

# INSTALACIÓN

## Requisitos previos:

- Sistema Operativo: Ubuntu 18.04.
- ROS. Versión según SO. Para 18.04, ROS Melodic.  
<https://wiki.ros.org/melodic/Installation>
- Cmake 3.0.0+.
  - Actualizar cmake con: `$ sudo apt install cmake`. Si la versión no es la más reciente, descargar desde la web: <https://cmake.org/download/> y seguir los pasos del README para lograr la instalación.
- gcc 4.8.1+

## Pasos para la instalación de FAST LIO:

### 1 Instalación de Livox SDK2:

- 1.1 Compilar e instalar Livox-SDK2:  
`$ git clone https://github.com/Livox-SDK/Livox-SDK2.git`  
`$ cd ./Livox-SDK2/`  
`$ mkdir build`  
`$ cd build`  
`$ cmake .. && make -j`  
`$ sudo make install`

### 2 Instalación de Livox ROS Driver:

- 2.1 Instalar en una carpeta /src dentro de un workspace:  
`$ git clone https://github.com/Livox-SDK/livox_ros_driver.git`
- 2.2 Compilar el paquete:  
`$ source /opt/ros/$ROS_DISTRO/setup.sh`
- 2.3 En el directorio del workspace, realizar:  
`$ catkin_make`

### 3 Instalación de Livox ROS Driver2:

- 3.1 Instalar en una carpeta /src dentro de un workspace:  
`$ git clone https://github.com/Livox-SDK/livox_ros_driver2.git`
- 3.2 Compilar el paquete:  
`$ source /opt/ros/$ROS_DISTRO/setup.sh`  
`$ ./build.sh ROS1`
- 3.3 En el directorio del workspace, realizar:  
`$ catkin_make`

### 4 Instalación de FAST-LIO:

- 4.1 Instalar en una carpeta /src dentro de un workspace:  
`$ git clone https://github.com/hku-mars/FAST_LIO.git`  
`$ cd FAST_LIO`  
`$ git submodule update -init`
- 4.2 Modificar en config el archivo mid360.yaml, y ponerle la matriz de rotación adecuada para la inclinación de 30 grados que tiene:  
extrinsic\_R: [ 0.86, 0, 0.5,  
                  0, 1, 0,  
                  -0.5, 0, 0.86]
- 4.3 En el directorio del workspace, realizar:  
`$ catkin_make`

## Pasos para la instalación de FAST LIO LOCALIZATION:

### 1 Dependencias:

- 1.1 `$ sudo apt install ros-$ROS_DISTRO-ros-numpy`
- 1.2 `$ sudo pip install --upgrade pip`
- 1.3 `$ sudo pip install ipywidgets==7.6.3`
- 1.4 `$ sudo pip install --user open3d==0.9`
  - 1.4.1 Si al hacer:  
`$ python`  
`>>>import open3d`

Falla la importación de open3d, emplear:  
`sudo -H pip install --user open3d==0.9`

### 2 Instalación de FAST-LIO-LOCALIZATION:

- 2.1 `cd ~/$A_ROS_DIR$/src`
- 2.2 `git clone https://github.com/HViktorTsoi/FAST_LIO_LOCALIZATION.git`

- 2.3      `cd FAST_LIO_LOCALIZATION`
- 2.4      `git submodule update --init`
- 2.5      Modificar archivos:
  - 2.5.1    Modificar CMakeLists.txt, en las 3 líneas finales donde crea el módulo `fastlio_mapping` con el mismo nombre que FAST-LIO Cambiarlo de alguna forma, por ejemplo añadiendo `_l`. Si no se hace, fallará `catkin_make`.
  - 2.5.2    Modificar en scripts, `global_localization.py`:  
`FOV = 6.2831, FOV_FAR=70.`
  - 2.5.3    Copiar archivo modificado `mid360.yaml` de `ws_livox/src/FAST_LIO/config` a `ws_livox/src/FAST_LIO_LOCALIZATION/config`.
  - 2.5.4    Modificar archivo `localization_avia.launch` en `ws_livox/src/FAST_LIO_LOCALIZATION/launch` para incluir el archivo de configuración `mid360.yaml`, y actualizar el nodo al que llama de `fastlio_mapping` al nombre que le pusimos en CMakeLists.txt, y guardar como `localization_mid360.launch`. Adicionalmente, guardar como mapa por defecto el archivo `.pcd` que guardemos en carpeta PCD de LOCALIZATION. Ruta de ejemplo:  
`"/home/grvc/catkin_ws/src/FAST_LIO_LOCALIZATION/PCD/scans.pcd"`.  
 Alternativamente, se puede dejar como ruta:  
`"/home/grvc/catkin_ws/src/FAST_LIO/PCD/scans.pcd"`, para hacer que el último mapa creado con FAST-LIO sea el que se emplee para la localización.
- 2.6      `catkin_make`

### Puesta en marcha FAST-LIO:

En el directorio del workspace:

```
$ source devel/setup.sh
$ roslaunch fast_lio mapping_mid360.launch
$ roslaunch livox_ros_driver2 msg_MID360.launch
```

### Puesta en marcha FAST-LIO-LOCALIZATION:

En el directorio del workspace:

```
$ source devel/setup.sh
$ roslaunch fast_lio_localization localization_mid360.launch
$ roslaunch livox_ros_driver2 msg_MID360.launch
```

### FAQ:

1. *No hay errores de compilación, pero no se muestra la nube de puntos en Rviz, ni se publican datos en el topic `/livox/lidar`.*

La configuración de Ethernet de la conexión con el LIDAR debe coincidir con la especificada en el archivo `MID360_config.json`, en el directorio:

`ws_livox/src/livox_ros_driver2/config`.

