

1 引言

1.1 编写目的

1.2 背景

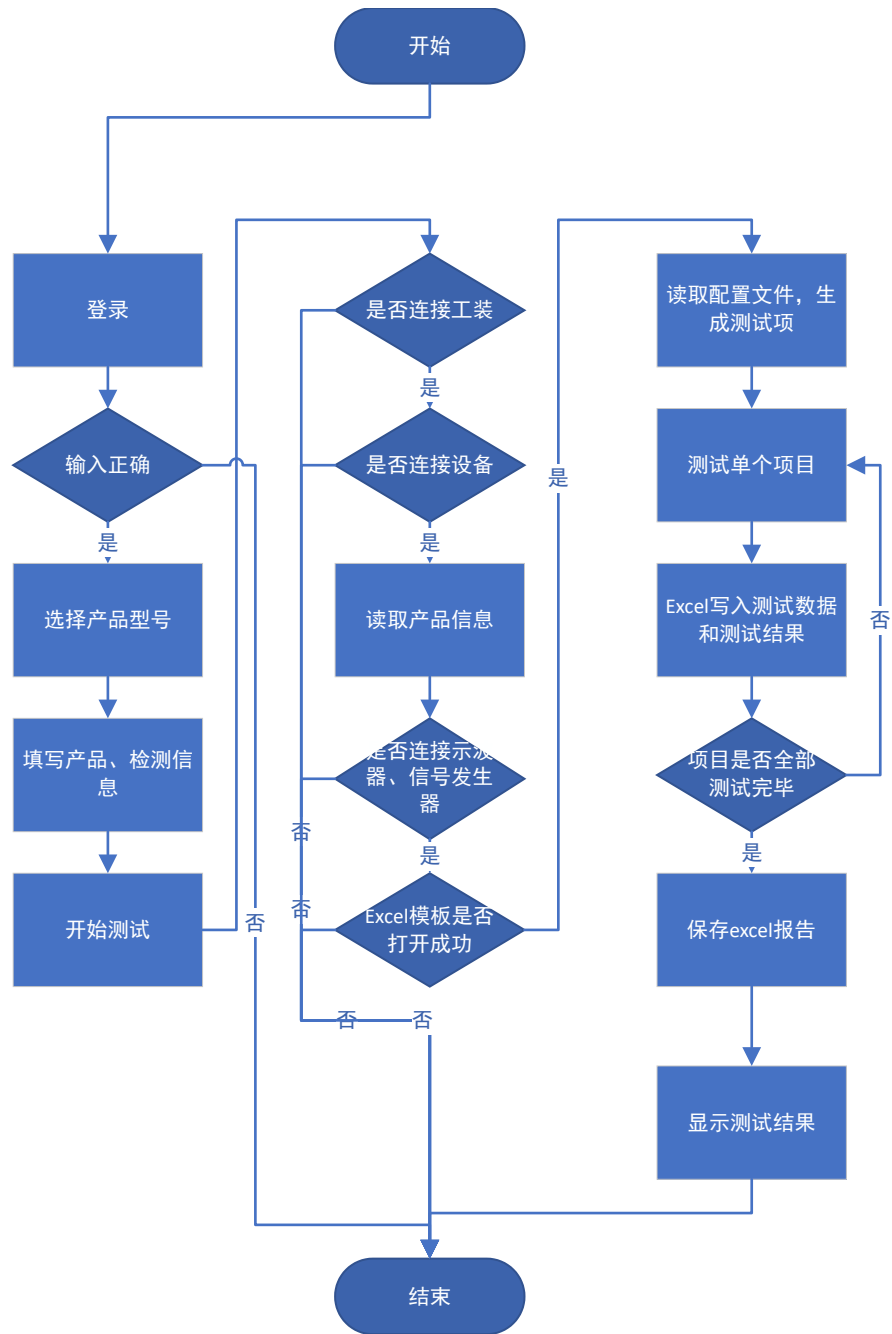
现有自动化测试工装软件代码因历史原因，导致结构相对较混乱，各产品共用同一套测试流程代码，耦合性太强，进行新产品添加时极其容易引入BUG；且软件部分使用技术较为过时。此次依托便携式小睿手自动化测试需求，对现有自动化测试工装软件进行重构。

2 需求分析

2.1 原始需求描述

原始需求来自对现有自动化工装测试软件分析。

- 登录：软件启动时展示登陆界面，用户可选择注册，注册成功后选择工位（半成品、成品、QC）后进入软件主界面；登陆时可选择保留选择，下次启动软件时候将自动回显上次输入。
- 主界面：
 - 主界面左侧提供型号树形结构，对不同产品线的型号进行归类分组，双击型号会进入该型号的测试界面；
 - 界面右侧为表单，需要填写生产日期、检验日期、产品编号、是否自动打印报告、检验站点、自动化工装编号，表单在双击进入测试界面时会进行校验；
 - 右上角设置选项：设置中常规tab页提供参数提升、示波器型号选择、是否蓝牙自动连接、蓝牙适配器类型、序列号规则设置项目几个设置项；此外，还有预设测试参数、测试流程、产品型号3个tab页；
 - 设置中提供历史记录查看，提供按照日期、产品型号进行筛选，并提供记录导出。
- 测试界面：测试界面分为两部分
 - 左侧：左侧为实时信息展示界面，提供设备电量显示（仅QC）、测试时间计时、通道实时采集值折线图、电流输出值以及产品名称、型号、编号、检验日期、生产日期、出厂编号字段展示
 - 右侧：右侧提供测试项列表、提供实时测试进度以及测试结果，且提供测试流程控制、测试结果数值查看。
- 测试流程：测试过程中的数据保存到数据中，同时按照格式生成json格式，在测试完成上传MES系统；所需测试项及测试参数根据配置文件确定。测试流程参考下图：



