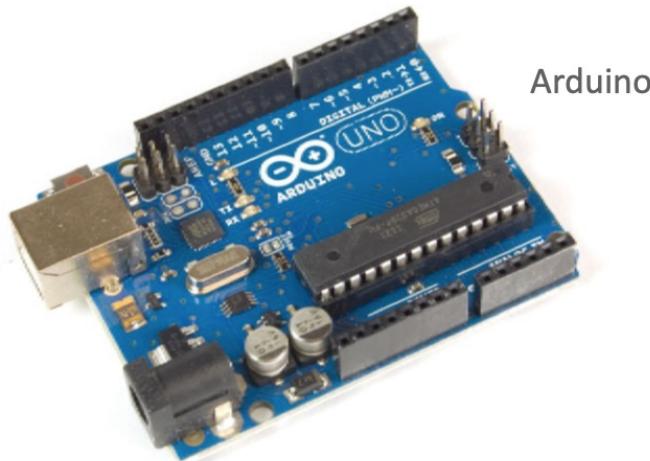


Atelier : les objets connectés

Arduino : Connected Fan

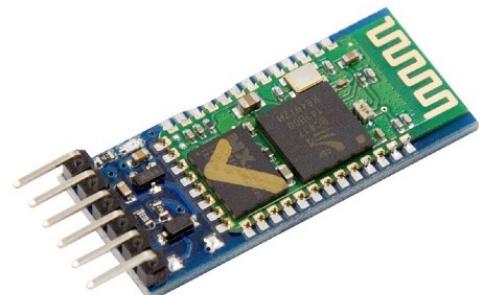


Arduino



Sensor

Temperature/pressure/humidity



Bluetooth emitter

Présentation du projet

Pour cet atelier vous allez créer votre propre objet connecté pour contrôler le déclenchement de votre ventilateur en fonction de la température ambiante. Le déclenchement peut se faire automatiquement depuis l'objet connecté.

Vous pouvez également déclencher le ventilateur et contrôler la température depuis votre smartphone en vous connectant à l'objet en bluetooth.

Qu'est-ce que Arduino ?

L'Arduino a été imaginé par une équipe de développeur mené par Massimo Banzi en 2005. Il s'agit d'une petite **carte électronique programmable** et un logiciel multiplateforme, accessible à tout un chacun dans le but de créer facilement des systèmes électroniques. Le **microcontrôleur** peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la domotique, le pilotage d'un robot, de l'informatique embarquée, etc... Auparavant, programmer un microcontrôleur était extrêmement long, fastidieux et onéreux.

