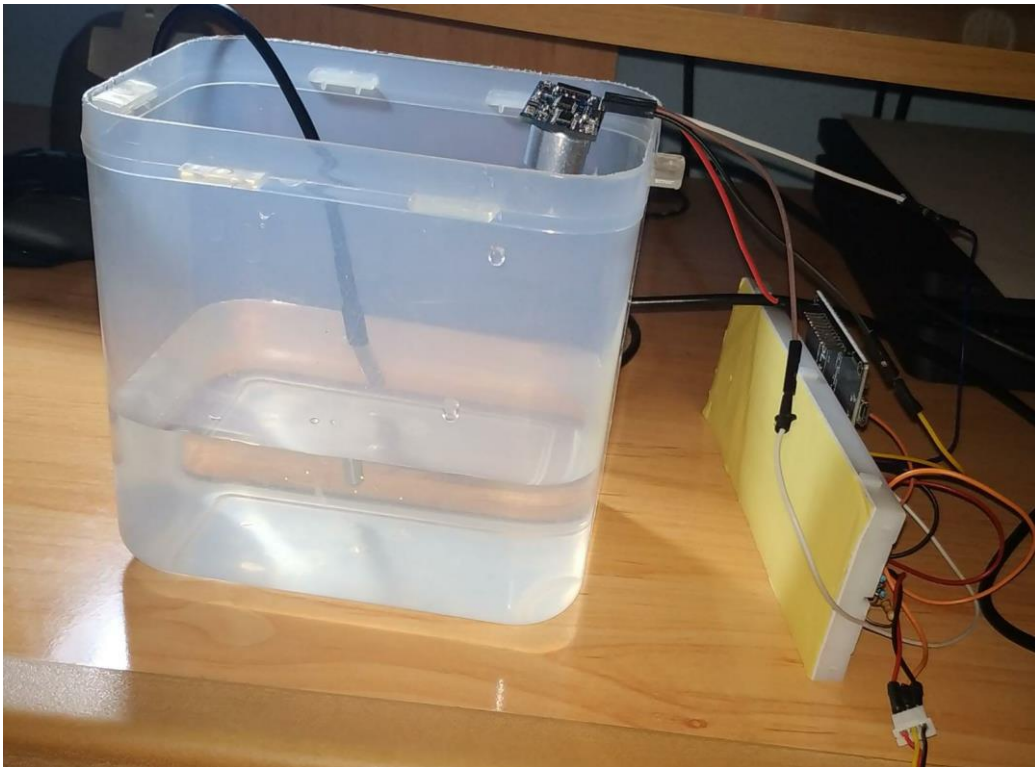


Miniproyecto con Arduino

Controlador de piscina con ESP8266



Autor: Antonio Blanco Castillo

ÍNDICE:

- **INTRODUCCIÓN (página 3)**
- **LISTA DE MATERIALES (página 3)**
- **MONTAJE (página 4)**
- **CÓDIGO (página 5)**
- **FUTUROS DESARROLLOS (página 7)**
- **CONCLUSIÓN (página 7)**
- **REFERENCIAS (página 7)**
- **IMÁGENES DEL CÓDIGO (página 8)**

INTRODUCCIÓN:

El proyecto consiste en, utilizando el ESP8266 conectado a wifi, crear una página web en la que se reciban y se muestren los datos de unos sensores conectados al chip, en este caso, al ser un miniproyecto, solo se utilizarán 2 sensores los cuáles nos permitirán obtener el nivel de altura del agua de una piscina, así como la temperatura del agua.

Para programar el chip mencionado anteriormente, utilizaremos el software “Arduino IDE”

LISTA DE MATERIALES:

- Protoboard:
- Varios cables para realizar conexiones en la protoboard
- Resistencia de 4700 ohmios
- ESP8266:



- Sensor HY-SRF05:



- Sensor DS18B20:



MONTAJE:

El montaje no es complejo, se conecta el ESP8266 a la protoboard, si se conecta en el centro no cabe, por lo que habrá que ponerla en un extremo quedando sin conectar una hilera de patas.

Posteriormente se conectarán los 2 sensores. La resistencia de 4700 se utilizará para el sensor de temperatura (DS18B20), en mi caso, no disponía de esa resistencia, por lo que en su lugar conecté en serie unas resistencias de 3300, 1000, 220 y 220, dando un total de 4740 lo cual sirve también.

El sensor de ultrasonidos (HY-SRF05), tiene 5 patas, que se conectarán con el ESP8266 de la siguiente forma:

Vcc → Vv, ya que necesita una alimentación de 5V para funcionar.

GND → G, tierra a tierra.

Trigger → D7, se puede usar cualquiera de las otras patas D

Echo → D6, se puede usar cualquiera de las otras patas D

El pin “OUT” quedará sin conectar.

