# Concept

Sniffer has 2 parts:

* Sniffer is installed outdoor, measures air quality data
* Hub receives data from Sniffer, displays on its screen and send data to Sniffer’s network

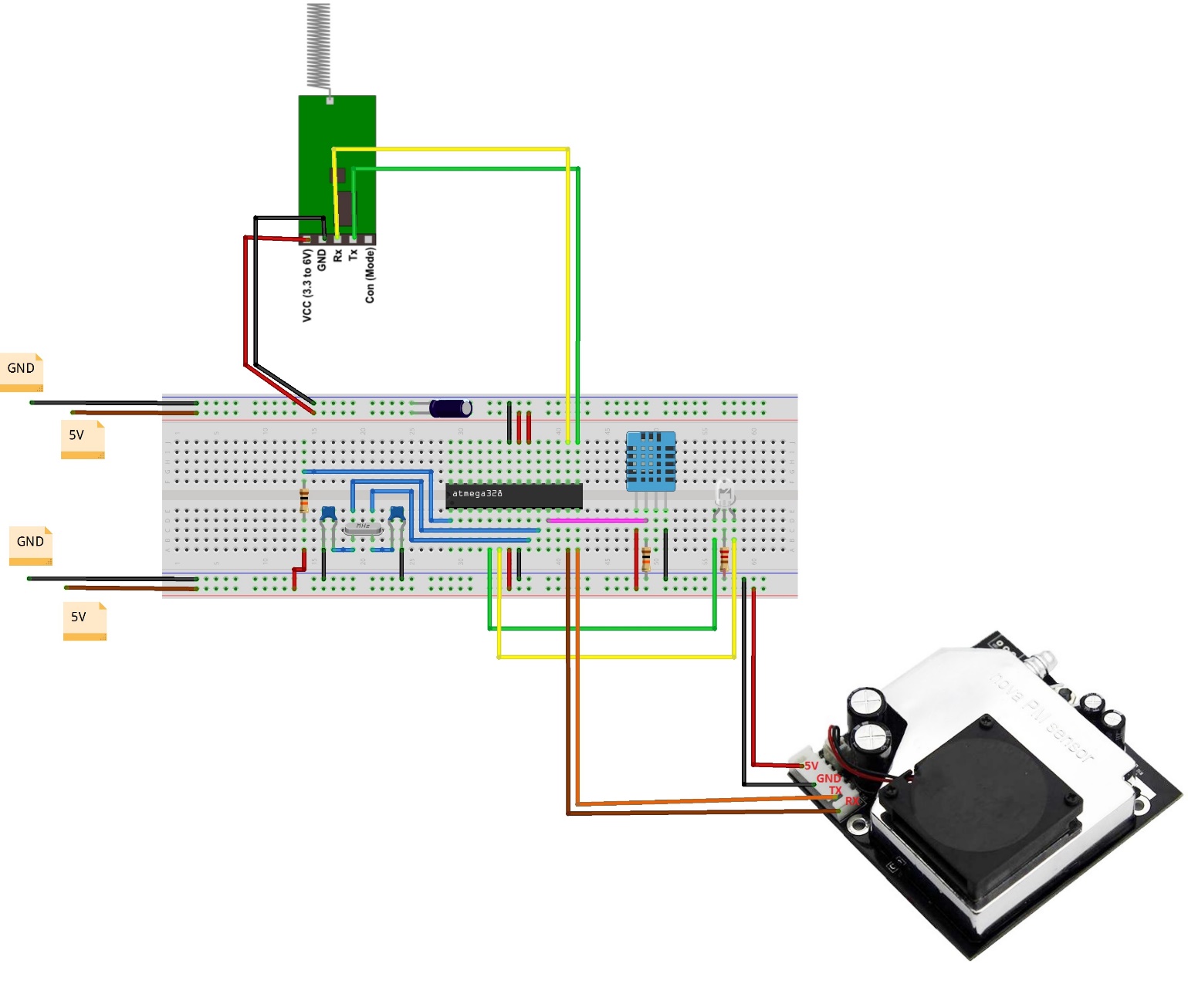
# Materials

1. Sniffer
   1. Arduino Uno or Atmel328P chip
   2. Breadboard
   3. DHT11 or DHT22 sensor
   4. Nova fitness SDS011 sensor
   5. Breadboard cables
   6. NRF cc1001
2. Hub
   1. ESP8266 – v12
   2. LCD 16x2
   3. I2C LCD module
   4. Breadboard
   5. Breadboard cables

