Lanhai-driver

**标准版SDK使用说明文档**

金华市蓝海光电技术有限公司

https://www.pacecat.com/

目录

[1.简介………………………………………………………………………………….1](#_Toc108529212)

[2. SDK文件组织 1](#_Toc108529213)

[3. SDK开发指南 1](#_Toc108529214)

[3.1 SDK构成 1](#_Toc108529215)

[3.2 头文件介绍 2](#_Toc108529216)

[3.3 SDK主要函数接口说明 2](#_Toc108529217)

[3.4 web服务接口说明 4](#_Toc108529218)

[3.5 注意事项 5](#_Toc108529219)

[4. 配置文件说明 5](#_Toc108529220)

[5. SDK编译 6](#_Toc108529221)

[6. 示例程序 6](#_Toc108529222)

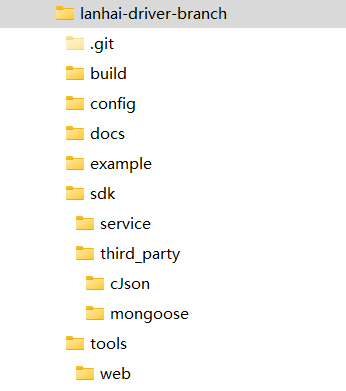
[7. 修订历史 6](#_Toc108529223)

1. 简介

本文档针对标准开源版本的 LIDAR SDK。目前该 SDK 可以在 Windows、和Linux 环境下使用。支持VC++ 2022 和g++ 98/gcc99)编译。

2. SDK文件组织

SDK 的文件结构如下图所示（cmake工程)：



Config:常用型号的雷达配置文件参数

example:测试例子，这里windows/linux通用

Docs:文档目录

Sdk:核心代码

Tools:生成的可执行文件目录，包括了windows和linux两种两种方式的可执行文件和web资源

CMakeLists.txt

Readme.md

3. SDK开发指南

### 3.1 [SDK构成](#_bookmark18)

这里以源码的方式提供，集成时需要添加sdk文件夹，引入单头文件standard\_interface.h，详细参考demo文件夹下的main.cpp文件，生成的工程中需要添加tools/web资源

### 3.2 [头文件介绍](#_bookmark20)

数据结构：data.h

错误定义头文件：error.h

全局通用类:Global.h,Global.cpp

数据收发类LidarDataProcess：

串口单独的函数（linux）Uart.c

服务类：

LidarCheckService.h 检测当前网段可用的串口/网络款/防区款雷达

LidarWebService.h web服务,用于浏览器端对雷达程序的控制以及数据获取等功能

ZoneAlarm.h 防区相关的功能(仅防区款生效)

第三方库：

cJSON

mongoose

Layui

echarts

jquery

对外提供：

standard\_interface.h 提供给用户使用的SDK头

### 3.3 [SDK主要](#_bookmark21)函数接口说明

这里主要解析standard\_interface.h/cpp文件，该文件直接提供给客户集成使用。

* read\_config

|  |  |
| --- | --- |
| 函数名称： | bool read\_config(const char\* cfg\_file\_name, RunConfig& cfg) |
| 函数参数： | 1.配置文件的绝对路径+名称[IN] 2.解析保存出的入参结构体[OUT] |
| 函数作用： | 如果使用配置文件入参的方式，则在第一步调用即可 |
| 返回值： | true/false |
| 其他说明： | 测试demo中支持单独传参或者使用配置文件传参两种方式 |

* openDev

|  |  |
| --- | --- |
| 函数名称： | int openDev(RunConfig& cfg) |
| 函数参数： | 1.运行参数[IN] |
| 函数作用： | 打开雷达设备，并且运行子进程单独处理数据 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 如果成功运行，将会打印成功信息 |

* getLidarData

|  |  |
| --- | --- |
| 函数名称： | int getLidarData(long threadID,bool dataGet) |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.是否打印[IN] |
| 函数作用： | 获取雷达点位数据 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 雷达信息包括角度，距离，强度 |

