2023-2024

TP1
Arduino

CHARPENTIER Baptiste
ROLAND Valentin

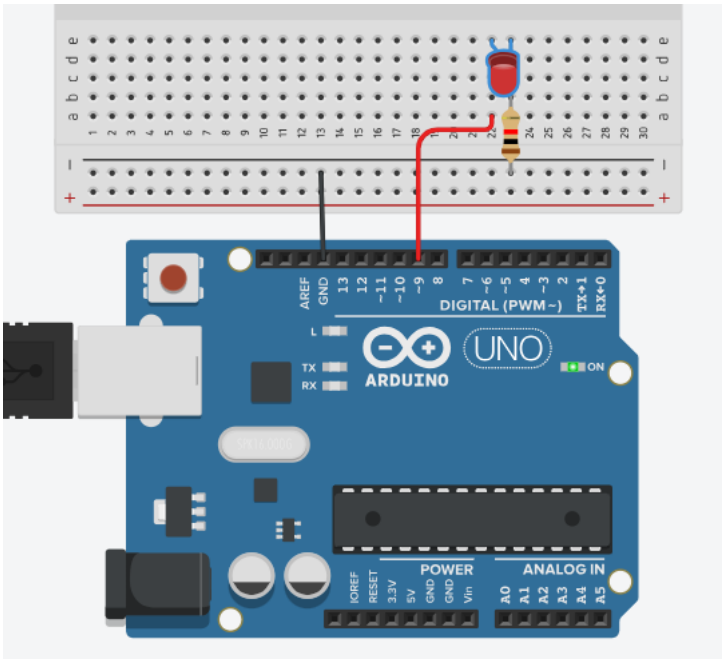
Sommaire

Introduction

L'industrie, comme tous les autres secteurs, a énormément bénéficié de l'énergie électrique. Grâce à celle-ci, l'humanité a pu connaitre son plus grand bond technologique et s'élever dans les cieux, créer des machines qui résolve et simule des situations de plus en plus complexes à gérer et même ; depuis peu ; pouvant réfléchir comme des humains. Maintenant, l'électricité fait place à l'automatisme comme avenir de l'humanité, ces machines qui peuvent dépasser l'homme dans le domaine des tâches manuelles. Lors de ce TP, nous allons faire une introduction au codage sur Arduino.

Exercice 1

1)Montage :

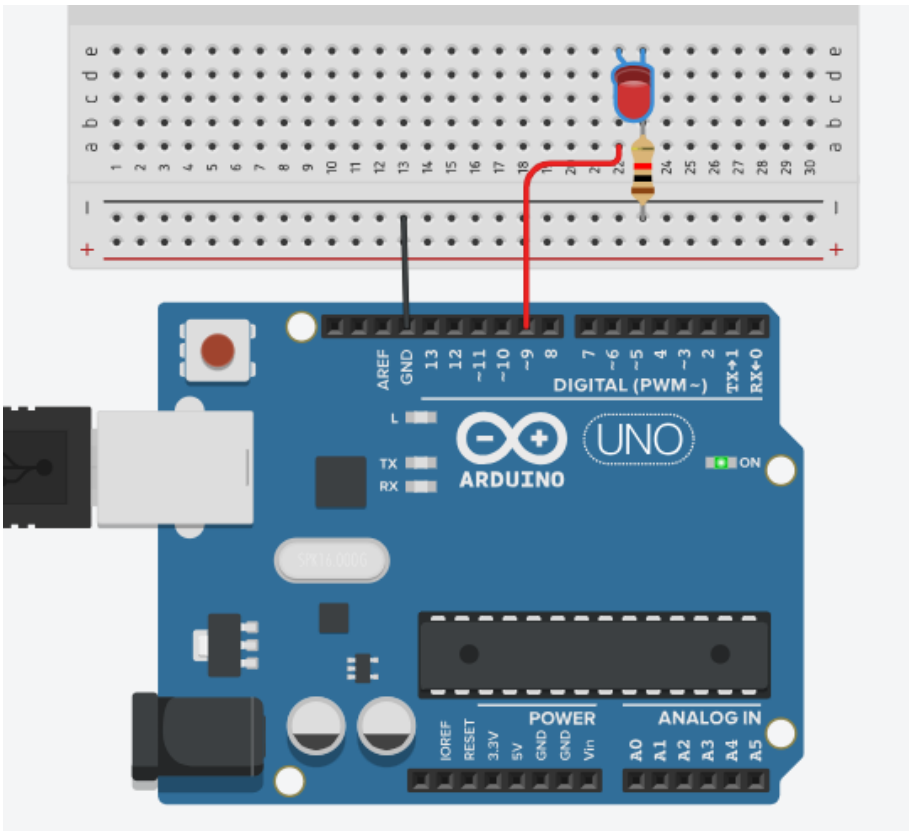


Nous avons branché la led au pin 9 qui sera utilisé pour envoyer le courant +, la masse est relié directement à l a deuxième broche de la led accompagné d'une résistance.

