Reading of sensor information using Sense HAT

Student: Yi Qiang Ji Zhang
Professor: Dr. Jaume Figueras Jove
Aerospace Engineering
Polythecnical University of Catalonia

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

30 April 2021

1 Session 1

This session intended to be an introductory session with Python programming. Various exercises are shown including how to work with dictionaries and checking if a word is a palindrome or not.

1.1 Class session

The followinf three codes

```python
# This code prompts the user for a number and tell if the number is even or not

for i in range(0,10):
    number = input("Enter a number: ")
    new_number = int(number)

    if new_number % 2 == 0:
        print("The number is even")
    else:
        print("The number is odd")
```

Listing 1: Is an integer even or not.

```python
# Working with lists

# Lists can contain both numbers and strings
a = list()
a.append(12)
print(a)

# Append a number in the last position of the vector
a.append(34)
print(a)
```

```python

# Append a string
a.append("hello")
print(a)

# The first element of the list
print(a[0])

# The last element of the list
print(a[-1])

# The first elements
print(a[0:2])

# Adding more elements
a.append("qwerty")
a.append(56)
a.append(578)

# Show the last two elements with reverse order
print(a)
print(a[-1:3:-1])

# The penultimate element of the list
print(a[-2])
```

Listing 2: Working with lists.

```python
# Dictionaries

# Create a dictionary
b = dict()
b['key1'] = 23
print(b)

# Create more instances
b['temp'] = 23
b['hum'] = 95
b['press'] = 1000

print(b)

# Change one instance to a string
b['press'] =  'data string'
print(b)

# Delete element
del b['key1']
print(b)
```

Listing 3: Dictionaries

1.2 Exercises

```python
# Ask for your name and surname and save it in different variables
```

```python

name = input("Enter your name: ")
surname = input("Enter your surname: ")

print("Hi " + name + " " + surname + " and welcome to ICT !")
```

Listing 4: Save name and surname into variables.

```python
# This code prompts the user for a number and tell if the number is even or not

for i in range(0,10):
    number = input("Enter a number: ")
    new_number = int(number)

    if new_number % 2 == 0:
        print("The number is even")
    else:
        print("The number is odd")
```

Listing 5: Is even or not.

```python
# Write a program that shows the divisors of an introduced number

# Prompt user to introduce an integer
number = int(input("Introduce an integer: "))

# Create an empty list
divisors = []

# Calculate divisors
for i in range(1,number):
    div = number % i
    if div == 0:
        divisors.append(i)

# Print
print("Your number was " + str(number) + " " + "and its divisors are: ")
# Print divisors
print(divisors)
```

Listing 6: Divisor of a number

```python
# Check if a word is a palindrome or not

# Function
def isPalindrome(str):
    # Run loop from 0 to len/2
    for i in range(0, int(len(word)/2)):
        if word[i] != word[len(word)-i-1]:
            return False
    return True

# Prompt user to introduce a word
word = str(input("Enter a word: "))

```

```
if isPalindrome(word)==True :
    print("The word " + word + " is a palindrome")
else:
    print("The word " + word + " is NOT a palindrome")
```

Listing 7: Is a word palindrome.

2 Session 2 and Session 3

The second session focuses on gathering data using *senseHat* from Raspberry Pi. The class session shows how to begin using the sense Hat API, for gathering data and how to format it to be shown in the terminal as well as managing the time calculation between each read. The follow-up exercise uses all the concepts explained in class and and writes the data into a csv file.

