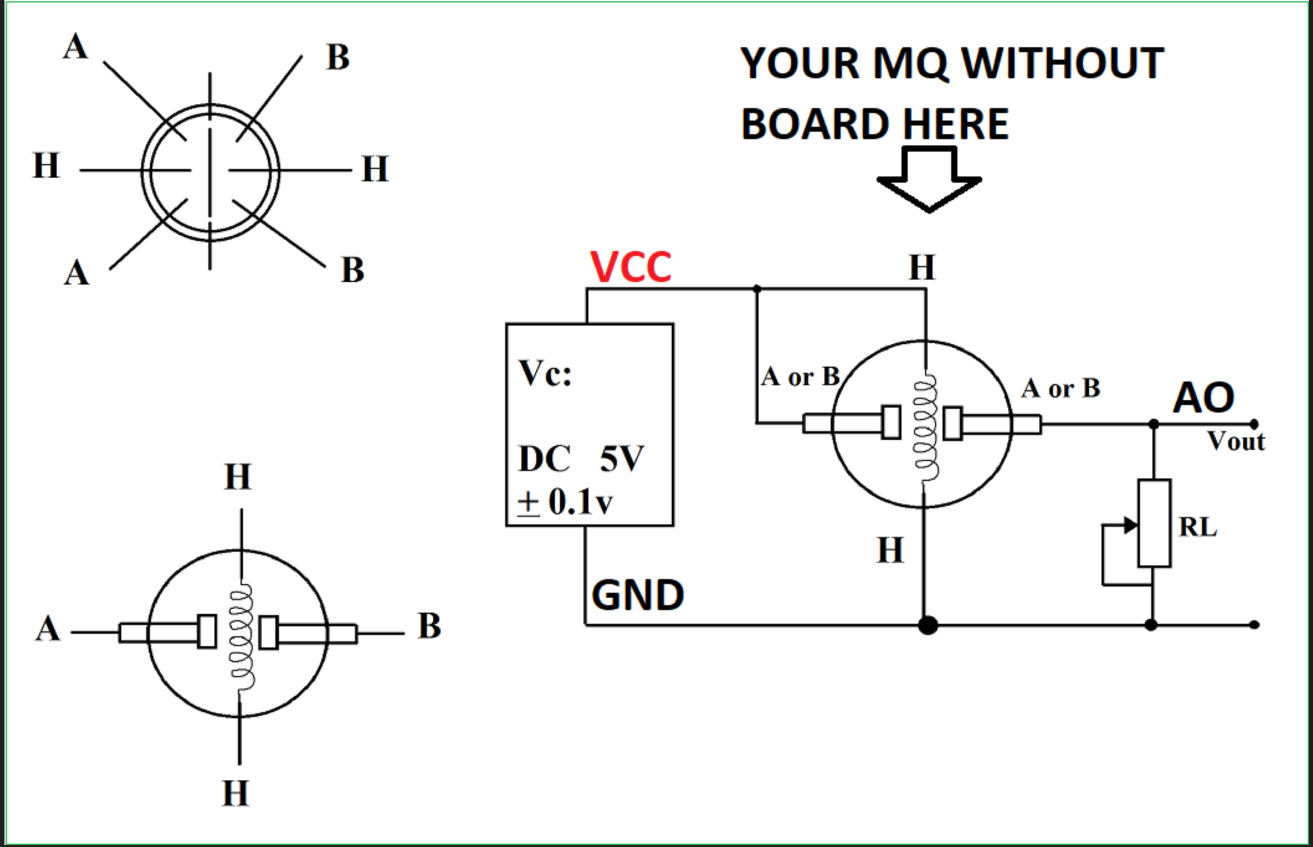
MQSensorLib v2.0

03/2020 By Miguel Califa – Yersson Carrillo – Ghiordy Contreras

New Features

- Implemented support for A2D external converters and ESP8266.
- (Example) Support to digital input, alarm status.
- Added calibration algorithm on the setup of all examples.
- Divided the calculation method on two methos (Linear – Exponential).
- Fixed words on spanish, traslated to english (Inicializar -> Init)

MQSensor Connection (Global)

