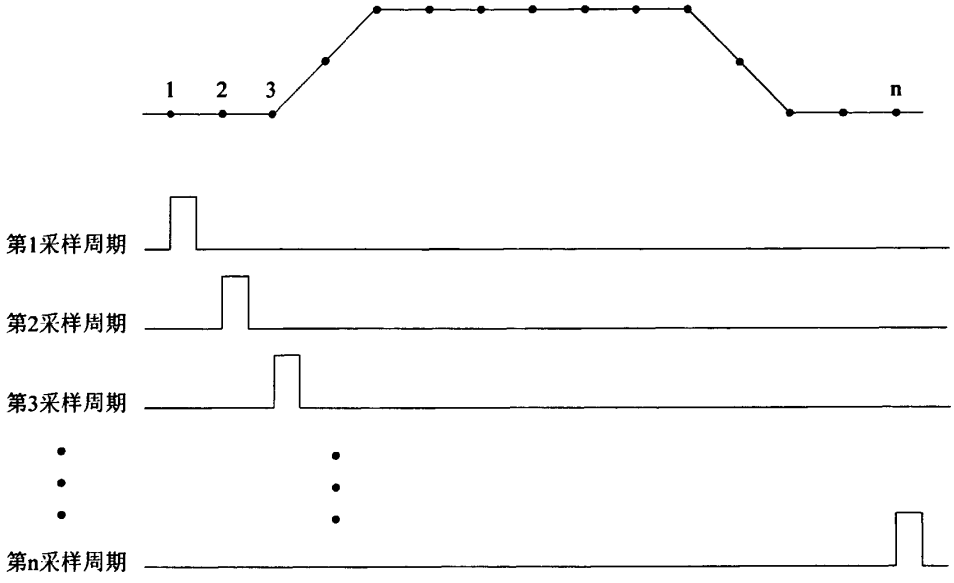


图 3-2 顺序采样

顺序采样的优点：最高等效取样率高达 50GHz；极低噪声；由于采样是以相对较慢速进行 A/D 转换器可使用高的分辨率。缺点：不具备预触发的能力；在慢扫描速度下对假波敏感；不能进行单次信号测量。

3.1.1.3 随机采样

随机采样，是指每个采样周期采集一定数量的采样点，经过多个采集周期的采样点积累，最终恢复出被测波形。图 3-3 是随机采样的原理图。由于信号与采样时钟之间是非同步的，使得每个采集周期的触发点与下一个采样点之间的间隔是随机的，所以称为随机采样。又因为信号是周期的，可以将每个采集周期的采样等效为对由触发点确定的“同一段时间”的采样。因而通过多个采样周期后，以触发点为基准将各采集周期的样点拼合，可以得到一个重复信号的由触发点确定的一段波形的密集的样点，这样就恢复了这段波形。

随机采样的优点：通常带宽高于实时采样率；预触发范围不受采样率和存储器容量限制；重复信号对假波不敏感。缺点是对高速单次信号测量不适合。

顺序采样与随机采样都属于等效时间采样，在这两种取样方式下，示波器的带宽称为等效带宽。它们的共同之处是都只能观测周期信号。但同时它们也有很大的不同。首先，顺序采样的采样点与触发脉冲有△t 的延迟时间关系，而随机采样的采样点与触发脉冲无任何关系，完全是随机的。其次，顺序采样触发后每个信号周期只有一个采样点，而随机采样每个信号周期可以获得一组采样点。随机采样相比于顺序采样最大的优点在于能够提供预触发信息。

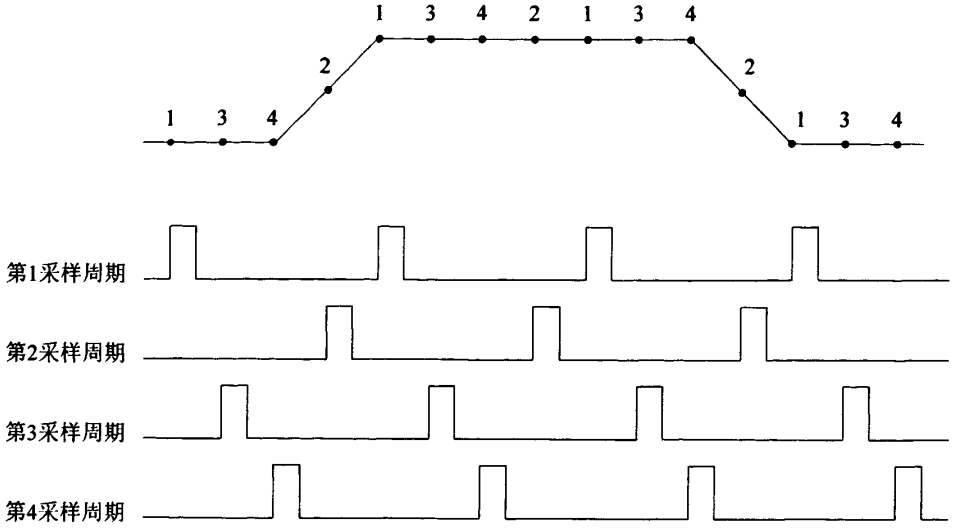


图 3-3 随机采样

3.1.2 插值原理

根据采样定理，对一个频带限制在 0- f_h 内的时间连续信号 $x(t)$ 进行采样，只要采样频率 $f_s \geq 2f_h$，就可以利用采样点序列无失真的重建原始信号 $x(t)$。

当采样频率满足采样定理时，让采样后信号通过一个截止频率为 π / T 的低通滤波器即可恢复波形。由采样序列值恢复出原信号的公式如下：

$$x(t)=\sum_{n=-\infty}^{\infty}x(nT)\frac{\sin\left[\frac{\pi}{T}(t-nT)\right]}{\frac{\pi}{T}(t-nT)} \tag{3.1}$$

其中，T 是采样间隔，是采样频率 f_s 的倒数，nT 代表第 n 个采样点的时间。

$\frac{\sin\left[\frac{\pi}{T}(t-nT)\right]}{\frac{\pi}{T}(t-nT)}$ 称为内插函数，它的普遍形式是 $\mathrm{Sa}(x)=\frac{\sin(x)}{x}$，函数波形如图 3-4 所示，它可以用来插补采样点间的空隙，相应内插方法称为正弦内插。内插函数的特点是：在采样点 nT 上函数值为 1，其余采样点函数值为 0。

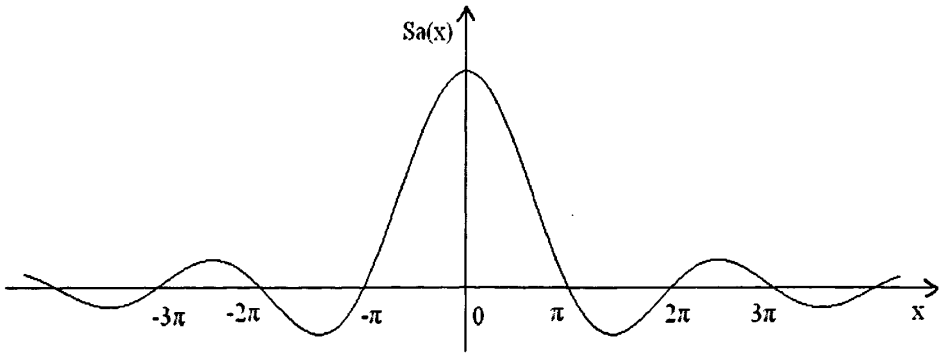


图3-4 Sa(x)函数

本数字存储示波器采用实时采样方式，示波器屏幕上看到的波形是由存储器中的采样点重建出来的信号波形。示波器在屏幕上显示出这些采样点，并在这些采样点之间画出连线。产生非采样点显示的数据点可以按几种方法来做，最简单的方法是在各个采样点之间用直线连接，这种方法称为为线性内插，只要各采样点之间距离得很近，用这种方法就能获得足够的重建波形，如果在信号跳变沿前后都采集了采样点，那么用这种方法就可以观察跳变沿。当波形的采样点在水平方向距离很远时，在屏幕上显示出一条通过各采样点的连续的曲线就比在采点之间用直线连接要好得多，为此可使用正弦内插法。采用这种方法时，在屏幕上将各个采集的采样点用幅度和频率均为可变的最佳正弦拟合曲线连接。采用了内插的方法以后，即使当幕上每格的采样点数较少时也能得到和模拟示波器显示波形类似的自然平滑的重建波形。

为了察看真正的采样点，示波器通常设有点显示方式，在此方式之下，不使用任何内插方法。选择这种方式以后，我们在屏幕上只能看到用离散亮点表示采样点，而在这些点之间没有任何连线。

3.1.3 插值程序

本数字存储示波器采用正弦内插法。因为实时采样为等间距采样，波形恢复公式可以变形为：

$$x_p(n_1,n_2)=\sum_{n=-\infty}^{+\infty}x(n)\frac{\sin(\pi*\Delta)}{\pi*\Delta} \tag{3.2}$$

式3.2中，$x_p(n_1,n_2)$为相邻第n_1个和第n_2个采样点之间要插的第p个点的值。$x(n)$为第n个采样点的值，Δ由两个采样点之间要插的点数m和p以及n_1和n位置来计算，若两个采样点之间要插m个点，p必定小于m，则根据公式$\Delta=(p/m)+n_1-n$可以计算出来。

理论上∑是从采样时间为负无穷大的采样点到采样时间正无穷大的采样点，这在编程实现起来不现实，经过实验，当一个插值点用80个采样点来计算时，已经满足本系统的精度，再增大采样点数来计算，插值误差几乎没有减小。从理论上来分析，把一个波形用傅立叶级数展开，主瓣以及前后两个旁瓣贡献的幅度能量最多，距离主瓣越远的旁瓣贡献的幅度能量越少。本示波器的垂直分辨率是8bits，也就是说插值点的插值精度要达到1/256，同时考虑计算速度，我们采用前后共80个点进行插值运算，插值公式变为

$$x_p(n_1,n_2)=\sum_{n=n_1-40}^{n_1+39}x(n)\frac{\sin[\pi(n_1-n+p/m)]}{\pi(n_1-n+p/m)} \tag{3.3}$$

由式 3.3 可以看出，计算一个插值点需要计算几十次的浮点数的乘法和除法，如果第n_1点和n_2点的插值点用n_1点前面的 40 个和包含n_1在内的后面的 40 个点来计算，第n_1点和n_2点之间要插值的点与第n_2和n_3点之间要插的具有相同的$\{\sin(\pi*\Delta)/(\pi*\Delta)\}$部分，所以可以把这部分数据先计算好做成表放在数组中，只需在计算的过程中根据m和p的值在数组中查出相应的$\{\sin(\pi*\Delta)/(\pi*\Delta)\}$值，再做一次与相应的采样点乘法运算，最后求 80 个乘积之和。另外$\{\sin(\pi*\Delta)/(\pi*\Delta)\}$值都是浮点数，可以将这些系数都乘以一个较大的正数，这里乘以 2 的 15 次方，使得这些系数变成一个整数，这样浮点运算就变为定点运算，所以将$2^{15}*\sin(\pi*\Delta)/(\pi*\Delta)$预先计算好，再做一次与相应的采样点乘法运算，然后求 80 个乘积之和，最后将这和除以2^{15}即可。

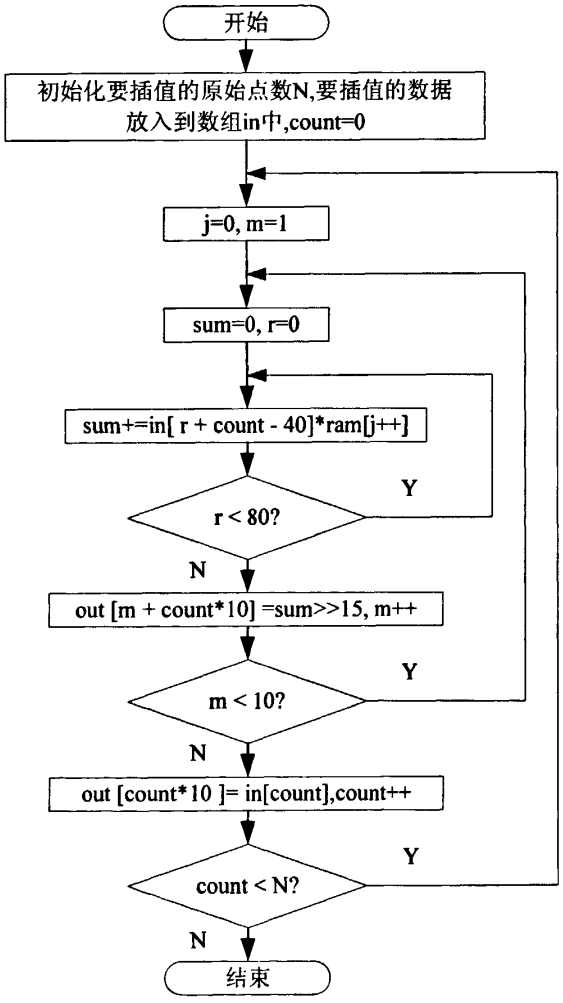


图3-5 9点插值流程图

本数字存储示波器采用实时采样方式，最高采样率为 1GSa/s，则最小采样间隔为 1ns ，而示波

器一个时基刻度含有 25 个点，若不插值则最低时基档位为 25ns/div,所以示波器处于 20 ns/div, 10 ns/div, 5 ns/div 和 2 ns/div 的时基需要插值才可以恢复显示波形，我们采用正弦内插法。20 ns/div, 10 ns/div, 5 ns/div 和 2 ns/div 都采用 1GSa/s 的采样率，20 ns/div 的实现是先隔 8 点抽取然后每两点间内插 9 个点，10 ns/div 的实现是先隔 4 点抽取然后每两点间内插 9 个点，5 ns/div 的实现是先隔 2 点抽取然后每两点间内插 9 个点，2 ns/div 的实现是先隔 2 点取然后每两点间内插 24 个点。

以9点插值为例，插值程序的流程图如图3-5所示。插值程序分为三层循环，最里面的循环是求两个采样点之间一个插值点的值，即80个采样点数据与对应的表格数据做80次循环的乘法。中间一层循环是要计算出两个采样点之间要插值的点的值，即m个插值点的值。最外层的循环是计算出所有采样点之间要插值的点的值。首先将 $2^{15} * \sin(\pi * \Delta) / (\pi * \Delta)$ 计算好存放到ram数组中以便下面的程序使用。初始化要插值的原始点数N，并将这N个原始数据以参数的方式传递给输入数组in。count变量是用来判断有没有已全部完成N个原始数据的插值，在开始将count置为0。变量j是用来查询数组ram中 $2^{15} * \sin(\pi * \Delta) / (\pi * \Delta)$ 对应的下标。变量m是用来判断有没有在两个原始点间插满9个点。变量r是用来判断求和是否累计了80次。程序按照如下公式

$$out[m + count * 10] = (\sum_{r=0}^{+79} in[r + count - 40] * ram[j++]) / 2^{15}, m = 1,2,\cdots,9$$

计算插值点的值，数组out是用来保存插值后的数据。

3.2 参数测量

参数测量分为幅度类参数测量和时间类参数测量。参数测量和显示有两种方法，一种方法是首先选择一级菜单 MEASURE（测量）菜单，然后在二级菜单中选出要测量的通道和参数，选定后，被选中的参数和通道都会显示于屏幕右边的菜单显示区。另一种测量参数的方法是用光标测量，首先选择一级菜单 CURSOR（光标）菜单，然后选择二级菜单中的幅度或时间，这样就会计算并显示出两条幅度光标线的各自的值及其差值或者两条时间光标线的各自的值及其差值。计算幅度类参数的基本依据是两个通道的量程，计算时间类参数的基本依据是时基。

参数测量都是通过对采集的数据进行分析来进行的，所以，参数测量的结果都源于在示波器中存贮的采集到的波形，这就意味着，示波器的设置情况对参数测量和结果会有影响。例如，如果示波器的时基速度设置得比较慢，比如说设置为 1ms/格，而要对一个为 50ns 至 100ns 的上升沿进行上升时间测量，那么由于采集过程中时间分辨率的限制，我们就无法测出正确的结果，为了进行这项参数测量，我们应当把示波器的时基档位设置得足够快，例如设置为 50ns/格以便以足够细的时间分辨率显示出被测波形的上升沿。

为了获得最精确的测量结果，一般来说参数测量应尽可能的在高扫速上进行。测量时的所需波形部分应显示在屏幕上[22]，即：

(1)周期和频率测量时至少有一个完整的周期被显示；

(2)脉宽测量要求整个脉冲被显示；

(3)下降时间测试时波形的下降沿必须被显示；

(4)上升时间测量时波形的上升沿必须被显示；

另外，如果显示不止一个波形、边沿或脉冲，测量将在最先出现的部分进行。

3.2.1 幅度类参数测量

幅度类参数包括：最大值、最小值、峰峰值、平均值、均方根值等。

本数字存储示波器采用的是 8 位 AD，所以波形数据的范围为 0-255。

最大值的计算是先将 max 赋值为 0，在整个数据长度中将每一个数据和 max 比较，如果大于

max，就将这个数据赋给 max，依次将所有数据和 max 比较，比较完后 max 中存放的就是最大值。

同理，最小值的计算是先将 min 赋值为 255，在整个数据长度中将每一个数据和 max 比较，如果小于 min，就将这个数据赋给 min，依次将所有数据和 min 比较，比较完后 min 中存放的就是最小值。

峰峰值就是最大值和最小值之差。

上面的最大值，最小值，峰峰值体现不出量程的信息，只有把得到的最大值，最小值，峰峰值与当前的量程档位结合起来才可以表达出准确的波形信息。

平均值的计算是先把整个数据长度的采样点进行平均，然后与采样值作为零点的 128 进行比较，若大于 128 认为平均值为正，若小于 128 认为平均值为负，然后在与当前的量程结合起来得到波形的平均值信息。

有关有效值的计算方法如下：

TDS700 定义：$V_{rms}=\sqrt{(\sum_{i=1}^{n}(V_i^2+V_i\times V_{i+1}+V_{i+1}^2)/3)/n}$

HP54501 定义：$V_{rms}=\sqrt{(\sum_{i=1}^{n}V_i^2-\sum_{i=1}^{n}V_i)/n}$

在这里，有效值按下面的公式计算：

$$V_{rms}=\sqrt{(x_1^2+x_2^2+\cdots+x_n^2)/n} \tag{3.4}$$

其中，$x_1\cdots x_n$ 为 n 个采样点与零电平之间的差值。由公式得到的 RMS 与量程结合起来可以得到波形的有效值信息。

3.2.2 时间类参数测量

时间类参数包括：频率、周期、上升时间、下降时间、正频宽、负频宽等。其中最为重要的是频率测量。信号频率测量的方法有很多，在本次设计中我们采用了软件测频与硬件测频相结合的方法。对频率在 10KHz 以上的信号采取硬件测频，10KHz 以下采用软件测频。

