



# MANUEL D'UTILISATION

# STEAMY PRO

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 PRO.

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

Fièrement propulsé par



# SOMMAIRE

01	Présentation «STEAMY PRO»	P.01
02	Le Logiciel ARDUINO	P.02
03	Tutoriel «STEAMY PRO»	P.03
	● Objectif	
	● Prérequis	
	● Résultats d'apprentissage	
	● Précautions	
	● Session 1 : STEAMY PRO -	
	● Session 2 : STEAMY PRO -	
	● Session 3 : STEAMY PRO -	
	● Session 4 : STEAMY PRO -	
	● Session 5 : STEAMY PRO -	
	● Session 6 : STEAMY PRO -	
	● Session 7 : STEAMY PRO -	

01

# Présentation STEAMY PRO

## 01 Présentation «STEAMY PRO»

“STEAMY PRO” est un système de prototypage modulaire composé d'une unité de base et de divers modules avec connecteur normalisé.

L'unité de base est l'ensemble d'une carte Arduino Uno (carte à microprocesseur qui permet de communiquer, de traiter et de contrôler l'entrée ou la sortie des modules “STEAMY PRO”) et d'un shield Arduino (carte électronique ayant la même forme et les mêmes dimensions qu'une carte Arduino Uno classique et pouvant s'emboîter parfaitement dans cette dernière). Chaque module “STEAMY PRO” s'adresse généralement à une seule fonction, allant d'un simple bouton à un capteur de distance à ultrason. Le connecteur “STEAMY PRO” standardisé permet à l'utilisateur d'assembler des unités “STEAMY PRO” avec une approche de blocs de construction. Il est facile à assembler ou à démonter, ce qui simplifie le système d'apprentissage pour l'expérimentation, la construction et le prototypage.



Ensemble des cartes du KIT STEAMY PRO

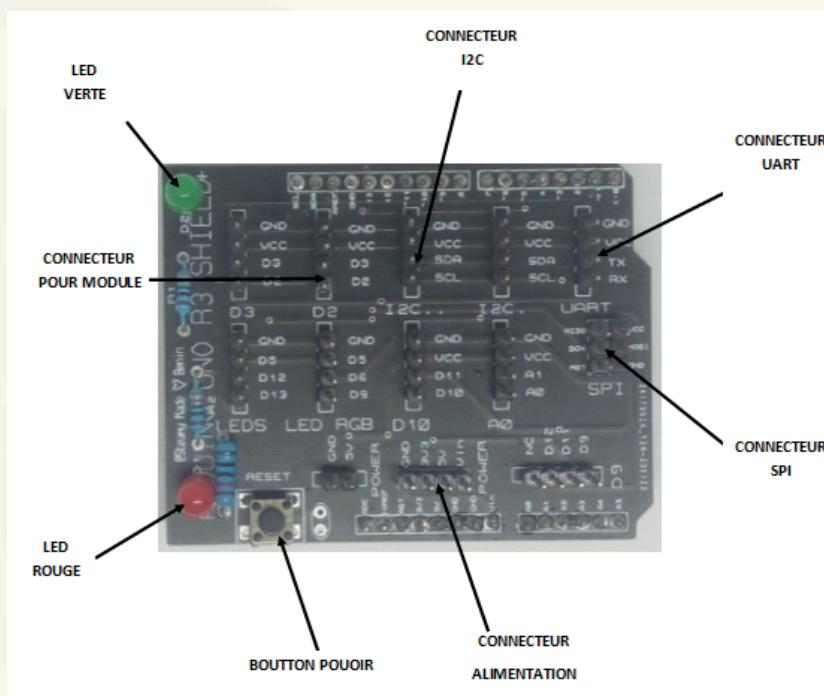
Avec la “STEAMY PRO”, les apprenants auront à assimiler les notions de systèmes, d’entrées et de sorties d’un système, d’entrées digitales et analogiques, de sorties digitales et analogiques, de protocoles de communications ainsi que le fonctionnement de petits systèmes intelligents.

Le kit “STEAMY PRO” contient une carte Arduino Uno, un Shield Arduino et 8 modules. Les informations détaillées sont répertoriées ci-dessous.

### SHIELD ARDUINO

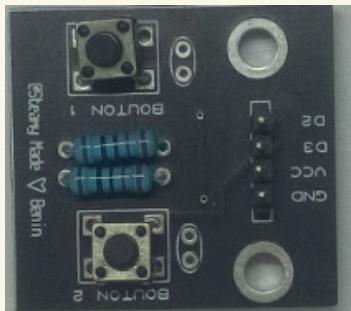
C'est une carte électronique ayant la même forme et les mêmes dimensions qu'une carte Arduino Uno classique et pouvant s'emboîter parfaitement dans cette dernière. On y retrouve :

- une led verte qui s'allume au branchement avec la carte Arduino et l'ordinateur ;
- une led rouge connectée à une pin digitale de la carte Arduino ;
- un bouton poussoir de réinitialisation permettant de reprendre le fonctionnement d'un processus depuis le début ;
- un connecteur pour l'utilisation du protocole de communication UART
- un jeu de deux connecteurs pour l'utilisation du protocole de communication SPI ;
- deux connecteurs pour l'utilisation du protocole de communication I2C ;
- deux connecteurs pour des alimentations externes 5V et 3,3V ;
- sept connecteurs pour les liaisons avec les différents modules de “STEAMY PRO”.



## CAPTEURS

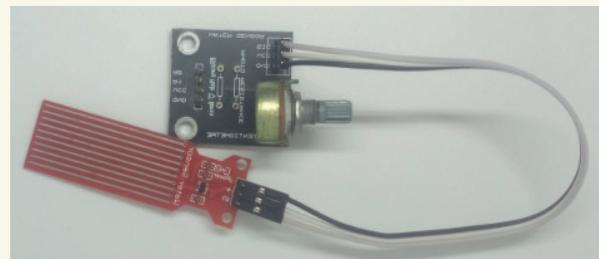
### Boutons poussoirs



Ce module utilise deux boutons poussoirs comme composants principaux. Un bouton poussoir est un dispositif de commande d'appareils électriques destiné à être actionné manuellement et ayant un rappel de ressort. Il est utilisé en détection de front montant et de front descendant. Le module sera utilisé pour commander des LEDS et d'autres actionneurs.

### Capteur d'eau

Le capteur d'eau est un capteur qui permet la détection de la présence de l'eau dans un milieu ou dans une enceinte. Pour que l'eau soit détectée, il faut que le capteur connecté au module soit immergé dans l'eau mais uniquement la partie qui semble rayée.



### Potentiomètre



Ce module utilise un potentiomètre comme composant principal. Un potentiomètre est une résistance variable. La valeur de la résistance change en fonction que l'utilisateur tourne le potentiomètre dans un sens ou dans l'autre. Le module du potentiomètre permet d'utiliser les valeurs analogiques.

### Capteur DHT22

Le capteur DHT22 (aussi connu sous la référence AM2302) est un capteur de température et d'humidité « deux en un ». Il est capable de mesurer des températures de -40 à +125°C avec une précision de +/- 0.5°C et des taux d'humidité relative de 0 à 100% avec une précision de +/- 2% (+/- 5% aux extrêmes, à 10% et 90%). Une mesure peut être réalisée toutes les 500 millisecondes (soit deux fois par seconde).



## CAPTEURS - SUITE

### Module Ultrason



Ce module utilise un capteur de distance comme composant principal. Il s'agit d'un capteur qui est basé sur les fréquences ultrason. Le détecteur HC-SR04 utilise les ultrasons pour déterminer la distance à laquelle se trouve un objet. Peu importe l'intensité de la lumière, la température ou le type de matière, le capteur pourra facilement détecter s'il y a un obstacle devant lui. Tout de fois, il peut être constraint sur certains types de couleurs tel que le noir (contraste), ou encore sur la matière comme le textile. Son champ de vision est de 90° environ selon l'environnement. Si une impulsion de plus de 10µS est détectée, alors le capteur envoie une série de 8 impulsions à ultrason de 40 kHz et attend le réfléchissement du signal. Ensuite, en ayant en tête la vitesse du son, il effectue un rapide calcul pour déterminer la distance.

## VOYANTS ET ACTIONNEURS

### Module LED RGB



Le principe de fonctionnement de la led RGB est simple. Les changements de couleurs sont obtenus par le mélange de 3 couleurs primaires: le rouge, le vert et le bleu. D'où l'abréviation RGB pour R (red – rouge), G (green – vert) et B (blue – bleu). Les leds flexibles RGB contiennent ces 3 couleurs primaires. Selon le "taux" de rouge, de vert ou de bleu, on peut obtenir une couleur précise.

## VOYANTS ET ACTIONNEURS - SUITE

### Module Buzzer



Ce module utilise un buzzer piézo comme composant principal, il peut produire un son aigu lorsqu'il est connecté à la sortie numérique et au niveau logique réglé sur élevé, sinon il peut produire diverses tonalités en fonction des fréquences générées par la sortie PWM analogique qui lui est connectée. (Remarque: la plage de fréquences que l'oreille humaine normale peut distinguer est comprise entre 20 Hz et 20 kHz.)

### Ecran LCD

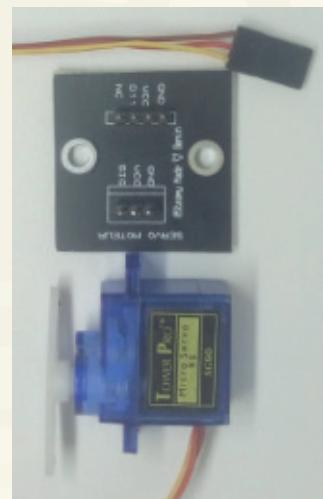


L'écran LCD de "STEAMY PRO" est un écran à 16 caractères à 2 lignes, il utilise une interface de bus I2C pour communiquer avec la carte de développement, ce qui réduira l'en-tête des broches de 10 à 2, ce qui est très pratique pour "STEAMY

PRO". Cet écran LCD permet l'affichage de messages comme notifications et informations venant des capteurs sur le milieu ambiant.

