

De l'instrument à la mesure

Master Analyse et Contrôle
Université Claude Bernard Lyon 1

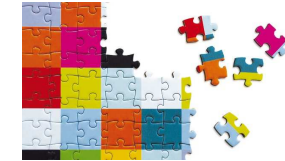
Une piste de formation... ...adaptable à différents niveaux

Construire l'instrument de mesure permet de mieux appréhender ses performances, et donc ses limites

Type de signal mesuré

Justesse et fidélité

Optimisation de l'instrument



Avoir été impliqué dans l'élaboration d'un instrument permet de mieux appréhender le fonctionnement d'autres instruments

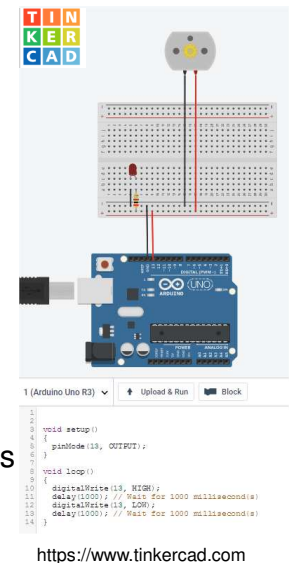
Possibilité de création d'instruments

- Electronique à bas coût
 - Basé sur des cartes Arduino opensource
- Grande variété de capteurs
- Grande variété d'actuateurs
- Facile à programmer
 - Outils de simulation : Tinkercad
 - Arduino IDE (Integrated Development Environment)



Est-ce difficile ?

- Carte électronique, composants
- Connaissances basiques d'électricité, d'informatique
- Outil de programmation pour non-spécialistes
- Bibliothèques numériques pour les capteurs et actuateurs



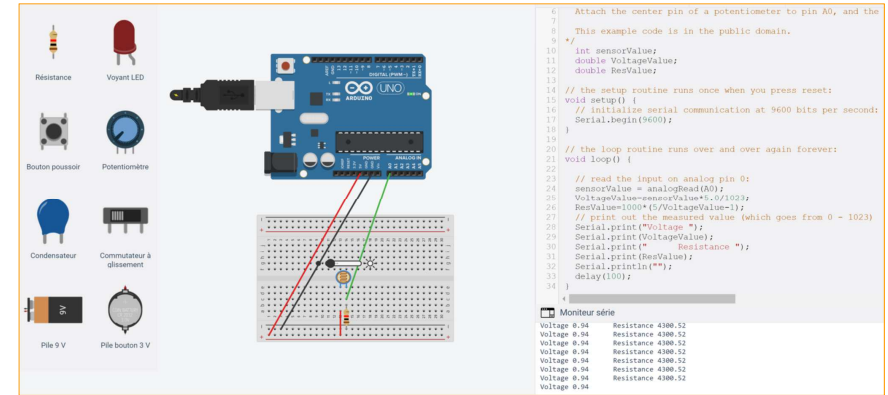
Un MOOC qui correspond à cette formation

Programmer un objet avec Arduino - session 3



<https://www.fun-mooc.fr/courses/course-v1%3AMinesTelecom%2B04017%2Bsession03/about>

Un simulateur de système Arduino



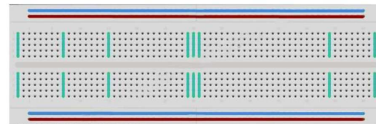
<https://www.tinkercad.com/>

L'instrumentation de base

- Carte Arduino



- Platine d'expérimentation



- Composants simples : Résistances, Diodes Electroluminescentes...

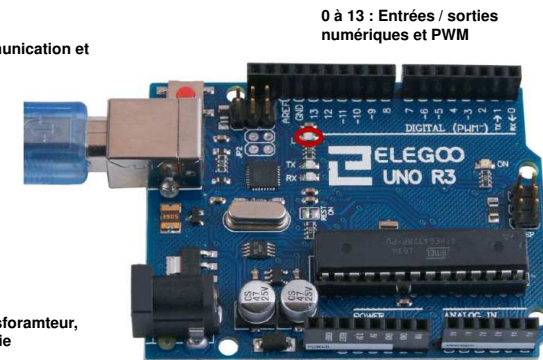


L'instrumentation de base

Carte Arduino

Port USB : communication et alimentation 5V.

