

ELECTRÓNICA DIGITAL II

PRÁCTICA CON ARDUINO: CONEXIÓN DE ARDUINO CON ETHERNET SHIELD

Esther Casero Rodríguez

Tercer curso en Ingeniería Electrónica Industrial y Automática

ÍNDICE

1. DESCRIPCIÓN DEL PROYECTO	3
2. BASE DE DATOS	3
3. SERVIDOR.....	4
4. FUNCIONES ESENCIALES EN PHP PARA ESTABLECER COMUNICACIÓN	5
5. CONEXIÓN DE ARDUINO A UNA BASE DE DATOS MYSQL	7
6. DESARROLLO DEL FORMATO DE LA PÁGINA WEB	10
7. MONTAJE	14

1. DESCRIPCIÓN DEL PROYECTO

Este proyecto consiste en usar los conocimientos de programación de Arduino para conseguir establecer la comunicación entre éste con la base de datos. En este caso, el proyecto trata de un regulador de temperatura y humedad de una cocina. Ambas magnitudes se van a regular mediante un humidificador y un ventilador. El usuario tendrá la posibilidad de visualizar los datos de temperatura y humedad en la cocina y activar cualquiera de los dos actuadores.

2. BASE DE DATOS

En esta práctica, se trabaja con una base de datos MySQL. MySQL es un sistema de gestión de bases de datos relacional de código abierto muy popular. El entorno que administra esta base de datos a través de una web es *phpMyAdmin*. Es un software herramienta que está escrito en PHP.

Desde la página web <http://176.32.230.52/phpmyadmin/> de este entorno se solicita un usuario y una contraseña. En el caso del usuario “p01”, la contraseña y el usuario para acceder a la base de datos son los siguientes:

Usuario: cl60-p01

Contraseña: EMGW ^ ^ zw!



Tras acceder con estos permisos de usuario, se observa a la izquierda el nombre de la base de datos (en este caso, “cl60-p01”). Una vez seleccionado este título, se tiene la opción de generar tablas donde se introduzcan los datos que Arduino va a necesitar para activar los actuadores y los datos de las lecturas que Arduino tome de los sensores.

En este proyecto, se han creado dos tablas. Una, con el nombre de “Tabla” y otra con el nombre de “valores”. En “Tabla” se introducen los valores del sensor de temperatura y humedad junto con la fecha que se actualiza tras cada refresco de la página. En “valores” se introducen los valores referentes a la activación de los actuadores como se explicará posteriormente.

La tabla “Tabla” cuenta con cuatro columnas: “ID”, “TEMPERATURA”, “HUMEDAD” y “FECHA”. La estructura de la tabla es la siguiente:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> 1	ID	int(11)			No	Ninguna	AUTO_INCREMENT	Cambiar Eliminar Primaria Unico Indice Espacial Más
<input type="checkbox"/> 2	TEMPERATURA	int(11)			No	Ninguna		Cambiar Eliminar Primaria Unico Indice Espacial Más
<input type="checkbox"/> 3	HUMEDAD	int(11)			No	Ninguna		Cambiar Eliminar Primaria Unico Indice Espacial Más
<input type="checkbox"/> 4	FECHA	int(11)			No	Ninguna		Cambiar Eliminar Primaria Unico Indice Espacial Más

Todas las filas se han declarado con el tipo *int* de tamaño 11 caracteres. “ID” se ha definido como una variable incremental sin embargo, en este caso no va a ser necesario puesto que los datos leídos procedentes de los sensores se va a actualizar o sobrescribir sobre los anteriores. De esta forma se evita la acumulación de valores sucesivos en la tabla.

La tabla “valores” cuenta exclusivamente con dos columnas: “ID” y “valor”. La estructura de la tabla es la siguiente:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> 1	ID	int(1)			No	Ninguna		Cambiar Eliminar Primaria Unico Indice Espacial Texto completo Más
<input type="checkbox"/> 2	valor	int(1)			Si	NULL		Cambiar Eliminar Primaria Unico Indice Espacial Texto completo Más

Esta tabla funciona de la misma forma que la anterior. La forma en la que se actualicen los datos vendrá dada por la programación PHP que esté disponible en el servidor tal y como se explicará posteriormente.

