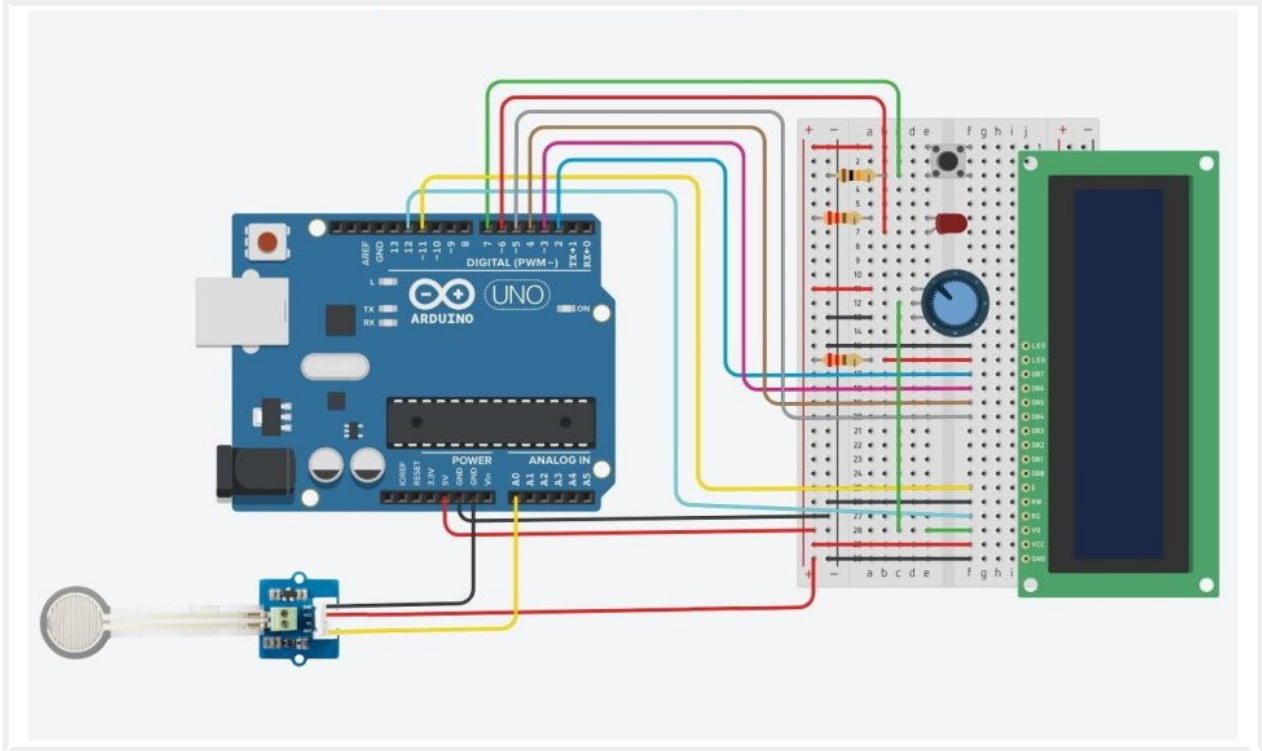


Capteur de Force

(Mesurer des forces à l'aide d'un capteur)



Liste des composants

- . 1 capteur de force Grove FSR402
- . 1 écran LCD
- . 1 bouton poussoir
- . 1 DEL rouge
- . 1 potentiomètre de 10 k Ω
- . 1 résistance de 10 k Ω
- . 2 résistances de 220 Ω
- . 1 plaque d'essais
- . Fils de connexion

Objectif

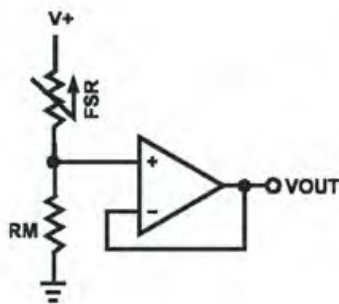
L'objectif de cette activité est de mesurer une force à l'aide d'un capteur FSR402 Grove puis d'afficher la valeur mesurée sur un écran LCD.

