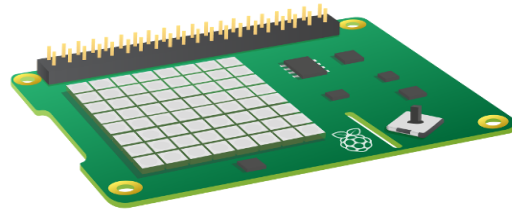


ΕΙΣΑΓΩΓΗ

Το Sense Hat αποτελεί ένα από τα κύρια κομμάτια του πειράματος. Είναι μια πλακέτα που συνδέεται με το Raspberry Pi και περιλαμβάνει ένα ενσωματωμένο χειριστήριο (joystick), και ένα προγραμματιζόμενο LED Matrix, στο οποίο μπορούν να προβληθούν οι μετρήσεις, και όχι μόνο, από τους ενσωματωμένους αισθητήρες του Sense Hat, οι οποίοι είναι:

1. Γυροσκόπιο
2. Βαρόμετρο
3. Θερμόμετρο
4. Μαγνητόμετρο
5. Υγρασιόμετρο
6. Φωτοαισθητήρα
7. Επιταχυνσιόμετρο



Προκειμένου να μπορέσουμε να λάβουμε δεδομένα από το Sense Hat, θα χρειαστούμε το λειτουργικό σύστημα Raspbian, οποιαδήποτε έκδοση της Python 3, καθώς και την δικιά του βιβλιοθήκη, την `sense_hat`, η οποία περιλαμβάνεται στο λειτουργικό.

Σε αυτόν τον συνοπτικό οδηγό, αναλύουμε πως να επικοινωνούμε με το Sense Hat, να λαμβάνουμε δεδομένα από τους αισθητήρες του και να εμφανίζουμε μηνύματα και άλλα δεδομένα που έχουμε συλλέξει, στην οθόνη του, ενώ στο τέλος υπάρχει μια συνοπτική λίστα με τις πιο κρίσιμες εντολές.

Πηγές:

- https://esamultimedia.esa.int/docs/edu/T05.2_Meet_the_Sense_HAT.pdf
- https://esamultimedia.esa.int/docs/edu/T05.3_How_to_collect_data_from_astropi.pdf
- <http://wsn.spaceflight.esa.int/docs/Factsheets/30%20ECLSS%20LR.pdf>
- <https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense-hat/0>

ΕΜΦΑΝΙΣΗ ΜΥΝΗΜΑΤΟΣ ΚΑΙ Η ΕΝΤΟΛΗ SLEEP

Προκειμένου να δείξουμε τις μετρήσεις μας, θα χρησιμοποιήσουμε την 8x8 LED οθόνη του Sense Hat. Αρχικά λοιπόν, για να εμφανίσουμε ένα μήνυμα στην οθόνη, θα πρέπει να χρησιμοποιήσουμε την εντολή **sense.show_message**. Το μήνυμα όμως, δεν μπορεί να περιλαμβάνει νούμερα, οπότε για να μετατρέψουμε τα νούμερα σε γράμματα, μπορούμε να χρησιμοποιήσουμε την εντολή **str**.

Χρησιμοποιώντας τις μεταβλητές **text_colour** και **back_colour**, μπορούμε να ορίσουμε το χρώμα των γραμμών και του παρασκήνιου, ορίζοντας πρώτα τρεις RGB τιμές, από το 0 έως το 255, οι οποίες ελέγχουν την φωτεινότητα του κάθε χρώματος. Για παράδειγμα, κόκκινο = (255,0,0)

Εκτός από το χρώμα, μπορούμε επίσης να θέσουμε και την ταχύτητα με την οποία εμφανίζεται το μήνυμα, χρησιμοποιώντας την μεταβλητή **scroll_speed**. Η προεπιλεγμένη ταχύτητα είναι 0.1. Όσο μικρότερος ο αριθμός, τόσο μεγαλύτερη και η ταχύτητα. Τέλος, αν χρειαστεί μπορούμε και να μεταβάλλουμε και τον προσανατολισμό του μηνύματος, με την εντολή **sense.set_orientation**.

