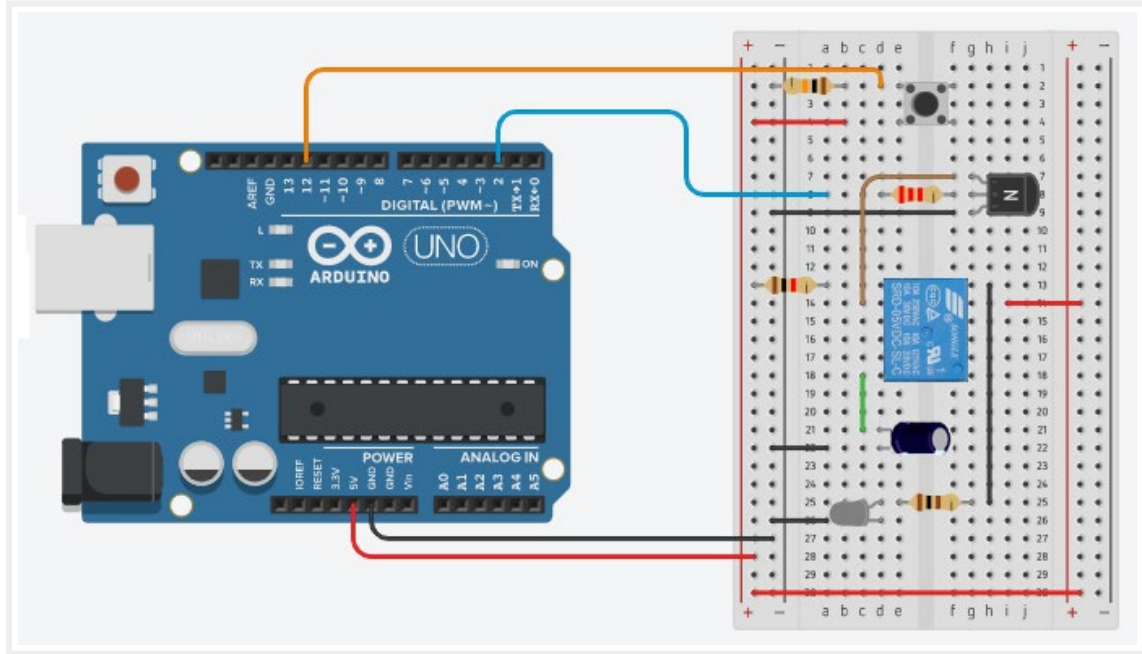


Dipôles RC – Flash

(Simulation d'un flash photographique)

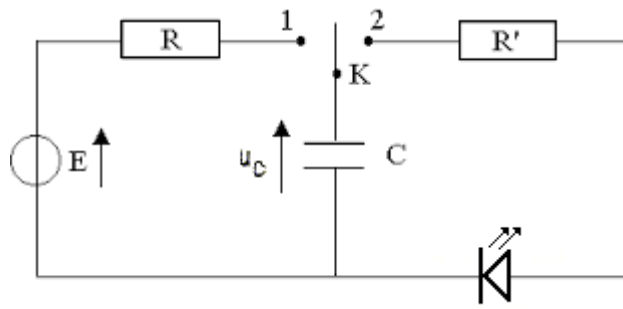


Liste des composants

- . 1 condensateur de 470 μF (C chimique : **attention à la polarité**)
- . 1 résistance de 10 $\text{k}\Omega$ (résistance du bouton poussoir)
- . 1 bouton poussoir
- . 1 transistor bipolaire NPN (BC547B)
- . 1 résistance de 2,2 $\text{k}\Omega$ (résistance du transistor)
- . 1 relais SRS-06VDC-SL
- . 1 résistance de 1 $\text{k}\Omega$ (résistance de charge du condensateur)
- . 1 résistance de 100 Ω (résistance de décharge du condensateur)
- . 1 DEL blanche

Objectif

L'objectif de cette activité est de simuler le flash d'un appareil-photo à l'aide d'une diode électroluminescente (DEL) blanche dans le circuit de décharge d'un condensateur selon le circuit suivant :



Dans ce circuit, quand l'interrupteur est en position 1, le condensateur se charge à travers la résistance R et quand l'interrupteur est en position 2, le condensateur se décharge dans la résistance R' et la DEL.

Pour simuler un flash, la DEL doit éclairer pendant une courte durée. La constante de temps du circuit de décharge doit donc être petite.

On prendra :

- . $R = 1 \text{ k}\Omega$, $R' = 100 \text{ }\Omega$, $C = 470 \text{ }\mu\text{F}$
- . $E = 5 \text{ V}$ (V_{cc} Arduino)
- . DEL blanche (tension de seuil = 2,5 V)

La constante de temps du circuit de décharge est alors :

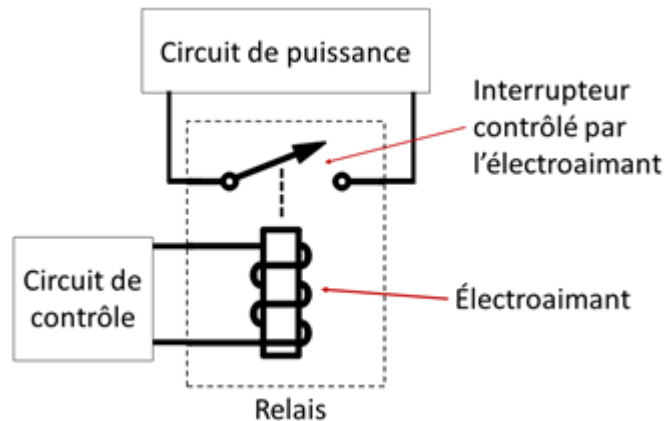
$$\tau = R' C = 100 \times 470 \cdot 10^{-6} = 47 \text{ mS}$$

On utilisera un relais électromécanique, par exemple le **SRS-06VDC-SL**, pour jouer le rôle de l'interrupteur.