3.2.2.1 硬件测频

硬件测频法，就是在给定的闸门信号中填入脉冲，通过必要的记数电路，得到填充的脉冲个数，从而算出待测信号的频率和周期。

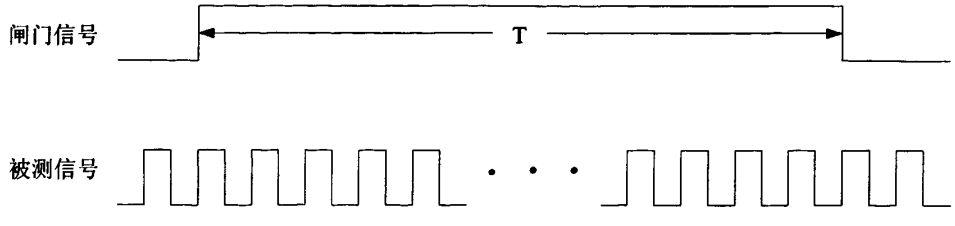


图 3-6 高频信号时硬件测频示意图

当被测信号频率较高时，通常选用频率较低的一个标准频率信号作为闸门信号，而将被测信号作为填补脉冲，在固定闸门的时间内对其记数，如图 3-6 所示。设闸门宽度为 T，记数值为 N，则这种测量方法的频率测量值为：

$$f_x=N/T \tag{3.5}$$

测量误差主要是对被测信号记数产生的±1 的误差，在忽略闸门信号自身误差的情况下，测量精度为：$\Delta f_x = \pm 1/T$。

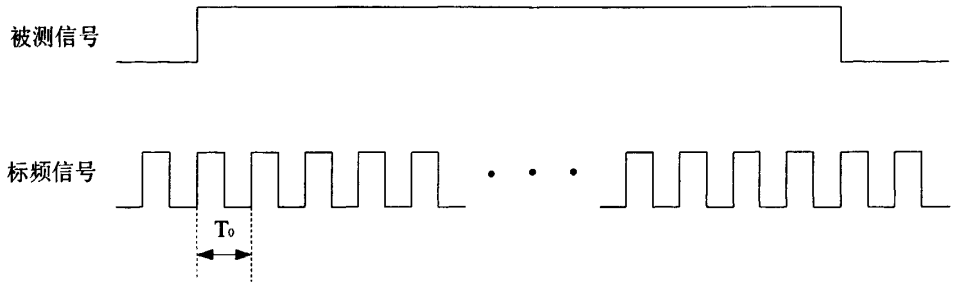


图 3-7 低频信号时硬件测频示意图

当被测信号频率较低时，通常选用被测信号作为闸门信号，而将频率较高的标准频率信号（标频信号）作为填充脉冲，进行记数，如图 3-7 所示。设计数值为 N，标频信号的频率为 f_0，周期为 T_0。则这种测量方法的频率测量值为：

$$f_x = f_0/N \tag{3.6}$$

3.2.2.2 软件测频

软件测频，与所选择的触发窗口的宽度和波形幅度相关[23]，在此我们定为波形峰峰值的 10%，计算在某一段波形穿过触发窗口的个数，然后计算出波形的周期和频率，在图 3-8 中，令 Vh=VPP/2+5%*VPP，Vl=VPP/2-5%*VPP，从数据区首地址开始找大于 Vh 的点，找到点后，再找小于 Vl 的点，把这一点的位置赋给 start，然后继续找大于 Vh 的点，再找小于 Vl 的点，令 periodCount=1。依次类推，每次找到大于 Vh 的点，再找到小于 Vl 的点后 periodCount 就加 1，这样将在数据区找到的最后大于 Vh 的点，小于 Vl 的点位置赋给 end，那么 periodCount 就是找到的周期个数，根据时基以及 start 和 end 的值，就可以计算出周期。

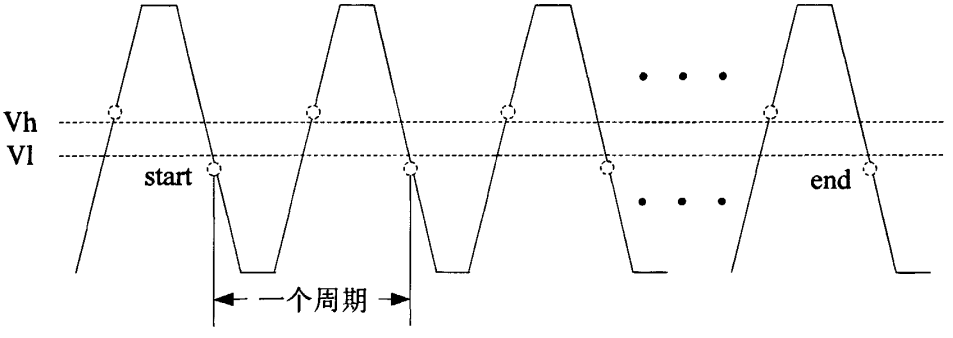


图 3-8 测量周期数示意图

周期计算公式如下：

$$\text{period}=(\text{end}-\text{start})/\ \text{periodCount}*(\text{TimeBase}/25) \tag{3.7}$$

其中 TimeBase 代表时基，由于时基是确定的，因此点与点之间的时间是确定，这样就可以根据起始点和结束点的位置（采样点的序号）和周期个数计算波形的周期。

测出了被测信号的周期后将该时间的倒数作为频率，使用公式：f=1/T，其准确度取决于周期

的测量精度。

3.2.2.3 上升时间、下降时间、正频宽、负频宽

脉冲波形参数[24][25]如图 3-9 所示。V_{top}（幅度顶值）和V_{base}（幅度底值）就是脉冲波形的 100%和 0%电平值，是脉冲参数自动测量的核心。确定了V_{top}和V_{base}值，才能计算脉冲其它参数值。

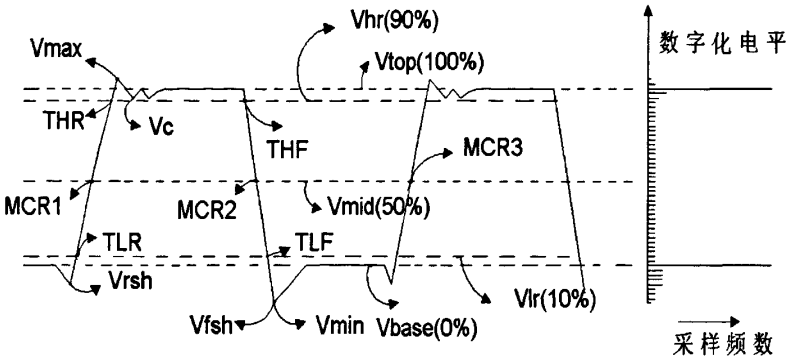


图 3-9 脉冲波形参数变量示意图

波形的顶值和底值不一定是波形的最大值和最小值。譬如，有小毛刺的脉冲波形，因为波形常常稳定在这些小毛刺下面，如果把最大值作为顶线就是错误的。

当脉冲顶部和底部有严重失真时，顶值和底值往往很难确定。有关脉冲参数标准文件推荐可用频数密度分布统计平均法或密度分布众数法确定V_{top}和V_{base}值。数字示波器借助自动参数测量功能，普遍采用“频数直方图密度分布众数算法”确定V_{top}和V_{base}值。

算法的基本出发点是做波形样点幅度频数直方图。直方图纵坐标是数字化电平，横坐标是数字化电平出现的样点次数。在波形中间电平上方和下方，样点出现次数最多的量化电平值就分别是V_{top}和V_{base}。HP 公司数字示波器还规定，最常出现的点是指样点出现次数要多于一屏样点总数的 5%，否则就认定$V_{top}=V_{\max}$、$V_{base}=V_{\min}$。顶值和底值的流程图如图 3-10 所示。

但对于正弦波，波形的$V_{top}=V_{\max}$，$V_{base}=V_{\min}$，在实际测量过程中，不能发现密度分布众数点，按 HP 公司的规定，自然取$V_{top}=V_{\max}$，$V_{base}=V_{\min}$。

求得顶值V_{top}和底值V_{base}后就可以按如下方法计算出其它参数：

1. 计算$V_{\mathrm{hr}}=V_{top}\times 90\%$，$V_{\mathrm{mid}}=V_{top}\times 50\%$，$V_{\mathrm{lr}}=V_{top}\times 10\%$

2. 在一个周期内寻找对应V_{hr}的时间点 THR 和 THF，对应V_{mid}的时间点 MCR1、MCR2 和

MCR3，对应V_{lr}的时间点 TLR 和 TLF

3. 上升时间：$t_r = (THR - TLR) \times TimeBase / 25$

4. 下降时间：$t_f = (THF - TLF) \times TimeBase / 25$

5. 正频宽：$+\tau = (MCR2 - MCR1) \times TimeBase / 25$

6. 负频宽：$-\tau = (MCR3 - MCR2) \times TimeBase / 25$

其中 TimeBase 是时基档位，时基档位的每一格中有 25 个点。

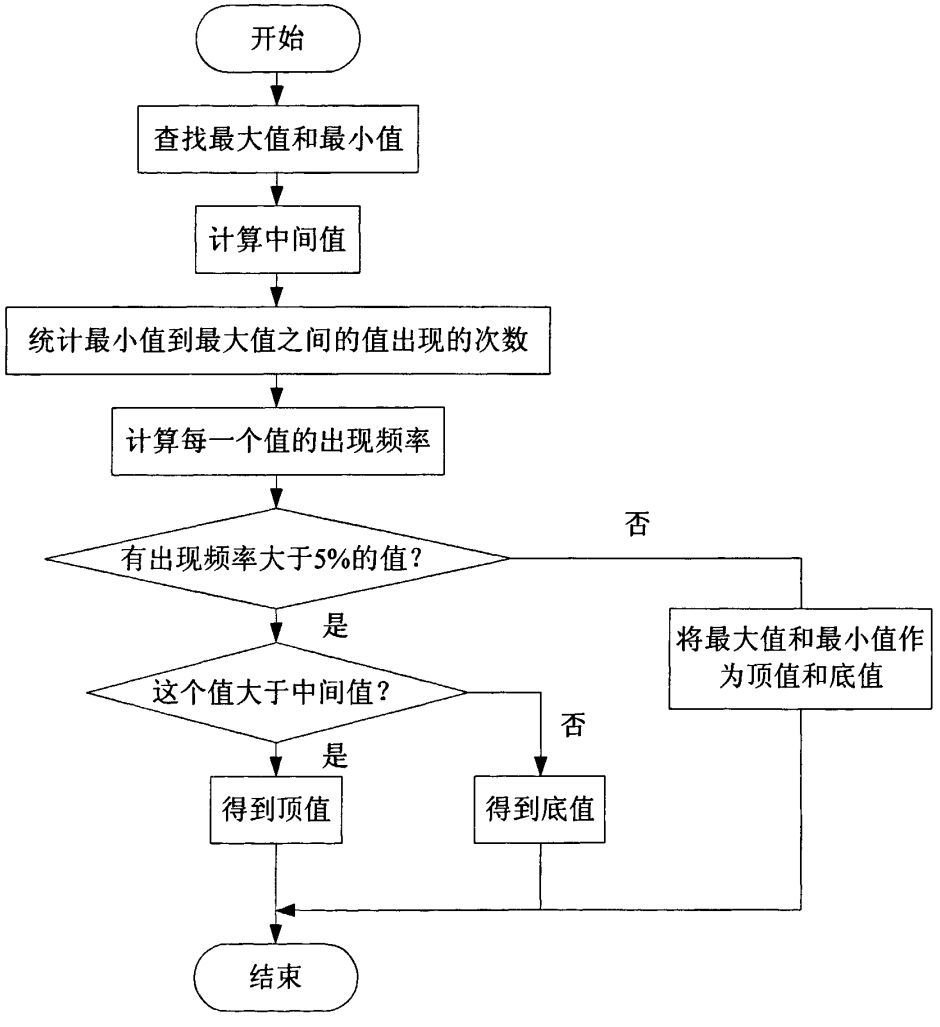


图 3-10 求顶值和底值的流程图

3.2.3 光标测量

在本课题中，数字存储示波器除了能提供信号幅度，周期等参数的自动测量外，还提供光标测量功能，以满足灵活的测量需要。最常用的光标测量功能，是用两条可在屏幕上任意移动的水平光标线，测量波形两线之间的幅度差；用两条可在屏幕上任意移动的垂直光标线，测量两线之间的时

间差。测量结果在屏幕上的自动显示。这种测量易于满足用户的某些特殊需要，例如可方便的测出脉冲顶部寄生振荡的周期，振荡振幅等，使用非常灵活。

光标测量的实现如下：首先判断光标的类型，如果光标的类型是电压，则进一步判断测量的通道，得到要测量的通道的幅度档位，根据两条光标线的位置（相对于零电平）和所测量的通道的幅度档位计算出光标所在位置的电压，它们的差的绝对值即为增量；如果光标的类型是时间，就根据两光标线的位置（相对于触发时间为 0）和时间档位计算出光标所在位置的时间，它们的差的绝对值即为增量。

3.3 波形运算

数字存储示波器除了能准确捕获信号，并给出精确的测试结果，还希望有一定的波形运算能力。国内外数字示波器无一例外地采用了一些常见的波形处理技术：如加，减及乘等。根据不同档次的数字示波器，具有不同的波形处理技术，归纳起来，常常采用的有波形相加、波形相减、波形相乘、波形相比、波形取反和波形积分等波形处理技术。

本数字存储示波器实现了波形相加、波形相减、波形取反、波形相乘。在做波形相加、波形相减和波形相乘时，通道 1 和通道 2 的幅度档位可以不同，这时波形相加、波形相减和波形相乘后的波形是两通道波形的对应点的真观之和、之差、之积。如果希望看到它们的真正之和、之差、之积，只需将两通道的幅度档位设置相同即可。波形取反是将原始通道信号相对地电位旋转 180 度，这可以通过求原始信号关于地电位对称的值而得到。

3.4 频域分析

3.4.1 窗函数理论

在信号处理中不可避免的要遇到数据截断问题。例如在应用 DFT 时，数据 $x(n)$ 总是有限长，在功率谱估计中也要遇到对自相关函数的截断问题。总之，我们在实际工作中能处理的离散序列总是有限长，把一个长序列变成有限长的短序列不可避免的要遇到窗函数[26]。因此，窗函数本身的研究及应用是信号处理中的一个基本问题。

设 $x(n)$ 为一长序列，$w(n)$ 是长度为 N 的窗函数，$n = 0,1,\cdots,N-1$，用 $w(n)$ 乘 $x(n)$ 得：

$$x_N(n) = x(n)w(n) \tag{3.8}$$

$x_N(n)$ 为 N 点序列，其 DTFT 是

$$X_N(e^{jw}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\theta})W(e^{j(w-\theta)})d\theta \tag{3.9}$$

由此可以看出，窗函数 $w(n)$ 不仅影响原来信号在时域的形状，也影响了其在频域的形状。

下面给出几种常用的窗函数。

（1） 矩形窗

$$w(n) = R_N(n) \tag{3.10}$$

（2） 汉宁（Hanning）窗

又称升余弦窗，$w(n) = \frac{1}{2}\left[1-\cos\left(\frac{2\pi n}{N-1}\right)\right]R_N(n)$ （3.11）

（3） 汉明（Hamming）窗

又称改进的升余弦窗，$w(n) = \left[0.54-0.46\cos\left(\frac{2\pi n}{N-1}\right)\right]R_N(n)$ （3.12）

（4） 布莱克曼（Blackman）窗

又称二阶升余弦窗，窗函数如下：

$$w(n) = \left[0.42-0.5\cos\left(\frac{2\pi n}{N-1}\right)+0.08\cos\left(\frac{4\pi n}{N-1}\right)\right]R_N(n) \quad (3.13)$$

（5） 平顶（flat top）窗

$$w(n) = \left[1-1.93\cos(\frac{2\pi n}{N-1})+1.29\cos(\frac{4\pi n}{N-1})-0.388\cos(\frac{6\pi n}{N-1})+0.032\cos(\frac{8\pi n}{N-1})\right]R_N(n) \quad (3.14)$$

以矩形窗为例，当$w(n) = R_N(n)$时，它的频谱为

$$W(e^{jw}) = \sum_{n=0}^{N-1} w(n)e^{-jwn} = \sum_{n=0}^{N-1} R_N(n)e^{-jwn} = e^{-j(N-1)w/2}\sin\left(\frac{wN}{2}\right)\Big/\sin\left(\frac{w}{2}\right)$$

可用幅度函数与相位函数来表示$W(e^{jw})$为

$$W(e^{jw}) = W_R(w)e^{-jw\alpha} \quad (3.15)$$

其线性相位部分$e^{-jw\alpha}$只是表示时延$\alpha = (N-1)/2$，对频响起影响的部分是它的幅度函数

$$W_R(w) = \frac{\sin(wN/2)}{w/2} \quad (3.16)$$

它的形状如图3-11所示。

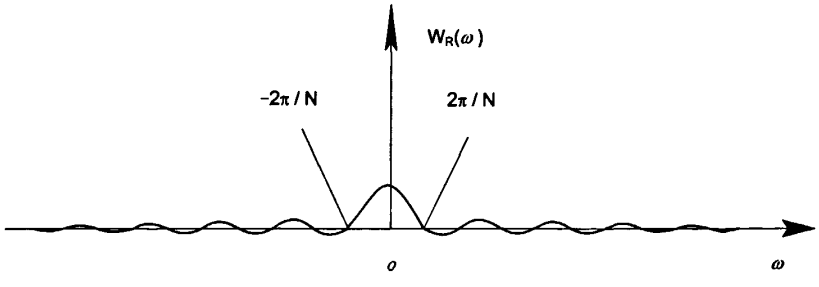


图3-11 $W_R(w)$函数

$|w| \le 2\pi/N$的部分称为窗函数的主瓣，在$|w| > 2\pi/N$部分称为窗函数的旁瓣。主瓣宽度决定

了被截断以后所得序列的频域分辨率，而旁瓣峰值有可能湮没信号频谱分量中较小的成分。

对窗函数总的要求是，希望它频谱的主瓣尽量的窄，旁瓣峰值尽量的小，使频域的能量主要集中在主瓣内。

归一化$|W(e^{jw})|=20\lg(|W(e^{jw})/W(e^{j0})|)$，下面几个频域指标[27]以定量的比较各种窗函数的性能：

1) 主瓣宽度B_0

2) 3db带宽B：它是主瓣归一化的幅度下降到-3db带宽时的带宽，令$\Delta w=\frac{2\pi}{N}$，则B可以以Δw为单位

3) 最大旁瓣峰值A（dB）

4) 旁瓣谱峰渐近衰减速度D（dB/oct）

表3.1是上面几种窗的性能指标[27][28][29]。

表3.1 几种窗函数的性能指标

窗类型	3dB带宽B	主瓣宽度B_0	最大旁瓣峰值A	旁瓣衰减速度D
矩形窗	0.89Δw	$4\pi/N$	-13dB	6dB/八倍频程
汉宁窗	1.44Δw	$8\pi/N$	-32dB	18dB/八倍频程
汉明窗	1.30Δw	$8\pi/N$	-43dB	6dB/八倍频程
布莱克曼窗	1.68Δw	$12\pi/N$	-58dB	18dB/八倍频程
平顶窗	3.72Δw	$20\pi/N$	-44dB	6dB/八倍频程