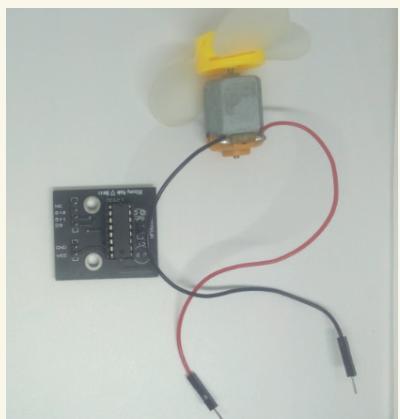
### Module Servo Moteur

A ce module, on connecte un servomoteur. Un servomoteur (souvent abrégé en « servo », provenant du latin *servus* qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi. Le servomoteur intègre dans un même boîtier, la mécanique (moteur et engrenage), et l'électronique, pour la commande et l'asservissement du moteur. La position est définie avec une limite de débattement d'angle de 180 degrés, mais également disponible en rotation continue.



## VOYANTS ET ACTIONNEURS - SUITE

### Module Moteur CC



Ce module est un module de commande de moteur à courant continu avec le circuit intégré L293. On dit qu'un moteur est un composant de conversion d'énergie électrique en énergie mécanique. Les moteurs à courant continu transforment l'énergie électrique en énergie mécanique de rotation, pour être précis. Le module permettra de commander un moteur pour plusieurs applications : changement de sens, accélération, décélération, et bien d'autres cas d'usage.

### Module LEDS Vert Jaune Rouge



Ce module présente trois leds de couleurs respectives verte, jaune et rouge. Il permet de faire de petits jeux de lumière, de clignotement ...

02

## Le logiciel Arduino

## 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**

Veuillez télécharger et installer l'IDE Arduino en fonction de votre système d'exploitation.

The screenshot shows the Arduino website's download section. At the top, there are navigation links: MATERIEL, LOGICIEL (with a dropdown arrow), DOCUMENTATION (with a dropdown arrow), COMMUNAUTÉ (with a dropdown arrow), BLOG, and SUR. Below these, there is a large button labeled "TÉLÉCHARGER ARDUINO IDE 1.8.13". To the left of the button is a circular icon containing the Arduino logo (an 'A' inside a circle). To the right of the button is a sidebar titled "OPTIONS DE TÉLÉCHARGEMENT" which lists download links for Windows, Application Windows, Linux, and Mac OS X. The main content area contains text about the Arduino IDE, links to source code and GitHub, and a note about PGP signatures.

MATÉRIEL    LOGICIEL ▾    DOCUMENTATION ▾    COMMUNAUTÉ ▾    BLOG    SUR

 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

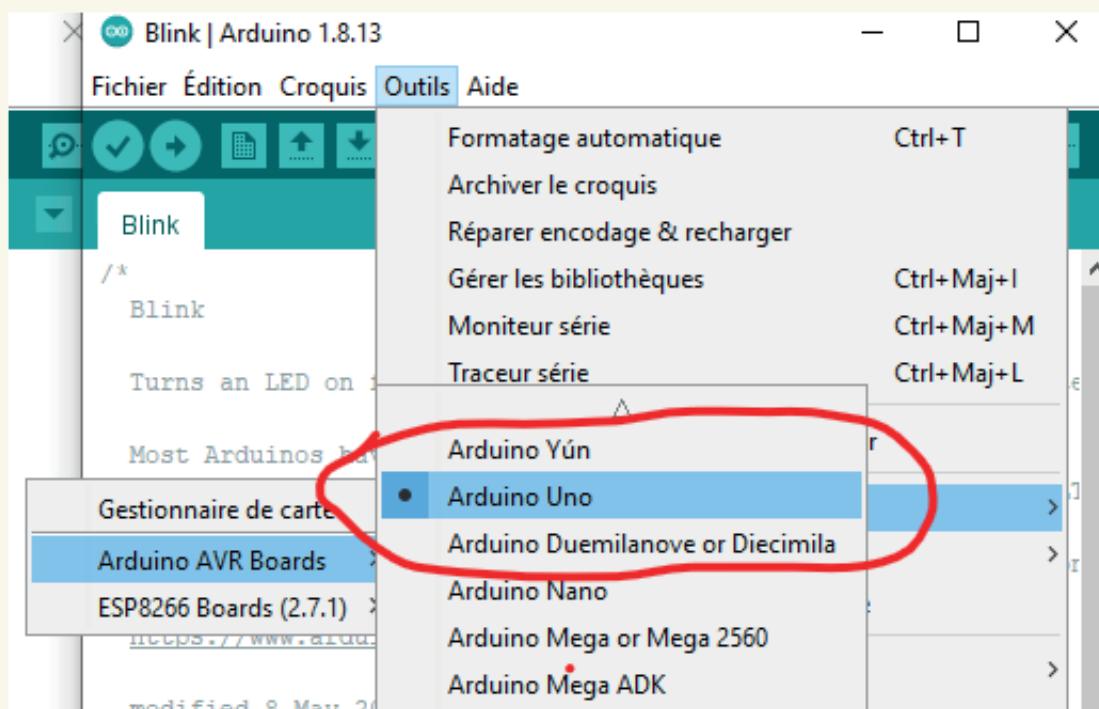
**Mac OS X** 10.10 ou plus nouveau

Notes de libération Checksums (sha512)

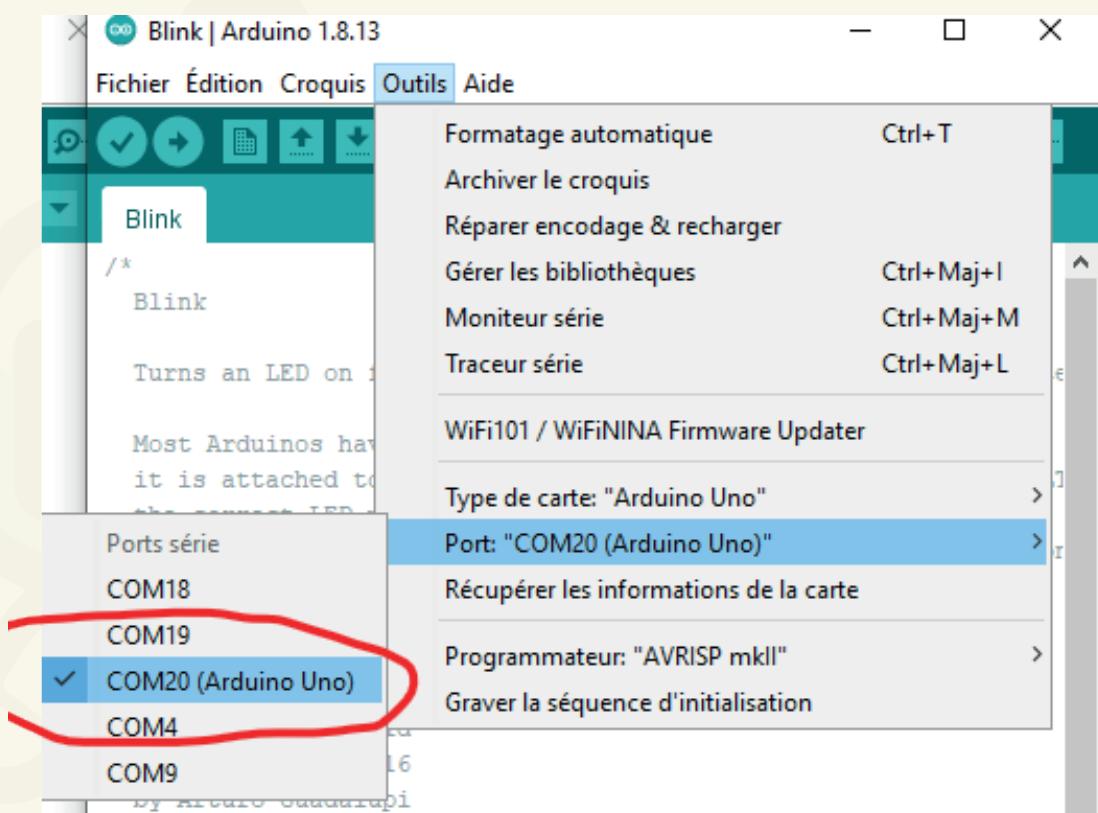
### TÉLÉVERSER UN PROGRAMME SUR LA CARTE ARDUINO

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, cliquer successivement sur Outils-Type de carte- Arduino AVR Boards- Arduino Uno comme le montre la figure ci-dessous :



