DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE116 2018-2019

Air quality detector

(Open Group Report)

Student Name	:	Mohan.Sun
Student ID	:	1717613
Student Name	:	Jingyu.Lyu
Student ID	:	1717610
Group		A25

Contribution form -- Smart car

Group__A25:

Name	Contribution (overall)	Circuit Building (100%)	Report (100%)	Arduino programing (100%)	Comments
Mohan.Sun	50%	50%	50%	50%	For example: Lead the whole project and contribute in Arduino coding and report writing
Jingyu.Lyu	50%	50%	50%	50%	For example: Work closely with the first group member, contribute in the report writing and poster design.

1. Abstract

Currently, sustainability is one of the most concerned subjects around the world. It is defined as the ability which develops in continues. The main idea is sustainable development, which meets the needs of the present without compromising the ability of future generations to meet their own needs [1]. Therefore, sustainable development requires what benefits to the living environments. In China, industry development has witnessed not only great achievements, but the gradually deteriorated living environment. Therefore, we decided to build a detector to detect air condition to help users to get the detailed information to judge if it is not suitable to do outdoor activities. This report will introduce the processes of building a Air quality detector based on Arduino and show the results of the finished work.

2. Introduction

As it is confirmed by US government, PM2.5 not only is damaged to the environment, but also is damaged to health [2]. Although this project, Air quality detector, does not work for the reduction, it provides PM2.5 information used to warn users to determine whether it is suitable or not to do outdoor activities. Due to this device is manufactured based on Arduino, it is necessary to introduce these basic components. This detector was designed with five main components: Arduino, PM2.5 sensor, OLED Screen 0.96 inch, Bluetooth (HC-06) and Temperature and Humidity (DTH11)

2.1 Arduino Nano

In this project, the most significant device is the Arduino Nano development board which likes the brain of human. There are 14 digital pins and 8 analog input pins on the Nano. Each of the 14 digital pins can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. In addition, some pins

have specialized functions. The 0(RX) and 1(TX) pins are used to receive and transmit TTL serial data. The 2 and 3 pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. The 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. What's more, There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. For the 8 analog inputs, each of them provides 10 bits of resolution (i.e. 1024 different values). The special pins are A4(SDA) and A5(SCL). They Support I2C (TWI) communication using the Wire library.

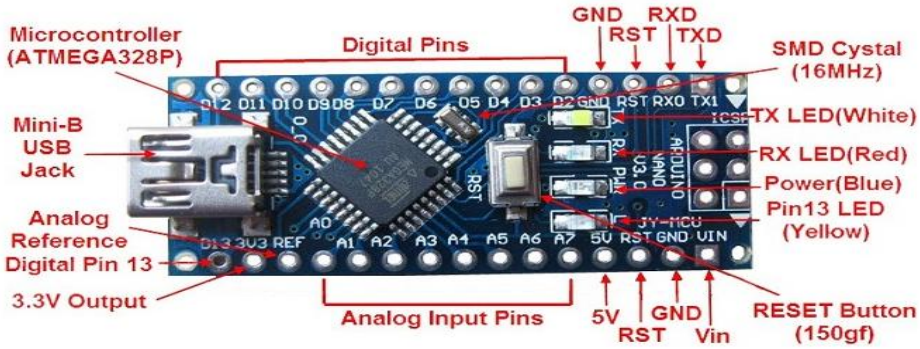

Figure 1: Arduino Nano

2.2 PM2.5 sensor

For the PM2.5 sensor, the dust concentration is determined by the height of the output pulse. The Figure 2 shows the relationship between dust density and output voltage.

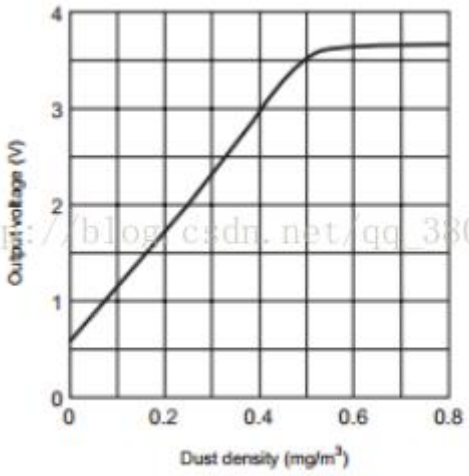

Figure 2: Relationship between dust density and output voltage

According to the Figure 2, we can find that when it reaches about 0.6mg/m3, the curve will no longer grow. At this time, the maximum output voltage is about 3.6v. In this case, we obtain a formula: Dust density = 0.17 * Output Voltage.

