

Atelier collectif : Réaliser un thermomètre à affichage numérique avec Arduino et l'impression 3D

⌚ 3 h

Objectif : A l'issue de cet atelier, l'apprenant est capable de réaliser un thermomètre avec affichage numérique et son boîtier. Il sait réaliser un montage d'un circuit électronique avec le microcontrôleur Arduino et est apte à le programmer grâce à l'IDE Arduino . Il saura enfin modéliser un boîtier en 3D avec Tinkercad pour assembler le tout.

Objectifs pédagogiques :

- Reproduire et comprendre un circuit électronique en utilisant des composants du starter kit Arduino (capteur de température, écran LCD).
- Reproduire, comprendre et modifier un programme sur l'IDE Arduino pour recueillir et afficher la température de la pièce dans le moniteur série de l'IDE Arduino puis sur l'écran LCD.
- Modéliser en 3D avec Tinkercad un boîtier dans lequel le montage électronique pourra être rangé.

Prérequis : Avoir suivi un atelier d'initiation à l'impression 3D, à la modélisation avec Tinkercad et un atelier d'initiation au microcontrôleur Arduino

Matériel nécessaire : Un ordinateur connecté à internet pour chaque participant et pour l'animateur équipé avec souris et clavier, 1 starter kit Arduino pour chaque participant, une imprimante 3D, 1 boîtier servant d'exemple d'assemblage, vidéoprojecteur, stylos et feuilles.

Besoin en logiciels : Arduino IDE, slicer pour l'impression 3D.

Public : Étudiants ou jeunes (18-25 ans) qui souhaitent s'approprier les outils d'impression 3D et microcontrôleur avant de réaliser un projet plus ambitieux

Taux d'encadrement : Un animateur pour 4 à 6 participants ou un animateur et un co-animateur pour 8 à 12 participants selon le matériel et la capacité de la salle.







Supports pédagogiques : [Fiche technique](#) de l'atelier à distribuer aux participants, [tutoriel](#) et [trame d'animation](#) à destination de l'animateur.

Modalités d'évaluation pendant et en aval de la séance : Exercices lors de la séance et questionnaire de satisfaction après l'atelier.

Modalités pédagogiques : Exercice-entraînement, Projet, Questionnement de groupe.

Contenu et déroulé de l'atelier

Contenu et déroulé de l'atelier par séquence : L'atelier se déroule en 6 parties :

- | | |
|---|--|
| 1. Lancement de l'atelier |  5' |
| 2. Utilisation du capteur de température et recueil de la température sur le moniteur série de l'IDE Arduino |  30' |
| 3. Montage de l'écran LCD et programmation pour afficher la température mesurée |  30' |
| 4. Modélisation du boîtier en 3D |  40' |
| 5. Assemblage et test |  10' |
| 6. Clôture de l'atelier |  5' |

Atelier collectif : Réaliser un thermomètre à affichage numérique avec Arduino et l'impression 3D

 3 h

TRAME D'ANIMATION

A destination de l'animateur

Prérequis : Avoir suivi un atelier d'initiation à l'impression 3D, à la modélisation avec Tinkercad et un atelier d'initiation au microcontrôleur Arduino

Avant de commencer : Imprimez votre trame d'animation, le [tutoriel](#) et la fiche support. Consultez si besoin ce tutoriel sur ce projet qui est très détaillé : <https://microcontrollerslab.com/tmp36-temperature-sensor-arduino-tutorial/> ou ce site qui explique les bases de l'électronique et d'Arduino : <https://fr.flossmanuals.net/arduino/les-bases-de-lelectronique/>

Allumez votre ordinateur et ouvrez les logiciels dont vous aurez besoin (IDE Arduino, navigateur). Vérifiez et préparez le matériel dans la salle.

Objectif : A l'issue de cet atelier, l'apprenant est capable de réaliser un thermomètre avec affichage numérique et son boîtier. Il sait réaliser un montage d'un circuit électronique avec le microcontrôleur Arduino et est apte à le programmer grâce à l'IDE Arduino . Il saura enfin modéliser un boîtier en 3D avec Tinkercad pour assembler le tout.

1. Lancement de l'atelier

🕒 5'

Présentez-vous aux participants et demandez-leur de faire de même. Recueillez leur attentes vis à vis de cet atelier.

Rappelez leur que cet atelier dure 3 heures.

Leur présentez les objectifs de cet atelier.

2. Utilisation du capteur de température et recueil de la température sur le moniteur série de l'IDE Arduino

🕒 30'

Distribuez aux participants la fiche support avec le circuit de montage et donnez leur un starter kit Arduino.

