Réalisé par Mabrouk Saifeddine

Mini-projet : Système de surveillance de distance et de température avec avertissement sonore.

Introduction :

J’ai réalisé un mini projet en utilisant une carte Arduino Uno et plusieurs composants électroniques. L'objectif de ce projet était de mesurer la distance à l'aide d'un capteur à ultrasons, d'afficher la distance sur un écran LCD, de mesurer la température et de produire un avertissement sonore si la distance mesurée était trop courte.

Composants utilisés :

- Carte Arduino Uno
- Capteur à ultrasons
- Capteur de température et d'humidité DHT11
- Écran LCD
- Buzzer
- cables

Mise en place du système :

J’ai connecté le capteur à ultrasons et le capteur DHT11 à la carte Arduino Uno et J’ai utilisé la bibliothèque LiquidCrystal_I2C pour l'écran LCD. J’ai ensuite programmé la carte Arduino Uno pour mesurer la distance et la température, afficher les résultats sur l'écran LCD et produire un avertissement sonore si la distance mesurée était inférieure à 5 cm.

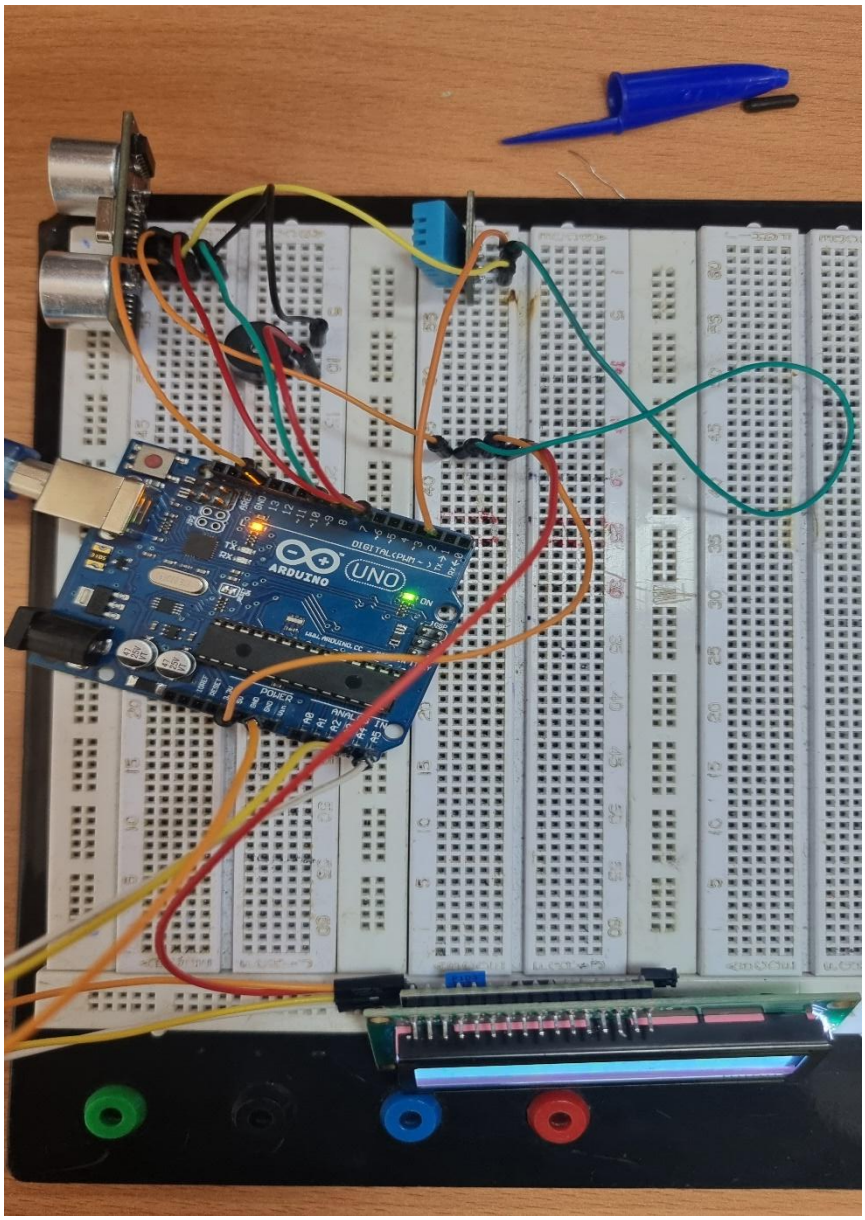
Résultats : J’ai obtenu des mesures de la distance et de la température, qui étaient affichées sur l'écran LCD. L'avertissement sonore a bien fonctionné lorsque la distance mesurée était inférieure à 5 cm.