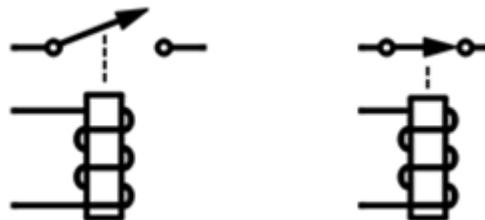
. Les relais



Un relais électromécanique, c'est un interrupteur commandé par un électroaimant. On l'utilise généralement pour isoler un circuit de commande (qui actionne l'électroaimant) d'un circuit de puissance (qui est contrôlé par l'interrupteur)



Le type de relais le plus simple est le relais SPST (« Single pole single throw »). Il est muni de 4 connecteurs : 2 connecteurs pour le contrôle de l'électroaimant, et deux connecteurs reliés à l'interrupteur.



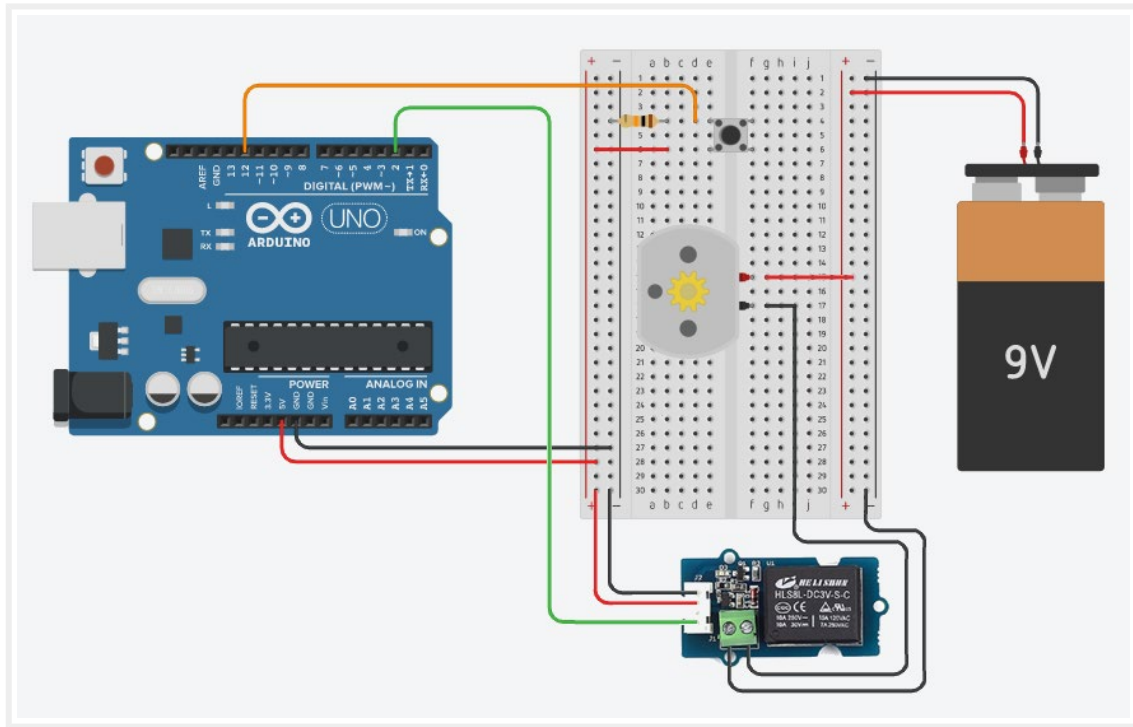
L'interrupteur bloque le courant du circuit de puissance si aucun courant ne circule dans l'électroaimant, et laisse circuler le courant du circuit de puissance si un courant circule dans l'électroaimant (on entend un petit « clic » lorsque l'interrupteur change d'état).

Par exemple, le relais **Grove 103020005** est un relais SPST qui agit comme un interrupteur normalement ouvert :



Il permet de commuter des charges plus élevées que ce que permettent les cartes Arduino. En effet, avec un relais, il est possible par exemple de contrôler un moteur fonctionnant sous 9 V continu.

Ce module est alimenté en 5V par l'Arduino et se raccorde (broche « **Signal** ») sur une sortie digitale, comme dans le circuit suivant :



Avec ce circuit, quand la sortie digitale 2 de l'Arduino est à un niveau bas, le relais agit comme un interrupteur ouvert, le circuit du moteur est donc ouvert et le moteur ne tourne pas. Au contraire, quand la sortie digitale 2 est à un niveau haut, le relais fait contact, la DEL du module s'allume, le circuit du moteur est fermé et le moteur tourne.

C'est avec le bouton poussoir qu'on donnera l'ordre à l'Arduino d'allumer ou d'éteindre le moteur :

- bouton appuyé : le moteur tourne (sortie digitale 2 à HIGH)
- bouton relâché : le moteur est arrêté (sortie digitale 2 à LOW)

Voici [le code](#) permettant de contrôler un relais Grove avec un bouton poussoir :

Relais

```
// Déclaration des constantes et variables

const int buttonPin = 12;
const int relayPin = 2;

// Initialisation des entrées et sorties

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

// Fonction principale en boucle

void loop()
{
    int buttonState = digitalRead(buttonPin);

    if (buttonState == 1)
    {
        digitalWrite(relayPin, HIGH);
    }
    else
    {
        digitalWrite(relayPin, LOW);
    }
    delay(10);
}
```

Déroulement du programme :

– 1. Déclaration des constantes et variables :

