

Fabrication d'une station météo

- L'objet du tutoriel : **Assembler une station météo à partir des composants**
- Difficultés : **Facile**
- Durée : **4 heures**
- Catégorie : **Energie renouvelable**
- Coût : **14 554 FCFA**
- Nombre de personnes : **5**

Sommaire

- Matériaux / outils
- Étapes de la réalisation
- Commentaires

Matériaux:

Capteur d'humidité de l'air et de température (référence : DHT11) **1 510 FCFA.**

Capteur d'humidité du sol (référence : FC-28) **4 489 FCFA.**

Capteur de luminosité (référence : KY-018) **548 FCFA.**

Ecran Lcd (référence : 11C/12C) **1 542 FCFA.**

Shield (Référence : Base shield V2.0) **3 542 FCFA.**

Câbles de raccordement **439 FCFA** (fils de récupération).

Carte arduino (Uno) **2 484 FCFA.**

Boîtier de la station météo (imprimé)(**paramètre**).

Condensateur (référence : Électrolytique 16v 100uF),(récupération).

Outils :

Fer à souder

Pistolet à colle

Imprimante 3D

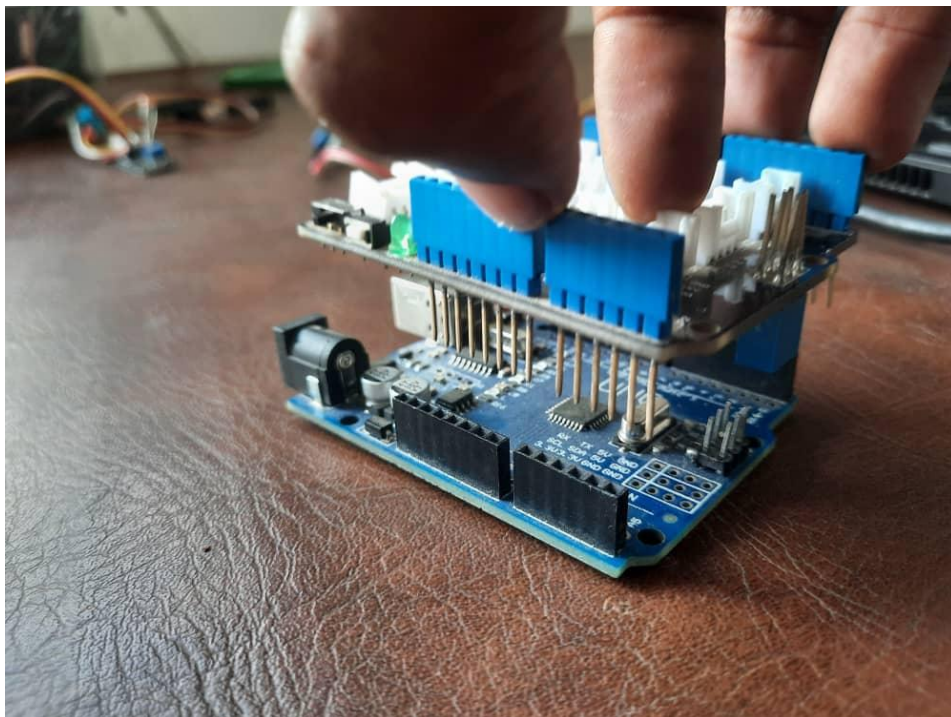
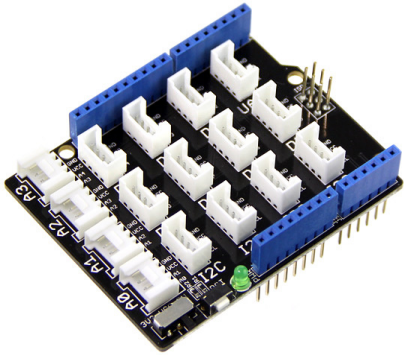
Etape 4 : Partie assemblage station météo

Une station météo est un appareil qui permet de connaître les caractéristiques de l'atmosphère de la pièce dans laquelle elle est placée (température, humidité, quantité de lumière etc...), ou éventuellement d'autres caractéristiques dépendamment de ce que l'on veut mesurer (l'humidité de l'aire, du sol dans notre cas).