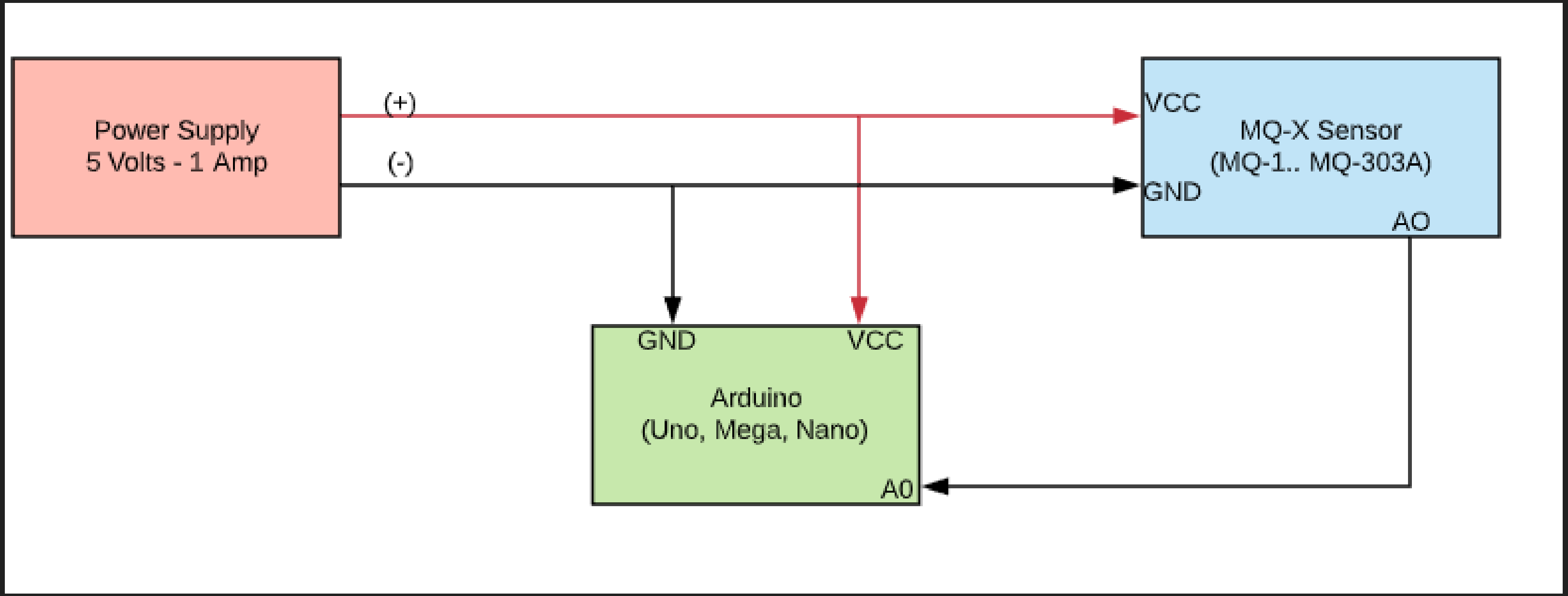


If you have an MQ into a board:
1. Identify whats the value of your RL.
2. Connect to your Arduino following next steps.

Or if you only have a sensor:

1. Build your circuit.
2. Identify your VCC, GND, AO.
3. Connect to your Arduino following next steps.

MQSensor and Arduino board (Wiring)



MQSensor and Arduino board (Program)

```
#include <MQUnifiedsensor.h>
#define         Board                   ("Arduino UNO")
#define         Pin                 (A3)  //Analog input 3 of your arduino
#define         Type                  ("MQ-3") //MQ3
#define         Voltage_Resolution      (5)
#define         ADC_Bit_Resolution      (10) // For arduino UNO/MEGA/NANO
#define         RatioMQ3CleanAir        (60) //RS / R0 = 60 ppm
MQUnifiedsensor MQ3(Board, Voltage_Resolution, ADC_Bit_Resolution, Pin, Type);
void setup() {
        Serial.begin(9600);
        MQ3.init();
        MQ3.setRegressionMethod(1); //_PPM =  a*ratio^b
        MQ3.setA(4.8387); MQ3.setB(-2.68); //Benzene
```

