



Redes Ambientales y Computación Ubicua

Bloque I. Computación Ubicua

Sesión 3

2018-19

I. Introducción

- Descripción de las sesiones
 - **Sesión 1:**
 1. Introducción a Raspberry Pi – Comp. Ubicua
 2. Tutorial de Python con algunos ejercicios
 - **Sesión 2:**
 1. Ampliación info sobre RaspPi
 2. Audio con Raspberry Pi
 3. Planteamiento y codificación del ejercicio “*Entrega1*”
 - **Sesión 3:**
 0. Información sobre cómo trabajar en esta sesión.
 1. Sense Hat
 2. Tempo
 3. Planteamiento y codificación del ejercicio “*Entrega2*”

CONTENIDOS

0. Información sobre cómo trabajar en esta sesión.

1. Sense Hat
2. Obtención del Tempo
3. Entregable 2

0. Como trabajar en esta sesión

En esta tercera sesión se recomienda trabajar empleado tanto los notebooks proporcionados como ejecutando comandos desde un terminal de la Raspberry.

Para ello, ejecute en primer lugar una conexión remota con la Raspberry Pi (ssh). Puede hacerlo desde Windows o desde Linux.

- Desde Windows necesitará una aplicación como Putty o similares. Descárguela, ejecútela y siga las instrucciones.
- Desde Linux debe abrir un terminal y ejecutar

`ssh pi@192.168.1.X` Siendo X el número de su equipo.

Una vez conectado a su equipo remotamente puede ejecutar jupyter con

`jupyter-notebook &`

Ahora podrá abrir un navegador en su PC y acceder a los notebooks bajo la url

`192.168.1.X:8888`

CONTENIDOS

1. Sense Hat
2. Obtención del Tempo
3. Entregable 2

2. Sense HAT

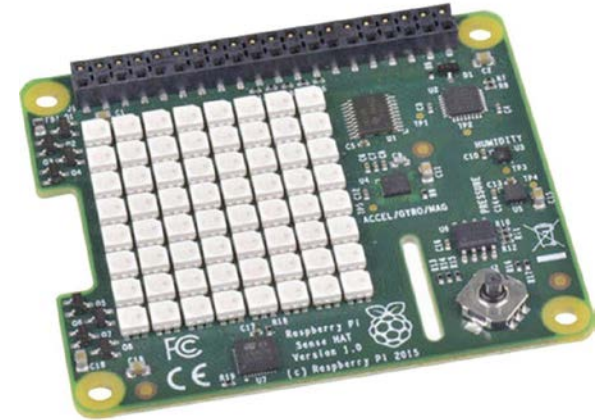
Placa (shield) – GPIO Características:

- Sensores (Cola FIFO) / I2C
 - sensor de presión (24 bits)
 - Sensor de temperatura -40 +120°C (16 bits).
 - Otro sensor de temperatura+sensor de humedad (16 bits ambos)
 - IMU (Inertial Measurement Unit 9)
 - Sensor de aceleración (16 bits, 3D)
 - Giroscopio (16 bits, 3D)
 - Campo magnético (16 bits, 3D)
- Joystick (←↑→↓)
- matriz LED de 8 por 8 elementos y con 15 bit de resolución RGB
- Librería completa en Python (<https://pythonhosted.org/sense-hat/>)

Referencias

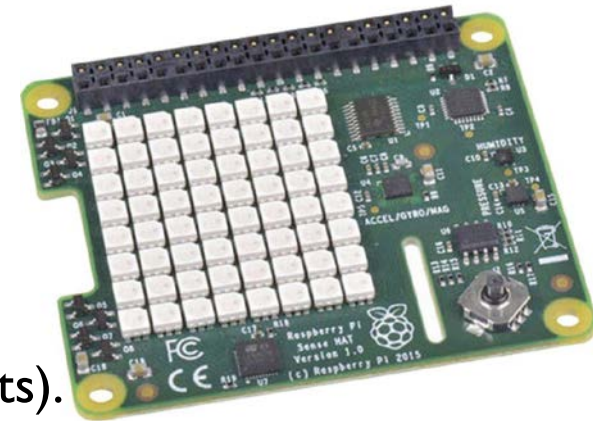
<https://www.raspberrypi.org/documentation/hardware/sense-hat/>