Montage :

Monter le shield sur la carte arduino.

NB : L'utilisation du shield facilite la connexion des différents éléments sur la carte.

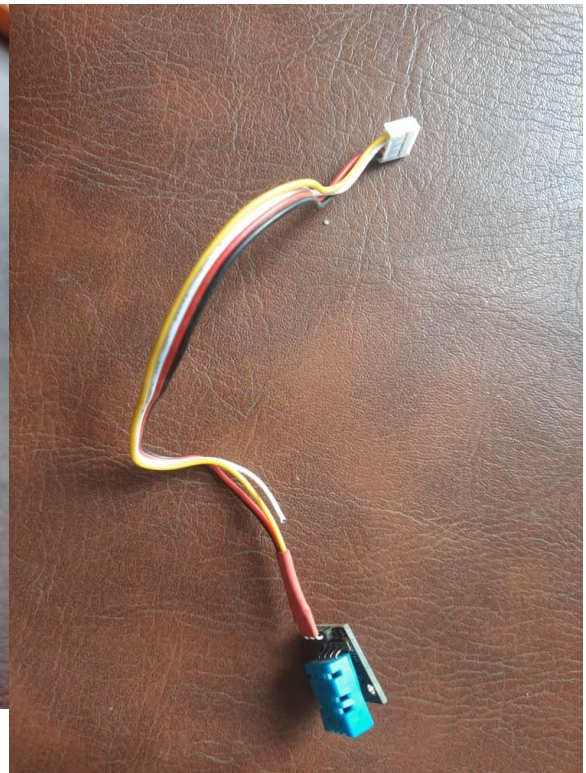
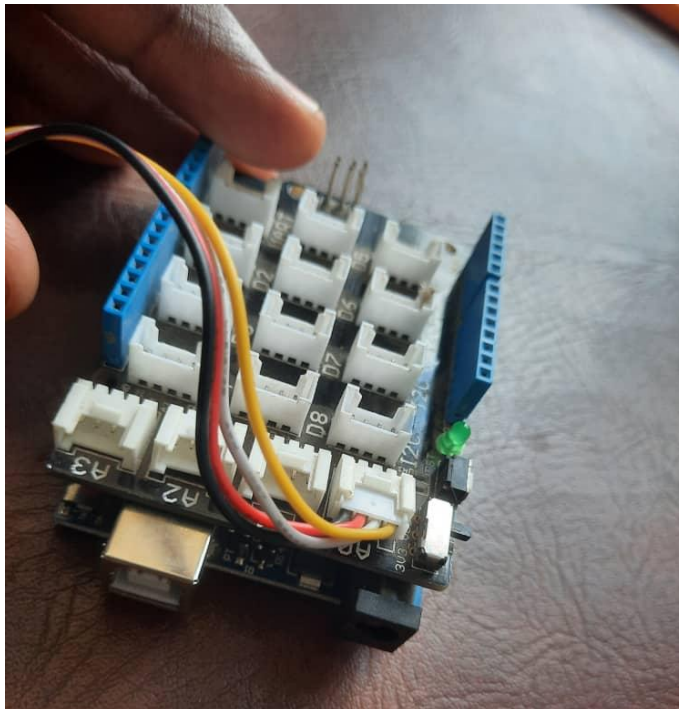


CAPTEUR DE TEMPÉRATURE ET HUMIDITÉ:

Comme son nom l'indique, il sert à mesurer l'humidité et la température d'un milieu. Sur l'image ci-dessous, le DHT11 (capteur de température et d'humidité) est relié à l'entrée analogique de la carte Arduino donc sur les ports A0 du shield.

