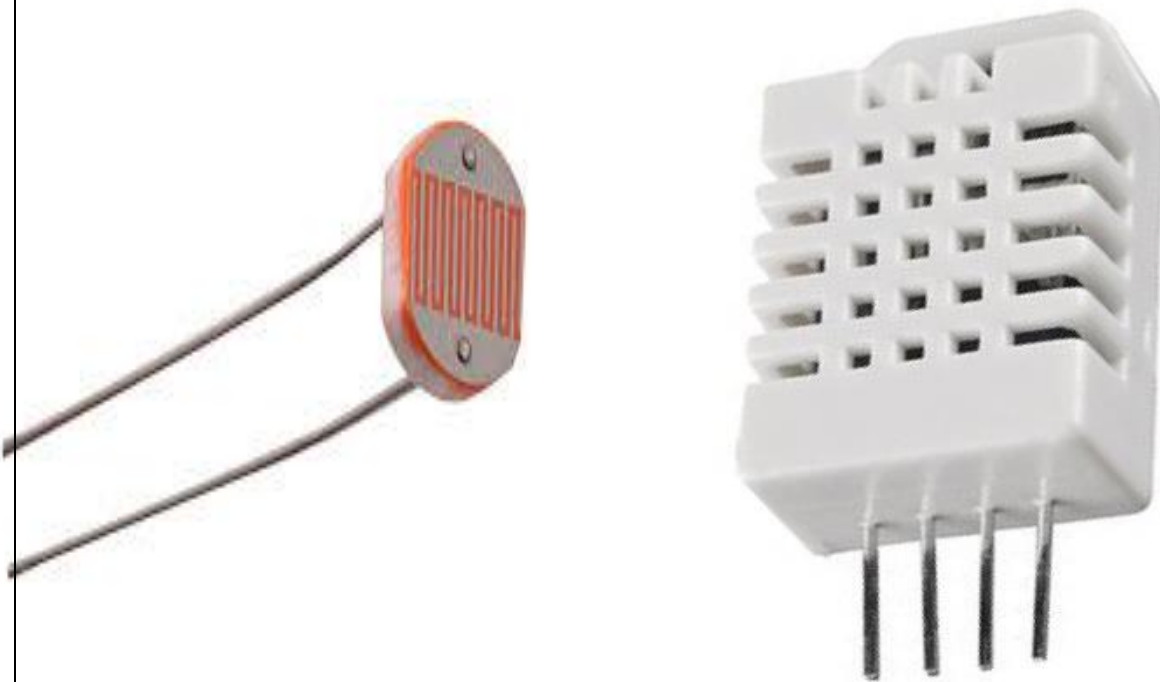


Etude de 2 capteurs:LDR et DHT22



Réalisé par : -BARRAD Marouane

-AIT BOURHIM Abderrahim

Encadré par :Mr.M.GHRBAOUI

Année Universitaire :2022/2023

Table de matières :

Introduction générale.....	5
Chapitre1 : Etude d'un capteur de luminosité.....	6
I . Généralités sur les photorésistances.....	7
1. Définition	7
2. L'éclairement.....	7
3. Principe de fonctionnement.....	7-8
4. Avantages.....	8
5. Inconvénients.....	8
II . Partie étude.....	9
1. Composants du montage.....	9-12
2. Principe de fonctionnement.....	12
3. Programme.....	13
4. Étalonnage.....	13/14
5. Fidélité et justesse.....	15
CHAPITRE 2 : ETUDIER LE CAPTEUR DHT22.	
I. DHT22 :.....	17
1. Présentation du capteur	17
2. Brochage.....	17
3. Protocole de communication	18
4. Traduction des mesures.....	19
II. Interfaces arduino/dht22.....	20
1. Programme arduino.....	20-21
2. Simulation ISIS.....	22
III. Réalisation.....	23
1. Description.....	23
2. Courbe etalonnage.....	24
3. Application.....	24
CONCLUSION.....	25

Liste des figures :

Figure1 : Composants d'un arduino UNO.

Figure2 :LCD 1602A

Figure3 : Module I2C LCD 1602A

Figure4:Les résistances utilisées dans le montage

Figure5 :Shéma ISIS

Figure6 :Simulation THINKERCAD

Figure7:Programme matlab d'étalonnage.

Figure8:Courbe d'étalonnage

Figure 9 : Brochage de DHT22.

Figure 10 : bus de communication (protocole)

Figure 11: communication format spécifier

Figure 12 : simulation isis du DHT22.

Figure 13 : image réelle montre la réalisation

Figure 14 : la courbe d'étalonnage.

Introduction générale :

Dans de nombreux domaines (industrie, recherche scientifique, services loisirs...), on a besoin de contrôler des paramètres physiques (température, force, position, vitesse, luminosité...).

Le capteur est l'élément indispensable à la détection de ces grandeurs physiques.

Un capteur est un organe de prélèvement d'informations qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (souvent électrique).

Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande.

Dans ce travail on va étudier 2 capteurs ,un capteur de luminosité(LDR) et un capteur de température(DHT22) ainsi que leurs courbes d'étalonnage et leurs justesses et fidélités.

Chapitre1 :

Etude d'un capteur de

luminosité :

Photorésistance.

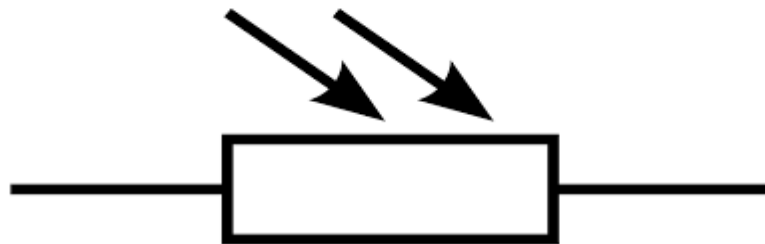
GENERALITES SUR LES PHOTORESISTANCES :

1.Définition de la LDR :

Comme son nom l'indique dans la langue de Shakespeare: LDR pour Light Dependent Resistor, la photorésistance est un dipôle dont la résistance varie en fonction de l'éclairement E qu'elle reçoit d'une source de lumière.

La partie sensible du capteur est une piste de sulfure de cadmium en forme de serpent : l'énergie lumineuse reçue déclenche une augmentation de porteurs de charges libres dans ce matériau, de sorte que sa résistance électrique évolue.

Son symbole normalisé dans un circuit est le suivant :



2.l'éclairement :

C'est une grandeur physique notée E , se mesurant à l'aide d'un luxmètre et qui permet de rendre compte de la luminosité plus ou moins forte d'une source lumineuse. Plus la source paraît intense, plus son éclairement E est élevé.

3.Principe de la photorésistance :

Un cristal de semi-conducteur à température basse contient peu d'électrons libres. La conductivité du cristal est très faible, proche de celle d'un isolant. Lorsque la température du cristal augmente de plus en plus d'électrons qui étaient immobilisés dans les liaisons covalentes s'échappent et peuvent participer à la conduction.

