# 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 verison

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](javascript:;):

connect drone,

parameter :

[optional](javascript:;):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](javascript:;):

Control drone takeoff  in [real](javascript:;) [time](javascript:;)

parameter :

led: The default value is 0，

format:{'r':0,'g':0,'b':0,'mode':1} r,g,b: color gamut, mode: 1/ [light](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) light,64/ breathing light

'''

Example code : api.single\_fly\_takeoff() api.single\_fly\_takeoff({'r':16,'g':15,'b':100,'mode':1})

[Landing](javascript:;)

single\_fly\_touchdown(led) '''

[description](javascript:;):

Control drone [Landing](javascript:;) in [real](javascript:;) [time](javascript:;)

parameter :

led: The default value is 0，

format:{'r':0,'g':0,'b':0,'mode':1} r,g,b: color gamut, mode: 1/ [light](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;):

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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;): [anticlockwise](javascript:;) ，[negative](javascript:;):[clockwise](javascript:;) )

led: The default value is 0，

format:{'r':0,'g':0,'b':0,'mode':1} r,g,b: color gamut, mode: 1/ [light](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;): [anticlockwise](javascript:;) ，[negative](javascript:;):[clockwise](javascript:;))

led: The default value is 0，

format:{'r':0,'g':0,'b':0,'mode':1} r,g,b: color gamut, mode: 1/ [light](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;)

False： clockwise

[Default](javascript:;) :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](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，16/ colorful light,32/ [blink](javascript:;) 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](javascript:;)， cm

way\_color // Line color gamut

，0- [blackest](javascript:;) 255- White

return:

return result = 1; // The result of the instruction execution：0- failed，1- [succeed](javascript:;) ,

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](javascript:;) [task](javascript:;)，

65-90：[capital](javascript:;) [letter](javascript:;)A-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](javascript:;)

'''

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](javascript:;)

'''

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](javascript:;)

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](javascript:;)

'''

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](javascript:;)

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](javascript:;)

'''

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](javascript:;) [gamut](javascript:;)

state:0 failed 1[succeed](javascript:;)

'''

Example code: ret = api.single\_fly\_getColor()#return：r,g,b: [colour](javascript:;) [gamut](javascript:;) state:0 failed 1 [succeed](javascript:;)

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](javascript:;) [gamut](javascript:;)

time: [duration](javascript:;) /s

mode: 1/ [light](javascript:;) on，2/[light](javascript:;) off，4/RGB cycle light，

16/ colorful light,32/ [blink](javascript:;) light,64/ breathing light

'''

Example code: api.single\_fly\_lamplight(255, 0, 0, 1, 1)# Set light color and mode

[lasing](javascript:;)

plane\_fly\_generating(type, data ,reserve) '''

description:

[lasing](javascript:;)

parameter:

type = 0; // [laser](javascript:;): 0- [single](javascript:;) [shot](javascript:;),1- keep [shot](javascript:;)ing，2- Turn on the laser receiver，

3- Turn off the laser receiver 4- keep firing 5- Turn off [laser](javascript:;)

data = 10; // frequency，times/s，[range](javascript:;) 1-14 reserve = 100 // Laser gap, Data Range 1-255

'''

Example code: api.plane\_fly\_generating(0, 10, 100)# [single](javascript:;) [shot](javascript:;) 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](javascript:;)), 2 and 3 algorithm control，

4- [calibration](javascript:;)，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() '''

plane\_fly\_disarm() '''

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](javascript:;) [value](javascript:;):

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](javascript:;) [value](javascript:;):

[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](javascript:;) [value](javascript:;):

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](javascript:;) [value](javascript:;):

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](javascript:;) [value](javascript:;):

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