Connecteur jack : alimentation transformateur, pile ou une batterie



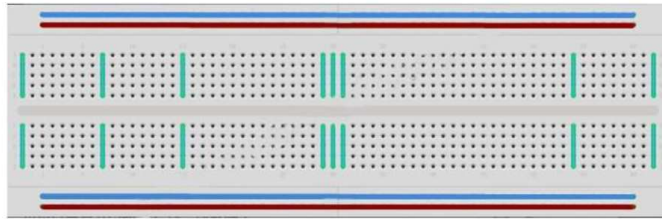
GND Terre ou masse (0V)
5V Alimentation +5V
3.3V Alimentation +3.3V
Vin Alimentation non stabilisée

A0 à A5 : Entrées / sorties analogiques

L'instrumentation de base

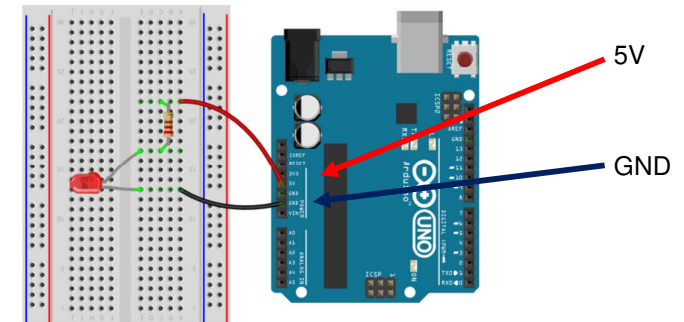
Platine d'expérimentation

Tous les connecteurs dans une colonne de 5 sont reliés entre eux
Tous les connecteurs sur les lignes rouge et bleu sont reliés entre eux



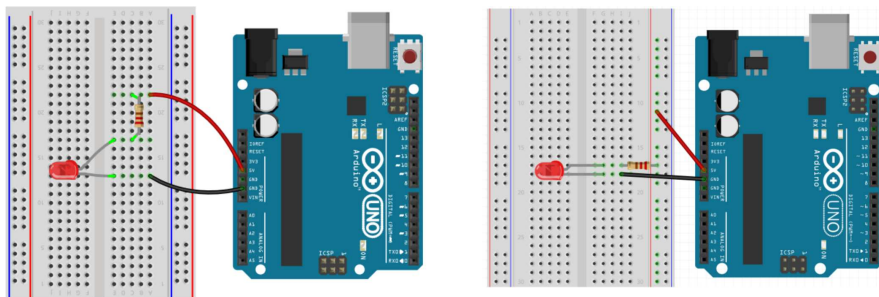
Un circuit électrique simple

- 1 LED
- 1 résistance de 220 Ω
- 2 câbles
- Branchement Carte Arduino sur un port USB => alimentation électrique



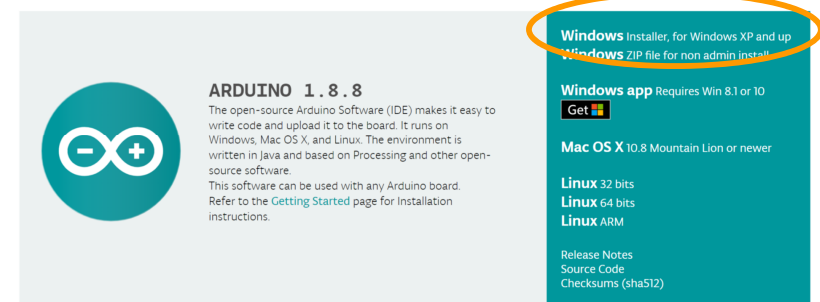
Un circuit électrique simple

Branchements équivalents ?



