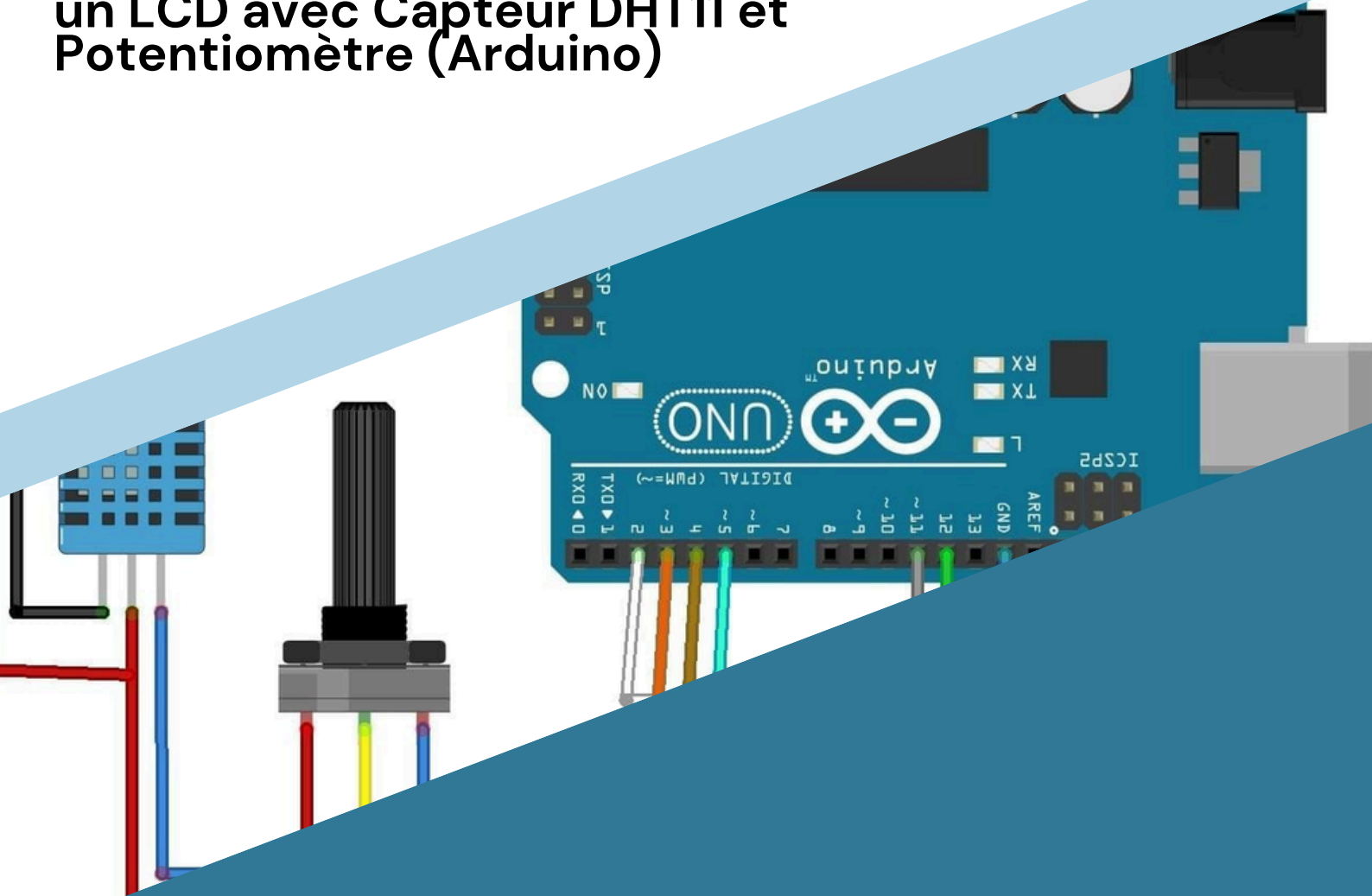


AVRIL 2025



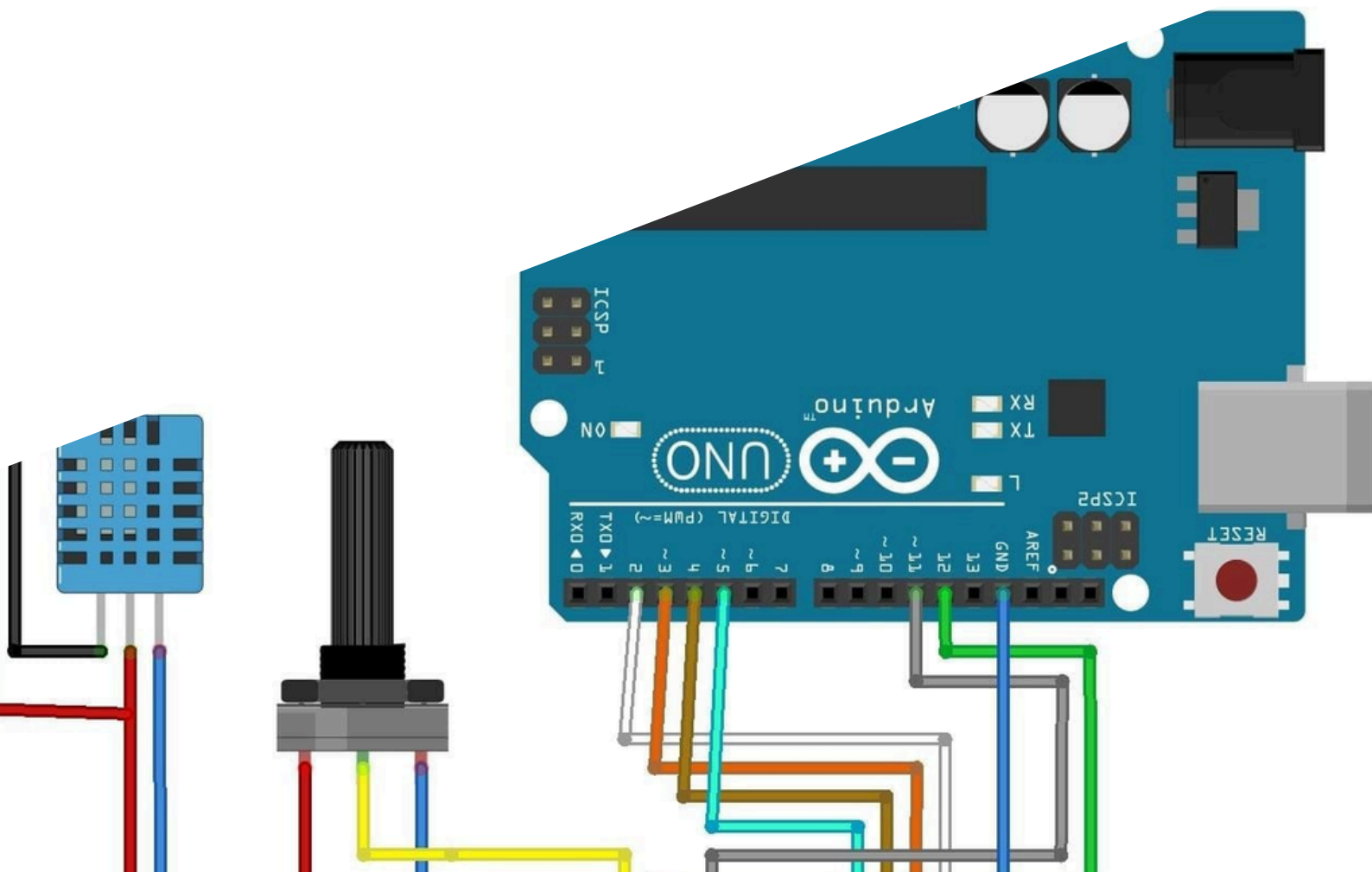
# Affichage de Température sur un LCD avec Capteur DHT11 et Potentiomètre (Arduino)



