

赛题4：基于 SQLite 的数据库插件开发

一、赛题背景和目标

随着软件系统对数据存储和管理需求的多样化，轻量级数据库 SQLite 因其零配置、高性能和跨平台特性，成为嵌入式系统各类科学计算应用的优选。北太天元科学计算与系统仿真软件计划增强数据库交互能力，实现结构化数据的高效读写与管理功能。

本赛题旨在设计一个符合北太天元插件规范的 SQLite 接口插件。该插件以动态库形式集成至主程序，实现数据库连接、查询、写入、事务控制等基础操作。通过本赛题，参赛队伍将综合展示对数据库接口设计、插件架构、数据交互和异常处理等方面的工程实现能力。

二、题目设置

本题共设置三个问题，分别聚焦于数据库连接与状态管理、数据查询与写入、事务控制与数据导入配置。

问题一：数据库连接与状态管理函数设计

请实现以下接口，确保插件能够与标准 SQLite3 数据库进行连接及关闭操作，并能正确维护连接状态：

- `conn = sqlite(dbfile, mode)`：建立或创建数据库连接。
- `close(conn)`：关闭连接并释放资源。
- `status = isopen(conn)`：返回当前连接是否有效。

参数及功能实现要求：

- `dbfile` SQLite数据库文件，支持绝对路径和相对路径输入。
- `mode` 可为自定义字符串（如 `'readonly'`、`'create'` 等），选手可自由设定其具体含义。
- `conn` 可设计为结构体或外部对象，应包含如 `Database`、`IsOpen`、`IsReadOnly` 等必要字段。
- `status` 可设计为布尔值或整数值，指示数据库是否仍处于打开状态。
- 函数实现应具备异常处理能力（如非法路径、无权限等）。

问题二：数据查询与写入操作接口设计

实现以下核心数据交互接口：

- `results = fetch(conn, sqlquery)`：执行任意 SELECT 查询并返回结果。

- `data = sqlread(conn, tablename]`) : 读取整张表。
- `sqlwrite(conn, tablename, data)` : 将北太天元表数据写入 SQLite 表。
- `execute(conn, sqlquery)` : 执行非 SELECT 类型的 SQL 语句 (如 INSERT、UPDATE、DELETE) 。

参数及功能实现要求:

- `conn` 可设计为结构体或外部对象, 应包含如 `Database`、`IsOpen`、`IsReadOnly` 等必要字段。
- `sqlquery` SQL 语句, 可设计为字符向量或字符串标量, 有关 SQL 查询语言的信息, 请参阅 SQL 教程。
- `tablename` 数据库表名, 可设计为字符串标量或字符向量。
- `results` 执行任意 SELECT 查询返回的结果。
- `data` 读取到的表数据。
- 结果 `data` 与 `results` 可采用北太天元平台内的 `table`、`struct` 或选手自定义的托管数据结构, 但要求各接口之间流转的数据采用统一的数据结构表示。
- 插件需处理 SQLite 到北太天元类型的自动转换, 并正确处理空值。类型转换对应关系, 可自行设计, 也可参考 matlab 的实现。
- 对 SQL 语法错误、空表等异常情形应有清晰反馈。
- 选手可为以上 4 个接口额外设计并自定义键值对参数名称及其功能, 形如 `results = fetch(conn, sqlquery, name, value)`, 例如控制是否清空表、设置事务模式、限制返回列等, **此项为加分项**。

问题三: 事务控制与数据导入配置接口设计

请实现以下事务控制与高级配置接口:

- `commit(conn)` : 提交事务。
- `rollback(conn)` : 回滚事务。
- `opts = databaseImportOptions(conn, source)` : 生成用于控制数据导入的选项对象 `opts` 。

参数及功能实现要求:

- `conn` 可设计为结构体或外部对象, 应包含如 `Database`、`IsOpen`、`IsReadOnly` 等必要字段。
- `source` 源, 可设计为字符向量或字符串标量。使用源输入参数指定数据库表的名称或用于从数据库导入数据的 SQL 查询。

- `opts` 控制数据导入的选项对象，可以采用北太天元平台内的 `struct` 或选手自定义的托管数据结构，包括以下字段（不限于）：
 - `ExcludeDuplicates`：是否排除重复项。
 - `VariableNames`：列名数组。
 - `VariableTypes`：数据类型列表。
 - `SelectedVariableNames`：列选择。
 - `FillValues`：缺失值默认填充。
 - `VariableNamingRule`：命名规则
- 插件应支持事务嵌套调用中的状态回滚与恢复。
- `databaseImportOptions` **为加分项**，选手可自行选择是否实现。

