

pyhula

A Python pack used by hula.

```
Python version:3.6.7
```

Installing

Input the following code in powershell(cmd.exe) to install pyhula.

```
pip install pyhula

pip install pyhula-1.1.7-cp36-cp36m-win_amd64.whl
```

Checking version

- ◆ Input "pip list" in powershell(cmd.exe) to get pyhula's version
- ◆ Using the following code.

```
import pyhula
ver = pyhula.get_version()
print(ver)
```

Usage

Use the following codes to create a userApi instance. Its interfaces can be used to control fylo plane. Go to doc/html/English/index.html to see the interface specification.

```
import pyhula
api = pyhula.UserApi()
if not api.connect():
    print("connect error")
else:
    print('connection to station by wifi')
```

```
api.single_fly_takeoff() #takeoff
api.single_fly_touchdown() # Landing
```

Interface

Connected drone

```
connect(server_ip)

...

    description:
    connect drone,

    parameter :
    optional:server_ip: If the drone IPv4 address is not specified,
    the return value is automatically obtained:
    True: True  False: False
...

Example code: api.connect('192.168.1.118')
Example code: api.connect()
```

Takeoff

```
single_fly_takeoff(led)
...

    description:
    Control drone takeoff in real time
    parameter :
    led: The default value is 0,
    format:{'r':0,'g':0,'b':0,'mode':1}  r,g,b: color gamut, mode: 1/ light on,
2/light off, 4/RGB cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...

Example code : api.single_fly_takeoff()
                api.single_fly_takeoff({'r':16,'g':15,'b':100,'mode':1})
```

Landing

```

single_fly_touchdown(led)
'''
    description:
    Control drone Landing in real time
    parameter :
    led: The default value is 0,
    format: {'r':0,'g':0,'b':0,'mode':1}  r,g,b: color gamut, mode: 1/ light on,
2/light off, 4/RGB cycle light, 16/ colorful light,32/ blink light,64/ breathing light
    '''
    Example code :api.single_fly_touchdown()
                    api.single_fly_touchdown({'r':16,'g':15,'b':100,'mode':1})

```

Hover

```

single_fly_hover_flight(time,led)
'''
    description:
    hover_flight
    parameter:
    time: Hover time (seconds)

    '''
    Example code : api.single_fly_hover_flight(10)
                    api.single_fly_hover_flight(10,{'r':16,'g':15,'b':100,'mode':1})

```

Fly_forward

```

single_fly_forward(distance,speed,led)
'''
    description:
    Control drone fly_forward in real time
    parameter:
    distance: Flight distance (cm)
    speed: If left blank, it will default to 100 speed (0-100cm) /s
    led: The default value is 0,
    format: {'r':0,'g':0,'b':0,'mode':1}  r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
    '''
    Example code : api.single_fly_forward(100)
                    api.single_fly_forward(100, 100,{'r':16,'g':15,'b':100,'mode':1})

```

Fly_backward

```
single_fly_back(distance,speed,led)
...
description:
Control drone fly_backward in real time
parameter:
distance: Flight distance (cm)
speed: If left blank, it will default to 100 speed (0-100cm) /s
led: The default value is 0,
format:{'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_back(100)
               api.single_fly_back(100, 100,{'r':16,'g':15,'b':100,'mode':1})
```

Fly to left

```
single_fly_left(distance,speed,led)
...
description:
Control drone fly to left in real time
parameter:
distance: Flight distance (cm)
speed: If left blank, it will default to 100 speed (0-100cm) /s
led: The default value is 0,
format:{'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_left(100)
               api.single_fly_left(100, 100,{'r':16,'g':15,'b':100,'mode':1})
```

Fly to right

```
single_fly_right(distance, speed,led)
...
description:
Control drone fly to right in real time
parameter:
distance: Flight distance (cm)
speed: If left blank, it will default to 100 speed (0-100cm) /s
led: The default value is 0,
format: {'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_right(100)
               api.single_fly_right(100, 100,{'r':16,'g':15,'b':100,'mode':1})
```

Fly_up

```
single_fly_up(distance, speed,led)
...
description:
Control drone fly_up in real time
parameter:
height:Flight height (cm)
speed: If left blank, it will default to 100 speed (0-100cm) /s
led: The default value is 0,
format: {'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_up(100)
               api.single_fly_up(100, 100,{'r':16,'g':15,'b':100,'mode':1})
```

Fly_down

```

single_fly_down(distance,speed,led)
...
description:
    Control drone fly_down in real time
parameter:
    height: Flight height (cm)
    speed: If left blank, it will default to 100 speed (0-100cm) /s
    led: The default value is 0,
    format:{'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
    cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_down(100)
                api.single_fly_down(100, 100,{'r':16,'g':15,'b':100,'mode':1})