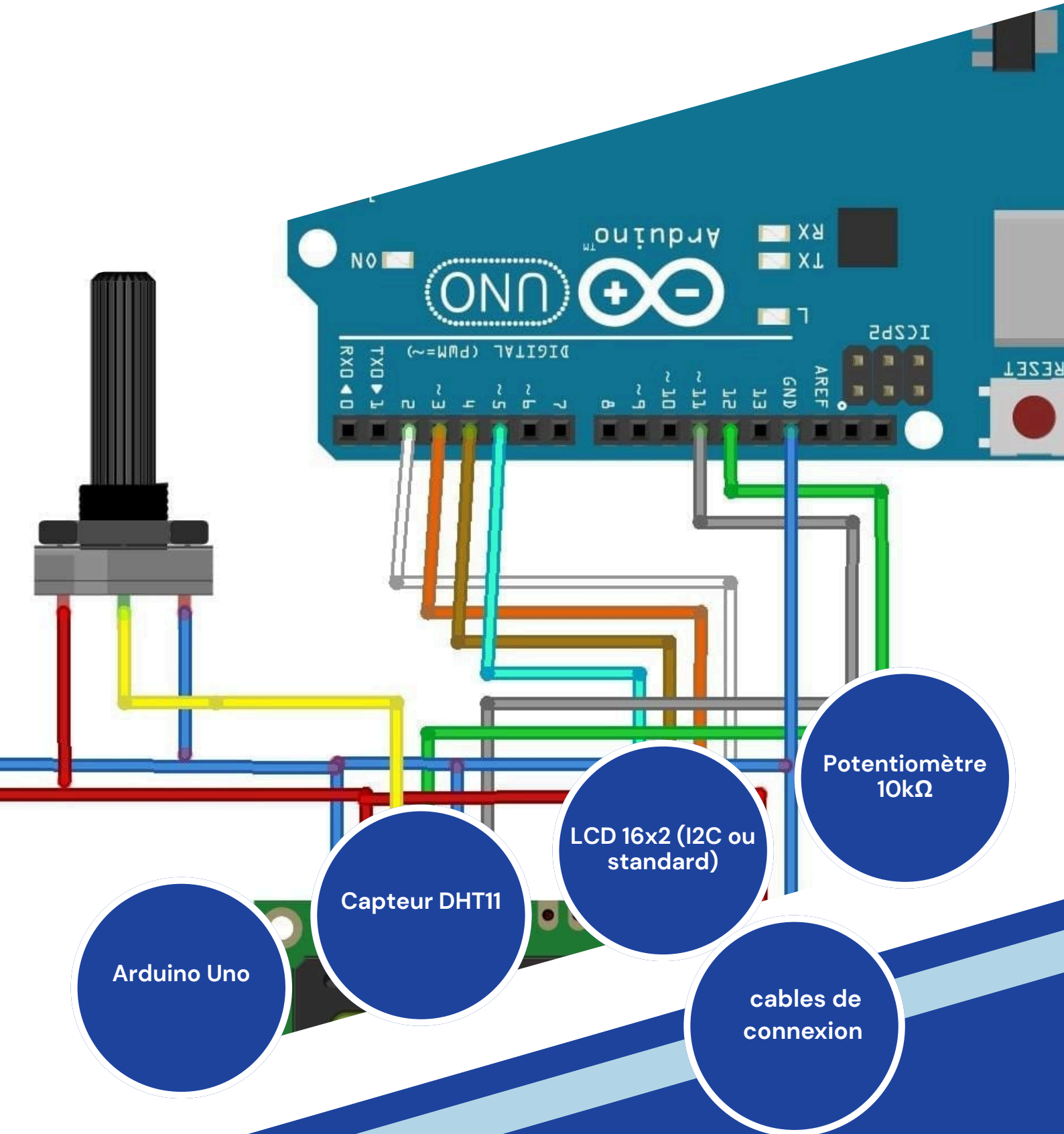
elaboré par :  
-Haj amor eslem  
-said mohamed  
-amiri eya

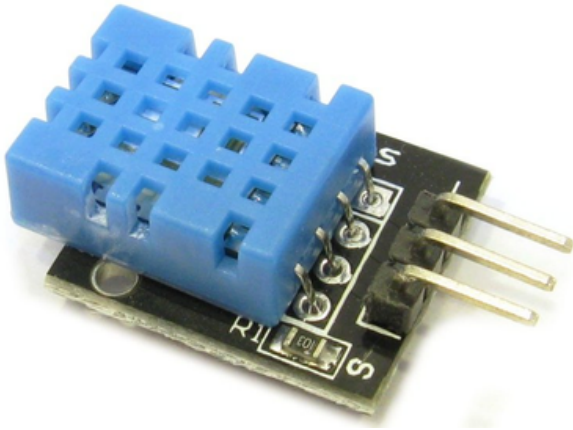
# Introduction

Ce rapport présente la conception et la réalisation d'un système permettant d'afficher la température et l'humidité mesurées par un capteur DHT11 sur un écran LCD 16x2, avec un réglage de luminosité via un potentiomètre. Le tout est contrôlé par une carte Arduino Uno.



# Matériel Utilisé





## ➔ Arduino Uno

- Microcontrôleur pour le traitement des données.

## ➔ Capteur DHT11

- Mesure la température et l'humidité ambiante.

## ➔ LCD 16x2 (I2C ou standard)

- Afficheur pour visualiser les données.

