

30-6-2020

PROYECTO

Monitoreo del sensor de temperatura a prueba de agua Nodemcu ESP8266

INTERNET OF THINGS

Academico:

Sergio Francisco Hernandez Machuca

Integrantes:

Citlalmina Cortés López

Gretell Yahaira Rios Fuentes

Brandon Emilio Mora Salgado

A) Titulo: **Monitoreo del sensor de temperatura a prueba de agua Nodemcu ESP8266**

B) Descripción detallada del funcionamiento del sistema:

Cómo funciona el sensor

El sensor de temperatura **DS18B20** es un dispositivo que se comunica de forma digital. Cuenta con tres terminales: Vcc, GND y el pin Data. Este sensor utiliza comunicación por  protocolo serial digital OneWire. Esté protocolo de comunicación permite enviar y recibir datos utilizando un solo cable. A diferencia de otros, que utilizan dos o más líneas de comunicación digital.

El dispositivo está compuesto por un sensor DS18B20. Los sensores de temperatura de un cable como el DS18B20 son dispositivos que pueden medir la temperatura con una cantidad mínima de hardware y cableado. Estos sensores utilizan un protocolo digital para enviar lecturas precisas de temperatura directamente a su placa de desarrollo sin la necesidad de un convertidor analógico a digital u otro hardware adicional. Puede obtener sensores de un solo cable en diferentes factores de forma, como sondas impermeables y de alta temperatura; estos son perfectos para detectar la temperatura en muchos proyectos y aplicaciones diferentes. Y dado que estos sensores usan el protocolo de un solo cable, incluso puede tener varios de ellos conectados al mismo pin y leer todos sus valores de temperatura de forma independiente.

El sensor de temperatura a prueba de agua DS18B20 tiene tres cables

* El cable rojo es el cable VCC: el voltaje de funcionamiento es de 3 a 5 voltios. En mi caso, usaré 3,3 voltios.
* El cable amarillo es el cable de datos: generalmente conectamos una resistencia entre el cable de datos y el cable VCC, lo explicaré en el diagrama del circuito.
* El cable negro es el cable de tierra. Este cable está conectado a tierra del módulo wifi Nodemcu Esp8266.

Este sensor de temperatura es capaz de medir temperaturas que van desde -55 ° C a 125 ° C

Para este proyecto se le debe dar el crédito al canal de youtube llamado : [Electronic Clinic](https://www.youtube.com/channel/UCo1jouP-SEy7Pjrk1p-lDaQ) ,que es del cual se saco la idea y el código para poder llevarlo acabo:

<https://www.youtube.com/watch?v=zNRmDQ3PVNA&list=LLBPjCp-x7dpaWeWQL__qy_Q&index=2&t=606s>

Planeamos que este proyecto, esencialmente pueda ser utilizado en una estación metereologica.

No solo se puede utilizar para ello pero es como nuestro objetivo principal del proyecto , si bien nuestro sensor puede alcanzar de -55 ° C a 125 ° C lo podemos utilizar para monitorear temperaturas que entren en ese rango todo dependerá del usuario poseedor del dispositivo otra opción de uso podría ser para monitorear el agua de una pecera ya que existen ciertos peces que necesitan una temperatura especifica.

Consideramos que nuestro proyecto es sencillo pero facilita mas la vida de una persona ya que no es necesario estar cerca del sensor para ver la temperatura ya que se puede hacer por medio de internet gracias a una aplicación lo cual hace mas sencillo monitorear la temperatura , ya que si se eleva o baja rápidamente se daría cuenta el usuario de dicho suceso.

C) Detalle de elementos:

Para la realización de este proyecto, necesitaremos los siguientes componentes:

1. ESP8266 $94 <https://articulo.mercadolibre.com.mx/MLM-600597145-modulo-wifi-esp8266-lua-nodemcu-node-mcu-_JM?quantity=1#position=1&type=item&tracking_id=47acd956-44f7-47e5-ac24-b2d2a128beee>

# 2. SENSOR DS18B20 $55

<https://articulo.mercadolibre.com.mx/MLM-562888040-sensor-de-temperatura-digital-ds18b20-_JM?quantity=1#position=2&type=item&tracking_id=78c7a82b-260e-46ac-af9b-1e72e8e6d464>

3. RESISTENCIAS $10 <https://articulo.mercadolibre.com.mx/MLM-754270720-resistencias-12w-5-paquete-10-piezas-varios-valores-_JM?quantity=1&variation=49892546291&onAttributesExp=true#position=3&type=item&tracking_id=05b6f36a-a1ea-492e-ae7a-b201d591a84a>

Total: $ 159

D) Detalle de las herramientas:

El software que vamos a utilizar para llevar a cabo el proyecto será Arduino.

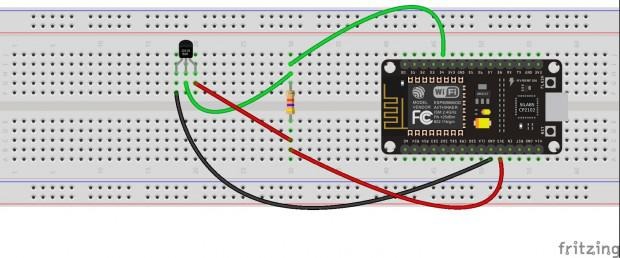
Arduino es una compañía de desarrollo de software y hardware libres, así como una comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan detectar y controlar objetos del mundo real. Al ser un software libre, aprovecharemos las facilidades que nos proporciona, interactuar con distintas ideas de otros proyectos o servicios que mejorarán el rendimiento de nuestro trabajo.

La enorme flexibilidad y el carácter libre y abierto de Arduino hacen que puedas utilizar este tipo de placas prácticamente para cualquier cosa, en nuestro caso el ESP8266.

*E) Bosquejo del esquema de construcción del proyecto:*

El proyecto todavia se encuentra en planeación pero he aquí unos diseños y esquemas:

*Circuito:*



*Código:*

#include <BlynkSimpleEsp8266.h>

#include <SimpleTimer.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#define BLYNK\_PRINT Serial // Comment this out to disable prints and save space

char auth[] = "1kiTyp8ZIBjEw12QY0yJhGiKtoM0-jtK";

/\* WiFi credentials \*/

char ssid[] = "Totalplay-A59F";

char pass[] = "A59F7F255PxBAn2d";

SimpleTimer timer;

#define ONE\_WIRE\_BUS 2 // DS18B20 on arduino pin2 corresponds to D4 on physical board "D4 pin on the ndoemcu Module"

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature DS18B20(&oneWire);

float temp;

float Fahrenheit=0;

void setup()

{

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

DS18B20.begin();

timer.setInterval(1000L, getSendData);

}

void loop()

{

timer.run(); // Initiates SimpleTimer

Blynk.run();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Send Sensor data to Blynk

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void getSendData()

{

DS18B20.requestTemperatures();

temp = DS18B20.getTempCByIndex(0); // Celcius

Fahrenheit = DS18B20.toFahrenheit(temp); // Fahrenheit

Serial.println(temp);

Serial.println(Fahrenheit);

Blynk.virtualWrite(V3, temp); //virtual pin V3

Blynk.virtualWrite(V4, Fahrenheit); //virtual pin V4

}