* GetDevInfo

|  |  |
| --- | --- |
| 函数名称： | int GetDevInfo(long threadID, EEpromV101& data) |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.接收的设备参数结构体[OUT] |
| 函数作用： | 获取雷达参数 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 网络款和防区款可以获得全部数据  串口款仅获得序列号 |

* SetDevInfo\_extre

|  |  |
| --- | --- |
| 函数名称： | int SetDevInfo\_extre(long threadID, DevData &data) |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.设置的设备参数结构体[IN/OUT] |
| 函数作用： | 设置雷达参数 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 这里的成功指的是雷达命令的收发成功，具体根据结构体中的result的信息判定，OK表示成功，NG表示失败  当前仅支持网络款以及防区款 |

* GetLidarTimestamp

|  |  |
| --- | --- |
| 函数名称： | int GetLidarTimestamp(long threadID,bool dataGet) |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.是否打印[IN] |
| 函数作用： | 获取雷达时间戳数据 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 当前仅支持网络款以及防区款  串口款不支持该功能 |

* OpenLocalService

|  |  |
| --- | --- |
| 函数名称： | void OpenLocalService(RunConfig& cfg) |
| 函数参数： | 1.运行参数结构体[IN] |
| 函数作用： | 启动本地服务，并且支持web访问控制，默认访问地址http://localhost:8888 |
| 返回值： | NULL |
| 其他说明： | 需要配置文件中设置  service\_port:8888 设置本地开发端口  is\_open\_service:1 0关闭 1打开 |

* CloseLocalService

|  |  |
| --- | --- |
| 函数名称： | void CloseLocalService() |
| 函数参数： |  |
| 函数作用： | 关闭本地服务 |
| 返回值： | NULL |
| 其他说明： |  |

* ControlDrv

|  |  |
| --- | --- |
| 函数名称： | int ControlDrv(long threadID, const char\*data); |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.控制雷达运行指令[IN] |
| 函数作用： | 控制雷达运行 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 指令包括以下4种：  LSTARH :开始运行  LSTOPH:停止运行  LRESTH:重新运行  LMEASH:固定测距 |

* getVersion

|  |  |
| --- | --- |
| 函数名称： | const char\* getVersion(); |
| 函数参数： | 无 |
| 函数作用： | 获取SDK版本号 |
| 返回值： | 版本号 |
| 其他说明： | 无 |

* GetAlarmZone

|  |  |
| --- | --- |
| 函数名称： | int GetAlarmZone(long threadID, RecvZoneDatas&data) |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.获取的防区数据[OUT] |
| 函数作用： | 获取雷达防区信息 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 当前仅支持矩形和扇形 |

* SetAlarmZone

|  |  |
| --- | --- |
| 函数名称： | int SetAlarmZone(long threadID, zones &data); |
| 函数参数： | 1.子线程ID/子线程队列ID[IN] 2.待设置的防区数据[IN] |
| 函数作用： | 设置雷达防区 |
| 返回值： | 0表示成功，其他均为失败，详情参考error.h头文件定义标准错误 |
| 其他说明： | 当前仅支持矩形和扇形 |

### 3.4 web服务接口说明

详情参考 雷达web接口.docx

### 3.5 **[注意事项](#_bookmark17)**

SDK使用C/C++方式开发。建议开发人员在使用lanhai-driver SDK前，对雷达的通讯协议(uart/udp)和工作模式(硬件正常工作的情况，包括指示灯等)有所了解。可以参考具体型号的使用手册获取相关细节。

