



**Benemérita
Universidad Autónoma de Puebla**

FACULTAD DE INGENIERÍA QUÍMICA

**“Construcción y validación del prototipo de
monitoreo basado en Arduino para el control de
calidad de agua”**

TESIS PROFESIONAL

Que para obtener el Título de:

Licenciatura en Ingeniería Química

Presenta:

JUAN JESÚS ACOCAL CAPILLA

Director de Tesis:

Dr. Alberto E. Campos Reales Pineda

Noviembre, 2018

RESÚMEN

En las últimas décadas se han presentado diversas alternativas para un mejor uso, reúso y tratamiento de las aguas en centros urbanos e industriales. En México el tratamiento y reúso del agua se ha venido incrementado conforme al paso de los años, pero de igual forma se han desarrollado los diferentes sectores como lo son: el industrial, energético, de la construcción. Sin embargo, hay muy pocas propuestas para mejorar el costo de los sistemas de monitoreo y darle un seguimiento apropiado a las descargas de residuos a los cuerpos de agua que los rodean como lo son lagos, ríos o presas; lo cual hace que su implementación a gran escala sea limitada y en algunos casos el mantenimiento o costo operativo se vuelve elevado por lo que éstos pueden ser abandonados eventualmente.

Por esta razón, en este trabajo se propone el diseño, construcción, programación y prueba de un prototipo de sensores de calidad de agua basados en el microcontrolador de código abierto Arduino.

Para lograr este objetivo, se utilizó una placa Arduino UNO, a la cual se conectaron sensores de pH, temperatura y turbidez. Se utilizó el entorno de programación Arduino IDE 1.8.5, con la cual se compilaban los datos de medición de dichos sensores y se corroboró la viabilidad técnica del prototipo, además de presentar un costo menor que el de las alternativas comerciales disponibles en el mercado.

Del presente trabajo se concluye que el prototipo diseñado puede ser utilizado como base para la implementación de una red de monitoreo para diversos cuerpos de agua que sea confiable y viable en sus costos de implementación, mantenimiento y personal, con la posibilidad de poder monitorear el agua en tiempo real y la capacidad de enviar alertas mediante red celular si se rebasan límites preestablecidos. Se esperaba que con la implementación del material propuesto para este trabajo reducirá el costo significativamente comparado con inversiones que se han hecho antes para este problema.

Palabras clave: Ríos y lagos, sensores, hardware abierto, bajo costo, parámetros.

ABSTRACT

In recent decades, different alternatives have been presented for better use, reuse and treatment of water in urban and industrial centres. In Mexico the treatment and reuse of the water has been increased according to the passage of the years, but in the same way the different sectors have developed as they are: the industrial, energetic, of the construction. However, there are very few proposals to improve the cost of monitoring systems and to give appropriate follow-up to waste discharges to the bodies of water surrounding them, such as lakes, rivers or dams; This makes its implementation on a large scale limited and in some cases the maintenance or operating cost becomes high so they can be abandoned eventually. For this reason, this paper proposes the design, construction, programming and testing of a prototype of water quality sensors based on the Arduino open source microcontroller. To achieve this objective, a Arduino UNO board was used, to which pH, temperature and turbidity sensors were connected. The Arduino IDE 1.8.5 programming environment was used, with which the measurement data of these sensors were compiled and the technical feasibility of the prototype was corroborated, in addition to presenting a lower cost than the commercial alternatives available in the Market. This paper concludes that the designed prototype can be used as a basis for the implementation of a monitoring network for various bodies of water that is reliable and viable in its implementation, maintenance and personnel costs, with the possibility of Be able to monitor the water in real time and the ability to send alerts via cellular network If you exceed preset limits. It would be expected that with the implementation of the proposed material for this work will reduce the cost significantly compared with investments that have been made earlier for this problem. Key words: rivers and lakes, sensors, open hardware, low cost, parameters.

LISTADO DE TABLAS

| | |
|--|----|
| Tabla 2.2 definición de: monitoreo y evaluación | 13 |
| Tabla 2.3a Sitios de la Red Nacional de Monitoreo..... | 14 |
| Tabla 2.7.1 Rangos de turbidez [UNT]..... | 22 |
| Tabla 4.1.1a NTU del agua a diferentes concentraciones de café | 35 |
| Tabla 4.1.1b NTU: prototipo, equipo comercial | 37 |
| Tabla 4.1.1c NTU del agua a diferentes concentraciones de café | 37 |
| Tabla 4.1.1d NTU: prototipo, equipo comercial | 39 |
| Tabla 4.1.1e NTU del agua a diferentes concentraciones de café | 39 |
| Tabla 4.1.1f NTU: prototipo, equipo comercial..... | 40 |
| Tabla 4.1.2 Temperaturas | 42 |
| Tabla 4.1.3a pHdesviación estándar..... | 46 |
| Tabla 4.1.3b pHdesviación estándar..... | 48 |
| Tabla 4.3a Costo de construcción | 54 |
| Tabla 4.3b Costo de equipos comerciales | 54 |

LISTADO DE FIGURAS

| | |
|---|----|
| Figura 1. Distribución de la Generación y Tratamiento de las aguas residuales en México durante el 2015 | 4 |
| Figura 2a Valores anuales de los componentes del ciclo hidrológico en México, 2015 | 9 |
| Figura 2b Principales lagos de la zona centro | 10 |
| Figura 2.2 La secuencia de los problemas de calidad del agua que surgen en los países industrializados | 12 |
| Figura 2.4 Tarjeta de circuito Arduino UNO | 15 |
| Figura 2.6a Pantalla principal Arduino. Estructura de programación | 18 |
| Figura 2.6b Ejemplo: función setup (), loop () | 19 |
| Figura 2.7.1 Turbidez | 20 |
| Figura 3 Metodología para la elaboración del prototipo de monitoreo de calidad de agua | 26 |
| Figura 3.2a Descarga de archivo para la instalación de Arduino | 27 |
| Figura 3.2b Programa ejemplo blink: encender y apagar un led. | 27 |
| Figura 3.1.1a Sensor de turbidez SKU: SEN0189..... | 28 |
| Figura 3.1.1 Diagrama de conexión del sensor de Turbidez: SEN0189 | 29 |
| Figura 3.1.2a Sensor de temperatura DS10B20 impermeable..... | 29 |
| Figura 3.1.2b Alimentación del DS10B20 con una fuente externa | 30 |
| Figura 3.1.1 Diagrama de conexión del sensor de temperatura DS18B20..... | 30 |
| Figura 3.3.3a Sensor de pH SKU: SEN0161..... | 31 |
| Figura 3.3.3b Diagrama de conexión del sensor de pH SKU: SEN0161..... | 31 |
| Figura 3.3.4a Diagrama de conexión del sensor de Oxígeno disuelto..... | 32 |
| Figura 3.3.4b Preparación de sonda nueva del sensor de pH SKU: SEN0161 | 32 |
| Figura 3.3.5 Diagrama de conexión del sensor de Oxígeno Disuelto SKU: DFR0300 | 33 |
| Figura 4.1.1a Interruptor en ajuste analógico..... | 34 |
| Figura 4.1.1b Código ejemplo DFRobot | 34 |

| | |
|---|----|
| Figura 4.1.1c Turbidímetro HACH2100N IS..... | 35 |
| Figura 4.1.1d Código de programación para NTU del sensor de turbidez de la práctica 1..... | 37 |
| Figura 4.1.1e Código de programación para NTU del sensor de turbidez de la práctica 2..... | 38 |
| Figura 4.1.1f Código de programación para NTU del sensor de turbidez de la práctica 3..... | 40 |
| Figura 4.1.2a Código de programación para temperatura del sensor DS10B20 | 41 |
| Figura 4.1.2b Montado de equipo | 42 |
| Figura 4.1.3a Código de ejemplo de programación... .. | 44 |
| Figura 4.1.3b Código de programación del sensor de pH. Práctica 1... .. | 46 |
| Figura 4.1.3c Buffer: 4.01, 7, 10 HYCEL... .. | 46 |
| Figura 4.1.3c Código de programación del sensor de pH. Práctica 1..... | 47 |

CONTENIDO

| | |
|--|-----|
| RESÚMEN..... | i |
| ABSTRACT | ii |
| LISTADO DE TABLAS | iii |
| LISTADO DE FIGURAS | iv |
| | |
| CAPITULO 1. INTRODUCCIÓN..... | 3 |
| 1. Introducción | 3 |
| 1.1. Justificación | 5 |
| 1.2. Objetivo | 7 |
| 1.2.1 Objetivo general..... | 7 |
| 1.2.2 Objetivos específicos | 7 |
| 1.3 Hipótesis | 7 |
| CAPITULO 2. MARCO TEÓRICO..... | 8 |
| Marco teórico | 8 |
| 2.1 Contaminación del agua..... | 10 |
| 2.1.1 Aguas residuales en México | 11 |
| 2.2 Calidad de agua | 12 |
| 2.3 Monitoreo de la calidad de agua..... | 14 |
| 2.4 Arduino..... | 14 |
| 2.4.1 Open Source..... | 16 |
| 2.4.2 Hardware libre | 16 |
| 2.4.3 Software libre..... | 16 |
| 2.5 Tendencia de Arduino en proyectos de investigación | 17 |
| 2.6 Lenguaje y estructura de programación..... | 18 |
| 2.7 Parámetros básicos de calidad del agua | 20 |
| 2.7.1 Turbidez..... | 20 |
| 2.7.2 Temperatura | 23 |
| 2.7.3 pH..... | 23 |
| 2.7.4 Oxígeno disuelto..... | 24 |
| 2.7.5 Conductividad | 25 |
| CAPITULO 3. METODOLOGÍA | 26 |
| Metodología | 26 |
| 3.1 Adquisición de materiales..... | 26 |
| 3.2 Instalación del software | 27 |
| 3.3 Sensores y su funcionamiento..... | 28 |
| 3.3.1 Sensor de Turbidez..... | 28 |

| | |
|--|----|
| 3.3.2 Sensor de Temperatura | 29 |
| 3.3.3 Sensor de pH..... | 31 |
| 3.3.4 Sensor de Oxígeno disuelto | 32 |
| 3.3.5 Sensor de Conductividad | 33 |
| CAPITULO 4.RESULTADOS..... | 34 |
| 4.1 Pruebas y comparación (prototipo- laboratorio)..... | 34 |
| 4.1.1 Sensor de turbidez..... | 34 |
| 4.1.2 Sensor de temperatura | 41 |
| 4.1.3 Sensor de pH..... | 43 |
| 4.2 Código para el monitoreo de: Turbidez, temperatura y pH..... | 48 |
| 4.3 Costos..... | 54 |
| CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES..... | 55 |
| CAPÍTULO 6. REFERENCIAS..... | 57 |
| CAPITULO 7. ANEXOS..... | 59 |

CAPITULO 1. INTRODUCCIÓN

1. Introducción

Desde la perspectiva ambiental, la industrialización ha causado graves problemas de contaminación que con el tiempo han aumentado de manera exponencial, con ello ha traído grandes problemas para el medio ambiente; uno de los recursos naturales que más ha sufrido este impacto ha sido el agua. Las industrias utilizan este recurso en diferentes labores dentro de la planta y después de ser utilizado en la mayoría de los casos es desechado con restos de residuos que son dañinos para el medio ambiente.

El agua es uno de los principales recursos naturales ya que es esencial para todo ser vivo. El agua es descrita por sus propiedades físicas como un líquido incoloro, inodoro e insípido, es definido como el “*disolvente universal*” debido a su estructura molecular polar que le confiere dicha propiedad.

El agua es un recurso indispensable para la mayoría de las industrias y no se ha podido sustituir su uso. Ya que es tan indispensable, es importante que se le dé un seguimiento al tema de su contaminación.

Nuestros océanos, mares, ríos y demás cuerpos de agua conforman el total del agua existente en nuestro planeta, de la cual 97.5% es agua salada (océanos y mares), un 2.5% constituye el agua dulce y solo el 1% está disponible (AgroDer, 2012). El agua dulce disponible proviene de dos principales fuentes de suministro, aguas subterráneas y aguas superficiales.

Respecto al tratamiento y reúso del agua en México. En el documento “Estadísticas del agua en México” en su última edición 2016 (CONAGUA, 2016), se clasifican las descargas de aguas residuales en municipales y no municipales. Las municipales pertenecen a las que son formadas en los núcleos de población y recolectadas en los sistemas de alcantarillado urbano y rural, por otra parte, las no municipales son aquellas generadas por otros usos como lo es el sector industrial.

En la Figura 1 se muestra que durante el año 2015, se generó un total de 443.7 metros cúbicos por cada segundo (m^3/s) de aguas residuales totales a nivel nacional, las 2,477 plantas en operación a lo largo del país, trataron 120.9 m^3/s de los 212 m^3/s recolectados de centros urbanos.

*Unidades de caudales m^3/s .

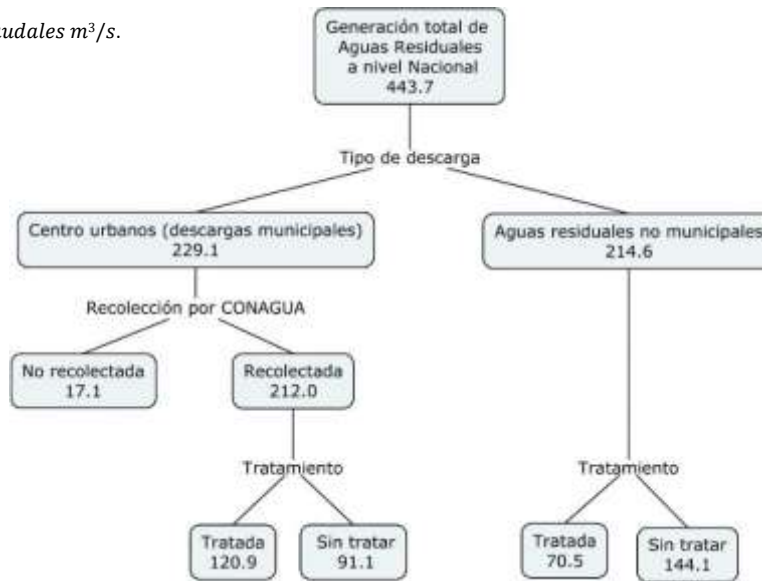


Figura 1. Distribución de la Generación y Tratamiento de las aguas residuales en México durante el 2015 (CONAGUA, 2016).

En el Estado de Puebla las principales industrias son del giro textil, químico, materiales de construcción, electromecánica y automotriz, el estado cuenta con 18 parques industriales y/o tecnológicos (SE, 2015), estos descargan aguas residuales a los ríos Atoyac y Alseseca. De igual forma las descargas municipales de las poblaciones por las que atraviesan en su recorrido hasta llegar a la presa Valsequillo (Saldaña, 2002).

El río Atoyac localizado principalmente en los estados de Puebla y Tlaxcala, es el principal cuerpo receptor que cruza el municipio de Puebla, nace de los deshielos y escurrimientos del volcán Iztaccihuatl; en su recorrido por el valle de Puebla recibe como tributarios a los ríos Xopanac, Zanja Real y La Cadena, así como al río Zahuapan, que en su trayectoria cruza por el estado de Tlaxcala; recibe además aguas de los ríos Zapatero, el Rabanillo, Ometlapanapa y San Francisco. Tiene una trayectoria orienteponiente a lo largo de los límites geográficos de los estados de Puebla y Tlaxcala.

Este río tiene una trayectoria de 28.65 kilómetros (km) de recorrido en el municipio de Puebla y 16 km dentro de la mancha urbana de la ciudad. El caudal medio anual antes de la descarga en la presa Valsequillo es de $6.7 m^3/s$. El caudal medio anual de las aguas residuales de Puebla es de $1.73 m^3/s$. De acuerdo a la información anterior, la participación de las aguas residuales representan el 26% del caudal medio anual del río Atoyac (Saldaña, 2002).

El río Alseseca nace de las numerosas barrancas y corrientes intermitentes que descienden de las faldas de la montaña Malinche; se forma básicamente con las descargas del drenaje sanitario de la ciudad de Puebla, su caudal medio anual es de

0.8 m³/segundo, compuesto por 0.7 m³/s de descargas de alcantarillado sanitario (88%) y 0.1 m³/s de otros escurrimientos (12%) (Saldaña, 2002).

El río Atoyac es el tercero más contaminado del país (Castillo, 2013) lo cual es consecuencia del impacto de diferentes descargas de tipo residual, así como la contribución natural de procesos erosivos, biológicos, y de influencia por actividad volcánica.