Dites-leur que nous allons utiliser un capteur de température TMP36 qui génère un courant en sortie qui est linéairement proportionnel à la température. Ce voltage va ensuite être converti en température en se basant sur un facteur de 10 mV/°C. Chaque fois que la température augmente d'1 degré, le courant augment de 10 mV. Ce capteur est déjà calibré et sa précision de mesure est de $\pm 2^{\circ}\text{C}$.

Précisez-leur que ce capteur a un sens de montage comme illustré sur la fiche support et demandez-leur de reproduire le montage.

Posez-leur régulièrement des questions comme : Comment le courant circule t'il ? Qu'est-ce que le ground ?

Laissez-les avancer tout en passant vers eux et les questionnant.

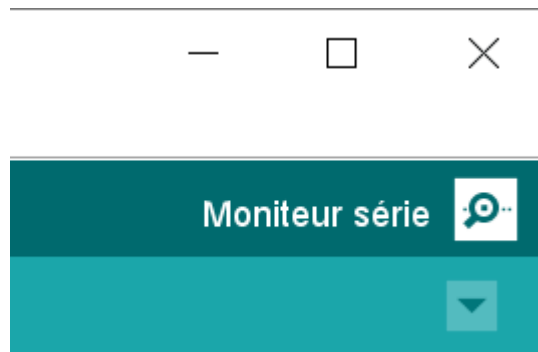
Lorsque le montage est réalisé, leur demandez d'utiliser le logiciel Arduino sur l'ordinateur et de copier/coller le code fourni.

Expliquez la structure du code void setup() est une fonction pour initialiser puis void loop() est une fonction qui va se répéter en boucle.

`/* */` permet de commenter le code sur plusieurs lignes et `//` sur une ligne. Commenter permet d'expliquer le code. Code == langage on doit parler un certain langage pour que l'ordinateur parle et contrôle le microcontrôleur Arduino. Expliquer la suite du code en utilisant les explications en commentaires.

Puis connectez Arduino au port USB de l'ordinateur et choisissez le bon port et téléversez le programme sur la carte Arduino.

Cliquez en haut à droite sur Moniteur série pour ouvrir le moniteur et voir les valeurs actualisées toutes les secondes. Vérifiez que vous avez la bonne connection au moniteur (ex : Serial.begin(9600) dans le code et sur le moniteur sélectionnez 9600 aussi)



Questionnez-les sur les différentes fonctions et blocs et vérifier qu'ils visualisent bien les valeurs de température sur le moniteur. Demandez-leur de changer la fréquence de l'actualisation et le texte qui s'affiche sur le moniteur.

3. Montage de l'écran LCD et programmation pour afficher la température mesurée

🕒 30'

Expliquez aux participants qu'à présent ils vont connecter un écran LCD pour afficher les valeurs mesurées par le capteur de température.

Dites-leur de reproduire le montage présenté dans la fiche support en expliquant les connexions et en les questionnant (qu'est-ce qu'une résistance? Utilisez le code couleur fourni pour identifier la bonne).

- * LCD VSS pin to ground
- * LCD VDD pin to 5V
- * LCD VO pin to ground
- * LCD RS pin to digital pin 12
- * LCD R/W pin to ground
- * LCD Enable E pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2

- * LCD LED+ pin to 5V
- * LCD LED- pin to ground via a 220-ohm resistance

Demandez-leur d'ouvrir l'IDE Arduino et de copier le code fourni non commenté et de téléverser le code sur l'arduino.

Utilisez le code commenté pour les interpellier sur les parties du code différente du code précédent. Quels sont les éléments différents ?

Pourquoi ? L'écran LCD a 16 colonnes et 2 lignes pour écrire et
`lcd.setCursor(0,0);` // set the cursor to column 0, line 0

Déplacez-vous et allez vérifier leur montage et leur code. Demandez-leur de modifier ce qui est affiché et la fréquence d'actualisation.

Maintenant dites-leur que nous allons emballer ce capteur dans un boîtier que nous allons modéliser en utilisant Tinkercad. Vérifiez avec eux si le montage électronique pourrait être optimisé pour que cela tienne le moins de place possible.

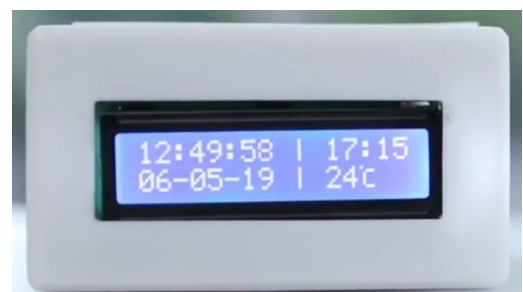
4. Modélisation du boîtier en 3D

🕒 40'

Demandez-leur de se connecter avec leur compte Tinkercad ou d'en créer un et de modéliser un boîtier pour ranger les composants du thermomètre et afficher la température.