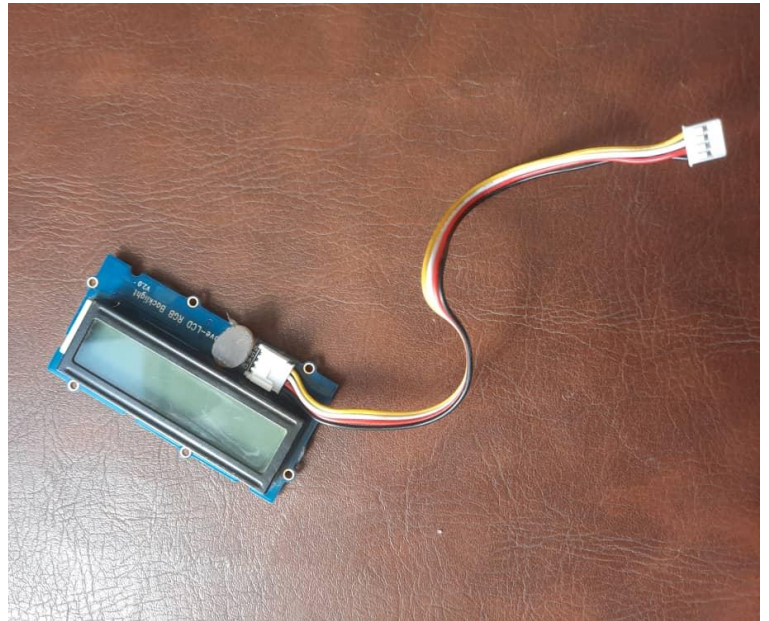
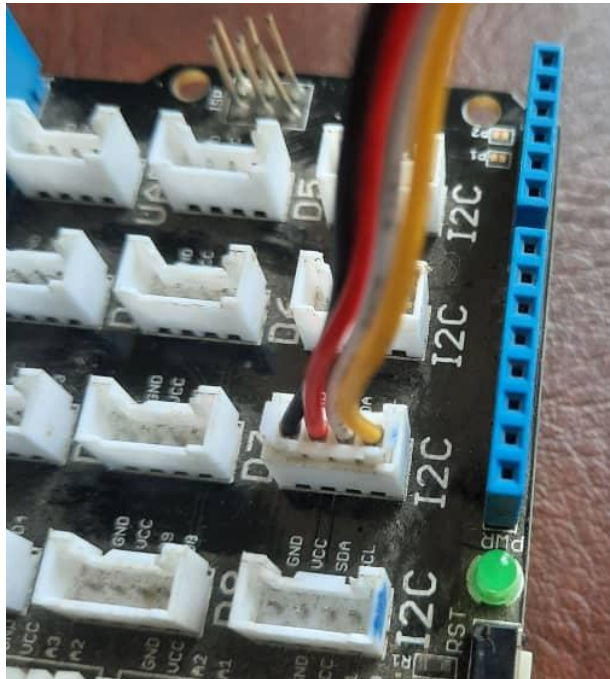
Pour les casbles, le jaune correspond à A0, le blanc correspond à A1, le rouge à Vcc et le noir à Gnd.

Etant donné que pour la connexion de ce capteur, le A1 n'est pas utilisé, il faut le déconnecter (fil blanc) ou à défaut le couper comme c'est le cas ici.