- . **const int buttonPin = 12** (broche du bouton poussoir)
- . **const int relayPin = 2** (broche du relais)

– 2. Initialisation des entrées et sorties :

- . **Initialisation de la broche du relais en sortie,**
- . **Initialisation de la broche du bouton poussoir en entrée.**

– 3. Fonction principale en boucle :

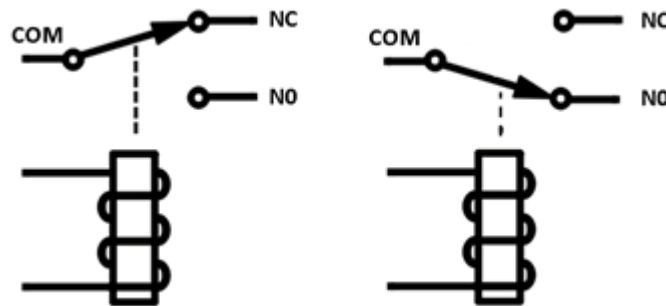
- . **Lecture de l'état logique de la broche du bouton poussoir**

. Mise à jour de la valeur de la broche du relais en fonction de la valeur de la broche du bouton poussoir :

- bouton appuyé (buttonState =1) : broche du relais à HIGH
- bouton relâché (buttonState =0) : broche du relais à LOW

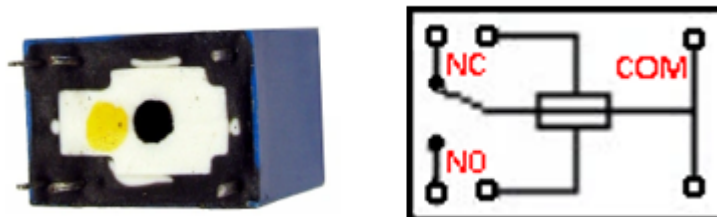
Cependant un relais SPST n'est pas adapté à notre activité. Nous allons plutôt utiliser un relais SPDT (« Single pole double throw »)

Dans ce type de relais, l'interrupteur est remplacé par un commutateur. Le relais comporte maintenant 5 connecteurs : en plus des deux connecteurs reliés à l'électroaimant, il y a un connecteur « COM » (commun), un connecteur « NC » (normally closed) et un connecteur « NO » (normally open).



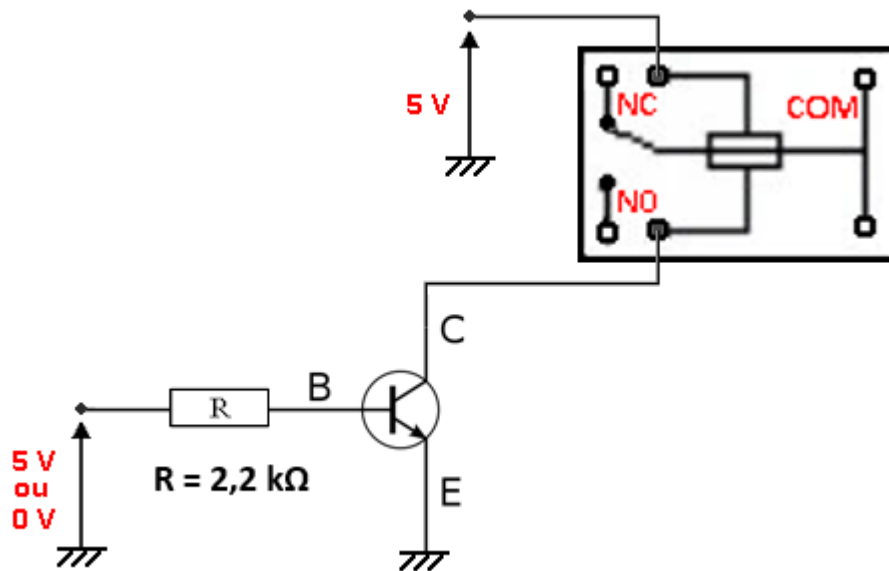
En absence ou en présence de courant circulant dans l'électroaimant, le connecteur COM est en contact avec le connecteur NC ou le connecteur NO.

Voici le schéma de câblage du relais SRS-06VDC-SL que nous allons utiliser (vue de dessous en absence de courant dans la bobine) :



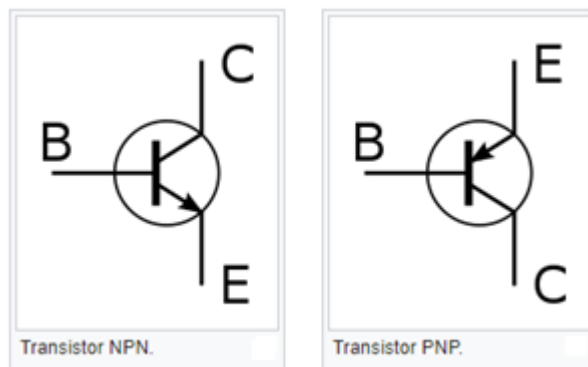
Contrairement au module Relais Grove, qui dispose déjà de son circuit de contrôle pour faire circuler ou pas un courant dans la bobine de commutation en fonction de la tension appliquée sur la broche "Signal", le relais SRS-06VDC-SL, lui n'en dispose pas.

Il faut donc ajouter un circuit de contrôle à notre relais pour commuter de la position NC à NO. Pour cela, nous allons utiliser un transistor bipolaire NPN (BC547B) en mode interrupteur commandé avec le circuit suivant :



Les transistors bipolaires (BJT pour Bipolar Junction Transistor) sont des composants à trois broches, sur lesquelles on peut appliquer une tension électrique. Les trois broches portent les noms suivants: **Collecteur**, **Base** et **Émetteur**.

