

LakiBeam SDK 用户手册

目录

1 体系结构	4
2 使用说明	5
2.1 环境要求	5
2.2 开发环境建设	5
3 类说明(UDP 协议)	9
3.1 结构体定义	9
3.2 函数说明	9
3.2.1 构造函数	9
3.2.2 析构函数	10
3.2.3 dots_valid()	10
3.2.4 get_pack()	10
3.2.5 restart()	10
3.2.6 wait()	10
4 类说明(HTTP 协议)	11
4.1 类定义	11
4.2 函数说明	11
4.2.1 delete_override()	11
4.2.2 get_filer_level()	11
4.2.3 get_firmware()	12
4.2.4 get_host()	12
4.2.5 get_host_IP()	12
4.2.6 get_host_port()	13
4.2.7 get_laser_enable()	13
4.2.8 get_laser_start()	13
4.2.9 get_lase_stop()	14
4.2.10 get_monitor()	14
4.2.11 get_motor_rpm()	14
4.2.12 get_network()	15
4.2.13 get_overview()	15
4.2.14 get_resolution()	15
4.2.15 get_scan_range()	16
4.2.16 get_scanfreq()	16
4.2.17 put_filter_level()	16
4.2.18 put_host_IP()	17
4.2.19 put_host_port()	17
4.2.20 put_laser_enable()	17
4.2.21 put_laser_start()	18
4.2.22 put_laser_stop()	18
4.2.23 put_override()	19
4.2.24 put_reset()	19
4.2.25 put_scanfreq()	19
5 SDK 开发流程	20

5.1 激光雷达主动上传数据模式流程 20

1 体系结构

Lakibeam 数据采集系统从软件的角度包括以下几个部分：

- (1) Lakibeam：采集环境数据和信息。
- (2) Etherent：Lakibeam 通过 Ethernet 与设备端连接。LakiBeam 采集到的数据通过 UDP 协议发送到设备端光束通过以太网连接到设备端。
- (3) SDK：可以通过 SDK 提供的接口接收 LakiBeam 采集的数据，并配置激光雷达参数、IP 地址等。