Code :

```
int LED = 9; //Création du port 9 pour la LED

void setup() {
   pinMode(LED, OUTPUT); //Initialisation du port 8 (LED) en sortie
}

void loop() {
  digitalWrite(LED, HIGH); //allumer la led
  delay(1000); //attendre 1s
  digitalWrite(LED, LOW); //éteindre la led
  delay(1000); //attendre 1s
}
```

2) Montage :

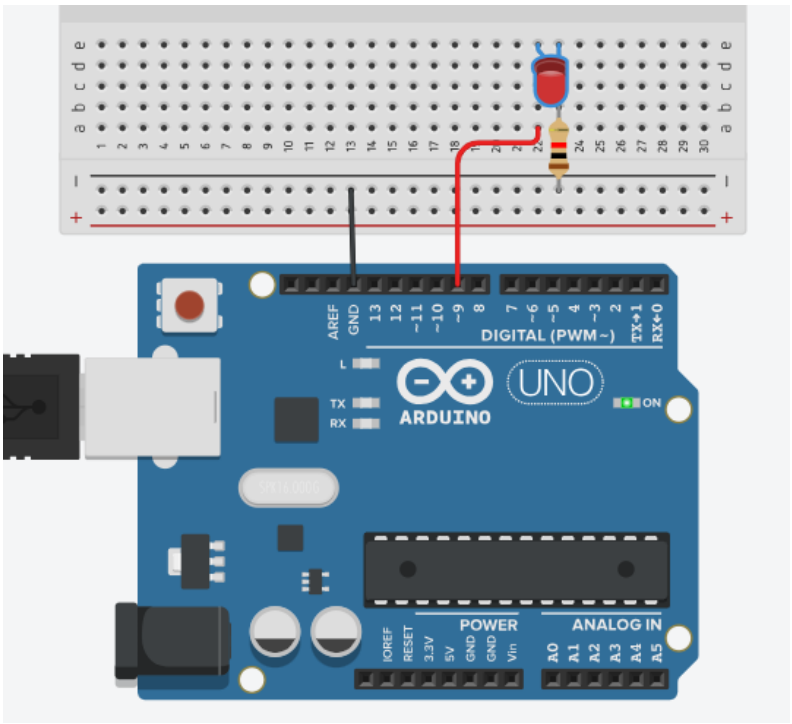


Le branchement est identique au premier montage, seul le code change.

Code :

```
int LED = 9 //création du port 9 pour la LED

void setup() {
   pinMode(LED, OUTPUT); //initialisation de la LED en sortie
}

void loop() {
  analogWrite(LED, 0); //luminausité de la led à 0%
  delay(100); //attendre 100ms
  analogWrite(LED, 64); //luminausité de la led à 25%
  delay(100); //attendre 100ms
  analogWrite(LED, 127); //luminausité de la led à 50%
  delay(100); //attendre 100ms
  analogWrite(LED, 191); //luminausité de la led à 75%
  delay(100); //attendre 100ms
  analogWrite(LED, 255); //luminausité de la led à 100%
  delay(100); //attendre 100ms
}
```

3) Montage :