2.1 Class session

```
# Sense data from the SenseHat

# More documentation: https://projects.raspberrypi.org/en/projects/getting-started-with-the-sense...
    -hat

# Import the SenseHat object from the emulator
from sense_emu import SenseHat
import time # Time library

# Create an object
sense = SenseHat()

# Infinite loop
while True:
    # Read temp from temperature sensor
    temp = sense.get_temperature()
    # Read temp from pressure sensor
    temp_p = sense.get_temperature_from_pressure()
    # Read temp from humidity sensor
    temp_h = sense.get_temperature_from_humidity()

    # Read pressure from pressure sensor
    pressure = sense.get_pressure()
    # Read humidity from humidity sensor
    humidity = sense.get_humidity()

    # Print temp data
    print("Temperature from pressure sensor: {0:3f} °C; from humidity is {1:3f} °C".format(temp_p...
    ,temp_h))
    # Print pressure and humidity data
    print("Pressure is: {0:0f} mbar".format(pressure))
    print("Humidity is: {0:0f} %".format(humidity))
    time.sleep(1) # Read data every second
```

Listing 8: Sense data from temperature pressure and humidity sensors.

```
# Read IMU data from the SenseHat

# More info: https://pythonhosted.org/sense-hat/api/

# Import the SenseHat object from the emulator
from sense_emu import SenseHat
import time # Time library

# Create an object
sense = SenseHat()
sense.clear()

# Loop infinitely
while True:
    # Get IMU data
    o = sense.get_orientation() # 'o' object is a dictionary
    pitch = o["pitch"]
    roll = o["roll"]
    yaw = o["yaw"]

    # Print the whole dictionary
    print(o)
    print(pitch)
    print(roll)
    print(yaw)
    print("pitch {0} roll {1} yaw {2}".format(pitch, roll, yaw))

    # Get accelerometer raw data
    accel = sense.get_accelerometer_raw()
    print(accel)
    accel2 = sense.get_accelerometer()
    print(accel2)

    # Get gyroscope raw data
    gyro = sense.get_gyroscope_raw()
    print(accel)

    # Get compass raw data
    compass = sense.get_compass_raw()
    print(compass)
```

Listing 9: Sense data from IMU.

```
# Sense data from the SenseHat

# Import the SenseHat object from the emulator
from sense_emu import SenseHat
import time # Time library
import datetime

# Create an object
sense = SenseHat()

# Infinite loop
while True:
    # Save current beginning time
    start = time.time()

    # Read temp from temperature sensor
    temp = sense.get_temperature()
    # Read temp from pressure sensor
    temp_p = sense.get_temperature_from_pressure()
    # Read temp from humidity sensor
    temp_h = sense.get_temperature_from_humidity()

    # Read pressure from pressure sensor
    pressure = sense.get_pressure()
    # Read humidity from humidity sensor
    humidity = sense.get_humidity()

    # Print temp data
    print("Temperature from pressure sensor: {0:3f} °C; from humidity is {1:3f} °C".format(temp_p,temp_h))
    # Print pressure and humidity data
    print("Pressure is: {0:0f} mbar".format(pressure))
    print("Humidity is: {0:0f} %".format(humidity))

    # Let the sensors have enough data by measuring multiple times
    for i in range(0,10):
        o = sense.get_orientation() # 'o' object is a dictionary
    print(o)

    # Read an write time
    current_time = datetime.datetime.now()
    print(current_time)

    # Save current end time
    end = time.time()
    elapsed_time = start - end
    time.sleep(1 - elapsed_time) # Sleep for 1 second taking into account the elapsed time
```

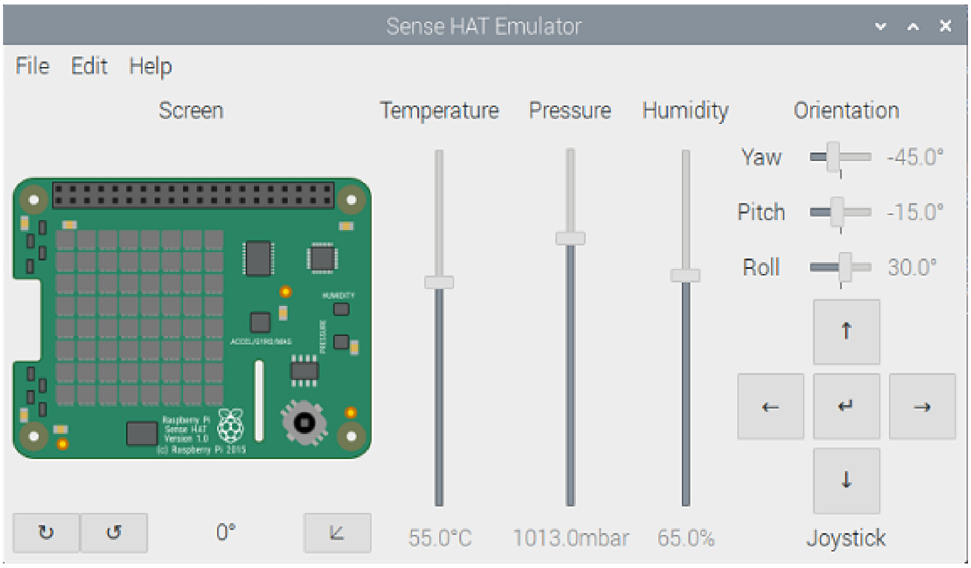
Listing 10: Adding time library.

