Report on:

Data logger for pressure

Project: Energy Supply Structures for the future

Guided By:

Dipl.-Ing. Stefan Balluff

Under the supervision of:

Dr.-Ing. habil. Stefan Krauter.

Submitted by:

Project By:

Ebeny wesley Tchapda Kouayep(6728360)

Sudhanraj ravikumar(6779070)

Tahir ahmad(6756398)

Sabarish Kumar Amaravadi(6778860)

CONTENTS

CHAPTER 1

Raspberry Pi

1.1 INTRODUCTION

The Raspberry Pi is a low cost, Bankcard or a visitor card sized tiny computer designed and developed by Raspberry Foundation in the United Kingdom. The Raspberry Pi consists of two USB ports, an HMDI port, an Ethernet port, a special Camera module, a port for a micro SD card and power supply port. It can be attached to a Monitor or TV and uses a standard USB keyboard and mouse. This device allows people of all ages to explore the computing, to learn programming languages like Scratch and Python, Sonic Pi to make code for creating music, to teach school students the basic computer science and every individual to learn computer hardware, programming and projects. It is capable of doing anything, which you can expect from a computer to do, for instance, browsing the internet, Playing HD videos, playing games etc.

Raspberry Pi has three versions:

Raspberry Pi 1, Raspberry Pi 2 and the Latest is Raspberry Pi3. Each raspberry pi has different subversions. The latest Raspberry Pi's hardware consists of:

- 1.2 GHz 64-bit Quad Core ARMv8 Processor
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy

The remaining components are the same as in the previous version of Raspberry Pi. The Device does not contain any built-in hard disk and have an external SD card port. The recommended SD card memory size is 8GB.

The Raspberry Pi is available for online purchase anywhere in the world and it costs around 35$.

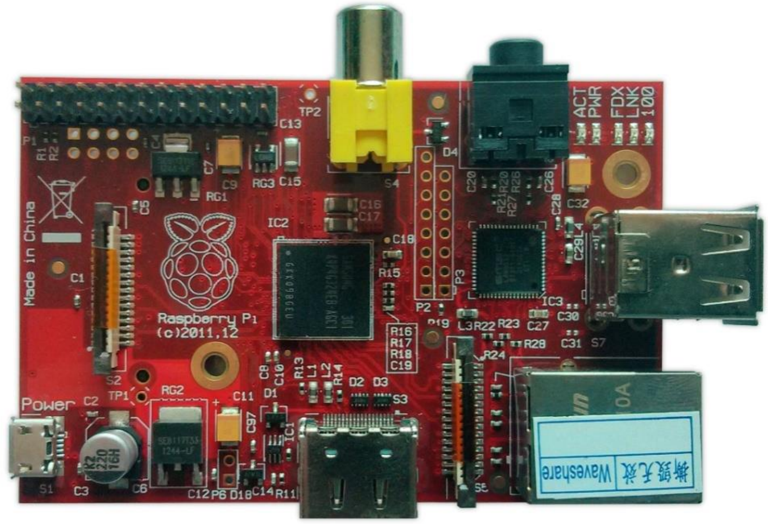

Fig. 1.1.1: Raspberry Pi (Top view)

1.2 HISTORY OF COMPUTER:

Computers in 1950's were very huge and very expensive. The applications of those days' computers were not much as we have it nowadays. The computers were used to get weather and other complex calculations. The huge computers did not have any operating systems, needed dedicated air-conditioned rooms, and trained persons to operate them. The inventors of Integrated Circuits made a huge difference in the performance and size of the huge machine to make them smaller.

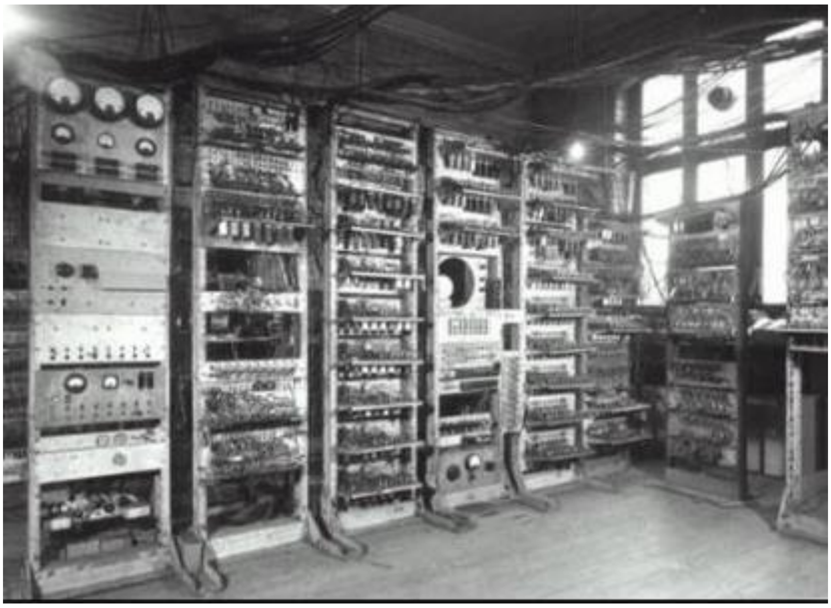

Fig. 1.2.1: Computer (in early 1950's)

In the early stages of computer, the portable computers were like a dream and were not available easily. Most of them have dedicated applications to work on or just single purpose usages. For example, for printing or for calculating.

A new innovation came in the history of computers when Apple had released, LISA in 1983, Graphical User Interface (GUI) was introduced for the first time in the history of computer.