由上表可以看出，矩形窗具有最窄的主瓣,但也有最大的旁瓣峰值和最慢的衰减速度，平顶窗的主瓣最宽，有较小的旁瓣峰值，布莱克曼窗有最小的旁瓣峰值和最快的衰减速度，但主瓣较宽。而汉宁窗和汉明窗主瓣稍宽，但有较小的旁瓣。

3.4.2FFT原理

傅里叶变换是一种将信号从时域变换到频域的变换形式，是信号处理中的重要的分析工具，卷积，相关，滤波，谱估计等都可以化为DFT（离散傅里叶变换）来实现。但是，直至20世纪60年代，由于数字计算机的处理速度较低以及离散傅里叶变换的计算量较大，离散傅里叶变换长期得不到真正的应用，直到快速离散傅里叶变换（FFT）算法的提出，才显现出离散傅里叶变换的强大功能，并被广泛应用于各种数字信号处理系统中[30]。

DFT的公式如下：

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk}\qquad k=0,1,\cdots,N-1 \tag{3.17}$$

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk}\qquad n=0,1,\cdots,N-1 \tag{3.18}$$

显然，求出N点$X(k)$需要N^2次复数乘法和$N(N-1)$次复数加法，而实现一次复数乘法需要四次实数乘法和两次实数加法，实现一次复数加法需要两次加法，所以需要整个DFT运算总共需要$4N^2$次实数乘法和$2N(2N-1)$次实数加法。随着序列长度N的增大，运算次数将急剧增加。20世纪

60 年代中期 Cooley 和 Tukey 提出了一种离散傅里叶变换的快速算法，它所需的运算量约为 $\frac{N}{2}\log_2 N$ 次复数乘法和 $N\log_2 N$ 次复数加法，这种快速算法的出现大大推动了离散傅里叶变换在各方面的应用，继 Cooley 和 Tukey 之后，又有许多人致力于进一步减少 DFT 的运算量，相继提出了一些改进算法。而目前比较普遍使用的算法，还是基于 Cooley 和 Tukey 提出的基二算法。

快速算法利用了 W_N^{nk} 的以下性质：

1. W_N^{nk} 的对称性：$(W_N^{nk})^* = W_N^{-nk}$
2. W_N^{nk} 的周期性：$W_N^{nk} = W_N^{(n+N)k} = W_N^{n(k+N)}$
3. W_N^{nk} 的可约性：$W_N^{nk} = W_{mN}^{nmk}, W_N^{nk} = W_{N/m}^{nk/m}$

基二算法又分为按时间抽取 (DIT) 法和按频率抽取 (DIF) 法。下面介绍按时间抽取 (DIT) 法。

设序列 $x(n)$ 长度为 N，且满足 $N= 2^M$，M 为正整数。如果不满足这个条件，可以人为的加上若干个零值点，使之达到这一要求。按 n 的奇偶把 $x(n)$ 分解为两个 $N/2$ 点的子序列：

$$X(k) = DFT[x(n)] = \sum_{n=0}^{N-1} x(n)W_N^{nk} = \sum_{\substack{n=0 \\ n\text{为偶数}}}^{N-1} x(n)W_N^{nk} + \sum_{\substack{n=0 \\ n\text{为奇数}}}^{N-1} x(n)W_N^{nk}$$

$$= \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} = \sum_{r=0}^{\frac{N}{2}-1} x(2r)W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x(2r+1)W_{N/2}^{rk}$$

$$= \sum_{r=0}^{\frac{N}{2}-1} g(r)W_N^{2rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} h(r)W_N^{2rk} \qquad g(r) = x(2r),\ h(r) = x(2r+1)$$

$$= G(k) + W_N^K H(k)$$

其中 $G(k) = \sum_{r=0}^{\frac{N}{2}-1} g(r)W_{\frac{N}{2}}^{rk}$，$H(k) = \sum_{r=0}^{\frac{N}{2}-1} h(r)W_{\frac{N}{2}}^{rk}$

利用 G（k）、H（k）的周期特性，有

$$X(k) = G(k) + W_N^k H(k), \qquad k = 0,1,2,\cdots,\frac{N}{2}-1 \tag{3.19}$$

$$X(k+\frac{N}{2}) = G(k) - W_N^k H(k), \qquad k = 0,1,2,\cdots,\frac{N}{2}-1 \tag{3.20}$$

可见，一个 N 点序列 $x(n)$ 的 DFT 可以从两个 N/2 点序列的 DFT 求出。以此类推，$G(k)$ 和 $H(k)$ 可以继续分解下去，这种按时间抽取算法是在输入序列分成越来越小的子序列上执行 DFT 运算，最后再合成为 N 点的 DFT。

公式 3.19 和 3.20 可以表示为蝶形运算结构，如图 3-12 所示。

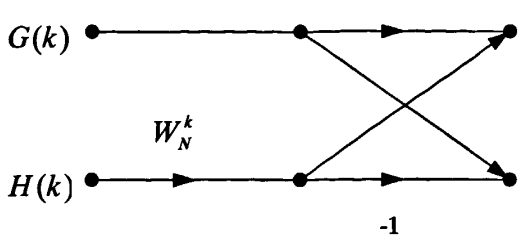


图 3-12 蝶形运算结构

N=8 按时间抽取法 FFT 运算流图如图 3-13 所示。由图可以看出，变换后的输出序列 X(k)依照正序排列，但输入序列 x(n)不再是原来的自然顺序，这种顺序被称作倒位序，即将原自然顺序的数的二进制数按位翻转的结果。例如 1 的二进制数为 001，按位翻转后变为 100，即为 4。所以在进行按时间抽取法的 FFT 前要先将按自然顺序排列的输入数据倒序。

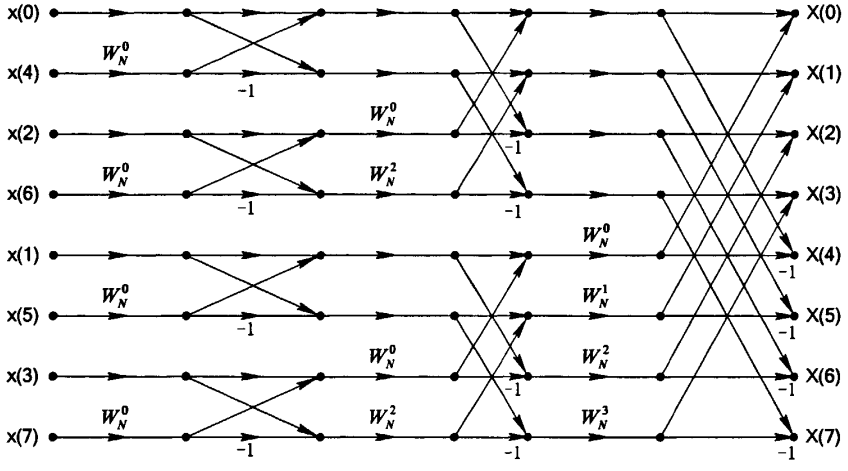


图 3-13 时间抽取法 8 点 FFT 运算流图

3.4.3DSO 中的窗函数和 FFT

数字存储示波器的 Math 菜单下有 FFT 的选项，可以通过选择来决定是做 CH1 还是 CH2 的 FFT，同时还有一个选项来决定加的窗函数的类型。窗函数的类型有矩形窗，汉宁窗和平顶窗三种。

表 3.2 信号频谱与窗函数的选择

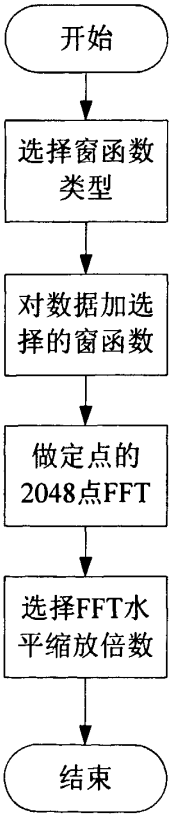
窗类型	信号类型	窗特性
矩形窗	宽带随机 分段逼近正弦波	主瓣窄、旁瓣衰减速度低、频率分辨率较低
汉宁窗	窄带随机信号 信号频谱未知 正弦波或正弦信号的组合	瓣幅峰值较大、频率分辨率高、可减小泄漏、旁瓣衰减速度快
平顶窗	要求较高幅度精度的正弦波	幅度精度较高、主瓣较宽、频率分辨率低、频谱泄漏大

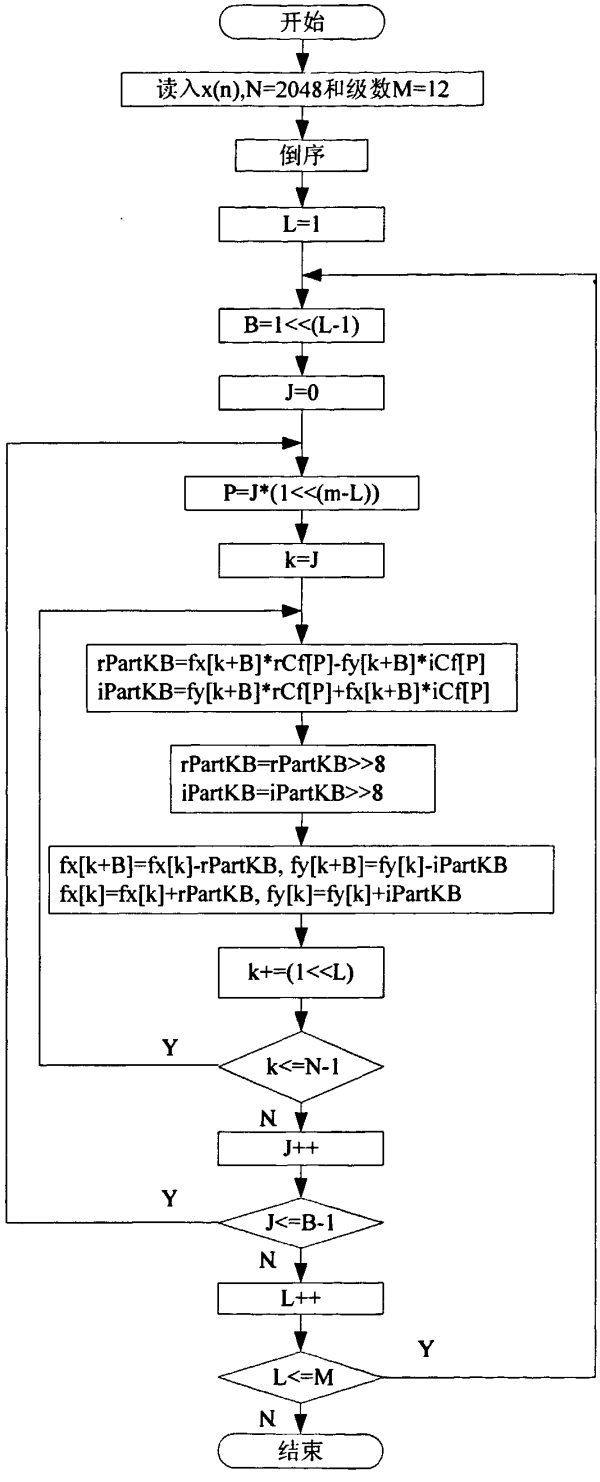
每种窗函数有其自身的特性,不同的窗函数适用于不同的应用[29]，如表 3.2 所示。要选择正确的窗函数,必须先估计信号的频谱成份。如若信号中有许多远离被测频率的强干扰频率分量，应选择旁

瓣衰减速度较快的窗函数；如果强干扰频率分量紧邻被测频率时，应选择旁瓣峰值较小的窗函数；如果被测信号含有两个或两个以上的频率成份，应选用主瓣很窄的窗函数；如果是单一频率信号，且要求幅度精度较高，则推荐用宽主瓣的窗函数。绝大多数应用中采用汉宁（Hanning）窗即可得到满意的结果，因为它具有较好的频率分辨率和抑制频谱泄漏的能力。

数字存储示波器中的 FFT 要求是 2048 点，并不要求编写一个通用的针对数据长度为任何 2 的倍数的 FFT 程序，由于 N=2048 是固定的，所以W_N^{nk}对于相应的n和k它们也是固定的，又因为它们是浮点的，所以将它们乘以一个较大的数转化为整数（在最后结果中要除以这个较大的数），存在一个表格里，这样就将浮点运算转化定点运算，进一步提高了运算速度。

关于对信号加窗和做 FFT 的流程图见图 3-14。由图 3-14（b）可以看出，FFT 运算包括三层循环。第一层循环是控制蝶形运算的级数，由于 N=2048=2^M，所以级数 M=12。第二层循环和第三层循环完成第 L 级蝶形运算。图中 rCf[]和 iCf[]数组即是W_N^{nk}的实部和虚部再乘以 256 得到的数据（它们是预先计算好的），所以流程图里有 rPartKB=rPartKB>>8;iPartKB=iPartKB>>8;是将数据结果右移 8 位（即除以 256），这样做的好处是避免了每次循环都计算W_N^{nk}，同时也避免了浮点运算，提高了 FFT 的速度。





（a）DSO 中加窗函数和 FFT 操作流程图　　(b)实现 FFT 的流程图

图 3-14 DSO 中窗函数和 FFT

第四章 键处理模块

作为一台智能化的现代电子测量仪器，良好的人机交互界面是必须的。本系统中选用了AT89C2051单片机来独立的管理示波器的键盘和旋钮，同时选用了一块320×240点阵的TFT液晶屏作为系统的显示部件。S3C2410微处理器通过串口和AT89C2051单片机进行通信，获取键语，并对键语进行分析，进而改变LCD屏幕上的波形及菜单的显示。S3C2410微处理器内置了LCD控制器，支持多种型号的LCD屏，可以很方便地和LCD屏接口。键盘控制与显示模块使得仪器的操作更加简单方便，显示更加直观，符合现代仪器小型化、智能化和具有良好人机交互的发展方向。本章就详细讨论键处理模块，下一章讨论显示处理模块。

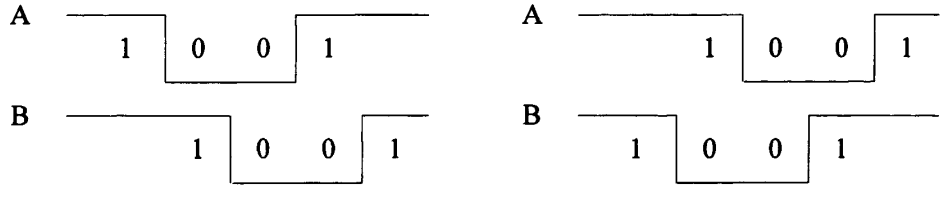
4.1 键盘和旋钮的原理

键盘作为传统的输入设备是和用户进行交互所不可缺少的部分。本仪器采用传统的结实耐用的机械式键盘。机械式键盘实质上是一组开关的集合，它利用触点的断开和闭合产生电信号[31]。本仪器采用矩阵式键盘，用AT89C2051单片机单独管理键盘，采用程序扫描工作方式，并且使用软件延迟法消除按键抖动。

除按键外，本仪器上还有旋钮。在数字仪器中，旋钮开关实际上是一个编码开关，它相当于将两个按键式开关的一端连在一起成为公共端，另一端分别引出，我们称之为引出端A和B。当旋钮开关旋动时，旋钮内部与A、B相连的两个金属簧片都会与接地的公共端相接，但是在时间上会有一个短暂的时间差，因此在A、B上就会产生如图4-1所示的两个有一定相位差的脉冲信号。例如，在旋钮逆时针旋的过程中，引出端A的簧片先接触公共端接地，引出端B的簧片经过一个短暂时差后也接触公共端接地，随后A、B簧片又先后与公共端断开，而呈现高电平。这样就会得到如图4-1（a）所示的两路脉冲信号。同理，当旋钮顺时针旋时，会得到如图4-1（b）所示的两路脉冲信号，此时B簧片要超前一个相位比A簧片先接触公共端。在键盘扫描过程中，如果我们能设法判断出两路脉冲到达的相位关系，就可以知道旋钮的旋动方向，从而进行相应的操作。

有两种方法可以用来判别脉冲到达的先后顺序，一种是硬件的办法，采用鉴相器，通过专门的电路判断，再将结果送至CPU；还有一种就是用软件的方法进行判断。对旋钮而言是要能表示两种动作：逆时针旋和顺时针旋。而在旋钮静止状态，旋钮所占用的信号线既可能导通又可能截止。所以在矩阵式键盘上对旋钮动作的识别是：首先行线上按一定顺序依次出现低电平，若旋钮无动作时，旋钮所占用的两条列线的状态不变；旋钮有动作时，旋钮所占用的两条列线的状态会连续变化，而且变化是有规律的：11->01->00->10（逆时针旋）或11->10->00->01（顺时针旋）。据此可以判断旋钮的旋转方向。