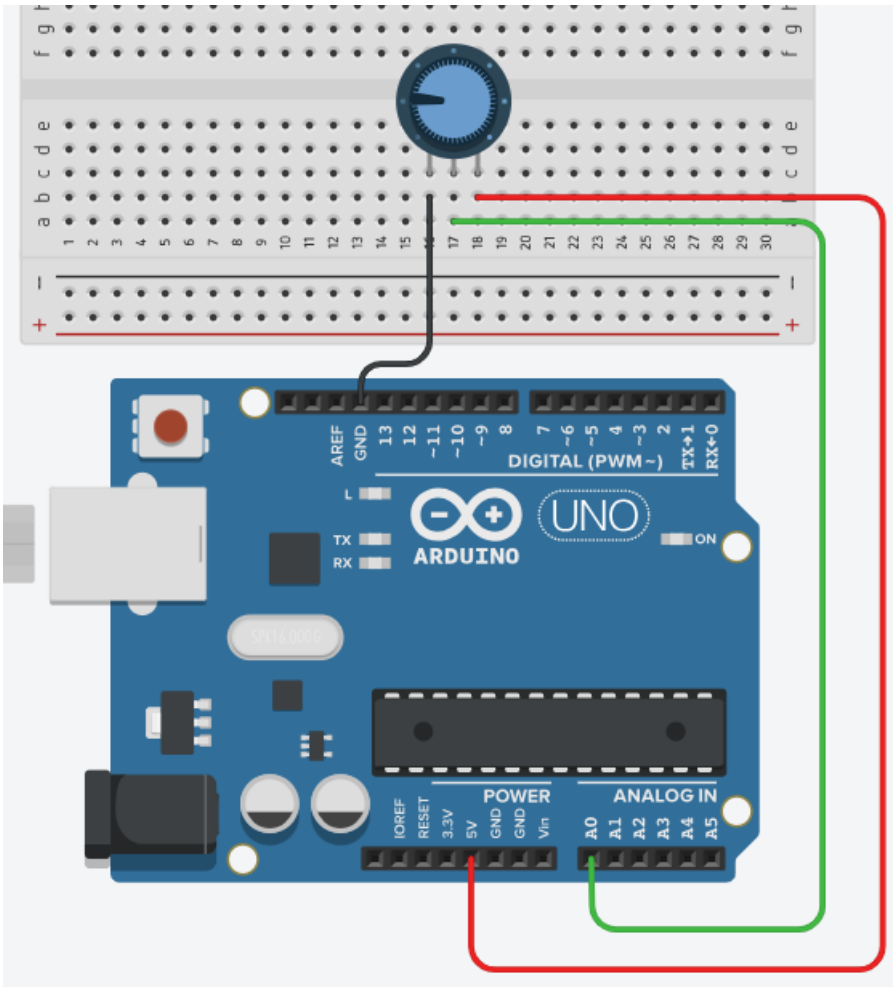


Montage toujours identique au 1.

Code :

```
int LED = 9 //création du port 9 pour la LED

void setup() {
   pinMode(LED, OUTPUT); //initialisation de la LED en sortie
}

void loop() {
  analogWrite(LED, 64); //luminausité de la led à 25%
  delay(100); //attendre 100ms
  analogWrite(LED, 191); //luminausité de la led à 75%
  delay(100); //attendre 100ms
  analogWrite(LED, 255); //luminausité de la led à 100%
  delay(100); //attendre 100ms
}
```

4)Montage :

