

VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

Abstract:

There are several critical machines or expensive equipment's which suffer damages due to vibrations and temperature. In such a case, a vibration and temperature sensor is required to find out whether the machine or equipment is producing vibrations, temperature or not. Identifying the object which is continuously vibrating is not a tricky job if the proper sensor is used to detect the vibration, temperature and humidity. So in this project, a basic **vibration sensor module ADXL345** and **DTH 11 sensor module** is interface with **Raspberry Pi 3+**.

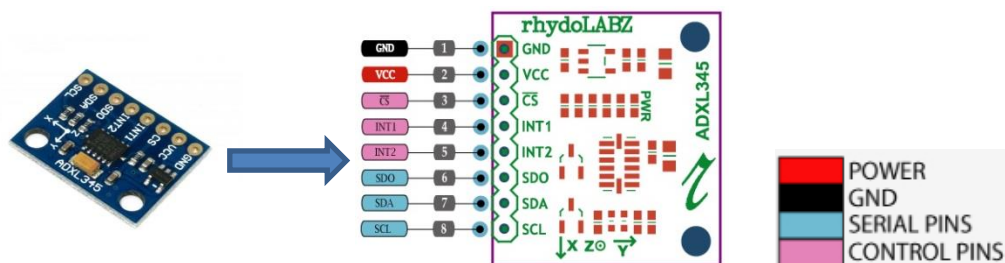
Components Required:

- Raspberry PI 3+
- ADXL-345 Vibration sensor
- DTH-11 Temp & Humid sensor
- Jumper wires
- USB Cable for uploading program

Vibration Sensor Module ADXL-345:

Vibration module, which can work from 3.3V to the 5V. The ADXL345 is a low-power, 3-axis MEMS accelerometer modules with both I2C and SPI interfaces. The Adafruit Breakout boards for these modules feature on-board 3.3v voltage regulation and level shifting which makes them simple to interface with 5v microprocessor such as the Raspberry pi 3+. The ADXL345 features 4 sensitivity ranges from +/- 2G to +/- 16G. And it supports output data rates ranging from 10Hz to 3200Hz.

Vibration Sensor Pin Diagram:

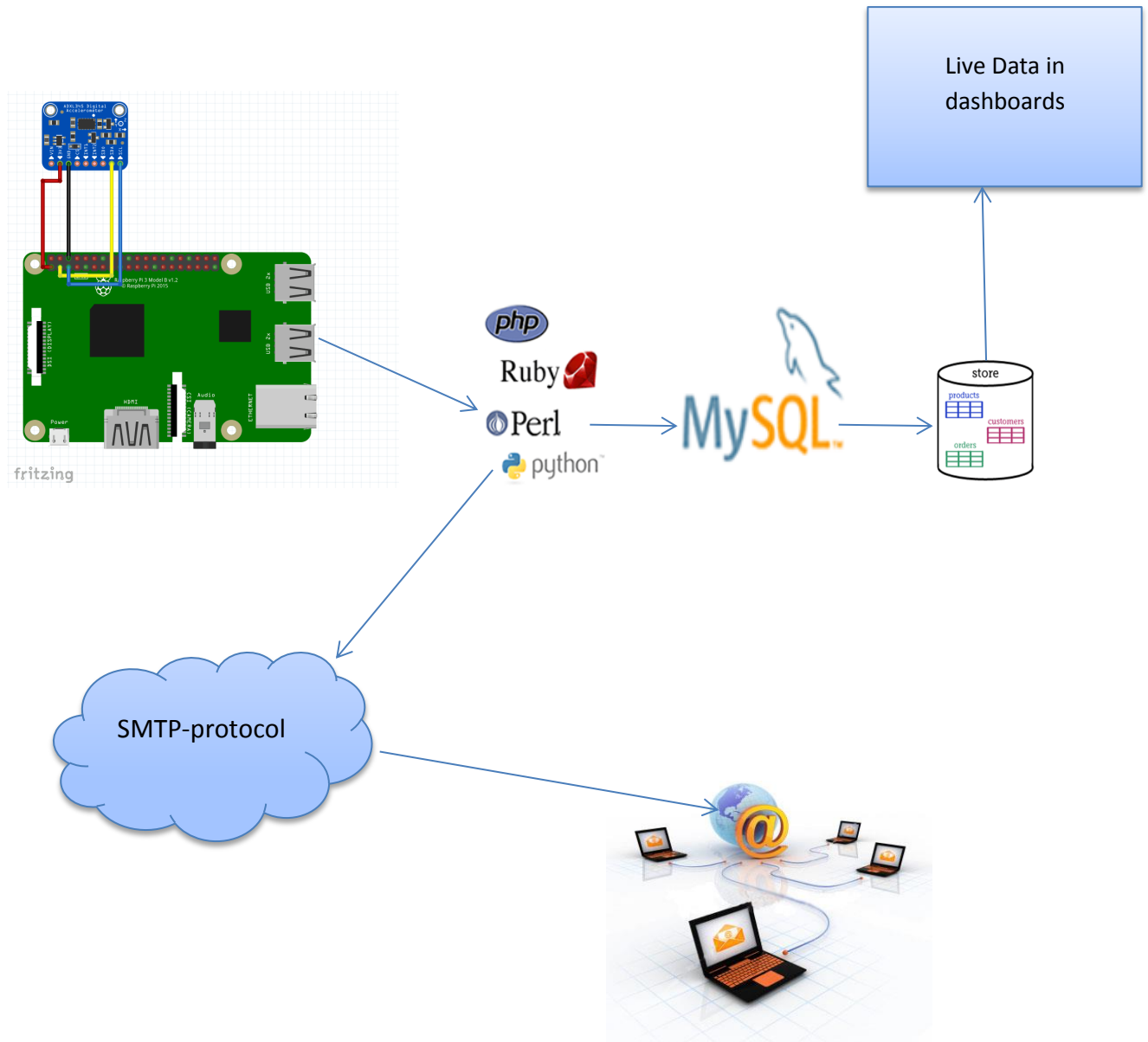


VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

Collect Data:(Sensor to DataBase)

The ADXL345 accelerometer is mounted on the monitoring object (for example on the milling head of CNC milling machine) and connected via I2C bus to the Raspberry Pi microcomputer. The microcomputer acquires (collects) data from the sensor and processes them. The ADXL345 accelerometer is used as a sensor for vibration measurements.

CIRCUIT DIAGRAM:

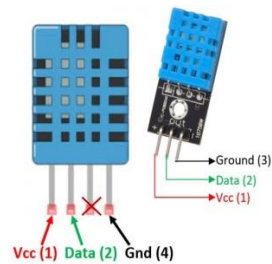


VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

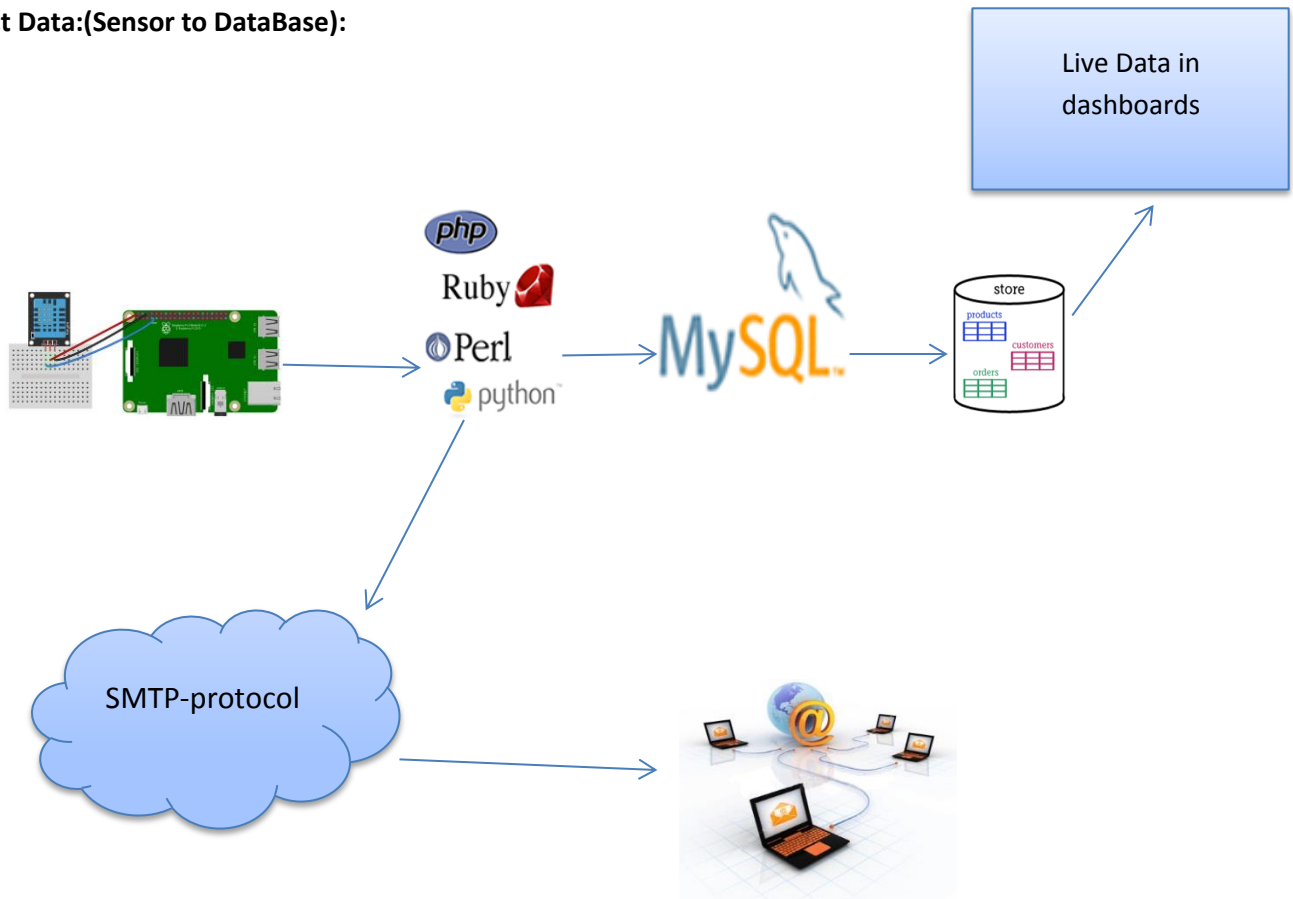
DTH-11 Sensor Module:

This is **DTH-11 Sensor module**, which can work from 3.3V to the 5V. The **DHT-11 sensors** are made of two parts, a capacitive humidity **sensor** and a thermistor. There is also a very basic chip inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity. The digital signal is fairly easy to read using any microcontroller.

Pin Diagram:

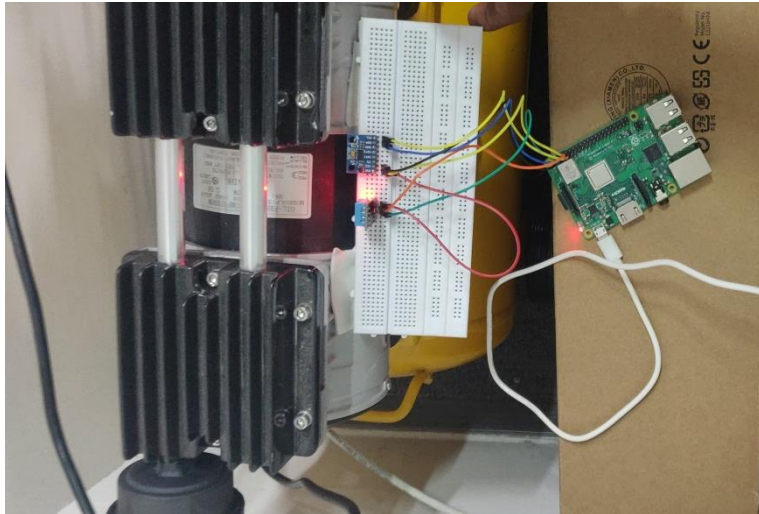


Collect Data:(Sensor to DataBase):



VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

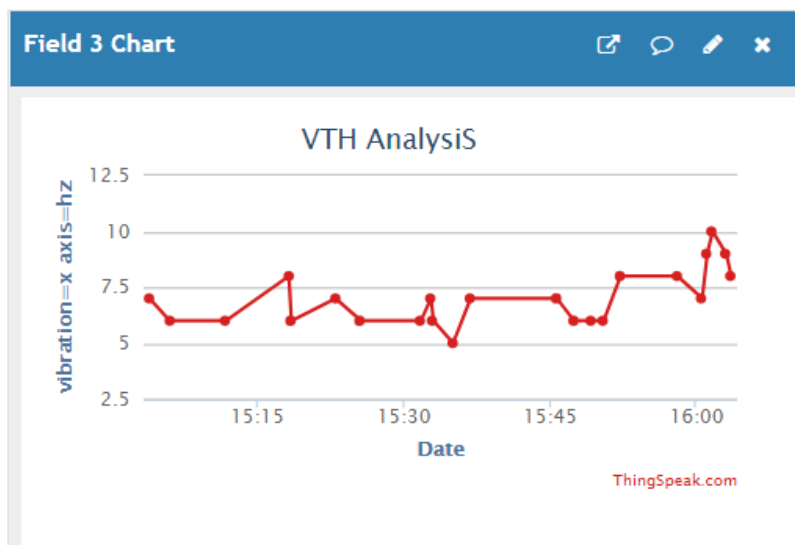
Live Demo For Compressor:



Research Results And Their Analysis:

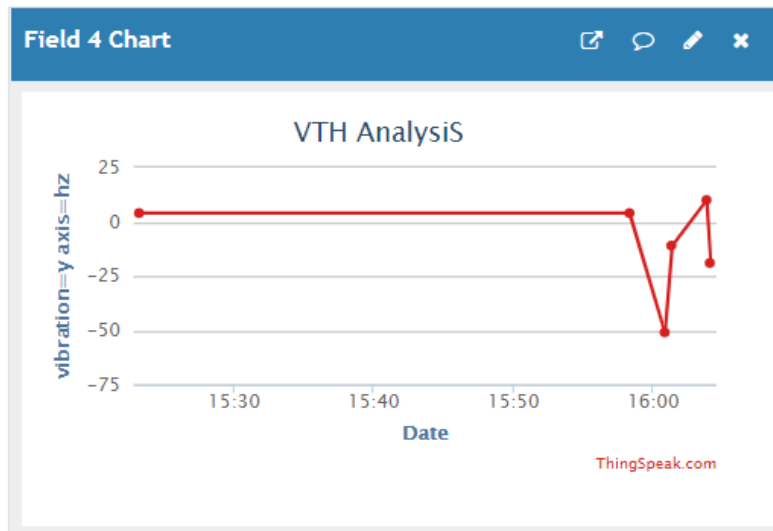
The research results are graphically presented of the vibration accelerations has been shown which are received from the accelerometer on the x-, y-, z - axes, respectively. The graphs show the spectrums of the vibration accelerations for each axis (X, Y and Z). And also temperature and humidity sensor data has been store database.