共性要求

- 插件应保证对数据库连接的稳定管理，避免资源泄露。
- 对异常场景（如连接失败、SQL 语法错误等）进行合理捕获和反馈。
- 对异常输入（如数据类型、数据维度等）进行合理拦截和报错，避免程序闪退。
- 采用北太天元插件规范进行接口设计，保证与主程序无缝集成。
- 函数参数与返回值类型应遵循既定约定，支持字符串、数值及结构化数据传递。
- 插件实现应尽量兼容多平台（Windows/Linux），**此项为加分项**。

三、技术规范

规范项	说明
编程语言	C/C++（推荐），允许适当使用 M 脚本封装接口层
SQLite 版本	SQLite 3.x，可使用官方提供的 C 语言接口
插件接口设计	统一接口函数声明，详情参考北太天元 SDK 开发文档
线程安全	插件设计需考虑多线程调用安全，保证数据库操作原子性
错误处理	返回详细错误码和错误信息，接口调用失败时不可崩溃
平台兼容	支持 Windows 或 Linux
代码规范	遵守北太天元团队代码规范，结构清晰，注释完善

四、提交材料

1. 完整源码（含编译脚本或工程文件）以及编译好的插件动态库。
2. 插件技术文档：包括但不限于如下内容

- a. 架构图与技术栈：包含插件的架构设计图，明确各模块的功能和关系；同时说明开发过程中使用的技术栈。
 - b. 插件的编译和构建方式：详细说明插件的编译环境要求、依赖项安装方法、编译步骤、以及构建过程中的注意事项。
 - c. 使用方式 / 入门教程：清晰介绍插件常见功能的使用步骤，帮助用户快速上手。
 - d. 用户接口文档：详细描述插件提供的各类接口，包括接口的名称、参数、返回值、功能说明及使用示例等。
3. 测试用例：需包含针对插件各项功能的测试用例源码。
 4. PPT 与讲解视频

附录：参考资料

1. 北太天元插件开发文档：北太天元安装目录/SDK/doc/baltamatica-sdk-manual.pdf
2. SQLite 官方网站与 C API 文档：
<https://www.sqlite.org/index.html>
<https://www.sqlite.org/c3ref/intro.html>
3. MATLAB SQLite 数据库文档：
https://ww2.mathworks.cn/help/database/matlab-interface-to-sqlite.html?s_tid=CRUX_topnav

赛题5：netCDF 插件开发

一、赛题背景和目标

在科学计算与工程模拟领域，netCDF（Network Common Data Form）作为一种支持多维数组结构的数据格式，被广泛应用于大气科学、海洋模拟、遥感数据等场景中。为进一步增强北太天元平台对科学数据的支持能力，本赛题旨在设计并实现一个通用的 netCDF 插件模块，使用户可直接在北太天元环境中读取、写入并提取 `.nc` 文件中的数据信息。

本赛题要求参赛队伍开发一个 C/C++ 实现的 netCDF 插件，具备良好的平台集成性、接口兼容性与使用便捷性。插件需支持以脚本函数方式完成常见的 netCDF 文件操作，同时兼顾性能优化与可维护性。

二、题目设置

请实现一个支持以下功能的 netCDF 插件模块，并将其注册至北太天元平台。

问题一：文件操作与管理接口设计

请实现以下基本文件操作函数，并确保用户可灵活使用这些接口打开、关闭及查询 `.nc` 文件的状态。

- `ncid = netcdf_open(source, mode, chunksize)`：打开或创建 netCDF 文件。
- `netcdf_close(ncid)`：关闭已打开的文件。
- `status = isopen(ncid)`：判断文件是否仍处于打开状态。

参数及功能实现要求：

- `source` NetCDF 文件路径，支持绝对路径和相对路径，格式为 `netcdf4` 文件。
- `mode` 访问类型，可为自定义字符串（如 `'WRITE'` - 读写访问、`'SHARE'` - 文件同步更新、`'NOWRITE'` - 只读访问等），选手可自由设定其具体含义。
- `chunksize`（可选开发）：块大小，用于性能调整的块大小参数，控制空间与时间之间的权衡。
- `ncid` NetCDF 文件句柄，选手可设计为整数、结构体或外部对象等合法数据类型，但必须可唯一标识文件。
- `status` 可设计为布尔值或整数值，指示文件是否仍处于打开状态。
- 提供异常处理机制（如路径不存在、权限问题等）。