3. Design procedure

This section is going to discuss the working principles of the Air Quality Detector and how these components are connected. In addition, the programming procedure will be analyzed.

3.1 Working Principles

For the Air Quality Detector, the temperature and humidity sensor (DHT11) and PM2.5 sensor are used to measure the air temperature, humidity and PM2.5 density. After the data is measured, it will be transferred to the Arduino Nano. Then the data will be calculated through a compiled program on Nano. Finally, the information will be printed on the OLED screen. The flow chart shows the process for this project in Figure 3.

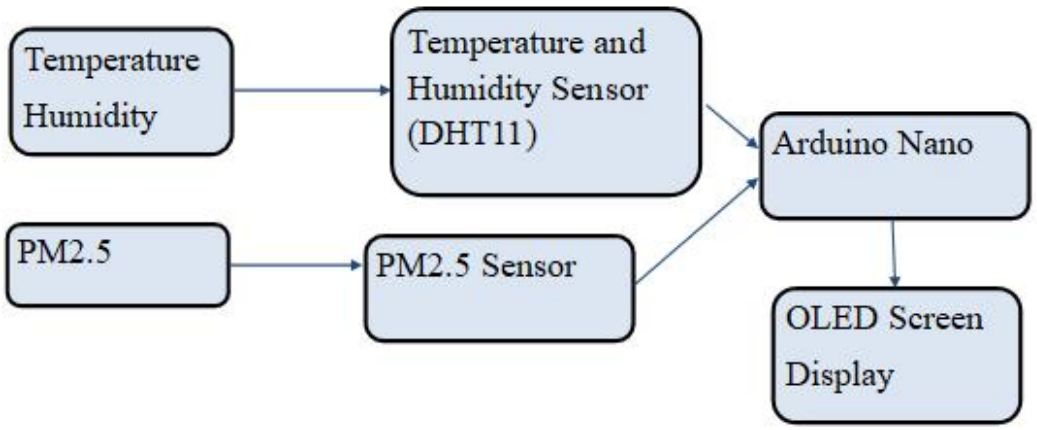

Figure 3. The design flow chart of the Air Quality Detector.

3.2 Connection Procedure

Firstly, connect the Arduino Nano to the power supply and ground on breadboard. Then connect the temperature and humidity sensor (DHT11) to Nano according to Figure 4.

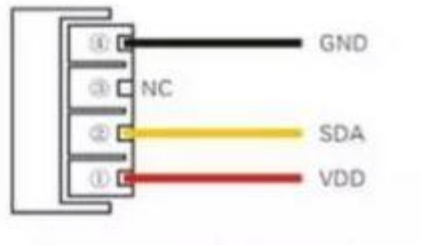

Figure 4: DHT11 pins.

Import the test code and open the serial monitor to see if the temperature and humidity can be measured. We obtain:

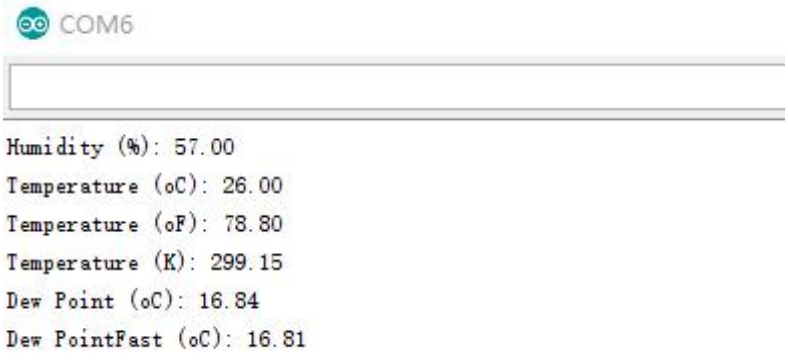

Figure 5: Measured temperature and humidity.