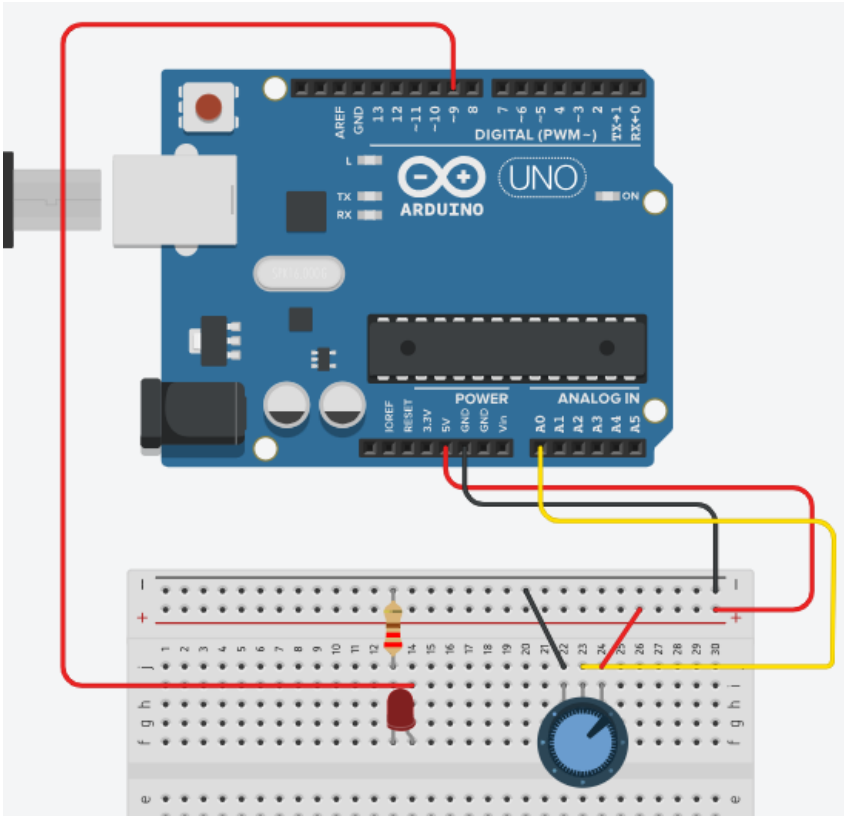


Le potentiomètre a 3 broches : 2 pour faire traverser le courant et 1(celle du milieu) pour prendre la valeur et la lire ensuite. La masse est relié au GND de la carte, le + directement sur le 5V de la carte et la broche du milieu directement sur A0.

Code :

```
int LED = 9; //Création du port 9 pour la led
int pot = A0; //création du port A0 pour le potentiomètre
int pot_value = 0; //création de la variable pot_value

void setup() {
  pinMode(LED, OUTPUT); //Initialisation de led en sortie
  pinMode(pot, INPUT); //Initialistaion de pot en entrée
  Serial.begin(9600); //Création du moniteur en fréquence 9600
}

void loop() {
  pot_value = analogRead(pot); //pot_value vaut la valeur lu du potentiomètre
  Serial.println(pot_value); //écrire la valeur lue
}
```

5) Montage :



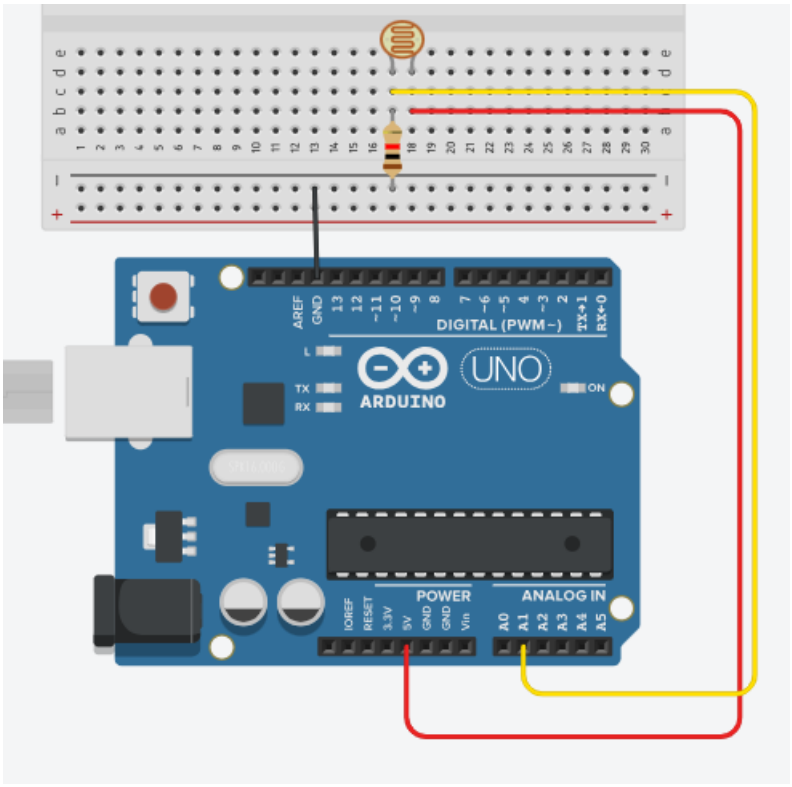
Le montage 1 et 4 sont fait en même temps :

La led est branché sur le pin 9 et le potentiomètre sur A0

Code :

```
int LED = 9; //Création du port 9 pour la led
int pot = A0; //Création du port A0 pour le potentiomètre
int pot_value = 0; //Création de la variable pot_value

void setup() {
  pinMode(LED, OUTPUT); //Initialisation de LED en sortie
  pinMode(pot, INPUT); //Initialisation de pot en entrée
}

void loop() {
  pot_value = analogRead(pot); //pot_value vaut la valeur lue de pot (port A0)
  pot_value = pot_value/4; //intensite vaut pot_value diviser par 4 (pour
attenidre 255 au max)
  analogWrite(LED, pot_value); //mettre la luminosité de la led a la valeur de
pot_value
  delay(50); //attendre 50ms
}
```

