
Les animations de la Vilaine Bidouille
à la Médiathèque de Redon

Saison 01 - Episode 05

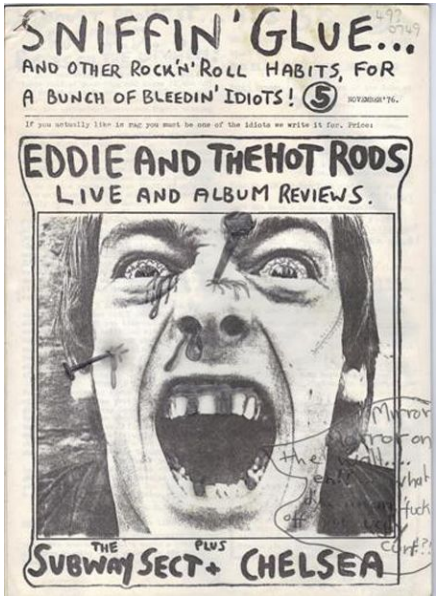
# Arduino pour le punks

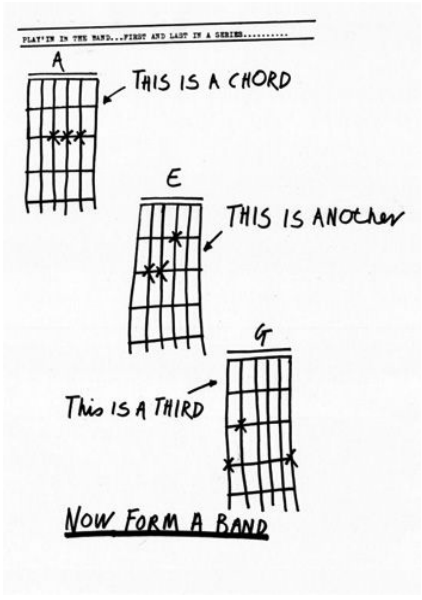
par Cyrille Laulier

LEBIOSPHERE

Clef de lecture du présent exposé :

En 1976 le mouvement punk repense les techniques d'enseignement de la musique.

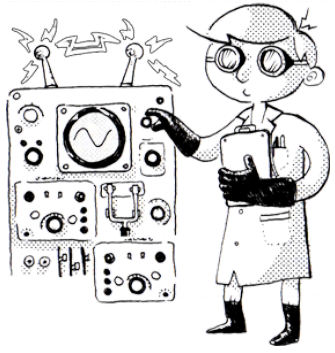



(ndlt : C'est un accord , c'en est un autre, c'en est un troisième. MAINTENANT FORMEZ UN GROUPE)

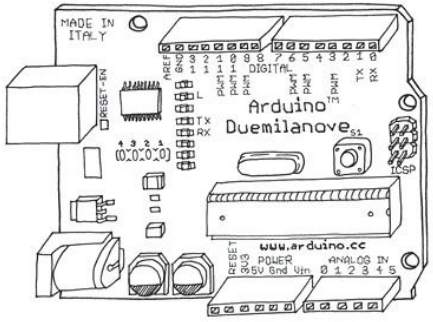
Arduino permet d'apprendre la mécatronique de la même manière.

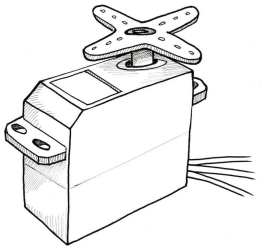
Alors ne laissez pas les experts d'un domaine vous dire que vous ne serez jamais des leurs. Ignorez les et surprenez les.

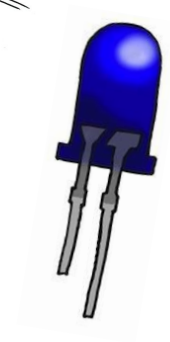


Automate :

Un automate est une boîte noire avec des entrées, des sorties et quelques lignes de code entre les deux.





CODONS !

Objectif : Ouvrir la porte du laboratoire secret selon un combinaison de bouton.


Un servomoteur représente la porte blindée.
Il y a trois boutons (A, B et C) pour taper son code.
Un buzzer représente l'alarme en cas de code erroné.
Une LED verte indique qu'on peut franchir la porte sans risque.
Une LED rouge clignote 2 secondes avant la fermeture automatique de la porte.

(ce cahier des charges est volontairement incomplet, c'est votre laboratoire secret après tout. Faites en ce que vous voulez)

# Astuces de câblage et de programmation :

## 1 - Copier c'est gagner

Sur le logiciel de programmation Arduino, ouvrez les exemples. Comprenez les. Et copier les lignes qui vous intéressent.

