

LNS-3000 智能巡检系统 V5.0
V.5.0.6-V5.0.7 升级部署说明手册

国网智能科技股份有限公司

1. 手册简介

本文档主要用于 5.0.6 后台升级到 5.0.7 后台，介绍如何进行升级。其它各模块的配置说明、功能介绍，请查看对应部分的详细文档。请注意，各模块需要配套使用。

2. 升级前准备工作

2.1. 原后台备份

停止 5.0.6 相关运行服务，包括 SubstationServer（后台服务）、OfflineClient（任务管理单元）、Tomcat（Web）、MySQL（数据库）等。

将 5.0.6 的数据库、后台、Web 等相关文件资源备份到硬盘其它位置。

确保原后台电子地图文件（.map 或.mtd）存在，并进行备份。

2.2. 版本检查

先进行原后台版本的检查，可通过原后台 SubStationServer 左上角“数据库版本”查看，或通过数据库工具 Navicat 查看（查看方式可参考附录 1）。

如原后台版本为 3.0 后台，需要根据原升级文档《老站升级手册》，进行 5.0 融合升级。

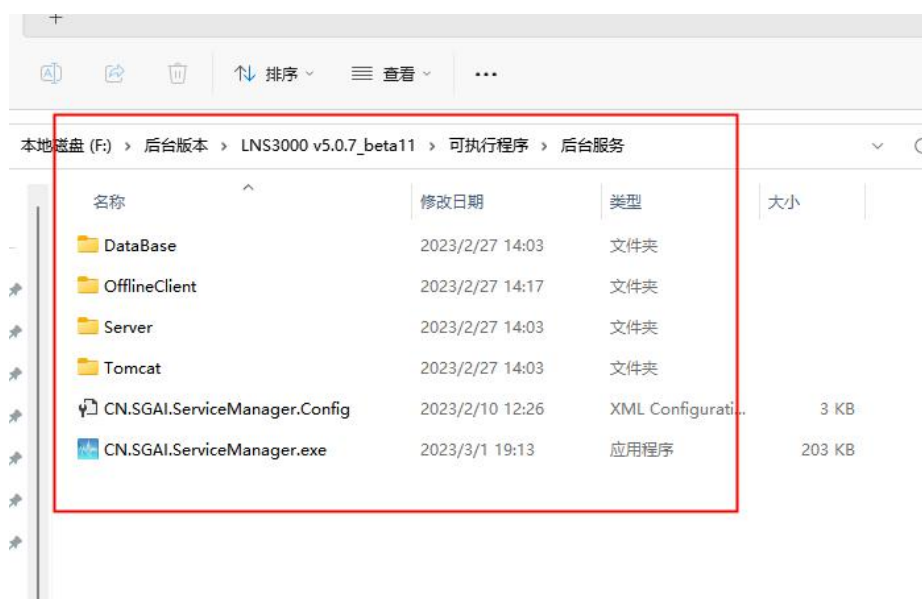
如原后台为 5.0 后台，且数据库版本小于等于 87，可直接应用本升级部署进行升级。

如原后台为 5.0 后台，且数据库版本大于 87，需要先对数据库版本进行降级处理后（降级到 87 数据库版本，降级方式见附录 1），再按此升级部署进行升级。

3. 升级过程

3.1. 版本介绍

5.0.7 版本相较于 5.0.6 版本做了较大的修改，具体体现在可执行程序——后台服务模块。5.0.7 版本将后台电脑上运行的程序整体打包，发版包中的后台服务文件夹内集成了数据库（DataBase）、任务管理单元（OfflineClient）、Sever（后台）、Tomcat（Web 服务），并新增服务管理模块（CN.SGAI.ServiceManager.exe 程序），如下图所示：



其中，服务管理模块 CN.SGAI.ServiceManager.exe 的作用是实现各程序的一键启动和进程守护，当所有模块部署完成后，可通过服务管理模块 CN.SGAI.ServiceManager.exe 对所有程序进行一键启动。服务管理模块的配置及使用介绍可参考《后台说明手册》第四章。

后台、Web 服务、任务管理单元的相应配置文件内 IP、端口均已默认配置，可直接使用，无需再进行 IP、端口配置。

3.2. 地图升级

因 5.0.7 支持多地图切换，在从老版本升级到 5.0.7 时，需要对老地图进行升级导入，请确保之前的地图文件（.mtd）文件可用，并将系统 C 盘的 MapCfg 文件夹备份后，删除该文件夹。