Actualmente el archivo está configurado para que la dirección IP del host sea 192.168.1.5 y la del LIDAR sea 192.168.1.196.

2. *Tras cambiar la configuración manual de la conexión por cable, la dirección IP de la conexión no se actualiza, y al hacer `$ ping` a la dirección no se transmiten los paquetes.*

Es necesario desconectar y reconectar usando el botón presente en la ventana de configuración de wired network tras aplicar los cambios.

3. *Problemas de instalación de open3d.*

La solución requiere de estos comandos, aunque puede que hagan falta más.

```
sudo pip install --upgrade pip
sudo pip install ipywidgets==7.6.3
sudo pip install --user open3d==0.9
```

Si la instalación de open3d no se lleva a cabo, o si al importar en python2.7 no aparece, emplear:

```
sudo -H pip install --user open3d==0.9
```

### Enlaces de utilidad:

```
https://github.com/Livox-SDK/Livox-SDK2
https://github.com/Livox-SDK/livox\_ros\_driver2
https://cmake.org/download/
https://wiki.ros.org/melodic/Installation
https://github.com/Livox-SDK/livox\_ros\_driver2/issues?q=is%3Aissue+
```

[https://terra-1-g.djicdn.com/851d20f7b9f64838a34cd02351370894/livox%20mid%20360%20%E7%94%A8%E6%88%B7%E4%BD%BF%E7%94%A8%E6%89%8B%E5%86%8C240222/Livox\\_Mid-360\\_User\\_Manual\\_EN.pdf](https://terra-1-g.djicdn.com/851d20f7b9f64838a34cd02351370894/livox%20mid%20360%20%E7%94%A8%E6%88%B7%E4%BD%BF%E7%94%A8%E6%89%8B%E5%86%8C240222/Livox_Mid-360_User_Manual_EN.pdf)  
<https://github.com/Livox-SDK/LIO-Livox>  
[https://github.com/hku-mars/FAST\\_LIO](https://github.com/hku-mars/FAST_LIO)  
[https://github.com/HViktorTsoi/FAST\\_LIO\\_LOCALIZATION](https://github.com/HViktorTsoi/FAST_LIO_LOCALIZATION)  
[https://github.com/HViktorTsoi/FAST\\_LIO\\_LOCALIZATION/issues/32](https://github.com/HViktorTsoi/FAST_LIO_LOCALIZATION/issues/32)

## USO

1.- Abrir Terminal y 5 pestañas.

2.- En cada pestaña acceder al ordenador a bordo a través de la red de Optitrack:  
\$ ssh grvc@DRONE.IP

```
////////////////////
// .BASHRC MODIFICADO, NO HACE FALTA EL PASO 3:
3.- Lanzar los siguientes comandos para poder conectar los cores de ROS (EN TODAS LAS
PESTAÑAS DEL ORDENADOR A BORDO):
    $ export ROS_MASTER_URI=http://DRONE.IP:11311
    $ export ROS_IP=DRONE.IP
// (ESTA DIRECCIÓN IP ES LA DE LA LATTEPANDA EN EL WIFI DE OPTITRACK)
////////////////////
```

4.- En el ordenador a bordo hay que lanzar los siguientes comandos:

```
// Envío de mensajes del LIDAR:
    $ roslaunch livox_ros_driver2 msg_MID360.launch

// Paquete de localización. Se puede cambiar el mapa global con el argumento
map:=/ruta/al/archivo.pcd
    $ roslaunch fast_lio_localization localization_mid360.launch

// Publicar pose inicial, bien desde GCS con nodo mencionado en el punto 5, o con:
    $ rosrunc fast_lio_localization publish_initial_pose.py 0 0 0 0 0 0

// Nodo para traducir /Odometry a /mavros/vision_pose/pose
    $ rosrunc pose_publisher pose_publisher_node

// Lanzar mavros, es necesario dar permiso.
    $ sudo chmod 666 /dev/ttyUSB0
    $ roslaunch mavros apm3.launch

// Para monitorizar los nodos.
    $ rostopic list
    $ rostopic echo nodo_de_interés
```

5.- En la GCS (Estación de Control de Tierra) hay que lanzar 2 comandos:

```
// En todas las pestañas de la GCS, hay que lanzar:
    $ export ROS_MASTER_URI=http://DRONE.IP:11311
    $ export ROS_IP=GCS.IP
// (ESTA DIRECCIÓN IP ES LA DE MI GCS EN EL WIFI DE OPTITRACK)

// PESTAÑA 1:
    $ roslaunch waypoint_publisher waypoint_publisher.launch

// PESTAÑA 2:
    $ rosrunc odom_to_global odom_to_global_node
```