```

Turn left

```

single_fly_turnleft(angle,led)
...
description:
    Control drone turn left in real time
parameter:
    angle: Rotation Angle (degree)
    led: The default value is 0,
    format:{'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
    cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_turnleft(90)
                api.single_fly_turnleft(90,{'r':16,'g':15,'b':100,'mode':1})

```

Turn right

```

single_fly_turnright(angle,led)
...
description:
    Control drone turn right in real time
parameter:
    angle: Rotation Angle (degree)
    led: The default value is 0,
    format:{'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
    cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_turnright(90)
                api.single_fly_turnright(90,{'r':16,'g':15,'b':100,'mode':1})

```

Bounce

```
single_fly_bounce(frequency, height, led)
...
description:
Control drone  bounce  in real time
parameter:
frequency: Bounce times
height: Bounce distance (cm)
led: The default value is 0,
format: {'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code: api.single_fly_bounce(3, 50)
              api.single_fly_bounce(3, 50, {'r':16, 'g':15, 'b':100, 'mode':1})
```

Straight_flight

```
single_fly_straight_flight( x, y, z, speed,led)
...
description:
straight_flight to co-ordinate (x,y,z)
parameter:
x: co-ordinate x (cm)  y: co-ordinate y (cm)  z: co-ordinate z (cm)
speed: If left blank, it will default to 100 speed (0-100cm) /s
led: The default value is 0,
format: {'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code: api.single_fly_straight_flight(100, 100, 100)
              api.single_fly_straight_flight(100, 100, 100, 50,{'r':16,'g':15,'b':100,'mode':1})
```

Flight around

```
single_fly_radius_around(radius,led)
...
description:
flight around A point ahead of the plane

parameter:
radius: radius ( cm,  positive: anticlockwise ,  negative:clockwise )
led: The default value is 0,
format: {'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
...
Example code : api.single_fly_radius_around(100)
               api.single_fly_radius_around(100,{'r':16,'g':15,'b':100,'mode':1})
```


Autogyration

```
single_fly_autogyration360(num,led)
'''
description:
Clockwise, counterclockwise rotation a certain number of turns
parameter:
num:( positive: anticlockwise , negative:clockwise)
led: The default value is 0,
format:{'r':0,'g':0,'b':0,'mode':1}   r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
'''
```

Example code : `api.single_fly_autogyration360(2)`
`api.single_fly_autogyration360(2,{'r':16,'g':15,'b':100,'mode':1})`

Somersault

```
single_fly_somersault(direction)
'''
description:
The drone forward, backward, left or right somersault
parameter:
DIRECTION_FORWARD=0, /* forward. | */
DIRECTION_BACK=1, /* back. | */
DIRECTION_LEFT=2, /* left. | */
DIRECTION_RIGHT=3, /* right. | */
led: The default value is 0,
format:{'r':0,'g':0,'b':0,'mode':1}   r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
cycle light, 16/ colorful light,32/ blink light,64/ breathing light
'''
```

Example code : `api.single_fly_somersault(0)`
`api.single_fly_somersault(0,{'r':16,'g':15,'b':100,'mode':1})`

Curvilinear Flight

```
single_fly_curvilinearFlight(x,y,z,speed,led)
...
description:
curvilinear Flight to (x,y,z)
parameter:
x: x co-ordinate (cm) (Body left and right, right is positive)
y: y co-ordinate (cm) (Front and back of the body, front is positive)
z: z co-ordinate (cm) (Body up and down, up is positive)
direction: True: anticlockwise
           False: clockwise
           Default :True
speed: If left blank, it will default to 100 speed (0-100cm) /s
led: The default value is 0,
format:{'r':0,'g':0,'b':0,'mode':1}    r,g,b: color gamut, mode: 1/ light on, 2/light off, 4/RGB
... cycle light, 16/ colorful light,32/ blink light,64/ breathing light

Example code: api.single_fly_curvilinearFlight(100, 100, 0 , True, 50)
api.single_fly_curvilinearFlight(100, 100, 0, False, 50,
{'r':16,'g':15,'b':100,'mode':1})
```

Turn on obstacle avoidance

```
single_fly_barrier_aircraft(mode)
...
    description:
    Turn on obstacle avoidance
    parameter:
    mode: True: Turn on   False: Turn off
...
Example code: api.single_fly_barrier_aircraft(True)
```

Line_walking

```
single_fly_Line_walking(fun_id, dist, way_color)
...
    description:
    line walking
    parameter:
    fun_id    = 0 //0: Forward patrol, Ignore a junction
    dist      // distance,   cm
    way_color // Line color gamut
    , 0- blackest 255- White
    return:
    return result = 1; // The result of the instruction execution: 0- failed, 1- succeed ,
    2- junction
...
Example code: api.single_fly_Line_walking(0, 100, 0)
```