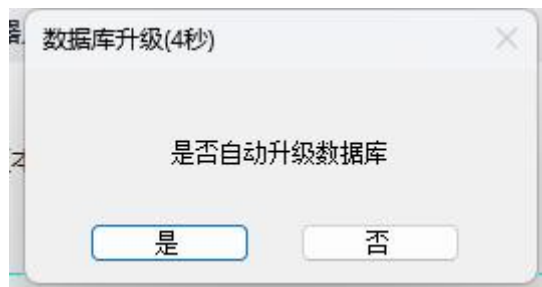
按《后台说明手册》中第五章电子地图工具使用说明，进行地图编辑维护。如无修改内容，可直接进行配置文件生成，填写必要信息后，确认即可。如存在多张地图，可在地图工具中进行新增、打开、生成配置文件。注意：请确保每张地图上有 1 到多个机器人，Web 会根据地图与机器人的关联关系进行地图展示，后台会根据地图与机器人的关联关系将地图信息的下发到对应机器人。

3.3. 后台升级

版本里提供的数据库为空库，当进行 5.0.7 升级时，需要将备份的 5.0.6 数据库放置在同目录下，即将 5.0.6 系统的 DataBsae 替换发版包中后台服务内的 DataBase。注意：需使用已进行过版本检查和相应处理的数据库

先启动数据库服务 mysqld 程序，然后直接启动 “\后台服务\Server\bin” 目录下的

SubstationServer.exe，会提示进行数据库升级（如下图），点击“是”，直接升级即可。



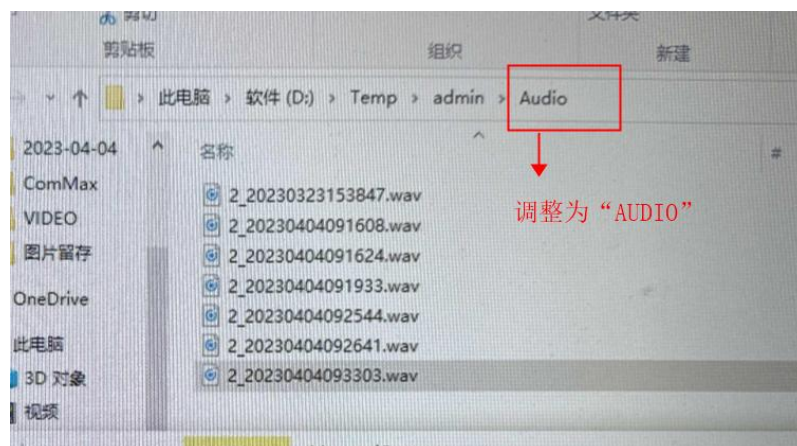
如升级过程出现问题，请准备好 ErrorLog 并联系研发处理。

注：完成数据库升级后，数据库内会自动将原 5.0.6 系统的巡检数据(包括巡检记录、识别结果、告警记录)另存到新的表中，并将相关表清空。如需要找回之前的巡检数据，请按附录 2 步骤操作。

完成以上操作后，点击 web 界面（巡检结果确认-设备告警确认/巡检历史数据/识别异常点查询/巡检结果浏览）可查询原系统的巡检数据是否存在。

如果 Web 端中图片或报告无法显示，可检查文件服务器配置，具体方式为：修改模型配置工具-系统-文件服务器配置内容，具体配置方法参见《巡检后台说明手册》6.2 部分。

