**氨氮分析仪嵌入式应用软件**

**概要设计**



编制：

校对：

审核：

审定：

批准：

日期： 年 月 日

**西安思坦科技有限公司**

**氨氮分析仪嵌入式应用软件**

**概要设计**



**西安思坦科技有限公司**

目录

[1. 项目简介 1](#_Toc481682058)

[2. 系统整体架构 1](#_Toc481682059)

[2.1. 结构视图 1](#_Toc481682060)

[2.2. 交互视图 2](#_Toc481682061)

[2.2.1. 主动发送 2](#_Toc481682062)

[2.2.2. 被动接收 4](#_Toc481682063)

[3. 关键技术方案 5](#_Toc481682064)

[3.1. 整体技术方案 5](#_Toc481682065)

[3.1.1. Qt开发框架 5](#_Toc481682066)

[3.1.2. 选择依据 5](#_Toc481682067)

[3.2. 用户图形界面方案 5](#_Toc481682068)

[3.2.1. QtQuick和QtQml 6](#_Toc481682069)

[3.2.2. 选择依据 6](#_Toc481682070)

[3.3. 数据持久化方案 7](#_Toc481682071)

[3.3.1. SQLite数据库 7](#_Toc481682072)

[3.3.2. 选择依据 7](#_Toc481682073)

[4. 模块设计 8](#_Toc481682074)

[4.1. 图形用户界面模块 8](#_Toc481682075)

[4.2. 数据存储模块 8](#_Toc481682076)

[4.3. 通信模块 9](#_Toc481682077)

[4.4. 控制器模块 9](#_Toc481682078)

[4.5. 数据模型定义 10](#_Toc481682087)

[5. 界面设计 10](#_Toc481682088)

[5.1. 用户与权限 10](#_Toc481682089)

[5.2. 数据测量 11](#_Toc481682090)

[5.2.1. 实时测量 11](#_Toc481682091)

[5.2.2. 计划测量 11](#_Toc481682092)

[5.3. 设置 12](#_Toc481682093)

[5.3.1. 系统设置 12](#_Toc481682094)

[5.3.2. 参数设置 12](#_Toc481682095)

[5.3.3. 周期设置 13](#_Toc481682096)

[5.4. 维护 13](#_Toc481682097)

[5.4.1. 系统维护 13](#_Toc481682098)

[5.4.2. 维护数据 14](#_Toc481682099)

[5.4.3. 功能测试 14](#_Toc481682100)

[5.5. 查询 14](#_Toc481682101)

[5.5.1. 历史数据 15](#_Toc481682102)

[5.5.2. 历史曲线 15](#_Toc481682103)

[5.5.3. 报警信息 16](#_Toc481682104)

[5.5.4. 日志信息 16](#_Toc481682105)

[5.5.5. 系统信息 17](#_Toc481682106)

# 项目简介

氨氮是水体的主要污染物质之一，氨氮分析仪是为了测量废水中的铵离子(NH+4) 浓度而设计的。

为了便于用户的操作与使用，氨氮在线分析仪的所有功能需要由软件来提供，并且需要具备操作简单，测量精度高、可靠性强、功能强大等特点，使用仪器自带液晶显示屏，可通过菜单完成仪器的设置、校准、清洗及氨氮数据的读取和查询等。

# 系统整体架构

## 结构视图

根据系统的功能性需求，系统整体可以划分为六大模块：图形用户界面模块（GUI）、控制器模块（Controller）、协议模块（Protocol）、通信模块（Communication）、数据存储模块（DataStore）、应用上下文模块（AppContext）。

图形用户界面模块提供了图形化的用户交互接口，提供了对用户输入数据的收集和下位机返回数据的显示；控制器模块对业务逻辑进行封装并向上层提供粗粒度功能接口；协议模块对具体协议格式进行定义并提供生成请求帧，解析响应帧以及格式验证等相关功能；通信模块提供了基于具体通信方式的粗粒度接口；数据存储模块提供了数据的持久化与逆持久化相关功能；应用上下文模块实现了基本的控制反转功能，为模块之间的解耦提供支持。

此外，控制器模块对其他模块的主动通信采用了观察者模式来避免模块之间的循环依赖。



## 交互视图

### 主动发送



### 被动接收



# 关键技术方案

## 整体技术方案

### Qt开发框架

Qt是一个1991年由奇趣科技开发的跨平台C++图形用户界面应用程序开发框架。它既可以开发GUI程序，也可用于开发非GUI程序，比如控制台工具和服务器。Qt是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器(Meta Object Compiler, moc)）以及一些宏，易于扩展，允许组件编程。