Faire de la carte un objet "actif"

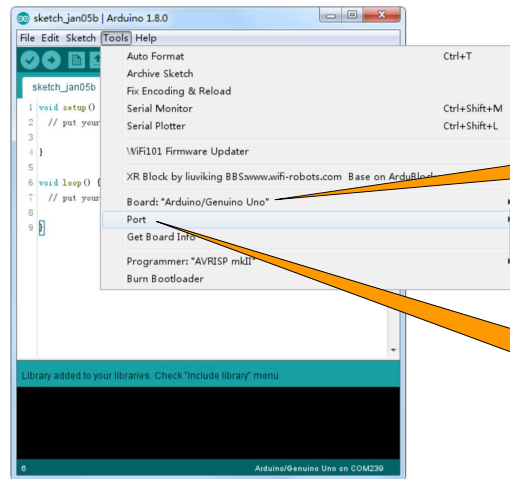
Download the Arduino IDE



<https://www.arduino.cc/en/main/software>

Puis installer le logiciel : executer arduino-1.8.8-windows.exe

Faire de la carte un objet "actif"



Type de carte

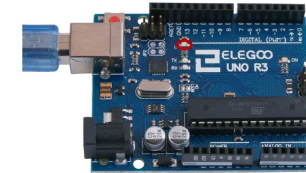
Branchement Carte
Arduino sur un port
USB => échange
d'information

Entrée/Sortie digitale (numérique)

Sortie 0 ou 5 V

LOW=0V ; HIGH=5V

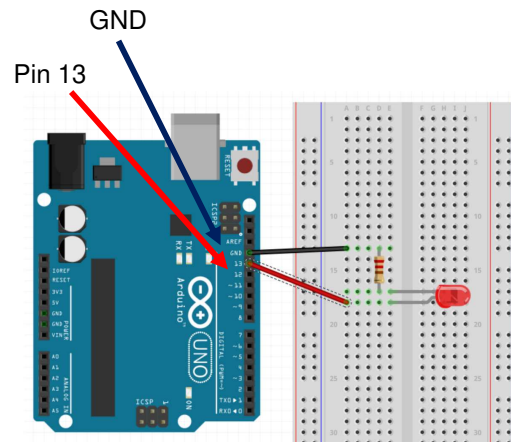
0 à 13 : Entrées / sorties
digitales et PWM



1^{er} programme : LED clignotante digitalWrite

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

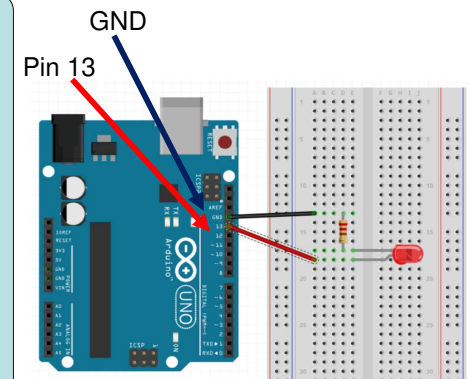
```
void loop(){  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```



1^{er} programme modifié : LED clignotante Utilisation d'une variable pour nommer le pin

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

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



Test du code

Transfert du code vers la carte

```
sketch_jan19a $  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

Déclaration
Variables

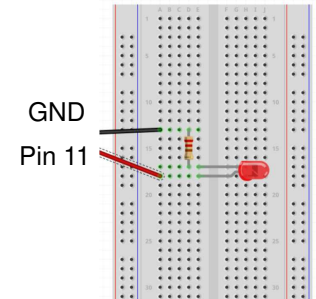
Initialisation

Programme
principal
(boucle)

2ème programme : LED clignotante

analogWrite (PWM : Pulse Width Modulation, pin 3, 5, 6, 9, 10, and 11)

```
int PinLED=11;  
  
void setup(){  
  pinMode(PinLED,OUTPUT);  
}  
  
void loop(){  
  analogWrite(PinLED, 255);  
  delay(1000);  
  analogWrite(PinLED, 0);  
  delay(1000);  
}
```