# Instruction

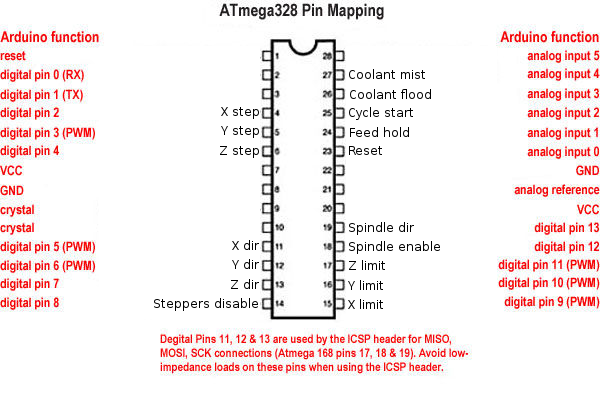
## Sniffer

Please find below the diagram showing how to connect parts to create Sniffer and each steps to make the circuit.



* 1. **Step 1: Assembling Atmel328P**

Our long-term solution is to reduce the power consumption for Sniffer. Therefore, we use Atmel328P chip to alter Arduino Uno which will consume a lot of energy if you intend to use battery.

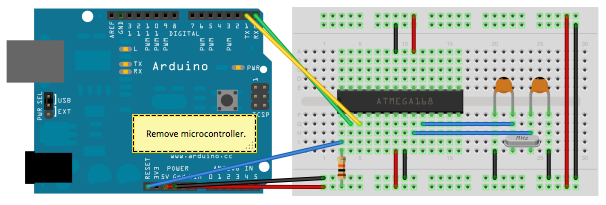


Atmel328P has the same functionality as an Arduino. However, you have to assemble the circuit and make it work as a real Arduino board.

Follow this construction to setup Atmel328P as a standalone microcontroller.

<https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard>

After assembling the circuit, you can program and upload code in Arduino IDE as usual.

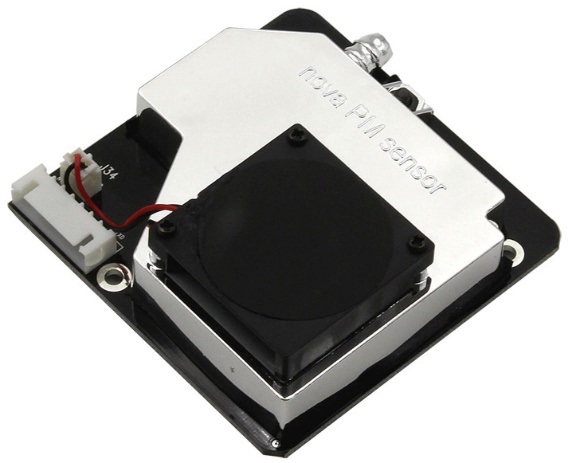


If you want to deploy your Sniffer in the environment, you can remove Arduino board which is only used for development.