采用这种根据序列码的规律来判断旋钮旋转方向的方法可以抗干扰，提高可靠性。因为逆时针旋和顺时针旋的序列都是唯一的，而且它们之间的差别很大。



(a)顺时针旋动　　(b)顺时针旋动

图4-1 编码开关引出端的脉冲信号图

4.2 键盘扫描程序

在本系统中，键盘由23个按键式开关和7个旋钮开关构成，相当于37个按键式开关，于是我

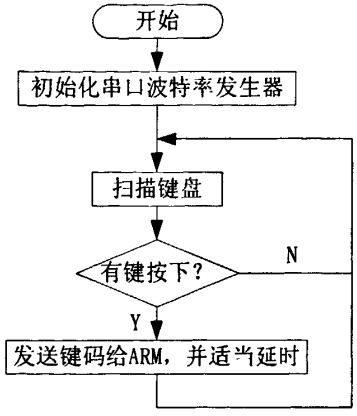


图 4-2 键盘主程序流程图

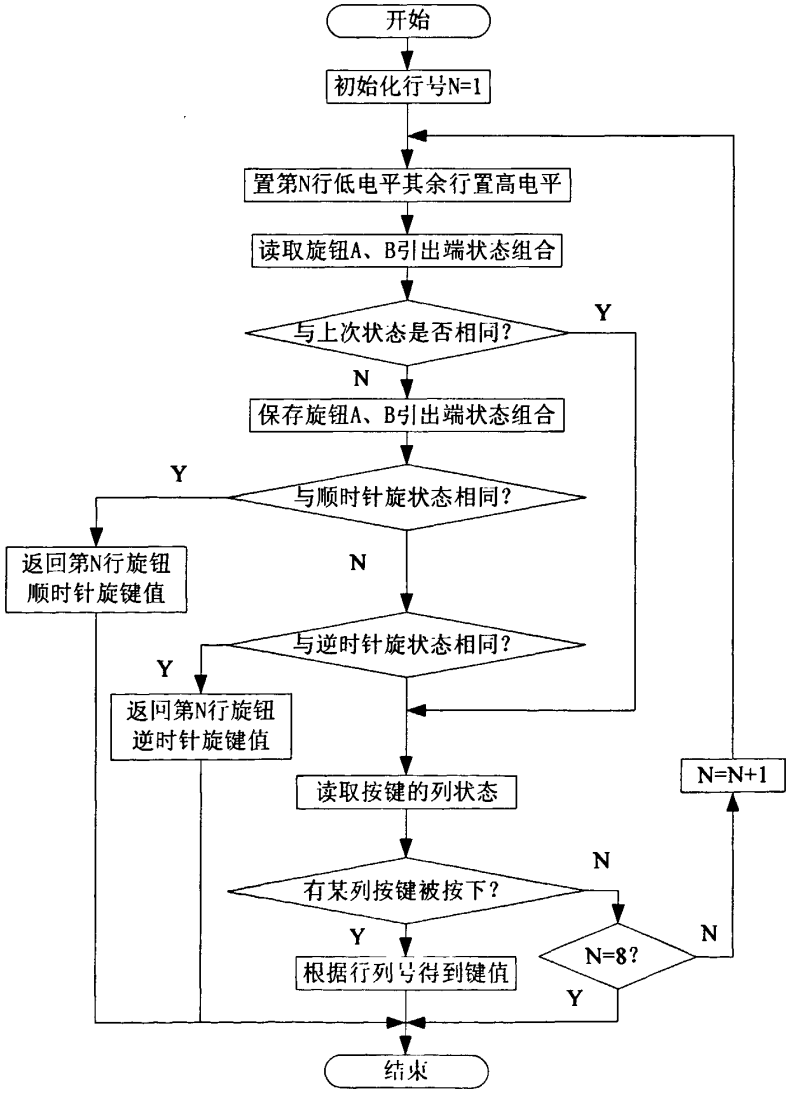


图 4-3 键盘扫描子程序流程图

们将键盘按照 8×5 的矩阵键盘结构进行组织，其中二列为按钮式按键，另两列分别接 7 个旋钮开关的 A、B 引出端。

采用编程扫描法识别按键，键盘主程序流程图和键盘扫描子程序流程图分别见图 4-2 和图 4-3。由于键盘面板中包括按键和旋钮，所以在编程的时候要区别对待。首先将第 N 行置低电平，其余行置高电平，读取旋钮的 A、B 引出端的状态值，若其与以前的值相等，依次检查各列电平变化，若某列电平由高变低则表示此列与置零电平的行的交叉处的按键被按下。若 A、B 引出端的状态值与之前的值不同则表示是旋钮的转动，通过判断旋的规律来确定是顺时针旋还是逆时针旋。

4.3 Linux 下串口接收程序的实现

由于示波器的键盘是通过串口与 ARM 连接的，所以还要编写 Linux 下的串口接收程序。在 Linux 下，所有的硬件资源都虚拟成文件，通过虚拟设备文件统一管理硬件设备。所有的设备文件一般都位于/dev 目录下，其中串口一和串口二对应的设备名依次为“/dev/ttyS0”，“/dev/ttyS1”。在 Linux 下对设备的操作方法和对文件的操作方法是一样的，因此对串口的读写就可以使用简单的 read，write 函数完成，所不同的是需要对串口的其它参数进行配置，下面就对常用的设置进行说明。

Linux 下串口[32]的设置包括波特率，数据位，校验位，停止位，输入输出方式等，而这些设置主要是通过设置 struct termios 结构体的各个成员值。这个结构体定义了串口配置需要使用的一些数据结构和参数，它的定义如下：

```
struct  termio
{
        unsigned short c_iflag;        //输入模式标志
        unsigned short c_oflag;        //输出模式标志
        unsigned short c_cflag;        //控制模式标志
        unsigned short c_lflag;        //本地模式标志
        unsigned char c_line;          //行标识
        unsigned char c_cc[NCC];       //控制字符
}
```

在这个结构体中最为重要的是 c_cflag，通过对它的赋值，用户可以设置波特率、字符大小、数据位、停止位、奇偶校验位和硬件流控等。另外 c_iflag 和 c_cc 也是比较常用的标志。

下面给出串口初始化设置：

```
termios_new.c_cflag  =  B19200 //设置波特率为 19200
termios_new.c_cflag &= ~CSIZE; //设置 8 位数据位
termios_new.c_cflag |= CS8;
termios_new.c_cflag &= ~PARENB; //设置无奇偶校验
termios_new.c_cflag &= ~CSTOPB; //设置一位停止位
/*下面三行把串口设置成 raw 模式，这样的话就可以保证文件传输过程中底层程序不会处理传输内容*/
termios_new.c_cflag |= CLOCAL | CREAD;
termios_new.c_iflag &= ~(INLCR | ICRNL | IGNCR | IXON | IXOFF | IUCLC);
termios_new.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG);
termios_new.c_oflag &= ~OPOST;
```

需要注意的是，串口的设置非常重要，关系到串口能否正常工作。由于键盘发送的是单字节的十六进制数据，所以必需将串口所对应的终端设置成原始（raw）工作方式，就是不必对输入输出处理。

初始化结束后就可以读取键盘发送过来的数据了。首先要打开串口：

fd=open(“/dev/ttyS1”,O_RDWR)；接下来用 read 函数进行读取 read(fd,buff,1)，表示读串口一个字节到变量 buff 中。另外还要注意的是 ARM 在 Linux 下的串口接收程序的波特率一定要和单片机键盘扫描程序的波特率一致，不然接收不到正确的数据。

4.4 各种键功能的实现

4.4.1 数字存储示波器的各种键

该数字存储示波器 23 个按键和 7 个旋钮，每一个按键和旋钮都对应着特定的功能。23 个按键对应的分别是：

1. DISPLAY（显示）
2. ACQUIRE：(获取)
3. SAVE/RECALL（存储/调出）
4. UTILITY（辅助功能）
5. CURSOR（光标）
6. MEASURE（测量）
7. AUTO SET（自动设置）
8. HELP（帮助）
9. CH1 MENU（通道 1 菜单）
10. CH2 MENU（通道 2 菜单）
11. MATH MENU（数学运算菜单）
12. HORI MENU（水平菜单）
13. TRIG MENU（触发菜单）
14. RUN/STOP（运行/停止）
15. SINGLE（单次）
16. SET TO 0（设置水平位置为 0）
17. FORCE TRIG（强制触发）
18. SET TO 50%（设置触发电平为最小和最大电压电平的一半）

另外还有 5 个二级菜单选项按键。1-13 的每一个按键对应的是一个一级菜单。当用户按下 1-13 中的一个一级菜单按键后，在屏幕上都会出现该一级菜单对应的二级菜单，通过按下二级菜单中的一个按键来改变二级菜单选项。14-18 的每一个按键没有自己的子菜单，当它们被按下后，它们会产生一个相应的动作。

七个旋钮分别是：CH1 量程旋钮，Vertical Position 1(垂直位置 1)旋钮，CH2 量程旋钮，Vertical Position 2(垂直位置 2)旋钮，时基旋钮，水平位置旋钮，触发电平旋钮。旋钮也只是改变状态，而没有自己的子菜单。

数字存储示波器的各种键的功能不同，但可以大致分为一级菜单按键，二级菜单按键，特殊功能键和旋钮，各种功能键的处理流程图如图 4-4 所示。

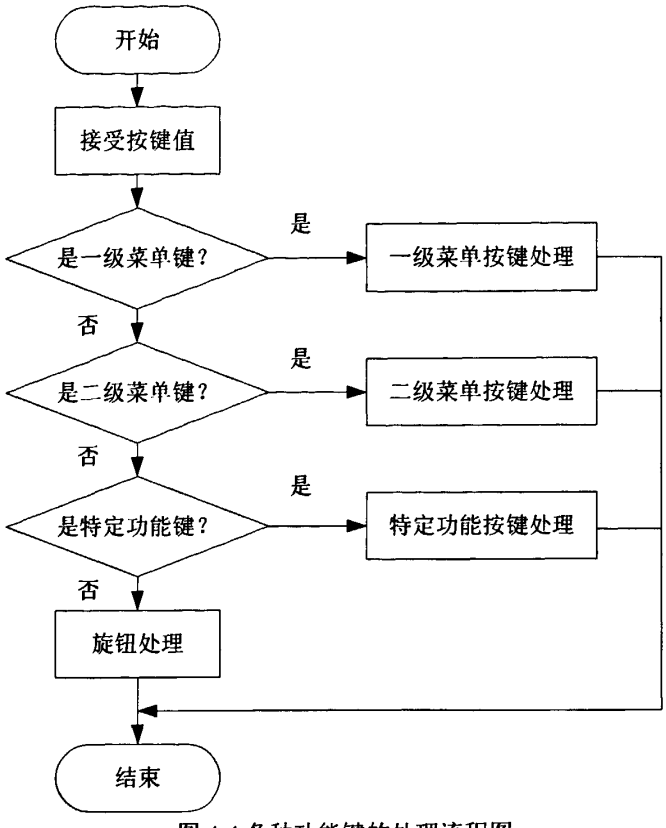


图 4-4 各种功能键的处理流程图

4.4.2 菜单功能键的设计与实现

一级菜单按键和二级菜单按键的通常的实现方法是通过 if 语句判断或是 switch 语句，例如像下面这样：

```
switch (keyValue)
 {
 case DISPLAY_KEY：
        menu= displayMenu;
        break;
 case SAVE_RECALL_KEY：
        menu= saveRecallMenu;
        break;
      ……              //还有十多个键，类似处理，这里省略
 case KEY1:
      swith(menu)
      {
      case displayMenu：
            DisplayMenuKey1();
            break;
      case saveRecallMenu:
           SaveRecallMenuKey1();
           break;
```

```
            ……              //还有十多个菜单，类似处理，这里省略
        }
    case KEY2:
        switch (menu)
        {
        case displayMenu：
              DisplayMenu Key2();
              break;
        case saveRecallMenu:
             SaveRecallMenuKey2();
             break;
             ……              //还有十多个菜单，类似处理，这里省略
        }
    ……
    case KEY5:
        switch (menu)
        {
        case displayMenu：
              DisplayMenu Key5();
              break;
        case saveRecallMenu:
             SaveRecallMenuKey5();
             break;
            ……      //还有十多个菜单，类似处理，这里省略
        }
    }
```

如果这样实现，使得程序结构不清晰且效率不高。C++虚函数具有在程序运行时能够依据类型的不同(通常称为运行时类型识别 RTTI) 而确定正确的匹配函数的能力，C++在编译时采用“滞后编译”(亦称后期绑定或动态绑定) 的技术来实现这种能力，以使函数能呈现“多态”[33]。本系统在设计一级菜单按键和二级菜单按键的实现时就是使用了 C++中的虚函数，以实现二级菜单按键的多态功能（在不同的一级菜单下，按下二级菜单按键，有不同的功能）。

首先是定义一个基类，在这个基类里用 virtual 声明成员函数为虚函数，这样就可以在派生类里重新定义此函数，为它定义新的功能。如下定义了一级菜单的抽象类，“=0”表示它是纯虚函数，没有自己的函数实现。包含一个或多个纯虚函数的类，称为抽象类。

```
class Menu
{
      public:
        virtual int Key1( ) =0;
        virtual int Key2( ) =0;
        virtual int Key3( ) =0;
        virtual int Key4( ) =0;
        virtual int Key5( ) =0;
};
```

然后，定义派生类。定义每一个一级菜单类，这些类是类 Menu 的派生类，是以公有继承的方

式继承类 Menu 的。并且重新定义派生类的函数体，例如类 DisplayMenu 的成员函数 Key1（）的功能是实现显示菜单的第一个二级菜单按键的功能，类 SaveRecallMenu 的成员函数 Key1（）的功能是实现存储调出菜单的第一个二级菜单按键的功能。

```
class DisplayMenu:public Menu
{
        public:
          virtual int Key1( );
          virtual int Key2( );
          virtual int Key3( );
          virtual int Key4( );
          virtual int Key5( );
};

class SaveRecallMenu:public Menu
{
        public:
          virtual int Key1();
          virtual int Key2();
          virtual int Key3();
          virtual int Key4();
          virtual int Key5();

};
……                          //还有其他一级菜单的类，类似定义，这里省略
```

然后定义指向基类的指针和派生类的对象。要在 C++中取得多态性的行为，被调用的函数必须是虚函数，而对象则必须是通过指针或者引用去操作的[34]。基类 Menu 的指针和派生类的对象定义如下：

```
Menu *menu;
DisplayMenu displayMenu;
SaveRecallMenu saveRecallMenu;
……                          //还有其他一级菜单的类的对象，类似定义，这里省略
```

最后，只需要将基类指针指向派生类的指针就可以调用派生类的函数。在本仪器中需要做的只是在那个一级菜单对应键按下后将它的对象赋给基类指针即可，而当二级菜单按下后，不用再用大量的 swith 语句，很简洁的实现了二级菜单按键的多态，如下面阴影部分所示。

```
switch (keyValue)
{
case DISPLAY_KEY：
        menu = &displayMenu;    //将显示菜单的对象赋给基类 menu 指针
        break;
case SAVE_RECALL_KEY：
        menu= &saveRecallMenu;
        break;
    ……                      //还有十多个键，类似处理，这里省略
```

```
    case KEY5:
        menu->Key5();
        break;
    case KEY4:
        menu->Key4();
        break;
    case KEY3:
        menu->Key3();
        break;
    case KEY2:
        menu->Key2();
        break;
  case KEY1:
        menu->Key1();
        break;

  }
```

综上所述，通常的实现（通过 if 语句判断或是 switch 语句）分支比较多，程序结构不清晰。根据按键的特点，我们利用了 C++编程中的虚函数机制创建一个虚基类，它下面有五个函数（它们是虚的，没有函数实现体），然后 13 个一级菜单都是由它派生而来，真正的函数实现是在派生类里完成的，这五个函数在五个按键之一被按下时就会调用对应的实现函数，实现了动态绑定。

一级菜单按键处理：将按下键的菜单的对象赋给指向父类的指针，并将它的二级菜单赋给要显示的菜单变量。

二级菜单按键处理：由于这时父类对象指针指向的就是当前的一级菜单，所以按下二级菜单键时，程序就会自动调用对应的函数，而不再需要判断现在的一级菜单是什么。不同的一级菜单下的二级菜单函数有不同的功能，我们需要做的是编写这些函数，而函数与函数之间是相对独立的，这样易于软件的团队开发。

4.4.3 旋钮功能的实现

量程、时基旋钮功能比较单一，量程、时基增大和减小每次只能改变一个档位，而且必须在量程时基范围之内。在液晶屏的左下角显示，如：CH2 2V 表示示波器的通道 2 量程为 2V/div。量程时基改变后要重新显示其最新状态。在示波器处于视窗扩展状态时，时基旋钮用来调整触发窗的大小。量程时基增大减小旋钮在系统处于回调波形状态或 STOP 状态时锁定。

当波形显示处于一般状态时，垂直位置 1 旋钮即是 CH1 垂直位置旋钮，旋动它可以调整通道 1 的波形的零点使得通道 1 的波形上下移动；垂直位置 2 旋钮即是 CH2 垂直位置旋钮，旋动它可以调整通道 2 的波形的零点使得通道 2 的波形上下移动。水平位置旋钮用来调触发点，选择了触发点后，左右移动此旋钮的作用就是调预触发点的位置，触发电平旋钮用以调节触发信号波形上的触发点的相应电平值，使在这一电平上启动扫描。用以改变扫描电路的工作状态，一般应为待触发状态，使用时只需调整旋钮，即能使波形稳定显示。当示波器处于视窗扩展状态时，水平位置旋钮移动的是视窗的位置。

在波形处于光标测量状态时，垂直位置 1 旋钮和垂直位置 2 旋钮分别用来调节光标 1 和光标 2，当测量幅度类光标的时候，逆时针旋、顺时针旋就可以使光标向上移动、向下移动，当测量时间类光标的时候，逆时针旋、顺时针旋就可以使光标向左移动、向右移动。

垂直位置 1 旋钮功能相对较为复杂，垂直位置 1 旋钮顺时针旋动功能的流程图如图 4-5 所示，其他的旋钮可以类似的实现。

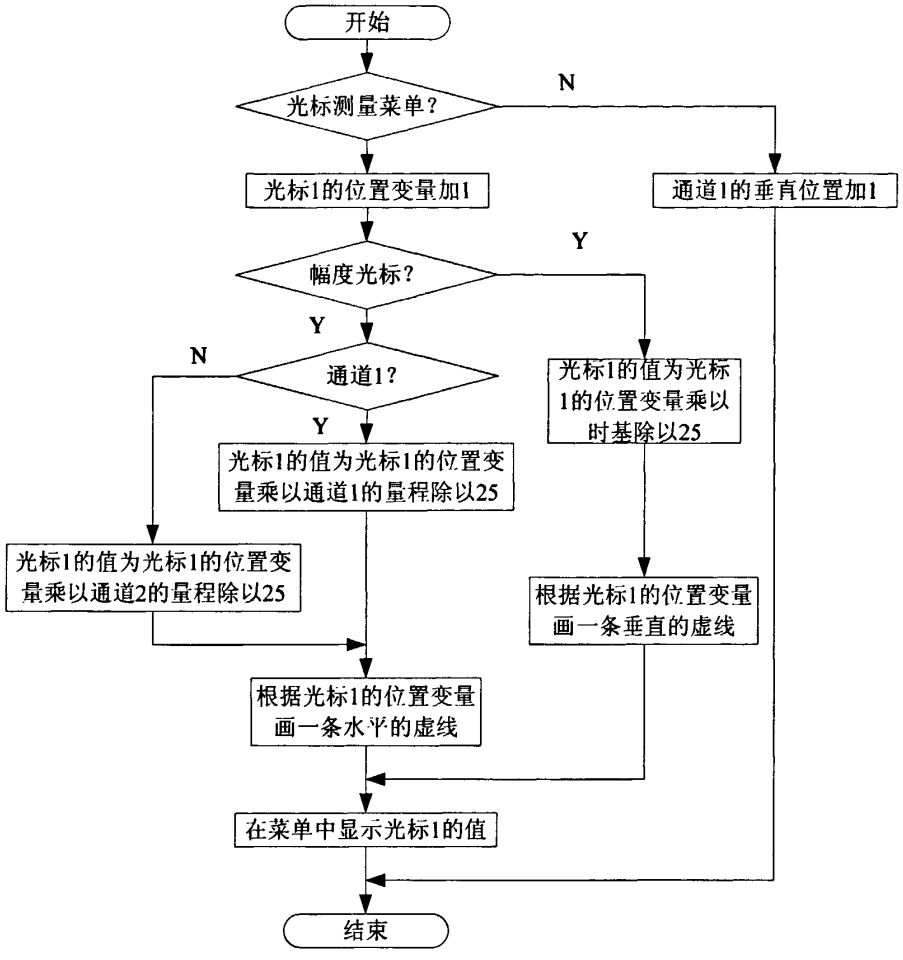


图 4-5 垂直位置 1 旋钮顺时针旋动功能流程图

4.4.3 特殊功能键的实现

本数字存储示波器有以下几个特殊功能键：

1. SET TO 50%键，此键功能是将触发电平设置为信号最小和最大电压电平间大约一半位置处，来快速稳定波形。
2. FORCE TRIG（强制触发）完成当前波形捕获，无论示波器是否检测到触发。在单次采样和正常采样工作模式中很有用。在“自动”模式下，如果未检测到触发，采取周期性地强制触发。
3. RUN/STOP（运行/停止）是开始和停止捕获。RUN 是开始捕获数据，STOP 是停止捕获，停止捕获时，显示最近完成的捕获。在捕获数据时，或当其停止时，可以缩放或平移图形。
4. SINGLE（单次）完成单次序列采集。
5. SET TO 0（设置为 0）用来将水平位置设置为零。
6. AUTO SET 键正确设置量程、时基等