2.2 整理需求描述

整理需求主要是对原始需求中，以当前来看需要改进的点。

- 登录：登录时选择工位可以放在主界面，避免切换工位时需要退出界面
- 主界面：设置选项中预设测试参数、测试流程、产品型号3个tab页实际使用过程中极少使用，当前版本考虑砍掉后续视情况而定。
- 测试流程：测试流程中需要在开始时就打开excel，每个测试项目完成需要写入，最后保存到本地；本次考虑保存到数据库，进行一次性导出，避免测试过程中excel写入异常打断测试流程

2.3 测试项描述

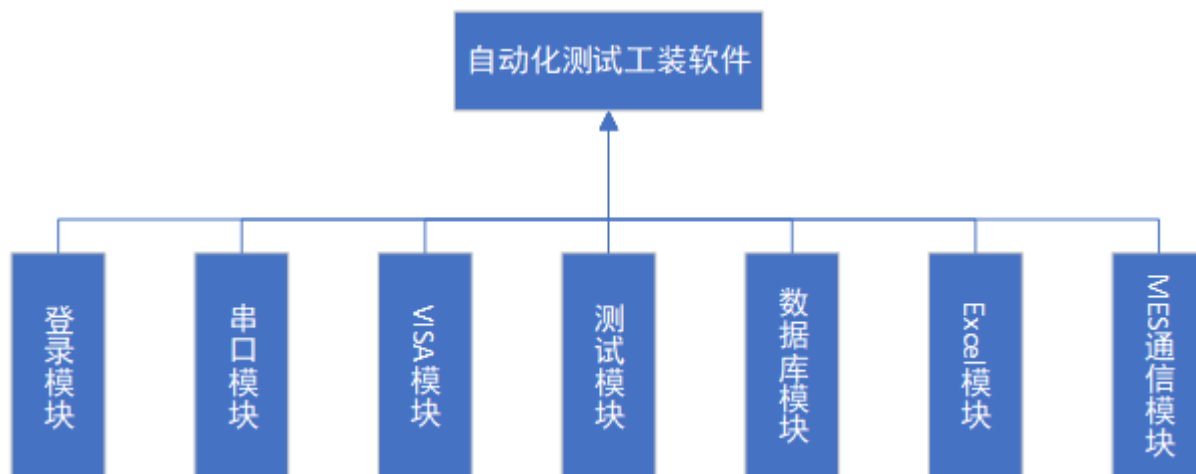
- 系统噪声：系统噪声意在测试不发射任何信号的情况下的设备干扰值。输入采集原始数据，将工装切到对应通道，每隔125ms读取一次采集数据的值，忽视前20个值（可能有波动），取第21~30个数值的平均值，记为系统噪声的结果值，输出结果。
- 工频陷波：按照要求将信号发生器调整到档位输出，将工装切到对应通道，每隔125ms读取一次采集数据的值，忽视前20个值（可能有波动），取第21~30个数值的平均值，记为工频陷波的结果值。
- 分辨率：此项目测试信号发射器发射两种不同信号时设备是否能够区别。按照要求使用信号发生器发射信号A，再发射与A差值较小的信号B，观察示值是否有变化。
- 示值准确度：根据配置控制信号发生器输出多个信号，判断采集值与标准值之间的误差是否符合结果要求。
- 反馈阈值准确度：
 - 信号发生器给预设值后，读取设备显示值P1（取10个值平均值），在上下调整信号发生器的幅值。例如：预设值1.2，读取值1.1，调整值0.1。；再次读取设备的显示值，每0.1上下调节，直至调节到范围在0.8~1.2之间，判定读取到有效值。
 - 2.若信号发生器给预设值后，设备上的显示值的 $0.5 < \text{显示值} > 3$ ，则直接判定不合格。
 - 分为3组参数
- 通频带：
 - 低频
 1. 将工装切到相应通道，信号发生器发送给定参数的信号；
 2. 设备上的显示值范围在68~72之间，信号发生器每0.1上下调节；
 3. 设备上的显示值范围不在68~72之间，每0.2上下调节。
 4. 最高调到18Hz，最低调到9.0Hz，调节范围内仍不满足要求，停止调节，判定不合格，切换下一个通道测试。
 5. 信号发生器给预设值后，设备上的显示值的 $40 < \text{显示值} > 90$ ，则直接判定不合格，切换下一个通道测试。
 - 高频
 1. 将工装切到相应通道，信号发生器发送给定参数的信号；
 2. 设备上的显示值范围在68~72之间，信号发生器每0.1上下调节；
 3. 设备上的显示值范围不在68~72之间，每0.2上下调节。
 4. 信号发生器，最高调到780Hz，最低调到540Hz，调节范围内仍不满足要求，停止调节，判定不合格，切换下一个通道测试。
 5. 信号发生器给预设值后，设备上的显示值的 $40 < \text{显示值} > 90$ ，则直接判定不合格
- 输出刺激强度：
 1. 将工装切到相应通道，令设备给定参数的输出电流；
 2. 记录示波器显示的电压，计算电流并记录；
 3. 依次测试三种电流并记录结果；
 4. 测试下个通道，直到全部通道测试完毕。
- 输出刺激频率：
 1. 将工装切到相应通道，令设备给定参数的输出电流；

