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/

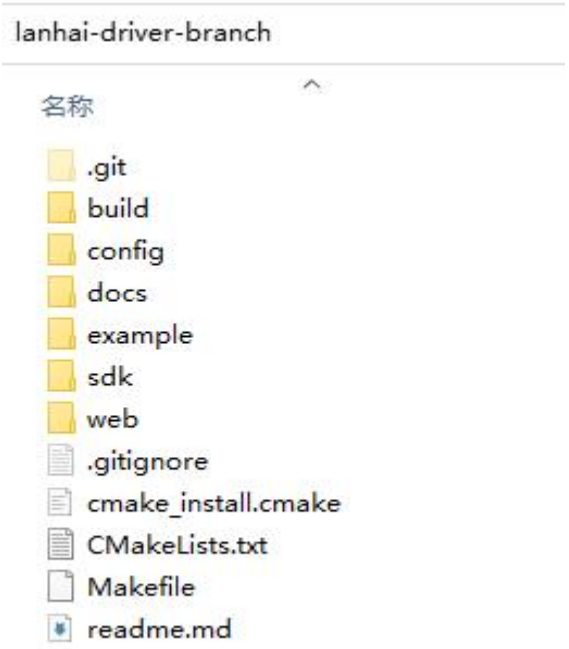
目录

1. 简介

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++6.0 以上版本(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(集成 demo)

Web:web 资源

CMakeLists.txt

Readme.md

3. SDK 开发指南

3.1 SDK 集成说明

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。

点云数据读取、同步时间戳打印、运行信息打印等可以参考

```
void CallBackMsg(int msgtype, void *param,int length)
```

函数

参数 1：自定义的信息类型

参数 2:回调的数据

参数 3：回调的数据长度

主要步骤说明：

1. 主线程通过 addLidarByPath 将配置文件中的数据添加，并返回唯一的 ID
2. 主线程通过 setCallBackPtr 设置每个雷达工作线程的回调函数，参考 CallBackMsg
3. 主线程通过 openDev 打开指定 ID 的雷达，并启动工作线程(必须)，web 可视化线程(可选)，心跳线程(可选)
4. 工作线程通过回调函数返回点云的具体信息
5. 主线程通过调用 standard_interface.h 中的接口读取/设置其他功能，详情参考以下的接口

3.2 头文件介绍

数据结构：data.h

错误定义头文件：error.h

通用接口:Global.h

数据处理 LidarDataProcess.h

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

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。

- addLidarByPath

函数名称：	int addLidarByPath(const char *cfg_file_name)
函数参数：	1.配置文件的绝对路径+名称　　[IN]
函数作用：	通过配置文件新增雷达
返回值：	雷达 ID
其他说明：	

- delLidarByID

函数名称：	bool delLidarByID(int ID);
函数参数：	1.雷达 ID[IN]
函数作用：	通过 ID 删除指定雷达
返回值：	True/false
其他说明：	

- setCallBackPtr

函数名称：	void setCallBackPtr(int ID,printfMsg ptr);
函数参数：	1.雷达 ID[IN]　　2.回调指针[IN]typedef void (*printfMsg)(int, void*,int);
函数作用：	根据雷达 ID，设置对应的回调指针
返回值：	
其他说明：	

- openDev

函数名称：	bool openDev(int ID);
函数参数：	1.雷达 ID[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：	True/false
其他说明：	如果成功运行，将会打印成功信息

- StopDev

函数名称：	void StopDev(int ID);
函数参数：	1.雷达 ID[IN]
函数作用：	关闭雷达设备及其子线程
返回值：	
其他说明：	

- GetDevInfo

函数名称：	bool GetDevInfo(int ID,EEpromV101 *eepromv101);
函数参数：	1.雷达 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：	Ture/false
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

- ControlDrv

函数名称：	bool ControlDrv(int ID, int num, char *cmd);
函数参数：	1.雷达 ID[IN]　2.指令长度[IN]　3.指令[IN]
函数作用：	控制雷达运行
返回值：	True/false
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

- 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

- ZoneSection

函数名称：	bool ZoneSection(int ID, char section);
函数参数：	1.雷达 ID　2.选择的防区
函数作用：	切换防区
返回值：	True/false
其他说明：	防区范围说明　0-9,A-F

- SetUDP

函数名称：	bool SetUDP(int ID, char* ip, char* mask, char* gateway, int port)

