

Télécommande Infrarouge et Arduino

Bonjour à tous.

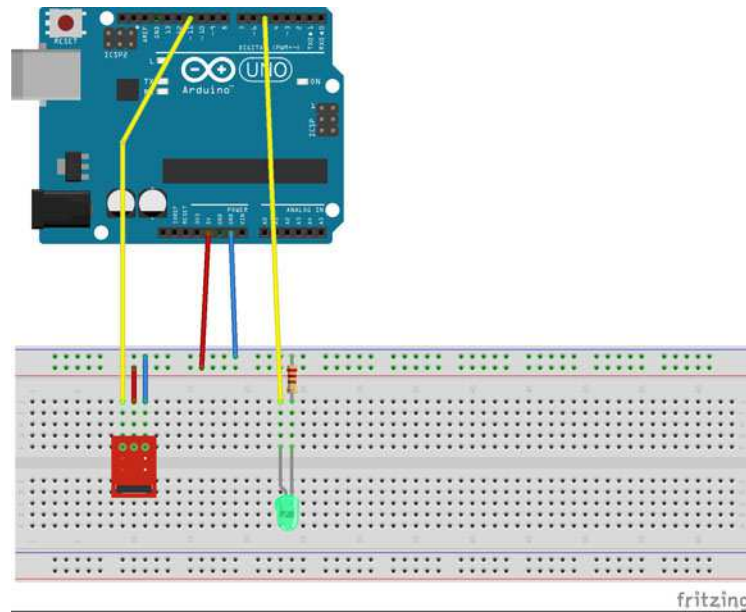
Aujourd'hui je vous propose, toujours avec la télécommande infrarouge, de commander :

- 1- L'allumage et l'extinction du led progressivement,
- 2- Commander un petit moteur électrique à vitesse variable.

Première partie :

L'Arduino possède des sorties qui sont utilisables en analogique (mais aussi en numérique tout ou rien). Comme cela nous allons pouvoir faire « varier la tension » aux bornes d'une led et ainsi faire varier sa luminosité.

Câblage à réaliser :



Le seul changement par rapport au montage précédent c'est qu'il faut IMPERATIVEMENT utiliser une broche de l'Arduino qui possède une sortie analogique (on dit PWM).

Aujourd'hui l'objectif est de commander une led avec la télécommande :

- La touche 0 pour éteindre la led,
- La touche 1 pour allumer la led progressivement (action momentanée sur la touche),
- La touche 2 pour éteindre la led progressivement (action momentanée sur la touche).

Avec la manip de l'étape précédente vous avez pu récupérer au moins ces trois codes. Pour ma télécommande les codes relevés sont :

- Touche 0 : 16738455
- Touche 1 : 16724175
- Touche 2 : 16718055

La led sera raccordée à une des bornes PWM : broche 5

Programme possible :

```
// Commander une led à allumage et extinction progressive avec une télécommande IR
#include <IRremote.h>
int IR = 11; // Récepteur IR en broche 11
int ledsana = 5; // led PWM N° 5
int i = 0; // variable pour effectuer la variation
IRrecv RCEP(IR);
decode_results RIR; //
void setup()
{
  RCEP.enableIRIn(); // Commencement de la réception
  pinMode(ledsana, OUTPUT); // la led est déclarée comme reliée à une sortie
}

void loop()
{
  if (RCEP.decode(&RIR)) // s'il y a une réception sur le capteur
  {
    if (RIR.value == 16738455) // si touche 0 actionnée
    {
      digitalWrite(ledsana, 0); // La LED s'éteint "brutalement"
    }
    if(RIR.value == 16724175) // si touche 1 actionnée
    {
      for(i=0; i<=255; i++)
      {
        analogWrite(ledsana, i); // La LED s'allume progressivement de 0 à max en 2.55 s (255 x 10 ms)
        delay (10);
      }
    }
    if(RIR.value == 16718055) // si touche 2 actionnée
    {
      for(i=0; i<=255; i++)
      {
        analogWrite(ledsana, 255-i); // La LED s'éteint de la valeur max à 0 en 2.55 s (255x10ms)
        delay (10);
      }
    }
    RCEP.resume(); // Réception de la prochaine valeur
  }
}
```

Le fichier `essai_cde_IR3_1_pour_ES.ino` est téléchargeable.

Deuxième partie :

Vous avez peut être envie de commander un moulin à eau, une éolienne (certes le mouvement sera produit par un petit moteur à courant continu).

J'ai pensé qu'il serait bien de faire varier la vitesse de ce moteur au moment du démarrage et aussi au moment de l'arrêt. C'est un peu le même principe que pour la led.

Mais car il y a un mais ... L'intensité que délivre un Arduino n'est pas suffisante pour alimenter un moteur.

Il est nécessaire de mettre en place un amplificateur.

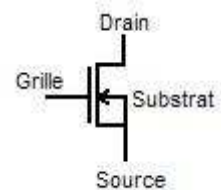
Deux solutions :

- Utiliser un transistor MOSFET
 - Utiliser un circuit intégré adapté à cette réalisation : L293D
- Les explications seront dans le prochain article.

Utilisation d'un transistor MOSFET :

Démystifions ce composant en indiquant tout simplement que ce composant va permettre de commander un courant fort (plusieurs ampères) à partir d'une tension de sortie de l'Arduino.

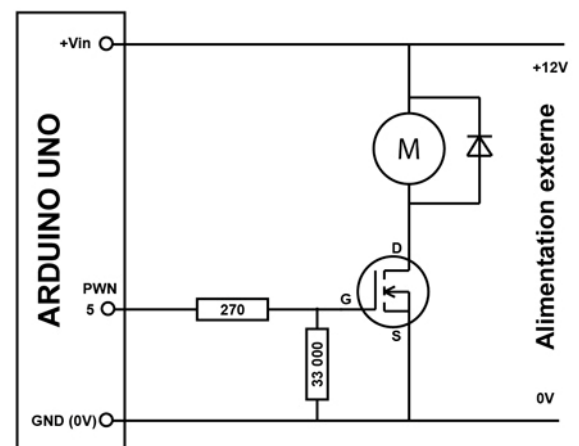
Son symbole simplifié est le suivant :



Pour l'utiliser il faut appliquer une commande sur l'entrée « grille » et la partie « puissance » sera raccordée à « Drain » et « Source ». Ce qui donne le schéma suivant :

Remarque : Ne pas oublier d'interconnecter les masses (Arduino et alimentation externe).

Petite précaution : Ce type de transistor (comme d'autre circuit électronique) est sensible à l'électricité statique. Aussi je vous conseille de vous relier à un conducteur de masse lui même relié à « la terre » (c'est une précaution que je prends dès que je travaille, par exemple, sur un décodeur de locomotive).



Aujourd'hui l'objectif est de commander un petit moteur courant continu 12V avec la télécommande :

- La touche 0 pour arrêter le moteur avec un ralentissement (action momentanée sur la touche) en 2.5 secondes,
- La touche 1 démarré le moteur progressivement (action momentanée sur la touche) en 5 secondes.

Avec la manip de l'étape précédente vous avez pu récupérer au moins ces 2 codes :

Pour ma télécommande les codes relevés sont :

- Touche 0 : 16738455
- Touche 1 : 16724175

La sortie de l'Arduino sera une des bornes PWM : broche 5

