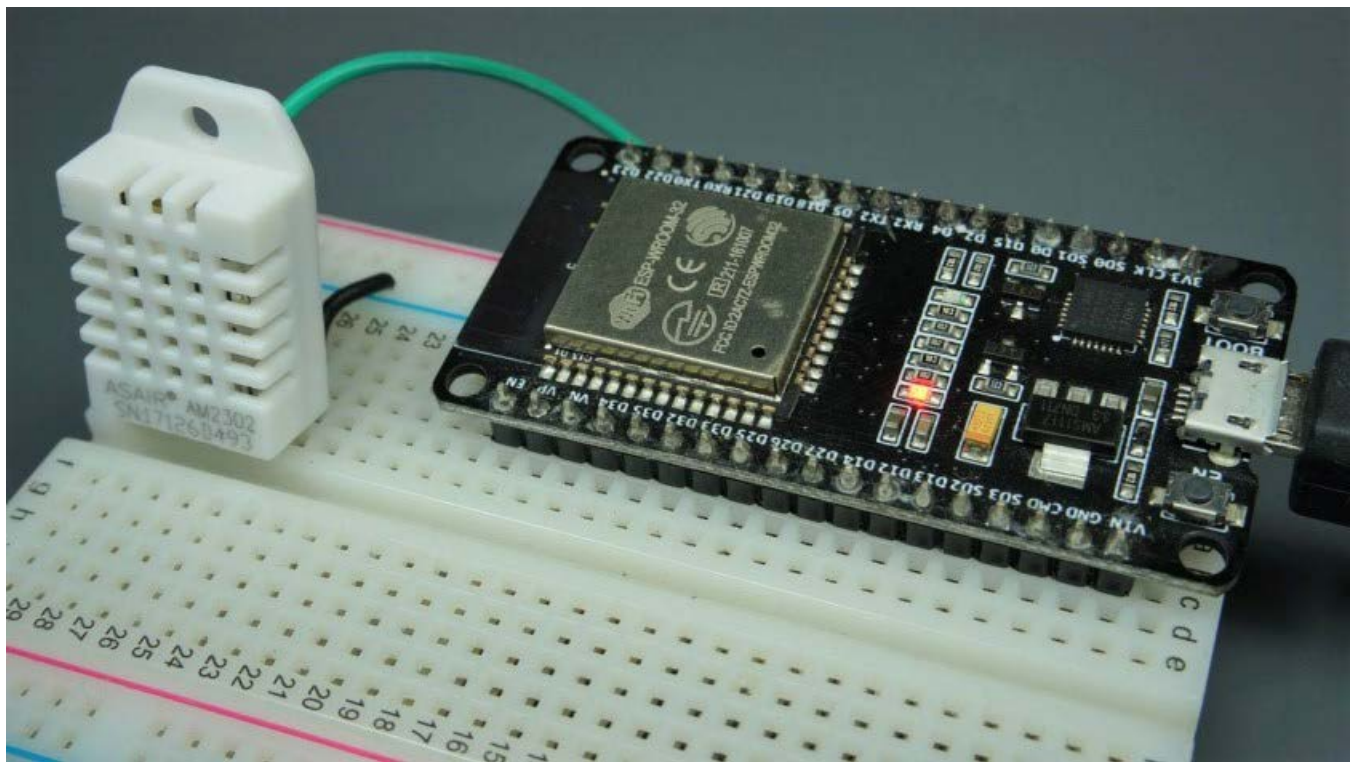


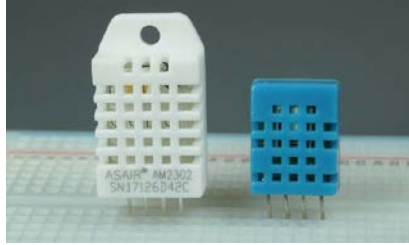
ESP32 avec capteur de température et d'humidité DHT11 / DHT22 utilisant Arduino IDE

Ce tutoriel montre comment utiliser les capteurs de température et d'humidité DHT11 et DHT22 avec l'ESP32 en utilisant Arduino IDE. Nous allons passer par une introduction rapide à ces capteurs, au brochage, au schéma de câblage et enfin au croquis Arduino.