<https://www.raspberrypi.org/learning/astro-pi-guide/>



2. Sense HAT

- **Sensores (Cola FIFO) / I2C**
 - sensor de presión (24 bits)
 - Sensor de temperatura -40 +120°C (16 bits).
 - Otro sensor de temperatura+sensor de humedad (16 bits ambos)
 - IMU (Inertial Measurement Unit 9 DoF)
 - Sensor de aceleración (16 bits, 3D)
 - Giroscopio (16 bits, 3D)
 - Campo magnético (16 bits, 3D)
- Joystick (←↑→↓)
- matriz LED de 8 por 8 elementos y con 15 bit de resolución RGB
- Librería completa en Python (<https://pythonhosted.org/sense-hat/api/>)



Referencias

<https://www.raspberrypi.org/documentation/hardware/sense-hat/>

<https://www.raspberrypi.org/learning/astro-pi-guide/>

2. Sense HAT

Emulador

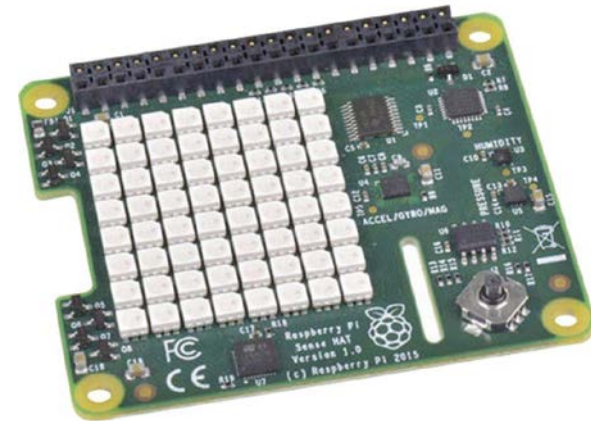
<https://trinket.io/sense-hat>

[Leer también](#)

<https://www.raspberrypi.org/blog/desktop-sense-hat-emulator/>

<https://www.raspberrypi.org/blog/sense-hat-emulator/>

<https://www.deviceplus.com/how-tos/raspberrypi-guide/the-sense-hat-add-on-board-for-raspberry-pi-sense-hat-emulator/>



2. Sense HAT

Instalación

- `sudo apt update`
- `sudo apt install sense-hat`
- `sudo pip3 install pillow`
- `sudo reboot` (sólo si no estaba instalado todo lo anterior)

- Ejemplos instalados en `/usr/src/sense-hat/examples`
- Copiar a home con `cp /usr/src/sense-hat/examples ~/ -a`
 - Ejemplos en python (`./nombredelprograma.py`)
 - en C/C++ (se compilan con `make` y se ejecutan `./nombredelEjecutable`)

<https://pythonhosted.org/sense-hat/>

2. Sense HAT

Primer programa: Hola Mundo (crear un archivo de texto)

hola.py

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_message("Hola Mundo!")
```

Ejecutar

```
python3 hola.py
```

Calibración de componentes:

