

# 山东科力 KLM 激光扫描仪 ROS 包

## 使用说明

该软件包是科力激光扫描仪的 ROS 驱动包。包含了从扫描仪获取数据，设置扫描仪参数等功能。该驱动包会发布标准的 `sensor_msgs::LaserScan` 消息。应用程序可以通过订阅该消息以获取扫描结果。

该说明阐述了怎样获取代码，怎样编译代码，怎样显示扫描结果以及怎样进行代码调试等。

### 获取代码

最新的 ROS 包代码可以从 [https://gitee.com/keli\\_tech31/sdkeli\\_klm\\_udp.git](https://gitee.com/keli_tech31/sdkeli_klm_udp.git) 获取。我们会不定时的将更新代码推送上去。

你直接获取最新代码的 ZIP 包。当然也可以通过 `git clone` 的方式获取代码，但在这之前你可能需要一个码云的账号。具体步骤如下：

1. 进入[~/catkin\_ws/]目录。
2. 执行[git init]。
3. 执行[git remote add origin [https://gitee.com/keli\\_tech31/sdkeli\\_klm\\_udp.git](https://gitee.com/keli_tech31/sdkeli_klm_udp.git)]
4. 执行[git clone [https://gitee.com/keli\\_tech31/sdkeli\\_klm\\_udp.git](https://gitee.com/keli_tech31/sdkeli_klm_udp.git)]

### 编译代码及运行

代码的编译相对简单，按照通常的 ROS 包编译步骤即可：

1. 进入[~/catkin\_ws/]目录。
2. 执行[catkin\_make]。如果 ROS 安装正确，应该可以正常编译通过。有问题请检查 ROS 安装是否正确。
3. 执行[source devel/setup.sh]，以保证 ROS 被正确的添加到了 ROS Package。
4. 你也可以执行[rospack list]来查看 sdkeli\_klm\_udp 包是否已经被正确添加。
5. 至此，ROS 包已经准备完毕。可以通过相应的 launch 文件运行我们的 ROS 驱动包。

6. 若是在编译过程中出现错误，有可能驱动包文件的权限不足，请尝试在 `catkin_ws/src` 目录下使用 `[sudo chmod 775 -R sdkeli_klm_udp/]` 指令修改权限，或者根据错误提示单独修改相应文件的权限。
7. 若是需要关闭雷达回波强度信息显示，可以在 `[~/catkin_ws/src/sdkeli_klm_udp/cfg/SDKeliKlm.cfg]` 中 “intensity” 参数配置为 `False`，修改之后需要重新编译才能生效。
8. Launch 文件说明：`sdkeli_klm_udp_with_1_lidar.launch` 为普通方式使用一个激光雷达；`sdkeli_klm_udp_with_2_lidars.launch` 为普通方式同时使用两个雷达；`sdkeli_klm_udp_nodelet_with_1_lidar.launch` 为 nodelet 方式使用一个雷达；`sdkeli_klm_udp_nodelet_with_2_lidars.launch` 为 nodelet 方式同时使用两个雷达。
9. 若是需要使用 nodelet 方式，并且需要使用其他 manager 请按照相应的 launch 文件指示修改相应的位置。
10. Launch 文件中默认使用了 topic remap 功能，单雷达工作默认的 topic 名称为 `keli_scan`，若要使用 RVIZ 查看图形显示扫描结果，需要注意选择正确的 topic。

为正确运行 ROS 包以前，你可能需要配置网络链接，以保证 ROS 能正确的从扫描仪获取数据：

1. 进入 `[~/catkin_ws/src/sdkeli_klm_udp/launch]` 目录。
2. 打开相应的 launch 文件，修改对应的 `[hostname]` 及 `[port]` 参数，具体的值请参照产品说明书。

## 使用 RVIZ 查看扫描结果

在 ROS 中，有很多方式可以查看扫描结果。在这里我们使用 RVIZ 来查看并调试扫描结果。具体步骤：

1. 启动扫描仪和 ROS 程序。
2. 在另外一个终端执行[roslaunch rviz rviz]以启动 RVIZ。
3. 在 RVIZ 左边的[Displays]窗口下面点击[Add]。
4. 在弹出的窗口选择[By Topic]页。
5. 选择[/keli\_scan]下面的[LaserScan]。然后点击 OK。
6. 回到[Displays]窗口，将[Global Options]下面的[Fixed Frame]值设置为[laser]。