2. 记录示波器显示的频率并计算误差和记录；
 3. 依次测试三种电流并记录结果；
 4. 测试下个通道，直到全部通道测试完毕。
- 输出脉冲宽度：
 1. 将工装切到相应通道，令设备给定参数的输出电流；
 2. 记录示波器显示的脉宽并计算误差和记录；
 3. 依次测试三种电流并记录结果；
 4. 测试下个通道，直到全部通道测试完毕

3 概要设计

3.1 软件功能模块

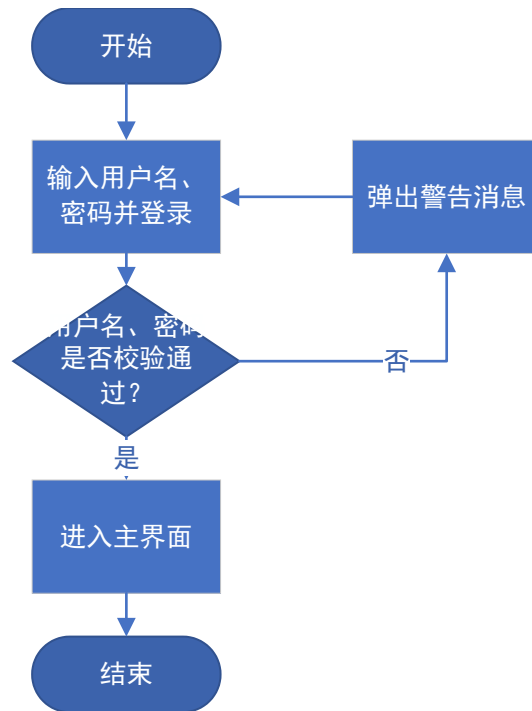
根据当前软件功能拆解，共整理出以下几个模块：



3.2 模块设计

3.2.1 登陆模块

登录模块流程图如下：

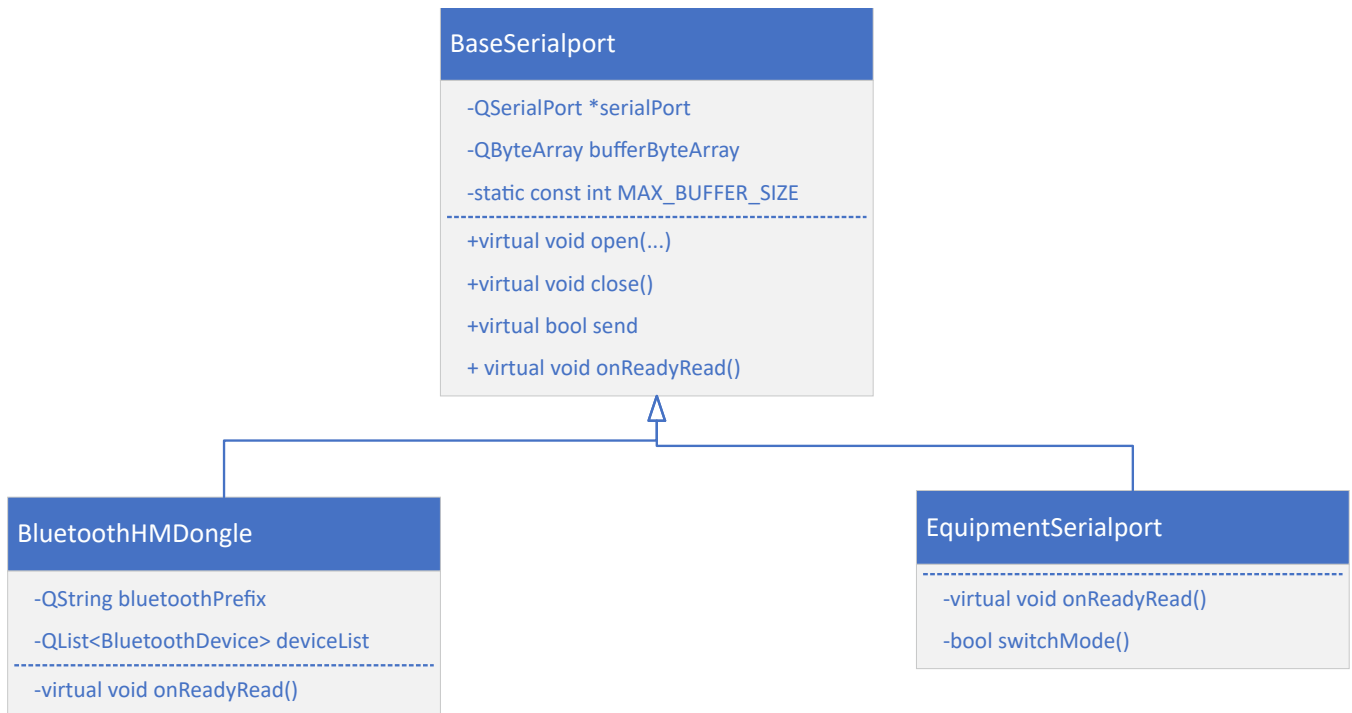


- 登录模块提供用户注册与登录功能，登录时选择工位来确定检测类型，后续根据检测类型确定测试流程（成品、半成品、QC），改成可在主界面中配置。
- 用户数据保存于数据库，登录时与数据库通信进行校验；后续测试流程中，测试数据中将会关联当前登录用户。
- 用户表字段结构如下：

