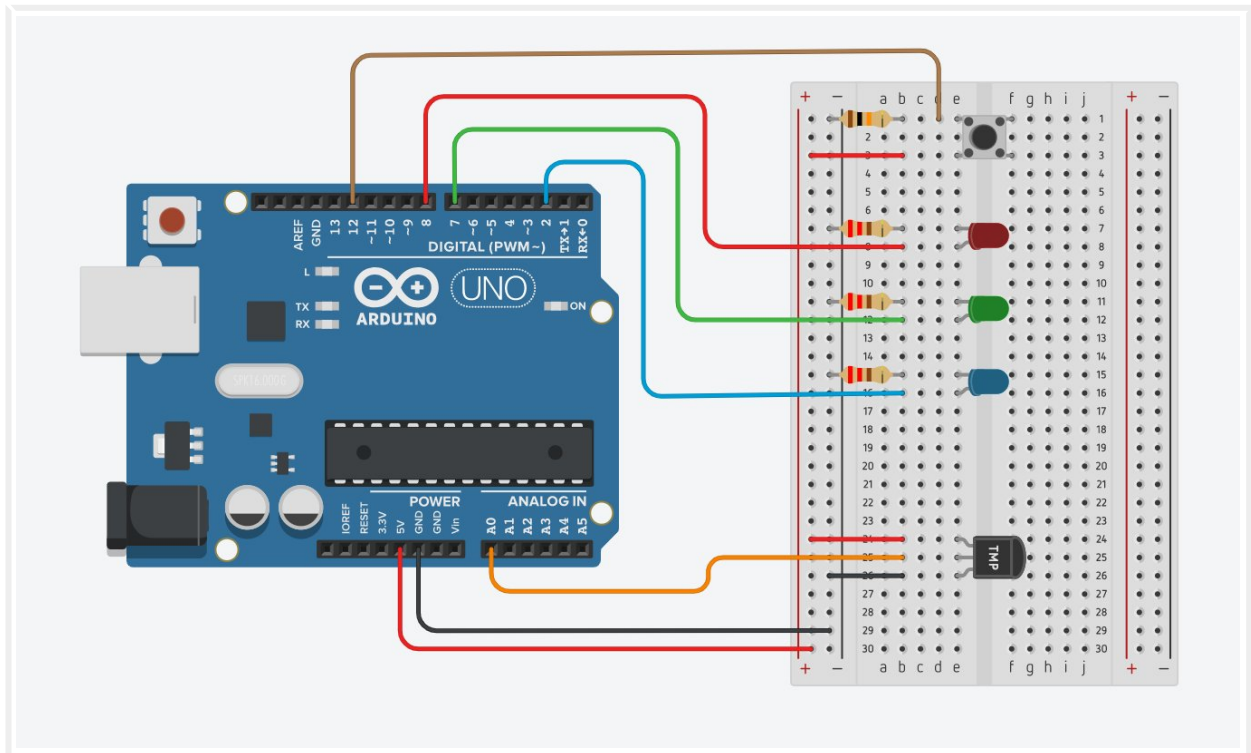


Températures & Dels

(Thermomètre à diodes électroluminescentes)



. Liste des composants

- . 1 capteur de température (TMP 36 ou LM 35)
- . 1 DEL rouge
- . 1 DEL verte
- . 1 DEL bleue
- . 3 résistances de 220 Ω (résistances de protection des DELs)
- . 1 résistance de 10 k Ω (résistance du circuit du bouton poussoir)
- . 1 bouton poussoir
- . 1 plaque d'essais
- . Fils de connexion

. Objectif

Dans cette activité, nous allons utiliser les DELS rouge, verte et bleue du circuit d'étude afin de visualiser la zone dans laquelle se situe la température mesurée ($T_{\text{mesurée}}$) par un capteur TMP 36 ou LM 35 par rapport à une valeur de référence (T_{ref}) et un écart de température (ΔT) à définir :

- si $T_{\text{mesurée}} < T_{\text{ref}} - \Delta T$: La DEL bleue est allumée,
- si $T_{\text{mesurée}} > T_{\text{ref}} + \Delta T$: La DEL rouge est allumée,
- $T_{\text{ref}} - \Delta T < T_{\text{mesurée}} < T_{\text{ref}} + \Delta T$: La DEL verte est allumée.

. Le programme

Le code de l'activité pourra être modifié pour voir l'influence des variables (température de référence, écart de température).

