

NOM :
Prénom :
DATE :

TP Arduino 01
Introduction à l'arduino



Introduction :

C'est quoi l'Arduino ?

Arduino est une plateforme open source de développement matériel et logiciel qui permet aux développeurs de créer des projets interactifs. Il est basé sur un microcontrôleur, qui est un petit ordinateur intégré sur une seule puce. Les développeurs peuvent utiliser le langage de programmation C++ pour contrôler les entrées et les sorties matérielles de leurs projets, comme les boutons, les capteurs, les LED et les moteurs.

A quoi sert l'Arduino ?

L'Arduino est utilisé pour créer des projets interactifs tels que des robots, des systèmes de contrôle d'éclairage, des capteurs environnementaux, des jeux électroniques, des systèmes de contrôle de température, et beaucoup plus. Il est particulièrement populaire pour les projets de démarrage, car il est facile à utiliser, peu coûteux et flexible.

Les différentes cartes Arduino

Il existe plusieurs modèles différents de cartes Arduino, chacun avec ses propres caractéristiques et utilisations. Les plus courants sont :

- Arduino Nano : une carte compacte qui est idéale pour les projets portables ou ceux qui nécessitent peu d'espace.
- Arduino Uno : la carte de base d'Arduino, elle est facile à utiliser et est souvent utilisée pour les projets débutants.
- Arduino Mega : une carte plus grande qui a beaucoup plus de mémoire et de ports d'entrée / sortie que les autres cartes. Elle est idéale pour les projets plus complexes.

Comment fonctionne l'Arduino ?

L'Arduino fonctionne en exécutant du code écrit dans le langage de programmation C++. Les développeurs écrivent le code dans un environnement de développement intégré (IDE) Arduino, puis téléversent ce code sur le microcontrôleur. Le code définit les entrées et les sorties que le projet utilisera, ainsi que les réactions aux entrées telles que les boutons ou les capteurs.

Comment configurer l'Arduino

Pour configurer l'Arduino, vous devez d'abord télécharger et installer le logiciel IDE Arduino sur votre ordinateur. Ensuite, branchez votre carte Arduino à l'ordinateur à l'aide d'un câble USB. Dans l'IDE, vous pouvez choisir le type de carte que vous utilisez et sélectionner le port série correspondant pour communiquer avec votre carte. Une fois que vous avez écrit votre code, vous pouvez le téléverser sur la carte en utilisant le bouton "Téléverser sur la carte" dans l'IDE Arduino. Une fois que votre code est téléversé sur la carte, il commencera à s'exécuter immédiatement. Vous pouvez utiliser les entrées et les sorties de votre projet pour obtenir des réactions en temps réel. Par exemple, si vous avez connecté un bouton à l'entrée, le projet peut changer d'état lorsque le bouton est pressé. De même, si vous avez connecté une LED à la sortie, le projet peut allumer ou éteindre la LED en fonction de l'état du bouton.

En conclusion, l'Arduino est un outil puissant pour les développeurs qui souhaitent créer des projets interactifs. Avec une variété de cartes disponibles et un environnement de développement facile à utiliser, il est accessible à une large variété de développeurs, allant des débutants aux utilisateurs expérimentés. Alors, pourquoi ne pas explorer l'univers de l'Arduino et voir ce que vous pouvez créer aujourd'hui ?

Question :

1 - Qu'est-ce qu'Arduino ?

Arduino est une plateforme de développement de matériel et de logiciel qui permet de créer des projets interactifs en utilisant des microcontrôleurs.

2 - Qu'est-ce qu'un microcontrôleur ?

Un microcontrôleur est un petit ordinateur intégré sur une seule puce qui peut être utilisé pour contrôler des entrées et des sorties matérielles.

3 - Pour quoi utilisons-nous les entrées et les sorties sur Arduino ?

Nous utilisons les entrées et les sorties sur Arduino pour faire réagir nos projets aux entrées telles que des boutons ou des capteurs, et pour produire des sorties telles que des LED ou des moteurs.

4 - Quelle est la différence entre une entrée numérique et une entrée analogique sur Arduino ?

Une entrée numérique peut uniquement être "on" ou "off", tandis qu'une entrée analogique peut prendre une plage de valeurs différentes.

5 - Quelle est la différence entre une sortie numérique et une sortie analogique sur Arduino ?

Une sortie numérique peut uniquement être "on" ou "off", tandis qu'une sortie analogique peut produire une plage de niveaux de tension différents.