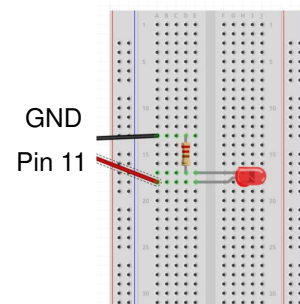


0 à 255
Modulation 0 à 5V

3ème programme : LED intensité variable

Boucle for

```
int PinLED=11;  
  
void setup(){  
  pinMode(PinLED,OUTPUT);  
}  
  
void loop(){  
  for (int i=0; i <= 255; i++){  
    analogWrite(PinLED, i);  
    delay(10);  
  }  
  analogWrite(PinLED, 0);  
  delay(1000);  
}
```



Quelques fonctions

Set up functions

pinMode (pin number, OUTPUT or INPUT); //sets one pin direction
pinMode (9, OUTPUT);
pinMode (6, INPUT);
Serial.begin(9600); // initializes the serial communication

Loop functions

SERIAL PRINT

Serial.print("character string"); // prints a text in the serial window
Serial.print(sensorValue); // prints the data sensorValue in the serial window

DELAY

delay(time_period); //waits during time_period ms

Quelques fonctions

INPUT :

digitalRead (pin number); // returns the digital level of the specified pin
SensorValue=digitalRead(6);

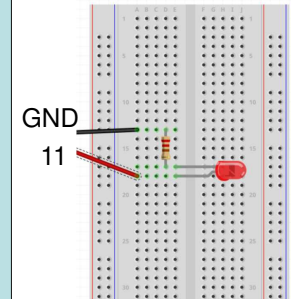
OUTPUT:

digitalWrite (pin number, HIGH or LOW); //sets the digital level of the specified pin
digitalWrite (9, LOW);

analogWrite(pin number, value between 0 and 255); //sets the analog voltage of the specified pin (3, 5, 6, 9, 10, 11)
analogWrite (9, 128); //mean voltage delivered at pin 9 is 2,5V

2ème programme modifié : Ecrire sur moniteur

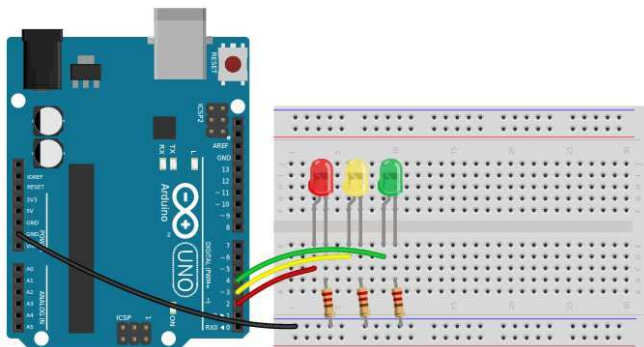
```
int PinLED=11;  
int Intensite=255;  
  
void setup(){  
  pinMode(PinLED,OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop(){  
  analogWrite(PinLED, Intensite);  
  Serial.print("Allumé : ");Serial.println(Intensite);  
  delay(1000);  
  analogWrite(PinLED, 0);  
  Serial.println("Eteint");  
  delay(1000);  
}
```



Quatrième programme : Feux tricolores

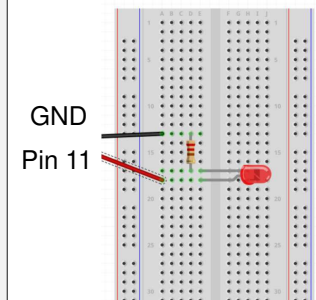
3 LED
3 résistances de 220 Ω

Ecrire sur le moniteur la couleur du feu allumé



Passer une commande via le moniteur

```
int PinLED = 11;  
String ChaineSaisie;  
  
void setup() {  
  pinMode(PinLED, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  //Teste si un texte arrive via le port série puis fait un echo  
  if (Serial.available()) {  
    ChaineSaisie = Serial.readString();  
    Serial.println(ChaineSaisie);  
  }  
  
  //Teste la commande et agit sur la LED  
  if (ChaineSaisie == "Allume") { digitalWrite(PinLED, HIGH); }  
  if (ChaineSaisie == "Eteint") { digitalWrite(PinLED, LOW); }  
  
  delay(100);  
}
```



