

# MANUEL D'UTILISATION **STEAMY STARTER**

Ce présent manuel d'utilisation comporte toutes les informations nécessaires pour apprendre et réaliser les différentes applications possibles grâce à l'utilisation de notre kit STEAMY STARTER.

*«STEAMY, Un apprentissage sur mesure du numérique»*

# SOMMAIRE



**01** Présentation «**STEAMY STARTER**»

**P.01**

**02** Le Logiciel ARDUINO

**P.02**

**03** Tutoriel «**STEAMY STARTER**»

**P.03**

- Objectif
- Prérequis
- Résultats d'apprentissage
- Précautions
- Session 1 : STEAMY STARTER - Diodes électroluminescentes LED
- Session 2 : STEAMY STARTER - Boutons poussoirs
- Session 3 : STEAMY STARTER - Lecteur de valeurs analogiques
- Session 4 : STEAMY STARTER - Buzzer
- Session 5 : STEAMY STARTER - Ecran LCD
- Session 6 : STEAMY STARTER - Module Ultrason
- Session 7 : STEAMY STARTER - Capteur de température et d'humidité

**01**

**Présentation  
«STEAMY STARTER»**

«**STEAMY STARTER**» est un ensemble de cartes Arduino Uno et d'un shield Arduino Uno sur lequel il y a plusieurs composants électroniques.

Avec la «**STEAMY STARTER**», les apprenants auront à assimiler les notions de systèmes, d'entrées et de sorties d'un système, d'entrées digitales et analogiques, de sorties digitales et analogiques, de protocoles de communications ainsi que le fonctionnement de petits systèmes intelligents.

Le shield présente l'avantage d'avoir les composants électroniques et capteurs déjà soudés ou à raccorder simplement sur elle. Plusieurs capteurs sont prévus sur cette carte pour recevoir des données de plusieurs types et provenant du milieu ambiant.

«**STEAMY STARTER**» interagit avec son environnement en fonction du programme qu'on lui implante. Pour cela, il est capable de collecter des informations grâce à ses capteurs et de réaliser des notifications grâce à des voyants visuels et sonores.

**Voyants visuels et sonores :**

«**STEAMY STARTER**» est vendu avec plusieurs voyants visuels et sonores :

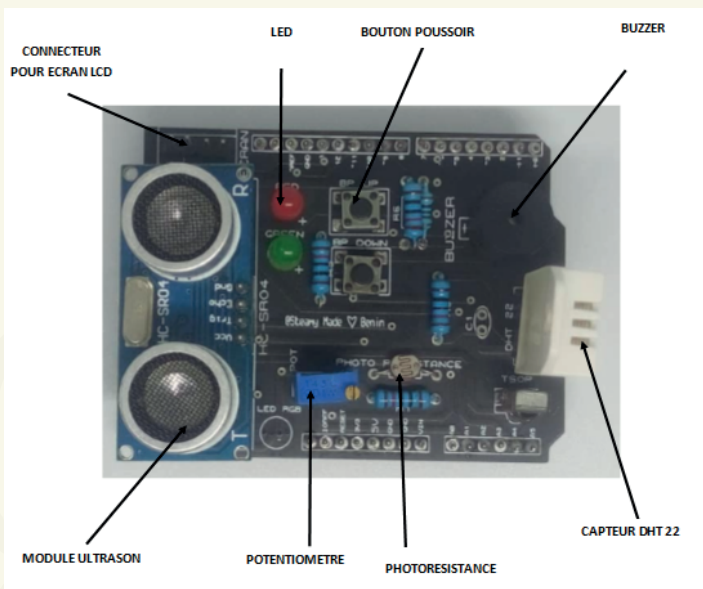
- un écran LCD utilisant le protocole de communication I2C qui affiche des informations.
- un buzzer grâce auquel il peut émettre des sons à des fréquences variées.
- deux diodes électroluminescentes DEL (rouge et verte) qui s'allument avec une intensité de lumière variée.

**Boutons et capteurs :**

Pour interagir avec son environnement et y recueillir des informations, on retrouve sur le shield:

- un capteur de température et d'humidité qui renseigne sur la température et d'humidité ambiante.
- un module à ultrasons qui permet de mesurer des distances.
- deux boutons poussoirs à appuyer pour déclencher une action.
- une photorésistance qui fonctionne comme un capteur de lumière et qui renseigne la luminosité ambiante.
- un potentiomètre qui fonctionne comme un capteur de position.

## LOCALISER LES CAPTEURS ET ACTIONNEURS



The background is a solid orange color. A large white triangle is positioned on the left side, pointing upwards. The triangle's base is at the bottom left, and its apex is at the top center. The text is located within the white triangle.

**02**

**Le logiciel ARDUINO**

Arduino IDE est une application multiplateforme qui est écrite dans des fonctions de C et C++. Il est utilisé pour écrire et télécharger des programmes sur des cartes compatibles Arduino. Il fournit plusieurs bibliothèques sous forme de classes (programmation orienté objet) qui sont utilisées notamment pour les différents capteurs et modules compatibles avec Arduino. Il utilise le programme **avrdude** pour convertir le code exécutable en un fichier texte au codage hexadécimal qui est chargé dans la carte Arduino par un programme de chargement dans le firmware de la carte.

Téléchargez et installez à l'adresse web <https://www.arduino.cc/en/software> **Arduino IDE**

**N.B.:** Veuillez à télécharger et installer l'IDE Arduino compatible avec votre système d'exploitation.

MATÉRIELLOGICIEL ▼DOCUMENTATION ▼COMMUNAUTÉ ▼BLOGSUR



## Arduino IDE 1,8,13

Le logiciel Open-source Arduino (IDE) facilite l'écriture du code et le téléchargement sur le plateau. Ce logiciel peut être utilisé avec n'importe quelle planche Arduino.

Consultez la page **Démarrer pour** les instructions d'installation.

SOURCE CODE

Le développement actif du logiciel Arduino est **hébergé par GitHub**. Voir les instructions pour la **construction du code**. Les dernières archives de code source de version sont **disponibles ici**. Les archives sont signées PGP afin qu'elles puissent être vérifiées à l'aide de cette clé gpg.

