作为物联网项目，此课题不可避免得需要考虑物联网所面对的诸多挑战，以下列出其中4点：

1、**技术标准的统一与协调**。物联网感知层的数据多源异构，不同的设备有不同的接口，不同的技术标准；网络层、应用层也由于使用的网络类型不同、行业的应用方向不同而存在不同的网络协议和体系结构。

2、**管理平台问题**。物联网自身就是一个复杂的网络体系，加之应用领域遍及各行各业，不可避免的存在很大的交叉性。如果这个网络体系没有一个专门的综合平台对信息进行分类管理，就会出现大量信息冗余、重复工作、重复建设造成资源浪费的状况。

3、**成本问题**。各国对物联网都积极支持，在看似百花齐放的背后，能够真正投入并大规模使用的物联网项目少之又少。在成本没有达到普遍可以接受的范围内，物联网的发展只能是空谈。

4、**安全性问题**。传统的互联网发展成熟、应用广泛，尚存在安全漏洞。物联网作为新兴产物，体系结构更复杂、没有统一标准，各方面的安全问题更加突出。其关键实现技术是传感网络，传感器暴露的自然环境下，特别是一些放置在恶劣环境中的传感器，如何长期维持网络的完整性对传感技术提出了新的要求，传感网络必须有自愈的功能。 这不仅仅受环境因素影响，人为因素的影响更严峻。

一、调查法与实验法

1、调查法，调查法是科学研究中最常用的方法之一，它是有目的有计划有系统的搜集有关研究对象现实状况，历史状况的材料的方法。调查方法是科学研究中常用的基本研究方法，它综合运用历史法、观察法等方法以及谈话问卷、个案研究、测验等科学方式，对教育现象进行有计划的周密的系统的了解，并对调查搜集到的大量资料进行分析，综合比较归纳，从而为人们提供规律性的知识。

调查法中最常用的是问卷调查法，它是以书面提出问题的方式搜集资料的一种研究方法，其调查者就调查项目编制表分发会邮寄给有关人员请示填写答案，然后回收整理统计和研究。

2、实验法，实验法是通过组织变革控制研究对象来发现与确认事物间的因果联系的一种科研方法，其主要特点是：第一，主动变革性观察与调查，都是在不干预研究对象的前提下去认识研究对象，发现其中的问题，而实验却要求主动操纵实验条件，人为的改变对象的存在方式变化过程，使它服从于科学认识的需要。第二，控制性，科学实验要求，根据研究的需要，借助各种方法技术，减少或消除各种可能影响科学的无关因素的干扰，在简化纯化的状态下认识研究对象。第三，因果性实验已发现确认事物之间的因果联系的有效工具和必要途径。

二、规范研究与实证研究，实证研究方法排斥价值判断，规范研究方法却以价值判断为基础。

1、规范研究方法，规范研究方法，以某种价值判断为基础，说明经济现象及其运行应该是什么的问题。规范研究方法研究客观现象的目的在于提出一定的标准作为经济理论的前提，并以该标准作为制定经济政策的依据，以及研究如何使经济现象的运行符合或实现这些标准。

规范研究方法以某种价值判断为基础，解决客观经济现象应该是什么的问题，需要说明，所要研究的对象本身是好还是坏，对社会具有积极意义，还是具有消极意义。规范研究方法，研究经济现象的出发点和归宿离不开价值判断。规范研究方法就是从价值判断出发来研究经济现象，并研究如何实现上述标准。

2、实证研究法，实证研究法是科学实践研究的一种特殊形式，其依据现有的科学理论和实践的需要提出设计，利用科学仪器和设备在自然条件下，通过有目的有步骤的操作，根据观察记录特定与此相伴随的现象的变化，来确定条件与现象之间的因果关系的活动。主要目的在于说明各种自变量与某一个因变量的关系。

实证研究法是认识客观现象，向人们提供实在有用确定精确的知识的研究方法，其重点是研究现象本身是什么的问题。实证研究法试图超越或排斥价值判断，直接是客观现象的内在构成因素及因素的普遍联系，归纳概括现象的本质及其运行规律。

实证研究法的步骤：确定所要研究的对象，分析研究对象的构成因素相互关系以及影响因素，搜集并分类相关的事实资料，按设定假设条件，在研究的过程中，研究对象的行为是由其特征所决定，试图把所有复杂因素都包括进去，显然是不现实也不可能的。为此必须对某一理论所使用的条件进行设定。

当然假设的条件有一些是不现实的，但没有假设条件则无法进行科学研究，运用实证研究法研究问题，必须正确设定假设条件。

提出理论假说，假说是对于现象进行客观研究所得出的暂时性结论，也就是未经过证明的结论。假说对研究对象现象的经验性概括和总结，但还不能说明它是否能成为具有普遍意义的理论。

验证在不同条件和不同时间对假说进行检验，用事实检验其正确与否，检验，包括应用假说对现象的运动发展进行预测。

三、定量分析与定性分析：

1，定量分析法。在科学研究中，通过定量分析法，可以使人们对研究对象的认识进一步精确化，以便更加科学的揭示规律，把握本质，理性关系预测事物的发展趋势。

2，定性分析法。定性分析法就是对研究对象进行多个方面的分析，具体的说是运用归纳和演绎，分析与综合以及抽象与概括等方法，对获得的各种材料进行思维加工，从而能去粗取精，去伪存真，由此及彼，由表及里，达到认识事物本质，既是内在规律。

四、文献综合法与个案研究法：

1，文献综合研究法。文献研究法是根据一定的研究目的或课题，通过调查文献来获得资料，从而全面的正确的了解掌握所要研究问题的一种方法；文献研究法被广泛用于各种学科研究中，其作用能了解有关问题的历史和现状，帮助确定研究课题，能形成关于研究对象的一般印象，有助于观察和访问；能得到现实资料的比较资料；有助于了解事物的全貌。

2，个案研究法，个案研究法是认定研究对象中的某一特定对象加以调查分析，弄清其特点及其形成过程的一种研究方法。个案研究有三种基本类型：个人调查，其对组织中的某一个人进行调查研究；团体调查，即对某个组织或团体进行调查研究；问题调查及对某个现象或问题进行调查研究。

五、跨学科研究法

运用多学科的理论方法和成果，从整体上对某一课题进行综合研究的方法，也称交叉研究法，科学发展运动的规律表明，科学在高度分化中又高度综合，形成一个统一的整体。据有关专家统计，现在世界上有2000多种学科，而学科分化的趋势还在加剧，但同时各学科间的联系越来越紧密，在语言方法和某些概念方面有日益统一化的趋势。

1、为什么使用python和qt作为开发语言，有什么好处？

答：pyqt5是一套Python绑定Digia QT5应用的框架。它可用于Python 2和3。Qt库是最强大的GUI库之一。Qt强大之处在于网上有很多pyqt的资源，而且qt技术已经相当成熟，pyqt是采用基本和qt一致的api，因此之前使用过qt的人，上手pyqt很容易。

pyqt5是一个跨平台的工具包，它可以运行在所有主流操作系统。QT最大的好处是QT Desiginer，可以方便我们进行页面的布局，可以说在Tkinter里面需要大量代码完成的页面布局，在QT里面只要拖一拖控件就搞定了。

2、数据分页是怎么实现的？

答：表格展示数据时获取表格页数和每页大小，通过这两个主要参数调用持久层查询语句。查询语句根据这两个参数计算查询结果的偏移量和结果数量，同时根据查询参数组成一条查询总数据量的语句，用于获取符合条件结果总数，分页表格获取到这个参数后即可根据每页大小计算出分页的总页数。

3、信号数据读写同时进行是如何做到的？

答：使用了多线程并发操作和读写锁。一个线程专门用于从设备读取数据，当用户发起采集指令时便不断从设备读取一定大小的数据并写入到内存数组中，这一步是数据的写入，需要添加写锁，避免其他线程读取脏数据。由于gui操作需要在主线程中进行，因此数据的读取和绘制发生在主线程。考虑到系统性能和肉眼观察的局限，主线程刷新的频率稍低，大概为读数据线程速度的1/5，这一步是数据的读取操作，需要添加读锁，避免数据被写入。

|  |  |  |
| --- | --- | --- |
| 序号 |  |  |
| 1 | 2021/11/1-  2021/11/6 | 讨论项目可行性，完成项目初期准备工作 |
| 2 | 2021/11/7-  2021/11/10 | 项目框架搭建，初步设计并绘制用户界面 |
| 3 | 2021/11/11-  2021/11/15 | 开发通用工具，完成基础数据的增删改查功能 |
| 4 | 2021/11/16-  2021/11/19 | 实现图像流式显示，能够实时展示数字信号 |
| 5 | 2021/11/16-  2021/11/19 | 展示信号频谱图、时频谱图，界面整合 |
| 6 | 2021/11/20-  2021/11/24 | 细节优化，项目雏形完成 |
| 7 | 2021/11/25-  2021/11/30 | 差异性分析功能，数据备份、恢复功能，项目基本可用 |
| 8 | 2021/12/1-  2021/12/6 | 项目重构优化，功能测试、完善，第一版发布 |

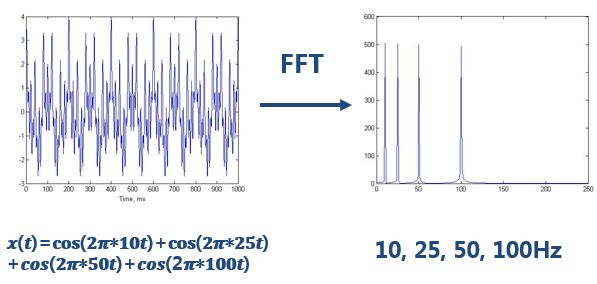
信号分析部分：

频谱图：

傅里叶变换是信号分析中的核心工具，它可分析信号的成分，也可用这些成分合成信号。许多波形可作为信号的成分，比如正弦波、方波、锯齿波等，傅立叶变换用正弦波作为信号的成分。

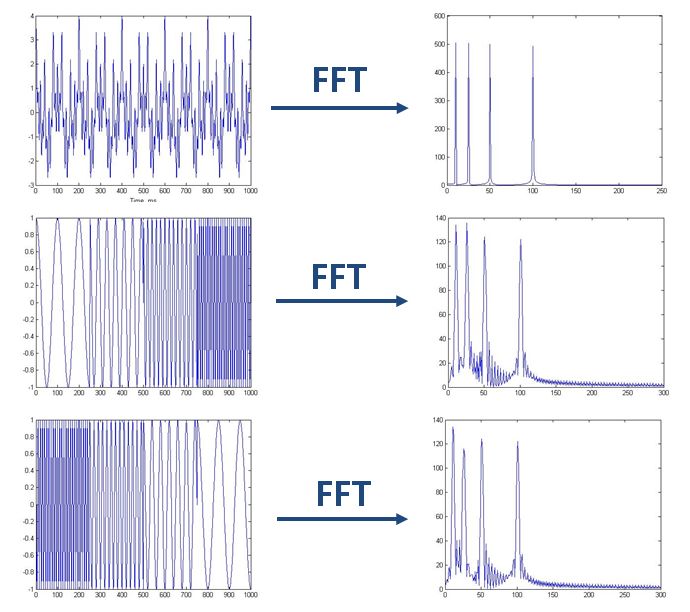
直接进行离散傅里叶变换的计算量较大，所幸存在一种快速傅立叶变换（fast Fourier transform）算法，简称FFT。由于FFT算法在信号分析领域应用甚广，在此就不再赘述。