问题二：变量数据的读取与写入接口设计

请实现以下读取和写入相关函数：

- `vardata = ncread(source, varname, start, count, stride)`：从netCDF文件中读取指定变量的数据。
- `ncwrite(source, varname, vardata, start, stride)`：向 netCDF 文件写入变量数据。
- `nccreate(source, varname)`：在 netCDF 文件中创建变量。

参数及功能实现要求：

- `source` NetCDF 文件路径，支持绝对路径和相对路径，格式为 `netcdf4` 文件。
- `varname` 变量名称，包含 netCDF 数据源中的变量名称。
- `start` 变量中数据的起始位置，长度为 N 的正整数向量，表示从每个维度的哪个位置开始读取，索引从 1 开始。默认从每维第一个元素开始读取。
- `count` 指定每个维度读取的数据量。选手可设计一个方式，支持用户通过向量形式灵活设定每个维度的读取长度，包括支持读取至某一维度末尾的能力。
- `stride`（可选开发，加分项）：指定在各维度上读写数据的步长，允许用户自定义数据访问的稀疏程度。若未指定，则默认为逐点连续访问。选手可制定详细的访问规则。
- `vardata`：变量数据，`varname` 的 netCDF 数据类型需与北太天元数据类型相匹配，数据类型的对应关系，可自行设计，也可参考matlab的实现。
- 支持读取/写入多维数据，起始位置、计数、步长参数应可选；
- 支持读取文件内任意变量、部分读取（start/count）；
- `nccreate` 函数可设计 Name, Value 参数方式进行接口扩展，形如 `nccreate(filename, varname, name, value)`，例如新变量的维度设置、指定北太天元数据类型、压缩参数等，**此项为加分项**。

问题三：访问文件元信息与属性接口设计

请实现以下函数接口，实现对 netCDF 文件元数据的访问与展示：

- `info = ncinfo(source)`：查询 netCDF 文件的元信息，如维度、变量、属性等；
- `ncdisp(source, location)`：树状结构展示文件结构；
- `attvalue = ncreadatt(source, location, attname)`：读取属性；
- `ncwriteatt(source, location, attname, attvalue)`：写入属性。

参数及功能实现要求：

- `source` NetCDF 文件路径，支持绝对路径和相对路径，格式为 `netcdf4` 文件。
- `info` 返回结果可为结构体或其他自定义格式，选手可以选择包含以下字段信息：Filename、Name、Dimensions、Variables、Attributes、Groups、Format、Datatypes。也可自行设计。

- `location` netCDF 数据源中变量或组的位置。
- `attname` 要读取的属性名称。
- `attvalue` 属性值，`attname` 的 netCDF 数据类型需与北太天元数据类型相匹配，数据类型的对应关系，可自行设计，也可参考 MATLAB 的实现。
- `ncdisp` 需能在终端显示层级结构，属性显示格式自定；
- 支持 netCDF4 的嵌套组结构与 UTF-8 属性名。

共性要求

- 支持大文件 (>2GB) 访问能力。
- 插件在连续打开关闭文件、读取多个变量等高频操作中应无内存爆炸或泄漏问题。
- 文件读取失败、变量不存在、维度越界等情况应能优雅处理异常。
- 采用北太天元插件规范进行接口设计，保证与主程序无缝集成。
- 函数参数与返回值类型应遵循既定约定，支持字符串、数值及结构化数据传递。
- 插件实现应尽量兼容多平台 (Windows/Linux) ，**此项为加分项**。

三、技术规范

规范项	说明
编程语言	C/C++ (推荐) ，允许适当使用 M 脚本封装接口层
插件接口设计	统一接口函数声明，详情参考北太天元 SDK 开发文档
线程安全	插件应支持线程安全调用，不阻塞主线程。
错误处理	返回详细错误码和错误信息，接口调用失败时不可崩溃
平台兼容	支持 Windows 或 Linux
代码规范	遵守北太天元团队代码规范，结构清晰，注释完善