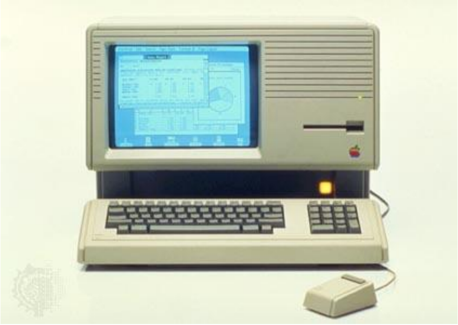

Fig. 1.2.2: Apple's LISA

As the GUI made by Apple, Microsoft also released DOS, which was a great achievement for IBM PC. The technology became cheap and affordable, now people could afford computers for their homes. IBM started creating massive amount of personal computers, enabling people to play games and do many other things.

At the same time, people have started considering creating of the portable computers. Many companies have launched the portable computers, which were the foundation for the modern Laptops. Osborne 1 was the first massively made portable computer released in 1981.

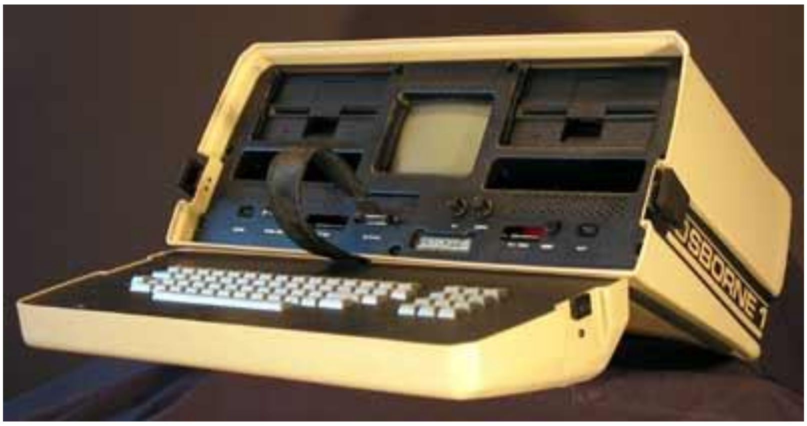

Fig. 1.2.3: Portable computer

As the technology improves during the 1990’s, the popularity and demand of the laptops increased. Several developments to laptops as if improved battery technology, power saving processors, improved LCD, Increased storage is implemented, improving usability and performance.

1.3 IDEA AND THE MAKING OF RASPBERRY PI:

The thought behind a tiny and affordable computer for kids came in 2006, when Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft, based at the University of Cambridge's Computer Laboratory, became concerned about the decline every year in the numbers and skill levels of the A Level students applying to read Computer Science. From a situation in the 1990s, where most of the kids applying were coming to interview programmers, the landscape in the 2000s was very different; a typical applicant might only have done a little web design.

The interest in interacting with the computer has changed, a number of problems were identified, majority of curriculum with lessons using Microsoft Word or Excel or making web pages. The students have decided to find a platform where they can easily boot in the programming environment. Thus the idea of creating such device which allows kids to perform specific tasks like programming or learn hardware, came and it was named as Raspberry PI.

The Raspberry Pi foundation created many designs and prototypes in the initial stages, the following is one of the examples.