Pour créer le boîtier, le perçage est beaucoup utilisé et il existe les formes des composants électroniques (arduino U3, écran LCD 16x2) pour ajuster au mieux. Montrez-leur cet [exemple](#) ou [celui-ci](#) et dites-leur d'en chercher.

Digital Clock & Thermometer



Laissez les participants pratiquer et passez régulièrement et montrez vous disponible pour les aider.

5. Assemblage et test

 10'

Utilisez un boîtier précédemment imprimé et le donnez aux participants pour qu'ils assemblent un thermomètre. Demandez-leur comment ce montage ou le design du boîtier pourrait être amélioré ?

6. Clôture de l'atelier

 5'

Présentez la synthèse de cet atelier.

Demandez aux participants ce qu'ils ont retenu de l'atelier, ce qu'ils ont le plus aimé, ce qu'il leur a déplu ou qu'ils ont trouvé difficile.

Recueillez leur envies de projets futurs, leur montrer ce site qui recense des chouettes projets :

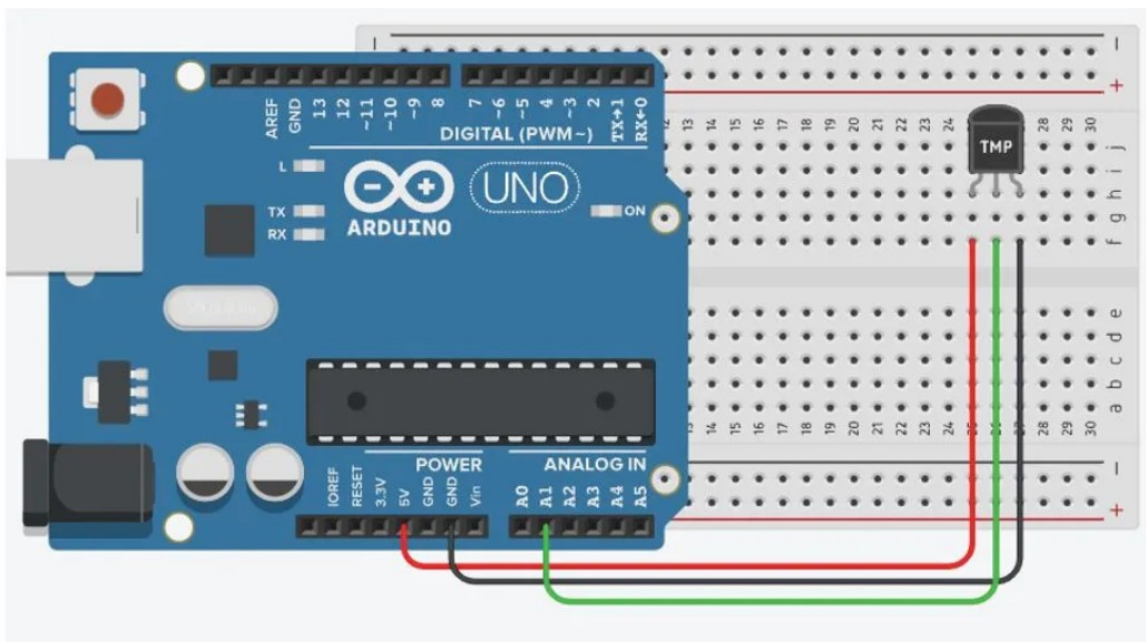
<https://create.arduino.cc/projecthub>

Réaliser un thermomètre numérique avec Arduino et l'impression 3D

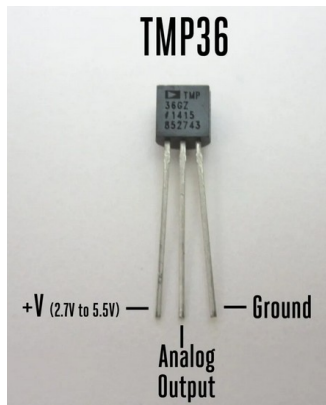
FICHE SUPPORT

2.Utilisation du capteur de température et recueil de la température sur le moniteur série de l'IDE Arduino

Circuit électronique et schéma de câblage



Capteur de température TMP36



Code à coller dans l'IDE Arduino

```
/*
Project Digital Thermometer
Parts required are in the Arduino Starter Kit: one TMP36 temperature sensor and wires
*/
```

```
// named constant for the pin the sensor is connected to
const int sensorPin = A0;
```

```
void setup() {
  // open a serial connection to display values
  Serial.begin(9600);
}
```

```
void loop() {
  // read the value on AnalogIn pin 0 and store it in a variable
  int sensorVal = analogRead(sensorPin);
```

```
// send the 10-bit sensor value out the serial port
Serial.print("sensor Value: ");
Serial.print(sensorVal);
```

```
// convert the ADC reading to voltage
float voltage = (sensorVal / 1024.0) * 5.0;
```