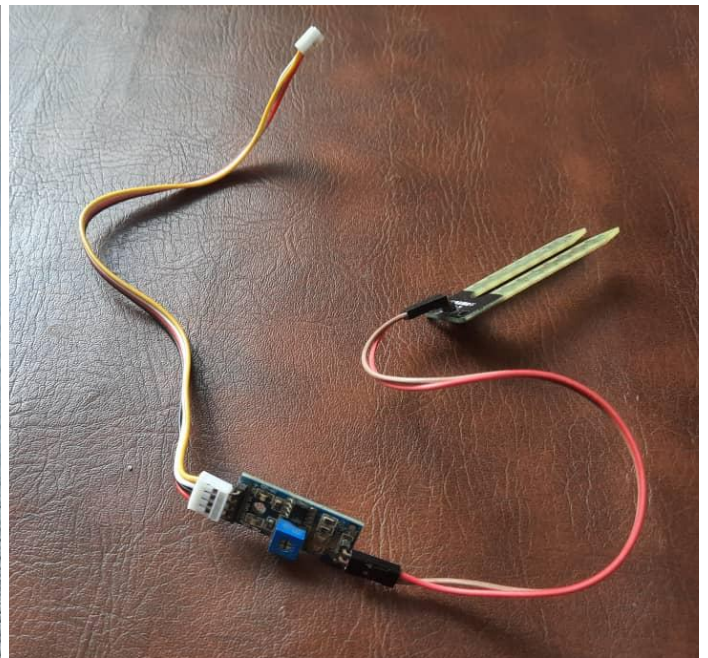
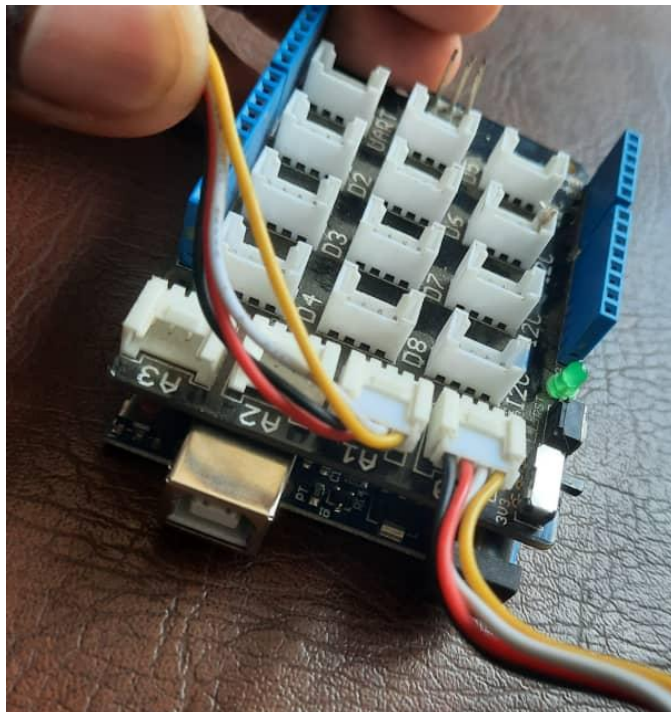
ECRAN LCD :

L'écran est utilisé pour afficher les valeurs mesurées par les capteurs.
Pour l'écran LCD le branchement se fait sur les I2C du shield.



CAPTEUR DE L'HUMIDITÉ DU SOL :

Le capteur de l'humidité du sol est relié au port A1 du shield.



CAPTEUR DE LUMINOSITÉ :

Pour le capteur de luminosité relier sur le port A2 du shield.

