A Manual on Outdoor Air Pollution Monitoring

Learning Objective: At the end of this workshop, participants will be able to:

a) analyze particulate matter in the surrounding by interfacing SDS sensor with ESP8266.
b) understand the behaviour of particulate matter with respect to variation of temperature and humidity.

Components Required:

ESP8266 - 1 pcs

SDS 011 - 1 pcs

AHT10 (for Temperature and Humidity) – 1 pcs

Jumper Wires

Breadboard

USB Cable - 1 pcs

Arduino IDE

Theory:

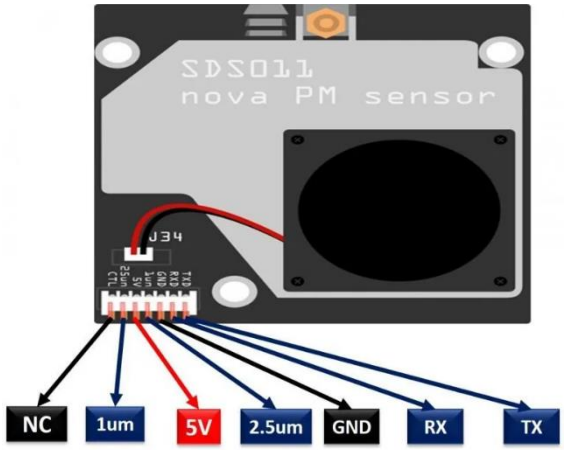
1. SDS 011:

SDS 011 is an air quality measurement sensor which can be used to get dust particles and smoke concentration in the air. More precisely, it can measure particulate matter (PM) concentrations in the air. It can detect the dust particles concentration between 0.3 to 10um. Most importantly, Nova PM dust sensor provides an interrupt-based response when the concentration of dust particles changes in the air

and the response time is less than 10 seconds. The operating voltage range is 4.7-5.3V which makes it suitable to use with standard voltage of 5 volts. Furthermore, it has a UART module and PWM outputs which can be used to get output from the SDS011 sensor.

Pinout:

The following picture shows the pinout diagram of the SDS011 dust particle sensor. It consists of 7 pins. But to interface it with microcontrollers such Arduino and ESP32, we can use only UART pins, such as RX, TX, to get dust concentration output from the dust sensor.



Pin Configuration Details:

Sr. No.	Pin Name	Function
1	NC	No connection/Not used.
2	1 µm	Range of PM2.5 value (0-999µg/m³) and provide output in PWM form on this pin.
3	5 V	Connect 5V to this pin to power dust sensor.
4	2.5 µm	Range of PM10 value (0-999µg/m³) and provide output in PWM form on this pin.
5	GND	Connect to ground terminal of power supply and common reference with Arduino or microcontroller
6	RX	Receiver terminal of UART module and used to receive commands from computer or microcontrollers.
7	TX	Transmitter pin of UART and used to transmit output to the target device such as Arduino or other microcontrollers.

SDS011 sensor can detect particulate matter (PM) in two ranges:

1. Ultrafine dust particles with PM range in 0 – 2.5 micrometres (µg/m³) and pin2 of SDS011 sensor provides PM2.5 output in PWM format.
2. Fine dust particles with PM range in 2.5 – 10 micrometres (µg/m³) and pin4 of SDS011 sensor provides PM10 output in PWM format.

Equations to calculate particulate matters:

- PM2.5 (µg / m³) = ((PM2.5 High byte *256) + PM2.5 low byte)/10
- PM10 (µg / m³) = ((PM10 high byte*256) + PM10 low byte)/10

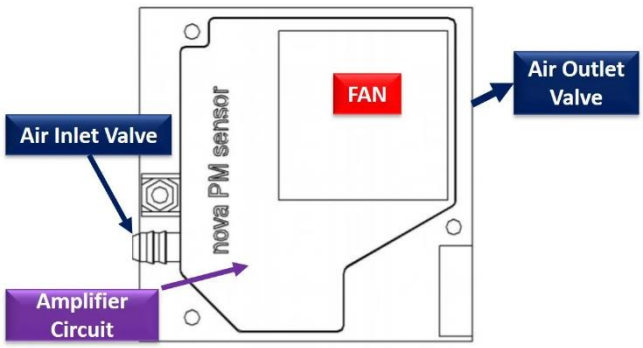
Features and Specifications:

- Particulate matter range: PM2.5 and PM10 (0.0-999.9 µg /m3)
- Operating voltage range: 4.7 5.3V
- Low current consumption: 70mA±10mA
- Sleep current consumption is less than 4mA
- Serial data output frequency: 1Hz
- Minimum dust particle detection resolution: 0.3 µm
- UART Interface with personal computers and microcontrollers
- Manual and periodic hibernation to receive dust sensor output
- Laser scattering working principle

Working:

Nova PM SDS011 air quality measurement sensor internally consists of following components:

Internal Fan
Laser Diode
Photo Diode
Air Inlet and Outlet Valves
Low Noise Amplifier circuit

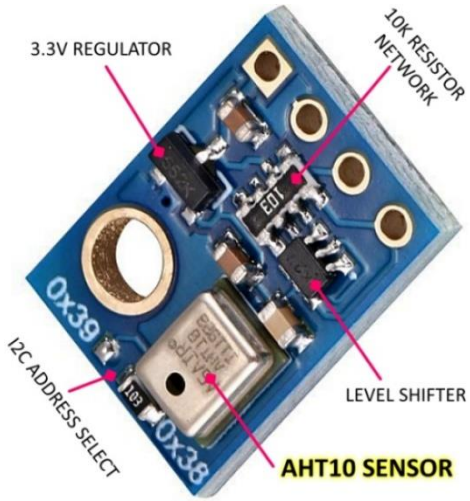


The Nova PM air quality measurement sensor has a built-in air circulation fan. This fan helps to circulate the air through the inlet valve to the output valve of the dust sensor. Moreover, it circulates air with a laser diode which is used to determine the size of dust particles in the air. When air enters the dust sensor through an inlet valve (as shown on the left-hand side of the SDS011 internal circuit) a laser diode illuminates the air and produces a scattered light. In response, the photodiode detects this scattered light and converts it into electrical signals. After that, these electrical signals are amplified through an amplification circuit and processed to get an amount of PM is in the air that is the number and diameter of dust particles.

AHT10 Sensor:

AHT10 is a digital temperature and humidity sensor embedded for reflow soldering. The dual-row flat leadless SMD package has a 4x5mm bottom and a height of 1.6mm.

The AHT10 sensor is equipped with a newly designed ASIC, an improved MEMS semiconductor capacitive humidity sensing element and a standard on-chip temperature sensing element. It can output a calibrated digital signal in standard I^2C format.



The minuscule sensor has a power supply range of 1.8-3.6V, but 3.3V is the recommended operating voltage.

Pinout:

1. Vin: Supply Voltage
2. Gnd: Ground
3. SCL: Serial Clock
   This pin is used for communication synchronization between the

microprocessor and AHT10 sensor. Since the interface contains completely static logic, there is no minimum SCL frequency.

4. SDA: Serial Data
   The SDA pin is used for data input and output of the sensor. SDA is active on the rising edge of the serial clock (SCL) when a command is sent to the sensor, and SDA must remain stable when SCL is high. After the falling edge of SCL, the SDA value can be changed.

Features and Specifications:

1. Interface type: I2C
2. Working voltage: 1.8 – 6.0 V
3. Interface size: 4*2.54mm pitch
4. Humidity accuracy: typical ± 2%
5. Humidity resolution: 0.024%
6. Temperature accuracy: typical ± 0.3 ° C
7. Temperature resolution: Typical 0.01 °C
8. Working temperature: -40°C–85°C

Procedure:

A) Interfacing of SDS sensor and AHT10 sensor with ESP8266:

The Fig. 1 below shows the circuit connection for interfacing of SDS011 and AHT10 sensors with ESP8266. Software setup remains as usual, and we will write a program without using the built-in library to get a clear understating of its functionality.

Connections:

SDS011	ESP8266
TX	D6
RX	D5
5 V	Vin
Gnd	Gnd

ESP8266	AHT10
3V3	Vin
Gnd	Gnd
D1	SCL
D2	SDA

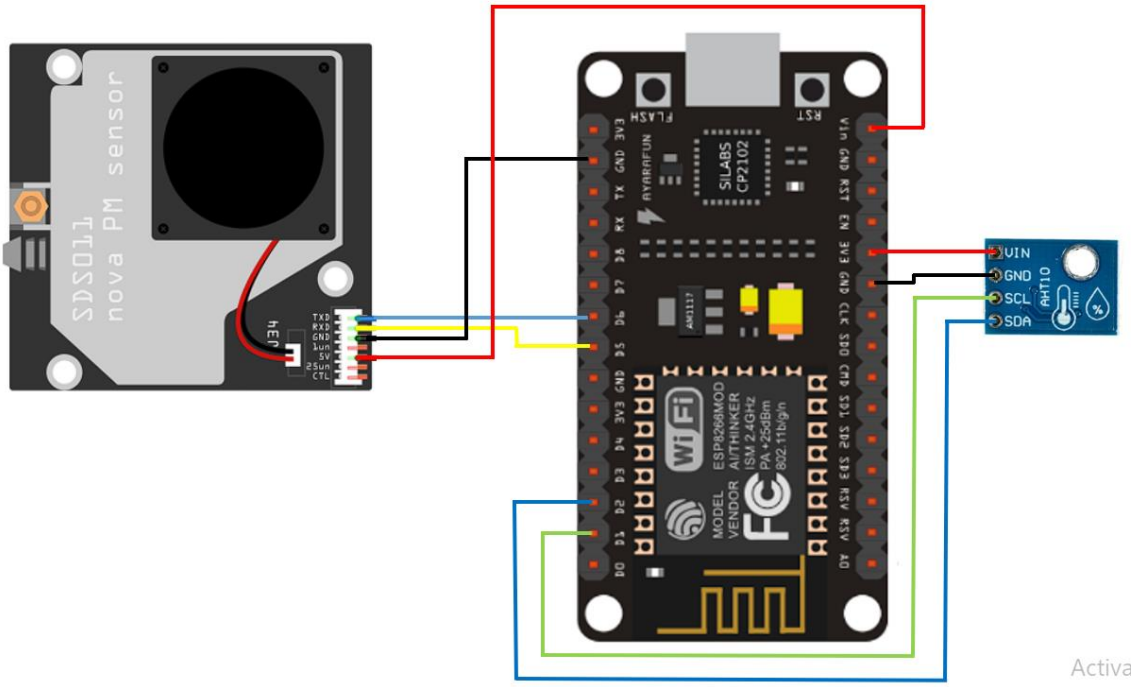


Figure 1: Interfacing of SDS and AHT sensors with ES8266

Steps for interfacing AHT10:

1. Open Arduino IDE
2. Go to file menu on top left corner of Arduino ide
3. Go to preferences and enter the url link for the esp8266 board.
4. url_link: http://arduino.esp8266.com/stable/package_esp8266com_index.json
5. Click on OK.
6. Now open tools -> board -> board manager and type esp8266.
7. After that press on install.
8. Once installed close the ide and reopen it.
9. Then go to tools -> board-> esp8266 -> Nodemcu 1.0. (ESP 12E module).
10. Go to sketch -> include library -> manage libraries -> type AdafruitAHTX0, after selecting the library click on install.
11. Follow step 10 to install Adafruit BUSIO library.
12. Write the code -> Go to tools -> Port -> Select the USB connected port.
13. Compile and upload the code.

Code for AHT10:

```
#include <Adafruit_AHTX0.h>

Adafruit_AHTX0 aht;

void setup() {
```

```
 Serial.begin(115200);
 Serial.println("Adafruit AHT10/AHT20 demo!");

 if (! aht.begin()) {
  Serial.println("Could not find AHT? Check wiring");
  while (1) delay(10);
 }
 Serial.println("AHT10 or AHT20 found");
}
void loop() {
 sensors_event_t humidity, temp;
 aht.getEvent(&humidity, &temp);// populate temp and humidity objects with fresh data
 Serial.print("Temperature: "); Serial.print(temp.temperature); Serial.println(" degrees C");
 Serial.print("Humidity: "); Serial.print(humidity.relative_humidity); Serial.println("% rH");
delay(500);
}
```

Steps for interfacing SDS011:

1. Go to sketch -> include library -> manage libraries -> type ESP8266 WIFI, after selecting the library click on install.
2. Follow step 1 to install SDS011.h (by R.Zchiegner) library.
3. Write the code -> Go to tools -> Port -> Select the USB connected port
4. Compile and upload the code.
5. Connect RX and TX pins of SDS sensor to D5 and D6 pins of ESP8266 respevctively.

Code for SDS011:

```
// SDS011 dust sensor example
// ----------------------------
//
// By R. Zschiegner (rz@madavi.de).
// April 2016
```

```
#include <SDS011.h>

float p10,p25;
int error;

SDS011 my_sds;

void setup() {
	my_sds.begin(D1,D2);
	Serial.begin(9600);
}

void loop() {
	error = my_sds.read(&p25,&p10);
	if (! error) {
		Serial.println("P2.5: "+String(p25));
		Serial.println("P10:  "+String(p10));
	}
	delay(100);
}
```