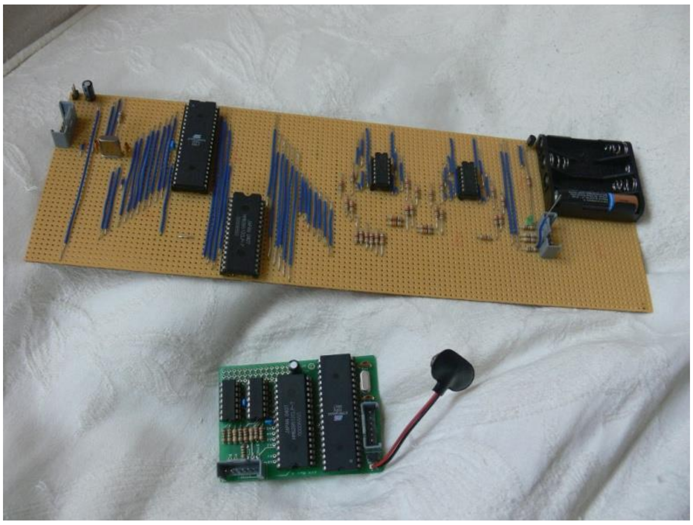

Fig. 1.3.1: Prototype of Raspberry Pi

1.4. HARDWARE AND LAYOUT OF RASPBERRY PI

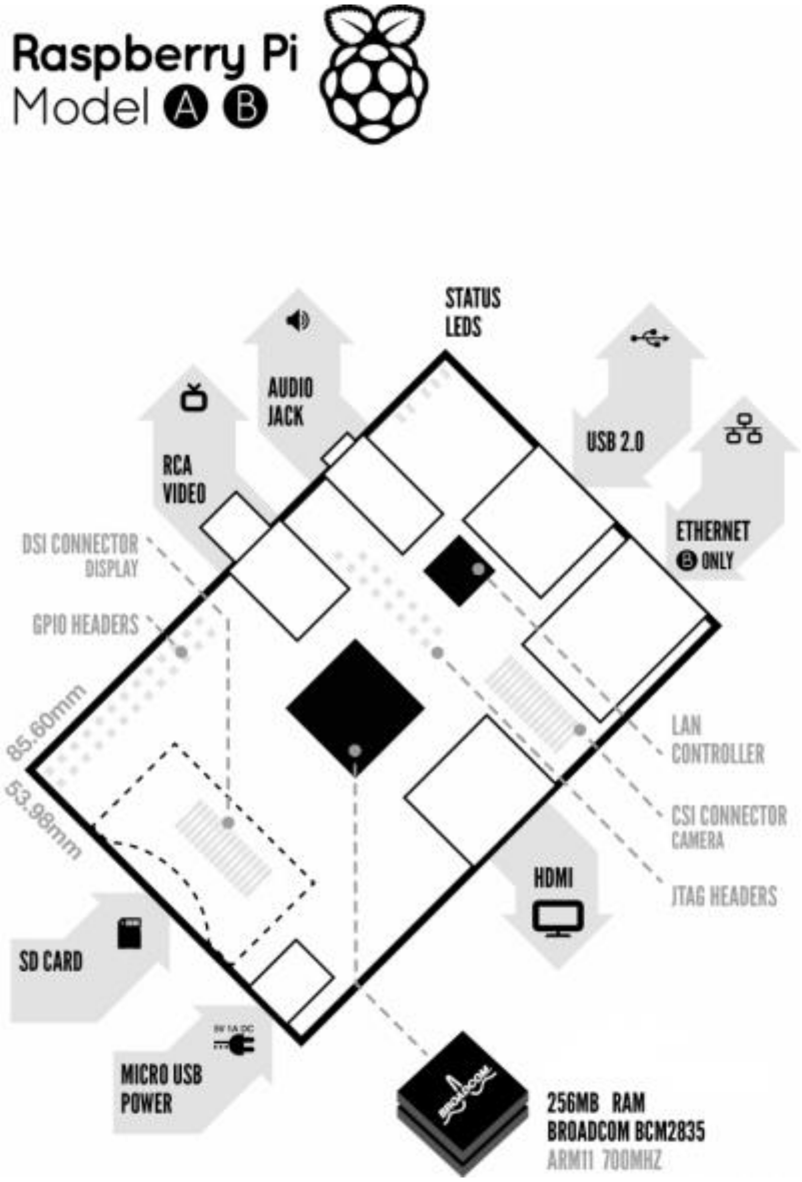

Fig. 1.4.1: Layout of Raspberry Pi

Components of Raspberry Pi:

1. Processor:
   The Raspberry Pi has a system on chip module, It has an ARM1176JZF-S processor. The system on chip is a complex that integrates the major functional elements into a single chip. The Raspberry Pi processor operating at 700 MHz, will not become hot and doesn’t require any cooling component.

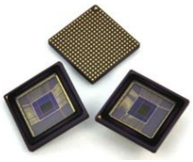

Fig. 4.2: ARM1176JZF-S processor

2. General Purpose Input Output. (GPIO)
   One powerful function of Raspberry Pi is GPIO pins along the edge of it, these pins are physical interface between Pi and the outside world. GPIO pins can be configured to be input or output and can be enabled and disabled.

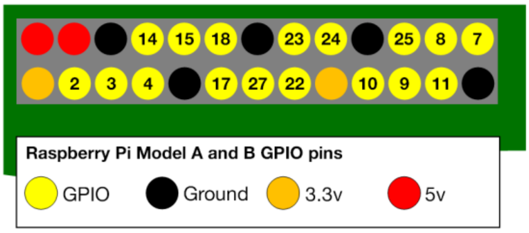

Fig. 4.3: GPIO

3. Power Supply.
   The Raspberry Pi consumes 700mA or 3W. The power is supplied through micro USB charger or GPIO header.

4. DSI Connector
   The Display Serial Interface defines a communication protocol between the host (source of the image data) and the device (destination of image data).

5. SD Card.
   The Pi does not have any built in storage. The operating System is loaded on a SD card, which is inserted into the SD card slot of Raspberry PI.

6. RCA Video
   RCA Video Outputs (PAL and NTSC) are available on all models of Pi. Any display with a RCA port can be connected easily.

7. Audio Jack.
   A standard 3.5mm audio jack is available in the Raspberry PI for stereo output.

8. USB Port.
   USB 2.0 ports are the means of connecting devices such as mouse, keyboard, Mic to the Raspberry Pi.

9. Ethernet Port
   Ethernet Port is available on the raspberry Pi. It can be connected to the internet using standard Ethernet cable.

10. CSI Port
    Camera Serial Interface is designed to be interfacing digital cameras.
11. HDMI
    High Definition Multimedia Interface designed to provide HD display to HDMI screens.

1.5. OPERATING SYSTEM

Operating System, The Raspberry Pi primarily uses Linux kernel-based operating systems. The ARM11 is based on version 6 of the ARM, which is no longer held by several popular versions of Linux, including Ubuntu. The install manager for Raspberry Pi is NOOBS. The OSs included with NOOBS are: Archlinux ARM, Open ELEC,Pidora (Fedora Remix), Raspbmc and the XBMC open source digital media centre, RISC OS – The operating system of the first ARM-based computer, Raspbian (recommended) – Maintained independently of the Foundation.

Noobs Installer

To get started with a Raspberry Pi, you need an operating system. Raspberry Pi itself does not contain any built in Operating system therefore it has to be install via a SD card, copying the operating system setup from computer.

1. Buy a Pre-Installed SD card.
   Pre-Installed Noobs SD cards can easily be found from Raspberry Pi vendors or private retailers. Examples of such operating systems are, Pimoroni, Adafruit and PiHut.
2. Download from Raspberry Pi website.
   - Raspberry Pi foundation recommends a minimum of 8GB SD card.
   - Using a computer with SD card reader, Download the operating system setup from following link: https://www.raspberrypi.org/downloads/
   - Download the file and extract it in any folder
   - Format the SD card
   - Follow the instructions of Installation process.
3. First Boot
   - Plug in The keyboard, mouse, monitor and the USB power cables.
   - A window will appear with the list of different Operating Systems, the recommended O.S is Raspbian, tick the box next to Raspbian and press Install.
   - When the setup is finished, the raspberry pi configuration menu will load. Here you can set up the time and date and other basic parameters.

1.6. APPLICATIONS:

The basic concept of creating raspberry pi was to educate people, particularly minors and teenagers about the programming and basic hardware interface.

Some of the applications of Raspberry Pi are as Follows:

- Teaching the programming languages
- Teaching the hardware interface
- It can be used in Robotics for controlling motors, sensors etc.
- It can measure temperature, pressure, humidity, acceleration using the sense hat.
- It is used in creating and using small servers.