### OPTIONS DE TÉLÉCHARGEMENT

**Windows** Gagnez 7 et plus nouveau  
**Windows** Fichier ZIP

**Application Windows** Gagnez 8,1 ou 10 

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 ou plus nouveau

Notes de libération Checksums (sha512)

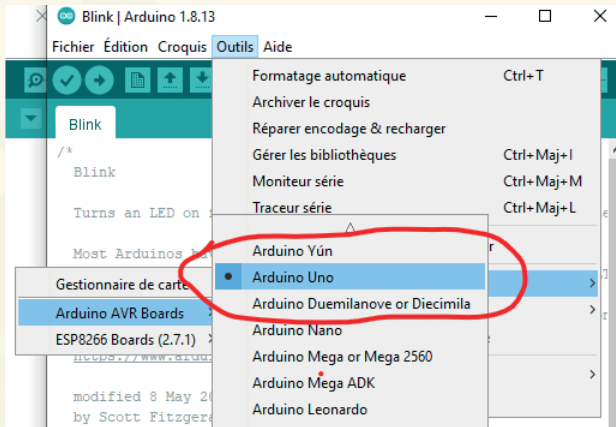
## TÉLÉVERSER UN PROGRAMME SUR LA CARTE ARDUINO

Téléverser un programme sur une carte Arduino consiste à charger sur la carte un programme informatique. Ce programme est converti en un autre programme plus complexe mais compréhensible par la carte. C'est lui qui contient le fonctionnement voulu, les interactions entre capteurs et voyants ...

Pour téléverser un programme sur la carte Arduino, il faut :

- ouvrir le fichier du programme avec le logiciel Arduino ;
- connecter la carte Arduino à l'ordinateur à travers un port USB ;
- choisir le type de carte Arduino utilisé (Arduino Uno). Pour ce faire, cliquez successivement

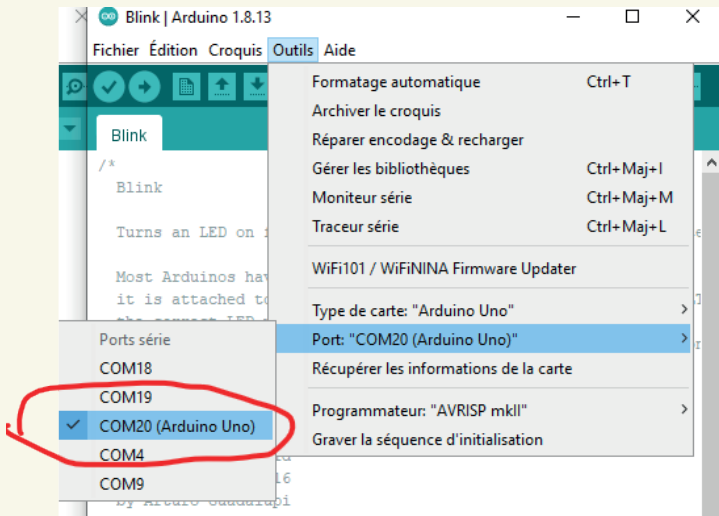
sur **Outils > Type de carte > Arduino AVR Boards > Arduino Uno** comme le montre la figure ci-dessous :





## TÉLÉVERSER UN PROGRAMME SUR LA CARTE ARDUINO

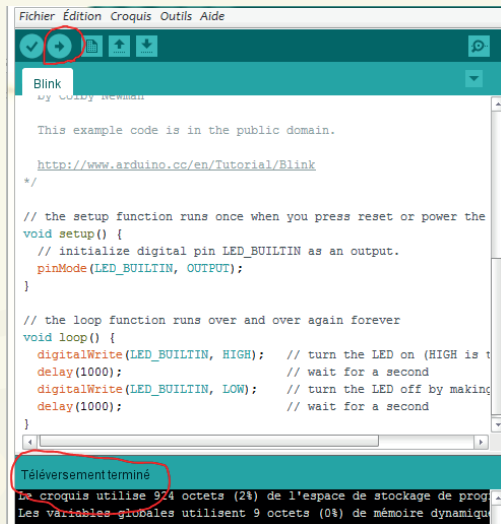
- choisir le port de communication entre l'ordinateur et la carte Arduino. Pour ce faire, cliquer successivement sur Outils- Port- COM.. comme sur la figure ci-dessous :



Attention le nom du port de communication varie d'un ordinateur à un autre et d'un port USB à un autre. Ici le port est le COM20.

## TÉLÉVERSER UN PROGRAMME SUR LA CARTE ARDUINO

- téléverser le code en cliquant sur l'icône correspondant dans le logiciel :



Une fois le téléversement terminé, le logiciel fait une notification comme sur la figure ci-dessus. En cas d'erreur de téléversement, vérifiez que vous avez bien copié le programme, bien choisi la carte et le port de communication. Au cas où l'erreur persiste, débrancher la carte Arduino et rebranchez la après quelques secondes.

**03**

**Tutoriel «STEAMY STARTER»**



## OBJECTIF

Cette section se compose de **7 tutoriels**, qui peuvent être divisés en plusieurs parties. Ils permettent de prendre en main le fonctionnement des différents capteurs et voyants de “STEAMY STARTER”. Certains exemples mettent en exergue des cas d'utilisations combinées des capteurs et voyants pour de petites applications réelles.

## PRÉREQUIS

Les connaissances fondamentales sur le fonctionnement de Arduino IDE et la compétence de codage sont essentielles à ce tutoriel. Par conséquent, veuillez vous assurer d'avoir terminé le didacticiel sur le logiciel ci-dessus et vous assurer que tout va bien.