Le module capteur de force compatible Grove basé sur un FSR402 permet la mesure d'une pression de 0,2 à 20 N. Ce capteur communique avec un microcontrôleur type Arduino via une liaison analogique :



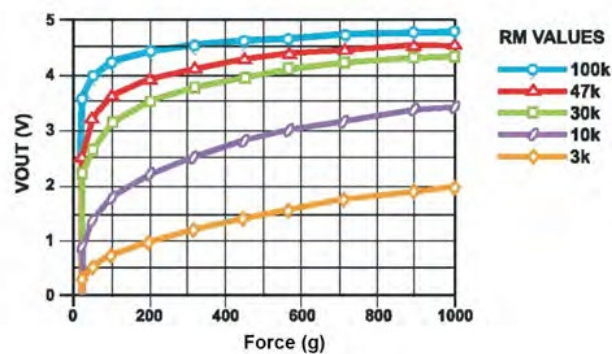
- ❶ GND: connect this module to the system GND
- ❷ VCC: you can use 5V or 3.3V for this module
- ❸ NC: none connected in this module
- ❹ OUT: output the Vout voltage signal

Pour convertir les forces exercées sur le capteur en tension, le module FSR 402 est lié à un système de mesure de résistance dans un diviseur de tension :

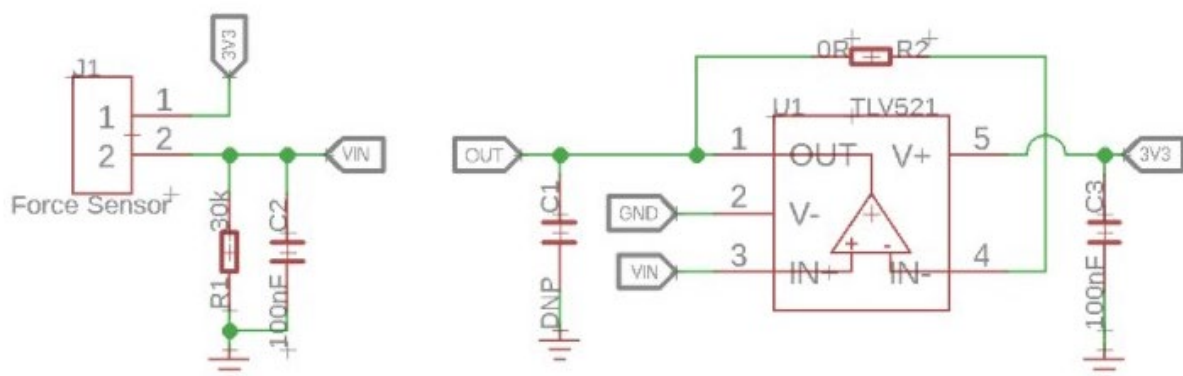


Avec :

$$V_{OUT} = \frac{R_M V +}{(R_M + R_{FSR})}$$



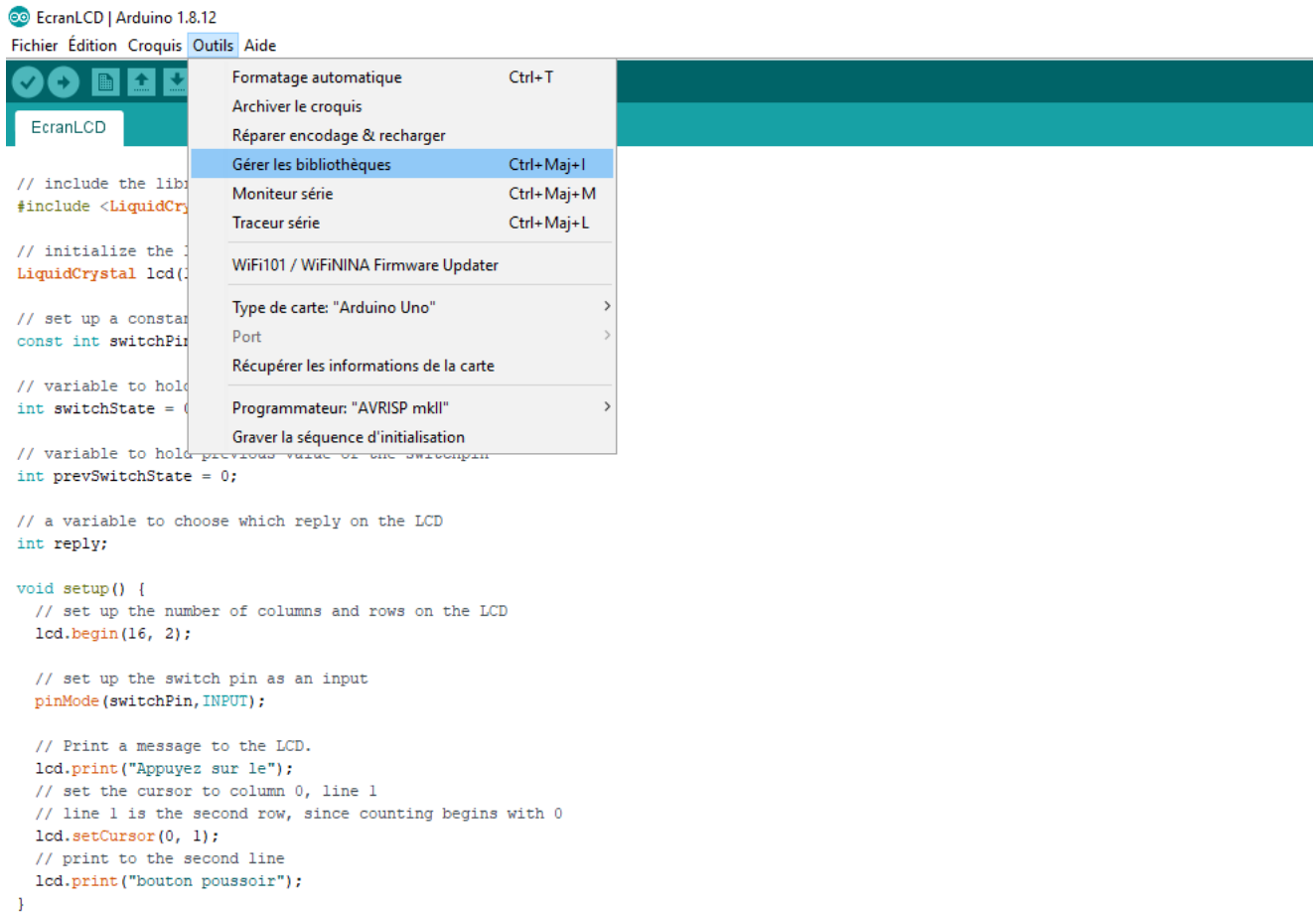
Le schéma électrique du capteur de force Grove est le suivant :