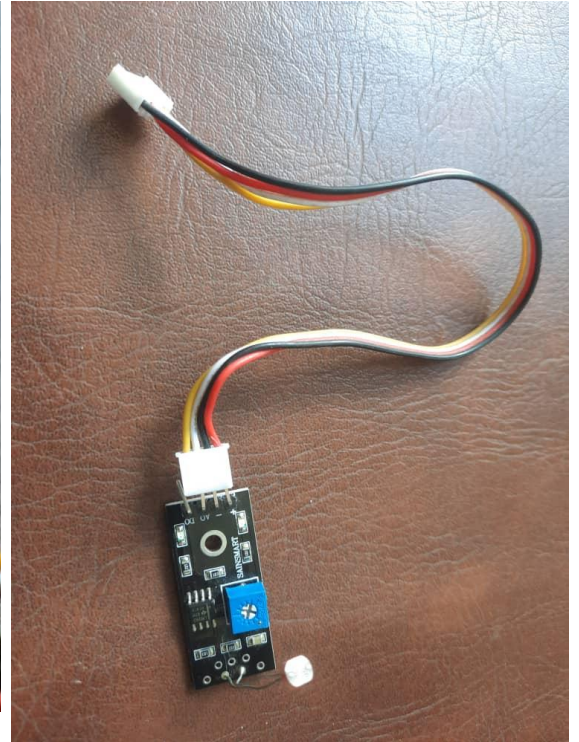
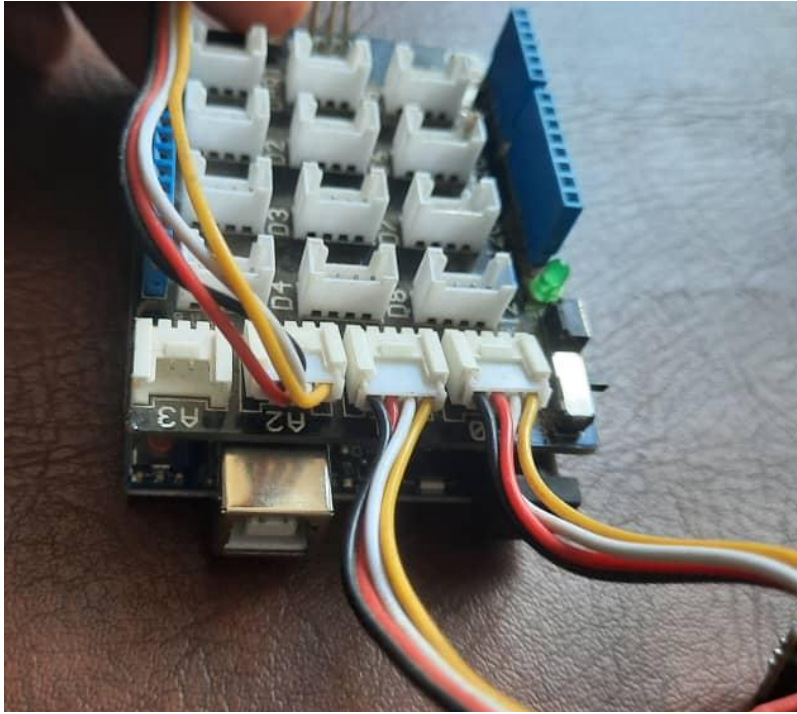
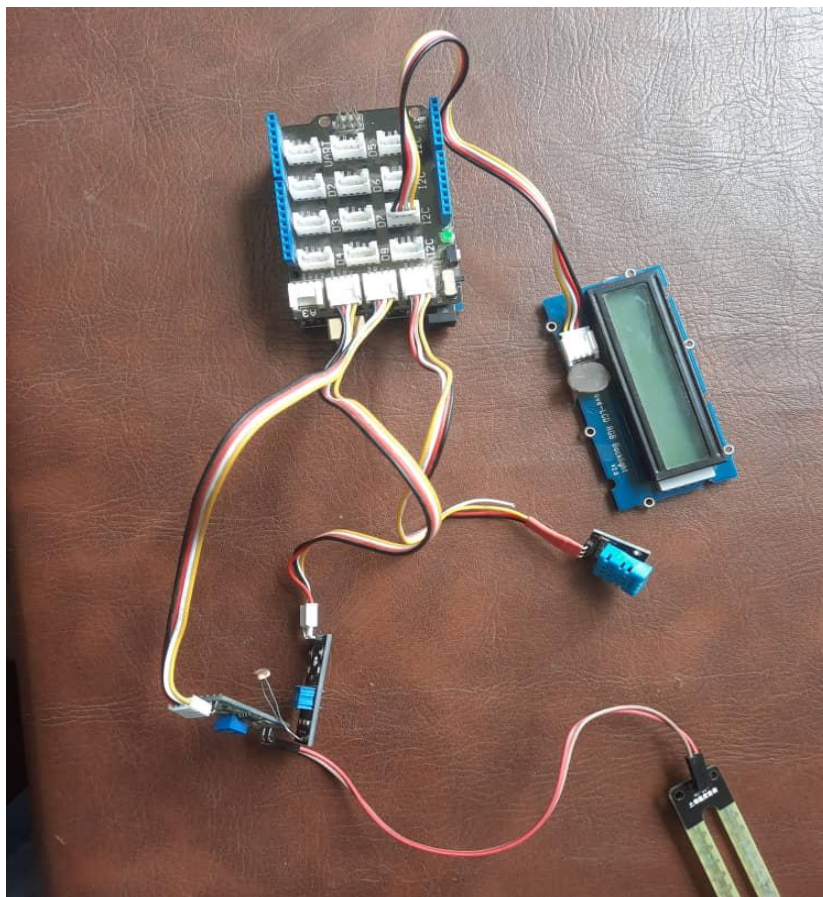


Image de l'ensembles des éléments



Etape 5 : Assemblage des panneaux solaires et de la station météo

Dans cette partie, il faut alimenter la station météo via le régulateur (+) batterie.

Pour cela,

Connecter la borne (+) de l'alimentation de la station météo à la borne 2 du régulateur

Et relier la borne (-) de l'alimentation de la station météo à la borne 3 du régulateur.

