

Gestion du clapet

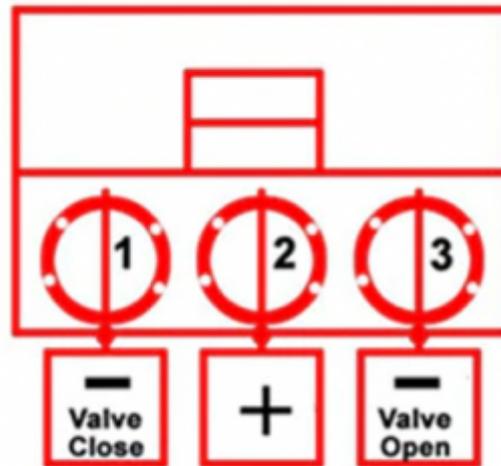
Le chauffage solaire est un dispositif qui permet de chauffer durant les journées d'hiver. Pour son fonctionnement un pilotage électronique est utilisé. En effet, si celui-ci est mal géré il peut refroidir durant les nuits en hiver ce qui serait contre-productif. Pour gérer correctement l'alternance entre les jours et les nuits, un clapet à fermeture automatique est piloté de manière électronique. Cette section présente son fonctionnement.



Le clapet peut être ouvert ou fermé en fonction du circuit alimenté. La connexion du milieu (n°2 sur le diagramme ci-dessous) correspond à l'alimentation 12 volts et est toujours alimenté. Si le port 1 est relié à la masse, le clapet se ferme. Si c'est le port 3 qui est relié à la masse, le clapet s'ouvre. Il n'est pas possible d'alimenter les 3 connectiques en même temps.

Electric air damper wiring diagram.

12VDC

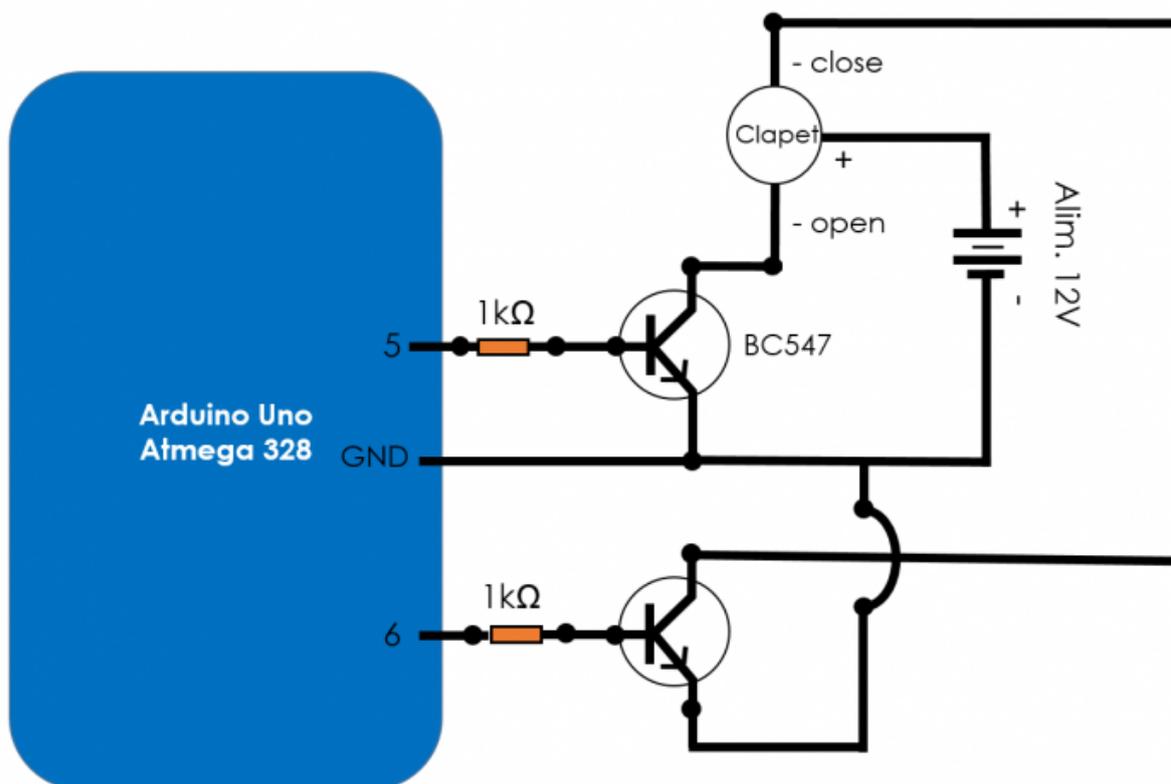


- Connect 2 and 1 wire, damper close.
- Connect 2 and 3 wire, damper Open.
- Can not connect all 3 wires at same time.