Temperatures_dels

```
// Déclaration des constantes et variables
```

```
const int PinSensor=0;
const int PinButton= 12;
const int PinLedR = 8;
const int PinLedV = 7;
const int PinLedB = 2;
const float TempRef = 20.0;
const float DT = 1.0;
```

```
int ValSensor = 0;
float tension = 0.0;
float Temp = 0.0;
float OldTemp = 0.0;
```

```
int ValButton = 0;
int OldValButton = 0;
int State = 0;
int OldState = 0;
```

```
// Initialisation des entrées et sorties
```

```
void setup() {
  Serial.begin(9600);
  pinMode(PinButton, INPUT);
  pinMode(PinLedR, OUTPUT);
  pinMode(PinLedV, OUTPUT);
  pinMode(PinLedB, OUTPUT);
  Serial.println("Appuyez sur le bouton poussoir pour commencer les mesures.");
}
```

```

// Fonction principale en boucle

void loop() {
  ValButton = digitalRead(PinButton);
  delay(10);

  if ((ValButton == HIGH) && (OldValButton == LOW))
  {
    State=1-State;
  }
  OldValButton = ValButton;

  if (State==1)
  {
    if (OldState == 0)
    {
      Serial.println("Mesure de la temperature en cours.");
      Serial.println("");
      Serial.println ("Temperature en degre Celsius:");
      OldState=1;
    }
    ValSensor = analogRead(PinSensor);
    tension = (ValSensor/1023.0)*5.0;

    // Capteur TMP 36
    Temp = (tension - 0.5) * 100;

    // Capteur LM 35
    //Temp = tension * 100;

    if (OldTemp != Temp)
    {
      Serial.println(Temp,1);
      OldTemp = Temp;
    }
    if ((Temp > TempRef - DT) && (Temp < TempRef + DT)) {
      digitalWrite(PinLedR, LOW);
      digitalWrite(PinLedV, HIGH);
      digitalWrite(PinLedB, LOW);
    }
  }
}

```

```

if (Temp > TempRef + DT) {
    digitalWrite(PinLedR, HIGH);
    digitalWrite(PinLedV, LOW);
    digitalWrite(PinLedB, LOW);
}

if (Temp < TempRef - DT) {
    digitalWrite(PinLedR, LOW);
    digitalWrite(PinLedV, LOW);
    digitalWrite(PinLedB, HIGH);
}
delay(100);
}
else
{
    if (OldState == 1){
        Serial.println("Fin des mesures.");
        digitalWrite(PinLedB, LOW);
        digitalWrite(PinLedR, LOW);
        digitalWrite(PinLedV, LOW);
        OldState = 0;}
    }
}

```

Déroulement du programme :

– Déclaration des constantes et variables :

- N° de la broche correspondant au bouton poussoir: **const int PinButton = 12**
- N° de la broche correspondant à la DEL rouge : **const int PinLedR = 8**
- N° de la broche correspondant à la DEL verte : **const int PinLedV = 7**
- N° de la broche correspondant à la DEL bleue : **const int PinLedB = 2**
- N° de la broche correspondant au capteur de température : **const int PinSensor = 0**
- Variable pour stocker la valeur de la broche du bouton poussoir: **int ValButton = 0**
- Variable pour stocker l'ancienne valeur de la broche du bouton poussoir: **int OldValButton = 0**
- Variable correspondant à l'action à effectuer: **int State = 0**
- Variable pour stocker l'ancienne valeur de la variable correspondant à l'action à effectuer: **int OldState = 0**
- Variable pour stocker la valeur de la broche du capteur de température: **int ValSensor = 0**
- Variable pour stocker la valeur en V de la tension correspondante à la valeur de la broche du capteur : **float tension = 0.0**
- Variable correspondant à la température calculée à partir de la valeur de la broche du capteur: **float Temp = 0.0**
- Variable correspondant à la température mesurée précédemment: **float OldTemp = 0.0**
- Variable correspondant à la température de référence: **float TempRef = 20.0**
- Variable correspondant à l'écart de température par rapport à la température de référence: **float DT = 1.0**

– Initialisation des entrées et sorties :

- le débit de communication en nombre de caractères par seconde pour la communication série est fixé à 9600 bauds: **Serial.begin(9600)**
- La broche du bouton poussoir est initialisée comme une entrée digitale. Des données seront donc envoyées depuis cette broche vers le microcontrôleur: **pinMode (PinButton, INPUT)**
- Les broches des DELs rouge, verte et bleue sont initialisées comme des sorties digitales. Des données seront donc envoyées depuis le microcontrôleur vers ces broches.

– Fonction principale en boucle :