A température constante si le même cristal semi-conducteur est soumis à une radiation lumineuse, l'énergie apportée par les photons peut suffire à libérer certains électrons utilisés dans les liaisons covalentes entre atomes du cristal. Plus le flux lumineux sera intense, plus le nombre d'électrons disponibles pour assurer la conduction sera grand, ainsi la résistance de la LDR est inversement proportionnelle à la lumière reçue.

Une photorésistance est composée d'un semi-conducteur à haute résistivité. Si la lumière incidente est de fréquence suffisamment élevée (donc d'une longueur d'onde inférieure à la longueur d'onde seuil), elle transporte une énergie importante. Au-delà d'un certain niveau propre au matériau, les photons absorbés par le semi-conducteur donneront aux électrons liés assez d'énergie pour passer de la bande de valence à la bande de conduction. La compréhension de ce phénomène entre dans le cadre de la théorie des bandes. Les électrons libres et les trous d'électron ainsi produits abaissent la résistance du matériau.

Lorsque le photon incident est suffisamment énergétique, la production des paires électron-trou est d'autant plus importante que le flux lumineux est intense. La résistance évolue donc comme l'éclairement.

4. Avantages :

- Faible coût
- Larges gammes spectrales
- Facilité de mise en œuvre
- Rapport de transfert statique
- Sensibilité élevée

5. Inconvénients :

- Non linéarité de la réponse en fonction du flux.
- La vitesse de variation de R avec l'éclairement est faible et non symétrique.
- Sensibilité thermique.
- Refroidissement nécessaire dans certains cas (capteurs thermiques).
- Temps de réponse élevé((0,1 μ s à 100 ms).
- Bande passante limitée.
- Instabilité dans le temps (vieillessement dû aux échauffements).

PARTIE ETUDE :

1.Composants du montage :

Une carte arduino Uno :est un cerveau qui permet de rendre intelligent des systèmesélectroniques et d'animer des dispositifs mécaniques.

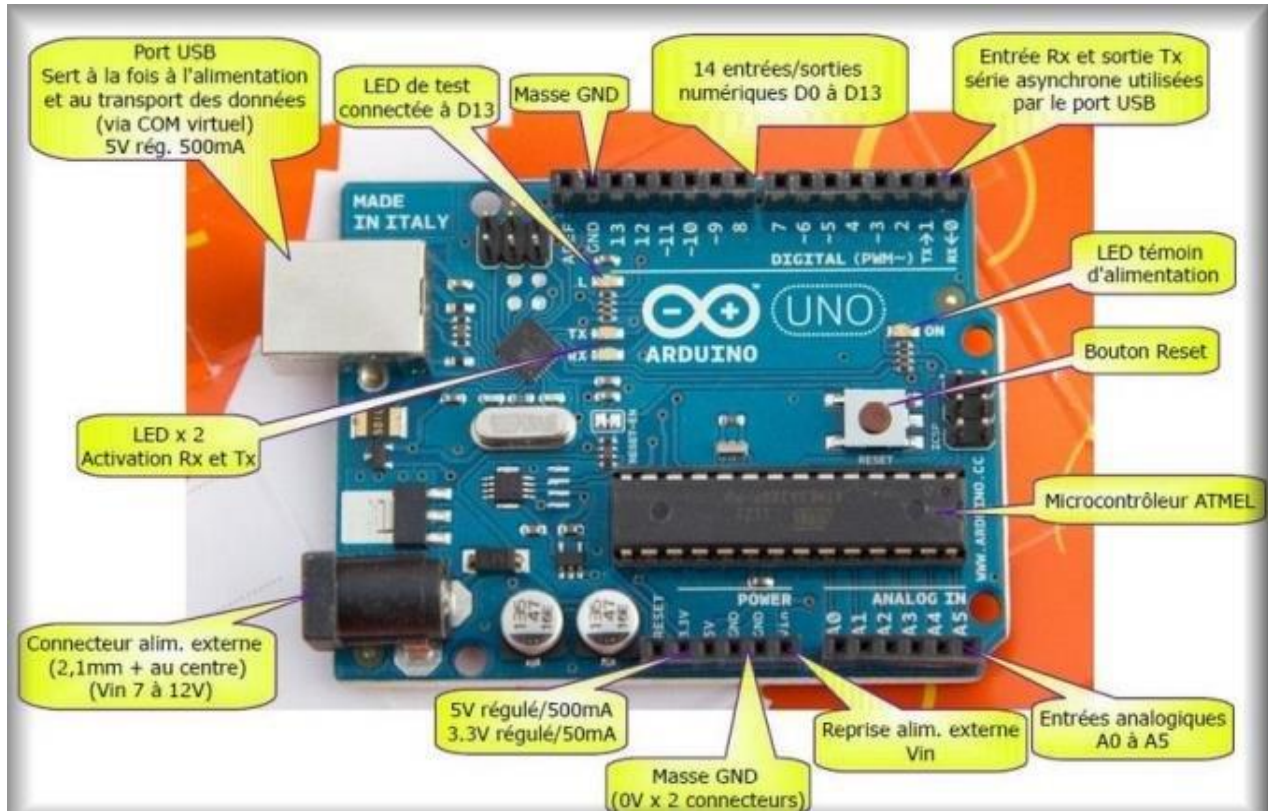


Figure1 :Composants d'un Arduino UNO

L'afficheur LCD I2C 160A : est un afficheur à cristaux liquides (LCD) qui utilise la communication en bus I2C pour se connecter à un microcontrôleur ou à une carte de développement. 1602 signifie que l'afficheur va pouvoir afficher 16 caractères sur chacune des 2 lignes.

Les afficheurs LCD I2C sont couramment utilisés dans les projets électroniques et de robotique pour afficher des données ou des messages. Ils sont généralement compacts et peu coûteux, et peuvent être facilement intégrés dans de nombreux projets.

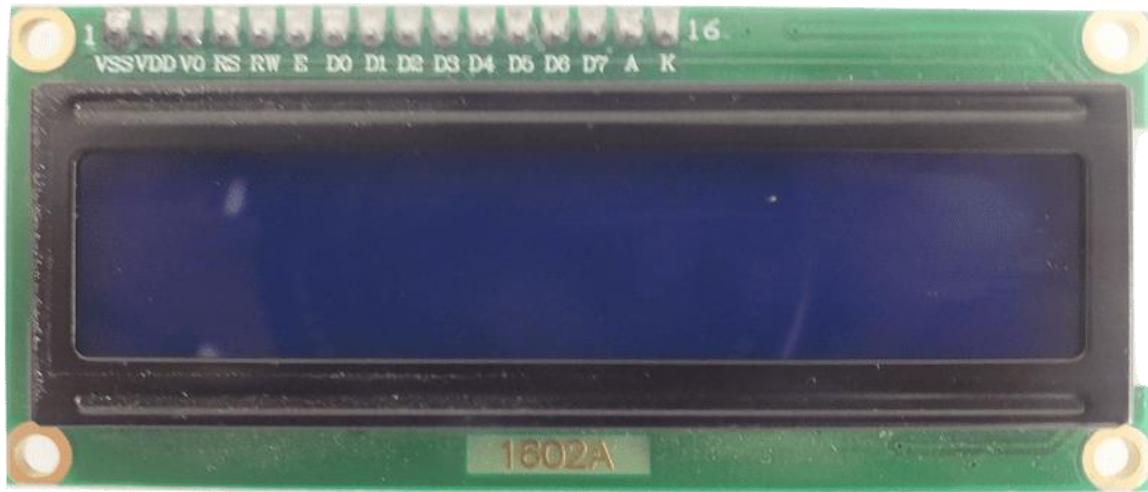


Figure2 :LCD 1602A

Il est fréquemment utilisé dans les applications à base microcontrôleur comme **arduino** et **micro:bit** .

