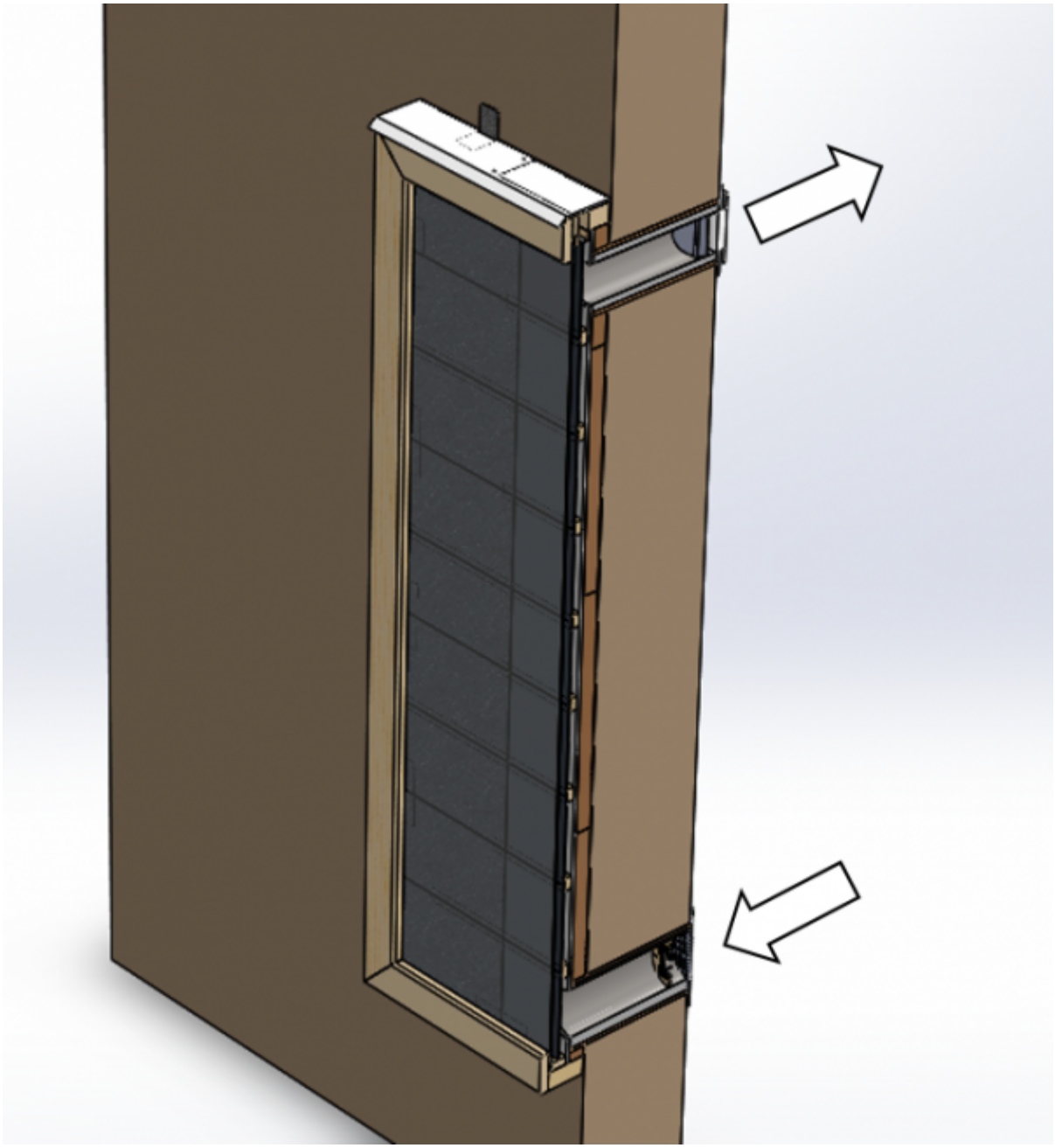
Régulation d'un ventilateur 12 volts avec une carte Arduino

Le fonctionnement du chauffage solaire électronique nécessite une gestion électronique permettant de piloter le ventilateur en fonction de la température au sein du panneau.

Pour le pilotage du ventilateur, deux capteurs de température sont utilisés :

- Le premier est placé en amont du ventilateur à l'intérieur de la conduite du bas du capteur. Sa température est proche de la température de l'habitat.
- Le second capteur de température est placé en aval du panneau solaire, au niveau de la sortie d'air.



Dans quelles conditions active t'on le ventilateur ?

