Alarm to System High Temperature

Assignment -2

By

Parvathi @ viji

952319106025

BUILD A PYTHON CODE,ASSUME U GET TEMPERATURE AND HUMIDITY VALUE AND WRITE A CONDITION TO CONTINUOUSLY DETECT ALARM IN CASE OF HIGH TEMPERATURE:

This article, we'll discuss interacting DHT11 with Raspberry Pi and see it working using Python code. Also, we'll display real-time Data on the 16×2 LCD. The code and explanation used in the code will be explained further below also all the modules regarding 16×2 LCD will be included with its article home page. So let's begin.

DHT-11
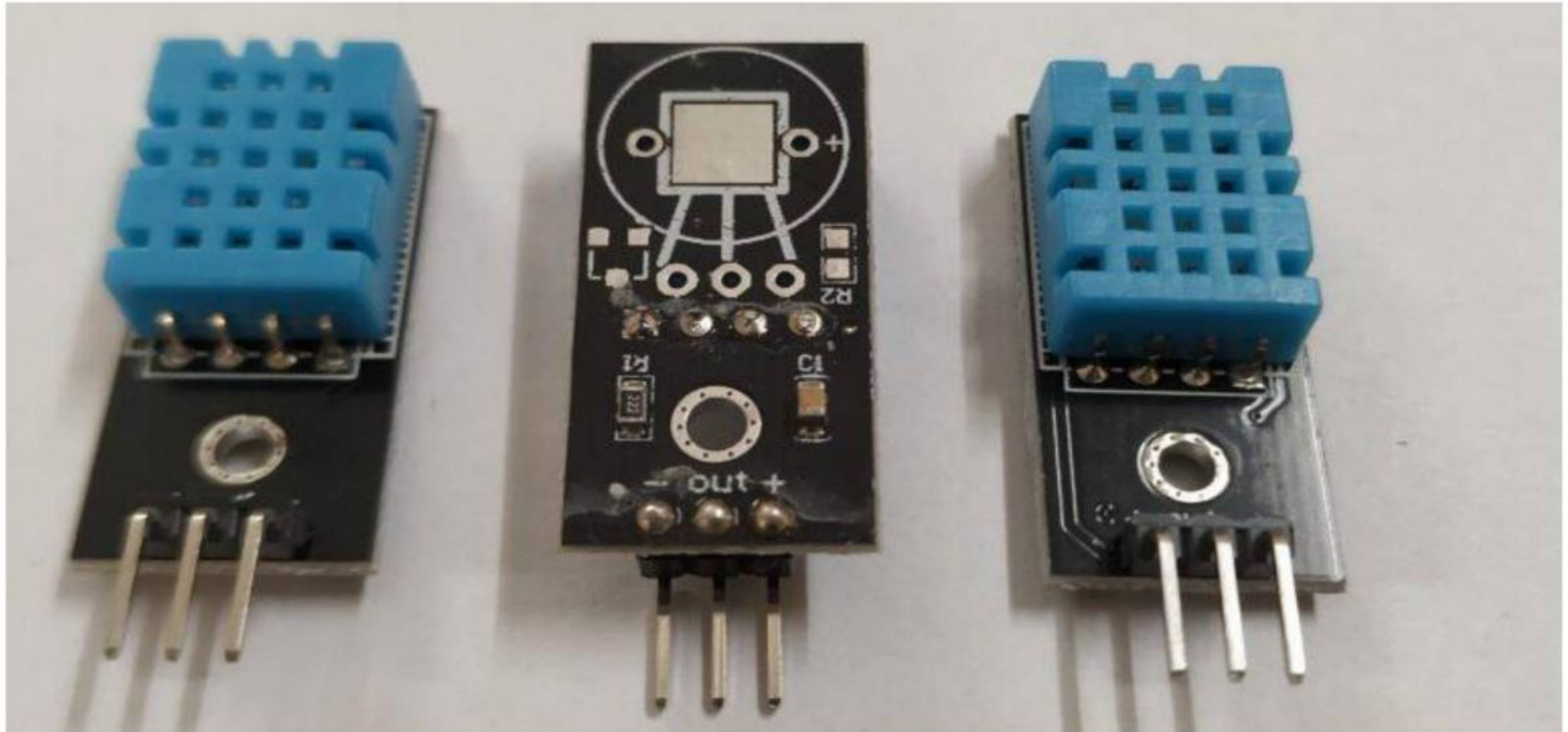
- It is the most common and famous temperature and humidity combined sensor you'll ever come to know. It has many tutorials with boards similar to Arduino.

