

Kit de programación Código Pi

Midamos el movimiento



Autoridades

Presidente de la Nación

Mauricio Macri

Jefe de Gabinete de Ministros

Marcos Peña

Ministro de Educación, Cultura, Ciencia y Tecnología

Alejandro Finocchiaro

Secretario de Gobierno de Cultura

Pablo Avelluto

Secretario de Gobierno de Ciencia, Tecnología e Innovación Productiva

Lino Barañao

Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología

Manuel Vidal

Secretaria de Innovación y Calidad Educativa

Mercedes Miguel

Subsecretario de Coordinación Administrativa

Javier Mezzamico

Directora Nacional de Innovación Educativa

María Florencia Ripani

ISBN en trámite



Este material fue producido por el Ministerio de Educación, Cultura, Ciencia y Tecnología en base a contenidos provistos sin cargo por la Fundación Raspberry Pi mediante licencias Creative Commons y han sido desarrollados en función de los Núcleos de Aprendizajes Prioritarios de educación digital, programación y robótica y los recursos tecnológicos propuestos en el marco del Plan Aprender Conectados.

Índice

Midamos el movimiento.....	5
¿Adónde estoy señalando?.....	7
Controlá el movimiento durante el transcurso del tiempo.....	8
Mostrar la orientación en la matriz led.....	9

Midamos el movimiento

La placa Sense HAT tiene un sensor de movimiento llamado IMU, que puede medir el tipo de movimiento que esté experimentando.

¿Qué es un sensor IMU?

La sigla **IMU** proviene del inglés y significa 'unidad de medición inercial'. En realidad, son tres sensores en uno:

- un giroscopio (mide la cantidad de movimiento y la rotación),
- un acelerómetro (mide las aceleraciones y se puede usar para encontrar la dirección de la gravedad),
- un magnetómetro (mide el campo magnético propio de la Tierra, que es algo parecido a una brújula).



Entonces, ¿por qué es importante un sensor de movimiento? Para que comprendas, te presentamos el siguiente ejemplo: en el espacio, hay una pregunta de fundamental importancia sobre la cual siempre debes saber la respuesta: **¿adónde está apuntando el objeto espacial?**

Si no sabés la orientación, estás en un gran problema; por eso se usa un sensor IMU como este en todas las naves espaciales tripuladas y controladas remotamente para rastrear los movimientos y para mantener un conocimiento de la orientación; incluso, la primera nave espacial tenía uno de estos sensores. Podés investigar acerca de las misiones Apolo que hicieron posible que los humanos aterrizaran en la superficie de la Luna.

Arriba figura una imagen de un sensor IMU del módulo de mando Apolo; compará el tamaño de esto con el sensor que se encuentra en la placa Sense HAT. Esta es una de las diferencias entre la tecnología de 1975 y la de la actualidad.

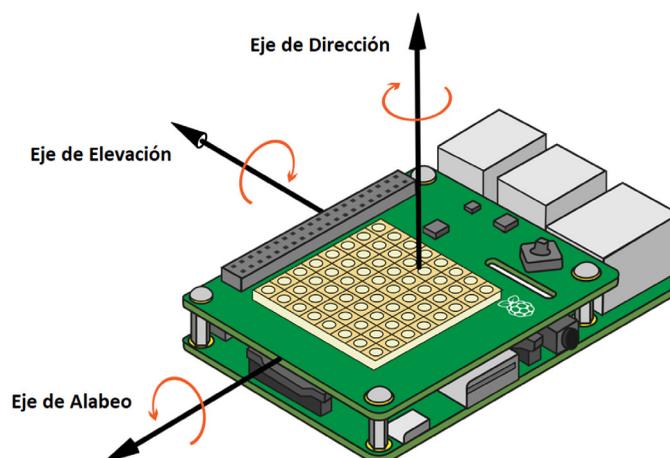
¿Cómo se representa la orientación?

La Tierra gira sobre un eje que va desde el Polo Norte hasta el Polo Sur. Todos los objetos en el espacio o en otro lugar tienen tres ejes en torno de los cuales giran. Si sabés cuánto giró cada eje, entonces sabés adónde está señalando el objeto.

Estos tres ejes son:

- **lateral (pitch):** como un avión cuando despega,
- **longitudinal (roll):** cuando el avión hace un giro de 360° sobre sí mismo,
- **vertical (yaw):** imaginate manejando el avión como un auto.

Mirá este breve [video](#) que muestra dónde están estos ejes en un avión. Tratá de imaginarte el avión apuntando a una dirección aleatoria. Para que el avión esté en esa posición, podés girarlo una cantidad determinada sobre su eje, para que alcance la orientación que imaginaste.



La imagen anterior muestra dónde están esos ejes en relación con la placa Sense HAT.

¿Adónde estoy señalando?

1. Hacé clic en **Menu** (Menú) > **Programming** (Programación) > **Python 3** (IDLE) para abrir una nueva ventana **Python Shell**.

2. Seleccioná **File** (Archivo) > **New Window** (Nueva ventana) e ingresá el siguiente código:

```
from sense_hat import
SenseHat sense =
SenseHat() sense.clear()
o = sense.get_orientation()
pitch =
o["pitch"]
roll =
o["roll"] yaw
= o["yaw"]
print("pitch {0} roll {1} yaw {2}".format(pitch, roll, yaw))
```

3. Seleccioná **File** (Archivo) > **Save** (Guardar) y elegí un nombre de archivo para tu programa.

4. Seleccioná **Run** (Ejecutar) > **Run module** (Ejecutar módulo).

5. Si ves el error `IMU Init Failed, please run as root / use sudo` en rojo, en la última línea, significa que no seguiste las instrucciones detalladas anteriormente. Cerrá todo y volvé al paso 1.