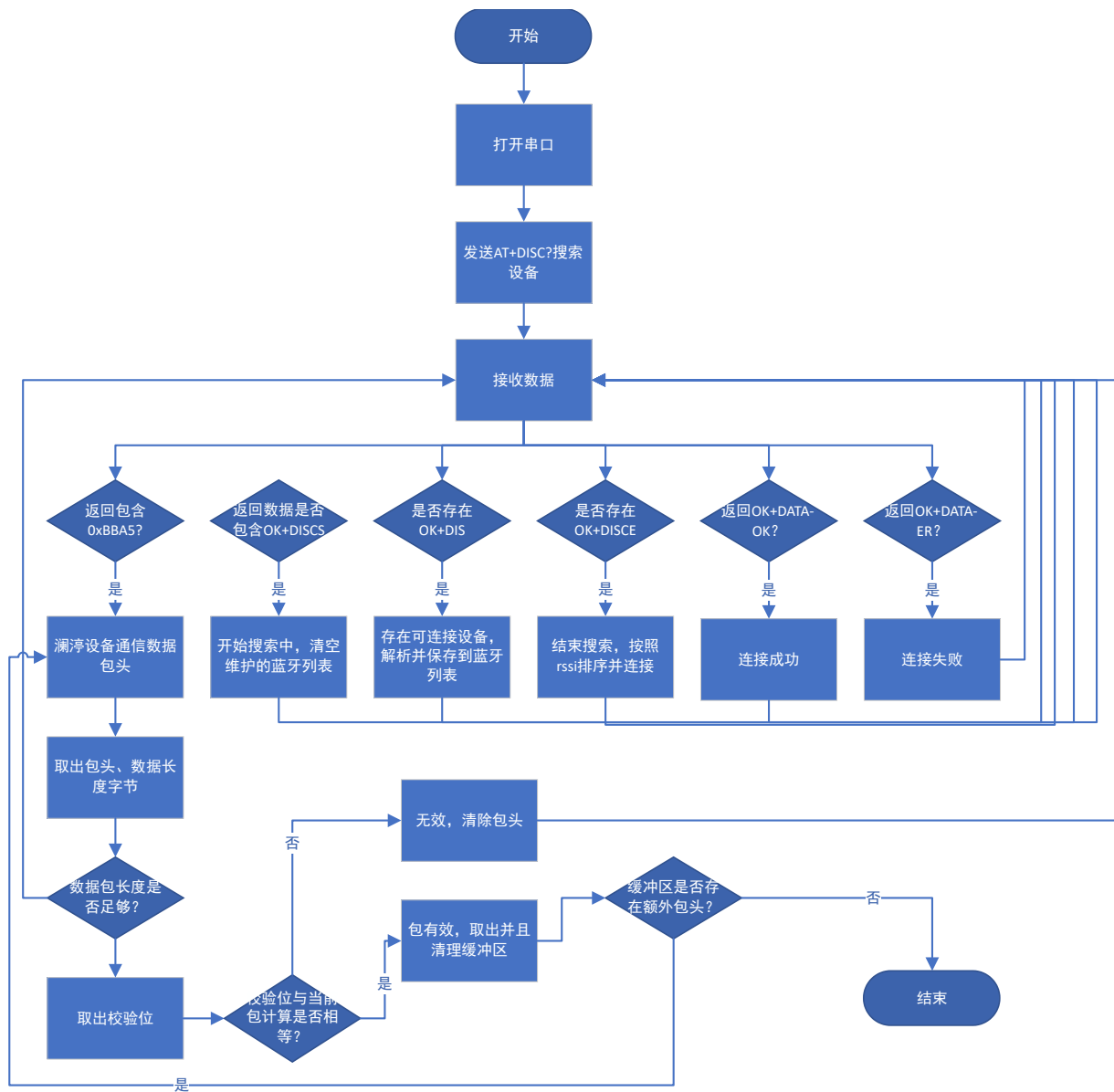
结构 数据 约束 索引 触发器 DDL									
AutoTestData 表名: User <input type="checkbox"/> WITHOUT ROWID <input type="checkbox"/> STRICT									
名称	数据类型	主键	外键	唯一	条件	非空	字符序	生成	
1 id	INTEGER								NULL
2 acoount	INTEGER								NULL
3 uname	TEXT								NULL
4 password	TEXT								NULL