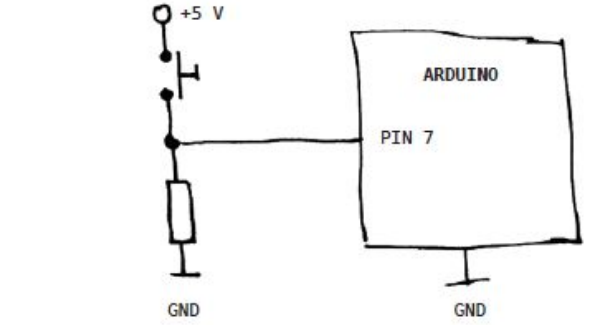
La structure est toujours la même :

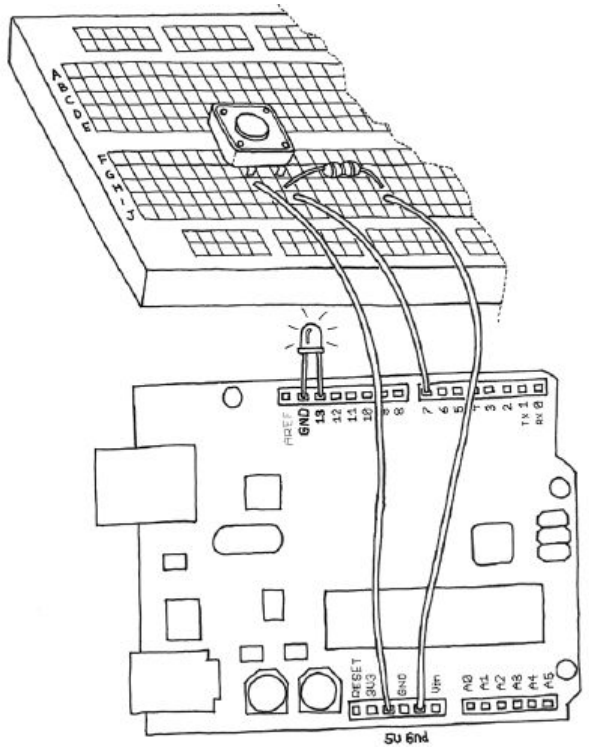
- déclarations des variables
- Programme d'initialisation (SETUP)
- Boucle à répéter à l'infini (LOOP)

## 2 - Boutons :

```
const int buttonPin = 7; // n° à changer si besoin
int buttonState =0;      // variable temporaire

void setup () {
      pinMode(buttonPin, INPUT);
}

void loop() {
      buttonState = digitalRead(buttonPin);
      ....
}
```

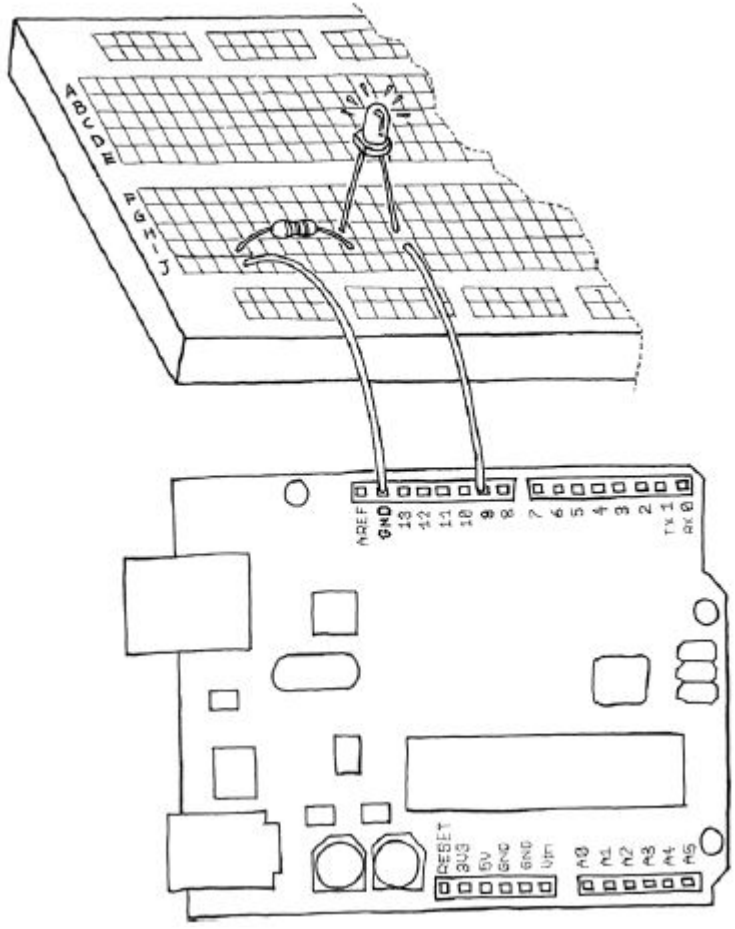
Pour le câblage, une résistance de 10K Ohm protège la carte des courts circuits