Change Of The Vibration Acceleration in the X-axis :

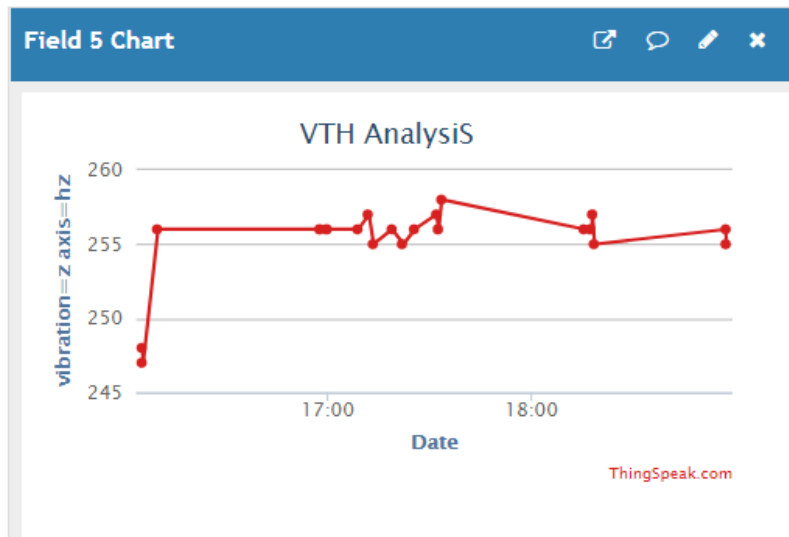


VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

Change Of The Vibration Acceleration in the Y-axis:



Change Of The Vibration Acceleration in the Z-axis:



Vibration acceleration sensor data store to MYSQL database:

phpMyAdmin

Server: MySQL 8.0 Server:3306 Database: sathishdb

Structure

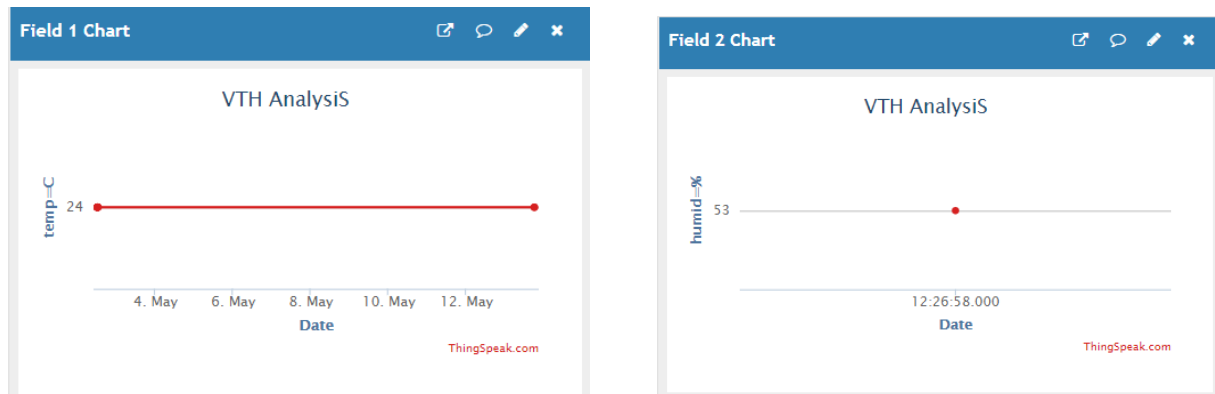
Show all | Number of rows: 25 | Filter rows:

+ Options

DATE TIME +5.30H	X axis Hz	Y axis Hz	Z axis Hz
2019-04-28 00:00:00.00000			
2019-05-13 11:49:04.00000	10	0	256
2019-05-13 11:55:41.00000	8	0	256
2019-05-13 11:55:49.00000	9	0	257
2019-05-13 12:02:06.00000	9	-1	257
2019-05-13 12:02:47.00000	8	-1	256
2019-05-13 12:03:39.00000	8	0	258
2019-05-13 12:45:50.00000	9	7	256
2019-05-13 12:45:58.00000	8	6	256
2019-05-13 12:47:48.00000	8	7	256
2019-05-13 12:47:56.00000	8	8	255
2019-05-13 12:48:03.00000	7	6	257
2019-05-13 12:48:40.00000	8	6	255
2019-05-13 13:27:16.00000	9	0	256
2019-05-13 13:27:25.00000	9	0	255
2019-05-13 13:27:32.00000	8	0	255

VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

Change of the DHT-11 sensor value(temp&humid):

