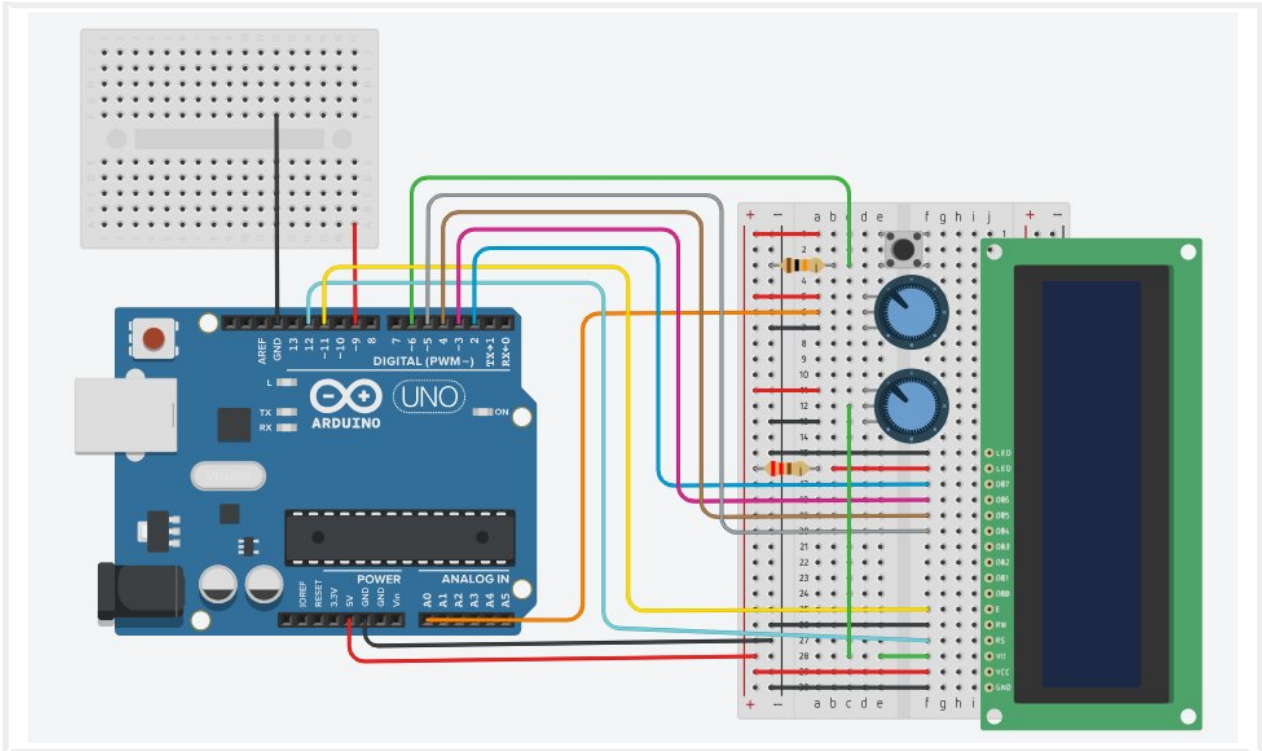


# Générateur 0-5 V

*(Réalisation d'un générateur 0-5 V)*



## . Liste des composants

- . 1 écran LCD 1602
- . 1 bouton poussoir
- . 2 potentiomètre de 10 k $\Omega$
- . 1 résistance de 10 k $\Omega$
- . 1 résistance de 220  $\Omega$
- . 2 plaques d'essais
- . Fils de connexion