3.2.2 串口模块

串口模块主要提供工装、蓝牙、气压计的串口通信功能，包含对应的通信协议。



- 串口模块实现基础串口类 `BaseSerialport`，内部维护一个串口对象 `serialPort` 和数据缓冲区，提供基础的打开、关闭、发送数据功能，还提供一个接收数据处理接口 `onReadyRead()`，该接口要求子类必须按照各自协议实现对应的数据处理功能，避免出现处理逻辑耦合。
- 当前**蓝牙**模块采用获取全部COM号，然后**遍历**的方式确定蓝牙串口，再根据**信号强度**进行设备连接；考虑适配器串口分配后一般不会改变，可在设置中根据增加【**蓝牙COM号**】设置项，后续使用直接打开COM，无需轮询。
- **气压计、工装**连接采用同样方式，将COM号写入**配置**。
- 蓝牙部分在连接设备后需要保持不间断处理数据，所以蓝牙串口需要放到**子线程**执行；工装与气压计串口无需保持长连接。
- 蓝牙部分通信协议解析流程图如下：

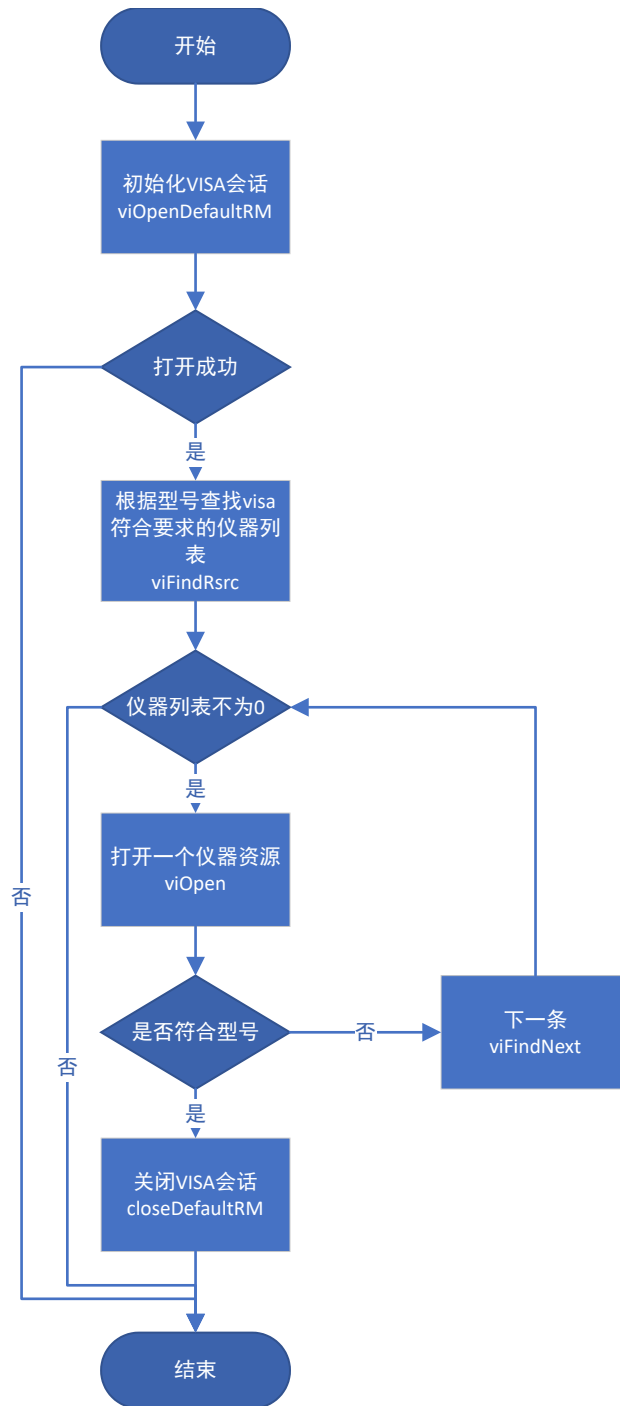


