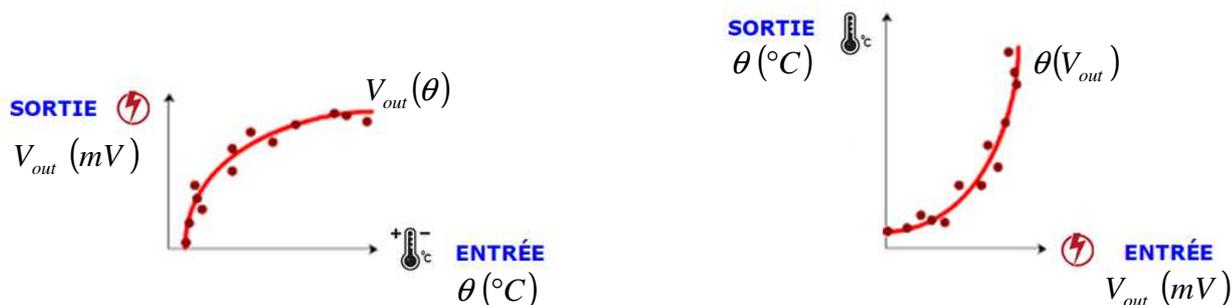


# Capteurs analogiques

## Notions avancées

Un capteur – de température pour l'exemple – délivre une **tension électrique**  $V_{out}$  (mV) qui varie à l'image des variations de la **grandeur physique mesurée**  $\theta$  (°C).

L'étalonnage du capteur donne deux relations :  $V_{out}(\theta)$  et sa réciproque,  $\theta(V_{out})$ .

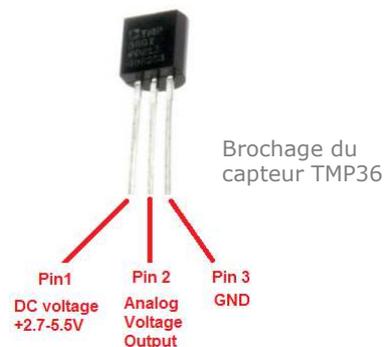
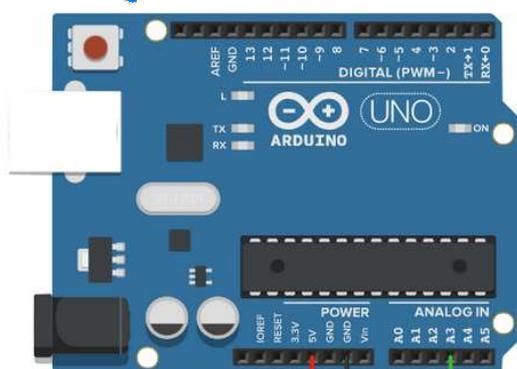


En Sciences de l'Ingénieur, la relation mathématique « Entrée / Sortie » a très souvent vocation à être manipulé dans un **programme informatique** (dans le cadre de la fonction « TRAITER »).

Dit autrement, pour un capteur de température, on veut manipuler des °C dans le programme. Mais, on va le voir dans un exemple, ce n'est pas immédiat.

### Capteur étudié : capteur de température TMP36 (TinkerCAD)

Carte ARDUINO



Capteur de température TMP36. Il est branché sur l'entrée analogique A3 de la carte.

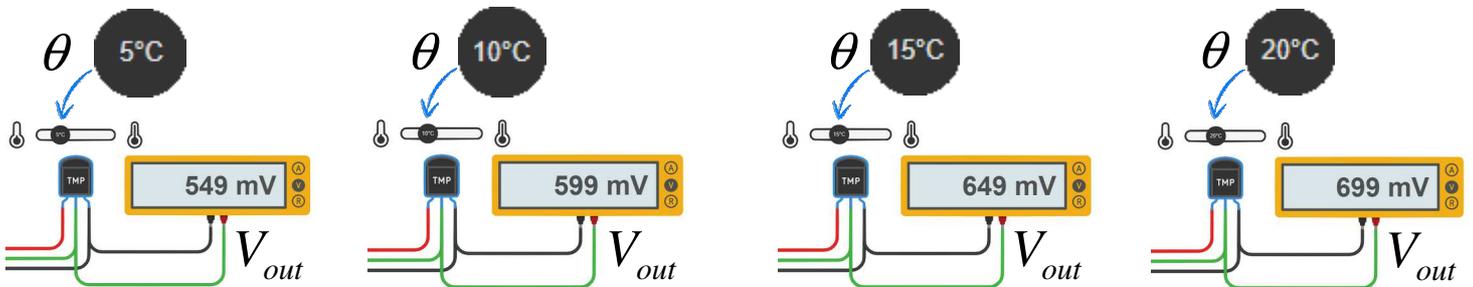
Le capteur est actif, il a besoin d'être alimenté.  $V_{in} = 5V$

Voltmètre pour mesurer  $V_{out}$

**i** Dans l'immédiat, il n'y a pas besoin de programme informatique dans la carte.

On étalonne le capteur en suivant la procédure d'étalonnage disponible en ligne (document 3).

