Projects

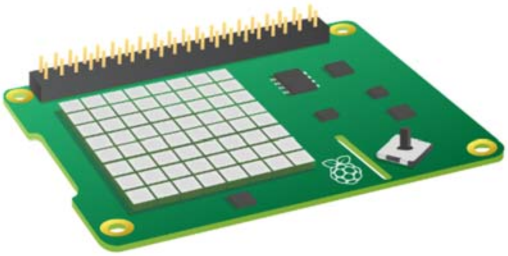
# Getting started with the Sense HAT

Explore the Sense HAT sensor board and its Python library

## Step 1 Introduction

### What you will make

The Sense HAT, which is a fundamental part of the **Astro Pi** (https://astro-pi.org/) mission, allows your Raspberry Pi to sense the world around it.



In this project, you will learn how to control the Sense HAT's LED matrix and collect sensor data, and you will combine these ideas in a number of small projects.

### What you will learn

By following this resource with your Raspberry Pi and Sense HAT you will learn how to:

- Communicate with the Sense HAT using Python
- Access the outputs of the Sense HAT
- Program the inputs of the Sense HAT
- Use the Sense HAT library to display messages and images

- Use variables to store sensor data
- Use loops to repeat behaviours

This resource covers elements from the following strands of the **Raspberry Pi Digital Making Curriculum** (https://www.raspberrypi.org/curriculum/):

- **Use basic programming constructs to create simple programs** (https://www.raspberrypi.org/curriculum/programming/creator)
- **Process input data to monitor or react to the environment** (https://www.raspberrypi.org/curriculum/physical-computing/developer)

## Step 2 What you will need

### Hardware

- Raspberry Pi
- Sense HAT

### Software

You will need the **latest version of Raspbian** (https://www.raspberrypi.org/downloads/) which already includes the following software packages:

- Python 3
- Sense HAT for Python 3

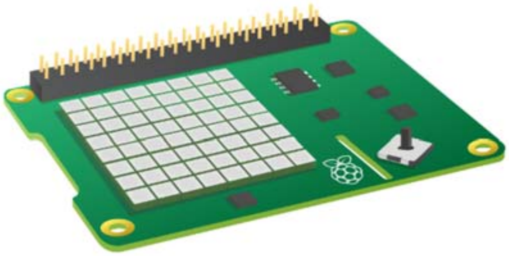
If for any reason you need to install a package manually, follow these instructions:

Type this command into the terminal to install the Sense HAT package:

```
sudo apt-get install sense-hat
```

## Step 3 What is a Sense HAT?

The Sense HAT is an add-on board for the Raspberry Pi, made especially for the **Astro Pi** (http://astro-pi.org/) competition. The board allows you to make measurements of temperature, humidity, pressure, and orientation, and to output information using its built-in LED matrix.



If you don't have access to a real Sense HAT, you can use an emulator.

## Step 4 Displaying text

- Display the text “Astro Pi is awesome” on your Sense HAT’s LED display.

We can change how the message is displayed by adding some extra **parameters** to the `show_message` command.

`scroll_speed`: affects how quickly the text moves across the screen. The default value is `0.1`. The bigger the number, the lower the speed.

`text_colour`: alters the colour of the text and is defined via three values to specify red, green, and blue. These are also called RGB values.

Check out the sections below to learn more about RGB values.

`back_colour`: alters the colour of the background and works in the same way as `text_colour`.

- Add a line of code before your message to define a variable called `blue` with the value `(0, 0, 255)`.

- Add another line of code to define a variable called `yellow` with the value `(255, 255, 0)`.

- Add parameters to the `show_message` command to display the text in yellow with a blue background.

- Add another parameter called `scroll_speed` to the `show_message` command and set the speed equal to `0.05` to speed up how quickly your message scrolls.

- Put your scrolling message in a while loop to make it repeat.

## Step 5 Displaying a single character

- Display the letter “A” on your Sense HAT’s LED display.

We can change how the letter is displayed by using two of the same parameters we used for the `show_message` command: `text_colour` and `back_colour`. Letters do not scroll, so there is no `scroll_speed` parameter.

- Display the letter “J” in red on a white background.

- Use the `sleep` function to display the letters of your name one at a time, each in a different colour, with a one-second pause between each.

- Randomly generate a colour by using `randint` to choose a number between `0` and `255` for each of the three RGB values that make up a colour.

- Use `sense.clear()` at the end of your code to clear the LED matrix.

## Step 6 Displaying images

You can fill the whole LED matrix with a single colour by using the `clear` method with the colour you've picked.

### Setting single pixels

The LED matrix can display more than just text! We can control each LED individually to create an image.

- Light up the pixels (LEDs) in the four corners of the matrix in a colour of your choice.

Can you guess what this code creates?

- Change the code to make a different pixel picture.

### Setting multiple pixels

Setting pixels individually can work brilliantly, but it gets rather complex when you want to set multiple pixels. To change all the pixels in one go with the `set_pixels` command.

- Use the `set_pixels` method to display an image on the LED matrix.

## Step 7 Setting orientation

So far, all our text and images have appeared the same way up, with the HDMI port at the bottom. However, this may not always be the true orientation of the Sense HAT (especially in outer space), so sometimes you might want to change the orientation of the LED matrix.

- Change the orientation of the pixel picture from the previous step by adding code to rotate the display below the code you wrote to connect to the Sense HAT:

You can create a simple animation by flipping an image repeatedly:

## Step 8 Sensing the environment

The Sense HAT has a set of environmental sensors for detecting the surrounding conditions; it can measure pressure, temperature, and humidity.

