

# UNIVERSIDAD TÉCNICA DEL NORTE

## FICA-CIERCOM

### SISTEMAS EMBEBIDOS

Benavides Wilmer, Farinango Luis, Velasco Angel

19 de noviembre de 2019

## 1. Introducción

La capacidad ambiental del agua indica el máximo de capacidad de contaminación permisible, utilizado para determinar el grado de contaminación y sus principales factores contaminantes (desperdicios industriales, aguas residuales domésticas), para poder recuperar y mejorar el medio ambiente del agua, con modelos de calidad (1-D y 2-D) donde se toma secciones para generar indicadores que se irán comparando consecuentemente y así poder medir la calidad del agua mediante una fórmula que abarca las siguientes variables: cantidad permisible de contaminante en el agua entre la sección de salida y la sección de control, capacidad de dilución y capacidad de auto purificación, donde se le relaciona a un grado de contaminación (entre I y V); posteriormente se obtendrá indicadores que conlleven a una hipótesis de qué tan contaminada se encuentra el agua y cómo mitigar la contaminación [1].

La condición ambiental del agua evaluada en términos de calidad y cantidad, se ha tomado muestras en 13 secciones distintas del río, en un periodo de tiempo de 8 años se ha encontrado altas concentraciones de DQO, NH<sub>3</sub>-N, BOD 5 y DO que exceden los estándares nacionales en las secciones con menos caudal en comparación con las secciones con corriente media y baja. Donde 5 estaciones hidrológicas han empleado métodos de evaluación (método de evaluación del índice, método difuso, método gris) y la identificación de la matriz de decisión, matriz de materia y matriz objetiva las cuales establecen que la explotación y el uso irrazonable de los recursos hídricos ponen en peligro el medio ambiente con el 80 % de grado de contaminación obteniendo una calificación de quinta clase o peor [2]. En la actualidad se llevan a cabo innumerables métodos para realizar el análisis del agua, uno de los más confiables es el conocido como GIS (Geographical Information System). ARK/INFO GIS guarda una red como una combinación de características de línea (links) y características de punto (nodos) con atributos asociados. Como las bases de datos GIS son estáticas y los lenguajes de programación GIS no están diseñados para análisis numérico, a menudo es difícil e ineficiente incorporar procedimiento numéricos complejos direc-

tamente dentro de un GIS [3].

El análisis hídrico también es llevado a cabo bajo el algoritmo de agrupamiento (clustering), mismo que hace referencia a una tecnología de reconocimiento exploratorio de patrones. En el estudio del análisis espacial de la calidad del agua, el análisis de agrupamiento jerárquico es el más usado, acorde al grado de afinidad-desafinidad de observación entre variables. Para obtener una lectura de calidad se buscan parámetros de características físicas y químicas de la calidad del agua, tales como: temperatura, pH, concentración de Nitrato, nitrógeno, etc. y evaluamos los resultados obtenidos [4].

Los sensores de nueva generación son desarrollados en base a sistemas multiparamétricos, por lo que mayormente se usan sistemas de sensores conectados entre sí para la recolección de datos. A diferencia de los sistemas tradicionales, estos pueden recolectar información de mejor calidad. Para la medición de calidad de agua se usa un parámetro llamado nivel de polución, dicho dato presenta un nivel de dificultad bastante alto para un sistema tradicional ya que se debe analizar en tres aspectos: biológico, químico y físico [5].

Ahora hay que sumar que la contaminación que afecta al agua de un río o vertiente es del tipo no puntual, es decir que no existe una fuente única que afecte al río de forma general, por lo que es mucho más difícil de aplacar. Por lo que a los factores ya mencionados hay que sumar niveles de concentración por zonas, los niveles de concentración por temporadas. Es decir que una muestra tomada en verano puede diferir bastante de una tomada en invierno incluso si se toma dicha muestra en el mismo lugar. Los parámetros químicos del agua que se obtienen en base a las mediciones son: COD, TP, NH<sub>3</sub>-N, NO<sub>2</sub>-N, NO<sub>3</sub>-N y TN. Las muestras según normal internacionales deben tomarse de acuerdo al método de sifón es decir se hace fluir un poco de agua entre dos recipientes, en este caso se debe tomar al menos agua de 5 centímetros de profundidad y el procedimiento debe durar al menos 30 minutos. Se recomienda tomar al menos tres muestras de agua [6].