对**非平稳**过程，傅里叶变换有局限性”。看如下一个简单的信号：



做完FFT（快速傅里叶变换）后，可以在频谱上看到清晰的四条线，信号包含四个频率成分。

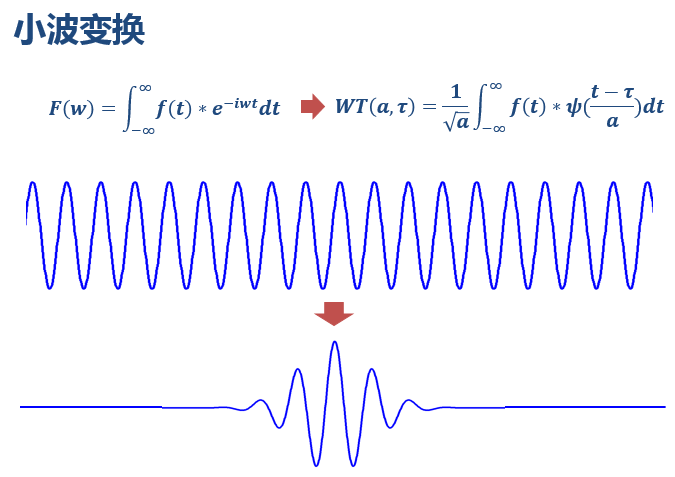
一切没有问题。但是，如果是**频率随着时间变化的非平稳信号**呢？



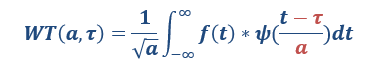
如上图，最上边的是频率始终不变的平稳信号。而下边两个则是频率随着时间改变的非平稳信号，它们同样包含和最上信号相同频率的四个成分。  
做FFT后，我们发现这三个时域上有巨大差异的信号，频谱（幅值谱）却非常一致。尤其是下边两个非平稳信号，我们从频谱上无法区分它们，因为它们包含的四个频率的信号的成分确实是一样的，只是出现的先后顺序不同。

可见，傅里叶变换处理非平稳信号有天生缺陷。它只能获取一段信号总体上包含哪些频率的成分，但是对各成分出现的时刻并无所知。因此时域相差很大的两个信号，可能频谱图一样。

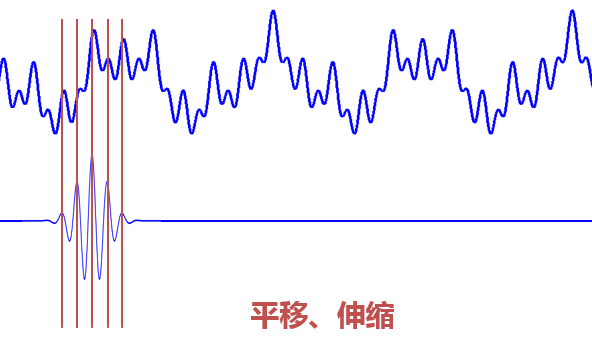
小波做的改变就在于，将无限长的三角函数基换成了有限长的会衰减的小波基



这就是为什么它叫“小波”，因为是很小的一个波嘛~



从公式可以看出，不同于傅里叶变换，变量只有频率ω，小波变换有两个变量：尺度a（scale）和平移量 τ（translation）。**尺度**a控制小波函数的**伸缩**，**平移量** τ控制小波函数的**平移**。**尺度**就对应于**频率**（反比），**平移量**τ就对应于**时间**。

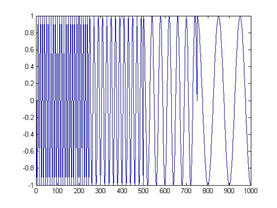


当伸缩、平移到这么一种重合情况时，也会相乘得到一个大的值。这时候和傅里叶变换不同的是，这**不仅可以知道信号有这样频率的成分，而且知道它在时域上存在的具体位置。**

而当我们在每个尺度下都平移着和信号乘过一遍后，我们就知道信号**在每个位置都包含哪些频率成分**。

看到了吗？有了小波，我们从此再也不害怕非稳定信号啦！从此可以做时频分析啦！

做傅里叶变换只能得到一个**频谱**，做小波变换却可以得到一个**时频谱**！



**信号采集与分析软件**

# 项目背景

## 项目需求

核心需求是采集数字信号持久化存储并提供一定的信号分析功能，能将正常信号收录到标准信号库，后续采集到的数据与标准信号进行对比，差异较大的标记为异常信号。

信号检测时应具备必要参数，如信号类别、检测地点，信号对比时优先选取参数相同的两个信号进行差异性分析。

程序还需具备基本数据的管理功能，包括用户登录与管理，普通用户与管理员权限分离，基本数据如信号类型、采集地点的增删改查功能。

软件需要在内网中运行，真实环境下只能使用本地数据库。

# 项目设计

## 逻辑结构设计

程序主体为桌面应用窗口程序，主界面展示各个菜单。首页需要登录名，密码。

（一）系统设置

设备自检：查看与虚拟示波器的连接状态，状态：正常，失败；操作：断开、连接，重连（先断开再连接）

检测点位：实现对舰上常用检测点位的管理，后续信号采集时需要对应具体点位；

信号类别：实现对信号类别的管理，如搜索雷达信号、导航雷达信号、火控雷达信号灯，后续信号采集时需要对应信号类别。

关闭系统：关闭整个主机。

（二）信号编辑

标准信号库：对检测点位+信号类别的标准信号库进行管理，添加信号、删除信号等；也可以直接从历史数据中直接点击“加入标准信号库”添加。

异常信号库：允许用户在常用信号的基础上注入非标准信号，形成异常信号库。

（三）检测维修

数据采集：选择检测点位、选择信号类别，点击开始采集，获取到示波器的数据并进行显示，底部显示信号分离的相关信息，并显示数据与标准信号库中信号对比的异常百分比；所有采集的数据需要自动保存到数据库中。