6)Montage :

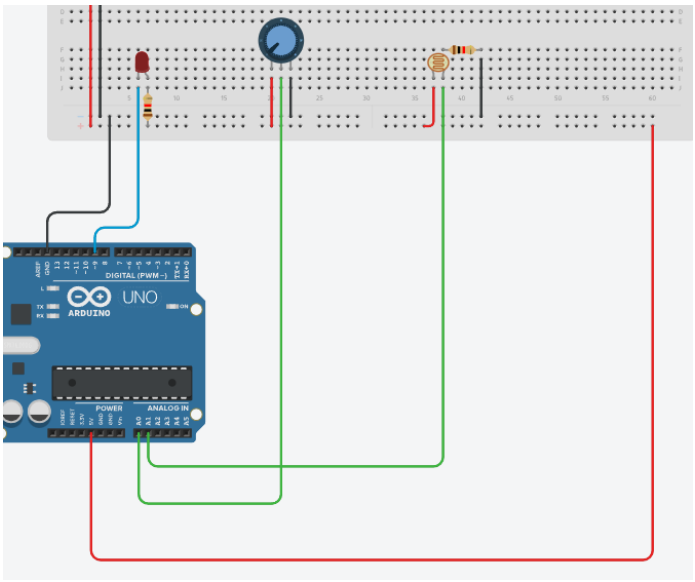


La photorésistance est branchée sur le pin A1 pour récupérer sa valeur, il est sur la même broche que la masse. La deuxième broche est sur le 5V.

Code :

```
int photo_res = A1; //Création du port A1 pour photo_res
int lumi = 0; //Création de la variable lumi

void setup() {
  Serial.begin(9600); //Connexion au moniteur 9600
  pinMode(photo_res, INPUT); //Initialisation de photo_res en entrée
}

void loop() {
  lumi = analogRead(photo_res); //lumi vaut la valeur lue de photo_res
  Serial.println(lumi); //écrire sur le moniteur les valeurs de lumi en
passant des lignes
  delay(50); //attendre 50ms
}
```

7)Montage :



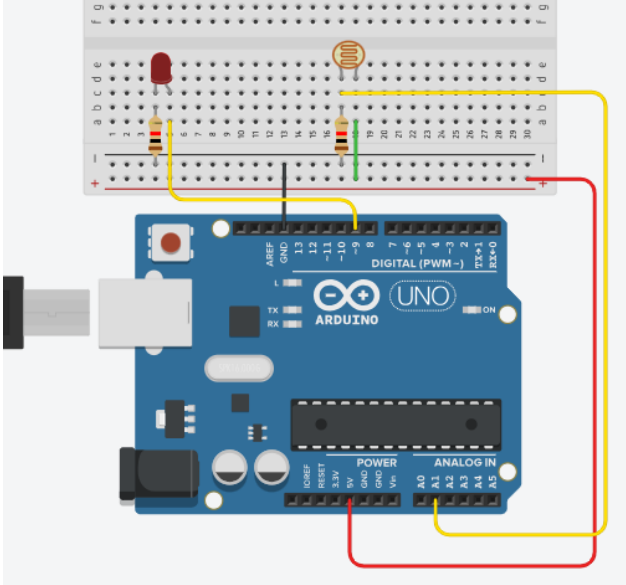
Dans ce montage, le potentiomètre est relié au A0, la photorésistance au A1 et la led au pin 9. La photorésistance et le potentiomètre sont branchés sur le 5V et le GND, la led est seulement branché au GND(car c'est le port 9 qui envoie le niveau de tension.