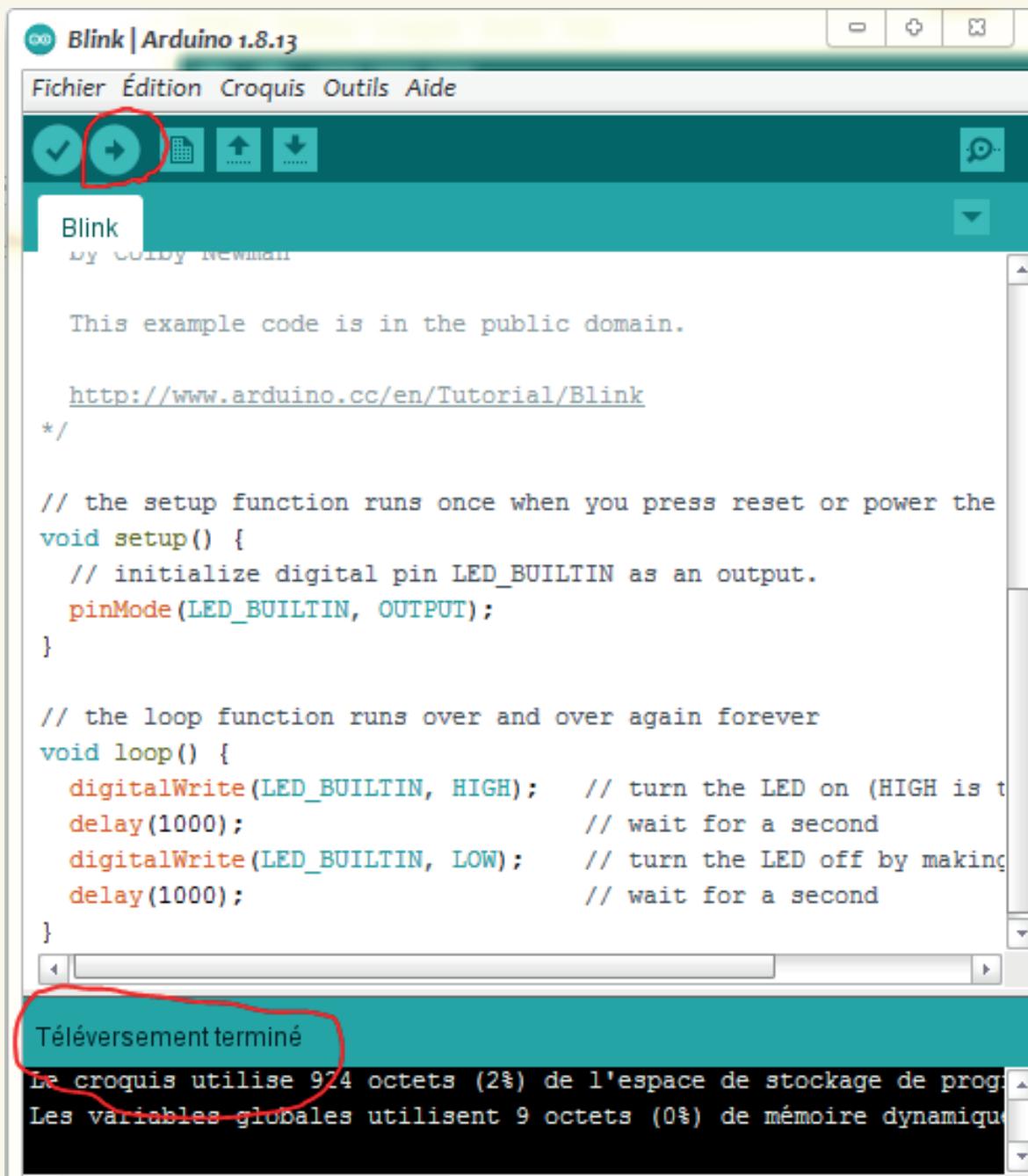
- choisir le port de communication entre l'ordinateur et la carte Arduino.  
Pour ce faire, cliquer successivement sur Outils- Port- COM.. comme sur la figure ci-dessous :



## 02 Le logiciel Arduino

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



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

**03**

## Tutoriel “STEAMY PRO”

### Objectif

Cette section se compose de 11 tutoriels, les tutoriels peuvent être divisés en plusieurs parties. Ils permettent de prendre en main le fonctionnement des différents capteurs, voyants et actionneurs de “STEAMY PRO”. 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 PRO”.
- Être capable d'identifier les différents modules ( capteurs, voyants et actionneurs ) de “STEAMY PRO” et leurs applications.
- Être en mesure d'imaginer d'autres applications et d'autres fonctionnements.

### 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 ;
- S'assurer de la bonne connexion entre le shield et les différents modules

## Session 1 : STEAMY PRO - LED rouge

### Objectif :

Clignotement d'une LED - Utilisez la logique numérique haut / bas pour faire «clignoter» la LED rouge.

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

```
//assign led as pin 13
#define led 13

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

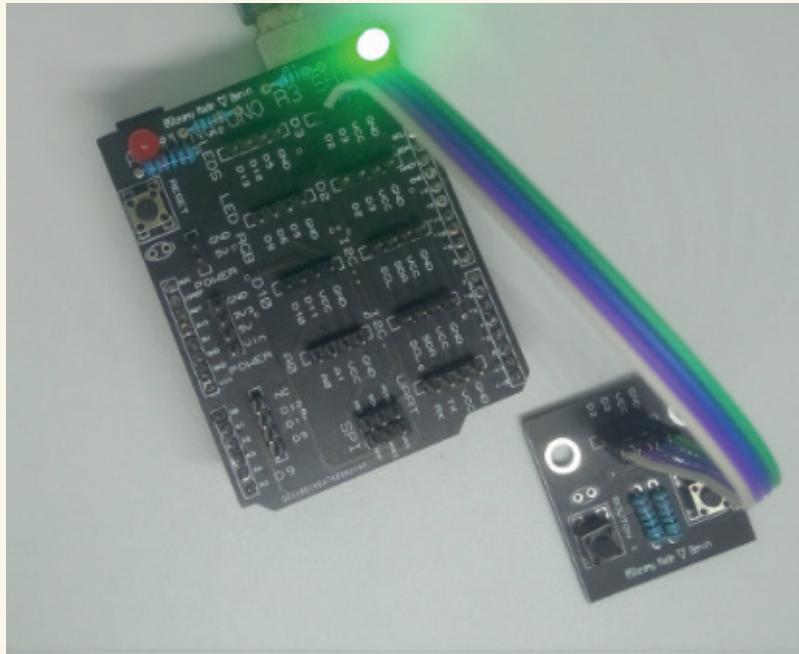
void loop()
{
    //turn on led(set logic level high)
    digitalWrite(led, HIGH);
    //wait 500 ms
    delay(500);
    //turn off led(set logic level low)
    digitalWrite(led, LOW);
    //wait 500 ms
    delay(500);
}
```

- Téléverser le code sur la carte Arduino.

- Observez le résultat. Une fois le téléchargement terminé, vous devriez voir la LED rouge (qui est sur le shield )clignoter, c'est-à-dire s'allumer et s'éteindre toutes les demi-secondes .

### Session 2 : STEAMY PRO - Boutons poussoirs

Connecter le module des boutons poussoirs au shield comme sur la figure ci dessous (les inscriptions sur le module et le shield devrait vous aider)



#### Objectif :

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

- 

Copiez et collez le code suivant dans l'IDE Arduino

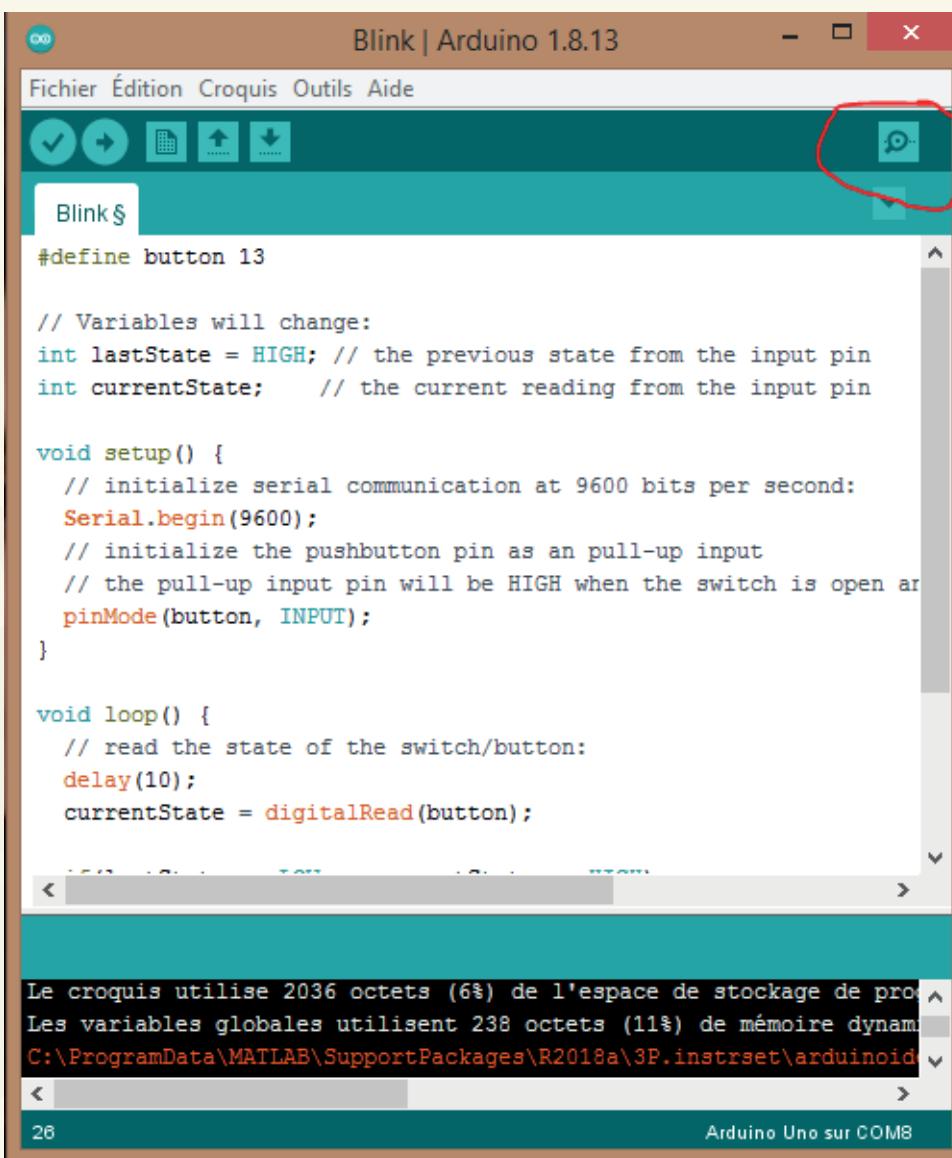
```
#define button 2

// 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);
}
```