诊断分析：对所有历史数据进行分析，针对异常数据，允许用户查看更为详细数据。

历史数据：显示所有历史检测的信号数据，可以查看详情（以数据刚采集时的显示界面进行显示），加入标准信号库（该条信号记录进入对应检测点+信号类别的标准信号库中）。

（四）数据管理

数据设置：设置连接数据库的相关信息，只有admin才有此功能，允许设置连接数据库的用户名和密码。

用户管理：实现对登录用户的管理，初始默认需要有admin用户

数据备份：将数据库数据备份到目录文件；

数据恢复：将目录文件数据恢复到数据库；

（五）窗口管理

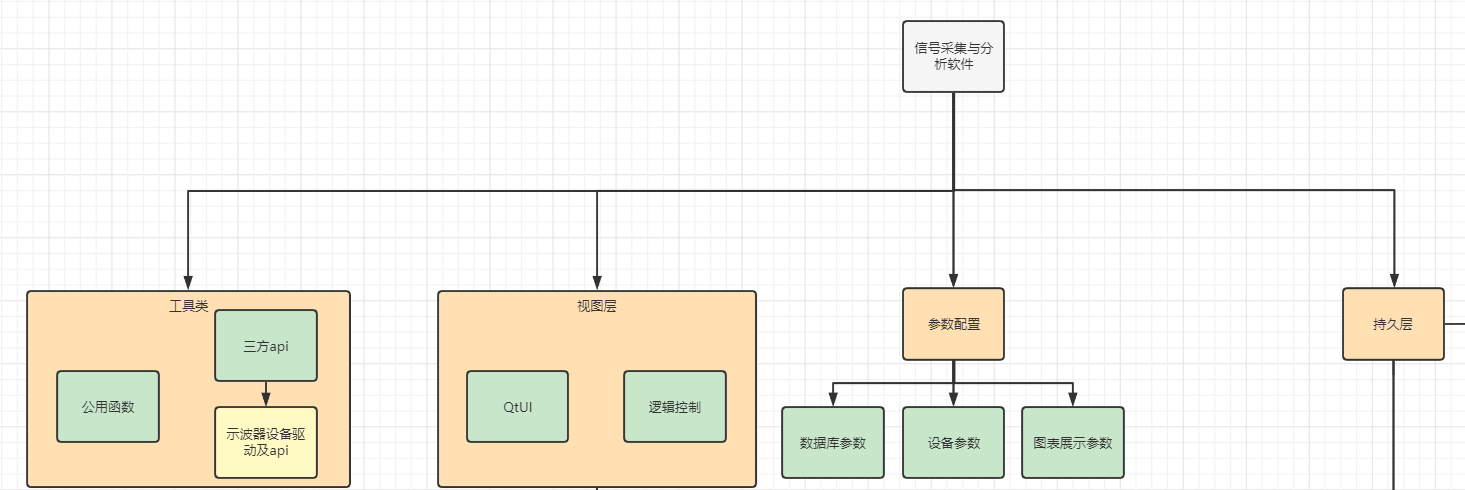
常规软件的窗口管理功能，如最大化、最小化等

（六）在线帮助

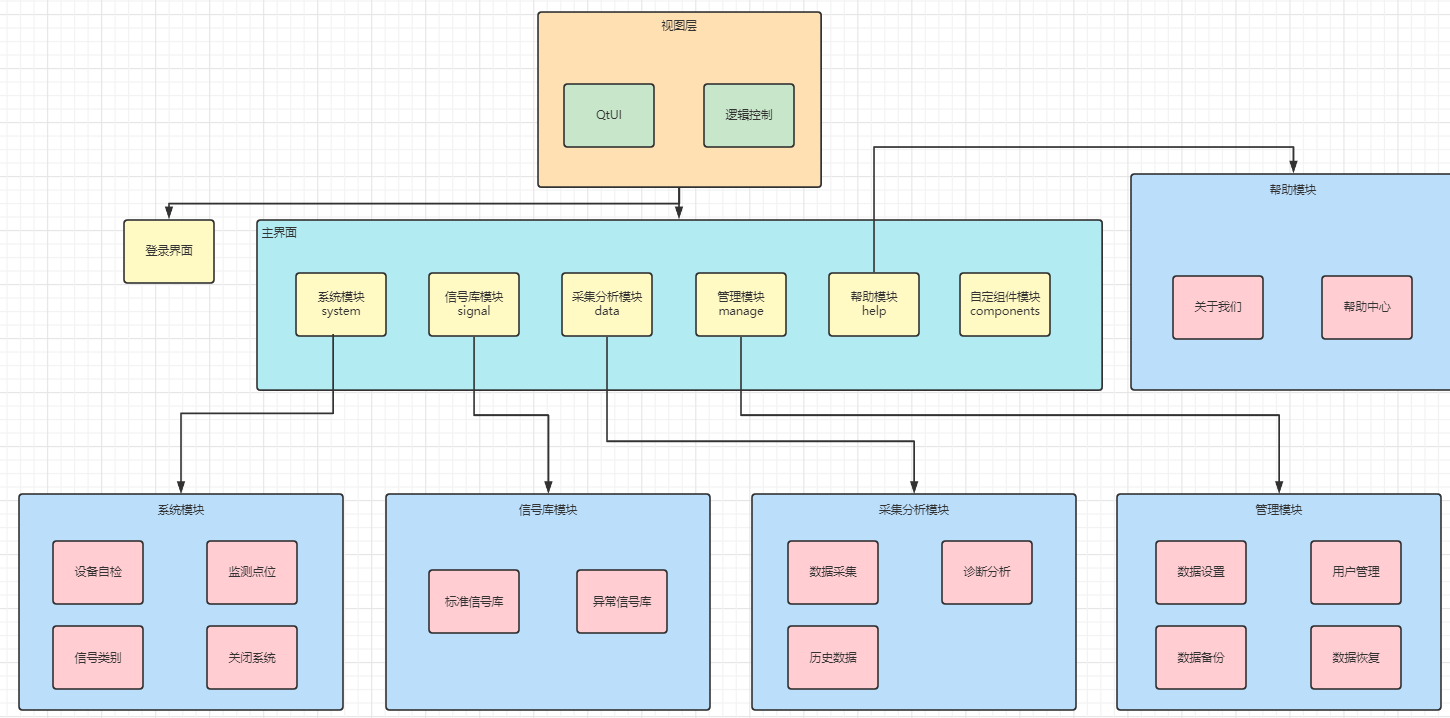
帮助中心

关于我们

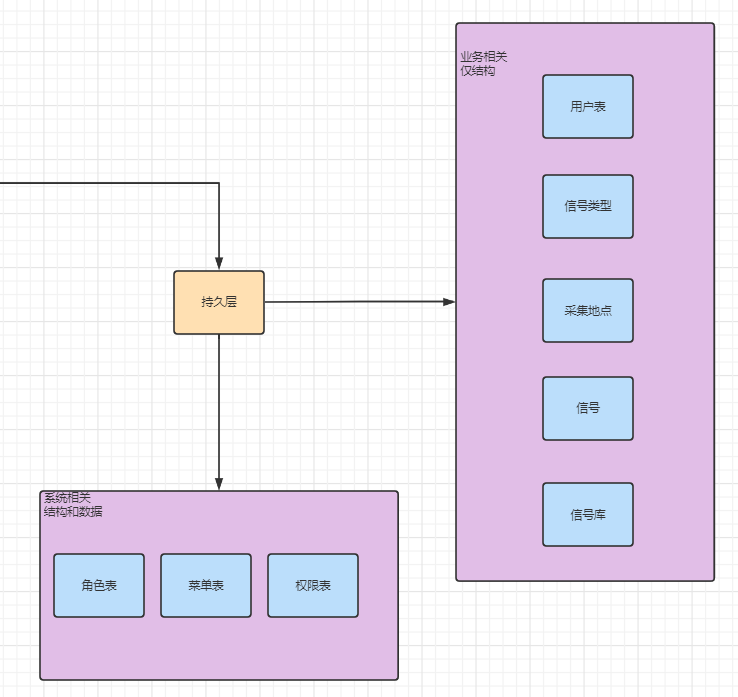
## 代码结构设计



图表 1系统整体结构

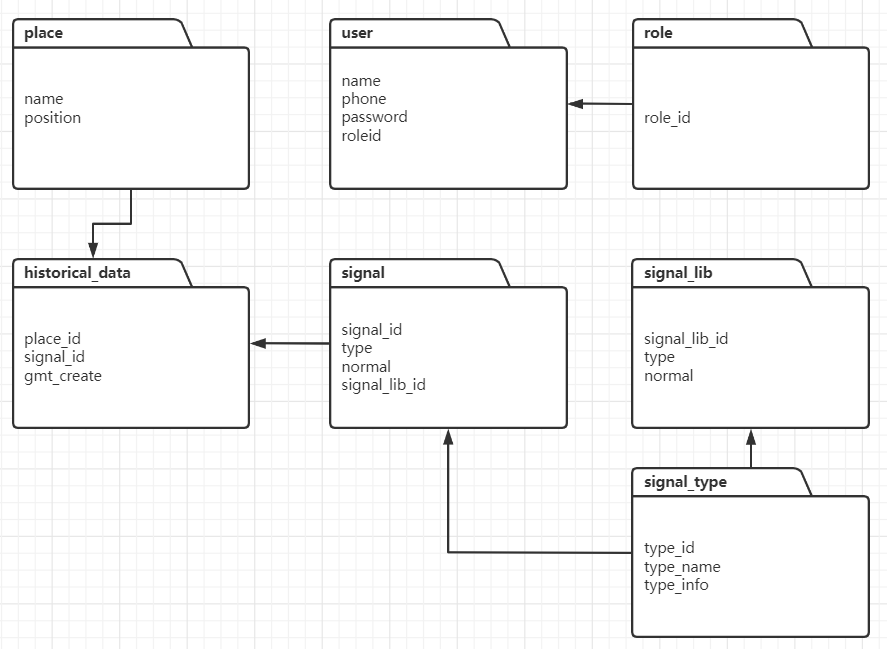


图表 2视图层结构



图表 3持久层结构

## 数据库设计

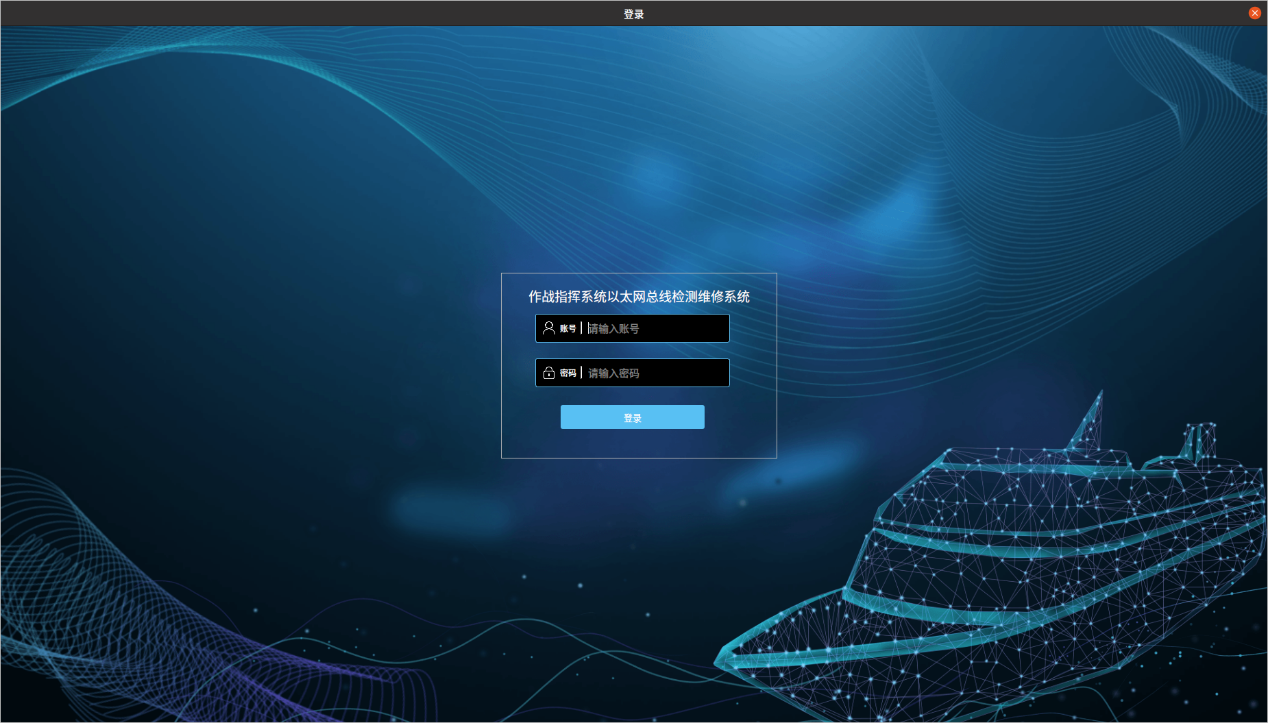


图表 4数据库设计