Pour obtenir un clapet qui alterne entre l'ouverture et la fermeture en fonction de la température du panneau solaire, deux transistor sont utilisés et pilotés par la carte Arduino. De la même manière que pour [la régulation du ventilateur 12volts](#), un signal sur le transistor concerné est envoyé dès lors que la température du capteur franchit la température de consigne définie.

Le schéma ci-dessous présente le fonctionnement du circuit :

- Une alimentation 12Volts alimente le clapet de manière continue sur le connectique +.
- Un transistor est branché sur le port 5 de la carte arduino. Quand la carte Arduino alimente le port, le transistor laisse circuler le courant jusqu'à la masse de l'alimentation 12 volts et permet d'ouvrir le clapet. Quand le port 5 n'est plus alimenté, le clapet arrête de s'ouvrir
- Comme il existe une butée pour le clapet, il suffit d'alimenter durant une période assez longue pour que le clapet s'ouvre complètement avant de couper le signal de la carte Arduino. Une temporisation de 15 secondes est suffisante.
- De la même manière, un transistor est branché en série avec le clapet du côté du connectique (- close). Lorsque la carte Arduino alimente le port 6, le transistor laisse circuler le courant et permet au clapet de se fermer.



Code

Le code présenté ci-dessous permet de gérer le clapet en fonction des températures mesurées par les capteurs au sein du panneau solaire. Le code [est téléchargeable ici](#) et contient également la [gestion du ventilateur présenté précédemment](#).

Preambule du code

```
#include <DHT.h>

#define DHTPIN_1 2    // what pin we're connected to
#define DHTPIN_2 3    // what pin we're connected to
#define DHTTYPE DHT11 // DHT 11
DHT dht_1(DHTPIN_1, DHTTYPE);
DHT dht_2(DHTPIN_2, DHTTYPE);
```

Définition des ports de la carte Arduino attribués au transistor pour l'ouverture et la fermeture du clapet

```
#define clapet_open 5
#define clapet_closed 6

#define fan 7

int maxTemp = 28;
```

Boucle d'initialisation

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

Définition du mode de fonctionnement des ports digitaux du clapet 5 et 6 en signal de sortie. C'est la carte Arduino qui envoie un signal vers le transistor (et non l'inverse)

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

    Serial.begin(9600);
    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()
{
    // Attente de 5secondes entre chaque mesure.
    delay(5000);
    // Lecture temperature et humidite (temps de lecture>250ms)
    // Capteur #1
    float h_1 = dht_1.readHumidity();
    // Temperature en degres Celsius
    float t_1 = dht_1.readTemperature();
    // Capteur #2
    float h_2 = dht_2.readHumidity();
    // Temperature en degres Celsius
    float t_2 = dht_2.readTemperature();
```

Si la température du capteur 1 dépasse la température de consigne

```
if(t_1 > maxTemp)
{
```

Attente d'une seconde

```
    delay(1000);
```

Affichage Ouverture clapet dans la console

```
Serial.println("Ouverture clapet");
```

Le signal d'ouverture du clapet est alimenté (port 5) par la carte Arduino tandis que le signal de fermeture du clapet n'est pas alimenté (port 6).

```
digitalWrite(clapet_open, HIGH);  
digitalWrite(clapet_closed, LOW);
```

Attente de 15 secondes puis tous les signaux sont coupés (sauf ventilation qui s'active)

```
delay(15000);  
digitalWrite(clapet_open, LOW);  
digitalWrite(clapet_closed, LOW);  
digitalWrite(fan, HIGH);  
}
```

Si la température est inférieure à la température de consigne, fermeture des clapets

```
else  
{
```

Attente d'une seconde

```
delay(1000);
```

Affichage dans la console "Fermeture clapet"

```
Serial.println("Fermeture clapet");
```

Le signal de fermeture du clapet est alimenté (port 6) par la carte Arduino tandis que le signal d'ouverture du clapet n'est pas alimenté (port 5).

```
digitalWrite(clapet_open, LOW);  
digitalWrite(clapet_closed, HIGH);  
delay(15000);
```

Attente de 15 secondes puis tous les signaux sont coupés.

```
digitalWrite(clapet_open, LOW);  
digitalWrite(clapet_closed, LOW);  
digitalWrite(fan, LOW);  
}
```

Affichage des mesures de température et d'humidité dans la console

```
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 ");
}
```

From:

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

Permanent link:

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



Last update: **2021/03/10 18:11**