Il existe trois cas de figure :

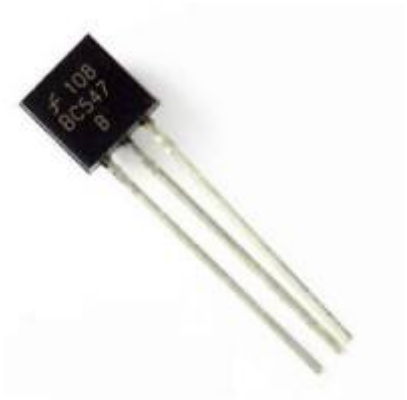
- Lorsque la température T1 du capteur en amont du ventilateur est inférieure à une température de confort défini 22°C et que la température en aval du panneau T2 est supérieure à 25°C, le panneau solaire est en mesure de chauffer l'habitat et le ventilateur est activé.
- Lorsque la température T1 du capteur en amont du ventilateur est supérieure à la température de confort de 22°C, il n'est pas nécessaire de chauffer et le ventilateur peut être coupé.
- Lorsque la température T1 du capteur en amont du ventilateur est inférieure à la température de 22°C mais que la température en aval du panneau est inférieure à 25°C, le panneau ne peut pas chauffer la pièce. Le ventilateur reste donc coupé pour ne pas consommer inutilement de l'électricité.

Liste de matériaux

Pour assurer le contrôle du ventilateur, nous allons utiliser :

- Une diode 1N4007

- Un transistor BC547



- Une carte Arduino ATmega 328 Uno

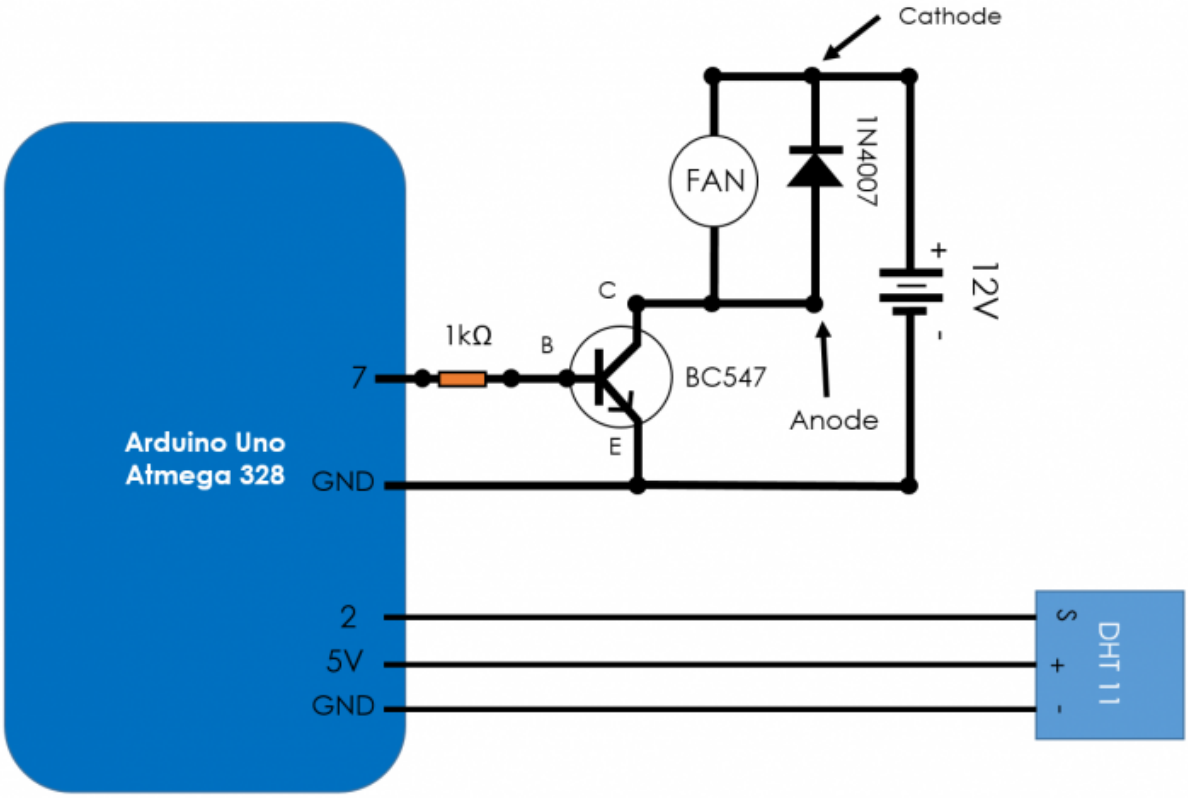
- Deux capteurs de températures DHT11

- Une résistance de 1kOhms

- Un convertisseur 230Volts-12Volts

- Un câble secteur 230Volts
- Câbles mâle-mâle
- Câbles femelle-mâle
- Un ventilateur 12V

Fonctionnement



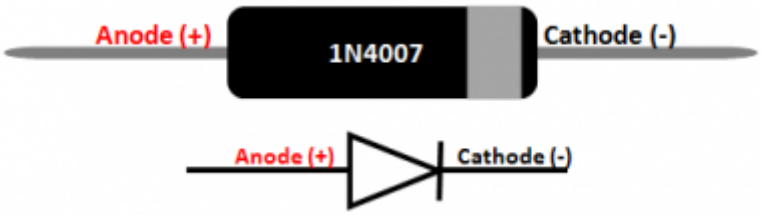
L'objectif est d'activer et de désactiver un ventilateur en fonction de la température au sein du capteur. Pour cela, un capteur de température, est branché sur une carte Arduino, afin de mesurer la température et de définir quand activer le ventilateur. Son fonctionnement est détaillé dans la présentation de l'utilisation du capteur de température et d'humidité du Wiki.