Advantages of Raspberry Pi

- The raspberry Pi is although a credit card sized small machine but it functions same as a normal computer at relatively very low cost.
- The device is energy efficient and does not consume huge amount of energy unlike big servers that requires a lot of energy and cooling systems
- The pi is portable and can be held in your hand.
- The Pi is completely silent making no noises at all.
- The device has a built in HDMI port that enables the resolution of 1920*1200.

Disadvantages of Raspberry Pi

- The processor is not as fast as a normal computer.
- The device is not compatible with other operating systems such as Windows.
- This device is not efficient for bigger business, which requires big servers.
- The RAM is not upgradable.

Conclusion

The Raspberry Pi is surely an innovative product that can help anyone who wants to learn electronics and computer programming and hardware. Increasing the processing power can help its working capabilities. Supplying the case and user manual will improve the user’s feedback about the product. Operating systems such as Windows can also boom the device.

It can be used effectively if its processing power is increased. It can work as a personal computer but it cannot replace it.

References

https://www.raspberrypi.org/help/what-is-a-raspberry-pi/

https://www.raspberrypi.org/help/quick-start-guide/

http://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading03.htm

https://www.raspberrypi.org/about/

http://www.slideshare.net/nipunmaster/a-seminar-report-on-raspberry-pi

http://elinux.org/RPi_Hardware

https://www.raspberrypi.org/documentation/usage/gpio/

https://www.raspberrypi.org/downloads/

https://www.raspberrypi.org/documentation/installation/installing-images/

https://www.raspberrypi.org/help/noobs-setup/

http://www.zdnet.com/article/raspberry-pi-11-reasons-why-its-the-perfect-small-server/

https://sites.google.com/site/bus237raspberrypi/customize

CHAPTER 2

SENSE HAT

2.1 INTRODUCTION

The Raspberry Pi Sense Hat (Figure 2.1) is the housing of multiple sensors mounted onto the top of Raspberry Pi.

The integrated circuit based sensors include:

1. Accelerometer

2. Gyroscope

3. Magnetometer

4. Temperature sensor

5. Humidity Sensor

6. Pressure sensor

In addition, Sense Hat also consists of an 8*8 LED matrix and a small 5 button joystick.

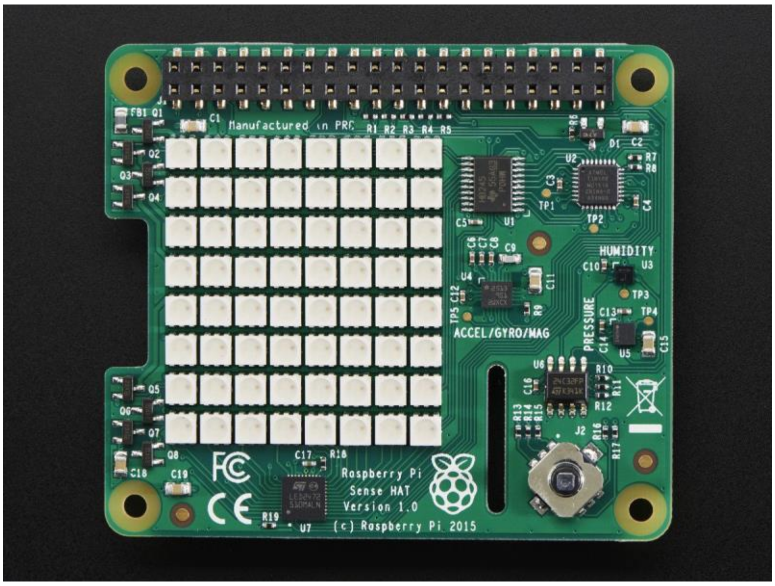

Figure 2.1.1: Raspberry Pi Sense Hat

(Source: adafruit.com)

TECHNICAL DETAILS:

- Gyroscope
    - Angular rate sensor (dps): ~245/500/2000
- Accelerometer
    - Linear acceleration sensor (g): ~2/4/8/16
- Magnetometer
    - Magnetic sensor (gauss): ~4/8/12/16
- Barometer: 260 - 1260 hPa absolute range (accuracy depends on the temperature and pressure, ~0.1 hPa under normal conditions)
- Temperature sensor: Accurate to ~2°C in the 0-65°C range
- Relative humidity sensor: Accurate to ~4.5% in the 20-80%rH range, accurate to ~0.5°C in 15-40°C range
- 8x8 LED matrix display
- Small 5 button joystick

A Raspeery pi equipped with sense

2.2 INSTALLATION:

In order to use the sense hat, the user has to update the sense hat. Updating sense hat ensures update of kernel, enables I2.C and installs necessary libraries and programs.

The following are the steps ,the user has to follow sequentially for updating the sense hat.

1. sudo apt-get update

2. sudo apt-get install sense-hat

3. sudo reboot

2.3 SOFTWARE OVERVIEW:

Once the sense hat is updated,the user is able to find some example programs in

/usr/src/sense-hat/examples

For absolute beginners,it is very easy to learn from the examples.