Εισάγοντας την βιβλιοθήκη **time**, μας δίνεται η δυνατότητα να βάλουμε το πρόγραμμά μας σε κατάσταση αναμονής, για ένα καθορισμένο χρονικό διάστημα. Αυτό επιτυγχάνεται με την εντολή **sleep**. Ορίζουμε τον χρόνο που θέλουμε το πρόγραμμα να “κοιμηθεί” σε δευτερόλεπτα.

```
1 from sense_hat import SenseHat
2 from time import sleep
3 sense = SenseHat()
4 red = (255,0,0) #Ορίζουμε το χρώμα κόκκινο
5 year = int(input("What is the current year"))
6 year = str(year) #Μετατρέπουμε την χρονιά σε κείμενο
7 message = "The year is: " + year #Το μήνυμα που θα εμφανίσει στην οθόνη
8 while True: #Αυτή η διαδικασία επαναλαμβάνεται ανά 2 δευτερόλεπτα
9     sense.show_message(message, text_colour=red, scroll_speed=0.05)
10    sleep(2)
```

(Παράδειγμα προγράμματος που διαβάζει την χρονιά, και στην συνέχεια την εκτυπώνει ανά 2 δευτερόλεπτα)

ΠΕΡΙΣΤΡΟΦΗ ΜΥΝΗΜΑΤΟΣ:

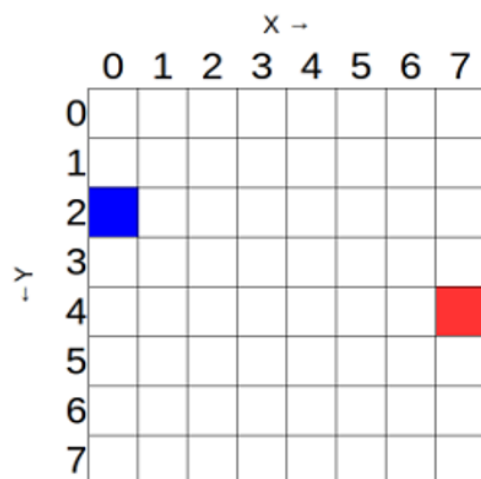
Αν χρειαστεί, μας δίνεται η ευκαιρία να περιστρέψουμε το κείμενο που εμφανίζεται στην οθόνη, κατά 90, 180, 270 και 360 μοίρες. Αυτό πραγματοποιείτε με την εντολή **set.orientation** ή απλούστερα, με την **flip_h()** μπορούμε να την περιστρέψουμε οριζόντια και αντίστοιχα καθέτα, με την **flip_v()**.

Όπως θα μάθουμε αργότερα και για το επιταχυνσιόμετρο, αν ένας άξονας έχει επιτάχυνση $\pm 1g$, τότε αυτός ο άξονας λέμε πως “κοιτάει προς τα κάτω”. Με αυτήν την πληροφορία, μπορούμε να δημιουργήσουμε έναν κώδικα, που αν περιστραφεί το Astro-Pi , θα περιστραφούν μαζί και τα μηνύματα που εμφανίζονται στην οθόνη. Συγκεκριμένα:

- Αν ο άξονας x έχει τιμή $-1g$, να περιστραφεί κατά 180 μοίρες
- Αν ο άξονας y έχει τιμή $+1g$, να περιστραφεί κατά 90 μοίρες.
- Αν ο άξονας y έχει τιμή $-1g$, να περιστραφεί κατά 270 μοίρες.

ΕΜΦΑΝΙΣΗ ΕΙΚΟΝΑΣ

Εκτός από μηνύματα και ενδείξεις , με την εντολή **set.pixel**, μπορούμε να ελέγξουμε ξεχωριστά το κάθε pixel της οθόνης του Sense Hat και να ορίσουμε το χρώμα που επιθυμούμε να εμφανίσει, δημιουργώντας έτσι “εικόνες” . Πρώτα όμως πρέπει να ορίσουμε τις x και y συντεταγμένες του επιθυμούμενου pixel. Το σύστημα συντεταγμένων του Sense Hat (το οποίο απεικονίζεται δεξιά), έχει ως αρχή των αξόνων το 0 και όχι το 1.



```
1 from sense_hat import SenseHat
2 sense = SenseHat()
3
4 red = (255,0,0)
5 blue = (0,0,255)
6
7 sense.set_pixel(0,2,red)
8 sense.set_pixel(7,4,blue)
```