Capteurs de température et d'humidité DHT11 et DHT22

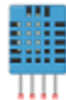

Les capteurs DHT11 et DHT22 sont utilisés pour mesurer la température et l'humidité relative. Ceux-ci sont très populaires parmi les fabricants et les amateurs d'électronique.



Ces capteurs contiennent une puce qui effectue une conversion analogique-numérique et crache un signal numérique avec la température et l'humidité. Cela les rend très faciles à utiliser avec n'importe quel microcontrôleur.

DHT11 vs DHT22

Le DHT11 et le DHT22 sont très similaires, mais diffèrent dans leurs spécifications. Le tableau suivant compare certaines des spécifications les plus importantes des capteurs de température et d'humidité DHT11 et DHT22. Pour une analyse plus approfondie de ces capteurs, veuillez consulter la fiche technique des capteurs.

	DHT11	DHT22
		
Écart de température	0 à 50 °C +/- 2 °C	-40 à 80 °C +/- 0,5°C
Plage d'humidité	20 à 90% +/- 5%	0 à 100% +/- 2%
Résolution	Humidité: 1% Température: 1°C	Humidité: 0,1% Température: 0,1°C
Tension de fonctionnement	3 - 5,5 V CC	3 - 6 V CC
Offre actuelle	0,5 - 2,5 mA	1 - 1,5 mA
Période d'échantillonnage	1 seconde	2 secondes
Prix	1 \$ à 5 \$	4 \$ à 10 \$

Le capteur DHT22 a une meilleure résolution et une plage de mesure de température et d'humidité plus large. Cependant, c'est un peu plus cher, et vous ne pouvez demander des lectures qu'à 2 secondes d'intervalle.

Le DHT11 a une portée plus petite et il est moins précis. Cependant, vous pouvez demander des lectures de capteur toutes les secondes. C'est aussi un peu moins cher.

Malgré leurs différences, ils fonctionnent de manière similaire, et vous pouvez utiliser le même code pour lire la température et l'humidité. Il vous suffit de sélectionner dans le code le type de capteur que vous utilisez.

Brochage DHT

Les capteurs DHT ont quatre broches comme indiqué dans la figure suivante. Cependant, si vous obtenez votre capteur DHT dans une carte de dérivation, il est livré avec seulement trois broches et avec une résistance de rappel interne sur la broche 2.



Le tableau suivant montre le brochage du DHT22 / DHT11. Lorsque le capteur est face à vous, la numérotation des broches commence à 1 de gauche à droite

Broche DHT Se connecter à

- | | |
|----------|---|
| 1 | 3,3 V |
| 2 | Tout GPIO numérique; connectez également une résistance de rappel de 10k ohms |
| 3 | Ne vous connectez pas |
| 4 | GND |