1.1. Justificación

Algunos estudiosos de la crisis del agua señalan que éste no es un problema de escasez física, sino de una deficiente gestión y para apoyar esta idea habrá que hacer un recuento de la evolución del desarrollo humano y su vinculación con el agua. El agua juega uno de los papeles más importantes para el mantenimiento de la vida en el planeta, su funcionalidad biológica la hace indispensable para la creación y la sobrevivencia de todos los seres vivos y como si esto por sí sólo no fuera lo suficientemente importante, el agua también es un pilar del desarrollo. En la publicación clásica de Bernard Frank, se destaca que el agua ha participado de manera muy importante en el origen de las civilizaciones, menciona a los Babilonios y a los Egipcios y describe como la infraestructura que permitía el almacenamiento y la transportación del agua fue la base del crecimiento y desarrollo de estos pueblos. Menciona que el agua además de ser fundamental para la vida, ya que el cuerpo humano contiene un 78% de agua, ha permitido el desarrollo económico y comercial. En los tiempos de las primeras civilizaciones, el agua, además de ser empleada para riego, servía como vía de transportación, por lo cual los pueblos se desarrollaron en las cercanías de las fuentes naturales de agua, posteriormente a través de la ingeniería, la infraestructura artificial sirvió para los mismos fines. (Frank, 1955)

El tratamiento y control de la contaminación del agua son de gran importancia para lograr una significativa mejoría en los mantos acuíferos que se encuentran en un nivel de contaminación alto y de la misma manera para evitar una afectación a aquellos que no han sido dañados de esta forma. En el estado de Puebla se ha considerado bastante el tema del estado actual de los recursos hídricos de los centros urbanos, pero poco se ha propuesto para el monitoreo, control y seguimiento de las descargas de residuos a los lagos, ríos etc. que ya están demasiado contaminados haciendo que su recuperación sea más lenta.

En el año 2013 la Secretaría de Desarrollo Rural, Sustentabilidad y Ordenamiento Territorial (SDRSOT) instaló *La red de Estaciones de Monitoreo para la Preservación, Conservación y Mejoramiento de la Calidad del Agua en la Cuenca del Alto Atoyac*

(REMPCMCA) la cual se conforma de 12 sitios de monitoreo a lo largo de 35 km del río Atoyac. Las estaciones de monitoreo miden los parámetros asociados a la calidad del agua como lo son: la temperatura, pH, conductividad, turbidez, potencial óxido-reducción y coeficiente de absorción espectral. Estas mediciones se realizan en intervalos de 15 minutos de forma continua, registrando los datos en un ordenador local, para posteriormente transmitir vía internet a un servidor donde se visualiza a través de una plataforma informática.

Adicionalmente, las estaciones cuentan con un sistema de muestreo automático acoplado al sistema de medición con la capacidad de muestrear y preservar hasta ocho litros de agua (ADASA, 2013).

Teniendo una gran cantidad de aguas residuales es necesario implementar un sistema de control, en la capital de Puebla, existen alrededor de 18 mil empresas y de acuerdo con la ley, éstas necesitan contar con un permiso para realizar sus descargas al drenaje público, lamentablemente solo el 17 por ciento cuentan con las autorizaciones correspondientes. Una de las razones principales del incumplimiento se debe a que implica inversiones y servicios.

Por ello la presidenta de la Comisión de Desarrollo Urbano y Medio Ambiente del ayuntamiento de Puebla, Myriam Arabian Couttolenc expreso que se enfocarán en que empresas clasificadas como “3 E” que incluye 16 giros: plástico, textiles, automotrices, metálicos, químicos, piel, entre otros, cumplan con la normatividad, pues estos son los que más contaminantes arrojan (Olivera, 2017).

Con ello se han desarrollado diferentes métodos para lograr una recuperación de este recurso pero debido a la insuficiente infraestructura, los altos costos que implican su uso y adquisición, la falta de mantenimiento y de personal capacitado, sólo 36 % de las aguas residuales generadas reciben tratamiento, lo cual crea la necesidad de desarrollar tecnologías para su depuración (Romero, 2009)

Con base en lo anterior, es indispensable contar con un sistema de monitoreo de los desechos que son arrojados a algún tipo de manto acuífero, en el caso del estado de Puebla específicamente en el río Atoyac.

Siendo este un problema grave ya que lagos y ríos se han convertido en fuentes de infección las alternativas para tratar agua son poco efectivas o no pueden dar una buena eficiencia si estos lugares siguen siendo contaminados, por ello es necesario incorporar una red de monitoreo que nos permita evaluar las características del agua en tiempo real que permita conocer las variaciones en algunos parámetros básicos de calidad del

agua del Río Atoyac para poder establecer qué tratamientos son los más efectivos para el saneamiento del mismo.

En este trabajo se propone el diseño y prueba de un prototipo que pueda servir como base para implementar una red de monitoreo que sea viable tanto en los resultados obtenidos, como en los costos de implementación, mantenimiento y personal que nos permita monitorear el agua que está en proceso de saneamiento.

Se esperaría que con la implementación del material propuesto para este trabajo reducirá el costo significativamente comparado con inversiones que se han hecho antes para este problema.

1.2. Objetivo

El presente trabajo tuvo como objetivo general y específicos los siguientes

1.2.1 Objetivo general

Construir un prototipo funcional de bajo costo para evaluar parámetros de calidad de agua.

1.2.2 Objetivos específicos

- 1.- Construir y programar un prototipo basado en Arduino que nos permita evaluar algunas características del agua a monitorear como lo son el pH, temperatura, oxígeno disuelto, conductividad y turbidez.
- 2.- Validar los datos obtenidos, comparándolos con pruebas de laboratorio y métodos estandarizados de calidad de agua.
- 3.- Estimar el costo y beneficios potenciales del prototipo.

1.3 Hipótesis

Los parámetros del agua: pH, temperatura, oxígeno disuelto, conductividad y turbiedad; son determinados a través de un prototipo de monitoreo basado en Arduino. Comparando el prototipo con equipos comúnmente utilizados en laboratorios haciendo un análisis técnico y económico, es notable la diferencia en el costo favoreciendo al prototipo, y confiable en cuanto a los datos obtenidos.

CAPITULO 2. MARCO TEÓRICO

Marco teórico

El agua es considerada como fuente de vida debido su función esencial en los procesos biológicos y a su importancia como elemento fundamental de desarrollo; sin embargo la desigualdad social respecto al acceso y disponibilidad del agua, aunada a la pérdida de la calidad para su utilización directa de la fuente han sido el origen de la crisis del agua en México.

Para el caso de México, la totalidad de habitantes del país 70% vive en zonas urbanas, el 89% cuenta con servicio de agua potable y 77% tiene servicio de alcantarillado, lo que indica, en términos proporcionales, que prácticamente la totalidad de los habitantes de las zonas urbanas cuentan con estos servicios, y quienes no disponen de ellos son las comunidades indígenas o rurales siendo el sector con menor poder adquisitivo. En el presente trabajo se exhibe la problemática y los componentes que atentan contra la sustentabilidad de este recurso, y la relación existente entre la disponibilidad y su gestión de este recurso natural (García, 2015).

Estrictamente hablando según la ley de la conservación de la materia de Lavoisier el volumen total de agua en el planeta no ha cambiado, sin embargo existe una percepción de incremento en la escasez del vital líquido en la mayoría de las regiones del planeta. Esta percepción tiene un sustento real, y es debido a los cambios en el ciclo hidrológico y a los problemas de contaminación que han reducido el volumen de agua dulce disponible en ciertas regiones.

Toledo menciona que solamente el 0.007% del agua del planeta se encuentra disponible para uso directo de los seres humanos siendo un total de

4,200 km³, esta cantidad dividida entre 6,000 millones de habitantes representa un volumen de 700 m³ por persona por año, sin embargo esta cifra promedio no refleja la disponibilidad real del agua para todos los habitantes del planeta debido a que el agua se distribuye de manera geográfica, no de manera administrativa, es decir, la distribución del agua depende de la altitud, latitud, clima entre otros factores ambientales y no de la manera en que los habitantes seleccionan los espacios territoriales para formar sus comunidades influenciados por intereses políticos y económicos (Toledo, 2002).

Anualmente México obtiene aproximadamente 1, 449,471 millones de metros cúbicos de agua en forma de precipitación. De esta agua, se estima que el 72.5% se evapora y regresa a la atmosfera, el 21.2% escurre por los ríos o arroyos, y el 6.3% restante se filtra al subsuelo de forma natural y recarga los acuíferos. Tomando en cuenta los fluidos

de salida (exportaciones) y de entrada (importaciones) de agua con los países vecinos, el país cuenta anualmente con 446,777 millones de metros cúbicos de agua dulce renovable.

La figura 2a muestra los componentes y valores que conforman el cálculo del agua renovable.

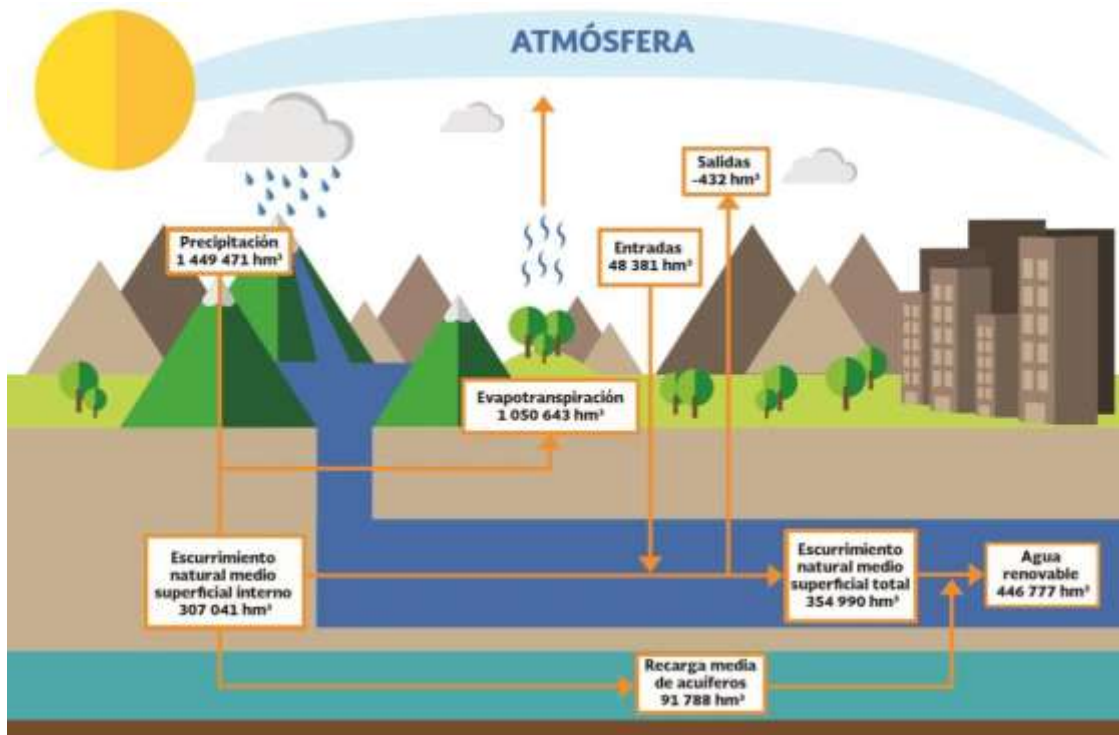


Figura 2a Valores anuales de los componentes del ciclo hidrológico en México, 2015 (CONAGUA, 2016b)

Los flujos de entrada representan el volumen de agua que escurre hacia país, generado en las cuencas transfronterizas que comparte México con sus países vecinos (Estados Unidos de América, Guatemala y Belice). Los flujos de salida representan el volumen de agua que México debe entregar a Estados Unidos de América conforme al “Tratado de Aguas” de 1994.

El agua renovable se debe analizar desde tres perspectivas:

- Distribución temporal: en México existen grandes variaciones del agua renovable a lo largo del año. La mayor parte de la lluvia ocurre en el verano, mientras que el resto del año es relativamente seco.
- Distribución espacial: en algunas regiones del país ocurre precipitación abundante y existe una baja densidad de población, mientras que en otras sucede lo contrario.

- Área de análisis: la problemática del agua y su atención es predominante de tipo local. Los indicadores calculados a gran escala esconden las fuertes variaciones que existen a lo largo y ancho del país.

Aguas superficiales

Los ríos y arroyos del país constituyen una red hidrográfica de 633 mil kilómetros de longitud, en la que destacan cincuenta y un ríos principales por los que fluye el 87% del escurrimiento superficial del país y cuyas cuencas cubren el 65% de la superficie territorial continental del país.

Principales lagos de México

En la siguiente figura se presentan algunos de los principales lagos de México en la zona centro del país.

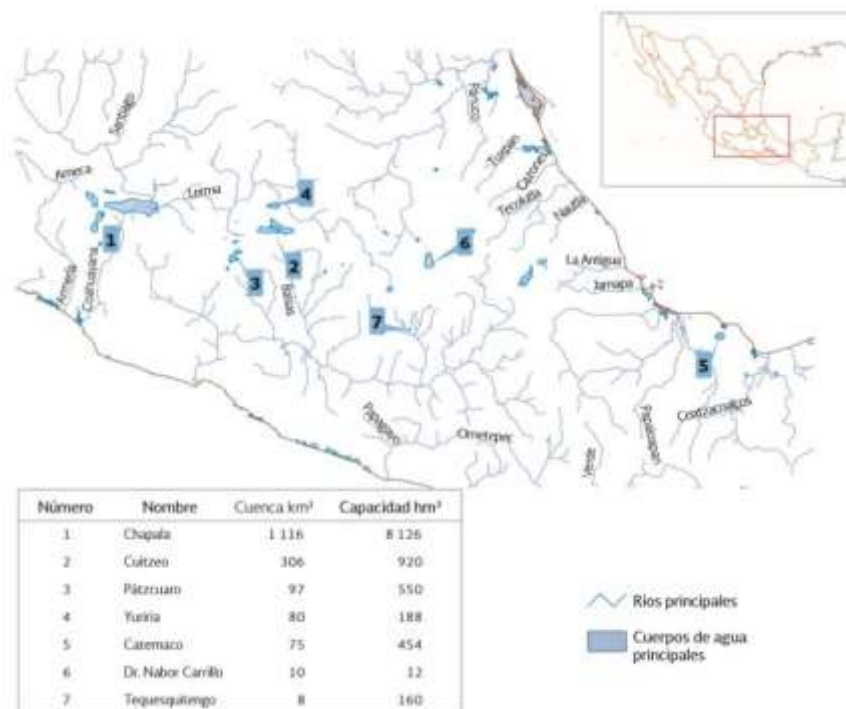


Figura 2b Principales lagos de la zona centro (CONAGUA, 2016b)

2.1 Contaminación del agua

La contaminación del agua en nuestros mantos acuíferos frecuentemente contiene un gran grupo de químicos que incluyen desde píldoras anticonceptivas, cremas solares, pesticidas y hasta petróleo, por mencionar algunos.

Además de la contaminación sintética, el agua dulce también es el punto final de nuestros desperdicios biológicos, en forma de aguas residuales, excrementos animales, y agua de lluvia mezclada con fertilizantes provenientes de jardines y granjas (NatGeo,

2011). Es importante resaltar que las descargas residuales principalmente se llevan a cabo en los ríos, los cuales se abren camino hacia los mares, costeras de océanos, causando una conexión dolorosamente obvia entre la tierra y el mar. En la última década la tecnología ha jugado un lugar importante al ofrecer soluciones como lo son las filtraciones y plantas de tratamiento cada vez más eficaces. Pero en contraparte, en partes del mundo en un nivel subdesarrollado, la infraestructura política, económica y técnica no son lo suficientemente capaces de eliminar los contaminantes del agua residual.

Sin embargo es difícil creer que en los países desarrollados, el 70% de los desperdicios industriales se vierten sin un tratamiento previo, contaminando así el suministro de agua potable.

2.1.1 Aguas residuales en México

Teniendo en cuenta que el agua residual se divide en “municipal” y “no municipal”, la primera refiriéndose al agua que proviene de la población y alcantarillas públicas, y la segunda provenientes de industrias.

El tratamiento de aguas residuales ha incrementado a lo largo de los últimos años gracias a las plantas de tratamiento en operación.

El progreso del caudal municipal tratado anualmente se puede apreciar en la gráfica 2.1a



Gráfica 2.1a Caudal de aguas residuales municipales tratadas a nivel nacional, m^3/s (CONAGUA, 2016).



Gráfica 2.1b Caudal de aguas residuales no municipales tratadas a nivel nacional, m^3/s (CONAGUA, 2016).

Por otra parte, en el año 2015, las plantas de tratamiento de aguas residuales industriales denominadas “no municipales”, trataron $70.5 m^3/s$ de aguas residuales en 2,832 plantas en operación a nivel nacional, como se observa en la gráfica 2.1b

A esta agua de desecho, se le denomina como residual. Por las raíces donde se genera, se define como la combinación de los residuos líquidos procedentes de residencias, instituciones públicas, establecimientos industriales y comerciales. En ocasiones, se incluyen a las aguas subterráneas, superficiales y pluviales (Metcalf, 1996).

2.2 Calidad de agua

La calidad de agua se determina mediante la caracterización física y química de muestras de agua y su comparación con normas y estándares de calidad. De esta forma se puede identificar si el agua es idónea para los requerimientos de calidad asociados a un uso determinado (ONU, 2016).

Pero en vista de la complejidad de los factores que determinan la calidad de agua y la gran variedad de parámetros utilizados para describir el estado de los cuerpos de agua en términos cuantitativos, es difícil proporcionar una definición simple de la calidad de agua (Singh, 2004). Además, la comprensión de la calidad de agua ha evolucionado con el transcurso de los años, a partir del incremento de requisitos de uso del agua, los avances en la tecnología que han permitido aumentar la capacidad de medir e interpretar las características del agua.

La figura 2.2 es un claro ejemplo de los parámetros que en su momento eran indispensables ser medidos en los países industrializados. Con el tiempo el principal parámetro a medir fue cambiando de acuerdo al tipo de industria que en cada año fue importante analizar en cuanto a su contaminación.

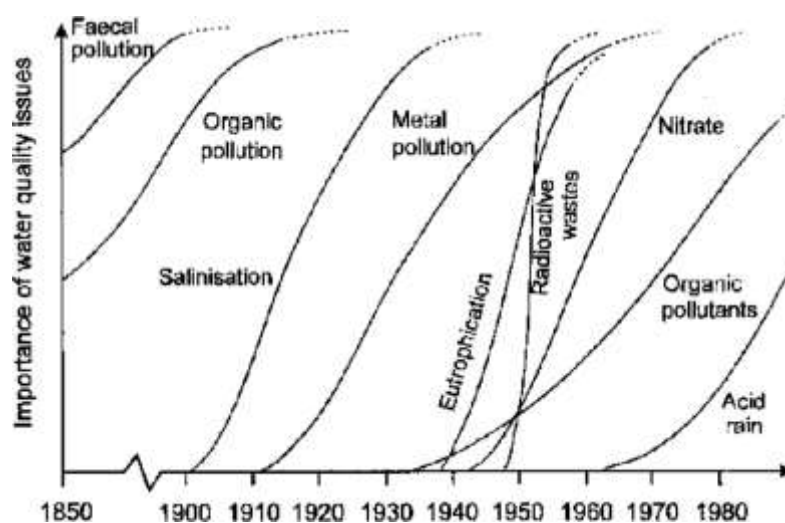


Figura 2.2 La secuencia de los problemas de calidad del agua que surgen en los países industrializados (Meybeck 1989).

Los términos de “monitoreo” y “evaluación” de calidad de agua se confunden con frecuencia y se usan como sinónimos (Bartram, 1996). Para los fines de este documento tomaremos las definiciones de Bartram, que se muestran en la tabla 2.2.

| Tabla 2.2 definición de: monitoreo y evaluación (Bartram, 1996). | |
|--|--|
| Término | Definición |
| Evaluación de calidad de agua. | Es el proceso general de avluación de la naturaleza física, química y biológica del agua en relación con la calidad natural. Los efectos humanos y los usos previstos; en particular los usos que pueden afectar la salud humana y la salud del propio ecosistema. |
| Monitoreo de calidad de agua. | Es la recopilación real de información en ubicaciones establecidas y en intervalos regulares con el fin de proporcionar los datos que pueden utilizarse para definir las condiciones actuales, establecer tendencias, etc. |

La evaluación de la calidad del agua incluye el uso de monitoreo para definir la condición del agua, para proporcionar la base para detectar tendencias y para proporcionar la información que permita el establecimiento de relaciones de causa y efecto. Los aspectos importantes de una evaluación son la interpretación y el informe de los resultados del monitoreo y la elaboración de recomendaciones para acciones futuras.

Por lo tanto, hay una secuencia lógica que consta de tres componentes: monitoreo, evaluación y administración en este orden. Además, también hay un ciclo de retroalimentación porque la administración inevitablemente requiere monitoreo de cumplimiento para hacer cumplir las regulaciones, así como evaluaciones a intervalos periódicos para verificar la efectividad de las decisiones de la administración. El objetivo principal del proyecto de monitoreo de la calidad del agua dulce global, GEMS / WATER, proporciona un ejemplo ilustrativo de la complejidad de la tarea de evaluación y su relación con el manejo (WHO, 1991):

- Proporcionar evaluaciones de la calidad del agua a los gobiernos, la comunidad científica y el público, sobre la calidad del agua dulce del mundo en relación con la salud de los ecosistemas humanos y acuáticos, y las preocupaciones ambientales globales, específicamente:
- Definir el estado de la calidad del agua;

- identificar y cuantificar las tendencias en la calidad del agua;
- definir la causa de las condiciones y tendencias observadas;
- identificar los tipos de problemas de calidad del agua que ocurren en áreas geográficas específicas; y
- proporcionar la información y las evaluaciones acumuladas en una forma que las agencias reguladoras y de gestión de recursos puedan utilizar para evaluar alternativas y tomar las decisiones necesarias.

2.3 Monitoreo de la calidad de agua

En el 2015, la Red Nacional de Monitoreo contaba con 4,999 sitios, distribuidos a lo largo y ancho del país, como se describe en la tabla 2.3a

| Red | Área | Sitios (número) |
|-------------------------|-------------------------------|-----------------|
| Superficial | Cuerpos de agua superficiales | 2706 |
| Subterránea | Cuerpos de agua subterráneos | 1065 |
| Estudios especiales | Cuerpos de agua superficiales | 32 |
| Costeros | Zonas costeras | 920 |
| Descargas superficiales | | 267 |
| Descargas subterráneas | | 9 |
| Total | | 4999 |

Adicionalmente a los parámetros fisicoquímicos y microbiológicos monitorizados por la Red, a partir de 2005 se han realizado monitoreos biológicos en algunas regiones del país, los cuales permiten evaluar la calidad de agua utilizando métodos sencillo y de bajo costo, tales como el índice de diversidad con organismos bentónicos.

2.4 Arduino

El proyecto Arduino, de origen italiano, pertenece a un movimiento denominado hardware de código abierto.

Arduino fue creado en el Ivrea Interaction Design Institute como un aparato sencillo para el prototipado, en un principio fue diseñado para estudiantes sin experiencia en electrónica y programación. Al presentarse ante la sociedad se tuvo la necesidad de ser adaptado acorde a las necesidades que exigían los nuevos usuarios, con el transcurso del tiempo el desafío era más grande, sin dejar a un lado que las tarjetas siguieron

siendo de código abierto, lo que proporciona a los usuarios fabricar de forma independiente y provisionalmente adaptarlos a sus exigencias particulares.

Las placas Arduino (Figura 2.4) pueden leer entradas mediante botones, sensores incluso mensajes de texto para luego convertirlos en una salida que puede ser encender un LED, un motor, activar una cámara etc. Para llevar a cabo este proceso se tienen que enviar un conjunto de instrucciones al microcontrolador utilizando un lenguaje de programación en este caso basado en el cableado y el software basado en el procesamiento.

La plataforma Arduino se le conoce como un entrenador de microcontrolador (pequeñas computadoras contenidas en un solo circuito integrado), para la cual hace uso de herramientas de software en PC para su programación. De tal forma que la programación del Arduino puede realizarse en un lenguaje de alto nivel como lo es C++, permitiendo manipular elementos de hardware externos con relativa facilidad, pues tanto el hardware como el entorno de desarrollo de software han sido diseñados considerando la facilidad de uso como una prioridad.

Por tanto, la plataforma Arduino es un proyecto tanto de software libre como de Hardware abierto, donde los autores hacen de conocimiento público todos los detalles de su implementación, ayudándose asimismo de otros proyectos de software libre para su creación.

Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.



Figura 2.4 Tarjeta de circuito Arduino UNO.

2.4.1 Open Source

La iniciativa Open Source (OSI, por sus siglas en inglés Open Source Initiative) celebra su vigésimo aniversario en 2018. La etiqueta de "código abierto" se creó en una sesión de estrategia celebrada el 3 de febrero de 1998 en Palo Alto, California. Ese mismo mes, se fundó la OSI. Como una organización global sin fines de lucro, OSI protege y promueve el software de fuente abierta, el desarrollo y las comunidades, defiende la libertad del software en la sociedad a través de la educación, la colaboración y la infraestructura, cuida la Definición de Open Source (OSD por sus siglas en inglés Open Source Definition) y evita el abuso de los ideales y la ética. Inherente al movimiento de código abierto (Open Source, 2018).

Open Source o también llamado "Código abierto" es un término que se utiliza para denominar a cierto tipo de software que se distribuye mediante una licencia que le permite al usuario, utilizar el código fuente del programa para estudiarlo, modificarlo y realizar mejoras en el mismo, pudiendo incluso hasta redistribuirlo.

A los programas que aceptan el acceso a su código de programación de los conoce como Open Source, lo cual permite ser modificado por otros programadores ajenos a la creación de los autores originales del software.

El software Open Source es aquel que presenta su código abierto, y el software libre lo cual permite ser descargado y distribuido de forma gratuita. Necesariamente tienen que cumplir con ambos requisitos para poder llamarse de esta forma, ya que existen software libre que no brindan acceso al código y no están considerados en este grupo (Rivera, 2015).

Cuando se brinda acceso al código fuente, la comunidad de programadores puede hacer sus aportes para solucionar eventuales fallos, incrementar la usabilidad y mejorar el programa a nivel general.

2.4.2 Hardware libre

De acuerdo a la asociación de Hardware de fuentes abiertas (OSHWA por sus siglas en inglés: Open Source Hardware Association) consiste en compartir los archivos de diseño de una pieza de hardware (máquina, herramienta, artículo útil u otra pieza tangible, mecánica o electrónica) para que otros puedan modificarla o fabricarla, inclusive con propósitos comerciales (OSHWA, 2018).

2.4.3 Software libre

Software significa que los usuarios tienen la libertad para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

Ventajas

Ahorro de costos, independencia respecto a empresas suministradoras, posibilidades de hacer adaptaciones a la medida del prototipo. Total independencia de un proveedor.

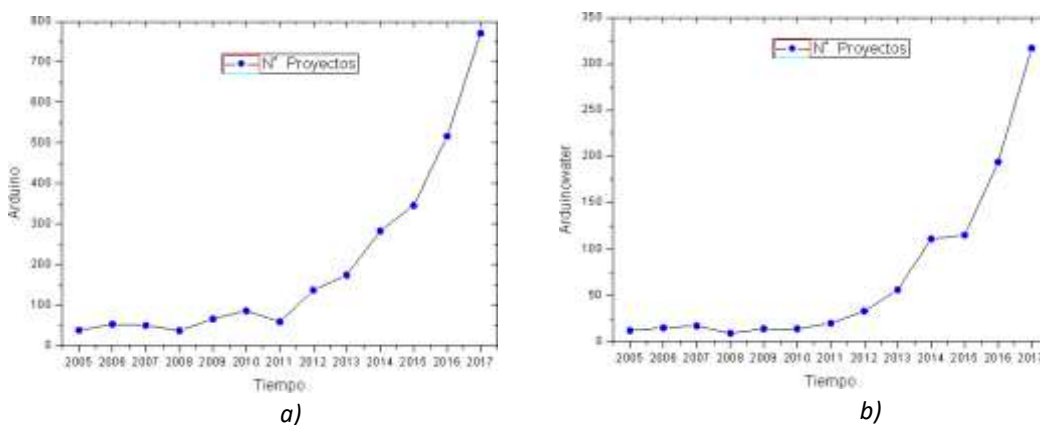
El usuario puede administrar libremente su crecimiento de acuerdo a las necesidades del usuario.

Desventajas

Interfaces graficas poco amigables. Además generalmente no se brinda soporte y buena documentación. Para su configuración se requieren conocimientos previos de funcionamiento del sistema operativo.

2.5 Tendencia de Arduino en proyectos de investigación

Arduino es una herramienta que con el paso de los años ha crecido de forma exponencial, gracias a que es un material que ha sido indispensable en proyectos de todo tipo, desde caseros hasta importantes trabajos científicos. En la gráfica 3 se representa la cantidad de proyectos científicos reportados en revistas arbitradas que se han llevado a cabo con Arduino en los últimos años. Las palabras clave que se utilizaron para buscar estos proyectos son “Arduino” y “Arduino wáter” ya que son temas de gran interés para este proyecto que se han dado a conocer en la plataforma ScienceDirect que es un centro líder de información de Elsevier para investigadores, profesores, estudiantes, profesionales de la salud y profesionales de la información. Combina publicaciones científicas, técnicas y de salud autorizadas y de texto completo (Elsevier, 2018).



Gráfica 2.5 Crecimiento de proyectos a) “Arduino”, b) “Arduino wáter”

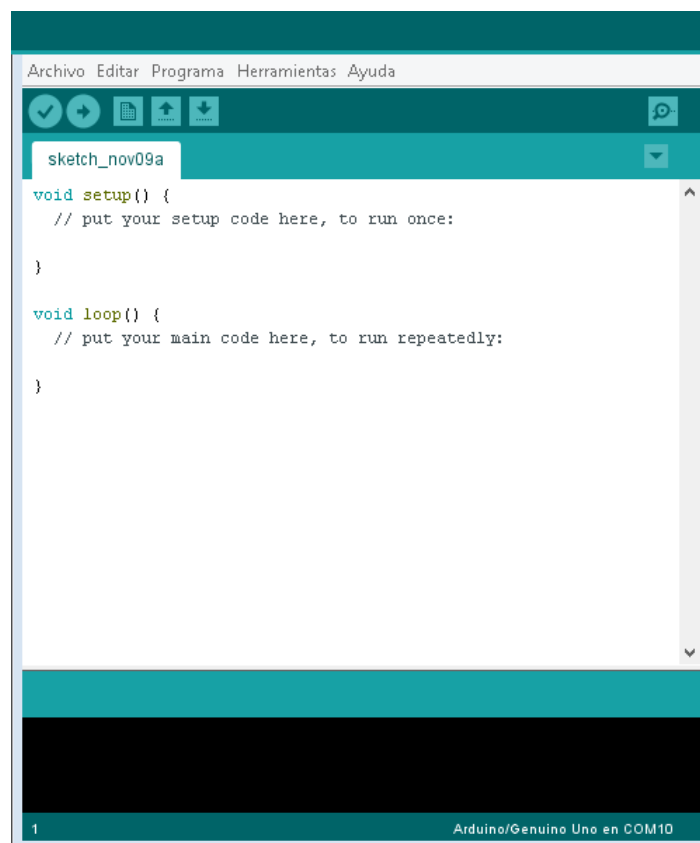
Como se puede observar con el paso del tiempo ha aumentado la cantidad de proyectos. Cabe mencionar que para febrero del año en curso (2018) se han rebasado ya los trabajos de los años 2013 y 2015 para Arduino y Arduino water, respectivamente.

Al trabajar con un material de esta índole (código abierto) es necesario respaldar la información y detalles del mismo, Creative commons que cuenta con una red de personal, juntas directivas y afiliados de todo el mundo ofrece licencias de copyright gratuitas y fáciles de usar, de una forma simple y estandarizada. Otorgando al público permisos para compartir y utilizar los materiales y las condiciones de trabajo de su elección.

2.6 Lenguaje y estructura de programación

El lenguaje de programación en Arduino deriva del lenguaje C++, que nos permite definir todas las operaciones que llevara a cabo el microcontrolador. Arduino ofrece un api o Core que facilitan la programación de los pines de entrada y salida y de los puertos de comunicación, así como otras librerías para operaciones específicas. El propio IDE ya incluye estas librerías de forma automática y no es necesario declararlas expresamente.

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes (Figura 2.6). Estas dos partes necesarias, o funciones, encierran bloques que contienen o instrucciones.



```
Archivo Editar Programa Herramientas Ayuda
sketch_nov09a
void setup() {
  // put your setup code here, to run once:
}
void loop() {
  // put your main code here, to run repeatedly:
}
1 Arduino/Genuino Uno en COM10
```

Figura 2.6a Pantalla principal Arduino. Estructura de programación

void setup()

Esta función es la primera que se ejecuta cuando Arduino da inicio al programa, y es efectuada una sola vez. Su propósito es llevar a cabo todas las tareas de inicialización antes que inicie el programa principal. Por ejemplo la configuración de los pines.

void loop()

Es ejecutada tras terminar la función setup(), y en caso que la función loop() termine, esta es re-ejecutada continuamente por el Arduino hasta que se reinicie, se re programe o se apague el suministro. Su propósito es llevar a cabo el programa principal mientras el Arduino permanezca operando.

```
void setup()
{
  pinMode(pin, OUTPUT); // configura el "pin" como salida
  digitalWrite(pin, HIGH); // pone el "pin" en estado HIGH
}

void loop()
{
  digitalWrite(pin, HIGH); // pone en uno (on, 5v)
  delay(1000); // el 'pin' espera un segundo (1000 ms)
  digitalWrite(pin, LOW); // pone en cero (off, 0v.) el 'pin'
  delay(1000);
}
```

Figura 2.6b Ejemplo: función setup (), loop ().

Arduino, a diferencia de una computadora personal, no cuenta con un sistema operativo al cual retornar en caso que los programas terminen. Por tanto los programas tienen que ser ejecutados de manera permanente mientras el sistema cuente con suministro eléctrico. Sin embargo, si el usuario así lo desea, puede detener la ejecución del programa por medio de atraparlo en un lazo infinito. Sentencias como for(;;); o bien while(1){ } pueden ser usadas en ese caso.

Existen diferentes funciones de acuerdo a la clase de hardware con que se interactúa, de las cuales las que se presentan a continuación son las referentes a las entradas y salidas digitales:

- pinMode (numero_pin, modo): Permite establecer la modalidad de los pines ya sea como entradas o como salidas. Los nombres de pines validos son los números del 0 al 13 y los pines analógicos del A0 al A5, mientras que los modos válidos son INPUT

(entrada) y OUTPUT (salida). Nótese que si se configuran los pines del A0 al A5 con esta función, los mismos se pueden usar como cualquier otra entrada o salida digital.

- `digitalRead(numero_pin)`: Efectúa la lectura de un pin digital que haya sido configurado como entrada, devolviendo su valor lógico ya sea como la constante LOW (0) o la constante HIGH (1).
- `digitalWrite(numero_pin, valor)`: Efectúa la escritura sobre un pin digital que haya sido configurado como salida. El valor puede especificarse como la constante LOW (0) o la constante HIGH (1).

2.7 Parámetros básicos de calidad del agua

Los procesos disponibles para mejorar la calidad de las aguas son muchos y muy diferentes, evidentemente en este trabajo es de gran interés aquellos que tienen que ver con la mejora de ríos y lagos, pero los parámetros utilizados para conocer y evaluar la calidad de agua son importantes y debemos describirlos.

Los parámetros físicos son los que definen las características del agua que responden a los sentidos de la vista, del tacto, gusto y olfato, como pueden ser los sólidos suspendidos, turbidez, color, sabor, olor, conductividad, temperatura entre otros.

El agua es llamada el solvente universal y los parámetros químicos están relacionados con la capacidad del agua para disolver diversas sustancias entre las que podemos mencionar a los sólidos disueltos totales, alcalinidad, dureza, fluoruros, metales, materias orgánicas y nutrientes.

Los parámetros biológicos son indicativos de la contaminación orgánica y biológica; tanto la actividad natural como la humana contribuyen a la contaminación orgánica de las aguas.

Ejemplos: demanda biológica de oxígeno, demanda química de oxígeno, gases disueltos.

2.7.1 Turbidez

La turbidez en el agua es causada por material suspendido y coloidal, como la arcilla, limo, materia orgánica e inorgánica finalmente dividida, plancton y otros organismos microscópicos. (APHA-AWWA-WEF, 2005)

La turbidez es una expresión de la propiedad óptica que hace que la luz se disperse y absorba en lugar de transmitirse en línea recta a través de la muestra. La correlación de la turbidez con la concentración de peso de la materia suspendida es difícil porque el tamaño, la forma y el índice de refracción de las partículas también afectan las

propiedades de dispersión de la luz de la suspensión. Las partículas ópticamente negras, como las de carbón activado, pueden absorber la luz y aumentar efectivamente las mediciones de turbidez.



Figura 2.7.1 *Turbidez*

Históricamente, el método estándar para la determinación de la turbidez se ha basado en el turbidímetro Jackson, sin embargo, el valor de turbidez más bajo que se puede medir directamente en este dispositivo es 25 Unidades Jackson de turbiedad (UJT). Debido a que las turbiedades del tratamiento del agua mediante procesos convencionales de separación de partículas de fluidos generalmente caen dentro del rango de 0 a 1, se desarrollaron métodos indirectos secundarios para estimar la turbidez. Los nefelómetros electrónicos son los instrumentos preferidos para la medición de la turbidez.

La mayoría de los turbidímetros comerciales diseñados para medir bajas turbiedades dan indicaciones comparativamente buenas de la intensidad de la luz dispersa en una dirección particular, predominantemente en ángulo recto con la luz incidente. Los turbidímetros con detectores de luz dispersa ubicados a 90° del haz incidente se llaman nefelómetros. Los nefelómetros son relativamente poco afectados por pequeñas diferencias en los parámetros de diseño, por lo tanto, se especifican como el instrumento estándar para la medición de bajas turbiedades.

Es preferible utilizar el método nefelométrico a los métodos visuales ya que su precisión, sensibilidad y aplicación se lleva a cabo en un amplio rango de turbidez. Los resultados de la medición nefelométrica se miden en unidades nefelométricas de turbiedad (UNT).

2.7.1.1 *Almacenamiento de la muestra*

Se recomienda determinar la turbidez tan pronto como sea posible después de tomar la muestra. Agitar suavemente todas las muestras antes del examen para garantizar una

medida representativa. Refrigerar o enfriar a 4°C, para minimizar la descomposición microbiológica de sólidos, si se requiere almacenamiento. Para obtener los mejores resultados, medir la turbidez inmediatamente sin alterar las condiciones originales de la muestra, como la temperatura o el pH. (APHA-AWWA-WEF, 2005)

2.7.1.2 Principio

El principio de este método se basa en una comparación de la intensidad de la luz dispersada por la muestra en condiciones definidas con la intensidad de la luz dispersada por una referencia estándar bajo las mismas condiciones. Cuanto mayor sea la intensidad de la luz dispersa, mayor será la turbidez. El polímero de formazina se usa como la suspensión primaria de referencia estándar. La turbidez de una concentración especificada de suspensión de formazina se define como 4000 UNT. (APHA-AWWA-WEF, 2005)

La turbidez puede determinarse para cualquier muestra de agua que esté libre de desechos y asentamiento rápido de sedimentos gruesos. Los objetos de vidrio sucios y la presencia de burbujas de aire dan falsos resultados.

2.7.1.3 Equipo

Nefelómetro de laboratorio o proceso: consiste en una fuente de luz para iluminar la muestra y uno o más detectores fotoeléctricos con un dispositivo de lectura para indicar la intensidad de la luz dispersada a 90° con respecto a la trayectoria de la luz incidente. La sensibilidad del instrumento permite detectar diferencias de turbidez de 0.02 NTU o menos, en el rango más bajo en aguas con una turbidez de menos de 1 UNT. Varios rangos pueden ser necesarios para obtener cobertura adecuada y sensibilidad suficiente para las turbiedades. (APHA-AWWA-WEF, 2005)

Reportar la turbidez en los siguientes rangos.

| Tabla 2.7.1 Rangos de turbidez [UNT] (APHA-AWWA-WEF, 2005) | |
|--|-------------------------------------|
| <i>Rango de turbidez [UNT]</i> | <i>Reporte al más cercano [UNT]</i> |
| 0-1.0 | 0.05 |
| 0.1-10 | 0.1 |
| 10-40 | 1 |
| 40-100 | 5 |
| 100-400 | 10 |
| 400-1000 | 50 |
| >1000 | 100 |

2.7.2 Temperatura

Las lecturas de temperatura se utilizan en el cálculo de varias formas de alcalinidad, en los estudios de la saturación y la estabilidad con respecto a carbonato de calcio, en el cálculo de la salinidad, y en las operaciones de laboratorio en general. Las plantas industriales a menudo requieren datos sobre la temperatura del agua para el uso de proceso o cálculos de transmisión de calor.

Como mínimo, el termómetro debe tener una escala marcada por cada 0.1° C, con marcas grabadas en el cristal capilar. El termómetro debe tener una capacidad térmica mínima para permitir el rápido equilibrio. Controlar periódicamente el termómetro contra una precisión certificada por National Institute of Standards and Technology (NIST, antes National Bureau of Standards). (APHA-AWWA-WEF, 2005)

2.7.3 pH

La medición del pH es una de las pruebas más importantes y más utilizadas en la química del agua. Prácticamente cada fase de suministro de agua y tratamiento de aguas residuales, por ejemplo, neutralización ácido-base, ablandamiento del agua, precipitación, coagulación, desinfección y control de la corrosión, depende del pH. El pH se usa en mediciones de alcalinidad y dióxido de carbono y muchos otros equilibrios ácido-base. A una temperatura dada, la intensidad del carácter ácido o básico de una solución se indica por el pH o la actividad de iones de hidrógeno. La alcalinidad y la acidez son las capacidades neutralizadoras de ácidos y bases de un agua y generalmente se expresan en miligramos de Carbonato de calcio ($CaCO_3$) por litro.

(APHA-AWWA-WEF, 2005)

El pH tal como lo define Sorensen es $-\log[H^+]$; es el factor de "intensidad" de la acidez. El agua pura está ligeramente ionizada y en equilibrio, el producto de iones es

$$\begin{aligned} [H^+][OH^-] &= K_w \\ &= 1.01 * 10^{-14} \text{ a } 25^\circ C \end{aligned} \quad (1)$$

y

$$\begin{aligned} [H^+] &= [OH^-] \\ &= 1.005 * 10^{-7} \end{aligned}$$

donde

$$[H^+] = \text{actividad de iones hidrogeno, } \frac{\text{moles}}{L}$$

$$[OH^-] = \text{actividad de iones hidroxilo, } \frac{\text{moles}}{L}, \text{ y}$$

$K_w = \text{ion producto de agua.}$

Debido a las interacciones iónicas en todas las soluciones menos diluidas, es necesario utilizar la "actividad" de un ion y no su concentración molar. El uso del término pH supone que se está considerando la actividad del ion hidrógeno a_{H^+} . El aproximado equivalente a molaridad, $[H^+]$ puede presumirse solo en soluciones muy diluidas (fuerza iónica < 0.1).

Una escala logarítmica es conveniente para expresar una amplia gama de actividades iónicas. La siguiente ecuación en forma logarítmica y corregida para reflejar la actividad es:

$$(-\log_{10} a_{H^+}) + (-\log_{10} a_{OH^-}) = 14$$

o

$$pH + pOH = pK_w$$

donde

$$pH = \log_{10} a_{H^+} \text{ y}$$

$$pOH = \log_{10} a_{OH^-}.$$

La ecuación 2 establece que a medida que el pH aumenta, el pOH disminuye de forma correspondiente y viceversa porque pK_w es constante para una temperatura dada. A 25 °C, pH 7.0 es neutro, las actividades de los iones hidrógeno e hidroxilo son iguales, y cada una corresponde a una actividad aproximada de 10^{-7} moles/L . El punto neutro depende de la temperatura y tiene un pH de 7.5 a 0 °C y un pH de 6.5 a 60 °C. El valor de pH de una solución altamente diluida es aproximadamente el mismo que el logaritmo común negativo de la concentración de iones de hidrógeno. Las aguas naturales generalmente tienen valores de pH en el rango de 4 a 9, y la mayoría son ligeramente básicas debido a la presencia de bicarbonatos y carbonatos de los metales alcalinos y alcalinotérreos.

2.7.4 Oxígeno disuelto

Los niveles de oxígeno disuelto (DO) en aguas naturales y residuales dependen de las actividades físicas, químicas y bioquímicas en el cuerpo de agua. El análisis de DO en agua es una prueba clave en el control del proceso de contaminación y tratamiento de residuos. (APHA-AWWA-WEF, 2005)

Se describen dos métodos para el análisis de DO:

1. Winkler o iodométrico y sus modificaciones
 - Modificación ácida,
Modificación permanganato,
 - Modificación de floculación de alumbre,
 - Modificación de floculación de sulfato de cobre-ácido sulfámico
2. Electrodo de membrana

El método Winkler es un procedimiento titrimétrico basado en la propiedad oxidante de DO mientras que el procedimiento de electrodo de membrana se basa en la tasa de difusión de oxígeno molecular a través de una membrana. La elección del procedimiento depende de las interferencias presentes, la precisión deseada.

2.7.5 Conductividad

La conductividad (k), es una medida de la capacidad de una solución acuosa para transportar una corriente eléctrica. Esta habilidad depende de la presencia de iones; en su concentración total, movilidad y valencia; y en la temperatura de medición. Las soluciones de la mayoría de los compuestos inorgánicos son conductores relativamente buenos. (APHA-AWWA-WEF, 2005)

Los instrumentos estándares para su medición son los siguientes:

- a. Instrumentos de conductividad autónomos: utilice un instrumento capaz de medir la conductividad con un error que no exceda el 1% o $1 \mu mho/cm$, el que sea mayor.
- b. Termómetro, capaz de leerse con una precisión de $0.1^\circ C$ y cubrir el rango de 23 a $27^\circ C$. Muchos medidores de conductividad están equipados con un sensor de temperatura automático.
- c. . Celda de conductividad:
 1. Tipo de electrodo de platino: las células de conductividad que contienen electrodos platinizados están disponibles en forma de pipeta o de inmersión. La elección de celda depende del rango esperado de conductividad.
 2. Tipo de electrodo no platino: utilice celdas de conductividad que contengan electrodos construidos con metales comunes duraderos (acero inoxidable, entre otros) para monitoreo continuo y estudios de campo.

Para más detalles de los métodos estándares para medir los diferentes parámetros mencionados anteriormente consultar las referencias correspondientes.

CAPITULO 3. METODOLOGÍA

Metodología

Como podemos observar en la figura 3, se fijó una secuencia de pasos a seguir para poder cumplir satisfactoriamente este proyecto.

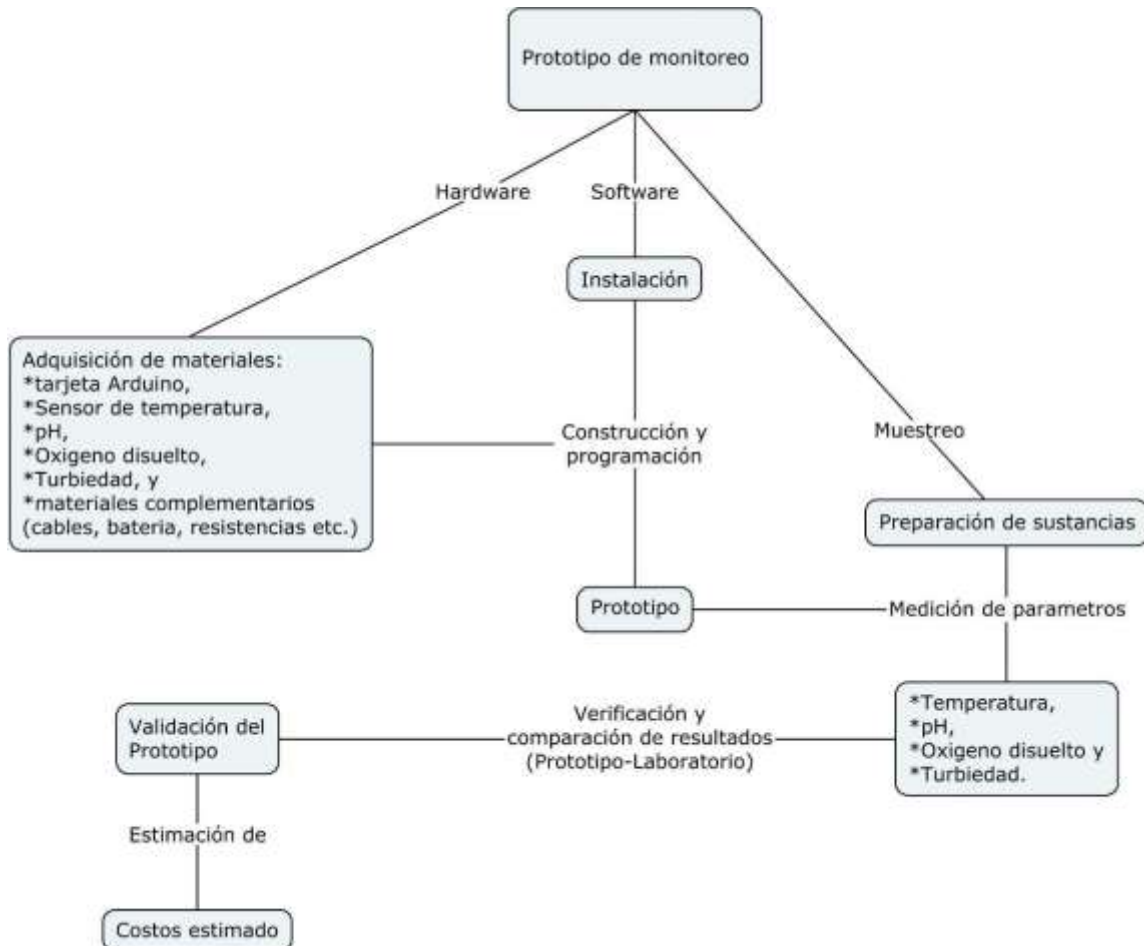


Figura 3 Metodología para la elaboración del prototipo de monitoreo de calidad de agua.

A continuación se presenta el desarrollo de cada etapa del proyecto.

3.1 Adquisición de materiales

Para adquisición de los materiales, se realizó una extensa búsqueda de proveedores para los diferentes componentes del proyecto.

Como se abarco en el capítulo anterior existe una gran variedad de tarjetas Arduino ya que varían principalmente dependiendo del uso que se le valla a dar. Para este proyecto se optó por adquirir la tarjeta Arduino UNO. Para los sensores el trabajo fue más profundo, ya que se corrobora la compatibilidad de ellos con Arduino. En cuanto a los materiales complementarios como lo son cables, resistencias, cautiñ, protoboar

etc. Fue necesario revisar las condiciones de cada sensor y tarjeta Arduino para poder adquirirlos.

3.2 Instalación del software

Para poder llevar a cabo correctamente la instalación de Arduino, se comienza por visitar la página oficial de Arduino, inmediatamente damos click en la pestaña Software en la cual nos permitirá visualizar la disponibilidad de archivos para las condiciones y particularmente el sistema operativo con que contemos (Figura 3.1). En este caso se instaló el archivo compatible con Windows 8.1



Figura 3.2a Descarga de archivo para la instalación de Arduino.

Una vez realizada la descargada del programa, se procede a llevar a cabo la instalación, lo cual se realiza de una manera sencilla siguiendo las indicaciones que nos muestra el instalador.

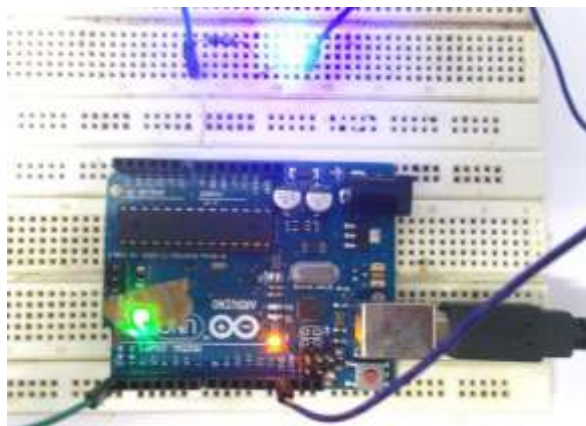


Figura 3.2b Programa ejemplo blink: encender y apagar un led.

Para terminar de manera segura con la instalación, se corrobora su funcionamiento realizando una prueba con un programa ejemplo que contiene el software llamado blink que permite prender y apagar un led, en este caso el led ya lo trae incluido la Tarjeta Arduino UNO (led 13).

3.3 Sensores y su funcionamiento

Se denomina sensor a un “sistema de medida” formado por el conjunto, transductor, sensor y acondicionador de señal, y que permite la medición de valores de características físicas, químicas, mecánicas, etc. Para tal objetivo adapta la señal obtenida desde un transductor que suele ser en el orden de lo mili voltios (mV) a valores más comprensibles según el objetivo del sistema general a implementar.

3.3.1 Sensor de Turbidez

El sensor de turbidez Arduino detecta la calidad del agua al medir los niveles de turbidez. Utiliza la luz para detectar partículas suspendidas en el agua al medir la transmitancia de la luz y la tasa de dispersión, que cambia con la cantidad de sólidos suspendidos totales (SST) en el agua. A medida que aumentan los SST, aumenta el nivel de turbidez del líquido.



Figura 3.1.1a Sensor de turbidez SKU: SEN0189.

Los sensores de turbidez se utilizan para medir la calidad del agua en ríos y arroyos, las mediciones de aguas residuales y efluentes, la instrumentación de control para el asentamiento de estanques, la investigación del transporte de sedimentos y las mediciones de laboratorio. Este sensor de líquido proporciona modos de salida de señal analógica y digital. El umbral es ajustable cuando está en modo de señal digital. Puede seleccionar el modo de acuerdo con su MCU.

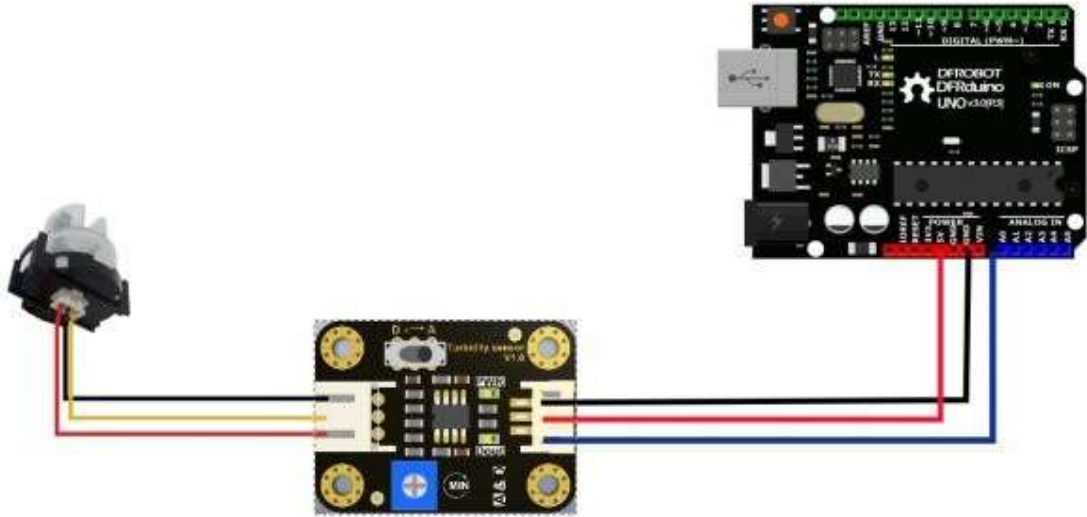


Figura 3.1.1 Diagrama de conexión del sensor de Turbidez: SEN0189

En la figura 3.1.1 se muestra la conexión correcta del sensor a la tarjeta Arduino UNO.

3.3.2 Sensor de Temperatura

El sensor de temperatura DS10B20 es un dispositivo que se comunica de forma digital. Cuenta con tres terminales: Vcc, GND y el pin Data. Este sensor utiliza comunicación OneWire, este protocolo permite enviar y recibir datos utilizando un solo cable, a diferencia de la mayoría de los protocolos que requieren dos cables.

Esta es una versión impermeable del sensor de temperatura Arduino DS18B20. Útil para cuando necesitas medir algo lejos, o en condiciones húmedas. Si bien el sensor es bueno hasta 125°C, el cable está revestido en PVC, por lo que sugerimos mantenerlo por debajo de 100 °C. Debido a que son digitales, no se produce ninguna degradación de la señal, incluso en largas distancias.



Figura 3.1.2a Sensor de temperatura DS10B20 impermeable.

Debido a que cada DS18B20 contiene un número de serie único de silicón, pueden existir múltiples DS18B20 en el mismo bus 1-Wire. Esto permite colocar sensores de temperatura en muchos lugares diferentes.

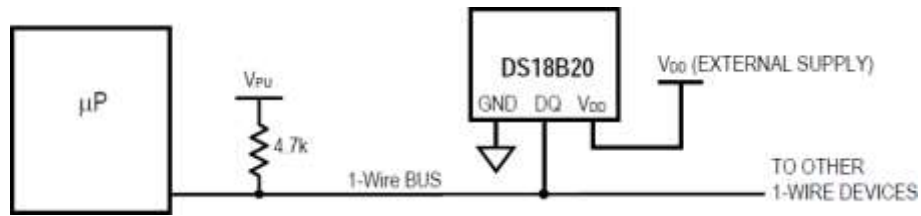


Figura 3.1.2b Alimentación del DS18B20 con una fuente externa

Para el correcto funcionamiento del sensor hay que poner una resistencia de 4.7K del pin de Datos y Vcc, Normalmente este sensor viene blindado en un cable largo para aplicaciones donde es necesario sumergirlo en líquidos u otras sustancias. Esta presentación del sensor solo trae 3 terminales o cables de conexión, El pin de Vcc es el cable Rojo, GND es el cable Negro y el Cable de datos puede ser de color Amarillo o Blanco.

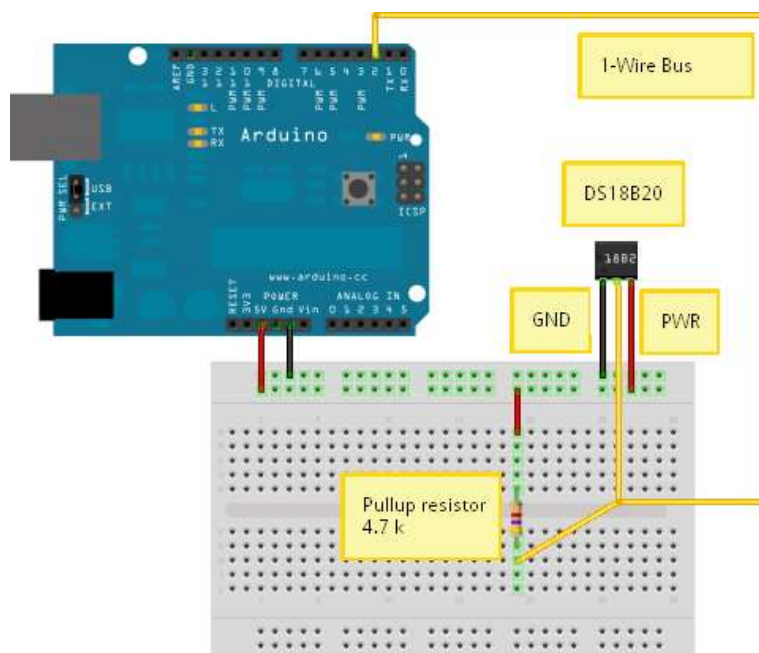


Figura 3.1.1 Diagrama de conexión del sensor de temperatura DS18B20.

3.3.3 Sensor de pH

El medidor de pH SKU: SEN0161 fue diseñado especialmente para los controladores Arduino. Contiene un LED que funciona como indicador de energía, un conector BNC y una interfaz de sensor pH. Para poder usarlo se conecta el sensor de pH con el conector BNC y conectamos la interfaz pH en el puerto de entrada analógica de cualquier controlador Arduino.

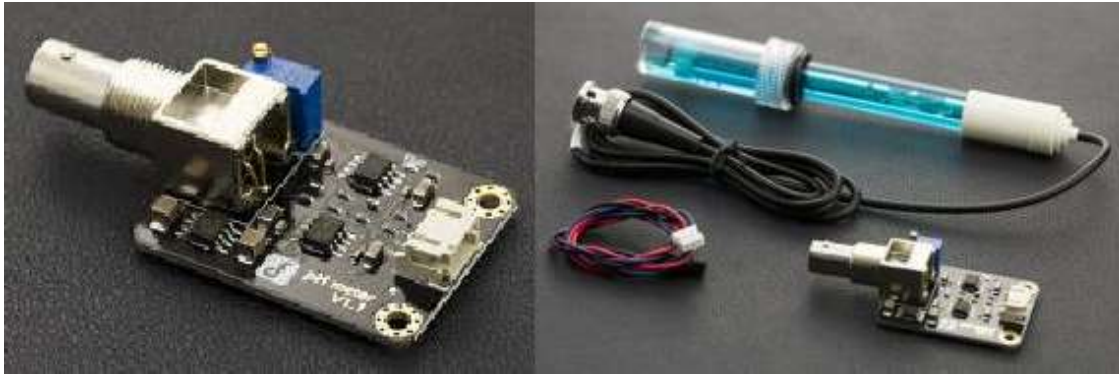


Figura 3.3.3a Sensor de pH SKU: SEN0161.

Los equipos se conectan de acuerdo a la figura 3.3.3b, es decir el electrodo de pH se inserta al conector BNC de la placa del medidor de pH, y luego use las líneas de conexión, la placa del medidor de pH se conecta a un puerto analógico de la tarjeta Arduino.

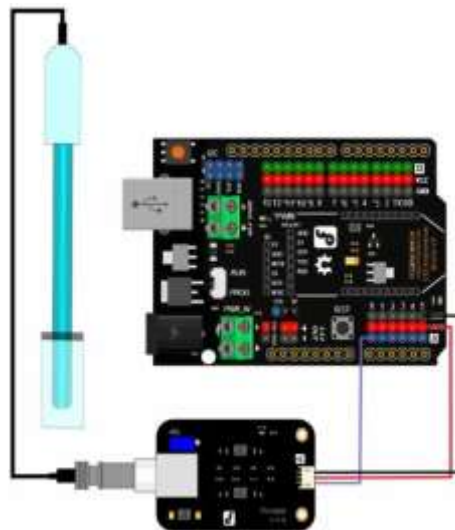


Figura 3.3.3b Diagrama de conexión del sensor de pH SKU:SEN0161.

3.3.4 Sensor de Oxígeno disuelto

Este sensor de Oxígeno disuelto SKU: SEN0237 se utiliza para medir el oxígeno disuelto en agua. Se aplica ampliamente en sectores como lo son la agricultura, el monitoreo del medio ambiente, las ciencias naturales etc. Contiene una sonda galvanizada que no necesita tiempo de polarización lo que permite estar disponible en cualquier momento. La solución de llenado y la tapa de la membrana son reemplazables, lo que conlleva un bajo costo de mantenimiento.

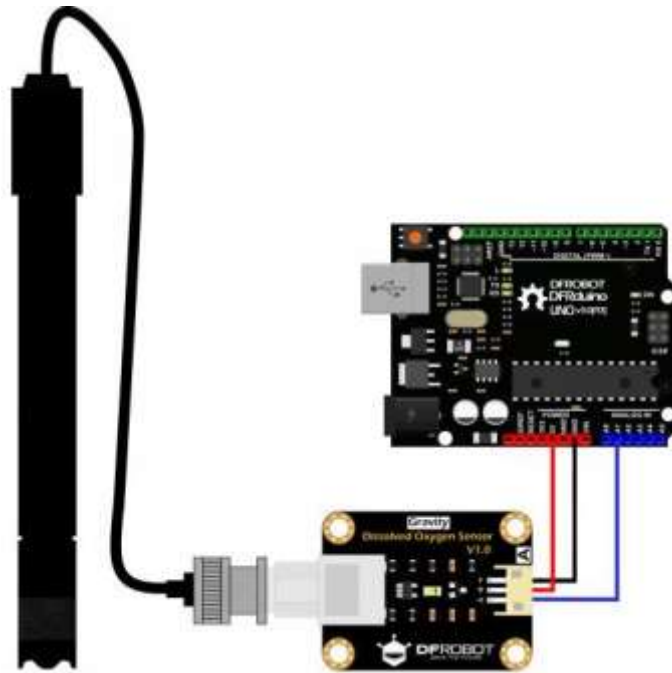


Figura 3.3.4a Diagrama de conexión del sensor de Oxígeno disuelto

Para una nueva sonda, primero se debe agregar 0.5 mol/L de solución de NaOH en la tapa de la membrana aproximadamente dos tercios de su capacidad máxima como solución de relleno. En la figura 3.3.4b se detalla el procedimiento de llenado de la sonda con solución de NaOH.

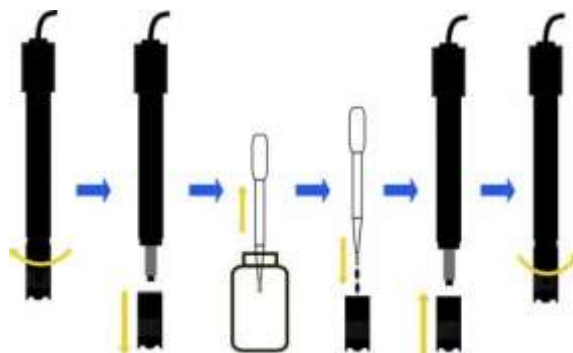


Figura 3.3.4b Preparación de sonda nueva del sensor de pH SKU:SEN0161.

3.3.5 Sensor de Conductividad

El sensor de conductividad eléctrica SKU: DFR0300 se utiliza especialmente para evaluar la calidad del agua, que se utiliza a menudo en cultivos, acuicultura, etc.

La biblioteca de software utiliza un método de calibración de dos puntos, y puede identificar automáticamente la solución de búfer estándar, de manera simple y conveniente.