3.2.3 VISA模块

visa模块主要通过visa库实现与测量设备的通信。

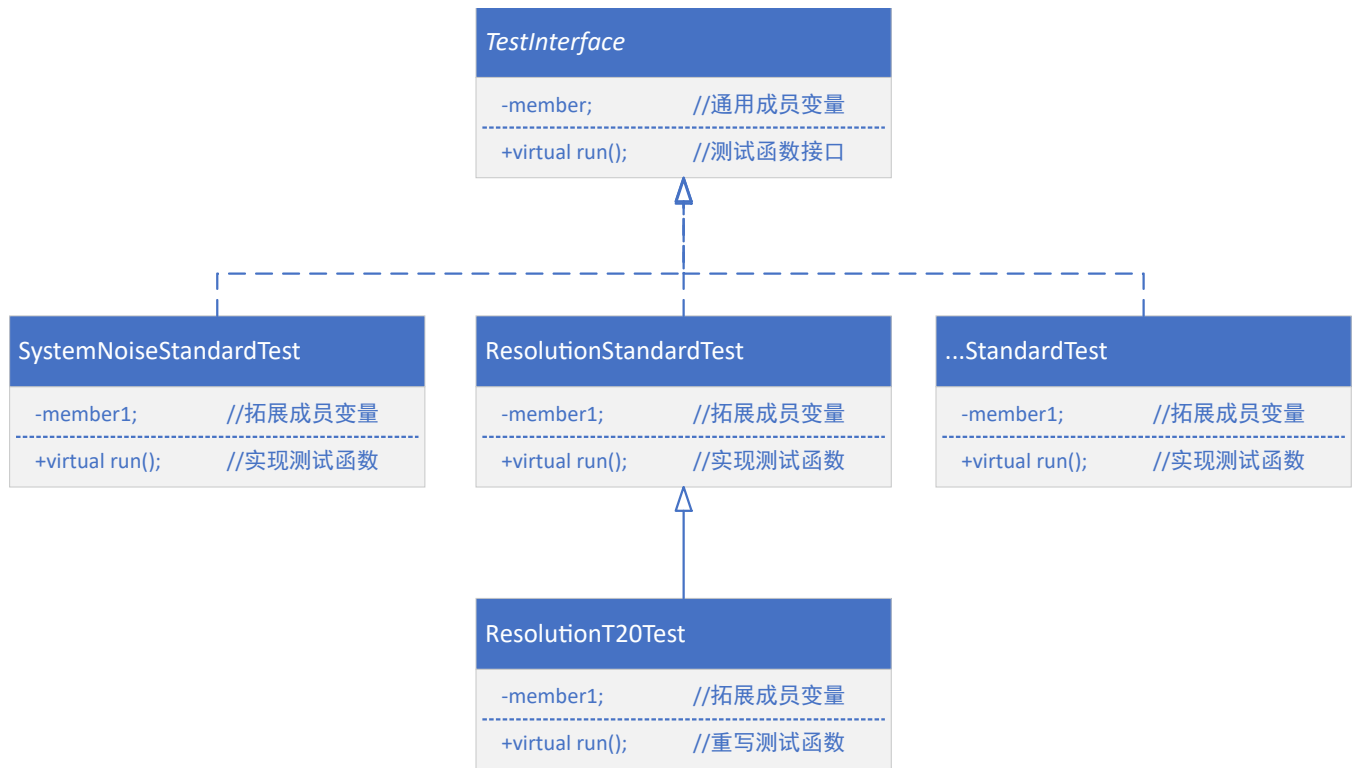


- 将visa库封装成通用基类，内部维护与仪器通信的会话，提供打开、关闭会话，打开、关闭、读写仪器的通用方法，各型号仪器继承通用类，无需关注与设备的通信，只需要根据协议实现各自的业务功能。
- visa通信流程图如下：



3.2.4 测试模块

- 主要功能模块，实现对产品的功能测试，通过配置中的测试项目，协调软、硬件资源进行流程控制；
- 当前测试流程为测试线程处理函数循环调用测试函数，测试函数与流程处理函数在同一个类中，耦合度强，拓展性差，流程处理繁琐；考虑拆分为测试线程与测试类；

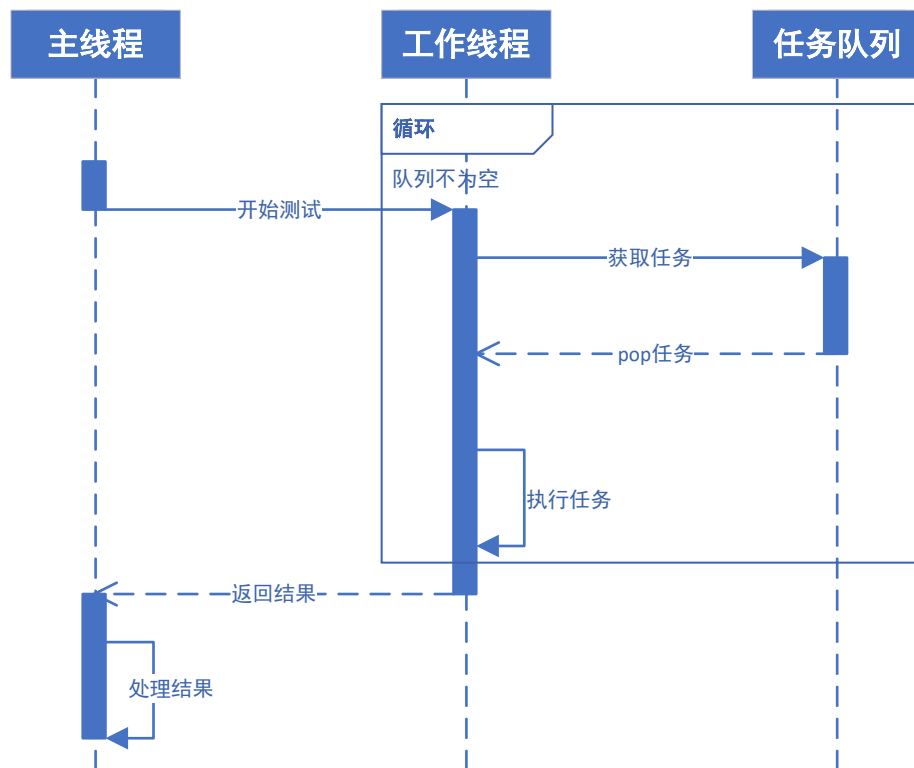


- **测试函数**结构如上图，抽象测试接口，所有标准测试类实现该接口；标准测试类不足以完成测试时，可以继承某个标准测试项类，重写测试函数在父类的基础上进行拓展；**测试结果**保存在**成员变量**中。

