```
#coding: utf-8

from smbus2 import SMBus
import time
from time import sleep
from urllib2 import urlopen

bus_number  = 1
i2c_address = 0x76

bus = SMBus(bus_number)

digT = []
digP = []
digH = []

t_fine = 0.0
pressure = 0.0
temperature = 0.0
var_h = 0.0


def writeReg(reg_address, data):
	bus.write_byte_data(i2c_address,reg_address,data)

def get_calib_param():
	calib = []

	for i in range (0x88,0x88+24):
		calib.append(bus.read_byte_data(i2c_address,i))
	calib.append(bus.read_byte_data(i2c_address,0xA1))
	for i in range (0xE1,0xE1+7):
		calib.append(bus.read_byte_data(i2c_address,i))

	digT.append((calib[1] << 8) | calib[0])
	digT.append((calib[3] << 8) | calib[2])
	digT.append((calib[5] << 8) | calib[4])
	digP.append((calib[7] << 8) | calib[6])
	digP.append((calib[9] << 8) | calib[8])
	digP.append((calib[11]<< 8) | calib[10])
	digP.append((calib[13]<< 8) | calib[12])
	digP.append((calib[15]<< 8) | calib[14])
	digP.append((calib[17]<< 8) | calib[16])
	digP.append((calib[19]<< 8) | calib[18])
	digP.append((calib[21]<< 8) | calib[20])
	digP.append((calib[23]<< 8) | calib[22])
	digH.append( calib[24] )
	digH.append((calib[26]<< 8) | calib[25])
```

```
        digH.append( calib[27] )
        digH.append((calib[28]<< 4) | (0x0F & calib[29]))
        digH.append((calib[30]<< 4) | ((calib[29] >> 4) & 0x0F))
        digH.append( calib[31] )

        for i in range(1,2):
                if digT[i] & 0x8000:
                        digT[i] = (-digT[i] ^ 0xFFFF) + 1

        for i in range(1,8):
                if digP[i] & 0x8000:
                        digP[i] = (-digP[i] ^ 0xFFFF) + 1

        for i in range(0,6):
                if digH[i] & 0x8000:
                        digH[i] = (-digH[i] ^ 0xFFFF) + 1

def readData():
        data = []
        for i in range (0xF7, 0xF7+8):
                data.append(bus.read_byte_data(i2c_address,i))
        pres_raw = (data[0] << 12) | (data[1] << 4) | (data[2] >> 4)
        temp_raw = (data[3] << 12) | (data[4] << 4) | (data[5] >> 4)
        hum_raw  = (data[6] << 8)  |  data[7]

        compensate_T(temp_raw)
        compensate_P(pres_raw)
        compensate_H(hum_raw)
      data = str(temperature) + "," + str(var_h) + "," + str(pressure)
#namaeの"imura"を変更してください。
        url = "http://iot.iway.jp/?namae=imura&ondo=%s" % data
        response = urlopen(url)

def compensate_P(adc_P):
        global  t_fine
      global  pressure
        pressure = 0.0

        v1 = (t_fine / 2.0) - 64000.0
        v2 = (((v1 / 4.0) * (v1 / 4.0)) / 2048) * digP[5]
        v2 = v2 + ((v1 * digP[4]) * 2.0)
        v2 = (v2 / 4.0) + (digP[3] * 65536.0)
        v1 = (((digP[2] * (((v1 / 4.0) * (v1 / 4.0)) / 8192)) / 8)  + ((digP[1] * v1) / 2.0)) / 262144
        v1 = ((32768 + v1) * digP[0]) / 32768

        if v1 == 0:
                return 0
        pressure = ((1048576 - adc_P) - (v2 / 4096)) * 3125
```

```
        if pressure < 0x80000000:
                pressure = (pressure * 2.0) / v1
        else:
                pressure = (pressure / v1) * 2
        v1 = (digP[8] * (((pressure / 8.0) * (pressure / 8.0)) / 8192.0)) / 4096
        v2 = ((pressure / 4.0) * digP[7]) / 8192.0
        pressure = pressure + ((v1 + v2 + digP[6]) / 16.0)

        print("pressure : %7.2f hPa") % (pressure/100)

def compensate_T(adc_T):
        global t_fine
        global temperature
        v1 = (adc_T / 16384.0 - digT[0] / 1024.0) * digT[1]
        v2 = (adc_T / 131072.0 - digT[0] / 8192.0) * (adc_T / 131072.0 - digT[0] / 8192.0) *
digT[2]
        t_fine = v1 + v2
        temperature = t_fine / 5120.0
        print("temp : %-6.2f ℃") % (temperature)

def compensate_H(adc_H):
        global t_fine
        global var_h
        var_h = t_fine - 76800.0
        if var_h != 0:
                var_h = (adc_H - (digH[3] * 64.0 + digH[4]/16384.0 * var_h)) * (digH[1] /
65536.0 * (1.0 + digH[5] / 67108864.0 * var_h * (1.0 + digH[2] / 67108864.0 * var_h)))
        else:
                return 0
        var_h = var_h * (1.0 - digH[0] * var_h / 524288.0)
        if var_h > 100.0:
                var_h = 100.0
        elif var_h < 0.0:
                var_h = 0.0
        print("hum : %6.2f %") % (var_h)


def setup():
        osrs_t = 1                      #Temperature oversampling x 1
        osrs_p = 1                      #Pressure oversampling x 1
        osrs_h = 1                      #Humidity oversampling x 1
        mode   = 3                      #Normal mode
        t_sb   = 5                      #Tstandby 1000ms
        filter = 0                      #Filter off
        spi3w_en = 0                    #3-wire SPI Disable

        ctrl_meas_reg = (osrs_t << 5) | (osrs_p << 2) | mode
        config_reg    = (t_sb << 5) | (filter << 2) | spi3w_en
```

```
        ctrl_hum_reg  = osrs_h

        writeReg(0xF2,ctrl_hum_reg)
        writeReg(0xF4,ctrl_meas_reg)
        writeReg(0xF5,config_reg)


setup()
get_calib_param()


if __name__ == '__main__':
        try:
                while True:
                        readData()
                        sleep(10)
        except KeyboardInterrupt:
                pass
```