当配置文件从windows复制到linux上时，因为平台间回车换行符的问题，可以通过vim打开xxx.txt文件，set ff=unix 来一键修改文件格式。

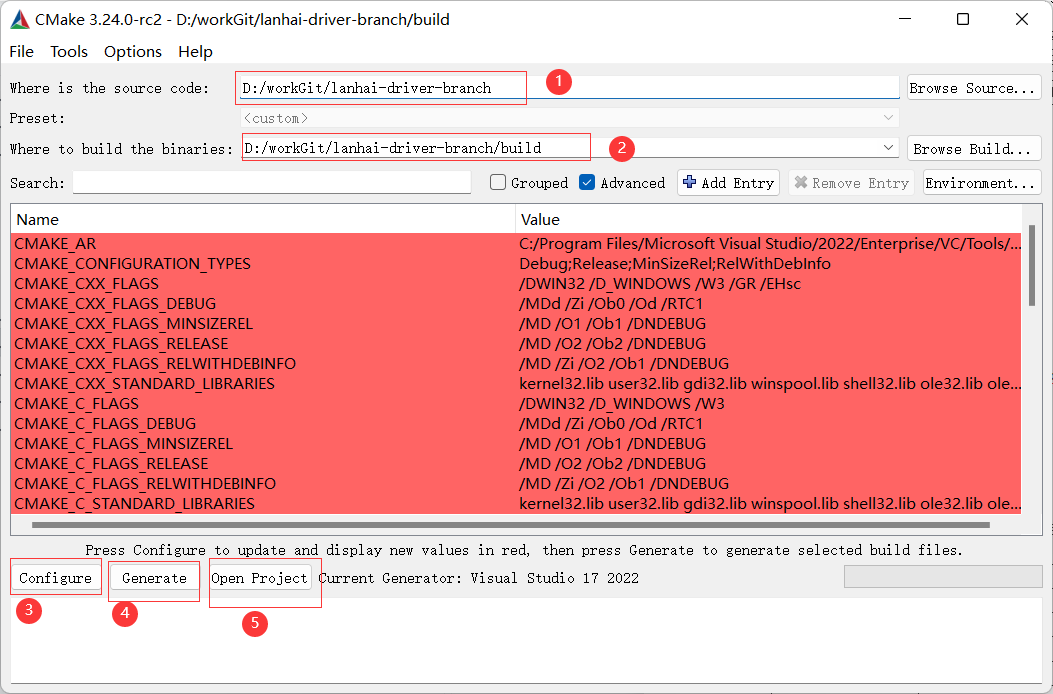
4. 配置文件说明

|  |  |  |
| --- | --- | --- |
| type | 雷达的类型 | uart:串口  udp：网络款  udp\_uart：虚拟串口 |
| port | 串口雷达的USB口号 | 举例：  Win:com1  Linux:/dev/ttyUSB0  Linux VPC:/dev/ttyACM0 |
| baud\_rate | 串口雷达的波特率 | 举例：768000 |
| lidar\_ip | 目标雷达的IP | 举例：192.168.0.98 |
| lidar\_port | 目标雷达的访问端口 | 举例：6543 |
| local\_port | 主机的接收端口 | 举例：6888 |
| raw\_bytes | 打包字节数 | 2/3 |
| unit\_is\_mm | 是否是毫米为数据单位 | 0/1 |
| with\_confidence | 是否数据带强度 | 0/1 |
| with\_checksum | 是否数据校验 | 0/1 |
| with\_smooth | 是否数据平滑 | 0/1 |
| with\_deshadow | 是否去拖点 | 0 |
| resample | 角分辨率 | 举例：400 |
| rpm | 转速 | 举例：600 |
| output\_scan | 是否打印点位输出 | 0/1 |
| output\_360 | 是否按360度一次输出  0：部分扇区 1：全扇形 | 0/1 |
| from\_zero | 是否零度输出  0 -180°-180° 1 0°-360 | 0/1 (暂不生效) |
| output\_file | 是否将点云数据保存的到文件 | 0/1 |
| is\_group\_listener | 对组播模式的设置 | 0正常模式 1监听模式 2发送模式 |
| group\_ip | 组播模式的IP | 举例：224.0.0.99 |
| service\_port | Web本都服务的开放端口 | 举例：8888 |
| is\_open\_service | 是否启动web服务 | 0/1 |

