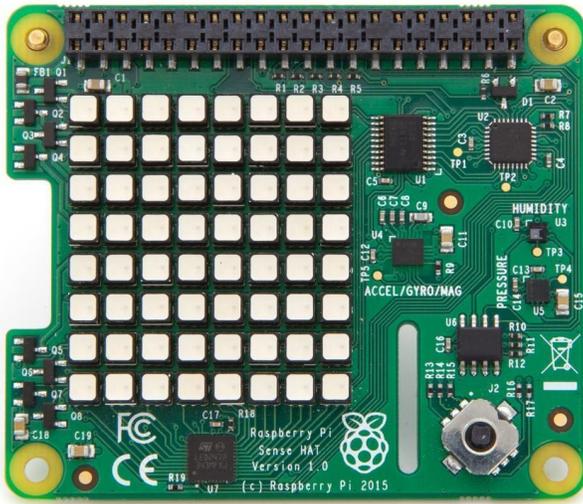


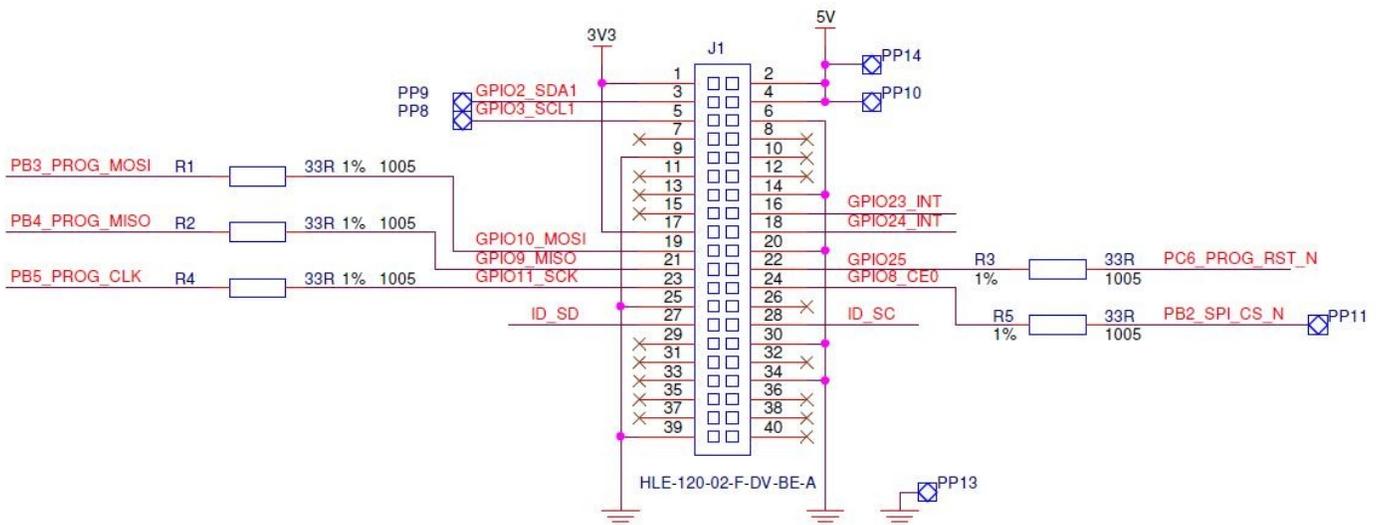
RASPBERRY E SENSE-HAT

Una delle schede accessorie più interessanti della raspberry, è la **sense hat**.

La scheda possiede un connettore femmina e dei fori di fissaggio adattaibili alla scheda Raspberry.



La scheda si interfaccia con la Raspberry tramite l'I2C e l'SPI, in tutti i piedini utilizzati sono quelli in figura.



Pertanto i pin 3-5-19-21-22-23-24-27-28 sono utilizzati dalla scheda sense hat e non potranno essere utilizzati per altre applicazioni.

La scheda utilizza inoltre i piedini 1 e 17 (3,3Volt) 2 e 4 (5Volt) e 6,9,14,20,25,30,34,39 (GND).

La prima operazione da fare per poter utilizzare questa scheda è attivare il protocollo I2C nella configurazione della scheda Raspberry.

