



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





Enabling the  
AstroPi mission

# Sense Hat

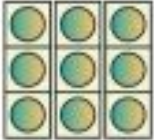

## Python 3 Cheatsheet






To add **Sense HAT** functionality to your python programs add the following lines to import the library for the Sense HAT library:

```
from sense_hat import SenseHat
sense = SenseHat()
```

From that point forwards you can use any of the set of functions from the Sense HAT Library.

 <p><b>LED Matrix</b></p>	<code>sense.set_pixel(0, 0, 255, 0, 0)</code>	Sets the top left LED to the colour red.
	<code>sense.show_letter("J", 0, 0, 255)</code>	Displays the letter "J" on the screen in blue.
	<code>sense.show_message("msg", text_colour=[0, 255, 0])</code>	Displays the message "msg" on the matrix in green.
	<code>sense.load_image("creeper.png", redraw=True)</code>	Load an 8x8 image "creeper.png" image and display it.
	<code>sense.clear()</code>	Clears the LED and switches them all off.
	<code>sense.set_rotation(r=0)</code>	Sets the rotation of the LED matrix.
	<code>sense.set_pixels(pixelList)</code>	Uses pixelList to draw a picture, each item is an [R,G,B] list
 <p><b>Movement</b></p>	<code>yaw,pitch,roll = sense.get_orientation().values()</code>	Gets the orientation data and stores their values as <b>yaw, pitch, roll</b>
	<code>m_x, m_y, m_z = sense.get_compass_raw().values()</code>	Gets the compass data and stores as <b>m_x, m_y, m_z</b>
	<code>x, y, z = sense.get_accelerometer_raw().values()</code>	Gets the accelerometer data and stores as <b>x, y, z</b>
	<code>g_x,g_y,g_z = sense.get_gyroscope_raw().values()</code>	Gets the orientation data and stores as <b>g_x, g_y, g_z</b>

 Temperature  Humidity	<code>t = sense.get_temperature_from_humidity()</code>	Uses the humidity sensor to get temperature and stores it as <b>t</b> .
	<code>t = sense.get_temperature_from_pressure()</code>	Uses the pressure sensor to get temperature and stores it as <b>t</b> .
	<code>h = sense.get_humidity()</code>	Measures the humidity and stores it as <b>h</b> .
	<code>p = sense.get_pressure()</code>	Measures the pressure and stores it as <b>p</b> .

 Joystick	<p>There are a number of ways to capture the input from the joystick. You could use either the <b>pygame</b> or <b>curses</b> library. However for this example we're going to use the evdev system, which you'll need to install using "sudo pip3 install evdev"</p>	
	<pre> from evdev import InputDevice, ecodes, list_devices from select import select  devices = [InputDevice(fn) for fn in list_devices()] for dev in devices:     if dev.name == "Raspberry Pi Sense HAT Joystick":         js = dev  while True:     r, w, x = select([dev.fd], [], [], 0.01)     for fd in r:         for event in dev.read():             if event.type == ecodes.EV_KEY:# and event.value == 1:                 if event.code == ecodes.KEY_UP:                     print("up")                 elif event.code == ecodes.KEY_LEFT:                     print("left")                 elif event.code == ecodes.KEY_RIGHT:                     print("right")                 elif event.code == ecodes.KEY_DOWN:                     print("down")             else:                 print("enter") </pre>	<p>The code on the left looks through the available input devices and finds the Sense-HAT joystick.</p> <p>It then continually check the joystick device and creates a list of events call <b>r</b>.</p> <p>For each event in the list it checks whether it was a keyboard style event.</p> <p>It then compares the the key code to the values for up, down, left and right and pirnts a corresponding message</p>

## Scrolling Message

```
from sense_hat import SenseHat

sense = SenseHat()

while True:
    sense.show_message("Spaaaaaaace!!!", scroll_speed=0.05, text_colour=[255,255,0], back_colour=[0,0,255])
```

## Environmental Sensing

```
from sense_hat import SenseHat

sense = SenseHat()

while True:
    t = sense.get_temperature()
    p = sense.get_pressure()
    h = sense.get_humidity()

    t = round(t,1)
    p = round(p,1)
    h = round(h,1)

    msg = "Temp = %s, Pressure=%s,
Humidity=%s" % (t,p,h)

sense.show_message(msg,scroll_speed=0.05)
```

## Rotating letter "J"

```
from sense_hat import SenseHat
import time

sense = SenseHat()

sense.show_letter("J")

while True:
    x, y, z = sense.get_accelerometer_raw().values()

    x = round(x, 0)
    y = round(y, 0)

    if x == -1:
        sense.set_rotation(180)
    elif y == -1:
        sense.set_rotation(90)
    elif y == 1:
        sense.set_rotation(270)
    else:
        sense.set_rotation(0)

    time.sleep(0.1)
```

## Reaction Game

```
from sense_hat import SenseHat
import time
import random

sense = SenseHat()

# set up the colours (white, green, red, empty)
w = [150,150,150]
g = [0,255,0]
r = [255,0,0]
e = [0,0,0]

# create three different coloured arrows
arrow = [e,e,e,w,w,e,e,e,
         e,e,w,w,w,w,e,e,
         e,w,e,w,w,e,w,e,
         w,e,e,w,w,e,e,w,
         e,e,e,w,w,e,e,e,
         e,e,e,w,w,e,e,e,
         e,e,e,w,w,e,e,e,
         e,e,e,w,w,e,e,e]

arrow_red = [e,e,e,r,r,e,e,e,
            e,e,r,r,r,r,e,e,
            e,r,e,r,r,e,r,e,
            r,e,e,r,r,e,e,r,
            e,e,e,r,r,e,e,e,
            e,e,e,r,r,e,e,e,
            e,e,e,r,r,e,e,e,
            e,e,e,r,r,e,e,e]

arrow_green = [e,e,e,g,g,e,e,e,
              e,e,g,g,g,g,e,e,
              e,g,e,g,g,e,g,e,
              g,e,e,g,g,e,e,g,
              e,e,e,g,g,e,e,e,
              e,e,e,g,g,e,e,e,
              e,e,e,g,g,e,e,e,
              e,e,e,g,g,e,e,e]

pause = 3
score = 0
angle = 0
play = True

sense.show_message("Keep the arrow pointing up", text_colour=[100,100,100])

while play == True:
    last_angle = angle
    while angle == last_angle:
        angle = random.choice([0,90,180,270])
        sense.set_rotation(angle)
        sense.set_pixels(arrow)
        time.sleep(pause)

    x, y, z = sense.get_accelerometer_raw().values()
    x = round(x, 0)
    y = round(y, 0)

    if x == -1 and angle == 180:
        sense.set_pixels(arrow_green)
        score = score + 1
    elif x == 1 and angle == 0:
        sense.set_pixels(arrow_green)
        score = score + 1
    elif y == -1 and angle == 90:
        sense.set_pixels(arrow_green)
        score = score + 1
    elif y == 1 and angle == 270:
        sense.set_pixels(arrow_green)
        score = score + 1
    else:
        sense.set_pixels(arrow_red)
        play = False

    pause = pause * 0.95
    time.sleep(0.5)

msg = "Your score was %s" % (score)
sense.show_message(msg, scroll_speed=0.05, text_colour=[100,100,100])
```