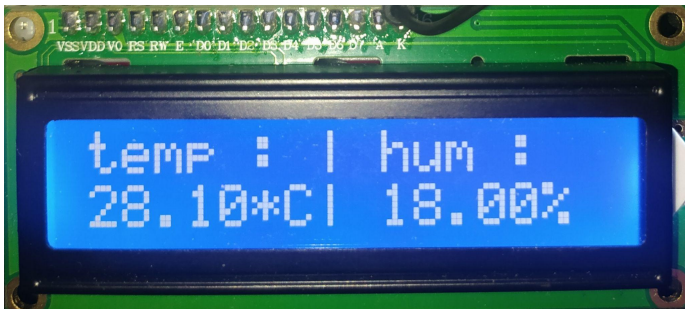
6 - Quel langage est utilisé pour programmer un arduino

L'Arduino utilise un langage de programmation appelé C++ pour faire fonctionner les projets. Les développeurs peuvent écrire du code dans un programme appelé IDE Arduino, puis téléverser ce code sur le microcontrôleur pour le faire exécuter.

TP Lecture et affichage de donnée :

I - Objectif :

En utilisant un capteur DHT11, un afficheur LCD 2x16 et un arduino uno obtenir le résultat suivant :



II - câblage :

Rappelle des bases :

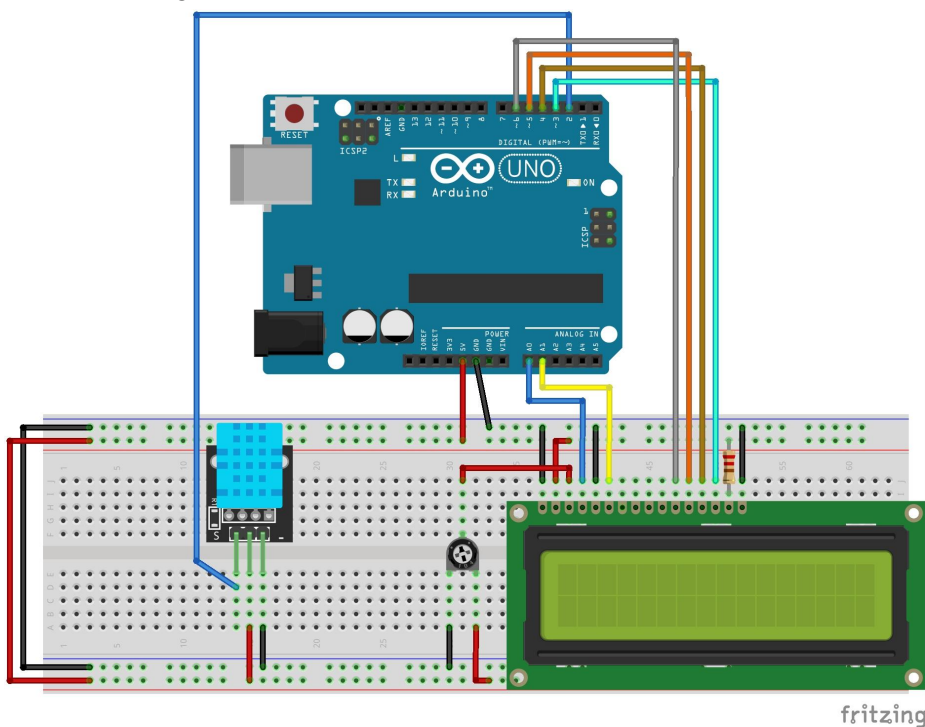
fil rouge = VCC

fil noir = GND / masse

résistance = 220 ohm

potentiomètre = 10k ohm

Faite le câblage suivant :



III - Programmation :

En utilisant internet écrivez un code qui va lire les données que le capteur reçoit et les afficher sur l'afficheur lcd comme montré dans objectif.

```
//Library pour le projet
#include <DHT.h>
#include <LiquidCrystal.h>

//Pins de l'écran LCD 2*16
#define rs A0
#define en A1
#define d4 6
#define d5 5
#define d6 4
#define d7 3
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
#define data 2
DHT dht(2, DHT11);

void setup() {
  dht.begin();          // Initialisation du capteur
  Serial.begin(9600);    // Initialisation du port série et son port
}

//Création de la boucle
void loop() {
  float h = dht.readHumidity();    //création d'une variable flottante qui stocke la valeur de l'humidité
  float t = dht.readTemperature(); //création d'une variable flottante qui stocke la valeur de la température

  //Affichage sur le port série
  Serial.print("Humidity: ");
  Serial.println(h);
  Serial.print("Temperature: ");
  Serial.println(t);

  //Affichage sur l'écran LCD
  lcd.begin(16, 2);          //Initialisation de l'écran
  lcd.setCursor(0, 0);      //Position du curseur (colonne 0, ligne 0)
  lcd.print("temp : | hum :");
  lcd.setCursor(0, 1);      //Position du curseur (colonne 0, ligne 1)
  lcd.print(t); lcd.print("°C| ");
  lcd.print(h); lcd.print("%");

  delay(1000);              //Attendre 1000 ms soit 1 seconde avant de refaire la boucle
}
```

IV - Pouvez-vous reformuler les résultats obtenus et expliquer les raisons?

Rédiger un petit paragraphe pour expliquer pourquoi vous avez réussi ou échoué.