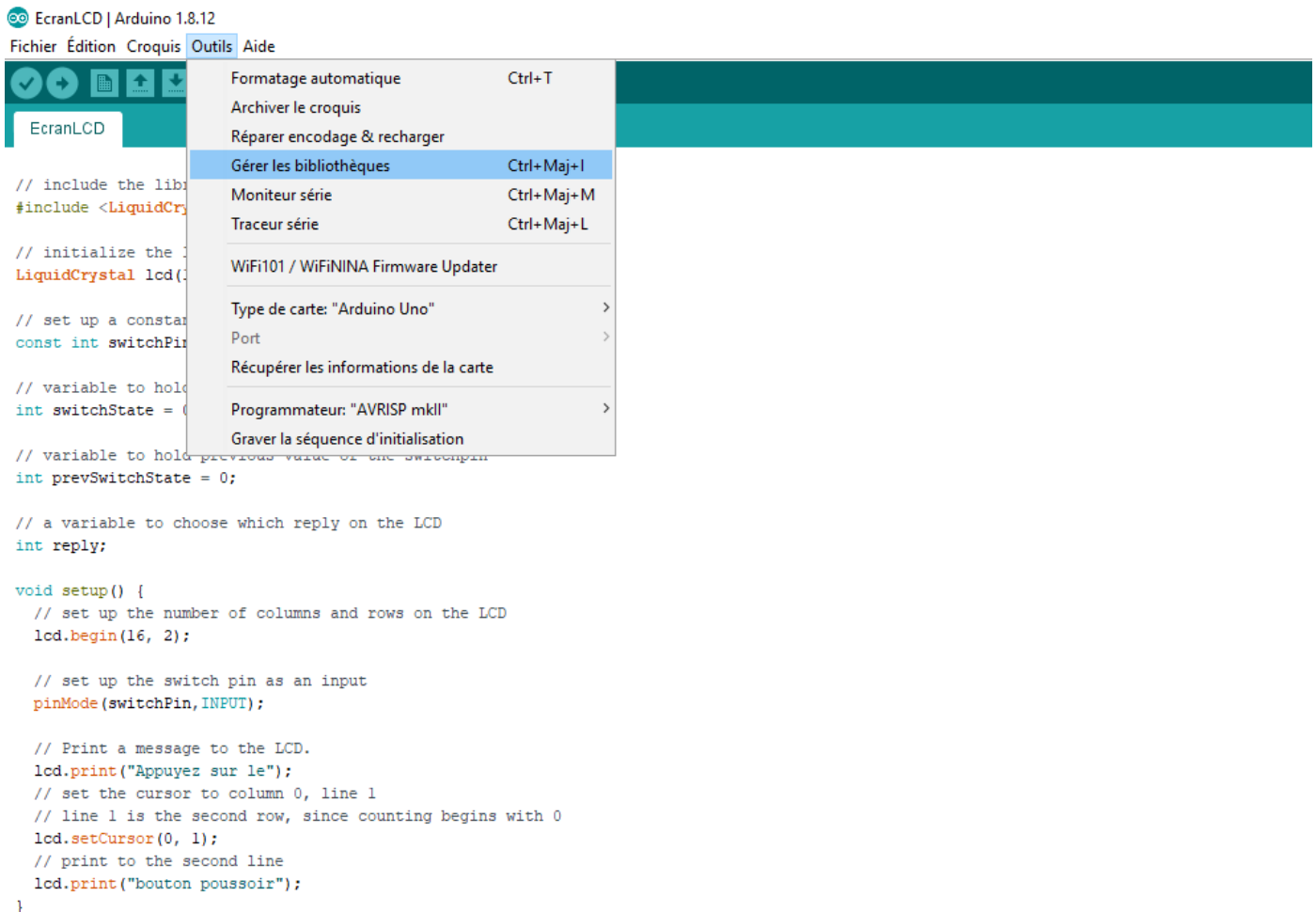
## . Objectif

L'objectif de l'activité est de réaliser un générateur 0-5 V pour, par exemple, alimenter une DEL. Le bouton poussoir permet d'allumer ou d'éteindre le générateur. La valeur de la tension délivrée entre la sortie 9 (sortie PWM) et la masse de l'Arduino Uno est réglée par le premier potentiomètre. La valeur de la tension est affichée sur un écran LCD.

