



Conciencia Tecnológica
ISSN: 1405-5597
contec@mail.ita.mx
Instituto Tecnológico de Aguascalientes
México

Una Guía Práctica para Desarrollar Equipo de Laboratorio con Arduino

Zambrano-de la Torre, M.; Guzmán-Fernández, M.; Sifuentes-Gallardo, C.; Manuel Ortiz-Romero, V.; Cruz-Dominguez, Oscar; Fraire-Hernández, M.; Pérez-Martínez, J.; Durán-Muñoz, H.
Una Guía Práctica para Desarrollar Equipo de Laboratorio con Arduino
Conciencia Tecnológica, núm. 59, 2020
Instituto Tecnológico de Aguascalientes, México
Disponible en: <https://www.redalyc.org/articulo.oa?id=94463783002>



Una Guía Práctica para Desarrollar Equipo de Laboratorio con Arduino

A practical guide to develop laboratory equipment with Arduino

M. Zambrano-de la Torre 1
Universidad Autónoma de Zacatecas, México

Redalyc: <https://www.redalyc.org/articulo.oa?id=94463783002>

M. Guzmán-Fernández 1
Universidad Autónoma de Zacatecas, México

C. Sifuentes-Gallardo 1
Universidad Autónoma de Zacatecas, México

V. Manuel Ortiz-Romero 1
Universidad Autónoma de Zacatecas, México

Oscar Cruz-Dominguez 2
Universidad Politécnica de Zacatecas, México

M. Fraire-Hernández 1
Universidad Autónoma de Zacatecas, México

J. Pérez-Martínez 1
Universidad Autónoma de Zacatecas, México

H. Durán-Muñoz 1
Universidad Autónoma de Zacatecas, México
hectorduranm@hotmail.com

Recepción: 08 Enero 2020
Aprobación: 18 Mayo 2020

RESUMEN:

Adquirir equipo para realizar prácticas de laboratorio se ha vuelto muy complicado debido a su alto costo. Para resolver este problema, la comunidad de docentes ha utilizado la placa de adquisición de datos Arduino. Esta placa es de bajo costo y permite la automatización de equipos de una manera simple y práctica. Sin embargo, ya no es suficiente utilizar solamente dicha placa,

NOTAS DE AUTOR

- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México
- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México
- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México
- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México
- 2 Carrera de Ingeniería Industrial. Universidad Politécnica de Zacatecas, Fresnillo, Zac. México
- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México
- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México
- 1 Universidad Autónoma de Zacatecas, Unidad Académica de Ingeniería Eléctrica, Zacatecas Zac. México

*Corresponding author: hectorduranm@hotmail.com

ahora es necesario combinarla con la implementación de un software de fácil uso (Visual Studio), y que permita ser controlada en tiempo real. En este trabajo se detalla el diseño e implementación de una placa de adquisición de datos basada en Arduino, y una interfaz gráfica para su control y automatización. Para esto, se muestran tres aplicaciones simples que pueden ser reproducidas por cualquier maestro, investigador o estudiante que no tenga conocimientos básicos sobre electrónica o programación. Al combinar las tres aplicaciones anteriores, fue posible fabricar un medidor de pH con sensor de temperatura y con control de agitación. El costo comercial de este equipo de laboratorio es elevado, pero con la configuración propuesta el costo disminuye 80%.

PALABRAS CLAVE: Arduino, Nuevas Tecnologías y Educación.

ABSTRACT:

Acquiring equipment to carry out laboratory practices has become very complicated due to its high cost. To solve this problem, the teacher community has used the Arduino data acquisition board. This board is inexpensive and allows automation of equipment in a simple and practical way. However, it is no longer enough to use only this board, now it is necessary to combine it with the implementation of an easy-to-use software (Visual Studio), and that allows it to be controlled in real time. This work details the design and implementation of an Arduino-based data acquisition board, and a graphical interface for its control and automation. For this, three simple applications are shown that can be reproduced by any teacher, researcher or student who does not have basic knowledge of electronics or programming. By combining the three previous applications, it was possible to manufacture a pH meter with temperature sensor and with agitation control. The commercial cost of this laboratory equipment is high, but with the proposed configuration the cost decreases 80%.

KEYWORDS: Arduino, New Technologies and Education.

INTRODUCCIÓN.

El uso de nuevas tecnologías en la educación ha revolucionado el proceso de aprendizaje. Por ejemplo, el uso de aplicaciones de redes sociales [1], libros electrónicos [2] y la implementación de equipos de laboratorio desarrollados por docentes. Además, esta nueva tendencia podría resolver diferentes problemas en la educación y la investigación.

Uno de estos problemas es adquirir equipo de laboratorio de alto costo, lo cual genera una creciente necesidad de desarrollar nuevas tecnologías de bajo costo [3, 4]. Una de estas nuevas tecnologías es la placa de adquisición de datos Arduino (DAB). Esta placa tiene un bajo costo, y el usuario no necesita tener un conocimiento profundo sobre electrónica y programación.

Básicamente, esta placa es una unidad de control y el corazón de toda la automatización [5]. En la literatura, existe una amplia gama de trabajos que utilizan la placa Arduino para desarrollar equipos que permitan resolver necesidades particulares de la comunidad docente. Por ejemplo: sistemas de reconocimiento llanto de infantes [6], sistemas de reconocimiento de ADN [7], sistemas de bajo costo para el estudio del comportamiento animal [8], el estudio de la corrosión de metal [9] e incluso sistemas para detección de fluorescencia [10], entre otros [11]. También existe un amplio uso de Visual Basic para diferentes aplicaciones, por ejemplo, la caracterización de materiales [12, 13], análisis estadístico [14] y simulaciones computacionales para procesos químicos [15].

La combinación de Arduino y Visual Studio resulta ser una herramienta extremadamente poderosa, ya que permiten el desarrollo de equipos de laboratorio de bajo costo y resuelven el problema actual de adquirir equipos de laboratorio.

El objetivo de este trabajo es presentar una solución fácil, de bajo costo y multifuncional que pueda implementarse en las universidades para reemplazar equipos de alto costo, con la finalidad de realizar prácticas de laboratorio con equipos caseros. Para ello no es necesario tener un conocimiento profundo sobre electrónica y programación. Por lo que, fue necesario fabricar de una DAB, basada en Arduino, este proceso se describe paso a paso en las siguientes secciones. Posteriormente, la automatización de la placa se realizará mediante el diseño de una interfaz gráfica amigable en Visual Studio. El desarrollo de la interfaz gráfica también se lleva a cabo paso a paso y permite al usuario tener control sobre los actuadores conectados a la placa. Además, se realizan tres aplicaciones simples que pueden implementarse fácilmente y que buscan

resolver las necesidades de cualquier investigador, docente o estudiante para desarrollar equipo de laboratorio o un dispositivo complementario al equipo de laboratorio. Finalmente, al combinar las tres aplicaciones anteriores, fue posible fabricar un medidor de pH (Potencial de Hidrógeno) con sensor de temperatura y control de agitación. El costo comercial de este equipo de laboratorio es de aproximadamente 600 dólares, pero con este sistema casero, el costo total es de 75 dólares.

MATERIALES Y MÉTODOS.

Placa de Adquisición de Datos (DAB).

La característica principal de una placa de adquisición de datos (DAB) es utilizar el mínimo de componentes, que sea de bajo costo y que tenga una operación óptima para resolver las tareas necesarias del usuario. El microcontrolador ATmega328P fue utilizado para diseñar la DAB, su característica principal es su bajo costo y versatilidad. Para el diseño y construcción de la placa, se seleccionaron componentes de bajo costo y fáciles de obtener. Los componentes se muestran en la Tabla 1.

TABLA 1
Componentes utilizados para el DAB.

Tabla 1. Componentes utilizados para el DAB.
Resistencias de 1000 Ω y 330 Ω (1/4 W).
Diodo emisor de luz. 3 mm de diámetro, 4.2 V de voltaje máximo y 20 mA. Indica visualmente que se energiza la placa.
Pulsador con dos terminales. Tiene la función de reiniciar el microcontrolador.
Condensador cerámico 0.1 μ F. Necesario para la comunicación del microcontrolador con el módulo "FT232".
Dos condensadores cerámicos de 22pF. Necesario en la conexión del cristal oscilador externo del microcontrolador.
Oscilador de cristal de 16MHz. Necesario para el funcionamiento del microcontrolador.
FTDI FT232. Componente responsable de la comunicación entre el microcontrolador y la computadora.
Microcontrolador ATmega328P-PU.
Placa fenólica de base para DAB, lámina de cobre con dimensiones de 6 x 8 cm.
Cloruro férrico. Compuesto químico utilizado para la construcción de la placa, circuito impreso (PCB).
Conexión de cable USB-Mini B. Componente necesario para la comunicación con la computadora.

El dispositivo FTDI FT232 (adaptador serie) permite la comunicación entre el microcontrolador a través de los periféricos y el software de programación. La configuración electrónica de la placa se muestra en la Fig. 1. La programación del microcontrolador se puede hacer utilizando el software "Arduino IDE" de uso gratuito.

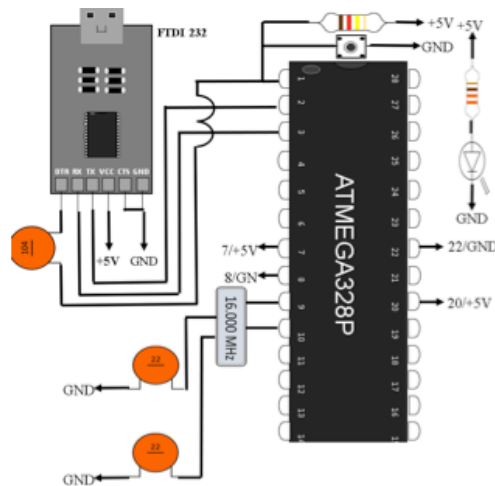


FIGURA 1

Diagrama electrónico de la placa de adquisición de datos con el microcontrolador ATmega328p.

Para el diseño de la placa de circuito impreso (PCB), se utilizó el software en línea de fácil uso EasyEDA, que ofrece herramientas esenciales, fáciles de usar en el diseño de circuitos electrónicos y la generación de circuitos impresos (Fig. 2). Teniendo en cuenta cada uno de los componentes del diseño anterior, se genera la plantilla de circuito impreso, necesaria para la creación de la PCB.

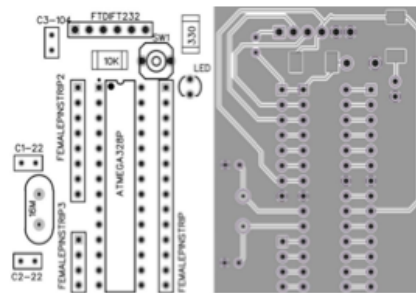


FIGURA 2

Circuito DAB diseñado con el software en línea EasyEDA de uso gratuito, con el microcontrolador ATmega328p.

Construcción. Para la construcción de la DAB se utiliza una placa fenólica. En ella, se vierte el cloruro férrico para así trazar el circuito en el cobre. Mientras que el circuito diseñado está impreso en una hoja de papel tipo fotografía. Luego, se utiliza el método de "planchado", la transferencia de calor al diseño del circuito del papel fotográfico, se impregna en la placa fenólica. Como resultado del método de "planchado", se obtiene la placa fenólica con el circuito impregnado con tinta, proveniente del papel fotográfico.

En el siguiente paso, la placa fenólica se coloca en un recipiente para exponerla al cloruro férrico y marcar las líneas de cobre correspondientes al circuito. El siguiente paso es perforar la placa PCB para ensamblar los componentes, utilizando como herramienta un mini taladro manual con una broca similar a los pines de los componentes del circuito. En el ensamblaje de componentes, se utilizan herramientas como soldador de 40w, soldadura de estaño y pasta. Después del montaje de los componentes, se realizan pruebas de cortocircuito y continuidad antes del primer encendido. El resultado final es la construcción del primer prototipo de la placa de adquisición de datos, donde se concluye con el primer encendido (Fig. 3).

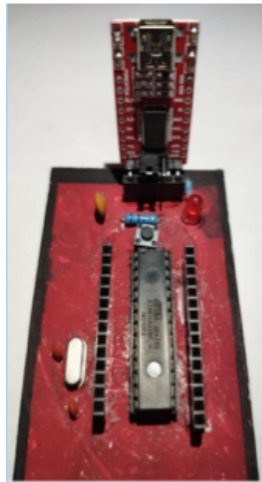


FIGURA 3

Proceso de ensamblaje de componentes en la placa de adquisición de datos

Software de Control. El propósito de esta sección es mostrar las características básicas del software utilizado, en secciones posteriores se muestra en detalle el procedimiento para desarrollar una interfaz gráfica implementada para tres aplicaciones básicas. El software gratuito utilizado es Visual Studio 2019. Este software está disponible en diferentes sistemas operativos, tiene un lenguaje de programación simple, una gran variedad de características en las opciones de diseño y una interfaz de programación amigable. Otra ventaja importante de este software es que genera el código de programación automáticamente al agregar elementos gráficos en la interfaz de usuario y finalmente ofrece múltiples archivos y plantillas para crear diferentes proyectos o aplicaciones. Para nuestro caso, utilizamos la plantilla Aplicación de formulario de Windows (Fig. 4), esta plantilla le permite crear un archivo ejecutable (.exe).



FIGURA 4

Creación de un proyecto en Visual Studio 2019

RESULTADOS.

Implementación de DAB y Software de Control en Tres Aplicaciones Diferentes. La siguiente sección aborda en detalle tres aplicaciones diferentes: (1) Control de apagado / encendido de un LED (diodo emisor de luz), (2) Lectura de temperatura con valores mostrados en un gráfico y (3) Control de un servomotor. Finalmente, con la combinación de las tres aplicaciones previas es posible desarrollar un sistema de medición de pH, agitación y medición de temperatura. Para cada aplicación, se muestran los pasos necesarios para desarrollar la interfaz gráfica, el código de programación del microcontrolador y un diagrama electrónico. El objetivo principal de este trabajo es mostrar, de manera sencilla y paso a paso, cómo la combinación de una DAB y la interfaz gráfica es posible potenciar diferentes aplicaciones para resolver las necesidades de investigadores, docentes y estudiantes. Además, este trabajo busca ser una base para que cualquier usuario, sin un conocimiento profundo de la electrónica y la programación, pueda desarrollar sus propios equipos, evitando la adquisición de equipos de laboratorio de alto costo.

LED controlado por Visual Studio. El primer paso es crear el proyecto en el área de trabajo (Fig. 5-a), agregando la caja de herramientas. (Fig. 5-b). En este caso, se realizó una interfaz gráfica para la comunicación en serie con la DAB para controlar el encendido / apagado de un LED.

El primer paso será declarar la variable "flag" (Fig. 6-a), para las etapas del programa. Además, se definirán los atributos de cada elemento de la interfaz gráfica y la habilitación del puerto serie. Luego, se programan las condiciones que se realizarán, dependiendo del estado de la variable "flag", con un valor de 1, el elemento "ENCENDIDO" se deshabilitará y los demás elementos se habilitarán.

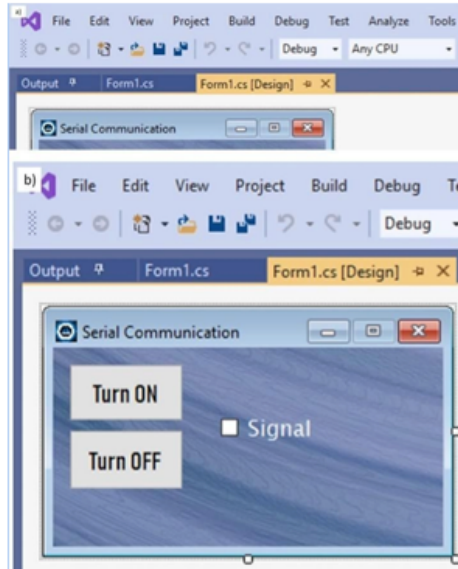


FIGURA 5

a) Área de trabajo y b) Elementos para apagar/encender el LED.

Cuando la variable "flag" tiene el valor de 2, sucederá lo contrario y también se escribirá un "0" en el puerto serie (Fig. 6-b).

```
private void TURNON_Click(object sender, EventArgs e)
{
    flag = 1;
    TurnON.Enabled = false;
    TurnOFF.Enabled = true;
    LED.Enabled = true;
}

private void TURNOFF_Click(object sender, EventArgs e)
{
    flag = 2;
    TurnON.Enabled = true;
    TurnOFF.Enabled = false;
    LED.Enabled = false;
    serialPort1.Write("0");
    LED.Checked = false;
    label7.Text = "";
}

public partial class Form1 : Form
{
    int flag = 0;

    public Form1()
    {
        InitializeComponent();
        TurnOFF.Enabled = false;
        LED.Enabled = false;
        try{ serialPort1.Open(); }
        catch(Exception msg) { MessageBox.Show(msg.ToString()); }
    }
}
```

FIGURA 6

a) Inicialización de los elementos, b) Configuración del estado.

Control de LED con Arduino. El IDE Arduino es un entorno de programación con las características de editor de código, compilador y depurador. Para crear el proyecto en el IDE de Arduino, es necesario seleccionar la pestaña "Archivo" y presionar la opción "Nuevo". El código utilizado para controlar el LED (Fig. 7) se basa en la lectura de información del puerto serie que se envía desde la interfaz gráfica.

```

int led=10;
int dato;

void setup() {
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  dato=Serial.read();
  if (dato == '1') {
    digitalWrite(led, HIGH);
  }
  if (dato == '0') {
    digitalWrite(led, LOW);
  }
}
    
```

FIGURA 7
Código Arduino para control LED

Una vez que se tenga el código en el IDE, se procede a compilarlo presionando la pestaña "verificar", luego el puerto de conexión COM se verifica en la pestaña "herramientas" y "puerto", una vez con el programa cargando como último paso para subir a la placa de adquisición de datos, solo es presionar en la pestaña "subir".

Configuración Electrónica LED. Finalmente, el diagrama electrónico del sistema para prender/apagar un LED se muestra en la Fig. 8.

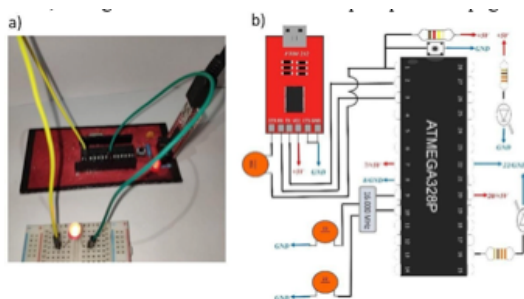


FIGURA 8
a) Configuración electrónica para controlar un led. b) Diagrama esquemático.

Implementación de DAB y Software de Registro de temperatura. Registro de temperatura usando Visual Studio. El diseño de la interfaz para el registro de temperatura consta de cuatro botones, que están programados para enviar señales a la placa. Para saber si cada botón está activado, se le asocia un LED; La programación de cada LED es la misma y se muestra en la Fig. 9-a). La función "temporizador" se utilizó para registrar los valores de temperatura, lo que permite enviar los valores a la interfaz gráfica. Por ejemplo, si la variable "flag" se declara como "1", el valor de "9" debe escribirse en el puerto serie. En la aplicación presentada, hay un contador llamado "time", que representa el tiempo (eje X) en el gráfico. Mientras que la variable "temperatureProBar" está trazando la información proveniente de la respuesta de la placa acerca de los valores de temperatura (Fig. 9-b). Finalmente, se programan las condiciones para la "barra de progreso", Fig. 9-c), que controla el aumento de temperatura. En esa barra están los valores de la temperatura y un indicador de "frío, cálido o caliente".

```

a) private void LED1_CheckedChanged(object sender, EventArgs e)
{
    if (flag == 1)
    {
        if (LED1.Checked == true)
        {
            serialPort1.WriteLine("1");
        }
        else
        {
            serialPort1.WriteLine("2");
        }
    }
}

b) private void Timer1_Tick(object sender, EventArgs e)
{
    if (flag == 1)
    {
        serialPort1.WriteLine("0");
        times++;
        if (time == 10)
        {
            time = 0;
            chart1.Series[0].Points.Clear();
            chart1.Series["TEMPERATURE"].Points.Add(temperatureProbar);
        }
    }
}

c) private void Progressbar1_Click(object sender, EventArgs e)
{
    progressBar1.Value = temperatureProbar;
    if (temperatureProbar <= 30)
    {
        label1.Text = "TEMPERATURE: " + temperatureProbar.ToString() + "°C = COLD";
    }
    if (temperatureProbar > 30 && temperatureProbar < 50)
    {
        label1.Text = "TEMPERATURE: " + temperatureProbar.ToString() + "°C = WARM";
    }
    if (temperatureProbar >= 50)
    {
        label1.Text = "TEMPERATURE: " + temperatureProbar.ToString() + "°C = HOT";
    }
}
    
```

FIGURA 9

a) Programación de cada "Herramienta" (LED), b) Programación del elemento "Temporizador" c) Programación del elemento "Barra de progreso".

Finalmente, el software funciona midiendo la temperatura que se muestra en el "Gráfico" y la "Barra de programación" y tiene cuatro "cuadros de selección", que pueden asignarse a las diferentes tareas que el usuario pueda necesitar, en este caso se representan con "LED" (Fig. 10).

Registro de temperatura usando Arduino. El código de programación Arduino se muestra en la Fig. 11. Consiste en identificar el valor del puerto serie y asignarlo a la variable "datos". Después de identificar este valor, el programa realizado en Visual Basic lo registra y lo grafica en la interfaz.

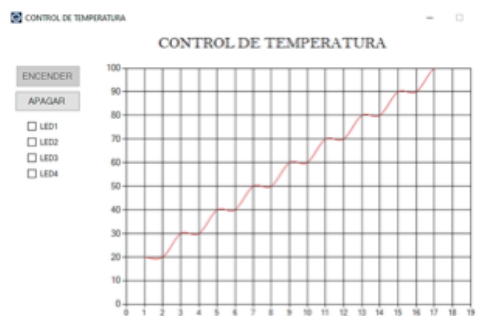


FIGURA 10

Interfaz de grabación de temperatura en funcionamiento.

```

int LM35; int dato; int led1=9; int led2=10; int led3=11; int led4=12; float TEMP;
void setup() { Serial.begin(9600);}
void loop() {
    dato=Serial.read();
    if (dato == "1") (digitalWrite(led1,HIGH));
    if (dato == "2") (digitalWrite(led2,LOW));
    if (dato == "3") (digitalWrite(led3,HIGH));
    if (dato == "4") (digitalWrite(led4,LOW));
    if (dato == "5") (digitalWrite(led1,HIGH));
    if (dato == "6") (digitalWrite(led2,LOW));
    if (dato == "7") (digitalWrite(led3,HIGH));
    if (dato == "8") (digitalWrite(led4,LOW));
    if (dato == "9") {LM35 = analogRead(A0);
        TEMP = ((LM35 * 5000.0) / 1023) / 10;
        Serial.println(TEMP, 3);}}
    
```

FIGURA 11.

Código Arduino para registrar la temperatura.

Configuración electrónica para el registro de temperatura. Para la conversión del fenómeno físico de la temperatura a una señal eléctrica y luego adaptarla y leerla mediante la DAB es necesario utilizar el sensor "LM35". Este sensor es económico, accesible y utiliza un voltaje de alimentación de 5V. Además, es un sensor

de 3 pines, que proporciona una salida de 10 mV por cada ° C. La configuración electrónica se muestra en la Fig. 12, y la configuración de la implementación se muestra en la Fig. 13.

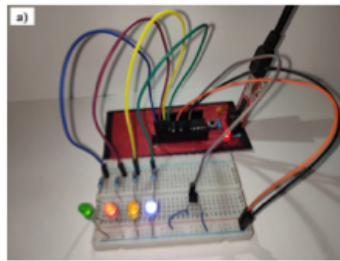


FIGURA 12.
Configuración electrónica para registrar la temperatura.

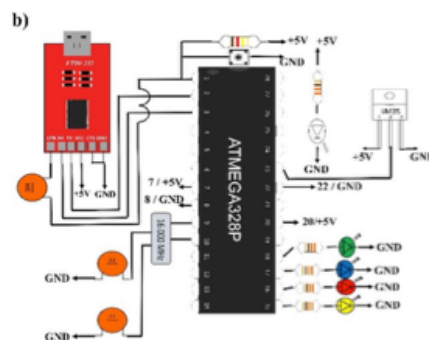


FIGURA 13
Diagrama esquemático para configurar la electrónica y registrar la temperatura.

Implementación de DAB y Software de Control de Servomotor. Servomotor controlado por Visual Studio. Para esta aplicación, la posición de un servomotor debe ser controlada. Básicamente, la operación es por medio de la comunicación en serie. La cual permite enviar los valores correspondientes a la posición deseada por el usuario. Un servomotor se controla mediante modulación de ancho de pulso (PWM) y, según el nivel de voltaje, se mueve de posición, comúnmente mediante movimientos de 0 a 180 °.

Para el diseño de la interfaz gráfica, primero se crea una función que permitirá configurar el puerto serie para la comunicación, esta vez se realizará mediante la configuración del puerto deseado (Fig. 14-a). Una vez que se realiza la comunicación, simplemente se programa el elemento llamado "TrackBar", que funcionará solo si el puerto está "abierto", escribiendo el valor proporcionado por "TrackBar" y mostrando el valor en una "Label o etiqueta" (Fig. 14-b).

```

a) private void PortProperties()
{
    Port = new SerialPort();
    Port.PortName = "COM5";
    Port.BaudRate = 9600;
    try
    {
        Port.Open();
    }
    catch (Exception e1)
    {
        MessageBox.Show(e1.Message);
    }
}

b) private void TrackBar_Scroll(object sender, EventArgs e)
{
    if (Port.IsOpen)
    {
        Port.WriteLine(Val_TrackBar.Value.ToString());
        Label1.Text = "DEGREE = " + Val_TrackBar.Value.ToString();
    }
}
    
```

FIGURA 14.

a) Función para configurar el puerto de comunicación, b) Programación del “TrackBar”. Una vez hecho esto, simplemente se ejecuta el software y se mueve el puntero a través de la barra (Fig. 15).

Una vez hecho esto, simplemente se ejecuta el software y se mueve el puntero a través de la barra (Fig. 15).



FIGURA 15.

Interfaz de control del servomotor.

Servomotor controlado por Arduino. El código de programación para Arduino se muestra en la Fig. 16. En este código se agrega la biblioteca "<Servo.h>".

```

Servomotor
#include <Servo.h>
Servo myservo;
int val;
void setup() {
    Serial.begin(9600);
    myservo.attach(10);}
void loop() {}
void serialEvent() {
    val = Serial.parseInt();
    if (val!=0){ myservo.write(val);}}
    
```

FIGURA 16.

Código de programación Arduino para el control del servomotor.

Configuración electrónica para controlar el servomotor. La configuración electrónica para controlar la posición de un servomotor se muestra en la Fig. 17. Mientras que el diagrama esquemático de las conexiones eléctricas se muestra en la Fig. 18



FIGURA 17.
Configuración electrónica para controlar un servomotor.

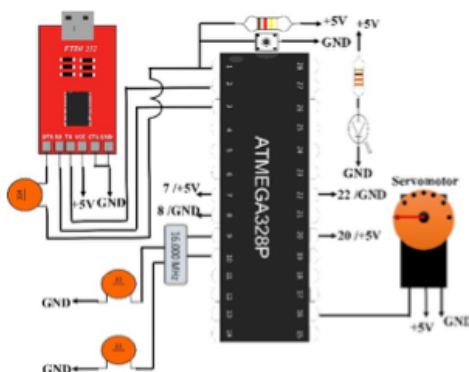


FIGURA 18.
Diagrama esquemático para controlar un servomotor.

Implementación de un medidor de pH con sensor de temperatura y configuración de agitación. Finalmente, esta última sección presenta un nuevo medidor de pH casero con detección de temperatura y control de agitación. Esta configuración casera tiene un bajo costo, y es muy fácil de hacer por maestros, investigadores y estudiantes. Además, los ejemplos anteriores sirven como base para que los maestros diseñen nuevas configuraciones didácticas y eviten pagar altos costos por equipo de laboratorio. La configuración final se presenta en la Fig. 18, y el diagrama electrónico se presenta en Fig. 19.

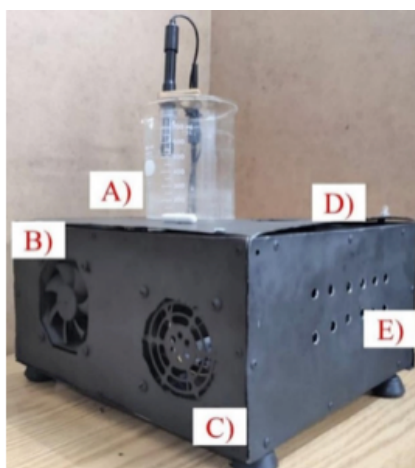


FIGURA 18.
Medidor de pH con sensor de temperatura y sistema de agitación, A) Parrilla (Resistencia) B) Ventilador Expulsor de Calor C) Sistema de Rejillas para Ventilación. D) Ventilador Enfriador de Placa Electrónica y TRIAC. E) Placa Electrónica.

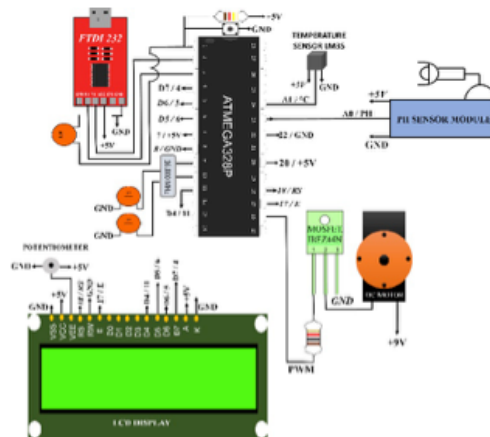


FIGURA 19
a) Diagrama electrónico.

Criterio de calibración del sensor de pH. Para la calibración del sensor de pH fueron considerados los siguientes criterios. Se realiza el enjuague del electrodo en agua destilada. Después, se realiza una lectura inicial, es decir, se introduce el sensor de pH en una disolución certificada, conocida también como referencia, y con un valor de pH conocido. Si el valor arrojado en la pantalla LCD coincide con la referencia, entonces se prosigue a determinar el PH de otra disolución. Se determinará otro valor de pH conocido. En caso de que el primer valor obtenido no coincida con el valor de la referencia, se procede a modificar la ecuación de calibración que es programada desde el Arduino IDE, y es con la cual se realiza la conversión voltaje-pH. Para la calibración de este sensor también se debe tomar en cuenta el valor de la incertidumbre. Cabe mencionar que, para tener una respuesta adecuada del sensor se recomienda generar un plan de mantenimiento y mantener el equipo en funcionamiento ideal. El sensor de pH se debe mantener húmedo y en una solución de pH de 4. No se debe guardar el electrodo en agua destilada, porque causaría que los iones resbalaran por el sensor y el electrodo se volvería inservible. Para la calibración del sensor también fueron consideradas las normas NMX-F-317-S-1978 y NMX-F-266-SCFI-2012.

Perspectivas. La perspectiva del presente manuscrito pretender ser un documento base para la comunidad universitaria, es decir, se busca presentar de manera pedagógica, clara y gradualmente como desarrollar y armar equipo de experimentación. Haciendo posible que docentes puedan implementar arreglos experimentales sencillos para realizar prácticas de las materias que imparten con los estudiantes. En el caso de docentes-investigadores, que no tengan bases profundas en electrónica, es posible que incorporen arreglos experimentales complementarios a equipos de laboratorio, y con ello realizar mediciones mas complejas. Finalmente, para los estudiantes busca ser un documento introductorio a tópicos de electrónica y química.

CONCLUSIONES.

En este trabajo, se muestra de manera clara y concreta el diseño e implementación de una placa de adquisición de datos de bajo costo. Además, a través de tres aplicaciones simples, se dieron a conocer los principios básicos para que alguien sin un profundo conocimiento en electrónica o programación pueda desarrollar su propio equipo de bajo costo. En cada aplicación, se diseña la interfaz gráfica para el control de diferentes actuadores. Finalmente, en este trabajo se presentó un nuevo medidor de pH casero con detección de temperatura y sistema de agitación. Este sistema casero es de bajo costo y es muy fácil de hacer por maestros, investigadores y estudiantes. Además, los ejemplos anteriores sirven como base para que los maestros diseñen nuevos sistemas didácticos y eviten pagar altos costos por equipo de laboratorio.

AGRADECIMIENTOS

Queremos agradecer a Christian Centeno-Alvarado por su valiosa ayuda.

REFERENCIAS

- [1] Cheng Yung-hsun and Fang Chen Tung. (2018), *Enhancing classroom management through parental involvement by using social networking apps*, South African Journal of Education. No. 38(2), p. 1427-1441. <https://doi.org/10.15700/saje.v38ns2a1427>
- [2] Sackstein Suzanne, Spark Linda and Jenkins Amy, (2015), *Are e-books effective tools for learning? Reading speed and comprehension: iPad vs. paper*, South African Journal of Education, No. 35(4), p. 1202-1216.
- [3] Hercog D and Gergiç B, (2014), *A Flexible Microcontroller-Based Data Acquisition Device*, Sensors, No. 14, p. 9755-9775. <https://doi.org/10.3390/s140609755>
- [4] Oates M, Ruiz-Canales M, Ferrández-Villena M and Fernández-López A, (2017), *A low cost sunlight analyser and data logger measuring radiation*, Computers and Electronics in Agriculture, No. 143, p. 38–48.
- [5] Bajer L and Krejcar O, (2015), *Design and Realization of Low Cost Control for Green house Environment with Remote Control*, IFAC-Papers On Line, No. 48(4). p. 368–373. <https://doi.org/10.1016/j.ifacol.2015.07.062>
- [6] Schubert T, D'Ausilio A and Canto R, (2013), *Using Arduino microcontroller boards to measure response latencies*. Behav Res. No. 45. p. 1332–1346.
- [7] Kyung-Won Kim, Mi-So Lee, Mun-Ho Ryu and Jong-Won Kim, (2016), *Arduino-based automation of a DNA extraction system*, Technology and Health Care. No. 24, p. 105–112. <https://doi.org/10.3233/THC-151048>
- [8] Kavya Devarakonda P, Nguyen Alexxai V. Kravitz. (2016), *ROBucket: A low cost operant chamber based on the Arduino microcontroller*, Behav Res. No. 48, p. 503-509. <https://doi.org/10.3758/s13428-015-0603-2>
- [9] Grassini S, Corbellini S, Parvis M, Angelini E and Zucchi F, (2016), *A simple Arduino-based EIS system for in situ corrosion monitoring of metallic works of art*. Measurements. <http://dx.doi.org/10.1016/j.measurement.2016.07.0140263-2241>
- [10] Bueno-Hernández D, Rupesh K, Roberto M and Jean Louis Marty, (2017), *Low cost optical device for detection of fluorescence from Ochratoxin A using a CMOS sensor*, Sensors and Actuators B. No. 246, p. 606–614.
- [11] Kuan Wen-Hsuan, Tseng Chi-Hung, Chen Sufen and Wong Ching-Chang, (2016), *Development of a Computer-Assisted Instrumentation Curriculum for Physics Students: Using LabVIEW and Arduino Platform* J Sci Educ Technol, No. 25(427). <https://doi.org/10.1007/s10956-016-9603-y>
- [12] Durán#Muñoz H, Hernández#Ortiz M, Sifuentes#Gallardo C, Galván#Tejeda I, Sánchez#Zeferino R and Castaño#Meneses V, (2018), *Comparative study of kinetic parameters induced by different excitation sources: using a novel and user-friendly glow curve deconvolution spreadsheet*. J Mater Sci: Mater Electron, No. 29, <https://doi.org/10.1007/s10854-018-9226-6>
- [13] Muñoz I, Brown F, Durán-Muñoz H, Cruz-Zaragoza E, Durán-Torres B and Álvarez-Montaña V, (2014), *Thermoluminescence response and glow curve structure of Sc₂TiO₅ β-irradiated*. Elsevier Ltd, 0969-8043 <http://dx.doi.org/10.1016/j.apradiso.2014.03.011>
- [14] Poppe L, Eliason H and Hastings M, (2004), *A Visual Basic Program to Generate Sediment Grain-Size Statistics and to Extrapolate Particle Distributions*, Computers & Geosciences, No. 30, p. 791–795.
- [15] Megumi Irie, Tomohiro Terada, Toshiya Katsura, Satoshi Matsuoka and Ken-ichi Inui, (2005), *Computational modelling of H⁺-coupled peptidetransport via human PEPT1*. J Physiol, p. 429–439.