AUTO SET 键的功能较为复杂，下面重点介绍 AUTO SET 键的实现。AUTO SET（自动设置）是数字存储示波器中非常特殊的功能键，是示波器中非常有用的按键，它使得示波器更加容易使用。AUTO SET 设置适当的量程、时基及触发条件以便获得最佳的信号观察效果。

本数字存储示波器的 AUTO SET 键的功能如下：设置适当的量程、时基、触发模式、触发源、触发电平等，当两个通道都有信号时，按照最低频率的信号设置触发和时基，当未发现信号，显示

通道 1。同时，根据波形的形状是类似于正弦波还是脉冲显示不同的菜单。

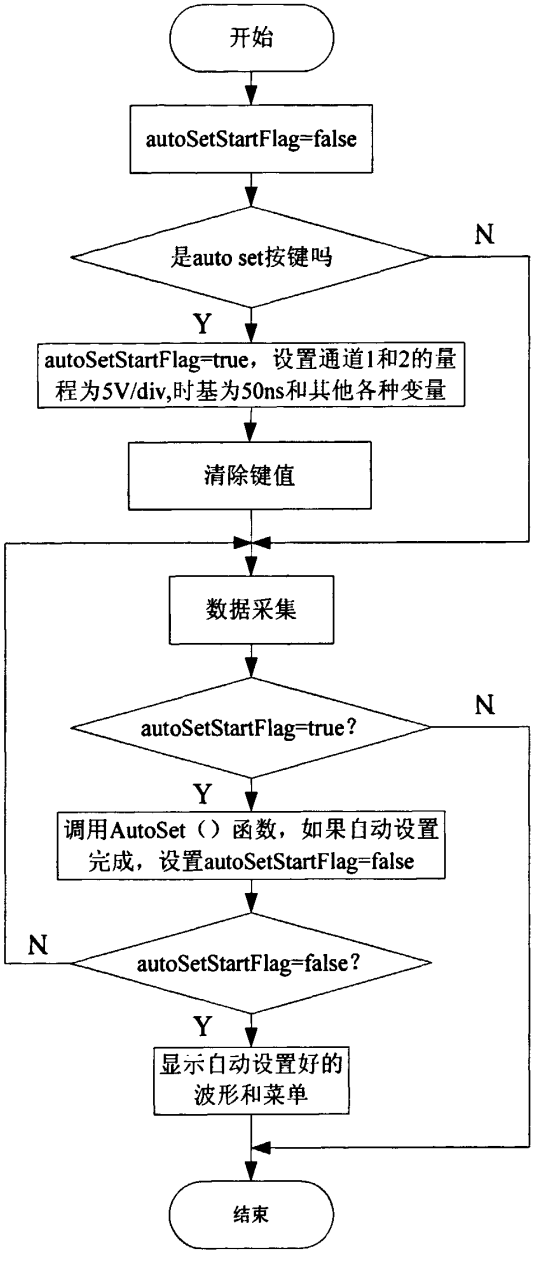


图 4-6 自动设置的流程图

实现自动设置功能的流程图如图 4-6。当按下 AUTO SET 键后，设置 autoSetStartFlag=true（表示自动设置还没有设置好，需要调用 AutoSet 函数）。为防止假波现象，时基从快时基向慢时基变化，量程从大量程向小量程减小，所以设置通道 1 和 2 的量程为 5V/div,时基为 50ns/div（没有插值的最快时基），此外，还要设置其他变量：触发方式为自动触发，耦合方式为直流耦合，并且设置两通道的直流电平在屏幕的中间位置。然后清除 AUTO SET 按键值以防止又重复了上面的设置。等到采集数据后，就判断 autoSetStartFlag 的值，如果为 true，则调用 AutoSet 函数，AutoSet 函数主要的功能是调整量程和时基的档位等，档位的改变会影响下一次采集来的数据。AutoSet 函数会根据有没有寻找到正确的量程、时基档位来设置 autoSetStartFlag，如果设置 autoSetStartFlag 为 false，表示自动设

置已完成，就显示波形和菜单；如果设置 autoSetStartFlag 为 true，表明自动设置还没有完成，所以需要采集新的数据，并调用 AutoSet 函数。当判断 autoSetStartFlag 的值为 false 则不需要再自动设置。

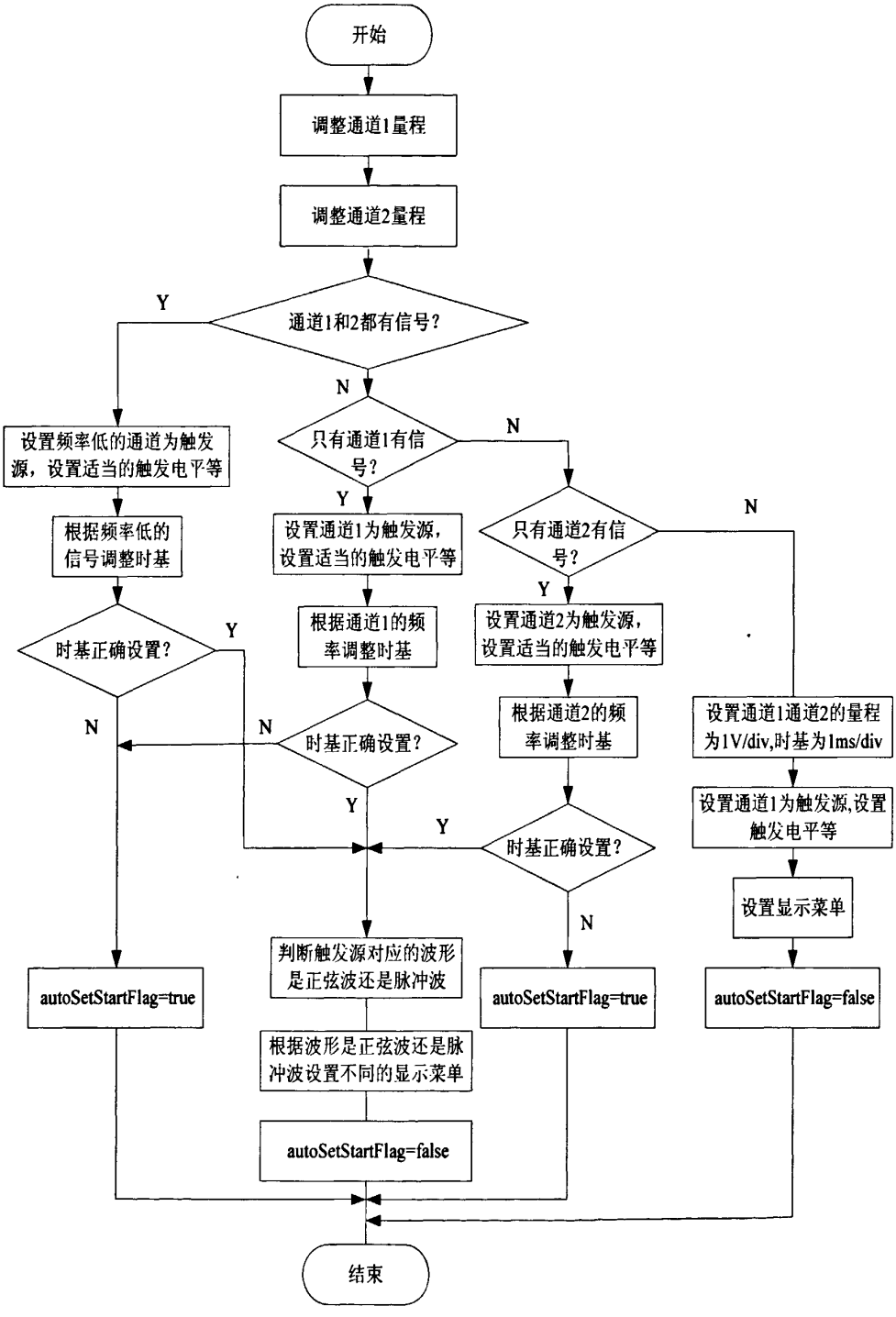


图 4-7 AutoSet 函数流程图

实现自动设置功能的主要函数是 AutoSet（）。AutoSet 函数的流程图如图 4-7。首先是调整通道 1 和 2 的量程，根据量程来判断通道 1 和 2 上有无信号。当通道 1 和 2 上都有信号时则计算两个通

道上的信号的频率，并按照频率低的信号调整时基；当只有通道 1（通道 2）上有信号时，则根据通道 1（通道 2）的频率来调整时基；如果通道 1 和 2 都没有信号则设置一特定的量程、时基、触发等，并设置 autoSetStartFlag 为 false 结束自动设置。当通道上有信号，并且时基调整正确，就判断触发源对应的波形是类似于脉冲波还是正弦波，并根据波形来设置不同的显示菜单，并设置 autoSetStartFlag 为 false 表示自动设置已完成。当通道上有信号，但时基还没有调整正确，则设置 autoSetStartFlag 为 true，表示需要采集新的数据并再次调用 AutoSet 函数直到自动设置完成。

第五章 显示处理模块

5.1 液晶显示原理

显示器是现代电子测量仪器的一种重要的输出设备，它以可视化的手段给出了测量结果，是测量仪器和操作员之间实现人机交互的一种不可或缺的部件。随着通信、IT 与多媒体市场的兴起，耗电少、辐射低、易于携带的液晶显示器（Liquid Crystal Display，LCD）成为显示器的首选。

LCD 本身不发光，依靠对外界光线的不同反射呈现不同的对比度，达到显示的目的，或者通过液晶的背面照光（背光）把显示内容显现出来[23]。LCD 显示按驱动方式可以分为静态驱动和动态驱动，静态驱动使用于字段型液晶显示器件，但每个字段都要引出电极。动态驱动多用于点阵式液晶显示器，点阵式液晶显示器显示的象素众多，为节省硬件驱动电路，通常采用矩阵式结构，在这种结构中把水平方向的一组象素的背电极连在一起，称行电极；把垂直方向的段电极连在一起，称列电极。在显示器件上的每一个象素，都由所在的行列来选择。动态驱动循环的选择点以驱动脉冲，显示某一象素是由行列选择电压合成实现的，例如扫描至某点显示，就需要同时在相应的列和行上施选择电压，以使该点的电场超过显示的阀值，这种扫描法周期很短，使得在显示屏上呈现稳定的波形。

本系统使用的是 TFT 型液晶显示模块（LCD Module）。所谓 TFT LCD 即 Thin film transistor liquid crystal display 的缩写，中文翻译为薄膜晶体管液晶显示器，它是利用薄膜晶体管来产生电压，以控制液晶转向的显示器。所谓液晶显示模块是一种常见的液晶显示器件的商品形式，它是一种将液晶显示器件、连接件、集成电路、PCB 线路板、背光源、结构件装配在一起的组件。液晶显示模块可分为数显液晶模块、点阵字符液晶模块和点阵图形液晶模块三种。数显液晶模块只能显示数字和一些标识符号，点阵字符液晶模块可以显示数字和英文字符，但是不能显示图形。

数字存储示波器的显示特点是不仅需要显示波形图像，还要显示中文或西文的操作菜单和各种测量参数，所以只能选用点阵图形液晶模块。其特点是点阵像素连续排列，行和列之间在排列时没有间隔，因此可以显示连续、完整的图形。由于它也是由 X－Y 矩阵象素构成的，所以除显示图形外，也可以显示字符。本系统选择的是晶采光电科技股份有限公司的 TFT 型 320×240 的 LCD。

5.2 S3C2410 内置 LCD 控制器

一块 LCD 屏显示图像，不但需要 LCD 驱动器，还需要有相应的 LCD 控制器。通常 LCD 驱动器会以 COF/COG 的形式与 LCD 玻璃基板制作在一起，而 LCD 控制器则有外部电路来实现[35]。而 S3C2410 内部已经集成了 LCD 控制器，因此可以很方便地去控制各种类型的 LCD 屏，例如：STN 和 TFT 屏。其中对于 TFT 屏，S3C2410 LCD 控制器的特性有：

1. 支持单色、4 级灰度、256 色的调色板显示模式；
2. 支持 64K 和 16M 色非调色板显示模式；
3. 支持分辨率为 640×480，320×240 及其它多种规格的 LCD。

对于控制 TFT 屏来说，除了要给它送视频数据（VD[23:0]）以外，还有以下一些信号是必不可少的，分别是：

1. VSYNC（VFRAME）：LCD 控制器和 LCD 驱动器之间的帧同步信号。该信号告诉 LCD 屏新一帧开始了。LCD 控制器在一帧显示完成后立即插入一个 VFRAME 信号，开始新一帧的显示。
2. HSYNC（VLINE）：LCD 控制器和 LCD 驱动器之间的行同步脉冲信号。该信号用于 LCD 驱动器将水平线(行)移位寄存器的内容传送给 LCD 屏显示。LCD 控制器在整行数据移入 LCD 驱动器后，插入一个 VLINE 信号。
3. VCLK：LCD 控制器和 LCD 驱动器之间的像素时钟信号。LCD 控制在 VCLK 的上升沿处

送出数据，LCD 驱动器在 VCLK 的下降沿处采样。

4. VDEN（VM）：数据有效标志信号。VM 信号被 LCD 驱动器用于改变行和列的电压 极性，从而控制像素点的显示或熄灭。VM 信号可以与每个帧同步，也可以与可变数量的 VLINE 信号同步。

图 5-1 是 S3C2410 LCD 控制器结构框图。S3C2410 的 LCD 控制器由 REGBANK、LCDCDMA、VIDPRCS、TIMEGEN 和 LPC3600 组成。其中 REGBANK 有 17 个可编程寄存器组和 256×16 的调色板存储器,可用来设定 LCD 控制器；LCDCDMA 是一个专用 DMA，可自动从帧存储器传输视频数据到 LCD 控制器，通过这个特殊的 DMA，视频数据可不经过 CPU 处理就在屏幕上显示；VIDPRCS 可接收从 LCDCDMA 来的视频数据并将其修改到合适数据格式，然后经 VD[23：0]送到 LCD 驱动器；TIMEGEN 和 LPC3600 负责产生 LCD 所需要的控制时序例如 VSYNC、HSYNC、VCLK、VDEN，然后从 VIDEO MUX 送给 LCD 屏。

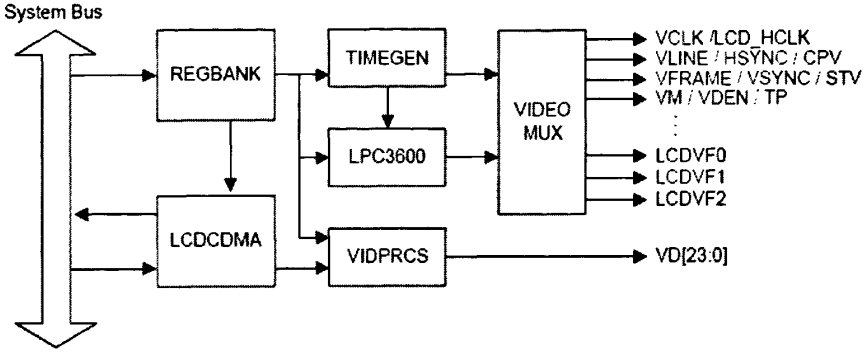


图 5-1 S3C2410 LCD 控制器结构框图

S3C2410 的 LCD 控制寄存器主要有：LCDCON1 寄存器、LCDCON2 寄存器、LCDCON3 寄存器、LCDCON4 寄存器、LCDCON5 寄存器。这些寄存器的设置与显示屏信息、控制时序和数据传输格式等密切相关，在设计中需要根据显示设备的具体信息正确设置这些寄存器才能使 S3C2410 正常控制驱动不同的显示屏。

5.3 Linux 下 LCD 驱动

嵌入式 Linux 是工作在保护模式下，所以用户态进程是无法像 DOS 那样使用显卡 BIOS 里提供的中断调用来实现直接写屏，Linux 抽象出 FrameBuffer 这个设备来供用户态进程实现直接写屏。

Framebuffer 机制[17]模仿显卡的功能，将显卡硬件结构抽象掉，可以通过 Framebuffer 的读写直接对显存进行操作。用户可以将 Framebuffer 看成是显示内存的一个映像，将其映射到进程地址空间之后，就可以直接进行读写操作，而写操作可以立即反应在屏幕上。这种操作是抽象的，统一的。用户不必关心物理显存的位置、换页机制等等具体细节。这些都是由 Framebuffer 设备驱动来完成的。

Framebuffer 本身不具备任何运算数据的能力,就好比是一个暂时存放水的水池.CPU 将运算后的结果放到这个水池,水池再将结果流到显示器。中间不会对数据做处理。

帧缓冲驱动的应用广泛，在 Linux 的桌面系统中，Xwindow 服务器就是利用帧缓冲进行窗口的绘制。嵌入式系统中的 Qt/Embedded 等图形用户界面环境也基于帧缓冲而设计。另外，通过帧缓冲可以支持汉字点阵的显示，因此帧缓冲也成为 Linux 汉化的可行方案。

本系统在网络上原有的 LCD 驱动程序基础上，针对硬件的连接和本液晶屏的特性做了移植。移植工作包括：

1. 设置屏幕的大小，由于此 LCD 是 320×240 大小，所以设置 xres=320，yres=240

2. 设置颜色位数，硬件采用 565 的连接方式，所以设置颜色位数为 16
3. 设置 LCD 控制寄存器，寄存器的设置主要根据 LCD 的参数进行设置，这些参数包括：VBPD，VFPD，VSPW，HBPD，HFPD，HSPW 等，设置如下：

```
lcdcon1 : LCD1_BPP_16T | LCD1_PNR_TFT | LCD1_CLKVAL(7) ,
lcdcon2 : LCD2_VBPD(17) | LCD2_VFPD(20) | LCD2_VSPW(4),
lcdcon3 : LCD3_HBPD(67) | LCD3_HFPD(10),
lcdcon4 : LCD4_HSPW(6) | LCD4_MVAL(13),
lcdcon5: LCD5_FRM565 | LCD5_INVVCLK | LCD5_INVVLINE | LCD5_INVVFRAME |
         LCD5_INVVDEN | LCD5_HWSWP
```