### 选择依据

采用Qt作为基础开发框架主要由以下几个优势：

1）优良的跨平台特性:Qt支持下列操作系统: Microsoft Windows 95/98， Microsoft Windows NT， Linux， Solaris， SunOS， HP-UX， Digital UNIX (OSF/1， Tru64)， Irix， FreeBSD， BSD/OS， SCO， AIX， OS390，QNX 等等。

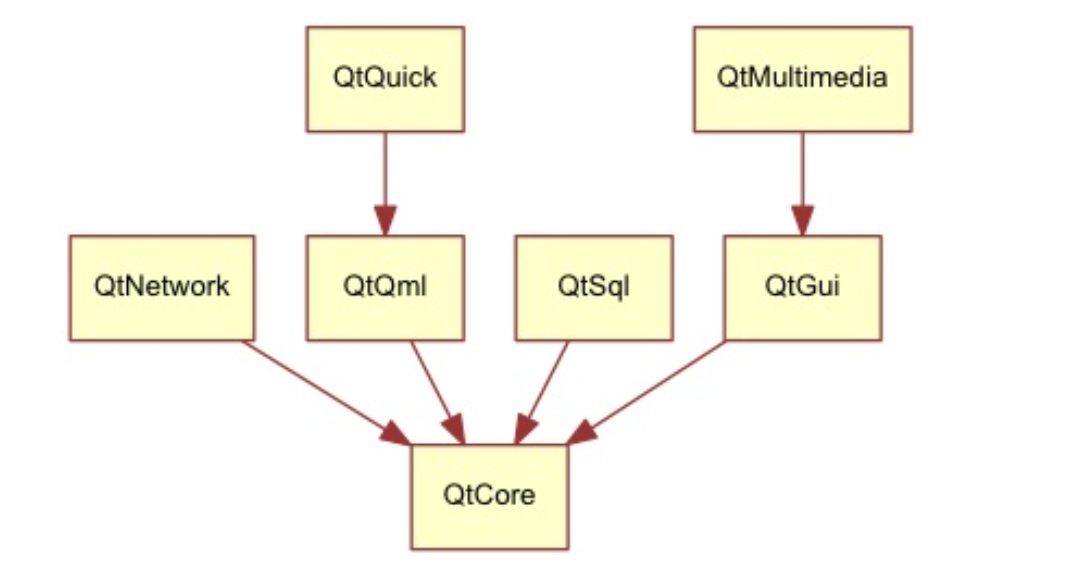
2）面向对象：Qt 的良好封装机制使得 Qt 的模块化程度非常高，可重用性较好，对于用户开发来说是非常 方便的。 Qt 提供了一种称为 signals/slots 的安全类型来替代 callback，这使得各个元件 之间的协同工作变得十分简单。

3）丰富的 API：Qt 包括多达 250 个以上的 C++ 类，还提供基于模板的 collections， serialization， file， I/O device， directory management， date/time 类。甚至还包括正则表达式的处理功能。

C++语言作为Qt框架的原生语言，能够最大限度的发挥出Qt框架的优势，同时C++语言运行时的高效性也保障了应用系统在配置较低的工控机上的流畅运行。

## 用户图形界面方案

Qt开发框架的主要强大之处在于跨平台图形化界面开发，现阶段有两种技术方案。传统的Qt应用界面开发过程中采用基于QtWidgets组件的声明式方式来实现用户界面，组件以C++类的方式由用户调用，因此界面的实现包含在用户编写的代码之中；另一种开发方式是基于QtQuick和QML的描述式方式。



### QtQuick和QtQml

QML是Qt推出的Qt Quick技术的一部分，是一种新增的简便易学的语言。QML是一种陈述性语言，用来描述一个程序的用户界面：无论是什么样子，以及它如何表现。在QML，一个用户界面被指定为具有属性的对象树。 这使得Qt更加便于很少或没有编程经验的人使用。

QML实际上是Qt Quick （Qt4.7.0中的新特性）核心组件之一：Qt Quick是一组旨在帮助开发者创建在移动电话，媒体播放器，机顶盒和其他便携设备上使用越来越多的直观、现代、流畅UI的工具集合。