如 Web 端手动录音无法播放，可检查文件服务器中临时文件目录下存放在录音文件夹名称是否全部为大写，如下图所示。



3.4. Web 与管理单元升级

修改任务管理单元配置文件（配置文件路径 OfflineClient\config\ClientConfig.json）内“AudioFilePath”、“XmlFilePath”路径，具体配置方法参见《任务管理单元说明手册》3 部分。



```
*ClientConfig.json - 记事本
文件(F)  编辑(E)  格式(O)  查看(V)  帮助(H)
//是否使用后台识别: (解释: 这里分后台识别和客户端识别, 下位机识别暂时归属后台识别类型中)
"IsConnectICE": "true",
// 后台ICE服务连接使用的协议
"ICEProtocol": "default",
// 后台ICE服务的连接地址
"ICECommunicatorIP": "127.0.0.1",
// 后台ICE的服务端口
"ICECommunicatorPort": "60000",
// 后台ICE连接的服务名称
"ICEServiceName": "OfflineSender",
// 本地ICE的身份名称
// 身份名称开头固定使用 "OfflineClient_" 字符串, 请参照下方进行设置
// 用于巡检机器人或轨道机器人 : OfflineClient_Robot_1
// 用于固定点机器人 : OfflineClient_Fix_1
// 用于顺序导航机器人 : OfflineClient_Fating_1
"LocalIdentityName": "OfflineClient_Robot_1",
// 下发音频文件文件夹
"AudioFilePath": "D:\\5.0.7-beta14\\Server\\Project\\AudioFile",
// 下发xml文件文件夹
// 配置文件confex.xml的存放地
"XmlFilePath": "D:\\5.0.7-beta14\\Server\\Project\\Ini",
// 下发路径文件文件夹, 联合巡检用
"RoadInfoPath": "D:\\Source5.0_Main\\Project\\Ini",
// 后台类型设定 请参照枚举全局常量【Constants.ConsolePGStyle】进行设定
// 3.0后台 : 1
// 标准化后台 : 2
// 隧道后台 : 3
// 容弗后台 : 4
// 联合巡检 : 5
"ConsolePGStyle": "2",
// 生成配置文件保存路径 //此处无效, 已修改为自动生成到exe文件同级目录
"CfgFilePath": "C:\\Cfg"
```

服务管理模块已经集成了 Web 与管理单元的启动, 如无特殊配置直接使用即可, 如需要修改配置, 参见《Web 部署手册》和《任务管理单元说明手册》。管理单元可根据现场需要按需部署, 例现场不存在阀厅机器人, 可不用启动阀厅任务管理单元。

3.5. 任务执行单元和下位机升级

3.5.1. 执行单元升级

固定点执行单元, 需要部署 Ubuntu1604 虚拟机, 请按照《执行单元说明手册》2.1 进行升级部署; 阀厅执行单元, 需要部署 Ubuntu1204 虚拟机, 请按照《执行单元说明手册》2.1 进行升级部署; 室内轨道机器人执行单元按工控机的系统环境进行相关程序替换, 按《执行单元说明手册》2.2.8 进行升级部署; 室外巡检机器人执行单元需要按工控机的系统环境进行相关程序替换, 请按照《执行单元说明手册》2.3.7 进行升级部署。

3.5.2. 91S 机器人工控机部署

该部分为 91S 机器人新部署工控机程序, 非新装 (或已部署过执行单元或下位机程序), 跳过此步骤。

91S 一键升级包安装方式参见《任务执行单元说明手册》。

3.5.3. 下位机升级

机器人下位机按工控机的系统环境进行相关程序替换, 请按照《下位机部署说明手册》2.2 进行升级部署。注意修改程序名称为 robot, 并将程序文件属性调整为可读写 (通过 FTP 工具, 将文件属性修改为 777)。

