

casTelloPy DOC

BORRADOR / DRAFT - 02/2020

--- INSTANCIACIÓN ---

Tello()

__init__()

Ejecuta entre otros:

creacion y bindeo general de sockets

statesocket

clientsocket

thread1 = Thread(run_udp_receiver()).start()

Ejecuta entre otras cosas, el siguiente bucle infinito:

while True:

self.response = self.clientSocket.recvfrom(1024) # tamaño buffer 1024 bytes

thread2 = Thread(obtener_estados()).start()

Ejecuta entre otros:

while True:

self.response_state = self.stateSocket.recvfrom(256) # tamaño buffer 256 bytes

self.estados = objeto con atributos decodificando response_state

--- Atributos Importantes ---

--- General ---

stream_on

esta_volando

mpads_enabled

sdk

--- Video ---

cap

background_frame_read

--- Estados ---

obtencion_estados_habilitada

obtencion_estados_forzada

response_state - mensaje entero recibido del dron sin segmentar

bat

.....

--- Métodos VIDEO ---

obtener_udp_video_adress()

obtener_video_capture()

obtener_frame_read()

Ejecuta:

self_background_fame_read = BackgroundFrameRead(obtener_ud_video_adress()).start()

Thread(update_frame).start()

stop_video_capture()

Ejecuta:

setear_stream_off()

--- Métodos Comandos Generales ---

--- Tabulados por cadenas de callbacks solo para comprensión; todos son métodos generales de la clase ---

enviar_comando_con_respuesta()

enviar_comando_de_control()

coenctar() ["command"]

setear_credenciales_wifi(self,ssid,password)

conectar_a_wifi(self,ssid,password)

setear_stream_on()

setear_stream_off()

setear_velocidad(x) cm/s - velocidad general.

habilitar_mpads()

dehsabilitar_mpads()

despegar()

aterrizar()

emergencia()

mover(direction, x) "up" "left", cm

mover_arriba(x)

mover_abajo(x)

mover_izquierda(x)

mover_derecha(x)

mover_adelante(x)

mover_atras(x)

rotar_sentido_horario(x°)

rotar_anti_sentido_horario(x°)

voltear_flip(direction) "l" "b"

voltear_flip_izquierda()

voltear_flip_derecha()

voltear_flip_adelante()

voltear_flip_atras()

enviar_comando_de_lectura()

obtener_velocidad() "speed?"

obtener_bateria() "battery?"

obtener_tiempo_vuelo() "time?"

obtener_altura() "height?"

obtener_temperatura() "temp?"

obtener_attitude() "attitude?"

obtener_barometro() "baro?"

obtener_distancia_tof() "tof?"

obtener_numero_serie() "sn?"

obtener_version_sdk() "sdk?"

setear_datos_sdk()

enviar_comando_sin_respuesta()

volar_hacia_xyz_velocidad(x,y,z,speed) "go"

curvar_xyz_velocidad(x1,y1,z1,x2,y2,z2,velocidad) "curve"

enviar_velocidades_control_remoto() "rc"

--- Finalización ---

end()

Ejecuta entre otros:

```
    aterrizar()
    setear_stream_off()
    background_grame_read.stop()
    cap.release()
```

```
__del__()  
    Ejecuta entre otros:  
    end()
```

TelloAutomatico(Tello)

```
--- Para acceder sin necesitar más interfaces, solo ejecución directa de comandos ---  
__init__(self)  
    super().__init__()  
    conectar() # "command" - Habilita envío de otros comandos.  
    setear_datos_sdk()  
    setear_velocidad(10)  
    setear_stream_off()  
    --- A partir de ahora el usuario puede ejecutar cualquier método en su script ---
```

TelloSinEstados(Tello)

```
--- Tello normal, inhabilita recepción continua de estados ante problemas sdk
```

TelloAutomaticoSinEstados(TelloAutomatico)

```
--- Tello Automatico, inhabilita recepción continua de estados ante problemas sdk
```