# 程序展示

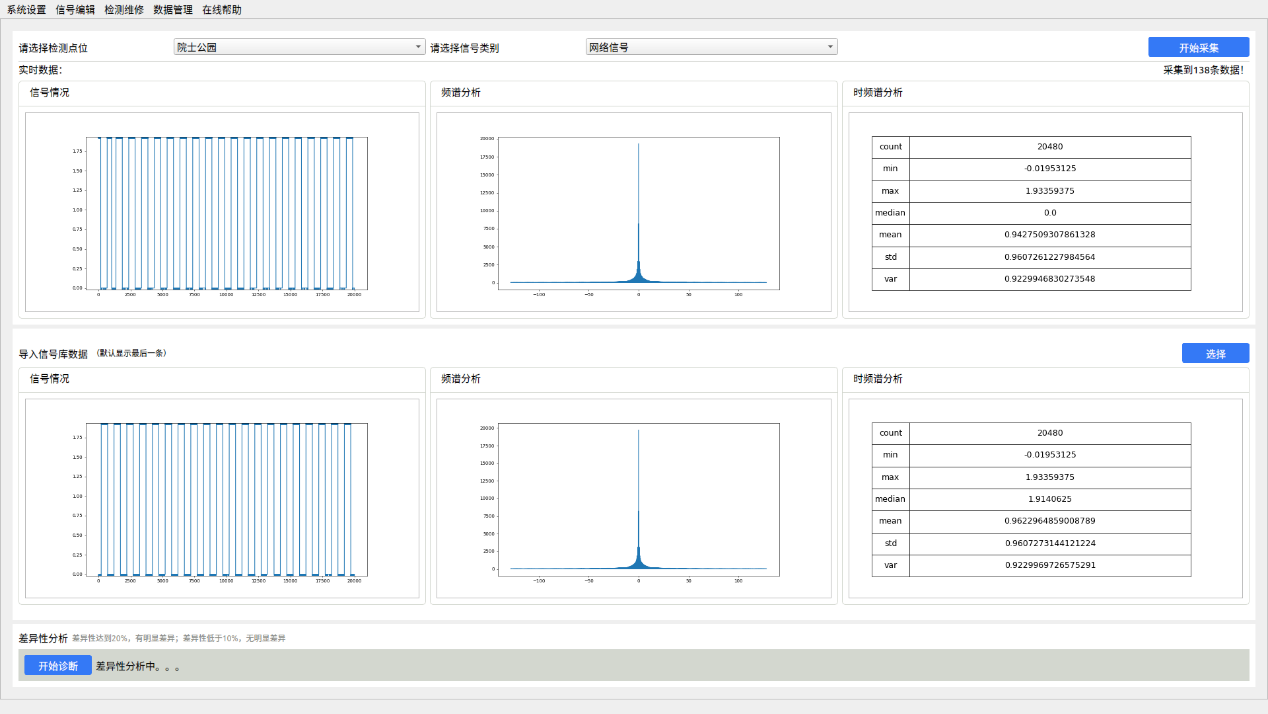
## 系统登录

系统界面如下图所示，输入登录账号和密码进入系统。



图表 5登录界面

登录成功后默认进入检测维修界面，如下图所示。



图表 6默认页面

## 系统菜单

本系统共有五个主要功能模块，分别是系统设置、信号编辑、检测维修和数据管理。

系统设置：包含“检测点位”、“信号类别”和“关闭系统”三个子模块。

信号编辑：包含“标准信号库”、“异常信号库”两个子模块。

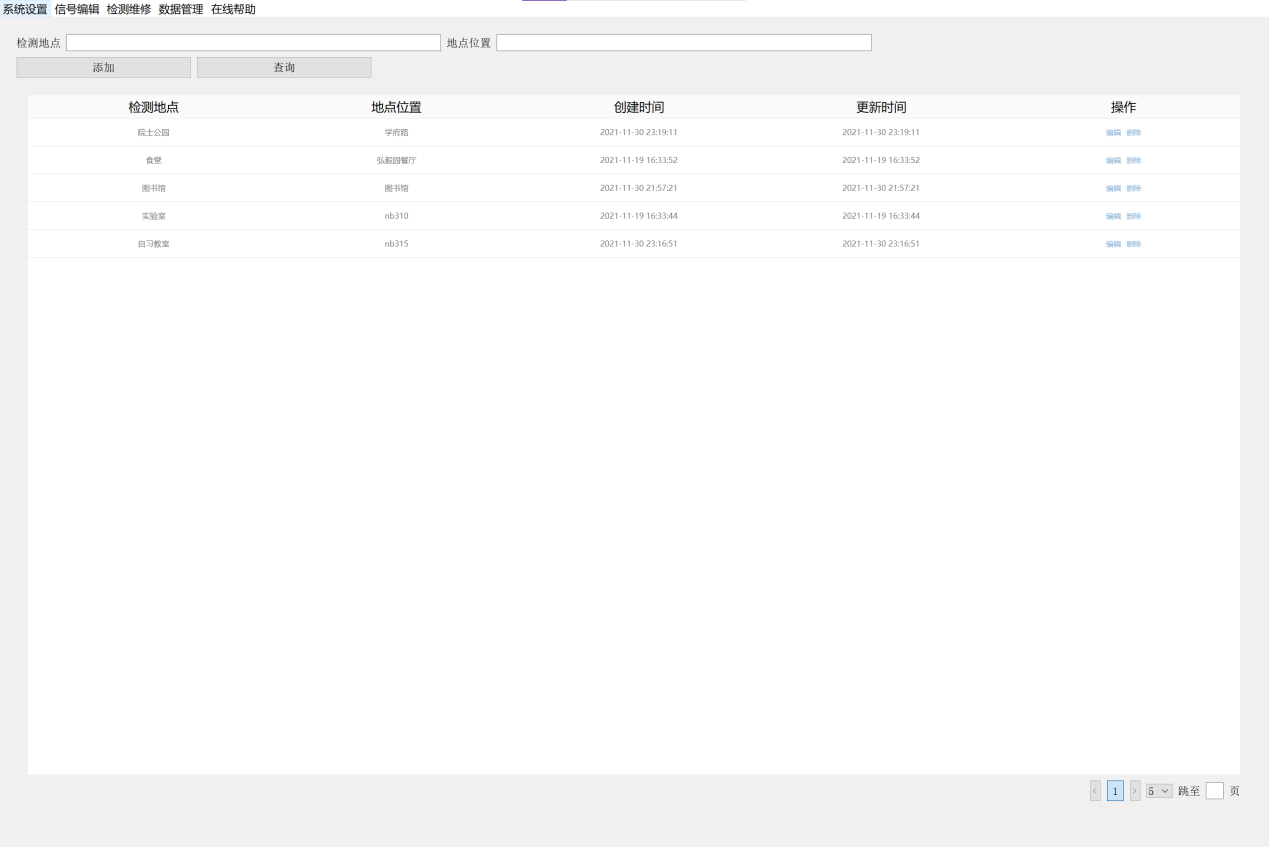
检测维修：包含“信号采集”、“诊断分析”和“历史数据”三个子模块。

数据管理：包含“用户管理”、“数据备份”和“数据恢复”三个子模块。

在线帮助：包含“关于我们”子模块。

## 系统设置

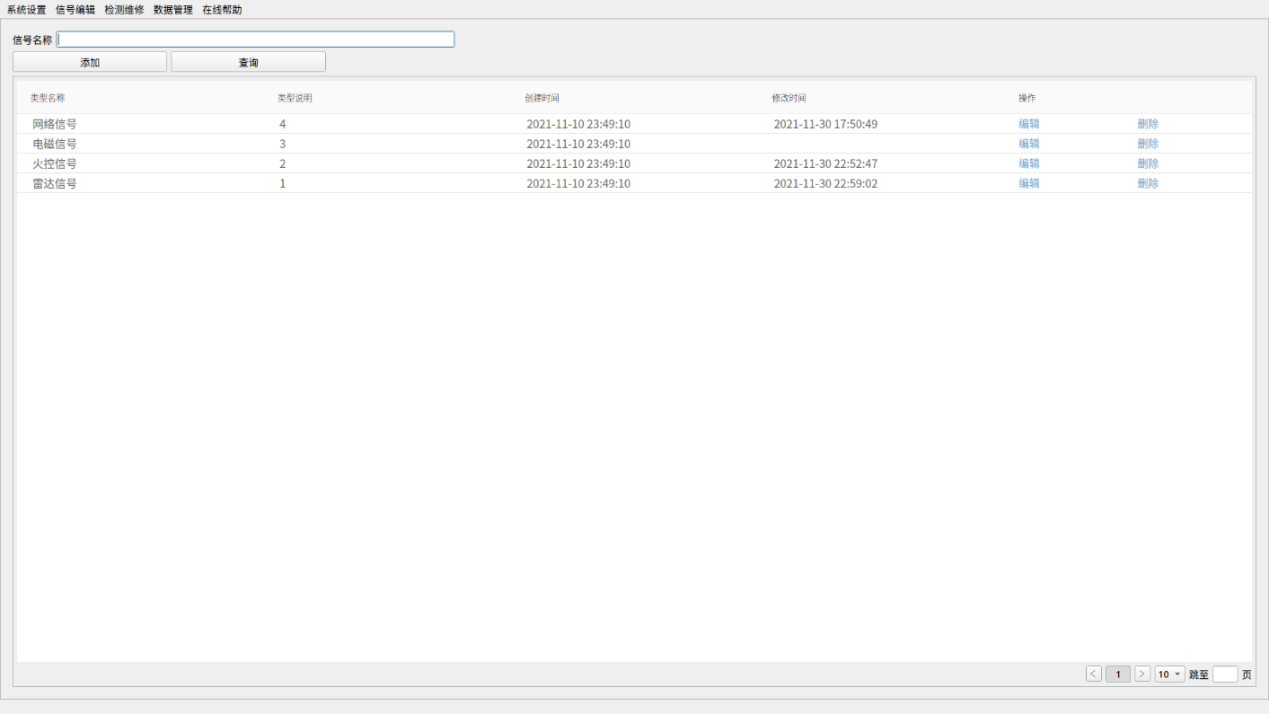
### 检测点位



图表 7监测点位

检测地点页面存储信号采集所可选的检测地点**。**具有基本的增加、删除、修改、查询功能

### 信号类别



图表 8信号类别

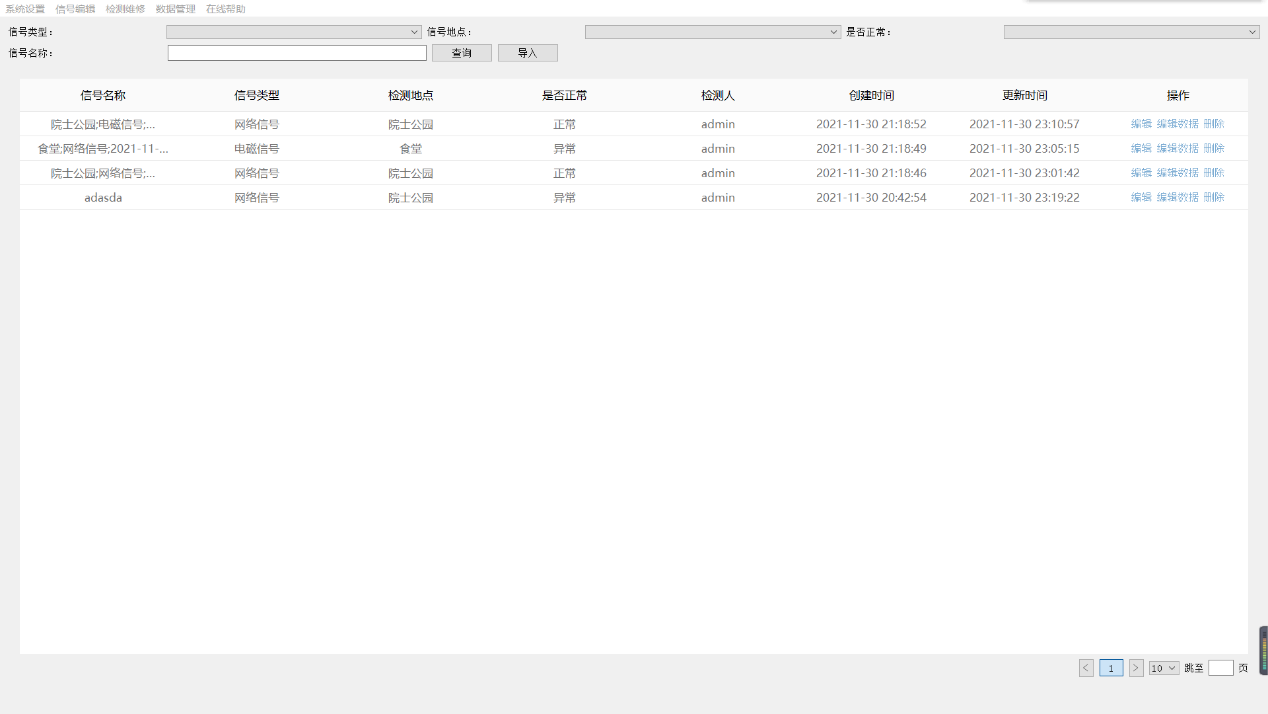
信号类别页面存储信号采集所可选的信号类别**。**具有基本的增加、删除、修改、查询功能。

### 关闭系统

关闭程序占有的所有资源（数据库连接、设备连接、子线程等）并退出系统。

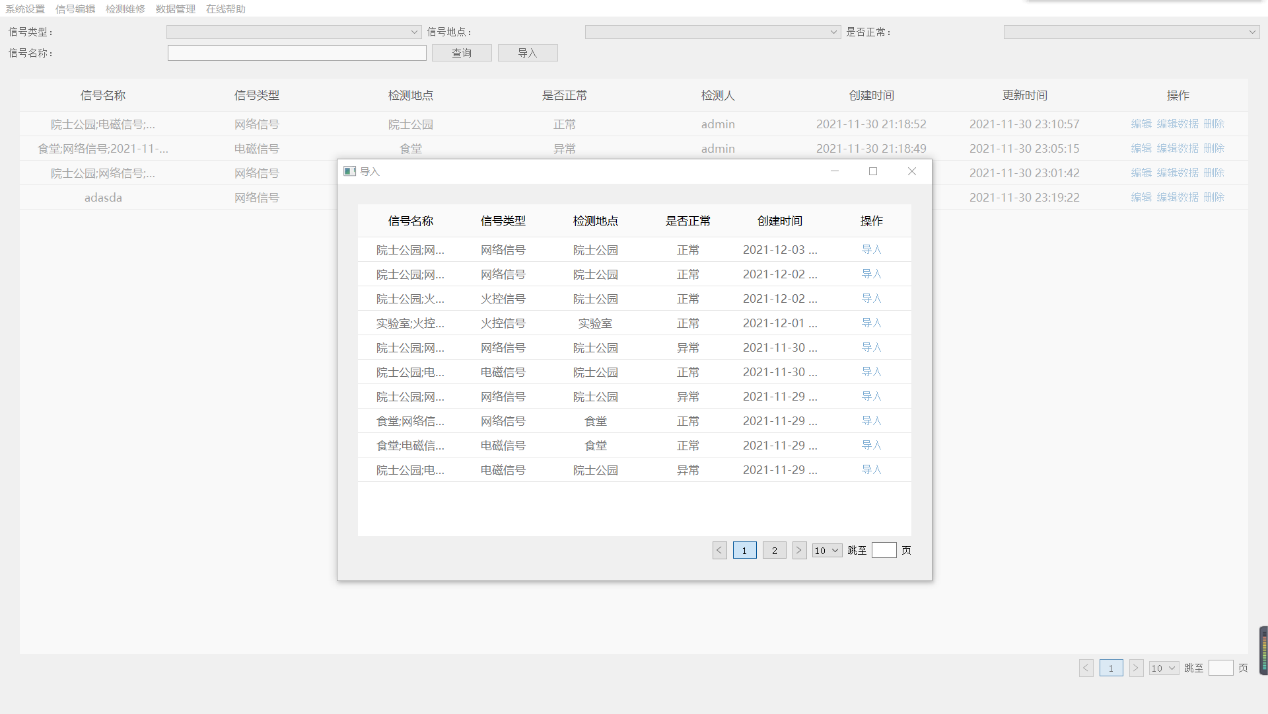
## 信号编辑

### 标准信号库



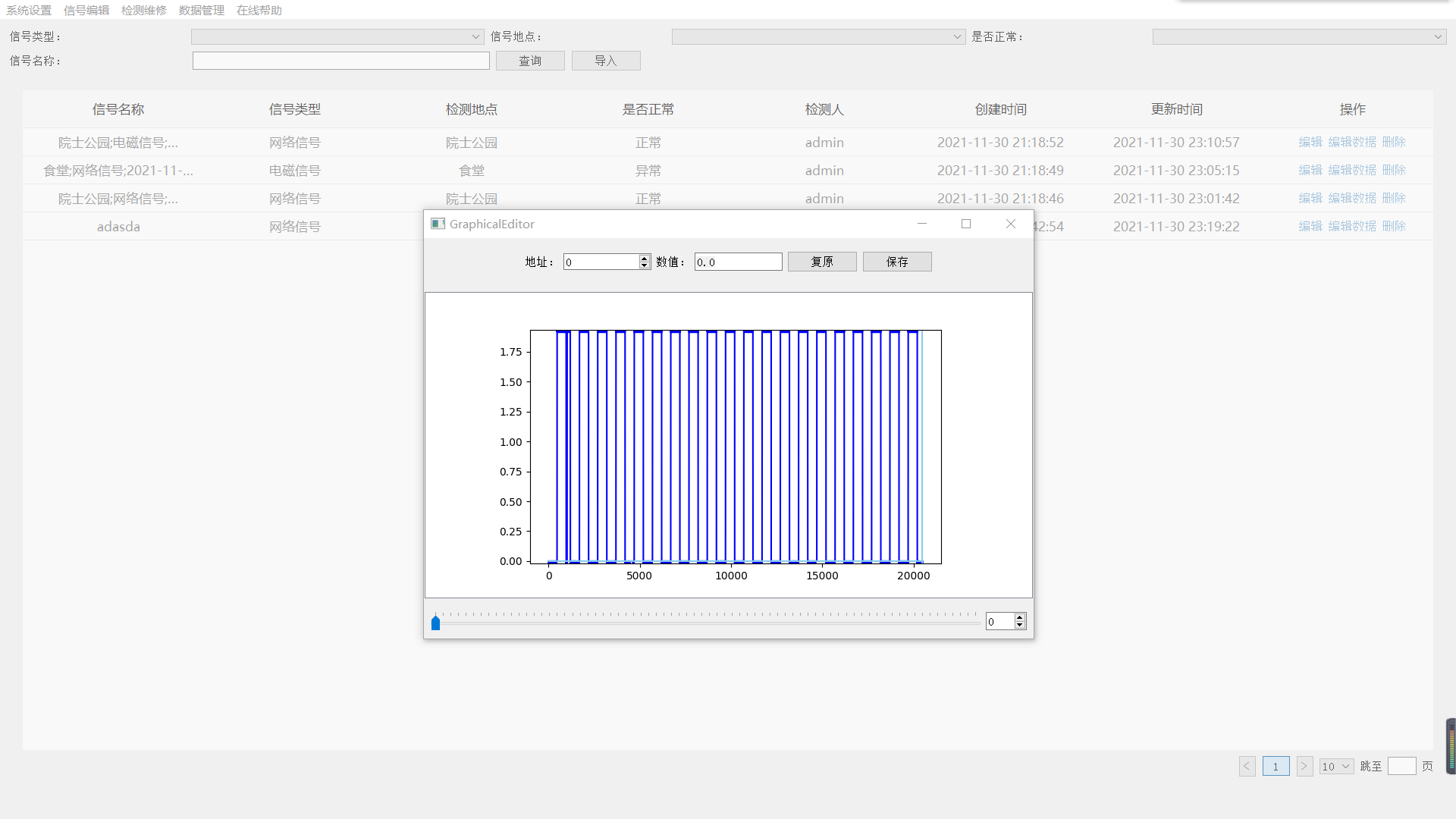
图表 9信号库页面

标准信号库页面存储关于采集标准信号的基本信息。具有基本的删除、修改、查询功能



图表 10导入信号库功能

可从历史数据中导入数据



图表 11编辑信号功能

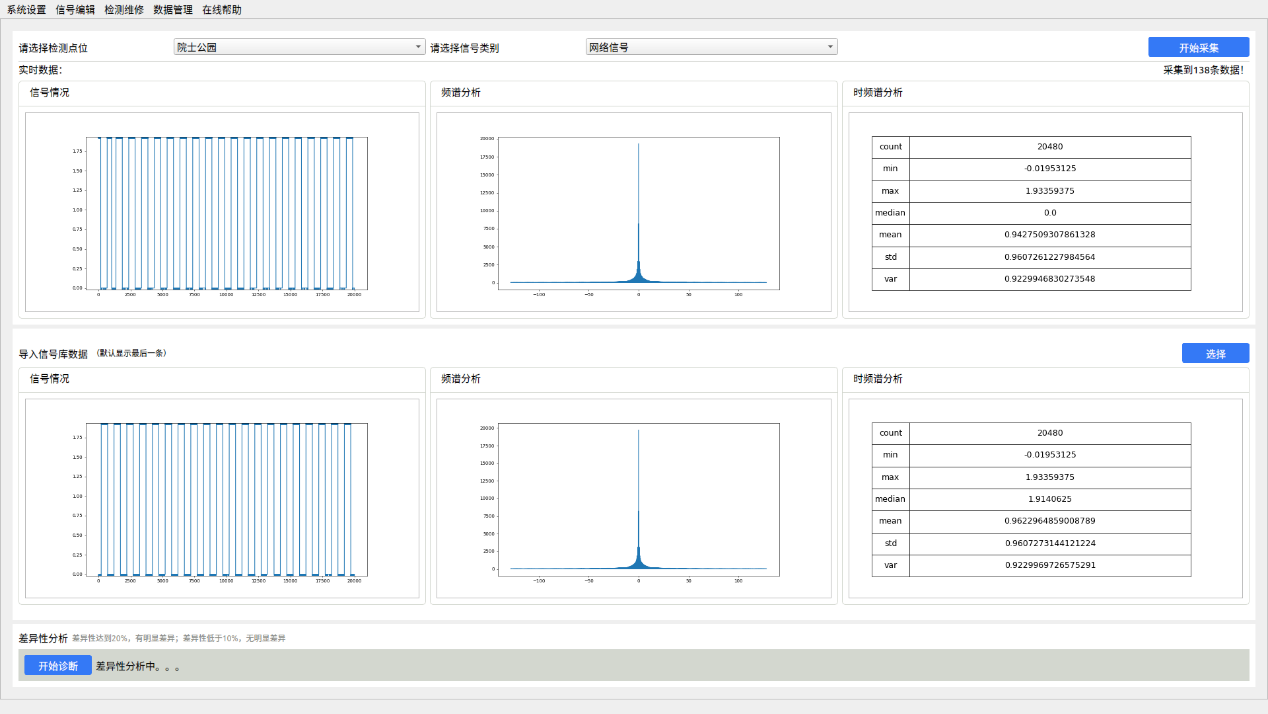
也可以编辑数据

## 检测维修

### 数据采集

进入页面后自动检测设备，一旦连接设备成功即可显示数据图表。选定监测点位和信号类别后等待信号稳定，点击右上角按钮即可随时开始采集数据。采集时间默认为3秒，3秒内采集到的数据将自动保存到数据库。

如果选定的监测点位和信号类别中恰有信号库数据，则自动获取到最后一条并播放一遍数据内容。此外，也可以点击右侧选择按钮，从信号库中手动选取一条数据与当前数据进行对比。

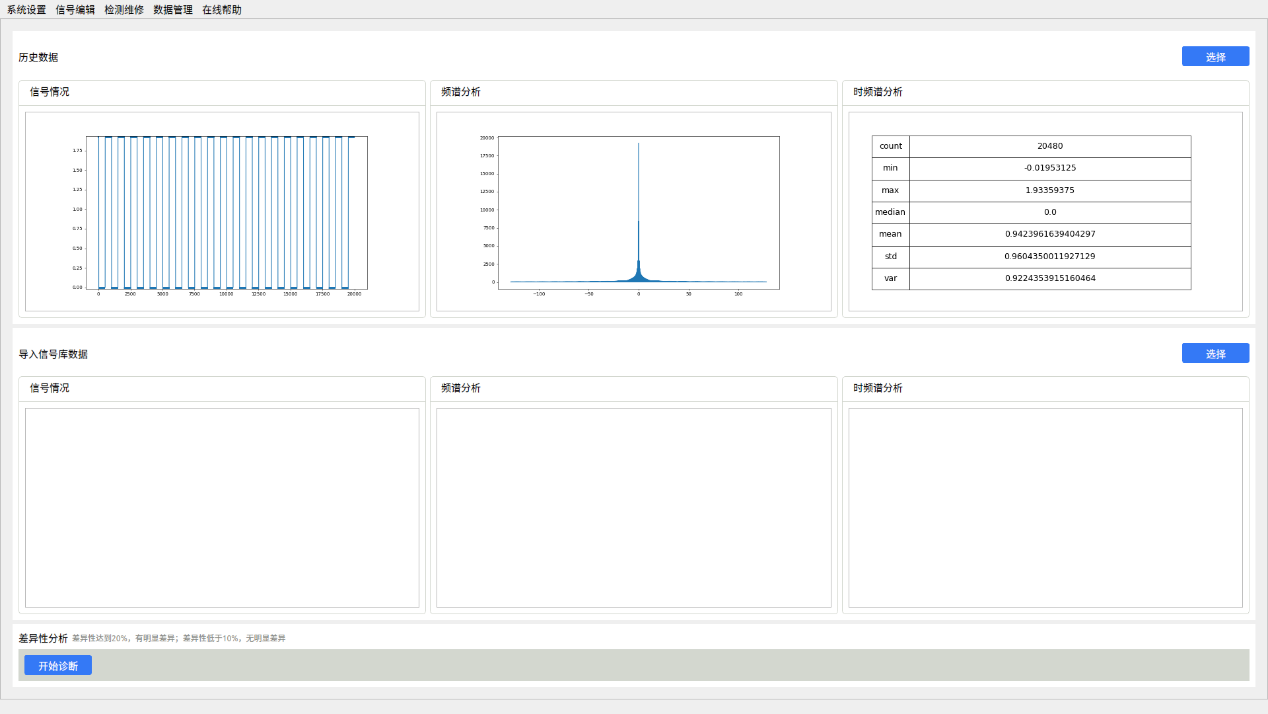


图表 12数据采集页面

当数据采集完毕并且选取了一条信号库数据时（即右上角显示采集到了…条数据并且下方信号库数据的图表成功展示时），即可点击左下角开始诊断按钮进行差异性分析。由于采集数据量较大，分析将置于后台处理，在大约3分钟后显示两条数据相似度，并保存到数据库，同时根据结果更新信号的正常状态。