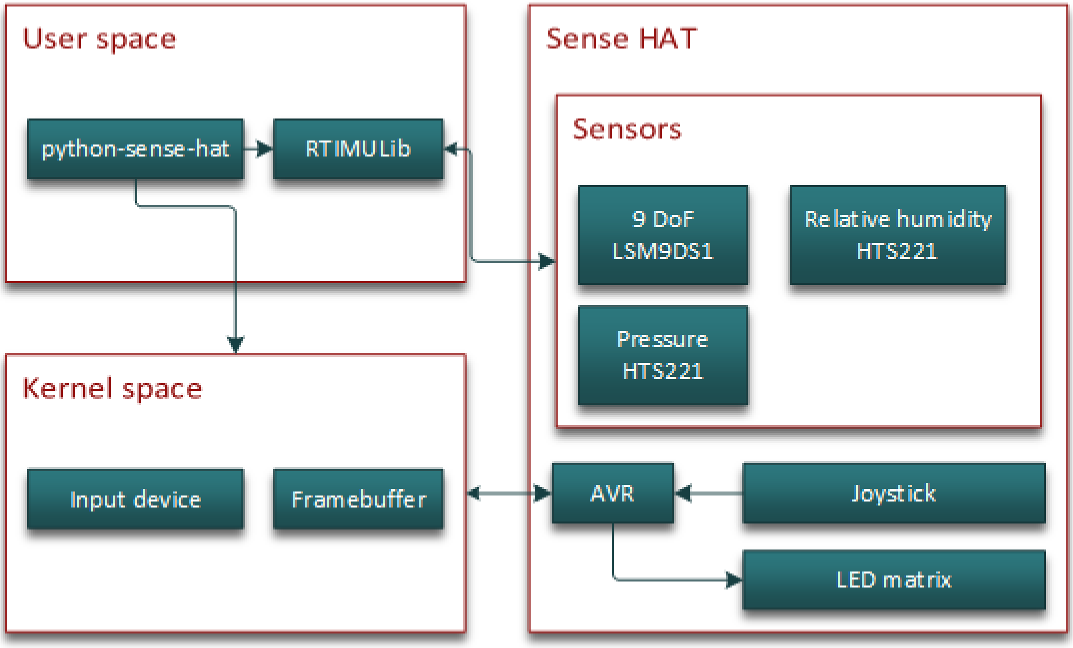

Fig. 2.3.1: Block diagram of sense hat with user space and kernel

(Source : www.raspberrypi.org/documentation/hardware/sense-hat/)

As mentioned earlier, sense hat is composed of multiple sensors. User has to import official sense-hat library to access all the sensors.

2.4 PROGRAMS:

Raspberry Pi equipped with sense hat can do almost anything. Some examples are given below

LED Matrix

set_rotation

When ths esne hat is upside down,the user can use following code to show the image normally

```
from sense_hat import SenseHat

sense = SenseHat()

sense.set_rotation(180)

# alternatives

sense.rotation = 180
```

Parameter	Type	Valid values	Explanation
r	Integer	0 90180 270	The angle to rotate the LED matrix though. 0 is with the Raspberry Pi HDMI port facing downwards.
redraw	Boolean	True False	Whether or not to redraw what is already being displayed on the LED matrix. Defaults to True

flip_h

To flip the images on led horizontally ,the following code is used.

```
from sense_hat import SenseHat

sense = SenseHat()

sense.flip_h()
```

Parameter	Type	Valid values	Explanation
redraw	Boolean	True False	Whether or not to redraw what is already being displayed on the LED matrix. Defaults to True

Returned type	Explanation
List	A list containing 64 smaller lists of [R, G, B] pixels (red, green, blue) representing the flipped image.

flip_v

The image on led can be flip vertically with the following set of lines

```
from sense_hat import SenseHat

sense = SenseHat()

sense.flip_v()
```

Parameter	Type	Valid values	Explanation
redraw	Boolean	True False	Whether or not to redraw what is already being displayed on the LED matrix when flipped. Defaults to True

Returned type	Explanation
List	A list containing 64 smaller lists of [R, G, B] pixels (red, green, blue) representing the flipped image.

show_message

The user can show text message on led.one can program the scroll speed and text color.

```
from sense_hat import SenseHat

sense = SenseHat()

sense.show_message("One small step for Pi!", text_colour=[255, 0, 0])
```

The above program displays a single character

Parameter	**Type**	**Valid values**	**Explanation**
text_string	String	Any text string.	The message to scroll.
scroll_speed	Float	Any floating point number.	The speed at which the text should scroll. This value represents the time paused for between shifting the text to the left by one column of pixels. Defaults to 0.1
text_colour	List	[R, G, B]	A list containing the R-G-B (red, green, blue) colour of the text. Each R-G-B element must be an integer between 0 and 255. Defaults to[255, 255, 255] white.
back_colour	List	[R, G, B]	A list containing the R-G-B (red, green, blue) colour of the background. Each R-G-B element must be an integer between 0 and 255. Defaults to [0, 0, 0] black / off.

2.5 Environmental sensors

get_humidity

To get the percentage of relative humidity from the humidity sensor,the following code is used:

```
from sense_hat import SenseHat


sense = SenseHat()

humidity = sense.get_humidity()

print("Humidity: %s %%rH" % humidity)
```

Returned type	Explanation
Float	The percentage of relative humidity.

With a minor changes in the program one can sense temperature using sense hat.

get_temperature

Calls get_temperature_from_humidity below.

```
from sense_hat import SenseHat


sense = SenseHat()

temp = sense.get_temperature()

print("Temperature: %s C" % temp)
```

The current project is about measuring pressure and the following code is useful for measuring pressure using sense hat.

```
from sense_hat import SenseHat


sense = SenseHat()

pressure = sense.get_pressure()

print("Pressure: %s Millibars" % pressure)
```

.

Returned type	Explanation
Float	The current pressure in Millibars.

REFERENCES:

raspberrypi.org/products/sense-hat/

raspberrypi.org/blog/the-sense-hat-headgear-for-the-terminally-curious/

raspberrypi.org/documentation/hardware/sense-hat/

CHAPTER 3

PYTHON

3.1 INTRODUCTION:

Python is a high level programming language. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It is a free and open source software.