## RÉSULTAT D'APPRENTISSAGE

- Être capable d'utiliser Arduino IDE pour écrire du code pour le shield de “STEAMY STARTER” ;
- Être capable d'identifier les capteurs et voyants du shield et leurs applications ;
- Être en mesure d'imaginer d'autres applications et d'autres fonctionnements faisant intervenir les capteurs et voyants du shield ;

## PRÉCAUTIONS

- Emboîter parfaitement et dans le bon sens le shield dans la carte Arduino ;
- Connecter la carte Arduino à l'ordinateur et assurer que la connexion s'est établie correctement

## SESSION 1 : STEAMY STARTER - Diodes électroluminescentes LED

### Objectif

Clignotement d'une LED et changement de luminosité

### Exemple 1 : Utilisez la logique numérique haut / bas pour faire «clignoter» la LED verte.

- Copiez et collez le code suivant dans l'IDE Arduino

```
#define led_green 12

void setup(){
  //set leds as output
  pinMode(led_green , OUTPUT);
}

void loop(){
  digitalWrite(led_green, HIGH);
  delay(500);
  //wait 500 ms

  digitalWrite(led_green,LOW);
  delay(500);
  //wait 500 ms
}
```

- Téléversez le code sur la carte Arduino.
- Observez le résultat. Une fois le téléchargement terminé, vous devriez voir la LED verte clignoter, c'est-à-dire s'allumer et s'éteindre toutes les demi-secondes .



## SESSION 1 : STEAMY STARTER - Diodes électroluminescentes LED

### Exemple 2 : Utilisez la logique numérique PWM pour changer la luminosité de la LED rouge

- Copiez et collez le code suivant dans l'IDE Arduino

```
#define led_red 11

void setup(){
  //set leds as output
  pinMode(led_red , OUTPUT);
}

void loop(){
  for(int i=256;i>-1;i-=25)
  {
    analogWrite(led_red,i);
    delay(1000);
    //wait 1000 ms
  }
  for(int i=0;i<256;i+=25)
  {
    analogWrite(led_red,i);
    delay(1000);
    //wait 1000 ms
  }
}
```

- Téléverser le code sur la carte Arduino.
- Observez le résultat. Une fois le téléchargement terminé, vous devriez voir la LED verte clignoter, c'est-à-dire s'allumer et s'éteindre toutes les demi-secondes .

## SESSION 2 : STEAMY STARTER - Boutons poussoirs

### Objectif

Voir le changement d'état du bouton dans le moniteur série et commander une LED via les deux boutons poussoirs

### Exemple 1 : Utilisez le moniteur série pour avoir la notification de changement d'état du bouton

- Copiez et collez le code suivant dans l'IDE Arduino

```
#define button 13

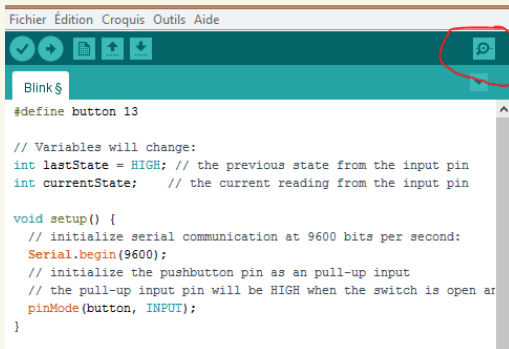
// Variables will change:
int lastState = HIGH; // the previous state from the input pin
int currentState; // the current reading from the input pin

void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // initialize the pushbutton pin as an pull-up input
  //the pull-up input pin will be HIGH when the switch is open and LOW when the switch is closed.
  pinMode(button, INPUT); }
void loop() {
  // read the state of the switch/button:
  delay(10);
  currentState = digitalRead(button);
  if(lastState == LOW && currentState == HIGH)
    Serial.println("L'etat du bouton a change du bas vers le haut.");
  // save the the last state
  lastState = currentState;
}
```

## SESSION 2 : STEAMY STARTER - Boutons poussoirs

**Exemple 1 : Utilisez le moniteur série pour avoir la notification de changement d'état du bouton**

● Observez le résultat. Une fois le téléchargement terminé, ouvrez le moniteur série. Pour l'ouvrir, il suffit de cliquer sur son icône dans le logiciel Arduino comme le montre la figure ci-dessous.

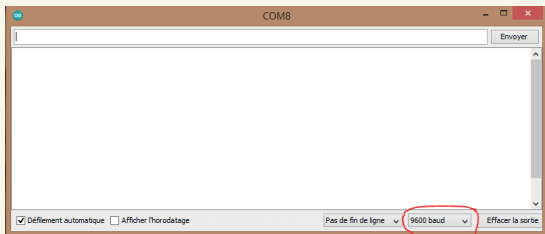


● Observez le résultat. Une fois le téléchargement terminé, ouvrez le moniteur série. Pour l'ouvrir, il suffit de cliquer sur son icône dans le logiciel Arduino comme le montre la figure ci-dessous.



**SESSION 2 : STEAMY STARTER - Boutons poussoirs**

**Exemple 1 : Utilisez le moniteur série pour avoir la notification de changement d'état du bouton**



Appuyez sur le bouton 1 du shield et relâchez ensuite. Vous devriez recevoir la notification dans le moteur série comme dans la figure ci-dessous :



## SESSION 2 : STEAMY STARTER - Boutons poussoirs

### Exemple 2 : Utilisez le bouton 1 pour allumer une LED et le bouton 2 pour l'éteindre

- Copiez et collez le code suivant dans l'IDE Arduino

```
#define led 12
#define button_1 13
#define button_2 2

// Variables will change:
int lastState_1 = HIGH; // the previous state from the button_1
int currentState_1; // the current reading from the button_1

int lastState_2 = LOW; // the previous state from the button_2
int currentState_2; // the current reading from the button_2

void setup() {
  pinMode(led, OUTPUT);

  // initialize the pushbutton pin1 as an pull-up input
  /*the pull-up input pin will be HIGH when the switch is open
  and LOW when the switch is closed.*/
  pinMode(button_1, INPUT);

  // initialize the pushbutton pin2 as an pull-down input
  /* the pull-up input pin will be LOW when the switch is open
  and HIGH when the switch is closed.*/
  pinMode(button_2, INPUT);
  digitalWrite(led, LOW); //Shut down the led first
}

void loop() {
  // read the state of the buttons:
  currentState_1 = digitalRead(button_1);
  currentState_2 = digitalRead(button_2);

  if(lastState_1 == LOW && currentState_1 == HIGH)
    digitalWrite(led, HIGH);

  if(lastState_2 == HIGH && currentState_2 == LOW)
    digitalWrite(led, LOW);

  // save the the last state
  lastState_1 = currentState_1;
  lastState_2 = currentState_2;
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous pouvez allumer la LED verte avec le bouton 1 et l'éteindre avec le bouton 2.

## SESSION 3 : STEAMY STARTER - Lecture de valeurs analogiques

### Exemple 1 : Lire les valeurs analogiques provenant du potentiomètre

- Copiez et collez le code suivant dans l'IDE Arduino