Le module **I2C LCD 1602A** contient 4 broches :

- Signal de donnée : SDA
- Signal d'horloge : SCL
- Signal d'alimentation : +VCC (5V)
- Signal de masse : GND

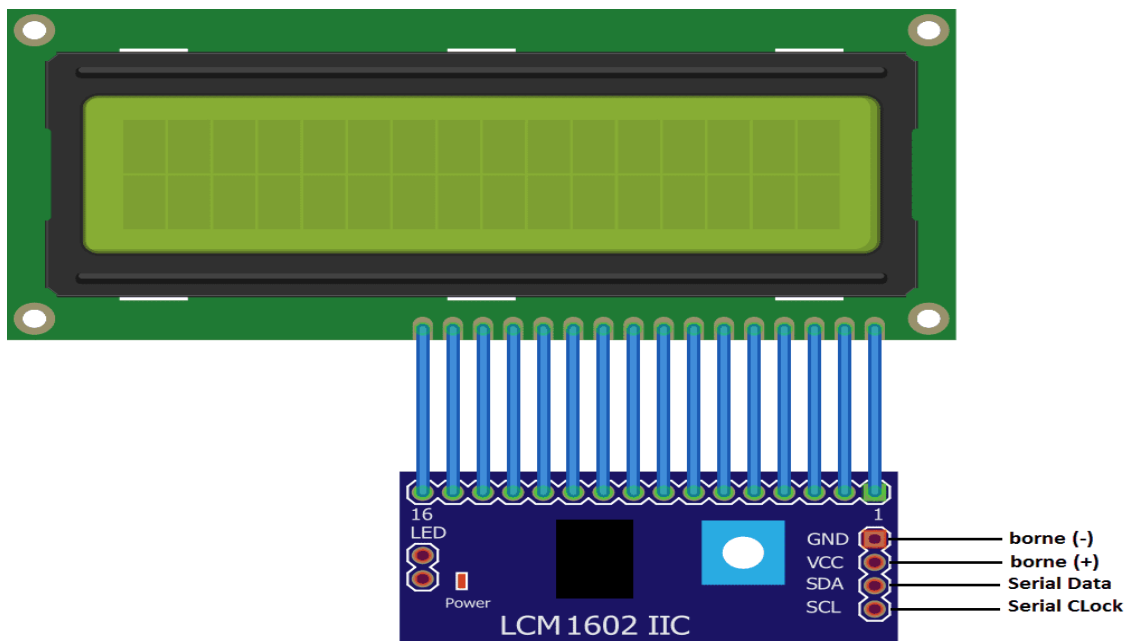


Figure3 :Module I2C LCD 1602A

Une LED et 2 résistances :

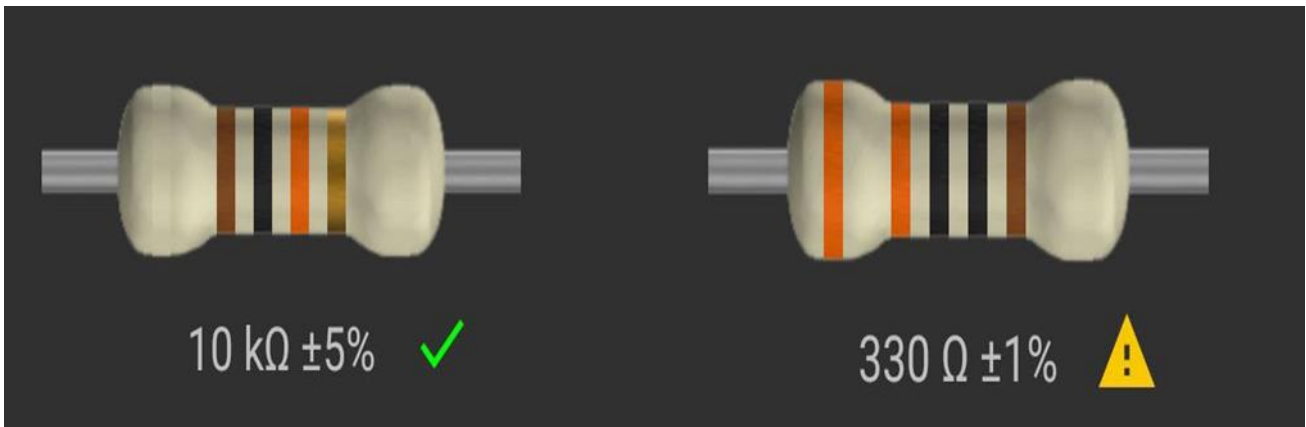
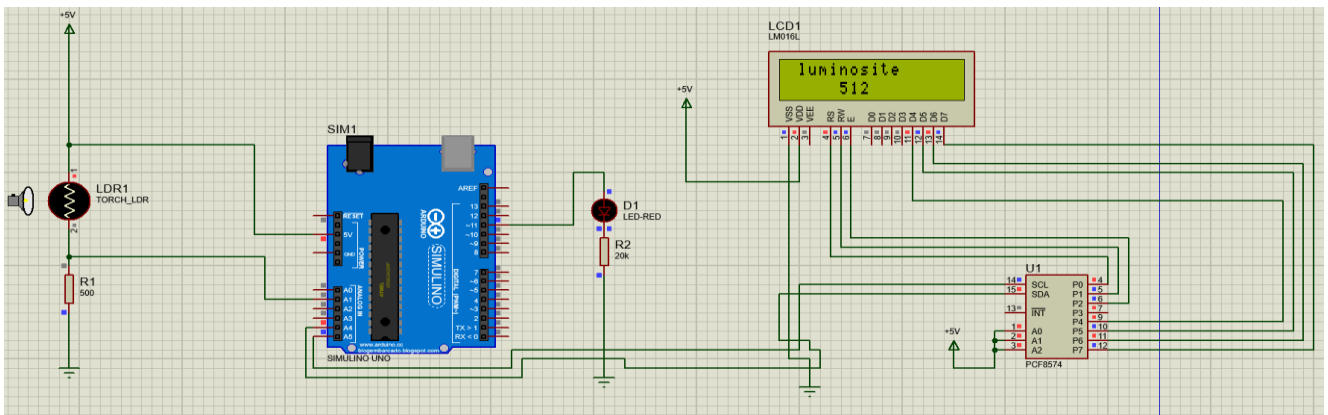


Figure4 :Les résistances utilisées dans le montage

Quand la valeur de luminosité est supérieure à 500 la LED s'éteint :



Quand la valeur de luminosité est inférieure à 500 la LED s'allume :