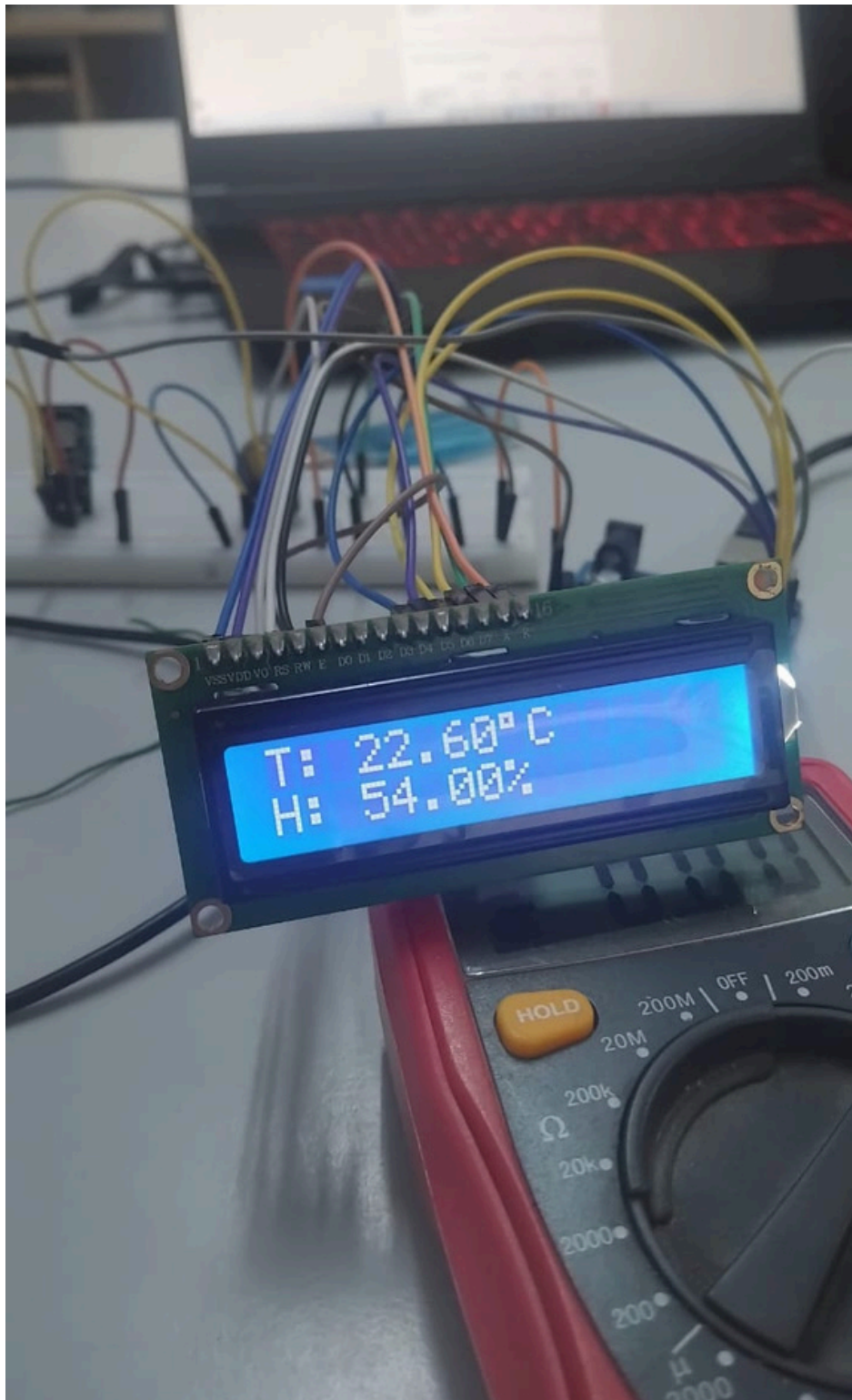
## ➔ Potentiomètre 10kΩ

- Permet d'ajuster la luminosité du rétroéclairage du LCD.