```
#define potentiometre A1
```

```
void setup(){
```

```
  Serial.begin(9600); // Communication started with 9600 baud
```

```
}
```

```
void loop(){
```

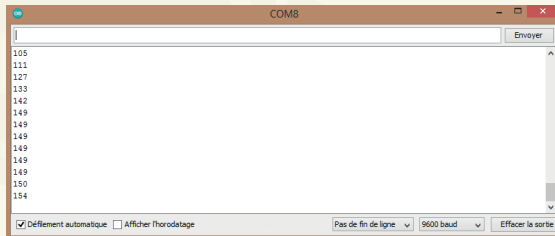
```
  int potentiometre_value=analogRead(potentiometre); // Incoming analog signal read and appointed  
  //sensor
```

```
  Serial.println(potentiometre_value); //Wrote serial port
```

```
  delay(1000);
```

```
}
```

- Observez le résultat. Une fois le téléchargement terminé, ouvrez le moniteur série. Vous devriez voir des valeurs entre 0 et 1023 défilier. Ces valeurs correspondent à la position du potentiomètre. Faites le tourner dans un sens et vous verrez les valeurs croître ou décroître. Tournez dans le sens contraire et vous verrez l'effet contraire.





## SESSION 3 : STEAMY STARTER - Lecture de valeurs analogiques

### Exemple 2 : Lire les valeurs analogiques provenant de la photorésistance

Une photorésistance est une résistance dont la valeur change en fonction de la quantité de lumière qu'elle reçoit.

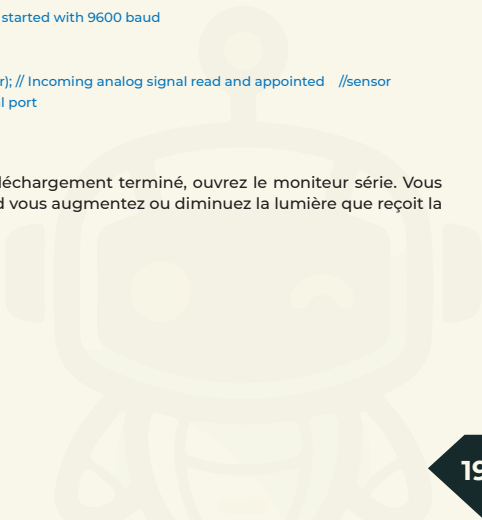
- Copiez et collez le code suivant dans l'IDE Arduino

```
#define light_sensor A0

void setup(){
  Serial.begin(9600); // Communication started with 9600 baud
}

void loop(){
  int light_value=analogRead(light_sensor); // Incoming analog signal read and appointed //sensor
  Serial.println(light_value); //Wrote serial port
  delay(1000);
}
```

- Observez le résultat. Une fois le téléchargement terminé, ouvrez le moniteur série. Vous verrez les valeurs affichées changer quand vous augmentez ou diminuez la lumière que reçoit la photorésistance.



## SESSION 3 : STEAMY STARTER - Lecture de valeurs analogiques

### Exemple 3 : Changer l'éclat d'une LED avec la photorésistance

- Copiez et collez le code suivant dans l'IDE Arduino

```
#define light_sensor A0
#define led_red 11

void setup(){
  pinMode(led_red , OUTPUT);
}
void loop(){
  int light_value=analogRead(light_sensor); // Incoming analog signal read and appointed //sensor
  int led_value = map(light_value, 0 , 1023 , 0 , 255);
  analogWrite(led_red,led_value);
  delay(1000);
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous verrez la LED rouge s'allumer. Vous pouvez changer l'éclat de la LED en jouant sur la lumière qui frappe la photorésistance.

### Exemple 4 : Changer la vitesse du clignotement avec le potentiomètre

Copiez et collez le code suivant dans l'IDE Arduino

```
#define potentiometre A1
#define led_green 12

void setup(){
  pinMode(led_green , OUTPUT);
}
```



## SESSION 3 : STEAMY STARTER - Lecture de valeurs analogiques

### Exemple 4 : Changer la vitesse du clignotement avec le potentiomètre - Suite

```
void loop(){  
  int pot_value=analogRead(potentiometre ); // Incoming analog signal read and appointed //sensor  
  int delay_value = map(pot_value, 0 , 1023 , 0 , 2000);  
  digitalWrite(led_green,HIGH);  
  delay(delay_value);  
  digitalWrite(led_green,LOW);  
  delay(delay_value);  
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous verrez la LED verte clignoter. Vous pouvez changer la vitesse du clignotement de la LED en tournant le potentiomètre.



## SESSION 4 : STEAMY STARTER - Buzzer

### Exemple 1 : Faire bourdonner un buzzer en utilisant logique numérique haut / bas

- Copiez et collez le code suivant dans l'IDE Arduino

```
//assign buzzer as pin 9
#define buzzer 9

void setup()
{
  //set buzzer as output
  pinMode(buzzer, OUTPUT);
}
void loop()
{
  //turn on buzzer(set logic level high)
  digitalWrite(buzzer, HIGH);
  //wait 1s
  delay(1000);
  //turn off buzzer(set logic level low)
  digitalWrite(buzzer, LOW);
  //wait 1s
  delay(1000);
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous entendrez le buzzer émettre un son. Ce son n'est pas très audible. C'est le premier test.