### 选择依据

|  |  |  |
| --- | --- | --- |
|  | QtWidgets | QtQuick/Qml |
| 优点 | * 组件丰富、稳定 * 运用比较成熟 | * 开发效率的提升 * UI 与逻辑分离 |
| 缺点 | * 界面与代码耦合 | * 发展不成熟、可能存在问题 |

通过对项目需求和规模的分析，以及未来的技术趋势，决定采用基于QtQuick/Qml的技术作为图形界面开发技术方案。

## 数据持久化方案

### SQLite数据库

SQLite，是一款轻型的数据库，是遵守ACID的关系型数据库管理系统，它包含在一个相对小的C库中。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。它能够支持Windows/Linux/Unix等等主流的操作系统，同时能够跟很多程序语言相结合，比如 Tcl、C#、PHP、Java等，还有ODBC接口，同样比起Mysql、PostgreSQL这两款开源的世界著名数据库管理系统来讲，它的处理速度比他们都快。

### 选择依据

选择SQLite数据库主要基于以下几点原因：

1）使用数据库系统便于数据的存储、读取与查询；

2）SQLite满足系统的数据规模需求；

3）SQLite的特性更适合运行在配置较低的工控机环境下；

4）Qt框架对SQLite数据库实现了集成并提供了友好的编程接口。

# 模块设计

## 图形用户界面模块



QMLViewPage中主要包含了使用QML语言设计的用户界面；AbstractViewManager定义了QML界面与后台C++代码交互的抽象基类；ViewManager中包含了继承了AbstractViewManager的与特定的QML界面进行交互的类。

## 数据存储模块



CommonDBStore中包含了提供通用数据库操作的相关类；AppDBStore中包含了与特定业务逻辑相关的数据库操作类。

## 通信模块



SerialHelper包含了对串口通信相关功能的封装；CommunicationServer包含了与业务逻辑相关的通信接口类。

## 控制器模块



BasicController类定义了控制器的基本功能和所需实现的接口；AppController中包含了特定功能相关的控制器类实现。



## 数据模型定义



AbstractDataModel定义了抽象的数据模型基类；DataModel中包含了特定功能所需特性集合的数据模型。

# 界面设计

## 用户与权限

系统用户类型分为管理员和操作员两类。操作员可进行日常检查或定期检查，但测量条件和系统设置须由管理员负责。



## 数据测量

数据测量分为实时测量和计划测量两种类型。

### 实时测量

实时测量在用户启动后立刻启动测量。



### 计划测量

计划测量在到达用户设定测量时间后启动测量。



## 设置

### 系统设置



系统设置中，可以对系统日期时间、报警开关、报警上下限、水泵预起时间及继电器、开关量输入、4 – 20mA 电流输出、自动注销时间等进行设置。

### 参数设置



参数设置功能，可以对标样浓度、测量量程、消解温度及时间等参数进行设置修改。

### 周期设置



周期设置主要是设定系统自动校准、自动清洗和自动测量的时间周期。

## 维护

维护功能主要提供操作员对设备进行标定校准、清洗、各主要部件测试等维护工具。

### 系统维护



### 维护数据



### 功能测试



## 查询

查询功能，可以对系统历史数据、测量数据趋势图、报警信息及系统信息进行查询。

### 历史数据



进入历史数据功能界面后，点击【查询】按钮，可以查询满足指定条件的历史数据；点击【当天】按钮，显示当天测量结果数据；点击【前一天】、【后一天】按钮向前或向后翻看前一天或后一天的历史数据。

### 历史曲线



历史曲线查询将以趋势图形的方式，将历史数据呈现出来，可以直观的查看数据变化趋势情况同时也可以向前或向后翻看前一天或后一天的趋势图形。

### 报警信息



进入报警信息功能界面后，点击【查询】按钮，可以查询满足指定条件的报警信息数据；点击【当天】按钮，显示当天报警信息数据；点击【前一天】、【后一天】按钮向前或向后翻看前一天或后一天的报警信息数据。

### 日志信息



进入日志信息功能界面后，点击【查询】按钮，可以查询满足指定条件的日志信息数据；点击【当天】按钮，显示当天日志信息数据；点击【前一天】、【后一天】按钮向前或向后翻看前一天或后一天的日志信息数据。

### 系统信息



系统信息显示本设备固件版本号及硬件版本号，便于系统软硬件升级管理和维护。