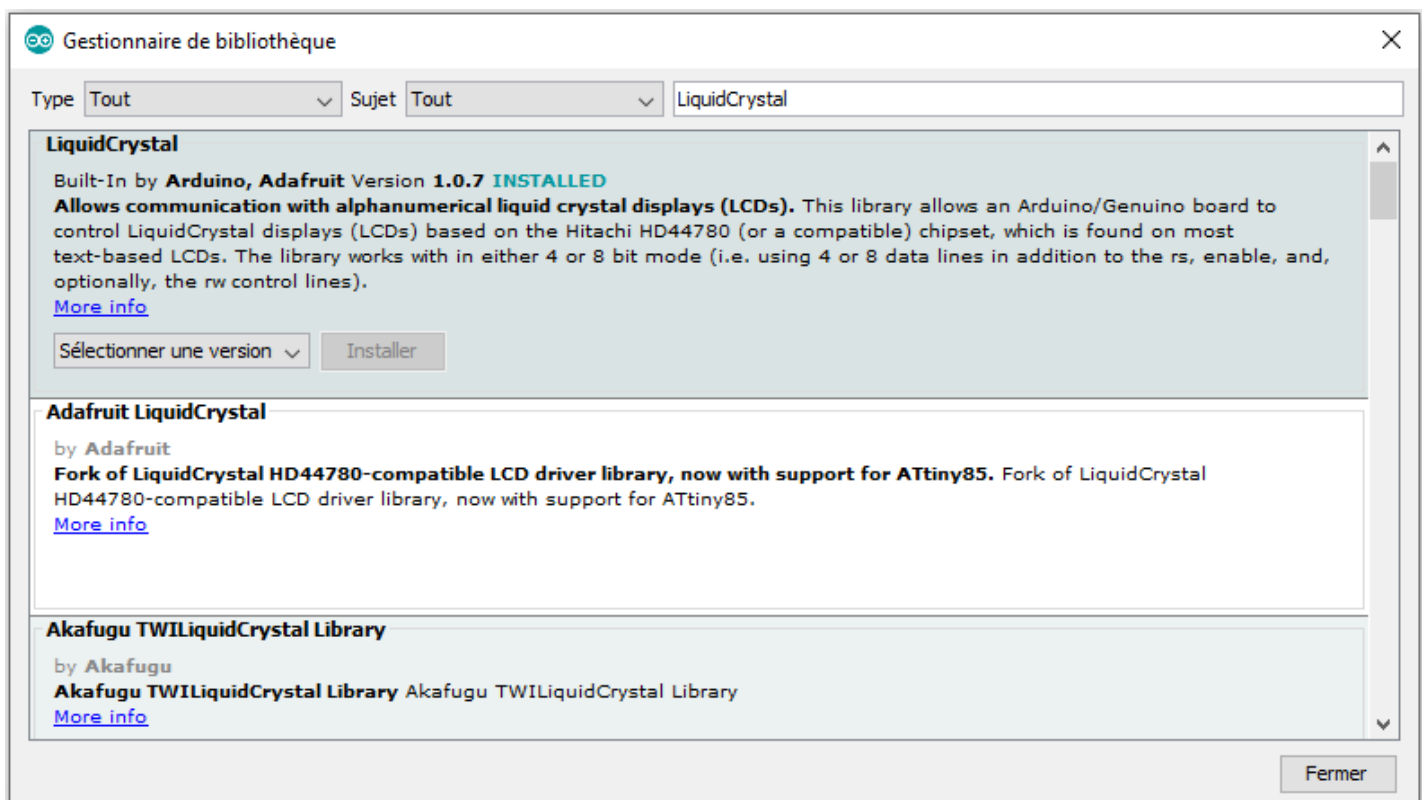
L'écran à cristaux liquides utilisé comporte deux rangées de 16 caractères, d'où son nom de LCD 1602 et le deuxième potentiomètre du montage permet de régler sa luminosité.

Le code de l'activité nécessite l'installation au préalable de la librairie " **LiquidCrystal** " d'Adafruit.

