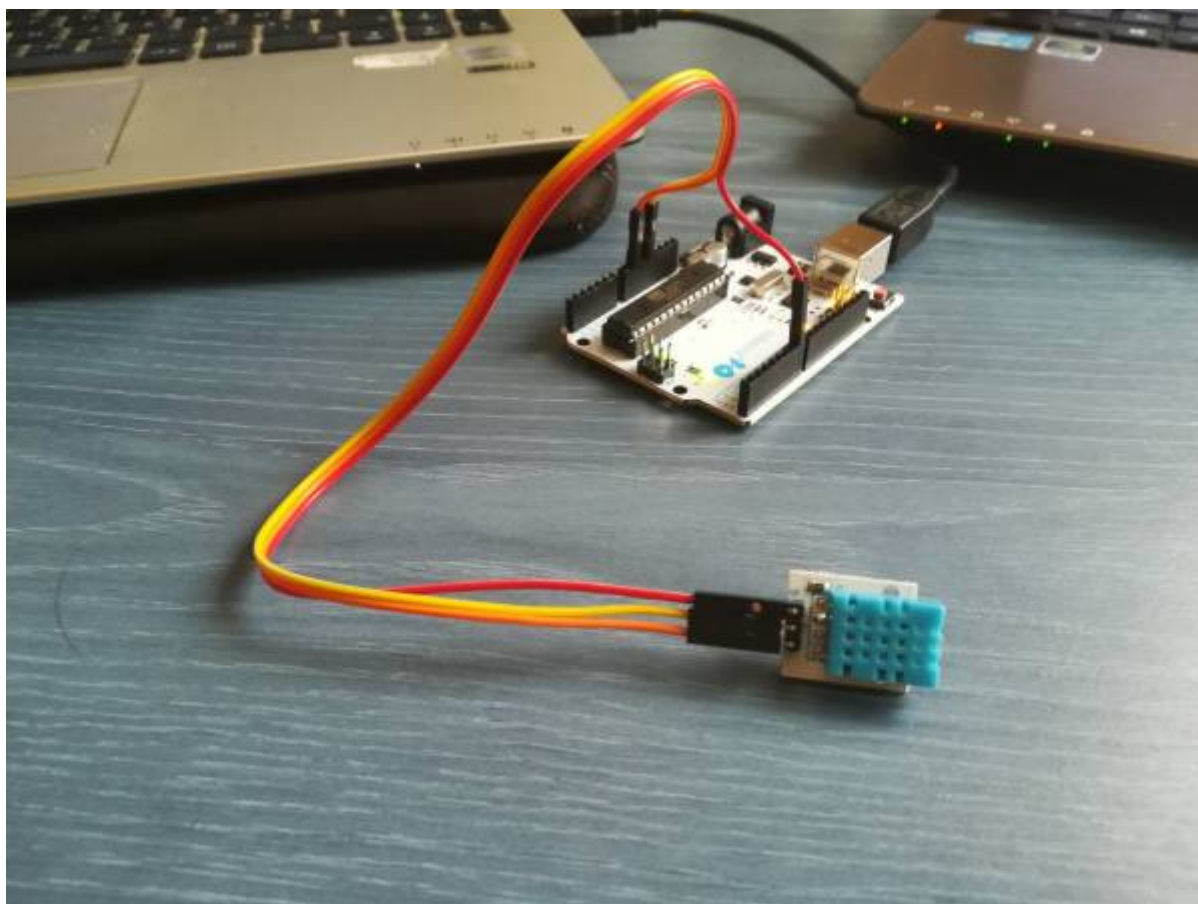


Utilisation du capteur de température et d'humidité

→ Etape précédente : [Prise en main de l'Arduino](#)