Identification label

```
single_fly_AiIdentifies(mode)
...
description:
identification label
parameter:
mode:
0-9: Identify numeric labels 0-9,
10: Identify left arrow,
11: Identify right arrow,
12: Identify the up arrow,
13: Identify down arrow,
20: end task,
65-90: capital letterA-Z;
The recognition process continues after the recognition is triggered 300ms, If the
identification is successful, it ends immediately

return:
x; The x-coordinate of the label card relative to the drone
y; The y-coordinate of the label card relative to the drone
z; The z-coordinate of the label card relative to the drone
angle; The Angle between the label card and the drone
result; //False: failed, True: succeed
...
Example code: api.single_fly_AiIdentifies(1)
```

Optical flow align QR code

```
single_fly_Optical_flow_alignment(qr_id, qr_size, angle = 0)
...
description:
Optical flow align QR code
parameter:
qr_id; To QR code id[0-9],
qr_size: The physical size of the QR code, range [6, 30], default value 20,
unit :cm
angle: Align the QR code and rotate it by n degrees
return:
result; // False: failed, True: succeed
...
Example code: api.single_fly_Optical_flow_alignment(1, 20, 0)
```

Optical flow recognition QR code

```
single_fly_Optical_flow_recognition(qr_id, qr_size)
...
    description:
    Optical flow recognition QR code
    parameter:
    qr_id; To QR code id[0-9],
    qr_size: The physical size of the QR code, range [6, 30], default value 20,
    unit :cm
    return:
    {
    result; // False: failed, True: succeed
    x;// Distance between the drone and the QR code
    y;// Distance between the drone and the QR code
    z;// Distance between the drone and the QR code
    yaw;// Angle between the drone and the QR code
    qr_id;// Identified the QR code id
    }
    ...
Example code: api.single_fly_recognition_Qrcode(0, 1)
```

Front camera Align QR code

```
single_fly_Proactive_alignment(qr_id)
...
    description:
    Front camera align QR code
    parameter:
    qr_id; To QR code id[0-9],
    return:
    result; // False: failed, True: succeed
    ...
Example code: api.single_fly_Proactive_alignment(1)
```

Front camera recognition QR code

```

single_fly_Anticipatory_recognition(qr_id)
...
    description:
    Front camera recognition QR code
    parameter:
    qr_id; To QR code id[0-9],
    return:
    {
        result; // False: failed, True: succeed
        x; // Distance between the drone and the QR code
        y; // Distance between the drone and the QR code
        z; // Distance between the drone and the QR code
        yaw; // Angle between the drone and the QR code
        qr_id; // Identified the QR code id
    }
...
Example code: api.single_fly_Anticipatory_recognition(1)

```

Track_Qrcode

```

single_fly_track_Qrcode(qr_id, time)
...
    description:
    Track QR code [0-9] [time] seconds
    parameter:
    qr_id: QR code id
    time: Tracking time
    return:
    result: 0: failed, 1: succeed
...
Example code: api.single_fly_track_Qrcode(1, 10)

```

Color recognition, get the color of the current video stream frame

```
single_fly_getColor()
'''
    description:
    Color recognition, get the color of the current video stream frame parameter:
    Mode:1star, Run a frame

    return:
    r,g,b: colour gamut
    state:0 failed 1succeed

'''
Example code: ret = api.single_fly_getColor()#return: r,g,b: colour gamut state:0 failed 1
succeed
```

Set light color and mode, Does not block the main thread

```
single_fly_lamplight(r, g, b, time, mode)
'''
    description:
    Set light color and mode
    parameter:
    r,g,b: colour gamut
    time: duration /s
    mode: 1/ light on, 2/light off, 4/RGB cycle light,
          16/ colorful light,32/ blink light,64/ breathing light

'''
Example code: api.single_fly_lamplight(255, 0, 0, 1, 1)# Set light color and mode
```

lasing

```
plane_fly_generating(type, data ,reserve)
...
    description:
    lasing
    parameter:
    type = 0;    // laser: 0- single shot,1- keep shoting, 2- Turn on the laser receiver,
                  3- Turn off the laser receiver 4- keep firing 5- Turn off laser
    data = 10; // frequency, times/s, range 1-14 reserve = 100 // Laser gap, Data Range 1-
    255
...
Example code: api.plane_fly_generating(0, 10, 100)# single shot
              api.plane_fly_generating(2, 10, 100) #Turn on the laser receiver
```


Laser receiver been hit

```

    plane_fly_laser_receiving()
    ...
    description:
    Laser receiver been hit
    return:
    True: Received laser
    False: No laser was received
    ...
    Example code: api.plane_fly_laser_receiving()