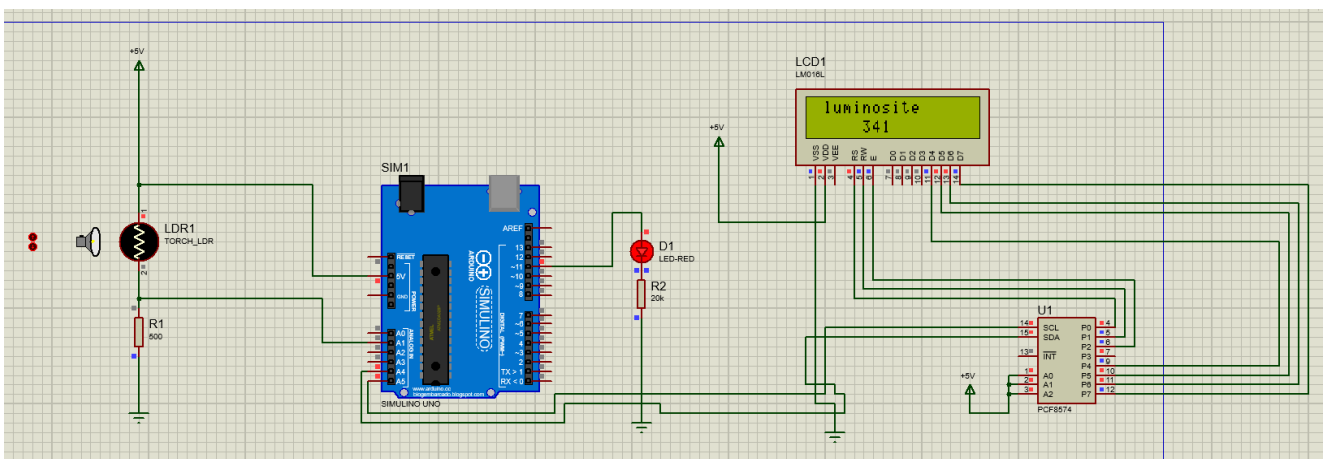


Figure5 :Shéma ISIS

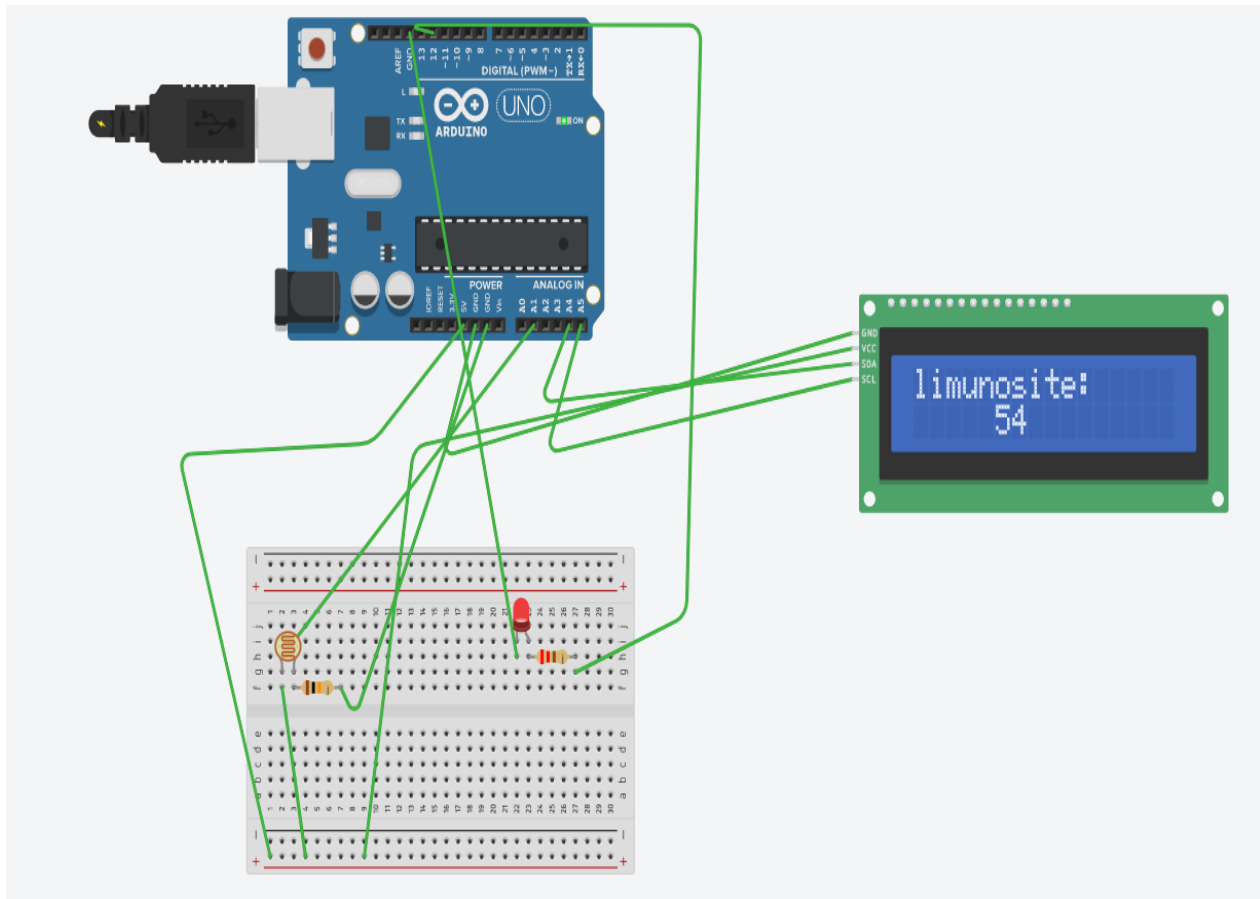


Figure6 :Simulation THINKEDCAD

2.Principe de fonctionnement :

Le montage permet de mesurer la luminosité à travers le capteur de luminosité LDR ,ainsi l'éclairement de la lampe change en fonction de la valeur de la luminosité détectée par le capteur ,elle s'allume quand la valeur de luminosité est inférieure à 500 et s'éteint lorsque cette valeur est supérieure à 500,la valeur de luminosité est affichée dans La LCD .On utilise Le module I2C LCD 1602A pour simplifier les choses, il contient 4 broches : on branche son GND et VCC respectivement avec le GND et VCC de l'arduino et SDA(signal de donnée) avec l'entrée analogique A4 de l'arduino et SCL(signal d'horloge) avec l'entrée analogique A5 et on branche la LDR en série avec la résistance de 10kohms et la LED avec une résistance de 330 ohms .