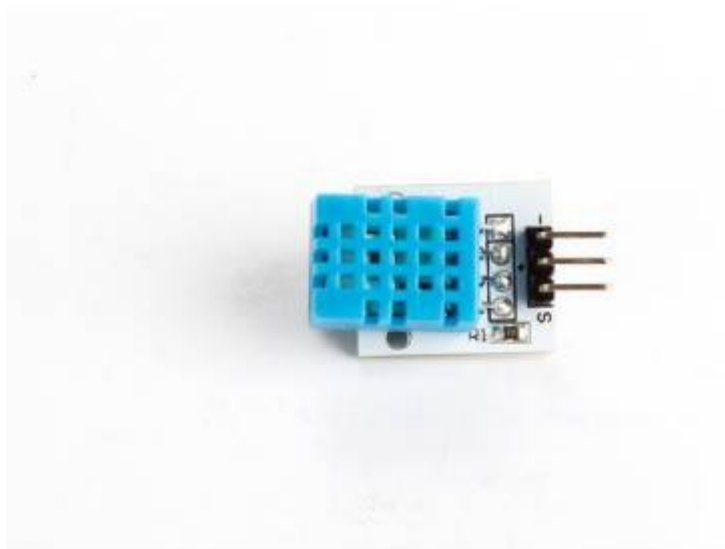
Dans le cadre de la construction du chauffage solaire aérothermique, un capteur de température et d'humidité est utilisé pour piloter de manière électronique la ventilation de l'air dans le panneau.



Fonctionnement du capteur

Nous allons ici utiliser un capteur de température DHT11¹⁾. Ce offre un bon rapport qualité-performance et est calibrée pour fonctionner sur une carte arduino car **son alimentation est en 5V**. Attention à ne pas alimenter ce capteur avec une tension d'alimentation de 12V qui détériorerait le capteur.

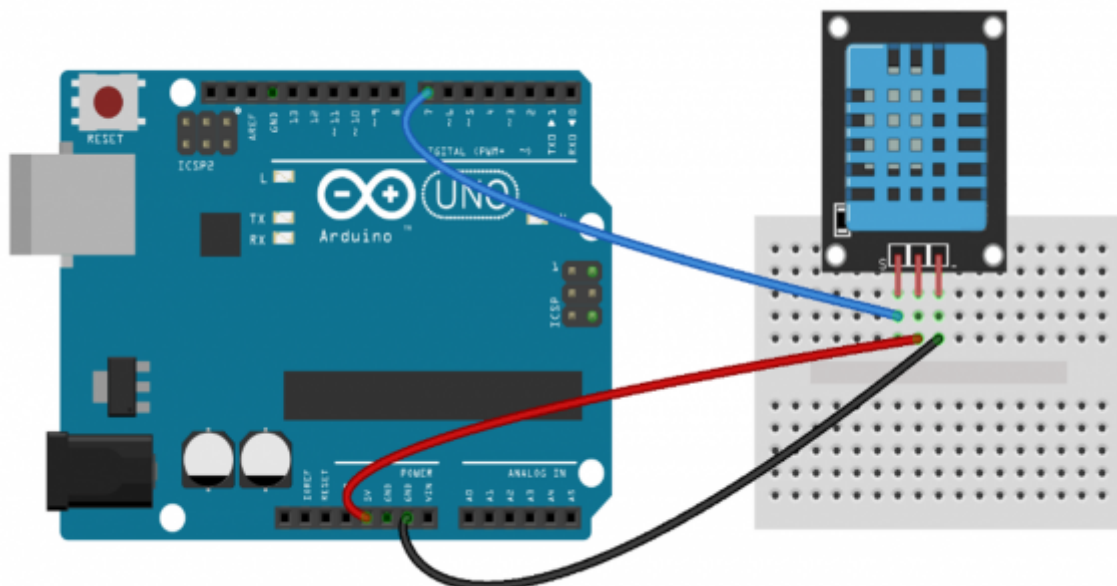
Sa plage de température est d'environ 0 à 50°C avec une erreur de $\pm 2^\circ\text{C}$. La plage de mesure de l'humidité relative est de 20 à 90% avec une erreur de $\pm 5\%$. Il existe un modèle plus précis : le DHT22. Il est de la même marque mais est deux fois plus cher. Sa plage de mesure de température et d'humidité s'étend respectivement à (-40 à 80°C $\pm 0.5^\circ\text{C}$) et à 0 à 100% HR ($\pm 2\%$)



Le capteur DHT11 est composé de 3 broches, respectivement de gauche à droite :

- Broche Signal
- Broche +
- Broche -

La broche signal doit être reliée à une piste de lecture numérique de la carte arduino (piste 7 sur la figure ci-dessous). La broche + doit être reliée à la masse (ground) de la carte Arduino repérée par les lettres GND. Enfin la broche - est reliée à l'alimentation 5V de la carte Arduino.



