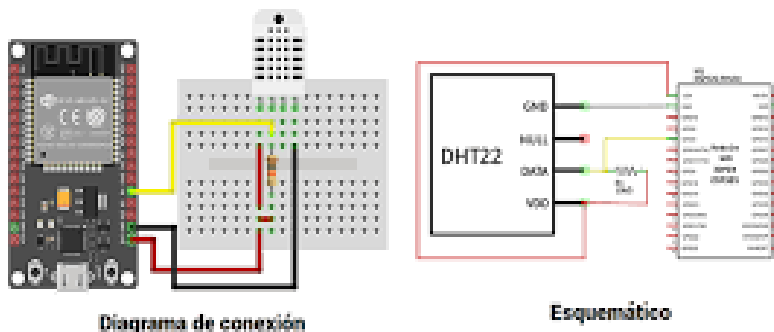


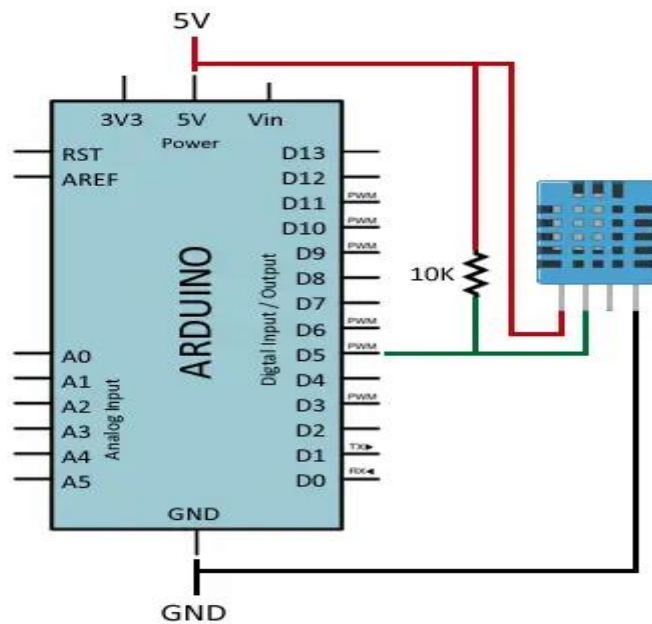
## ACTIVIDAD DE SENSORES

### EJERCICIO N.º 1

“Supervisión de la temperatura y la humedad de los alimentos con AWS IoT Core y Esp32”



Conectar el sensor es sencillo, simplemente alimentamos desde Arduino al sensor a través de los pines GND y Vcc del mismo. Por otro lado, conectamos la salida Output a una entrada digital de Arduino. Necesitaremos poner una resistencia de 10K entre Vcc y el Pin Output.



También  
observar que  
la placa  
hacemos  
con un

cuenta con sensores de temperatura HDC1880 – 12C

- Protocolo con 2 puertos digitales y unos sensores
- Cables y roles con sensores
- Roles de 5 voltios y unos puertos
- Wifi y cuenta con archivos secretos

podemos  
el requisito de  
Esp32, lo  
trabajando  
modulo que

El sensor que utilizaremos está compuesto por una sonda de dos electrodos y un módulo convertidor.

La conexión con nuestra placa Arduino es muy simple, si te fijas, el convertidor tiene dos lados, uno con cuatro pines y otro con dos. El lado de cuatro pines va hacia nuestra placa Arduino, y el lado de dos pines va hacia los electrodos.

Si estamos utilizando el sensor en modo analógico, la placa Arduino detectará el nivel de voltaje entregado por este, y lo convertirá en un número equivalente a la cantidad de humedad detectada entre 0 y 1023. Cuando el sensor detecta nada de humedad (seco) entrega un valor de 1023 y cuando detecta un valor de alta humedad entrega un valor de 0 (o cercano a 0).

El **HDC1080** es un sensor de **temperatura y humedad** con una **precisión excelente** y un consumo muy bajo. Es compatible con **Arduino y ESP8266** gracias a que utiliza el protocolo de comunicación **I2C**.

En términos generales, **cuanto más voltaje de alimentación y cuanta más temperatura, el consumo es mayor.**

Por último y dentro de lo que es el consumo es muy importante fijarse en los **tiempos de respuesta** que tiene un sensor. Básicamente viene a decirnos **cuánto tarda en tomar la medida** y volver al modo dormido de bajo consumo.



## “ CÓDIGO DE PROGRAMAR LA TEMP

```
#define _DEBUG_

#include <Wire.h>
#include "ClosedCube_HDC1080.h"

// Instancia a la clase
ClosedCube_HDC1080 hdc1080;

void setup() {
#ifdef _DEBUG_
  Serial.begin(9600);
#endif

  // Iniciar sensor al que se pasa la dirección I2C
  // Por defecto es un número de 7-bit 1000000 que
  // equivale al 40 en hexadecimal
  // Por defecto la precisión es de 14-bit para temperatura y humedad
  hdc1080.begin(0x40);
#ifdef _DEBUG_
  Serial.println(" ");
  Serial.println("Inicado sensor HDC1080");
#endif
}

void loop() {
  // Leer temperatura y humedad del sensor
  double temperatura = hdc1080.readTemperature();
  double humedad = hdc1080.readHumidity();

#ifdef _DEBUG_
  // Mostrar datos en el monitor serie
  Serial.print("Temperatura = ");
  Serial.print(temperatura);
  Serial.print("'C Humedad = ");
  Serial.print(humedad);
  Serial.println("'C");
#endif

  delay(3000);
}
```

AWS IoT Core es una plataforma en la nube administrada que permite a los dispositivos conectados interactuar con facilidad y seguridad con las aplicaciones en la nube y otros dispositivos. AWS IoT Core admite miles de millones de dispositivos y billones de mensajes, y es capaz de procesarlos y direccionarlos a

puntos de enlace de AWS y a otros dispositivos de manera fiable y segura. Con AWS IoT Core, sus aplicaciones pueden realizar un seguimiento de todos los dispositivos y comunicarse con ellos en todo momento, incluso cuando no están conectados.

**Conectividad entre dispositivos y la nube de AWS.** En primer lugar, con AWS IoT Core puede comunicarse de forma segura con dispositivos conectados, con baja latencia y baja sobrecarga. La comunicación puede escalar para alcanzar a tantos dispositivos como sea necesario. AWS IoT Core es compatible con los protocolos de comunicación estándar (por el momento, se admiten HTTP, MQTT, WebSockets y LoRaWAN). La comunicación está protegida con TLS.