3.2 DATE AND TIME :

Date and time is the most important data to analyze the output for every particular instant of needed time. Date and time can be imported to python to read the year, month, days, hours, minutes and seconds more precisely.

import time

import datetime

Time library provides various time related functions. The date time library is used to provide classes to access and calculate the date and time function in all kind of ways.

time.time()

The time.time() function is the most common function used in python for working with times and for converting between the representations. This function returns the data in floating point which is expressed in seconds from the epoch.

fromtimestamp()

fromtimestamp function is used to return the local date corresponding to the POSIX timestamp which is same as it is returned by time.time() function. This may lead to raise the Value Error, if the timestamp is out of the range of values.

strftime('format’)

strftime(format) function is used to convert the tuple to the format of string as per it is specified in format. The format can be any type of time formats which is represented in directives.

Table 3.1 Directives

DIRECTIVES	MEANING
%S	Seconds as a decimal number
%M	Minutes as a decimal number
%H	Hours as a decimal number
%d	Day as decimal number
%m	Month as decimal number
%Y	Year as decimal number

localtime()

The localtime() function is used to convert the number of seconds to local time. If seconds is not provided then the current time is returned by time() function. The sec parameter is used to convert the number of seconds into the structure struct_time representation.

The time values returned by localtime() function can also be accessed by attributes for every individual fields.

Table 3.2 Attributes

ATTRIBUTE	VALUES
tm_sec	range [0,61]
tm_min	range [0,59]
tm_hour	range [0,23]

3.3 MATHEMATICAL FUNCTIONS:

Import math

Math library is used to import the mathematical functions. These functions cannot be used for complex numbers.

max()

The max() function is a built in function which is used to find the maximum value of an array, list or from the sequence.

min()

The min() function is a built in function which is used to find the minimum value from an array, list or sequence.

abs(x)

The abs(x) function is a built in function which is used to return the absolute value of the number. The argument may be a plain or long integer or a floating point number. If the argument is a complex number, its magnitude is returned.

3.4 NUMPY:

import numpy as np

NumPy library is used to efficiently access and calculate large multi-dimensional arrays of arbitrary records without sacrificing too much speed for small multi-dimensional arrays. For the simplification the numpy can be called as np. Advantages of using Numpy with Python are array oriented computing, efficiently implemented multi-dimensional arrays and designed for scientific computation.

np.array()

The np.array() function is used to create a single or multi-dimensional array from the sequence, list or tuple.

np.std()

The np.std() function is the built in function of NumPy which is used to claculate the standard deviation from the array.

np.mean()

The np.mean() function is the built in function of NumPy which is used to calculate the mean value of the array.

3.5 SENSOR:

import SenseHat

sense=SenseHat()

The Sensehat library is imported to the python to enable the functions available in the sense-hat device. Before importing the SenseHat library the SenseHat package is installed to ensure whether the kernel is up to date and to enable I2C. It provide access to all sensors and LED matrix in the sensehat board. The details about Sensehat are explained in the Chapter-2.

The SenseHat() function is called to return the sensing values to sense variable.

sense.get_pressure()

The sense.get_pressure() function is used to invoke the pressure sensor in the sensehat board to measure the sensed pressure values.

3.6 CSV:

CSV is abbreviated as comma separated values. It’s a most common format which is used to import and export data from spreadsheets and databases. The format is operationally defined by many apllications to read and write it.

import csv

The csv library is used to read and write the tabular data in CSV format. It allows to read and write the data into excel without knowing the precise details of the CSV format used by Excel.

csv.writer

The csv.writer function is a writer object which is used to convert the users data into delimited strings on the required file. It opens a file and writes the data into the file. Normally it is first written in a CSV format file.

open(filename, mode)

CSV format file should be opened before writing the data. Open function is used to open the file. The first argument is a string containing the filename and the second argument is another string which uses character to define the mode. For example to write data the mode is ‘w’ and ‘r’ to read the data.

csvwriter.writerows(rows)

The above function is used to write the data in the format of rows. Rows parameter are the parameters which has to be formatted to rows.

3.7 SQLITE3:

SQlite3 is the lighter version of SQl. SQlite is a user server or database without a server. The difference between SQl and SQlite3 is that in SQl the username, password and server is required whereas in SQlite3 all these are not required. With SQlite the entire database that contains in a single file and also which pulls less memory.

```
import sqlite3
```

The sqlite3 library is imported to use the functions of sqlite3.

```
sqlite3.connect('database.db')
```

To access the module the database should be created. Connect function is used to create the database. The parameter represents the name of the database to be created.

```
conn.cursor()
```

The conn.cursor() function is used to create the cursor. The cursor object is created make any operation with the database and pass the SQL statements to execute them.

```
cursor.execute('''CREATE TABLE users(column1 DATA TYPE,column2 DATA TYPE''')
```

The above function is used to create the table in database with the required column and for each column the data type should be mentioned.

```
cursor.execute('''INSERT INTO users(column1, column2) VALUES(?,?,?,?)''',
(column1,column2))
```

The above function is used to insert data into the table. “?” is used to import the python variables into table. The string operations or concatenation is not used to make queries because it is very insecure.

```
cursor.close()
```

```
sqlite3.connect('database.db').close()
```

The close() function is used to disconnect the database. The cursor and the database should be closed after the execution of database.

CHAPTER 4

4.1 Programming the data logger:

This program of the data logger is the code we have round to become the 1 Hz measurement and the Storage of 10 minutes values continually.