### 诊断维修

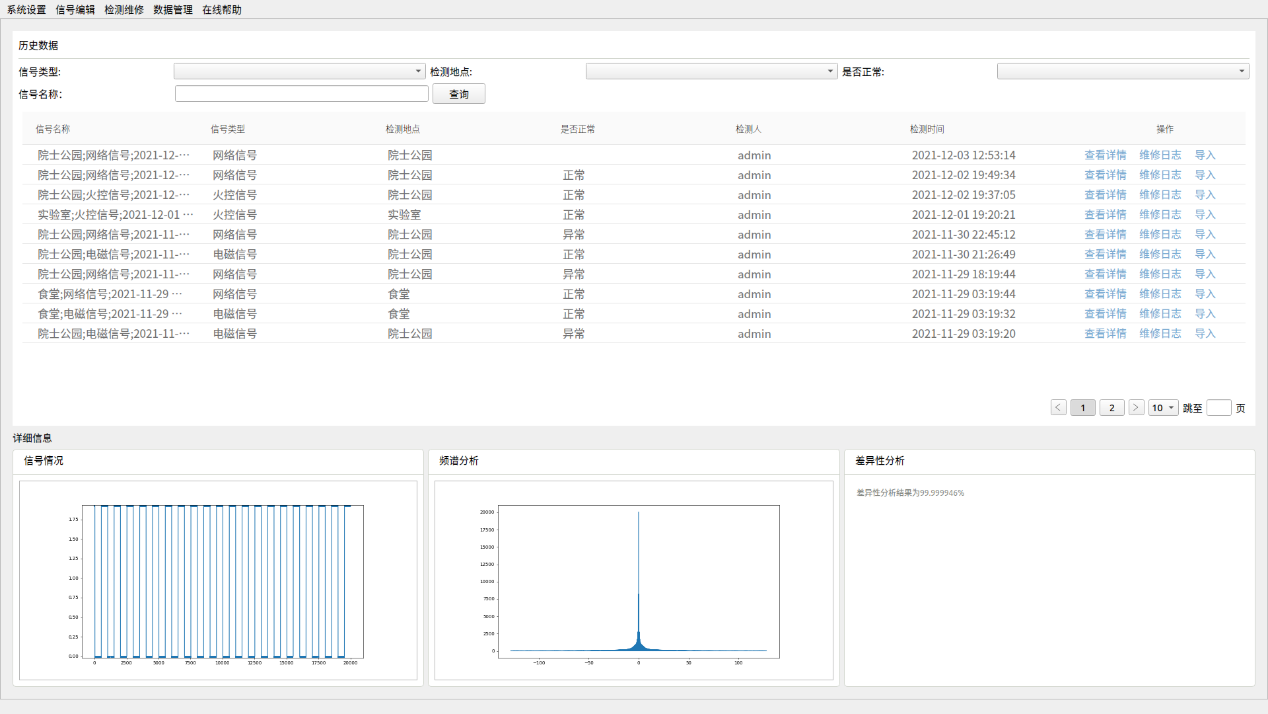
如果某条数据第一时间错过了差异性分析，而后发现该数据有参考价值时，可以在此页面进行历史采集数据与信号库数据的差异性分析。功能与数据采集界面类似，只是将实时采集的数据更换为数据库中取出的历史数据，降低了界面的复杂度。



图表 13诊断维修页面

### 历史数据

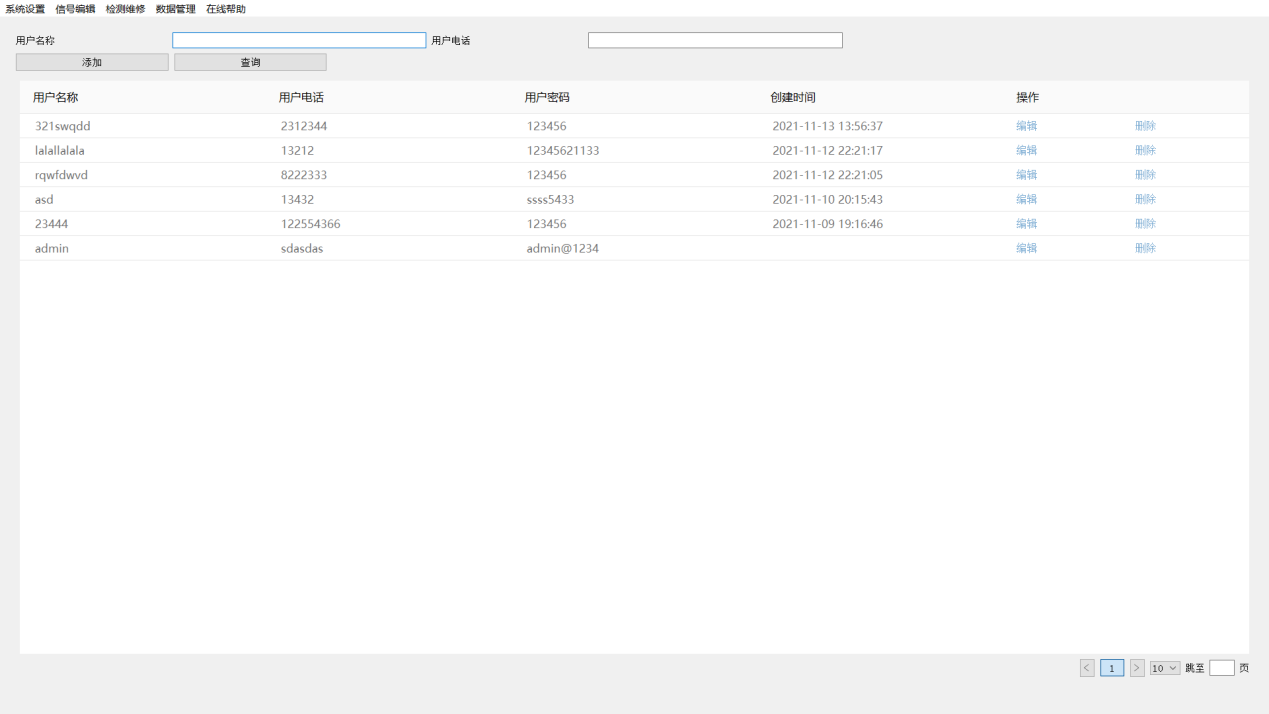
查看历史数据的具体情况，包括其数据参数、差异性分析结果、维修日志等。历史数据仅支持对维修日志的修改。点击导入按钮可以对应的历史数据导入到标准信号库中。



图表 14历史数据页面

## 数据管理

### 用户管理



图表 15用户管理页面

用户管理页面存储系统用户的信息**。**具有基本的增加用户、删除用户、修改用户、查询用户的功能

### 数据备份

数据备份实现备份数据库的功能，备份后可通过同模块的数据恢复功能来恢复数据库数据。

在选择数据库备份文件的存放路径及输入文件名后即可完成数据库的备份，数据库备份要花一点时间，请耐心等待，完成后备份文件便会存入到指定的文件路径中

### 数据恢复

数据恢复实现通过本地的数据库备份文件来恢复数据库的功能

选择数据库备份文件后，再输入系统提示的文字即可完成数据库的恢复，数据恢复要花一点时间，请耐心等待

# 总结

该项目是实验室定制的Linux GUI程序，用于收集并处理分析实时雷达信号。项目采用python和qt作为主要语言，持久层使用MySQL。信号采集模块使用usb外接虚拟示波器采集数据，通过github上第三方驱动和api与其通信，获取实时采集的数字信号数据。

我作为主要设计和开发人员，运用敏捷开发方法，快速设计搭建了项目架构。我手写了基于qt的分页器和分页表格，作为基本数据展示的基础组件。我封装了数据收集线程、波形展示等组件，用建造者、策略设计模式及多线程编程实现了信号数据的采集和展示。我在项目中还使用了单例、装饰器、备忘录、模板等设计模式，用广搜实现动态菜单、权限分离功能。

项目初步实现了核心需求，部分细节还需程序进一步迭代完善开发。

DESIGN OF A LOW-COST PC-BASED DIGITAL STORAGE OSCILLOSCOPE USING VIRTUAL INSTRUMENTATION

CHINMAY VILAS SAMAK & TANMAY VILAS SAMAK

Student, Department of Mechatronics Engineering, SRM Institute of Science and Technology, Tamil Nadu, India

ABSTRACT

Virtual instrumentation with minimal hardware setup is the current need of the engineering industry, and this research was particularly focused on virtualization of a Digital Storage Oscilloscope, one of the most crucial and widely used electronic measurement instruments. A PC-based digital storage oscilloscope was designed using NI LabVIEW and a compatible hardware interfacing unit was developed using Arduino Nano, an ATmega328 based development board. Keysight InfiniiVision DSO-X 2002A, a commercially available DSO, was used as a function generator and the performance analysis of the virtual oscilloscope was compared against the same standard. Six different test signals, viz. DC, pulse, square, triangle, sawtooth and sine were considered in order to analyze the comparative results acquired from the standard and the virtual DSOs, respectively. Results confirmed optimum performance of the designed virtual oscilloscope and assured its potential applications in the field of electronic measurements and instrumentation.

KEYWORDS:

Engineering Measurements, Oscilloscope, Signal Analysis & Virtual Instrumentation

Received: Feb 24, 2020; Accepted: Mar 16, 2020; Published: Apr 04, 2020; Paper Id.: IJECIERDJUN20202

1. INRTODUCTION

Oscilloscope is an electronic measurement instrument that displays voltage signals with respect to time. In effect, it plots the instantaneous voltage values of the signal as a function of time. Conventional oscilloscopes used cathode ray tubes (CRTs) to display the waveforms, and were known as cathode ray oscilloscopes (CROs). CROs are still used today by hobbyists to analyze continuous analog signals. However, the digital storage oscilloscopes (DSOs) have revolutionized the process of signal analysis. DSOs have an analog-to-digital converter (ADC), which converts the continuous analog signal into discrete samples of digital data with the bandwidth not exceeding half the sampling rate of the ADC, known as Nyquist limit [1]. This data is then processed using digital signal processors and displayed on an LCD or LED screen. DSOs can also save the signal data in various formats for subsequent analysis or documentation. Furthermore, DSOs can analyze multiple parameters of a signal at once and can also display overlays and annotations representing the same, which is a key factor responsible for their worldwide fame.

Instrumentation for the purpose of precise and accurate measurements and data acquisition is the spine of all engineering domains. However, traditional measuring instruments lack flexibility and scope for modifications, have greater costs, larger sizes and require more maintenance as well. Hence, there is a great need of virtual instrumentation – the notion of designing fully functional instruments virtually, on a software level with very minimal or absolutely no hardware components. A virtual instrument (VI) principally constitutes of a PC with the VI application and driver software installed, and a hardware interfacing unit in order to interface them with the physical world. A virtual instrument generally resembles the actual instrument in terms of visual appearance and can be operated using a graphical user interface (GUI).

The greatest advantage of virtual instruments is that, being software applications, they can be upgraded without many issues. Moreover, they cut down the cost of the actual hardware instruments and also the resources used for producing them. Furthermore, multiple virtual instruments can be installed on a single PC and used as and when required, rather than carrying heavy instruments everywhere. All in all, virtual instruments are modular, cost-effective and extremely convenient.

Virtual instrumentation is an emerging technology and some of the previous works in this field have addressed the same problem statement. However, each approach, including ours, is unique in terms of the hardware and software utilization for realizing the design and the overall functionality of the developed instrument. K. P. S. Rana et al. [2] developed a mixed signal virtual oscilloscope using NI LabVIEW and a DAQ card. They pointed out the problem of DSOs to not effectively process frequency-varying time signals and stated that mixed signal oscilloscopes (MSOs) used for this purpose are very costly; virtual instrumentation helped them achieve the goal of cost reduction. Ping Gong et al. [3] developed a virtual oscilloscope using NI LabVIEW and NI USB data acquisition card. They also discussed the timedomain analysis and frequency-domain analysis of the signals processed using their virtual oscilloscope. Norizam Sulaiman et al. [4] designed a PC-based 4-channel DSO using digital signal processing (DSP) techniques. They used peripheral interface controller (PIC) as the interfacing device and developed the software using Microsoft Visual C++. Wei Jiang et al. [5] designed an oscilloscope using virtual instrumentation technique. They used LabVIEW for designing the virtual instrument and utilized PCI-6024 data acquisition card as the interfacing device. Chandan Bhunia et al. [6] developed a PC-based virtual oscilloscope by designing a parallel port data acquisition device and a software package for displaying acquired signal. They also suggested the use of this low-cost oscilloscope for undergraduate laboratory demonstration and instruction purposes.

2. SYSTEM DESIGN

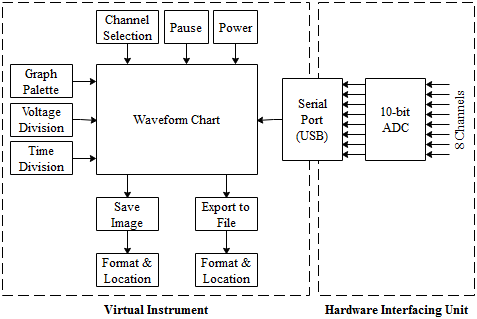


Figure 1: Block Diagram of the Proposed System.