```
cd  
cp /usr/share/librtimulib-utils/RTEllipsoidFit ./ -a  
cd RTEllipsoidFit  
RTIMULibCal
```

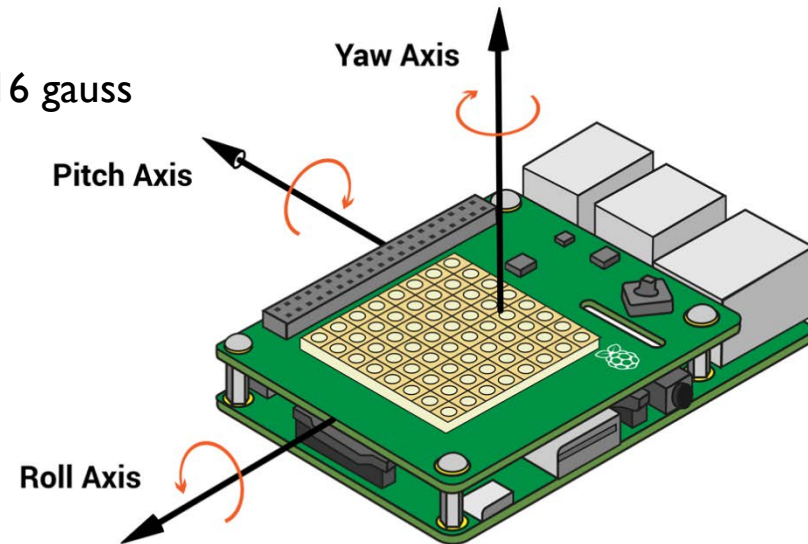
Referencia

<https://www.raspberrypi.org/documentation/hardware/sense-hat/README.md>

2. Sense HAT

IMU (9DoF)

- Inertial Measurement Sensor: ST LSM9DSI (16 bits)
- 3D Orientación- Gyro – Angular rate sensor $\pm 245/500/2000$ dps
- 3D Acc $\pm 2/\pm 4/\pm 8/\pm 16$ g
 - Pitch
 - Roll
 - Yaw
- 3D magnetic field sensor $\pm 2/\pm 4/\pm 8/\pm 16$ gauss

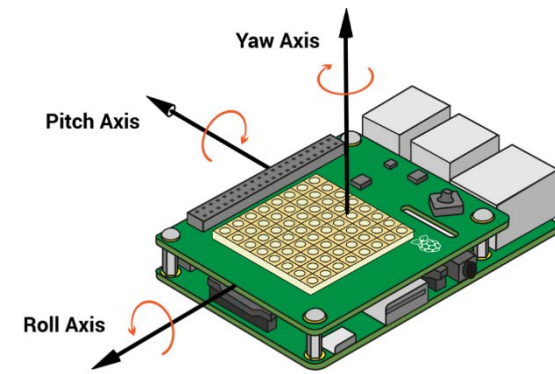


Referencias

<http://www.st.com/web/en/resource/technical/document/datasheet/DM00103319.pdf>

2. Sense HAT

IMU (9DoF)



I. Ejemplo de uso de la IMU

