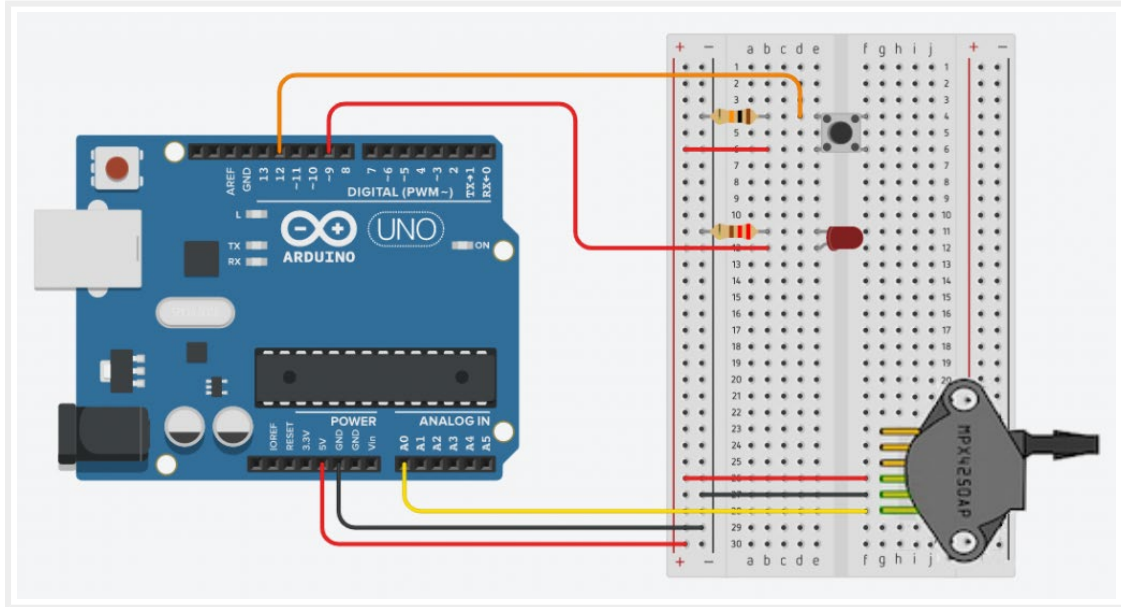


Pression – Indicateur

(Indicateur de pression à DEL)



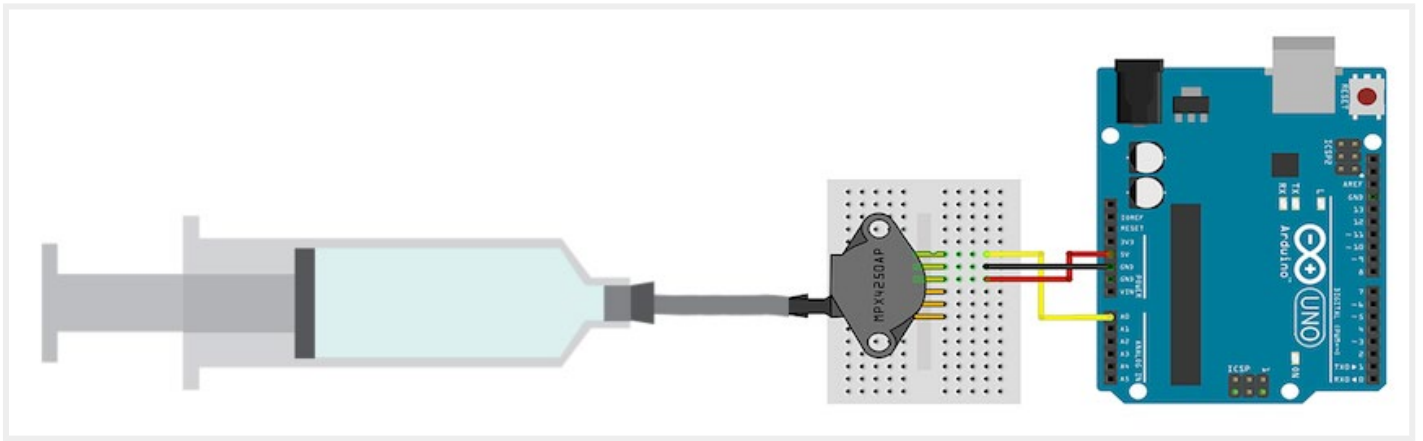
Liste des composants :