函数参数：	1.雷达 ID　2.雷达 ip　3.掩码 4.网关　5 端口
函数作用：	设置目标雷达的网络
返回值：	True/false
其他说明：	当切换雷达 ip 后，需要重新设置 socket 才能接收以后的数据

- SetDST

函数名称：	`bool SetDST(int ID, char* ip, int port);`
函数参数：	1.雷达 ID　2.雷达数据上传的 ip 3 端口
函数作用：	设置目标雷达的数据上传地址
返回值：	True/false
其他说明：	

- SetRPM

函数名称：	`bool SetRPM(int ID,int RPM);`
函数参数：	1.雷达 ID　2.转速
函数作用：	设置雷达的转速
返回值：	True/false
其他说明：	

- SetTFX

函数名称：	`bool SetTFX(int ID,bool tfx);`
函数参数：	1.雷达 ID　2.是否固定上传
函数作用：	设置固定上传
返回值：	True/false
其他说明：	默认关闭

- SetDSW

函数名称：	`bool SetDSW(int ID,bool dsw);`
函数参数：	1.雷达 ID　2.是否去拖点
函数作用：	设置去拖点
返回值：	True/false
其他说明：	默认打开

- SetSMT

函数名称：	`bool SetSMT(int ID,bool smt);`

函数参数：	1.雷达 ID　2.是否数据平滑
函数作用：	设置数据平滑
返回值：	True/false
其他说明：	默认打开

- SetPST

函数名称：	`bool SetPST(int ID,int mode);`
函数参数：	1.雷达 ID　2.数据上传类型
函数作用：	设置 SetPST
返回值：	True/false
其他说明：	数据上传类型:0 无数据 1 数据 2 报警 3 数据+报警

- SetDID

函数名称：	`bool SetDID(int ID,unsigned int number);`
函数参数：	1.雷达 ID　2.雷达编号
函数作用：	设置雷达编号
返回值：	True/false
其他说明：	与雷达 ID 不同

- getLidarsList

函数名称：	`std::vector<DevConnInfo> getLidarsList()`
函数参数：	
函数作用：	获取网络款雷达的心跳包数据和当前可用的串口雷达
返回值：	雷达信息结构体数组
其他说明：	