Afin d'ajouter une librairie à l'IDE Arduino, il faut aller dans le menu « **Outils ->Gérer les bibliothèques** » :



Il suffit ensuite de rechercher et d'ajouter la librairie " **LiquidCrystal** " d'Adafruit:



## . Le programme

Voici le code de l'activité :

#### Generateur

```
const int switchPin = 6;
int const PotPin=A0;
int const PinTension=9;
int PotVal=0;
int OldPotVal=0;
float Tension=0.00;
int switchState = 0;
int prevSwitchState = 0;
int state=0;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  pinMode(PinTension,OUTPUT);
  pinMode(switchPin,INPUT);
  lcd.begin(16, 2);
  Serial.begin(9600);
}

void loop() {

  switchState = digitalRead(switchPin);

  if ((switchState == HIGH)&&(prevSwitchState == LOW)) {
    state=1-state;
    delay(10);
  }
  else {
    if ((switchState == LOW)&&(prevSwitchState == HIGH)) {
      delay(10);
    }
  }
  prevSwitchState = switchState;

  if (state==1) {

    PotVal=analogRead(PotPin);
    analogWrite(PinTension,PotVal/4);

    if (PotVal != OldPotVal) {
      Tension=PotVal*5.00/1023;
      lcd.clear();
```

```
    lcd.setCursor(0, 0);  
    lcd.print("Tension (V):");  
    lcd.setCursor(0, 1);  
    lcd.print(Tension);  
    Serial.println(Tension);  
    delay(10);  
}  
  
OldPotVal=PotVal;  
  
}  
else {  
    analogWrite(PinTension,0);  
    lcd.clear();  
}  
  
}
```

## Déroulement du programme :

### – 1. Insertion des bibliothèques :

- . Insertion de la librairie " LiquidCrystal " d'Adafruit,
- . Initialisation de la librairie avec les broches utilisées pour l'écran LCD.

### – 2. Déclaration des constantes et variables :

- . const int switchPin = 6 (constante nombre entier correspondant à la broche du bouton poussoir)
- . int switchState = 0 (variable nombre entier pour stocker la valeur du potentiel de la broche du bouton poussoir)
- . int prevSwitchState = 0 (variable nombre entier pour stocker l'ancienne valeur du potentiel de la broche du bouton poussoir)
- . int state=0 (variable nombre entier correspondant à l'action à effectuer)
- . int const PotPin=A0 (constante nombre entier correspondant à la broche du premier potentiomètre)
- . int const PinTension=9 (constante nombre entier correspondant à la borne + du générateur)
- . int PotVal=0 (variable nombre entier pour stocker la valeur du potentiel de la broche du premier potentiomètre)
- . int OldPotVal=0 (variable nombre entier pour stocker l'ancienne valeur du potentiel de la broche du premier potentiomètre)
- . float Tension=0.00 (variable nombre décimal pour stocker la valeur du potentiel en V délivré sur la broche correspondant à la borne + du générateur)

### – 3. Initialisation des entrées et sorties :

- . **Initialisation du nombre de lignes et de colonnes de l'écran LCD,**
- . **Initialisation de la broche du bouton poussoir en entrée,**
- . **Initialisation de la borne + du générateur en sortie,**
- . **Initialisation de la liaison série à un débit de 9600 bauds.**

### – 4. Fonction principale en boucle :

- > **Lecture de la valeur de la broche du bouton poussoir,**
- > **Mise à jour de la variable « state » si changement de la valeur du bouton poussoir,**
- > **Mise en route ou arrêt du générateur en fonction de la variable « state »,**

- Lecture de la valeur du potentiel de la broche du premier potentiomètre,
- Application de cette valeur de potentiel sur la broche correspondant à la borne + du générateur,
- Conversion en V de la valeur du potentiel de la broche du premier potentiomètre et affichage de cette valeur sur l'écran LCD et le moniteur série, si celle-ci a changée.