Passer une commande via le moniteur

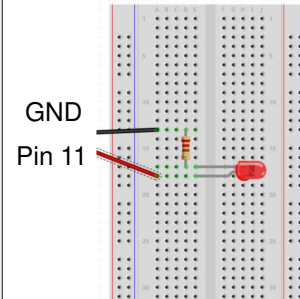
```
int PinLED = 11;
String ChaineSaisie;

void setup() {
  pinMode(PinLED,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  //Teste si un texte arrive via le port série puis fait un echo
  if (Serial.available()) {
    ChaineSaisie = "";
    while (Serial.available()) {
      ChaineSaisie = ChaineSaisie + char(Serial.read());
    }
    Serial.println(ChaineSaisie);
  }

  //Teste la commande et agit sur la LED
  if (ChaineSaisie == "Allume") { digitalWrite(PinLED, HIGH); }
  if (ChaineSaisie == "Eteint") { digitalWrite(PinLED, LOW); }

  delay(100);
}
```

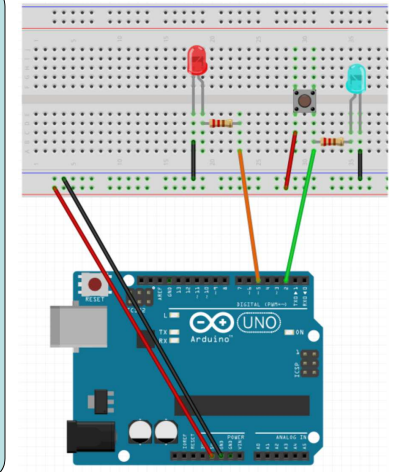


Intégration d'un bouton provoquant une action

```
int PinLED=5;
int PinTestBouton=2;

void setup(){
  pinMode(PinLED,OUTPUT);
  pinMode(PinTestBouton, INPUT);
}

void loop(){
  if (digitalRead(PinTestBouton)==HIGH){
    analogWrite(PinLED, 255);
    delay(5000);
    analogWrite(PinLED, 0);
  }
}
```



Feux tricolores avec bouton piétons

Le feu tricolore est vert pour les voitures

Quand le bouton est pressé :

- le feu orange s'allume pendant 2 secondes
- le feu rouge s'allume pendant 5 secondes
- le feu repasse au vert



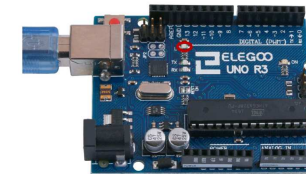
On peut aussi créer le feux piétons correspondant !



Entrée analogique

analogRead (pin number); // returns the numerical value (0-1023)
proportional to the analog voltage of the specified pin (between A0 to A5)

SensorValueA=analogRead(A0);



A0 à A5 : Entrées Analogiques

Convertisseur Analogique/Numérique 10 bits (0 à 1023)

Création d'un thermomètre

Le capteur : thermistance

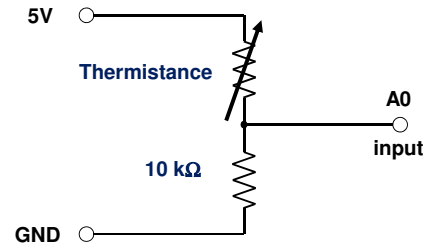


Données fiche technique

$$R_{(T)} = R_{(25)} \cdot e^{\left(B \cdot \left(\frac{1}{T} - \frac{1}{298,15}\right)\right)}$$

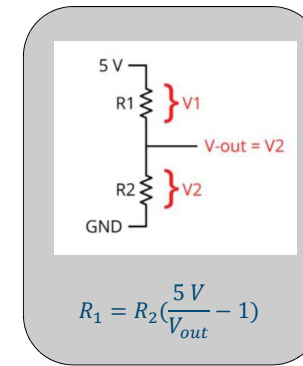
$$R_{(25)} = 10\,000\,\Omega \quad B = 3950$$