Code :

```
int LED = 9; //Créatio du port 9 pour la led
int pot = A0; //Création du port A0 pour le potentiomètre
int pot_value = 0; //création de la variable pot_value
int intensite = 0; //Création de la variable intensite
int photo_res = A1; //Création du port A1 pour photo_res
int lumi = 0; //Création de la variable lumi

void setup() {
  pinMode(LED, OUTPUT); //Initialisation de led en sortie
  pinMode(pot, INPUT); //Initialisation du potentiomètre en entrée
  Serial.begin(9600); //Connexion au moniteur 9600
  pinMode(photo_res, INPUT); //Initialisation de photo_res en entrée
}

void loop() {
  pot_value = analogRead(pot); //pot_value vaus la valeur lue de pot
  lumi = analogRead(photo_res); //lumi vaut la valeur lue de photo_res
  Serial.println(lumi); //écrire sur le moniteur les valeurs de lumi en
passant des lignes
  intensite = pot_value/4; //intesite vaut la valeur de pot_value diviser par
4
  analogWrite(LED, intensite); //mettre la luminosité de la led a la valeur de
intensite
  delay(50); //attendre 50ms
}
```

Exercice 2

1)Montage :

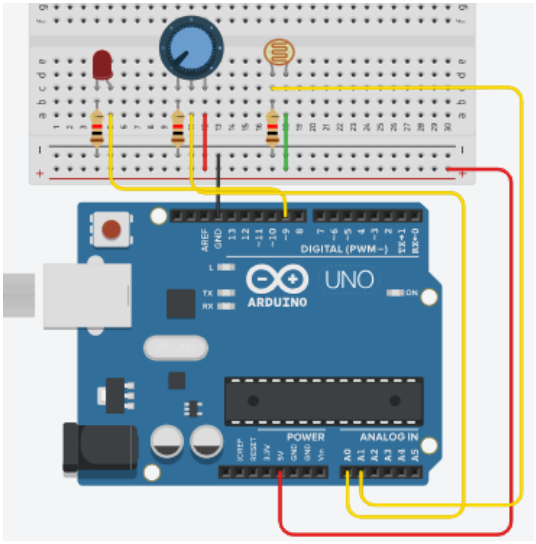


Pour le 1 de exercice 2 il faut reprendre le branchement de la led et de la photorésistance, le port de la led est connecté au port 9 de la carte en sortie et celui de la photorésistance en A1 en entrée.

Code:

```
#include <PID_v1.h>

#define PIN_INPUT 1 //défini le port A1 pour la photorésistance
#define PIN_OUTPUT 9 //défini le port 9 pour la led

double Setpoint, Input, Output; //défini les varaibles connectée entre eux

double Kp=0, Ki=10, Kd=0; //Caractéristique des paramètres
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup()
{
  Input = analogRead(PIN_INPUT); //Input vaut la valeur lue en PIN_INPUT

  myPID.SetMode(AUTOMATIC); //allume le PID
  Setpoint = 100; //point de "sensibilité"
}

void loop()
{
  Input = analogRead(PIN_INPUT); //Input vaut la valeur lue en PIN_INPUT
  myPID.Compute(); //PID en fonctionnement
  analogWrite(PIN_OUTPUT, Output); //écrire la valeur de PIN_OUTPUT en Output
}
```

2)Montage :



Pour la suite des exercices il faut rajouter le potentiomètre qui sera branché en A0 en entrée.

Montage :

```
#include <PID_v1.h>

#define PIN_INPUT 1 //défini le port A1 pour la photorésistance
#define PIN_OUTPUT 9 //défini le port 9 pour la led
#define PIN_pot 0 //défini le port A0 pour le potentiomètre

int pot = 0; //Création de variable pot

double Setpoint, Input, Output; //défini les varaibles connectée entre eux

double Kp=0, Ki=10, Kd=0; //Caractéristique des paramètres
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup()
{
  Input = analogRead(PIN_INPUT); //Input vaut la valeur lue en PIN_INPUT
  pinMode(PIN_pot, INPUT); //Initialistaion de PIN_pot en entrée

  myPID.SetMode(AUTOMATIC); //allume le PID
}

void loop()
{
  pot = analogRead(PIN_pot); //pot vaut la valeur lue en PIN_pot
  pot = pot/4; //pot vaut pot/4
  Setpoint = pot; //point de "sensibilité" en fonction de la valeur du
potentiomètre
  Input = analogRead(PIN_INPUT); //Input vaut la valeur lue en PIN_INPUT
  myPID.Compute(); //PID en fonctionnement
  analogWrite(PIN_OUTPUT, Output); //écrire la valeur de PIN_OUTPUT en Output
}
```