至此，扫描图形应该能显示在 RVIZ 的视图窗口了。

## 关于参数配置

在 ROS 包中，可以有两种 ROS 参数配置方式：configure 和 launch 的参数配置。其中，configure 通常用来配置扫描仪的一些基本信息；而 launch 用来配置启动参数。

configure 配置的值可以在[~/catkin\_ws/src/sdkeli\_klm\_udp/cfg/SDKeliKlm.cfg]中直接修改，修改之后需要重新编译才能生效。实际上，由于我们使用了 ROS 动态配置的功能，我们还可以通过[roslaunch rqt\_reconfigure rqt\_reconfigure]方式修改 configure 配置。比如，调整扫描角度，以及启用/禁用 debug 等。

launch 的配置主要用于配置启动参数，需要注意的是，修改完成后需要重新启动 ROS 包。

## 获取调试信息

调试信息包括 debug 信息和 ROS 的调试信息。目前 debug 仅包括从扫描仪获取的扫描结果。debug 信息在 configure 配置中启用/禁用。因此，我们可以使用[roslaunch rqt\_reconfigure rqt\_reconfigure]来动态启用/禁用 debug 信息。

ROS 的调试信息可以使用[rqt\_logger\_level]来动态调整 log 级别(Debug, Info, Warn, Error, Fatal)。默认的级别是 Info，如果你想显示更多的信息可以通过[rqt\_logger\_level]来动态调整。

## 在自己的 ROS 程序中订阅扫描数据

前面已经提到，sdkeli\_klm\_udp 发布的是标准的 sensor\_msgs::LaserScan 消息。我们可以通过订阅获取 sensor\_msgs::LaserScan 消息。sdkeli\_klm\_udp 中发布 sensor\_msgs::LaserScan 的 topic 是 scan。也就是说，我们通过订阅 scan 的 topic 就能收到 sdkeli\_klm\_udp 发送的消息：

```
int main(int argc, char **argv)
{
    /*Initialize ROS system and node*/
    ros::init(argc, argv, "sdkeli_klm_demo");
    ros::NodeHandle nhSDKLKlmMsg;
    ros::Subscriber subSDKLKlmMsg = nhSDKLKlmMsg.subscribe("scan", 100, &OnSDKLKlmMessageReceived);

    /*Let ROS take over*/
    ros::spin();
}
```

下面是关于处理数据（OnSDKLKlmMessageReceived）的一个简单实例：

```
void OnSDKLKlmMessageReceived(const sensor_msgs::LaserScan &msg)
{
    ROS_INFO("sdkeli_klm - Laser Scan message received:");
    ROS_INFO("sdkeli_klm - frame_id: %s", msg.header.frame_id.c_str());
    ROS_INFO("sdkeli_klm - timestamp: %d.%d", msg.header.stamp.sec, msg.header.stamp.nsec);
    ROS_INFO("sdkeli_klm - angle_min: %f", msg.angle_min);
    ROS_INFO("sdkeli_klm - angle_max: %f", msg.angle_max);
    ROS_INFO("sdkeli_klm - angle_increment: %f", msg.angle_increment);
    ROS_INFO("sdkeli_klm - time_increment: %f", msg.time_increment);
    ROS_INFO("sdkeli_klm - scan_time: %f", msg.scan_time);
    ROS_INFO("sdkeli_klm - range_min: %f", msg.range_min);
    ROS_INFO("sdkeli_klm - range_max: %f", msg.range_max);

    /*Dump data...*/
    int dataCount = msg.ranges.size();
    ROS_INFO("sdkeli_klm - Data Count: %d", dataCount);
    for(int index = 0; index < dataCount; index++)
    {
        if(((index + 1) % 60 == 0) && (index != 0))
        {
            printf("\n");
        }

        printf("%.2f ", msg.ranges[index]);
    }

    printf("\n\n");
}
```