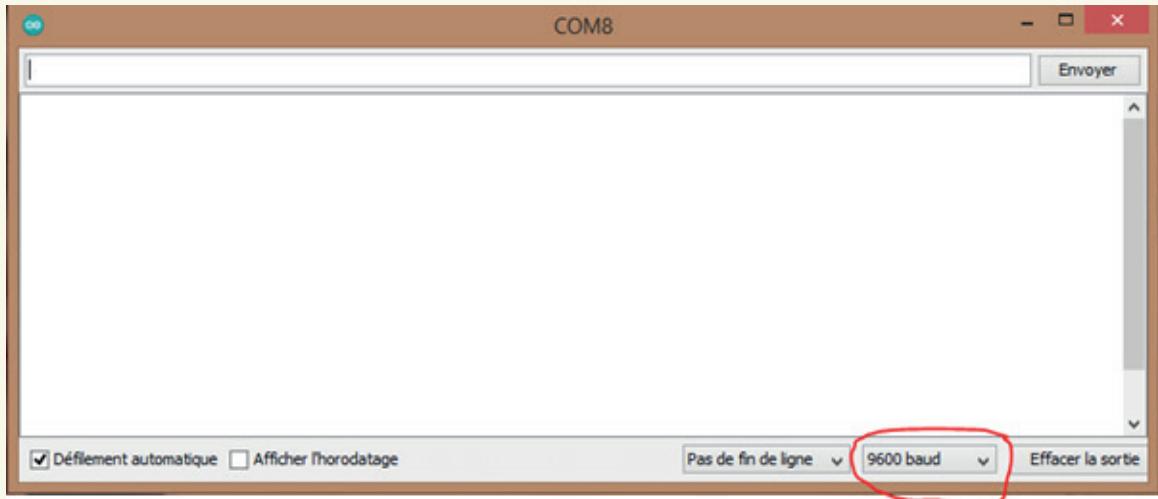
```
void loop() {
  // read the state of the switch/button:
  delay(500);
  currentState = digitalRead(button);
  if(lastState == LOW && currentState == HIGH)
    Serial.println("L'état du bouton a change du bas vers le haut.");
  // save the the last state
  lastState = currentState;
}
```

- 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.

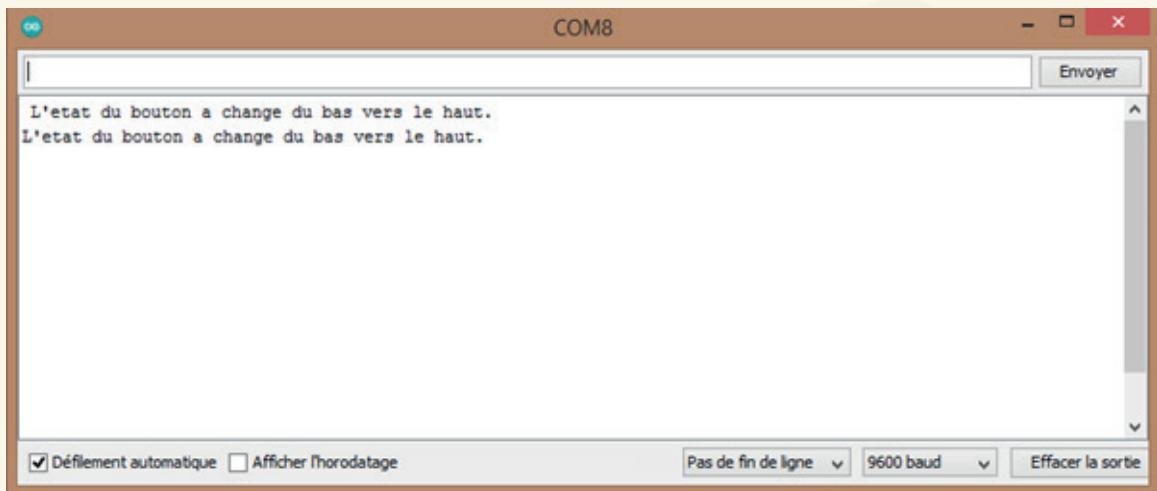


## 03 Tutoriel “STEAMY PRO”

- Quand vous ouvrez le moniteur série, assurez-vous d'avoir choisi la bonne vitesse de réception des données. Ceci est très important. Pour ce faire, il suffit de le sélectionner (nous utilisons ici la vitesse 9600 bauds) en bas de la fenêtre du moniteur série comme ci-dessous indiqué :

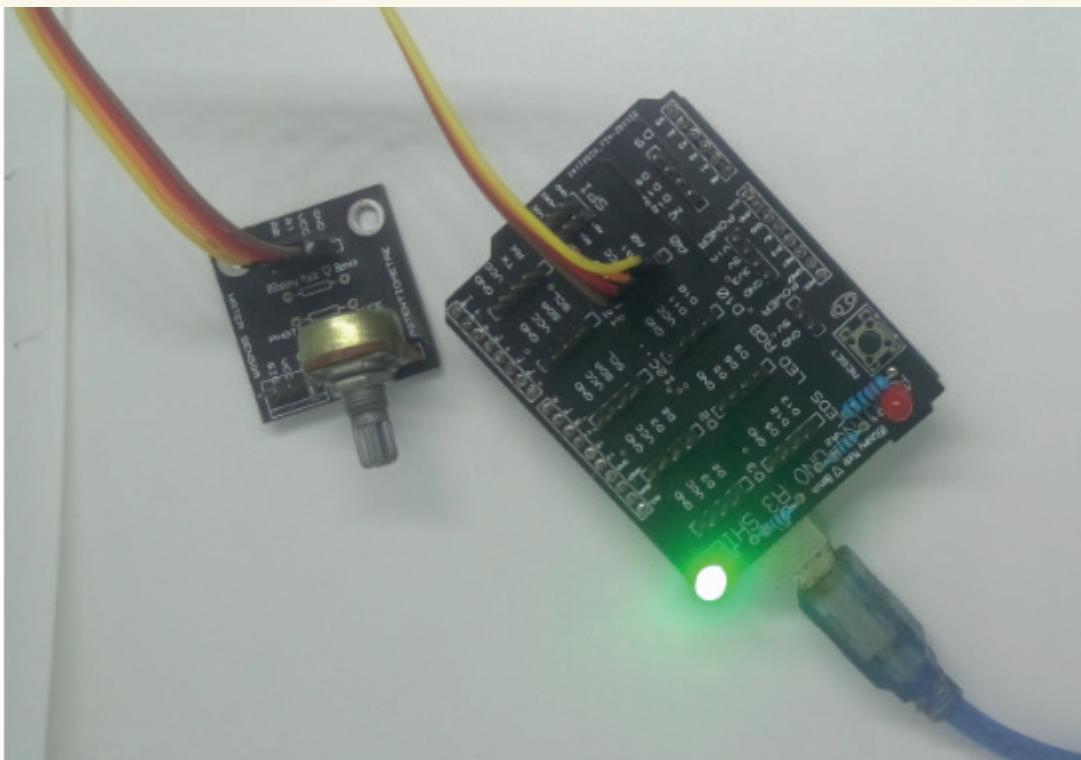


- Appuyez sur le bouton 1 du shield et relâchez ensuite. Vous devriez recevoir la notification dans le moteur série comme dans la figure ci-dessous :



## Session 3 : STEAMY PRO - Lecture de valeurs analogiques

Connecter le module du potentiomètre et du capteur d'eau au shield comme sur la figure ci dessous (les inscriptions sur le module et le shield devrait vous aider)



### Objectif :

Lire les valeurs analogiques provenant du potentiomètre.

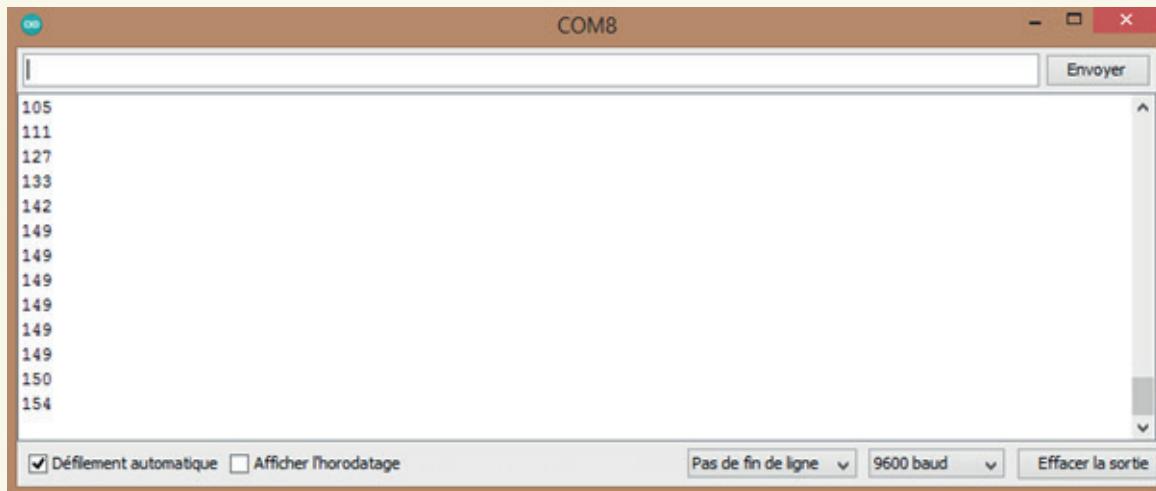
- Copiez et collez le code suivant dans l'IDE Arduino

```
#define potentiometre A0
```

```
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);  
}
```

