PH Sensor E-201-C

Technical Manual Rev 1r0

The pH sensor Module consist of PH sensor also called as PH probe and a Signal Conditioning Board which gives an output which is proportional to the PH Value and can be interfaced directly to any Micro-controller.

The pH sensor components are usually combined into one device called a combination pH electrode. The measuring electrode is usually glass and quite fragile. Recent developments have replaced the glass with more durable solid-state sensors. The preamplifier is a signal-conditioning device. It takes the high-impedance pH electrode signal and changes it into a low impedance signal which the analyzer or transmitter can accept. The preamplifier also strengthens and stabilizes the signal, making it less susceptible to electrical noise.

pH and ORP probes are both used for measuring the acidic intensity of liquid solutions. A pH probe measures acidity on a scale from 0 to 14, with 0 being the most acidic and 14 being the most basic. Similarly, an Oxidation-Reduction Potential (ORP) probe returns a voltage proportional to the tendency of the solution to gain or lose electrons from other substances (which is linked directly to the pH a substance).

Applications:

 - Can be used as different tester, pH meter or controller
 - Suitable for most aquariums, hydroponics, laboratory etc

General Specifications:

Input supply voltage: 5V
Working current: 5 – 10mA
Detection concentration range: PH 0 – 14
Detection range of temperature: 0 – 80 degC
Response Time:: ≤ 5S
Stability Time: ≤ 60S
Output: Analog
Power Consumption: ≤ 0.5W
Working Temperature: -10 to +50 deg C
Working Humidity: 95%RH (nominal humidity 65%RH)
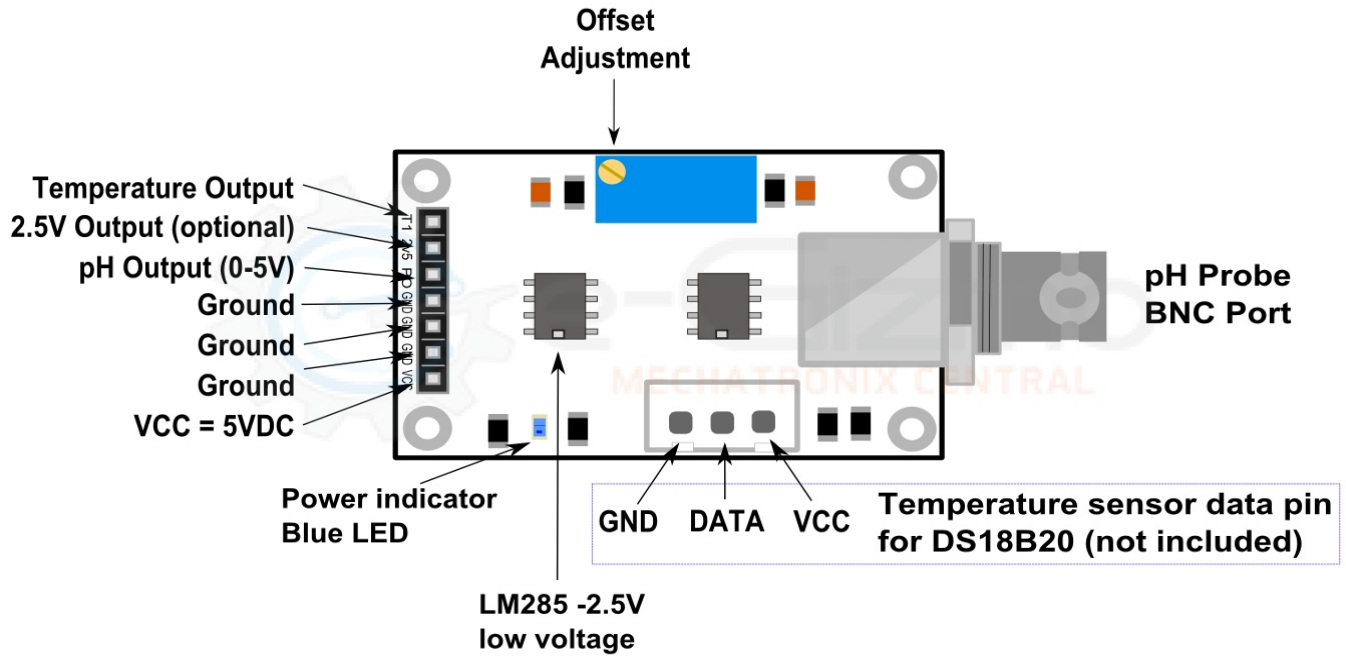**Weight:** 25g
PCB Dimension: 42mm x 32mm x 20mm

Figure 1: pH Sensor module kit v2 pinouts

Figure 2: pH Subtance (solutions for testing reference)

IMPORTANT TIPS:

On How to Clean pH Electrodes:

1. Do not "wipe" or rub the electrode.

2. Swirl the electrode gently in the cleaning soultion.

3. Gently rinse with deionized or distilled water.

4. Store in a storage solution.

5. When possible, use a specialized electrode.

Which Cleaning Solution to use?