## SESSION 4 : STEAMY STARTER - Buzzer

## Exemple 2 : Jouer des notes de musique avec le buzzer en utilisant des fréquences

- Copiez et collez le code suivant dans l'IDE Arduino

```
//assign buzzer as pin 9
#define buzzer 9

void setup()
{
}

void loop()
{
  /* tone(pin, frequency, duration) */
  //set buzzer pin to play 264Hz for 300ms
  tone(buzzer, 262, 300);
  //wait 1s
  delay(1000);
  //set buzzer pin to play 297Hz for 300ms
  tone(buzzer, 294, 300);
  //wait 1s
  delay(1000);
  //set buzzer pin to play 330Hz for 300ms
  tone(buzzer, 330, 300);
  //wait 1s
  delay(1000);
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous entendrez le buzzer émettre trois sons différents en boucle. Ecoutez-bien ! Vous reconnaîtrez les trois premières notes de la musique Do - Re - Mi ...



## SESSION 4 : STEAMY STARTER - Buzzer

### Exemple 3 : Jouer des notes d'une musique célèbre dans le monde

- Copiez et collez le code suivant dans l'IDE Arduino

```
// initialise the frequency of the notes
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_C4 262
#define NOTE_D4 294
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_G4 392
#define NOTE_CS 523

//assign buzzer as pin 9
#define buzzer 9

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_F4, NOTE_E4,
  NOTE_C4, NOTE_C4, NOTE_D4, NOTE_C4, NOTE_G4, NOTE_F4,
  NOTE_C4, NOTE_C4, NOTE_CS, NOTE_A4, NOTE_F4, NOTE_E4, NOTE_D4,
  NOTE_AS4, NOTE_AS4, NOTE_A4, NOTE_F4, NOTE_G4, NOTE_F4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  8, 8, 4, 4, 4, 2,
  8, 8, 4, 4, 4, 2,
  8, 8, 4, 4, 4, 4, 4,
  8, 8, 4, 4, 4, 2,
};
```

## SESSION 4 : STEAMY STARTER - Buzzer

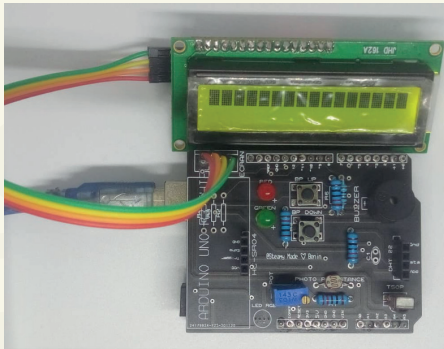
## Exemple 3 : Jouer des notes d'une musique célèbre dans le monde - Suite

```
void setup() {  
  // iterate over the notes of the melody:  
}  
  
void loop() {  
  // no need to repeat the melody.  
  for (int thisNote = 0 ; thisNote < 25 ; thisNote++) {  
  
    // to calculate the note duration, take one second divided by the note type.  
    //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.  
    int noteDuration = 1000 / noteDurations[thisNote];  
    tone(buzzer, melody[thisNote], noteDuration);  
  
    // to distinguish the notes, set a minimum time between them.  
    int pauseBetweenNotes = noteDuration * 1.30;  
    delay(pauseBetweenNotes);  
    noTone(buzzer);  
  }  
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous entendrez la musique. Vous la connaissez, n'est-ce pas ? C'est la célèbre musique de la chanson du ""Joyeux anniversaire""

## SESSION 5 : STEAMY STARTER - Ecran LCD

L'écran LCD nous servira à afficher des messages. Avant toute utilisation, vous devez la brancher au shield arduino. L'emplacement des connecteurs pour l'écran est indiqué sur le shield. Une fois branchée au shield, l'écran doit s'allumer. Si elle ne s'allume pas, changer l'ordre de branchement des 4 fils dans le sens contraire.



### Exemple 1 : Afficher un message à l'écran

- Copiez et collez le code suivant dans l'IDE Arduino et téléverser le :

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display
```



## SESSION 5 : STEAMY STARTER - Ecran LED

### Exemple 1 : Afficher un message à l'écran - Suite

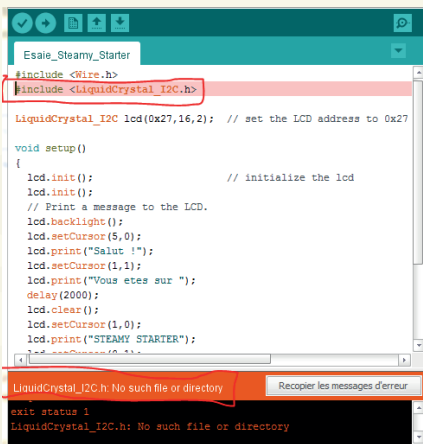
```
void setup()
{
  lcd.init();           // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(5,0);
  lcd.print("Salut !");
  lcd.setCursor(1,1);
  lcd.print("Vous etes sur ");
  delay(2000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("STEAMY STARTER");
  lcd.setCursor(0,1);
  lcd.print("Amusez-vous bien");
}
void loop()
{
}
```

Je suis sûr que vous n'êtes pas parvenu à téléverser le code. Vous avez sans doute une erreur comme le montre la figure ci-dessous :



## SESSION 5 : STEAMY STARTER - Ecran LED

## Exemple 1 : Afficher un message à l'écran - Suite



```
Esale_Steamy_Starter

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27

void setup()
{
  lcd.init();           // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(5,0);
  lcd.print("Salut !");
  lcd.setCursor(1,1);
  lcd.print("Vous etes sur ");
  delay(2000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("STEAMY STARTER");
}