El trigger se encarga de enviar pulsos de ultrasonido, mientras que el

Echo se encarga de recibir dicho pulso una vez haya rebotado.

El sensor de temperatura se conectará de la siguiente forma:

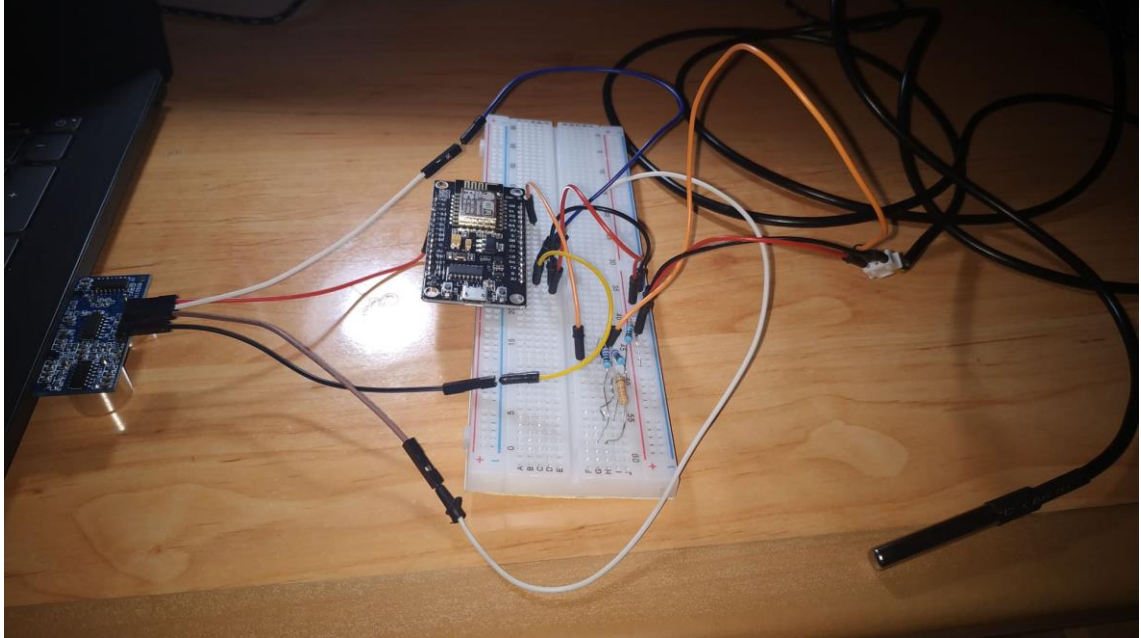
Cable negro (tierra) → GND

Cable rojo (Vdd) → 3V

Cable amarillo (Data) → D1, se puede usar cualquiera de las otras D

La resistencia irá conectada entre el cable amarillo y la pata D1.

El circuito montado completo quedaría así, teniendo en cuenta que en la foto no se aprecia perfectamente:



CÓDIGO:

En lo referente al código, para conectar los sensores con las patas D, hay que tener en cuenta que a cada pata se le asigna un número, por ejemplo, para conectar el Trigger con la pata D7, hay que poner `TRIGGER = 13`, puesto que 13 es el número asignado a D7.

Esta información y estos valores de los datos los encontramos en la referencia [1].

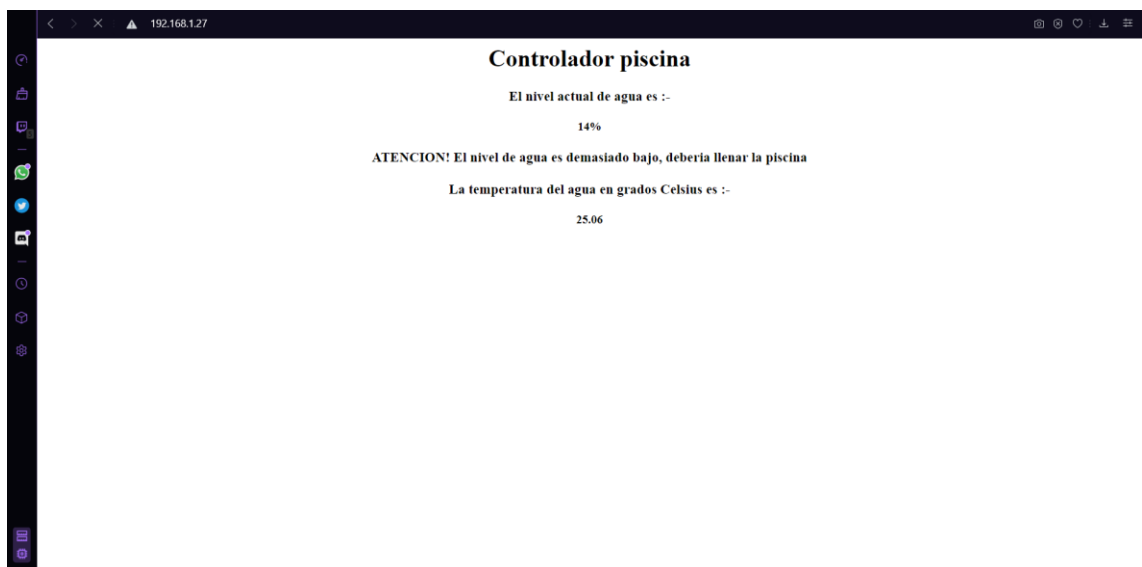
En la parte `setup` del código, se inician los sensores, la conexión wifi, y se crea la web en lenguaje html, la cual mostrará unas líneas de texto con la información del porcentaje de agua y la temperatura del agua.