(Παράδειγμα προγράμματος που φωτίζει με κόκκινο χρώμα το pixel με συντεταγμένες 0,2 και με μπλέ, αυτό με συντεταγμένες 7,4)

Παρόλα αυτά, αν θελίσουμε να εμφανίσουμε κάποιο σχήμα ή σχέδιο, θα ήταν αρκετά πιο εύκολο, αντί να χρησιμοποιούσαμε επανηλημένα την `set.pixel`, επιλέγαμε την ***set.pixels***, η οποία μας επιτρέπει να φωτίσουμε και τα 64 pixels απευθείας. Για την χρησιμοποιήσουμε όπως, πρέπει αρχικά να δημιουργήσουμε μια λίστα σε μορφή 8x8, που περιλαμβάνει μια μεταβλητή για το κάθε pixel, στην οποία ορίζουμε το χρώμα του. Για παράδειγμα:

```
1 from sense_hat import SenseHat
2
3 sense = SenseHat()
4
5 #Ορίζουμε τα χρώματα
6 B = (102, 51, 0)
7 b = (0, 0, 255)
8 S = (205,133,63)
9 W = (255, 255, 255)
10
11 #Με αυτήν την λίστα ορίζουμε που θα εμφανιστεί το κάθε χρώμα,
12 #δηλαδή τα pixels θα δείξουν ένα συγκεκριμένο χρώμα
13 image = [
14     B, B, B, B, B, B, B, B,
15     B, B, B, B, B, B, B, B,
16     B, S, S, S, S, S, S, B,
17     S, S, S, S, S, S, S, S,
18     S, W, b, S, S, b, W, S,
19     S, S, S, B, B, S, S, S,
20     S, S, B, S, S, B, S, S,
21     S, S, B, B, B, B, S, S
22 ]
23
24 #Εμφανίζουμε τα επιλεγμένα χρώματα, με βάση την λίστα
25 #που δημιουργήσαμε παραπάνω
26 sense.set_pixels(image)
```

(Παράδειγμα προγράμματος που εμφανίζει τον χαρακτήρα Steve, από το βιντεοπαιχνίδι της Mojang Studios, Minecraft)

ΑΝΙΧΝΕΥΟΝΤΑΣ ΤΟ ΠΕΡΙΒΑΛΛΟΝ

ΘΕΡΜΟΚΡΑΣΙΑ:

Το Sense Hat διαθέτει 2 αισθητήρες που μπορούν να μετρήσουν την θερμοκρασία του περιβάλλοντος, το υγρασιόμετρο καθώς και το πιεσόμετρο. Η εντολή `sense.get_temperature_from_humidity`, (ή αλλιώς απλούστερα `sense.get_temperature`), διαβάζει την θερμοκρασία από το υγρασιόμετρο, ενώ αντιστοίχη, η `sense.get_temperature_from_pressure`, την διαβάζει από πιεσόμετρο.

Παρόλ' αυτά, και οι δύο αισθητήρες, κατάγραφουν την ίδια θερμοκρασία, σε βαθμούς Κελσίου, οπότε δεν μας επηρεάζει ποια μέθοδο θα χρησιμοποιήσουμε. Στον Διεθνή Διαστημικό Σταθμό, η θερμοκρασία διατηρείτε από τους 18 μέχρι και τους 27 βαθμούς Κελσίου. Στον δεξιά πίνακα εμφανίζονται οι τιμές της θερμοκρασίας μέσα στο Columbus module, παρμένες από ένα Astro Pi, ενώ παρακάτω, ένα παράδειγμα προγράμματος που μετράει την θερμοκρασία, χρησιμοποιώντας το υγρασιόμετρο.

| Experimental data from Columbus module | |
|--|---------------|
| Temperature (°C) | Date and time |
| 27,53 | 16/2/16 10:45 |
| 27,52 | 16/2/16 10:45 |
| 27,54 | 16/2/16 10:45 |
| 27,55 | 16/2/16 10:45 |
| 27,53 | 16/2/16 10:45 |
| 27,55 | 16/2/16 10:45 |
| 27,54 | 16/2/16 10:46 |
| 27,54 | 16/2/16 10:46 |
| 27,53 | 16/2/16 10:46 |
| 27,52 | 16/2/16 10:46 |
| 27,53 | 16/2/16 10:46 |
| 27,53 | 16/2/16 10:46 |

```
1 from sense_hat import SenseHat
2
3 sense = SenseHat()
4 sense.clear
5
6 temp = sense.get_temperature()
7 print (temp)
```