Questa operazione si può effettuare sia da terminale con il comando **sudo raspi-config** accedendo alla sezione **interfacing options**, che da interfaccia grafica tramite **preferenze-Raspberry pi configuration** accedendo sempre alla sezione **interfaces**. In entrambi i casi al termine dell'attivazione del protocollo I2C occorre fare il reboot della scheda.

La seconda operazione da fare è quella di installare le librerie per la gestione della scheda tramite i seguenti comandi da terminale:

```
sudo apt-get update          scarica la lista aggiornata dei pacchetti presenti nelle repository
sudo apt-get install sense-hat per installare le librerie
sudo reboot                 per riavviare
```

A questo punto abbiamo a disposizione tutti i dispositivi presenti sulla scheda sense-hat.

Dopo l'installazione potremo trovare i programmi di esempio in linguaggio C nella cartella:
/usr/src/sense-hat/examples

Possiamo spostarci sotto la cartella con i file di esempio e compilarli utilizzando il comando:

```
sudo gcc nomefile.c -o nomefile.out
```

lanciando poi il programma eseguibile digitando: **./nomefile.out**

Ad esempio possiamo provare a compilare l'esempio snake.c, scrivendo nel terminale:

```
cd /home/pi <invio>
sudo gcc /usr/src/sense-hat/examples/snake/snake.c -o /home/pi/snake.out <invio>
./snake.out <invio>
```

E' inoltre presente un programma di esempio che possiamo lanciare con il seguente comando da terminale:

```
RTIMULibDrive11
```

A bordo della scheda possiamo trovare i seguenti dispositivi:

- Un I.M.U. (IMU (Inertial **M**asurement **U**nit) cioè un sistema per il controllo della posizione composto da:
 - Giroscopio
 - Accelerometro
 - Sensore di campo magnetico
- Un barometro range 260 - 1260 hPa (ettoPascal corrispondente al mBar)
- Un sensore di temperatura, range 0-65 °C
- Un sensore di umidità relativa
- Uno schermo a matrice di LED RGB 8x8
- Un joystick a 5 pulsanti

Per la gestione di questi componenti hardware viene fornita la libreria **SenseHat** per Python.

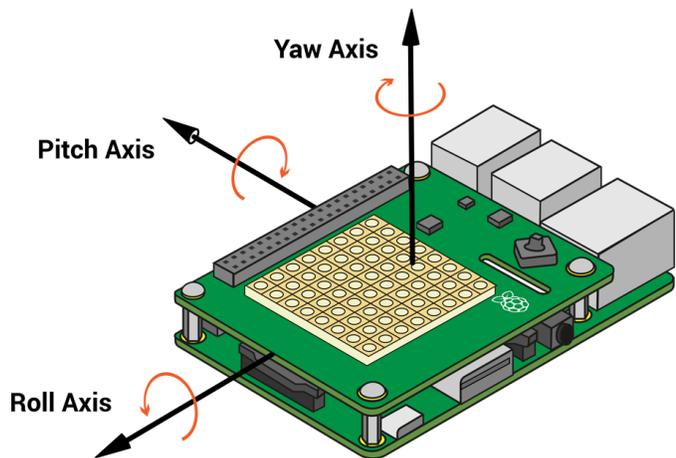
Vediamo di seguito i programmi di esempio per l'utilizzo di tutti i dispositivi hardware menzionati sopra.

GIROSCOPIO

Il giroscopio fornisce i 3 valori angolari in gradi indicati in figura, **Pitch**, **Roll** ed **Yaw**.

Possiamo testare il suo funzionamento scrivendo il seguente programma in Python:

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



Possiamo scrivere il programma da terminale con il comando **sudo nano giroscopio.py**

Salvando poi il programma con la sequenza di tasti:

- **CTRL+X**
- **S**
- **invio**

Per lanciare il programma scrivere da terminale: `python giroscopio.py` per uscire dal programma premere **CTRL+C** sulla finestra del terminale

ACCELEROMETRO

L'accelerometro ci fornisce l'accelerazione espressa in g nelle 3 direzioni X,Y e Z.