```
sudo pip3 install pi3d  
git clone git://github.com/astro-pi/apollo-soyuz  
cd apollo-soyuz  
sudo ./soyuz.py
```

Revise el código para comprender qué hace.

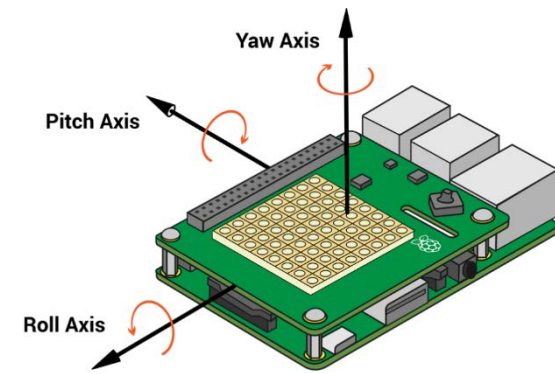
Teclas asociadas: m, g, a, =, -

Referencias

<https://www.raspberrypi.org/learning/astro-pi-guide/sensors/movement.md>

2. Sense HAT

Presión, temperatura y humedad



- **Barometric pressure and temperature sensor: ST LPS25H**
 - Presión: en pascles o milibares ($1 \text{ Pa} = 10^{-2} \text{ mbar}$)
 - Temperatura en $^{\circ}\text{C}$ $-40 + 120^{\circ}\text{C}$ (16 bits)
- **Relative humidity and temperature sensor: ST HTS221**
 - Temperatura en $^{\circ}\text{C}$ $-40 + 120^{\circ}\text{C}$ (16 bits)
 - Humedad relativa

Ejemplos de prueba en <https://github.com/raspberrypilearning/astro-pi-guide>
Pressure, humidity, temperature, movement.

Referencias

<http://www.st.com/web/en/resource/technical/document/datasheet/DM00066332.pdf>

<http://www.farnell.com/datasheets/1836732.pdf>

2. Sense HAT

Información extra y tutoriales

Referencias

<https://github.com/RPi-Distro/python-sense-hat/blob/master/examples/README.md>

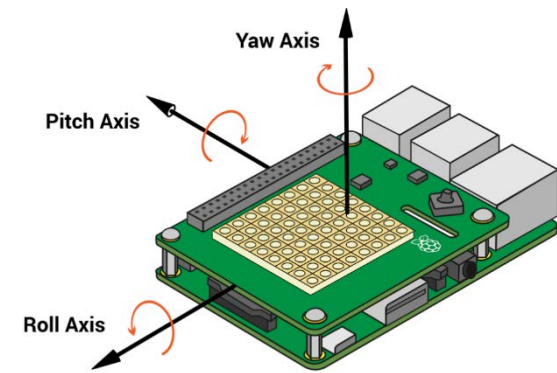
<https://github.com/bennuttall/sense-hat-examples/tree/master/python>

<https://www.raspberrypi.org/documentation/hardware/sense-hat/README.md>

<https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat>

<https://www.hackster.io/raspberry-pi/products/sense-hat>

https://es.pinout.xyz/pinout/sense_hat



CONTENIDOS

1. Sense Hat
2. Obtención del Tempo
3. Entregable 2

3. Obtención del Tempo/Ritmo/Beat

Términos relacionados

¿Qué es el **ritmo**? Definición intuitiva – repetición periódica de elementos débiles y fuertes (ej. Ritmo cardiaco).

Tempo – (ppm / bpm) Termino musical que establece la velocidad a la que debe interpretarse una pieza musical.

Beat?

Físicamente - Componente de baja frecuencia de la música.

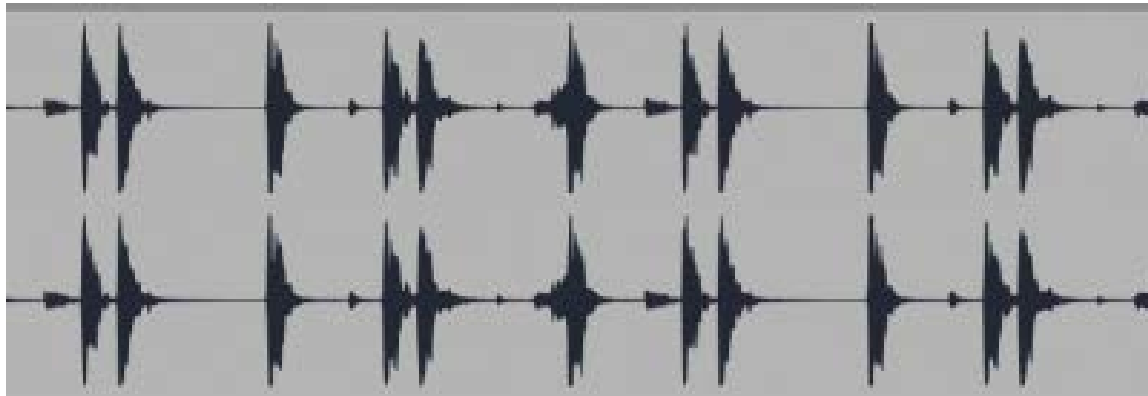
Problema para analizarlo matemáticamente – Diferentes algoritmos.

3. Obtención del Tempo/Ritmo/Beat

Beat

Intuitivamente, se asocia “beats” a cambios en la Energía sonora significativos con respecto al resto.

- Picos en la energía sonora.
- Normalmente asociados a instrumentos graves como bajo, batería.
- Frecuencias Bajas → del orden de 0 a 6Hz.



3. Obtención del Tempo/Ritmo/Beat

Trabajos relacionados (evaluación de 23 métodos).

	Aubio	BeatIt	Beatroot	Ellis	Essentia	Fixedtempo	IBT	jAudio	MIRTempo	Mpeg7-xm	qmtempo	Tzanetakis
Evaluation 1	39,35%	60,43%	23,23%	45,59%	24,51%	24,73%	35,91%	5,16%	30,97%	48,39%	43,23%	25,59%
Evaluation 2	67,31%	78,28%	67,96%	80,65%	83,44%	50,75%	79,78%	32,26%	65,59%	70,54%	80,43%	66,45%
Evaluation 3	67,31%	78,49%	69,03%	81,29%	85,37%	50,75%	80,43%	36,56%	65,59%	70,54%	80,86%	67,74%
	AlonsoACF	AlonsoSP	DixonACF	DixonI	DixonT	Klapuri	Scheirer	Tzan_H	Tzan_MM	Tzan_MS	Uhle	
Evaluation 1	23,44%	37,42%	16,99%	28,60%	19,35%	58,49%	37,85%	21,29%	18,71%	27,53%	41,94%	
Evaluation 2	58,28%	68,60%	76,99%	62,58%	68,82%	91,18%	69,46%	47,74%	41,08%	52,47%	71,83%	
Evaluation 3	59,35%	69,25%	81,29%	64,95%	71,83%	91,18%	70,11%	51,40%	42,58%	53,55%	72,47%	

Table 4: Evaluations Results

Fuente: Comparative evaluation and combination of audio tempo estimation approaches (2011). Jose R. Zapata et al.

3. Obtención del Tempo/Ritmo/Beat

Primera aproximación, usando el comando
soundstretch de SoundTouch
(audio processing library)

3. Obtención del Tempo/Ritmo/Beat

SoundTouch audio processing library

1. Descripción de la librería

SoundTouch librería de código abierto para el procesamiento de audio diseñada para cambiar el Tempo, el Pitch y el Playback de archivos de audio y flujos (streaming).

Permite desarrollar aplicaciones pero también incluye un comando por consola que permite procesar archivos de audio en formato **.wav**.

2. Posible utilización en la asignatura

Opción a) Crear un código que haga uso de la librería para obtener el tempo.
Opcionalmente puede enviarlo a otro equipo.

Opción b) Usar el comando por consola en un Shell-script para obtener el tempo.
Opcionalmente puede enviarlo a otro equipo.

Fuente: <https://www.surina.net/soundtouch/index.html>

3. Obtención del Tempo/Ritmo/Beat

SoundTouch audio processing library

3. ¿Cómo obtener archivos para hacer las pruebas?

El comando youtube-dl permite descargar vídeos de youtube (o audio de estos, que es lo que interesa en esta asignatura).

Para usarlo basta con ejecutar en un terminal (todo va en la misma línea)

