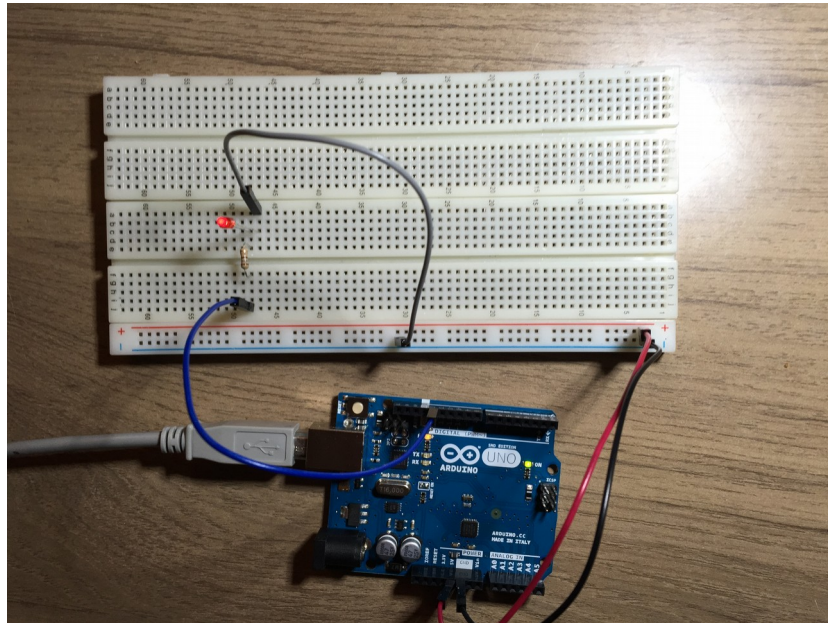


Partie 1 :

1. Installation : Suivre les instructions du TD
2. Réaliser le tuto Blink

```
int led1 = 13;
void setup() {
  pinMode(led1, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH);
  delay(1000);
  digitalWrite(led1, LOW);
  delay(1000);
}
```

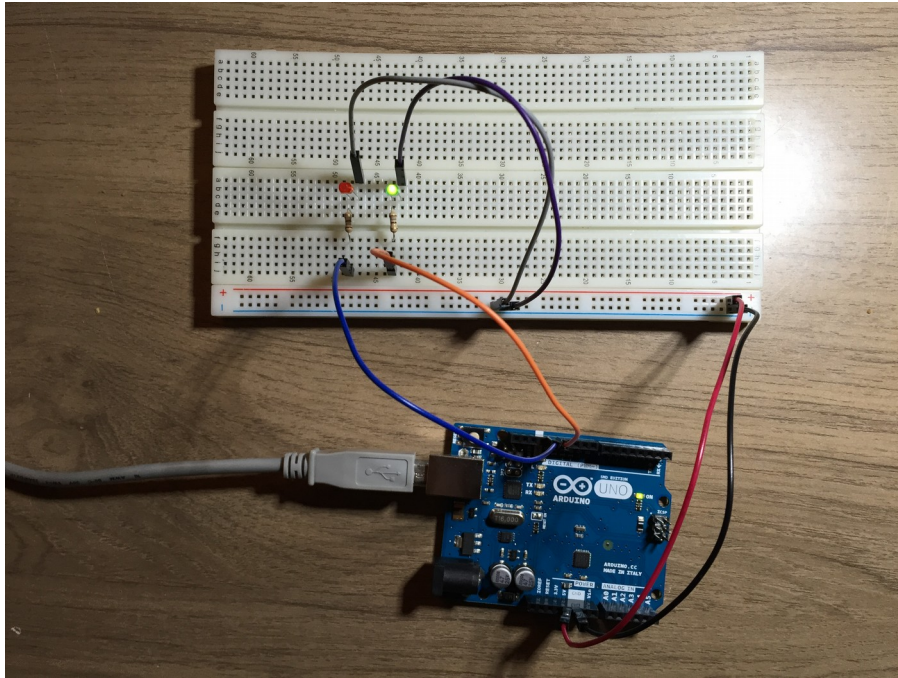


Voir la vidéo (TD1_1_1.MOV)

3. Rajouter led2
On rajoute la led 2 sur la pine 12

```
int led1 = 13;
int led2 = 12;
void setup() {
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH);
  digitalWrite(led2, LOW);
  delay(1000);
  digitalWrite(led1, LOW);
  digitalWrite(led2, HIGH);
  delay(1000);
}
```



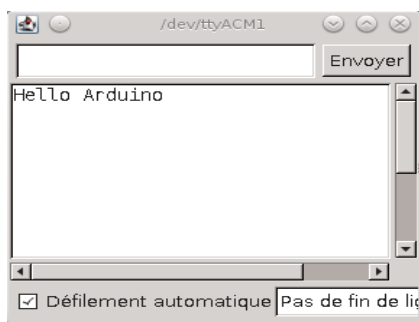
Voir la vidéo (TD1_1_2.MOV)