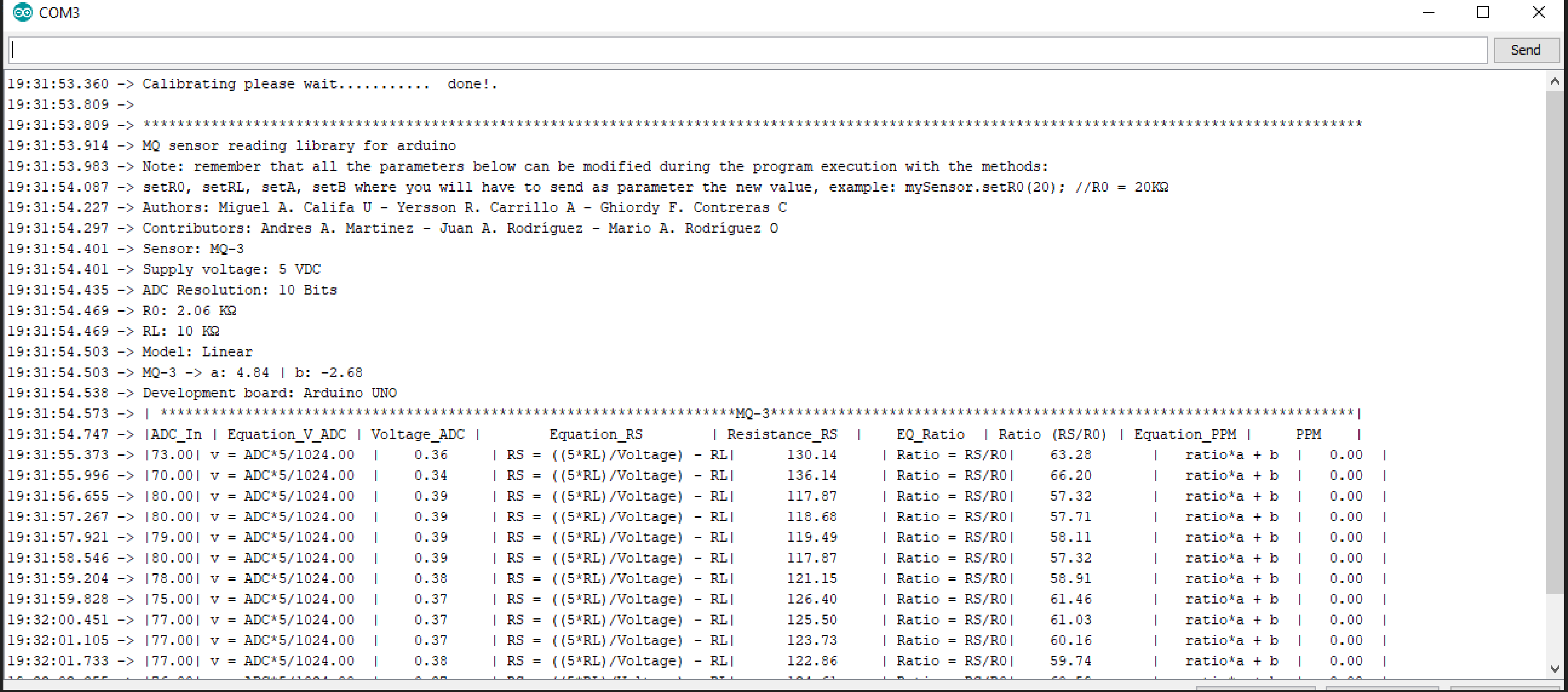
MQSensor and Arduino board (Program)

```
    float calcR0 = 0;
     for(int i = 1; i<=10; i ++)
     {
      MQ3.update();
      calcR0 += MQ3.calibrate(RatioMQ3CleanAir);
      Serial.print(".");
     }
     MQ3.setR0(calcR0/10);
     Serial.println("  done!.");
}
```

MQSensor and Arduino board (Program)