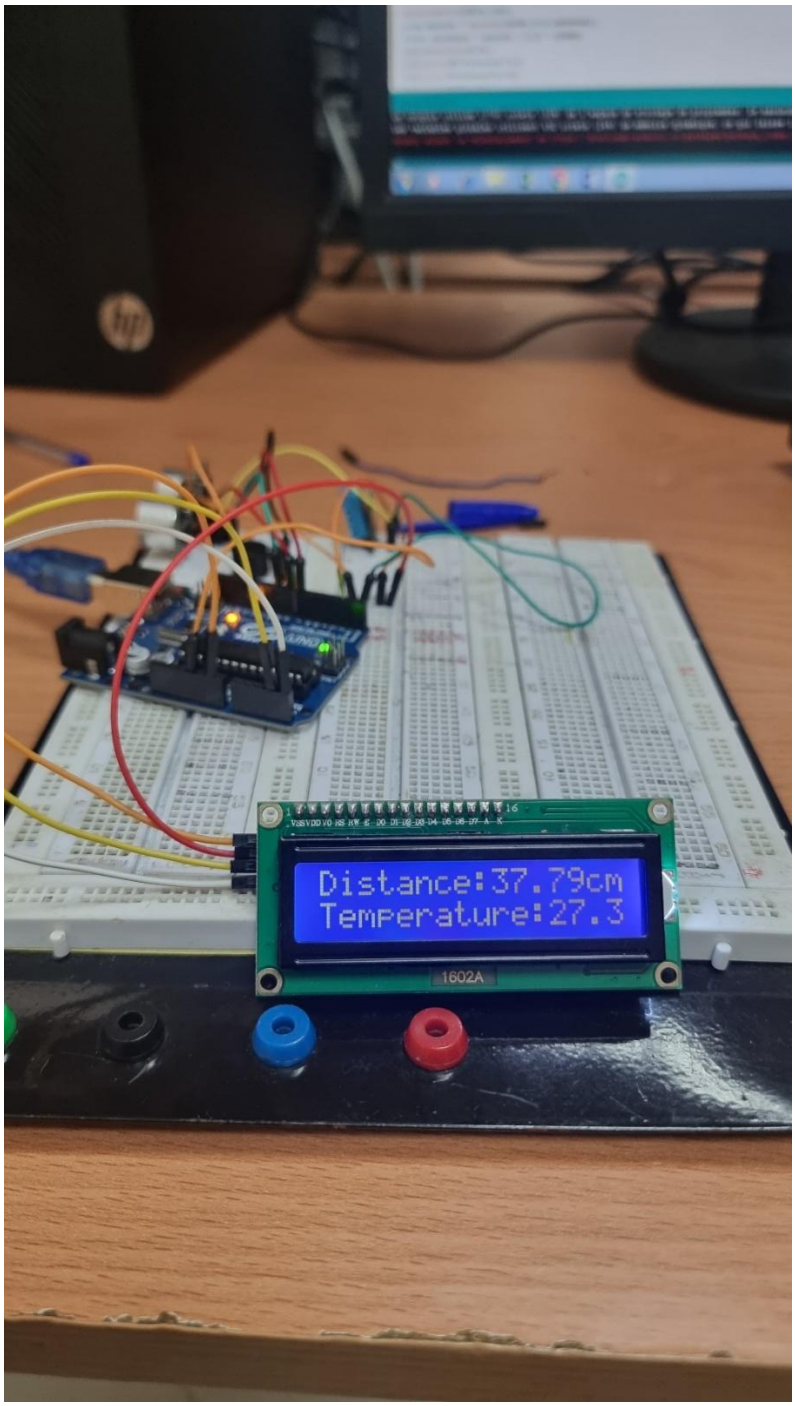
Code :

```
#include <DHT.h> // Importation de la bibliothèque DHT
#include <DHT_U.h> // Importation de la bibliothèque DHT_U
#include <Adafruit_Sensor.h> // Importation de la bibliothèque Adafruit_Sensor
#include <LiquidCrystal_I2C.h> // Importation de la bibliothèque LiquidCrystal_I2C
LiquidCrystal_I2C lcd(0x27,16,2) ; // Initialisation d'un objet de type LiquidCrystal_I2C avec l'adresse I2C 0x27, 16 colonnes et 2 lignes
const unsigned long MEASURE = 25000UL; // Définition de la constante MEASURE à 25000 microsecondes
const float SOUND = 340.0 / 1000; // Définition de la constante SOUND à la vitesse du son (340 m/s) divisée par 1000 pour obtenir la vitesse en millisecondes
const byte TRIG = 8; // Définition de la broche TRIG à 8
const byte ECHO = 9; // Définition de la broche ECHO à 9
int buzz = 7; // Déclaration de la variable buzz et initialisation à 7
DHT temp(2, DHT11); // Initialisation d'un objet de type DHT pour mesurer la température sur la broche 2 avec un capteur DHT11
void setup() {//Cette fonction est exécutée une seule fois au démarrage du programme. Elle est généralement utilisée pour initialiser les broches d'entrée/sortie et les paramètres de configuration de la carte Arduino.
  Serial.begin(115200) ; // Initialisation de la communication série à une vitesse de 115200 bauds
  pinMode(TRIG, OUTPUT); // Configuration de la broche TRIG en sortie
  digitalWrite(TRIG, LOW); // Mise à LOW de la broche TRIG
  pinMode(ECHO, INPUT); // Configuration de la broche ECHO en entrée
  pinMode(buzz, OUTPUT); // Configuration de la broche buzz en sortie
  lcd.init(); // Initialisation de l'écran LCD
  lcd.clear(); // Effacement de l'écran LCD
  lcd.backlight(); // Activation du rétro-éclairage de l'écran LCD
  temp.begin(); } // Initialisation du capteur DHT11
void loop() {//Cette fonction est exécutée en boucle indéfiniment jusqu'à ce que la carte Arduino soit éteinte ou réinitialisée. C'est dans cette fonction que la plupart des instructions du programme sont écrites. Elle permet de faire fonctionner les différents composants de la carte Arduino et de traiter les données obtenues à partir des capteurs ou autres sources.