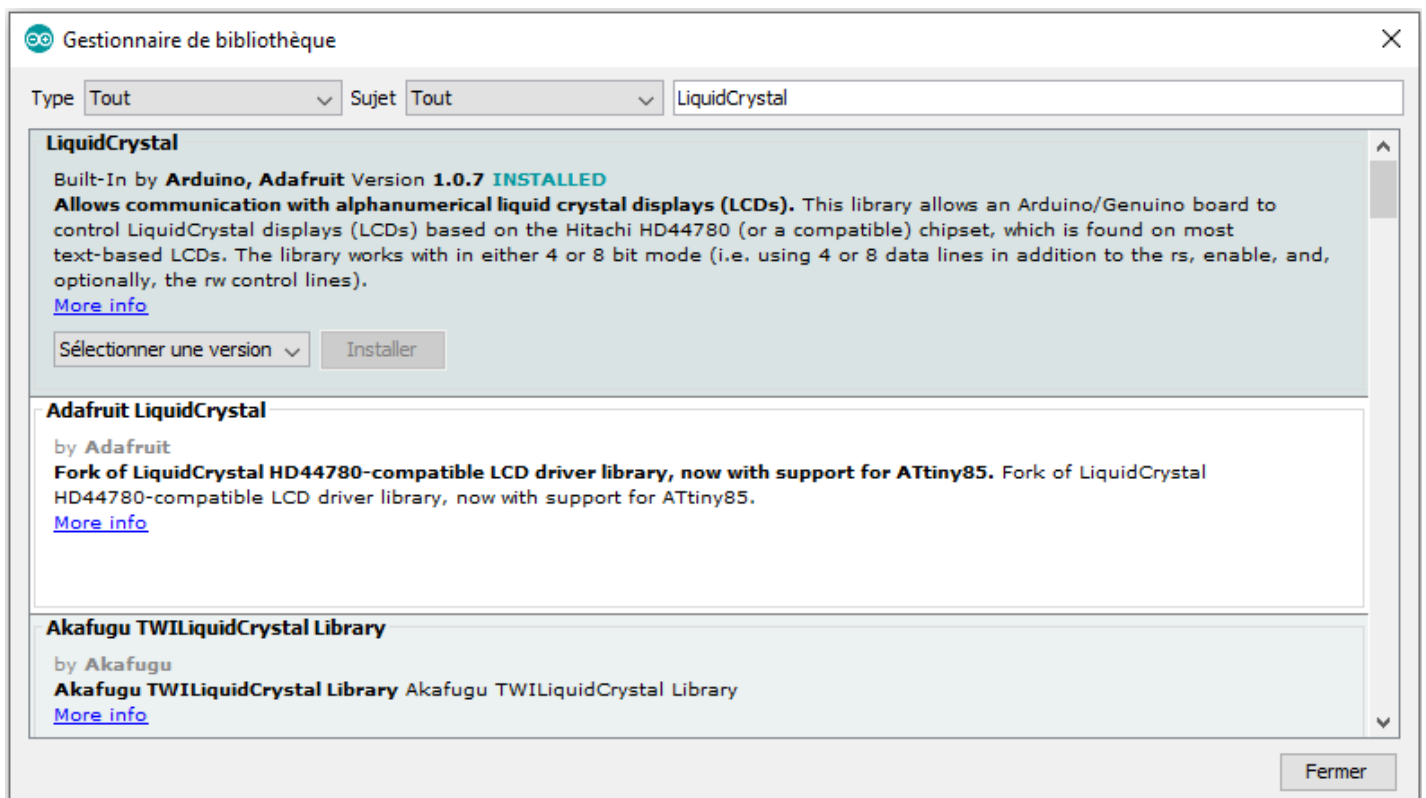
L'écran à cristaux liquides utilisé comporte deux rangées de 16 caractères, d'où son nom de LCD 1602 et le potentiomètre du montage permet de régler sa luminosité.