On distingue deux types de transistors bipolaires : les transistors NPN et les transistors PNP.



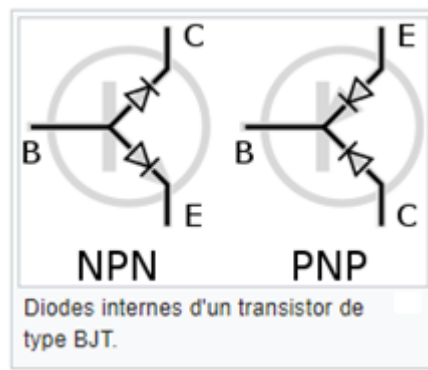
Il existe une tension entre chaque paire de broche, ainsi qu'un courant qui passe dans chaque broche. Cela fait en tout trois tensions notées V_{CE} , V_{CB} , V_{BE} et trois courants notés I_C , I_B et I_E .

Ceux-ci sont reliés par les équations suivantes :

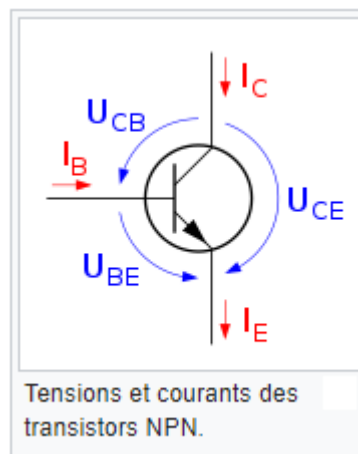
$$V_{CE} = V_{CB} + V_{BE}$$

$$I_E = I_B + I_C$$

On peut considérer, à quelques détails près, qu'un transistor est composé de deux diodes mises en série dans des sens opposés :



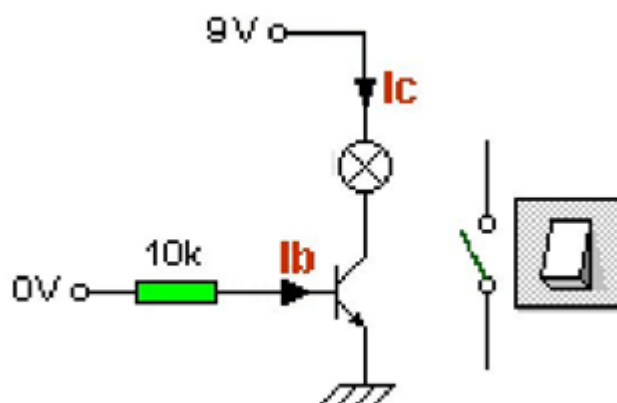
Le fonctionnement d'un transistor bipolaire NPN en interrupteur commandé consiste à activer la base, pour qu'elle permette au courant présent dans le collecteur de s'écouler jusqu'à l'émetteur.



Quand le courant de base est nul, le transistor est bloqué :

$$V_{BE} = 0 \rightarrow I_C = I_E = 0$$

Il est équivalent à un interrupteur ouvert :

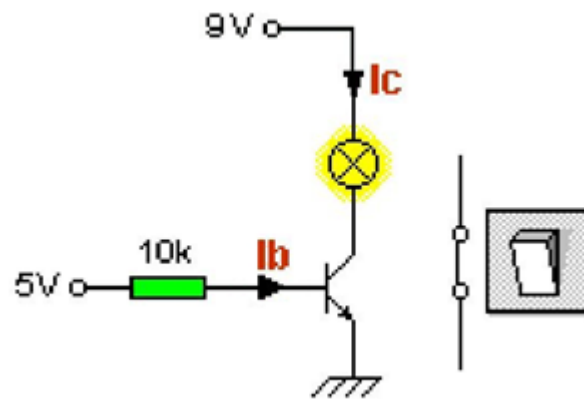


Quand le courant de la base est suffisant, le transistor est saturé :

. $V_{BE} = 0,7 \text{ V}$ (tension de seuil de la diode base-émetteur), le transistor est alors passant.

. Pour être saturé, il faut que : $I_b > I_c / \beta$ où β est le gain en courant du transistor ($I_b \times \beta = I_c$), aussi souvent appelé Hfe dans les fiches techniques des constructeurs.

Le transistor est alors équivalent à un interrupteur fermé :



La résistance de la base doit être calculée pour avoir un courant I_b suffisant.

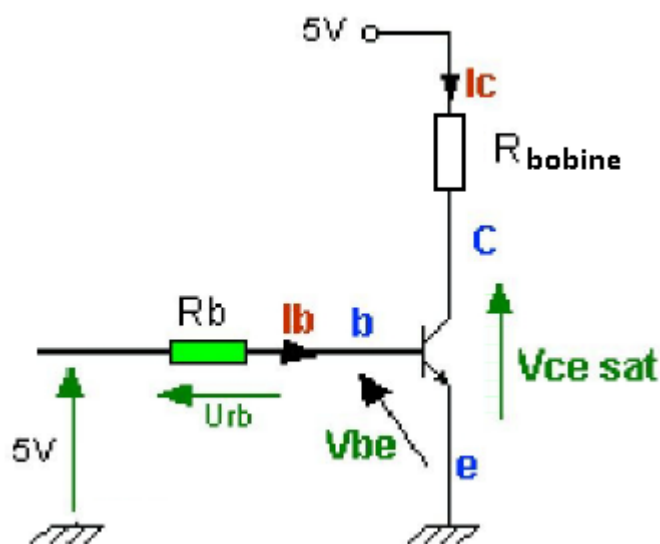
Avec le circuit de contrôle pour notre relais, on peut calculer la résistance de la base. En premier il faut déterminer le courant I_c :

$$I_c = U_{\text{bobine relais}} / R_{\text{bobine relais}}$$

$$. R_{\text{bobine}} = 120 \, \Omega$$

$$. U_{\text{bobine}} = 5 - V_{\text{ce sat}}$$

(La tension $V_{\text{ce sat}}$ est proche de 0 V mais pas nulle. $V_{\text{ce sat}} \approx 0,2 \text{ V}$)