Pièces nécessaires

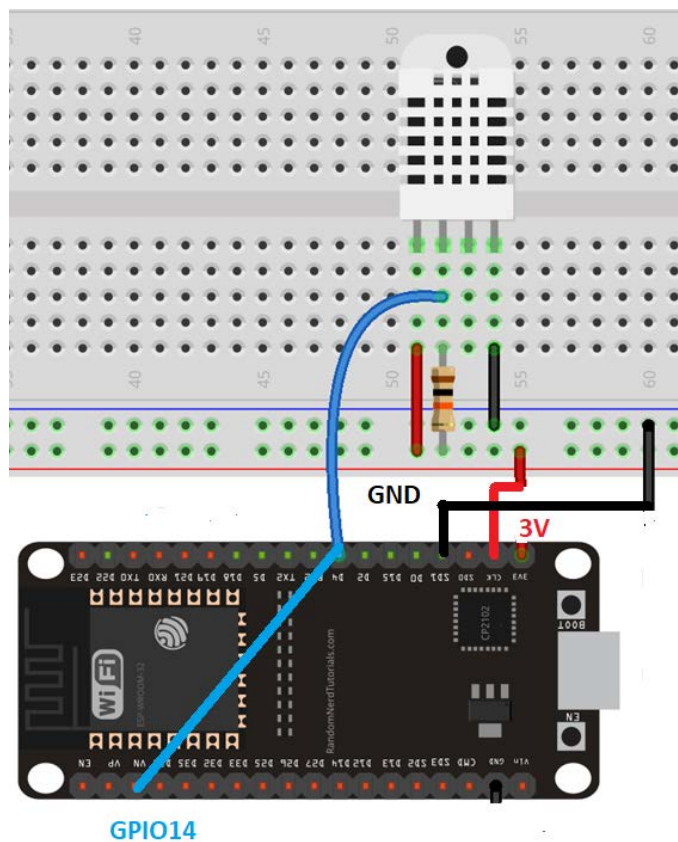
Pour suivre ce didacticiel, vous devez câbler le capteur de température DHT11 ou DHT22 à l'ESP32. Vous devez utiliser une résistance de rappel de 10k ohms.

Voici une liste des pièces dont vous avez besoin pour construire le circuit:

- ESP32
- Capteur de température et d'humidité DHT11 ou DHT22
- Résistance 10k Ohm
- Planche à pain
- Fils de connexion

Schéma électrique

Câblez le capteur DHT22 ou DHT11 à la carte de développement ESP32 comme indiqué dans le schéma suivant.



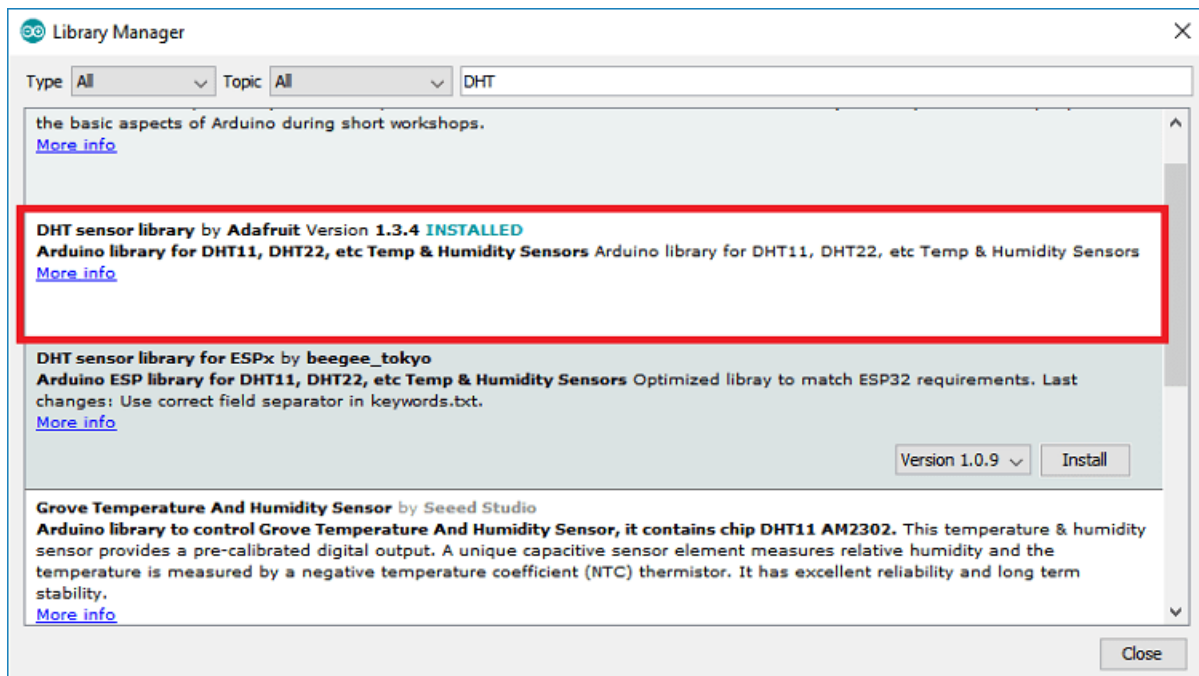
Dans cet exemple, nous connectons la broche de données DHT à GPIO 14. Cependant, vous pouvez utiliser n'importe quelle autre broche numérique appropriée.