- In this part section, we are going to present different steps of the program. Before start with these Steps, we are going to explain first all the python script we used for the implementation.

import time	For manipulation of the time in python
import datetime	
import math	for the math bibliotheca in python
import numpy as np	For the numerical python
import csv	For comma separate file in python
import string	For the character in python
import sqlite3	For the database in python
from array import array	For the array in python

- The second intersection in the program was the creation of a sense object which represents the Sense HAT. To do it, we used this lines.

```
from sense_hat import SenseHat

variable = SenseHat()

sense.clear()
```

- After that, we need also to connect to the internet to create the database that we are going to use. Here is the code for this connection.

```
sqlite3.connect ('pressure.db')
```

- At the same time, we try also to add a cursor for the facility of the manipulation of the database. To add it, this line help us do it.

```
conn . cursor ()
```

- That our goal is to make a database and calculate some precise values, we need to store it in the array, the raison is that can help to calculate those parameters, we need. Here is the initialization of the array:

```
Array = []
```

- The program is going to measure the pressure for every second, but we need the values for every ten minutes. To achieve this goal we use the counter and here is the initialization of the counter.

```
Counter = 0
```

- Now, we can start the process by taking the first time measurement, which can help us to verify the first condition of the logic (measurement start at modulo ten minutes and zero second). Here is the line, it gives the timestamp at the beginning.

```
ts = datetime.datetime.fromtimestamp (time.time()).strftime('%Y-%m-%d %H:%M:%S')
```

- '%Y-%m-%d %H: %M: %S': this expression means year, month, day, hours, minutes and second.

- For the storage latter, we need to open a comma separate file and write the tittle of every column before. To make it we implement this.

```
with open ('name.csv', 'w') as csv_writer:
    csv_writer = csv.writer (csv_writer)
    csv_writer. writerow(('timestamp', ' press_avg', 'press_min', ' press_max', ' press_std'))
```

- After doing this, we can make sure the code are going to start the implementation of the values at every ten minutes and zero second on actually start the time.

- Example, if we start at 23.03:2016 at 10:02:15, the program is going to wait until 10:10:00 before start.

- Here is the part of code, which implement this condition.

```
if (time.localtime(time.time()).tm_sec==0 and time.localtime(time.time()).tm_min%10==0):
```

- The measurement of one hertz pressure can be made with the timesleep () function of python, but when we have implemented it in the code, we have got a bigger time delay and the time result was not what we expected to have.

  To eliminate this delay we try another approach, what give the better result. This approach is to take the time two times and make a comparison for the one second condition. Here is the first current time measurement to eliminate the time slipe function.

```
 ts = time.time()
mt1 = time.localtime (ts)
```

- After taking the first current time, we can now enter inside the loop and make a measurement for the second time and forward make the condition (this condition means if the difference between those two measurements of the time is not one second, keep trying until that.)

```
while True:
     ts = time.time()
     mt2 = time.localtime (ts)
```

```
mt3=mt1.tm_sec-mt2.tm_sec
 mt4=mt1.tm_min-mt2.tm_min
```

```
if ( abs(mt3)==1 or abs(mt3)==59):
    mt1=mt2
    pressure =sense.get_pressure()
    a.append(pressure)
    arrval=a
    q=np.array (arrval)
```

- Here is the part where we use a counter for ten minutes, to do it we are counting for every one minute and make a counter for ten times, that means 600 values will be store in the array at this range of the time.

```
if ( abs(mt4)==1 or abs(mt4)==59 ):
    Counter = counter + 1
```

- Then come the condition for this ten minutes measurement and the reset of the counter, because after every ten minutes, it is important to start the counter from zero, when we need to respect the time range for the exercise.

```
if (Counter==10):
    Counter = 0
```

- In this small section, we are going to take the time for the measurement, when all the conditions before are true. Like the first time up, it is going to be in the format of year, month, day and then hours, minutes and second.

```
ts =time.time()

st=datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')

print (st)
```

- The next subsection is calling writing values in the comma separate file (csv). What are doing here? : we are going to open the same file we have already open at the beginning for the csv-file and write all the measuring and calculating values inside of it in the order we need. In this case, we start by writhing the time stamp and then the average values, after that the minimum, maximum and standard deviation of the measuring values.

```
with open('second.csv', 'a') as csv_writer:

    csv_writer = csv.writer(csv_writer)

    row = (st, np.mean(q), q.min(), q.max(), np.std(q))

    csv_writer.writerow(row)
```

- After write the values a comma separate file, here we are starting to create a database for the storage of the values. To make this, we create a table if is not exist and inside of the table we create the rows and columns and the name of the different values, we need inside of this table. These commands, we are going to insert that.

```
def create_table():

     c.execute ('CREATE TABLE IF NOT EXISTS pressure

    (timestamp TEXT, mean_pressure REAL, min_pressure REAL,

    max_pressure, REAL,std_pressure REAL)')
```

- After the creation of the table and writing the different tittle of the column inside, now we are going to entry those values with this function.

```
def data_entry():
   c.execute ('''INSERT INTO pressure VALUES (?,?,?,?,?)''',
   (st, np.mean (q), q.min (),  q.max(), np.std(q)))
   conn.commit()
```

- After storage the values, it is also important to call the functions, which can create and execute the table. This both lines are help to do it.

```
create_table()
data_entry()
```

- After this process , it will be good to close the connection and the cursor, but in the this program, we are going to store a data continually, that means , we don't need this both lines , which are also part of the database program.

```
c.close()
conn.close ()
```

- After all this operations, we have to empty the array every ten minutes that is the code to reset the array.

```
del a[:]
```

4.2 EXPERIMENTAL RESULTS:

The data logger for pressure is developed using the Raspberry pi and sense hat and the results (for one hour)are shown below

timestamp	press_avg	press_min	press_max	press_std
3/18/2016 18:00	1009.38801	1009.243408	1009.539063	0.070716685
3/18/2016 18:10	1009.28524	1009.007324	1009.400146	0.078543768
3/18/2016 18:20	1009.0008	1008.894531	1009.104248	0.036953728
3/18/2016 18:30	1008.85551	1008.721191	1008.968506	0.051249957
3/18/2016 18:40	1008.90951	1008.822266	1009.011475	0.040935122
3/18/2016 18:50	1009.0047	1008.9375	1009.09375	0.026158369
3/18/2016 19:00	1009.14177	1008.932373	1009.30957	0.110457499