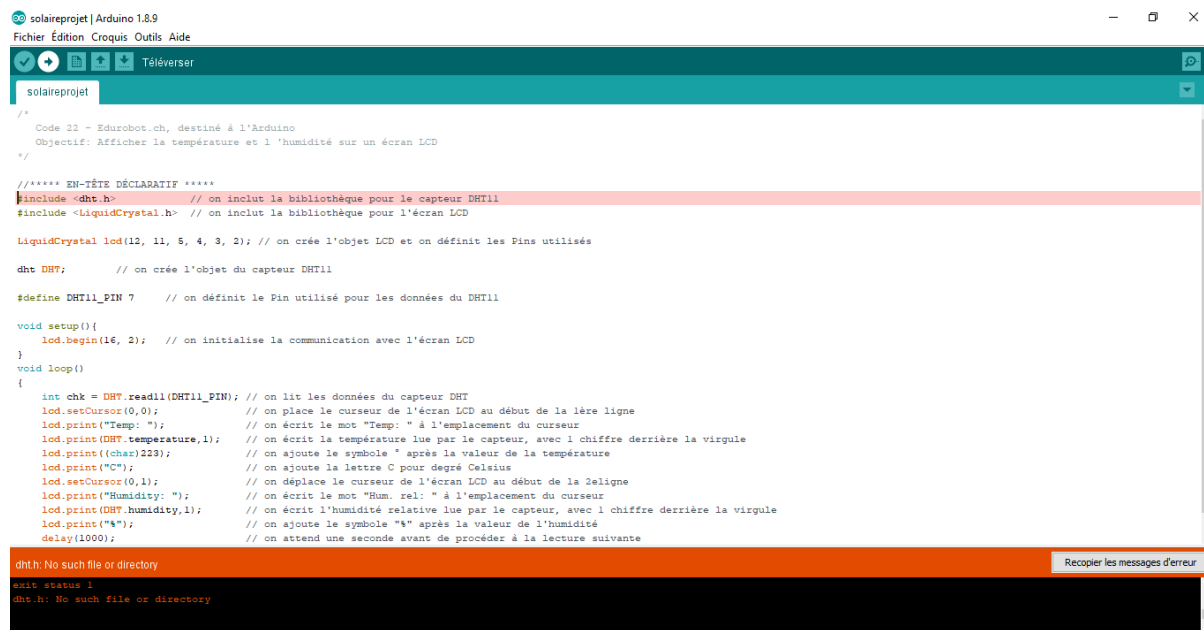
Etape 6 : Le Code

Dans cette partie nous allons vous montrer les étapes à suivre pour mettre en place le code de la station météo.

Tout d'abord il faut faire des recherches pour trouver un code qui répond à votre besoins nous avons trouvé celui ci :

source :

<https://arduino.developpez.com/tutoriels/arduino-a-l-ecole/?page=projet-11-construire-une-station-meteo#LXIX-B>



```
solairprojet | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Téléverser
solairprojet
/*
  Code 22 - Edurobor.ch, destiné à l'Arduino
  Objectif: Afficher la température et l'humidité sur un écran LCD
*/

//***** EN-TÊTE DÉCLARATIF *****/
#include <dht.h> // on inclut la bibliothèque pour le capteur DHT11
#include <LiquidCrystal.h> // on inclut la bibliothèque pour l'écran LCD

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // on crée l'objet LCD et on définit les Pins utilisés

dht DHT; // on crée l'objet du capteur DHT11

#define DHT11_PIN 7 // on définit le Pin utilisé pour les données du DHT11

void setup() {
  lcd.begin(16, 2); // on initialise la communication avec l'écran LCD
}

void loop() {
  int chk = DHT.read11(DHT11_PIN); // on lit les données du capteur DHT
  lcd.setCursor(0,0); // on place le curseur de l'écran LCD au début de la 1ère ligne
  lcd.print("Temp: "); // on écrit le mot "Temp: " à l'emplacement du curseur
  lcd.print(DHT.temperature,1); // on écrit la température lue par le capteur, avec 1 chiffre derrière la virgule
  lcd.print((char)223); // on ajoute le symbole "°" après la valeur de la température
  lcd.print("C"); // on ajoute la lettre C pour degré Celsius
  lcd.setCursor(0,1); // on déplace le curseur de l'écran LCD au début de la 2ème ligne
  lcd.print("Humidity: "); // on écrit le mot "Hum. rel: " à l'emplacement du curseur
  lcd.print(DHT.humidity,1); // on écrit l'humidité relative lue par le capteur, avec 1 chiffre derrière la virgule
  lcd.print("%"); // on ajoute le symbole "%" après la valeur de l'humidité
  delay(1000); // on attend une seconde avant de procéder à la lecture suivante

dht.h: No such file or directory
exit status 1
dht.h: No such file or directory
Recopier les messages d'erreur
```