```
youtube-dl -x --audio-format wav
```

```
https://www.youtube.com/watch?v=gsJEMH\_emBM
```

Descargará el audio de ese vídeo en formato wav, con el nombre

60 BPM Metronome-gsJEMH_emBM.wav

4. Instalación de SoundStretch

```
sudo apt-get install soundstretch
```

5. Comprobación

```
soundstretch 60 BPM Metronome-gsJEMH_emBM.wav -bpm
```

Debe dar un valor de 60 BPMs

3. Obtención del Tempo/Ritmo/Beat

SoundTouch audio processing library

6. Capturando audio y procesando con soundstretch

Desde un terminal, se puede ejecutar estos dos comandos

```
arecord -d 10 -F 10 -D plughw:1,0 -f cd capt.wav  
soundstretch capt.wav -bpm
```

Para guardar en un archivo de texto la información mostrada por pantalla, añade al final del comando

```
& tee info.txt
```

7. Se puede procesar ese archivo con AWK para obtener sólo el valor de los BPMs

```
awk 'BEGIN {BPM=0} $1==Detected" {BPM=$4} END  
{print BPM}' info.txt
```

8. Empleado dentro de un script (ver anexo I a continuación)

9. Empleado en un programa en Python (se verá después).

3. Obtención del Tempo/Ritmo/Beat

SoundTouch.ANEXO I. Script disponible en el aula virtual
scriptConsola.sh

```
#!/bin/bash
if [ "$#" -ne 1 ]; then
    echo "Usage: $0 Time to capture" >&2
    exit 1
fi
while [ 1 ]
do
    echo "1. captura de 10 segundos de audio "
    echo "  arecord -d" $1 " -F 10 -D plughw:1,0 -f cd capt.wav"
    arecord -d $1 -F 10 -D plughw:1,0 -f cd capt.wav

    echo "2. Obtencion de los BPMs"
    echo "  soundstretch capt.wav -bpm |& tee info.txt "
    soundstretch capt.wav -bpm |& tee info.txt

    echo "3. Awk para obtener el valor numerico "
    echo "  awk ' BEGIN {BPM=0}  \$1== " Detected " {BPM=\$4} END {print BPM} ' info.txt"
    BPMs="$(awk ' BEGIN {BPM=0} \$1=="Detected" {BPM=\$4} END {print BPM}' info.txt)"

    echo "valos BPMs " $BPMs
    if [ "$BPMs" == "0" ]; then
        echo "No encontro los BMPs"
    else
        echo "enviando al servidor" $BPMs
        exec 3<>/dev/udp/192.168.1.137/4952
        echo -e $BPMs>&3
    fi
done
```

Código para enviar a
otro equipo.

3. Obtención del Tempo/Ritmo/Beat

SoundTouch. Comando con un programa de python

Ya se vio cómo emplear `os.system` en sesiones anteriores, sin embargo en esta ocasión se empleará `subprocess`. Se puede usar de la siguiente forma:

(antes renombrar el archivo `.wav` a `60.wav` con el comando `mv`)