3.4 注意事项

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,也可以通过售后使用说明查看常见问题。

4. 配置文件说明

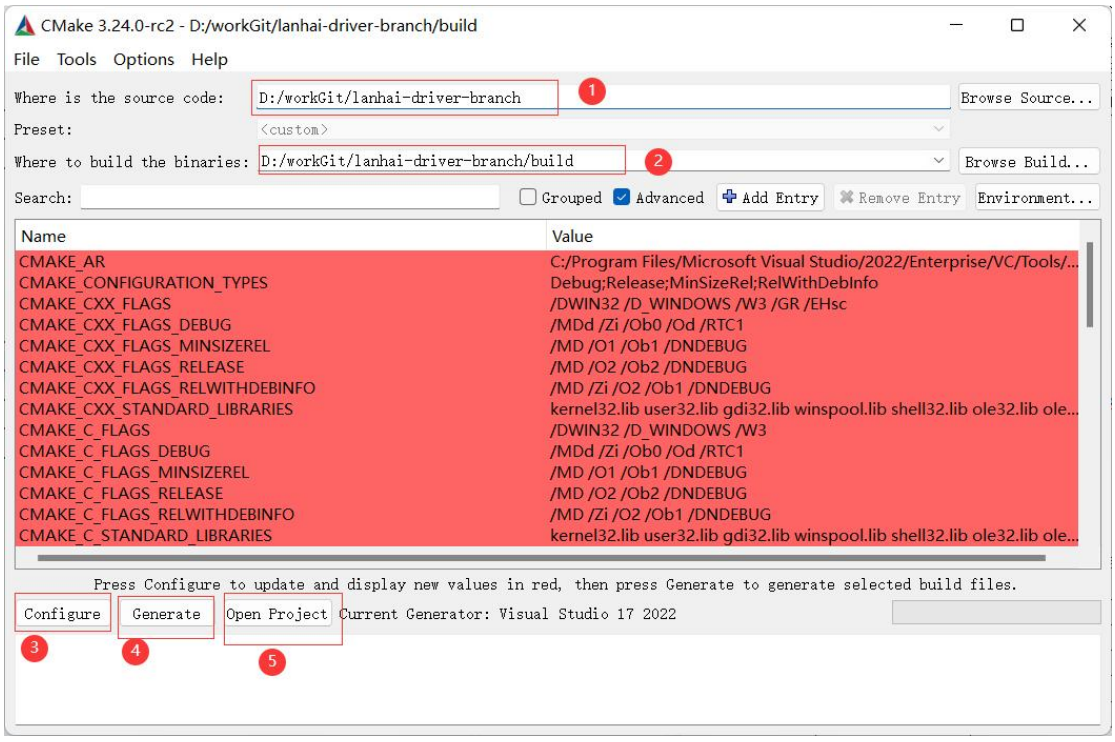
connect		
type	雷达的类型	uart:串口
		udp：网络款
		udp_uart：虚拟串口
connectArg	串口雷达的 USB 口号	Win:com1 Linux:/dev/ttyUSB0 Linux VPC:/dev/ttyACM0
	目标雷达的 IP	举例：192.168.158.98
connectArg2	串口雷达的波特率	举例：768000
	目标雷达的访问端口	举例：6543
local_port	主机的接收端口	举例：6888
is_group_listener	是否开启广播	0/1
group_ip	广播 IP	224.0.0.99
data		
output_360	是否按 360 度一次输出	0：部分扇区 1：全扇形
from_zero	是否零度输出	0/1 0 -180°-180° 1 0°-360
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
error_circle	统计符合要求（距离为 0 的点数量/总数大于指定系数）的错误圈数	默认 3
error_scale	比例系数 距离为 0 的点:总点数	默认 0.9
filter		
filter_open	滤波生效开关	1
max_range	滤波生效的最大距离	20
min_range	滤波生效的最小距离	0.5
max_range_difference	离异点判断的物理范围	0.1
filter_window	判断离异点下标的范围	1
get		
model	查询型号	-1 不执行 >=0 执行
version	查询硬件版本号	-1 不执行 >=0 执行

set		
unit_is_mm	是否是毫米为数据单位	-1/0/1　不执行/否/是
with_confidence	是否数据带强度	-1/0/1　不执行/否/是
with_smooth	是否数据平滑	-1/0/1　不执行/否/是
with_deshadow	是否去拖点	-1/0/1　不执行/否/是
resample	角分辨率	-1/0/1/xxx　不执行/非固定/固定角分辨率/具体的固定值
rpm	转速	-1/xx　不执行/具体转速
direction	旋转方向	-1/0/1　不执行/顺时针/逆时针　仅部分雷达生效
alarm_msg	报警信息上传	-1/0/1　不执行/仅数据/数据加报警
ats	连接方式	-1/0/1　不执行/网络/USB
with_start	启动时默认发送开始旋转指令	-1/0/1　不执行/否/是

5. SDK 编译

Windows:

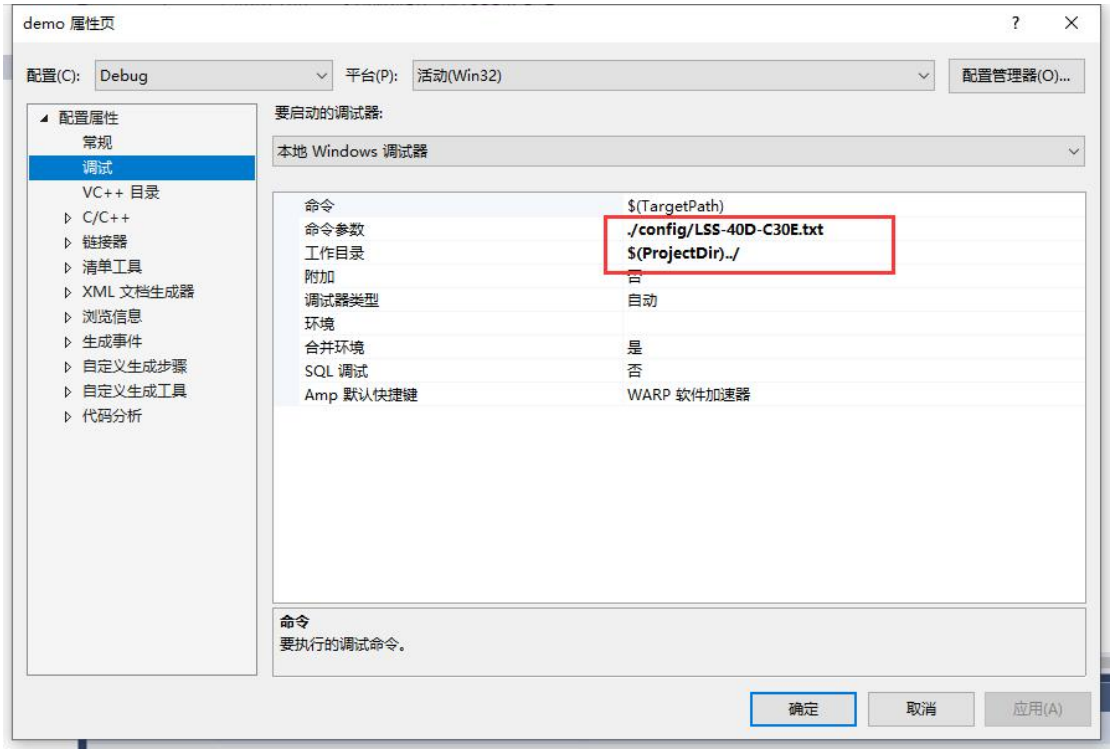
5.1 Widows:需要下载 cmake 工具，然后使用 cmake-gui 生成指定的 vs 工程，这里以 vs2022 为例(cmake 需要比较新的版本)



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 ./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

```
type:udp
port:com7
baud_rate:768000
lidar_ip:192.168.0.75
lidar_port:6543
lidar_ip1:192.168.0.98
lidar_port2:6543
local_port:6669
raw_bytes:3
unit_is_mm:1
with_confidence:1
with_checksum:1
with_smooth:1
with_deshadow:1
resample:1
rpm:600
output_scan:1
output_360:1
from_zero:1
output_file:C:\\qwer.txt
is_group_listener:0
group_ip:224.0.0.99
service_port:8889
is_open_service:1
```

```
type:udp
port:com7
baud_rate:768000
lidar_ip:192.168.0.9
lidar_port:6543
local_port:6669
raw_bytes:3
unit_is_mm:1
with_confidence:1
with_checksum:1
with_smooth:1
with_deshadow:1
resample:200
rpm:600
output_scan:1
output_360:1
from_zero:0
output_file:C:\\123.
is_group_listener:0
group_ip:224.0.0.99
service_port:8889
is_open_service:1
```

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

```
----> start udp 192.168.0.75:6543 udp 576
send command : 'LSTARH'
set LiDAR LSTARH  OK
send command : 'LXVERH'
set LiDAR LXVERH  OK V301 221101
Lidar is open success
Please control it through a browser and enter the default address: http://localhost: 8889
angle sum 3510, drop 38 fans 2925 points
getLidarData recv MSG
GetDevInfo recv MSG
dev info: 设备编号:1     序列号:LH64012104000011  类型:LDS-E310-E
dev info: ip地址:192.168.0.75 子网掩码:255.255.255.0 网关地址:192.168.0.1 默认目标IP:192.168.0.49  默认目标udp端口号:6669   默认UDP对外服务端口号:6543
dev info: 转速:600 电机启动参数:1000 FIR滤波阶数:1 圈数:1  分辨率:1   开机自动上传:1 固定上传:1  数据点平滑:1 去拖点:1   网络心跳:0  记录IO口极性:0
dev info:平滑系数: 20  激活防区: 0  上传数据类型: 1----> start udp 192.168.0.98:6543 udp 620
send command : 'LSTARH'
set LiDAR LSTARH  OK
send command : 'LXVERH'
set LiDAR LXVERH  OK V301 221104
send command : 'LSRES:0200H'
set LiDAR resample 200 OK
Lidar is open success
Please control it through a browser and enter the default address: http://localhost: 8888
angle sum 1800, drop 10 fans 900 points
getLidarData recv MSG
angle sum 3420, drop 19 fans 1710 points
GetDevInfo recv MSG
dev info: 设备编号:1     序列号:GS230103012      类型:LDS-50C-C30E
dev info: ip地址:192.168.0.98 子网掩码:255.255.255.0 网关地址:192.168.0.1 默认目标IP:192.168.0.48  默认目标udp端口号:6668   默认UDP对外服务端口号:6543
dev info: 转速:600 电机启动参数:1000 FIR滤波阶数:3 圈数:2  分辨率:0   开机自动上传:1 固定上传:0  数据点平滑:1 去拖点:1   网络心跳:0  记录IO口极性:0
dev info:平滑系数: 14  激活防区: 15  上传数据类型: 1
```

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

6.2 雷达数据整圈为 0 提示

```
type:uart
port:com10
baud_rate:500000
lidar_ip:192.168.0.208
lidar_port:6543
local_port:6668
raw_bytes:3
unit_is_mm:1
with_confidence:1
with_checksum:1
with_smooth:1
with_deshadow:1
resample:400
rpm:600
output_scan:1
output_360:0
from_zero:0
output_file:/tmp/udp_data2.txt
is_group_listener:0
group_ip:224.0.0.99
service_port:8888
is_open_service:1
error_circle:3
error_scale:0.9
```

配置文件新增：

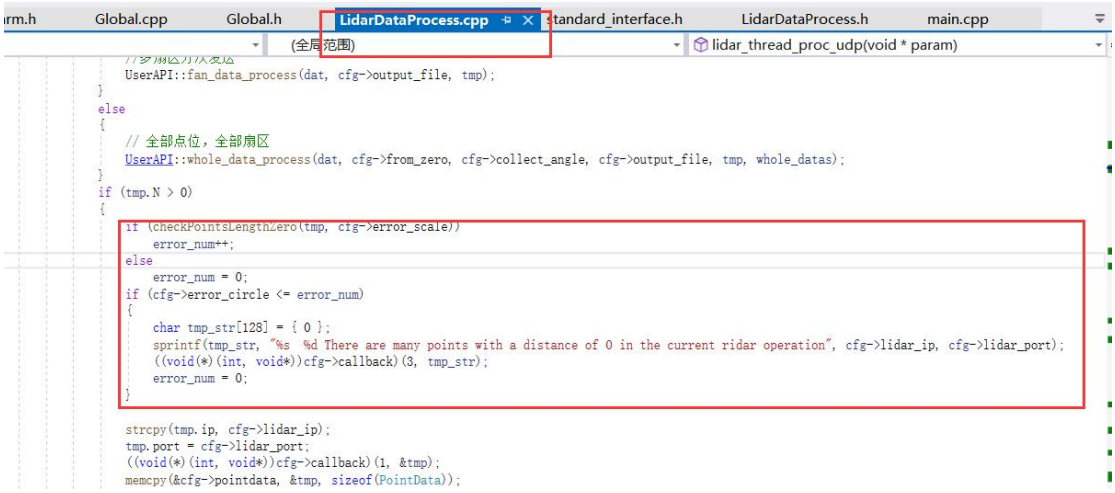
error_circle：这里指连续 3 圈出现点距离为 0，并且超过指定比例个数

error_scale: 这个是自定义的距离为 0 系数，即总点数 * 指定系数 = 判定的点数

```
//传入回调指针的方式打印
void CallBackMsg(int msgtype, void* param)
{
    //实时雷达数据返回
    if (msgtype == 1) { ... }
    //实时防区数据返回
    else if (msgtype == 2) { ... }
    //获取错误信息
    else if (msgtype == 3)
    {
        char* result = (char*)param;
        INFO_PR("Error Info : %s\n", result);

    }
    //获取雷达时间戳打印信息
    else if (msgtype == 4) { ... }
}
```

在最外层的 demo.cpp 对回调函数返回的错误打印，仅收到一个扇区/一圈全部点数的长度都为 0 时返回报错（扇区/圈根据 output_360 参数）



内部调整：当获取一圈/一个扇区信息时，对所有的点位长度进行判断，如果符合要求的点达到指定系数，并且持续指定圈数，则通过回调函数返回错误，如果中间有符合要求的点数，则重置错误圈数累计。

6.3 雷达各项数据的获取(集成使用)

详细参考 main.cpp 中的 CallBackMsg 回调函数，具体返回数据有

- 点云数据
- 报警数据　需要在配置文件加上 alarm_msg:1(打开报警数据上传)
- 报错信息
- 时间戳打印

雷达全局参数考察　PrintMsg 函数

7. 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
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
LSUID:%sH	设置机器序号　　例如 LSUID:201812030001H
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