Cuando el porcentaje del agua sea menor del 70%, se activará un `if` que hará que aparezca un mensaje de precaución avisando de que el nivel del agua es demasiado bajo.

Para que el cálculo del porcentaje funcione bien, será necesario ajustar el valor de la constante “profundidad”, para que se ajuste al tamaño de la profundidad de la piscina o recipiente en el que se encuentre nuestro sensor.

La dirección web aparecerá en el monitor de Arduino IDE al pulsar el botón reset del ESP8266

El resultado final de la web se verá así:



FUTUROS DESARROLLOS POSIBLES:

Algunas de las ideas que se me ocurren para futuras mejoras son:

- Incluir un apartado en la web que te permita introducir el valor de la profundidad de la piscina, ya que eso sería más cómodo que abrir el código, modificar el valor, y volver a cargar el código en el chip.
- Un sensor que mida el estado de limpieza del agua.
- Que cuando el volumen del agua esté por debajo de determinado porcentaje, se active automáticamente alguna forma que vuelva a llenar la piscina.

CONCLUSIÓN:

La conclusión que he sacado del trabajo, es que me ha servido para aprender y realizar por primera vez algo relacionado con IoT, recordando y aprendiendo nuevas cosas sobre la programación en Arduino y HTML, lo cual ha sido entretenido.

REFERENCIAS:

[1] https://github.com/esp8266/Arduino/blob/master/variants/d1_mini/pins_arduino.h

Otras referencias:

- <https://randomnerdtutorials.com/esp8266-ds18b20-temperature-sensor-web-server-with-arduino-ide/>
- https://naylampmechatronics.com/blog/46_Tutorial-sensor-de-temperatura-DS18B20.html
- <https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>
- https://naylampmechatronics.com/blog/56_usando-esp8266-con-el-ide-de-arduino.html
- <https://minibots.wordpress.com/2017/11/26/ejemplo-de-utilizacion-del-protocolo-mqtt-con-esp8266/>

IMÁGENES DEL CÓDIGO:

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <OneWire.h>
#include <DallasTemperature.h>

int TRIGGER = 13; //D7
int ECHO = 12; //D6
const int oneWireBus = 5; //D1
const int profundidad = 20; //profundidad de la piscina en cm

const char* ssid = "MiFibra-EE1A";
const char* password = "Skr6N9DD";
ESP8266WebServer server(80); //inicia el servidor en el puerto 80 (http port)

OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

String page = "";
int data;
float data2;
void setup(void) {

  pinMode(TRIGGER, OUTPUT);
  pinMode(ECHO, INPUT);
  delay(1000);
  Serial.begin(115200);
  sensors.begin(); //Inicio sensor temperatura
  WiFi.begin(ssid, password); //Inicio conexion wifi
  Serial.println("");
  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.print(".");
}

Serial.println("");
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
server.on("/", []() {
  page = "<head><meta http-equiv='refresh' content='3'></head><center><h1>Controlador piscina</h1><h3>El nivel actual de agua es :-</h3> <h4>" + String(data) + "%";
  if (data <= 70) {
    page = page + "</h4><h3>ATENCIÓN! El nivel de agua es demasiado bajo, debería llenar la piscina</h3> <h4>";
  }
  page = page + "</h4><h3>La temperatura del agua en grados Celsius es :-</h3> <h4>" + String(data2) + "%</h4></center></center>";
  server.send(200, "text/html", page);
});
server.begin();
Serial.println("Web server started!");
}

void loop(void) {
  sensors.requestTemperatures();
  float temperatura = sensors.getTempCByIndex(0);
  data2 = temperatura;
  delay(5000);

  digitalWrite(TRIGGER, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIGGER, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIGGER, LOW);
  long duration = pulseIn(ECHO, HIGH);
  float distance = (duration/2)/29.09;
  float agua = (profundidad-distance)/profundidad*100; //calcula el porcentaje de agua en función de la profundidad de la piscina
  data = agua; //calcula el porcentaje de agua en función de la profundidad de la piscina
  server.handleClient();
}
```