Par contre dans ce code nous avons un problème avec la bibliothèque dht11. Rechercher et Télécharger la bibliothèque, nous avons choisi celui-ci

<https://github.com/adidax/dht11>

Après avoir ajouter la bibliothèque **#include <dht11.h>**

Il faut supprimer la première importation et changer les lignes de codes.

```
dht DHT;          // on crée l'objet du capteur DHT11

#define DHT11_PIN 7    // on définit le Pin utilisé pour les données du DHT11

void loop()
{
    int chk = DHT.read11(DHT11_PIN); // on lit les données du capteur DHT
```

Par ceux là

```
dht11 DHT;
#define DHT11_PIN A0

void loop() {

    int chk =DHT.read(DHT11_PIN);    // on lit les données du capteur DHT
```

Pour LCD ECRAN :

Les écrans **LCD compatibles Grove** utilisent le **protocole I2C** pour communiquer avec la carte Arduino.

L'écran doit donc être relié à une des **broches I2C** de la **base Grove shield**.

source

https://icn.lycee-valin.fr/dokuwiki/doku.php?id=arduino:plaque_grove_du_lycee_valin:comment_utiliser_un_ecran_lcd_grove

Ajouter la bibliothèque dans : Croquis → Inclure une bibliothèque → Gérer les bibliothèques → grove_rgb_lcd.h

Nous avons enlevé ceci

```
..
-----
#include <LiquidCrystal.h> // on inclut la bibliothèque pour l'écran LCD

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // on crée l'objet LCD et on définit les Pins utilisés
```

Et le remplacer par cela

```
#include <Wire.h> // inclure la bibliothèque permettant la communication par le protocole I2C
#include "rgb_lcd.h" // inclure la bibliothèque pour l'écran LCD

rgb_lcd lcd; // création d'un objet lcd
```

Il faut remplacer le void setup par ce bout de code

```
void setup() {
  Serial.begin(9600);
  //pinMode(9,OUTPUT); Réglez la broche 9 comme broche de sortie, pour envoyer le signal au relay.
  // pinMode(8,INPUT); Réglez la broche 8 comme broche d'entrée pour recevoir les données du capteur d'humidité.
  lcd.begin(16, 2); // configurer le nombre de colonnes et de lignes de l'écran LCD :
  pinMode(A2, INPUT);
  analogWrite(A2, LOW); //Regler la sortie A2 pour envoyer les données du capteur de l'humidité du sol
  pinMode(A3, INPUT);
  analogWrite(A3, LOW); //Regler la sortie A3 pour envoyer les données du capteur de luminosité
}
void loop() {
```


De même que le void loop par ceci :

```
void loop() {  
  int sol = analogRead(A2);  
  int light3 = analogRead(A4);  
  int chk =DHT.read(DHT11_PIN); // on lit les données du capteur DHT  
  lcd.setCursor(0,0); // on place le curseur de l'écran LCD au début de la 1ère ligne  
  lcd.print("T: "); // on écrit le mot "Temp: " à l'emplacement du curseur  
  lcd.print(DHT.temperature); // on écrit la température lue par le capteur,  
  lcd.print((char)223); // on ajoute le symbole ° après la valeur de la température  
  lcd.print("C"); // on ajoute la lettre C pour degré Celsius  
  lcd.setCursor(9,0); // on déplace le curseur de l'écran LCD au x=9 de la 1ère ligne  
  lcd.print("H: "); // on écrit le mot "Hum. rel: " à l'emplacement du curseur  
  lcd.print(DHT.humidity); // on écrit l'humidité relative lue par le capteur, avec 1 chiffre derrière la virgule  
  lcd.print("%"); // on ajoute le symbole "%" après la valeur de l'humidité  
  lcd.setCursor(0,1); // on déplace le curseur de l'écran LCD au début de la 2ème ligne  
  lcd.print("S: "); // on écrit le mot "Sol. rel: " à l'emplacement du curseur  
  lcd.print(sol); // on écrit l'humidité relative lue par le capteur, avec 1 chiffre derrière la virgule  
  lcd.setCursor(7,1); // on déplace le curseur de l'écran LCD au début de la 2ème ligne et à x=7  
  lcd.print("L: "); // on écrit la lettre "L. rel: " à l'emplacement du curseur  
  lcd.print(light3); // on écrit la quantité de lumière relative lue par le capteur, avec 1 chiffre derrière la virgule  
  delay(1000); // on attend une seconde avant de procéder à la lecture suivante  
}
```