四、提交材料

1. 完整源码 (含编译脚本或工程文件) 以及编译好的插件动态库。
2. 插件技术文档：包括但不限于如下内容
 - a. 架构图与技术栈：包含插件的架构设计图，明确各模块的功能和关系；同时说明开发过程中使用的技术栈。
 - b. 插件的编译和构建方式：详细说明插件的编译环境要求、依赖项安装方法、编译步骤、以及构建过程中的注意事项。
 - c. 使用方式 / 入门教程：清晰介绍插件常见功能的使用步骤，帮助用户快速上手。

- d. 用户接口文档：详细描述插件提供的各类接口，包括接口的名称、参数、返回值、功能说明及使用示例等。
3. 测试用例：需包含针对插件各项功能的测试用例源码。
4. PPT 与讲解视频

附录：参考资料

1. 北太天元插件开发文档：北太天元安装目录/SDK/doc/baltamatica-sdk-manual.pdf
2. NetCDF 官方文档：
<https://docs.unidata.ucar.edu/netcdf-c/current/index.html>
3. MATLAB 中 netcdf 接口文档（供接口设计参考）：
https://ww2.mathworks.cn/help/matlab/network-common-data-form.html?s_tid=CRUX_lftnav

赛题6：数据降维插件（t-SNE 算法）开发

一、赛题背景和目标

赛题背景

t-分布随机邻域嵌入（t-Distributed Stochastic Neighbor Embedding, t-SNE）是当前最流行的高维数据可视化技术之一，广泛应用于机器学习、生物信息学、图像处理等领域。本题目要求参赛队伍开发一个功能完整的基于t-SNE 算法的数据降维北太天元插件，通过C/C++调用开源库实现核心算法。

插件需要处理如下数据降维问题：

代码块

- 1 输入：高维数据矩阵 $X \in R^{n \times d}$ （ n 个样本， d 维特征）
- 2 输出：低维嵌入矩阵 $Y \in R^{n \times k}$ （通常 $k=2$ 或 3 ）
- 3 目标：保持局部邻域结构，使相似样本在低维空间中距离较近

赛题目标

参赛队伍需要基于成熟的开源 t-SNE 实现（如van der Maaten的C++代码），开发一个高性能的北太天元插件，支持精确算法和Barnes-Hut加速算法，具备良好的用户体验和接口兼容性。

二、题目设置

问题一：t-SNE 算法接入

请以插件的方式实现 `tsne_plugin` 函数，具体要求如下：

1. 插件框架开发：

- a. 实现标准的北太天元插件接口
- b. 函数签名：`[Y, loss, exitflag] = tsne_plugin(X, varargin)`
- c. 支持键值对参数输入：`tsne_plugin(X, 'Perplexity', 30, 'Algorithm', 'barneshut')`

`tsne_plugin` 调用的参数说明如下：

输入参数：

X - 需要进行降维的高维数据

varargin - 可变参数列表

采用'参数名', 参数值的配对形式输入, 支持以下参数:

'Algorithm' - 算法选择, 默认值: 'barneshut', 可选值: 'exact': 精确t-SNE算法; 'barneshut': Barnes-Hut近似算法

'Perplexity' - 困惑度, 默认值: 30, 取值范围: $5 \leq \text{Perplexity} \leq \min(50, n-1)$

'NumDimensions' - 输出维度。默认值: 2, 取值范围: 2或3

'MaxIter' - 最大迭代次数, 默认值: 1000, 取值范围: $100 \leq \text{MaxIter} \leq 5000$

'LearnRate' - 学习率 (可选)

'Exaggeration' - 早期放大系数 (可选)

'Theta' - Barnes-Hut精度参数 (可选)

输出参数:

Y - 低维嵌入结果, 输入数据在低维空间中的嵌入坐标

loss - 最终损失值, 最终的KL散度值, 衡量低维表示与高维数据的拟合程度

exitflag - 退出状态标志, 算法退出的原因和状态

取值含义:

- **1**: 正常收敛, 梯度范数小于容差
- **0**: 达到最大迭代次数但未收敛
- **-1**: 输入数据错误或参数无效
- **-2**: 内存不足或计算错误
- **-3**: 数值计算异常 (如出现NaN/Inf)

2. 开源库集成:

- 选择并集成高质量的t-SNE开源实现
- 将C++代码封装为北太天元可调用的插件形式
- 处理内存管理和数据类型转换

3. 核心算法支持:

- 精确t-SNE算法 (适用于小规模数据 ≤ 1000 样本)
- Barnes-Hut近似算法 (适用于大规模数据)
- 算法自动选择机制或用户指定选择

4. 基础参数支持:

- `Perplexity`: 困惑度控制 (默认30)