1. SDK编译

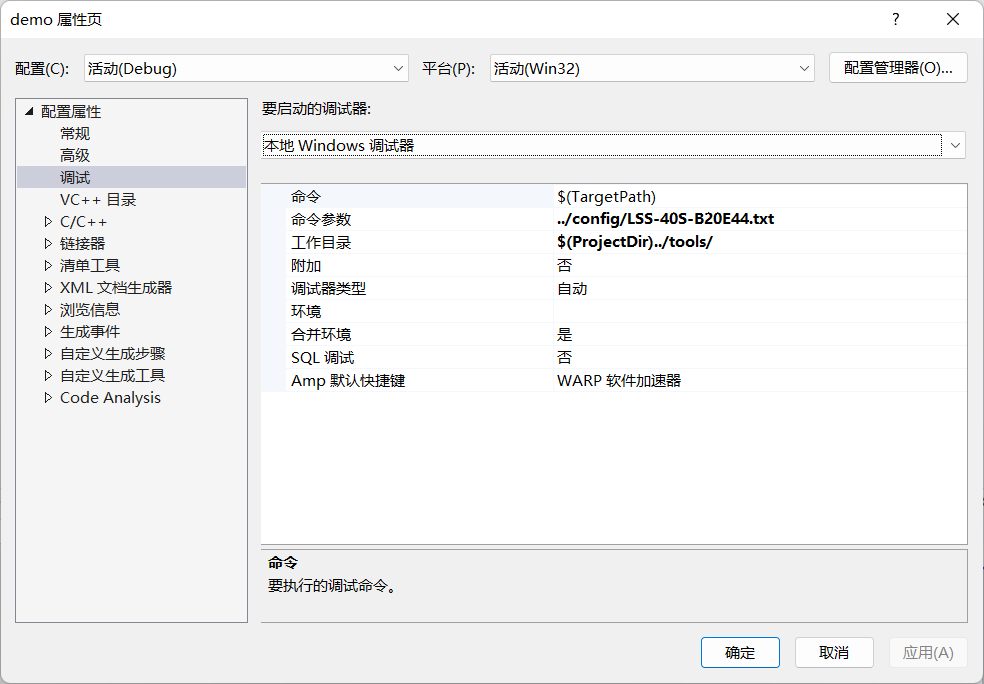
Windows:

5.1 Widows:需要下载cmake工具，然后使用cmake-gui生成指定的vs工程，这里以vs2022为例。

5.2 vs工程：解决方案->设置demo为启动项

5.3 项目->属性->调试，设置命令参数(调用雷达的配置文件)以及工作目录(固定)。

，



5.4 生成->生成解决方案

5.5 调试->开始运行

Linux

5.6 cmake CMakeList.txt

5.7 make

5.8 make install

5.9 cd tools

5.10 ./demo ../config/xxxx.txt

注:如果是虚拟机下的串口设备，还需要sudo chmod 777 /dev/ttyUSB\*,给设备赋权

6. 示例程序

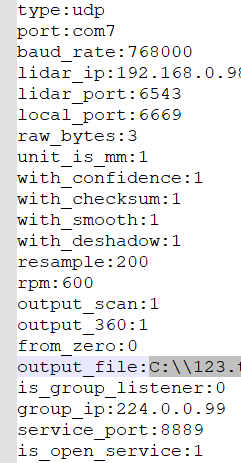
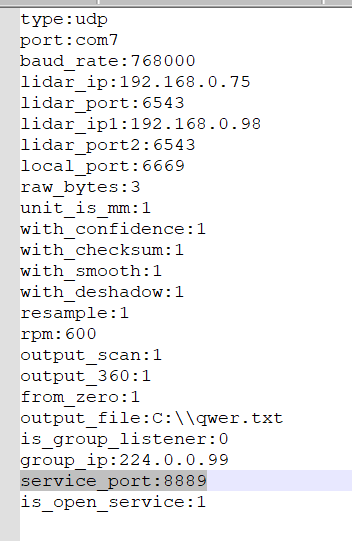
源码详情参考example目录下的main.cpp文件

