

PUESTA EN FUNCIONAMIENTO DEL SENSOR RPLIDAR A2 M8 USANDO LINUX Y PYTHON

SET UP OF RPLIDAR A2 M8 SENSOR USING LINUX AND PYTHON

Pedro Correa

Tecnológico Nacional de México en Celaya
D1803025@itcelaya.edu.mx

Alejandro Barranco

Tecnológico Nacional de México en Celaya
israel.barranco@itcelaya.edu.mx

Francisco Pérez

Tecnológico Nacional de México en Celaya
francisco.perez@itcelaya.edu.mx

Pedro Bautista

Tecnológico Nacional de México en Celaya
D1803004@itcelaya.edu.mx

Horacio Rostro

Universidad de Guanajuato
hrostrog@ugto.mx

Resumen

El sensor RPLIDAR A2 M8 de la marca SLAMTEC, mediante el uso de las propiedades del láser, es capaz de detectar objetos estáticos y en movimiento en un rango de 360°. Este dispositivo viene con un software basado en lenguaje C para acceder a todas sus funciones (el SDK *Open-Source* y un programa Demo) y que permite su implementación en múltiples plataformas como Windows, Linux, MacOS. Estas ventajas se reflejan en la creación de sistemas de detección de obstáculos en vehículos autónomos. En este trabajo se presenta una manera de configurar el LIDAR con PYTHON sobre Linux.

Palabra(s) Clave: Autónomos, Linux, Python, RPLIDAR, Sensor.

Abstract

The RPLIDAR A2 M8 sensor of the SLAMTEC, use the properties of the laser to detect static and moving objects in a range of 360°; This device comes with a

software based on C language to access all its functions (the Open-Source SDK) and allows its implementation on several platforms such as Windows, Linux and MacOS. These advantages are reflected in the creation of obstacle detection systems in autonomous vehicles. In this paper we present a way to configure LIDAR with PYTHON over Linux.

Keywords: *Autónomos, Linux, Python, RPLIDAR, Sensor.*

1. Introducción

Para la detección de objetos en movimiento, como por ejemplo obstáculos que se acercan a un automóvil, existe varios tipos de sensores que funcionan bajo distintos tipos de principios físicos entre ellos están los sensores LIDAR que funcionan aprovechando las propiedades del láser.

El sensor RPLIDAR A2 tiene como emisor un rayo láser de alta intensidad que sale del sensor y recorre el espacio hasta chocar con un objeto cuya superficie refleja el láser. El receptor del sensor detecta el rayo reflejado y mediante el conocimiento de la constante de la velocidad de la luz se logra calcular la distancia a la que se encuentra el objeto (RPLIDAR, 2014). Este sensor cuenta con un motor, cuyos parámetros pueden ser consultados en el *datasheet* (RPLIDAR A2, 2014), el cual gira 360° para proyectar el láser alrededor de todo el sensor.

El propósito de este documento es describir, a modo de instructivo, una forma de utilizar el sensor RPLIDAR A2 utilizando plataformas basadas en Linux que ayuden a cumplir con el objetivo de integrar este tipo de dispositivos en aplicaciones de navegación autónoma para vehículos terrestres.

El sensor RPLIDAR es comúnmente utilizado para aplicaciones en interiores como se muestra en (H. Singh Grewal, 2018), donde se desarrolla un sistema mediante el sensor láser y una cámara estéreo que permite la detección de las paredes y bordes del piso de una casa para permitir el desplazamiento autónomo de una silla de ruedas. Una aplicación similar se observa en (S. Ho Yeon, 2017), donde el sensor LIDAR se usa para la detección de pelotas de tenis y determinar la distancia a la que se encuentran de un robot que las recolecta de forma autónoma, este robot también se apoya de una cámara web y sensores ultrasónicos.

Este sensor también puede utilizarse en exteriores proporcionando buena respuesta en la detección de objetos y bordes del camino, como se muestra en (J. Kolarik, 2017), donde se implementa un algoritmo difuso para la fusión de datos de sensores ultrasónicos, unidades de medición inercial y el RPLIDAR con el que se logra el movimiento autónomo de un carro a escala.