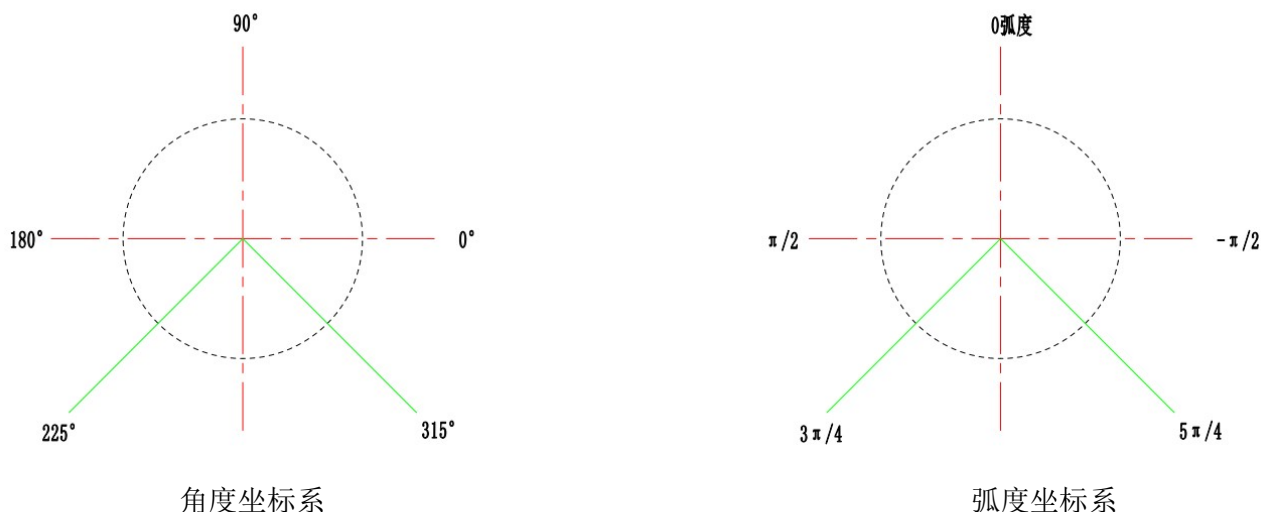
实例代码已经上传到了 git 库 [https://gitee.com/keli\\_tech31/sdkeli\\_klm\\_demo.git](https://gitee.com/keli_tech31/sdkeli_klm_demo.git)。该实例代码仅做参考，用户需要根据自己的业务逻辑实现相应的数据处理。

## 调整扫描角度

这里调整的扫描角度是调整 ROS 包的输出角度。默认的角度是 0 度到 360 度。默认配置定义在 configure 配置中[~/catkin\_ws/src/sdkeli\_klm\_udp/cfg/SDKeliKlm.cfg]。我们可以通过[roslaunch rqt\_reconfigure rqt\_reconfigure]来对其进行动态调整。需要注意的是，我们的配置值是角度，默认范围是[-1.570796 ~ 4.712388] ( $-0.5\pi \sim 1.5\pi$ )，对应的范围即 0 度到 360 度。

在 rqt\_reconfigure 中拖放起始角度和终止角度范围就能很清楚的看见 RVIZ 中的扫描范围发生了变化。当然，我们也可以在[~/catkin\_ws/src/sdkeli\_klm\_udp/cfg/SDKeliKlm.cfg]进行手动配置。

需要注意的是，我们的扫描仪的扫描范围在 0 度到 360 度之间，按照逆时针方向扫描。其中的 0 度和 360 度是角度值。而标准的 sensor\_msgs::LaserScan 消息要求的是弧度值，以正前方为 0 弧度。因此，我们需要将角度转化为对应的弧度。下面是角度与弧度的坐标系：



由上面可见，角度和弧度的坐标差了 90 度(即： $\pi/2$  个弧度)。我们在进行角/弧度转换的时候首先需要进行坐标转换。下面是角度转换成弧度的公式：

$$\text{弧度值} = ((\text{角度值} \times \pi) / 180) - (\pi/2)$$

其中， $\pi/2$  表示角度到弧度的坐标平移。同理，弧度转化为角度的公式为：

$$\text{角度值} = ((\text{弧度值} \times 180) / \pi) + 90$$

因此，假设我们需要将扫描角度调整为 0 角度到 180 角度。那么计算方法如下：

$$\text{起始弧度} = ((0 \times \pi) / 180) - (\pi / 2) = -\pi/2;$$

$$\text{终止角度} = ((180 \times \pi) / 180) - (\pi / 2) = \pi/2;$$