## NAME

## LibraryTT (Libreria)

PACKAGE CONTENTS

* Lidar3D
* Servo
* txt2array

FILE

LibraryTT/\_\_init\_\_.py

### NAME

### LibraryTT.Servo (Paquete)

CLASSES

builtins.object

Servo

### class Servo(builtins.object)

# | Servo(Angle) FUNCION

|

| Clase para el control de servomotor

| Este siempre se pondrá en 90° por default

| Parametros

| ----------

| Angle => [0,180]

|

| METODOS

|

| \_\_del\_\_(self)

| Destrucción del Objeto

|

| \_\_init\_\_(self, Angle)

| Initialize self. See help(type(self)) for accurate signature.

|

| setAngle(self, angle)

| Función para cambiar la posición del servomotor

| solo acepta valores entre 0 y 180° por seguridad

| Parametros

| ------------

| angle => [0,180]

|

| stop(self)

| Funcion para parar el servor

|

| ----------------------------------------------------------------------

### NAME

### LibraryTT.txt2array (Paquete)

METODOS

array2csv(data)

Generacion de un archivo csv de la nube de puntos

lidar en coordenadas cartesianas a partir de un array,

que se guardará en la carpeta "./Sets\_CSV"

@input LidarPoints\_array dimensions: nx3

@output N/A

array2txt(data)

Generacion de un archivo de texto \*.txt de la nube de puntos

lidar en coordenadas cartesianas a partir de un array,

que se guardará en la carpeta "./Sets"

@input LidarPoints\_array dimensions: nx3

@output N/A

csv2array(file\_path=None)

Lectura de nube de puntos [x,y,z] desde archivo CSV

a un array de la de dimensiones nX3 donde n=numero de puntos

@input file\_path \*.csv

@output array dimensions: nx3

imprimir3D(array)

Ploteo en 3D de un array con nube de puntos

Parametros

----------------

array dimensiones 3xn

txt2array(file\_path=None)

Lectura de nube de puntos [x,y,z] desde archivo de texto \*.txt

a un array de la de dimensiones nx3 donde n=numero de puntos

@input file\_path \*.txt

@output array dimenions: nx3

--------------------------------------------------------------------------------------------------------------------

### NAME

### LibraryTT.Lidar3D (Paquete)

CLASSES

builtins.object

Scaner3D

class Scaner3D(builtins.object)

| Clase para crea objeto de escaneo 3D para

| la adquisición de la nube de puntos LMS,

| mediante el LiDAR

|

| METODOS

|

| Scanear(self, Angulo\_Init=60, Angulo\_Fin=50, paso=-1, plotear=False)

| Función para escanear los puntos en el espacio

| Parametros

| ------------

| int Angulo\_Init Angulo Inicial de escaneo [0,180]°

| int ANgulo\_Fin Angulo Final de escaneo [0,180]°

| int paso paso de los angulos del servo para la reconstruccion

| bool plotear ¿Desea que se ploté lo escaneado?

|

| Regresa

| ------------

| array nx3 Array con la nube de puntos escaneada en forma cartesina

| de la forma [x,y,z]xn

|

| \_\_init\_\_(self)

| Inicialización de la clase para escaneo con

| le LiDAR

| Parametros

| ----------

| Ninguno

|

| ploteo3D(self)

| Función para plotear el último set de nube de puntos

| escaneado

| ----------------------------------------------------------------------

| METODOS ESTATICOS

|

| recorte(scan)

| Filtra los puntos de la parte frontal del LiDAR

|

| ----------------------------------------------------------------------