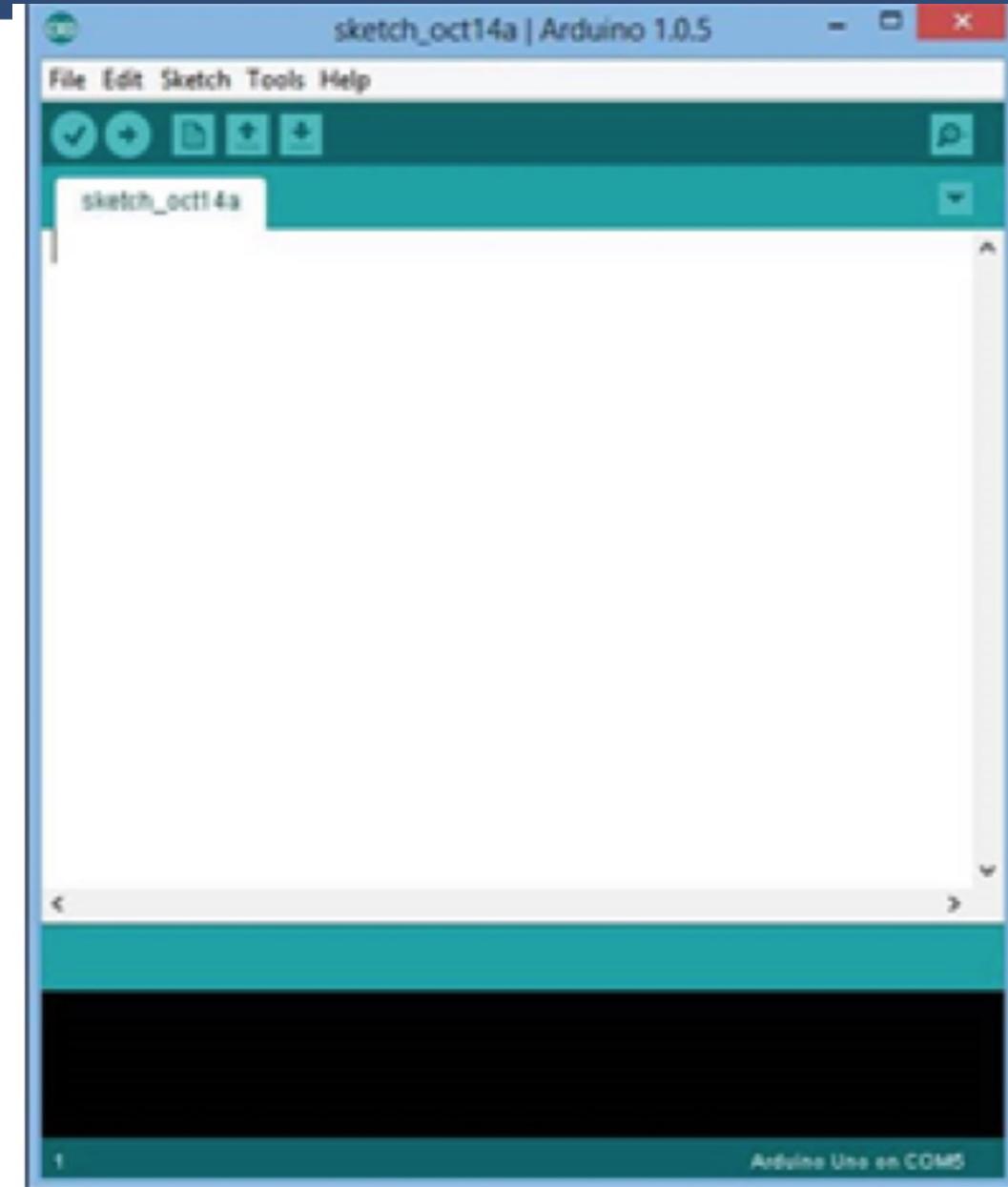
Atelier : les objets connectés

Arduino : Connected Fan

Matériel nécessaire

- Un Arduino
- Un capteur de température
- Un module bluetooth
- Un ventilateur
- Des fils électriques
- Un PC avec un logiciel Arduino IDE permettant d'écrire un programme et de le téléverser à l'Arduino

Si votre éditeur n'est pas déjà lancé sur votre poste, ou si un des composants est manquant, appelez un assistant.



Quelques notions de C

Le programme pour l'arduino s'écrira en C.

Un programme est une liste d'instructions qui est exécutée par un système.

Le C est un langage de programmation inventé au début des années 70. Il est devenu aujourd'hui l'un des langages les plus utilisés en programmation.

De nombreux langages plus récents s'en inspirent comme le C++, le Java ou encore le PHP.

Le programme terminé sera envoyé à l'arduino qui l'exécutera.

Les variables

Une variable est un espace mémoire contenant un nom et une valeur.

On peut l'assimiler à une boîte avec un nom dans laquelle on range certains objets.

On définit une variable de cette façon :

```
int temp = 0;
```

Ici, nous créons la variable nommée *temp* qui est du type *int* (nombre). On initialise sa valeur à 0.

Cette variable sera utilisée dans notre programme pour stocker la température qu'il fait.

N'oubliez pas le point-virgule à la fin !

Les fonctions

Les fonctions vous permettent de regrouper plusieurs instructions de votre programme.
Il y en a deux déjà définies **setup()** et **loop()**

```
void setup()
{
    //écrire les instructions ici
}

void loop()
{
    //écrire les instructions ici
}
```

Les conditions

Les conditions vous permettent d'exécuter une instruction seulement si une condition est validée.