Anche in questo caso possiamo testare il funzionamento scrivendo come fatto precedentemente un programma chiamato **accelerometro.py** con il seguente contenuto.

```
from sense_hat import SenseHat
sense = SenseHat()
while True:
    acceleration = sense.get_accelerometer_raw()
    x = acceleration['x']
    y = acceleration['y']
    z = acceleration['z']

    x=round(x, 0)    #arrotondamento al numero di decimali scelto
    y=round(y, 0)
    z=round(z, 0)

    print("x={0}, y={1}, z={2}".format(x, y, z))
```

SENSORE MAGNETICO (BUSSOLA)

Il sensore magnetico fornisce l'angolo in gradi rispetto al nord magnetico terrestre.

Il programma di esempio è il seguente:

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()  
while True:  
    nord = sense.get_compass()  
    print("Nord: %s" % nord)
```

SENSORE DI PRESSIONE (BAROMETRO)

Il sensore di pressione fornisce la pressione espressa in ettoPascal (corrispondente al mBar)

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()  
while True:  
    pressione= sense.get_pressure()  
    print("Pressione: %s" % pressione)
```

SENSORE DI TEMPERATURA (TERMOMETRO)

Il sensore di temperatura fornisce la temperatura espressa in gradi centigradi.

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()  
while True:  
    temp= sense.get_temperature()  
    print("Temperatura: %s" % temp)
```

SENSORE DI UMIDITA' (IGROMETRO)

Il sensore di temperatura fornisce l'umidità relativa in percentuale. Cioè il rapporto tra vapore acqueo presente e massima quantità di vapore acqueo che la stessa massa d'aria riesce a contenere, il 100% indica che è presente solo acqua.

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()  
while True:  
    umid= sense.get_humidity()  
    print("Umidità: %s" % umid)
```

DISPLAY A MATRICE DI LED RGB

Il display è composto da 64 LED RGB disposti come nella matrice in figura.

I LED possono assumere un diverso colore a seconda della quantità di Rosso (R) Verde (G) e Blu (B) espresse singolarmente con un valore che va da 0 a 255.

L'istruzione `sense.set_pixel(R,G,B)` ad esempio definisce il colore di un pixel

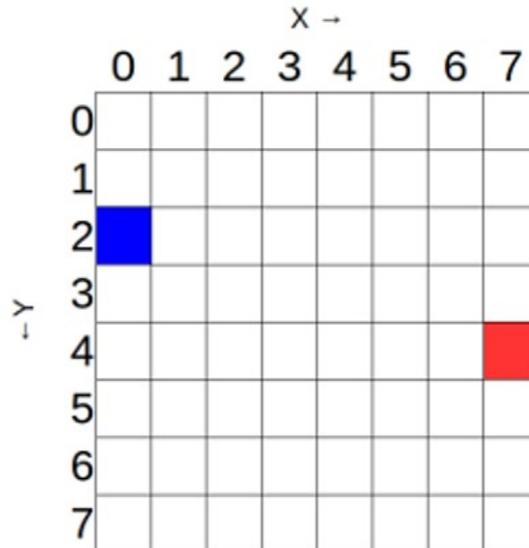
```
from sense_hat import SenseHat
sense = SenseHat()
```

```
blue = (0, 0, 255)
```

```
red = (255, 0, 0)
```

```
sense.set_pixel(0, 2, blue)
```

```
sense.set_pixel(7, 4, red)
```



La stessa funzione può essere utilizzata definendo prima la matrice di pixel nel seguente modo:

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

```
# definisco dei colori
```

```
g = (0, 255, 0) # Green
```

```
b = (0, 0, 0) # Black
```

```
# definisco il colore dei led della matrice
```

```
immagine = [
```

```
    g, g, g, g, g, g, g, g,
```

```
    g, g, g, g, g, g, g, g,
```

```
    g, b, b, g, g, b, b, g,
```

```
    g, b, b, g, g, b, b, g,
```

```
    g, g, g, b, b, g, g, g,
```

```
    g, g, b, b, b, b, g, g,
```

```
    g, g, b, b, b, b, g, g,
```

```
    g, g, b, g, g, b, g, g
```

```
]
```

```
sense.set_pixels(immagine)
```

L'immagine visualizzata potrà essere specchiata in orizzontale con la funzione ***sense.flip_h()*** in verticale con ***sense.flip_v()*** e ruotata con ***sense.set_rotation(angolo)*** con quest'ultimo comando si definisce il valore di rotazione in maniera assoluta e non relativa, pertanto eseguendo più volte ***sense.set_rotation(90)*** avremo solo una rotazione alla posizione di 90°.