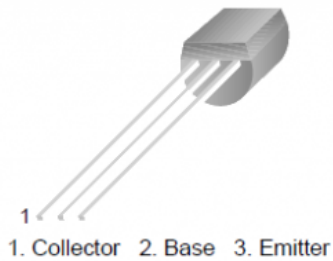


$$\text{Donc : } I_c = 4,8 / 120 = 0,04 \text{ A} = 40 \text{ mA}$$

Le courant de la base I_b doit être suffisant pour saturer le transistor:

$$I_b > I_c / \beta$$

D'après la documentation du constructeur du transistor **BC547B**, β est au moins égal à 200 :



h_{FE} Classification

| Classification | A | B | C |
|----------------|-----------|-----------|-----------|
| h_{FE} | 110 ~ 220 | 200 ~ 450 | 420 ~ 800 |

Il faut donc : $I_b \text{ min} = 40 / 200 = 0,2 \text{ mA}$

Connaissant I_b , il est maintenant possible de calculer R_b :

$$R_b = U_{Rb} / I_b$$

$$V_{BE} + U_{Rb} = 5 \text{ V} \text{ avec } V_{BE} = 0.7 \text{ V (tension de seuil de la diode)}$$

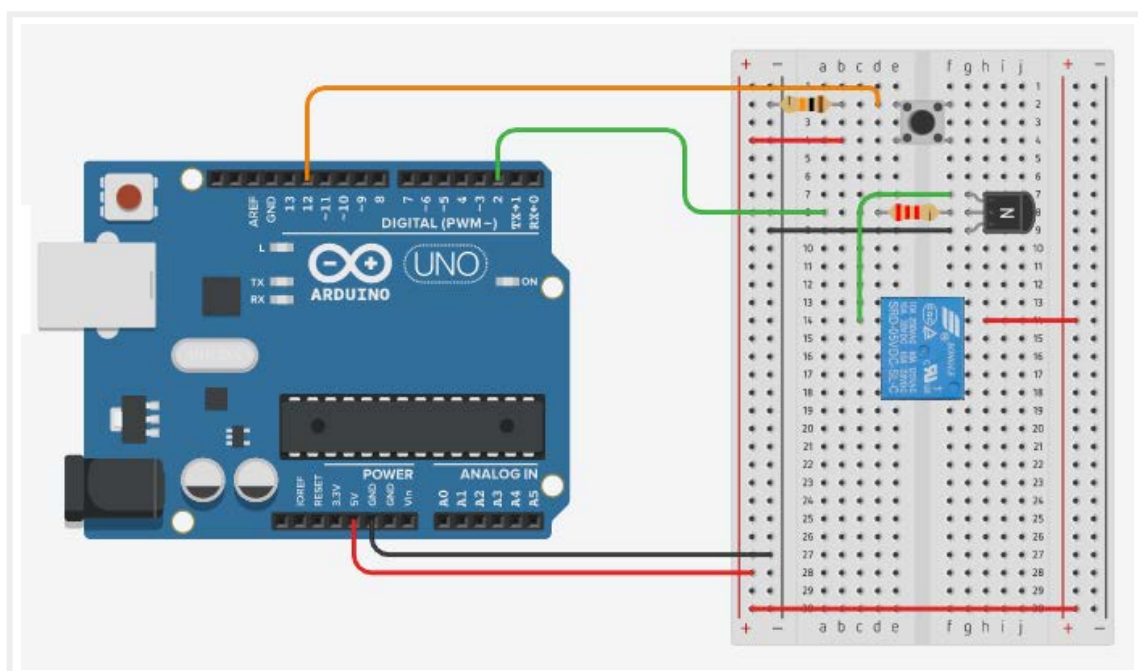
$$\text{Donc : } U_{Rb} = 5 - V_{be} = 5 - 0,7 = 4,3 \text{ V}$$

$$R_b = 4,3 / 0,2 \cdot 10^{-3} = 21\,500 \, \Omega = 21,5 \text{ k}\Omega$$

La résistance de la base doit donc être au maximum égale à 21,5 k Ω pour que le courant i_b soit au minimum de 0,2mA.

Dans notre circuit de contrôle du relais, la résistance R_b utilisée étant de 2,2 k Ω , le courant i_b est alors d'environ 2 mA.

Nous sommes donc assurés de saturer le transistor quand une sortie digitale de l'Arduino reliée à R_b est à un niveau haut (5V) et ceci sans danger pour la sortie de l'Arduino (rappel : $I_{\text{max sortie}} = 20 \text{ mA}$), selon le circuit suivant :



Avec le même programme que pour le contrôle du relais Grove, il est possible de contrôler le relais SRS-06VDC-SL :

– bouton appuyé :

- Sortie digitale 2 à HIGH
- Le transistor est saturé
- Un courant circule dans la bobine du relais
- Le relais commute en position NO