## 3 - Lumière LED :

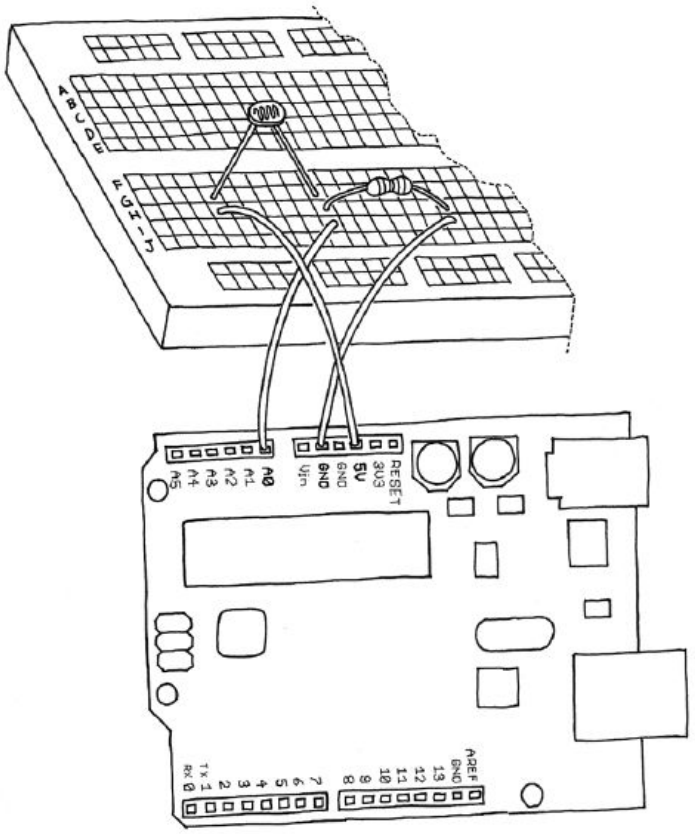
```
const int ledPin = 9;

void setup() {
      pinMode(ledPin, OUTPUT);
}

void loop() {
      digitalWrite(ledPin, HIGH); // ou LOW
      ....
}
```




Pour le câblage, une résistance de 220 Ohm suffit généralement à protéger la LED quand la carte est alimentée en 5V.

## 4 - Potentiomètre ou détecteur de lumière :

Uniquement sur les entrées analogiques A0 à A5.



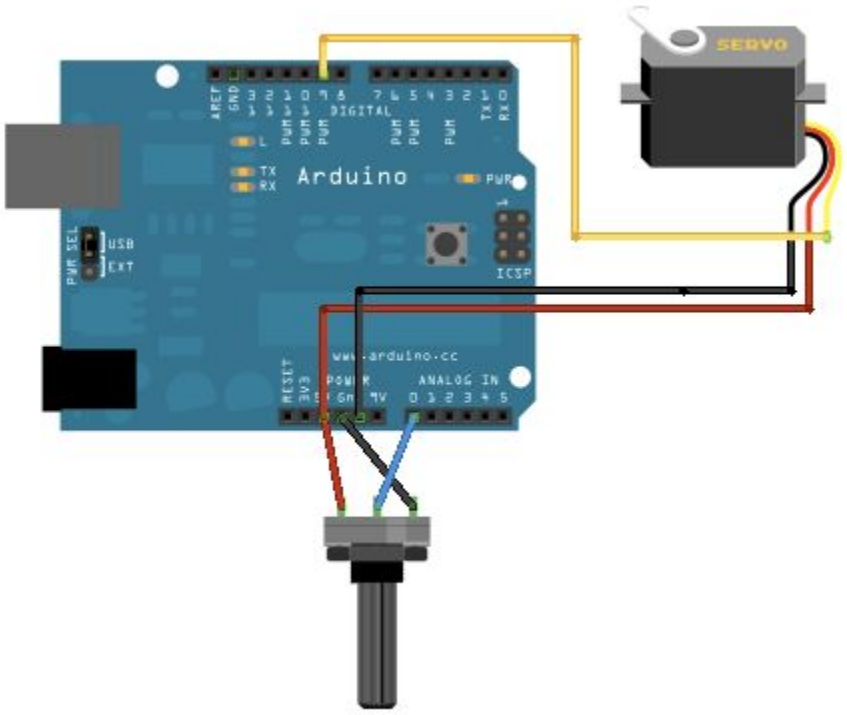
```
#define potar A0  // pas de point-virgule.
int val=0;

void setup() {
      // les entrée analogiques sont par défaut des
      entrées
}

void loop() {
      val = analogRead(potar);
      ....
}
```