* 1. **Step 2: Adding your sensors**

Our Sniffer uses 2 kind of sensors to measure air quality data

* + - Nova fitness SDS011 sensor: to measure dust level in the environment: PM 2.5 and PM 10
    - DHT11 sensor: to measure temperature and humidity



Next, assemble the sensors to Atmel

Nova Sensor ------------ Arduino/Atmel

**VCC** ------------------ **VCC 5V**

**GND** ----------------- **GND**

**TX** ------------------ **digital pin 7 (leg 13 on Atmel)**

**RX** ---------------- **digital pin 8 (leg 14 on Atmel)**

DHT 11 ------------ Arduino/Atmel

**VCC** ------------------ **VCC 5V**

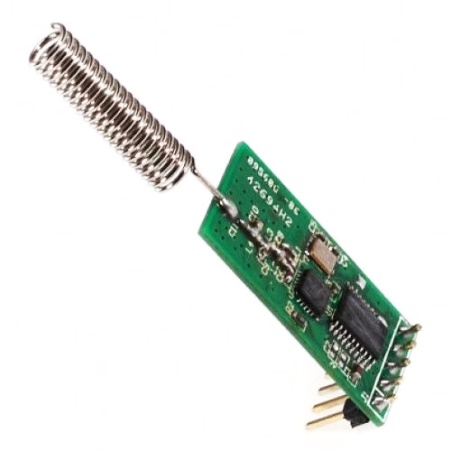
**GND** ----------------- **GND**

**DATA** ----------------**digital pin 5 (leg 11 on Atmel)**

* 1. **Step 3: Adding transmitter device to send air data to Hub**

We use RF module CC1101 to transmit air data: PM2.5, PM 10, temperature and humidity from Sniffer to Hub

The transmitted distance is around 20 meters through the obstacle. We already tested the case that Sniffer was put on 4th floor and the Hub at the ground floor then it works perfectly.



CC1101 ------------ Arduino/Atmel

**VCC** ------------------ **VCC 5V**

**GND** ----------------- **GND**

**TX** ----------------**digital pin 9 (leg 15 on Atmel)**

**RX** ----------------**digital pin 10 (leg 16 on Atmel)**

## Hub

How to use I2C module with LCD display

Connect I2C **GND** to power source **GND**, **VCC** to **5V**

Please note that LCD use 5V but your ESP only running on 3.3V. Hence, you should provide different voltages for them.