Le circuit



Création d'un thermomètre

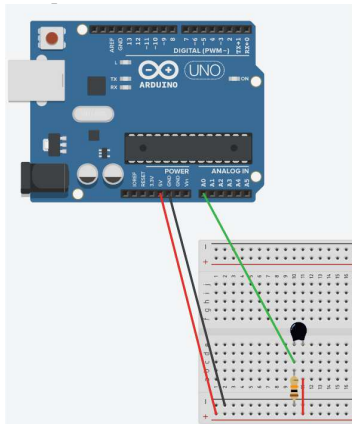
Le circuit



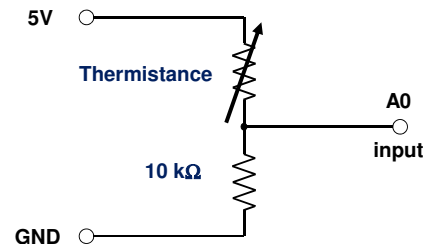
$$R_{(T)} = R_{(25)} \cdot e^{\left(B \cdot \left(\frac{1}{T} - \frac{1}{298,15}\right)\right)}$$

$$R_{(25)} = 10\,000\,\Omega \quad B = 3950$$

Création d'un thermomètre



Le circuit

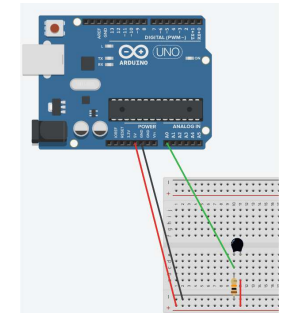


Création d'un thermomètre

```
float TensionLue;
float ResistanceCapteur;
float Temperature;
float R2 = 10000;
```

```
void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  TensionLue = analogRead(A0) * 5.0 / 1023;
  ResistanceCapteur = R2 * (5 / TensionLue - 1);
  Temperature = 1 / ((log(ResistanceCapteur / 10000)) / 3950 + 1 / 298.15);
  Serial.println(Temperature - 273.15);
  delay(200);
}
```

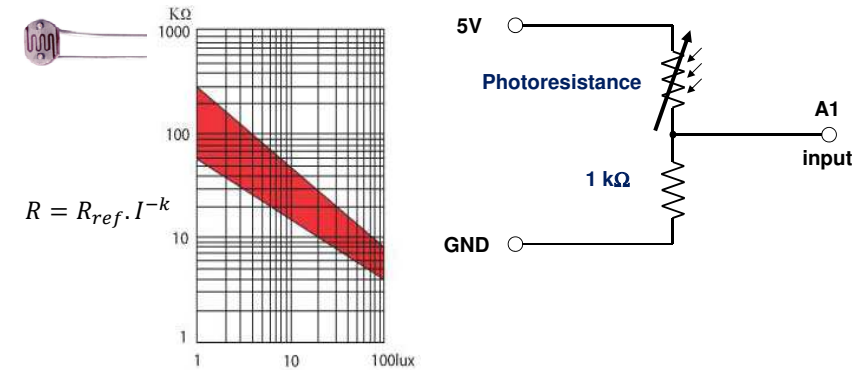


Rétroaction

- Allumer une diode bleu si $T \leq 20^\circ\text{C}$
- Allumer une diode verte si $20 < T \leq 25^\circ\text{C}$
- Allumer une diode orange si $25 < T \leq 30^\circ\text{C}$
- Allumer une diode rouge si $T > 320^\circ\text{C}$

Mesure d'un flux lumineux

Le capteur : photoresistance

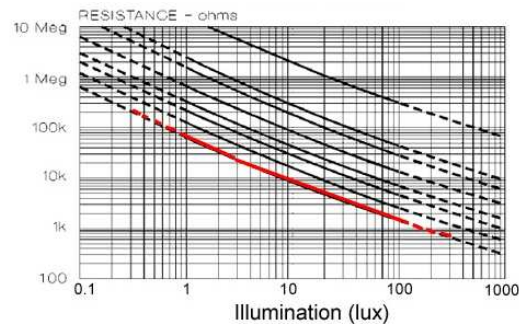


Mesure d'un flux lumineux

photoresistance



$$R = R_0 \cdot I^{-k} \Leftrightarrow \log(R) = \log(R_0) - k \cdot \log(I)$$



Mesure d'un flux lumineux



$$R = R_{ref} \cdot I^{-k}$$

$$\ln(R) = 11,6 - 0,7 \cdot \ln(I)$$

```
float TensionLue;  
float ResistanceCapteur;  
float Intensite;  
float R2 = 10000;
```