3.Le programme :

```
#include<Wire.h>

#include<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);//0x27 l'adresse du I2C,16 nombre de
colonnes et 2 nombre de lignes.
int LED =11;//La LED est branchée à La sortie numérique 11 de l'arduino.
int LDR =A1;//La LDR est branchée à l'entrée analogique A1 de l'arduino.
int val ; //Déclaration de la variable de luminosité.
void setup()
{
    pinMode(LED,OUTPUT);//Déclaration de la LED comme sortie.
    lcd.init();//initialisation de la LCD.
    lcd.backlight();//le rétro éclairage de la LCD.
}

void loop()
{
    val=analogRead(LDR);//Lire la valeur indiqué par la LDR.
    if(val<500){
        digitalWrite(LED,HIGH);//allumer la LED.
        lcd.setCursor(1,0);//Placer le curseur dans la colonne 1 et la ligne zéro.
        lcd.print("luminosité");//Ecrire « luminosité »
        lcd.setCursor(5,1);//Placer le curseur dans la colonne 5 et la ligne 1.
        lcd.print(val);//Ecrire la valeur de la LDR.
    }
    else{
        digitalWrite(LED,LOW);//Eteindre la LED.
        lcd.setCursor(1,0);//Placer le curseur dans la colonne 1 et la ligne zéro.
        lcd.print("luminosité");//Ecrire « luminosité »
        lcd.setCursor(5,1);//Placer le curseur dans la colonne 5 et la ligne 1.

        lcd.print(val);Ecrire la valeur de la LDR.
    }
}
```

4.Etalonnage :

Etalonner un capteur ou une chaîne de mesure, c'est établir une relation entre les grandeurs d'entrée et de sortie d'un capteur ou d'une chaîne de mesure. L'étalonnage doit permettre à partir de la grandeur de sortie, de fournir la valeur de la grandeur d'entrée assortie d'une incertitude-type.

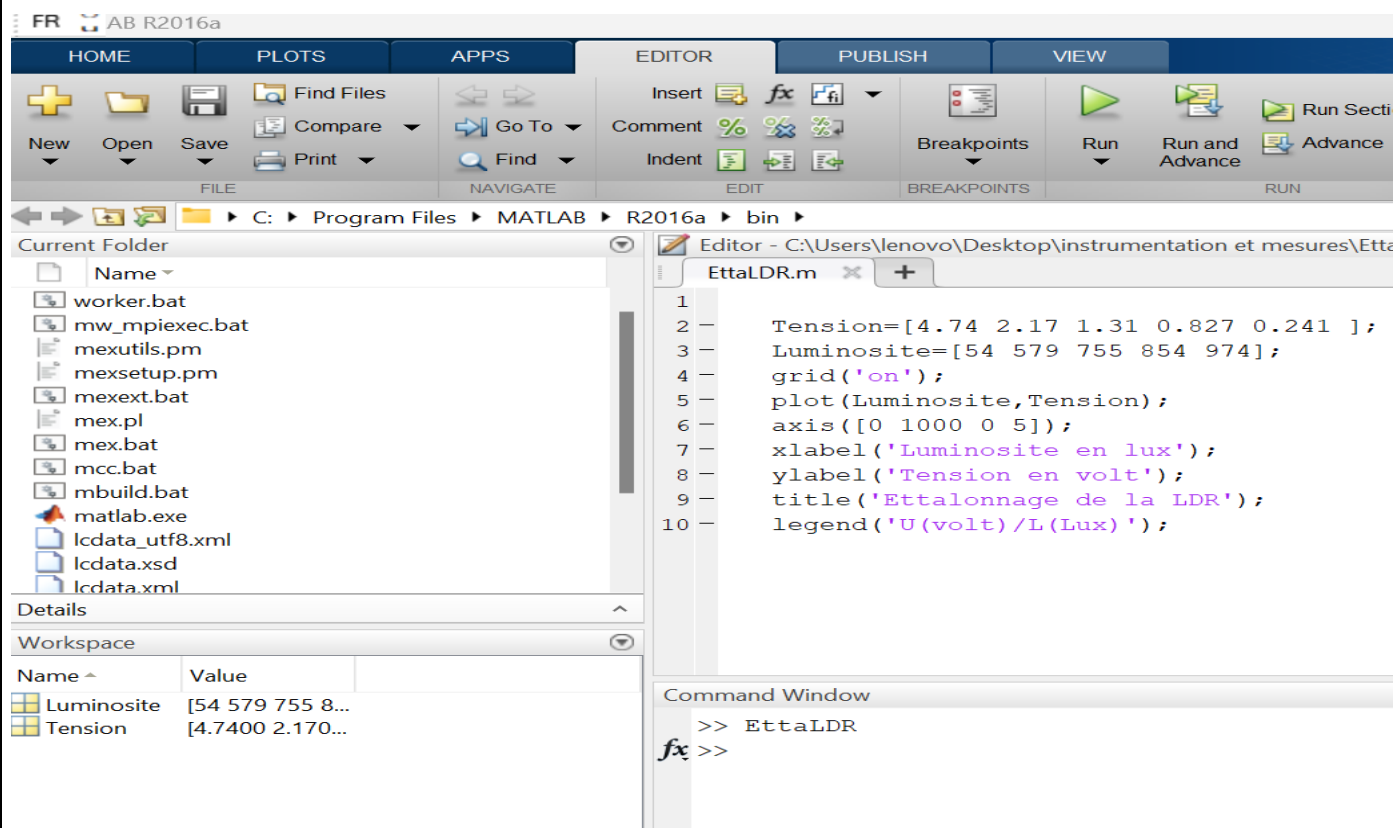


Figure7 :Programmematlab d'étalonnage.

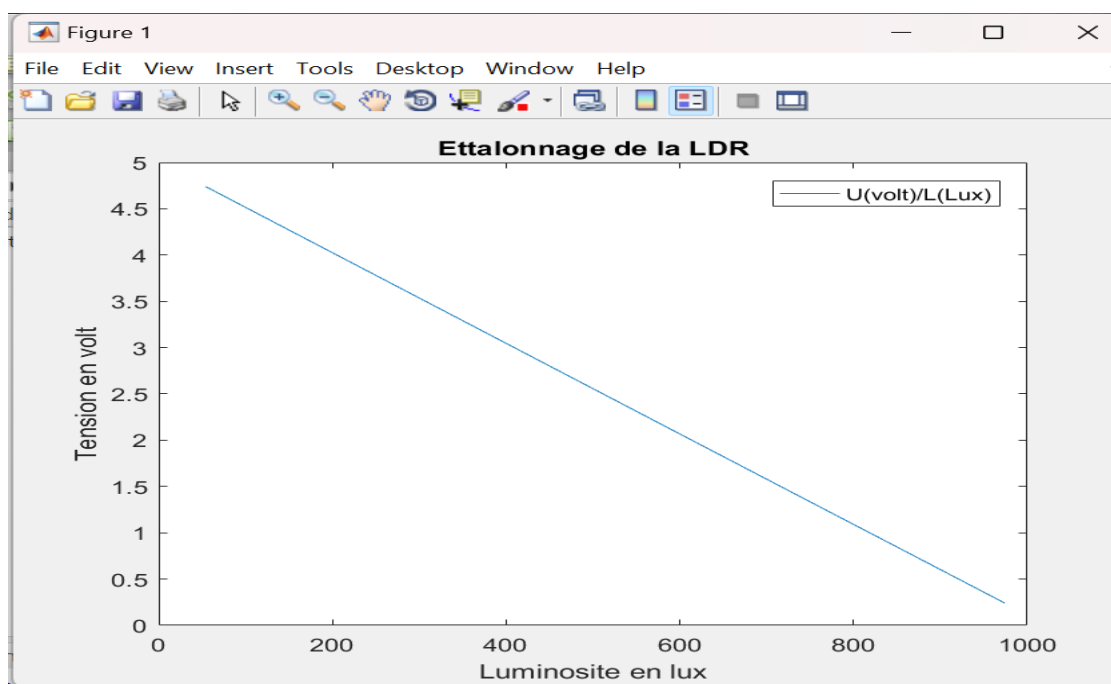


Figure8 :Courbe d'étalonnage.