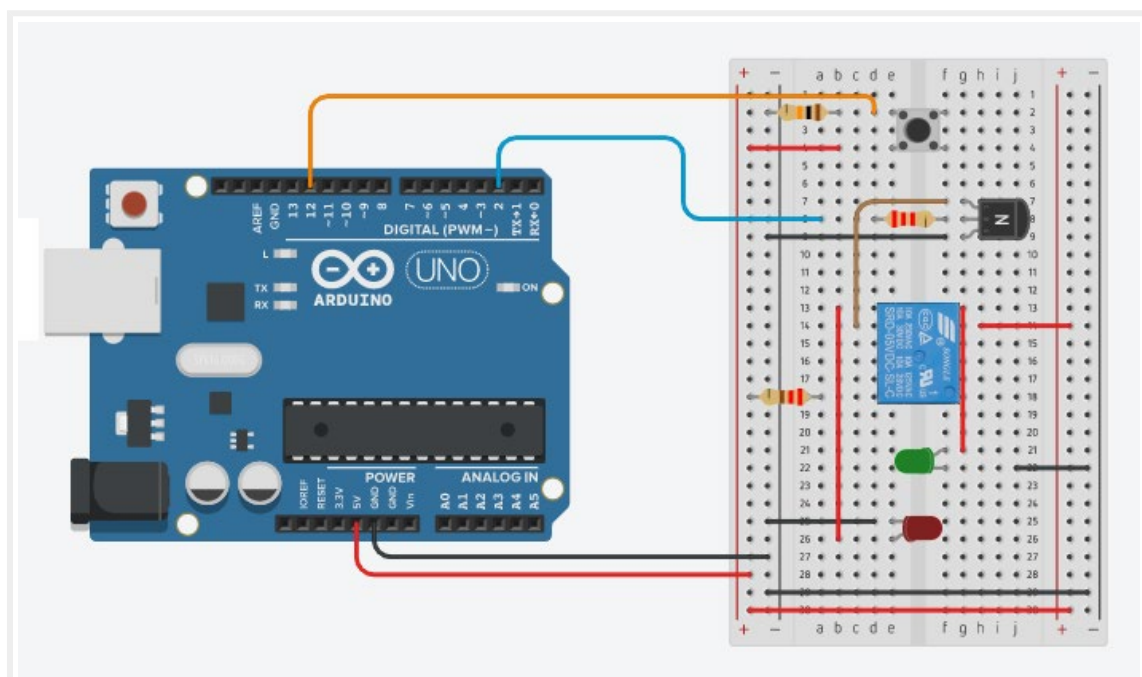
– bouton relâché :

- Sortie digitale 2 à LOW
- Le transistor est bloqué
- Aucun courant dans la bobine du relais
- Le relais commute en position NC

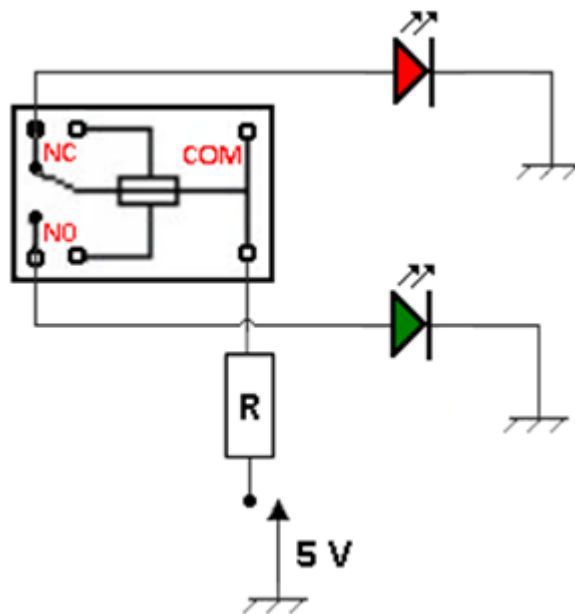
Exemple d'application :

Le circuit suivant permet d'allumer la DEL rouge ou la DEL verte avec un bouton poussoir et le programme de contrôle d'un relais :

- bouton relâché : La DEL rouge est allumée et la DEL verte est éteinte,
- bouton appuyé : La DEL verte est allumée et la DEL rouge est éteinte.

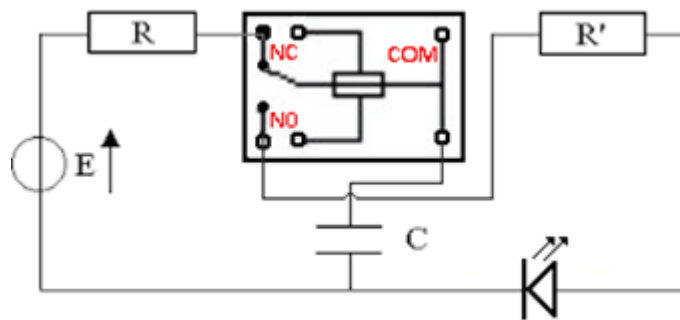


Le circuit de puissance est alors (relais vu de dessous) :



Avec $R = 220 \, \Omega$

En transposant cet exemple à notre circuit de simulation de flash, le circuit de puissance devient :



Ainsi en l'absence de courant dans la bobine de l'électroaimant, le condensateur se charge à travers la résistance R et quand un courant circule dans la bobine, le condensateur se décharge dans R' et la DEL.