La conductividad es el recíproco de la resistividad, que está relacionada con la capacidad del material para transportar la corriente. En el líquido, el recíproco de la resistencia es la conductividad que es la medida de su capacidad para conducir electricidad. La conductividad es un parámetro importante de la calidad del agua. Puede reflejar la extensión de los electrolitos presentes en el agua.

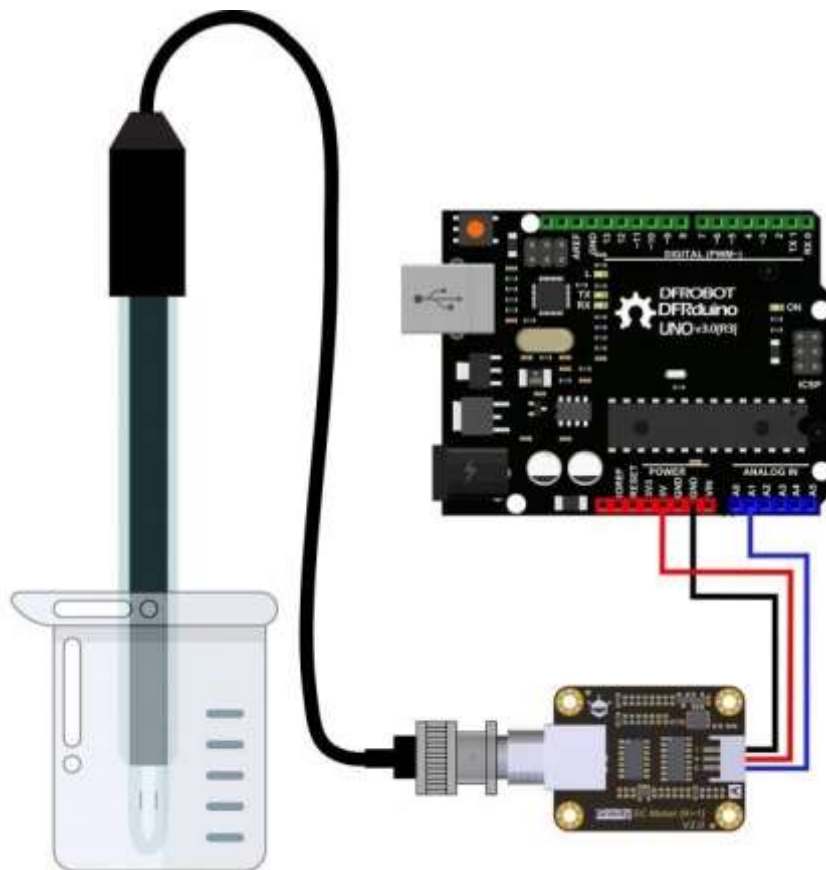


Figura 3.3.5 Diagrama de conexión del sensor de Oxígeno Disuelto SKU:DFR0300

CAPITULO 4.RESULTADOS

4.1 Pruebas y comparación (prototipo- laboratorio)

Los sensores fueron armados y probados inicialmente por separado para verificar su buen funcionamiento, posteriormente se agruparon para formar un solo equipo.

4.1.1 Sensor de turbidez

Para el prototipo se adquirió el sensor de turbidez SKU: SEN0189 el cual fue armado y adaptado de la siguiente manera:

La conexión del sensor se llevó a cabo de acuerdo con la figura 3.1.1b.

El sensor es capaz de funcionar en modo analógico o en modo digital ajustando en interruptor que controla esta señal.

- La señal analógica de salida (A), toma un valor de tensión e irá disminuyendo a medida que el valor de la turbidez aumente en el líquido.
- La señal digital (D), puede tomar valores de señal a nivel alto o bajo, en función de si están por encima o por debajo del umbral marcado por el potenciómetro.

En este caso ajustamos el interruptor en la señal analógica. Figura 4.1.1a



Figura 4.1.1a Interruptor en ajuste analógico.

Una vez armado el sensor, se procede a trabajar con el código, el cual partimos del “código ejemplo” del proveedor.

```
void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0); // read the :
  float voltage = sensorValue * (5.0 / 1024.0);
  Serial.println(voltage); // print out the valu
  delay(500);
}
```

Figura 4.1.1b Código ejemplo DFRobot.

Como se puede observar el “código ejemplo”, nos devuelve como resultado el voltaje que es medido con el sensor.

Para poder realizar la conversión de voltaje (mili Voltios) a Unidades Nefelométricas de Turbidez (NTU). Realizamos una gráfica del voltaje que nos proporciona el sensor (prototipo) contra las NTU de un turbidímetro comercial (figura 4.1.1c).



Figura 4.1.1c Turbidímetro HACH 2100N IS.

Para las pruebas se utilizó el siguiente material:

- Turbidímetro HACH y accesorios
- 5 vasos de precipitados de 150 mL.
- 1 Vaso de precipitados de 500 mL.
- 1 probeta 50 mL.
- Café soluble
- Computadora
- Sensor de turbidez

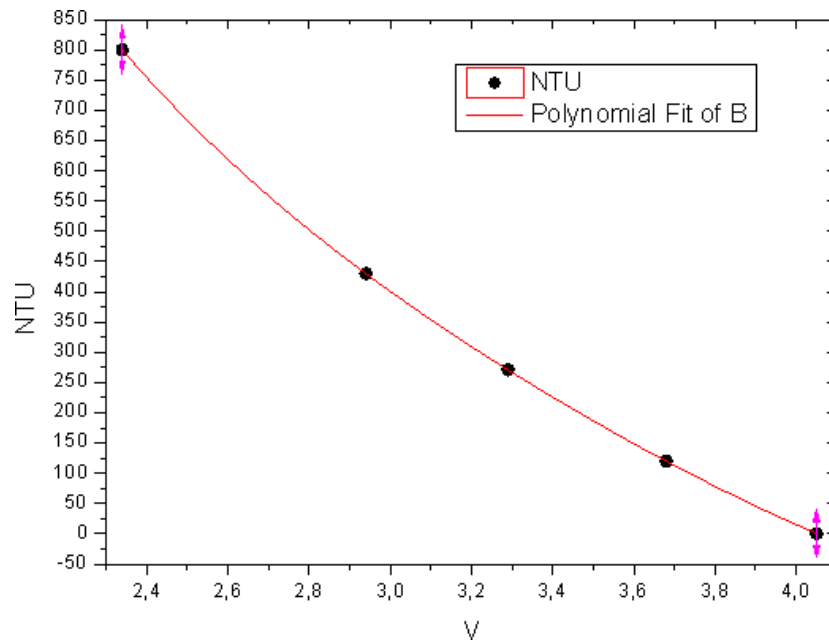
Siguiendo las condiciones y pasos apropiados para la medición de la turbidez anteriormente aclaradas.

Practica 1 Sensor sin modificaciones.

Preparamos las siguientes soluciones y obtuvimos el voltaje con ayuda de nuestro sensor y la turbiedad con el equipo de laboratorio (Tabla 4.1.1a).

| Tabla 4.1.1a NTU del agua a diferentes concentraciones de café | | | | | | | | | | |
|--|-----------|------|------|------|------|-------------------|-------|-------|-------|-------|
| ramos de café | Prototipo | | | | | Turbidímetro HACH | | | | |
| | V1 | V2 | V3 | V4 | V5 | NTU1 | NTU2 | NTU3 | NTU4 | NTU5 |
| 0.0 | 4.03 | 4.05 | 4.05 | 4.05 | 4.06 | 0.714 | 0.715 | 0.716 | 0.715 | 0.715 |
| 0.1 | 3.68 | 3.69 | 3.67 | 3.69 | 3.67 | 119 | 120 | 120 | 120 | 121 |
| 0.2 | 3.3 | 3.29 | 3.31 | 3.26 | 3.29 | 271 | 271 | 271 | 272 | 270 |
| 0.3 | 2.93 | 2.91 | 2.94 | 2.96 | 2.96 | 431 | 432 | 428 | 429 | 430 |
| 0.5 | 2.32 | 2.34 | 2.34 | 2.35 | 2.35 | 800 | 800 | 800 | 800 | 800 |

Después de 5 repeticiones tomamos el promedio de ellas para el Voltaje y NTU. Pudiendo así graficar Voltaje vs NTU y realizando un ajuste polinomial.



Gráfica 4.1.1a Voltaje vs NTU con línea de tendencia.

La ecuación de la línea de tendencia es la siguiente:

$$y = (32.15472 * x^4) - (453.77837 * x^3) + (2477.98528 * x^2) - (6568.7422 * x) + 7452.55014$$

Lo que es igual a

$$NTU = (32.15472 * V^4) - (453.77837 * V^3) + (2477.98528 * V^2) - (6568.7422 * V) + 7452.55014$$

De esta manera es posible convertir el voltaje en NTU, así que se programa dicha ecuación (Figura 4.1.1d).

```

void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0);

  //voltageNTU = (sensorValue * (5.0 / 1024.0));
  //NTU = (32.15472*(voltageNTU^4)-(453.77837*(voltageNTU^3))+(2477.98528*(voltageNTU^2))
  -(6568.7422*voltageNTU)+7452.55014);

  float NTU = (32.15472*((sensorValue * (5.0 / 1024.0))*(sensorValue * (5.0 / 1024.0))*
  (sensorValue * (5.0 / 1024.0))*(sensorValue * (5.0 / 1024.0)))-(453.77837*
  ((sensorValue * (5.0 / 1024.0))*(sensorValue * (5.0 / 1024.0))*(sensorValue *
  (5.0 / 1024.0)))+(2477.98528*((sensorValue * (5.0 / 1024.0))*(sensorValue *
  (5.0 / 1024.0)))-(6568.7422*(sensorValue * (5.0 / 1024.0)))+7452.55014);
  Serial.print("                                las NTU son");
  Serial.println(NTU); // imprime el valor que es

  delay (1000);
}

```

Figura 4.1.1d Código de programación para NTU del sensor de turbidez de la práctica 1.

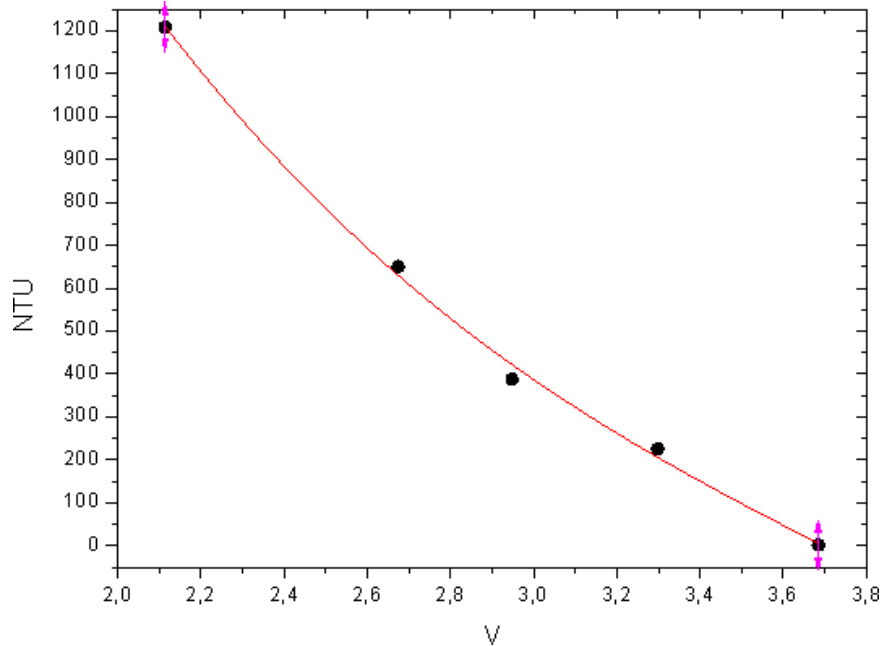
De esta forma se llevó a cabo la calibración del sensor. Una vez calibrado el sensor para un rango de 0 a 800 NTU. Realizamos una vez más 5 pruebas de las cuales calculamos el promedio, obteniendo los siguientes datos:

| Gramos de café | Prototipo | | Turbidimetro HACH | |
|----------------|-----------|---------------------|-------------------|---------------------|
| | NTUprom | Desviación estándar | NTUprom | Desviación estándar |
| 0.00 | 1.29 | 2.781 | 0.715 | 0.001 |
| 0.10 | 120.01 | 3.190 | 120 | 0.632 |
| 0.20 | 271.02 | 7.023 | 271 | 0.632 |
| 0.30 | 430.05 | 9.440 | 430 | 1.414 |
| 0.50 | 800.04 | 8.560 | 800 | 0.000 |

Practica 2 Sensor aislado (medición en completa oscuridad)

Los pasos son los mismos que la practica 1.

| ramos de café | Prototipo | | | | | Turbidimetro HACH | | | | |
|---------------|-----------|------|------|------|------|-------------------|------|------|--------|--------|
| | V1 | V2 | V3 | V4 | V5 | NTU1 | NTU2 | NTU3 | NTU4 | NTU5 |
| 0.00 | 3.68 | 3.68 | 3.69 | 3.68 | 3.69 | 1.32 | 1.16 | 1.23 | 1.3 | 1.18 |
| 0.10 | 3.29 | 3.3 | 3.3 | 3.29 | 3.31 | 227 | 225 | 225 | 226.3 | 225.1 |
| 0.20 | 2.95 | 2.95 | 2.94 | 2.95 | 2.95 | 389 | 387 | 389 | 389 | 388.4 |
| 0.30 | 2.68 | 2.67 | 2.67 | 2.67 | 2.68 | 651 | 645 | 647 | 650.1 | 659.3 |
| 0.50 | 2.12 | 2.11 | 2.12 | 2.11 | 2.11 | 1211 | 1204 | 1208 | 1209.3 | 1208.4 |



Gráfica 4.1.1b Voltaje vs NTU con línea de tendencia.

La ecuación de la línea de tendencia es la siguiente:

$$y = -(90.32767 * x^3) + (1031.64643 * x^2) - (4415.80939 * x) + 6788.38359$$

Lo que es igual a

$$NTU = -(90.32767 * V^3) + (1031.64643 * V^2) - (4415.80939 * V) + 6788.38359$$

```
void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0); // read the input on analog pin 0:

  //voltageNTU = (sensorValue * (5.0 / 1024.0)); // Convert the analog reading (which goes from 0 to 5V) to voltage
  //NTU = -(90.32767*(voltageNTU^3))+(1031.64643*(voltageNTU^2))-(4415.80989*voltageNTU)+6788.38359

  float NTU = -(90.32767*((sensorValue * (5.0 / 1024.0))*((sensorValue * (5.0 / 1024.0))*
(sensorValue * (5.0 / 1024.0))))+(1031.64643*((sensorValue * (5.0 / 1024.0))*((sensorValue *
(5.0 / 1024.0))))-(4415.80989*(sensorValue * (5.0 / 1024.0)))+6788.38359;
  Serial.print("          las NTU son");
  Serial.println(NTU); // imprime el valor que es

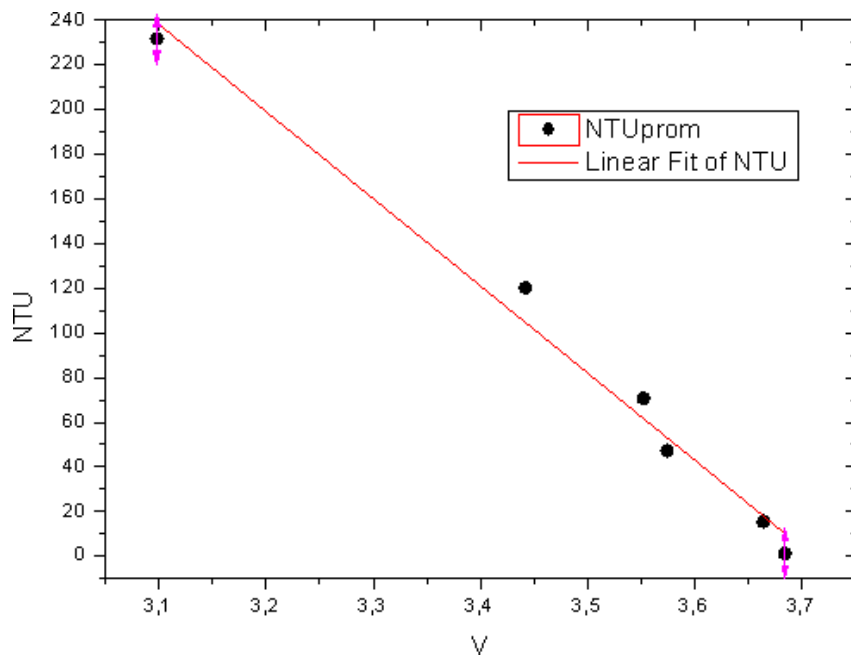
  delay (1000);
}
}
```