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

```
w = (150, 150, 150)  
b = (0, 0, 255)  
e = (0, 0, 0)
```

```
image = [  
e,e,e,e,e,e,e,e,  
e,e,e,e,e,e,e,e,  
w,w,w,e,e,w,w,w,  
w,w,b,e,e,w,w,b,  
w,w,w,e,e,w,w,w,  
e,e,e,e,e,e,e,e,  
e,e,e,e,e,e,e,e,  
e,e,e,e,e,e,e,e  
]
```

```
sense.set_pixels(image)  
while True:  
    sleep(1)  
    sense.set_rotation(90)  
    sleep(1)  
    sense.flip_h()  
    sleep(1)  
    sense.flip_v()
```

Per definire il colore dello sfondo del display invece si può utilizzare la funzione ***sense.clear()*** nel seguente modo

```
from sense_hat import SenseHat  
sense = SenseHat()  
r = 255  
g = 255  
b = 255  
sense.clear((r, g, b))
```

Per visualizzare una lettera invece si utilizza la funzione ***sense.show_letter("carattere")***

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_letter("Z")
```

Per visualizzare un messaggio la funzione è ***sense.show_message("testo")***

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.show_message("Hello world")
```

La funzione ***sense.show_message*** ha anche altri parametri per definire il colore del testo dello sfondo ed il movimento della scritta (*scroll_speed*).

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

```
blue = (0, 0, 255)  
yellow = (255, 255, 0)
```

```
while True:
```

```
    sense.show_message("Hello world", text_colour=yellow, back_colour=blue, scroll_speed=0.05)
```

Se voglio visualizzare il valore di una variabile posso sempre utilizzare la stessa funzione trasformando la variabile in una stringa:

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

```
while True:
```

```
    valore=100,3  
    messaggio=str(valore)  
    sense.show_message(messaggio)
```

JOYSTICK

Il Joystick ha 5 pulsanti, 4 per le direzioni ed il centrale la funzione è **`sense.stick.get_events()`** he restituisce gli eventi di direzione (direction) ed azione (action).

Direction può valere **up**, **down**, **right**, **left** e **medium**.

Action può valere **pressed** (premuta) **released** (rilasciato) ed **held** (mantenuto premuto)

Con il seguente esempio visualizziamo sul terminale lo stato del Joystick

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

while True:
    for event in sense.stick.get_events():
        print(event.direction, event.action)
```

Con il seguente esempio visualizziamo sul display a LED lo stato del Joystick

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

e = (0, 0, 0)
w = (255, 255, 255)

sense.clear()
while True:
    for event in sense.stick.get_events():
        if event.action == "pressed": #controllo la pressione del tasto
            # Controllo la direzione
            if event.direction == "up":
                sense.show_letter("U") # Up arrow
            elif event.direction == "down":
                sense.show_letter("D") # Down arrow
            elif event.direction == "left":
                sense.show_letter("L") # Left arrow
            elif event.direction == "right":
                sense.show_letter("R") # Right arrow
            elif event.direction == "middle":
                sense.show_letter("M") # Enter key

    sleep(0.5)
    sense.clear()
```

EMULATORE SENSE HAT

Nella scheda Raspberry è presente un simulatore dell Sense_Hat, per utilizzarlo bisogna eseguire le seguenti operazioni:

- 1) Aprire il simulatore presente nel menù programmazione
- 2) Lanciare l'IDE di Python presente nel menù programmazione
- 3) Aprire un nuovo file dall'IDE (file, new file)
- 4) Scrivere il programma modificando la riga iniziale:

```
from sense_hat import SenseHat
```

```
con
```

```
from sense_emu import SenseHat
```

- 5) Salvare il file
- 6) Nel menù **RUN** fare **RUN Module**

A questo punto nell'emulatore verrà avviato il programma scritto.

I programmi di esempio e tutte le informazioni sono tratte dal sito ufficiale della Raspberry ai seguenti link:

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

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