## 2. Marco Teórico

### 2.1. Node MCU

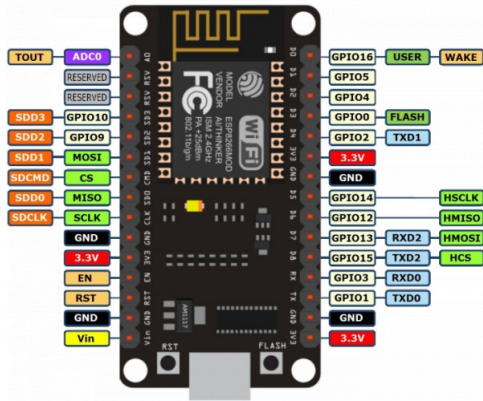


Figura 1: Node MCU( Esp8266)

ESP8266 no es un microcontrolador. Dentro si que lleva uno y se llama Tensilica L106 de 32-bit. La MCU se va a encargar de gestionar todas las entradas, salidas y cálculos necesarios para hacer funcionar el programa que hayamos cargado [7].

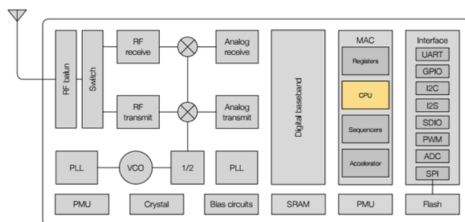


Figura 2: ESP8266EX Datasheet

### 2.2. Sensor de PH

El pH-4502C es una medida de acidez o alcalinidad de una disolución. El pH indica la concentración de iones hidronio  $[H_3O^+]$  presentes en determinadas sustancias. Este sensor permite medir el pH de un líquido mediante un electrodo ph y su placa controladora que ofrece un valor analógico proporcional a la medición.



Figura 3: Sensor module PH-4502C

### 2.3. Sensor de temperatura de agua

El DS18B20 ofrece 9 a 12 bits lecturas de temperatura más una interfaz 1-Wire, por lo que sólo un hilo debe conectarse desde un microprocesador central con un voltaje de entrada 3.0 V-5.5 V y un rango de temperatura de -55 hasta + 125 grados centígrados [8].



Figura 4: Sensor de temperatura DS18B20

### 2.4. Sensor TDS conductividad

TDS (total de sólidos disueltos): indica cuántos miligramos de sólidos disueltos se disuelven en 1 litro de agua. En general, cuanto mayor es el valor de TDS, más impura es el agua, posee un amplio suministro de voltaje de 3.3 5.5V y la salida de señal analógica de 0 2.3V.

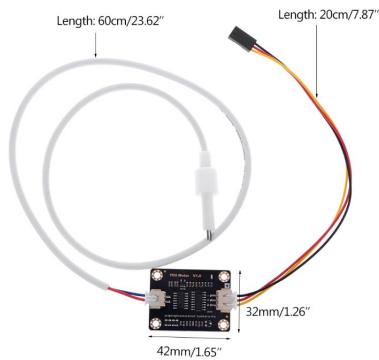


Figura 5: Sensor TDS

ra detectar la turbidez de la solución líquida, convierte la señal de corriente del sensor a voltaje de salida del módulo, cuanto menor es el voltaje de salida, mayor es el valor de turbidez.



Figura 6: Sensor NTU

## 2.5. Sensor de Turbidez (NTU)

La turbidez es la medición de partículas en el agua. Este sensor de turbidez adopta un principio óptico pa-

## 3. Desarrollo

### 3.1. Calibración de Sensor de PH

Para calibrar el sensor de PH se toma en cuenta que el rango promedio de la sonda oscila entre valores negativos y positivos. El 0 representa un pH de 7.0. Para poder usarlo con Arduino, este circuito agrega un valor de compensación al valor medido por la sonda, por lo que el ADC solo tendrá que tomar muestras de valores de voltaje positivos. Por lo tanto, forzaremos un pH de 7.0 desconectando la sonda del circuito y cortocircuitando el interior del conector BNC con el exterior. Con un multímetro, mida el valor del pin Po y ajuste el potenciómetro para que sea 2.5V.