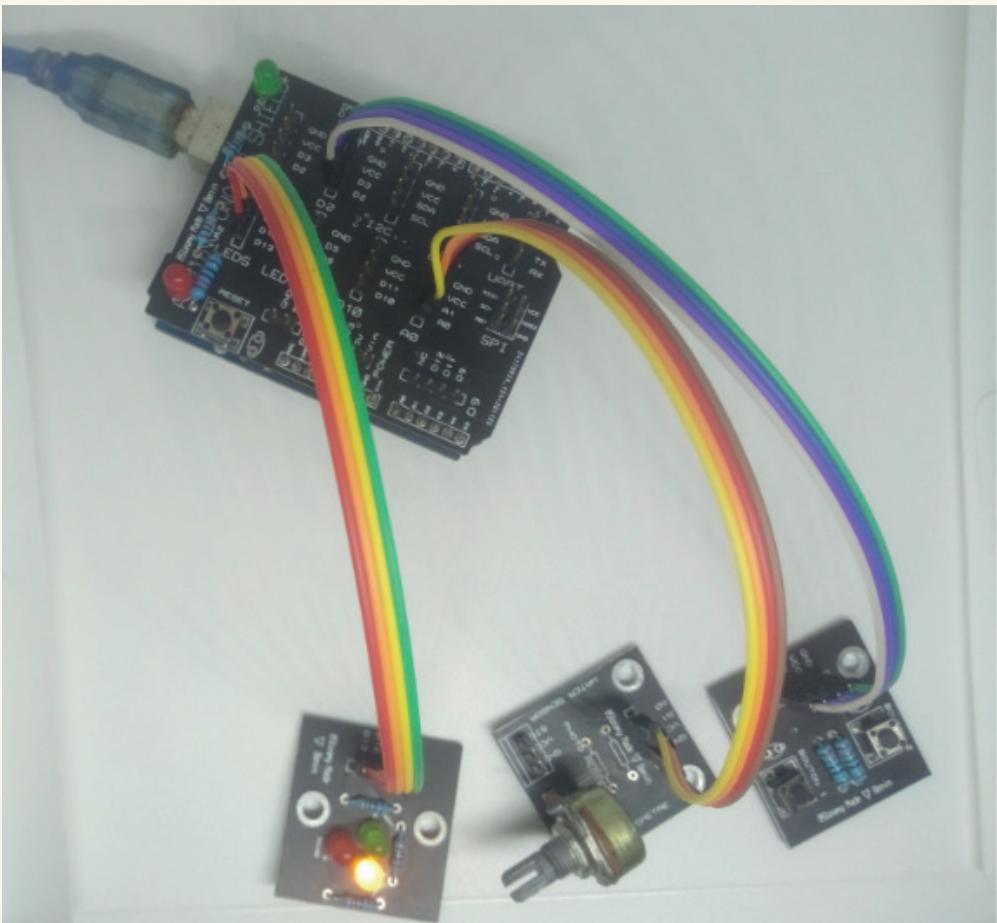
## 03 Tutoriel “STEAMY PRO”

- Observez le résultat. Une fois le téléchargement terminé, ouvrez le moniteur série. Vous devriez voir des valeurs entre 0 et 1023 défiler. 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 4 : STEAMY PRO - Module LEDs

Connecter le module LEDS , le module bouton poussoirs et le module du potentiomètre comme sur la figure ci dessous (les inscriptions sur le module et le shield devrait vous aider)



### Objectif :

Allumer chaque led l'une après l'autre dans un sens constant.

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

```
#define led_green 5  
#define led_yellow 12  
#define led_red 13
```

## 03 Tutoriel “STEAMY PRO”

```
void setup(){
    //set leds as output
    pinMode(led_green , OUTPUT);
    pinMode(led_yellow , OUTPUT);
    pinMode(led_red , OUTPUT);
}
void loop(){
    digitalWrite(led_green, HIGH);
    digitalWrite(led_yellow, LOW);
    digitalWrite(led_red, LOW);
    delay(500);
    //wait 500 ms

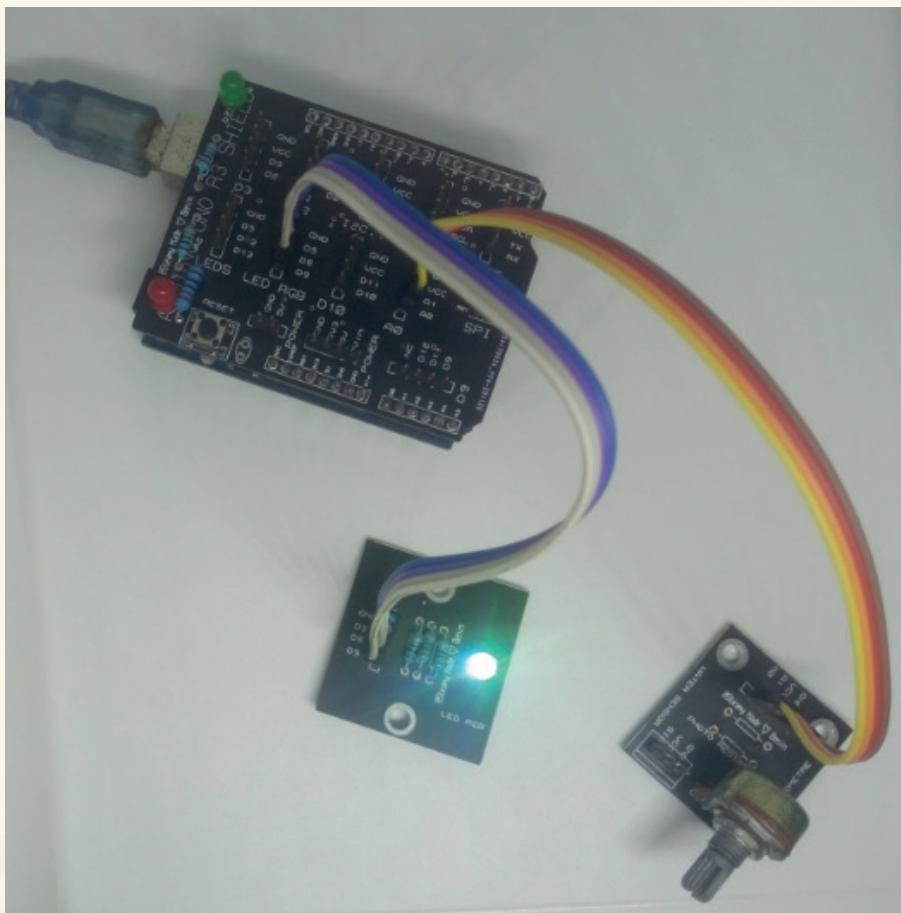
    digitalWrite(led_green,LOW );
    digitalWrite(led_yellow, HIGH);
    digitalWrite(led_red,LOW );
    delay(500);
    //wait 500 ms

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

- Observez le résultat. Une fois le téléchargement terminé, vous verrez la LED verte s'allumer et s'éteindre après une demi-seconde, ensuite c'est le tour de la LED jaune et enfin celui de la LED rouge . Et le cycle recommence.

## Session 5 : STEAMY PRO - Module LED RGB

Connecter le module LED RGB et le module du potentiomètre comme sur la figure ci dessous (les inscriptions sur le module et le shield devrait vous aider)



## Objectif :

Afficher une couleur fixe au niveau du module RGB.

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

```
#define led_red 5  
#define led_green 6  
#define led_blue 9  
  
int red_value = 200;  
int green_value = 100  
int blue_value = 10;
```

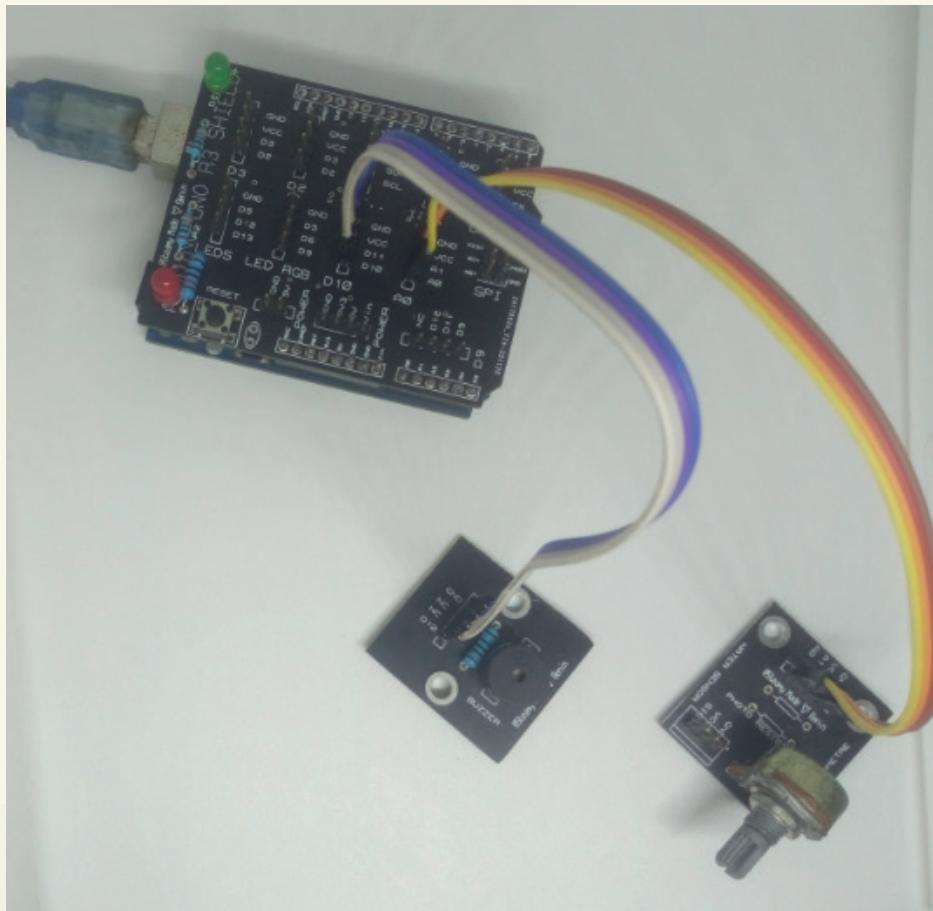
## 03 Tutoriel “STEAMY PRO”

```
void setup(){  
    //set led as output  
    pinMode(led_red , OUTPUT);  
    pinMode(led_green , OUTPUT);  
    pinMode(led_blue , OUTPUT);  
  
}  
void loop() {  
    analogWrite(led_red, red_value);  
    analogWrite(led_green, green_value);  
    analogWrite(led_blue, blue_value);  
}
```

- Observez le résultat. Une fois le téléchargement terminé, la LED affichera une certaine couleur. Jouer sur les valeurs de **red\_value**, **green\_value** et de **blue\_value** dans le code en les changeant (ne pas dépasser 255) et téléverser à nouveau le code. La couleur de la LED changera.

## Session 6 : STEAMY PRO - Module Buzzer

Connecter le module LED Buzzer ainsi que celui du potentiomètre comme sur la figure ci dessous (les inscriptions sur le module et le shield devrait vous aider)



### Objectif :

Réaliser des notes de musique.

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

