

Conception et réalisation d'un « Détecteur de fièvre »

Fiche professeur

Contexte : Gauthier a des soucis de fièvres se produisant n'importe quand dans la semaine à la suite d'un problème de santé mettant en jeu son système rénal. Il souhaite créer un montage à partir d'un capteur thermique afin de vérifier rapidement s'il a de la fièvre. Un voyant vert permettrait de vérifier que son montage fonctionne, un voyant orange s'allumerait si sa température dépasse 38 °C et enfin un voyant rouge s'allumerait à son tour si sa température dépasse 40 °C.

Il commence par faire différentes recherches sur internet et trouve les renseignements des documents fournis concernant un capteur de température référencé TMP36 et la programmation correspondante d'une carte Arduino™.

Doc. 1 Le capteur de température TMP36

Le composant



Caractéristiques données par le constructeur

- Tension d'entrée : de 2,7 à 5,5 V en continu
- Facteur d'échelle de 10 mV/°C
- Plage de fonctionnement : - 40 à + 125 °C
- Tension de sortie : 0.1 V (-40°C) à 2.0 V (150°C, mais perte de précision au-delà de 125 °C)

Doc.2 Programme de la carte ARDUINO™ pour l'allumage d'une seule diode électroluminescente, dès que la température mesurée dépasse 20 °C

```
//initialisation des constantes SensorPin pour l'entrée analogique et baselineTemp pour la
température de référence
const int sensorPin = A0;
const float baselineTemp = 18.0;
const float temp1 = baselineTemp+2.0;
// ouverture du port série de connexion entre la carte Arduino™ et l'ordinateur, lecture en
boucle des broches de sortie vers les LED et initialisation de ces sorties
void setup() {
  Serial.begin(9600);
  for(int pinNumber = 2; pinNumber<5; pinNumber++){
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
}
// sensorVal stocke la lecture du capteur, analogRead() lit la tension sur SensorPin (A0 ici)
```

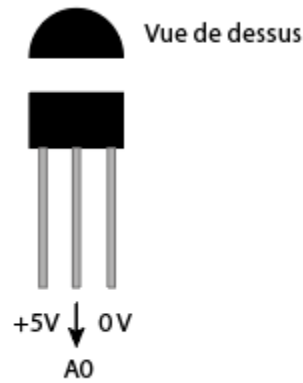
```

void loop() {
  int sensorVal = analogRead(sensorPin);
  Serial.print("Valeur du capteur :");
  Serial.print(sensorVal);
  // conversion en tension puis conversion de la tension en température
  float tension = (sensorVal/1023.0)*5.0;
  Serial.print(", Tension en V : ");
  Serial.print(tension);
  Serial.print(", Température en °C : ");
  float temperature = (tension-0.5)*100;
  Serial.print(temperature);
  if(temperature < temp1){
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }else if((temperature >= temp1){
    digitalWrite(2, HIGH);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
  }
  delay(1);
}

```

Matériel à disposition :

- Un ordinateur possédant un port USB et équipé du [logiciel](#) gratuit développé par Arduino pour ses microcontrôleurs ainsi que du logiciel gratuit de schématisation [fritzing](#) ;
- Une carte à microcontrôleur, de type Arduino™, ou autre ;
- Une plaquette de montage breadboard ;
- 8 fils électriques ;
- 3 DEL de couleurs verte, jaune et rouge (la borne d'entrée du courant est la patte longue pour que la DEL soit passante) et 3 résistances de 220 Ω à brancher en série avec les DEL afin d'éviter qu'elles ne grillent;
- Un capteur de température TMP36. Ce composant délivre une tension proportionnelle à la température, et ce dans une large gamme de températures. Remarque : avec le méplat vu vers soi, la patte de gauche est à relier au +5 V de la carte et la patte de droite à la masse (0 V). La patte du milieu est reliée à l'entrée A0 analogique de la carte.