The cleaning solution you use will depend upon your particular process and the residues you are trying to remove. There is a wide range of pre-mixed cleaning solutions available online or you can make your own. Make sure you take care when handling any cleaning solution – some can be hazardous so make sure you follow all safety instructions and wear appropriate protection equipment!

(For more information - Please go to reference link)
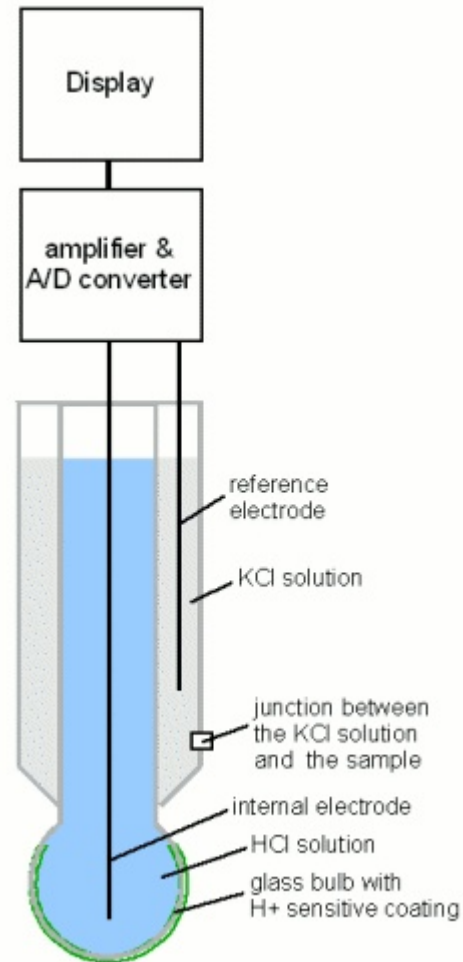
Reference:
https://www.southforkinst.com/ph-electrodes-clean-repair/

Figure 3: pH Electrode parts

Figure 1: Reference pH Value and Output voltage.

pH Value	Output
4	3.071
7	2.535
10	2.066

Table 2. pH Value of the known substances.

|	
0	5% Sulphuric acid, H2SO4, battery acid.
1	0.1 N HCl, hydrochloric acid (1.1)
2	Lemon juice. Vinegar (2.4-3.4)
3	wine (3.5-3.7)
4	Orange juice. Apple juice (3.8). Beer. Tomatoes.
5	Cottage cheese. Black coffee. Rain water 5.6.
6	Milk. Fish (6.7-7). chicken (6.4-6.6).
7	Neutral: equal numbers of hydrogen and hydroxyl ions. Blood (7.3-7.4). Distilled water without CO2, after boiling.
8	Sea water (8.1). Egg white.
9	Borax. baking soda.
10	Milk of magnesia, Magnesium hydroxide Mg (OH)2.
11	Household ammonia
12	Photographic developer, household bleach
13	Oven cleaner
14	Sodium lye NaOH, 1 mol/litre.