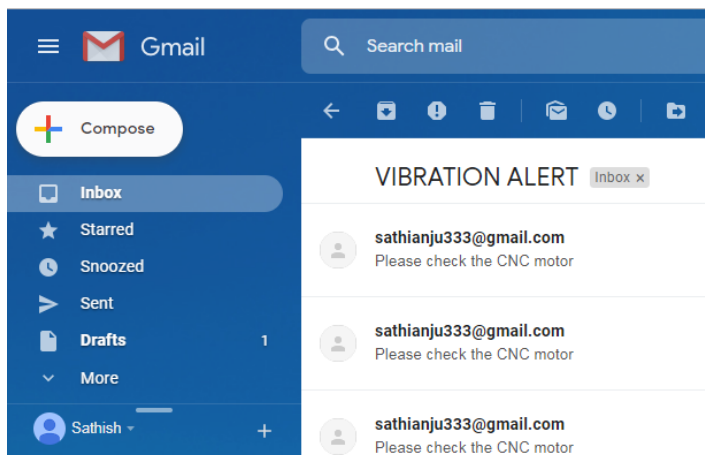


DHT-11 sensor data store to MYSQL database:

The screenshot shows the phpMyAdmin interface. On the left, the database structure is displayed, including 'information_schema', 'sathishdb', and various databases like 'customers', 'data_th', 'dht11_db', 'dht_db', 'distance', 'sensor_db', 'SONY', and 'temp_rec'. On the right, the 'Browse' tab is selected, showing a table with the following data:

DATE	TEMP C	HUMID %
2019-04-30 12:12:12	22	70
2019-04-30 12:12:18	22	76
2019-04-30 12:12:22	22	76
2019-04-30 12:14:46	22	75
2019-04-30 12:14:51	22	76
2019-05-02 05:47:33	23	56
2019-05-02 05:47:37	23	56
2019-05-02 05:47:41	23	56
2019-05-02 05:49:45	23	56
2019-05-02 05:49:49	23	58
2019-05-02 05:49:54	23	58
2019-05-02 06:31:21	23	55
2019-05-02 06:31:27	23	55
2019-05-02 06:31:31	23	55
2019-05-02 06:31:35	23	55
2019-05-02 06:31:39	23	55
2019-05-02 06:31:43	23	55

SMTP-protocol using vibration alert through Mail:



CODE:(ADXL-345)

```
import time
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
import urllib.request as ur
from urllib.request import urlopen
from bs4 import BeautifulSoup
#from httplib import HTTPResponse
import urllib
#import dht11
import time
import mysql.connector
from mysql.connector import Error
from mysql.connector import errorcode
# Import the ADXL345 module.
import Adafruit_ADXL345

# Create an ADXL345 instance.
accel = Adafruit_ADXL345.ADXL345()
connection = mysql.connector.connect(host='db4free.net',
                                     database='sathishdb',
                                     user='sathish',
                                     password='sathish@14')

cursor = connection.cursor()

# Alternatively you can specify the device address and I2C bus with parameters:
#accel = Adafruit_ADXL345.ADXL345(address=0x54, busnum=2)

# You can optionally change the range to one of:
# - ADXL345_RANGE_2_G = +/-2G (default)
# - ADXL345_RANGE_4_G = +/-4G
# - ADXL345_RANGE_8_G = +/-8G
# - ADXL345_RANGE_16_G = +/-16G
# For example to set to +/- 16G:
#accel.set_range(Adafruit_ADXL345.ADXL345_RANGE_16_G)

# Or change the data rate to one of:
# - ADXL345_DATARATE_0_10_HZ = 0.1 hz
# - ADXL345_DATARATE_0_20_HZ = 0.2 hz
# - ADXL345_DATARATE_0_30_HZ = 0.30 hz
```