- You may visit It's Arduino Tutorial to have a more clear Idea of Its working if you are

working so, here we are giving you thetutorial on how to connect dht11 with Raspberry Pi.

- DHT11 is a simple sensor and has a very simple structure for measuring temperature and humidity. Basically, it is an enclosed structure that consists of two wires which are responsible for checking humidity and temperature.

Material Requirement

- Raspberry Pi with keyboard and mouse. Or putty terminal.

- Breadboard

- Jumper wires

- DHT11 Sensor

DHT11 with Raspberry pi Circuit Diagram

DHT11 Temperature & Humidity Sensor

Raspberry Pi
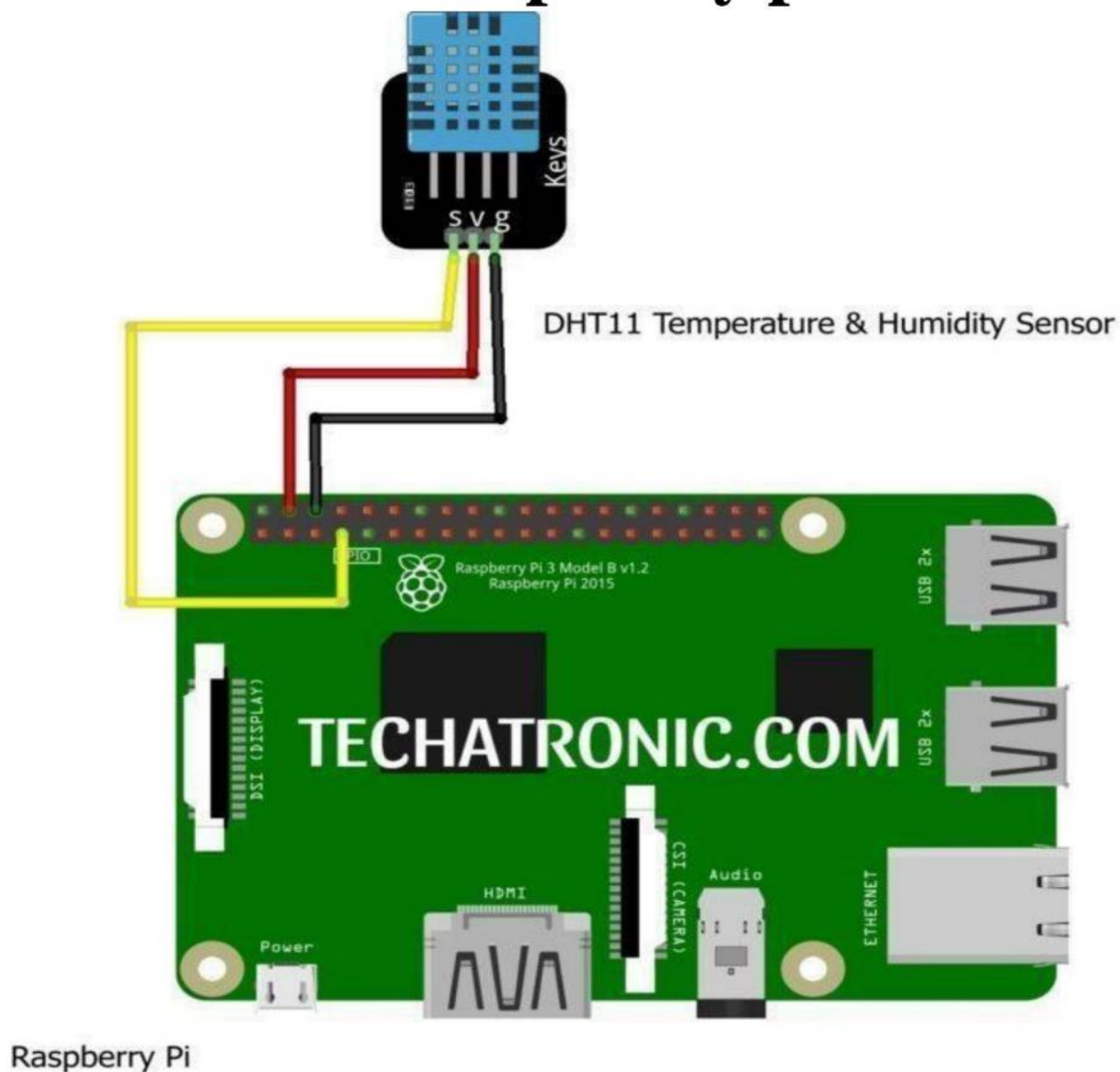
Import Adafruit_DHT
DHT11=Adafruit_DHT.DHT11 # Adafruit_DHT.DHT22 for DHT22 sensor.