PH Sensor E-201-C
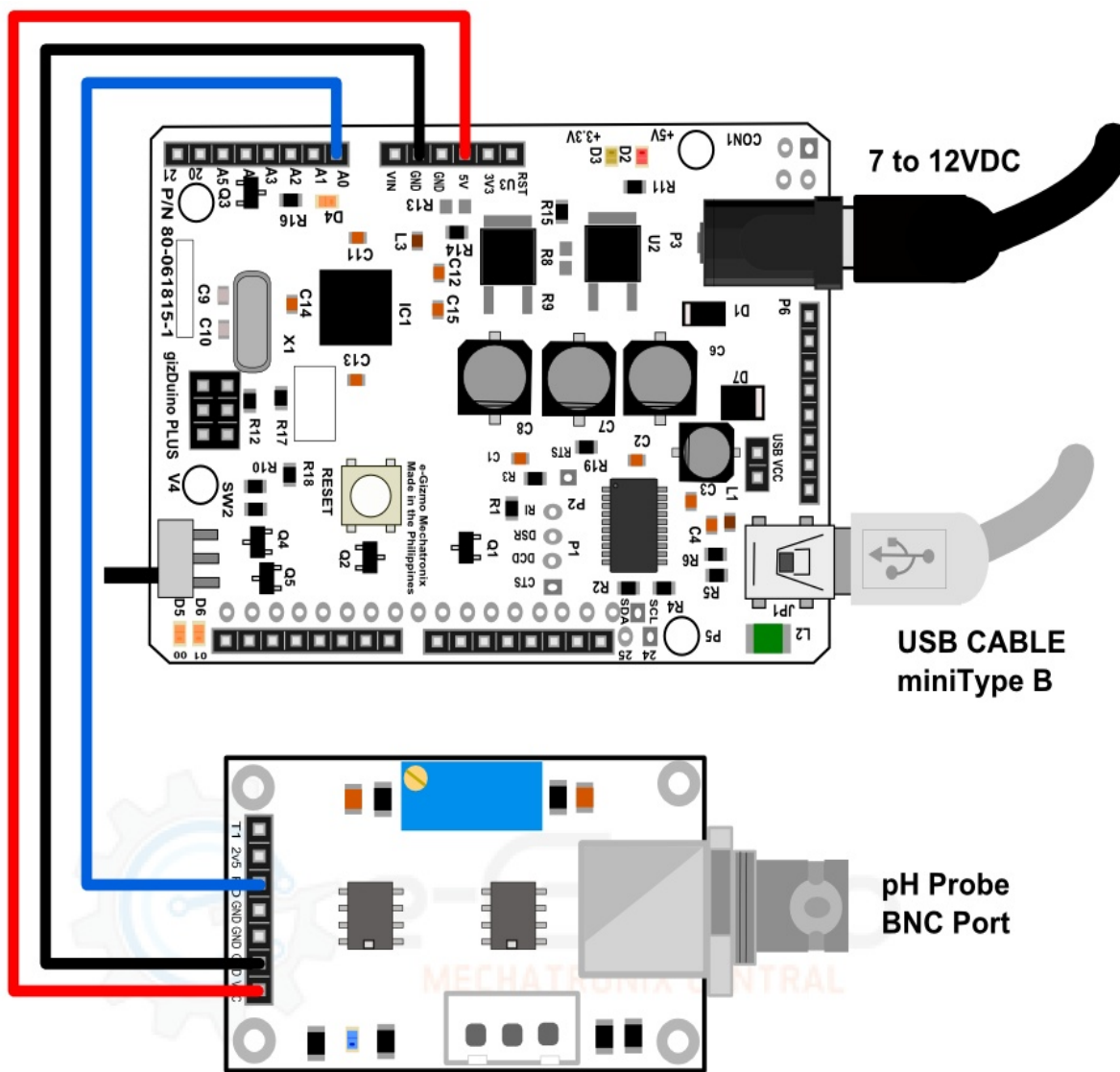
Figure 4: Sample connections.

```
/*
    e-Gizmo pH Sensor kit module v2.0

    This sample sketch is for pH sensor module
an easy output voltage to compare it to pH value.
See the table below:
pH Value  |  Output (V)
=========================
    4        3.071
    7        2.535
    10       2.066

For pH scale table for reference of the known substance.
Here's the link: http://www.seafriends.org.nz/dda/ph.htm
(See the pH scale)

Modified by e-Gizmo Mechatronix Central
http://www.e-gizmo.com
July 18,2017

*/
#define SENSOR A0          //pH meter Analog output to Arduino Analog Input 0
#define OFFSET 0.00         //deviation compensate
#define LED 13
#define SAMPLING_INTERVAL 20
#define PRINT_INTERVAL 800
#define ARRAY_LENGTH  40    //times of collection
int PH_ARRAY[ARRAY_LENGTH];   //Store the average value of the sensor feedback
int PH_ARRAY_INDEX=0;
void setup(void)
{
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("PH SENSOR KIT VOLTAGE TEST!");    //Test the serial monitor
}
void loop(void)
{
  static unsigned long SAMPLING_TIME = millis();
  static unsigned long PRINT_TIME = millis();
  static float VOLTAGE;
  if(millis()-SAMPLING_TIME > SAMPLING_INTERVAL)
  {
    PH_ARRAY[PH_ARRAY_INDEX++]=analogRead(SENSOR);
    if(PH_ARRAY_INDEX==ARRAY_LENGTH)PH_ARRAY_INDEX=0;