ΥΓΡΑΣΙΑ ΚΑΙ ΠΙΕΣΗ:

Με το sense hat μας δίνετε επίσης η ευκαιρία να μετρήσουμε και την υγρασία και την πίεση, χωρίς την θερμοκρασία. Για να μετρήσουμε λοιπόν την υγρασία, σε ποσοστά τοις εκατό (%), θα χρησιμοποιήσουμε την εντολή **sense.get_humidity**. Παρόμοια, μπορούμε να υπολογίσουμε και την ατμοσφαιρική πίεση, σε millibars, με την εντολή **sense.get_pressure**. Στον ISS, η υγρασία διατηρείτε στο 60%, ενώ η πίεση από 980 μέχρι 1030 millibars.

Αξίζει να αναφερθεί ότι στο παρακάτω πρόγραμμα, στο οποίο μετριοούνται και εμφανίζονται οι τιμές της υγρασίας καθώς και της πίεσης, χρησιμοποιείτε η εντολή **round**, με την οποία μπορούμε να στρογγυλοποιήσουμε μια μεταβλητή, (στην συγκεκριμένη περίπτωση αυτές που έχουν αποθηκευτεί η πίεση και η υγρασία), με όσα δεκαδικά ψηφία θέλουμε, ακόμα και κανένα.

```
1 from sense_hat import SenseHat
2
3 sense = SenseHat()
4 sense.clear
5
6 pressure = sense.get_pressure()
7 humidity = sense.get_humidity()
8
9 pressure = round(pressure,2)
10 humidity = round(humidity,2)
11
12 print(pressure, humidity)
```

ΦΩΣ:

Το Sense Hat, περιέχει και έναν φωτοαισθητήρα, ο οποίος μπορεί να μετρήσει την ποσότητα φωτός που “χτυπάει” τον αισθητήρα, καθώς και ν’ ανιχνεύσει τις ποσότητες του μπλε, κόκκινου και πράσινου χρώματος. Οι μετρήσεις του πέρνουν τιμές από το 0 έως το 256.

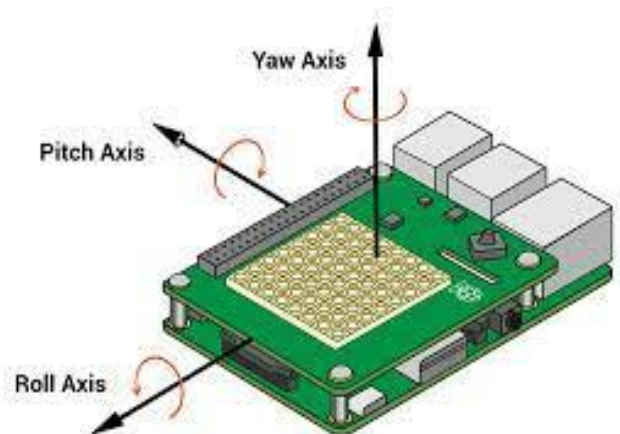
Με την μεταβλητή `sense.color.gain`, η οποία δέχεται ως τιμές μόνο το 1, το 4, το 16 ή και το 60, μπορούμε να ορίσουμε την ευαισθησία του ανιχνευτή. Παράλληλα, με την `sense.color.integration_cycles`, ορίζουμε τον χρόνο που περνάει σε milliseconds, μέχρι ο υπολογιστής να μετρήσει την ποσότητα του φωτός. Δέχεται τιμές από 1 ως και 256, ενώ η προεπιλεγμένη είναι 2,4 milliseconds. Παρακάτω εμφανίζεται ένα πρόγραμμα που εκτυπώνει τις τιμές του κόκκινου, μπλέ και πράσινου, καθώς και του ολικού φωτός που ανιχνεύονται.

```
1 from sense_hat import SenseHat
2 import time
3
4 sense = SenseHat()
5 sense.color.gain = 60
6 sense.color.integration_cycles = 64
7
8 while True:
9     sleep(2 * sense.colour.integration_time)
10    red, green, blue, clear = sense.colour.colour # readings scaled to 0-256
11    print(f"R: {red}, G: {green}, B: {blue}, C: {clear}")
```