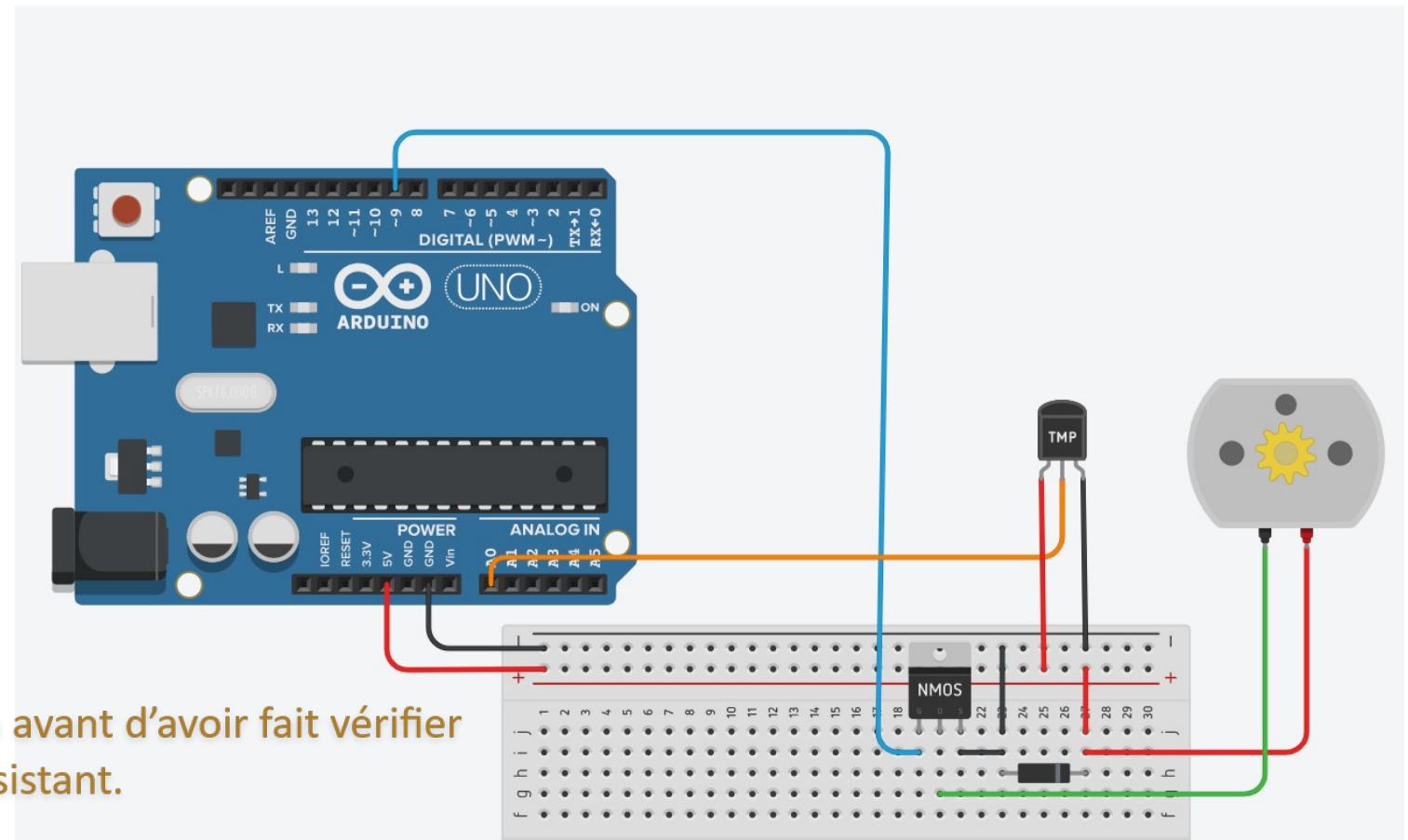
```
if (temp > 1000) {  
    //écrire les instructions ici  
}  
  
else {  
    //écrire les instructions ici  
}
```