Installation de la librairie

Pour utiliser le capteur, ouvrez l'[IDE Arduino](#) puis :

- Installer ensuite la librairie dans l'IDE en cliquant sur : Croquis/Inclure une bibliothèque/Gérer les bibliothèques
- Rechercher la bibliothèque DHT sensor library de Adafruit et installer la.
- Télécharger le [code Lecture_T_h_final.ino](#) écrit pour lire la température

Une fois cette étape réalisée, vous pouvez désormais utiliser la librairie DHT. Nous allons voir en détail le programme écrit dans l'IDE.

Preamble du code

Ouverture de la librairie utilisée. Celle-ci comporte des commandes déjà configurées. La commande `#include` permet de charger la librairie dont le nom est compris entre les symboles `<>`

```
#include <DHT.h>
```

Déclaration du nom attribué au port de lecture numéro 7. Ici, le nom utilisé est `DHTPIN_1` mais on aurait pu utiliser `toto` ou n'importe quoi d'autre tant que l'ensemble du code est cohérent.

```
#define DHTPIN_1 7
```

Définition du type de capteur utilisé. Ici c'est un capteur DHT11 mais l'on aurait pu utiliser un capteur DHT22.

```
#define DHTTYPE DHT11
```

Commande qui attribue au nom `dht_1` un capteur DHT de type `DHTTYPE` et branché sur le port `DHTPIN_1`.

```
DHT dht_1(DHTPIN_1, DHTTYPE);
```

Initialisation

```
void setup()  
{
```

Définition du débit de données avec la commande `Serial.begin` en bits par seconde à une valeur de 9600 bits par secondes.

```
Serial.begin(9600);
```

Démarrage du capteur de température avec la commande `dht_1.begin()`.

```
dht_1.begin();  
}
```

Boucle infinie

```
void loop()
{
```

Création d'une nouvelle variable nommée `h_1` de type float (variable réelle avec une réelle dite flottante) et attribution de la valeur de l'humidité lue par le capteur avec la commande `dht_1.readHumidity()`

```
// Capteur #1
float h_1 = dht_1.readHumidity();
```

Création d'une variable `t_1` qui lit la température avec la commande `dht_1.readTemperature()`;

```
float t_1 = dht_1.readTemperature();
```

Affichage des valeurs dans la console avec la commande `Serial.print` Affichage du texte avec des apostrophes ("Comme ceci") pour afficher **Comme ceci** dans la console. Affichage de la valeur des variables `t_1` et `h_1` avec les commandes `Serial.print(t_1)` et `Serial.print(h_1)`

```
Serial.print(" Humidity #1: ");
Serial.print(h_1);
Serial.print(" %\t");
Serial.print("Temperature #1: ");
Serial.print(t_1);
```

Pour sauter une ligne à la fin de la ligne utiliser la commande `Serial.println` à la place de `Serial.print`