3. SERVIDOR

En el servidor se van a encontrar colgados todos los archivos y scripts que se encarguen de comunicar Arduino con la base de datos y viceversa, así como de la impresión de los datos por pantalla en la página que se usa para ello. También se incluye un script programado en HTML que contiene el formato de la página además de la carpeta que contiene toda la información multimedia.

Para subir todos los archivos y scripts al servidor se usa un programa denominado *Filezilla*. Este programa solicita al usuario el servidor, el nombre de usuario y la contraseña:

Servidor: ftp.wiringbits.com

Usuario: p01@wiringbits.com

Contraseña: EMGW ^ ^ zw!

Tras introducir los datos, se establece la comunicación con la base de datos.

Al final de la práctica los archivos que habrá subidos al servidor son:

Nombre de archivo	Tamaño...	Tipo de archivo	Última modificación	Permisos	Propietario...
..					
imagenes		Carpeta de archivos	13/05/2017 12:52:22	flcdmpe (0...	1689077 16...
index.php	11.831	Archivo PHP	14/05/2017 21:52:21	adfrw (0644)	1689077 16...
sensor.php	2.615	Archivo PHP	13/05/2017 15:07:49	adfrw (0644)	1689077 16...

Posteriormente se contará la utilidad de cada script.

4. FUNCIONES ESENCIALES EN PHP PARA ESTABLECER COMUNICACIÓN

A continuación, se va a describir cada función en programación PHP para establecer comunicación entre la base de datos y Arduino:

- Función que permite iniciar la conexión con la base de datos:

```
<?php
// Llamada a la funcion "conexion"
$conn = conexion();
// Creacion de la funcion "conexion"
function conexion() {
    $servername = "localhost";
    $username   = "cl60-p01";
    $password   = "EMGW^^zw!";
    $dbname     = "cl60-p01";

    $conn = new mysqli($servername, $username, $password, $dbname);
    if (!$conn) {
        die("Conexion fallada: " . mysqli_connect_error());
    }
    return $conn;
}
?>
```

- Función que inserta los datos de temperatura y humedad recogidos por el sensor junto con la fecha en la tabla "Tabla":

```
<?php
// Inicializacion de variables
$variable= 0;
$fecha= time();
$humedad= "15";
$temperatura= "50";

// Llamada a la funcion "insertar"
$variable = insertar($temperatura, $humedad, $fecha, $conn);
// Creacion de la funcion "insertar"
function insertar($temperatura, $humedad, $fecha, $conn){
    $ins = "INSERT INTO Tabla (TEMPERATURA, HUMEDAD, FECHA)
    VALUES ($temperatura, $humedad, $fecha)";
    if (mysqli_query($conn, $ins)) {
        $variable=1;
        echo "Si";
    }
    else{
        $variable=0;
        echo "No";
    }
    mysqli_close($conn);
    return $variable;
}
?>
```

- Función que actualiza un dato de temperatura en una determinada “ID” de la columna “TEMPERATURA” de la tabla “Tabla”:

```
<?php
// Inicializacion de variables
$variable2= 0;
$id= 2;
// Llamada a la funcion "actualizar"
$variable2= actualizar($id, $dato, $conn);

// Creacion de la funcion "actualizar"
function actualizar ($id, $dato, $conn) {

$sql = "UPDATE Tabla SET TEMPERATURA=$dato WHERE ID=$id";

if (mysqli_query($conn, $sql)) {
echo "Actualizacion satisfactoria";
$variable2=0;
} else {
echo "No hay conexion";
$variable2=1;
}

mysqli_close($conn);
return $variable2;
}
?>
```

- Función que recoge un dato de la columna “TEMPERATURA” de la tabla “Tabla” de una “ID” concreta:

```
<?php
// Inicializacion de variables
$variable3= 0;
$id= 2;
// Llamada a la funcion "recuperar"
$recu = recuperar($id, $conn);

// Creacion de la funcion "recuperar"
function recuperar ($id, $conn) {
$rep = "SELECT TEMPERATURA FROM Tabla WHERE ID=$id";
$resultado = mysqli_query($conn, $rep);
if (!$resultado) {
echo "No se pudo ejecutar la consulta: " . mysqli_error();
exit;
$variable3=0;
}
while ($fila =mysqli_fetch_assoc($resultado)) {

echo $fila["TEMPERATURA"];
$variable3=1;
}

mysqli_close($conn);
return $variable3;
}
?>
```