Pour le capteur TMP36 du simulateur TinkerCAD, avec 4 valeurs relevées entre +5 °C et +20 °C, on obtient l'étalonnage suivant : (le capteur est linéaire)



$mV$  →      →  $^{\circ}C$

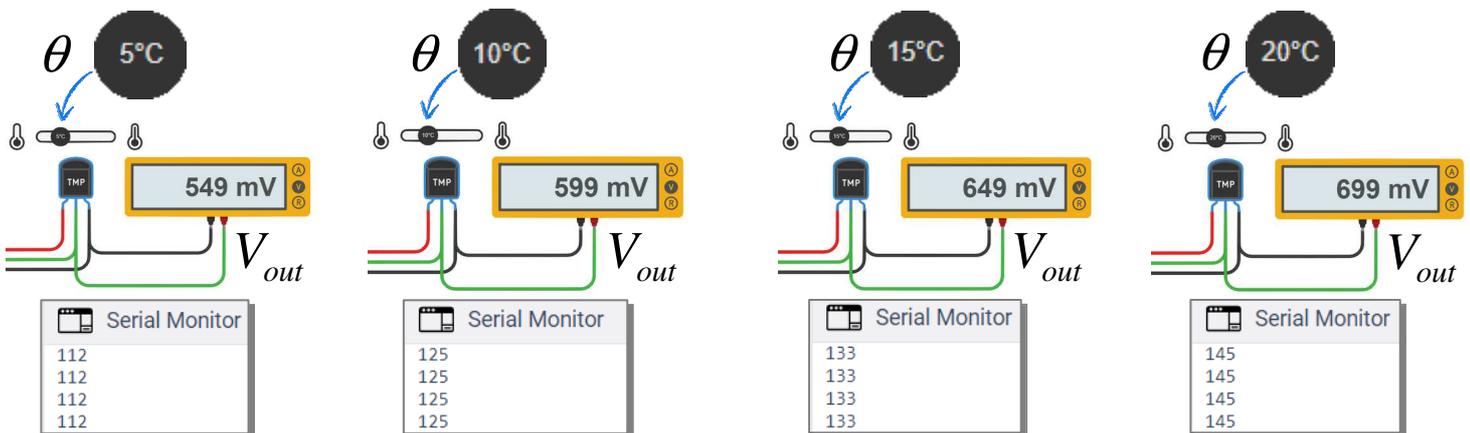
$$V_{out}(\theta) = 10 \cdot \theta + 499$$

$^{\circ}C$  →      →  $mV$

$$\theta(V_{out}) = 0,1 \cdot V_{out} - 49,9$$

Allons voir maintenant la façon dont un programme informatique gère tout cela.

```
1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   int sensorValue = analogRead(A3);
7   Serial.println(sensorValue);
8   delay(100);
9 }
```



On constate dans le moniteur série que ce qui est en fait manipulable dans le programme, c'est une valeur numérique. A chaque température  $\theta$  correspond donc une tension  $V_{out}$  mais aussi une valeur numérique.

Soit  $N$  cette valeur numérique.

Quelle est la relation entre  $N$  et  $\theta$  ? Car au final, dans le programme, c'est bien cette relation qu'il va falloir exploiter !

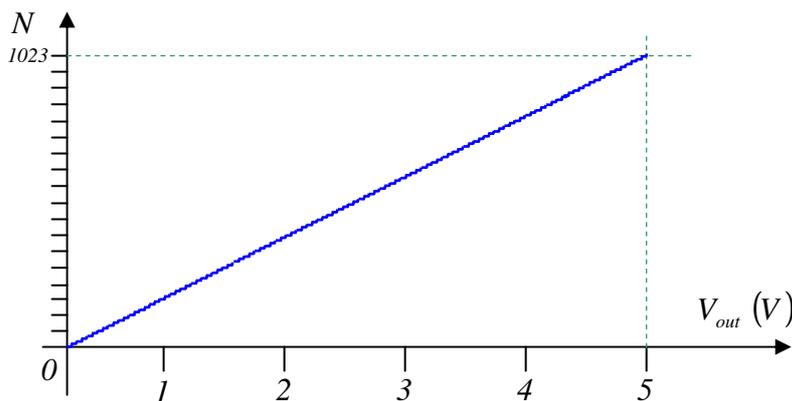
Sur la broche A3 de la carte, on a bien la tension électrique  $V_{out}$ . Donc entre la tension  $V_{out}$  sur la broche et le programme avec ses valeurs numériques, il se passe quelque chose...

Ce quelque chose est une **numérisation du signal** correspondant à la tension  $V_{out}$ .

Pour une carte Arduino, d'après sa documentation :

- La tension admise sur les broches A0 à A5 doit être comprise entre **0V et 5V**.
- La tension est codée sur **10 bits**, soit  $2^{10} = 1024$  valeurs numériques comprise entre **0 et 1023**.