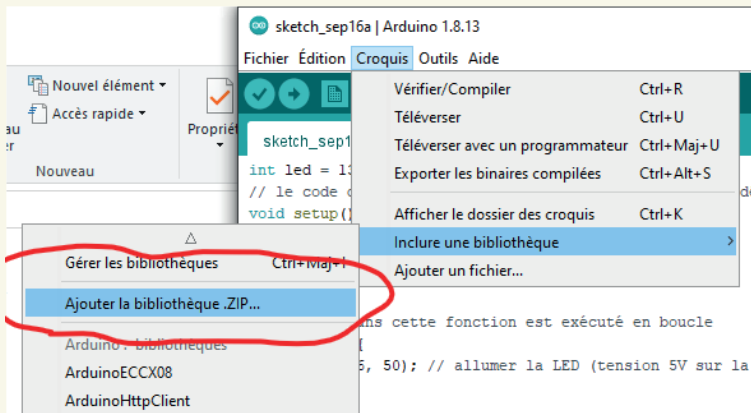
exit status 1
LiquidCrystal_I2C.h: No such file or directory
```

• En réalité, il manque des fichiers à votre logiciel Arduino pour comprendre le code. Pour arriver à téléverser ce code, il faut ajouter les fichiers au logiciel: on parle d'inclure une bibliothèque. Pour cela, il faut :

- télécharger la bibliothèque à l'adresse [www.tiaes.co/steamy](http://www.tiaes.co/steamy). Il s'agit d'un fichier .zip
- cliquer successivement dans le logiciel sur Croquis-Inclure une bibliothèque. Ajouter la bibliothèque .ZIP comme le montre la figure ci-dessous :

## SESSION 5 : STEAMY STARTER - Ecran LED

## Exemple 1 : Afficher un message à l'écran - Suite



- naviguer dans votre explorateur de fichiers, choisissez le fichier .zip et ouvrez-le.

La bibliothèque est maintenant ajoutée au logiciel. Une notification est donnée par le logiciel comme sur la figure ci-dessous:

## SESSION 5 : STEAMY STARTER - Ecran LED

## Exemple 1 : Afficher un message à l'écran - Suite

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to

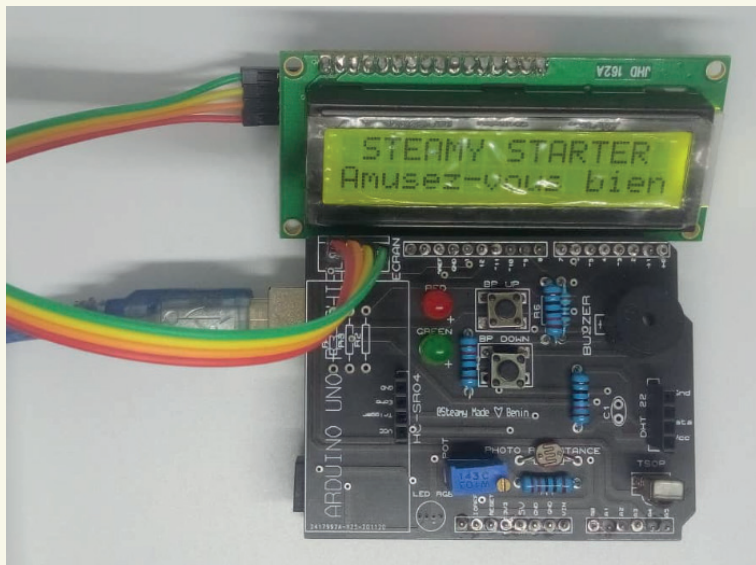
void setup()
{
  lcd.init();                      // initialize the lcd
  lcd.init();
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(5,0);
  lcd.print("Salut !");
  lcd.setCursor(1,1);
  lcd.print("Vous etes sur ");
  delay(2000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("STEAMY STARTER");
}
```

La bibliothèque a été ajoutée à votre dossier de bibliothèques. Veuillez regarder  
Compilation terminated.  
exit status 1  
LiquidCrystal\_I2C.h: No such file or directory

- Téléverser le code à nouveau. Cela devrait marcher maintenant
- Observez le résultat. Une fois le téléchargement terminé, vous verrez un message affiché à l'écran en deux temps.

## SESSION 5 : STEAMY STARTER - Ecran LED

## Exemple 1 : Afficher un message à l'écran - Suite





## SESSION 5 : STEAMY STARTER - Ecran LCD

### Exemple 2 : Saisir et afficher un message à l'écran

- Copiez et collez le code suivant dans l'IDE Arduino et téléversez le :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup() {
  lcd.init();           // initialize the lcd
  lcd.backlight();
  Serial.begin(9600);
}

void loop() {
  // when characters arrive over the serial port...
  if (Serial.available()) {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0) {
      // display each character to the LCD
      lcd.write(Serial.read());
    }
  }
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous pouvez afficher votre propre message à l'écran. Il suffit de le saisir dans le moniteur série. Ouvrez-le. Vous savez déjà le faire. Saisissez votre message et validez en appuyant sur la touche "Enter" de votre clavier.



## SESSION 6 : STEAMY STARTER - Module Ultrason

Le module ultrason est un capteur de distance. Il permet d'évaluer la distance entre un obstacle et lui. Pour l'utiliser avec le logiciel Arduino, il faut également une bibliothèque spéciale. Télécharger le à cette adresse [www.](http://www.) et ajouter le au logiciel comme dans la session précédente.

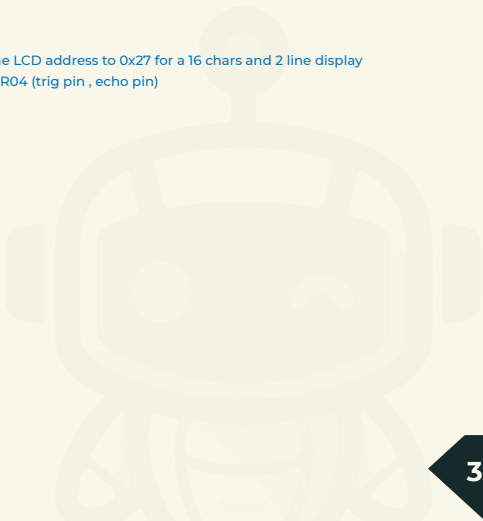
### Exemple 1 : Mesurer une distance et l'afficher l'écran LCD

