



# **RAPPORT DU MINI-PROJET**

## **CAPTEUR ET ACTIONNEUR**



*Sujet :*

## ***StationMétéo***

**Realiser par :**

Harraoui sohaib  
Eddouche yasser  
Oumaima ighil

**Présenté devant le jury**  
**compose par :**

- Pr. TAOUSSI Mohamed
- Pr. JAI Otman
- Pr. CHAIBI Radouane

**Encadre par :**

Pr. JAI Otman

# **SOMMAIRE**

<b>Introduction .....</b>	<b>4</b>
---------------------------	----------

<b>I. Etude du Projet.....</b>	<b>5</b>
--------------------------------	----------

1.1 Definition du projet. ....	5
--------------------------------	---

1.2 Choix des logiciels utilisées.....	5
--	---

1.3 Choix de materiel. ....	5
-----------------------------	---

<b>II. Realisation du Projet.....</b>	<b>3</b>
---------------------------------------	----------

2.1 Realisation virtuel.....	
------------------------------	--

2.2 Cablage de Station.....	
-----------------------------	--

2.3 Fonctionnement de Station.....	
------------------------------------	--

<b>Conclusion .....</b>	<b>4</b>
-------------------------	----------

# Introduction

Au cours de la première période de notre première année de DUT, nous devons faire un projet en groupes variables de trois à quatre personnes sur différents sujets. Ce petit projet nous concerne

Amène à travailler en équipe et de manière autonome sur une réalisation correspondante nos compétences.

Ce sujet nous mène à réaliser une station météo

## **I. ETUDE DU PROJET :**

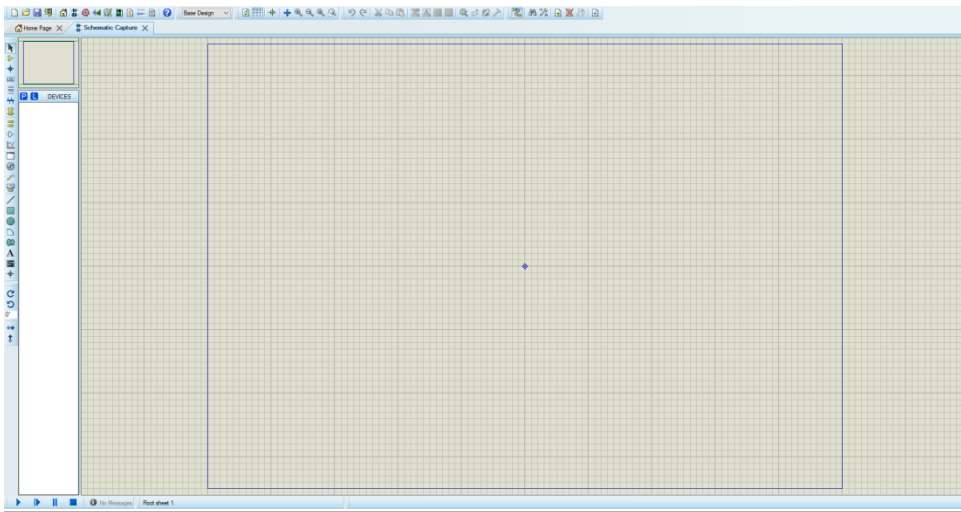
### **1.1 Définition du projet:**

Une station météorologique, parfois appelée abri météorologique synecdoque, est un ensemble de capteurs qui enregistrent et fournissent des mesures physiques et des paramètres météorologiques liés aux changements climatiques, ces capteurs sont logés dans un abri météorologique, qui réalise et protège l'équilibre thermique du thermomètre avec l'air. De la lumière du soleil <sup>1</sup>. Les variables à mesurer sont la température et la pression, et les stations peuvent comporter des capteurs pour tout ou partie de ces informations uniquement.

### **1.2 Choix des logiciels utilisés:**

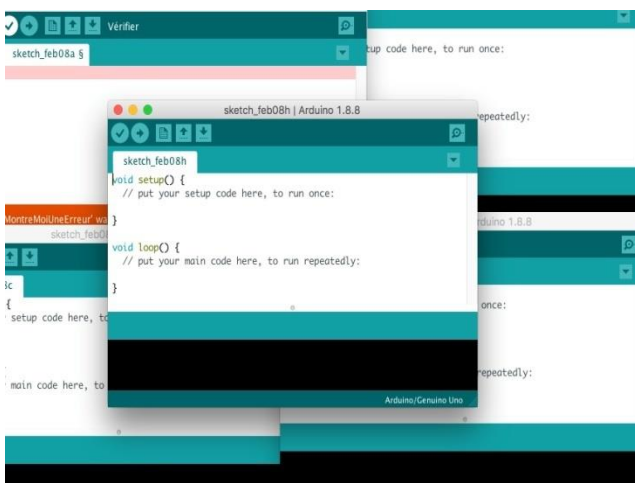
**Proteus électronique :**

**Proteus** est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels inclus dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES, PROSPICE et VSM.



## Arduino ide :

Un Environnement de Développement Intégré (**IDE**) open source et gratuit, téléchargeable sur le site officiel **Arduino**.



<<pour écrire le programme

## 1.3 Choix demateriel:

### Arduino MEGA :

La carte Arduino Mega 2560 est basée sur un ATmega2560 cadencé à 16 MHz. Elle dispose de 54 E/S dont 14 PWM, 16 analogiques et 4 UARTs. Elle est idéale pour des applications exigeant des caractéristiques plus complètes que la Uno.

Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

Cette carte peut se programmer avec le logiciel Arduino disponible gratuitement (IDE).

#### Caractéristiques:

- Alimentation:
  - via port USB ou
  - 7 à 12 V sur connecteur alim
- Microprocesseur: ATmega2560
- Mémoire flash: 256 kB
- Mémoire SRAM: 8 kB
- Mémoire EEPROM: 4 kB
- 54 broches d'E/S dont 14 PWM
- 16 entrées analogiques 10 bits
- Intensité par E/S: 40 mA
- Cadencement: 16 MHz
- 3 ports série
- Bus I2C et SPI
- Gestion des interruptions
- Fiche USB B
- Version:Rev 3



- Dimensions: 107 x 53 x 15 mm

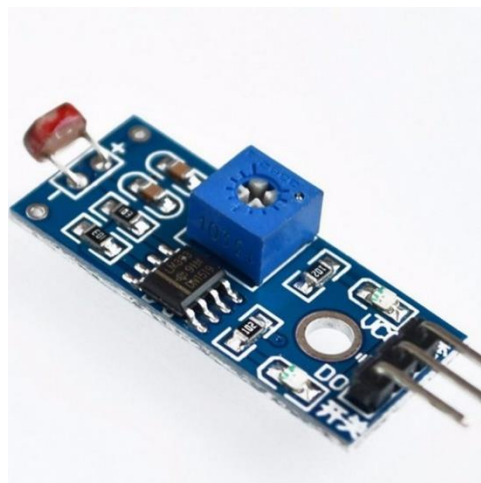
## Les Capteurs :

- **Capteur photosensible :**  
**LDR** ou photorésistance est un composant électronique dont la résistance varie en fonction de la lumière perçue. La valeur de la luminosité est convertit en résistance qu'il est facile d'acquérir avec l'aide d'une carte Arduino au travers d'une entrée analogique.

**Potentiomètre** est une résistance variable manuellement.

- **DHT11** : Le capteur DH11 mesure la température et l'humidité. L'utilisation de ce type de capteur est intéressante pour un débutant car il permet de mesurer une grandeur physique accessible à tous.

Le capteur DHT11 a 4 broches, mais il est souvent vendu sur une carte support qui possède 3 broches. Il communique avec l'Arduino très simplement au travers d'une de ses entrées numériques. Les 2 autres broches sont pour son alimentation 5 V et la masse (GND).





**LIBRERIE :** #include <DHT.h>

➤ 5 bouton poussoir :

Un **bouton** (ou **bouton poussoir** et **bouton-poussoir**) est un interrupteur simple qui permet de contrôler les capacités d'un processus

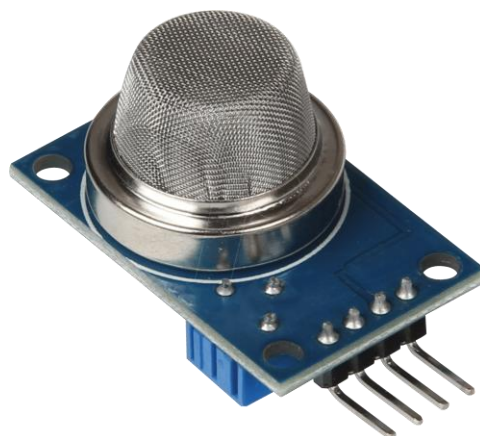


Les boutons sont généralement fabriqués à partir de matériaux durs, habituellement en plastique ou en métal mais peuvent également être constitués de caoutchouc. On distingue deux types de boutons : le bouton normalement ouvert et celui normalement fermé (contact repos (CR) ou contact travail (CT))



### ➤ MQ2 :

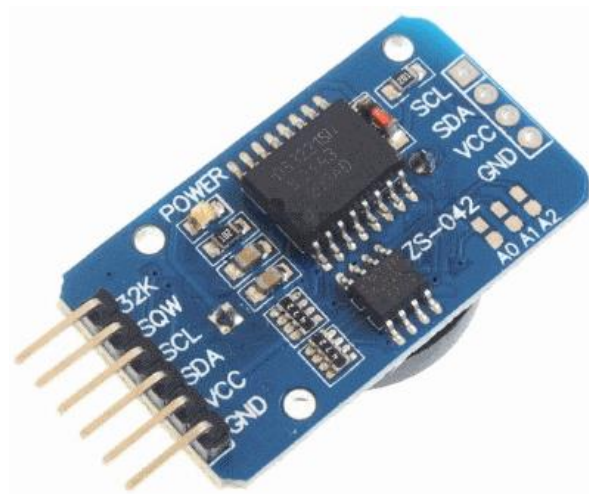
Le **MQ2** est un capteur qui permet de détecter du gaz ou de fumée à des concentrations de 300 ppm à 10000 ppm. Après calibration, le **MQ2** peut détecter différents gaz comme le GPL (LPG), l'i-butane, le propane, le



méthane, l'alcool, l'hydrogène ainsi que les fumées.

### ➤ RTC DS3231 :

Le module **DS3231** est une horloge temps réel (**RTC**) qui contient un quartz compensée pour les dérives suivant la température. La communication se fait via le bus I2C. Les dates et heures sont mémorisées grâce à une pile CR2032



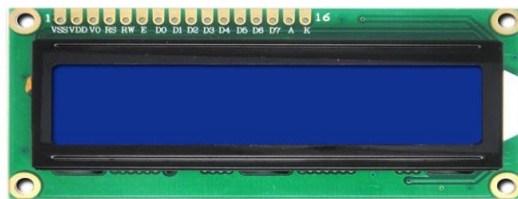
**LIBRERIE** : `#include <TimeLib.h>`

`#include <DS3231.h>`

`#include <Wire.h>` : pour la communication I2C de DS3231

`#include <EEPROM.h>` : pour enregistrer la date d'alarme

- **Potentiomètre** : Un potentiomètre (appelé familièrement potard) est un type de résistance variable à trois bornes, dont une est reliée à un curseur se déplaçant sur une piste résistante terminée par les deux autres bornes.
- **LCD 16x2** : L'LCD permet la création d'écrans plats à faible consommation d'électricité. Aujourd'hui ces écrans sont utilisés dans presque tous les affichages

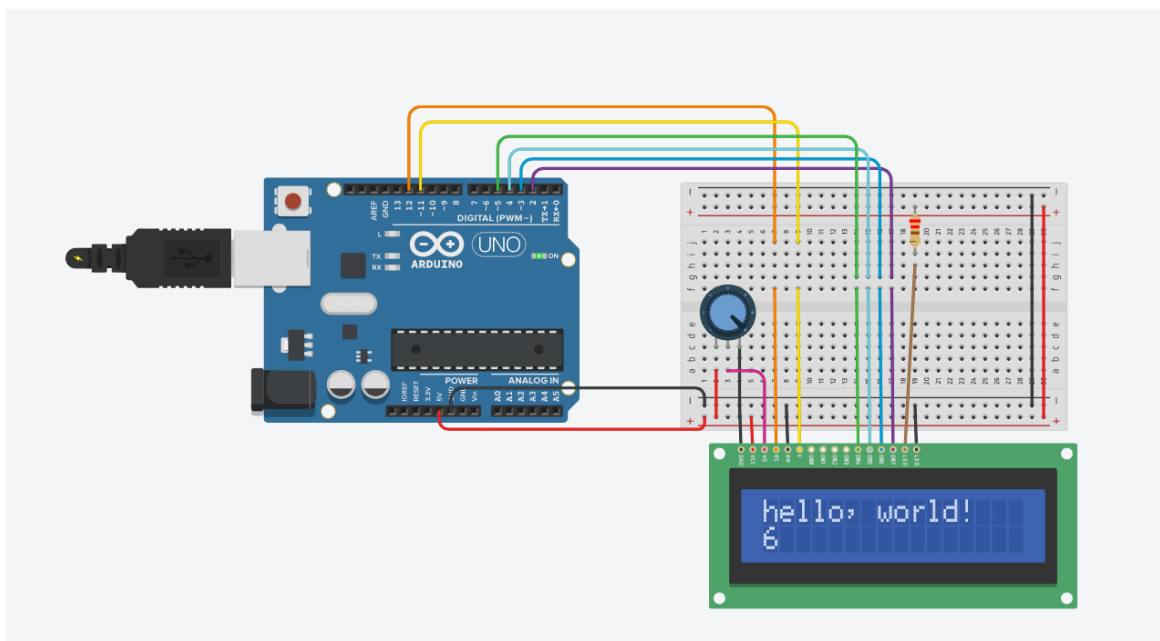


électroniques.

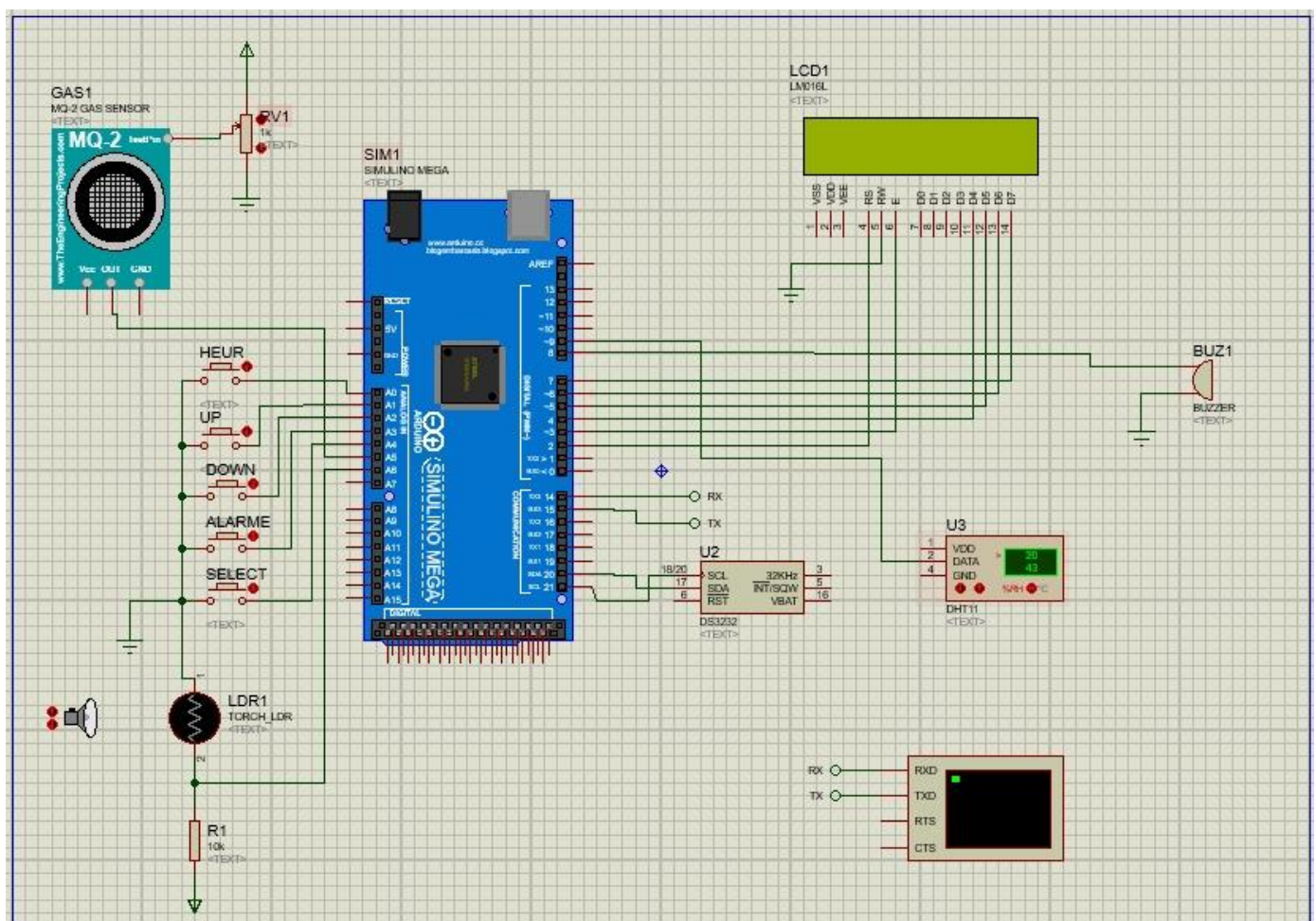


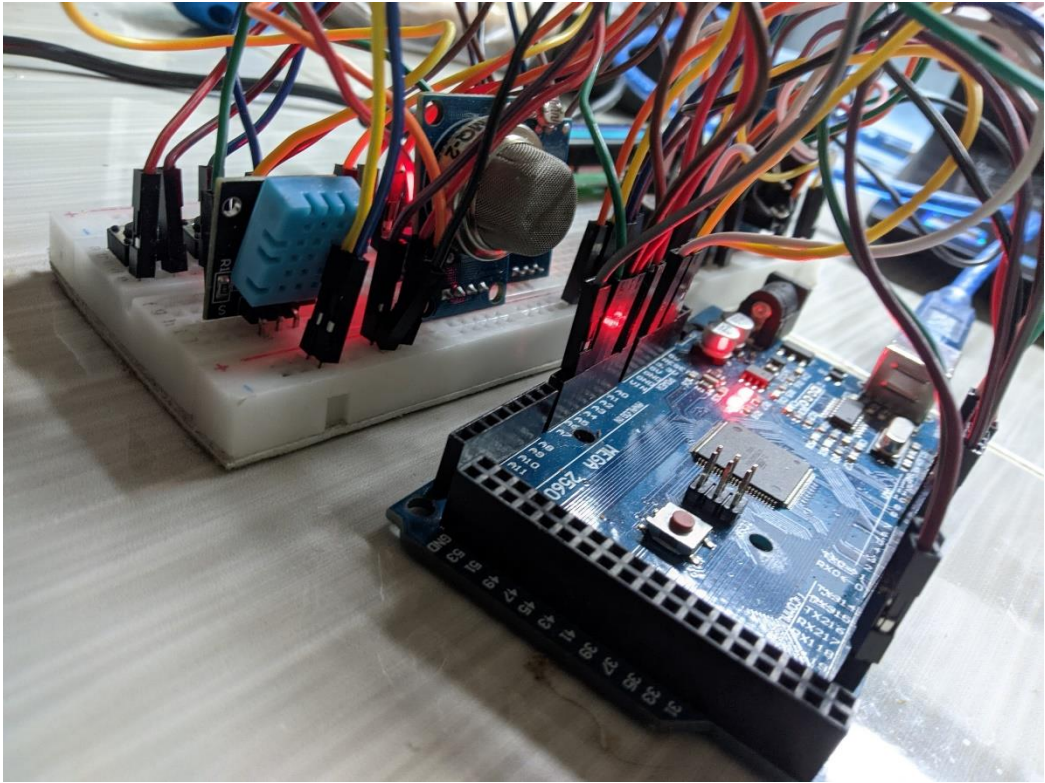
## II. ETUDE DU PROJET :

## 2.1 Réalisation virtuel :



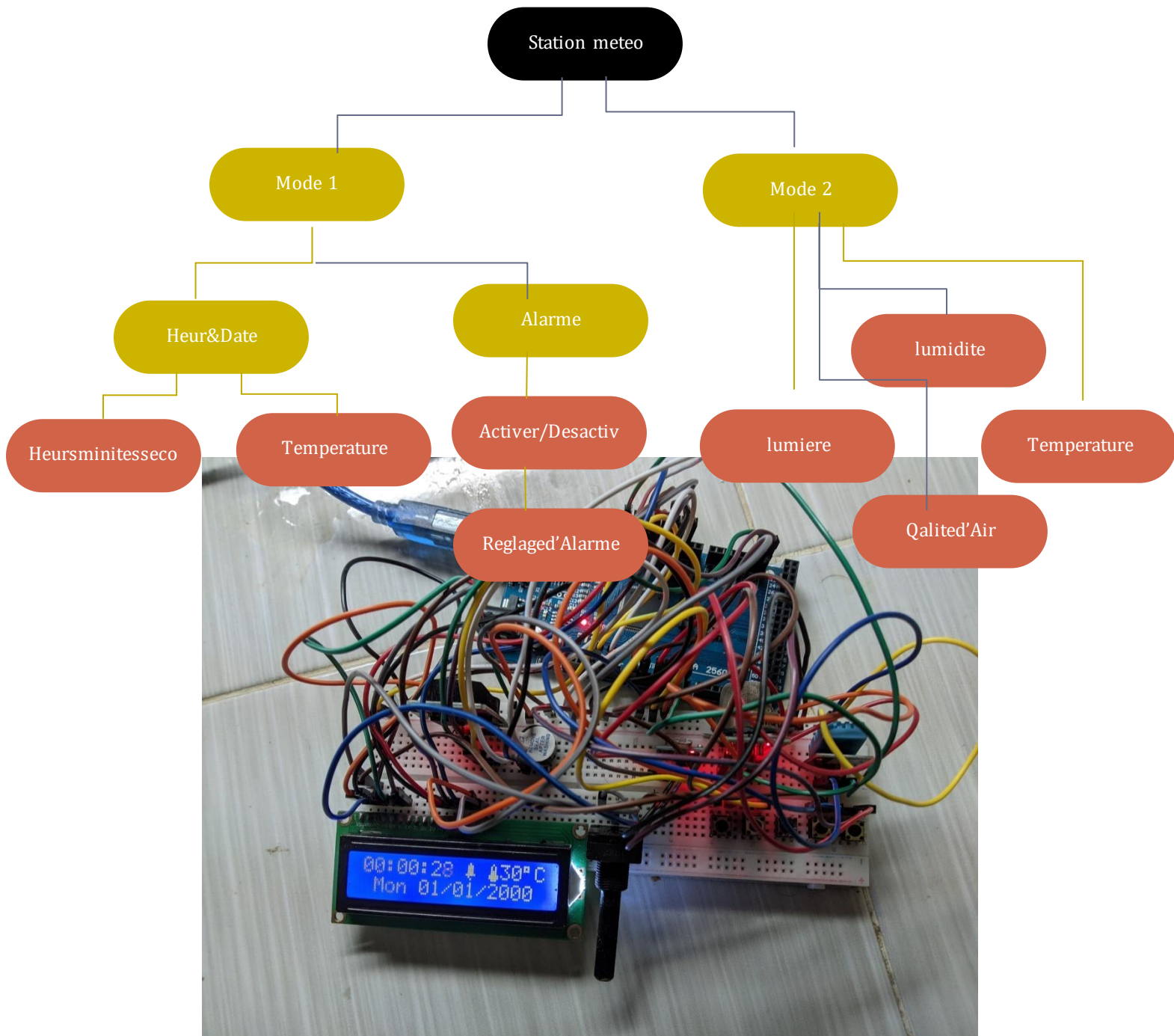
## 2.2 Cablage du Station :







## 2.3 Fonctionnement du station :







Le jour

La Date

Notre station météo affiche sur LCD au moment de démarrage le nom de projet pendant deux secondes puis il entre au premier mode de fonctionnement dans lequel il y a une horloge d'heure et minute et seconde et on a dans le deuxième ligné de LCD le jour et la date , on règle le paramètre précédent : premièrement par select le paramètre à l'aide de boutons poussoirheure&date et l'incrémentation ou décrémentation s'effectuer par bouton UP et DOWN, en fin de régler tous les paramètres dans première fenêtre (setup end) la date et enregistrer dans RTC, en cas de perdre l'énergie la station gardera le temps .

Un click sur le bouton d'alarme station entrera dans la deuxième fenêtre de réglage de l'alarme et il déclenchera un buzzer quand le temps vient.

Un click sur le bouton de select mode affichera les mesures en temps réel des grandeurs atmosphériques physiques.

### 3.Le code source :

```
#include<LiquidCrystal.h>
#include<TimeLib.h>
#include<DHT.h>
#include<DS3231.h> //Memanggil RTC3231 Library
#include<Wire.h>    // i2C Connection Library
#include<EEPROM.h>

//----DEFINE-----
#define smokeA0    A5
#define sensorThres 400
#define DHTPIN9
#define DHTTYPE DHT11
#define LDRA6
#define bt_time    A0
#define bt_up      A1
#define bt_down    A2
#define bt_alarm   A3
#define bt_selectA4
#define buzzer8
//-----
//-----
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(2,3,4,5,6,7);
DS3231 rtc(20,21);
// Init a Time-data structure
Time t; //pencacah string time()

int hh=0, mm=0, ss=0, dd=0, bb=0, set_day;
int yy=0;
String Day="  ";

int AlarmHH =21, AlarmMM =22, AlarmSS =23, setMode=0, setAlarm=0, alarmMode=0;
bool select_mode=false;
int stop=0, mode=0, flag=0;

//Eeprom Store Variable
uint8_t HH;
uint8_t MM;

byte bell_symbol[8]={
    B00100,
    B01110,
    B01110,
    B01110,
    B01110,
    B11111,
    B01000,
    B00100};
byte thermometer_symbol[8]={
    B00100,
    B01010,
    B01010,
    B01110,
    B01110,
    B11111,
    B11111,
```

```
B01110};
```

```
void setup() {
  Serial.begin(9600);
  dht.begin();
  lcd.begin(16,2); // initialize the lcd
  rtc.begin(); // memulaikoneksi i2c dengan RTC

  pinMode(bt_time, INPUT_PULLUP);
  pinMode(bt_up, INPUT_PULLUP);
  pinMode(bt_down, INPUT_PULLUP);
  pinMode(bt_alarm, INPUT_PULLUP);
  pinMode(bt_select, INPUT_PULLUP);
  pinMode(smokeA0, INPUT);
  pinMode(LDR, INPUT);
  pinMode(buzzer, OUTPUT);

  lcd.createChar(1, thermometer_symbol);
  lcd.createChar(2, bell_symbol);

  lcd.setCursor(0,0);
  lcd.print(" Station Meteo");
  lcd.setCursor(0,1);
  lcd.print(" WithAlarm ");
  delay(2000);
  lcd.clear();

  stop=EEPROM.read(50);
  if(stop==0){
  }else{WriteEeprom();}

  EEPROM.write(50,0);
  ReadEeprom();
}

void loop() {
  int analogSensor=analogRead(smokeA0);
  int ldrStatus=analogRead(LDR);
  //-----
  t=rtc.getTime();
  Day=rtc.getDOWStr(1);
  int state=digitalRead(bt_select);

  if(state==LOW){delay(100);select_mode=!select_mode;delay(50);lcd.clear();}

  if(!select_mode){
    Serial.println("nonapuuier");
    if(setMode==0){
      hh=t.hour,DEC;
      mm=t.min,DEC;
      ss=t.sec,DEC;
      dd=t.date,DEC;
      bb=t.mon,DEC;
      yy=t.year,DEC;
    }

    if(setAlarm==0){
      lcd.setCursor(0,0);
      lcd.print((hh/10)%10);
      lcd.print(hh%10);
    }
  }
}
```

```

    lcd.print(":");
    lcd.print((mm/10)%10);
    lcd.print(mm%10);
    lcd.print(":");
    lcd.print((ss/10)%10);
    lcd.print(ss%10);
    lcd.print(" ");
    if(mode==1){lcd.write(2);}
    else{lcd.print(" ");}
    lcd.print(" ");
    lcd.write(1);
    lcd.print(rtc.getTemp(),0);
    lcd.write(223);
    lcd.print("C");
    lcd.print(" ");

    lcd.setCursor(1,1);
    lcd.print(Day);
    lcd.print(" ");
    lcd.print((dd/10)%10);
    lcd.print(dd%10);
    lcd.print("/");
    lcd.print((bb/10)%10);
    lcd.print(bb%10);
    lcd.print("/");
    lcd.print((yy/1000)%10);
    lcd.print((yy/100)%10);
    lcd.print((yy/10)%10);
    lcd.print(yy%10);
}

setupClock();
setTimer();
delay(100);
blinking();

//Alarm
if(alarmMode==1&&mode==1&&hh==AlarmHH&&mm==AlarmMM&&ss>=AlarmSS){
    digitalWrite(buzzer,HIGH);
    delay(300);
    digitalWrite(buzzer,LOW);
}else{digitalWrite(buzzer,LOW);}

delay(100);
}
if(select_mode){
    //lcd.clear();
    //lcd.setCursor(0,0);
    //lcd.print("Mode 2");
    Serial.println("appuier");
    //*****
    //if (state== LOW) {delay(100);select_mode =
!select_mode;delay(50);lcd.clear();}
    select_mode=!select_mode;
if(analogSensor>sensorThres)
{
    lcd.setCursor(0,0);
    lcd.print("qualite d'Air:");
    lcd.setCursor(4,1);
    lcd.print("dangereux");
}

```

```

        tone(buzzer,1000,200);
    }
    else
    {
        lcd.setCursor(0,0);
        lcd.print("qualite d'Air:");
        lcd.setCursor(4,1);
        lcd.print("Bonne");
        noTone(buzzer);
    }
    delay(3000);
    lcd.clear();
    delay(100);
    lcd.setCursor(0,0);
    lcd.print("Temperature = ");
    lcd.setCursor(4,1);
    lcd.print(String(dht.readTemperature())+" C");
    delay(3000);
    delay(100);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Humidite = ");
    lcd.setCursor(4,1);
    lcd.print(String(dht.readHumidity())+" %");
    delay(3000);
    lcd.clear();
    delay(100);

    lcd.setCursor(5,0);
    if(ldrStatus<=250){
    lcd.print("jour");
    }else{
    lcd.print("nuit");
    }
    delay(3000);
    lcd.clear();
    delay(100);

    }
}

voidblinking(){
//BLINKING SCREEN
if(setAlarm<2&&setMode==1){lcd.setCursor(0,0);  lcd.print("  ");}
if(setAlarm<2&&setMode==2){lcd.setCursor(3,0);  lcd.print("  ");}
if(setAlarm<2&&setMode==3){lcd.setCursor(6,0);  lcd.print("  ");}
if(setAlarm<2&&setMode==4){lcd.setCursor(1,1);  lcd.print("  ");}
if(setAlarm<2&&setMode==5){lcd.setCursor(5,1);  lcd.print("  ");}
if(setAlarm<2&&setMode==6){lcd.setCursor(8,1);  lcd.print("  ");}
if(setAlarm<2&&setMode==7){lcd.setCursor(11,1);lcd.print("  ");}
//Alarm
if(setMode==0&&setAlarm==1){lcd.setCursor(6,0);lcd.print("          ");}
if(setMode==0&&setAlarm==2){lcd.setCursor(4,1);lcd.print("  ");}
if(setMode==0&&setAlarm==3){lcd.setCursor(7,1);lcd.print("  ");}
if(setMode==0&&setAlarm==4){lcd.setCursor(10,1);lcd.print("  ");}
}

//SetingJam ,Tanggal,Alarm/Timer
voidsetupClock(void){

    if(setMode==8){

```

```

    lcd.setCursor(0,0);
    lcd.print(F("Set Date Finish "));
    lcd.setCursor(0,1);
    lcd.print(F("Set Time Finish "));
    delay(1000);
    rtc.setTime(hh,mm,ss);
    rtc.setDate(dd,bb,yy);
    lcd.clear();
    setMode=0;
}

if(setAlarm==5){
    lcd.setCursor(0,0);
    lcd.print(F("Set Alarm Finish"));
    lcd.setCursor(0,1);
    lcd.print(F("-EEPROM Updated-"));
    WriteEeprom();
    delay(2000);
    lcd.clear();
    setAlarm=0;
    alarmMode=1;
}

if(setAlarm>0){alarmMode=0;}

if(digitalRead(bt_time)==0&&flag==0){flag=1;
if(setAlarm>0){setAlarm=5;}
else{setMode=setMode+1;}
}

if(digitalRead(bt_alarm)==0&&flag==0){flag=1;
if(setMode>0){setMode=8;}
else{setAlarm=setAlarm+1;}
    lcd.clear();}

if(digitalRead(bt_time)==1&&digitalRead(bt_alarm)==1){flag=0;}

if(digitalRead(bt_up)==0){
    if(setAlarm<2&&setMode==1)hh=hh+1;
    if(setAlarm<2&&setMode==2)mm=mm+1;
    if(setAlarm<2&&setMode==3)ss=ss+1;
    if(setAlarm<2&&setMode==4)set_day=set_day+1;
    if(setAlarm<2&&setMode==5)dd=dd+1;
    if(setAlarm<2&&setMode==6)bb=bb+1;
    if(setAlarm<2&&setMode==7)yy=yy+1;
    //Alarm
    if(setMode==0&&setAlarm==1)mode=1;
    if(setMode==0&&setAlarm==2&&AlarmHH<23)AlarmHH=AlarmHH+1;
    if(setMode==0&&setAlarm==3&&AlarmMM<59)AlarmMM=AlarmMM+1;
    if(setMode==0&&setAlarm==4&&AlarmSS<59)AlarmSS=AlarmSS+1;

if(hh>23)hh=0;
if(mm>59)mm=0;
if(ss>59)ss=0;
if(set_day>7)set_day=0;
if(dd>31)dd=0;
if(bb>12)bb=0;
if(yy>2030)yy=2000;
rtc.setDOW(set_day);
}

```

```

if(digitalRead(bt_down)==0){
    if(setAlarm<2&&setMode==1)hh=hh-1;
    if(setAlarm<2&&setMode==2)mm=mm-1;
    if(setAlarm<2&&setMode==3)ss=ss-1;
    if(setAlarm<2&&setMode==4)set_day=set_day-1;
    if(setAlarm<2&&setMode==5)dd=dd-1;
    if(setAlarm<2&&setMode==6)bb=bb-1;
    if(setAlarm<2&&setMode==7)yy=yy-1;
    //Alarm
    if(setMode==0&&setAlarm==1)mode=0;
    if(setMode==0&&setAlarm==2&&AlarmHH>0)AlarmHH=AlarmHH-1;
    if(setMode==0&&setAlarm==3&&AlarmMM>0)AlarmMM=AlarmMM-1;
    if(setMode==0&&setAlarm==4&&AlarmSS>0)AlarmSS=AlarmSS-1;
if(hh<0)hh=23;
if(mm<0)mm=59;
if(ss<0)ss=59;
if(set_day<0)set_day=7;
if(dd<0)dd=31;
if(bb<0)bb=12;
if(yy<0)yy=2030;
rtc.setDOW(set_day);
}

}

voidsetTimer(){
    //Alarm
if(setMode==0&&setAlarm>0){
    lcd.setCursor(0,0);
    lcd.print("Alarm ");
if(mode==0){lcd.print("Deactivate");}
    else{lcd.print("Activated ");}

    lcd.setCursor(4,1);
    lcd.print((AlarmHH/10)%10);
    lcd.print(AlarmHH%10);
    lcd.print(":");
    lcd.print((AlarmMM/10)%10);
    lcd.print(AlarmMM%10);
    lcd.print(":");
    lcd.print((AlarmSS/10)%10);
    lcd.print(AlarmSS%10);
}
}

voidReadEeprom(){
    AlarmHH=EEPROM.read(1);
    AlarmMM=EEPROM.read(2);
    AlarmSS=EEPROM.read(3);

    mode=EEPROM.read(4);
}

voidWriteEeprom(){
    EEPROM.write(1,AlarmHH);
    EEPROM.write(2,AlarmMM);
    EEPROM.write(3,AlarmSS);

    EEPROM.write(4,mode);
}

```

## **CONCLUSION:**

La réalisation de ce projet nous a apporté, sur un plan personnel, une vision différente du travail en équipe et une nouvelle façon de travailler. En effet, il a fallu diviser le travail en fonction des possibilités et implications de chacun. Nous avons tous été réactifs afin de trouver des créneaux horaires en commun. A l'inverse nous avons appris à travailler en autonomie sur une petite partie d'un projet plus grand. Nous avons su adapter notre comportement au travail réalisé par les autres membres de l'équipe.