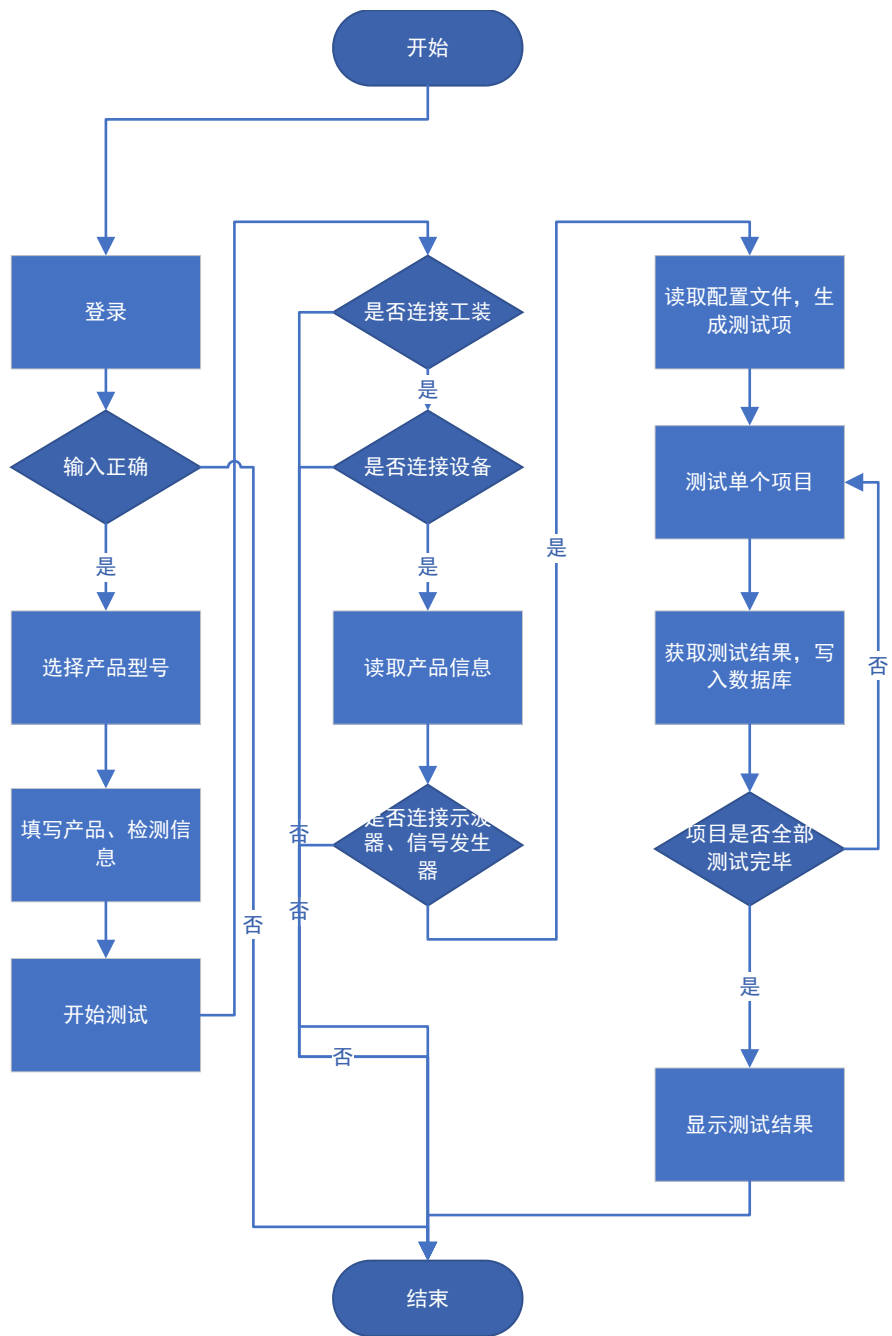
```

1  class Child
2  {
3      ...
4      void run() override
5      {
6          //定制内容
7          ...
8          Parent::run();
9          //定制内容
10         ...
11     }
12 };
  