  digitalWrite(TRIG, HIGH); // Envoi d'une impulsion HIGH de 10 microsecondes sur la broche TRIG
  delayMicroseconds(10); // Attente de 10 microsecondes
```

```
  digitalWrite(TRIG, LOW); // Mise à LOW de la broche TRIG
  long mesure = pulseIn(ECHO, HIGH, MEASURE); // Mesure de la durée de l'impulsion sur la broche ECHO et stockage dans la variable mesure
  float distance = mesure / 2.0 * SOUND; // Calcul de la distance en utilisant la durée de l'impulsion et la vitesse du son
  lcd.setCursor(0, 0); // Positionnement du curseur de l'écran LCD en haut à gauche
  lcd.print(F("Distance:")); // Affichage du texte "Distance:"
  lcd.print(distance / 10.0); // Affichage de la distance en centimètres
  lcd.print(F("cm ")); // Affichage de l'unité (cm)
  delay(500); // Attente de 500 millisecondes
  if ((distance / 10.0) <= 5 && (distance / 10.0) >= 1) { //Si la distance est comprise entre 1 et 5 cm
tone(buzz, 1000); // Émission d'un signal sonore de 1000 Hz
delay(500); // Attente de 500 millisecondes
noTone(buzz); // Arrêt du signal sonore
} else if ((distance / 10.0) == 0) {  // Si la distance est nulle (capteur trop proche de l'objet)
 tone(buzz, 1000); }  // Émission d'un signal sonore de 1000 Hz
else { // Si la distance est supérieure à 5 cm
noTone(buzz); }  // Arrêt du signal sonore
lcd.setCursor(0, 1); // Positionnement du curseur de l'écran LCD sur la deuxième ligne
lcd.print(F("Temperature:")); // Affichage du texte "Temperature:"
lcd.print(String(temp.readTemperature() - 0.7)); // Affichage de la température en Celsius, avec une correction de -0,7 degré pour compenser une erreur connue du capteur DHT11
lcd.print(F("°C")); }  // Affichage de l'unité (degré Celsius)
```

Capture photo :





Conclusion :

Ce mini projet ma permis de mieux comprendre le fonctionnement de la carte Arduino Uno et des différents composants électroniques. J'ai également appris à utiliser différentes bibliothèques Arduino pour simplifier la programmation.