2 使用说明

2.1 环境要求

此 SDK 是源码发布，请用户集成到目标系统中。

SDK，使用 boost 库，在编译之前，请准备 boost 库，推荐 boost 库的 1.63 及以上版本。

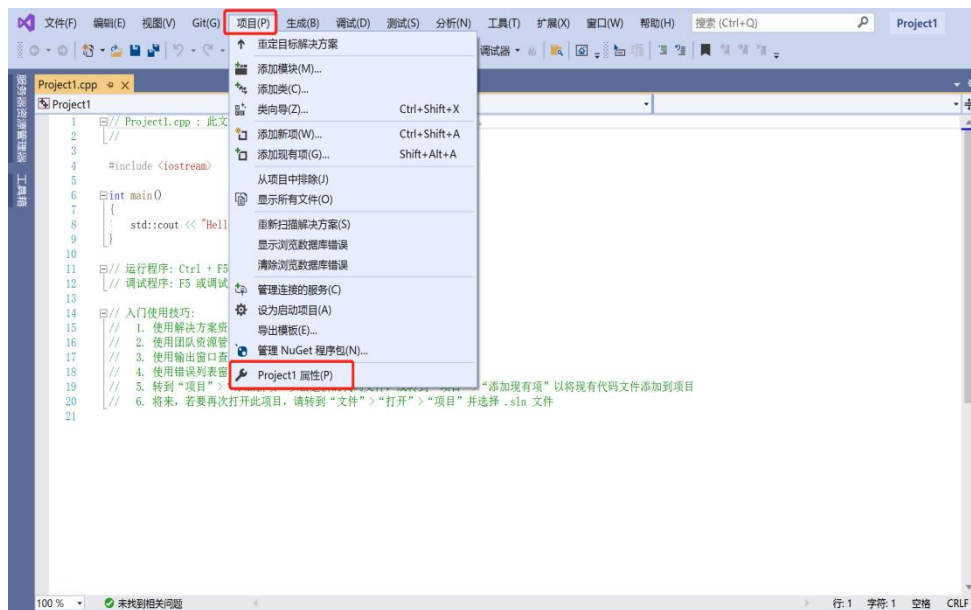
2.2 开发环境建设

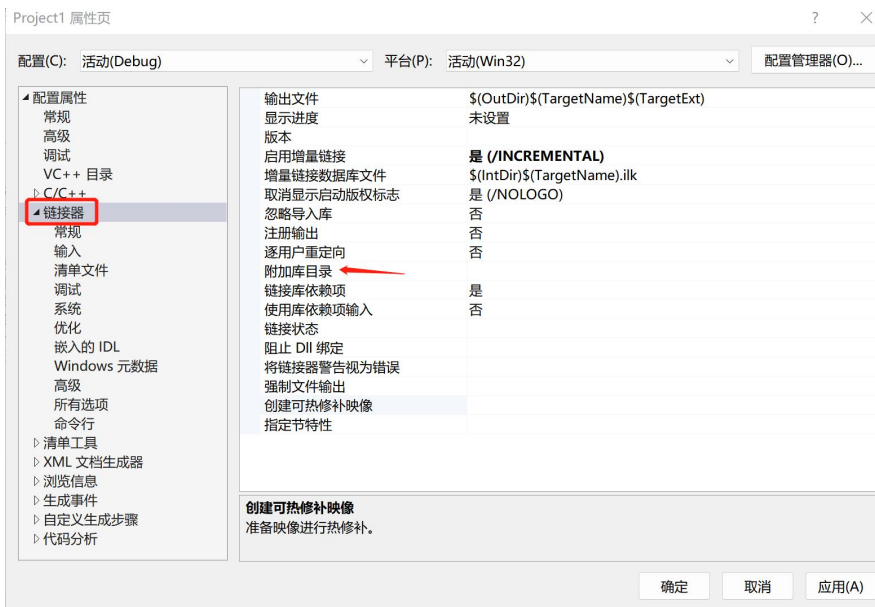
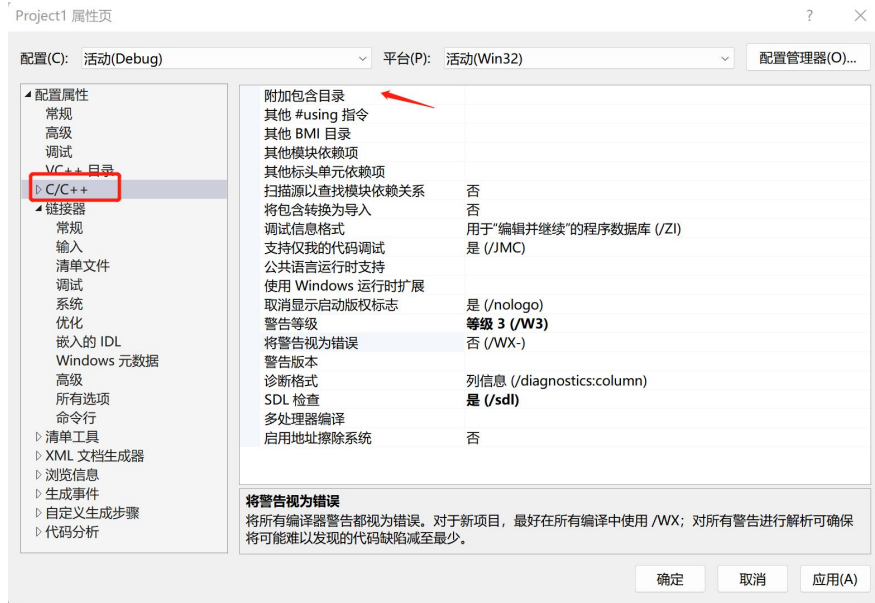
windows (Visual Studio):

对于基于 Windows 操作系统的 LakiBeam SDK 应用程序开发，您的开发环境应该满足以下条件：建议您的计算机安装 Visual Studio 2019 并安装编译的 boost 库。

(1) 配置编译选项:

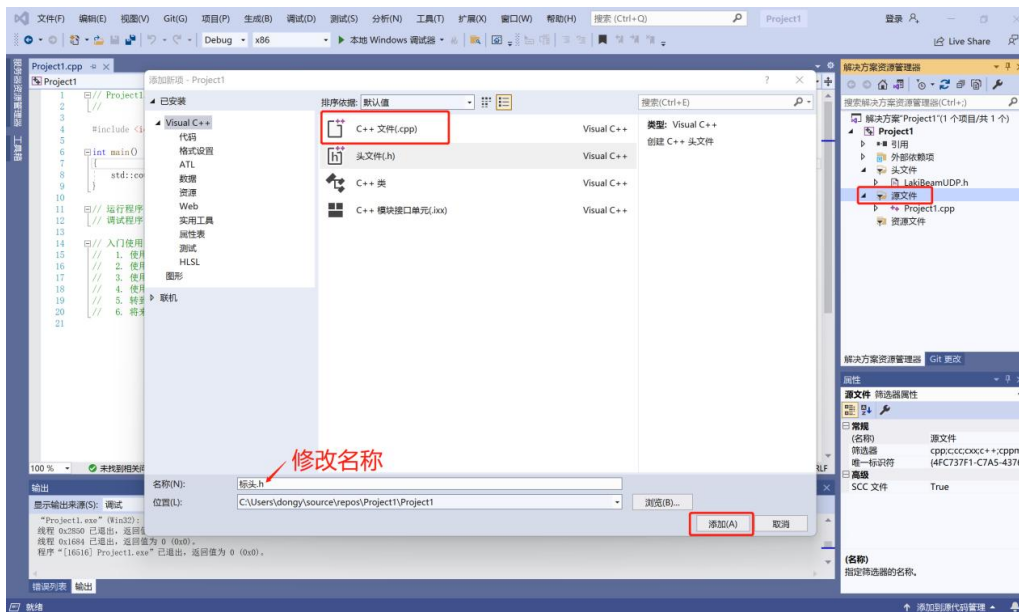
a. 打开项目属性”面板





- 在左侧列表中，分别选择 C / C + +和链接器
- 选择右侧的附加包含目录和附加库目录，然后单击下拉按钮
- 选择“编辑”
- 将 boost 依赖项添加到该列表中
- 点击“应用”，完成配置

(2) 创建源文件



(3) 添加代码：在弹出的编辑器中放入 LakiBeamUDP.cpp 文件中的代码

(4) 编译运行：在 Visual Studio 的主菜单中，单击调试->开始调试即可编译运行您的程序

ubuntu (Visual Studio Code):

(1) 安装编译 boost 库

- a. 下载 boost (版本高于 1.63)，以 1.78 版本举例
- b. 将压缩包解压
- c. 打开终端

```
cd boost_1_78_0 // 进入到文件夹中, boost_1_78_0 为文件名 //
```

```
./bootstrap.sh // 执行命令 //
```

```
./b2 install --prefix=/home/fen/boost // home/fen/boost 为自己的  
安装路径, 可以新建一个文件夹当作安装路径 //
```

```
cd ~/boost // 进入新建的 boost 目录 //
```

```
ls // 目录下会有 include 和 lib 两个文件夹, 一个  
为头文件, 一个是库文件 //
```

```
sudo mv -f ./lib/* /usr/lib // 将库文件移动到/usr/lib 下, 显示无法移  
动也不要紧, 继续下一步 //
```

```
sudo cp -rf ./include/boost /usr/include // 将头文件复制到  
/usr/include 下, boost 编译完成 //
```

(2) 获取 sdk

- a. 解压 sdk (sdk 压缩包为.rar, 如无法解压, 按以下步骤进行, 如可以解压, 直接跳到第三步)
- b. 打开终端

```
sudo apt-get install rar
sudo apt-get install unrar
```

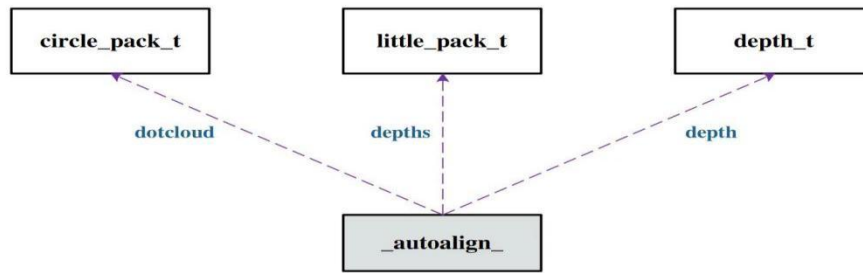
(3) 测试代码

- c. 下载 vscode 编译工具
- d. 在 vscode 里打开 sdk
- e. 运行和调试

3 类说明(UDP 协议)