1. Travail à réaliser : utilise les documents à disposition pour réaliser le détecteur de fièvre souhaité par Gauthier.

Travail du professeur : suivre les élèves et valider les étapes des raisonnements

2. Questions sur le montage :

- a. Donne une estimation de la précision sur la mesure de la température dans la gamme de tensions fournie par le constructeur. Compare cette précision sur la mesure avec la précision donnée par la console de mesures.

La carte fournit 1024 valeurs de 0 à 1023 correspondant à des tensions de 0 à 5,0 V. Chaque pas de mesure correspond à $5,0/1023 = 4,9$ mV. Le capteur a pour valeurs de conversion 10 mV par °C. 4,9 mV correspondent ainsi à 0,49 °C soit environ un intervalle de confiance de 0,5 °C. La précision de l'affichage est donc illusoire car trop précise.

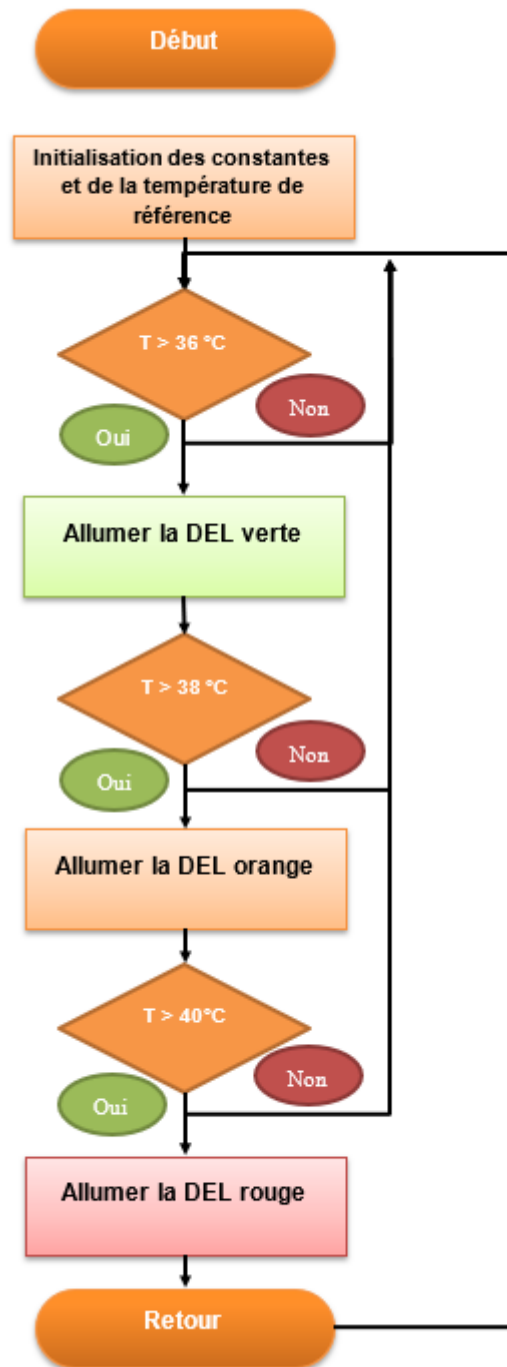
- b. En utilisant les renseignements de la fiche outil, indique les modifications à faire dans les formules du programme afin d'utiliser un capteur LM335 à la place du capteur TMP36.

Exemple de correction avec un raisonnement en volts :

```
float tension = (sensorVal/1023.0)*5.0;
Serial.print(" Tension en V : ");
Serial.print(tension);
Serial.print(" Température en °C : ");
float temperature = tension*100-273,15;
```

En annexe des éléments de correction pour chacune des étapes du raisonnement :

1. Exemple d'organigramme :



2. Schématisation

Le logiciel Fritzing permet de faire des schémas électriques normalisés (onglet vue schématique) ou des schémas réalistes (onglet « Platine d'essai).