The overall design of the proposed system (Figure 1) can be divided into two sub-domains, viz. the virtual instrument and the hardware interfacing unit. The virtual instrument is a software package designed to mimic the actual instrument that it represents, while the hardware interfacing unit establishes a connection between the real and the virtual world.

2.1 Virtual Instrument

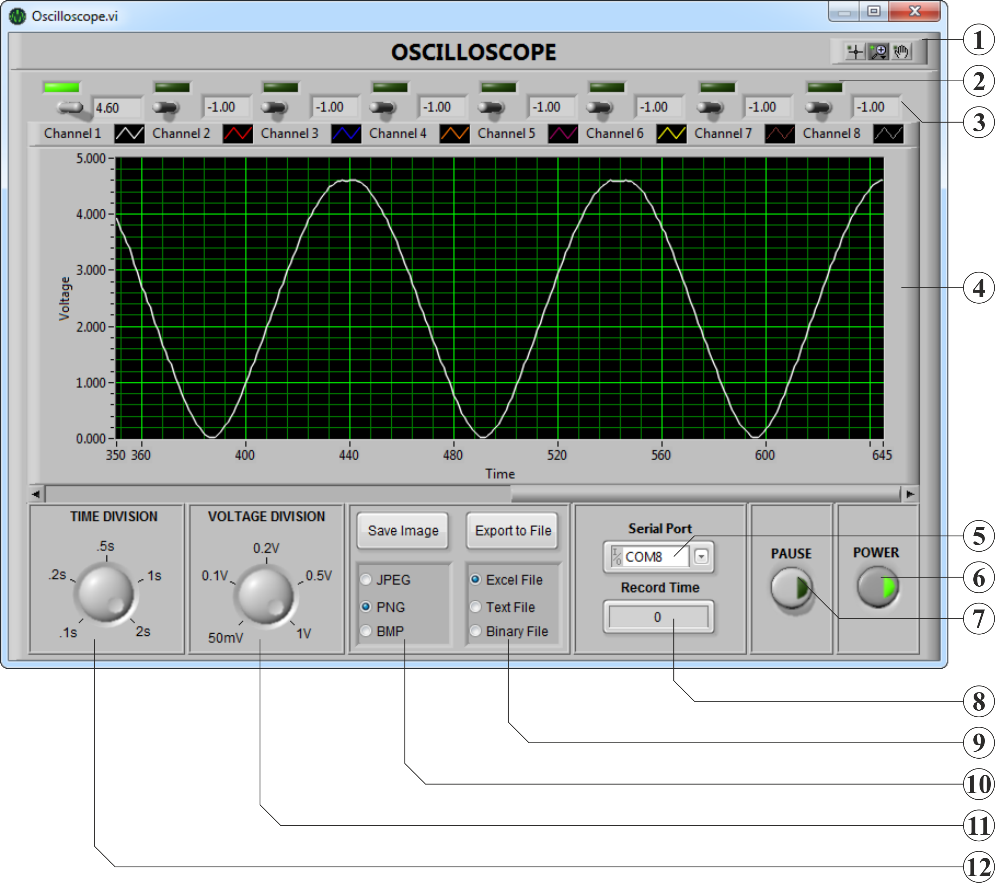


Figure 2: Front Panel (GUI) of the Virtual Instrument.

The virtual instrument was designed using NI LabVIEW and a standalone executable file of the virtual instrument was exported as a GUI application. Figure 2 illustrates the GUI or what is called the front panel of the virtual instrument and Table 1 holds the description of all the callouts represented in Figure 2.

Table 1: Description of Figure 2

|  |  |
| --- | --- |
| Label | Description |
| 1 | Graph Palette is used to move cursors, zoom and pan the graph. |
| 2 | Indicator LED indicates whether the channel is active or not. |
| 3 | Channel Readout displays the instantaneous voltage value. |
| 4 | Waveform Chart displays the real-time waveform. |
| 5 | Serial Port is used to select the COM Port for connection. |
| 6 | Power Button is used to switch the VI on or off. |
| 7 | Pause Button is used to pause the waveform. |
| 8 | Record Time is the time through which data has been exported. |
| 9 | Export to File is used to write data to a specified file format. |
| 10 | Save Image is used to export the instantaneous image of the VI. |
| 11 | Voltage Division is used to adjust the vertical divisions. |
| 12 | Time Division is used to adjust the horizontal divisions. |

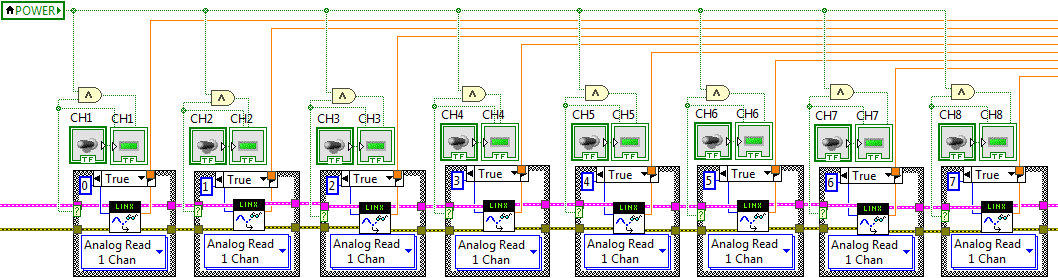


Figure 3: Block Diagram of Data Acquisition and Channel Controls.

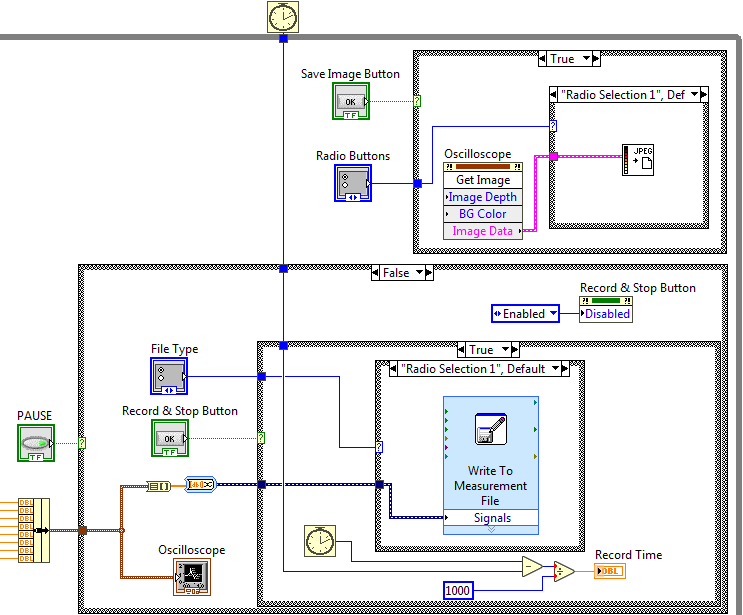
The channel controls (Figure 3) allow the user to select between 8 different channels, even multiple at a time, and the status of each channel is indicated by the corresponding indicator LED on the front panel. This sub-system also takes care of data acquisition of the individual channels.

Figure 4: Block Diagram of Waveform Display and Data Storage Controls.

Figure 4 illustrates the sub-system responsible for displaying the waveform and storing the data either as an image or as discrete values exported to a file. A waveform chart is used to graphically display the input signal with respect to time and the graph palette is used to move cursors, pan and zoom the display. The design supports consecutive viewing of signals from 8 channels at a time and the instantaneous voltage values of each channel are displayed next to their respective controls; if a channel is disabled, its voltage value is regarded as -1.00.

The data storage feature is what differentiates this virtual instrument from regular virtual oscilloscopes to be regarded as a virtual DSO. The data storage controls allow the user to save the waveform data as an image (JPEG, PNG or BMP), which is mostly used for the purpose of documentation or export the discrete values to a file (XLSX, LVM or TDMS) for subsequent analysis using third-party software such as Microsoft Excel, MATLAB, etc. In the latter case, record time is displayed in order to indicate the time through which data has been exported.

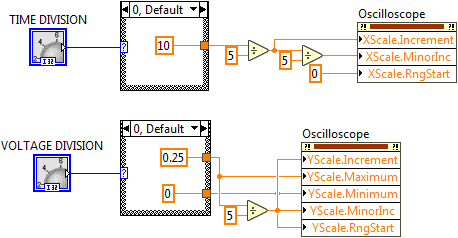


Figure 5: Block Diagram of Time and Voltage Division Controls.

The time and voltage division controls (Figure 5) are used to alter the X-axis (time) and Y-axis (voltage) scales, respectively. Suitable scaling factors are provided so as to obtain the time divisions of 100 ms, 200 ms, 500 ms, 1 s and 2 s and voltage divisions of 50 mV, 100 mV, 200 mV, 500 mV and 1 V, respectively.

Hardware Interfacing Unit

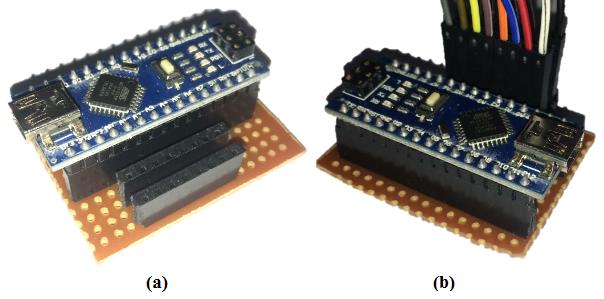


Figure 6: Hardware Interfacing Unit (a) Without Probes and (b) with Probes.

Since this research was focused on designing a cost-effective virtual DSO, the hardware interfacing unit was designed using Arduino Nano, a development board based on ATmega328. The analog inputs of Arduino were configured as 8 different channels and the built-in 10-bit ADC was used to digitize the input signals from (0-5) V to (0-1023) discrete values. The acquired data was transferred to the virtual instrument hosted on a PC using serial communication through the USB.

Figure 6 (a) shows the hardware interfacing unit without any probes so as to clearly display the connections for 8 different channels; the top rail is for signal lines of individual channels and the lower one is the GND rail. On the contrary, Figure 6 (b) shows the hardware interfacing unit with the probes attached. Probes of 8 different colors, viz. white, red, blue, orange, violet, yellow, brown and grey were used to indicate the respective channels and the same color convention was also followed while designing the front panel of the virtual instrument. All the GND probes are black in color.

Arduino Nano operates at clock frequency of 16 MHz and has its ADC set to 125 kHz (Equation 1). Since each conversion in AVR takes 13 ADC clock cycles, the maximum sampling rate was only about 10 kHz (Equation 2).

6 MHz/128 = 125 kHz (1)

125 kHz/13 = 9615 Hz ≈ 10 kHz (2)

RESEARCH METHODOLOGY

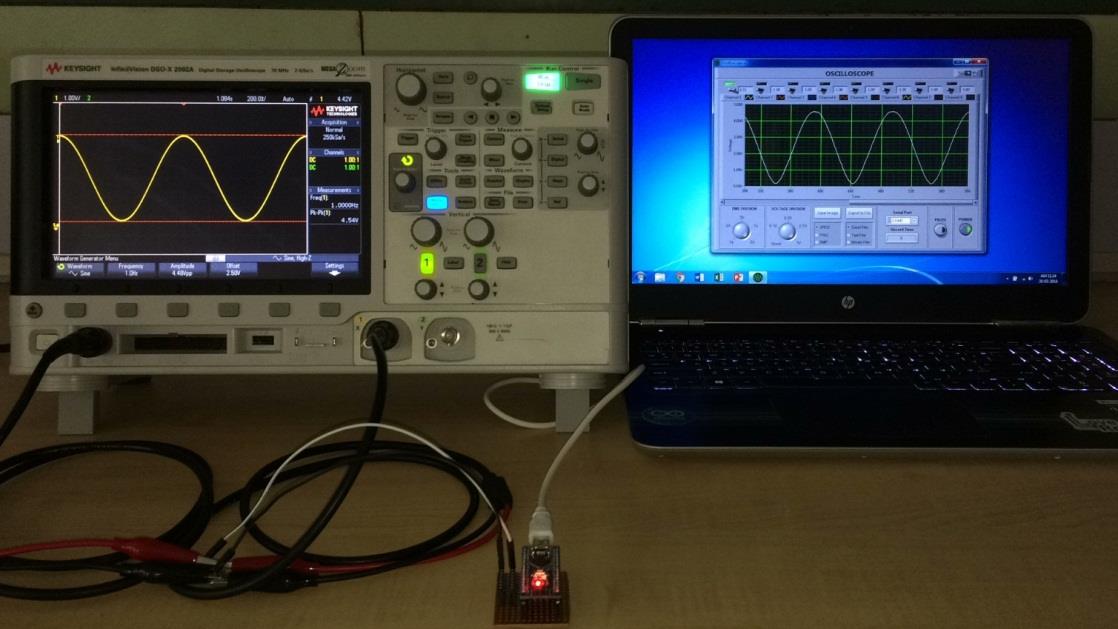


Figure 7: Experimental Setup.

