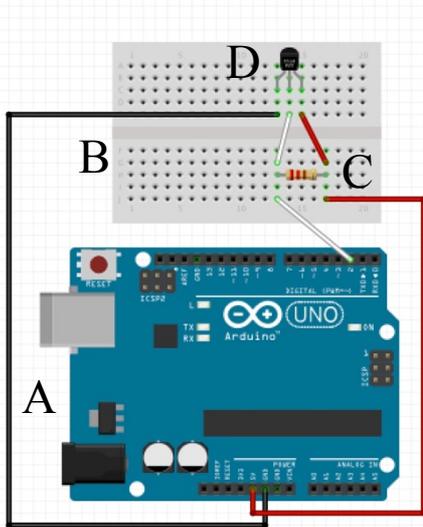


Création d'une sonde thermomètre à l'aide d'une carte à microcontrôleur

Pour débiter, j'ai besoin de :

A. Hardware : Matériel nécessaire et réalisation du circuit :



- Une carte Arduino et son câble de liaison USB (A)
- Un ordinateur avec l'IDE* Arduino installé
- Une *breadboard* (B)
- Une résistance de 4,7kΩ (C)
- Un capteur de température DS18B20 DALLAS Semiconductor (D)
- Des conducteurs avec connecteurs Dupont

La carte Arduino porte un microcontrôleur capable de communiquer avec un ordinateur via le port USB. Il utilise également des broches digitales et analogiques pour se connecter à des modules électroniques.

La *breadboard* permet de réaliser facilement des circuits simples : on peut épingler dans les trous des broches ou les pattes de composants électroniques. Les trous de chaque colonne sont déjà reliés électriquement.

Figure 1 : Réaliser le montage ci-dessus en connectant la borne (-) du capteur à la broche GND (Ground) de l'Arduino. Les données circuleront par un port OneWire (1 fil) connecté à la broche digitale 2. Le capteur est alimenté en 5V mais il ne faut pas oublier de rajouter une résistance de tirage de 4,7kΩ associée au fil OneWire.

B. Software : paramétrer l'IDE Arduino et écrire le code :

1. IDE* et connexion au port série.

Après avoir vérifié que la carte est bien branchée à l'ordinateur (LED allumée), on commence par paramétrer le type de carte Arduino utilisée (ici Arduino UNO) et le port série sur lequel est branché la carte (elle est reconnue mais vous devez tout de même sélectionner le bon port). Pour cela, on utilise le menu outil de l'IDE (figure 5).

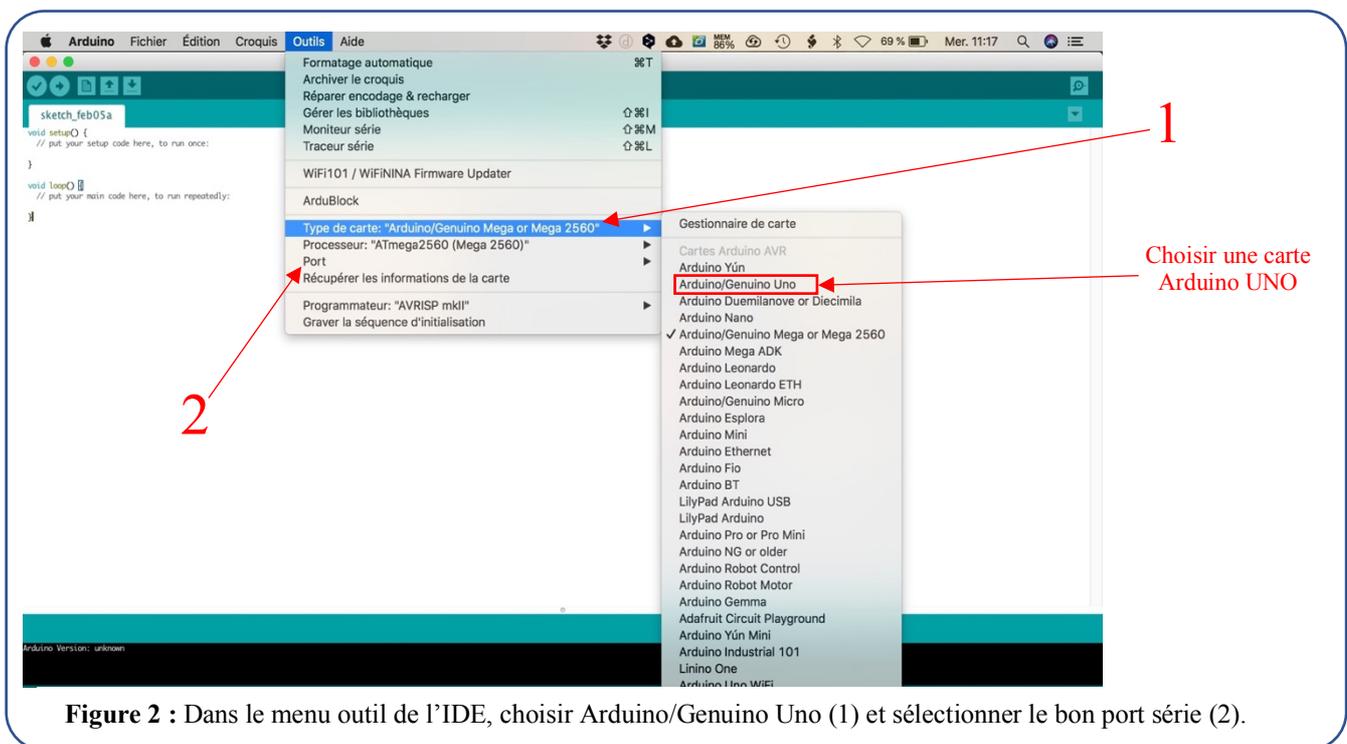
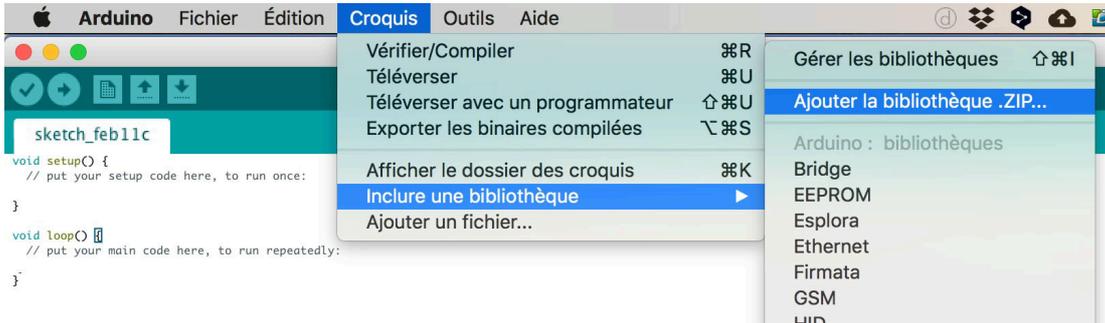


