```
Help on package tellopy:

NAME
    tellopy - DJI Tello controller

FILE
    /Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/site-
packages/tellopy/__init__.py

DESCRIPTION
    This is a python package which controlls DJI toy drone 'Tello'. The
major portion of the source
    code was ported from the driver of GOBOT project. Please refer their
blog post at
    https://gobot.io/blog/2018/04/20/hello-tello-hacking-drones-with-go

PACKAGE CONTENTS
    _internal (package)
    examples (package)

CLASSES
    __builtin__.object
        tellopy._internal.tello.Tello

    class Tello(__builtin__.object)
      |  Methods defined here:
      |
      |  __init__(self, port=9000)
      |
      |  backward(self, val)
      |      Backward tells the drone to go in reverse.
      |      Pass in an int from 0-100.
      |
      |  clockwise(self, val)
      |      Clockwise tells the drone to rotate in a
      |      clockwise direction.
      |      Pass in an int from 0-100.
      |
      |  connect(self)
      |      Connect is used to send the initial connection request
      |          to the drone.
      |
      |  counter_clockwise(self, val)
      |      CounterClockwise tells the drone to rotate in a
      |      counter-clockwise direction.
      |       Pass in an int from 0-100.
      |
      |  down(self, val)
      |      Down tells the drone to descend. Pass in an int from 0-100.
      |
      |  flip_back(self)
      |      flip_back tells the drone to perform a backwards flip
      |
      |  flip_backleft(self)
```

```
|      flip_backleft tells the drone to perform a
|          backwards left flip
|
|  flip_backright(self)
|
|     right flip
|
|  flip_forward(self)
|      flip_forward tells the drone to perform a forwards flip
|
|  flip_forwardleft(self)
|      flip_forwardleft tells the drone to perform a forwards
|.     left flip
|
|  flip_forwardright(self)
|      flip_forwardright tells the drone to perform a
|      forwards right flip
|
|  flip_left(self)
|      flip_left tells the drone to perform a left flip
|
|  flip_right(self)
|      flip_right tells the drone to perform a right flip
|
|  forward(self, val)
|      Forward tells the drone to go forward.
|          Pass in an int from 0-100.
|
|  get_video_stream(self)
|      Get_video_stream is used to prepare buffer object
|     which receive video data from the drone.
|
|  land(self)
|      Land tells the drone to come in for landing.
|
|  left(self, val)
|      Left tells the drone to go left. Pass in an int from 0-100.
|
|  palm_land(self)
|      Tells the drone to wait for a hand underneath it
|       and then land.
|
|  quit(self)
|      Quit stops the internal threads.
|
|  recv_file_data(self, data)
|
|  right(self, val)
|      Right tells the drone to go right. Pass in an int from 0-100.
|
|  send_packet(self, pkt)
|      Send_packet is used to send a command packet to the drone.
|
|  send_packet_data(self, command, type=104, payload=[])
```

```
|
|  set_exposure(self, level)
|      Set_exposure sets the drone camera exposure level.
|          Valid levels are 0, 1, and 2.
|
|  set_loglevel(self, level)
|      Set_loglevel controls the output messages. Valid levels are
|      LOG_ERROR, LOG_WARN, LOG_INFO, LOG_DEBUG and LOG_ALL.
|
|  set_pitch(self, pitch)
|      Set_pitch controls the forward and backward tilt
|     of the drone.
|      Pass in an int from -1.0 ~ 1.0.
|     (positive value will make the drone move forward)

|  set_roll(self, roll)
|      Set_roll controls the the side to side tilt of the drone.
|      Pass in an int from -1.0 ~ 1.0.
|     (positive value will make the drone move to the right)
|
|  set_throttle(self, throttle)
|      Set_throttle controls the vertical up and down
|     motion of the drone.
|      Pass in an int from -1.0 ~ 1.0. (positive value means upward)
|
|  set_video_encoder_rate(self, rate)
|      Set_video_encoder_rate sets the drone video encoder rate.
|
|  set_video_mode(self, zoom=False)
|       Tell the drone whether to capture 960x720 4:3 video,
|       or 1280x720 16:9 zoomed video.
|       4:3 has a wider field of view (both vertically and
|      horizontally), 16:9 is crisper.
|
|  set_yaw(self, yaw)
|      Set_yaw controls the left and right rotation of the drone.
|      Pass in an int from -1.0 ~ 1.0.
|     (positive value will make the drone turn to the right)
|
|  start_video(self)
|      Start_video tells the drone to send start info (SPS/PPS)
|     for video stream.
|
|  subscribe(self, signal, handler)
|      Subscribe a event such as EVENT_CONNECTED, EVENT_FLIGHT_DATA,
|     EVENT_VIDEO_FRAME and so on.
|
|  take_picture(self)
|
|  takeoff(self)
|      Takeoff tells the drones to liftoff and start flying.
|
|  up(self, val)
|      Up tells the drone to ascend. Pass in an int from 0-100.
```

```
|
|  wait_for_connection(self, timeout=None)
|      Wait_for_connection will block until the connection
|     is established.
|
|  ----------------------------------------------------------------------
|  Data descriptors defined here:
|
|  __dict__
|      dictionary for instance variables (if defined)
|
|  __weakref__
|      list of weak references to the object (if defined)
|
|  ----------------------------------------------------------------------
|  Data and other attributes defined here:
|
|  CONNECTED_EVENT = Event::connected
|
|  EVENT_CONNECTED = Event::connected
|
|  EVENT_DISCONNECTED = Event::disconnected
|
|  EVENT_FILE_RECEIVED = Event::file received
|
|  EVENT_FLIGHT_DATA = Event::fligt_data
|
|  EVENT_LIGHT = Event::light
|
|  EVENT_LOG = Event::log
|
|  EVENT_TIME = Event::time
|
|  EVENT_VIDEO_DATA = Event::video data
|
|  EVENT_VIDEO_FRAME = Event::video frame
|
|  EVENT_WIFI = Event::wifi
|
|  FLIGHT_EVENT = Event::fligt_data
|
|  LIGHT_EVENT = Event::light
|
|  LOG_ALL = 99
|
|  LOG_DEBUG = 3
|
|  LOG_ERROR = 0
|
|  LOG_EVENT = Event::log
|
|  LOG_INFO = 2
|
|  LOG_WARN = 1
```

```
      |
      |  STATE_CONNECTED = State::connected
      |
      |  STATE_CONNECTING = State::connecting
      |
      |  STATE_DISCONNECTED = State::disconnected
      |
      |  STATE_QUIT = State::quit
      |
      |  TIME_EVENT = Event::time
      |
      |  VIDEO_FRAME_EVENT = Event::video frame
      |
      |  WIFI_EVENT = Event::wifi

DATA
    __all__ = ['Tello']
```