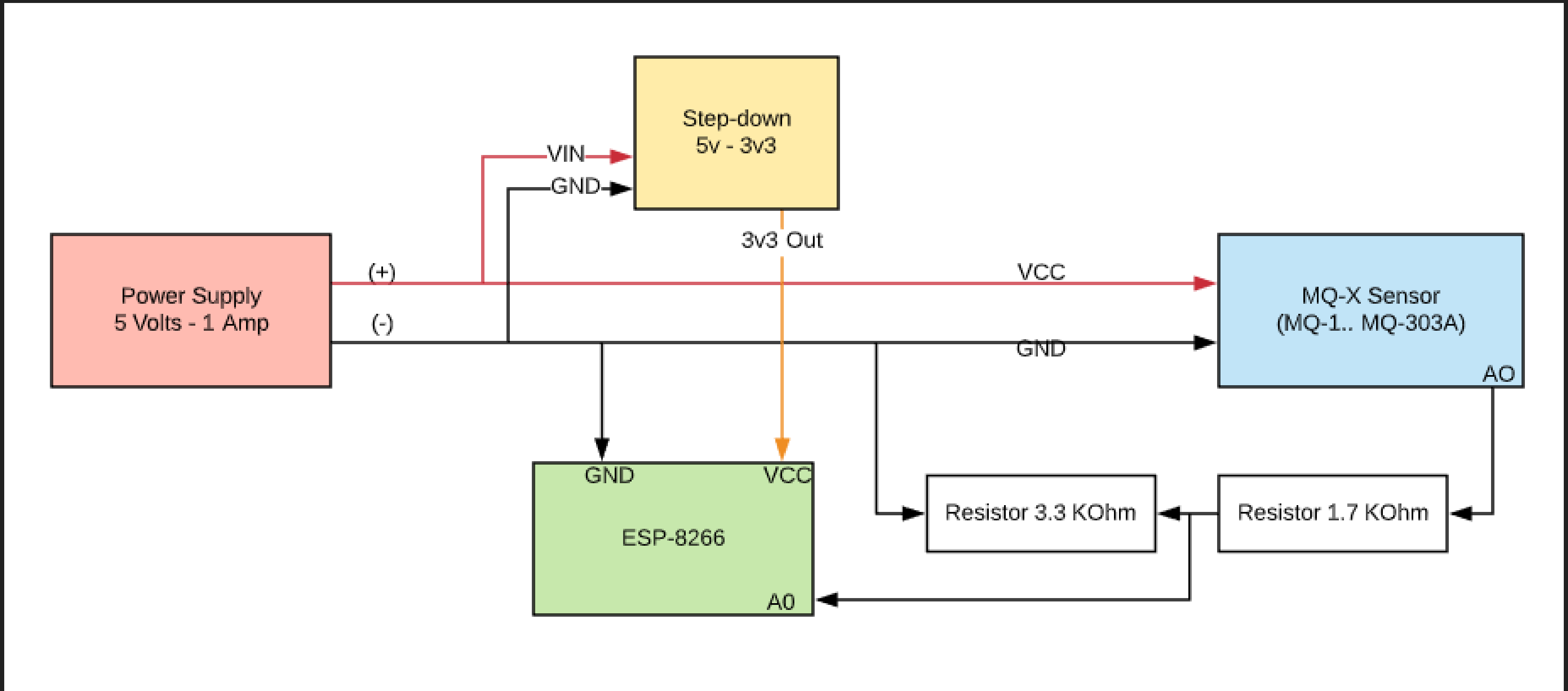
```
void loop() {
  MQ3.update(); // Update data, the arduino will be read the voltage on the analog pin
  MQ3.readSensor(); // Sensor will read PPM concentration using the model and a and b
values setted before or in the setup
  MQ3.serialDebug(); // Will print the table on the serial port
  delay(500); //Sampling frequency
}
```