```

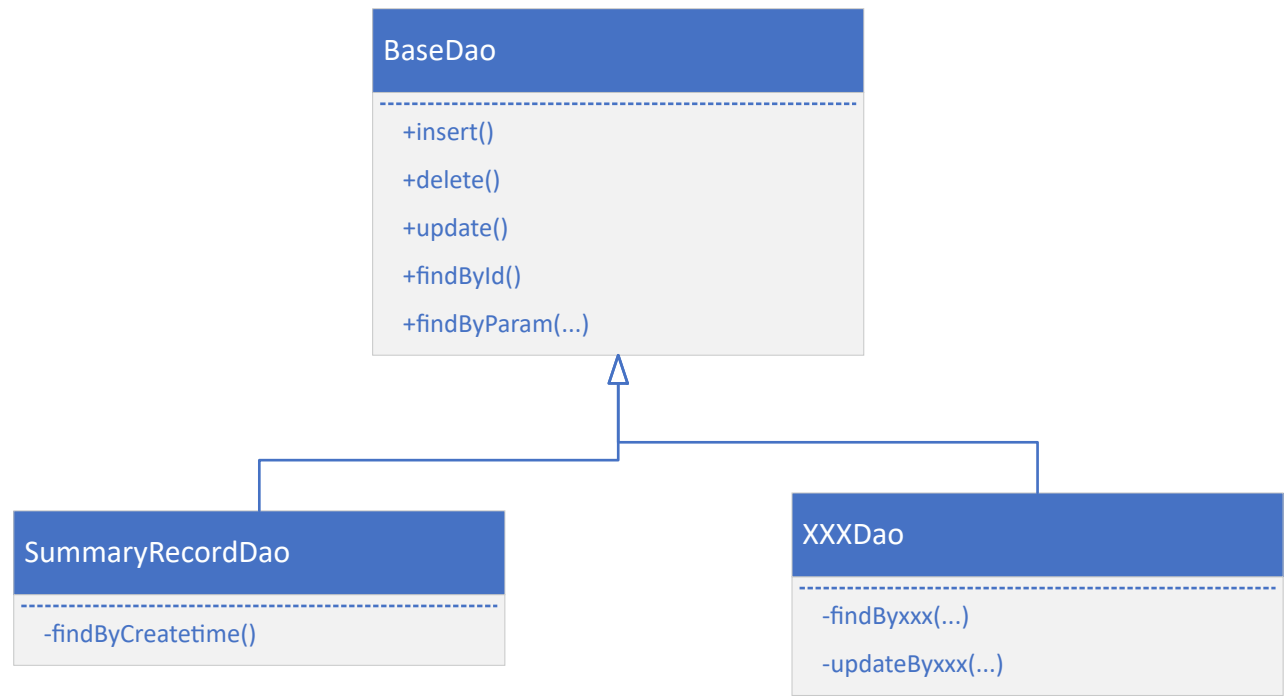
- **测试线程**分为两部分：**工作线程**、**任务队列**。初始化时将需要执行的测试类对象放入任务队列，后续启动工作线程从任务队列中不断取出测试项并执行，直到队列为空；



- 在并行测试下，由于可用仪器资源是唯一的，在某些测试项中会多次调用，所以在每次测试时生成测试项，保证测试项唯一并且在入口处增加互斥锁，保证仪器资源不会冲突。
- 每个测试项的结果都保存在测试项中，在单个测试项结束后从中取出并重置，同时进行处理保存到数据库，方便并行设备使用。
- 新测试流程对比当前，去掉了在测试过程中多次读写excel，测试流程如下：

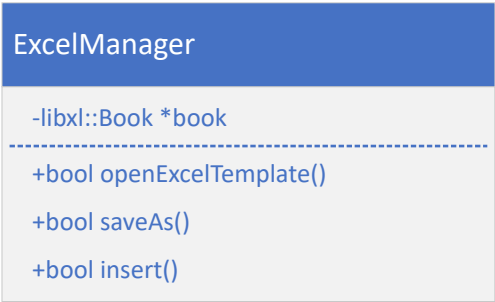


3.2.5 数据库模块



- 数据库模块基类提供基础的增删改查功能，子类继承后可以根据自身业务定制实现新的功能。
- 每个型号的测试数据对应一张数据表，每张数据表对应一个子类，方便业务层逻辑隔离。
- 各个型号之外存在一张汇总表，记录测试记录，关联其他型号的数据表，需要时通过连接语句查询。

3.2.6 Excel模块



- `ExcelManager` 类提供基础文档打开、保存、插入功能，需要时根据对应的模板填充数据，进行Excel文件导出。
- Excel模块使用第三方LibXL库，文档丰富，无依赖库，无需本地安装office。



LibXL

excel library for developers

[Home](#) [Index](#) [Book](#) [Sheet](#) [Format](#) [Font](#) [AutoFilter](#) [FilterColumn](#) [RichString](#) [FormControl](#)
[ConditionalFormatting](#) [ConditionalFormat](#) [Examples](#)

Factory functions

C++ ▼

- `Book* book = xlCreateBook();`
Creates a [book](#) instance for working with XLS files. This function should be called first for receiving a book pointer. This function and other classes are in the libxl namespace.
- `Book* book = xlCreateXMLBook();`
Creates a [book](#) instance for working with XLSX files. This function should be called first for receiving a book pointer. This function and other classes are in the libxl namespace.

Book class

- `bool load(const wchar_t* filename, const wchar_t* tempFile = 0)`
- `bool loadSheet(const wchar_t* filename, int sheetIndex, const wchar_t* tempFile = 0)`
- `bool loadPartially(const wchar_t* filename, int sheetIndex, int firstRow, int lastRow, const wchar_t* tempFile = 0)`
- `bool loadWithoutEmptyCells(const wchar_t* filename)`
- `bool loadInfo(const wchar_t* filename)`
- `bool save(const wchar_t* filename, bool useTempFile = false)`
- `bool loadRaw(const char* data, unsigned size, int sheetIndex = -1, int firstRow = -1, int lastRow`

HOME

DOWNLOAD

DOCUMENTATION

EXAMPLES

SETUP

FAQ

PURCHASE

CONTACT

[C++ Excel Library to read/write xls/xlsx files - LibXL](#)

3.2.7 MES通信模块

MES通信模块是按照格式将测试结果数据封装成json格式，通过HTTP将数据通过指定接口上传到MES系统。