```
// 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
```

## 03 Tutoriel “STEAMY PRO”

```
#define NOTE_F4 349
#define NOTE_G4 392
#define NOTE_C5 523

//assign buzzer as pin 10
#define buzzer 10

// 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_C5, 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,
    8, 8, 4, 4, 4, 2,
};
void setup() {}
void loop() {
    // iterate over the notes of the melody:
    for (int thisNote = 0 ; thisNote < 25 ; thisNote++) {

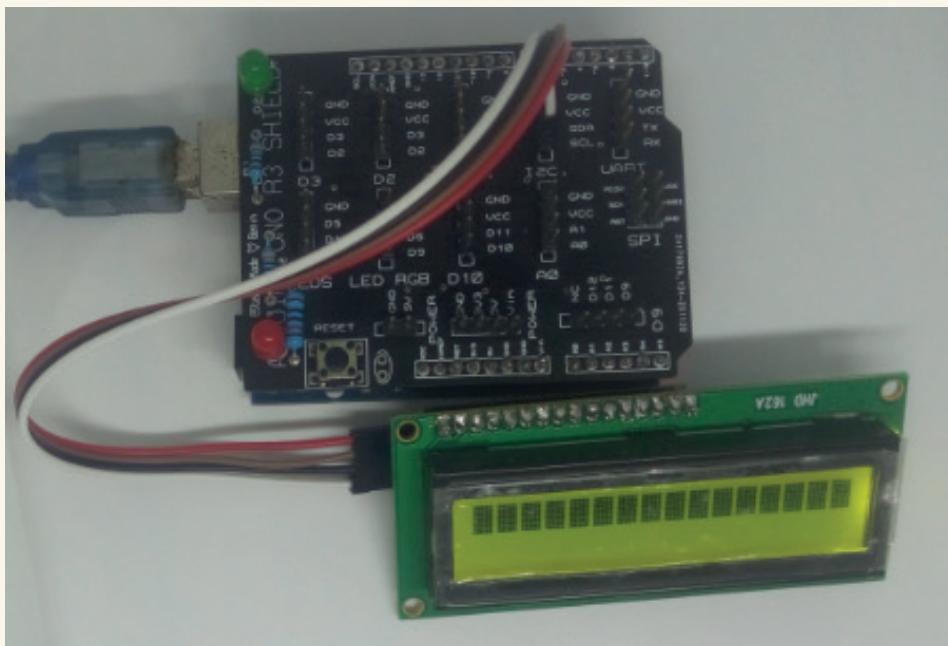
        // to calculate the note duration, take one second divided by the note type.
        //e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
        int noteDuration = 1000 / noteDurations[thisNote];
        tone(buzzer, melody[thisNote], noteDuration);

        // 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 7 : STEAMY PRO - Ecran LCD

L'écran LCD nous servira à afficher des messages. Avant toute utilisation, vous devez la brancher au Shield Arduino à un des connecteurs I2C. 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



### Objectif :

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.init();
  // Print a message to the LCD.
```

## 03 Tutoriel “STEAMY PRO”

```
lcd.backlight();
lcd.setCursor(5,0);
lcd.print("Salut !");
lcd.setCursor(1,1);
lcd.print("Vous etes sur ");
delay(2000);
lcd.clear();
lcd.setCursor(3,0);
lcd.print("STEAMY PRO");
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 :

The screenshot shows the Arduino IDE interface with the following details:

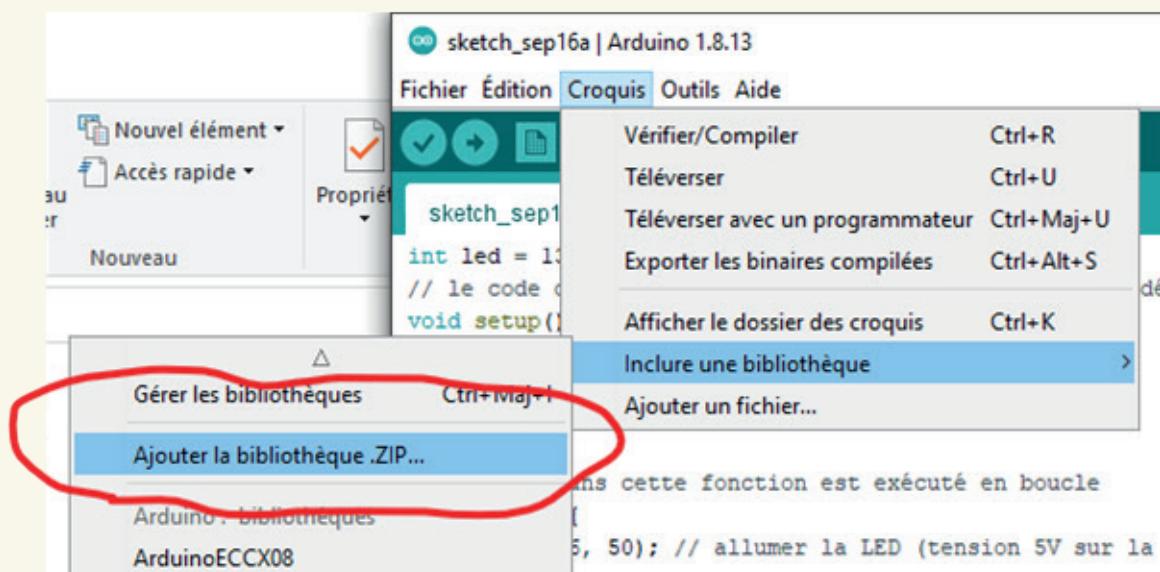
- Title Bar:** Esaie\_Steamy\_Starter | Arduino 1.8.13
- Menu Bar:** Fichier Édition Croquis Outils Aide
- Toolbar:** Includes icons for file operations like Open, Save, and Print.
- Code Editor:** Displays the provided Arduino sketch. The line `#include <LiquidCrystal_I2C.h>` is highlighted with a red box.
- Compile Output:** Shows the terminal window with the following error message:

```
LiquidCrystal_I2C.h: No such file or directory
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 :



- 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:

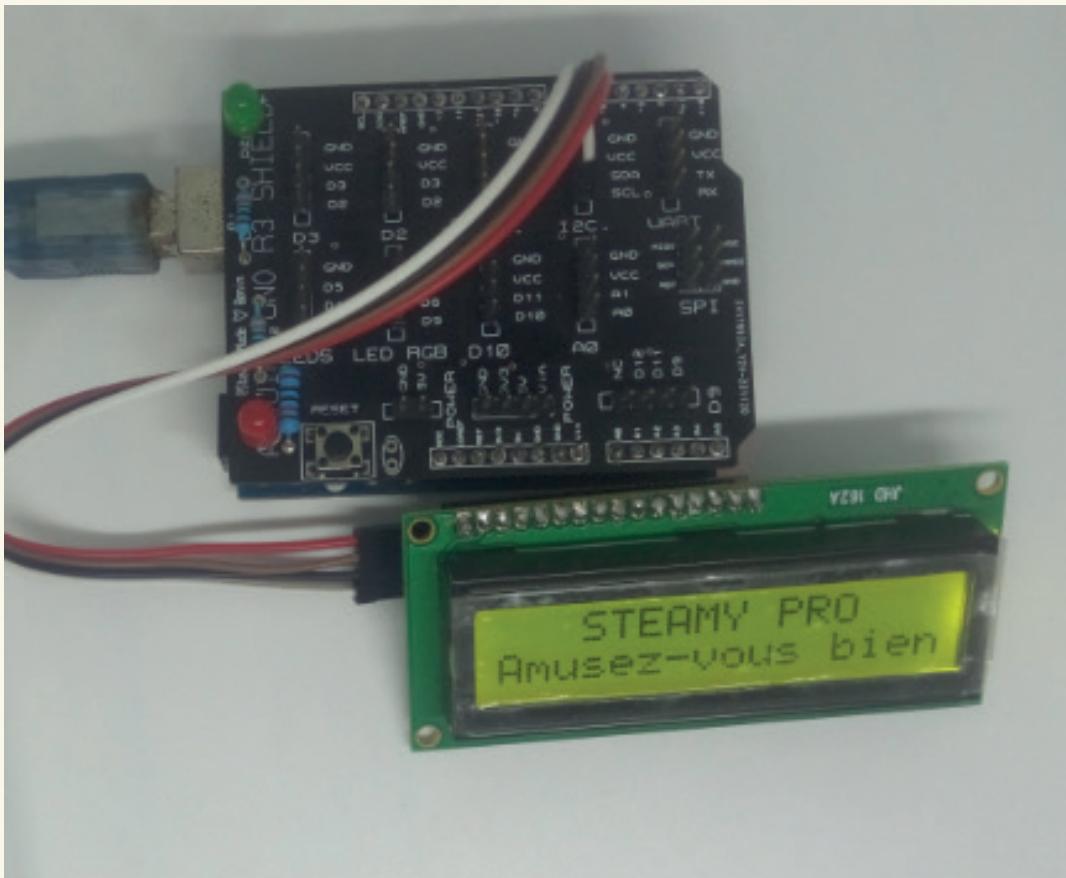
The screenshot shows the Arduino IDE interface with the code editor containing the following code:

```
lcd.setCursor(1,1);
lcd.print("Vous etes sur ");
delay(2000);
lcd.clear();
lcd.setCursor(1,0);
lcd.print("STEAMY STARTER");
---- ---- / \ / \
```

A red box highlights a message in the status bar at the bottom: "La bibliothèque a été ajoutée à votre dossier de bibliothèques. Veuillez regarder l Compilation terminated. exit status 1 LiquidCrystal\_I2C.h: No such file or directory".

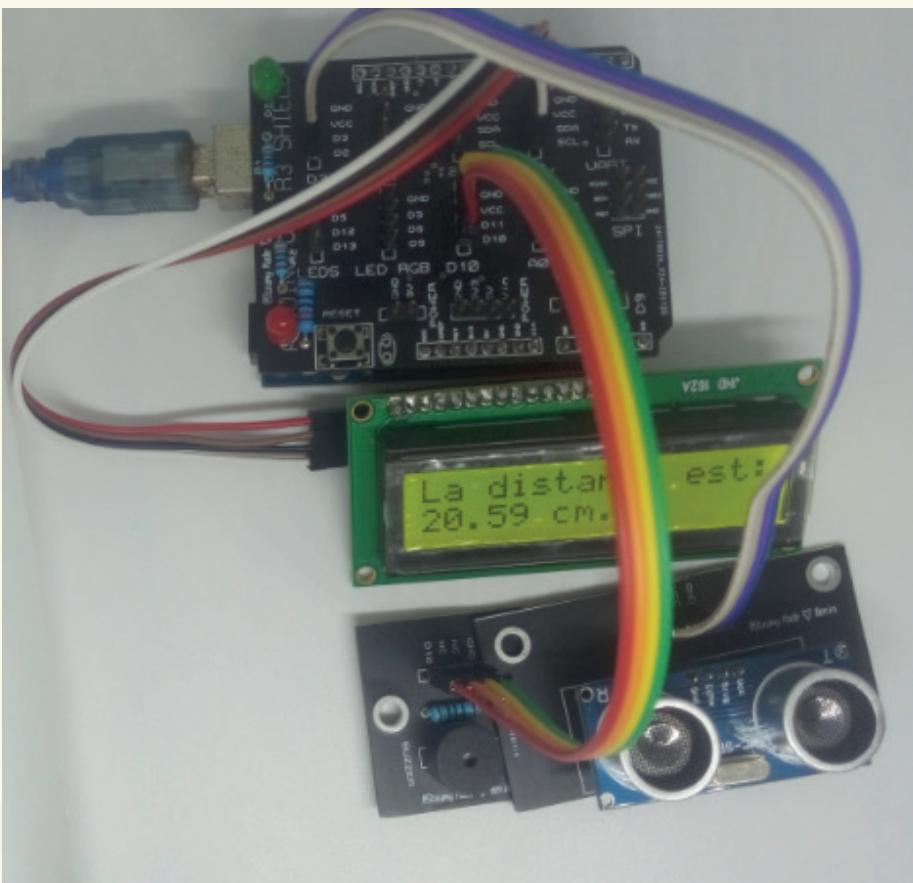
## 03 Tutoriel “STEAMY PRO”

- 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 8 : STEAMY PRO - Module Ultrason

Connecter le module Ultrason, le module buzzer ainsi que l'écran LCD du comme sur la figure ci-dessous (les inscriptions sur le module et le Shield devrait vous aider).



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.](#) et ajouter le au logiciel comme dans la session précédente.

### Objectif :

Mesurer une distance et l'afficher l'écran LCD.

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

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

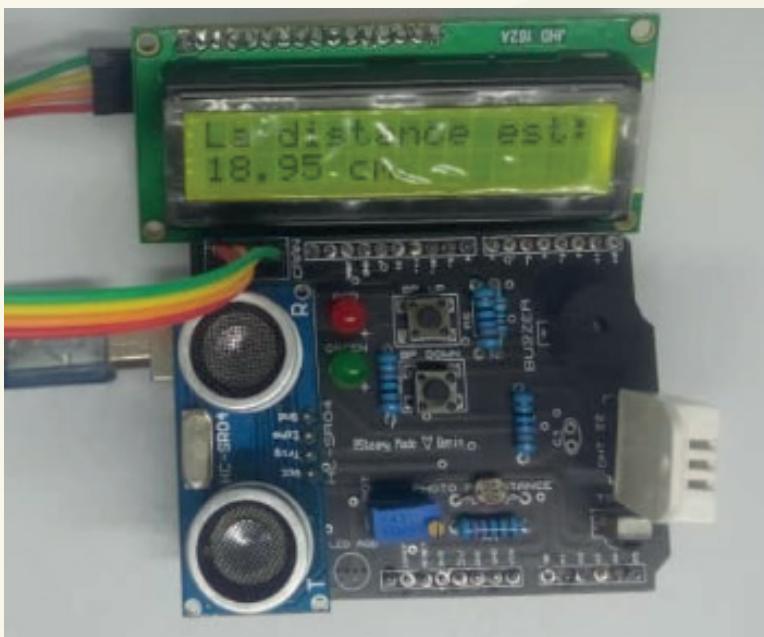
## 03 Tutoriel “STEAMY PRO”

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

```
HCSR04 hc(2,3); //initialisation class HCSR04 (trig pin , echo pin)
```

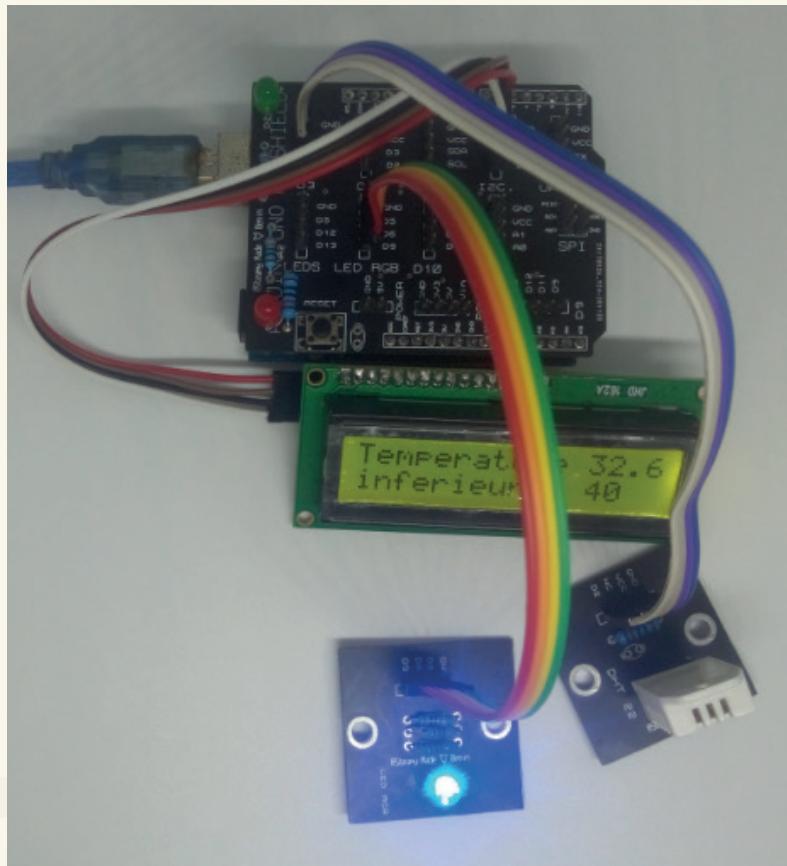
```
void setup()  
{  
    lcd.init();           // initialize the lcd  
    lcd.backlight();  
}  
void loop()  
{  
    lcd.backlight();  
    lcd.setCursor(0,0);  
    lcd.print("La distance est:");  
    lcd.setCursor(0,1);  
    lcd.print(hc.dist());  
    lcd.print(" cm.");  
    delay(1000); 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

- 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 9 : STEAMY PRO - Capteur de température et d'humidité

Connecter le module DHT 22, le module LED RGB, ainsi que l'écran LCD comme sur la figure ci-dessous (les inscriptions sur le module et le Shield devrait vous aider).



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.](#) et ajouter le au logiciel comme dans la session précédente.

### Objectif :

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 téléversez le :

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

## 03 Tutoriel “STEAMY PRO”

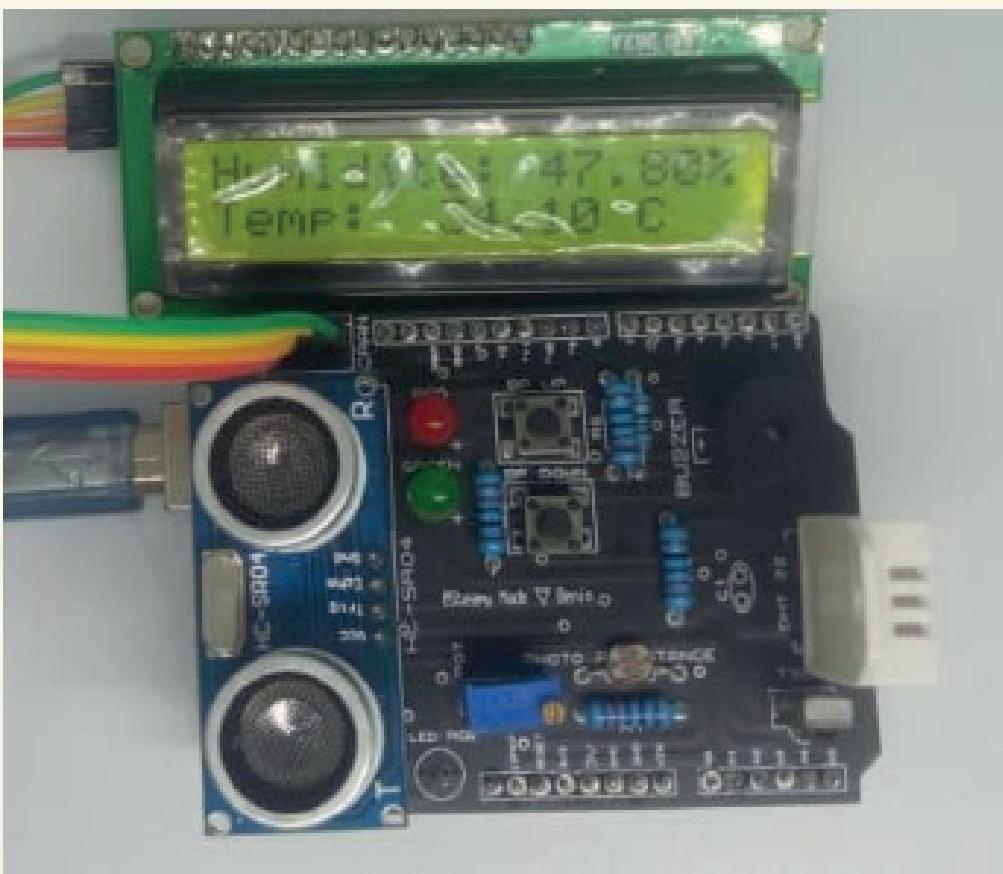
```
#include <LiquidCrystal_I2C.h>

#define DHTPIN 2 // 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();
}
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("%"));
}
```