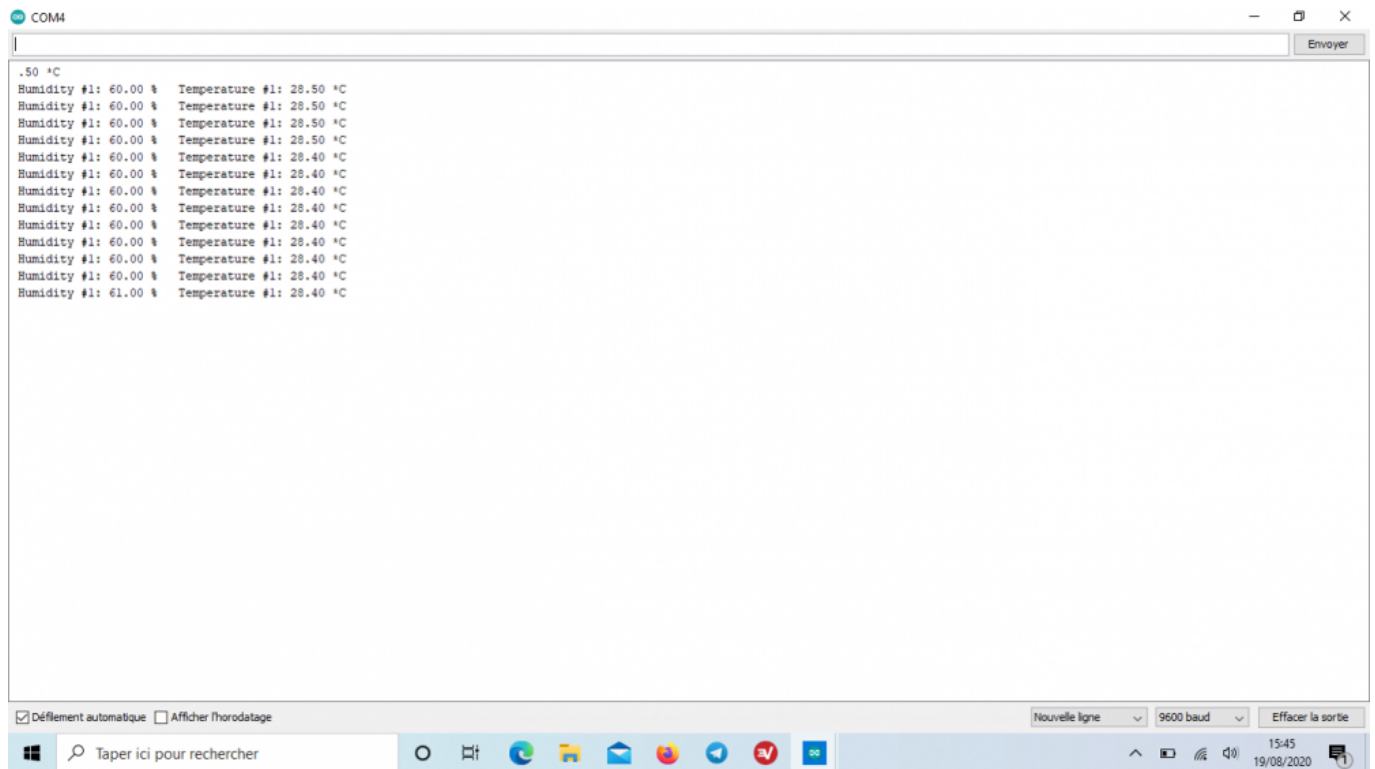
```
Serial.println(" °C ");
```

La commande `delay` permet de définir un laps de temps de pause permettant de définir la fréquence de mesure entre deux mesures. Ici nous utilisons 1000ms soit 1seconde.

```
delay(1000);
}
```

Exécution du code

- Vérifier/Compiler le code
- Téléverser le code
- Afficher les mesures de température dans le Moniteur Série



```
.50 °C
Humidity #1: 60.00 % Temperature #1: 28.50 °C
Humidity #1: 60.00 % Temperature #1: 28.50 °C
Humidity #1: 60.00 % Temperature #1: 28.50 °C
Humidity #1: 60.00 % Temperature #1: 28.50 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 60.00 % Temperature #1: 28.40 °C
Humidity #1: 61.00 % Temperature #1: 28.40 °C
```

Etape suivante: [Régulation d'un ventilateur 12 volts avec une carte Arduino](#)

1)
[Notice d'utilisation du capteur de température et d'humidité DHT 11 de Velleman VMA311](#)

From:

<https://wiki.enerlog.fr/> - **Wiki Enerlog**

Permanent link:

https://wiki.enerlog.fr/doku.php?id=espace_public:electronique:arduino_capteur_temp



Last update: **2021/05/19 14:40**