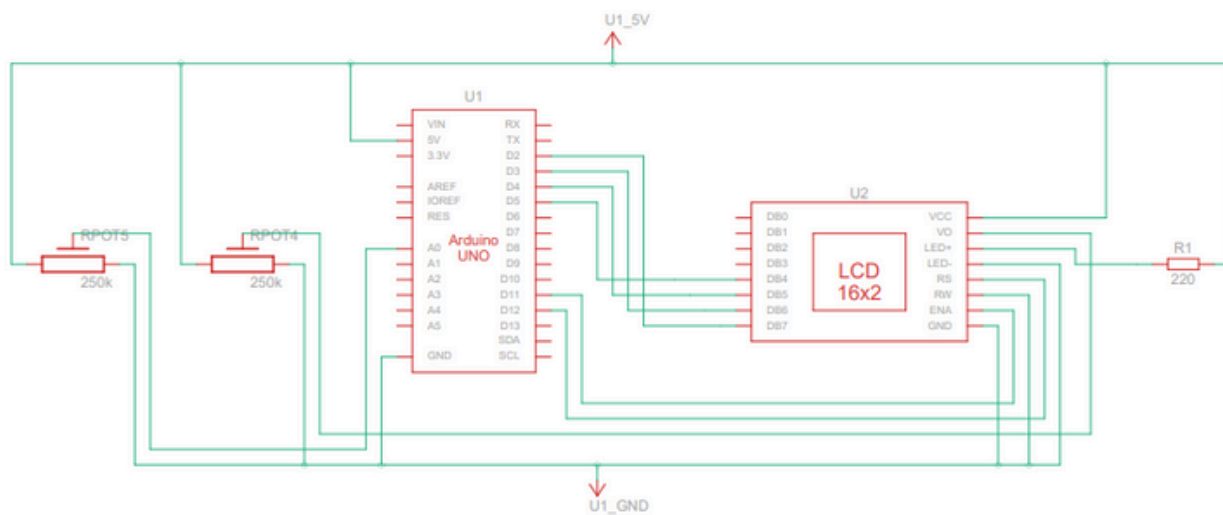
## ➔ câbles

- Pour les connexions.

# Photo réelle du projet



# Schéma de Câblage





# Programmation arduino



```
1 #include <DHT.h> // Inclusion de la bibliothèque pour le capteur DHT (température et humidité)
2 #include <LiquidCrystal.h> // Inclusion de la bibliothèque pour l'écran LCD
3
4 // Définir les broches
5 #define DHTPIN 7 // Définition de la broche de données du capteur DHT11
6 #define DHTTYPE DHT11 // Spécification du type de capteur utilisé (DHT11)
7
8 // Initialiser le capteur DHT avec la broche et le type définis
9 DHT dht(DHTPIN, DHTTYPE);
10
11 // Initialiser l'écran LCD avec les broches connectées : RS, EN, D4, D5, D6, D7
12 const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
13 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
14
15 void setup() {
16   Serial.begin(9600); // Initialisation de la communication série à 9600 bauds (pour affichage dans le moniteur série)
17   dht.begin(); // Initialisation du capteur DHT11
18   lcd.begin(16, 2); // Initialisation de l'écran LCD (16 colonnes, 2 lignes)
19
20   lcd.print("Initialisation..."); // Affichage d'un message d'initialisation sur l'écran LCD
21   delay(2000); // Attente de 2 secondes pour permettre la lecture du message
22   lcd.clear(); // Effacement de l'écran LCD
23 }
24
25 void loop() {
26   delay(1000); // Pause d'1 seconde entre chaque mesure
27
28   float h = dht.readHumidity(); // Lecture de l'humidité relative en pourcentage
29   float t = dht.readTemperature(); // Lecture de la température en degrés Celsius
30
31   // Vérification si la lecture a échoué (NaN = Not a Number)
32   if (isnan(h) || isnan(t)) {
33     lcd.clear(); // Efface l'écran
34     lcd.print("Erreur lecture!"); // Affiche un message d'erreur
35     return; // Quitte la boucle loop et recommence
36   }
37
38   lcd.clear(); // Efface l'écran pour afficher de nouvelles données
39
40   // Affichage de la température sur la première ligne
41   lcd.setCursor(0, 0); // Positionnement du curseur au début de la première ligne
42   lcd.print("T: "); // Affiche "T: " pour Température
43   lcd.print(t); // Affiche la température
44   lcd.print((char)223); // Affiche le symbole du degré (°)
45   lcd.print("C"); // Affiche "C" pour Celsius
46
47   // Affichage de l'humidité sur la deuxième ligne
48   lcd.setCursor(0, 1); // Positionnement du curseur au début de la deuxième ligne
49   lcd.print("H: "); // Affiche "H: " pour Humidité
50   lcd.print(h); // Affiche l'humidité
51   lcd.print("%"); // Affiche le symbole de pourcentage
52
53   // Affichage des valeurs sur le moniteur série
54   Serial.print("Humidite: "); // Affiche "Humidite: " dans le moniteur série
55   Serial.print(h); // Affiche la valeur de l'humidité
56   Serial.print("% Temperature: "); // Affiche "Temperature: " après l'humidité
57   Serial.print(t); // Affiche la température
58   Serial.println("°C"); // Affiche le symbole °C et passe à la ligne suivante
59 }
```



- Ajoutant un affichage graphique (OLED).
- Utilisation iot pour la supervision de données avec un module Wi-Fi(ESP32).