MQSensor and Arduino board (Working)



```
19:31:53.360 -> Calibrating please wait...........  done!.
19:31:53.809 -> 
19:31:53.809 -> ************************************************************************************************************************************************
19:31:53.914 -> MQ sensor reading library for arduino
19:31:53.983 -> Note: remember that all the parameters below can be modified during the program execution with the methods:
19:31:54.087 -> setR0, setRL, setA, setB where you will have to send as parameter the new value, example: mySensor.setR0(20); //R0 = 20KΩ
19:31:54.227 -> Authors: Miguel A. Califa U - Yersson R. Carrillo A - Ghiordy F. Contreras C
19:31:54.297 -> Contributors: Andres A. Martinez - Juan A. Rodríguez - Mario A. Rodríguez O
19:31:54.401 -> Sensor: MQ-3
19:31:54.401 -> Supply voltage: 5 VDC
19:31:54.435 -> ADC Resolution: 10 Bits
19:31:54.469 -> R0: 2.06 KΩ
19:31:54.469 -> RL: 10 KΩ
19:31:54.503 -> Model: Linear
19:31:54.503 -> MQ-3 -> a: 4.84 | b: -2.68
19:31:54.538 -> Development board: Arduino UNO
19:31:54.573 -> | ********************************************************************MQ-3*********************************************************************|
19:31:54.747 -> |ADC_In | Equation_V_ADC | Voltage_ADC |        Equation_RS        | Resistance_RS  |    EQ_Ratio   | Ratio (RS/R0) | Equation_PPM |     PPM    |
19:31:55.373 -> |73.00| v = ADC*5/1024.00  |     0.36     | RS = ((5*RL)/Voltage) - RL|     130.14     | Ratio = RS/R0|    63.28       |   ratio*a + b  |   0.00   |
19:31:55.996 -> |70.00| v = ADC*5/1024.00  |     0.34     | RS = ((5*RL)/Voltage) - RL|     136.14     | Ratio = RS/R0|    66.20       |   ratio*a + b  |   0.00   |
19:31:56.655 -> |80.00| v = ADC*5/1024.00  |     0.39     | RS = ((5*RL)/Voltage) - RL|     117.87     | Ratio = RS/R0|    57.32       |   ratio*a + b  |   0.00   |
19:31:57.267 -> |80.00| v = ADC*5/1024.00  |     0.39     | RS = ((5*RL)/Voltage) - RL|     118.68     | Ratio = RS/R0|    57.71       |   ratio*a + b  |   0.00   |
19:31:57.921 -> |79.00| v = ADC*5/1024.00  |     0.39     | RS = ((5*RL)/Voltage) - RL|     119.49     | Ratio = RS/R0|    58.11       |   ratio*a + b  |   0.00   |
19:31:58.546 -> |80.00| v = ADC*5/1024.00  |     0.39     | RS = ((5*RL)/Voltage) - RL|     117.87     | Ratio = RS/R0|    57.32       |   ratio*a + b  |   0.00   |
19:31:59.204 -> |78.00| v = ADC*5/1024.00  |     0.38     | RS = ((5*RL)/Voltage) - RL|     121.15     | Ratio = RS/R0|    58.91       |   ratio*a + b  |   0.00   |
19:31:59.828 -> |75.00| v = ADC*5/1024.00  |     0.37     | RS = ((5*RL)/Voltage) - RL|     126.40     | Ratio = RS/R0|    61.46       |   ratio*a + b  |   0.00   |
19:32:00.451 -> |77.00| v = ADC*5/1024.00  |     0.37     | RS = ((5*RL)/Voltage) - RL|     125.50     | Ratio = RS/R0|    61.03       |   ratio*a + b  |   0.00   |
19:32:01.105 -> |77.00| v = ADC*5/1024.00  |     0.37     | RS = ((5*RL)/Voltage) - RL|     123.73     | Ratio = RS/R0|    60.16       |   ratio*a + b  |   0.00   |
19:32:01.733 -> |77.00| v = ADC*5/1024.00  |     0.38     | RS = ((5*RL)/Voltage) - RL|     122.86     | Ratio = RS/R0|    59.74       |   ratio*a + b  |   0.00   |
```

MQSensor and ESP8266 (Wiring)



MQSensor and ESP8266 (Program)

```
//Include the library
#include <MQUnifiedsensor.h>
#define         Board                   ("Arduino UNO")
#define         Pin                     (A3)  //Analog input 3 of your Arduino
#define         Type                    ("MQ-3") //MQ3
#define         Voltage_Resolution      (3.3)
#define         ADC_Bit_Resolution      (10) // For arduino UNO/MEGA/NANO
MQUnifiedsensor MQ3(Board, Voltage_Resolution, ADC_Bit_Resolution, Pin, Type);
void setup() {
      Serial.begin(9600); //Init serial port
      MQ3.setRegressionMethod(1); //_PPM =  a*ratio^b
      MQ3.setA(4.8387); MQ3.setB(-2.68); //Benzene
       MQ3.init();
```