4. 配置检查

4.1. 任务启动方式

通过系统配置工具可配置任务启动的方式，任务启动方式分四种：立即启动、空闲启动、高级启动、优先级启动。

立即启动：新任务执行，旧任务终止；

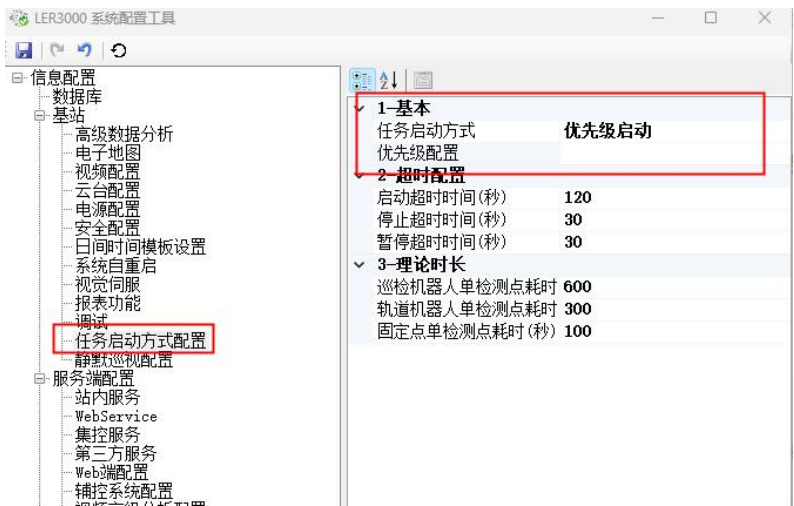
空闲启动：旧任务不终止，新任务不执行；

高级启动：根据优先级，新任务级别高，终止旧任务，开始新任务，否则新任务不执行；

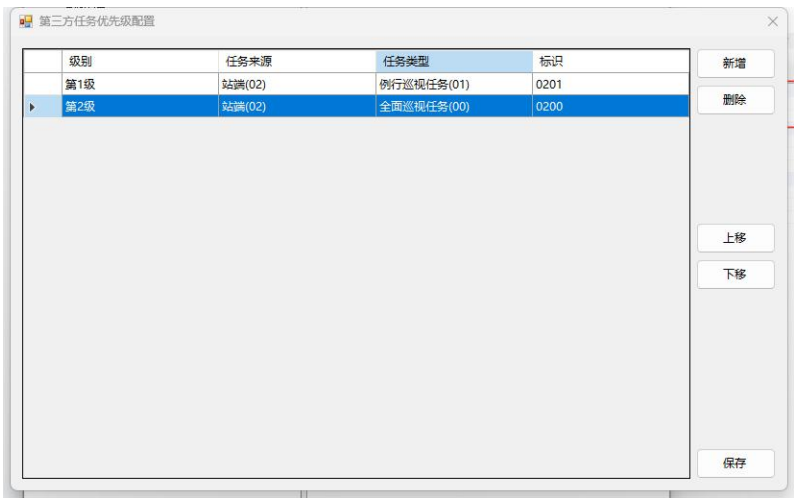
优先级启动：任务按优先级别进行任务队列的排序，支持高级别任务打断低级别任务，同级别按顺序执行。（一键返航、低电量返航优先级别最高）。

4.2. 任务优先级配置方式

通过系统配置工具中的任务启动方式进行配置，如下图：



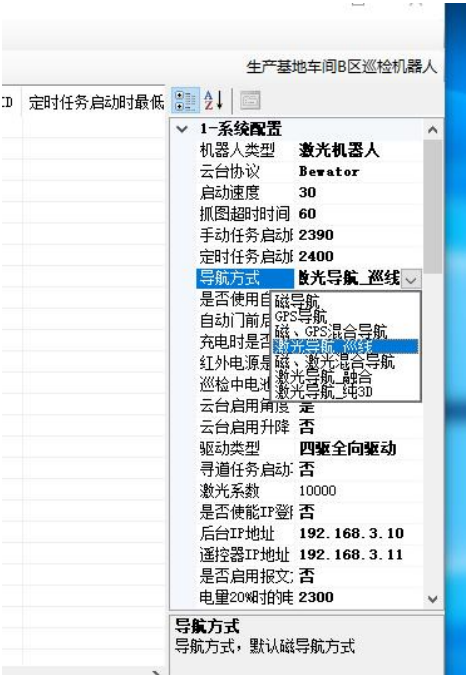
配置优先级，如下图：



可进行优先级的新增、删除、上移和下移，任务级别分三个层级：任务来源、任务类型、任务等级，该处可通过配置任务来源和任务类型的组合来确定任务优先级的高低，任务等级为任务的一个属性，目前不支持配置，默认为 Level9。