    VOLTAGE = AVERAGE_ARRAY(PH_ARRAY, ARRAY_LENGTH)*5.0/1024;
    SAMPLING_TIME=millis();
  }
  if(millis() - PRINT_TIME > PRINT_INTERVAL)   //Every 800 milliseconds, print a numerical, convert
the state of the LED indicator
  {
        Serial.print("VOLATGE OUTPUT: ");
```

```
        Serial.println(VOLTAGE,2);
        digitalWrite(LED,digitalRead(LED)^1);
        PRINT_TIME=millis();
    }
}
double AVERAGE_ARRAY(int* ARR, int NUMBER){
  int i;
  int max,min;
  double AVG;
  long AMOUNT=0;
  if(NUMBER<=0){
    Serial.println("ERROR!/n");
    return 0;
  }
  if(NUMBER<5){   //less than 5, calculated directly statistics
    for(i=0;i<NUMBER;i++){
      AMOUNT+=ARR[i];
    }
    AVG = AMOUNT/NUMBER;
    return AVG;
  }else{
    if(ARR[0]<ARR[1]){
      min = ARR[0];max=ARR[1];
    }
    else{
      min=ARR[1];max=ARR[0];
    }
    for(i=2;i<NUMBER;i++){
      if(ARR[i]<min){
        AMOUNT+=min;        //arr<min
        min=ARR[i];
      }else {
        if(ARR[i]>max){
          AMOUNT+=max;    //arr>max
          max=     AMOUNT+=ARR[i];
        }else{
          AMOUNT+=ARR[i]; //min<=arr<=max
        }
      }//if
    }//for
    AVG = (double)AMOUNT/(NUMBER-2);
  }//if
  return AVG;
}
```
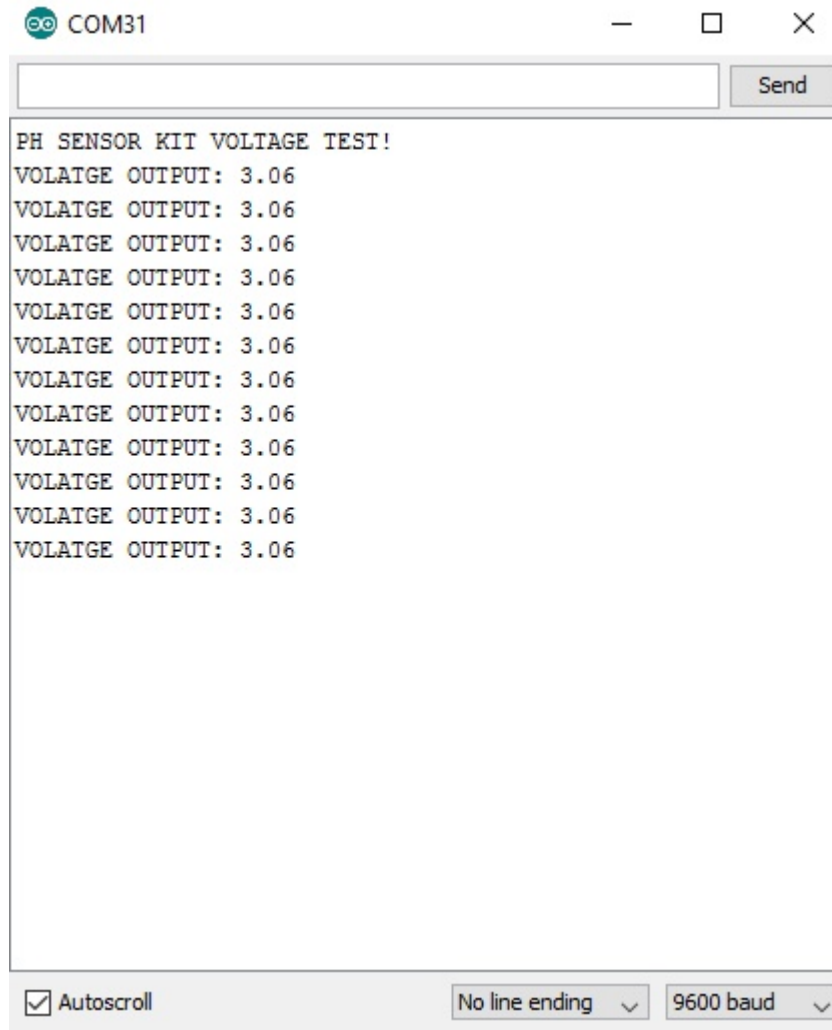
Figure 5: Serial print output from pH sensor.

PH Sensor E-201-C