- NumDimensions : 输出维度, 2或3 (默认2)
- Algorithm : 'exact'或'barneshut' (默认'barneshut'), 以及自动选择机制
- MaxIter : 最大迭代数 (默认1000)

问题二: 插件高级功能

参赛队伍可根据时间和技术能力自由选择实现以下功能, 作为加分项:

1. 距离度量扩展

欧几里得距离: 默认的标准距离度量, 适用于连续数值特征且各特征量纲相似的数据, 具有直观的几何意义。

余弦距离: 专门适用于高维稀疏数据 (如文本TF-IDF特征、图像HOG特征), 通过计算向量夹角来衡量相似性, 不受向量长度影响。

马哈拉诺比斯距离: 考虑特征间相关性的高级距离度量, 能够自动处理特征缩放问题, 特别适用于金融数据、传感器数据等存在特征相关性的场景。

切比雪夫距离: 基于 L_∞ 范数的距离度量, 关注各维度的最大差异, 适用于需要考虑最坏情况的应用场景。

2. 数据预处理功能

数据标准化: 提供Z-score和Min-Max两种标准化方式, 自动消除不同特征间的量纲和尺度差异, 确保所有特征对距离计算的贡献均衡。

PCA预降维: 在t-SNE计算前先使用主成分分析降维至30-50维, 既能保留主要信息又可显著减少计算复杂度, 特别适用于超高维数据 (如图像、基因数据)。

NaN值处理: 智能检测和处理缺失值, 根据NaN比例自动选择删除样本、均值插补或K近邻插补等策略, 确保算法能够稳定运行。

数据验证: 自动检查数据的维度、数值类型、方差特性和重复样本等, 在预处理阶段就发现和解决潜在的数据质量问题。

3. 其他优化方向还有: 用户体验优化、性能优化、错误处理等

问题三: 完成插件在标准测试集的测试

插件开发完成后, 需要使用如下的标准数据集 (文件在附件) 进行测试, 测试结果需提交。

测试集1: Fisher鸢尾花数据集

数据规模: 150样本 \times 4维特征 **测试参数:** 默认参数 **性能基准:**

- 精确算法: ≤ 200 次迭代, 运行时间 ≤ 5 秒
- 最终损失值: ≤ 0.5
- 聚类分离度: 3个类别应基本可分

测试集2: MNIST数字子集

数据规模：2000样本 × 784维特征（每类200个样本） **测试参数：**Perplexity=50, Algorithm='barneshut' **性能基准：**

- Barnes-Hut算法：≤1000次迭代，运行时间≤60秒
- 最终损失值：≤1.2
- 聚类效果：10个数字类别应有明显聚类结构

测试集3：高维高斯聚类数据

数据规模：5000样本 × 100维特征（5个高斯聚类） **测试参数：**Perplexity=30, Algorithm='barneshut' **性能基准：**

- Barnes-Hut算法：≤1000次迭代，运行时间≤120秒
- 最终损失值：≤0.8
- 聚类保持度：5个原始聚类应在2D空间中保持分离

三、技术规范

1. 开发语言：主要使用与所选开源 C/C++ 库相适配的编程语言，同时需结合北太天元插件开发要求选用合适的开发语言。
2. 开发工具：无强制要求，参赛选手可根据所选开源库和北太天元插件开发需求选择合适的开发工具。
3. 兼容性：插件需兼容北太天元 4.0 及以上版本，且需兼容 Linux 或 Windows 平台。
4. 安全性：保证插件在调用开源库功能及与北太天元进行数据交互过程中的数据安全性，防止数据泄露或被篡改。
5. 第三方库使用：插件开发过程中如涉及使用其他第三方库，请确保该库以及其源码能从公开渠道获取。
6. 性能基准：在标准测试集上收敛，损失值满足基准要求。
7. 稳定性要求：3个测试集全部通过。

四、提交材料

1. 完整源码（含编译脚本或工程文件）以及编译好的插件动态库。
2. 插件技术文档：包括但不限于如下内容
 - a. 架构图与技术栈：包含插件的架构设计图，明确各模块的功能和关系；同时说明开发过程中使用的技术栈。
 - b. 插件的编译和构建方式：详细说明插件的编译环境要求、依赖项安装方法、编译步骤、以及构建过程中的注意事项。
 - c. 使用方式 / 入门教程：清晰介绍插件常见功能的使用步骤，帮助用户快速上手。