Figure 7 shows the experimental setup used to analyze the performance of the proposed virtual DSO. The setup included commercially available standard DSO (Keysight InfiniiVision DSO-X 2002A [7]), hardware interfacing unit of the proposed virtual DSO and a personal computer (laptop PC) running the virtual instrument (stand-alone GUI application developed using NI LabVIEW).

Keysight InfiniiVision DSO-X 2002A, a standard DSO, was used as a function generator to generate the 6 standard test signals viz. DC, pulse, square, triangle, sawtooth and sine, each of 5 V amplitude (TTL logic preset) and 1.00 Hz frequency (except DC signal), which were fed back to the same DSO for analyzing their actual parameters.

The same test signals generated by the function generator were simultaneously fed to the proposed virtual DSO through the hardware interfacing unit by tapping the connections from Keysight InfiniiVision DSO-X 2002A and its performance was analyzed based on its ability to accurately display these test signals.

A comparative study was made in order to analyze and verify up to the mark behavior of the proposed virtual DSO with respect to a commercially available, standard DSO (Keysight InfiniiVision DSO-X 2002A).

Another set of experiments included signal input from various standard as well as non-standard sources. These signals mostly consisted of responses from actual engineering systems like mobile robots [8] – sensor data, controller input and output, feedback signals, etc. Acquisition of these signals was tested in terms of parameter accuracy and noise ratio. Overall performance of the designed oscilloscope was determined based on accuracy of displayed waveform and resolution of stored data.

4. RESULTS AND DISCUSSIONS

The proposed virtual DSO and standard DSO (Keysight InfiniiVision DSO-X 2002A) exhibited very small deviation in terms of output waveforms obtained from the experiments.

Table 2 displays peak-to-peak voltage and frequency values of signals analyzed on the standard and the virtual DSOs. Note that, voltage axis had an error of 1.3% and no particular error was observed for time axis.

The fact that virtual DSO could not analyze signals beyond 10 kHz, as opposed to 70 MHz for its counterpart, was a limiting factor for the design.

Table 2: Experiment Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signal | Keysight DSO |  | Virtual DSO |  |
|  | Vp-p | Frequency | Vp-p | Frequency |
| DC | 80 mV | - | 100 mV | - |
| Pulse | 4.54 V | 1.00 Hz | 4.60 V | 1.00 Hz |
| Square | 4.54 V | 1.00 Hz | 4.60 V | 1.00 Hz |
| Triangle | 4.54 V | 1.00 Hz | 4.60 V | 1.00 Hz |
| Sawtooth | 4.54 V | 1.00 Hz | 4.60 V | 1.00 Hz |
| Sine | 4.54 V | 1.00 Hz | 4.60 V | 1.00 Hz |

Figure 8 shows the test signals as observed on commercially available Keysight InfiniiVision DSO-X 2002A and the proposed PC-based virtual DSO comparatively.

• Figure 8 (a) shows DC signal with 5 V amplitude and 2.5 V offset as observed on the Keysight InfiniiVision DSO-X 2002A whereas Figure 8 (g) shows the same signal as observed on the proposed virtual DSO.

• Figure 8 (b) shows pulse signal of 100 µs pulse width with 5 V amplitude, 1 Hz frequency and 2.5 V offset as observed on the Keysight InfiniiVision DSO-X 2002A whereas Figure 8 (h) shows the same signal as observed on the proposed virtual DSO.

• Figure 8 (c) shows square wave of 50% duty cycle with 5 V amplitude, 1 Hz frequency and 2.5 V offset as observed on the Keysight InfiniiVision DSO-X 2002A whereas Figure 8 (i) shows the same signal as observed on the proposed virtual DSO.

• Figure 8 (d) shows triangle wave of 50% symmetry with 5 V amplitude, 1 Hz frequency and 2.5 V offset as observed on the Keysight InfiniiVision DSO-X 2002A whereas Figure 8 (j) shows the same signal as observed on the proposed virtual DSO.

• Figure 8 (e) shows sawtooth wave of 100% symmetry with 5 V amplitude, 1 Hz frequency and 2.5 V offset as observed on the Keysight InfiniiVision DSO-X 2002A whereas Figure 8 (k) shows the same signal as observed on the proposed virtual DSO.

• Figure 8 (f) shows sine wave with 5 V amplitude, 1 Hz frequency and 2.5 V offset as observed on the Keysight InfiniiVision DSO-X 2002A whereas Figure 8 (l) shows the same signal as observed on the proposed virtual DSO.

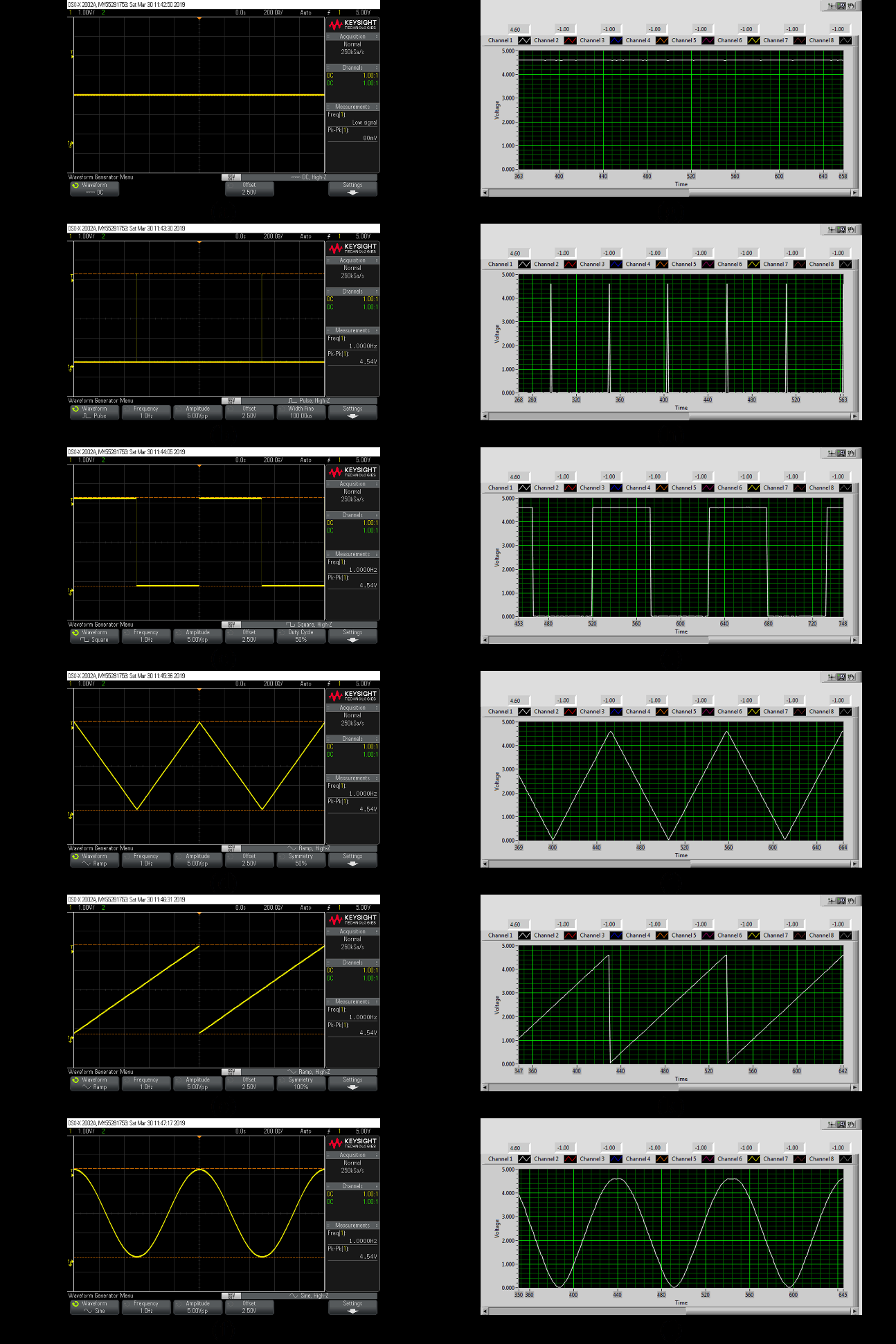


Figure 8: Comparative Results: (a)-(f) Output of Keysight InfiniiVision DSO-X 2002A and (g)-(l) Output of Proposed Virtual DSO

5. CONCLUSIONS

A low-cost PC-based virtual DSO (virtual instrument as well as hardware interfacing unit) was designed and its performance analysis exhibited optimal behavior in response to input signals. The tolerance limits of the designed virtual DSO lied within the range of mere ±1.3%; however, the sampling rate was limited to just 10 kHz and the DSO could only analyze voltage signals between 0 V and 5 V DC; both the restrictions being imposed due to the microcontroller’s limitations.

Future works of this research include, adopting sophisticated data acquisition system with high resolution ADC in order to achieve higher bandwidth and acquire AC and DC signals of greater magnitude, and adding more features to the virtual instrument such as triggering, time-domain analysis, frequency-domain analysis, virtual function generator, etc.

REFERENCES

1. *J. L. Hancock, “Evaluating Oscilloscope Sample Rates vs. Sampling Fidelity: How to Make the Most Accurate Digital Measurements”, American Society for Engineering Education, pp. 1-10, 2011.*
2. *K. P. S. Rana, et al., “A DAQ card based mixed signal virtual oscilloscope”, Elsevier Measurement., vol. 41, pp. 1032-1039, 2008.*
3. *P. Gong and W. Zhou, “Design and Implementation of Multifunctional Oscilloscope Using USB Data-Acquisition Card”, in Elsevier Procedia Engineering, IWIEE, vol. 29, pp. 3245–3249, 2012.*
4. *N. Sulaiman and N. A. Mahmud, “Designing the PC-Based 4-Channel Digital Storage Oscilloscope by using DSP Techniques”, IEEE, SCOReD, Malaysia, 2007.*
5. *W. Jiang and F. Yuan, “Design of Oscilloscope Based on Virtual Instrument Technique”, IEEE, 2nd ICPEITS, pp. 284–287, 2009.*
6. *C. Bhunia, et al., “A Low-Cost PC-Based Virtual Oscilloscope”, IEEE Transactions on Education, vol. 47, no. 2, pp. 295-299, May 2004.*
7. *7. InfiniiVision 2000 X-Series Oscilloscopes – Data Sheet, Keysight Technologies, Santa Rosa, CA, USA, 2018.*
8. *8. C. V. Samak and T. V. Samak, “Design of a Two-Wheel Self-Balancing Robot with the Implementation of a Novel State Feedback for PID Controller using On-Board State Estimation Algorithm”, IJRRD, vol. 8, no. 2, pp. 1-10, 2018.*