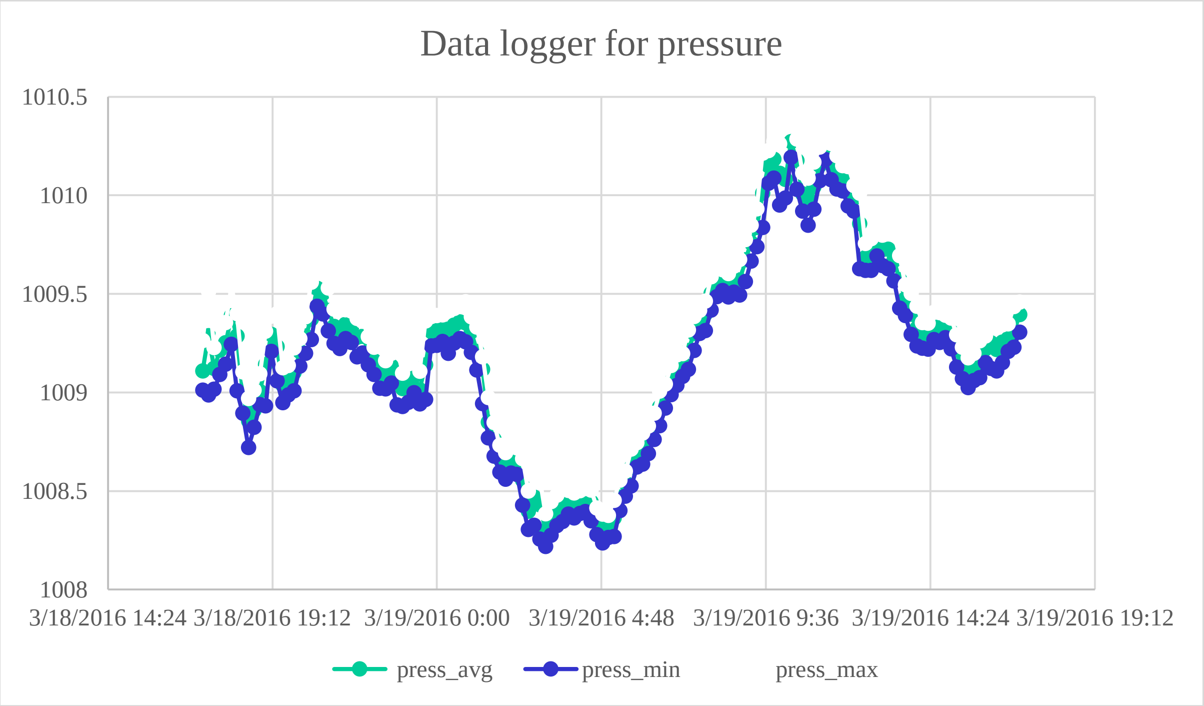

Fig. 4.2.1: Pressure readings

4.3 PLAUSIBILITY:

From the weather forecast, the pressure (March 11, 2016) is 1002 mbar. the minimum and maximum values of pressure varies from 1008 to 1009.

ACKNOWLEDGEMENT:

We express our deep gratitude to Dipl.-Ing. Stefan Balluff for his support throughout the project.

REFERENCES:

Python standard library : https://docs.python.org/2/library/

Numpy : http://docs.scipy.org/doc/numpy-1.10.1/user/install.html

Counter : https://docs.python.org/3.1/library/collections.html

Csv: https://docs.python.org/3/library/csv.html

DB: http://www.tutorialspoint.com/python/python_database_access.html

APPENDIX:

Code:

```
import time
import datetime
import math
import numpy as np
import csv
import string
import sqlite3
from array import array
from sense_hat import SenseHat

sense=SenseHat()
sense.clear()

conn=sqlite3.connect('pressure.db')
c=conn.cursor()


a= []
m = 0


ts=time.time()
st=datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
print (st)


with open('second.csv', 'w') as csv_writer:
    csv_writer = csv.writer(csv_writer)
    csv_writer.writerow(('timestamp', ' press_avg', 'press_min', ' press_max', ' press_std'))


while True:
    if (time.localtime(time.time()).tm_sec==0 and time.localtime(time.time()).tm_min%10==0):
```

```
ts=time.time()
mt1 = time.localtime(ts)



while True:
    ts=time.time()
    mt2 = time.localtime(ts)
    mt3=mt1.tm_sec-mt2.tm_sec
    mt4=mt1.tm_min-mt2.tm_min

    if ( abs(mt3)==1 or abs(mt3)==59):
        mt1=mt2
        pressure =sense.get_pressure()
        a.append(pressure)
        arrval=a
        q=np.array(arrval)


    if ( abs(mt4)==1 or abs(mt4)==59 ):
        m = m + 1
        print(m)
        ts =time.time()
        st=datetime.datetime.fromtimestamp(ts).strftime
         ('%Y-%m-%d %H:%M:%S')
        print (st)

    if (m==10):
        m=0
        ts =time.time()
        st=datetime.datetime.fromtimestamp(ts).strftime
        ('%Y-%m-%d %H:%M:%S')
        print (st)

        with open('second.csv', 'a') as csv_writer:
            csv_writer = csv.writer(csv_writer)
            row = (st, np.mean(q), q.min(), q.max(), np.std(q))
            csv_writer.writerow(row)

            def create_table():
                c.execute('CREATE TABLE IF NOT EXISTS pressure
                (timestamp TEXT,mean_pressure REAL,
    min_pressure REAL,max_pressure REAL,std_pressure REAL)')




def data_entry():
    c.execute('''INSERT INTO pressure VALUES(?,?,?,?,?)''',
    (st, np.mean(q), q.min(), q.max(), np.std(q)))
    conn.commit()

    #c.close()
    #conn.close()

create_table()
data_entry()
del a[:]
```