La liste des composants se trouve dans la fenêtre de droite. La carte Arduino y figure aussi :

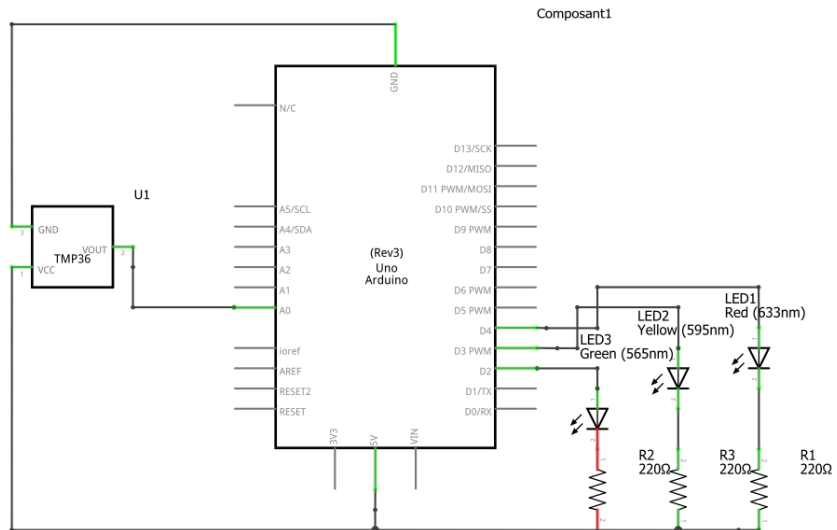
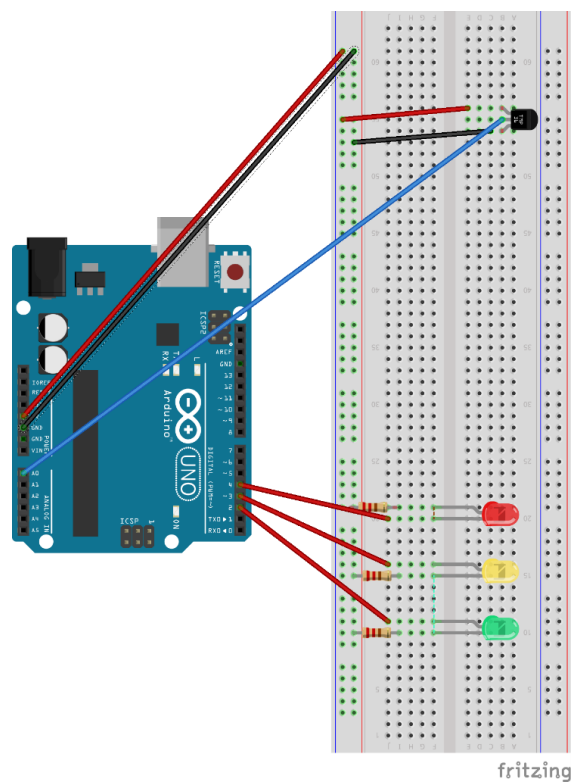
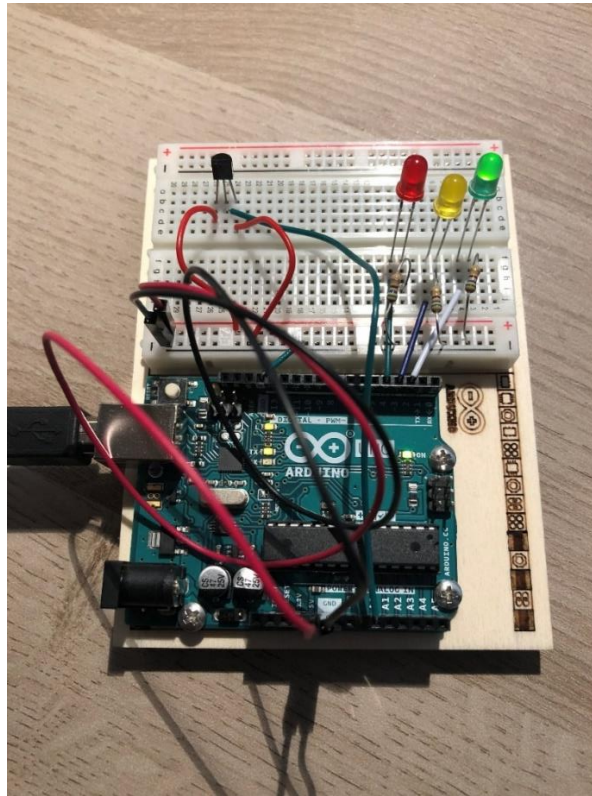