Figure 2 : Dans le menu outil de l'IDE, choisir Arduino/Genuino Uno (1) et sélectionner le bon port série (2).

*IDE : *Integrated Development Environment*, logiciel gratuit fourni par Arduino et assurant l'interface entre la carte et l'ordinateur.

2. Importation des bibliothèques

Pour simplifier l'écriture des programmes Arduino, il existe des bibliothèques pour la plupart des capteurs que nous utiliserons. Dans cet exemple, nous aurons besoin des bibliothèques **Dallas.zip** qui gère le capteur



lui-même et la bibliothèque **OneWire.zip** qui gère les communications sur un fil avec l'Arduino (Figure 3).

Figure 3 : Ajouter des bibliothèques à l'IDE.

3. Programmation de la carte.

Vous pouvez, soit directement ouvrir le fichier « thermometer.ino » présent sur la clé USB ou l'écrire directement dans l'IDE (Figure 6).

Un code Arduino comprend au minimum 3 parties :



(1) Un entête pour nommer les broches, déclarer les variables et les bibliothèques qui seront utilisées

(2) Une fonction setup qui ne sera réalisée qu'une seule fois au démarrage de la carte

(3) Une fonction loop qui sera répétée indéfiniment par la carte

Enfin, on téléverse le code dans la carte Arduino !

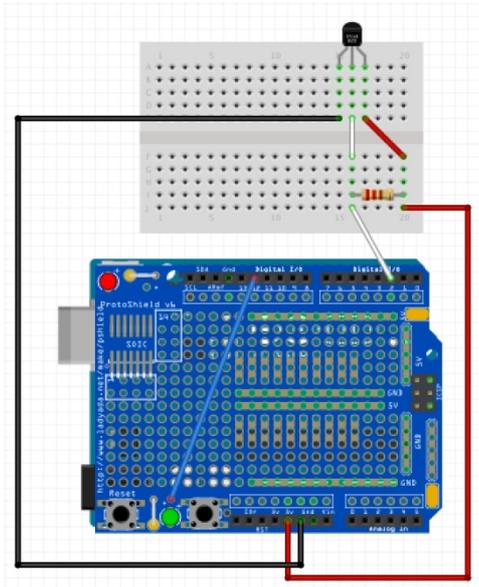
Figure 4 : Le code Arduino est écrit dans l'IDE puis envoyé dans le microcontrôleur de la carte.

On peut visualiser les données envoyées par la carte en cliquant sur le moniteur série en haut à droite de la fenêtre du logiciel.



D. Pour aller plus loin

1 : Créer une alarme en rajoutant une condition if... else



Nous allons créer une condition qui allumera une LED si la température mesurée dépasse 25°C ! Sur le montage précédent, on se contente de connecter la LED intégrée au shield (fil bleu) à une broche digitale de l'Arduino (la broche 12 par exemple).

Figure 5 : Connecter la LED du shield (fil bleu) à la broche digitale 12 de l'Arduino.

Modifions un peu le code en ajoutant :

(1) Entête

```
#define LED 12 // on appelle LED la broche digitale 12
```

(2) Setup

```
pinMode(LED, OUTPUT) ; //la broche LED est déclarée comme une sortie de courant
```

(3) Loop

Ecrire avant le délai de 1000 ms :

```
if(Temp>25) {digitalWrite(LED, HIGH);} //si Temp>25, la broche LED delivera 5 volts  
else {digitalWrite(LED, LOW);} // sinon, la broche deliver 0 volt.
```

2 : Exploiter les données sur un ordinateur

thermometer_Data

```
#include <OneWire.h>  
#include <DallasTemperature.h>  
  
#define ONE_WIRE_BUS 2  
  
float Temp;  
  
OneWire oneWire(ONE_WIRE_BUS);  
DallasTemperature sensors(&oneWire);  
  
void setup(void)  
{  
  Serial.begin(9600);  
  sensors.begin(); //on démarre le capteur  
}  
  
void loop(void)  
{  
  sensors.requestTemperatures();  
  Temp = sensors.getTempCByIndex(0);  
  Serial.println(Temp);  
  delay(1000);  
}
```

L'objectif est de pouvoir récupérer les données de température pour les exploiter sur un logiciel comme Libre-office CALC ou Microsoft Excel. Pour cela, il faut suffisamment simplifier le code pour n'afficher **que** la température. On copie le moniteur série dans le tableur.