- Create a scrolling text display which keeps people informed about the current pressure, temperature, and humidity readings. You can use the scrolling text display code you wrote in the 'Displaying text' step to help you.

According to **online documentation** (http://wsn.spaceflight.esa.int/docs/Factsheets/30%20ECLSS%20LR.pdf) , the International Space Station maintains these conditions at the following levels:

Temperature: 18.3-26.7 Celsius

Pressure: 979-1027 millibars

Humidity: around 60%

- Define variables for the colours green (0, 255, 0) and red (255, 0, 0).
- Use an if statement in your code to check whether the temperature is between 18.3 and 26.7 degrees Celsius.
- If the temperature is within this normal range, display the scrolling message with a green background. If not, display a red background.
- Add more if statements to test for normal pressure and humidity conditions as well.

## Step 9 Detecting movement

The Sense HAT has an IMU (**I**nertial **M**easurement **U**nit) chip which includes a set of sensors that detect movement:

- A gyroscope (for detecting which way up the board is)
- An accelerometer (for detecting movement)
- A magnetometer (for detecting magnetic fields)

- Write a program to detect the current pitch, roll, and yaw. Run the program and move the Sense HAT around. Watch how the values change as the Sense HAT moves.

**Note:** When using the movement sensors, it is important to take frequent readings. If you take readings too slowly, for example by putting `time.sleep(0.5)` in your loop, you will see strange results. This is because the code needs lots of measurements in order to successfully combine the data coming from the gyroscope, accelerometer, and magnetometer.

### Which way up?

The `sense.get_accelerometer_raw()` method tells you the amount of G-force acting on each axis (x, y, z). If any axis has ±1G, then you know that axis is pointing downwards.

In this example, the amount of gravitational acceleration for each axis is extracted and is then rounded to the nearest whole number:

- Rotate the Sense HAT. You should see the values for `x` and `y` change between `-1` and `1`. If you place the Pi flat or turn it upside down, the value for the `z` axis will be `1` and then `-1`.

Use this information to set the orientation of the LED matrix.

- Starting with the code above, add some code before the while loop to display the letter "J" on the LED matrix. Use the `show_letter` method you already learned about.

- After the code which displays the G-force values for the x, y and z axes , add an if statement to check which way up the Sense HAT is pointing. Update the orientation of the display using the `set_rotation` method you learned about earlier. Here is some pseudo-code to get you started:

```
If the x axis has -1 G, rotate 180 degrees
Else if the y axis has 1 G, rotate 90 degrees
Else if the y axis has -1 G, rotate 270 degrees
Else rotate 0 degrees
```

### Shake the board

If the board is only rotated, it will only ever experience 1 G of acceleration in any direction; if we were to shake it, the sensor would experience more than 1 G. We could then detect that rapid motion and respond.

For this program we will introduce the `abs()` function, which is not specific to the Sense HAT library but instead is part of standard Python. `abs()` gives us the absolute figure of a value and ignores whether the actual value is positive or negative — for example, `abs(1)` and `abs(-1)` both return `1`. This function is helpful because we don't care in which direction the sensor is being shaken, just that it is shaken.

```
from sense_hat import SenseHat

sense = SenseHat()

red = (255, 0, 0)

while True:
    acceleration = sense.get_accelerometer_raw()
    x = acceleration['x']
    y = acceleration['y']
    z = acceleration['z']

    x = abs(x)
    y = abs(y)
    z = abs(z)

    if x > 1 or y > 1 or z > 1:
        sense.show_letter("!", red)
    else:
        sense.clear()
```

This is a little tricky to emulate, so you should try this one using a real Sense HAT if you can. If you find the program is too sensitive (that is, it thinks the Sense HAT is constantly being shaken), try changing the value `1` to a larger value to raise the threshold of what is defined as a “shake”.

## Step 10 Using the joystick

You can detect when the Sense HAT's joystick is pressed, held, and released in five different directions: up, down, left, right, and middle.

- Depending on which way the joystick was pressed, display one of the letters U, D, L, R or M on the LED matrix.

You can also call a function whenever the Sense HAT's joystick is moved in a particular direction.

- Create functions to fill the LED matrix with four different colours. Add triggers to call one function for each possible direction in which the joystick can be pressed.

## Step 11 Challenge: putting it all together

Now that you've explored most of the features of the Sense HAT, you can combine them to create a project. Here's an example reaction game, which could be used by the astronauts to test their reflexes:

Rotate the board to make the arrow point up. If you match it in time, the arrow turns green and your score increases; if not, your arrow turns red and the game ends. The game keeps showing arrows in new orientations until you lose, and each turn gets faster.

This idea combines:

- Showing messages and images on the LED matrix
- Setting and detecting the orientation
- Use of variables, randomisation, iteration, and selection

As this is more complicated than previous programs in this resource, it's worth planning out the steps involved in pseudo-code:

```
Import the required libraries (sense_hat, time, random)
Create a sensehat object

Define variables for the colours needed (white, green, red, blank)
Create three different arrows (white, green, red)
Set a variable pause to 3 (the initial time between turns)
Set variables score and angle to 0
Set a variable called play to True (this will be used to stop the game later)

while play == True
Choose a new random angle
Display the white arrow
Sleep for current pause length
If orientation matches the arrow…
Add a point and turn the arrow green
Otherwise set play to False and display the red arrow

Shorten the pause duration slightly
Pause before the next arrow

When loop is exited, display a message with the score
```