3.1 结构体定义

autoalign 结构体的协作图:



circle_pack_t 结构体: 一个重新打包的深度包最小单元

depth_t 结构体: 激光雷达深度数据的最小包裹

little_pack_t 结构体: 一帧激光雷达深度数据帧的数据结构的最小单位

repack_t 结构体: 一帧表示一块深度数据

udp_pack_t 结构体: 一帧 udp 数据包的数据结构

LakiBeamUDP: 激光雷达 UDP 数据包拆包解包类, 接收 lakibeam 雷达的深度数据, 重新组装数据块, 每个数据块包含一圈深度数据, 块与块之间顺序连接

3.2 函数说明

3.2.1 构造函数

LakiBeamUDP()

LakiBeamUDP::LakiBeamUDP (

string local IP,

string local port,

string laser IP,

string laser port)

参数:

local IP: 本地 IP

local port: 本地端口号

laser IP: 雷达 IP

laser port: 雷达端口号

3.2.2 析构函数

~ LakiBeamUDP()

LakiBeamUDP::~~ LakiBeamUDP ()

3.2.3 dots_valid()

u32_t LakiBeamUDP::dots_valid (void)

获取激光雷达包裹有效值

返回值:

激光雷达的数据有效个数

3.2.4 get_pack()

bool LakiBeamUDP::get_pack (repack_t & pack)

获取激光雷达数据包:

参数:

pack: 雷达数据返回值, 具体包括时间戳、角度值、深度数据、回波强度

返回值:

true:操作成功

false:操作失败

3.2.5 restart()

void LakiBeamUDP::restart (void)

重启雷达 UDP 的 SDK 功能包

3.2.6 wait()

void LakiBeamUDP::wait (void)

等待雷达新的一包数据到来

4 类说明(HTTP 协议)

4.1 类定义

filter_c 类: 雷达滤波等级类, 滤波器参数

firemwared_c 类: 获取雷达固件信息封装类

host_c 类: 主机配置类, 主机的配置参数

monitor_c 类: 获取系统监控数据类封装, 系统数据信息

network_c 类: 网络信息类封装, 网络信息各项参数

overview_c 类: 复位服务器数据类, 服务器各项参数

scan_range_c 类: 获取雷达角度扫描范围封装类, 雷达扫描起始位置和结束位置

LakiBeamHTTP 类: 雷达配置文件类, 可以配置雷达所有的运行参数

4.2 函数说明

4.2.1 delete_override()

bool LakiBeamHTTP::delete_override (string & result)

删除静态模式配置并设置为 DHCP 模式

参数:

result: 本次操作的结果

返回值:

true: 操作成功

false: 操作失败

4.2.2 get_filer_level()

bool LakiBeamHTTP::get_filter_level (filter_c & level)

设置激光扫描结束角度

参数:

level:滤波器信息结构体返回值

返回值:

true:参数信息操作成功

false:参数信息操作失败

4.2.3 get_firmware()

bool LakiBeamHTTP::get_firmware (firmware c & firmware)

获取雷达固件信息

参数:

firmware: 雷达固件信息的引用类参数引用

返回值:

true: 操作成功

false: 操作失败

4.2.4 get_host()

bool LakiBeamHTTP::get_host (host c & host)

获取当前雷达 Host 端 IP 配置

参数:

host: Host 端 IP 配置信息返回值

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.5 get_host_IP()

bool LakiBeamHTTP::get_host_IP (string & IP)

获取当前雷达 Host 端 IP 地址

参数:

IP: 雷达 Host 端 IP 地址返回值

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.6 get_host_port()

bool LakiBeamHTTP::get_host_port (string & port)

获取当前雷达 Host 端 IP 地址端口号

参数:

port: Host 端 IP 地址端口号返回值

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.7 get_laser_enable()

bool LakiBeamHTTP::get_laser_enable (string & laser state)

获取激光状态:

参数:

motor rpm: 当前激光状态返回值 (true: 使能 false: 失能)

返回值:

true: 操作成功

false: 操作失败

4.2.8 get_laser_start()

bool LakiBeamHTTP::get_laser_start (string & start)

获取激光扫描起始角度

参数:

start: 雷达扫描起始角度返回值 (单位: 度)