Pour un potentiomètre, pas de protection à prévoir (resist if you must). Pour une résistance photosensible, il vaut mieux une résistance de 10k Ohm.

## 5 - Servomoteur :

Uniquement sur les sorties PWM

```
#include <Servo.h>

Servo myservo; // objet servomoteur
int potpin = 0;  // n° du potar
int val;    // variable

void setup() {
        myservo.attach(9);
}

void loop() {
        val = analogRead(potpin);
        val = map(val, 0, 1023, 0, 180);
        myservo.write(val);
        delay(15);
}
```

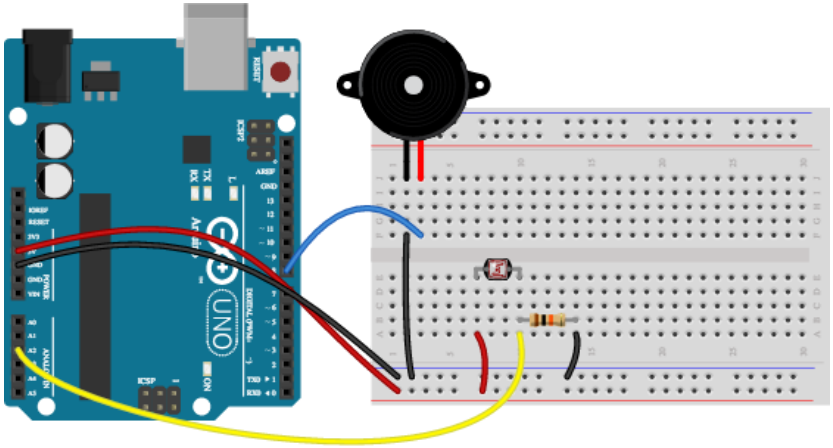


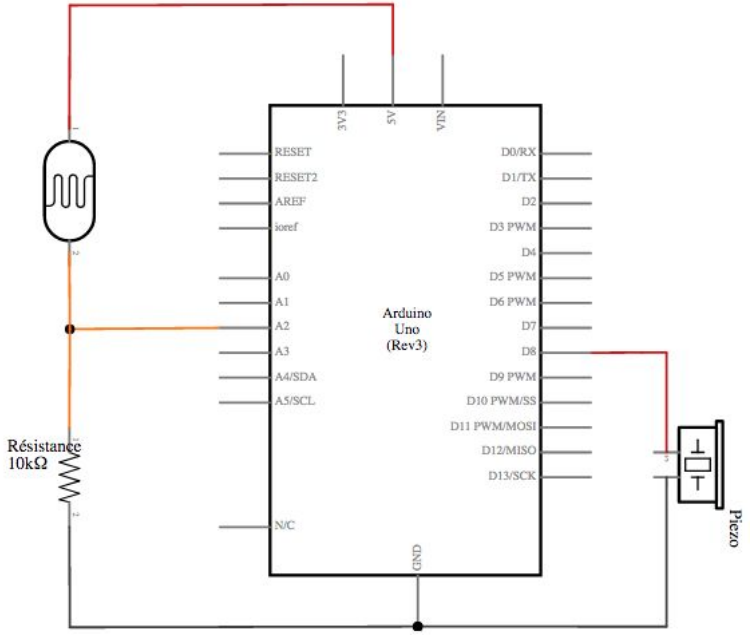

Pas mal de nouvelles notions : bibliothèque, objet, temporisation, sortie PWM ...

## 6 - buzzer :

```
int sensorValue;
int pitch;

void setup() {
}

void loop() {
  sensorValue = analogRead(A0);
  pitch = map(sensorValue, 0, 1023, 50, 30000);
  tone(8, pitch);
  delay(10);
}
```

On met la résistance pour le capteur de luminosité.





## 7 - La logique :

### Action selon une condition

```
if (condition) {
        action ;
}
else {
    autre action ;
}
```

### Boucle tant qu'une condition est vraie

```
while (condition) {
        action;
}
```