Next, connect the OLED screen to Nano and import the test code. We obtain:

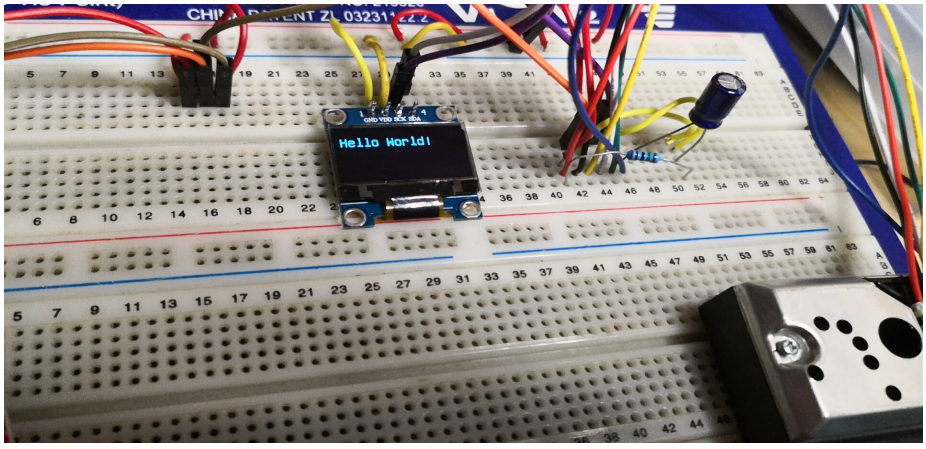

Figure 6: Test of OLED screen.

Finally, connect the PM2.5 sensor to Nano and import the test code. We obtain:

Figure 7: Measured PM2.5 density.

The connection is finished. Then import the program into Arduino Nano, there information of air quality will be printed on the screen. Figure 8 shows the Air Quality Detector.