返回值:

true: 操作成功

false: 操作失败

4.2.9 get_lase_stop()

bool LakiBeamHTTP::get_laser_stop (string & stop)

获取激光扫描结束角度

参数:

start: 雷达扫描结束角度返回值 (单位: 度)

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.10 get_monitor()

bool LakiBeamHTTP::get_monitor (monitor c & monitor)

获取雷达监测信息

参数:

monitor: 获取系统监控数据, 包含系统平均负载、 内存使用率和系统运行时间

返回值:

true: 操作成功

false: 操作失败

4.2.11 get_motor_rpm()

bool LakiBeamHTTP::get_motor_rpm (string & motor rpm)

获取雷达实时转速

参数:

motor_rpm: 当前雷达转速返回值 (单位: rpm)

返回值:

true: 操作成功

false: 操作失败

4.2.12 get_network()

bool LakiBeamHTTP::get_network (network c & network)

获取网络信息

参数:

network: 网络信息返回值, 包含以太网链路状态、双工模式、MAC 地址、主机名称、IPv4 信息以及以太网传输速率

返回值:

true: 操作成功

false: 操作失败

4.2.13 get_overview()

bool LakiBeamHTTP::get_motor_rpm (string & motor rpm)

获取雷达实时转速

参数:

overview: 包含雷达扫描频率、雷达实时转速、激光开关状态、分辨率、扫描角度范围、滤波等级和 Host 端 IP 地址及端口号

返回值:

true: 操作成功

false: 操作失败

4.2.14 get_resolution()

bool LakiBeamHTTP::get_resolution (string & resolution)

获取雷达当前水平角分辨率

参数:

resolution: 雷达水平角分辨率返回值, 各扫描频率对应的水平角分辨率分别为: 10Hz/0.1° , 20Hz/0.25° , 25Hz/0.25° , 30Hz/0.25°

返回值:

true: 操作成功

false: 操作失败

4.2.15 get_scan_range()

bool LakiBeamHTTP::get_scan_range (scan range c & scan range)

获取雷达扫描范围

参数:

scan_range: 雷达扫描范围返回值 单位: 度

返回值:

true: 操作成功

false: 操作失败

4.2.16 get_scanfreq()

bool LakiBeamHTTP::get_scanfreq (string & result)

获取雷达扫描频率

参数:

result: 当前雷达扫描频率返回值 (单位: Hz)

返回值:

true: 操作成功

false: 操作失败

4.2.17 put_filter_level()

bool LakiBeamHTTP::put_filter_level (string & level)

设置当前雷达数据滤波器件等级

参数:

level: 设置滤波器级别, 分为 0~3 共 4 个等级, 0 为关闭

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.18 put_host_IP()

bool LakiBeamHTTP::put_host_IP (string & IP)

设定当前雷达 Host 端 IP 地址

参数:

IP: 雷达 Host 端 IP 地址设定值

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.19 put_host_port()

bool LakiBeamHTTP::put_host_port (string & IP)

设定当前雷达 Host 端 IP 地址端口号

参数:

IP: Host 端 IP 地址端口号设定值

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.20 put_laser_enable()

bool LakiBeamHTTP::put_laser_enable (string & config state)

切换激光状态

参数:

config_state: 激光状态设定值 (true: 使能 false: 失能)

返回值:

true: 操作成功

false: 操作失败

4.2.21 put_laser_start()

bool LakiBeamHTTP::put_laser_start (string & start)

设置激光扫描起始角度

参数:

start: 设置雷达扫描起始角度设定值 范围: 0 度~ 315 度 注意: 设定的起始角度 应小于结束角度, 否则设定失败

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.22 put_laser_stop()

bool LakiBeamHTTP::put_laser_stop (string & stop)

设置激光扫描结束角度

参数:

stop: 设置雷达扫描结束角度设定值 范围: 0 度~ 315 度 (注意: 设定的起始角度 应小于结束角度, 否则设定失败)

返回值:

true: 参数信息操作成功

false: 参数信息操作失败

4.2.23 put_override()

bool LakiBeamHTTP::put_override (string override, string & result)