Schéma réalisé avec le logiciel gratuit [Fritzing](#)

Fritzing permet de faire aussi des **schémas réalistes** :



3. Montage final à réaliser :



Conseil pour un fonctionnement correct : ne pas brancher les DEL et le capteur de température sur la même partie de la breadbord afin d'éviter des courants de fuite nuisibles à un fonctionnement correct.

4. Exemple de programme complet pour le montage à 3 DEL

```
//initialisation des constantes SensorPin pour l'entrée analogique et baselineTemp pour la
température de référence
const int sensorPin = A0;
const float baselineTemp = 35.0;
const float temp1 = baselineTemp+1.0;
const float temp2 = baselineTemp+3.0;
const float temp3 = baselineTemp+5.0;
```

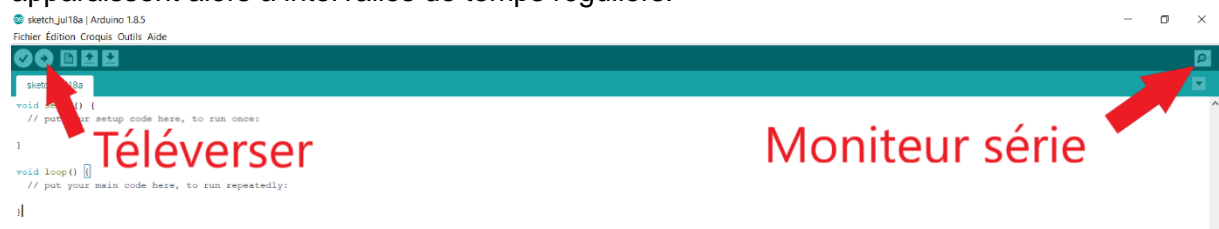
```
// ouverture du port série de connexion entre l'arduino et l'ordinateur, lecture en boucle des
broches de sortie vers les LED et initialisation de ces sorties
void setup() {
  Serial.begin(9600);
  for(int pinNumber = 2; pinNumber<5; pinNumber++){
    pinMode(pinNumber, OUTPUT);
    digitalWrite(pinNumber, LOW);
  }
}
```

```
// sensorVal stocke la lecture du capteur, analogRead() lit la tension sur SensorPin (A0 ici)
void loop() {
```

```
int sensorVal = analogRead(sensorPin);
Serial.print("Valeur du capteur :");
Serial.print(sensorVal);
// conversion en tension
float tension = (sensorVal/1024.0)*5.0;
Serial.print(", Tension en V : ");
Serial.print(tension);
```

```
// conversion de la tension en température et affichage (une variation de tension de 10 mV
correspond à une variation de température de 1°C)
Serial.print(", Température en °C : ");
float temperature = (tension-0.5)*100;
Serial.print(temperature);
if(temperature < temp1){
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
}else if((temperature >= temp1) && (temperature < temp2)){
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
}else if((temperature >= temp2) && (temperature < temp3)){
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
}else if(temperature >= temp3){
  digitalWrite(2, HIGH);
  digitalWrite(3, HIGH);
  digitalWrite(4, HIGH);
}
delay(1);
}
```

Une fois le programme tapé, il faut le téléverser vers la carte Arduino™. Le fonctionnement peut être testé à tout moment avec le moniteur série. Les valeurs de températures apparaissent alors à intervalles de temps réguliers.



Professeur : vérifier si erreurs de syntaxe lors du téléversement.