3)Même montage seulement le code qui change :

```
#include <PID_v1.h>

#define PIN_INPUT 1 //défini le port A1 pour la photorésistance
#define PIN_OUTPUT 9 //défini le port 9 pour la led
#define PIN_pot 0 //défini le port A0 pour le potentiomètre

int pot = 0; //Création de variable pot

double Setpoint, Input, Output; //défini les varaibles connectée entre eux

double Kp=0, Ki=10, Kd=0; //Caractéristique des paramètres
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

void setup()
{
  Input = analogRead(PIN_INPUT); //Input vaut la valeur lue en PIN_INPUT
  pinMode(PIN_pot, INPUT); //Initialistaion de PIN_pot en entrée

  myPID.SetMode(AUTOMATIC); //allume le PID
  Serial.begin(9600); //synchronise avec le moniteur 9600Hz
}

void loop()
{
  pot = analogRead(PIN_pot); //pot vaut la valeur lue en PIN_pot
  pot = pot/4; //pot vaut pot/4
  Setpoint = pot; //point de "sensibilité" en fonction de la valeur du
potentiomètre
  Input = analogRead(PIN_INPUT); //Input vaut la valeur lue en PIN_INPUT
  myPID.Compute(); //PID en fonctionnement
  analogWrite(PIN_OUTPUT, Output); //écrire la valeur de PIN_OUTPUT en Output
  Serial.println(Output); //écrire sur le moniteur la valeur "output"
  Serial.print("   "); //écrire un espace
  Serial.println(pot); //écrire la valeur de "po"
}
```

4)code :

```
#include <PID_v1.h>
const int photores = A1; // entree photoresistance
const int pot = A0; // entree potentiometre
const int led = 9; // pin de la led
double lightLevel; // variable qui stocke le niveaue de lumiÃ¨re entrant

// parametres de rÃ©gulation
float Kp=0; // valeur initiale du gain proportionnel
float Ki=10; // valeur initiale du gain intÃ©gral
float Kd=0;  // valeur initiale du gain diffÃ©rentiel

double Consigne, Entree, Sortie;  //Variables pour stocker les valeurs
PID myPID(&Entree, &Sortie, &Consigne, Kp, Ki, Kd, DIRECT);

const int sampleRate = 1; // Variable qui dÃ©termine la vitesse Ã  laquelle le
PID tourne

// Parametrage de la communication
const long serialPing = 500; //Cela determine a quelle periode on ping la
boucle (en ms)
unsigned long now = 0; //Cette variable sert Ã  compter le temps
unsigned long lastMessage = 0; //Cette variable permet de garder une trace de
quand notre boucle a communique avec le port serie pour la derniere fois

void setup(){
  lightLevel = analogRead(photores); //Lire le niveau
  Entree = map(lightLevel, 0, 1024, 0, 255); //Changer les amplitudes
  Consigne = map(analogRead(pot), 0, 1024, 0, 255);  //Lire la consigne
  Serial.begin(9600); //Commencer une communication via le port serie
  myPID.SetMode(AUTOMATIC);  //Demarrer la regulation PID
  myPID.SetSampleTime(sampleRate); //Regle la periode d'echantillonnage

  Serial.println("Debut"); // On commence
  lastMessage = millis(); // Marque du temps
}

void loop(){
  Consigne = map(analogRead(pot), 0, 1024, 0, 255); //Lire la consigne
  lightLevel = analogRead(photores); //Obtenir le niveau lumineux
  Entree = map(lightLevel, 0, 900, 0, 255); //En deduire l'entree
  myPID.Compute();  //Faire tourner la boucle PID
  analogWrite(led, Sortie);  //Ecrire la sortie du PID comme entree MLI de la
LED

  now = millis(); // Marque du temps
  if(now - lastMessage > serialPing) {  // A commenter !!!
    Serial.print("Consigne = ");
```

```
    Serial.print(Consigne);
    Serial.print(" Entree = ");
    Serial.print(Entree);
    Serial.print(" Sortie = ");
    Serial.print(Sortie);
    Serial.print("\n");
    if (Serial.available() > 0) { //Nous voulons mettre a jour Kp, Ki et Kd
      for (int x = 0; x < 4; x++) {
        switch (x) {
          case 0:
            Kp = Serial.parseFloat();
            break;
          case 1:
            Ki = Serial.parseFloat();
            break;
          case 2:
            Kd = Serial.parseFloat();
            break;
          case 3:
            for (int y = Serial.available(); y == 0; y--) {
              Serial.read();
            }
            break;
        }
      }
      Serial.print(" Kp,Ki,Kd = ");
      Serial.print(Kp);
      Serial.print(",");
      Serial.print(Ki);
      Serial.print(",");
      Serial.println(Kd);  //On affiche les valeurs pour verifier qu'elles ont
bien ete mises a jour
      myPID.SetTunings(Kp, Ki, Kd); //On regle les nouveaux parametres du PID
et on relance la regulation
    }

    lastMessage = now;
    //Marque du temps.
  }

}
```