- Función que borra el contenido de la tabla “Tabla” correspondiente a la “ID” indicada.

```
<?php
// Inicializacion de variables
$variable4= 0;
$id= 2;
// Llamada a la funcion "delete"
$borrar= delete($conn, $id);

// Creacion de la funcion "delete"
function delete ($conn, $id) {
$del = "DELETE FROM Tabla WHERE ID=$id";
if (mysqli_query($conn, $del)) {
echo "Record deleted successfully";
$variable4=1;
}
mysqli_close($conn);
return $variable4;
}
?>
```

Hay que tener en cuenta que antes de llamar a las funciones correspondientes se debe llamar a la función conexión para establecer la comunicación con la base de datos.

5. CONEXIÓN DE ARDUINO A UNA BASE DE DATOS MYSQL

Es necesario que Arduino recoja los datos capturados por los sensores y los transfiera a la base de datos para posteriormente mostrarlos por pantalla con las funciones que se han visto en el apartado anterior y que, además, sea capaz de obtener un valor escrito sobre la posición de una de las tablas de la base de datos.

A continuación, se expone el programa completo usado en Arduino seguido de la explicación del código y de las librerías empleadas:

```
////////////////////////////////////
////////// LIBRERIAS //////////
////////////////////////////////////
#include <SPI.h>
#include <Ethernet.h>
#include <SimpleDHT.h>

////////////////////////////////////
////////// CONFIGURACION INTERNET //////////
////////////////////////////////////

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEA };
char server[] = "p01.wiringbits.com";
IPAddress ip(192, 168, 1, 140);
EthernetClient client;
```

La librería *SPI* permite la comunicación con dispositivos SPI. SPI (Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

La librería *Ethernet* se usa para manejar el Ethernet Shield que permite a Arduino conectarse a internet.

SimpleDHT es la librería por defecto del sensor de temperatura y humedad.

- ❖ Aquí se establece la dirección MAC del router. Es un identificador de 48 bits asignada a la tarjeta de red.
- ❖ Con una cadena de caracteres se define la dirección del servidor.
- ❖ En *IPAddress* se declara la IP asignada a Arduino.
- ❖ Con *EthernetClient* crea un cliente con el nombre de “cliente”.

```

////////////////////////////////////
///////// DECLARACION E INICIALIZACION DE VARIABLES //////////
////////////////////////////////////

int tiempo = 1000;
int pinDHT11 = 2;
SimpleDHT11 dht11;
byte temperature;
byte humidity;
byte data[40] = {0};
int valor1=0;

////////////////////////////////////
/////////                               SETUP                               //////////
////////////////////////////////////

```

```

void setup() {
  Serial.begin(9600);
  Ethernet.begin(mac);
  delay(tiempo);
}

```

La parte de *Setup* solo se inicia una vez y al comenzar el programa. En este momento, el monitor de Arduino se activa y también comienza la conexión con el servidor usando la dirección MAC fijada anteriormente.

```

////////////////////////////////////
/////////          FUNCION DE ENVIO DE DATOS          //////////
////////////////////////////////////

```

```

void temphum() {

  Serial.println("=====");
  Serial.println("Sample DHT11...");
  if (dht11.read(pinDHT11, &temperature, &humidity, data)) {
    Serial.println("Read DHT11 failed");
    return;
  }

  Serial.print("Sample RAW Bits: ");
  for (int i = 0; i < 40; i++) {
    Serial.print((int)data[i]);
    if (i > 0 && ((i + 1) % 4) == 0) {
      Serial.print(' ');
    }
  }

  Serial.println("");

  //Serial.print("Sample OK: ");
  Serial.print((int)temperature); Serial.print(" *C, ");
  Serial.print((int)humidity); Serial.println(" %");

  delay(2000);
}

```

Esta función denominada *temphum* recoge los valores captados por el sensor de humedad y temperatura.


```

////////////////////////////////////
/////////          ENVIO DE DATOS          \\\\\\\\\\\
////////////////////////////////////

```

```

void envio () {
  delay(4000);
  if (client.connect(server, 80)) {
    Serial.println("Conectando...");
    temphum();
    client.print("GET
http://p01.wiringbits.com/sensor.php?temperatura=");
    client.print(temperature);
    client.print("&humedad=");
    client.print(humidity);
    client.println(" HTTP/1.1");
    client.print("Host: ");
    client.print(server);
    client.println("User-Agent: Arduino-Ethernet");
    //client.println("Connection: close");
    client.println();
  }
  else {
    Serial.println("No hay conexion");
  }
}

```

Esta función denominada *envio* se encarga de comunicarse con el script que se encarga de recoger estos valores y subirlos a la base de datos que se encuentra dentro del servidor.

client.connect conecta Arduino al servidor y al puerto 80 por defecto.

client.print se encarga de escribir en la URL del buscador de internet “temperatura=n” y “humedad=m” siendo “n” y “m” los valores recogidos por el sensor almacenados en las variables de Arduino “temperature” y “humidity”. Todo ello debe escribirse en la misma línea de la dirección web.

Posteriormente se indica al cliente el servidor que se trata y el dispositivo conectado a la red (Arduino-Ethernet).

```

////////////////////////////////////
/////////          LOOP          \\\\\\\\\\\
////////////////////////////////////

```

```

void loop () {
  envio();
  if (client.connected()) {
    Serial.println(" Connected");
    while(client.available()) {
      delay(3000);
      Serial.println("Connection OK");
      if(client.find("<sensor>")) {
        valor1 = client.parseInt();
      }
      client.println( " HTTP/1.1");
      client.print( "Host: " );
      client.println(server);
      client.println( "Connection: close" );
      client.println();
      client.stop(); }
    }
    Serial.println("dato:");
    Serial.println(valor1);

```

El bloque *loop* se encarga de ejecutar las instrucciones en bucle. Primero llama a la función *envio* y posteriormente establece la conexión para recibir los datos que el script *sensor.php* recibe de la base de datos. Arduino accede a este dato desde un tag situado en la parte de debajo del código PHP del mismo script.

Si detecta que el cliente está conectado mientras este esté disponible busca lo que hay contenido en el tag llamado “sensor” y lo introduce en la variable *valor1*. Una vez hecho esto, se desconecta el cliente del servidor mediante *client.stop*.

```

if (valor1==1) {

digitalWrite(7,HIGH);

}
if (valor1==0) {

digitalWrite(7,LOW);

```

```
}  
  
if (valor1==2) {  
  digitalWrite(8,HIGH);  
}  
if (valor1==3) {  
  digitalWrite(8,LOW);  
}  
}
```

Si Arduino recibe un '1' activa el ventilador, si recibe un '0', lo desactiva. Si recibe un '2', activa el humidificador y si recibe un '3', lo desactiva. Para ello, el ventilador y el humidificador deben estar conectados en el pin 7 y 8 respectivamente. La activación de las salidas puede comprobarse fácilmente mediante el uso de leds como se verá más adelante. De esta forma, se observará que la comunicación se establece adecuadamente.

Para que el script *sensor.php* (que se encarga de la comunicación con la base de datos y Arduino) recoja los valores de temperatura y humedad escritos en la URL basta con recogerlos en dos variables (\$temperatura y \$humedad) de la siguiente manera:

```
$temperatura= $_GET['temperatura'];  
$humedad= $_GET['humedad'];
```

El tag se escribe fuera del código PHP que contiene las funciones principales dentro del script *sensor.php*. Todo lo que se imprime por pantalla a través de *echo* se introduce en la variable *valorI*:

```
<sensor>  
  <?php  
  
    echo $data;  
  ?>  
</sensor>
```

6. DESARROLLO DEL FORMATO DE LA PÁGINA WEB

La página web consta de cuatro botones para la activación y desactivación del ventilador y del humidificador y de la visualización de la temperatura, humedad y fecha.

La decoración está ambientada en un entorno de postres para relacionar la utilidad con el lugar de uso, es decir, tal y como se ha comentado en la descripción del proyecto, se trata de un regulador de temperatura y humedad de una cocina.

El diseño se ha elaborado en un programa denominado *Dreamweaver*. El resultado se muestra a continuación:



Dentro del script principal denominado *index.php* (al que llama el servidor por defecto) se incluye tanto código PHP como HTML. En la zonas del código HTML donde se pretende mostrar alguna variable procedente del código PHP se introduce `<?php ...?>` en cuyo interior se llama a las funciones correspondientes que deben situarse a parte del código HTML. Estas funciones en este caso, son las que recogen los datos de la base de datos que se pretenden mostrar en la tabla HTML.

El código PHP de las funciones es:

```
<?php
    $variable= 0;
    $fechaa=0;
    $variable2= 0;
    $id=10;
    $variable3=0;
    $variable4=0;
    $variable6=0;
    $activar = $_POST['activacion'];

    // FUNCION QUE ESTABLECE LA CONEXION

function conexion(){
    $servername = "localhost";
    $username = "cl60-p01";
    $password = "EMGW^^zw!";
    $dbname = "cl60-p01";

    $conn = new mysqli($servername, $username, $password, $dbname);
    if (!$conn) {
        die("Conexion fallada: " . mysqli_connect_error());
    }
    return $conn;
}
```

```
function recuperartemp ($id, $conn) {
    $rep = "SELECT TEMPERATURA FROM Tabla WHERE
ID=$id";

    $resultado = mysqli_query($conn, $rep);
    if (!$resultado) {
        echo 'No se pudo ejecutar la consulta: '
. mysqli_error();
        exit;
        $variable3=0;
    }
    while ($fila =mysqli_fetch_assoc($resultado)) {
        echo $fila["TEMPERATURA"];
        $variable3=1;
    }
    mysqli_close($conn);
    return $variable3;
}
```

Función que recoge la variable temperatura de la base de datos.

```
function recuperarhum ($id,$conn) {
    $rep = "SELECT HUMEDAD FROM Tabla WHERE
ID=$id";

    $resultado = mysqli_query($conn, $rep);
    if (!$resultado) {
        echo 'No se pudo ejecutar la consulta: '
. mysqli_error();
        exit;
        $variable2=0;
    }
    while ($fila =mysqli_fetch_assoc($resultado)) {
        echo $fila["HUMEDAD"];
        $variable2=1;
    }
    mysqli_close($conn);
    return $variable2;
}
```

Función que recoge la variable humedad de la base de datos.

```
function recuperarfecha ($id,$conn) {
    $rep = "SELECT FECHA FROM Tabla WHERE ID=$id";
    $resultado = mysqli_query($conn, $rep);
    if (!$resultado) {
        echo 'No se pudo ejecutar la consulta: '
. mysqli_error();
        exit;
        $variable7=0;
    }
    while ($fila =mysqli_fetch_assoc($resultado)) {
        $fecha= $fila["FECHA"];
        $variable7=1;
    }
    mysqli_close($conn);
    return $fecha;
}
```

Función que recoge la variable fecha de la base de datos.

```
$variableactualizar=0;
```

```
$conn = conexion();
```

```
$variableactualizar= actualizar($activar,$conn);
```

```
function actualizar ($activar,$conn) {

$var = "UPDATE valores SET valor=$activar WHERE ID=1";

if (mysqli_query($conn, $var)) {
//echo"Actualizacion satisfactoria";
$variableactualizar=0;
} else {
echo "No hay conexion";
$variableactualizar=1;
}

mysqli_close($conn);
return $variableactualizar;
}

?>
```

Función que introduce la variable de activación de los actuadores en la base de datos.

En el PHP dentro del HTML se llama a alguna de las tres funciones de arriba para mostrar el dato recogido de la base de datos. Un ejemplo es el siguiente:

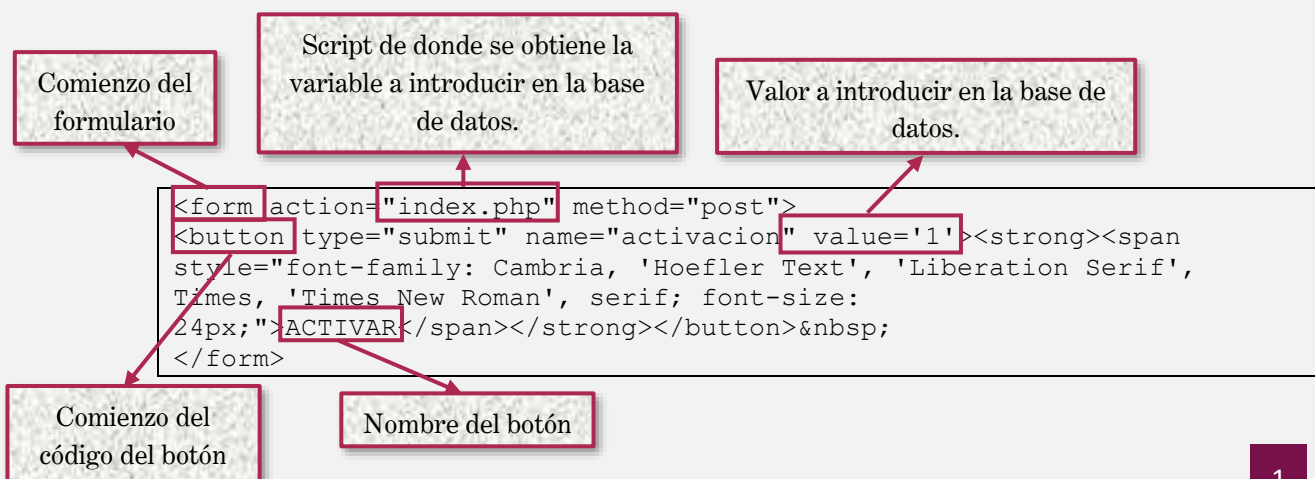
```
<?php
date_default_timezone_set('Europe/Madrid');
$id=1;
$conn = conexion();
$fexa = recuperarfecha($id,$conn);
$fexal= date("F j, Y, h:i A", mktime($fexa));
echo $fexal;

?>
```

En el lugar donde esté este trozo de código, se mostrará en la página web. Un caso especial es el de mostrar la fecha ya que no sólo hay que llamar a la función *recuperarfecha* como en el caso de mostrar la temperatura o la humedad sino que debe establecerse el formato de la misma para que convierta el valor asignado por la variable de la fecha introducido en la base de datos mediante *time()* a una hora y fecha propias de la zona europea (Madrid).

Esa variable de actuación (\$activar) cambia de valor al pulsar alguno de los botones de activación o desactivación y mediante la función *actualizar* se introduce en la base de datos para que Arduino acceda a ella.

En HTML los botones se crean con formularios. Un ejemplo es el siguiente:



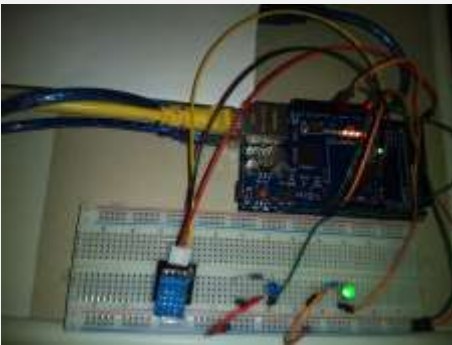
Para llevar el valor introducido por el botón a una variable que después se usa para actualizar la base de datos se asigna a la variable lo siguiente:

```
$activar = $_POST['activacion'];
```

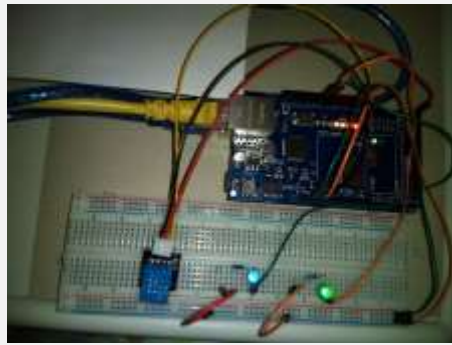
7. MONTAJE

Para comprobar si funciona y se transmiten los datos a Arduino se introducen dos leds. El led verde se activa cuando el botón del ventilador se pulsa. Lo mismo sucede con el led azul al activar el botón del humidificador.

Activación del ventilador



Activación de ambos actuadores



Activación del humidificador