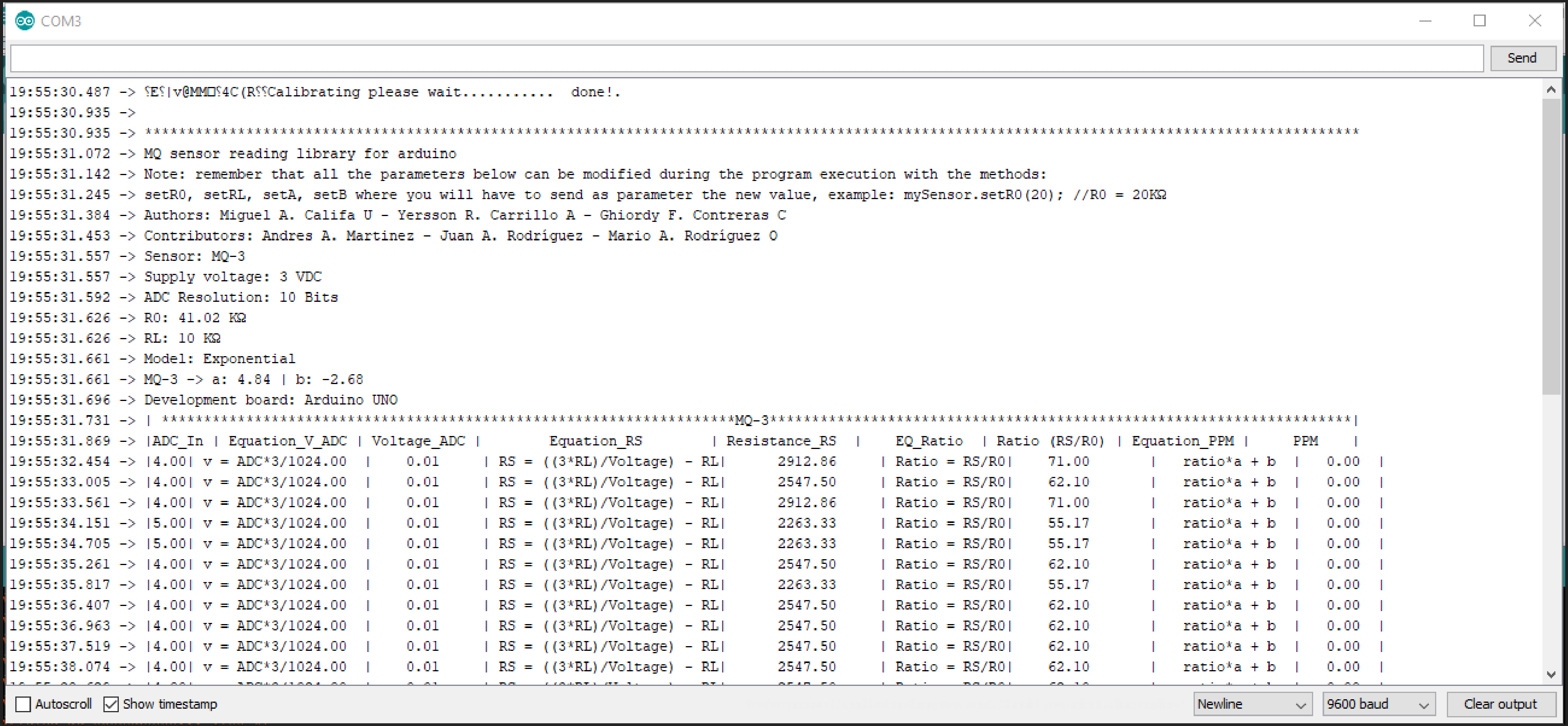
MQSensor and ESP8266 (Program)

```
    Serial.print("Calibrating please wait.");
    float calcR0 = 0;
    for(int i = 1; i<=10; i ++)
    {
      MQ3.update(); // Update data, the arduino will be read the voltage on the analog pin
      calcR0 += MQ3.calibrate(RatioMQ3CleanAir);
      Serial.print(".");
    }
    MQ3.setR0(calcR0/10);
    Serial.println("  done!.");
}
```