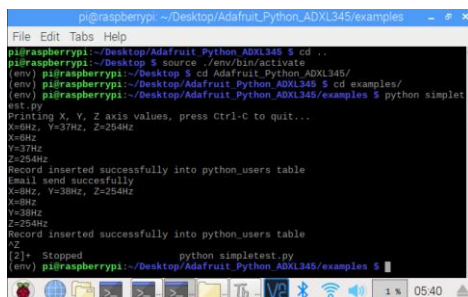
VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

```
print('Printing X, Y, Z axis values, press Ctrl-C to quit...')
while True:
    # Read the X, Y, Z axis acceleration values and print them.
    x, y, z = accel.read()
    print('X={0}Hz, Y={1}Hz, Z={2}Hz'.format(x, y, z))
    print('X={0}Hz'.format(x))
    print('Y={0}Hz'.format(y))
    print('Z={0}Hz'.format(z))
    # Wait half a second and repeat.
    time.sleep(0.5)
    X='{0}'.format(x)
    #req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2BJX$
    req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2BJXDA7Z517Y7IZM&field3="+str(x))
    Y='{0}'.format(y)
    #req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2BJX$
    req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2BJXDA7Z517Y7IZM&field4="+str(y))
    Z='{0}'.format(z)
    #req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2BJX$
    req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2BJXDA7Z517Y7IZM&field1="+str(z))
    c = connection.cursor()
    c.execute("insert into `SONY`(`DATE TIME` +5.30H`,`X axis Hz`,`Y axis Hz`,`Z axis Hz`)values(sysdate(),%s,%s,%s)",(X,Y,Z))
    connection.commit()
    print ("Record inserted successfully into python_users table")
    while(x>38):
        fromaddr="sathianju333@gmail.com"
        toaddr="rajkumar.m@wabco-auto.com"
        msg = MIMEText(body, 'plain')
        msg['From']=fromaddr
        msg['To']=toaddr
        msg['subject']="VIBRATION ALERT"
        body="Please check the CNC motor"
        msg.attach(MIMEText(body, 'plain'))
        server=smtplib.SMTP('smtp.gmail.com',587)
        server.ehlo()
        server.starttls()
        server.ehlo()
        server.login("sathianju333@gmail.com","anjugam14")
        server.sendmail(fromaddr,toaddr,msg.as_string())
        print("Email send succesfully")
        break
    server.quit()
```


VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

```
server.quit()
while (y>238):
    fromaddr="sathianju333@gmail.com"
    toaddr="rajkumar.m@wabco-auto.com"
    msg = MIMEMultipart()
    msg['From']=fromaddr
    msg['To']=toaddr
    msg['subject']="VIBRATION ALERT"
    body="Please check the CNC motor"
    msg.attach(MIMEText(body,'plain'))
    server=smtplib.SMTP('smtp.gmail.com',587)
    server.ehlo()
    server.starttls()
    server.ehlo()
    server.login("sathianju333@gmail.com","anjugam14")
    server.sendmail(fromaddr,toaddr,msg.as_string())
    print("Email send succesfully")
    break
server.quit()
while (z>=235):
    fromaddr="sathianju333@gmail.com"
    toaddr="rajkumar.m@wabco-auto.com"
    msg = MIMEMultipart()
    msg['From']=fromaddr
    msg['To']=toaddr
    msg['subject']="VIBRATION ALERT"
    body="Please check the CNC motor"
    msg.attach(MIMEText(body,'plain'))
    server=smtplib.SMTP('smtp.gmail.com',587)
    server.ehlo()
    server.starttls()
    server.ehlo()
    server.login("sathianju333@gmail.com","anjugam14")
    server.sendmail(fromaddr,toaddr,msg.as_string())
    print("Email send succesfully")
    break
server.quit()
```