Tomando valores de tablas ya existentes de PH de diversos líquidos se va calibrando hasta tener unos similares.

pH Meter Measurement Results v0.3 ttfong01 2019apr26hkt0928					
Sample	Output (V)	Googled pH	pH per V (1/V)	Measured pH	Error (%)
Water	2.50	7.00	n/a	n/a	n/a
Drain Pipe Cleaner	1.33	14.00	-5.98	13.44	-4
Bleach	1.89	13.00	-9.84	10.36	-25
Green Tea	2.58	7.20	2.50	6.56	-10
Milk	2.62	6.60	-3.33	6.34	-4
Body Wash pH 5.5	2.77	5.50	-5.56	5.51	0
Soy Sauce	2.91	5.00	-4.88	4.74	-5
Chinese Madarin	2.96	3.00	-8.70	4.47	33
Coffee	2.90	5.00	-5.00	4.80	-4
Vinegar	3.07	2.00	-8.77	3.86	48
Min	1.33	2.00	-9.84	3.86	-25
Max	3.07	14.00	2.50	13.44	48
Average	2.56	6.81	-5.51	6.68	3

Figura 7: Tabla de PH

Como se muestra en la Figura 7, la gráfica resultante usando la fórmula general  $y = mx + b$  para calcular  $m$  y  $b$ , donde  $x$  serían el voltaje y  $y$  el pH. El resultado es  $y = -5.70x + 21.34$ .

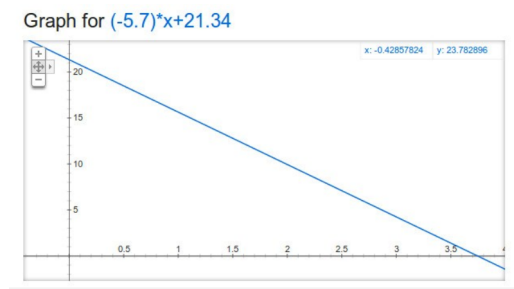


Figura 8: Ecuación de voltaje y pH medido.  $y = -5.70 * x + 21.34$

### 3.1.1. Programa utilizado

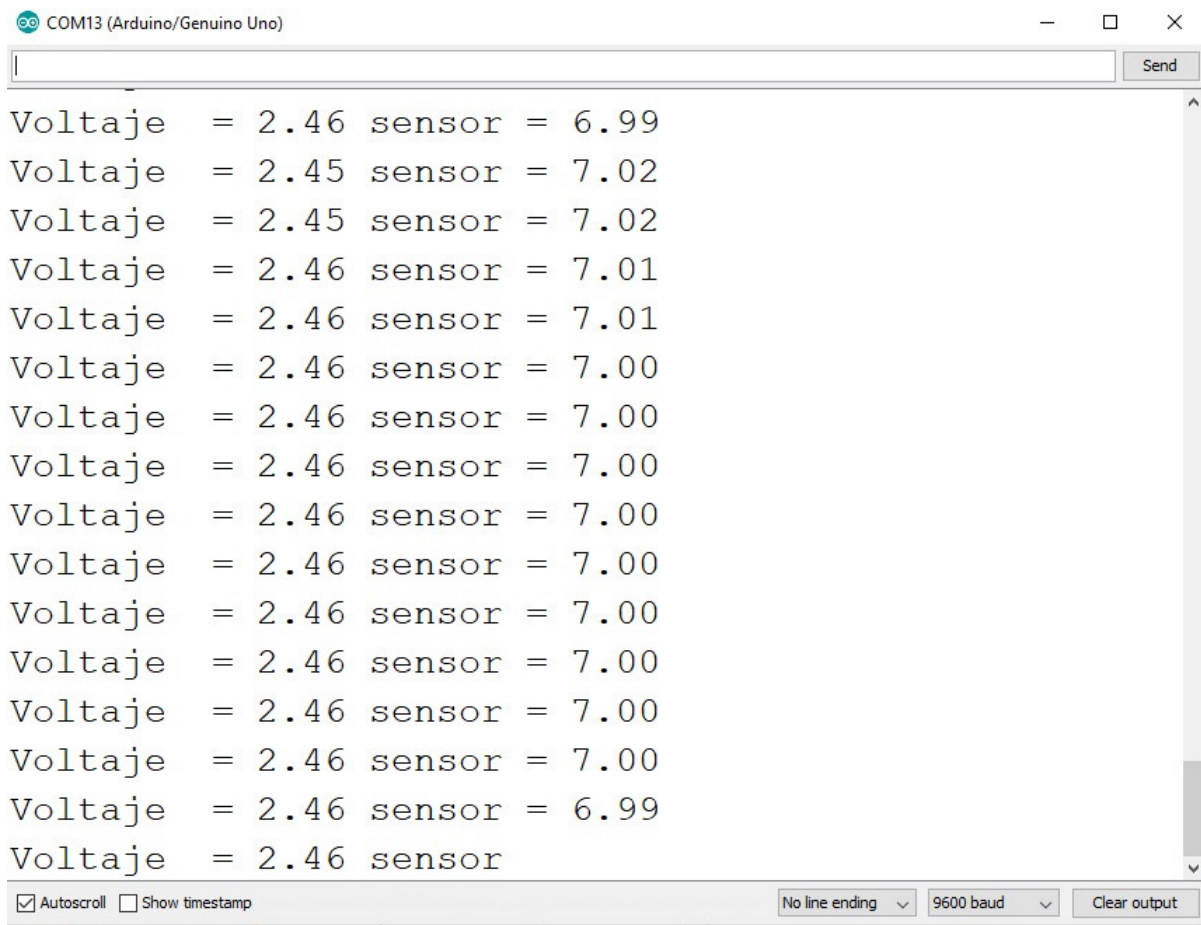
Programa 1: Código utilizado para el sensor de pH

```

1 float b, calibration = 21.34; //change this value to calibrate
2 const int analogInPin = A0;
3 int buf[10], temp, sensorValue = 0;
4 unsigned long int avgValue;
5
6 void setup() {
7     Serial.begin(9600);
8 }
9
10 void loop() {
11     for (int i = 0; i < 10; i++) {
12         buf[i] = analogRead(analogInPin);
13         delay(30);
14     }
15     for (int i = 0; i < 9; i++) {
16         for (int j = i + 1; j < 10; j++) {
17             if (buf[i] > buf[j]) {
18                 temp = buf[i];
19                 buf[i] = buf[j];
20                 buf[j] = temp;
21             }
22         }
23     }
24     avgValue = 0;
25     for (int i = 2; i < 8; i++)
26         avgValue += buf[i];
27     float pHVol = (float)avgValue * 5.0 / 1024 / 6;
28     float pHValue = -5.70 * pHVol + calibration;
29     Serial.print("vol = ");
30     Serial.println(pHVol);
31     Serial.print("sensor = ");
32     Serial.println(pHValue);
33     delay(500);
34 }