其中，LCD1_BPP_16T 表示设置为 16 位彩色；LCD1_PNR_TFT 表示设为 TFT 模式；LCD1_CLKVAL(7)是设置 CLKVAL 值为 7；LCD2_VBPD 表示垂直同步信号的后肩；LCD2_VFPD 表示垂直同步信号的前肩；LCD2_VSPW 垂直同步信号的脉宽；LCD3_HBPD 表示水平同步信号的后肩；LCD3_HFPD 表示水平同步信号的前肩；LCD4_HSPW 水平同步信号的脉宽，关于 LCD2_VBPD、LCD2_VFPD、LCD2_VSPW、LCD3_HBPD、LCD3_HFPD 和 LCD4_HSPW 的值可查找 LCD 手册；LCD4_MVAL 是 STN 模式下的参数，无需调整；LCD5_FRM565 是设置 16bpp 视频输入数据的格式为 565 方式，即 R、G、B 分别用 5 位、6 位、5 位表示。LCD5_INVVCLK、LCD5_INVVLINE、LCD5_INVVFRAME、LCD5_INVVDEN 是设置控制信号的脉冲极性。LCD5_HWSWP 设置颜色信息为半字交换。

5.4 图形用户界面开发工具

5.4.1 选择 Qt/Embedded 为 GUI 开发工具

对于示波器来说，把采集到的数据以波形的形式在 LCD 上实时的显示是基本的要求。因此，为本系统选择一个高性能、高可靠、占用资源少的嵌入式图形用户界面(Graphical User Interface, GUI)的开发环境显得至关重要。目前应用于嵌入式 Linux 的主流 GUI 系统主要有 MiniGUI、Microwindows、OpenGUI、Qt/Embedded，这些 GUI 在接口定义、体系结构、功能特性上存在很大差别，采取的技术路线也有所不同[36]。

1. MiniGUI：它建立在比较成熟的图形引擎之上，开发的重点在于窗口系统。其小巧精致并且尽量与 Win32 兼容。MiniGUI 可以应用在包括手持设备、机顶盒、游戏终端等在内的各种高端或低端的嵌入式系统中。其技术上的创新包括：图像抽象层 API 基本没有影响，但大大方便了 MiniGUI 应用程序的移植、调试等工作；多字体和多字符集支持；拥有两个不同架构的版本。但是 MiniGUI 也同时存在很多问题，例如：本身存在的代码质量参差不齐，影响整体系统稳定性；开发没有一定的时间限制，很难达到商用的要求。

2. MicroWindows：其主要特色在于提供了 Server/Client 体系结构，与 Win32 和 XWindows 窗口系统保持兼容，其采用分层设计，以便不同的层面能在需要的时候改写，基本上使用 C 语言，其提供了相对完善的图形功能，目前开发的重点在底层的图形引擎。但是其没有任何硬件加速能力，窗口系统和图形接口方面功能也比较欠缺。

3. OpenGUI：基于一个用汇编实现的 x86 图形内核，并且利用 MMX 指令进行了优化，因此其运行速度非常快。OpenGUI 采用分层结构：最底层是使用汇编语言编写的高速图形引擎，中间层实现了用于图形绘制的 API，包括线条、矩形和圆弧等，第三层是使用 C++编写的完整的 GUI 对象集。OpenGUI 的特点是资源消耗小、可移植性差、不支持多进程。

4. Qt/Embedded：是著名的Qt 库开发商 Trolltech 公司开发的面向嵌入式系统的Qt版本，它提供了丰富的窗口小部件，并且还支持窗口部件的定制，因此可以为用户提供漂亮的图形界面，为带有轻量级窗口系统的嵌入式设备提供了标准的API。Linux桌面系统的KDE就是基于Qt库开发的。Qt可以帮助开发者为满足嵌入式系统小而快捷的要求开发稳定的应用程序。

作为嵌入式平台的 QT 版本，Qt/Embedded 具有一些其它的特点[37]：

1. 节省内存，大小可定制。Qt/Embedded 的设计特点保证了它可以高效的利用内存资源。同时程序员可以根据开发环境的需要和内存资源的大小，对 Qt/Embedded 库进行裁剪。

2. 支持多种 CPU 体系结构。Qt/Embedded 可以方便地移植到 Linux 系统支持的多种 CPU 体系结构平台上。包括当前主流的嵌入式应用，例如 X86、ARM、MIPS 以及 PowerPC 等多种 CPU。

3. 优异的屏幕与色彩支持，功能完善的仿真环境。Qt/Embedded 可以支持多种屏幕大小、分辨率和色深。实际上，Qt/Embedded 是硬件独立的，可以在 Linux 支持的处理器及图形卡的任意组合上正常工作。在使用 Qt/Embedded 开发嵌入式应用时，并不需要将所有的应用程序都下载到目标机才能进行仿真，而可以在宿主机上进行基本的调试。

在综合分析和比较以上各 GUI 的特点之后，本系统采用 Qt/Embedded 作为 GUI 开发工具。

5.4.2 Qt/Embedded 的特征

Qt/Embedded 的底层图形引擎基于 framebuffer，它直接对帧缓冲 FrameBuffer 进行读和写的操作，相对于 QT 在 X11 环境下运行的体系结构提高了程序运行的效率，图 5-2 为 Qt/Embedded 的实现结构[38]。在 Qt/Embedded 中 QScreen 类为抽象出的底层显示设备，基类其中声明了对于显示设备的基本描述和操作方式如打开、关闭、获得显示能力、创建 GFX 操作对象等。另外一个重要的基类是 QGfx 类，该类抽象出对于显示设备的具体操作接口(图形设备环境) 如选择画刷、画线、画矩形操作等。以上两个基类是 Qt/Embedded 图形引擎的底层抽象。其中所有具体函数基本都是虚函数，Qt/Embedded 对于具体的显示设备如 Linux framebuffer、Qt Virtual Framebuffer 做的抽象接口类全都由此继承并重载基类中的虚函数实现。

Qt/Embedded 在体系上为 C/S 结构，任何一个 Qt/Embedded 程序都可以作为系统中唯一的一个 GUI Server 存在。

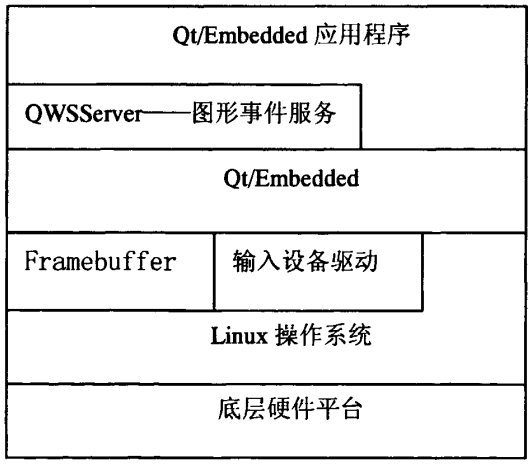


图 5-2 Qt/Embedded 的实现结构

Qt/Embedded 包含了许多支持嵌入式系统开发的工具，有两个最实用的工具是 qmake 和 Qt designer(图形设计器)。qmake 是一个为编译 Qt/Embedded 库和应用而提供的 Makefile 生成器。它能够根据一个工程文件（.pro）产生不同平台下的 Makefile 文件。qmake 支持跨平台开发和影子生成（shadow builds），影子生成是指当工程的源代码共享给网络上的多台机器时，每台机器编译链接这个工程的代码将在不同的子路径下完成，这样就不会覆盖别人的编译链接生成的文件。qmake 还易于在不同的配置之间切换。开发者可以使用 Qt 图形设计器可视化地设计对话框而不需编写一行代码。使用 Qt 图形设计器的布局管理可以生成具有平滑改变尺寸的对话框，qmake 和 Qt 图形设计器是完全集成在一起的。

图形用户界面需要对用户的动作做出响应。例如，当用户点击了一个菜单项或是工具栏的按钮

时，应用程序会执行某些代码。大部分情况下，是希望不同类型的对象之间能够进行通信。程序员必须把事件和相关代码联系起来，这样才能对事件做出响应。以前的工具开发包使用的事件响应机制是易崩溃的，不够健壮的，同时也不是面向对象的。Trolltech 已经创立了一种新的机制，叫做“信号与插槽”。信号与插槽是一种强有力的对象间通信机制，它完全可以取代原始的回调和消息映射机制；信号与插槽是迅速的、类型安全的、健壮的、完全面向对象并用 C++来实现的一种机制[39]。

在以前，当使用回调函数机制来把某段响应代码和一个按钮的动作相关联时，通常把那段响应代码写成一个函数，然后把这个函数的地址指针传给按钮，当那个按钮被按下时，这个函数就会被执行。对于这种方式，以前的开发包不能够确保回调函数被执行时所传递进来的函数参数就是正确的类型，因此容易造成进程崩溃。另外一个问题是，回调这种方式紧紧的绑定了图形用户接口的功能元素，因而很难把开发进行独立的分类。

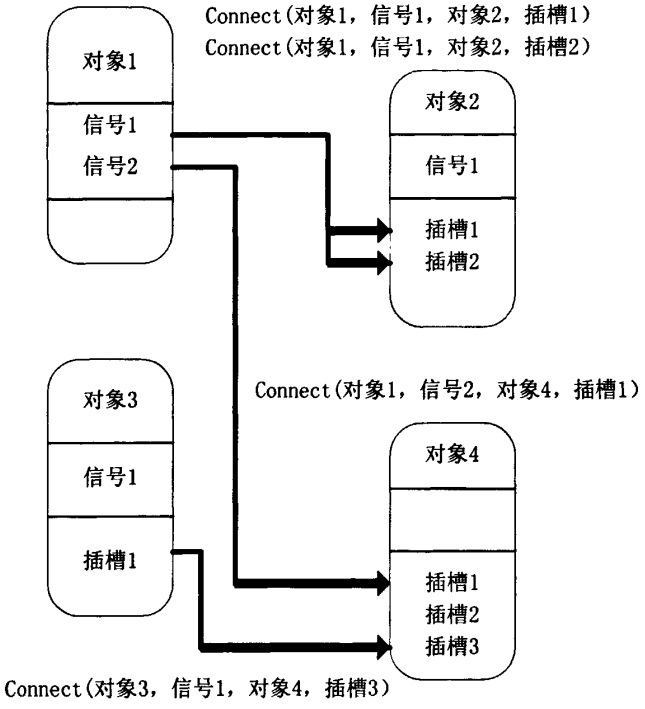


图 5-3 信号和插槽连接示意图

Qt 的信号与插槽机制是不同的。Qt 的窗口在事件发生后会激发信号。例如一个按钮被点击时会激发一个“clicked”信号。程序员通过建立一个函数（称作一个插槽），然后调用 connect 函数把这个插槽和一个信号连接起来，这样就完成了一个事件和响应代码的连接。信号与插槽机制并不要求类之间互相知道细节，这样就可以相对容易的开发出代码可高重用的类。信号与插槽机制是类型安全的，它以警告的方式报告类型错误，而不会使系统产生崩溃。图 5-3 是信号和插槽连接示意图。

5.5 界面设计和实现

本系统采用的 LCD 是 320×240，示波器的液晶显示区可分为 5 块，如图 5-4 所示。

1 区为屏幕状态区，占用 260×10 大小，主要显示采集模式，触发状态和中心刻度线的时间读数等。2 区为零点显示区，占用 10×200 大小，主要显示波形的接地参考点。3 区为波形和触发电平显示区，占用 250×200 大小，主要显示波形、触发电平和水平触发位置。4 区为菜单和参数显示区，占用 60×210 大小，这部分主要显示菜单和菜单选项，当在测量波形的有关参数时，它还显示参数名称和参数测量值。5 区为量程（垂直灵敏度）、时基和信息显示区，占用 320×30 大小，主要显示

量程、时基、触发源、触发电平等。

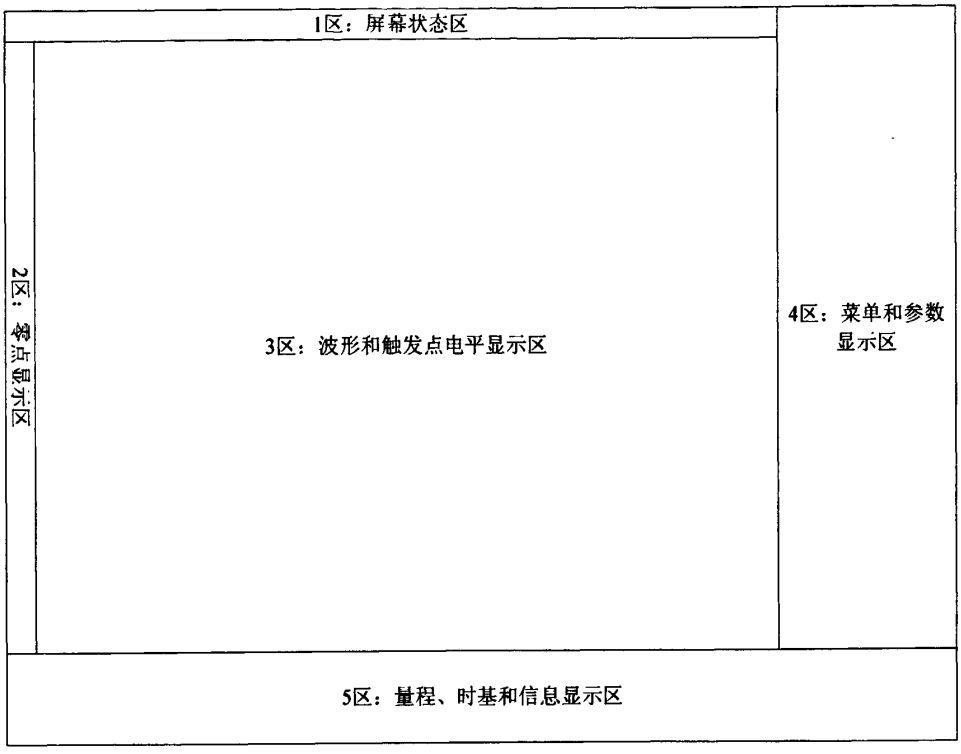


图 5-4 屏幕区域划分示意图

需要注意的是 LCD 的左上角的坐标（0，0），右下角坐标为（320，240）。为了和通常的大家的习惯一致，即左下角的的坐标（0，0），右上角坐标为（320，240），在编程的时候需要做个简单的线性变换即可，坐标变换示意图如图 5-5。

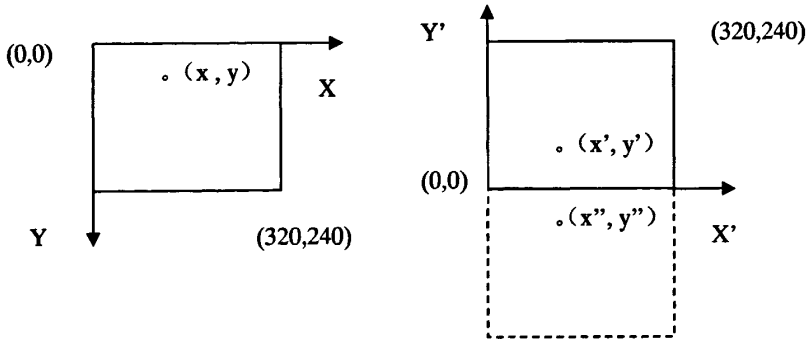


图 5-5 坐标变换示意图

（x", y"）是由（x , y）向下平移 240 所得，所以 y"=y+240; x"=x;

（x', y'） 和（x", y"）关于原坐标系 x=240 对称,所以 y"+y'=240*2; x"=x';

所以得 x' =x; y'=240- y。在将波形显示到屏幕上时需要使用这一变换。

Qt/Embedded 拥有丰富的满足不同需求的窗口部件，如按钮、滚动条等，而且还允许创建自定义窗口部件。这里我们使用有框架的窗口部件的基类 QFrame 绘制 LCD 上的各个显示区。类 QFrame 有很多成员函数，如 setBackgroundColor（）是设置背景颜色，setGeometry（）设置窗口大小。下

面是示波器界面的各个窗口的实现。

```
screenStateDispFrm = new QFrame( this, "screenStateDispFrm" );//创建屏态状态区
screenStateDispFrm->setBackgroundColor( QColor( 90, 90, 90 ) ); //设置背景颜色
screenStateDispFrm->setGeometry( QRect( 0, 0, 260, 10 ) ); //设置屏态状态区大小

zeroDispFrm = new QFrame( this, "zeroDispFrm" ); //创建屏零点显示区
zeroDispFrm->setBackgroundColor(QColor( 90, 90, 90 ));
zeroDispFrm->setGeometry( QRect( 0, 10, 10, 200 ) );

waveformDispFrm = new QFrame( this, "waveformDispFrm" ); //创建波形显示区
waveformDispFrm->setBackgroundColor(black);
waveformDispFrm->setGeometry( QRect( 10, 10, 250, 200 ) );

menuDispFrm = new QFrame( this, "menuDispFrm" ); //创建菜单和参数显示区
menuDispFrm->setBackgroundColor(QColor( 90, 90, 90 ));
menuDispFrm->setGeometry( QRect( 260, 0, 60, 240 ) );

//创建量程、时基显示区
rangeTimeBaseDispFrm = new QFrame( this, "rangeTimeBaseDispFrm" );
rangeTimeBaseDispFrm->setBackgroundColor(QColor( 90, 90, 90));
rangeTimeBaseDispFrm->setGeometry( QRect( 0, 210, 260, 30 ) );
```

5.6 菜单设计和实现

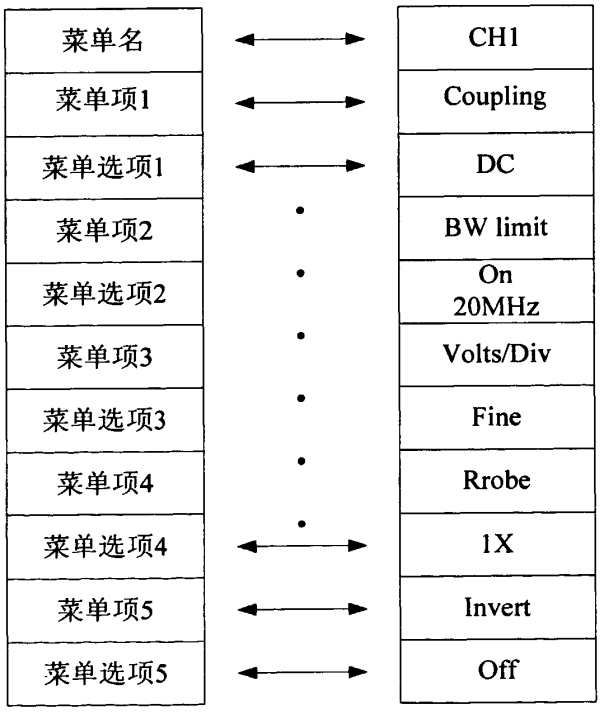


图 5-6 菜单结构示意图和实例

该数字存储示波器有十三个一级菜单，在每个菜单下还有自己的二级菜单，不同的二级菜单有不同的菜单选项。为了编程的统一，菜单结构设计如图 5-6 所示。

其中菜单名是一级菜单，菜单项 1，菜单项 2, ……, 菜单项 5 是二级菜单。菜单选项 1 包含了菜单项 1 下的选项，可以通过按键来进行选择其中的一项。例如按下 CH1 按钮，菜单名显示 CH1，CH1 菜单下有 5 个二级菜单，分别是：耦合、带宽限制、伏/格、探头、反相。在二级菜单项耦合的下面有三个菜单选项：直流、交流和接地，当按下耦合对应的按键时，会在直流、交流和接地选项间切换。