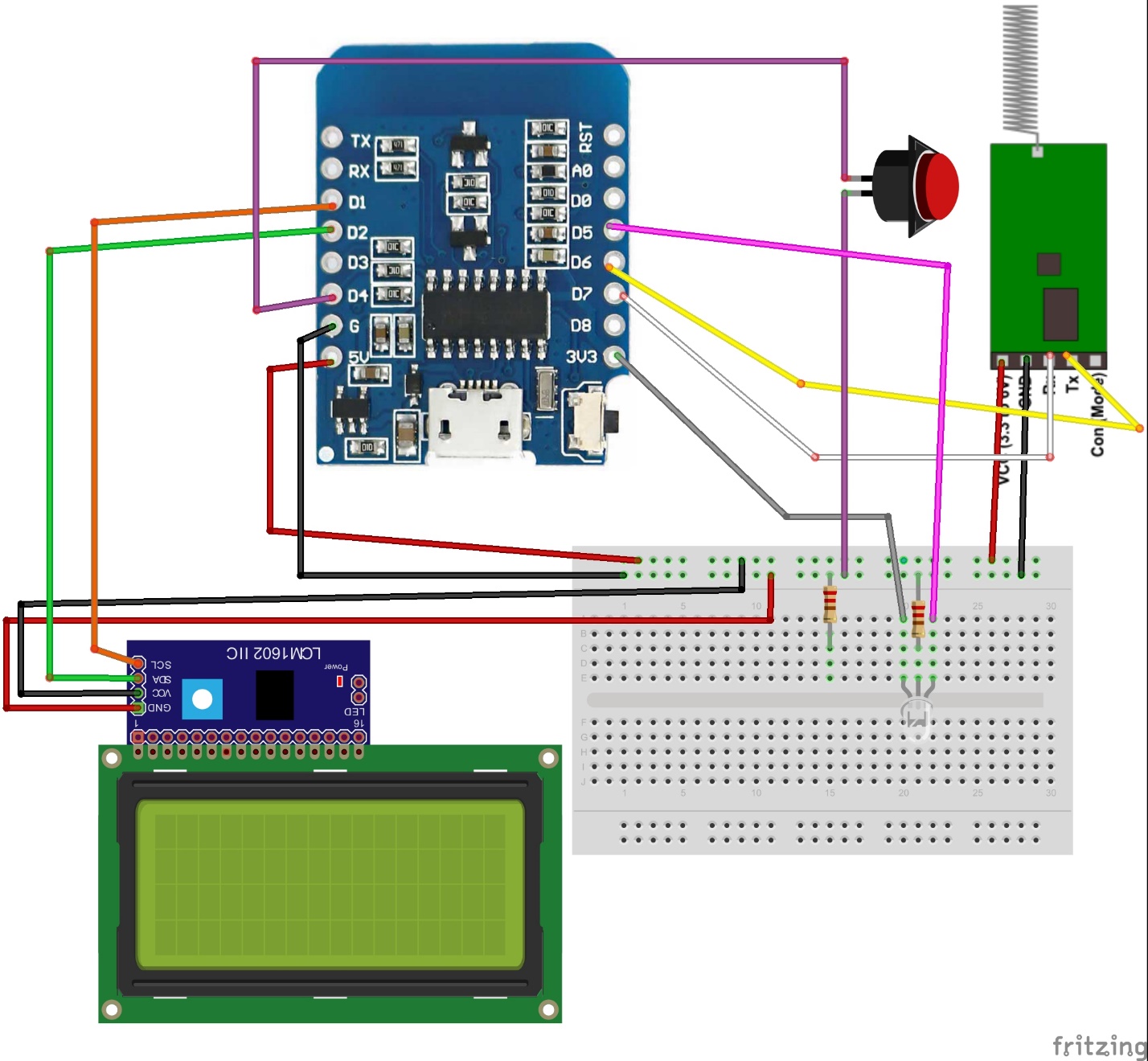
Pin mapping between I2C module and ESP

I2C ------------ ESP

**SDA** ------------------ **GPIO-04**

**SCL** ----------------- **GPIO-05**

Please find below the diagram showing how to connect parts to create Hub



### Test and Calibrate your LCD

Use the LiquidCrystal\_I2C library, upload the sample code from Files\Examples\LiquidCrystal\_I2C\HelloWorld. If you can see the LCD displaying Hello World, then congratulation, you do not need to calibrate it.

If you see the LCD background glowing but cannot see any character like in the following picture (TODO: add picture), then you can calibrate your LCD using a small screw driver, change the blue knob (on the I2C module) clock-wise or anti-clockwise until you can see the words on screen.

### How to upload code to ESP8266

If you have the ESP8266 v12 mini version D1 like us then all you have to do is plugin the USB cable, select the correct COM port, then you can start uploading the code.

If you only have the version that comes with bare pins like we used to have (<https://www.google.com.vn/#safe=strict&q=esp8266+v12+yellow>). The you can use the following methods to upload code:

There are two ways that we used to connect ESP8266 to PC for uploading code:

1. Using CP2101 module
2. Via Arduino Uno

Please find below how to put in ESP in upload mode

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ESP8266 | CP2102 |  |  | ESP8266 | Arduino |
| TX | RX |  |  | TX | TX |
| RX | TX |  |  | RX | RX |
| VCC | 3.3V |  |  | VCC | 3.3V |
| CH\_PD | 3.3V |  |  | CH\_PD | 3.3V |
| GND | GND |  |  | GND | GND |
| GPIO15 | GND |  |  | GPIO15 | GND |
| GPIO00 | GND |  |  | GPIO00 | GND |
|  |  |  |  |  | RESET - GND |
|  |  |  |  |

Note: While developing, there is need to upload code many times. Before re-upload code, un-plug then re-plug ESP88266’s VCC to re-enable upload mode.