4.3. 机器人导航方式

V5.0.7 版本支持巡线导航、融合导航和纯 3D 导航三种方式，可根据机器人本体的导航方式进行切换，具体配置位置如下图：



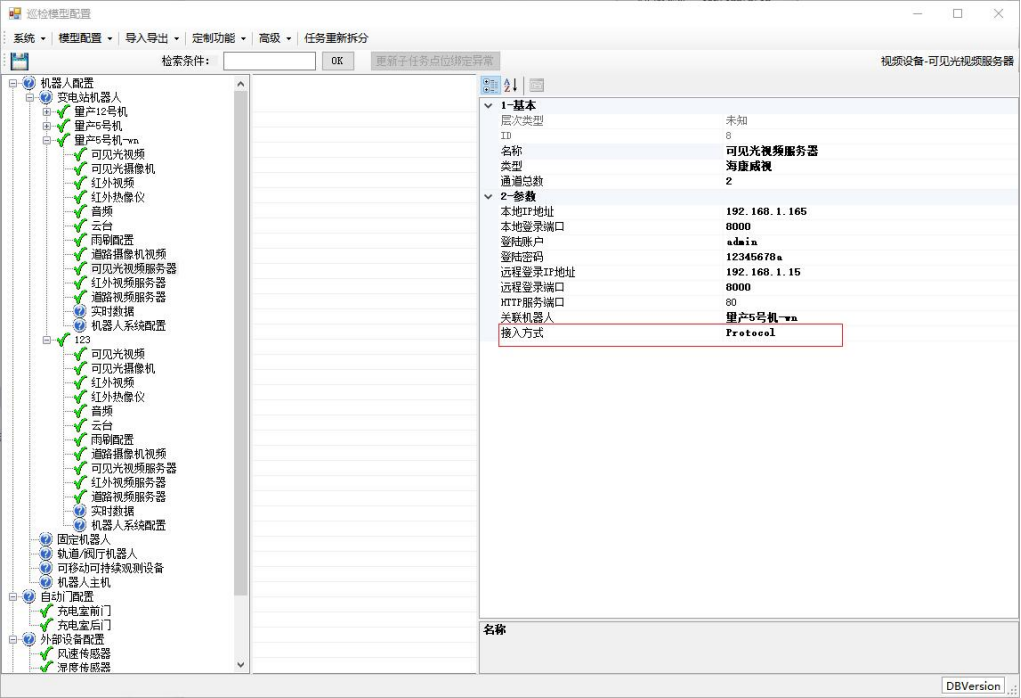
其中激光导航分为三种方式，一种为巡线导航，一种为融合导航，一种是纯 3D 导航，默认为巡线导航。

4.4. 相机控制方式

3200稳定测试机器人-可见光摄像头	
1-基本	
ID	3
设备名称	可见光摄像头
2-参数	
拉焦协议	MaiKang
相机地址	1
默认焦距	10
码流类型	子码流
控制设备	可见光视频服务器-192.168.3.26
控制设备通道号	4
拉焦的控制方式	函数控制控制方式
抓图模式	视频设备抓图
相机接入方式	SDK
厂家类型	海康
型号	SERVO_3007SY
伺服系数文件	ServoYunTaiCoefMap_Hikon.xml
视觉伺服抓图分辨率	D1
聚焦伺服抓图分辨率	1080P
识别抓图分辨率	1080P

新版本支持通过 SDK 和 HTTP 协议的方式控制相机,HTTP 协议的方式目前仅支持 3007NT 相机,如现场为 3007NT 现场,请调整相机接入方式为:Protocol(协议方式),伺服系数文件选择:ServoYunTaiCoefMap_Hikon_3007NT.xml。注:海康 3007SY 相机接入方式选择:SDK,伺服系数选择:ServoYunTaiCoefMap_Hikon.xml

另外,相机接入方式放在了可见光视频服务器里面,如下图:

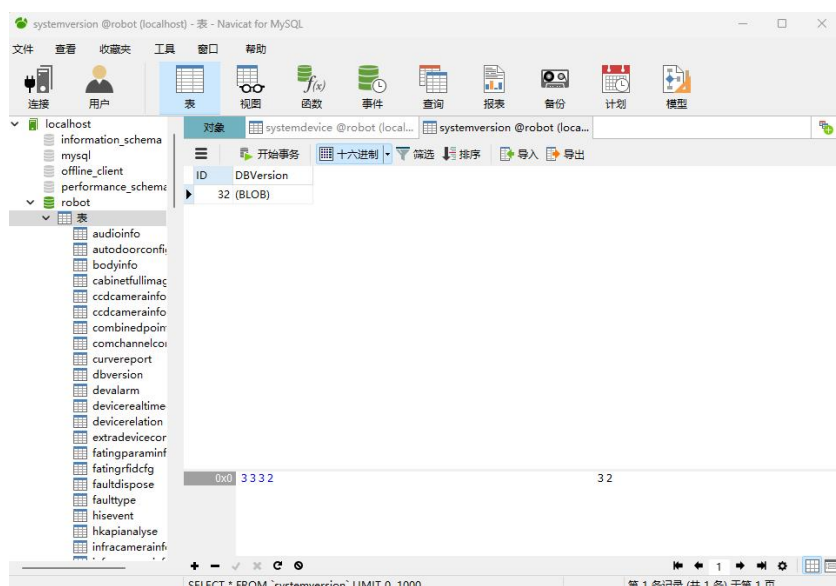


附录 1：5.0 后台数据库降级方法

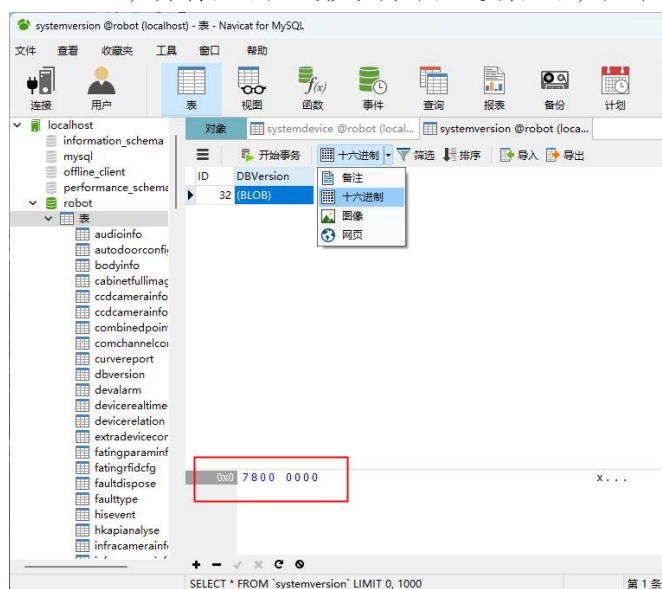
1.降级需要通过 Navicat 工具（Navicat 工具需使用发版包“部署工具中”的 Navicat for MySQL,解压后即可使用）实现。首先需要通过 Navicat 连接上对应数据库（数据库默认端口 3306，默认用户 robot，默认密码 LNINT-mysql2016）。

2.找到 robot 数据库，并打开。

3.找到表 systemversion，并双击打开，表中仅有一条记录，如下图：



4.选中该行的 DBVersion，并将显示方式修改为十六进制显示，如下图：



5.上图红框中的内容即为当前的数据库版本信息，为十六进制。如显示 7800 0000，则表示 0x78，即当前版本为 120。如果要降级到 87 版本，将 87 转换为 16 进制，即 0x57，将红框中的内容修改为“5700 0000”，注意输入位数，一定为 8 位。