Atelier : les objets connectés

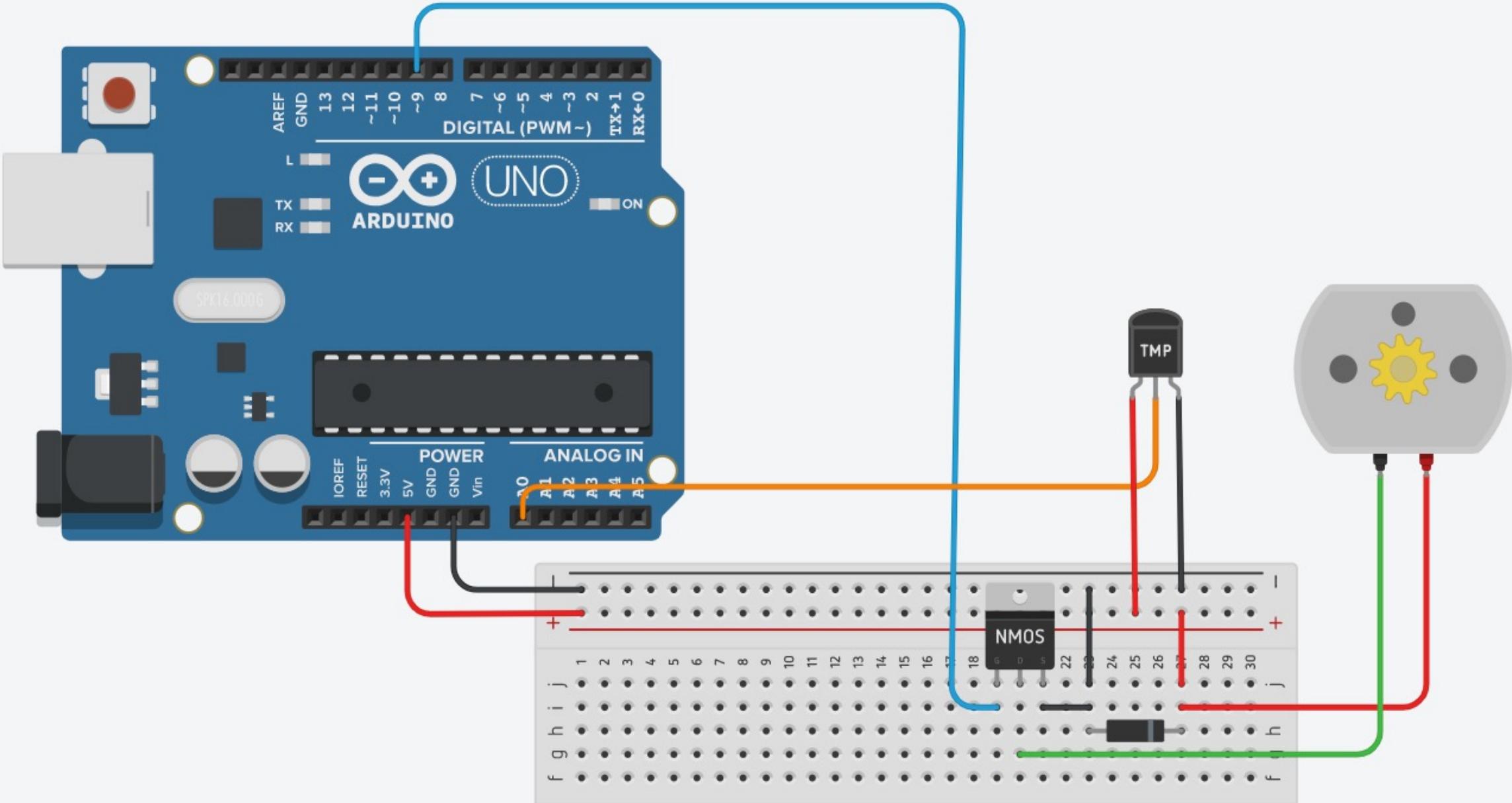
Arduino : Connected Fan

Branchement :

Avant de commencer notre programme, préparons tout d'abord notre Arduino. En vous référant au schémas suivant, effectuez les branchements :