本示波器要求：进入一个菜单 A 后，经过一些操作后进入另一个菜单 B，又经过一系列操作，当再次返回到菜单 A 后，菜单 A 的选项应该保持为操作者先前选择的选项。所以在编程时为每一个一级菜单创建 11 个字符串（1 个菜单名，5 个二级菜单项和 5 个二级菜单选项），用于保存二级菜单和二级菜单选项，另外还有一组特殊的字符串（11 个字符串），这组字符串是专门用于显示用的，当一个一级菜单对应的键被按下时，就将该一级菜单的字符串的值赋给显示的字符串，所以每次都是显示那组要显示的字符串，当用户按下二级菜单选项键时，会改变二级菜单选项的字符串的值，由于二级菜单选项的字符串会把值赋给要显示的字符串，所以用户改变菜单选项时，菜单会根据用户的操作正确的显示。

编程时没有使用 C 语言中的类型 char*，也没有使用 C++语言中的 string 类，而是使用了 QT 中的 QString 类型来表示字符串，这是因为在 QT 中，如果要国际化（支持多种语言），要求可见的文本使用 QString，而本仪器要求支持多种语言。另外一个原因是 QString 功能强大，本仪器的参数测量结果有的会是浮点数，用 QString 可以很容易的显示浮点数。

5.7 波形显示的实现

屏幕的波形与触发点显示区通常情况下需要显示网格、两个通道波形、触发点、触发电平以及光标线（在光标测量时）等。如果是在 Math 菜单下则显示 Math 下操作的结果（例如两波形相加的结果），但网格是必须要画的，波形显示的流程图如图 5-7。

首先是画网格，然后判断是否要画 Math 菜单下的波形，如果是，则根据 Math 菜单下的二级菜单是 ch1+ch2、ch1-ch2 、ch2-ch1、ch1×ch2、FFT 来画相应的波形；如果不需要画 Math 波形则判断是否显示 ch1 波形，如果需要显示 ch1，则根据是点方式还是矢量方式画 ch1 波形，然后画 ch1 的零点，如果不需要显示 ch1，则判断是否显示 ch2 波形，下面的和画 ch1 波形类似，画完 ch2 波形后判断是否显示光标，如果要显示光标，则根据是幅度光标还是时间光标进行不同的画图，若是要画幅度光标，则根据光标的位置画两行虚线，若是要画时间光标，则根据光标的位置画两列虚线。

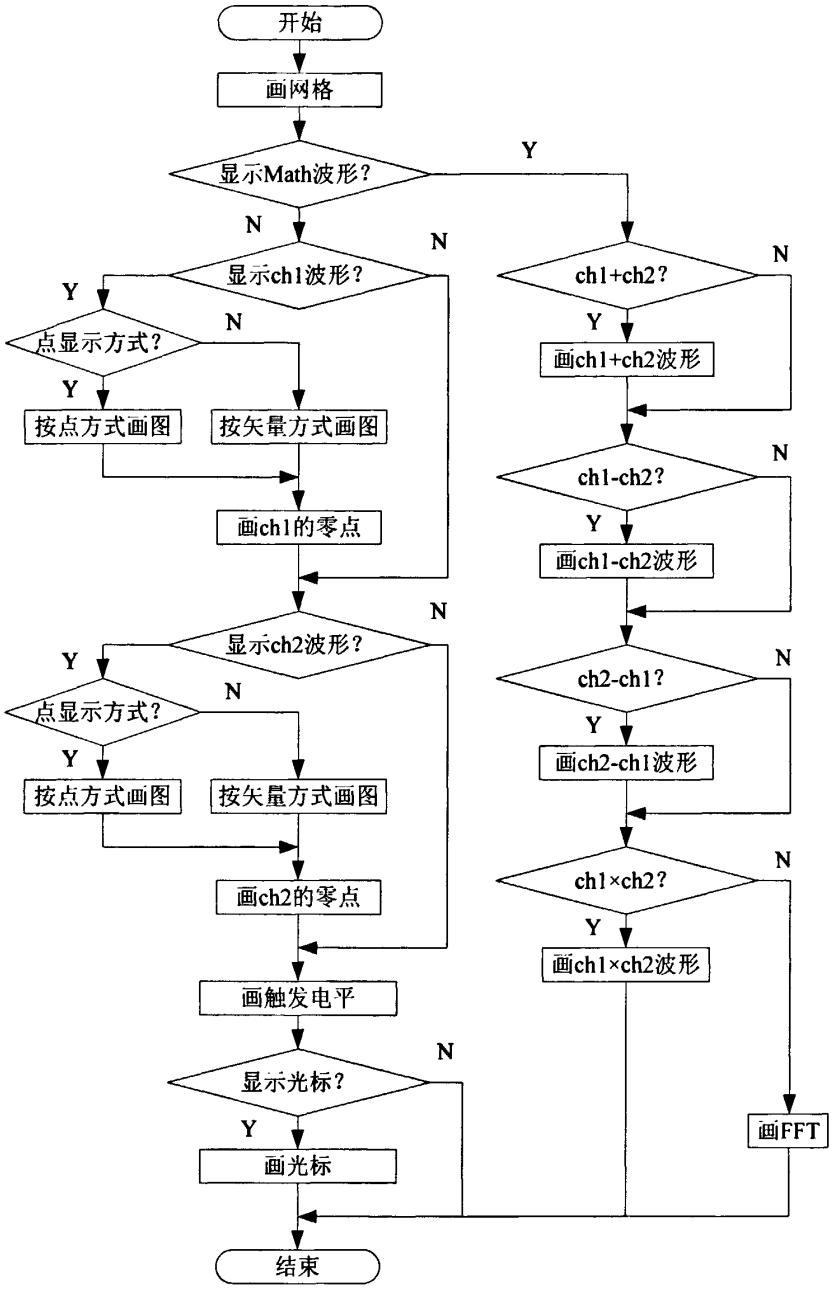


图 5-7 波形显示流程图

5.8 解决屏幕闪烁的办法

在非常短的时间内，同一个像素被使用不同颜色多次绘制，会发生闪烁。双缓冲[40]技术是 GUI 编程中常用的技术，它是一种用来提供漂亮的用户界面并且避免闪烁的技术。所谓的双缓冲就是在内存中创建一个和屏幕绘图区域一致的对象，先将图像绘制到内存中的这个对象上，再一次性将这个对象上的图像拷贝到屏幕上，这样能大大加快绘图的速度。双缓冲不仅仅避免闪烁有用，如果窗口部件非常复杂并且需要重复绘制时，它也是非常有用的。

在本示波器编程的早期，屏幕会闪烁。使用双缓冲技术后，解决了闪烁的问题。下面就讨论在

QT 下如何使用双缓冲技术来避免闪烁。

首先，把一个 QPixmap 的大小定义为至少和重绘区域的边缘矩形一样大；然后使用 QPixmap::fill（）函数用整个窗口部件的擦除色或者背景色来填充整个 QPixmap；接着以像素映射为参数创建一个画笔 QPainter；最后，使用 bitBlt（）函数（它的名字叫“位块搬移”）把像素映射复制到窗口部件中。

```
QRect ur=e->rect();//得到组件尺寸
QPixmap pix(ur.size());//以此为参数创建一个像素映射变量
pixmap.fill(menuDispFrm, rect.topLeft( ) );//填充像素映射，以菜单显示区为例
QPainter paint(&pix);//以像素映射为参数创建一个 QPainter 对象
paint.translate(-ur.x(),-ur.y());//在 QPainter 上绘画
bitBlt(this, 0, 0, &pix);// 位块搬移
```

5.9 进一步提高刷新率的方法

双缓冲其实就是一种提高屏幕刷新率的方法，通过提高屏幕刷新率，屏幕不再闪烁。为了进一步提高刷新率，分析了本仪器的特点，采取了下面的方法进一步刷新率，流程图如图 5-8。

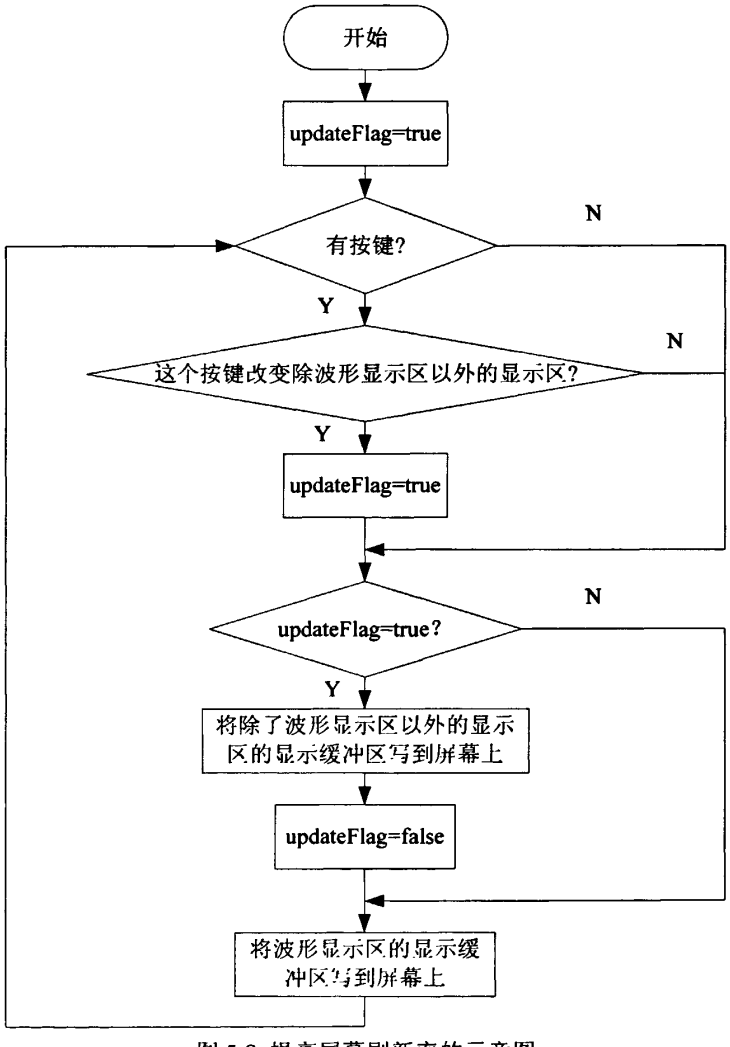


图 5-8 提高屏幕刷新率的示意图

当没有按键按下时，本仪器只需刷新波形与触发点显示区，当有按键按下时菜单区、量程时基等显示区才会相应的变化，这时才需要刷新这些显示区。 实现的方法如下：设置一全局变量updateFlag，并初始化为true，这样保证第一次也绘制除波形显示区以外的显示区，即绘制所有的显示区。判断是否有按键，如果有按键按下则进一步判断该按键是否会改变除了波形显示区以外的显示区，如果会，则置updateFlag=true，没有按键或者该按键只改变波形显示区则不用置updateFlag=true，接下来判断变量updateFlag是否为true，如果为true，则将除了波形显示区以外的显示区的显示缓冲区写到屏幕上，即刷新除了波形显示区以外的显示区，并置updateFlag=false，这一点很重要，重置updateFlag=false会保证按下一个键只会刷新一次，如果updateFlag为false，则不刷新波形显示区以外的显示区，最后将波形显示区的显示缓冲区写到屏幕上，即刷新波形显示区。

5.10 国际化——支持多种语言

由于本数字存储示波器的菜单要求中英文操作界面，并且可以自由的切换，因此QT的国际化功能显得十分重要。Qt完全支持Unicode——国际标准字符集。程序员可以在他们的程序里自由的混用阿拉伯语、英语、希伯来语、日语、俄语和其他Unicode所支持的语言。QT还包括支持程序翻译的工具，以帮助各个公司进军国际市场[41]。

QT国际化的步骤如下：

首先，对所有用户可见的文本使用QString。用户可见的文本也就是以任何方式显示给用户的文本[42]。与char相比，QString内部使用Unicode编码。这使QString能够毫无问题地处理世界上的所有语言。相反，如果使用char*，这很可能造成某些系统无法以其希望的方式显示文本。通过使用QString显示用户可见文本，能够确保不会出现这种问题。当然对于仅仅程序员可见的文本并不需要都变成QString对象，可利用Qt提供的QCString或者原始的char *。

其次，对所有文字形式的文本使用tr()函数。在Qt的翻译机制下，QObject::tr()函数可以帮我们取得翻译之后的文本。

再次，利用lupdate工具从源代码中扫描并提取需要翻译的字符串，生成.ts文件。运行lupdate，以提取出Qt应用的C++源代码中的可翻译文本，会产生一个给翻译者的信息文件（.ts文件）。

然后使用Qt语言学家linguist工具来协助完成翻译工作，即打开前面用lupdate生成的.ts文件，对其中的字符串逐条进行翻译并保存。

最后，利用lrelease工具处理翻译好的.ts文件，生成格式更为紧凑的.qm文件。运行lrelease，以从.ts文件中得到只适用于最后使用的轻量级的信息文件（.qm文件）。

如果要增加支持其他语言，只要在第四步骤中增加其他语言的翻译译文即可。本数字存储示波器的默认语言是英文，可以通过按下二级菜单的相应按键来实现中英文菜单的切换。

第六章　系统调试与结果

6.1 系统调试

本系统的调试包括：硬件调试、驱动程序调试、应用程序调试。首先进行硬件调试，保证硬件连接正确的情况下，然后再调试驱动程序。根据硬件的连接编写驱动程序和驱动测试程序，由于本系统在调试驱动时采用模块加载的方式，所以每一个设备驱动可以单独的使用相应的驱动测试程序调试，如果驱动程序能正确的控制硬件，则将其加入到应用程序中。应用程序使用 Linux 提供的系统调用函数操作各个硬件设备，并实现数字存储示波器的各种功能。

由于本系统是嵌入式系统，所以驱动程序、应用程序的开发与调试和在通用 PC 机上开发程序不同。在嵌入式应用系统中，通常将运行目标程序的计算机系统称为目标机。由于目标机没有足够的资源运行开发工具和调试工具，就需要在另外一台计算机上运行和调试程序。这个运行调试程序的计算机通常是一台 PC 机，称为宿主机（或者调试机，主机）。通常的嵌入式系统的软件开发采用一种交叉编译调试的方式。 交叉编译调试环境建立在宿主机上，开发时使用宿主机上的交叉编译、汇编及连接工具形成可执行的二进制代码，这种可执行代码并不能在宿主机上执行，而只能在目标机上执行，然后把可执行文件下载到目标机上运行。调试时的方法很多，可以使用串口，以太网口等。图 6-1 是驱动程序和应用程序的开发流程。

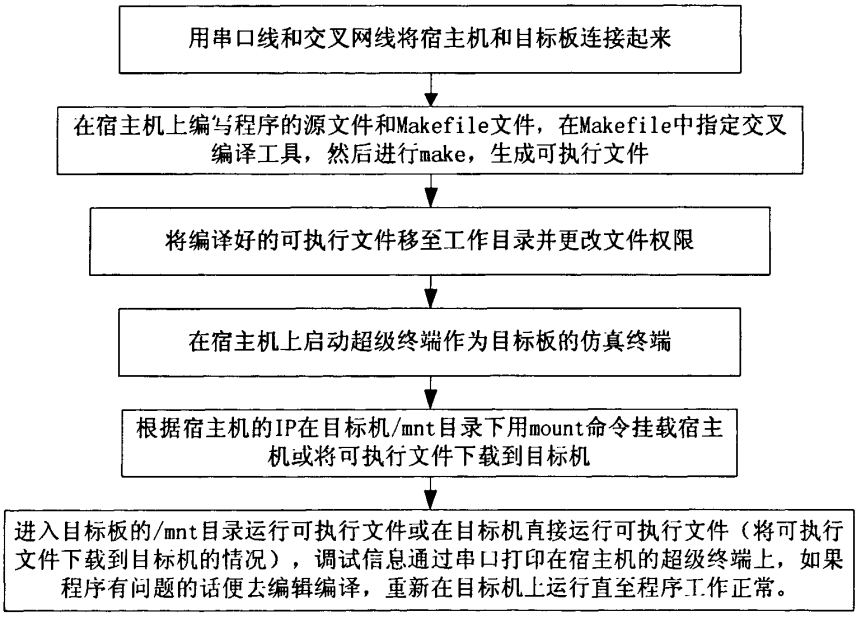


图 6-1 驱动程序和应用程序的开发流程

6.2 系统调试结果

到目前为止，本系统设计了一个实时采样的带宽为 200MHz 数字存储示波器的样机，已完成的工作包括：硬件电路的制作和调试；系统控制模块的硬件驱动设计；数据处理、键盘、显示模块的绝大部分软件设计。样机基本完成了本数字存储示波器要求的各种功能，可以实现的功能如下：

1. 时基，幅度档位旋钮的变化，快时基时信号的恢复（插值），波形上下左右移动显示
2. 自动测量和光标测量功能：参数包括最大值、最小值、峰峰值、平均值、有效值、频率、 周期、上升时间、下降时间、正频宽、负频宽等。用光标测量幅度类和时间类参数。
3. 波形和设置的存储调出功能：可以将屏幕上显示的波形或设置参数存储并且可以重新调出。

4. 自动设置功能：对任何一种未知的信号，此功能能以适当的时基和量程显示。
5. 触发功能：有内部/外部，上升和下降触发，触发方式有自动，正常，和单次。
6. 获取方式：有采样，峰值检测和平均三种捕捉方式。
7. 数学运算功能：通道 1 和通道 2 加，减，乘和 FFT 功能。
8. 两种语言界面：操作的界面语言可选择中文，英文。