lcd.setCursor(0,0); // Il faut placer le curseur de l'écran LCD au début de la 1ère ligne

lcd.setCursor(9,0); // Il faut placer le curseur de l'écran LCD à x=9 de la 1ère ligne

lcd.setCursor(0,1); // Il faut placer le curseur de l'écran LCD au début de la 2ème

ligne lcd.setCursor(7,1); // Il faut placer le curseur de l'écran LCD à la 2ème ligne et à x=7

Le code en Intégralité

code source :

```
#include <dht11.h>
dht11 DHT;
#define DHT11_PIN A0

#include <Wire.h> // inclure la bibliothèque permettant la communication par le
protocole I2C
#include "rgb_lcd.h" // inclure la bibliothèque pour l'écran LCD

rgb_lcd lcd; // création d'un objet lcd
int val; // Cette variable stocke la valeur reçue du capteur d'humidité du sol.
void setup() {
  Serial.begin(9600);
  //pinMode(9,OUTPUT); Réglez la broche 9 comme broche de sortie, pour envoyer le
signal au relay.
  // pinMode(8,INPUT); Réglez la broche 8 comme broche d'entrée pour recevoir les
données du capteur d'humidité.
  lcd.begin(16, 2); // configurer le nombre de colonnes et de lignes de l'écran LCD :
  pinMode(A2, INPUT);
  analogWrite(A2, LOW); //Regler la sortie A2 pour envoyer les données du capteur de
l'humidité du sol
  pinMode(A3, INPUT);
  analogWrite(A3, LOW); //Regler la sortie A3 pour envoyer les données du capteur de
luminosité
}
void loop() {
```

```

int sol = analogRead(A2);
int light3 = analogRead(A4);
int chk = DHT.read(DHT11_PIN); // on lit les données du capteur DHT
lcd.setCursor(0, 0); // on place le curseur de l'écran LCD au début de la 1ère
ligne
lcd.print("T: "); // on écrit le mot "Temp: " à l'emplacement du curseur
lcd.print(DHT.temperature); // on écrit la température lue par le capteur,
lcd.print((char)223); // on ajoute le symbole ° après la valeur de la température
lcd.print("C"); // on ajoute la lettre C pour degré Celsius
lcd.setCursor(9, 0); // on déplace le curseur de l'écran LCD au x=9 de la 1eligne
lcd.print("H: "); // on écrit le mot "Hum. rel: " à l'emplacement du curseur
lcd.print(DHT.humidity); // on écrit l'humidité relative lue par le capteur, avec 1
chiffre derrière la virgule
lcd.print("%"); // on ajoute le symbole "%" après la valeur de l'humidité
lcd.setCursor(0, 1); // on déplace le curseur de l'écran LCD au début de la
2eligne
lcd.print("S: "); // on écrit le mot "Sol. rel: " à l'emplacement du curseur
lcd.print(sol); // on écrit l'humidité relative lue par le capteur, avec 1 chiffre
derrière la virgule
lcd.setCursor(7, 1); // on déplace le curseur de l'écran LCD au début de la
2eligne et a x=7
lcd.print("L: "); // on écrit la lettre "L. rel: " à l'emplacement du curseur
lcd.print(light3); // on écrit la quantité de lumiere relative lue par le capteur, avec
1 chiffre derrière la virgule
delay(1000); // on attend une seconde avant de procéder à la lecture
suivante
}

```

[Senfablab](#) - Dakar.