- Copiez et collez le code suivant dans l'IDE Arduino et téléverser le :

```
#include <HCSR04.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

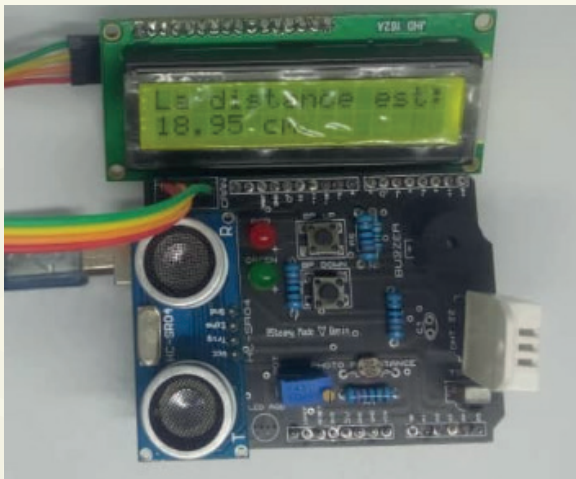
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display
HCSR04 hc(7,8); //initialisation class HCSR04 (trig pin , echo pin)

void setup()
{
  lcd.init();           // initialize the lcd
  lcd.backlight();
}
void loop()
{
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("La distance est:");
  lcd.setCursor(0,1);
  lcd.print(hc.dist());
  lcd.print(" cm.");
  delay(1000);
}
```



**SESSION 6 : STEAMY STARTER - Module Ultrason****Exemple 1 : Mesurer une distance et l'afficher l'écran LCD - Suite**

- Vous l'aurez sans doute compris, l'ECRAN LCD sera utilisé encore ici. Observez le résultat. Une fois le téléchargement terminé, vous pouvez voir affiché à l'écran la distance mesurée par le capteur. Mettez devant le capteur un obstacle et faites varier la distance. Elle changera automatiquement à l'écran.





## SESSION 6 : STEAMY STARTER - Module Ultrason

### Exemple 2 : Changer la vitesse du clignotement avec le module ultrason

- Copiez et collez le code suivant dans l'IDE Arduino et téléverser le :

```
#include <HCSR04.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define led_green 12
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display
HCSR04 hc(7,8);//initialisation class HCSR04 (trig pin , echo pin)

void setup() {
  lcd.init();           // initialize the lcd
  lcd.backlight();
  pinMode(led_green , OUTPUT);
}

void loop() {
  int distance = hc.dist();
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("La distance est:");
  lcd.setCursor(0,1);
  lcd.print(distance);
  lcd.print(" cm.");
  int delay_value = map(distance, 0 , 30 , 0 , 1000);
  digitalWrite(led_green,HIGH);
  delay(delay_value);
  distance = hc.dist();
  delay_value = map(distance, 0 , 30 , 0 , 1000);
  digitalWrite(led_green,LOW);
  delay(delay_value);
}
```

## SESSION 6 : STEAMY STARTER - Module Ultrason

### Exemple 2 : Changer la vitesse du clignotement avec le module ultrason - Suite

- Vous l'aurez sans doute compris, l'écran LCD sera utilisé encore ici. Observez le résultat. Une fois le téléchargement terminé, vous pouvez voir affiché à l'écran la distance mesurée par le capteur. Aussi la vitesse de clignotement de la LED verte change en fonction de la distance mesurée par le capteur. Mettez devant le capteur un obstacle et attendez quelques secondes. Faites varier la distance pour faire varier la vitesse de clignotement de la LED.

### Exemple 3 : Capteur de proximité avec le module ultrason

- Copiez et collez le code suivant dans l'IDE Arduino et téléverser le :

```
#include <HCSR04.h>
#define led_red 11
#define buzzer 9
HCSR04 hc(7,8);//initialisation class HCSR04 (trig pin , echo pin)

void setup() {
  pinMode(led_red , OUTPUT);
  pinMode(buzzer , OUTPUT);
}

void loop() {
  if (hc.dist()<10) {
    digitalWrite(led_red,HIGH);
    tone(buzzer, 1000, 10);
  }
  else {
    digitalWrite(led_red,LOW);
    digitalWrite(buzzer,LOW);
  }
}
```



## SESSION 6 : STEAMY STARTER - Module Ultrason

### Exemple 3 : Capteur de proximité avec le module ultrason - Suite

- Observez le résultat. Une fois le téléchargement terminé, le module ultrason est utilisé comme un capteur de proximité. Dès que votre main ou tout autre objet est proche du module, cela est signalé automatiquement. La LED rouge s'allume et le buzzer sonne pour signaler.



## SESSION 7 : STEAMY STARTER - Capteur de température et humidité

Le capteur DHT22 est un capteur de température et d'humidité. Il permet d'évaluer la température et l'humidité du milieu ambiant. Pour l'utiliser avec le logiciel Arduino, il faut également une bibliothèque spéciale. Télécharger le à cette adresse [www.adafruit.com/products/285](http://www.adafruit.com/products/285) et ajouter le au logiciel comme dans la session précédente.

### Exemple 1 : Mesurer la température et l'humidité du milieu et l'afficher l'écran LCD

- Copiez et collez le code suivant dans l'IDE Arduino et téléverser le :

```
#include "DHT.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 4 // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

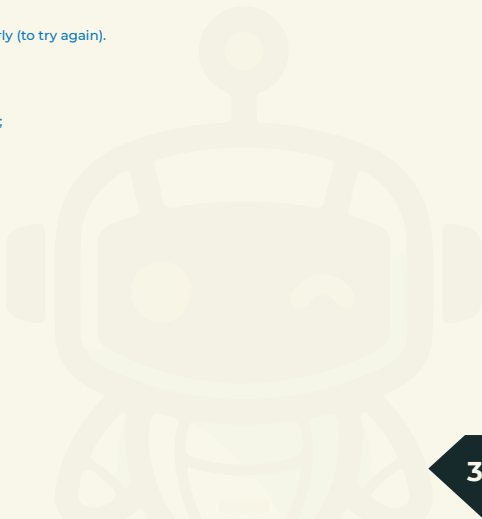
void setup() {
  lcd.init();           // initialize the lcd
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("DHT 22 test!...");
  dht.begin();
  delay(2000);
  lcd.clear();
}
```