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 :

Force_Sensor

```
#include <LiquidCrystal.h>

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

//Constants:
const int sensorPin = A0; //pin A0 to read analog input

const int onOffSwitchStateSwitchPin = 7;
const int PinLED = 6;

int onOffSwitchState = 0;
int previousOnOffSwitchState = 0;
int state=0;

//Variables:
int value = 0; //save analog value
int previousvalue = 0;

void setup(){

    Serial.begin(9600);          //Begin serial communication
    pinMode(onOffSwitchStateSwitchPin, INPUT);
    pinMode(PinLED,OUTPUT);

    lcd.begin(16, 2);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print ("ARDUINO -");
    lcd.setCursor(0, 1);
    lcd.print ("CAPTEUR FORCE");

}

void loop(){

    onOffSwitchState = digitalRead(onOffSwitchStateSwitchPin);
    delay(1);
    if ((onOffSwitchState == HIGH)&&(previousOnOffSwitchState == LOW))
    {
        state=1-state;
        delay(10);
    }
    else
    {
        if ((onOffSwitchState == LOW)&&(previousOnOffSwitchState == HIGH))
        {
            delay(10);
        }
    }
    previousOnOffSwitchState = onOffSwitchState;
```

```

if (state==1)
{
  digitalWrite(PinLED, HIGH);
  value = analogRead(sensorPin);      //Read and save analog value from sensor
  Serial.println(value);               //Print value

  if (value!=previousvalue and value>1) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Valeur capteur:");
    lcd.setCursor(0, 1);
    lcd.print (value);
    previousvalue = value;
  }

  delay(100);                         //Small delay
}
else
{
  digitalWrite(PinLED, LOW);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print ("ARDUINO -");
  lcd.setCursor(0, 1);
  lcd.print ("CAPTEUR FORCE");

  delay(100);
}
}

```

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 sensorPin = A0** (constante nombre entier correspondant à la broche de l'arduino reliée à la sortie « signal » du capteur)
- . **const int onOffSwitchStateSwitchPin = 7** (constante nombre entier correspondant à la broche du bouton poussoir)
- . **const int PinLED = 6** (constante nombre entier correspondant à la broche de la DEL)
- . **int onOffSwitchState = 0** (variable nombre entier pour stocker la valeur du potentiel de la broche du bouton poussoir)
- . **int previousOnOffSwitchState = 0** (variable nombre entier pour stocker la précédente valeur du potentiel de la broche du bouton poussoir)
- . **int state=0** (variable nombre entier correspondant à l'action à effectuer)
- . **int value = 0** (variable nombre entier correspondant à la valeur de la broche du capteur)
- . **int previousvalue = 0** (variable nombre entier correspondant à la valeur précédente de la broche du capteur)

– 3. Initialisation des entrées et sorties :

- . Initialisation de la liaison série
- . Initialisation du nombre de lignes et de colonnes de l'écran LCD,
- . Initialisation de la broche du bouton poussoir en entrée,
- . Initialisation de la broche de la DEL en sortie,
- . Affichage du titre de l'activité « ARDUINO – CAPTEUR FORCE » sur l'écran LCD.

– 4. Fonction principale en boucle :

- > Lecture de la valeur de la broche du bouton poussoir,
- > Si le bouton poussoir est appuyé, les mesures de force commencent (la DEL est allumée),
- > Affichage des valeurs mesurées sur l'écran LCD et le moniteur série,
- > Si le bouton poussoir est de nouveau appuyé, les mesures sont arrêtées (la DEL est éteinte).