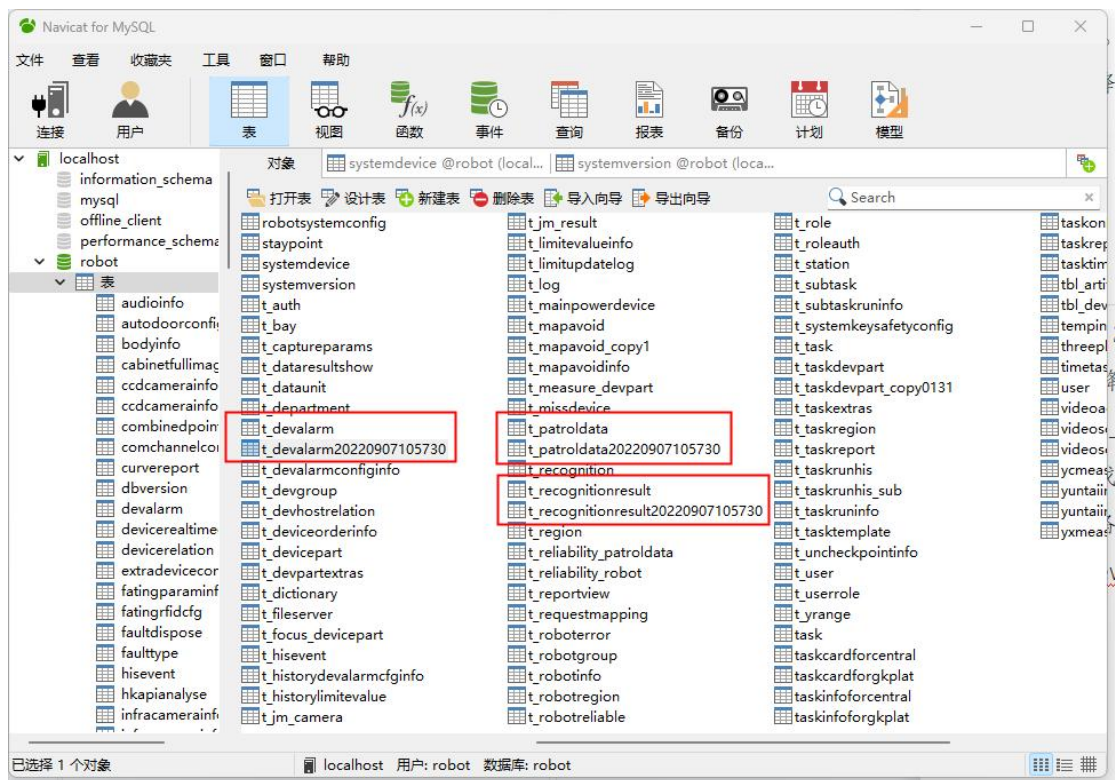
6.降级完成。

附录 2：升级后原巡检结果恢复方法

1.降级需要通过 Navicat 工具实现。首先需要通过 Navicat 连接上对应数据库。注意：Navicat 工具不要使用 12 版本，可选用部署工具中的 Navicat。

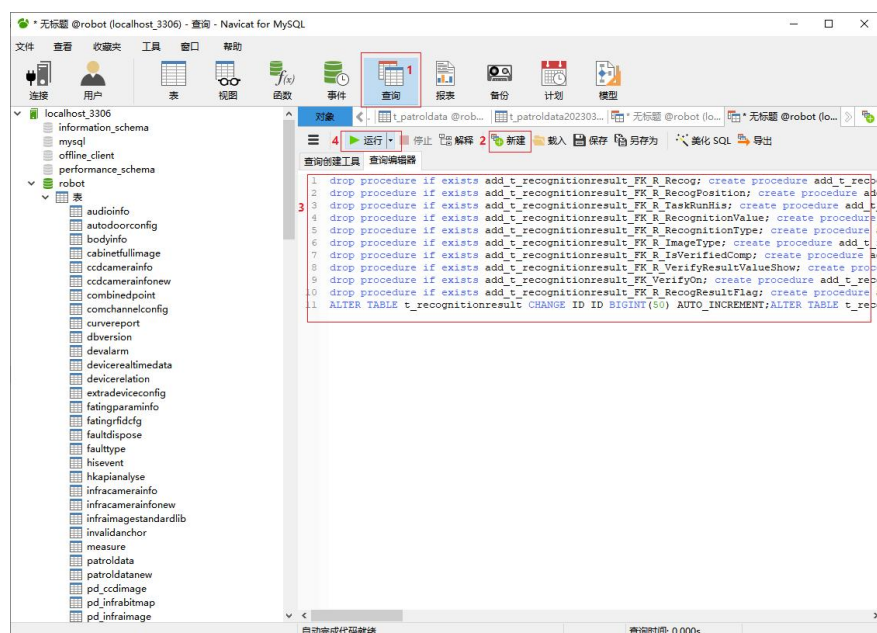
2.找到 robot 数据库，并打开。

3.与巡检数据相关的三个表为 t_patroldata、t_recognitionresult、t_devalarm，带后缀的表为原 5.0.6 数据库的巡检数据，后缀数字代表数据库升级完成的时间。如下图：



4.将新生成的 t_patroldata、t_recognitionresult、t_devalarm 三个表删除，将带后缀的三个表改名（去除后缀）。

5.通过 Navicat 工具，点击“查询”-“新建”，在新建的查询编辑器下输入以下 SQL 语句，点击“运行”。此步骤是对原表建立索引并进行表结构修改，建立索引速度由表内数据量决定。如下图：