ΑΔΡΑΝΕΙΑΚΗ ΜΟΝΑΔΑ ΜΕΤΡΗΣΗΣ (IMU)

Το Sense Hat, περιλαμβάνει μια αδρανειακή μονάδα μέτρησης, ή αλλιώς Inertial Measurement Unit (IMU), η οποία αποτελείτε από:

- Γυροσκόπιο, το οποίο μετράει ορμή και περιστροφή (μπορεί να βρει τον προσανατολισμό, δηλαδή που είναι το “πάνω”)
- Επιταχυνσιόμετρο, το οποίο μετράει επιτάχυνση και μπορεί να χρησιμοποιηθεί για να βρεί και την κατεύθυνση της βαρύτητας, όταν το αντικείμενο ηρεμεί



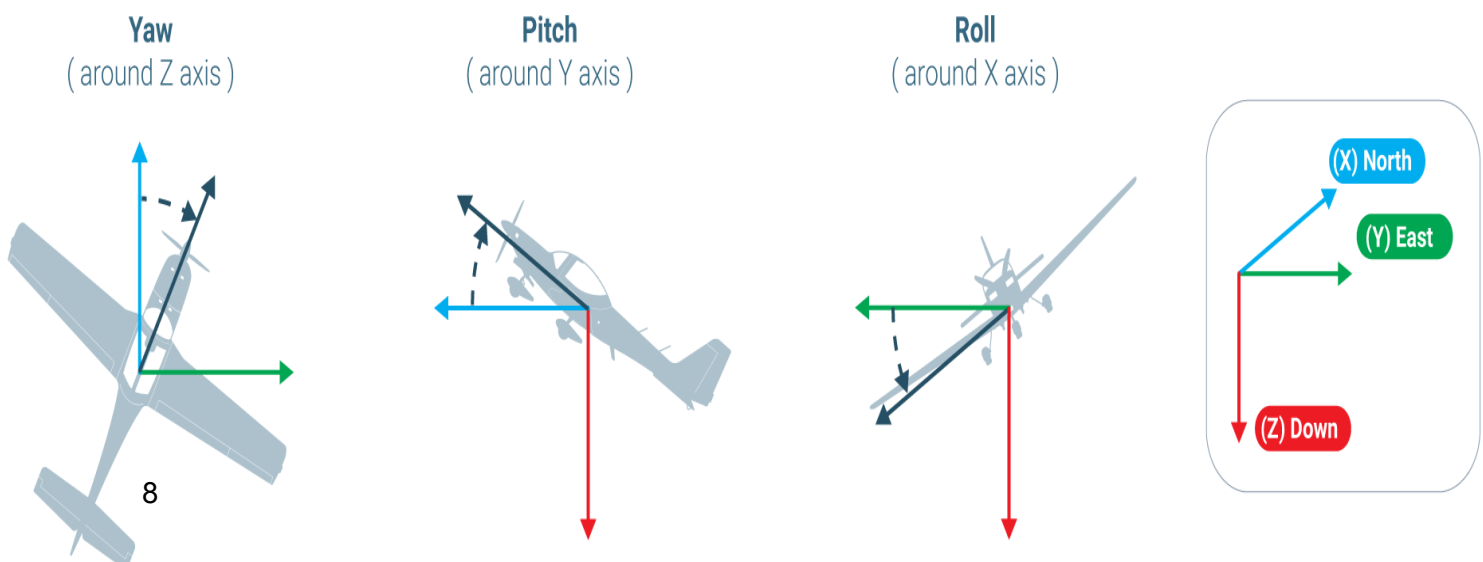
- Μαγνητόμετρο, που μετράει το μαγνητικό πεδίο της Γης, και λειτουργεί σαν μια πυξίδα

ΓΥΡΟΣΚΟΠΙΟ:

Το γυροσκόπιο υπολογίζει την ορμή του Sense Hat, καθώς και την περιστροφή του γύρω από τους 3 άξονες (x,y,z), όταν αυτό κινείται. Αυτή η περιστροφή, αναλόγως και στον άξονα όπου γίνεται, ονομάζεται Pitch, Roll και Yaw. Καλό θα ήταν να αναλύσουμε και να κατανοήσουμε καλά τι εκφράζουν αυτά τα τρία μεγέθη. Συγκεκριμένα:

- Το Pitch εκφράζει την περιστροφή γύρω από τον γ'γ άξονα (Για παράδειγμα ένα αεροπλάνο να απογειώνεται)
- Το Roll, εκφράζει την περιστροφή γύρω από τον x'x άξονα (Για παράδειγμα ένα αεροπλάνο να κάνει πλάγιες τούμπες)
- Το yaw, εκφράζει την περιστροφή γύρω από τον z'z άξονα (Για παράδειγμα να στρίβει σαν ένα αυτοκίνητο)

Παρακάτω εμφανίζεται μια εικόνα που απεικονίζει τα παραπάνω παραδείγματα, όπου το κόκκινο συμβολίζει τον άξονα z, το πράσινο τον y, και το μπλέ τον άξονα x. Σε αυτό το σημείο είναι καλό να υπενθυμίσουμε πως όλοι οι άξονες είναι κάθετοι μεταξύ τους.



Προκειμένου λοιπόν να καλέσουμε το γυροσκόπιο και να υπολογίσουμε ένα από αυτά τα μεγέθη, θα χρησιμοποιήσουμε την εντολή `get_gyroscope` ή `get_rotation`, όπου και με τις δύο εντολές θα μας επιστραφεί μία τιμή για τον κάθε άξονα. Παρακάτω εμφανίζεται ένα πρόγραμμα, το οποίο μετράει το Pitch, Roll και το Yaw.

```
1 from sense_hat import SenseHat
2
3 sense = SenseHat()
4 sense.clear()
5
6 o = sense.get_orientation()
7
8 #Εδώ ορίζουμε την μεταβλητή στην οποία αποθηκεύεται η τιμή για τον κάθε άξονα
9 pitch = o["pitch"]
10 roll = o["roll"]
11 yaw = o["yaw"]
12
13 #Εμφανίζουμε τις μετρήσεις
14 print("pitch {0} roll {1} yaw {2}".format(pitch, roll, yaw))
```

ΕΠΙΤΑΧΥΝΣΙΟΜΕΤΡΟ:

Το επιταχυνσιόμετρο μετράει το διάνυσμα της επιτάχυνσης , και συγκεκριμένα, το διάνυσμα της βαρυτικής επιτάχυνσης, όταν ένα αντικείμενο ηρεμεί. Υπολογίζει το μέτρο της σε g-forces , όπου $1g\text{-force} = 1g = 9.81 \text{ m/s}^2$, στην επιφάνεια της Γης. Μπορούμε να καλέσουμε το επιταχυνσιόμετρο με την εντολή `sense.get_accelerometer_raw`, η οποία όπως συμβαίνει και με το γυροσκόπιο, θα μας επιστρέψει 3 ξεχωριστές τιμές, μια για τον κάθε άξονα. Αξίζει να σημειωθεί πως αν σε έναν άξονα η επιτάχυνση είναι $\pm 1g$, τότε ο συγκεκριμένος άξονας βρίσκεται πάνω στη διεύθυνση της βαρύτητας.

Παρακάτω εμφανίζεται ένα πρόγραμμα το οποίο μετράει την επιτάχυνση στον κάθε άξονα, σε g-forces, και στην συνέχεια εμφανίζει τις μετρήσεις.

```

1 from sense_hat import SenseHat
2
3 sense = SenseHat()
4
5 while True:
6     acceleration = sense.get_accelerometer_raw()
7
8     #Ορίζουμε μια μεταβλήτη για την τιμή στον κάθε άξονα
9     x = acceleration['x']
10    y = acceleration['y']
11    z = acceleration['z']
12
13    #Στρογγυλοποιούμε τις μετρήσεις
14    x=round(x, 0)
15    y=round(y, 0)
16    z=round(z, 0)
17
18    #Τέλος, εμφανίζουμε τις στρογγυλοποιημένες μετρήσεις μας
19    print("x={0}, y={1}, z={2}".format(x, y, z))

```

Τέλος, στον παρακάτω πίνακα, εμφανίζονται οι μετρήσεις του επιταχυνσιομέτρου ενός Astro-Pi, μέσα στον Διεθνή Διαστημικό Σταθμό, σε g-foces.