```

### 3.1.2. Resultados obtenidos

The image shows a screenshot of the Arduino IDE serial monitor window. The title bar indicates 'COM13 (Arduino/Genuino Uno)'. The monitor displays a series of data points in a monospaced font. Each line contains two values: 'Voltaje' followed by an equals sign and a number, and 'sensor' followed by an equals sign and a number. The 'Voltaje' values are mostly 2.46, with some 2.45. The 'sensor' values are mostly 7.00, with some 7.01, 7.02, and 6.99. The data is scrolling upwards. At the bottom of the window, there are controls: a checked 'Autoscroll' checkbox, an unchecked 'Show timestamp' checkbox, a 'No line ending' dropdown menu, a '9600 baud' dropdown menu, and a 'Clear output' button.

```
COM13 (Arduino/Genuino Uno)
Send

Voltaje = 2.46 sensor = 6.99
Voltaje = 2.45 sensor = 7.02
Voltaje = 2.45 sensor = 7.02
Voltaje = 2.46 sensor = 7.01
Voltaje = 2.46 sensor = 7.01
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 7.00
Voltaje = 2.46 sensor = 6.99
Voltaje = 2.46 sensor

☒ Autoscroll ☐ Show timestamp
No line ending 9600 baud Clear output
```

Figura 9: Datos adquiridos

### 3.2. Calibración de Sensor de temperatura

Para el sensor de temperatura se tomo muestras con agua de diferentes temperaturas, como el agua en 100 grados llega a su punto de ebullición

<b><i>Tabla Punto de Ebullición de algunas sustancias a 1 atm</i></b>	
<b>Sustancia</b>	<b>Punto de Ebullición (°C)</b>
Agua	100
Etanol	78
Sal de mesa	1.465
Cobre	2.300
Ácido acético	118
Hierro	3.000
Plata	1.950