5)code :

```
#include <PID_v1.h>
const int photores = A1; // entree photoresistance
const int pot = A0; // entree potentiometre
const int led = 9; // pin de la led
double lightLevel; // variable qui stocke le niveaue de lumiÃ¨re entrant

// parametres de rÃ©gulation
float Kp = 0; // valeur initiale du gain proportionnel
float Ki = 10; // valeur initiale du gain intÃ©gral
float Kd = 5;  // valeur initiale du gain diffÃ©rentiel

double Consigne, Entree, Sortie;  //Variables pour stocker les valeurs
PID myPID(&Entree, &Sortie, &Consigne, Kp, Ki, Kd, DIRECT);

const int sampleRate = 1; // Variable qui dÃ©termine la vitesse Ã  laquelle le
PID tourne

// Parametrage de la communication
const long serialPing = 500; //Cela determine a quelle periode on ping la
boucle (en ms)
unsigned long now = 0; //Cette variable sert Ã  compter le temps
unsigned long lastMessage = 0; //Cette variable permet de garder une trace de
quand notre boucle a communique avec le port serie pour la derniere fois

void setup(){
  lightLevel = analogRead(photores); //Lire le niveau
  Entree = map(lightLevel, 0, 1024, 0, 255); //Changer les amplitudes
  Consigne = map(analogRead(pot), 0, 1024, 0, 255);  //Lire la consigne
  Serial.begin(9600); //Commencer une communication via le port serie
  myPID.SetMode(AUTOMATIC);  //Demarrer la regulation PID
  myPID.SetSampleTime(sampleRate); //Regle la periode d'echantillonnage

  Serial.println("Debut"); // On commence
  lastMessage = millis(); // Marque du temps
}

void loop(){
  Consigne = map(analogRead(pot), 0, 1024, 0, 255); //Lire la consigne
  lightLevel = analogRead(photores); //Obtenir le niveau lumineux
  Entree = map(lightLevel, 0, 900, 0, 255); //En deduire l'entree
  myPID.Compute();  //Faire tourner la boucle PID
  analogWrite(led, Sortie);  //Ecrire la sortie du PID comme entree MLI de la
LED

  now = millis(); // Marque du temps
  if(now - lastMessage > serialPing) {  // A commenter !!!
    Serial.print("Consigne = ");
```

```
    Serial.print(Consigne);
    Serial.print(" Entree = ");
    Serial.print(Entree);
    Serial.print(" Sortie = ");
    Serial.print(Sortie);
    Serial.print("\n");
    if (Serial.available() > 0) { //Nous voulons mettre a jour Kp, Ki et Kd
      for (int x = 0; x < 4; x++) {
        switch (x) {
          case 0:
            Kp = Serial.parseFloat();
            break;
          case 1:
            Ki = Serial.parseFloat();
            break;
          case 2:
            Kd = Serial.parseFloat();
            break;
          case 3:
            for (int y = Serial.available(); y == 0; y--) {
              Serial.read();
            }
            break;
        }
      }
      Serial.print(" Kp,Ki,Kd = ");
      Serial.print(Kp);
      Serial.print(",");
      Serial.print(Ki);
      Serial.print(",");
      Serial.println(Kd);  //On affiche les valeurs pour verifier qu'elles ont
bien ete mises a jour
      myPID.SetTunings(Kp, Ki, Kd); //On regle les nouveaux parametres du PID
et on relance la regulation
    }

    lastMessage = now;
    //Marque du temps.
  }

}
```

Voici l'équation utilisé dans les systèmes d'automates :

$$MV(t) = Kp \times \left[E(t) \quad + \quad Ki \times \int E(t)dt \quad + \quad Kd \times \frac{dE(t)}{dt} \right]$$

Kp, Ki et Kd sont les gains de portions P, I et D et définissent l'intensité de chaque action.

Conclusion

Ce TP nous a permis d'utiliser plusieurs fonctions d'Arduino pour créer de petits automates.

Grâce à ça, nous pouvons maintenant coder différentes machines sur Arduino