Installation de bibliothèques

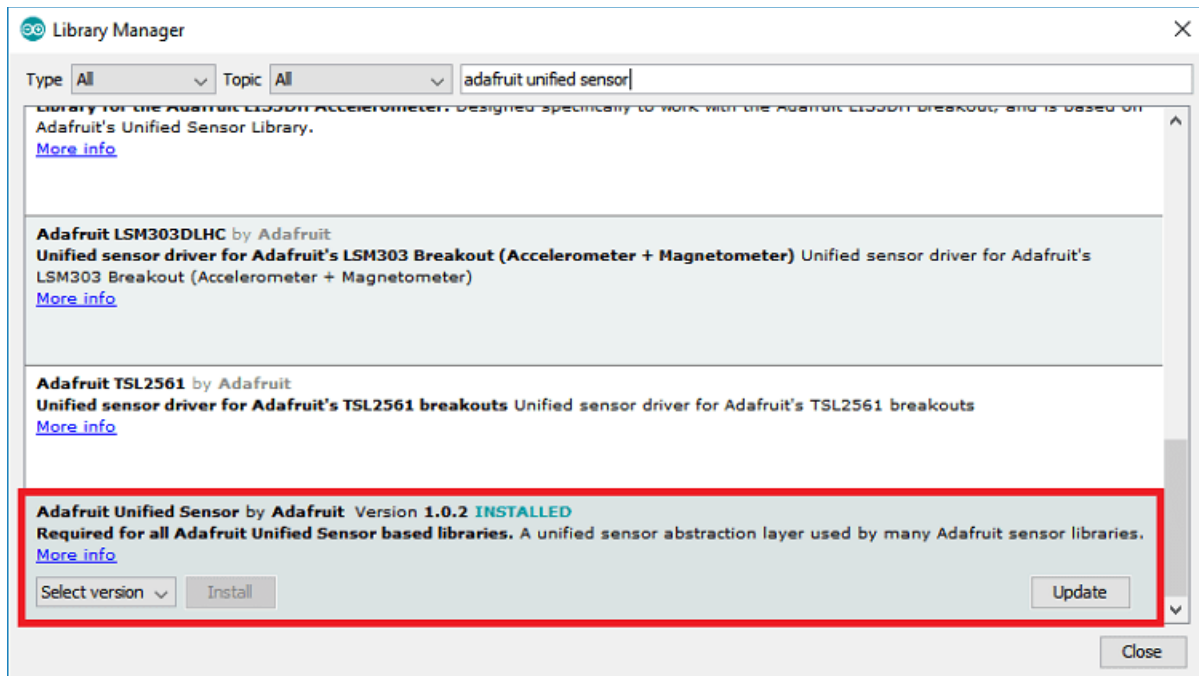
Pour lire à partir du capteur DHT, nous utiliserons la [bibliothèque DHT d'Adafruit](#) . Pour utiliser cette bibliothèque, vous devez également installer la [bibliothèque Adafruit Unified Sensor](#) . Suivez les étapes suivantes pour installer ces bibliothèques.

Ouvrez votre Arduino IDE et accédez à **Sketch > Inclure la bibliothèque > Gérer les bibliothèques** . Le gestionnaire de bibliothèque devrait s'ouvrir.

Recherchez « **DHT** » dans la zone de recherche et installez la bibliothèque DHT d'Adafruit.



Après avoir installé la bibliothèque DHT d'Adafruit, saisissez « **Adafruit Unified Sensor** » dans la zone de recherche. Faites défiler vers le bas pour trouver la bibliothèque et l'installer.



Redémarrez Arduino pour prendre en compte les modifications.