Figura 10: Punto de ebullición

### 3.2.1. Programa utilizado

Programa 2: Código utilizado para el sensor de Temperatura

```
1 #include <OneWire.h>
2 #include <DallasTemperature.h>
3 #define ONE_WIRE_BUS D1 //Se establece el pin D1 como bus OneWire
4 OneWire oneWire(ONE_WIRE_BUS);
5 DallasTemperature sensors(&oneWire); //Se declara una variable u objeto para
   nuestro sensor
6
7 void setup() {
8   Serial.begin(9600);
9 }
10
11 void loop() {
12   sensors.requestTemperatures(); // Enviamos este comando para obtener del
   sensor el valor de la temperatura
13   float temp = sensors.getTempCByIndex(0);
14   Serial.println("Temperatura is: ");
15   Serial.println(temp); // Podemos tener más de un sensor en el mismo bus, 0 se
   refiere al primer sensor del bus
16   delay(3000);
17 }
```

### 3.2.2. Resultados obtenidos

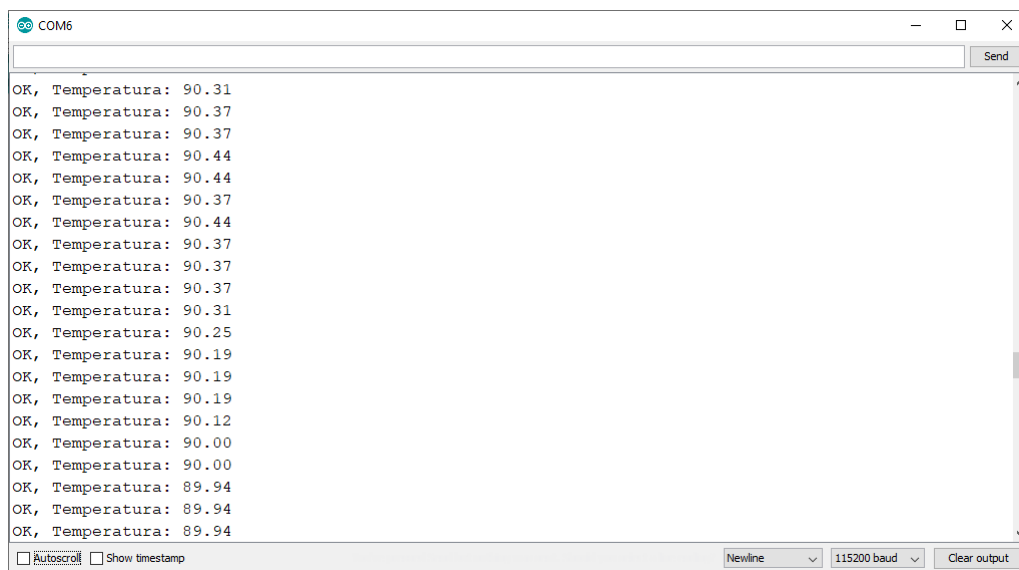


Figura 11: Datos adquiridos

### 3.3. Calibración de Sensor TDS

Tomando de referencia el color del agua se puede usar distintos colores de agua y tomando tablas ya existentes de referencia para calibrar el sensor.



Figura 12: Conductividad del agua

### 3.3.1. Programa utilizado

Programa 3: Código utilizado para el sensor TDS

```
1 const float ArduinoVoltage = 3.3; // Voltaje usado
2 const float ArduinoResolution = ArduinoVoltage / 1024;
3 const float resistorValue = 10000.0;
4 int threshold = 3;
5 int inputPin = A0;
6 int ouputPin = A5;
7
8 void setup() {
9     Serial.begin(9600);
10    pinMode(ouputPin, OUTPUT);
11    pinMode(inputPin, INPUT);
12 }
13
14 void loop(){
15     int analogValue = 0;
16     int oldAnalogValue = 1000;
17     float returnVoltage = 0.0;
18     float resistance = 0.0;
19     double Siemens;
20     float TDS = 0.0;
21
22     while (((oldAnalogValue - analogValue) > threshold) || (oldAnalogValue < 50))
23     {
24         oldAnalogValue = analogValue;
25         digitalWrite( ouputPin, HIGH );
26         delay(10); // allow ringing to stop
27         analogValue = analogRead( inputPin );
28         digitalWrite( ouputPin, LOW );
29     }
30
31     Serial.print("Return voltage = ");
32     returnVoltage = analogValue * ArduinoResolution;
33     Serial.print(returnVoltage);
34     Serial.println(" volts");
35     Serial.print("That works out to a resistance of ");
36     resistance = ((5.00 * resistorValue) / returnVoltage) - resistorValue;
37     Serial.print(resistance);
38     Serial.println(" Ohms.");
39     Serial.print("Which works out to a conductivity of ");
40     Siemens = 1.0 / (resistance / 1000000);
41     Serial.print(Siemens);
42     Serial.println(" microSiemens.");
43     Serial.print("We can estimate Total Dissolved Solids to be on the order of ");
44     TDS = 500 * (Siemens / 1000);
45     Serial.print(TDS);
46     Serial.println(" PPM.");
47     if (returnVoltage > 4.9) Serial.println("Are you sure this isn't metal?");
48     delay(5000);
49 }
```



