3/14/2019

Connecting to two Tellos and issuing commands can be relatively easily implemented. By putting the Tellos into router moder where they connect to a common base station, commands can be issued to several drones using the code snippit below:

```
# Import the necessary modules
import socket
import threading
import time

# IP and port of Tello
tello1_address = ('192.168.0.101', 8889)
tello2_address = ('192.168.0.102', 8889)

# IP and port of local computer
local1_address = ('', 9010)
local2_address = ('', 9011)

# Create a UDP connection that we'll send the command to
sock1 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock2 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind to the local address and port
sock1.bind(local1_address)
sock2.bind(local2_address)

# Send the message to Tello and allow for a delay in seconds
def send(message, delay):
  # Try to send the message otherwise print the exception
  try:
    sock1.sendto(message.encode(), tello1_address)
    sock2.sendto(message.encode(), tello2_address)
    print("Sending message: " + message)
  except Exception as e:
    print("Error sending: " + str(e))
```

When connected to a Tello via its broadcasted WiFi network, video can be pulled using the TelloPy library and the code below:

```
def main():

    drone = tello.Tello('', 8889)
    vplayer = TelloUI(drone,"./img/")

    # start the Tkinter mainloop
    vplayer.root.mainloop()
```

The TelloPy library defines a Tello object as follows:

```
class Tello:
    """Wrapper class to interact with the Tello drone."""

    def __init__(self, local_ip, local_port, imperial=False, command_timeout=.3, te
                 tello_port=8889):
        """
        Binds to the local IP/port and puts the Tello into command mode.

        :param local_ip (str): Local IP address to bind.
        :param local_port (int): Local port to bind.
        :param imperial (bool): If True, speed is MPH and distance is feet.
                                If False, speed is KPH and distance is meters.
        :param command_timeout (int|float): Number of seconds to wait for a respons
        :param tello_ip (str): Tello IP.
        :param tello_port (int): Tello port.
        """

        self.abort_flag = False
        self.decoder = libh264decoder.H264Decoder()
        self.command_timeout = command_timeout
        self.imperial = imperial
        self.response = None
        self.frame = None  # numpy array BGR -- current camera output frame
        self.is_freeze = False  # freeze current camera output
        self.last_frame = None
        self.socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  # socket fo
        self.socket_video = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  # soc
        self.tello_address = (tello_ip, tello_port)
        self.local_video_port = 11111  # port for receiving video stream
        self.last_height = 0
        self.socket.bind((local_ip, local_port))

        # thread for receiving cmd ack
        self.receive_thread = threading.Thread(target=self._receive_thread)
        self.receive_thread.daemon = True

        self.receive_thread.start()

        # to receive video -- send cmd: command, streamon
        self.socket.sendto(b'command', self.tello_address)
        print ('sent: command')
        self.socket.sendto(b'streamon', self.tello_address)
        print ('sent: streamon')

        self.socket_video.bind((local_ip, self.local_video_port))

        # thread for receiving video
        self.receive_video_thread = threading.Thread(target=self._receive_video_thr
        self.receive_video_thread.daemon = True

        self.receive_video_thread.start()
```