. [Le programme](#)

Voici le code de l'activité :

Dipôles_RC_Flash

```
// Déclaration des constantes et variables

const int PinUC = 0;
const int PinButton = 12;
const int PinRelay = 2;

int ValPinUC = 0;
float UC = 0.0;
int StateHigh = 0;
int StateLow = 0;

int ValButton = 0;

// Initialisation des entrées et sorties

void setup() {
  Serial.begin(9600);
  pinMode(PinButton, INPUT);
  pinMode(PinRelay, OUTPUT);
  Serial.println("Charge du condensateur en cours.");
  digitalWrite(PinRelay, 0);
  while (ValPinUC<1022) {
    ValPinUC = analogRead(PinUC);
  }
  Serial.println("Condensateur charge.");
  Serial.print("Maintenez le bouton poussoir appuyé ");
  Serial.println("pour decharger le condensateur.");
}

// Fonction principale en boucle

void loop() {
  ValButton = digitalRead(PinButton);

  if (ValButton == HIGH)
  {
    if (StateHigh==0){
      Serial.println("Decharge du condensateur:");
      StateHigh = 1;
      StateLow = 0;
    }
    digitalWrite(PinRelay, 1);
    ValPinUC = analogRead(PinUC);
    UC = (ValPinUC/1023.0)*5.0;
    Serial.print("UC (V): ");
    Serial.println(UC,2);
    delay(200);
  }
}
```

```

else {
  if (StateLow==0){
    ValPinUC = analogRead(PinUC);
    if (ValPinUC<1022){
      Serial.println("Charge du condensateur:");
    }
    StateLow = 1;
    StateHigh = 0;
  }
  digitalWrite(PinRelay, 0);
  while (ValPinUC<1022) {
    ValPinUC = analogRead(PinUC);
    UC = (ValPinUC/1023.0)*5.0;
    Serial.print("UC (V): ");
    Serial.println(UC,2);
    delay(200);
  }
}
}

```

Déroulement du programme :

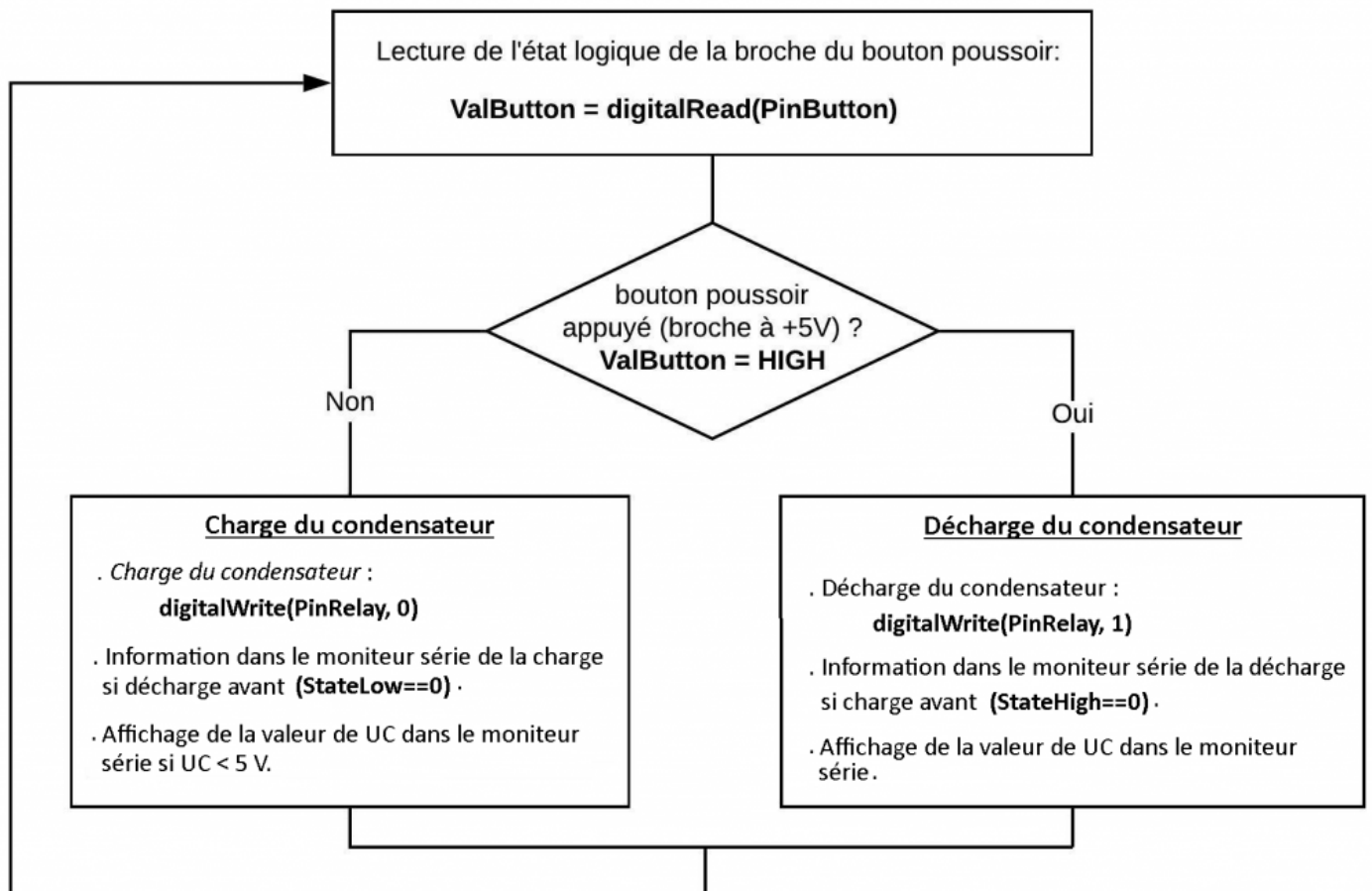
– 1. Déclaration des constantes et variables :

- . **const int PinUC = 0** (broche A0 du condensateur)
- . **const int buttonPin = 12** (broche du bouton poussoir)
- . **const int PinRelay = 2** (broche du relais)
- . **int ValPinUc = 0** (variable nombre entier pour stocker la valeur de la broche du condensateur)
- . **float Uc = 0.0** (variable nombre décimal pour stocker le résultat du calcul de la tension Uc)
- . **int StateHigh = 0** (variable nombre entier indiquant si la broche du relais est à l'état haut)
- . **int StateLow = 0** (variable nombre entier indiquant si la broche du relais est à l'état bas)
- . **int ValButton = 0** (variable nombre entier pour stocker la valeur de la broche du bouton poussoir)