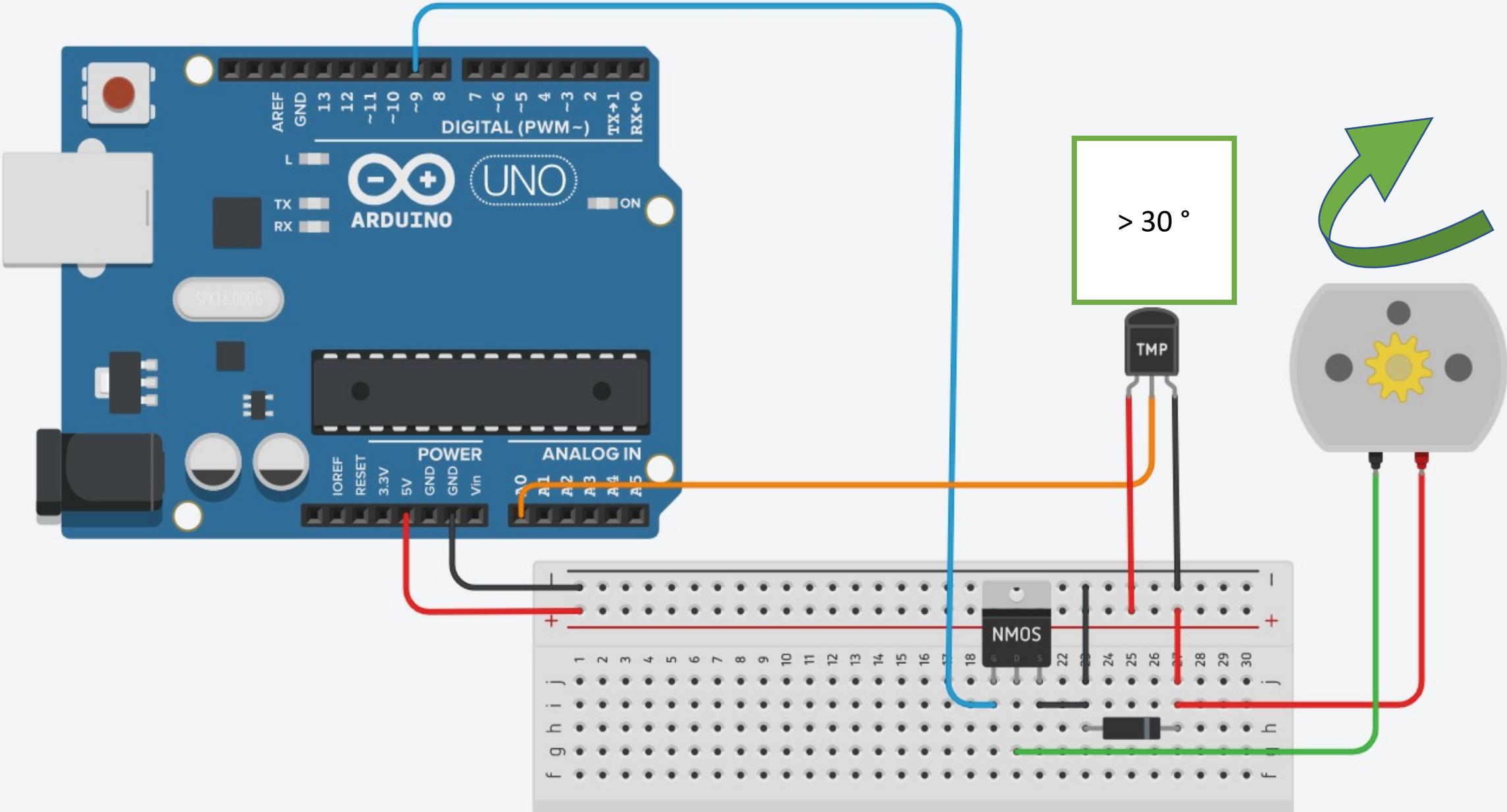


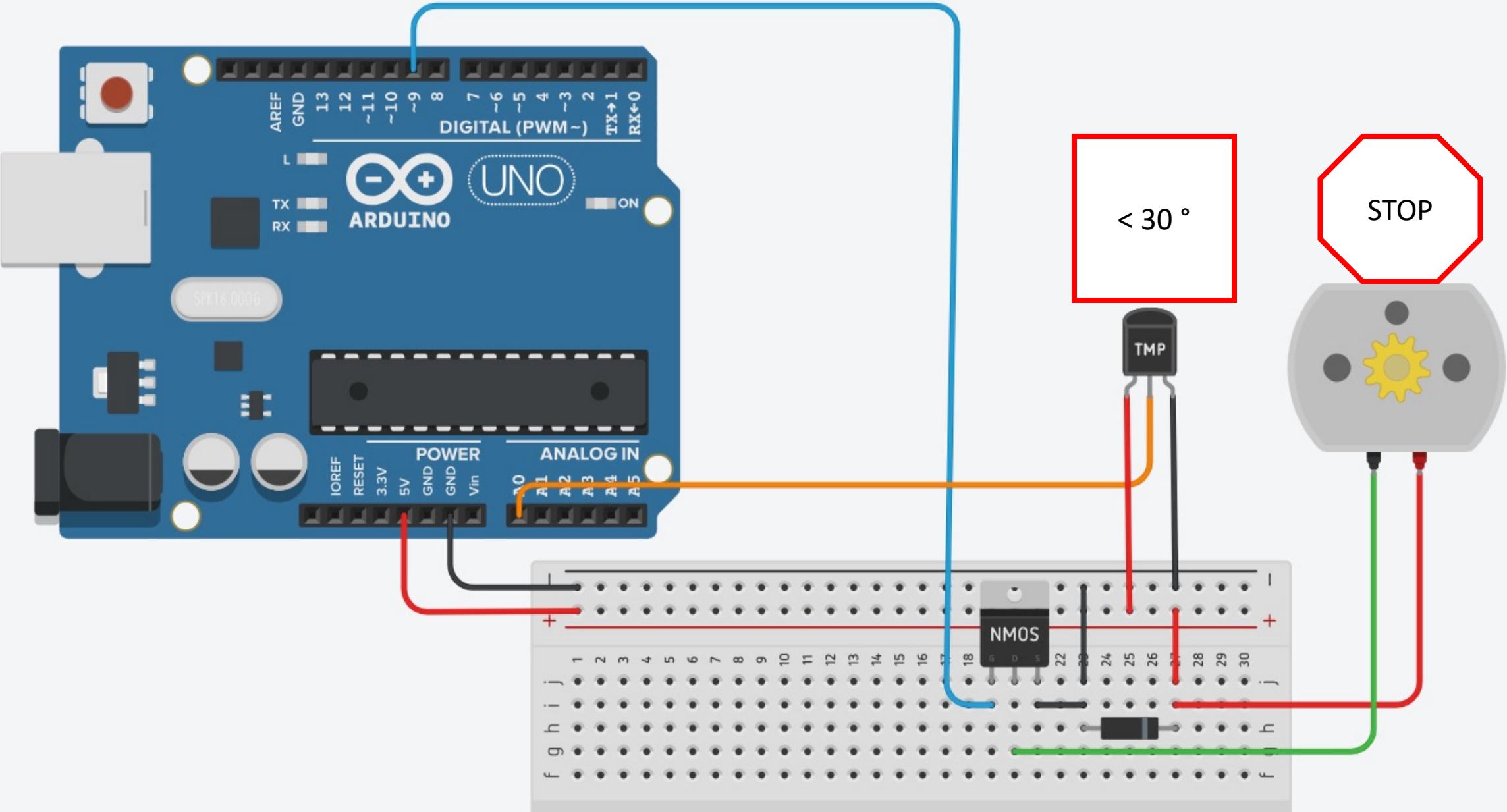
Ne mettez jamais la carte Arduino sous tension avant d'avoir fait vérifier vos branchements par un assistant.