SQL 语句如下（灰底部分）：

drop procedure if exists add_t_recognitionresult_FK_R_Recog; create procedure

```

add_t_recognitionresult_FK_R_Recog() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_Recog')THEN ALTER TABLE t_recognitionresult ADD INDEX FK_R_Recog(PatrolDataID);End IF;END;
call add_t_recognitionresult_FK_R_Recog();drop procedure if exists add_t_recognitionresult_FK_R_Recog;
drop procedure if exists add_t_recognitionresult_FK_R_RecogPosition; create procedure
add_t_recognitionresult_FK_R_RecogPosition() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_RecogPosition')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_RecogPosition(DevicePartID);End IF;END; call add_t_recognitionresult_FK_R_RecogPosition();drop
procedure if exists add_t_recognitionresult_FK_R_RecogPosition;
drop procedure if exists add_t_recognitionresult_FK_R_TaskRunHis; create procedure
add_t_recognitionresult_FK_R_TaskRunHis() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_TaskRunHis')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_TaskRunHis(TaskRunHisID);End IF;END; call add_t_recognitionresult_FK_R_TaskRunHis();drop procedure
if exists add_t_recognitionresult_FK_R_TaskRunHis;
drop procedure if exists add_t_recognitionresult_FK_R_RecognitionValue; create procedure
add_t_recognitionresult_FK_R_RecognitionValue() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_RecognitionValue')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_RecognitionValue(RecognitionValue);End IF;END; call
add_t_recognitionresult_FK_R_RecognitionValue();drop procedure if exists
add_t_recognitionresult_FK_R_RecognitionValue;
drop procedure if exists add_t_recognitionresult_FK_R_RecognitionType; create procedure
add_t_recognitionresult_FK_R_RecognitionType() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_RecognitionType')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_RecognitionType(RecognitionType);End IF;END; call add_t_recognitionresult_FK_R_RecognitionType();drop
procedure if exists add_t_recognitionresult_FK_R_RecognitionType;
drop procedure if exists add_t_recognitionresult_FK_R_ImageType; create procedure
add_t_recognitionresult_FK_R_ImageType() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_ImageType')THEN ALTER TABLE t_recognitionresult ADD INDEX FK_R_ImageType(ImageType);End
IF;END; call add_t_recognitionresult_FK_R_ImageType();drop procedure if exists
add_t_recognitionresult_FK_R_ImageType;
drop procedure if exists add_t_recognitionresult_FK_R_IsVerifiedComp; create procedure
add_t_recognitionresult_FK_R_IsVerifiedComp() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_IsVerifiedComp')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_IsVerifiedComp(RecognitionTime,IsVerified,ValueShow);End IF;END; call
add_t_recognitionresult_FK_R_IsVerifiedComp();drop procedure if exists
add_t_recognitionresult_FK_R_IsVerifiedComp;
drop procedure if exists add_t_recognitionresult_FK_R_VerifyResultValueShow; create procedure
add_t_recognitionresult_FK_R_VerifyResultValueShow() BEGIN IF NOT EXISTS (SELECT * FROM

```

```

information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_VerifyResultValueShow')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_VerifyResultValueShow(VerifyResultValueShow);End IF;END; call
add_t_recognitionresult_FK_R_VerifyResultValueShow();drop procedure if exists
add_t_recognitionresult_FK_R_VerifyResultValueShow;

drop procedure if exists add_t_recognitionresult_FK_VerifyOn; create procedure
add_t_recognitionresult_FK_VerifyOn() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_VerifyOn')THEN ALTER TABLE t_recognitionresult ADD INDEX FK_VerifyOn(VerifyOn);End IF;END; call
add_t_recognitionresult_FK_VerifyOn();drop procedure if exists add_t_recognitionresult_FK_VerifyOn;

drop procedure if exists add_t_recognitionresult_FK_R_RecogResultFlag; create procedure
add_t_recognitionresult_FK_R_RecogResultFlag() BEGIN IF NOT EXISTS (SELECT * FROM
information_schema.STATISTICS WHERE table_name = 't_recognitionresult' AND index_name =
'FK_R_RecogResultFlag')THEN ALTER TABLE t_recognitionresult ADD INDEX
FK_R_RecogResultFlag(RecogResultFlag);End IF;END; call add_t_recognitionresult_FK_R_RecogResultFlag();drop
procedure if exists add_t_recognitionresult_FK_R_RecogResultFlag;

ALTER TABLE t_recognitionresult CHANGE ID ID BIGINT(50) AUTO_INCREMENT;ALTER TABLE
t_recognitionresult CHANGE PatrolDataID PatrolDataID BIGINT(50);ALTER TABLE t_patroldata CHANGE ID ID
BIGINT(50) AUTO_INCREMENT;ALTER TABLE t_devalarm CHANGE PatrolDataID PatrolDataID BIGINT(50);

```

6.执行完成后，结果如下图所示，结束。

