weather sensor that measures temperature, pressure, and humidity

|  |
| --- |
| #!/usr/bin/env python3 |
|  |
| import time |
| from pms5003 import PMS5003, ReadTimeoutError |
| import logging |
|  |
| logging.basicConfig( |
| format='%(asctime)s.%(msecs)03d %(levelname)-8s %(message)s', |
| level=logging.INFO, |
| datefmt='%Y-%m-%d %H:%M:%S') |
|  |
| logging.info("""particulates.py - Print readings from the PMS5003 particulate sensor. |
|  |
| Press Ctrl+C to exit! |
|  |
| """) |
|  |
| pms5003 = PMS5003() |
| time.sleep(1.0) |
|  |
| try: |
| while True: |
| try: |
| readings = pms5003.read() |
| logging.info(readings) |
| except ReadTimeoutError: |
| pms5003 = PMS5003() |
| except KeyboardInterrupt: |
| pass |

senses particulates of various sizes (PM1, PM2.5, PM10) from sources like smoke, dust, pollen, metal and organic particles, and more

|  |
| --- |
| import sys |
| import mock |
| import pytest |
| import struct |
|  |
|  |
| class MockSerialFail(): |
| def \_\_init\_\_(self): |
| pass |
|  |
| def read(self, length): |
| return b'\x00' \* length |
|  |
|  |
| class MockSerial(): |
| def \_\_init\_\_(self): |
| self.ptr = 0 |
| self.sof = b'\x42\x4d' |
| self.data = self.sof |
| self.data += struct.pack('>H', 28) |
| self.data += b'\x00' \* 26 |
| checksum = struct.pack('>H', sum(bytearray(self.data))) |
| self.data += checksum |
|  |
| def read(self, length): |
| result = self.data[self.ptr:self.ptr + length] |
| self.ptr += length |
| if self.ptr >= len(self.data): |
| self.ptr = 0 |
| return result |
|  |
|  |
| def \_mock(): |
| sys.modules['RPi'] = mock.Mock() |
| sys.modules['RPi.GPIO'] = mock.Mock() |
| sys.modules['serial'] = mock.Mock() |
|  |
|  |
| def test\_setup(): |
| \_mock() |
| import pms5003 |
| sensor = pms5003.PMS5003() |
| del sensor |
|  |
|  |
| def test\_double\_setup(): |
| \_mock() |
| import pms5003 |
| sensor = pms5003.PMS5003() |
| sensor.setup() |
|  |
|  |
| def test\_read(): |
| \_mock() |
| import pms5003 |
| sensor = pms5003.PMS5003() |
| sensor.\_serial = MockSerial() |
| data = sensor.read() |
| data.pm\_ug\_per\_m3(2.5) |
|  |
|  |
| def test\_read\_fail(): |
| \_mock() |
| import pms5003 |
| sensor = pms5003.PMS5003() |
| sensor.\_serial = MockSerialFail() |
| with pytest.raises(pms5003.ReadTimeoutError): |
| data = sensor.read() |
| del data |

Detects CO, NO2, Ethanol, Hydrogen, Ammonia and Methane – most important substances it detects are Carbon Monoxide and Nitrogen Dioxide gas.

|  |
| --- |
| #define Serial SerialUSB |
|  |  |
|  | //analog read |
|  | int co=0; |
|  | int no2=0; |
|  |  |
|  | //Value voltaje |
|  | float vco=0; |
|  | float vno2=0; |
|  |  |
|  | //Value resistencia |
|  | float rco=0; |
|  | float rno2=0; |
|  |  |
|  | //Value Rs/Ro |
|  | float conCO=0; |
|  | float conNO2=0; |
|  |  |
|  | //cálculo de ppm |
|  | double ppmCO=0; |
|  | double ppmNO2=0; |
|  |  |
|  | void setup() { |
|  | //resolution analog reads |
|  | analogReadResolution(12); |
|  | Serial.begin(9600); |
|  | //pre heating |
|  | while(!Serial) |
|  | pinMode(9,OUTPUT); |
|  | Serial.println("pre heating"); |
|  | digitalWrite(9, HIGH); |
|  | //delay (30000); |
|  | Serial.println("pre heating done"); |
|  | digitalWrite(9, LOW); |
|  | } |
|  |  |
|  | void loop() { |
|  |  |
|  | //Get Sensor Data |
|  | co=analogRead(A0); |
|  | no2=analogRead(A1); |
|  | //Convert to voltaje |
|  | vco=(3.3\*co)/4096; |
|  | vno2=(3.3\*no2)/4096; |
|  | //Convert to resist |
|  | rco=47000\*((3.3-vco)/vco);//load resistor in red 1ohm |
|  | rno2=((270\*(3.3-vno2))/vno2);//load resistor in ox 270ohm |
|  | //Convert to indicator concentration |
|  | conCO= 47000/rco; |
|  | conNO2= 270/rno2; |
|  | //Calculo de particulas por millon |
|  | ppmCO=(4.4138\*pow(conCO,-1.178)); |
|  | ppmNO2= ((-0.0003\*(conNO2\*conNO2))+(0.1626\*conNO2)-0.0217); |
|  | Serial.print("PPM CO"); |
|  | Serial.print(" "); |
|  | Serial.print(ppmCO); |
|  | Serial.print("PPM NO2"); |
|  | Serial.print(" "); |
|  | Serial.println(ppmNO2); |
|  |  |
|  | delay(1000); |
|  | } |

It detects VOCs (Volatile Organic Compounds) and has a gas sensor chip to detect carbon dioxide. It also has a temperature and humidity sensor (this sensor detects similar things to the Enviro + sensor).

#include "Adafruit\_CCS811.h"

Adafruit\_CCS811 ccs;

void setup() {

Serial.begin(9600);

Serial.println("CCS811 test");

if(!ccs.begin()){

Serial.println("Failed to start sensor! Please check your wiring.");

while(1);

}

//calibrate temperature sensor

while(!ccs.available());

float temp = ccs.calculateTemperature();

ccs.setTempOffset(temp - 25.0);

}

void loop() {

if(ccs.available()){

float temp = ccs.calculateTemperature();

if(!ccs.readData()){

Serial.print("CO2: ");

Serial.print(ccs.geteCO2());

Serial.print("ppm, TVOC: ");

Serial.print(ccs.getTVOC());

Serial.print("ppb Temp:");

Serial.println(temp);

}

else{

Serial.println("ERROR!");

while(1);

}

}

delay(500);

}