Partie 2 :

Utilisation de l'interface Serial Arduino

1. Hello Arduino !

```
void setup() {  
  Serial.begin(9600);  
  delay(1000);  
  Serial.println("Hello Arduino ");  
}  
void loop() {  
}
```



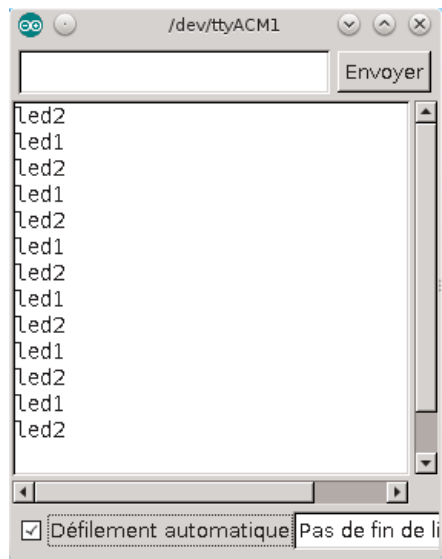
2. Afficher le numéro de la led!

```
int led1 = 13;  
int led2 = 12;  
void setup() {  
  Serial.begin(9600);  
  pinMode(led1, OUTPUT);  
  pinMode(led2, OUTPUT);  
}  
void loop() {  
  digitalWrite(led1, HIGH);  
  digitalWrite(led2, LOW);  
  Serial.println("led1");  
  delay(1000);  
  digitalWrite(led1, LOW);
```

```

digitalWrite(led2, HIGH);
Serial.println("led2");
delay(1000);
}

```



3. Morse

```

int led1 = 13;
short morseTable [26][5] = {
    {1,2,0,0,0}, // A
    {2,1,1,1,0}, // B
    {2,1,2,1,0}, // C
    {2,1,1,0,0}, // D
    {1,0,0,0,0}, // E
    {1,1,2,1,0}, // F
    {2,2,1,0,0}, // G
    {1,1,1,1,0}, // H
    {1,1,0,0,0}, // I
    {1,2,2,2,0}, // J
    {2,1,2,0,0}, // K
    {1,2,1,1,0}, // L
    {2,2,0,0,0}, // M
    {2,1,0,0,0}, // N
    {2,2,2,0,0}, // O
    {1,2,2,1,0}, // P
    {2,2,1,2,0}, // Q
    {1,2,1,0,0}, // R
    {1,1,1,0,0}, // S
    {2,0,0,0,0}, // T
    {1,1,2,0,0}, // U
    {1,1,1,2,0}, // V
    {1,2,2,0,0}, // W
    {2,1,1,2,0}, //X
    {2,1,2,2,0}, //Y
    {2,2,1,1,0} //Z
};

void allumerLed(int temp){
    digitalWrite(led1, HIGH);
    delay(temp * 500); // chaque unité représente 1/2 seconde
    digitalWrite(led1, LOW);
}

void sendMorse(char c){
    int nbr;
    short i;
    nbr = c-65; //table Ascii
    if ((nbr >=0 ) && (nbr <=26)){
        for( i=0; i<5; i++){
            allumerLed(morseTable[nbr][i]);
        }
    }
}

```

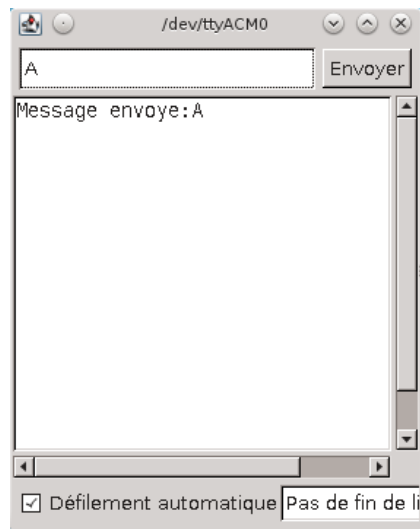
```

        delay(500); //attendre 1/2 seconde entre chaque signale
    }
}

void setup() {
    Serial.begin(9600);
    pinMode(led1, OUTPUT);
    digitalWrite(led1, LOW);
}

void loop() {
    if(Serial.available() > 0) {
        char c='\0';
        if (Serial.available()) {
            delay(2); //Pour laisser le temps au buffer
            c =Serial.read();
            sendMorse(c);
        }
        Serial.print("Message envoye:");
        Serial.println(c);
    }
}

```



Voir vidéo (TD1_2_3.MOV)

Partie 3 :