Dans un premier temps, on a donc les correspondances suivantes : (résultat de la numérisation du signal)



$$N(V_{out}) = V_{out} \times \frac{1024}{5}$$

*Valeurs valables pour une carte Arduino.*

$$V_{out}(N) = N \times \frac{5}{1024}$$

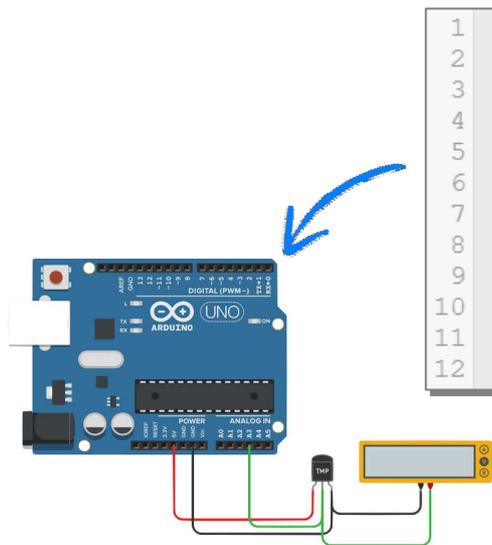
Correspondance entre la tension (V) et sa valeur numérique codée sur 10 bits

Dans un second temps, on obtient la relation  $\theta(N)$  en combinant  $\theta(V_{out})$  et  $V_{out}(N)$  :

$$\left\{ \begin{array}{l} \theta(V_{out}) = 0,1 \cdot V_{out} - 49,9 \\ V_{out}(N) = N \times \frac{5}{1024} \end{array} \right. \Rightarrow \theta(N) = 0,488 \cdot N - 49,9$$

*Attention : on a converti les mV en V pour être cohérent !*

C'est cette équation qu'il faut mettre dans le programme !

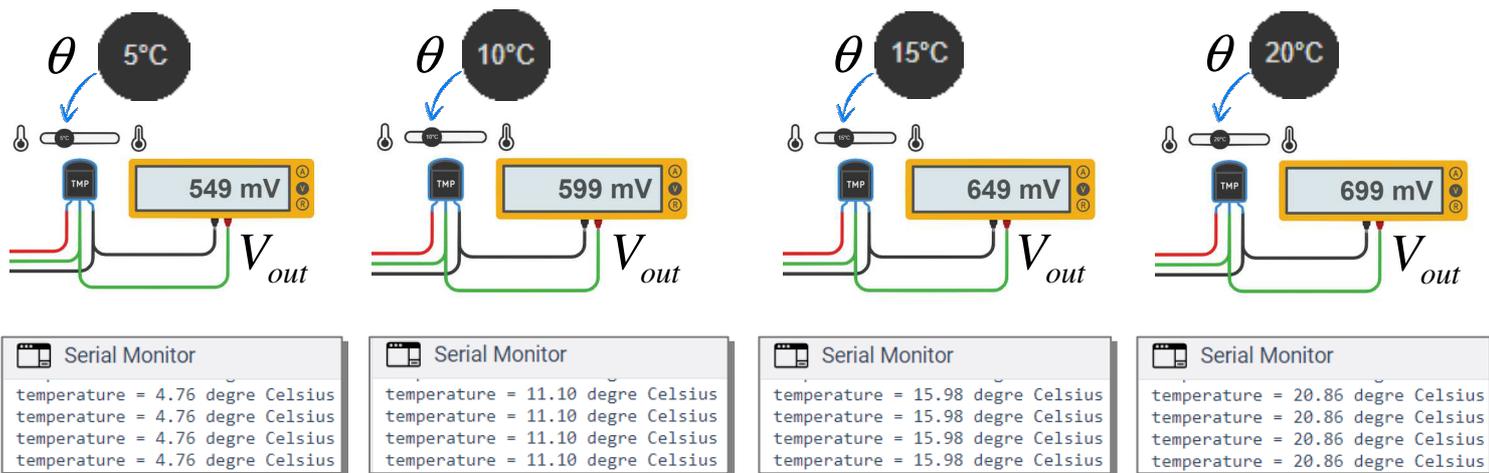


```

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6   int sensorValue = analogRead(A3);
7   float temperature = (sensorValue * 0.488) - 49.9;
8   Serial.print("temperature = ");
9   Serial.print(temperature);
10  Serial.println(" degre Celsius");
11  delay(100);
12 }

```

Il n'y a plus qu'à tester...



On note des écarts entre la température réelle et celle affichée dans le moniteur, mais **ça marche !**

Les écarts peuvent être réduits en jouant sur les coefficients de l'équation (coefficient directeur et/ou ordonnée à l'origine).



La valeur de la température a été simplement affichée dans moniteur série mais dans le programme, on peut la manipuler comme bon nous semble ! (comparaison à un seuil, etc.)