Partie 1

Déclencher le ventilateur en fonction de la température ambiante





Ecrivez votre programme

Tout en haut du programme, faites une nouvelle variable de type int pour représenter notre ventilateur

```
int fan = ?;
```

Quelle valeur allons nous donner à la variable *fan* (notre ventilateur) ?

Petit indice: regardez sur quelle est le numéro de la broche sur laquelle est connecté le signal du ventilateur

Initialisez votre programme

La fonction **setup()** est lancée à chaque démarrage de l'arduino. C'est ici que nous allons initialiser notre programme.
Ecrivez dans la fonction **setup()** l'instruction qui permet d'initialiser l'affichage:

Vous pouvez ensuite ajouter une instruction pour que notre Arduino nous affiche quelque chose avec la fonction *display()*

Par exemple *display("Initialisation terminée");*

```
init_display();
```

Atelier : les objets connectés

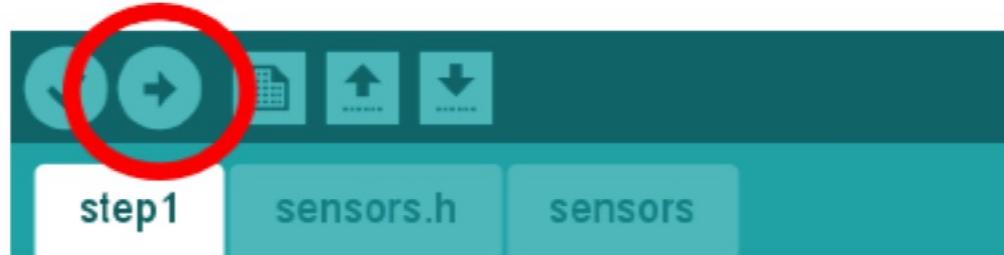
Arduino : Connected Fan

Vous êtes maintenant prêt à envoyer votre programme à l'arduino.

Sauvegarder vos modifications, et téléverser.

Un message vous confirmant le bon fonctionnement devrait s'afficher dans le Moniteur Série (Ctrl+Shift+M)

Appelez un assistant si ce n'est pas le cas. Vous êtes maintenant prêt à récupérer les données du capteur.



Récupérez les données

Dans la fonction **loop()** vous pouvez maintenant récupérer la température ambiante en utilisant la fonction **get_temp()**
Et stocker le résultat de cette fonction dans une variable que nous avons défini plus tôt.