2.2 Exercises

```
# Sense data from the SenseHat

# Import the SenseHat object from the emulator
from sense_emu import SenseHat
import time # Time library
import datetime # Date time
import csv # Import csv

# Create an object
sense = SenseHat()
sense.clear()

# Field Names
fieldnames = ["Current UTC Time","Temp (°C) [Temp sensor]", "Temp (°C) [Pressure sensor]", "Temp (°C) [Humidity sensor]", "Pressure (mbar) [Pressure sensor]", "Humidity (mbar) [Humidity sensor]", "Pitch (°) [IMU]", "Roll (°) [IMU]", "Yaw (°) [IMU]", "Pitch (°) [Accel sensor]", "Roll (°) [Accel sensor]", "Yaw (°) [Accel sensor]", "Pitch (°) [Gyro sensor]", "Roll (°) [Gyro sensor]", "Yaw (°) [Gyro sensor]", "North (%) [Compass sensor]"]

# Create the csv file:
with open('database_2.csv', 'w', newline='') as file:
    # Write csv
    writer = csv.writer(file)
    writer.writerow(fieldnames)

    # Infinite loop
    while True:
        # Save current beginning time
        start = time.time()

        # Read temp from temperature sensor
        temp = sense.get_temperature()
        # Read temp from pressure sensor
        temp_p = sense.get_temperature_from_pressure()
        # Read temp from humidity sensor
        temp_h = sense.get_temperature_from_humidity()
        # Read pressure from pressure sensor
        pressure = sense.get_pressure()
        # Read humidity from humidity sensor
        humidity = sense.get_humidity()

        # Let the sensors have enough data by measuring multiple times
        for i in range(0,10):
            o = sense.get_orientation() # 'o' object is a dictionary
        pitch_IMU = o["pitch"]
        roll_IMU = o["roll"]
        yaw_IMU = o["yaw"]

        for i in range(0,10):
            accel_only = sense.get_accelerometer()
        pitch_acc = accel_only["pitch"]
        roll_acc = accel_only["roll"]
        yaw_acc = accel_only["yaw"]

        for i in range(0,10):
            gyro_only = sense.get_accelerometer()
        pitch_gyro = gyro_only["pitch"]
        roll_gyro = gyro_only["roll"]
        yaw_gyro = gyro_only["yaw"]

        for i in range(0,10):
            north = sense.get_compass()

        # Read an write time
        current_time = datetime.datetime.utcnow()

        # writer.writerow(['1'])
        writer.writerow([current_time,temp,temp_p,temp_h,pressure,humidity,pitch_IMU,roll_IMU,...
    yaw_IMU,pitch_acc,roll_acc,yaw_acc,pitch_gyro,roll_gyro,yaw_gyro,north])

        # Save current end time
        end = time.time()
        elapsed_time = start - end

        # Sample sample_frequency
        sample_frequency = 1/elapsed_time

        time.sleep(1 - elapsed_time) # Sleep for 1 second taking into account the elapsed time
```