## SESSION 7 : STEAMY STARTER - Capteur de température et humidité

### Exemple 1 : Mesurer la température et l'humidité du milieu et l'afficher l'écran LCD - Suite

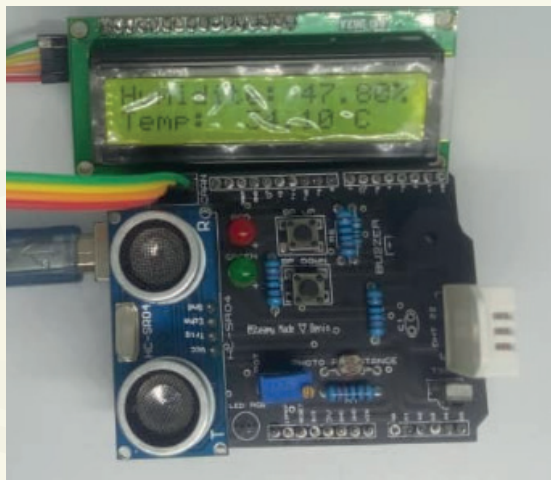
```
void loop() {  
  // Wait a few seconds between measurements.  
  delay(2000);  
  // Reading temperature or humidity takes about 250 milliseconds!  
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)  
  float h = dht.readHumidity();  
  // Read temperature as Celsius (the default)  
  float t = dht.readTemperature();  
  // Check if any reads failed and exit early (to try again).  
  if (isnan(h) || isnan(t) ) {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Le capteur a un problème.");  
    lcd.setCursor(0,1);  
    lcd.print("probleme.");  
    return;  
  }  
  lcd.setCursor(0,0);  
  lcd.print(F("Humidite: "));  
  lcd.setCursor(10,0);  
  lcd.print(h);  
  lcd.setCursor(15,0);  
  lcd.print(F("%"));  
  lcd.setCursor(0,1);  
  lcd.print(F("Temp: "));  
  lcd.setCursor(7,1);  
  lcd.print(t);  
  lcd.setCursor(12,1);  
  lcd.print(F(" C"));  
}
```





**SESSION 7 : STEAMY STARTER - Capteur de température et humidité****Exemple 1 : Mesurer la température et l'humidité du milieu et l'afficher l'écran LCD - Suite**

- Observez le résultat. Une fois le téléchargement terminé, vous pouvez voir affiché à l'écran la température et l'humidité de l'air ambiant. Approchez une source de chaleur et vous verrez la température augmenter et l'humidité diminuer.





## SESSION 7 : STEAMY STARTER - Capteur de température et humidité

### Exemple 2 : Capteur de seuil de température

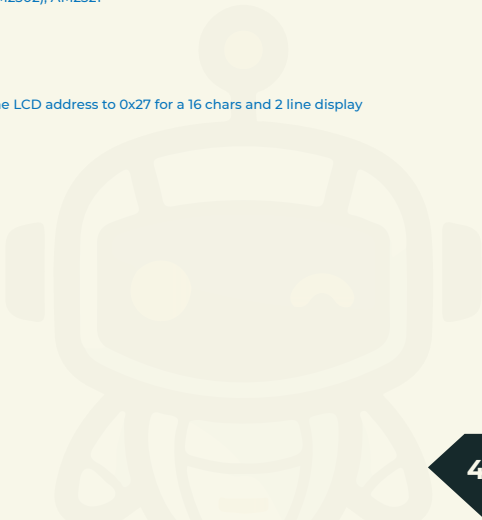
- Copiez et collez le code suivant dans l'IDE Arduino et téléverser le :

```
#include "DHT.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 4 // Digital pin connected to the DHT sensor
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
#define led_green 12
#define led_red 11

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,16,2); // set the LCD address to 0x27 for a 16 chars and 2 line display

void setup() {
  lcd.init();           // initialize the lcd
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("DHT 22 test!...");
  dht.begin();
  delay(2000);
  lcd.clear();
  pinMode(led_green , OUTPUT);
  pinMode(led_red , OUTPUT);
  digitalWrite(led_green, LOW);
  digitalWrite(led_red, LOW);
}
```



## SESSION 7 : STEAMY STARTER - Capteur de température et humidité

### Exemple 2 : Capteur de seuil de température - Suite

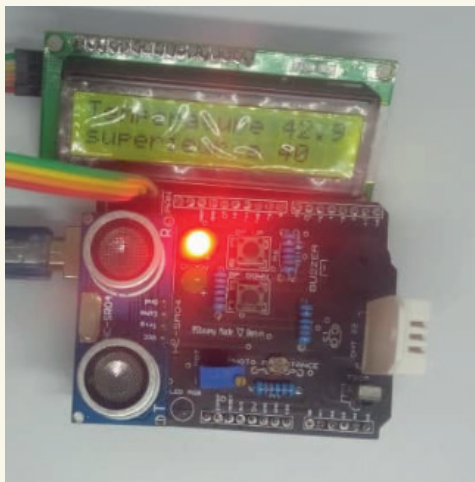
```
void loop() {  
  // Wait a few seconds between measurements.  
  delay(2000);  
  // Reading temperature or humidity takes about 250 milliseconds!  
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)  
  float h = dht.readHumidity();  
  // Read temperature as Celsius (the default)  
  float t = dht.readTemperature();  
  // Check if any reads failed and exit early (to try again).  
  if (isnan(h) || isnan(t) ) {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Le capteur a un problème.");  
    lcd.setCursor(0,1);  
    lcd.print("probleme.");  
    return;  
  }  
  if (t<40 ){  
    lcd.setCursor(0,0);  
    lcd.print(F("Temperature "));  
    lcd.print(t);  
    lcd.setCursor(0,1);  
    lcd.print(F("inferieur a 40"));  
    digitalWrite(led_green, HIGH);  
    digitalWrite(led_red, LOW);  
  }  
}
```

```
else  
{  
  lcd.setCursor(0,0);  
  lcd.print(F("Temperature "));  
  lcd.print(t);  
  lcd.setCursor(0,1);  
  lcd.print(F("superieur a 40"));  
  digitalWrite(led_green,LOW);  
  digitalWrite(led_red,HIGH);  
}  
}
```

## SESSION 7 : STEAMY STARTER - Capteur de température et humidité

### Exemple 2 : Capteur de seuil de température - Suite

- Observez le résultat. Une fois le téléchargement terminé, vous pouvez voir affiché à l'écran la température de l'air ambiant. L'écran indique si la température est inférieure ou supérieure à 40 degré Celsius. Dans le premier cas, la LED verte s'allume. Dans le second cas, la LED rouge s'allume.



Fièremment propulsé par