2. Métodos

Para obtener datos del sensor LIDAR mediante el uso de Python y Linux es necesario hacer algunos ajustes previos al software, estos requerimientos previos se explican a continuación:

a) Software:

- Linux: es necesario tener instalado cualquiera de los sistemas operativos basados en Linux, se recomienda Ubuntu, Debian o Raspbian (si se está trabajando con la tarjeta Raspberry Pi).
- Python: se recomienda usar para la aplicación usar Python 3.5 o superior para garantizar las compatibilidades entre las librerías.

b) Librerías Python 3: Para lograr iniciar el sensor LIDAR y adquirir los datos del mismo usando el Python se deben instalar las siguientes librerías:

- Numpy: Es un paquete hecho para Python el cual permite la computación científica y la integración con el lenguaje C++ (Scipy, 2018). Para instalar este paquete se debe acceder a una terminal de Linux y teclear el siguiente comando:

```
sudo apt-get install python3-numpy
```

- Serial: El paquete serial permite acceder a todos los puertos serial de nuestro Hardware desde Python; es necesaria su instalación para aprovechar el uso del adaptador USB que trae el sensor LIDAR. Para instalarlo se debe teclear en una terminal de linux el siguiente comando:

```
sudo apt-get install python3-serial
```

- Matplotlib: Esta librería permite mostrar los datos adquiridos de forma gráfica y en varios formatos y sistemas coordenados. Para instalarlo se debe teclear en una terminal de linux el siguiente comando:

sudo apt-get install python3-matplotlib

- Rplidar-Master: Esta librería es la que permite controlar el sensor RPLIDAR desarrollada principalmente para Python 3 (SkoltechRobotics, 2017). Para su instalación en Linux se deben seguir el siguiente instructivo:

- ✓ Después de instalar las librerías anteriores se recomienda siempre actualizar los paquetes de Linux con los comandos típicos, para evitar incompatibilidades en el sistema y liberar espacio:

sudo apt-get upgrade

sudo apt-get update

- ✓ Luego de actualizar el Sistema se debe ir al navegador de internet y entrar al siguiente enlace:

<https://github.com/SkoltechRobotics/rplidar>

En este sitio se encuentra la librería Rplidar-master desarrollada por Artyom Pavlov, la cual cuenta con licencia MIT que permite su libre distribución y uso. De sede descargar la librería.

- ✓ Luego de descargar la librería, se va a la carpeta de la descarga, se descomprime el archivo ZIP y se ingresa a la carpeta de la librería. Luego de eso se da clic derecho dentro de la carpeta y elige la opción “abrir en terminal”.
- ✓ Una vez abierta la carpeta en una terminal de Linux se debe teclear el siguiente comando para instalar la librería dentro del entorno de Python 3:

Python setup.py install

Python3 setup.py install

- ✓ Una vez cumplidos los pasos anteriores la librería ya debería estar funcionando y para comprobarlo se puede abrir el IDLE de Python 3 y teclear:

```
import rplidar
```

Si todo está bien no debería aparecer ningún error.

- c) **Puesta en marcha:** Se debe conectar el puerto UART que tiene el sensor al correspondiente lado del adaptador USB (figura 1) que trae de fábrica, así mismo mediante un cable que convierte de micro USB a USB se hace la conexión con el computador o Raspberry que estemos usando.

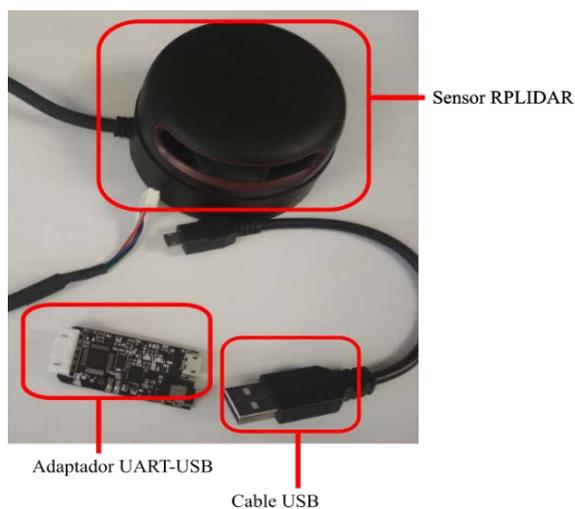


Figura 1 Sensor RPLidar y sus componentes.