```

Positioning QR code switch

```

Plane_cmd_switch_QR(type)
...
    description:
    Positioning QR code switch
    parameter:
    type:0-turn on    1- turn off
    ...
    Example code: api.Plane_cmd_switch_QR(0)

```

Shot

```

Plane_fly_take_photo()
...
    description:
    Before taking pictures, the video stream must be turned on
    ...
    Example code: api.Plane_fly_take_photo()#shot

```

Recording

```

Plane_cmd_switch_video(type)
...
    description:
    star recording
    parameter:
    type:// recording, 0-star, 1-end

```

```
...
```

Example code: `api.Plane_cmd_switch_video(0)# star recording`

Video stream turn on

```
Plane_cmd_swith_rtp(type)
...
```

description:
video stream turn on
parameter:
type:0- turn on, 1- turn off

```
...
```

Example code: `api.Plane_cmd_swith_rtp(0)# video stream turn on`

Open the video stream window

```
single_fly_flip_rtp()
...
```

description:
Open the video stream window (the video stream must be turned on)
parameter:

```
...
```

Example code: `api.single_fly_flip_rtp()#Open the video stream window`

Set the front camera tilt Angle

```
Plane_cmd_camera_angle(type, data)
...
```

description:
Set the front camera tilt Angle
parameter:
type = 0; // direction: 0-up,1-down(absolutely), 2 and 3 algorithm control,
4- calibration, 5- scratch block up, 6- scratch block down (relatively)
data = 30; // Angle range: 0~90

```
...
```

Example code: `api.Plane_cmd_camera_angle(0, 30)# Set the front camera tilt Angle`

Unlock the Low speed propeller

```
plane_fly_arm()
...
```

```
description:
Unlock drone motor propeller
parameter:
```

```
...
```

```
Example code: api.plane_fly_arm()#Low speed propeller
```

Lock the Low speed propeller

```
plane_fly_arm()
```

```
...
```

```
description:
turn off drone motor
parameter:
```

```
...
```

```
Example code: api.plane_fly_disarm()#Lock the Low speed propeller
```

Get obstacle avoidance direction information

```
Plane_getBarrier()
```

```
...
```

```
description:
Get obstacle avoidance direction information
parameter:
return: Dictionary Obstacle status in each direction
, True: There is an obstacle, False: There is No obstacle
{
'forward': True
'back': True,
'left': True,
'right': True,
}
```

```
...
```

```
Example code: ret = api.Plane_getBarrier()#Get obstacle avoidance direction information
```

Get drone battery percentage

```
get_battery()
...
    description:
    Get drone battery percentage
    returned value:
    Integer: battery percentage
...
Example code: ret = api.get_battery()#Get drone battery percentage
```

Get drone coordinates (x,y,z)

```
get_coordinate()
...
    description:
    Get drone coordinates [x, y, z]
    parameter:
    returned value:
    [x, y, z]
...
Example code: ret = api.get_coordinate()#Get drone coordinates [x, y, z]
```

Get drone Angle

```
get_yaw()
...
    description:
    Get drone Angle (degree)
    returned value:
    Integer :[yaw Angle, pitch Angle, roll Angle]
...
Example code: ret = api.get_yaw()
```

Get drone speed (X axis speed,Y axis speed,Z axis speed)

```

get_plane_speed()
...

description:
Get drone speed (X axis speed,Y axis speed,Z axis speed)
returned value:
Integer:[X,Y,Z]
...

Example code: ret = api.get_plane_speed()

```

Get the drone Tof height value

```

get_plane_distance()
...

description:
Get the drone Tof height value
returned value:
Integer: drone Tof height value
...

Example code: ret = api.get_plane_distance()

```

Get drone ID

```

get_plane_id()
...

description:
Get drone ID
返回值:
Integer: drone ID
...

Example code: ret = api.get_plane_id()

```

External electromagnet

```

Plane_cmd_electromagnet(type)
...

description:
External electromagnet
parameter:
type:2- Electromagnet attracts 3- Electromagnet pops out
...

Example code: ret = api.Plane_cmd_electromagnet(2)

```

External cramp

```
Plane_cmd_clamp(type,angle)
...
    description:
    External cramp
    parameter:
    type: 0: Clamp disable,1: Clamp enable,2: Clamp Angle,3: Electromagnet pops out,4:
    Electromagnet attracts
...
```

Example code:

```
api.Plane_cmd_clamp(0)## Clamp disable
api.Plane_cmd_clamp(1)## Clamp enable
api.Plane_cmd_clamp(2,30)## First, call the clip to enable, then the clip Angle to open at 30 degrees
api.Plane_cmd_clamp(3)## Electromagnet pops out
api.Plane_cmd_clamp(4)## Electromagnet attracts
```