

Projet Prototypage

Qiancheng Polina Olivier

Fatmagül

Notre projet consiste à concevoir un dispositif de mesure et de prévention de température, combiné avec une fonctionnalité d'affichage d'humidité. Il utilise plusieurs composants électroniques contrôlés par une carte Arduino UNO. L'objectif est de prévenir tout dépassement de seuil de température prédéfini et d'afficher l'humidité ambiante de manière visuelle.

Fonctionnement

1. Réglage de la température limite (potentiomètre)

Le dispositif intègre un potentiomètre qui permet à l'utilisateur de fixer une température limite souhaitée. Cette valeur servira de seuil critique pour déclencher une alerte en cas de dépassement.

2. Mesure de la température (thermomètre)

Un capteur de température mesure en temps réel la température ambiante. La valeur relevée est comparée à la température limite définie via le potentiomètre.

3. Alerte de dépassement (LED)

Si la température mesurée dépasse le seuil fixé, une LED rouge s'allume pour avertir l'utilisateur de ce dépassement.

4. Affichage de l'humidité (servo-moteur)

Un capteur d'humidité est relié à un servo-moteur. Celui-ci affiche visuellement l'humidité mesurée grâce à une aiguille qui se déplace en fonction du taux d'humidité ambiant sur une feuille sur laquelle on retrouve des degrés d'humidité.

Ce projet avait pour objectif d'être utilisé dans des laboratoires où des tests nécessitent un strict contrôle de la température. Il est essentiel de disposer d'un outil de prévention capable de signaler tout dépassement de seuil critique.

Cependant, en raison de codes d'erreur et de la complexité technique rencontrée, nous n'avons pas pu finaliser le projet et obtenir les résultats escomptés. Le nouveau code que nous avons développé permet néanmoins de faire varier le clignotement de la LED en fonction du réglage du potentiomètre, offrant une visualisation dynamique de l'état du système. Cette fonctionnalité peut constituer une base pour de futurs développements.

Voici le code utilisé dans ce projet:

```
int sensorPin = A0; // select the input pin for the potentiometer
int ledPin = 13;    // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
  Serial.begin(115200);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  Serial.print("potentiometer value : ");
  Serial.println(sensorValue);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
}
```

Revision #7

Created 7 February 2025 10:17:52 by Nguyen Olivier

Updated 24 September 2025 09:23:22 by Ozgul Fatmagul