- d. 用户接口文档：详细描述插件提供的各类接口，包括接口的名称、参数、返回值、功能说明及使用示例等。
 - e. 标准测试集测试结果：需包含针对标准测试集的测试结果。
3. 测试用例：需包含针对插件各项功能的测试用例源码。
 4. PPT 与讲解视频

推荐的开源库

- van der Maaten的原始实现：<https://github.com/lvdmaaten/bhtsne>
- 现代C++实现：<https://github.com/LTLA/qdtsne>
- GPU加速版本：https://github.com/georgedimitriadis/t_sne_bhcuda

数据集附件

<https://baltamatica.pku.edu.cn/cszl/zlxz/29451493f594463abcaad0ec54b306921.htm#/>

赛题7：自选开源库插件开发

一、赛题背景与目标

北太天元作为一款在工程计算、科学研究等领域具有广泛应用的软件，其功能的丰富性和扩展性对于满足工业界多样化的需求至关重要。当前，工业界存在大量功能强大、应用广泛的开源 C/C++ 库，这些库在数值计算、信号处理、图形图像等诸多方面具备独特优势。将合适的开源 C/C++ 库接入北太天元，能够有效拓展北太天元的功能边界，提升其在工业场景中的适用性和竞争力，更好地服务于工程实践和科学研究。

二、题目设置

参赛选手需自行选择一款开源的 C/C++ 库，基于北太天元软件进行插件开发，实现该开源库与北太天元的有效对接。最终开发的插件应能使北太天元软件可直接调用该开源库的核心功能，从而拓展北太天元在相关领域的应用能力，为工业界相关问题的解决提供更便捷、高效的工具支持。

1. 开源库选择

参赛者自主选择的库应具有一定的工业应用价值，在其所属领域具有代表性或独特优势，且需确保能通过公开渠道获取，在提交材料中明确提供该开源库的下载地址及获取方式说明。建议所选开源库的许可证为宽松型开源许可证（如 MIT、BSD 等），且不得包含与赛事规则或北太天元软件使用相冲突的条款。

2. 插件开发内容

严格遵循北太天元的插件接口规范与开发框架进行开发，需通过实现插件函数的方式实现北太天元 and 开源库的调用，且所实现的功能需完备，能够覆盖开源库的核心功能点。同时确保插件能够与北太天元软件稳定兼容，重点关注开源库与北太天元环境的数据交互效率，包括内存共享、数据格式转换等方面，保证插件调用的便捷性和稳定性。

3. 开发工作量

需自行对标其余 B 类赛题的要求，确保本次开发的工作量（插件函数个数和用法个数）与之相当，从开发难度、功能实现范围、技术复杂度等方面进行合理把控。

三、技术要求

- 开发语言：**主要使用与所选开源 C/C++ 库相适配的编程语言，同时需结合北太天元插件开发要求选用合适的开发语言。
- 开发工具：**无强制要求，参赛选手可根据所选开源库和北太天元插件开发需求选择合适的开发工具。

3. 兼容性：插件需兼容北太天元 4.0 及以上版本，且需兼容 Linux 或 Windows 平台。多平台支持为加分项。
4. 安全性：保证插件在调用开源库功能及与北太天元进行数据交互过程中的数据安全性，防止数据泄露或被篡改。
5. 第三方库使用：插件开发过程中如涉及使用其他第三方库，请确保该库以及其源码能从公开渠道获取。

四、提交材料

1. 完整源码（含编译脚本或工程文件）以及编译好的插件动态库。
2. 插件技术文档：包括但不限于如下内容
 - a. 插件的开发背景与接入意义：详细介绍所选开源库的特点、优势，阐述选择该库的原因；分析将该开源库接入北太天元对软件功能拓展的具体价值，同时阐述该插件在工业场景中的应用前景。
 - b. 架构图与技术栈：包含插件的架构设计图，明确各模块的功能和关系；同时说明开发过程中使用的技术栈。
 - c. 插件的编译和构建方式：详细说明插件的编译环境要求、依赖项安装方法、编译步骤、以及构建过程中的注意事项。
 - d. 使用方式 / 入门教程：清晰介绍插件常见功能的使用步骤，帮助用户快速上手。
 - e. 用户接口文档：详细描述插件提供的各类接口，包括接口的名称、参数、返回值、功能说明及使用示例等。
3. 测试用例：需包含针对插件各项功能的测试用例源码。
4. PPT 与讲解视频