Il faut donner un paramètre à la fonction **get_temp()**

Ce paramètre représente la broche sur laquelle est connectée le capteur de température

```
? = get_temp(?);
```

Rappel:

Affichez les données récupérées

On peut afficher les données ou du texte grâce à la fonction `display`.

```
display ("Bonjour")
```

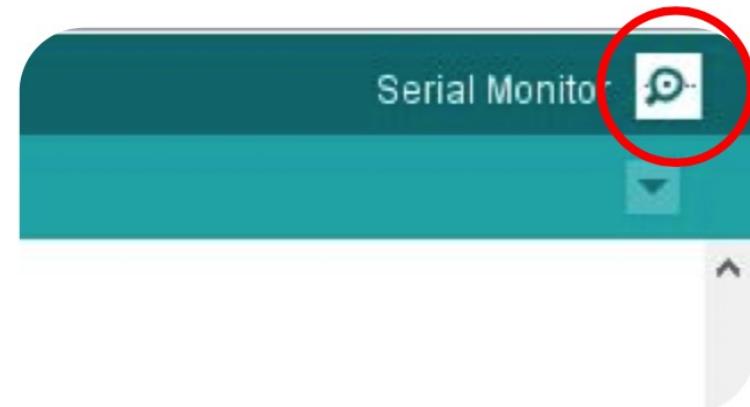
Toujours dans la fonction `loop()`, essayez d'afficher la valeur de `temp`

Atelier : les objets connectés

Arduino : Connected Fan

Envoyer votre programme à l'arduino.

Cliquer sur la petite loupe pour ouvrir l'écran d'affichage, qui nous permettra de visualiser les données qui sont récupérées par le capteur.



Créez une fonction pour déclencher le ventilateur

Dans cette fonction vous allez indiquer à l'arduino que **SI** la température dépasse 30°C de déclencher le ventilateur.

```
...
void setup()
{
    ...
}

void trigger_fan()
{
    //écrivez votre code ici
}
...
```

Voici les instructions pour allumer ou éteindre le ventilateur

```
turn(ON, fan);
```

```
turn(OFF, fan);
```

Atelier : les objets connectés

Arduino : Connected Fan

Votre fonction complète :

```
void trigger_fan()
{
    if (temp > 30.0) {
        turn(ON, fan);
    }
    else {
        turn(OFF, fan);
    }
}
```

Enfin, appeler la fonction **trigger_fan()** dans la fonction **loop()**

Téléverser le programme dans l'arduino

Vous pouvez maintenant vérifier que votre objet connecté fonctionne. Approchez une source de chaleur près du capteur et le ventilateur devrait se déclencher si la température dépasse 30°C

Partie 2

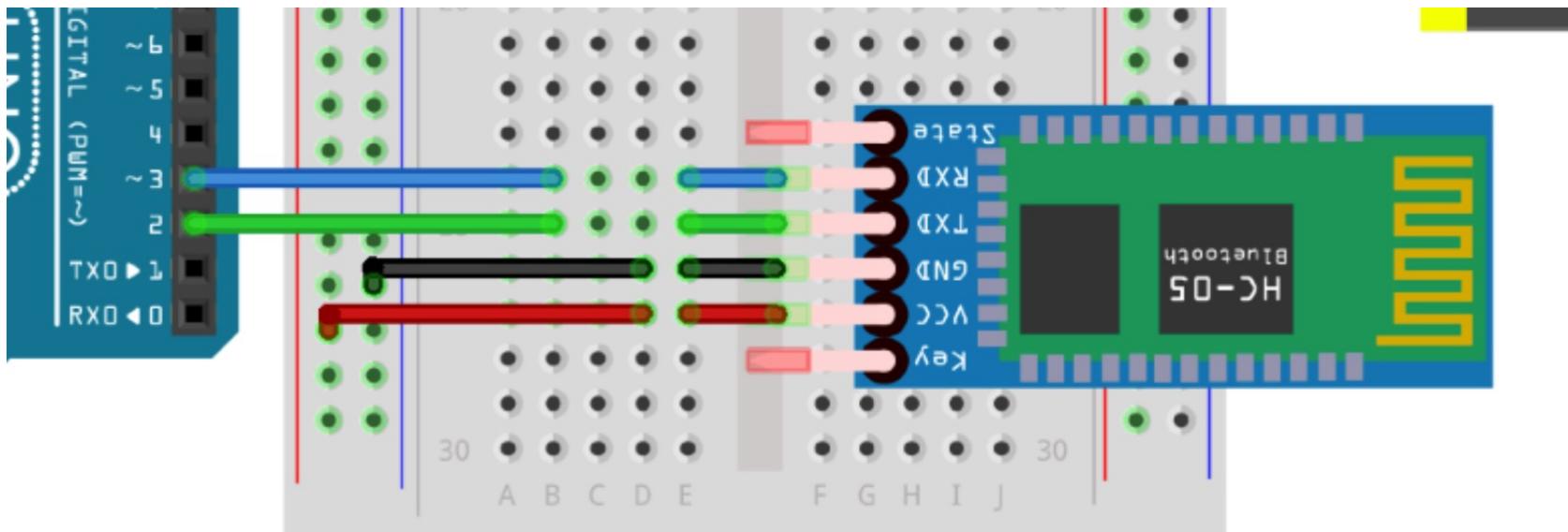
Récupérer les données du capteurs sur son smartphone en bluetooth