MQSensor and ESP8266 (Program)

```
void loop() {
  MQ3.update(); // Update data, the arduino will be read the voltage on the analog pin
  MQ3.readSensor();
  MQ3.serialDebug(); // Will print the table on the serial port
  delay(500); //Sampling frequency
}
```

MQSensor and ESP8266 (Result)



```
19:55:30.487 -> ߧEߧ|v@MM□ߧ4C(Rߧߧ Calibrating please wait...........  done!.
19:55:30.935 ->
19:55:30.935 -> ************************************************************************************************************************************************
19:55:31.072 -> MQ sensor reading library for arduino
19:55:31.142 -> Note: remember that all the parameters below can be modified during the program execution with the methods:
19:55:31.245 -> setR0, setRL, setA, setB where you will have to send as parameter the new value, example: mySensor.setR0(20); //R0 = 20KΩ
19:55:31.384 -> Authors: Miguel A. Califa U - Yersson R. Carrillo A - Ghiordy F. Contreras C
19:55:31.453 -> Contributors: Andres A. Martinez - Juan A. Rodríguez - Mario A. Rodríguez O
19:55:31.557 -> Sensor: MQ-3
19:55:31.557 -> Supply voltage: 3 VDC
19:55:31.592 -> ADC Resolution: 10 Bits
19:55:31.626 -> R0: 41.02 KΩ
19:55:31.626 -> RL: 10 KΩ
19:55:31.661 -> Model: Exponential
19:55:31.661 -> MQ-3 -> a: 4.84 | b: -2.68
19:55:31.696 -> Development board: Arduino UNO
19:55:31.731 -> | ********************************************************************MQ-3**********************************************************************|
19:55:31.869 -> |ADC_In | Equation_V_ADC | Voltage_ADC |        Equation_RS        | Resistance_RS  |    EQ_Ratio   | Ratio (RS/R0) | Equation_PPM |     PPM    |
19:55:32.454 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2912.86     | Ratio = RS/R0|    71.00       |   ratio*a + b  |   0.00  |
19:55:33.005 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2547.50     | Ratio = RS/R0|    62.10       |   ratio*a + b  |   0.00  |
19:55:33.561 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2912.86     | Ratio = RS/R0|    71.00       |   ratio*a + b  |   0.00  |
19:55:34.151 -> |5.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2263.33     | Ratio = RS/R0|    55.17       |   ratio*a + b  |   0.00  |
19:55:34.705 -> |5.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2263.33     | Ratio = RS/R0|    55.17       |   ratio*a + b  |   0.00  |
19:55:35.261 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2547.50     | Ratio = RS/R0|    62.10       |   ratio*a + b  |   0.00  |
19:55:35.817 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2263.33     | Ratio = RS/R0|    55.17       |   ratio*a + b  |   0.00  |
19:55:36.407 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2547.50     | Ratio = RS/R0|    62.10       |   ratio*a + b  |   0.00  |
19:55:36.963 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2547.50     | Ratio = RS/R0|    62.10       |   ratio*a + b  |   0.00  |
19:55:37.519 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2547.50     | Ratio = RS/R0|    62.10       |   ratio*a + b  |   0.00  |
19:55:38.074 -> |4.00| v = ADC*3/1024.00  |    0.01     | RS = ((3*RL)/Voltage) - RL|      2547.50     | Ratio = RS/R0|    62.10       |   ratio*a + b  |   0.00  |
```

MQSensor A2D External

Method **setADC(Your-ADC-VALUE)** replace **update()**

** Wiring is custom and it depends what A2D that you use.
** You need to set first the ADC Bit resolution and make sure is correct.
** After your read ADC method you Will set the value for MQ library using MQ.setADC(yourADCReadedValue).
** Use the library looks like sensor connected direct to your analog pin of board.
** Please remember, A0 pin doesnt matter, you can set 0…100, any.

MQSensor Digital Input

** Digital input is only alert output.
** Its setted before pre-heat and its experimental.
** Messages Will show on the serial output.

Any Questions:

Options:

1. Open issue on our github repo.
2. Contact us email: miguelangel5612@Gmail.com
3. Comment this video bellow section.