### 3.3.2. Resultados obtenidos

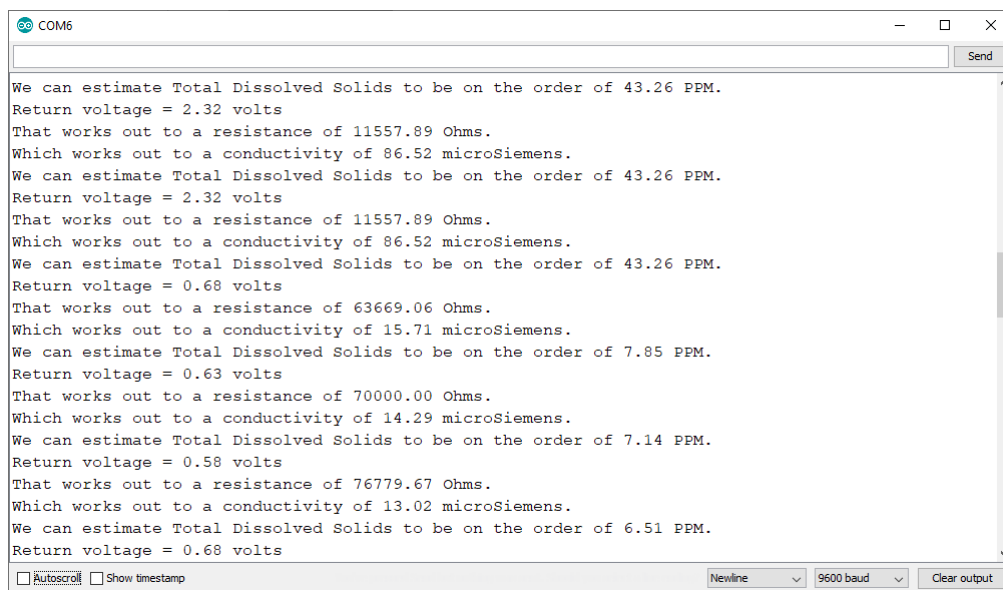


Figura 13: Datos adquiridos

### 3.4. Calibración de Sensor NTU

Ya que este módulo convierte la señal de corriente del sensor a voltaje de salida del módulo. Cuanto menor es el voltaje de salida, mayor es el valor de turbidez, y tomando como referencia para calibrar la turbidez en tablas ya existentes.

Masa de agua	Nivel de turbidez
Masas de agua con pocas plantas y animales	0 JTU
Agua potable	<0.5 JTU
Agua subterránea típica	<1.0 JTU
Masas de agua con cantidad moderada de plantas y animales	1 - 8 JTU
Masas de agua con cantidades grandes de vida planctónica	10 - 30 JTU
Agua turbia o flujos en ríos de las tormentas de invierno	20 - 50 JTU

Figura 14: Turbidez del agua dependiendo de la masa

#### 3.4.1. Programa utilizado

Programa 4: Código utilizado para el sensor de Turbidez

```
1 float voltaje ;
2 float NTU;
3
4 void setup() {
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   int sensorVal = analogRead(A0);
10  voltaje = sensorVal * (5.00 / 1023.0);
11  //y = 174.96x2 - 1357.1x + 2630.5
12  //285.48x2 - 2285.5x + 4574.8
13
```



```

14 NTU = (float)(285.48 * pow(voltaje , 2)) - (float)2285.5 * voltaje + (float)
    4574.8;
15 //NTU=(float)(-1120.4*pow(voltaje ,2))+(float)5742.3*voltaje -(float)4352.9;
16 float NTUC = NTU / 10.00;
17 Serial.print("volatje es :");
18 Serial.print(voltaje);
19 Serial.print(" Turbidez es :");
20 Serial.println(NTU);
21 delay(500);
22 }