OUTPUT:



```
pi@raspberrypi: ~/Desktop/Adafruit_Python_ADXL345/examples
File Edit Tabs Help
pi@raspberrypi:~/Desktop/Adafruit_Python_ADXL345 $ cd ..
pi@raspberrypi:~/Desktop $ source /env/bin/activate
(env) pi@raspberrypi:~/Desktop $ cd Adafruit_Python_ADXL345/
(env) pi@raspberrypi:~/Desktop/Adafruit_Python_ADXL345 $ cd examples/
(env) pi@raspberrypi:~/Desktop/Adafruit_Python_ADXL345/examples $ python simpletest.py
Printing X, Y, Z axis values, press Ctrl-C to quit...
X=8Hz, Y=37Hz, Z=254Hz
X=8Hz
Y=37Hz
Z=254Hz
Record inserted successfully into python_users table
Email send successfully
X=8Hz, Y=38Hz, Z=254Hz
X=8Hz
Y=38Hz
Z=254Hz
Record inserted successfully into python_users table
^Z
[2]+  Stopped                  python simpletest.py
(env) pi@raspberrypi:~/Desktop/Adafruit_Python_ADXL345/examples $
```

VIBRATION AND TEMPERATURE AND HUMIDITY ANALYSIS

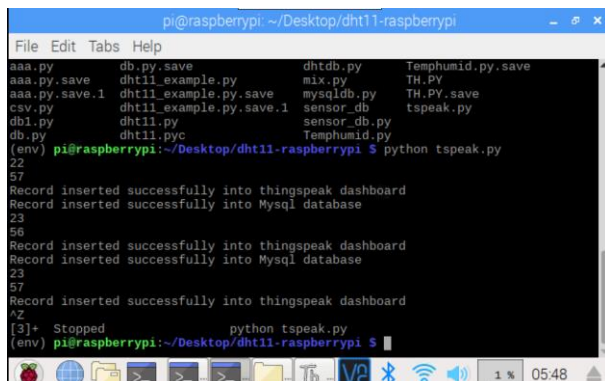
CODE:(dht-11)

```
Rpi.GPIO as GPIO
urllib.request as ur
l1ib.request import urlopen
4 import BeautifulSoup
ttmlib import HTTPResponse
urllib
dht11
time
mysql.connector
sql.connector import Error
sql.connector import errorcode
datetime
l1ib.error import HTTPError
alize GPIO
twarnings(False)
tmode(GPIO.BCM)
leanup()
ata using Pin GPIO21
e = dht11.DHT11(pin=4)

ion = mysql.connector.connect(host='db4free.net',
                               database='sathishdb',
                               user='sathish',
                               password='sathish@14')
= connection.cursor()
rue:|
ult = instance.read()
result.is_valid():
    temp="%d" % result.temperature
    print(temp)
    humid="%d" % result.humidity
    #humid="70%"
    print(humid)
```

```
humid="%d" % result.humidity
#humid="70%"
print(humid)
req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2B3XDA7Z517Y7IZM&field1="+str(temp))
#humid="%d" % result.humidity
req=urllib.request.urlopen("https://api.thingspeak.com/update?api_key=2B3XDA7Z517Y7IZM&field2="+str(humid))
print("Record inserted successfully into thingspeak dashboard")
c = connection.cursor()
c.execute("insert into `sensor_db` (`DATE`,`TEMP` C`,`HUMID` %) values(sysdate(),%s,%s)",(temp,humid))
connection.commit()
print ("Record inserted successfully into python_users table")
```

OUTPUT:



```
pi@raspberrypi: ~/Desktop/dht11-raspberrypi
File Edit Tabs Help
aaa.py db.py.save dhtdb.py Temphumid.py.save
aaa.py.save dht11_example.py mix.py TH.PY
aaa.py.save.1 dht11_example.py.save mysql_db.py TH.PY.save
csv.py dht11_example.py.save.1 sensor_db tspeak.py
db1.py dht11.py sensor_db.py
db.py dht11.pyc Temphumid.py
(env) pi@raspberrypi:~/Desktop/dht11-raspberrypi $ python tspeak.py
22
57
Record inserted successfully into thingspeak dashboard
Record inserted successfully into Mysql database
23
58
Record inserted successfully into thingspeak dashboard
Record inserted successfully into Mysql database
23
57
Record inserted successfully into thingspeak dashboard
^Z
[3]+ Stopped python tspeak.py
(env) pi@raspberrypi:~/Desktop/dht11-raspberrypi $
```