1. Question : a quoi sert la résistance de 1k Ω : Résistance de tirage ou résistance de rappel : située entre la source d'alimentation et une ligne, et qui amène délibérément cette même ligne soit à l'état haut (1 en électronique numérique) pour une résistance de tirage, soit à l'état bas (0 logique) pour une résistance de rappel. (wikipedia)
2. Afficher sur la console série la valeur obtenue sur A0.

```

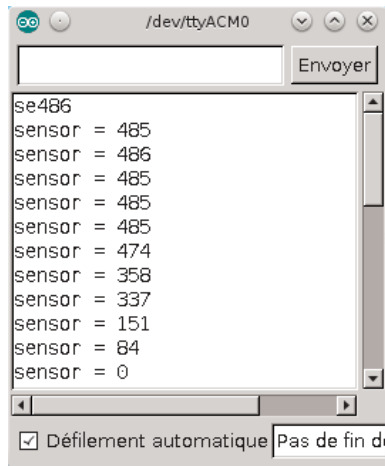
const int analogInPin = A0;

int sensorValue = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    sensorValue = anaalogRead(analogInPin);
    Serial.print("sensor = " );
    Serial.println(sensorValue);
    delay(500);
}

```



3. Remplacer la photorésistance par le potentiomètre

...

Partie 4 :

1. Écrire un programme pour que la led(led1) s'allume proportionnellement à la chute de luminosité :

```
const int sensorPin = A0;
const int led1 = 9;

int sensorValue = 0;

void setup() {
  // turn on LED to signal the start of the calibration period:
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
}

void loop() {

  sensorValue = analogRead(sensorPin);
  sensorValue = map(sensorValue, 0, 1023, 0, 255);
  sensorValue = constrain(sensorValue, 0, 255);
  analogWrite(led1, sensorValue);
}
```

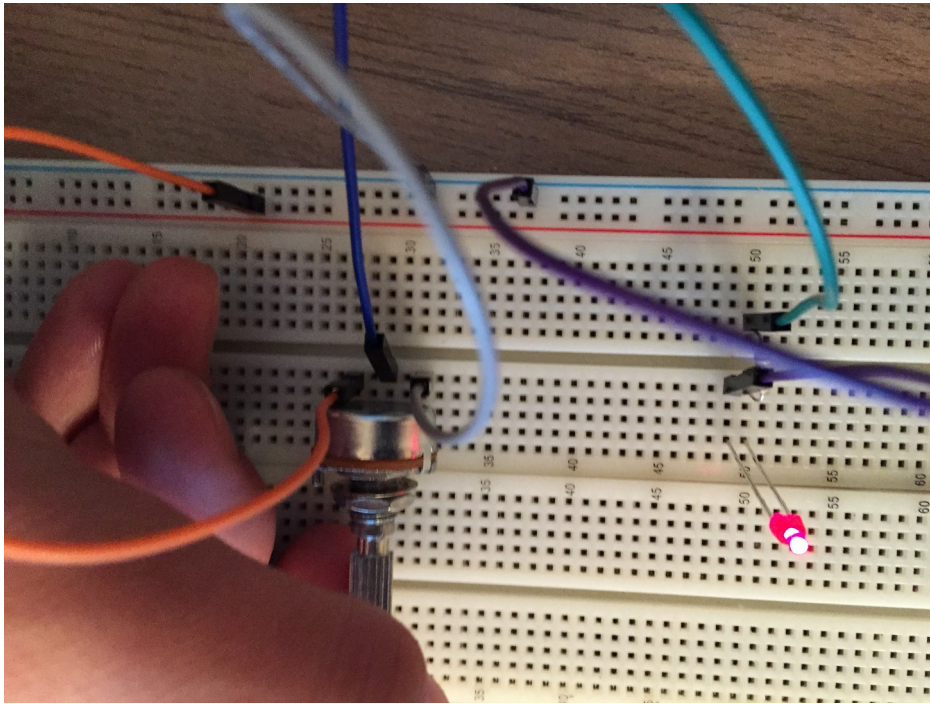
2. Calibrer

```
const int sensorPin = A0;
const int led1 = 9;
int sensorValue = 0;
int sensorMin = 1023;
int sensorMax = 0;

void setup() {
  pinMode(13, OUTPUT);
  digitalWrite(13, HIGH);
  while (millis() < 5000) {
    sensorValue = analogRead(sensorPin);
    if (sensorValue > sensorMax) {
      sensorMax = sensorValue;
    }
    if (sensorValue < sensorMin) {
      sensorMin = sensorValue;
    }
  }
  digitalWrite(13, LOW);
}

void loop() {
  sensorValue = analogRead(sensorPin);
```

```
sensorValue = map(sensorValue, sensorMin, sensorMax, 0, 255);  
sensorValue = constrain(sensorValue, 0, 255);  
analogWrite(led1, sensorValue);  
}
```



Voir vidéo (TD1_4_2)