可执行文件详情参考tools目录

### 6.1多雷达配置

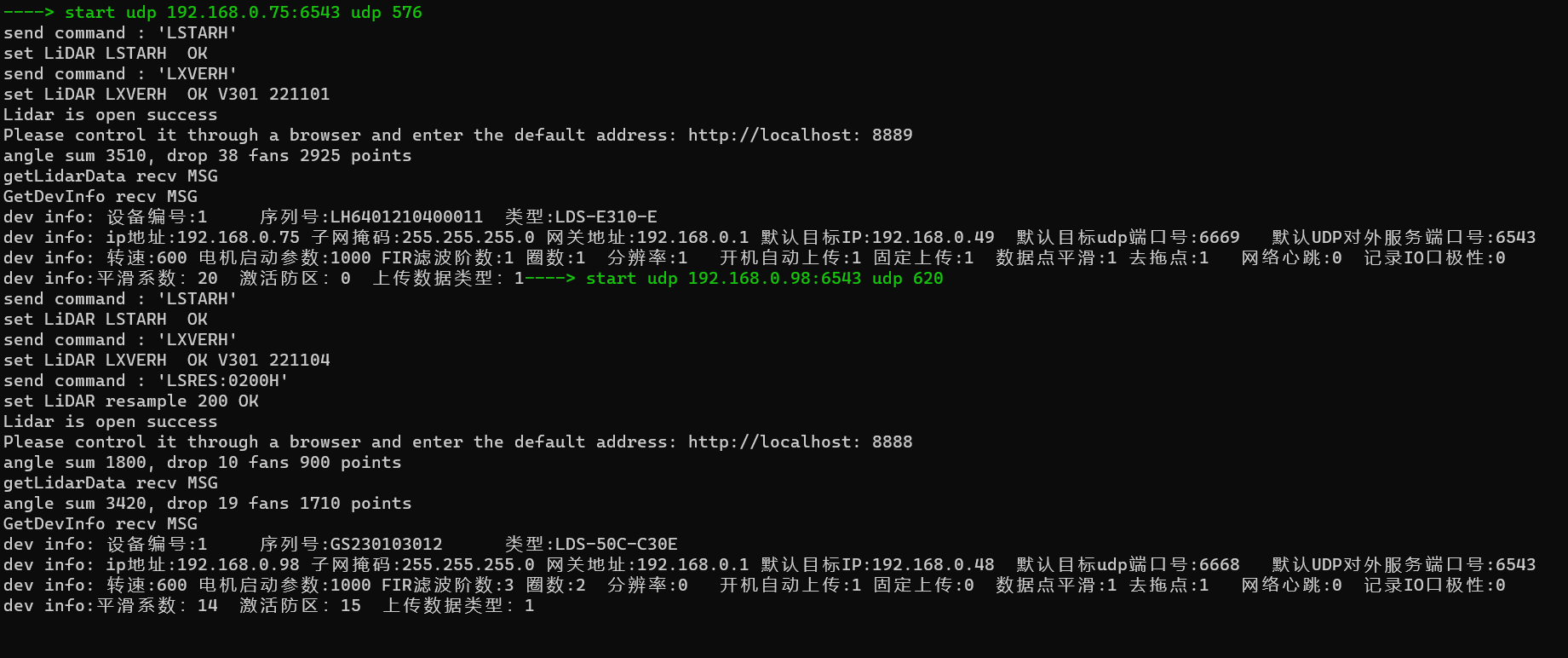
这里将配置文件的相对路径(绝对路径)传入，

例如 ./demo config/LDS-E310-E.txt config/LDS-50C-C30E.txt 具体配置信息如下



注意：必须要保持配置文件中的local\_port以及service\_port不同

成功运行后，会有两个运行成功打印，如果有一个失败，就会直接退出进程



如果要查看web显示，则需要打开两个tab页,端口对应着配置文件的service\_port

http:://localhost:8888

http:://localhost:8889

1. 控制指令表

说明：这里指SDK中涉及到的硬件指令含义,并不是所有型号都全部支持以下命令，具体型号支持的指令查看该雷达型号的用户手册。

|  |  |
| --- | --- |
| 指令 | 说明 |
| LSTARH | 开始运行 |
| LSTOPH | 停止运行 |
| LRESTH | 重新运行 |
| LMEASH | 固定测距 |
|  |  |
| LXVERH | 获取硬件信息 |
| LMDCMH | 设置雷达点云数据的单位 CM |
| LMDMMH | 设置雷达点云数据的单位 MM |
| LNCONH | 打开雷达强度数据 |
| LOCONH | 关闭雷达强度数据 |
| LFFF0H | 打开去拖点 |
| LFFF1H | 关闭去拖点 |
| LSSS0H | 打开数据平滑 |
| LSSS1H | 关闭数据平滑 |
| LSRES:000H | 设置默认的角分辨率 |
| LSRES:001H | 设置修正的角分辨率 |
|  |  |
| LSRPM:%04dH | 设置转速 范围0450-1200 例如：LSRPM:0450H |
| LSERR:+%dH | 设置角度偏差 范围-99-+99 例如:LSERR:-23H LSERR:+23H |
| LSUDP:%sH | 设置udp组合信息 设置雷达IP地址 子网掩码 网关 服务端口号，例如LSUDP:192.168.158.091 255.255.255.000 192.168.158.001 05000H |