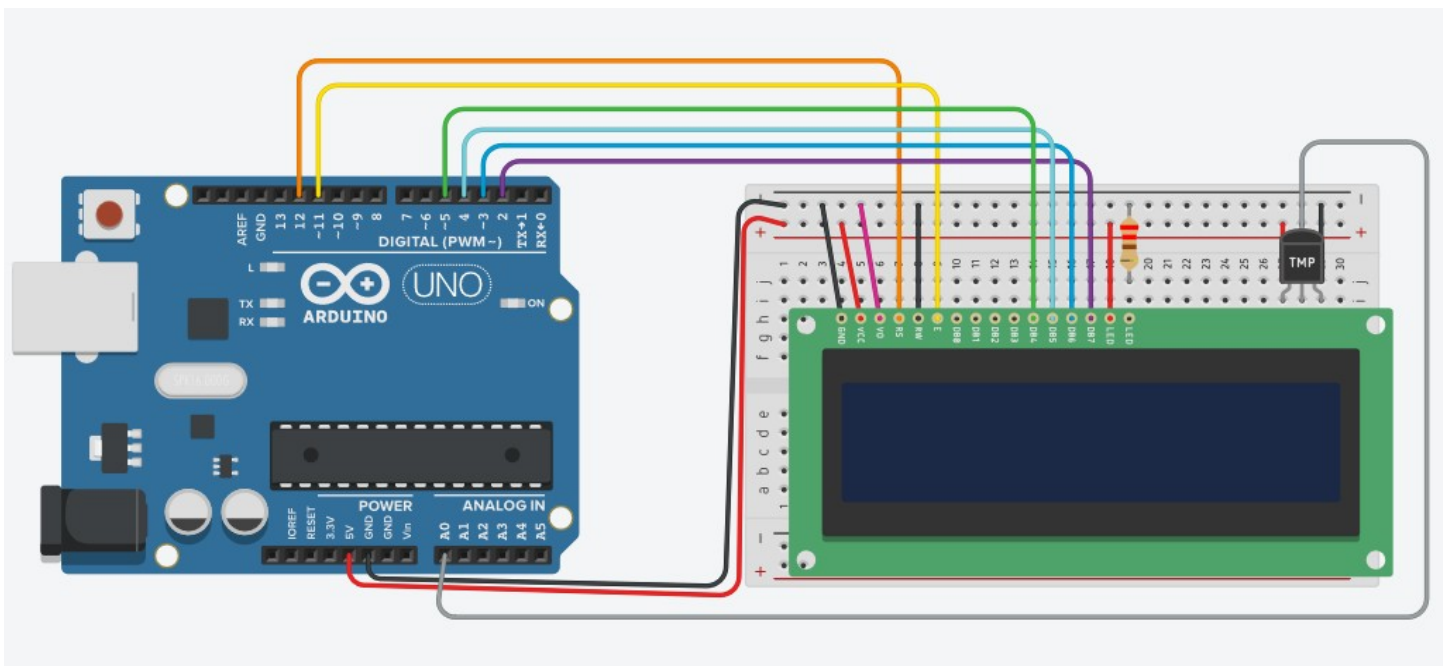
```
// Send the voltage level out the Serial port
Serial.print(", Volts: ");
Serial.print(voltage);
```

```
// convert the voltage to temperature in degrees C
// the sensor changes 10 mV per degree
// the datasheet says there's a 500 mV offset
// ((voltage - 500 mV) times 100)
Serial.print(", degrés Celcius: ");
float temperature = (voltage - .5) * 100;
Serial.println(temperature);
```

```
// printing delay on the serial monitor  
delay(1000);  
}
```

3.Montage de l'écran LCD et programmation pour afficher la température mesurée

Circuit électronique et schéma de câblage



Code à coller dans l'IDE Arduino

```
#include<LiquidCrystal.h>  
  
const int sensor_pin = A0;  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  
  
void setup()  
{  
  lcd.begin(16, 2);  
  lcd.print("Hello!");  
}  
  
void loop()  
{  
  int sensor_data = analogRead(sensor_pin);  
  float voltage = sensor_data * (5.0 / 1024.0);  
  float temperature_Celsius = (voltage - 0.5) * 100;  
  
  delay(1000);  
}
```

```
lcd.setCursor(0,0);  
lcd.print("Temperature");  
lcd.setCursor(0,1);  
lcd.print(temperature_Celsius);  
lcd.setCursor(5,1);  
lcd.write(223);  
lcd.print("C");  
}
```

