

Métodos de la clase Dron

<pre>def connect(self, connection_string, baud, id=None, blocking=True, callback=None, params = None)</pre>	
	<p>Conecta con el dron. Los parámetros <code>connection_string</code>, <code>baud</code> indican si hay que conectar con el simulador o con el dron real, y la velocidad de comunicación. Para conectar con el simulador típicamente el string de conexión es: <code>'tcp:127.0.0.1:5763'</code> y la velocidad es 115200. La conexión admite un identificador para el dron, que la librería añadirá como primer parámetro en todas las funciones callback.</p>
<pre>def disconnect (self)</pre>	
	<p>Desconecta el dron. Además, detiene el envío de datos de telemetría y espera 5 segundos antes de retornar.</p>
<pre>def arm(self, blocking=True, callback=None, params = None)</pre>	
	<p>Arma el dron.</p>
<pre>def takeOff(self, aTargetAltitude, blocking=True, callback=None, params = None)</pre>	
	<p>Despega el dron hasta alcanzar la altura indicada en el parámetro.</p>
<pre>def startGo(self)</pre>	
	<p>Pone al dron en modo navegación, para que acepte las ordenes de navegar en dirección Norte, Sur, etc.</p>
<pre>def stopGo(self)</pre>	
	<p>Para el modo navegación. Es necesario hacer esto antes de ordenar otras operaciones (como por ejemplo dirigirse a un punto determinado).</p>
<pre>def changeNavSpeed (self, speed)</pre>	
	<p>Cambia la velocidad del dron cuando está en modo navegación.</p>
<pre>def go(self, direction)</pre>	
	<p>Hace que el dron navegue en la dirección indicada. Las opciones son: <code>'North'</code>, <code>'South'</code>, <code>'West'</code>, <code>'East'</code>, <code>'NorthWest'</code>, <code>'NorthEast'</code>, <code>'SouthWest'</code>, <code>'SouthEast'</code>, <code>'Stop'</code>.</p>
<pre>def getParams(self, parameters, blocking=True, callback=None)</pre>	
	<p>Pide al dron el valor de los parámetros indicados. En el caso de que la llamada sea o bloqueante, ejecutará la función del callback pasándole como parámetro la lista de valores recibida (y el identificador del dron como primer parámetro en el caso de que el dron haya sido identificado en el momento de la conexión). Este ejemplo muestra cómo se pasa la lista de parámetros (en formato json) cuyo valor se quiere recuperar y como se revuelve la lista de valores de esos parámetros.</p> <pre>parameters = json.dumps(["RTL_ALT", "PILOT SPEED UP",</pre>

	<pre>"FENCE_ACTION" }) result = dron.getParams(parameters) values = json.loads(result) print('Valores:', values)</pre> <p>El resultado podría ser:</p> <p>Valores: [{'RTL_ALT': -1.0}, {'PILOT_SPEED_UP': 100.0}, {'FENCE_ACTION': 4.0}]</p>
	<pre>def setParams(self, parameters, blocking=True, callback=None, params = None)</pre>
	<p>Establece el valor de los parámetros que se le pasan en una lista en formato json. Un ejemplo de uso es este:</p> <pre>parameters = json.dumps([{'ID': "FENCE_ENABLE", 'Value': 1}, {'ID': "FENCE_ACTION", 'Value': 4}]) dron.setParams(parameters)</pre>
	<pre>def RTL (self, blocking=True, callback=None, params = None)</pre>
	<p>Ordena al dron que retorne a casa (Return to launch).</p>
	<pre>def Land (self, blocking=True, callback=None, params = None)</pre>
	<p>Ordena al dron que aterrice en el punto que está sobrevolando.</p>
	<pre>def setGEOFence(self, fence waypoints, blocking=True, callback=None, params = None)</pre>
	<p>Envía al dron la configuración del geofence, que consiste en la lista de las coordenadas geográficas que delimitan el polígono (al menos 3 puntos), en formato json. Un ejemplo puede ser este:</p> <pre>fence_waypoints = json.dumps([{'lat': 41.123, 'lon': 1.988}, {'lat': 41.234, 'lon': 1.999}, {'lat': 41.150, 'lon': 1.870}]) dron.setGEOFence(fence_waypoints)</pre>
	<pre>def send_telemetry_info(self, process_telemetry_info)</pre>
	<p>Pide al dron que envíe los datos de telemetría. Cuando el dron tiene un nuevo paquete de telemetría llama al callback y le pasa ese paquete en formato json (y el identificador del dron como primer parámetro en el caso de que el dron haya sido identificando en el momento de la conexión). El dron va a enviar 4 paquetes por segundo. El ejemplo siguiente muestra el formato en que se reciben los datos de telemetría:</p> <pre>def process_telemetry_info(self, telemetry_info): print ('info: ', telemetry_info)</pre> <p>El resultado podría ser:</p> <pre>info:{'lat':41.276,'lon':1.988, 'alt': -0.016, 'groundSpeed': 0.120, 'heading': 355.88, 'state': 'connected'}</pre> <p>Los posibles estados en los que puede estar el dron son: 'connected', 'disconnected', 'arming', 'armed', 'takingOff', 'flying', 'returning', 'landing'.</p>

```
def stop_sending telemetry info(self)
```

Detiene el envío de datos de telemetría.

```
def goto(self,  
    lat, lon, alt,  
    blocking=True, callback=None, params=None)
```

Dirige al dron al punto geográfico indicado. Esta no funciona si estamos en modo navegación.