获取当前静态 IP 重写值

参数:

override: 静态 IP 重写地址设定值, 当 DHCP 模式打开时操作有效, 否则为无效操作

result: 本次操作的结果或者错误信息

返回值:

true: 操作成功

false: 操作失败

4.2.24 put_reset()

bool LakiBeamHTTP::put_reset (string & result)

系统复位

参数:

result: 本次操作的结果

返回值:

true: 操作成功

false: 操作失败

4.2.25 put_scanfreq()

bool LakiBeamHTTP::put_scanfreq (string & freq)

设置雷达扫描频率

参数:

freq: 设置扫描频率设定值 (设定选项: 10 20 25 30 推荐值: 25)

返回值:

true: 操作成功

false: 操作失败

5 SDK 开发流程

5.1 激光雷达主动上传数据模式流程

上位机获取 LakiBeam 雷达的点云数据只有雷达主动上传数据这一种模式。雷达主动上传数据模式,指的是根据雷达中关于 HostIP 和 DataPort 的配置,设置与雷达相连的网口 IP,根据配置界面中 HostIP 和端口号打开接收数据的端口,根据雷达 IP 连接雷达,从而获取雷达主动上传数据。

要点:

- (1) 知晓雷达配置界面中的 HostIP、DataPort 信息;
- (2) 设置与雷达相连的网卡 IP 为配置界面中的 HostIP;
- (3) 根据雷达配置界面中的 HostIP 和 DataPort, 设置接收数据的端口。

雷达上电初始,默认的数据传输方式是主动上传数据模式,此时上位机可以采用雷达主动上传数据模式方案获取数据。在雷达主动上传数据模式流程下,可以通过 Type-c 数据线连接雷达,浏览器登录雷达配置界面查看获取雷达中的 HostIP、DataPort 配置信息,如图 5.1 所示。

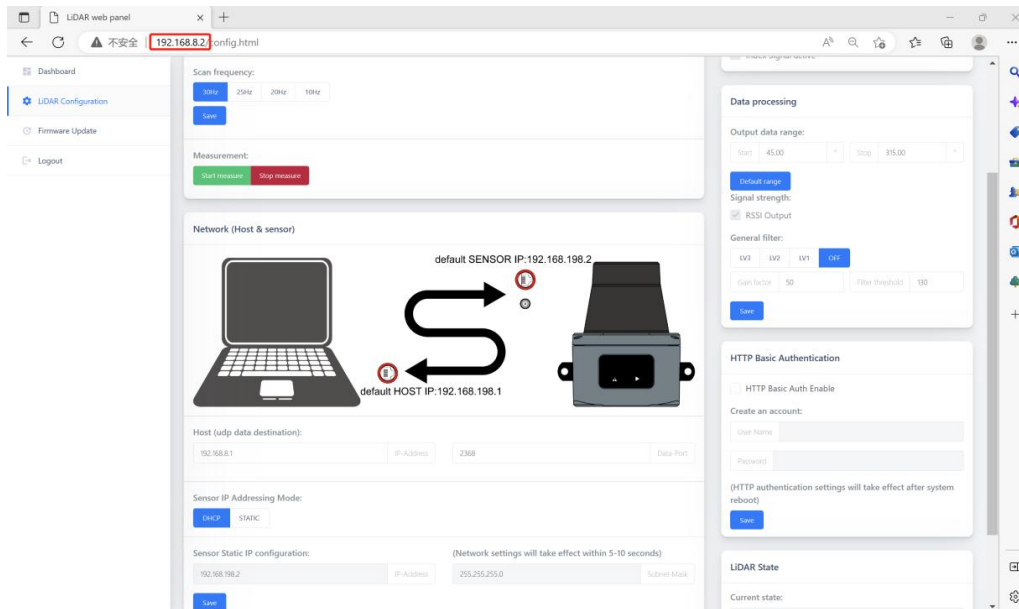


图 5.1: 激光雷达配置界面

当需要修改激光雷达配置信息时, 打开如图 5.1 所示的配置接口, 修改激光雷达网络配置, 要求 HostIP 和雷达 IP 在同一网段。

初始激光雷达的默认设置如下：

使用过程：

- (1) 建立设备对象，更改 HostIP 和数据端口，获取激光雷达数据；
- (2) 根据激光雷达的 IP，连接到激光雷达，并请求数据传输；
- (3) 设置激光雷达参数；
- (4) 轮询获取数据。