

---

# **YarpTp\_Module\_ Documentation**

**Kevin Pedroza Goenaga**

**20 de noviembre de 2018**



---

## Contenidos:

---

|   |          |
|---|----------|
| <b>1. Primeros pasos</b>                      | <b>3</b> |
| <b>2. Instalación</b>                         | <b>5</b> |
| 2.1. Requerimientos . . . . .                 | 5        |
| 2.2. Instalación . . . . .                    | 5        |
| <b>3. Métodos de movimiento</b>               | <b>7</b> |
| 3.1. Movimientos abiertos . . . . .           | 7        |
| 3.2. Movimientos cerrados . . . . .           | 7        |
| 3.3. Giros . . . . .                          | 7        |
| 3.4. Métodos de parada . . . . .              | 8        |
| <b>4. Métodos de configuración</b>            | <b>9</b> |
| 4.1. Desactivar control . . . . .             | 9        |
| 4.2. Definición de los pines a usar . . . . . | 9        |



YarpTp es un módulo para Python construido para un fácil control de la plataforma robótica YarpTp. Con él, cualquiera podrá construir scripts que ejecuten movimientos del robot de forma fácil.

En este espacio se encontrará toda la información para usar el módulo y de igual forma toda la referencia sobre la API del módulo.



# CAPÍTULO 1

---

## Primeros pasos

---

Un script sencillo para moverse hacia el frente puede ser así:

```
from YarpTp import YarpTp
my_car = YarpTp()
my_car.ForwardStep()
my_car.GoodBye()
```

¿Qué está haciendo?

1. En primer lugar se importa la clase `YarpTp` quien contiene los métodos de control.
2. Luego, se crea una instancia de la clase, es decir, un objeto que hará las veces del robot en el código.
3. Ahora es posible ejecutar cualquier método de movimiento, como se muestra al llamar a `ForwardStep()`, el cual se mueve cierto paso predefinido hacia delante. Para ver los demás movimientos, ir a la sección de movimientos.
4. Finalmente, hay que “apagar” el control del robot usando el método `GoodBye()`.





## 2.1 Requerimientos

El módulo está diseñado para usarse sobre la plataforma robótica YarpTp cuyo controlador es una Raspberry Pi. Por lo tanto, el módulo sólo tendrá efecto si se usa en la arquitectura de una Raspberry Pi.

La librería requerida por el módulo es:

- RPi.GPIO (versión mínima 0.6.3) módulo de python para el control de los pines GPIO.

## 2.2 Instalación

El paquete puede descargarse desde PyPi o usando el gestor de paquetes pip a través del comando:

```
$ pip install YarpTp
```



---

### Métodos de movimiento

---

Los movimientos soportados por el módulo incluyen el desplazamiento hacia delante, hacia atrás y los giros.

#### 3.1 Movimientos abiertos

Los métodos de movimiento que se presentan a continuación son ejecutados de forma libre y personalizada, como se muestra más adelante.

Cada uno de los movimientos puede ejecutarse de dos formas: movimiento continuo o personalizado:

- Cuando no se indica el parámetro del tiempo (tm) el movimiento será continuo hasta que una instrucción lo detenga y usando la velocidad indicada. Si la velocidad no se indica, se ejecuta el movimiento a la velocidad predefinida (50 %).
- Cuando se indica un tiempo del movimiento, se ejecutará hasta que transcurra y luego se detendrá. Para la velocidad, se comportará de la forma indicada en el punto anterior.

Los movimientos varían de acuerdo a la dirección (hacia delante o hacia atrás) y por el motor que se moverá (izquierdo, derecho o ambos motores). Los métodos se listan a continuación:

#### 3.2 Movimientos cerrados

Estos movimientos ocurren de forma limitada o definida, es decir, mueven al robot por un paso.

#### 3.3 Giros

Estos métodos permiten girar el robot hacia un lado.

## 3.4 Métodos de parada

Los siguiente métodos detienen el movimiento de los motores del robot.

---

### Métodos de configuración

---

#### 4.1 Desactivar control

Cuando se finalice el uso del robot, deben liberarse los pines de la Raspberry. Para esto, es importante que al final de cada script donde se controle el robot se ejecute el siguiente movimiento.

Al ejecutar el método anterior, no se podrá controlar más el robot. Sin embargo, si una nueva instancia de la clase `YarpTp` se crea usando la instrucción `YarpTp()` se podrá controlar de nuevo el robot.

#### 4.2 Definición de los pines a usar

Internamente, el control de las ruedas se da desde los pines GPIO de la Raspberry, los cuales están predefinidos en el módulo siguiendo la documentación de construcción del robot `YarpTp`. Sin embargo, si los pines son cambiados, se pueden usar los siguientes métodos: