

Tello Documentation

Hao Liu

March 2022

1 Introduction

This is a document to introduce and explain some of the built-in function used in this project of Dji Tello Edu Drone. Most of the information are acquired from the Github page of DJITelloPy (<https://github.com/damiafuentes/DJITelloPy>) and Tello user guide (<https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20SDK%202.0%20User%20Guide.pdf>).

2 Python package used

2.1 djitellopy

This is the package provided by Dji. Tello.py inside includes most of the functions we used to control the drone

2.2 cv2

We use cv2 package to modify on the image we obtained from the cameras of the drone and show it on computer.

2.3 pygame

We use pygame to write the function of keyboard reading so that we can send command to the drone with our keyboard.

3 Tello function introduction

3.1 Connection and video setup

1. `connect()` : Send the connection request to the drone
2. `streamon()` : Enable video stream
3. `streamoff()` : Disable video stream.
4. `set_video_direction(int)` : The Tello Edu drone has two cameras. One is in the front and one is at below. This function is used to indicate which camera is now being displayed on the screen. When the input is set 0, the front view is displayed and when the input is set 1, the down view is displayed.

5. `get_frame_read()` : Get the image from the camera.

3.2 Basic control Command

1. `takeoff()`: Automatic takeoff.
2. `land()`: Automatic landing.
3. `get_state_field(str)`: Get the a specific component of the drone's state. Possible entries a relisted as follows:

1. Tello EDU with mission pads enabled only 'mid', 'x', 'y', 'z', 'mpy': (custom format 'x,y,z')
2. Common entries: 'pitch', 'roll', 'yaw', 'vgx', 'vgy', 'vgz', 'templ', 'temph', 'tof', 'h', 'bat', 'time', 'agx', 'agy', 'agz')

All other `get_sth.` functions are written based on this function. The description of each input is shown as Table. 1

4. `send_rc_control(lr,fb,ud,yv)`: Four channels correspond to four different control command. `lr` represents the left_right velocity, `fb` represents the forward_backward velocity, `ud` represents the up_down velocity and `yv` represents the yaw velocity. The range of all inputs is -100 - 100.

Table 1: Input command description

mid	The ID of the Mission Pad detected. If no Mission Pad is detected, a "-1" message will be received instead
x,y,z	The "x,y,z" coordinate detected on the Mission Pad. If there is no Mission Pad, a "0" message will be received
mpry	The pitch roll and yaw angle relative to the detected Mission Pad
pitch,roll,yaw	The degree of the attitude pitch roll and yaw angle
vgx,vgy,vgz	The speed of the "x,y,z" axis
templ,temph	The lowest or highest temperature in degree Celsius
tof	The flight distance in cm
h	The height in cm
time	The amount of time the motor has been used
agx,agy,agz	The acceleration of the "x,y,z" axis

4 Mission Pad related command

1. `enable_mission_pads()`: Turn on the Mission Pad detection function.
2. `disable_mission_pads()`: Turn off the Mission Pad detection function.
3. `set_mission_pad_detection_direction(int)`: Set the detection direction. 0 enables downwards detection only. 1 enables forward detection only. 2 enables both forward and downward detection.
4. `go_xyz_speed_mid(x,y,z,speed,mid)`: Fly to x y z relative to the mission pad with id mid. Speed defines the traveling speed in cm/s. All the inputs are integers and if no Mission Pads are found, an error will be returned.
5. `go_xyz_speed_yaw_mid(x, y, z, speed, yaw, mid1, mid2)`: Fly to x y z relative to mid1. Then fly to 0 0 z over mid2 and rotate to yaw relative to mid2's rotation. Speed defines the traveling speed in cm/s. All the inputs are integers and if no Mission Pads are found, an error will be returned.