- . 1 capteur de pression MPX4250AP
- . 1 DEL rouge
- . 1 résistance de 220 Ω (résistance de protection de la DEL)
- . 1 résistance de 10 k Ω (résistance du circuit du bouton poussoir)
- . 1 bouton poussoir
- . 1 plaque d'essais
- . Fils de connexion

Objectif

De façon à s'assurer que la pression mesurée par notre capteur MPX4250AP ne soit pas supérieure ou inférieure à la pression maximale (250 kPa) ou minimale (20 kPa) admissible, nous allons dans cette activité utiliser une DEL rouge qui sera allumée quand la pression est supérieure ou inférieure à des seuils à définir afin de prévenir de leurs dépassements.

La pression mesurée sera modifiée avec une seringue d'un volume utile de 60 mL fixée au capteur par l'intermédiaire d'un tuyau:



En déplaçant le piston, initialement placé sur la graduation 30 mL, on fait varier le volume de l'air enfermé dans le corps de la seringue et donc la pression appliquée sur le capteur.

. Le programme

Voici le code de l'activité :

```
Pression_indicateur

// Déclaration des constantes et variables

const int PinSensor = 0;
const int PinButton = 12;
const int PinLed = 9;
const int PMax = 200;
const int PMin = 55;

int ValSensor = 0;
float tension = 0.0;
float Pression = 0.0;
float OldPression = 0.0;

int ValButton = 0;
int OldValButton = 0;
int State = 0;
int OldState = 0;

// Initialisation des entrées et sorties

void setup() {
  Serial.begin(9600);
  pinMode(PinButton, INPUT);
  pinMode(PinLed, OUTPUT);
  Serial.println("Appuyez sur le bouton poussoir pour commencer les mesures.");
}

// Fonction principale en boucle

void loop() {
  ValButton = digitalRead(PinButton);
  delay(10);
```

```

if ((ValButton == HIGH) && (OldValButton == LOW))
{
    State=1-State;
}
OldValButton = ValButton;

if (State==1)
{
    if (OldState == 0)
    {
        Serial.println("Mesure de la pression en cours.");
        Serial.println("");
        Serial.println ("Pression (en kPa):");
        OldState=1;
    }
    ValSensor = analogRead(PinSensor);
    tension = (ValSensor/1023.0)*5.0;

    Pression = tension / 0.02 + 10;

    if (OldPression != Pression)
    {
        Serial.println(Pression,1);
        OldPression = Pression;
    }

    if (Pression>PMax or Pression<PMin)
    {
        digitalWrite(PinLed,HIGH);
    }
    else {
        digitalWrite(PinLed,LOW);
    }
    delay(500);
}
else
{
    {
        if (OldState == 1){
            Serial.println("Fin des mesures.");
            digitalWrite(PinLed,LOW);
            OldState = 0;}
        }
}

```

Déroulement du programme :

– 1. Déclaration des constantes et variables :

- . **const int PinSensor = 0** (broche du capteur de pression)
- . **const int PinButton = 12** (broche du bouton poussoir)
- . **const int PinLed = 9** (broche de la DEL)
- . **const int PMax = 200** (constante nombre entier correspondant à la valeur de la pression maximale en kPa)

- . **const int PMin = 55** (constante nombre entier correspondant à la valeur de la pression minimale en kPa)
- . **int ValSensor = 0** (variable nombre entier pour stocker la valeur de la broche du capteur)
- . **float tension = 0.0** (variable nombre décimal pour stocker le résultat du calcul de la tension de la broche du capteur)
- . **float Pression = 0.0** (variable nombre décimal pour stocker le résultat du calcul de la pression)
- . **float OldPression = 0.0** (variable nombre décimal pour stocker le résultat du calcul de la pression précédent)
- . **int ValButton = 0** (variable nombre entier pour stocker la valeur de la broche du bouton poussoir)
- . **int OldValButton = 0** (variable nombre entier pour stocker la valeur précédente de la broche du bouton poussoir)
- . **int State = 0** (variable nombre entier correspondant à l'action à effectuer)
- . **int OldState = 0** (variable nombre entier correspondant à l'action effectuée précédemment)

– 2. Initialisation des entrées et sorties :

- . **Initialisation de la liaison série à un débit de 9600 bauds**
- . **Initialisation de la broche du bouton poussoir en entrée**
- . **Initialisation de la broche de la DEL en sortie**

– 3. Fonction principale en boucle