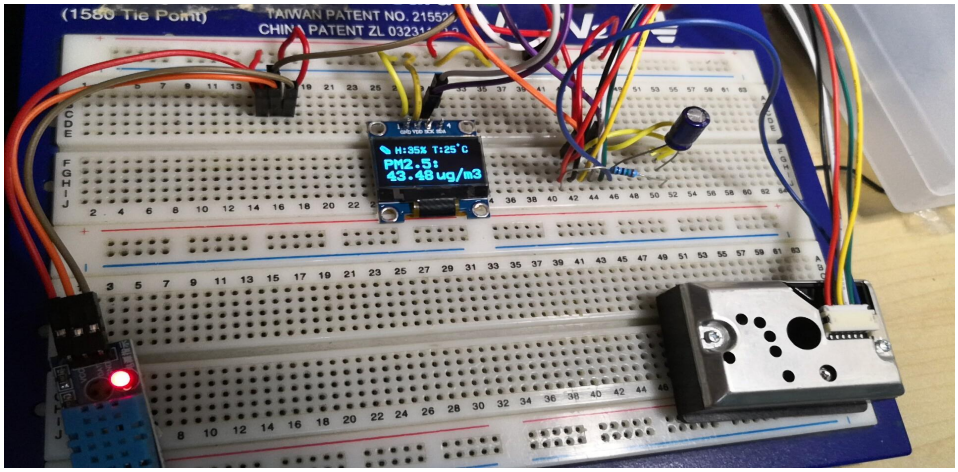

Figure 8. Air Quality Detector

3.3 Code

```
#include "U8glib.h"
#include <SoftwareSerial.h>
#include <dht11.h>
#define DHT11PIN A0
#define measurePin A1//输出引脚连接模拟口A1
#define ledPin 5 //LED引脚连接数字口5
dht11 dht;
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NO_ACK);  // Display which does not send AC
float h, t, hic;
void setup() {
   pinMode(ledPin, OUTPUT);
   pinMode(measurePin, INPUT);
   Serial.begin(9600);
if ( u8g.getMode() == U8G_MODE_R3G3B2 ) {
u8g.setColorIndex(255); // white
}
else if ( u8g.getMode() == U8G_MODE_GRAY2BIT ) {
u8g.setColorIndex(3); // max intensity
}
else if ( u8g.getMode() == U8G_MODE_BW ) {
u8g.setColorIndex(1); // pixel on
}
else if ( u8g.getMode() == U8G_MODE_HICOLOR ) {
u8g.setHiColorByRGB(255,255,255);
}
delay(2000);
}
int i=0;
void loop() {
u8g.firstPage();
pmRead();
if(i>=5) i=0;else i++;
dhtRead();
do {
draw();
} while( u8g.nextPage() );
delay(500);
}
unsigned int samplingTime = 280;
unsigned int deltaTime = 40;
unsigned int sleepTime = 9680;
```

```
float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;
void pmRead(){//get the PM2.5 densigy
 digitalWrite(ledPin, LOW);
 delayMicroseconds(samplingTime);
 voMeasured = analogRead(measurePin);
 delayMicroseconds(deltaTime);
 digitalWrite(ledPin, HIGH);
 delayMicroseconds(sleepTime);
 calcVoltage = voMeasured * (5.0 / 1024);
 dustDensity = 5000*calcVoltage/29 - 3000/29;
 if (dustDensity < 0) { dustDensity = 0.00;}
 Serial.println(dustDensity);
 delay(2 * 1000);
}
void dhtRead(){//get the temperature and humidity
// Reading temperature or humidity takes about 250 milliseconds!
// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
dht.read(DHT11PIN);
h = dht.humidity;// Read temperature as Celsius (the default)
t = dht.temperature;// Compute heat index in Celsius (isFahreheit = false)
}
void draw(void) {
// graphic commands to redraw the complete screen should be placed here
char sendBuff[20];
u8g.setFont(u8g_font_unifont_78_79);
sprintf(sendBuff,"%c",14);
u8g.drawStr( 0, 18, sendBuff);
u8g.setFont(u8g_font_unifont);
sprintf(sendBuff,"H:%2d%% T:%2d%cC",(int)h,(int)t,0xB0);
u8g.drawStr( 20, 18, sendBuff);
u8g.setFont(u8g_font_profont22);
sprintf(sendBuff,"PM2.5:",dustDensity);
u8g.drawStr( 5, 42, sendBuff);
u8g.setPrintPos(6,60);
u8g.print(dustDensity);
u8g.setPrintPos(70,60);
u8g.print("ug/m3");
}
```

4. Discussion and analysis

During the construction of the Air Quality Detector, there were amount of difficulties. At first, our group was not familiar with the Arduino Nano. In this case, the connection of pins

Bothered us for a long time. In addition, part of components' model were not specified which led to wrong connection. There is also limitation in the PM2.5 sensor that it can only measure particle density within a certain rang instead of PM2.5 density. As a result, the measured PM2.5 density was not accurate. What's more, due to the insufficient memory space of the Arduino Nano, the design of Bluetooth module was not completed.

5. Conclusion

In conclusion, due to it seems that air quality becomes more serious than ever before, especially smog weather happens more frequently, it is necessary for citizens to be prepared to handle this situation. Therefore, we designed and built a air quality detector to enable to collect information. This detector was consisted of five main digital devices, which were Arduino Nano, PM2.5 sensor, OLED Screen, Bluetooth (HC-06) and Temperature and Humidity sensor (DTH11). The program was designed and uploaded into the chip, and the sensors collected the information and sent it to the chip through pins, then the chip transferred it to signals and sent them to OLED screen. It works perfectly. However, there was a little shortage, the detector does not have a switch. For now, it works once the power is provided. And the Bluetooth module was not installed in this device. In addition, the accuracy of data measurement needs to be improved. Therefore, for the further study, we considered to designing a switch to control the power and design Bluetooth module for the air quality detector. It is also helpful method to apply more precise components in this detector. In conclusion, in this project, we study further about Arduino, and we learn better about the cooperation between the chip, displays and sensors, and how to transfer the information to the signals and how they propagate in the circuits. It is quite important to electric engineers.

References

[1] World Commission on Environment and Development (WCED). Our common future. Oxford:Oxford University Press, 1987 p.43.

[2] (July. 20, 2018) *Health and Environmental Effects of Particulate Matter (PM)* [Online]. Available: https://www.epa.gov/pm-pollution/health-and-environmental-effects-particulate-matter-pm