Figura 4.1.1e Código de programación para NTU del sensor de turbidez de la práctica 2.

| Tabla 4.1.1d NTU: Prototipo, equipo comercial | | | | |
|---|-----------|---------------------|-------------------|---------------------|
| Gramos de café | Prototipo | | Turbidimetro HACH | |
| | NTUprom | Desviación estándar | NTUprom | Desviación estándar |
| 0 | 5.631 | 2.412 | 1.238 | 0.063 |
| 0.1005 | 205.853 | 4.177 | 225.68 | 0.823 |
| 0.2013 | 422.099 | 2.759 | 388.48 | 0.776 |
| 0.3029 | 630.016 | 4.093 | 650.48 | 4.906 |
| 0.5012 | 1210.419 | 6.193 | 1208.14 | 2.313 |

Practica 3 Sensor aislado en nivel bajo

| Tabla 4.1.1e NTU del agua a diferentes concentraciones de café | | | | | | | | | | |
|--|-----------|------|------|------|------|-------------------|---------|--------|--------|---------|
| Gramos de café | Prototipo | | | | | Turbidimetro HACH | | | | |
| | V1 | V2 | V3 | V4 | V5 | NTU1 | NTU2 | NTU3 | NTU4 | NTU5 |
| 0.00 | 3.68 | 3.68 | 3.69 | 3.68 | 3.69 | 1.32 | 1.16 | 1.23 | 1.3 | 1.18 |
| 0.01 | 3.66 | 3.67 | 3.66 | 3.67 | 3.66 | 15.3 | 15.5 | 15.8 | 15.5 | 15.7 |
| 0.03 | 3.57 | 3.58 | 3.57 | 3.57 | 3.58 | 47.4 | 47.2 | 47.1 | 47.41 | 47.3 |
| 0.05 | 3.55 | 3.55 | 3.55 | 3.56 | 3.55 | 71.1 | 70.5 | 70.4 | 70.8 | 70.6 |
| 0.08 | 3.44 | 3.44 | 3.44 | 3.45 | 3.44 | 120.186 | 120.117 | 120.27 | 120.18 | 120.186 |
| 0.10 | 3.29 | 3.3 | 3.3 | 3.29 | 3.31 | 227 | 225 | 225 | 226.3 | 225.1 |



Gráfica 4.1.1c Voltaje vs NTU con línea de tendencia.

$$y = -(390.33989 * x) + 1448.20473$$

$$NTU = -(390.33989 * V) + 1448.20473$$

```

void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0); // read the input on analog pin 0:

  //voltageNTU = (sensorValue * (5.0 / 1024.0));
  //NTU = -(390.33989*voltageNTU)+1448.20473;

  float NTU = -(390.33989*(sensorValue * (5.0 / 1024.0)))+1448.20473;
  Serial.print("                                las NTU son");
  Serial.println(NTU); // imprime el valor que es

  delay (1000);
}

```

Figura 4.1.1f Código de programación para NTU del sensor de turbidez de la práctica 3.

| Tabla 4.1.1f NTU: Prototipo, equipo comercial | | | | |
|--|-----------|------------|-------------------|------------|
| Gramos de café | Prototipo | | Turbidimetro HACH | |
| | NTUprom | Desviación | NTUprom | Desviación |
| 0 | 10.193 | 1.912 | 1.238 | 0.063 |
| 0.0117 | 17.999 | 1.912 | 15.56 | 0.174 |
| 0.034 | 53.130 | 1.912 | 47.282 | 0.119 |
| 0.05 | 61.717 | 1.561 | 70.68 | 0.248 |
| 0.081 | 104.655 | 1.561 | 120.187 | 0.049 |
| 0.1005 | 160.864 | 2.921 | 225.68 | 0.823 |

4.1.2 Sensor de temperatura

El sensor de temperatura DS10B20 fue conectado, de acuerdo a sus especificaciones. Una vez echa esta acción, se procedió a realizar el código se programación.

```
#include <OneWire.h>
#include <DallasTemperature.h>

// Pin donde está conectado el sensor
const byte pinTemperatura = 9;

//Instancia a las clases OneWire y Dallas Temperature
OneWire oneWireObjeto(pinTemperatura);
DallasTemperature sensorTemperatura(&oneWireObjeto);

void setup() {
  // Iniciamos la comunicación serie
  Serial.begin(19200);
  //Iniciamos el bus 1-Wire
  sensorTemperatura.begin();
}

void loop() {
  // Mandamos comandos para toma de temperatura al sensor
  //Serial.println("Mandando comandos al sensor");
  sensorTemperatura.requestTemperatures();

  //Leemos y mostramos los datos del sensor DS18B20
  Serial.print("La Temperatura es: ");
  Serial.println(sensorTemperatura.getTempCByIndex(0));

  delay (1000);
}
```

Figura 4.1.2a Código de programación para temperatura del sensor DS10B20.

La programación de este sensor es sencilla ya que Arduino contiene las librerías que controlan al sensor.

Para validar la medición del sensor se realizó la siguiente práctica.

Materiales:

1 Termómetro -10 a 150°C LLOYD'S

1 vaso de precipitados 500mL,

1 parrilla,

1 soporte,

1 pinzas de tres dedos,

5 vasos de precipitados de 150mL,

1 agitador magnético.

El procedimiento es bastante sencillo, consiste en montar los materiales como se muestra en la figura 4.1.3 b. posteriormente, calentar agua hasta una temperatura de 70°C aproximadamente, apagamos la parrilla y le agregamos 50mL de agua fría cada cierto tiempo que se estabilice la temperatura.

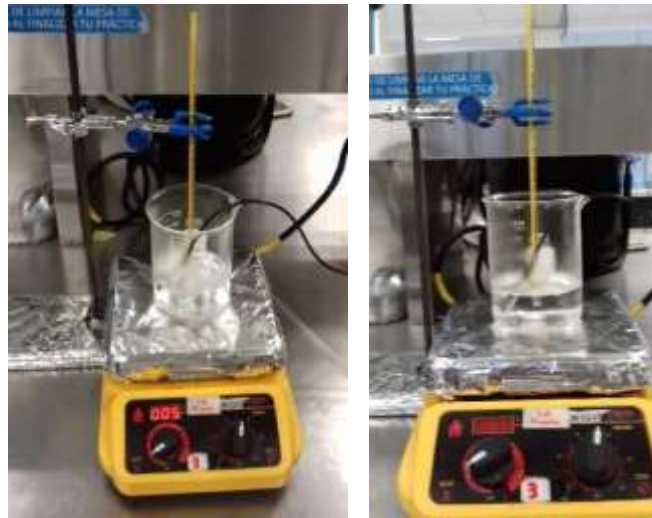


Figura 4.1.2b Montado de equipo

Registramos las diferentes temperaturas que nos proporciona el sensor del prototipo y el termómetro de laboratorio.

Los datos obtenidos fueron los siguientes:

| Tabla 4.1.2 Temperaturas °C | |
|------------------------------------|-----------------------------|
| Prototipo | Termometro comercial |
| 70.19 | 69 |
| 61.25 | 61 |
| 74.44 | 74 |
| 41.38 | 41 |
| 33.63 | 34 |
| 29.81 | 30 |
| 26.06 | 26 |
| 25.06 | 25 |

4.1.3 Sensor de pH

La programación se llevó a cabo partiendo del código ejemplo del proveedor.

```
#define SensorPin A0          //pH meter Analog output to
#define Offset 0.00          //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40      //times of collection
int pHArray[ArrayLenth];    //Store the average value of the
int pHArrayIndex=0;
void setup(void)
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  Serial.println("pH meter experiment!");    //Test the ser
}
void loop(void)
{
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval)
  {
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
  }
  if(millis() - printTime > printInterval)    //Every 800 m:
  {
```

```

        Serial.print("Voltage:");
        Serial.print(voltage,2);
        Serial.print("    pH value: ");
        Serial.println(pHValue,2);
        digitalWrite(LED,digitalRead(LED)^1);
        printTime=millis();
    }
}
double avergearray(int* arr, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/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=arr[i];
            }else{
                amount+=arr[i]; //min<=arr<=max
            }
        }
    }
    avg = (double) amount/ (number-2);
}
return avg;
}

```

Figura 4.1.3a Código de ejemplo de programación.

Como podemos observar, una vez más el código ejemplo del proveedor nos devuelve el voltaje (mili voltios) medido por el sensor.

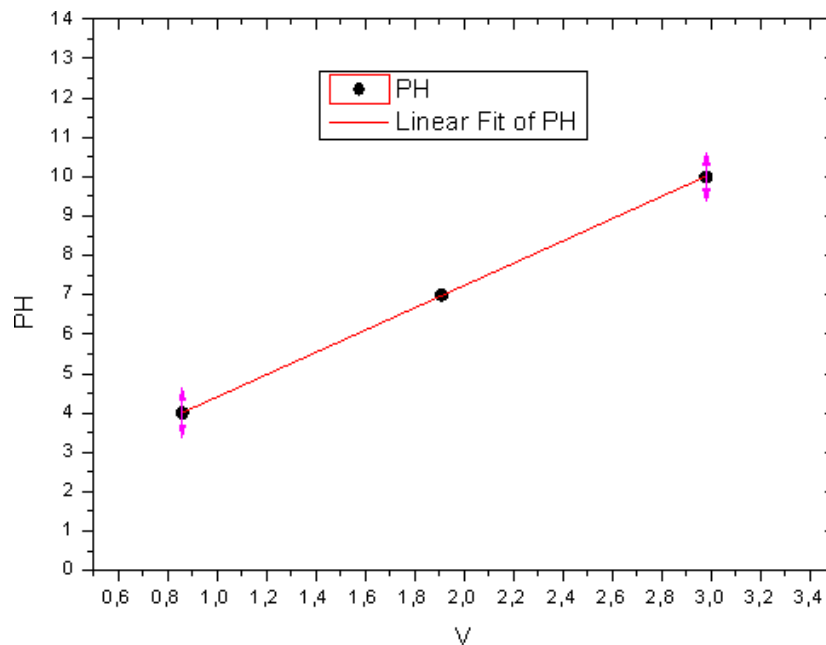
Así que es necesario convertir esa señal de voltaje a la escala adecuada de pH.

Para ello realizamos la calibración del sensor. Se han utilizado tres soluciones de con pH 4, 7 y 10 directamente proporcionados por el proveedor.

Practica 1 soluciones estándar del proveedor.

Comenzamos cargando el código a nuestra tarjeta Arduino UNO. Para la lectura de los datos se coloca el electrodo al cada estándar (4, 7, 10), limpiando el electrodo con agua destilada en cada medición.

Graficamos los puntos obtenidos y ajustamos una línea de tendencia. Obteniendo así una gráfica de calibración para el sensor.



Gráfica 4.1.3a Voltaje vs pH con línea de tendencia.

Ecuación de ajuste (calibración).

$$y = 2.8301 * x + 1.57563$$

A lo que se interpretamos como:

$$pH = 2.8301 * V + 1.57563$$

A continuación incorporamos la ecuación de ajuste el código.

```

void loop(void)
{
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval)
  {
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    //Calibramos, y=(2.8301*voltage)+1.57563+ Offset
    pHValue = ((2.8301*(voltage))+1.57563) + Offset;
    samplingTime=millis();
  }
}

```

Figura 4.1.3b Código de programación del sensor de pH. Práctica 1.

| Tabla 4.1.3a pH desviación estándar | | |
|-------------------------------------|-------------|---------------------|
| Buffer | Calibración | Desviación estándar |
| 4 | 4.009516 | 0.004758 |
| 7 | 6.981121 | 0.0094395 |
| 10 | 10.009328 | 0.004664 |

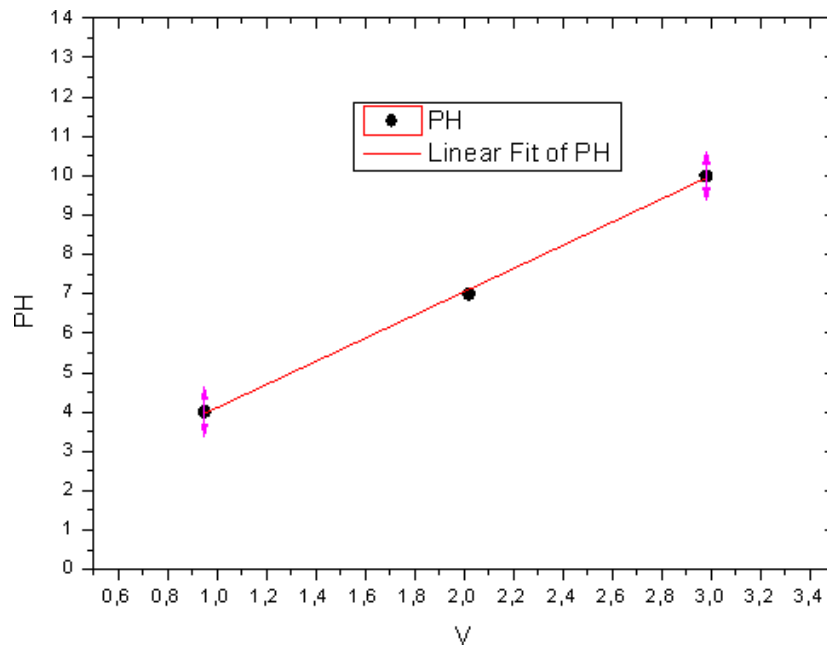
Practica 2 Estándares de calibración comerciales

Una manera de corroborar los datos obtenidos anteriormente es realizar una segunda calibración con diferentes estándares comerciales. En este caso hacemos uso de buffers de la marca HYCEL (4.01, 7, 10).



Figura 4.1.3c Buffer: 4.01, 7, 10 HYCEL.

Realizamos el mismo procedimiento de la práctica 1 para obtener la gráfica de calibración.



Gráfica 4.1.3b Voltaje vs pH con línea de tendencia.

Ecuación de ajuste (calibración).

$$y = 2.94776 * x + 1.15693$$

A lo que se interpretamos como:

$$pH = 2.94776 * V + 1.15693$$

A continuación incorporamos la ecuación de ajuste el código.

```
void loop(void)
{
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval)
  {
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    //Calibramos, y=(2.94776*voltage)+1.15693+ Offset
    pHValue = {(2.94776*(voltage))+1.15693} + Offset;
    samplingTime=millis();
  }
}
```

Figura 4.1.3c Código de programación del sensor de pH. Práctica 1.

| Tabla 4.1.3b pH desviación estándar | | |
|-------------------------------------|-------------|---------------------|
| Buffer | Calibración | Desviación estándar |
| 4.01 | 3.23983438 | 0.38508281 |
| 7 | 6.88891099 | 0.0555445 |
| 10 | 10.1628489 | 0.08142445 |

4.2 Código para el monitoreo de: Turbidez, temperatura y pH.

/*Este programa calcula del pH, temperatura y turbidez de una sustancia

*Para Turbiedad conectar en A0

*Para Temperatura conectar en pin 9

*Para pH conectar en A3

*/

//definimos parámetros para pH-----

#define Sensor Pin A3

#define Offset 0.0

#define LED 13

#define samplingInterval 20

#define printInterval 1000

#define ArrayLenth 40

int pHArray[ArrayLenth];

int pHArrayIndex=0;

//definimos librerias y parámetros para la Temperatura-----

#include <OneWire.h>

#include <DallasTemperature.h>

// Pin donde está conectado DQ

const byte pinTemperatura = 9;

//Instancia a las clases OneWire y Dallas Temperature

```

OneWire oneWireObjeto(pinTemperatura);

DallasTemperature sensorTemperatura(&oneWireObjeto);

//.....

void setup(void)

{

  pinMode(LED,OUTPUT);

  //iniciamos comunicación serie

  Serial.begin(19200);

  Serial.println("COMENZANDO :D");

  //Iniciamos el bus 1-Wire para temperatura-----

  sensorTemperatura.begin();

}

//.....

void loop(void)

//se programa el sensor de pH para poder calcular y leer los datos-----

{

  static unsigned long samplingTime = millis();

  static unsigned long printTime = millis();

  static float pHValue,voltagepH;

  if(millis()-samplingTime > samplingInterval)

  {

    pHArray[pHArrayIndex++]=analogRead(SensorPin);

    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;

    voltagepH = avergearray(pHArray, ArrayLenth)*5.0/1024;

```

```

//Calibramos,reemplazando el 3.5 que trae el programa por la linea de tendencia
y=(-0.3867x^2)+3.9282x+0.9077)+ Offset, para lograr un mejor resultado

    pHValue = ((-0.3867*((voltagepH)*(voltagepH)))+(3.9282*(voltagepH))+0.9077) +
Offset;

    samplingTime=millis();

}

if(millis() - printTime > printInterval)

{

//Serial.print("Voltaje pH:");

    // Serial.print(voltagepH,2);

    Serial.print("El pH es: ");

Serial.println(pHValue,2);

    digitalWrite(LED,digitalRead(LED)^1);

    printTime=millis();

}

//se programa el sensor de Temperatura para poder calcular y leer los datos-----
-----

// Mandamos comandos para toma de temperatura al sensor

//Serial.println("Mandando comandos al sensor");

sensorTemperatura.requestTemperatures();

//Leemos y mostramos los datos del sensor DS18B20

Serial.print("          La Temperatura es: ");

Serial.println(sensorTemperatura.getTempCByIndex(0));

//Se programa el sensor de Turbiedad para poder calcular y leer los datos-----
-----

int sensorValue = analogRead(A0);

//voltageNTU = (sensorValue * (5.0 / 1024.0));

```

```

//NTU = (32.15472*(votageNTU^4))-
(453.77837*(votageNTU^3))+(2477.98528*(votageNTU^2))-
(6568.74222*votageNTU)+7452.55014; // Convert the analog reading (which goes from
0 - 1023) to a voltage (0 - 5V);

float NTU = (32.15472*((sensorValue * (5.0 / 1024.0))*(sensorValue * (5.0 /
1024.0))*(sensorValue * (5.0 / 1024.0))*(sensorValue * (5.0 / 1024.0))))-
(453.77837*((sensorValue * (5.0 / 1024.0))*(sensorValue * (5.0 /
1024.0))*(sensorValue * (5.0 / 1024.0))))+(2477.98528*((sensorValue * (5.0 /
1024.0))*(sensorValue * (5.0 / 1024.0))))-(6568.74222*(sensorValue * (5.0 /
1024.0)))+7452.55014;

Serial.print("                las NTU son: ");

Serial.println(NTU); // imprime el valor que es

Serial.println(sensorValue * (5.0 / 1024.0));

//Se programa el sensor de Oxigeno para poder calcular y leer los datos-----
---

float sensorValueO;

float sensorVoltageO;

float Value_O2;

sensorValueO = analogRead(A5);

sensorVoltageO =(sensorValueO/1024)*5.0;

sensorVoltageO = sensorVoltageO/201*10000;

Value_O2 = sensorVoltageO/7.43;

Serial.print("                O_2 en el ambiente: ");

Serial.print(Value_O2,1);

Serial.println("%");

delay (1000);

}

//-----

// función double para pH-----

```

```

double avergearray(int* arr, int number){

    int i;

    int max,min;

    double avg;

    long amount=0;

    if(number<=0){

        Serial.println("Error number for the array to avraging!/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=arr[i];  
  
    }else{  
  
        amount+=arr[i]; //min<=arr<=max  
  
    }  
  
} //if  
  
} //for  
  
avg = (double)amount/(number-2);  
  
} //if  
  
return avg;  
  
}
```

4.3 Costos

| Prototipo de monitoreo: PH, NTU y temperatura | Artículo | cantidad | Costo (M.N) | |
|---|-------------------------|------------------|-------------|---------|
| | Sensor de turbidez | 1 | 586.21 | |
| | sensor de temperatura | 1 | 142.24 | |
| | Sensor de pH | 1 | 1189.66 | |
| | Arduino UNO | 1 | 168.10 | |
| | Cautín | 1 | 135.34 | |
| | soldadura de estaño | 10 | 60.34 | |
| | Protoboard | 1 | 86.21 | |
| | Cables DUPONT M-H | 4 | 68.97 | |
| | Cables DUPONT M-M | 4 | 68.97 | |
| | Total | | | 2506.04 |
| | Sensor: | Oxigeno Disuelto | 1 | 3680.01 |
| | Conductividad electrica | 1 | 6889.99 | |
| Total | | | 10570.00 | |

| Equipo | Costo | Enlace |
|---|-------|---|
| Termometro de mercurio -20 a 150°C sumergible | 269 | https://articulo.mercadolibre.com.mx/MLM-602011183-termometro-de-mercurio-20-a-150c-sumergible- JM |
| Turbidimetro HACH 2100q portable | 20000 | https://articulo.mercadolibre.com.mx/MLM-641123507-hach-2100q-turbidimetro-portable- JM |
| Medidor de Ph Zeigen | 2380 | https://articulo.mercadolibre.com.mx/MLM-555040250-medidor-de-ph-de-escritorio-para-liquidos-y-agua-laboratorio- JM |
| Total | 22649 | |

CAPITULO 5 CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Con el desarrollo del presente trabajo, construimos y adaptamos un prototipo de monitoreo de calidad de agua. Partiendo del objetivo general podemos concluir que se cumplió satisfactoriamente ya que se cumplió con la construcción de un prototipo funcional de bajo costo para evaluar parámetros de calidad de agua.

Arduino fue el soporte principal de la investigación, permitió la configuración del prototipo tanto en software como en hardware, gracias a que Arduino forma parte del grupo Open Source. El código de programación de Arduino es bastante amigable, permitiendo una configuración sencilla del prototipo. En cuanto al costo de la tarjeta Arduino y sensores de la misma familia de código abierto son bastante accesibles.

Se propuso como objetivo específico evaluar 5 parámetros de calidad de agua: turbidez, temperatura, pH, conductividad eléctrica y oxígeno disuelto. En el caso de los sensores de conductividad y oxígeno disuelto, hubo una barrera fuera de nuestro alcance, ya que, como los demás sensores se hizo realizar una búsqueda muy extensa, contactando varios proveedores para de artículos electrónicos para revisar su existencia y disponibilidad de ser enviado. Para el caso de estos dos sensores, no fue posible comprarlos por los permisos en la aduana para entrar al país.

De igual forma se propuso realizar pruebas comparativas del prototipo con equipos comúnmente utilizados en laboratorio, esto para validar de alguna manera los parámetros obtenidos por el prototipo. Como se pudo observar en los resultados, el sensor de temperatura es demasiado exacto en su medición ya que se comparó con un termómetro de mercurio comercial. El sensor de pH muestra un rango promedio de error aproximadamente de 0.17 lo cual para una medición de pH es aceptable. En el caso del sensor de turbidez se obtuvo un error promedio de 3.5, en rangos de 200 NTU o más es aceptable, pero en rangos inferiores a 200NTU si es de considerar el error aunque dependiendo la aplicación, en este caso en aguas residuales es aceptable.

El costo total del prototipo para medir pH, turbidez y temperatura es aproximadamente 9 veces menor al costo de los equipos comúnmente utilizados en un laboratorio. Teniendo en cuenta siempre el porcentaje de que nos proporciona cada uno y el enfoque que le dimos a esta investigación podemos concluir que es factible implementar el prototipo de monitoreo.

RECOMENDACIONES

Para la mejora futura del prototipo, sugerimos realizar la calibración en intervalos específicos en los que se trabaje comúnmente para obtener datos más precisos.

El alcance del prototipo es muy alto en cuanto a tecnología, sugerimos mejorarlo incorporando un sistema que nos permita el almacenamiento de datos: sugerimos hacer uso de una Raspberry Pi.

Para un mejor monitoreo en tiempo real recomendamos implementar el uso de un módulo de Wifi o un circuito que nos permita la incorporación de Chip de telefonía, para lograr una transmisión de datos a una plataforma en la red o directamente a un dispositivo móvil e incluso un módulo de GPS para detectar la zona exacta los parámetros obtenidos. Incrementando el alcance del proyecto sugerido en un futuro no muy lejano concluimos recomendando a los trabajos futuros el uso de esta tecnología para lograr un mejor monitoreo de calidad de agua.

CAPÍTULO 6. REFERENCIAS

- ADASA, 2013** ADASA Sistemas Water and Environment Technology. (2013). Recuperado de: <http://www.adasasistemas.com/esp/home.php>
- AgroDer, 2012** AgroDer. (2012). Huella hídrica en México en el contexto de Norteamérica. WWF México y AgroDer. CDMX, México.
- APHA-AWWA-WEF, 2005** APHA-AWWA-WEF (2005) Standard Methods for the Examination of Water and Wastewater. 21th Edition. Washington DC, (2-8 a 2-11; 2-44 a 2-48; 2-61 a 2-62; 490;4-136 a 4-143).
- Bartram, 1996** Bartram, J. and Ballance, R. [Eds] 1996 Water Quality Monitoring; A Practical Guide to the Design and Implementation of Fresh Water Quality Studies and Monitoring Programmes. Chapman & Hall, London.
- CONAGUA, 2016** Estadísticas del Agua en México: Edición 2016. *Comisión Nacional del Agua. Secretaría de Medio Ambiente y Recursos Naturales.*
- CONAGUA. 2016b.** Subdirección General Técnica.
- Frank, 1995** Frank, B. (1955). La Historia del agua como historia del hombre En Agua su aprovechamiento en la agricultura. (Departamento de Agricultura de los Estados Unidos): 1-9
- García, 2015** García, G. M., & Martínez, P. C. C. (2015). Escenario del agua en México. *CULCyT*, (30).
- Metcalf, 1996** Metcalf (1996). Ingeniería de aguas residuales. Tratamiento, vertido y reutilización. McGraw-Hill. México, D.F
- Meybeck, 1989** Meybeck, M., Chapman, D. and Helmer, R. [Eds] 1989 Global Freshwater Quality: A First Assessment. Blackwell Reference, Oxford, 306 pp.
- NatGeo, 2011** National Geographic. (2011, Julio, 6). National Geographic. Recuperado de: <https://www.nationalgeographic.es/medio-ambiente/contaminacion-del-agua>
- Olivera, 2017** Alejandra Olivera. (2017, Octubre, 3). Angulo 7. Recuperado de: <http://www.angulo7.com.mx/2017/10/03/habra-laboratorio-mida-contaminacion-agua-costara-3-5mdp/>.
- Open Source, 2018** Open Source. (2018). Celebrate 20 years of Open Source. California, EC.: Open Source Initiative. Recuperado de: <https://opensource.org/>

- Open Source, 2018** Open Source. (2018). History of the OSI. California, EC.: Open Source Initiative. Recuperado de: <https://opensource.org/history>
- OSHW, 2018** Open Source Hardware Association (2018). Best Practices der Open-Source-Hardware 1.0. Recuperado de: <https://www.oshwa.org/?s=hardware+>
- Rivera, 2015** Rivera, D, & Yopez, E. (2015). Diseño e implementación de un prototipo para la medición de calidad del agua y control de la oxigenación en forma remota orientado a la producción acuícola (tesis de pregrado). Universidad Politécnica Salesiana, Guayaquil, Colombia.
- Romero, 2009** ROMERO-AGUILAR, M., COLÍN-CRUZ, A., SÁNCHEZ-SALINAS, E., & OrtizHernández, M. A. (2009). Tratamiento de aguas residuales por un sistema piloto de humedales artificiales: evaluación de la remoción de la carga orgánica. *Revista internacional de contaminación ambiental*, 25(3), 157-167.
- Saldaña, 2002** Saldaña, P., Lerdo, T. A., Gómez, M. A., & López, R. (2002). La importancia de incluir análisis de toxicidad en descargas industriales y municipales que afectan a los cuerpos receptores. *Instituto Mexicano de Tecnología del Agua*, 1-11.
- SE, 2015** Secretaría de Economía. *Información económica y estatal* [archivo PDF]. Recuperado de <https://www.gob.mx/cms/uploads/attachment/file/85638/puebla.pdf>
- Singh, 2004** Singh, K. P., Malik, A., Mohan, D., & Sinha, S. (2004). Multivariate statistical techniques for the evaluation of spatial and temporal variations in water quality of Gomti River (India)—a case study. *Water research*, 38(18), 3980-3992.
- Soto, 2013** Soto Montes de Oca, A. (2013). *La valoración económica del medio ambiente a través del método de valoración contingente: el caso de la Cuenca del Alto Atoyac en Puebla, México*(Master's thesis, Universidad Iberoamericana Ciudad de México. Departamento de Economía).
- Toledo, 2002** Toledo, A. 2002. El agua en México y el Mundo. *Gaceta Ecológica-Instituto Nacional de Ecología (México)* No. 64: 9-18.
- WHO, 1991** WHO 1991 GEMS/WATER 1990-2000. The Challenge Ahead. WHO/PEP/91.2, World Health Organization, Geneva.

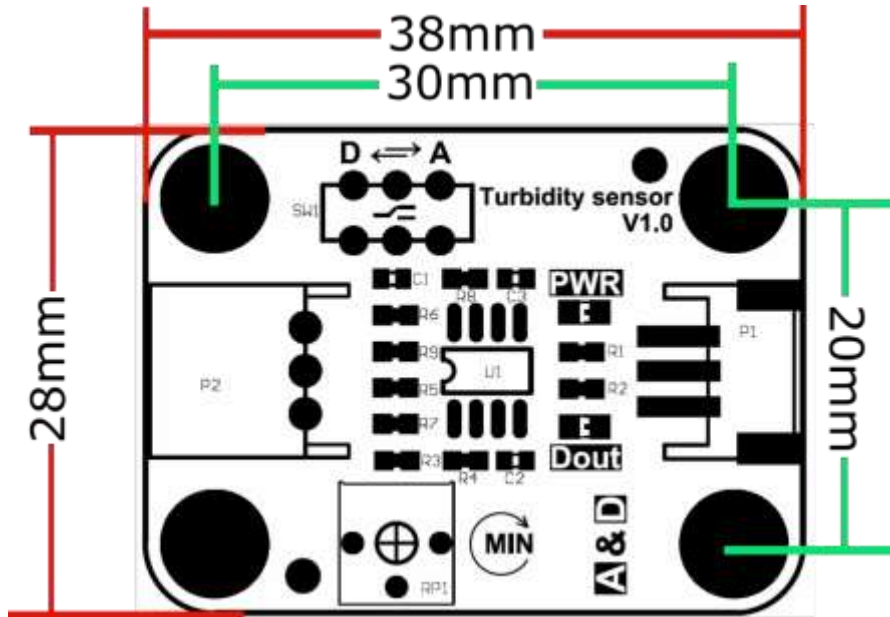


Turbidity sensor SKU: SEN0189



Specification

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
Analog output: 0-4.5V
Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5°C~90°C
- Storage Temperature: -10°C~90°C
- Weight: 30g
- Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches



Waterproof DS18B20 Digital Temperature Sensor for Arduino



SPECIFICATION

Usable with 3.0V to 5.5V power/data

±0.5°C Accuracy from -10°C to +85°C

Usable temperature range: -55 to 125°C (-67°F to +257°F)

9 to 12 bit selectable resolution

Uses 1-Wire interface- requires only one digital pin for communication

Unique 64 bit ID burned into chip

Multiple sensors can share one pin

Temperature-limit alarm system

Query time is less than 750ms

3 wires interface:

Type A

Red wire - VCC

Black wire - GND

Yellow wire - DATA

Type B (Recently there's a series of sensors using this pin mapping. Sorry for the inconvenience.)

Red wire - VCC

Yellow wire - GND

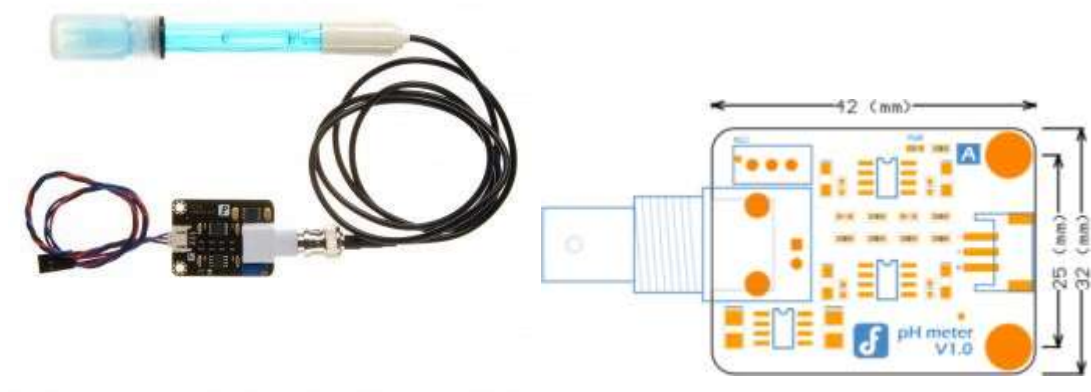
Green wire - DATA

Stainless steel tube 6mm diameter by 35mm long

Cable diameter: 4mm(0.16)

Length: 90cm(35.43")

PH meter(SKU: SEN0161)



Specification

Module Power : 5.00V

Module Size : 43mm×32mm

Measuring Range:0-14PH

Measuring Temperature :0-60 °C

Accuracy : ± 0.1pH (25 °C)

Response Time : ≤ 1min

pH Sensor with BNC Connector

PH2.0 Interface (3 foot patch)

Gain Adjustment Potentiometer

Power Indicator LED

Cable Length from sensor to BNC connector:660mm