ESP32 Lecture Température and Humidité

Pour lire la température et l'humidité du capteur DHT, nous utiliserons un exemple basé sur la bibliothèque Adafruit DHT. Copiez le disponible dans le répertoire sur votre IDE Arduino.

Comment fonctionne le code

Tout d'abord il faut importer la bibliothèque DHT :

```
#include "DHT.h"
```

Ensuite, définissez la broche numérique sur laquelle la broche de données du capteur DHT est connectée. Dans ce cas, il est connecté à GPIO 14.

```
#define DHTPIN 14      // Digital pin connected to the DHT sensor
```

Ensuite, vous devez sélectionner le type de capteur DHT que vous utilisez. La bibliothèque prend en charge DHT11, DHT22 et DHT21. Décommentez le type de capteur que vous utilisez et commentez tous les autres. Dans ce cas, nous utilisons le capteur DHT22.

```
//#define DHTTYPE DHT11    // DHT 11
#define DHTTYPE DHT22    // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21    // DHT 21 (AM2301)
```

Créer un objet DHT appelé dht sur la broche et avec le type définis précédemment.

```
DHT dht(DHTPIN, DHTTYPE);
```

Dans le setup(), initialiser la vitesse à 9600bauds et envoyer le message DHT22 test.

```
Serial.begin(9600);  
Serial.println(F("DHTxx test!"));
```

Initialiser le capteur DHT.

```
dht.begin();
```

Le programme loop() démarre avec un délai de 2000 ms (2 secondes), car la période d'échantillonnage maximale du DHT22 est de 2 secondes. Ainsi, nous ne pouvons obtenir des lectures que toutes les deux secondes.

```
delay(2000);
```

La température et l'humidité sont renvoyées au format float. Nous créons des variables flottantes h, t et f pour enregistrer respectivement l'humidité, la température en Celsius et la température en Fahrenheit.

Obtenir l'humidité et la température est aussi simple que d'utiliser les méthodes readHumidity () et readTemperature () sur l'objet dht, comme indiqué ci-dessous:

```
float h = dht.readHumidity();  
// Read temperature as Celsius (the default)  
float t = dht.readTemperature();
```

Si vous souhaitez obtenir la température en degrés Fahrenheit, vous devez passer le paramètre true comme argument à la méthode readTemperature ().

```
float f = dht.readTemperature(true);
```

Il existe également une instruction if qui vérifie si le capteur a renvoyé des lectures de température et d'humidité valides.

```
if (isnan(h) || isnan(t) || isnan(f)) {  
    Serial.println(F("Failed to read from DHT sensor!"));  
    return;  
}
```

Après avoir obtenu l'humidité et la température, la bibliothèque dispose d'une méthode qui calcule l'indice de chaleur. Vous pouvez obtenir l'indice de chaleur en degrés Celsius et Fahrenheit comme indiqué ci-dessous:

```
// Compute heat index in Fahrenheit (the default)  
float hif = dht.computeHeatIndex(f, h);  
// Compute heat index in Celsius (isFahreheit = false)  
float hic = dht.computeHeatIndex(t, h, false);  
Serial.print(F("Humidity: "));  
Serial.print(h);
```

Enfin, imprimez toutes les lectures sur le moniteur série avec les commandes suivantes:

```
Serial.print(F("%  Temperature: "));  
Serial.print(t);  
Serial.print(F("°C "));  
Serial.print(f);  
Serial.print(F("°F  Heat index: "));  
Serial.print(hic);  
Serial.print(F("°C "));  
Serial.print(hif);  
Serial.println(F("°F"));
```

Mise en œuvre

Téléchargez le code sur votre carte ESP32. Assurez-vous d'avoir la bonne carte et le bon port COM sélectionnés dans vos paramètres Arduino IDE.

Après avoir téléchargé le code, ouvrez le moniteur série à une vitesse de transmission de 9600. Vous devriez obtenir les dernières lectures de température et d'humidité dans le moniteur série toutes les deux secondes.