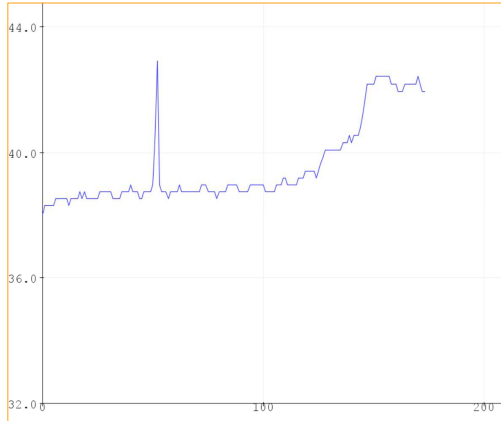
```
void setup() {  
  pinMode(A1, INPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {  
  TensionLue = analogRead(A1) * 5.0 / 1023;  
  ResistanceCapteur = R2 * (5 / TensionLue - 1);  
  Intensite = exp((log(ResistanceCapteur) - 11.6) / (-0.7));  
  Serial.println(Intensite);  
  delay(200);  
}
```

Flux lumineux, utilisation du traceur

Sans changer le programme

- Fermeture du moniteur nécessaire
- Ouverture du traceur



COM7 (Arduino/Genuino Uno)
5.18
3.54
4.70
7.86
38.54
38.98
38.98
38.98
39.20
39.20
39.20
39.20
39.20
39.20
39.20
38.98

☒ Défilement automatique

Moyenne d'une mesure

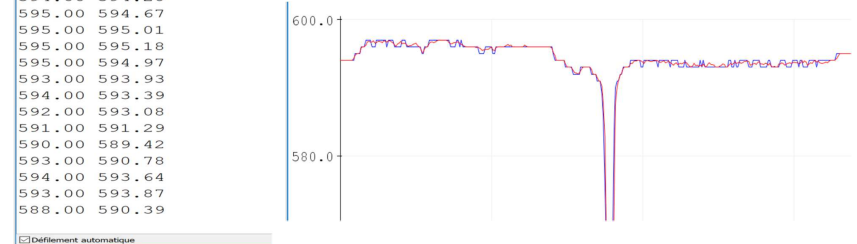
Mesurer le signal moyen sur une durée de 100 ms

Tracer en même temps

- La mesure instantanée toute les 100 ms
- La moyenne pendant le temps d'acquisition de 100 ms

```
Serial.print(SignalInst); Serial.print(" "); Serial.println(SignalMoy);
```

COM5 (Arduino/Genuino Uno)
594.00 594.20
595.00 594.67
595.00 595.01
595.00 595.18
595.00 594.97
593.00 593.93
594.00 593.39
592.00 593.08
591.00 591.29
590.00 589.42
593.00 590.78
594.00 593.64
593.00 593.87
588.00 590.39



Moyenne d'une mesure

```
int AnalogPinSuivi=A0;
float SignalInst;
float SignalMoy;
int duree=100;
```

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

```
void loop(){
  float SommeSignalLu=0; // initialise la sommation
  long NombreLecture=0;
  long MillisInit=millis(); // récupère le temps de démarrage
  while (MillisInit+duree>millis()) {
    SommeSignalLu=SommeSignalLu+analogRead(AnalogPinSuivi);
    NombreLecture= NombreLecture+1;
  }
  SignalMoy=SommeSignalLu/NombreLecture;
  SignalInst=analogRead(AnalogPinSuivi);
  Serial.print(SignalInst); Serial.print(" "); Serial.println(SignalMoy);
}
```

Rétroaction

ALERTER si le capteur se trouve dans le noir

Ecrire sur le moniteur

Allumer, faire clignoter une LED

Générer un son