Sur le schéma de fonctionnement présenté ci-dessus, ce capteur de température DHT11 est présenté en bas à droite et est relié à l'alimentation 5V de la carte Arduino. **Attention à ne pas alimenter vos capteurs de température avec l'alimentation 12 Volts pour ne pas les endommager !**

Sur la partie du haut, un ventilateur est branché en parallèle d'une diode et en série d'une alimentation 12V et d'un transistor.

La diode joue ici un rôle de protection du ventilateur. Lorsque le ventilateur n'est plus alimentée, l'inducteur du ventilateur ne peut pas être déchargé lentement et cela peut endommager le

ventilateur. En installant une diode, on fournie un chemin pour le courant inductif qui permet une décharge plus lente. La diode est un composant polarisé et possède un sens de montage [1)].



Le transistor BC547 permet d'assurer la liaison entre la carte Arduino et le circuit alimenté en 12 volts. Un transistor est composé de 3 broches. Lorsque l'on regarde la face plate du transistor BC547, les 3 broches (de gauche à droite) sont[2)] :

- Le collecteur
- La base
- L'émetteur

Le transistor laisse circuler le courant entre le collecteur et l'émetteur seulement lorsque la base est alimenté par la carte Arduino. Quand il n'y a pas de courant envoyé par l'Arduino sur la Base, le courant ne circule plus entre le collecteur et l'émetteur et le ventilateur s'arrête.

Code

Le code à exécuter pour réguler le ventilateur est présenté ci-dessous et est disponible à télécharger ici.

Préambule du code

Chargement de la bibliothèque du capteur de température

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

Définition des ports attribués aux capteurs de températures (ports 2 et 3)

```
#define DHTPIN_1 2     // what pin we're connected to
#define DHTPIN_2 3     // what pin we're connected to
```

Définition du type de capteur de température (DHT11)

```
#define DHTTYPE DHT11   // DHT 11
```

Définition de la valeur entière (int) de la température de consigne qui pilote le ventilateur

```
int maxTemp = 25;
```

Définition du port attribué au signal d'activation/désactivation du ventilateur

```
#define fan 7
```

Commande qui attribue au nom dht_1 et dht_2 des capteur DHT de type DHTTYPE et branché sur le port DHTPIN_1 et DHTPIN_2.

```
DHT dht_1(DHTPIN_1, DHTTYPE);
DHT dht_2(DHTPIN_2, DHTTYPE);
```

Initialisation

```
void setup()
{
```

Définition du mode de fonctionnement du port digital numéro 7 (fan) en signal de sortie. C'est la carte Arduino qui envoie un signal vers le transistor (et non l'inverse)

```
   pinMode(fan, OUTPUT);   // Signal de sortie pour activer ou non le
ventilateur
```

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 des capteurs de température..

```
   dht_1.begin();            // Mise en route du capteur de temperature et
d'humidite
   dht_2.begin();            // Mise en route du capteur de temperature et
d'humidite
 }
```

Boucle infinie

```
 void loop()
 {
```

Pause de 5000ms avec la commande delay.

```
   delay(5000);
```

Lecture de la température et de l'humidité définie dans des variables réelles flottantes (de type float) avec les commandes dht_1.readHumidity() et dht_1.readTemperature().

```
   float h_1 = dht_1.readHumidity();          // Humidité relative
   float t_1 = dht_1.readTemperature();       // Temperature en degres
Celsius
   float h_2 = dht_2.readHumidity();          // Humidité relative
   float t_2 = dht_2.readTemperature();       // Temperature en degres
Celsius
```

Activation et désactivation du ventilateur avec une condition de type if/else. Si la température t_1 est supérieure a maxTemp (25°C), le signal de sortie **fan** (port 7) est alimenté en courant (alimentation du Transistor). Sinon, le signal de sortie n'est pas alimenté

```
   if(t_1 > maxTemp)
   {
       digitalWrite(fan, HIGH);
   }
   else
   {
      digitalWrite(fan, LOW);
   }
```

Affichage dans la console de l'humidité et de la température mesurée

```
   Serial.print(" Humidity #1: ");
   Serial.print(h_1);
   Serial.print(" %\t");
   Serial.print("Temperature #1: ");
   Serial.print(t_1);
   Serial.print(" *C ");
   Serial.print(" Humidity #2: ");
   Serial.print(h_2);
   Serial.print(" %\t");
   Serial.print("Temperature #2: ");
   Serial.print(t_2);
   Serial.println(" *C ");
 }
```

Etape suivante:Gestion du clapet

1)

Branchement de la diode 1N4007

2)

Le transistor BC547

From:
https://wiki.enerlog.fr/ - **Wiki Enerlog**

Permanent link:
https://wiki.enerlog.fr/doku.php?id=espace_public:electronique:arduino_ventilateur

Last update: **2021/03/21 12:10**