Una vez conectado el sensor al puerto USB y antes de utilizar Python y la librería rplidar-master, se debe liberar dicho puerto de cualquier otra actividad que Linux le esté dando para que Python pueda acceder a él libremente; esto debe hacerse tecleando en una terminal de Linux el siguiente comando:

```
sudo chmod 777 /dev/ttyUSB0
```

Si este paso no se realiza cualquier Script que se ejecute desde Python para usar el puerto USB dará error. Cabe resaltar que el nombre ttyUSB0 es el que Linux le asigna siempre a la conexión con el sensor LIDAR.

Una vez cumplidos los procedimientos anteriores y se puede empezar con el uso de la librería y del sensor. En la figura 2 se muestra el dispositivo en funcionamiento en un computador con sistema operativo Ubuntu 18.04 y con Python 3.6.5.

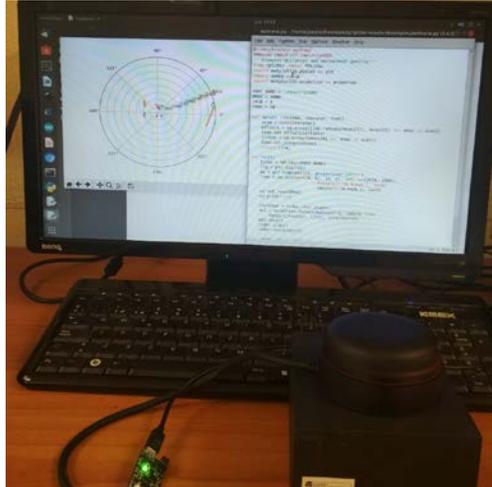


Figura 2 Conexión sensor RPLidar con Linux y Python.

4. Resultados

Dentro de la carpeta de la librería Rplidar-master esta la sub carpeta *examples* donde se encuentran varios *scripts* de Python a modo de ejemplos con los que se puede ver el funcionamiento del sensor. Principalmente el ejemplo llamado: *animate.py*, el cual muestra, en el sistema coordenado polar, las mediciones del sensor en un rango de 360° (figura 3).

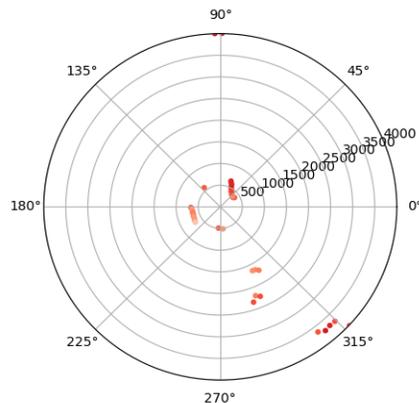


Figura 3 Funcionamiento Librería Rplidar-master

Para probar el funcionamiento del sensor se ubicó a una persona a 3.5 m de distancia frente al sensor y se fue acercando; de esta forma se pudo medir la distancia a la que el sensor LIDAR detecta a la persona. En la figura 4 se observan tres puntos, sobre el eje de 0° y cerca del círculo de radio de 3500 mm, que representan a la persona que está ubicada justamente a esa distancia (es decir 3.5 m).

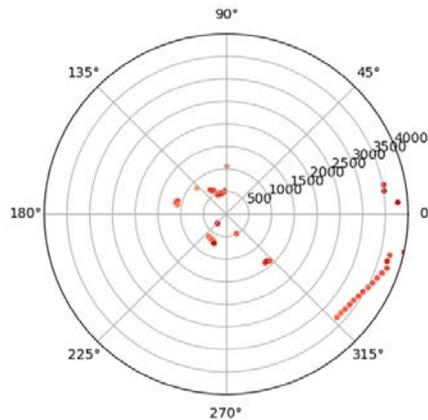


Figura 4 Detección de obstáculos a 3.5 m.

Luego la persona se acercó a dos metros del sensor y el resultado fue el mostrado en la figura 5.

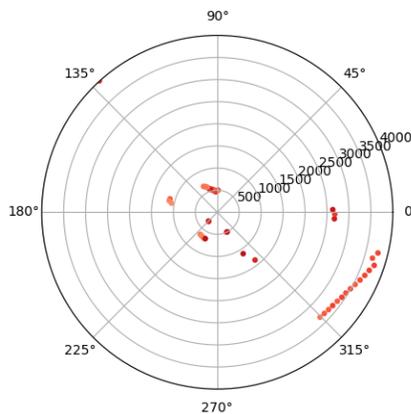


Figura 5 Detección de obstáculos a 2 m.