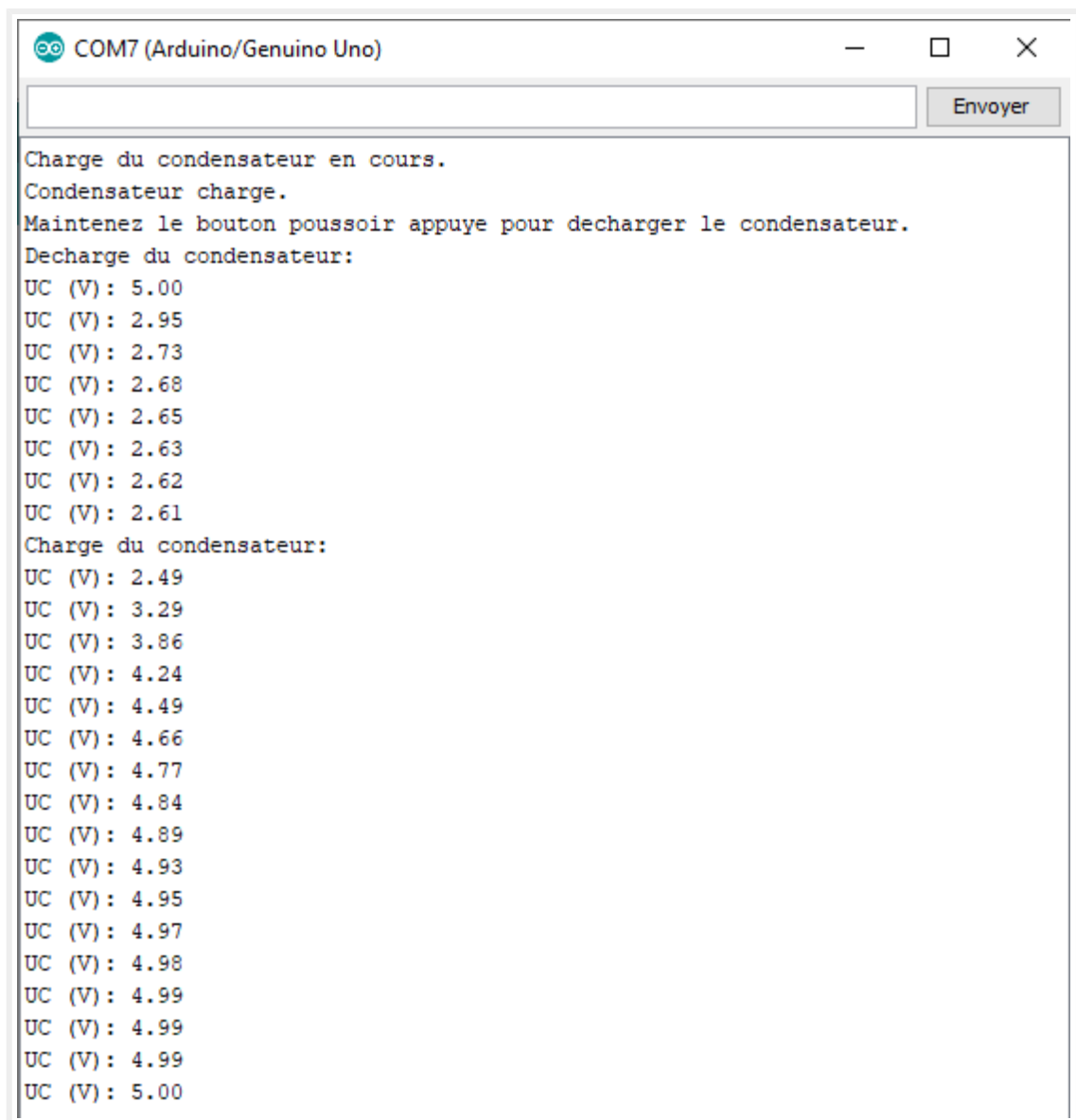
– 2. Initialisation des entrées et sorties :

- . **Initialisation de la broche du relais en sortie,**
- . **Initialisation de la broche du bouton poussoir en entrée,**
- . **Charge du condensateur.**

– 3. Fonction principale en boucle :



Résultats dans le moniteur série



The screenshot shows the 'COM7 (Arduino/Genuino Uno)' serial monitor window. It contains the following text:

```
Charge du condensateur en cours.  
Condensateur charge.  
Maintenez le bouton poussoir appuye pour decharger le condensateur.  
Decharge du condensateur:  
UC (V) : 5.00  
UC (V) : 2.95  
UC (V) : 2.73  
UC (V) : 2.68  
UC (V) : 2.65  
UC (V) : 2.63  
UC (V) : 2.62  
UC (V) : 2.61  
Charge du condensateur:  
UC (V) : 2.49  
UC (V) : 3.29  
UC (V) : 3.86  
UC (V) : 4.24  
UC (V) : 4.49  
UC (V) : 4.66  
UC (V) : 4.77  
UC (V) : 4.84  
UC (V) : 4.89  
UC (V) : 4.93  
UC (V) : 4.95  
UC (V) : 4.97  
UC (V) : 4.98  
UC (V) : 4.99  
UC (V) : 4.99  
UC (V) : 4.99  
UC (V) : 5.00
```

Remarque :

On peut voir, dans le moniteur série, que le condensateur se décharge jusqu'à atteindre la tension de seuil de la DEL blanche (environ 2,5 V).

En effet, quand la tension aux bornes de la DEL est inférieure à la tension de seuil, la diode n'est plus passante, l'intensité dans le circuit est nulle et le condensateur ne se décharge plus.