| LSDST:%sH | 设置接收雷达信息的ip地址和端口号 LSDST:192.168.158.043 12300H |
| LSNSP:%sH | 设置机器类型 LSNSP:LDS-50C-S-UH |
| LSUID:%sH | 设置机器序号 例如LSUID:201812030001H |
| LSFIR:%02dH | 设置滤波圈数 (范围01~99) 例如：LSFIR:03H |
| LSPUL:%04dH | 设置电机启动脉冲数 (范围0500~4000) 例如：LSPUL:2000H |
| LSVER:%04dH | 设置版本号 例如：LSPUL:2000H |
| LSPNP:%dH | 设置IO类型 比如设置LSNPN:1H 输出IO类型为PNP |
| LSSMT:%dH | 设置数据平滑 LSSMT:1H 打开 LSSMT:0H 关闭 |
| LSDSW:%dH | 设置去拖点 LSDSW:1H 打开 LSDSW:0H 关闭 |
| LSDID:%dH | 设置设备ID LSDID:xxxH |
| LSATS:%dH | 开机自动上传 LSATS:xH 1/0 打开/关闭 |
| LSTFX:%dH | 固定上传地址 LSTFX:xH 1/0 打开/关闭 |
| LSPST:%dH | 数据/报警上传类型 LSPST:xH 0:无 1:数据 2报警 3 数据+报警 |
| LSAF:%dH | 去拖点系数 |

8. 修订历史

|  |  |  |
| --- | --- | --- |
| 日期 | 内容 | 编写人 |
| 2022.5.10 | 初稿 | 王子宁 |
| 2022.7.1 | 新增本地服务，对web访问的支持  新增对当前网段可用雷达的检测服务  优化工作 | 王子宁 |
| 2022.8.30 | 新增节点防区功能(网络防区款以及虚拟串口款)  新增web绘制以及显示防区，当前仅支持矩形以及扇形  调整工程为cmake结构  更新相关的注释以及优化工作 | 王子宁 |
| 2022.9.14 | 版本号:V1.2  新增对LDS-E100-R以及LDS-E340-E的支持  对初始化设置配置参数的应答优化 | 王子宁 |
| 2023.2.14 | 版本号:V2.0  对内部文件结构重新调整  新增多雷达运行的支持 | 王子宁 |