5. Fidélité et justesse d'un LDR :

Justesse : Pour déterminer calculer la justesse on doit calculer la valeur moyenne et la comparer avec la valeur vraie :

$$M = \frac{S}{n}$$

$$M = \frac{351+349+352+353+348}{5} = 350.6$$

On considère que la valeur vraie égale à 350 lux d'une chambre très éclairée :

$$J = M - V = 0.6 < 1$$

Fédilité : Pour déterminer la fidélité on doit calculer la valeur de l'écart type :

$$\sigma = \sqrt{\frac{\sum (x_i - m)^2}{n - 1}}$$

$$\text{Ecart type} = 2.0736 > 1$$

Donc le capteur est juste et non fidèle donc il n'est pas précis.

CHAPITRE 2 : CAPTEUR DHT22

I. DHT22 :

1. Présentation :

Le capteur DHT22 est un capteur de température et d'humidité à sortie numérique série. La liaison entre la carte Arduino et le capteur est assurée par un microcontrôleur 8 bits intégré dans le corps du DHT22. Une bibliothèque permet la communication entre le capteur et la carte ARDUINO.

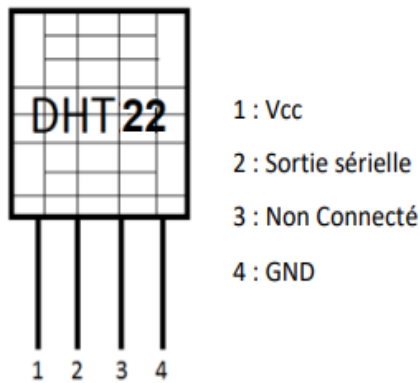


Fig.2 : Capteur DHT22

Caractéristiques :

Humidité (relative%) : 0 à 100 %
Précision (humidité) : +/- 2%
Température : -40°C à +150°C
Précision (température) : +/- 0,5°C
Fréquence mesure : 1/2 Hz (2 mesure par seconde)
Tension d'alimentation : 3 à 5 volts

Mesure de la température et de l'humidité

La mesure de ces 2 variables se fait via la bibliothèque `<dht.h>`, se référer à la notice d'installation de la carte CAPAX-Xtd® pour son utilisation.

2. Brochage :

Il faut alimenter le capteur (5V) et utiliser une résistance de *Pull-Up* (4,7kΩ à 10kΩ) pour le port GPIO de communication avec le capteur.

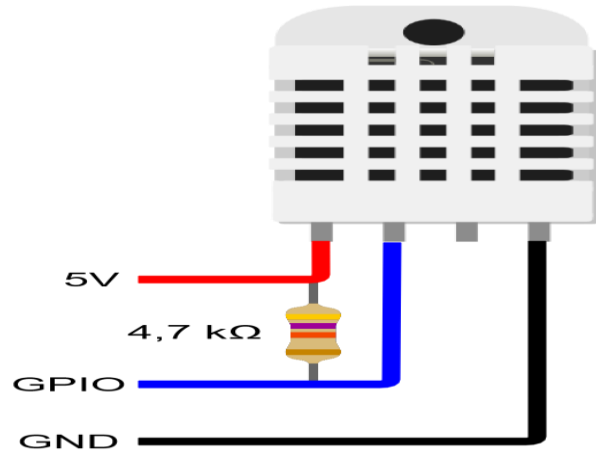


Figure 9: Brochage de DHT22.

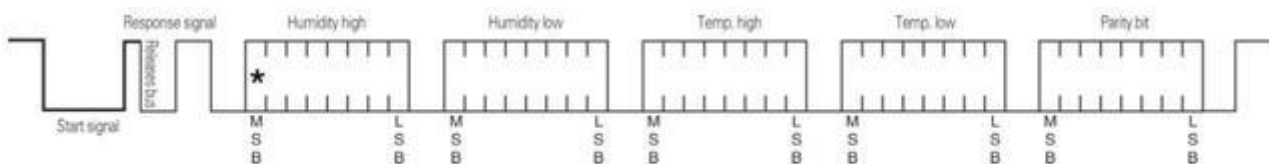
Remarque :

Le capteur DHT22 est un capteur numérique de température et d'humidité. Il utilise un capteur d'humidité Capacitif et une thermistance pour mesurer l'air ambiant et génère un signal numérique sur la broche de Données. Le seul inconvénient réel de ce capteur est sa période de mesure : une fois toutes les 2 secondes.

3. Protocole de communication :

La communication avec un capteur DHT se fait en 3 étapes :

- Tout d'abord, le microcontrôleur maître (la carte Arduino dans notre cas) réveille le capteur en plaçant la ligne de données à LOW pendant au moins 800µs (au moins 18ms pour le DHT11). Durant ce laps de temps, le capteur va se réveiller et préparer une mesure de température et d'humidité. Une fois le temps écoulé, le maître va libérer la ligne de données et passer en écoute.
- Une fois la ligne de données libérée, le capteur répond au maître (pour montrer qu'il est bien réveillé) en maintenant la ligne de données à LOW pendant 80µs puis à HIGH pendant 80µs.
- Le capteur va ensuite transmettre une série de 40 bits (5 octets). Les deux premiers octets contiennent la mesure de l'humidité. Les deux octets suivants contiennent la mesure de la température et le cinquième octet contient une somme de contrôle qui permet de vérifier que les données lues sont correctes.



Pic5: AM2302 Single-bus communication protocol

Figure 10 : bus de communication (protocole)

Table 5: AM2302 Communication format specifier

Name	Single-bus format definition
Start signal	Microprocessor data bus (SDA) to bring down a period of time (at least 800µ s) [1] notify the sensor to prepare the data.
Response signal	Sensor data bus (SDA) is pulled down to 80µ s, followed by high-80µ s response to host the start signal.
Data format	Host the start signal is received, the sensor one-time string from the data bus (SDA) 40 data, the high first-out.
Humidity	Humidity resolution of 16Bit, the previous high; humidity sensor string value is 10 times the actual humidity values.
Temp.	Temperature resolution of 16Bit, the previous high; temperature sensor string value is 10 times the actual temperature value; The temperature is the highest bit (Bit15) is equal to 1 indicates a negative temperature, the temperature is the highest bit (Bit15) is equal to 0 indicates a positive temperature; Temperature in addition to the most significant bit (Bit14 ~ bit 0) temperature values.
Parity bit	Parity bit = humidity high + humidity low + temperature high + temperature low

Figure 11 : communication format spécifier

4. Traduction des mesures :

Le circuit donne 16 bits pour l'humidité relative, exprimée en dixièmes et codé en binaire.