```
import os
import subprocess
from subprocess import PIPE, Popen

proceso = Popen([ 'soundstretch', '60.wav', '-bpm'],
                stdout=PIPE, stderr=PIPE)

errores = proceso.stderr.read()
salida = proceso.stdout.read()

print ("ERRORES ")
print (errores)
print ("SALIDA ")
print (salida)
```

Referencias: <http://python-para-impacientes.blogspot.com.es/2014/02/ejecutar-un-comando-externo.html>
<https://docs.python.org/3/library/subprocess.html>
http://librosweb.es/libro/python/capitulo_10/modulos_de_sistema.html

3. Obtención del Tempo/Ritmo/Beat

Segunda aproximación, Obtención del tempo con
AUBIO

3. Obtención del Tempo/Ritmo/Beat

Obtención del tempo con AUBIO

La librería AUBIO también tiene una herramienta para calcular el tempo de una canción (ver ejemplo **demo_bpm_extract()**, hay una copia en el aula virtual.). Se puede invocar de la siguiente forma:

```
python3 demo_bpm_extract.py archivo.wav -m default
```

- Tiene tres modos de funcionamiento (opción -m).
- Cuidado con el valor de sampleRate, si no coincide no funciona.
 - Para conocer los datos del archivo (sampleRate entre otros) puede usar el comando file

```
file 60.wav
```

 (dirá 48000 Hz posiblemente).

Referencias:

<https://github.com/aubio/aubio/tree/master/python/demos>

CONTENIDOS

1. Sense Hat
2. Obtención del Tempo
3. Entregable 2



3. Entregable 2

Entregable 2. Detección de los BPM y actuación sobre sense Hat.

Para este entregable debe realizar un código que,

1. Cree una función llamada `tempo` que admita como parámetro único el tiempo de captura (si desea añadir otros, puede hacerlo, pero como mínimo debe tener el tiempo en segundos).
2. Esta función debe:
 1. Capturar el sonido a través del micrófono,
 2. Procesar esa información para obtener los BPM.
 3. Devolver un valor de retorno que contenga el tempo.
 4. Con ese valor de retorno, el código principal debe mostrar en el display del senseHat el valor numérico del tempo.



3. Entregable 2

Entregable 2. Opciones de desarrollo.

En el código proporcionado a lo largo de las sesiones de este bloque se encuentra información sobre:

1. Como capturar el audio y guardarlo en un archivo de diversas formas.
 1. Con llamadas al SO.
 2. Empleando PyAudio.
2. Respecto a la obtención de los BPM, se proporcionan ejemplos para poder hacerlo de las siguientes formas
 1. Llamando al SO con el comando soundstretch (requiere tener un archivo guardado).
 2. Empleando aubio (el ejemplo proporcionado también requiere tener un archivo guardado).



3. Entregable 2

Entregable 2. Opciones de desarrollo - Evaluación.

Por lo tanto, el código se puede desarrollar con las siguientes variantes, incrementando el orden de complejidad (que se traducirá en una mayor nota según la complejidad. Todas sobre 10):

1. El código captura el audio con llamada al SO, lo guarda en archivo y lo procesa con soundstretch mediante una llamada al SO. Puntuación máxima de 4.
2. El código captura el audio con llamada al SO, lo guarda en archivo pero se procesa con aubio. Puntuación máxima de 5.
3. El código captura el audio con PyAudio, lo guarda en archivo y lo procesa con soundstretch mediante llamada al SO. Puntuación máxima de 5.
4. El código captura el audio con PyAudio, lo guarda en archivo y lo procesa con aubio desde ese archivo. Puntuación máxima de 6.
5. El código captura el audio con PyAudio, y sin pasar por guardado en archivo, se procesa con aubio. Puntuación máxima de 10.