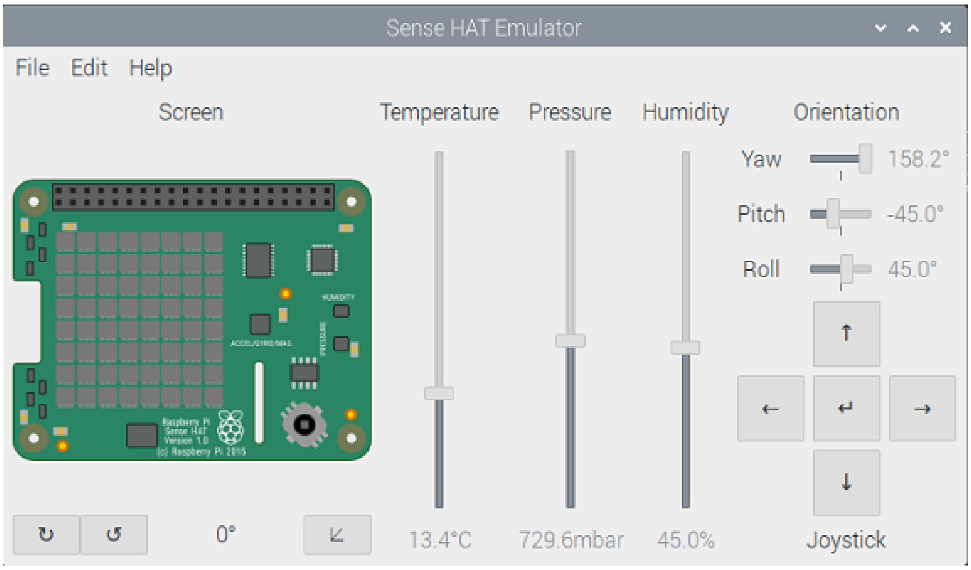
Listing 11: Write data into a csv file.

Below is presented how the SenseHAT Emulator was configured (see Figures 1 and 2):



Yaw -45.0°
Pitch -15.0°
Roll 30.0°
55.0°C
1013.0mbar
65.0%

Figure 1: Caption



Yaw 158.2°
Pitch -45.0°
Roll 45.0°
13.4°C
729.6mbar
45.0%

Figure 2: Caption

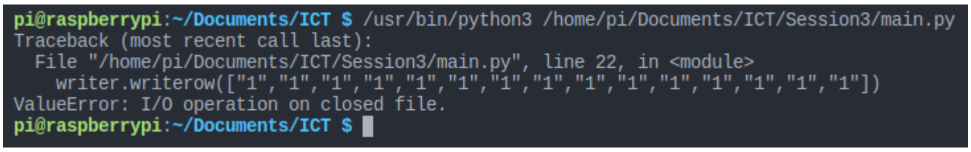
The output csv looks like (see Table 1):

Current UTC Time	Temp (ºC) [Temp sensor]	Temp (ºC) [Pressure sensor]	Temp (ºC) [Humidity sensor]	Pressure (mbar) [Pressure sensor]	Humidity (mbar) [Humidity sensor]	Pitch (º) [IMU]	Roll (º) [IMU]	Yaw (º) [IMU]	Pitch (º) [Accel sensor]	Roll (º) [Accel sensor]	Yaw (º) [Accel sensor]	Pitch (º) [Gyro sensor]	Roll (º) [Gyro sensor]	Yaw (º) [Gyro sensor]	North (%) [Compass sensor
2021-05-01 07:23:45.859898	54.984375	55.0958333333333	54.984375	1012.99755859375	64.77734375	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:23:48.240021	54.984375	54.9125	54.984375	1012.99072265625	64.9140625	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:23:50.619010	55.015625	54.9104166666667	55.015625	1012.99829101563	64.921875	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:23:52.983212	55.015625	55.15	55.015625	1012.9970703125	65.29296875	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:23:55.345296	54.921875	55.0875	54.921875	1013.00537109375	65	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:23:57.692786	54.953125	55.06875	54.953125	1012.99072265625	64.68359375	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:24:00.032648	55.015625	54.9520833333333	55.015625	1012.99584960938	64.8984375	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:24:02.413349	42.5625	14.2270833333333	42.5625	1012.99829101563	65.21875	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:24:04.797513	17.0625	13.5125	17.0625	729.661865234375	62.546875	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	345.000786117359	29.9984277652822	315.002358352077	315.002358352077
2021-05-01 07:24:07.157184	13.40625	13.39375	13.40625	729.630126953125	44.7265625	345.000786117359	29.9984277652822	158.189511395239	345.000786117359	29.9984277652822	158.189511395239	345.000786117359	29.9984277652822	158.189511395239	158.189511395239
2021-05-01 07:24:09.512372	13.390625	13.3291666666667	13.390625	729.619140625	45.2109375	0	29.9984277652822	158.189511395239	0	29.9984277652822	158.189511395239	0	29.9984277652822	158.189511395239	158.189511395239
2021-05-01 07:24:11.854421	13.40625	13.4583333333333	13.40625	729.616943359375	44.546875	0	0	158.189511395239	345.000786117359	0	158.189511395239	330.001572234718	0	158.189511395239	158.189511395239
2021-05-01 07:24:14.195830	13.359375	13.325	13.359375	729.562255859375	45.04296875	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239
2021-05-01 07:24:16.553766	13.359375	13.4458333333333	13.359375	729.520751953125	45.109375	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239
2021-05-01 07:24:18.904891	13.40625	13.4604166666667	13.40625	729.588134765625	44.9296875	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239
2021-05-01 07:24:21.254367	13.421875	13.4625	13.421875	729.666015625	44.83203125	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239
2021-05-01 07:24:23.620523	13.40625	13.4875	13.40625	729.535400390625	45.16796875	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239
2021-05-01 07:24:25.980654	13.40625	13.3666666666667	13.40625	729.626708984375	44.70703125	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239
2021-05-01 07:24:28.341278	13.421875	13.4354166666667	13.421875	729.581787109375	45.25390625	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	315.002358352077	44.9976416479234	158.189511395239	158.189511395239

Table 1: CSV database from Sense Hat sensors. Source: Own.

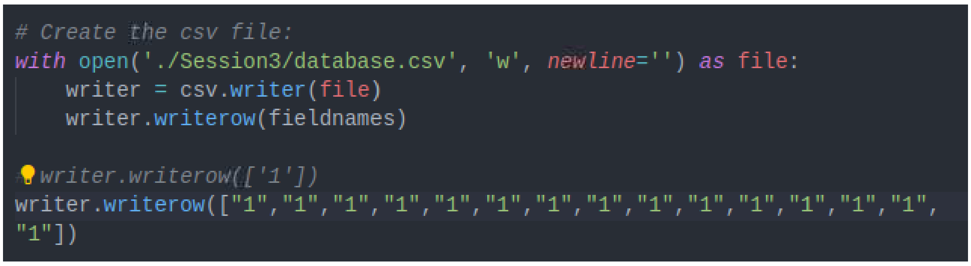
The main problem faced in exercise 3 was to create and write a csv file. The intention was to create a file within the current working directory. However, the code created the csv file outside the current directory to the main directory.

Another issue encountered is the fact that the csv file must be opened to be able to write information to it. Since it was not open, the following error comes:



```
pi@raspberrypi:~/Documents/ICT $ /usr/bin/python3 /home/pi/Documents/ICT/Session3/main.py
Traceback (most recent call last):
  File "/home/pi/Documents/ICT/Session3/main.py", line 22, in <module>
    writer.writerow(["1","1","1","1","1","1","1","1","1","1","1","1","1","1","1"])
ValueError: I/O operation on closed file.
pi@raspberrypi:~/Documents/ICT $ 
```

Figure 3: Terminal error. Source: Own.



```
# Create the csv file:
with open('./Session3/database.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(fieldnames)

#writer.writerow(['1'])
writer.writerow(["1","1","1","1","1","1","1","1","1","1","1","1","1","1",
"1"])
```

Figure 4: Error in code

The main problem was that

```
writer.writerow()
```

was not inside the action

```
with open() as file:
```

References

[1] Python Hosted. *Sense HAT API Reference*. 2021. URL: https://pythonhosted.org/sense-hat/api/#imu-sensor.