| Table A2 - Accelerometer readings from the Astro Pi on the ISS (in g) | | | |
|---|----------|----------|-------------|
| accel_x | accel_y | accel_z | time_stamp |
| -0,00057 | 0,019359 | 0,014357 | 10:45:00 AM |
| -0,00044 | 0,019405 | 0,014425 | 11:45:00 AM |
| -0,00056 | 0,019531 | 0,014597 | 12:45:00 AM |
| -0,00056 | 0,019506 | 0,014432 | 01:45:00 PM |
| -0,00058 | 0,019464 | 0,014569 | 02:45:01 PM |
| -0,00056 | 0,01939 | 0,014578 | 03:45:00 PM |
| -0,00053 | 0,019384 | 0,014389 | 04:45:00 PM |
| -0,00046 | 0,01926 | 0,01444 | 05:45:00 PM |
| -0,00053 | 0,019266 | 0,014568 | 06:45:01 PM |

ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟ ΧΕΙΡΙΣΤΗΡΙΟ

Το Sense Hat, μπορεί να ανιχνεύσει τις κινήσεις του χειριστηρίου, όταν αυτό κινείται προς τα δεξιά, αριστερά, πάνω, κάτω ή ακόμα και όταν πιέζεται προς την μέση. Έχοντας ακριβώς την ίδια χρήση, αντί για το χειριστήριο, μπορεί να χρησιμοποιηθεί και το πληκτρολόγιο του υπολογιστή, όπου τα “arrow keys” (που παριστάνονται στην δεξιά εικόνα) αντιστοιχούν στις κινήσεις του χειριστηρίου και το κουμπί “Enter”, στο πάτημά του.



Στο παρακάτω παράδειγμα, το οποίο μπορεί να ανιχνεύσει πότε ένα από αυτά τα κουμπιά πατιούνται, το `event.direction` αντιστοιχεί στην κατεύθυνση στην οποία θα πιεζόταν αντίστοιχα το joystick, πχ πάνω, δεξιά κτλ, ενώ το `event.action`, στην δράση που πραγματοποιήθηκε, δηλαδή αν το χειριστήριο ή τα κουμπιά πιάστηκαν, κρατήθηκαν παρατεταμένα ή απελευθερώθηκαν.

```
1 from sense_hat import SenseHat
2 sense = SenseHat()
3
4
5 while True:
6     for event in sense.stick.get_events():
7         print(event.direction, event.action)
```

ΣΥΝΟΠΤΙΚΗ ΛΙΣΤΑ ΕΝΤΟΛΩΝ

- str : Μετατρέπει αριθμούς σε κείμενο
- abs : Υπολογίζει την απόλυτη τιμή μίας μεταβλητής
- round : Στρογγυλοποιεί την τιμή μιας μεταβλητής
- show_message : Εμφανίζει ένα μήνυμα στην οθόνη του Sense Hat
- set_rotation : Περιστρέφει ό,τι εμφανίζεται στην οθόνη του SH*
- sleep : Τοποθετεί το πρόγραμμα σε κατάσταση αναμονής
- get_temperature : Μετράει την θερμοκρασία
- get_pressure : Μετράει την πίεση
- get_humidity : Μετράει υγρασία
- get_orientation : Μετράει το Pitch, Roll, Yaw
- get_accelerometer_raw : Μετράει το διάνυσμα της επιτάχυνσης
- set_pixel : Ορίζει το χρώμα ενός pixel της οθόνης του SH*

*SH = Sense Hat

Σε αυτό το σημείο, ολοκληρώνεται αυτός ο οδηγός. Ελπίζουμε να βοήθησε τους αναγνώστες να κατανοήσουν και να κατάλαβουν πως λειτουργούν οι αισθητήρες του Sense Hat, καθώς και πως οι ίδιοι μπορούν να τους χρησιμοποιήσουν για τα δικά τους πειράματα. Ευχόμαστε σε όποιον-α ή οποιαδήποτε ομάδα διαβάσει αυτόν τον οδηγό, καλή επιτυχία.

Ο οδηγός αυτός γράφτηκε από τον Στέλιο Στεφανή, με την βοήθεια της υπόλοιπης ομάδας GR.ISS, Παναγιώτη Σκούλη, Παναγιώτη Καλαμακίδη, Αλέξανδρο Καραδίμογλου, υπό την επίβλεψη του Παναγιώτη Πετρίδη και Χρυσόστου Σαμαρά.

23ο ΓΕΛ Θεσσαλονίκης, 2023