以下是本数字存储示波器的实物和工作时照片。

图 6-2 是数字存储示波器各部分实物图，包括 LCD，s3c2410，AD，FPGA，键盘和旋钮等。图 6-3 数字存储示波器的样机照片。

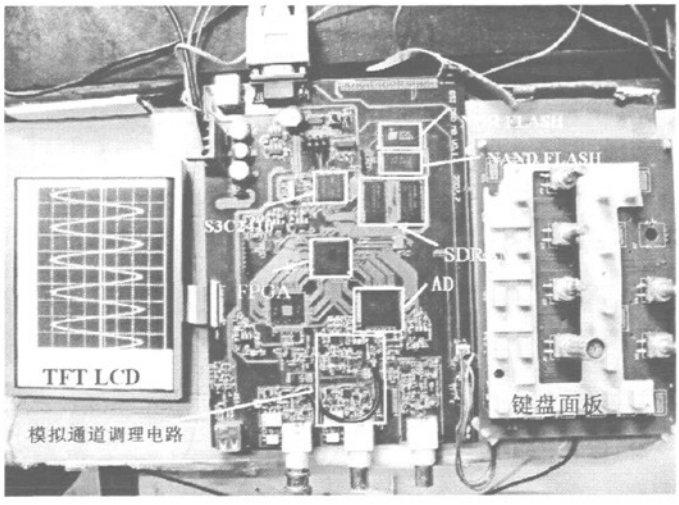


图 6-2 数字存储示波器各部分实物图

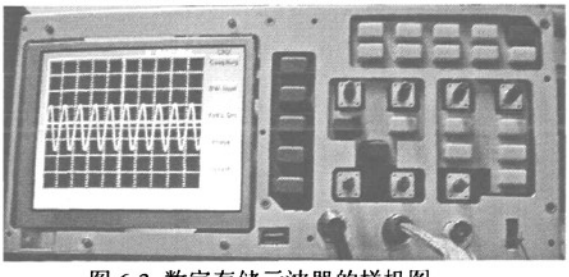


图 6-3 数字存储示波器的样机图

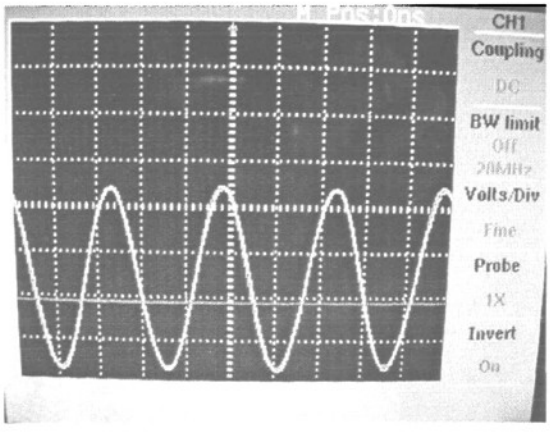
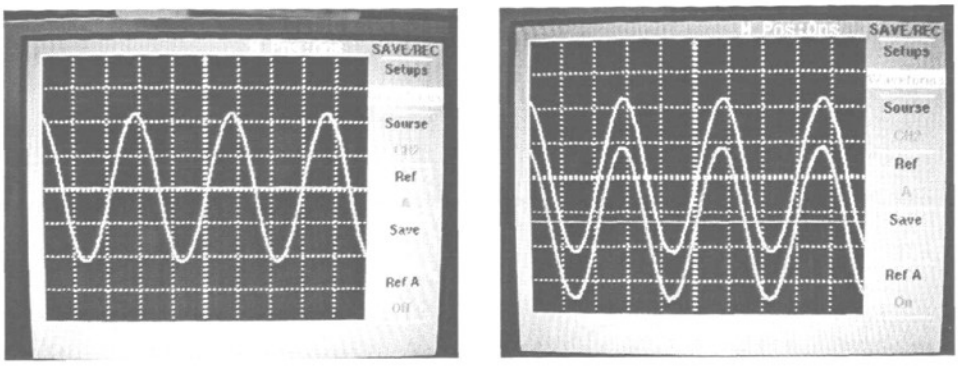


图 6-4 输入幅度是 500mV，频率是 200MHz 正弦波时的波形显示图

图 6-4 输入幅度是 500mV，频率是 200MHz 正弦波时的波形显示图，量程是 100mv，时基是 2ns，在这一时基采用了正弦插值以恢复波形。其中的一条水平红线是表示触发电平的位置。由于背景颜色和时基的颜色对比不是很明显和相机的原因，有的照片可能看不清楚时基。

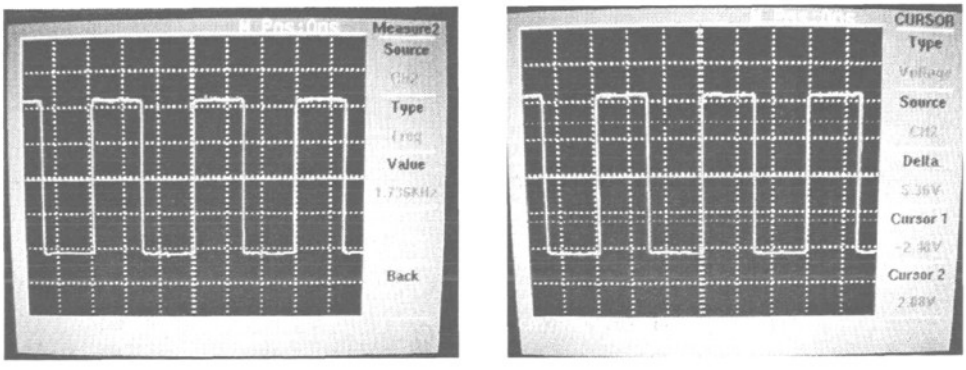
图 6-5 是存储/调出功能的照片，其中（a）为原波形，图（b）中白色波形是存储的原波形重新调出后的波形。



(a) 原波形　　(b)存储后调出的波形

图 6-5 存储/调出功能

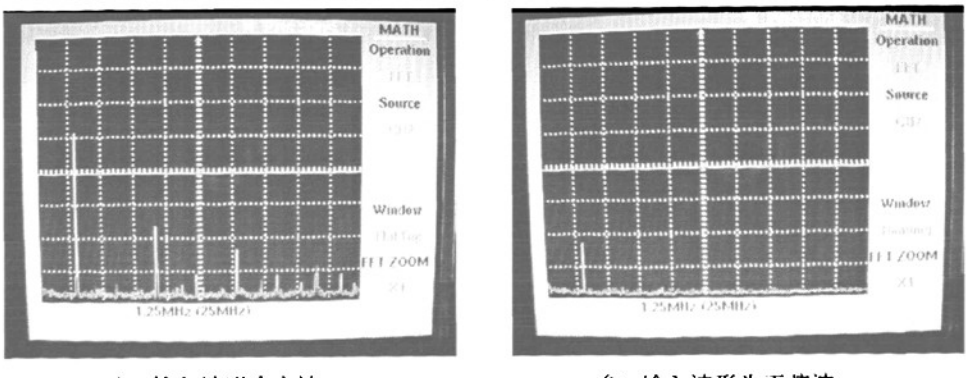
图 6-6 是参数测量功能的照片。 (a) 是自动测量通道 2 的频率，(b) 是光标测量通道 2 的幅度。



(a) 自动参数测量　　(b) 光标测量

图 6-6 参数测量功能

图 6-7 是通道 2 输入方波和正弦波并且加了不同窗的 FFT 的照片。



(a) 输入波形为方波　　(b) 输入波形为正弦波

图 6-7　FFT 功能

图 6-8 是自动设置功能的照片。

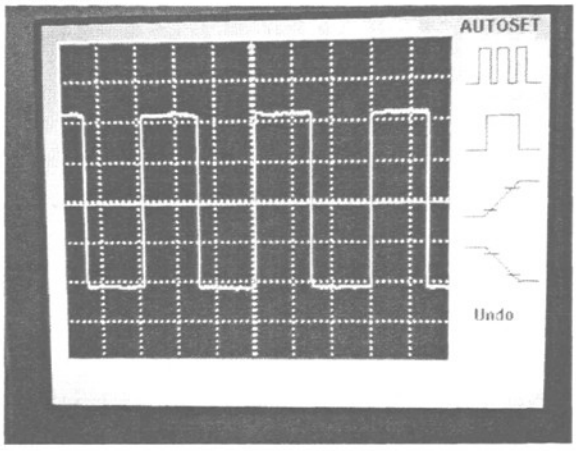


图 6-8 自动设置功能

图 6-9 是触发功能的照片。通道 1 和 2 都有信号，选择通道 2 为触发源，触发类型选择上升边沿触发，触发模式为自动。

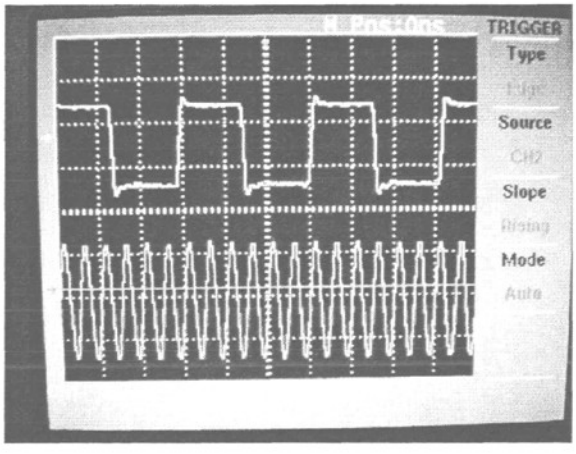


图 6-9 触发功能

图 6-10 中文界面的照片。

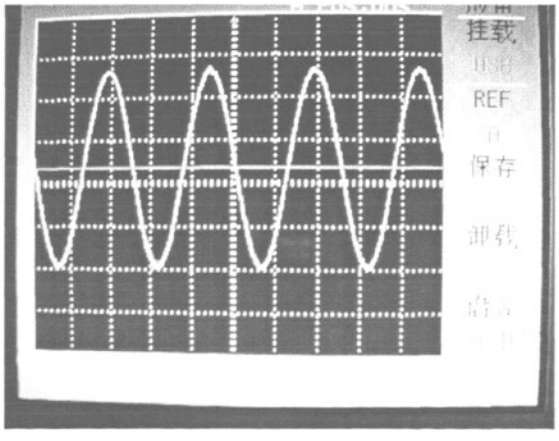


图 6-8 中文界面

第七章 总结与展望

示波器是最常用的电子测量仪器之一，在生产、科研、国防等许多领域有着广泛的应用。数字存储示波器更是以其特有的优势，正在逐渐取代模拟示波器而被广泛使用于各个测试领域。国外在数字存储示波器领域的技术已经非常成熟，并且占领了绝大部分的国内市场份额。而国内的数字存储示波器的研制尚处于起步阶段，因此自主研制数字存储示波器成为必要。

本课题是研制带宽为 200MHz 的数字存储示波器。本文根据数字存储示波器的原理，结合本课题数字存储示波器的设计指标要求，提出硬件采取高速 A/D＋FPGA＋ARM9，软件采用嵌入式 Linux 操作系统的技术方案。本系统采用 Samsung 公司的 ARM9 微处理器 S3C2410 作为控制核心，其强大的控制性能和灵活的接口技术可以大大简化系统的控制和编程。采用 National Semiconductor 公司的最高采样率为每秒 1G 采样点的高速模数转化器 ADC08D500 和 Xilinx 公司的性价比较高的 FPGA 芯片 XC3S500E，实现数据的高速采集和存储。利用嵌入式 Linux 操作系统支持各种设备的特点，充分利用了系统的片上资源，方便实现整个系统的功能扩展和软硬件升级和移植。

本论文主要完成了以下工作：

1. 提出硬件采取高速 A/D＋FPGA＋ARM9，软件采用嵌入式 Linux 操作系统的设计方案。分析了信号调理模块、数据采集和存储模块、系统控制模块等几个模块的硬件构成和功能以及实现方法。
2. 通过分析比较各种嵌入式操作系统，采用嵌入式 Linux 作为本系统的软件开发平台。阐述了各个驱动程序的功能并设计了应用程序的整体框架。
3. 研究并实现了插值算法、时间类参数测量、幅度类参数测量、FFT 等。
4. 研究了键盘和旋钮的原理、键扫描与接受和本数字示波器的键功能的编程。
5. 研究了 LCD 的原理、Linux 下 LCD 驱动和基于 Qt/Embedded 的数字存储示波器的界面的设计和实现。

虽然现有的系统已经基本完成了 200MHz 数字存储示波器的功能，但是本系统还需要进一步的改善：

1. 本系统移植的嵌入式 Linux 内核是基于 2.4.18 版本的，现在 Linux 内核开发进展迅速，2.6 版本的内核已经相当成熟，内核 API 集成更多的功能并且更好的支持处理器平台和硬件驱动，因此以后可以考虑把内核版本升级到 2.6。
2. 虽然刷新率以满足不闪烁的需求，但可以选择更强的处理器或优化程序进一步提高刷新率。
3. 丰富屏幕上的信息，如将采集模式以图标的方式显示在屏幕状态区。
4. 加入支持其它语言（如法语、西班牙语、日语、韩语等）的功能。

致 谢

光阴似箭，转眼间两年半的研究生学习生活即将结束。东南大学信息处理与应用工程研究中心良好的工作环境和积极向上的学术气氛是本论文顺利完成的重要前提和保证，在东南大学的硕士学习生涯是我人生中非常重要的一段时期，从基础学习走向研究学习，从实践能力的锻炼走向理论分析能力的培养。在论文即将完成之际，向工程研究中心的各位领导，老师和同学致以衷心的谢意。

我首先要衷心感谢我的导师赵力教授。他严谨的治学态度、一丝不苟的工作作风和渊博的知识给我留下了深刻的印象。在攻读硕士研究生期间，他对我在学业和科研上的悉心指导和关心，在生活上的关怀与帮助使我一生难忘，并将激励我在今后的学习和工作中勇于拼搏。在他的细心指导下，我得以顺利完成学业和研究工作，也提高了自己独立研究学习的能力。他渊博的学识，认真严谨的学术态度和脚踏实地的工作作风永远都值得我学习。

非常感谢知识渊博的束海泉教授。作为项目组的负责人，束教授耐心细致的工作作风、严谨求实的科学态度给我留下了深刻印象。对他给予我科研方法上的悉心指导和学习生活上的热情帮助，我深表谢意。感谢彭魁师兄、周政、李奚鹏、鲁芳、杨漫，感谢在做项目期间的合作和帮助。

感谢实验室的罗琳副教授、奚吉博士、乔杰博士、丁晶师兄、张军师兄、刘壮师兄以及王选刚、王育峰、李涛、江乐、宋鹏等同学在我的学习、研究与生活过程中给予我的无私帮助。

最后我要感谢我的家人和朋友，正是他们一如既往的关心、爱护、支持和鼓励，给了我创造的灵感和激情以及战胜困难的最大勇气。

参考文献

[1] 蒋焕文，孙续．电子测量[M]．第三版．北京：中国计量出版社，2008．1-2

[2] 武举．便携式数字存储示波器的研究与设计[D]：[硕士学位论文]．大连：大连理工大学，2005

[3] 王化．200MHz 数字存储示波表软件系统设计[D]：[硕士学位论文]．成都：电子科技大学，2005

[4] 仇杰．数字存储示波器设计及基于 ARM 的主控模块的实现[D]：[硕士学位论文]．南京：南京航空航天大学，2006

[5] 古天祥，王厚军，习友宝，等．电子测量原理[M]．北京：机械工业出版社，2004．281-283

[6] 邓斌．电子测量与仪器[M]．北京：国防工业出版社，2008．92-94

[7] 林占江，林放．电子测量仪器原理与使用[M]．北京：电子工业出版社，2006．132-135

[8] 赵茂泰．智能仪器原理与应用[M]．北京：电子工业出版社，2004．231-232

[9] National Semiconductor, Inc. ADC08D500 Data Sheet. 2005

[10] Xilinx,Inc. Spatan-3E FPGA Family:Complete Data Sheet.2005

[11] Samsung Electronics Co.Ltd. S3C2410 32-Bit Risc Microprocessor User's Manual 1.2. 2003

[12] 王田苗．嵌入式系统设计与实例开发[M]．第二版．北京：清华大学出版社，2003．3-4

[13] 潘巨龙，黄宁，姚伏天，等．ARM9 嵌入式 Linux 系统构建与应用[M]．北京：北京航空航天大学出版社，2006．1-2

[14] 孙天泽，袁文菊，张海峰．嵌入式设计及 Linux 驱动开发指南——基于 ARM9 处理器[M]．北京：电子工业出版社，2005．7-9

[15] 王卫国．嵌入式 Linux 在高速数据采集系统中的应用研究[D]：[硕士学位论文]．西安：西北工业大学，2004

[16] Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman. Linux Device Drivers, Third Edition. New York: O'Reilly, 2005. 5-6

[17] 锐极电子科技有限公司．ARM & Linux 嵌入式系统开发详解[M]．北京：北京航空航天大学出版社，2007．146

[18] 华清远见嵌入式培训中心．嵌入式 Linux C 语言应用程序设计[M]．北京：人民邮电出版社，2007．334-336

[19] 卫刚．200MHz 手持式数字存储示波表软件系统设计[D]：[硕士学位论文]．成都：电子科技大学，2006

[20] 陈尚松，雷加，郭庆．电子测量与仪器[M]．北京：电子工业出版社，2005．263-265

[21] 蒋薇.数字存储示波表的软件设计：[硕士学位论文].成都：电子科技大学自动化工程学院，2003

[22] 程小瑜．随机采样数字化示波器监控软件系统研究[D]：[硕士学位论文]．成都：电子科技大学，2003

[23] 赵琪．100MHz 数字存储示波器的软件设计[D]：[硕士学位论文]．成都：电子科技大学，2001

[24] 郭允晟．脉冲参数与时域测量技术[M]．北京：中国计量出版社，1989．2-6

[25] 陈长龄，宋玉娥．按模拟示波器检定方法评价数字示波器的若干问题探讨[J]．计量技术．2000，No2：43-46

[26] Harris F J. On the use of windows for harmonic analysis with the discrete Fourier transform .Proc. IEEE, 1978, 66(1):51~83

[27] 胡广书. 数字信号处理——理论、算法与实现[M]. 第二版. 北京：清华大学出版社，2003. 303-306

[28] Gade, Svend and Herlufsen, H., "Use of Weighting Functions in DFT/FFT Analysis (Part I)," Brüel & Kjær, Windows to FFT Analysis (Part I) Technical Review, No. 3, 1987. 6-21

[29] Tanja C. Hofner，乔宗标．高速模数转换器动态参数的定义和测试[J]．电子产品世界，2001 (02) ：

34-37
[30] 吴镇扬. 数字信号处理[M]. 北京：高等教育出版社. 2004. 64-73
[31] 王刚. 基于双 CPU 的数字存储示波器的设计与实现[D]: [硕士学位论文]. 南京：东南大学，2004
[32] 孙琼. 嵌入式 Linux 应用程序开发详解[M]. 北京：人民邮电出版社，2006. 184-185
[33] 王彤. C++类的多态性的再探讨[J]: 科技信息. 2008(14) ：6-8
[34] Bjarne Stroustrup. The C++ Programming Language. Special Edition. New York: Addison Wesley,2000. 305-320
[35] 宋宝华. Linux 设备驱动开发详解[M]. 北京：人民邮电出版社，2008. 478-528
[36] 王丽洁，习勇，魏急波. 基于 Qt / Embedded 和 Qtopia 的 GUI 设计[J]. 测控技术，2006（25）:316-319
[37] 王国英. 基于 QTE 的嵌入式 GPSGIS 车载导航系统设计与实现[D]: [硕士学位论文]. 杭州：浙江大学，2007
[38] 徐广毅，张晓林，崔迎炜，蒋文军. Qt /Embedded 在嵌入式 Linux 系统中的应用[J]. 单片机与嵌入式系统应用，20004，12(1) ：15- 18
[39] 于明,范书瑞,曾祥烨. ARM9 嵌入式系统设计与开发教程[M] . 北京:电子工业出版社,2006.293-295
[40] Jasmin Blanchette, Mark Summerfield.C++GUI Programming with Qt 3.Prentice Hall PTR, 2004.101-122
[41] Trolltech 公司. Qt 4.2 白皮书. 2006. 43
[42] Daniel Solin. 24 小时学通 Qt 编程[M]. 袁鹏飞译. 北京：人民邮电出版社，2000. 261-264