Par exemple, si les 2 premiers bites sont 0000 0010 1001 0010 = 0x0292

La conversion en décimal donne 658, l'humidité est 65.8%.

Un Serial.print () Arduino affichera 658 et c'est toujours comme cela que cela se passe avec les langages "évolués": les variables sont en binaires en mémoire, mais on les montre en décimal.

Si les 2 bites suivant pour la température sont 1000 0000 0110 0101, le 1er bit à un indique que

la température est négative et sa valeur absolue suit: 0000 0000 0110 0101 = 0x0065

La conversion en décimal donne 101, donc la température est de -10,1 degrés

Il semble important de vérifier que la somme des 4 premières mesures (en ignorant les dépassements) est égale à la somme de contrôle. Il faut alors recommencer la mesure.

II. Interface Arduino/DHT22 :

1. Programme arduino :

```
#include <LiquidCrystal.h>           // include LCD library code

#include "DHT.h"                     // include DHT library code
#define DHTPIN 8                     // DHT22 data pin is connected to Arduino pin 8

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // LCD module connections (RS, E, D4, D5, D6, D7)

#define DHTTYPE DHT22               // DHT22 sensor is used

DHT dht(DHTPIN, DHTTYPE);           // Initialize DHT library

char temperature[] = "Temp = 00.0 C";

char humidity[] = "RH = 00.0 %";

void setup() {

  Serial.begin(9600);

  dht.begin();                      // set up the LCD's number of columns and rows

  lcd.begin(16, 2);

  dht.begin();

}

void loop() {

  delay(1000);                      // wait 1s between readings

  // Read humidity

  int RH = dht.readHumidity()*10;

  //Read temperature in degree Celsius

  int Temp = dht.readTemperature()*10;

  Serial.println("DHT22");

  Serial.print(Temp);

  Serial.println("°C");

  Serial.print(RH);

  Serial.println("%");              // Check if any reads failed and exit early (to try again)
```

```

if (isnan(RH) || isnan(Temp)) {

    lcd.clear();

    lcd.setCursor(5, 0);

    lcd.print("Error");

    return;

}

if(Temp < 0){

    temperature[6] = '-';

    Temp = abs(Temp);

}

else

    temperature[6] = ' ';

temperature[7] = (Temp / 100) % 10 + 48;

temperature[8] = (Temp / 10) % 10 + 48;

temperature[10] = Temp % 10 + 48;

temperature[11] = 223;    // Degree symbol ( °)

if(RH >= 1000)

    humidity[6] = '1';

else

    humidity[6] = ' ';

humidity[7] = (RH / 100) % 10 + 48;

humidity[8] = (RH / 10) % 10 + 48;

humidity[10] = RH % 10 + 48;

lcd.setCursor(0, 0);

lcd.print(temperature);

lcd.setCursor(0, 1);

lcd.print(humidity);

}

```

2. Simulation ISIS :

La simulation est réalisée via le Simulateur PROTEUS ISIS :

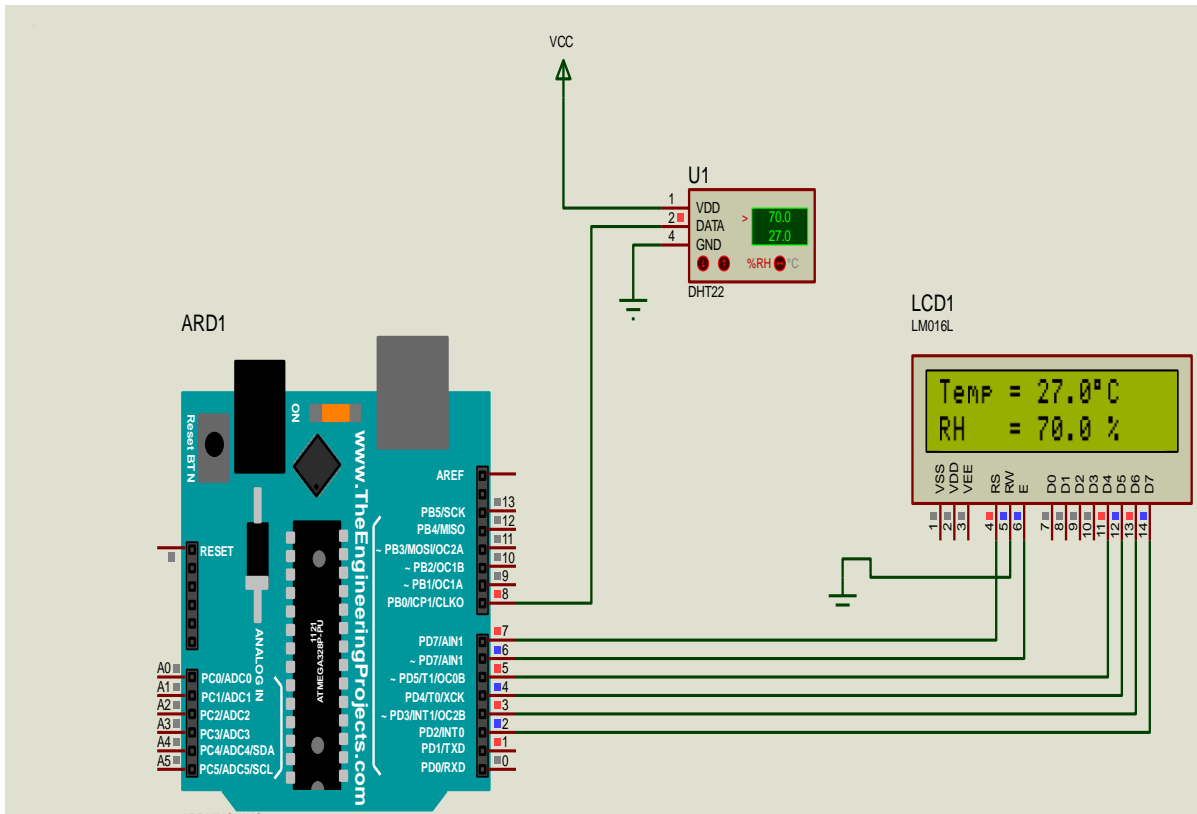


Figure 12: simulation isis du DHT22.

Notes :

- Le capteur DHT22 envoie un signal numérique dirigé vers les entrées digitales de la carte arduino.
- La carte arduino traite le signal délivrée par le capteur via le programme arduino puis Elle affiche la mesure effectuée sur la LCD (bus I2C).

III. Réalisation :

1. Description :

On représente ci- dessous une image montre la réalisation pour effectuer
Après une application simple .

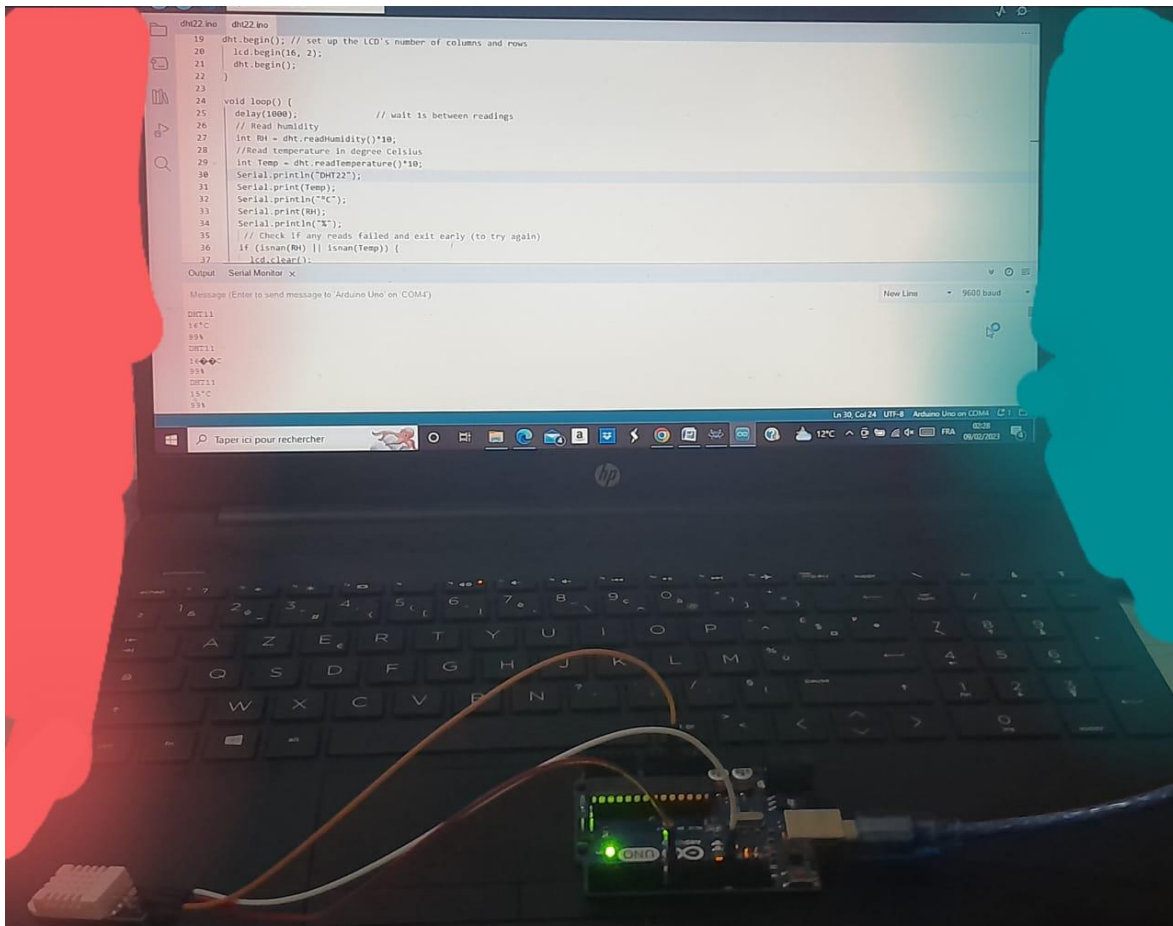


Figure 13 : image réelle montre la réalisation .

Notes :

La réalisation nécessite la connexion du PC (logiciel ARDUINO IDE) avec
La carte ARDUINO via un câble USB et configurer la connexion sur le logiciel
De programmation (port COM4) ainsi que ajouter un bloc d'instruction pour
Assurer la liaison .

2. Courbe d'étalonnage :

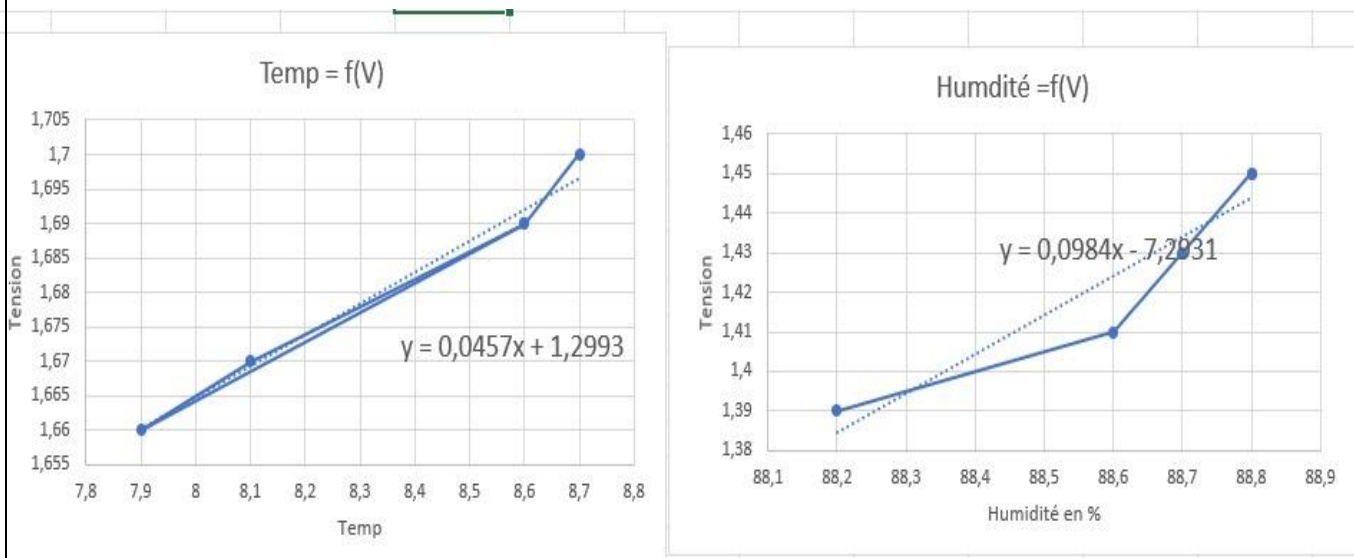


Figure 14 : la courbe d'étalonnage.

3. Application :

Objectif: mesurer la température du corps humain :

Séries de mesures : **24 °C, 26 °C , 27 °C , 28 °C, 29 °C ,30 °C, 31 °C, 32 °C ,33 °C.**

La valeur vraie : **36°C**

Déterminons la valeur moyenne :

$$M= 28.8$$

Déterminons l'erreur de fidélité : **l'écart type = 0.36**

Déterminons la justesse et la précision :

$$E_j = 7.2 \quad \text{et} \quad E_p = 7.208$$

Déductions :

- La fidélité : puisque les mesures sont proches l'une de l'autre donc Le capteur est fidèle.
- La justesse : puisque il ya une différence entre la valeur moyenne et la valeur vraie don le capteur n'est pas juste.
- La précision : le capteur n'est pas précis.

CONCLUSION

CE MINI-PROJET EST UN ESPACE POUR MAITRISER L'UTILISATION PRATIQUE DES CAPTEURS ET DECOUVRIR LEURS PRINCIPES DE FONCTIONNEMENT AINSI QUE ETUDIER LEURS CARACTERISTIQUES METROLOGIES.