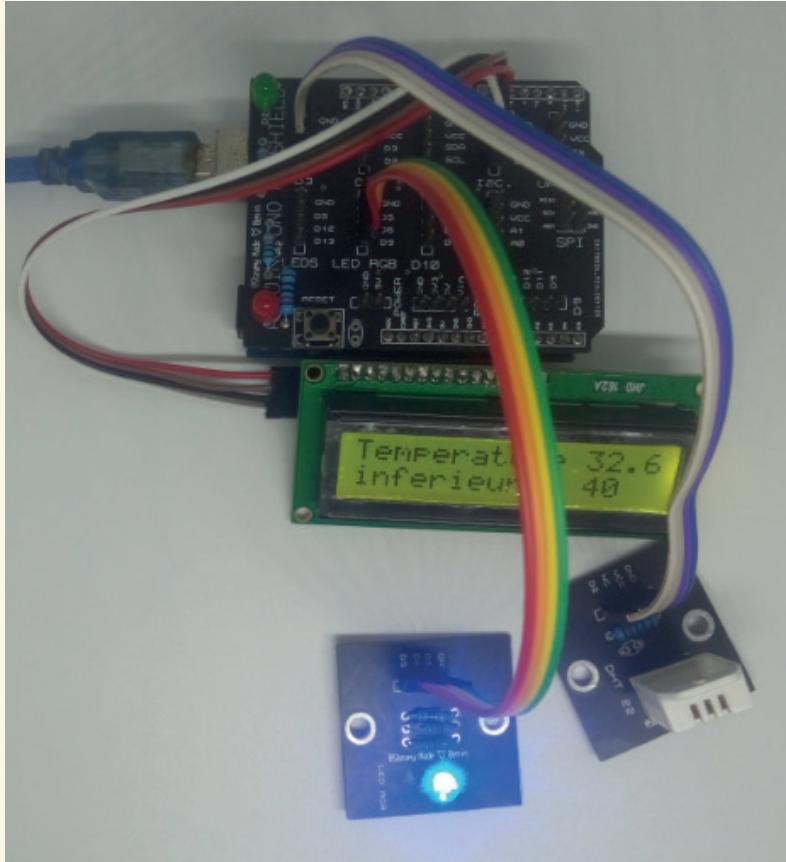
```
lcd.setCursor(0,1);
lcd.print(F("Temp: "));
lcd.setCursor(7,1);
lcd.print(t);
lcd.setCursor(12,1);
lcd.print(F(" C"));
}
```

- 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 10 : STEAMY PRO - Module Servo Moteur

Connecter le module Servo Moteur, et le module du potentiomètre comme sur la figure ci-dessous (les inscriptions sur le module et le Shield devrait vous aider).



Pour utiliser le servo moteur avec le logiciel Arduino, il faut également une bibliothèque spéciale. Télécharger le à cette adresse [www.](#) et ajouter le au logiciel comme dans la session précédente.

#### Objectif :

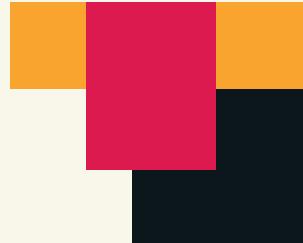
Contrôler la position du servo moteur avec potentiomètre.

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

```
#include <Servo.h>
#define potentiometre A0
Servo myservo; // create servo object to control a servo
int motor_value; // variable to read the value from the analog pin
```

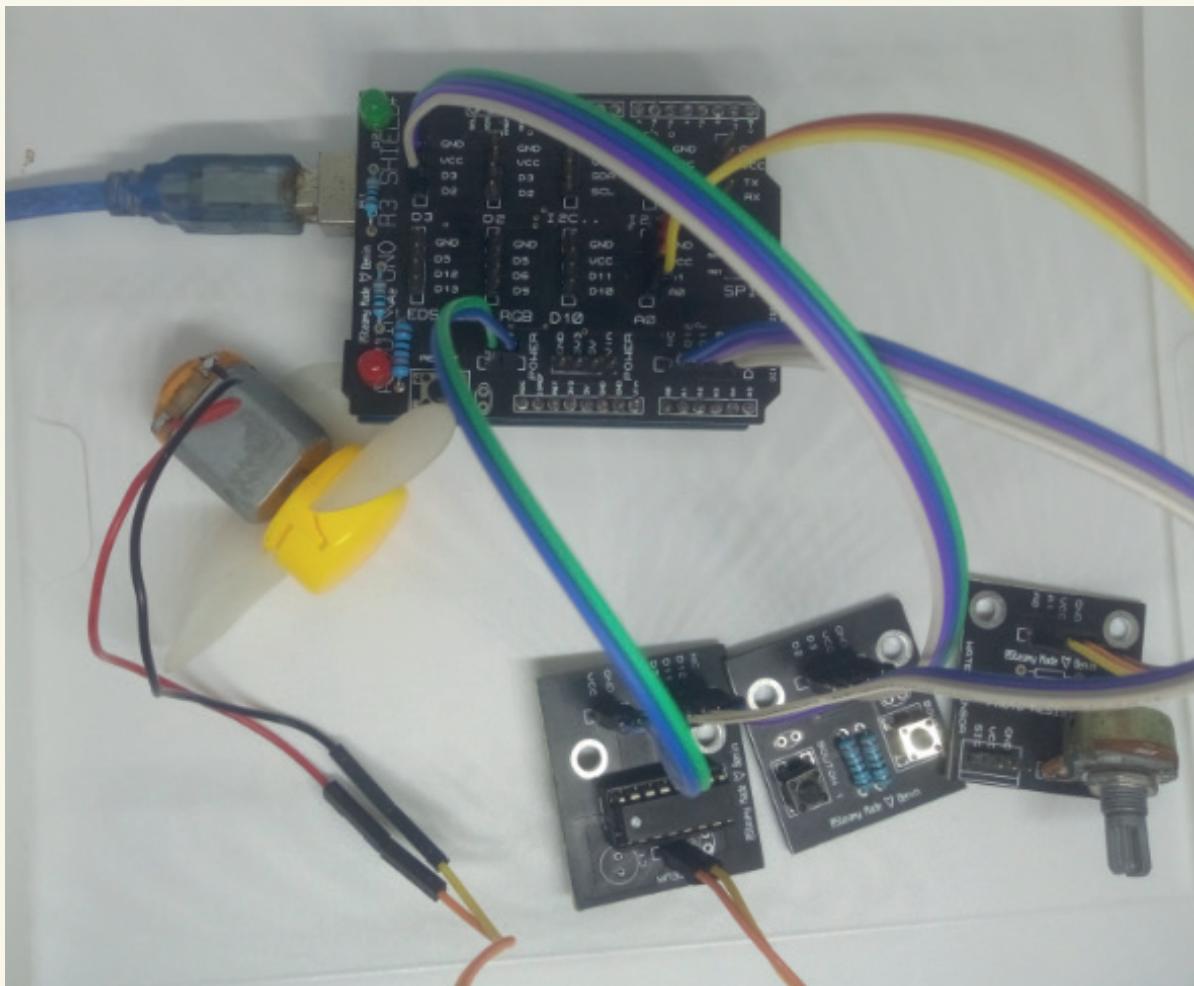
```
void setup() {  
    myservo.attach(11); // attaches the servo on pin 9 to the servo object  
}  
void loop() {  
    motor_value = map( analogRead(potentiometre), 0, 1023, 0, 180); // scale it  
    to use it with the servo (value between 0 and 180)  
    myservo.write(motor_value); // sets the servo position according to  
    the scaled value  
    delay(15); // waits for the servo to get there  
}
```

- Observez le résultat. Une fois le téléchargement terminé, vous pouvez commander le servomoteur, choisir la position de l'axe du moteur avec le potentiomètre.



### Session 11 : STEAMY PRO - Module Moteur CC

Connecter le module Moteur CC, le module du potentiomètre et le module des boutons poussoirs comme sur la figure ci-dessous (les inscriptions sur le module et le Shield devrait vous aider). Connectez également le moteur cc à son module.



#### Objectif :

Changement du sens du moteur.

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

```
// Motor connections  
int motor_en = 9;  
int motor_in1 = 11;  
int motor_in2 = 12;
```

```
void setup() {  
    // Set all the motor control pins to outputs  
    pinMode(motor_en, OUTPUT);  
    pinMode(motor_in1, OUTPUT);  
    pinMode(motor_in2, OUTPUT);  
  
    // Turn off motors - Initial state  
    digitalWrite(motor_in1, LOW);  
    digitalWrite(motor_in2, LOW);  
}  
void loop() {  
    directionControl();  
    delay(1000);  
}  
  
// This function lets you control spinning direction of motors  
void directionControl() {  
    // Set motor to maximum speed  
    // For PWM maximum possible values are 0 to 255  
    analogWrite(motor_en, 100);  
    // Turn on motor  
    digitalWrite(motor_in1, HIGH);  
    digitalWrite(motor_in2, LOW);  
    delay(2000);  
    // Turn off motors  
    digitalWrite(motor_in1, LOW);  
    digitalWrite(motor_in2, LOW);  
    delay(2000);  
    // Now change motor directions  
    digitalWrite(motor_in1, LOW);  
    digitalWrite(motor_in2, HIGH);  
    delay(2000);  
    // Turn off motors  
    digitalWrite(motor_in1, LOW);  
    digitalWrite(motor_in2, LOW);  
    delay(2000);  
}
```

Observez le résultat. Une fois le téléchargement terminé, vous pouvez voir le moteur changer de sens toutes les deux minutes.potentiomètre.



**Fièrement propulsé par**