While True:

  Try:

```
Temp,humid=Adafruit_DHT.read_retry(DHT
11,4) # 4 is ithe GPIO number you can change
this to your required need
    Print("TEMP={0:0.1f}°C
HUMIDITY={1:0.1f}%".format(temp,h
umid))     Except KeyboardInterrupt:
    Break
```

- The first line as we have said we have imported the library for the DHT11 Sensor to work i.e., Adafruit_DHT. You can use this library with DHt22 also, but you need to change the DHT11 object

line.

- Then we create a DHT object which store the DHT11 sensor configuration details and further in code we use this object name to refer to all working statements.

- Next we create an infinite while loop within Try and except method to create a keyboard interrupt terminating condition

  i.e., Ctrl+C

- In next line we read data from the DHT11 sensor and stores it in two variable as two

values are being received, one for

temperature and other for humidity.

CODING:

```python
!/usr/bin/python
import struct, array, time, io,
fcntl

I2C_SLAVE=0x0703

# find with  sudo i2cdetect -y 1

HDC1008_ADDR = 0x40

                            bus=1
fr = io.open("/dev/i2c-
    "+str(bus), "rb", buffering=0)
fw = io.open("/dev/i2c-
    "+str(bus), "wb", buffering=0)

# set device address
fcntl.ioctl(fr, I2C_SLAVE,
    HDC1008_ADDR)
```

```python
fcntl.ioctl(fw, I2C_SLAVE,
    HDC1008_ADDR)
time.sleep(0.015) #15ms startup
time
                  s = [0x02,0x02,0x00]
s2 = bytearray( s )
fw.write( s2 ) #sending config
    register bytes
time.sleep(0.015)
    # From the data sheet

                    s = [0x00] # temp
s2 = bytearray( s )
fw.write( s2 )
time.sleep(0.0625)
    # From the data sheet

    data = fr.read(2) #read 2 byte
                    temperature data
buf = array.array('B', data)
print ( "Temp: %f" % (
    (((((buf[0]<<8) +
    (buf[1]))/65536.0)*165.0 ) -
    40.0  )   )


    time.sleep(0.015)
    # From the data sheet

                    s = [0x01] # hum
```

```python
s2 = bytearray( s )
fw.write( s2 )
time.sleep(0.0625)
   # From the data sheet

      data = fr.read(2) #read 2 byte
                     temperature data
buf = array.array('B', data)
print ( "Humidity: %f" % (
   ((((buf[0]<<8) +
   (buf[1]))/65536.0)*100.0 ) ) )
```