Finalmente, la persona se ubicó a un metro del sensor y a medida que se va acercando la persona el sensor es capaz de detectar más puntos que se reflejan

en la superficie del cuerpo de la persona; como se muestra en la figura 6, la persona está representada por el conjunto de puntos que esta sobre el eje 0° y cerca del círculo de radio 1000 mm.

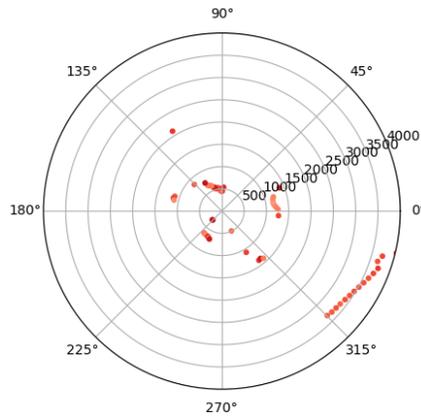


Figura 6 Detección de obstáculos a 1 m.

5. Discusión

La librería Rplidar-master para Python posee una gran resolución al momento de medir distancia ya que entrega los datos en milímetros. Además, al ser una librería de código abierto puede ser modificada tanto como se necesite dependiendo de la aplicación en la que se quiera implementar.

De acuerdo a lo visto en las figuras 4, 5 y 6, se hace necesaria una modificación a la librería Rplidar-master para ajustar la velocidad del motor del sensor, así como la cantidad de puntos que toma en cada giro ya que la configuración que viene por defecto dificulta la reconstrucción 2D del entorno que rodea al sensor.

El controlar este tipo de sensores mediante plataformas basadas en Linux presenta una gran ventaja al momento de desarrollar aplicaciones portables (o móviles) como las que se requieren para la construcción de vehículos autónomos aéreos o terrestres; esto, debido a que tarjetas de desarrollo como la Raspberry Pi 3 usan mediante sistemas operativos de Linux y tienen un poder de procesamiento capaz de procesar flujos de datos de mediana densidad como los entregados por el RPLIDAR A2.

El propósito principal de la implementación del sensor RPLIDAR es llegar a la fusión de datos con otros sensores de diferente naturaleza (o principio físico) para

lograr el movimiento autónomo de un vehículo terrestre y es por eso que se necesita aumentar la cantidad de puntos que se toman del ambiente al máximo posible. Para realizar modificaciones a la Rplidar-master es necesario consultar los *datasheets* del dispositivo vistos en (RPLIDAR, 2014)

6. Bibliografía y Referencias

- [1] RPLIDAR, low cost 360 degree 2D laser Scanner (LIDAR) System, Introduction and Datasheet, 2014-4-17, rev-6.
- [2] RPLIDAR A2: Low Cost 360 Degree Laser Range Scanner, Introduction and Datasheet, A2M8, 2014.
- [3] Scipy (2018), Installing packages, Scipy.org: <https://www.scipy.org/install.html>.
- [4] SkoltechRobotics (2017), Python module for RPLidar A1 and A2 rangefinder scanners, GitHub: <https://github.com/SkoltechRobotics/rplidar>.
- [5] Matplotlib version 3.0 (2002-2012), Python Plotting-Matplotlib v3.0 documentation, Matplotlib: <https://matplotlib.org/>.
- [6] SLAMTEC (2016-2018), Leading service Robot Localization and Navigation Solution Provider, SLAMTEC: <https://www.slamtec.com/en/Lidar/A3>.
- [7] J. Kolarik, Z. Slanina; "Fuzzy Control Application in Swarm Robotics Cars"; Department of Cybernetics and Biomedical Engineering, VSB Technical University of Ostrava; Ostrava, Czech Republic; 2017.
- [8] H. Singh Grewal, N. Thotappala Jayaprakash, A. Matthews; "Autonomous Wheelchair Navigation in Unmapped Indoor Environments"; Department of Computer Engineering; California State University; Fullerton, California 92831; 2018.
- [9] S. Ho Yeon, D. Kim; "System Design for Autonomous Table Tennis Ball Collecting Robot"; Department of Electrical and Computer Engineering; Seoul National University; Seoul, 08826, Korea; 2017.