6. Ahora deberías ver algo como lo siguiente:

```
IMU Init Succeeded
pitch 356.35723002363454 roll 303.4986602798494 yaw
339.19880231669873
```

7. No son necesarios todos los números después de la coma, así que vamos a redondear. Justo antes de la línea:

```
print("pitch %s roll %s yaw %s" % (pitch, roll, yaw))
```

agregá estas líneas debajo:

```
pitch = round(pitch, 1)
roll = round(roll, 1) yaw = round(yaw, 1)
```

Controlá el movimiento durante el transcurso del tiempo

1. Sería bueno controlar cómo cambian los valores de los ejes con los movimientos, entonces ingresá tu código en un bucle `while` y ejecutalo otra vez:

```
while True:
    o = sense.get_orientation()
    pitch = o["pitch"]
    roll = o["roll"]
    yaw = o["yaw"]
    pitch = round(pitch, 1) roll =
    round(roll, 1) yaw =
    round(yaw, 1)
    print("pitch {0} roll {1} yaw {2}".format(pitch, roll, yaw))
```

2. Mové la placa Sense HAT que estás sosteniendo en la mano hacia un lado y otro, y deberías ver cómo cambian los números. Fijate si podés hacer que cambie un eje solamente, por ejemplo, moviéndola en dirección del eje lateral. Hacé lo mismo con los tres ejes. Presioná `Ctrl + C` para detener el programa.

Mostrar la orientación en la matriz led

1. Mostrar algo que está en modo 3D en modo 2D siempre es un desafío, en especial cuando el tamaño de tu pantalla es de solo 8 x 8 píxeles. Una manera que podría funcionar es tener un led para cada eje y luego hacer que se muevan en distintas direcciones. Por ejemplo:

- El led del eje lateral podría ir de arriba abajo.
- El led del eje longitudinal podría ir de lado a lado.
- El led del eje vertical podría dar vueltas por todo el perímetro.

2. Este es un truco inteligente que podés aplicar:

La tabla que figura a continuación muestra los números de leds secuenciales dispuestos en filas horizontales.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Podés convertir cualesquiera de esos números en sus coordenadas **X** **Y** usando el código que se muestra a continuación.

```
y = number // 8  
x = number % 8
```

Para obtener el valor **Y**, tenés que dividir **//** el número por 8, descartando la parte fraccional. Esta es una división entera y no tiene en cuenta el resto. Luego, para

el valor `x`, hacés la función de módulo `%` de 8, que te da únicamente el resto.

Por ejemplo (usando el número 60 que está en la fila de abajo):

```
60 // 8 = 7
60 % 8 = 4
```

3. Probá con este código:

```
number = 60
y = number // 8
x = number % 8
sense.set_pixel(x, y, 255, 255, 255)
```

4. El truco es hacer una lista que tenga los números de leds para que la ruta que vos querés se mueva de un lado a otro. Supongamos que querés hacer que los leds giren por todo el perímetro; tendrías que leer los números que figuran en la parte de arriba de la tabla, hacia abajo en el lateral derecho, hacia atrás a lo largo de la parte de abajo y luego hacia arriba en el lateral izquierdo. Entonces sería:

```
edge = [0, 1, 2, 3, 4, 5, 6, 7, 15, 23, 31, 39, 47, 55, 63,
62, 61, 60, 59, 58, 57, 56, 4
```

Podemos encontrar la longitud de la lista usando la función `len`:

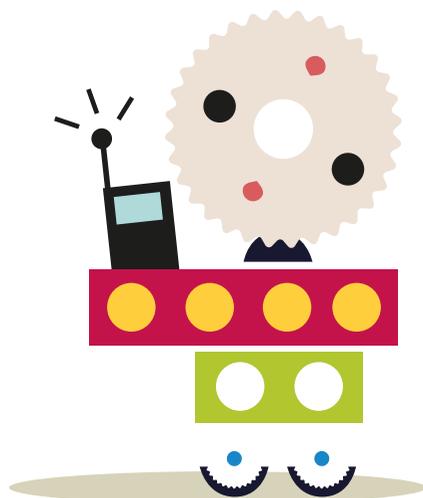
```
length = len(edge)
```

Vemos que la longitud es 28. Si dividimos 28 por 360, tendremos un coeficiente entre, digamos, la medición del eje vertical y las posiciones en nuestra lista (qué tan lejos vamos en el perímetro). Después podemos obtener el número de píxel secuencial de la lista en la posición calculada; probá sus coordenadas y encendé el led. De esta manera:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
edge = [0, 1, 2, 3, 4, 5, 6, 7, 15, 23, 31, 39, 47, 55, 63,
62, 61, 60, 59, 58, 57, 56, 4] length = len(edge)
ratio = length / 360.0
while True:
o = sense.get_orientation()
pitch = o["pitch"]
roll = o["roll"]
yaw = o["yaw"]
yaw_list_position = int(yaw * ratio)
yaw_pixel_number = edge[yaw_list_position]
y = yaw_pixel_number // 8 x = yaw_pixel_number % 8
sense.set_pixel(x, y, 255, 255, 255)
```

5. Lo que vas a notar es que el código que figura arriba solo enciende los leds, pero vas a tener que resolver cómo apagarlos vos mismo. Tratá de tener una variable para la x y la y anteriores de la última vez alrededor del bucle y, si es diferente de las nuevas x e y, usá `set_pixel` para configurar el led como negro.

La fundación Raspberry Pi suministra contenidos de aprendizaje de programación sin cargo. Encuentre más información en <https://projects.raspberrypi.org/en/> (inglés)



**APRENDER
CONECTADOS**



Ministerio de Educación,
Cultura, Ciencia y Tecnología
Presidencia de la Nación