Atelier : les objets connectés

Arduino : Connected Fan

Tout en haut du programme, en dessous des variables, écrivez la ligne suivante

```
BLUETOOTH my_bt_module(2, 3);
```



Initialisez le module bluetooth

Comme pour votre capteur et l'affichage, vous devez initialiser le module bluetooth
Ecrivez dans la fonction **setup()** cette instruction :

```
init_bluetooth();
```

Créez une fonction pour envoyer les données

Dans cette fonction vous allez envoyer la température ambiante en °C et en °F grâce à l'émetteur Bluetooth.

Et oui, vous devez également écrire une fonction qui fait la conversion entre °C et °F.

$$tempF = \frac{tempC \times 9}{5} + 32$$

```
int C_to_F(int tempC)
{
    return ?;
}
```

```
...
void setup()
{
    ...
}

...
void send_bt()
{
    //écrivez votre code ici
}
...
```

Envoyer les données

Voici l'instruction pour envoyer une donnée en bluetooth

```
bluetooth_send("Temperature " + String(temp) + " °C");
```

Faites de même pour envoyer la température en °F

Enfin appeler la fonction depuis votre fonction **loop()**

Recevoir les données

Si vous avez un smartphone Android, vous pouvez vous connecter à votre module bluetooth.

Dans la liste des appareils bluetooth, associer l'appareil FocusTech-01 ou FocusTech-02. Le numéro de l'émetteur est inscrit sur celui ci.

Le code PIN est : 4242

Télécharger l'application

Sur votre Play Store, télécharger l'application nommée **Arduino bluetooth controller**
Jumeler votre appareil et lancer le terminal.

Partie 3

Envoyez une commande depuis le smartphone pour allumer le ventilateur

Atelier : les objets connectés

Arduino : Connected Fan

Tout en haut du programme, déclarez une variable booléen

```
boolean my_bt = true;
```

Créez une fonction pour recevoir une commande

Dans cette fonction vous allez recevoir une commande depuis le smartphone pour allumer ou éteindre le ventilateur.

Dans cette fonction vous allez indiquerz **SI** le bluetooth reçoit "**On**" le ventilateur s'allume **SINON SI** le bluetooth reçoit "**Off**" le ventilateur s'éteint.

```
void receiv_bt()
{
}
```

Atelier : les objets connectés

Arduino : Connected Fan

Dans cette fonction vous allez tout d'abord créer une boucle qui permettra au ventilateur lorsqu'elle est allumé de ne pas s'éteindre.

```
void receiv_bt()
{
    while () {
        //instructions
    }
}
```

- 1 - Vous allez dire à votre boucle tant que la variable **my_bt** est égale à **true**
- 2 - Dans la boucle vous allez indiquerz **SI** la commande reçue est **On** le ventilateur s'allume.

```
Bluetooth_recv == "On"
```

- 3 - Après la première condition on indique **SINON SI** la commande reçue est **Off** on change l'état de la variable **my_bt** pour la mettre en **false**.
- 4 - Enfin vous sortez en dehors de la boucle, vous indiquez **SI** la variable **my_bt** est égale à **false** le ventilateur s'éteint et vous remettez la variable **my_bt** à **true** pour que vous puissiez réallumer le ventilateur.

Votre fonction complète :

```
void receiv_bt()
{
    while (my_bt == true) {
        if (bluetooth_recv() == "On") {
            turn(ON, fan);
        } else if (Bluetooth_recv() == "Off") {
            my_bt == false;
        }
        if (my_bt == false) {
            turn(OFF, fan);
            my_bt = true;
        }
    }
}
```

Enfin, appeler la fonction **receiv_bt()** dans la fonction **loop()**
Téléverser le programme dans l'arduino

Vous pouvez maintenant vérifier que votre objet connecté fonctionne. Sur le smartphone écrivez "On" sur la barre de commande.

FIN

**Vous avez maintenant fait votre tout
premier objet connecté.**

Félicitations !