```

### 3.4.2. Resultados obtenidos

```

volatje es :2.10 Turbidez es :1037.72
volatje es :2.16 Turbidez es :969.73
volatje es :2.10 Turbidez es :1037.72
volatje es :2.08 Turbidez es :1053.74
volatje es :2.10 Turbidez es :1037.72
volatje es :2.08 Turbidez es :1053.74
volatje es :2.72 Turbidez es :468.59
volatje es :2.71 Turbidez es :475.76
volatje es :2.71 Turbidez es :479.37
volatje es :2.69 Turbidez es :490.28
volatje es :2.70 Turbidez es :486.63
volatje es :2.70 Turbidez es :486.63
volatje es :2.69 Turbidez es :490.28
volatje es :2.69 Turbidez es :490.28
volatje es :2.70 Turbidez es :486.63

```

Figura 15: Datos adquiridos

## 4. Conclusiones y Recomendaciones

### Conclusiones:

- La conductividad del agua depende estrictamente de cuanto se encuentre contaminada ya que una mayor contaminación se traduce como una mayor cantidad de impurezas en el líquido, esto provoca una concentración de iones que hacen que la conductividad aumente.
- La calibración del sensor de turbidez y pH esta basado en ecuaciones que dan la relación de estas dos magnitudes en función del voltaje.
- El uso del componente electrónico ESP8266(NodeMCU) puede llegar a dar un sinnúmero de problemas, ya que no es un dispositivo utilizado muy a menudo y es complicado encontrar información acerca de este y de su funcionamiento, a mayores, este MCU cuenta con incompatibilidades con ciertas librerías de arduino, pero siempre se encuentra otra alternativa.

### Recomendaciones:

- \* Calibrar los sensores con un número de muestras amplia, ya que así se puede comprobar el correcto funcionamiento de dichos sensores.
- \* Revisar el datasheet para saber cual es el valor de alimentación de trabajo de los sensores para no averiarlos.
- \* Encontrar las ecuaciones adecuadas para cada calibración, existirán casos donde se deberá generar la ecuación en base a mediciones de los sensores.

## 5. Anexos



Figura 16: Pruebas realizadas

## Referencias

- [1] Z.-z. Zhang, F. Zhang, C.-g. Xu, J.-x. Xu, W.-w. Zhang, and Q.-q. Qi, "Study on the water environment capacity for the typical watershed in taizihe river," in *2011 International Symposium on Water Resource and Environmental Protection*, vol. 1. IEEE, 2011, pp. 486–488.
- [2] W. Guo, J. Chen, Y. Sheng, and J. Wang, "Integrated evaluation of water quality and quantity of the wei river reach in shaanxi province," in *2011 International Symposium on Water Resource and Environmental Protection*, vol. 2. IEEE, 2011, pp. 863–866.
- [3] Y. Jiang, "Application of gis network analysis in water pollution control of huaihe river basin," in *2011 19th International Conference on Geoinformatics*. IEEE, 2011, pp. 1–4.
- [4] W. Jingmeng, G. Xiaoyu, Z. Wenji, and M. Xiangang, "Research on water environmental quality evaluation and characteristics analysis of tonghui river," in *2011 International Symposium on Water Resource and Environmental Protection*, vol. 2. IEEE, 2011, pp. 1066–1069.
- [5] Y. Qin, H. Li, J. Li, and L. Zhu, "Impact of nonpoint source pollution on water quality of the bahe river," in *2011 International Symposium on Water Resource and Environmental Protection*, vol. 3. IEEE, 2011, pp. 2121–2124.
- [6] S. Randhawa, S. S. Sandha, and B. Srivastava, "A multi-sensor process for in-situ monitoring of water pollution in rivers or lakes for high-resolution quantitative and qualitative water quality data," in *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*. IEEE, 2016, pp. 122–129.
- [7] R. L. Boylestad and L. Nashelsky, *Electrónica: teoría de circuitos y dispositivos electrónicos*. PEARSON educación, 2003.
- [8] J. E. Kemmerly and W. H. Hayt, *Análisis de circuitos en ingeniería*. McGraw-Hill Companies, Incorporated, 1975.