

# Électronique :

## CR\_Arduino

## Table des matières

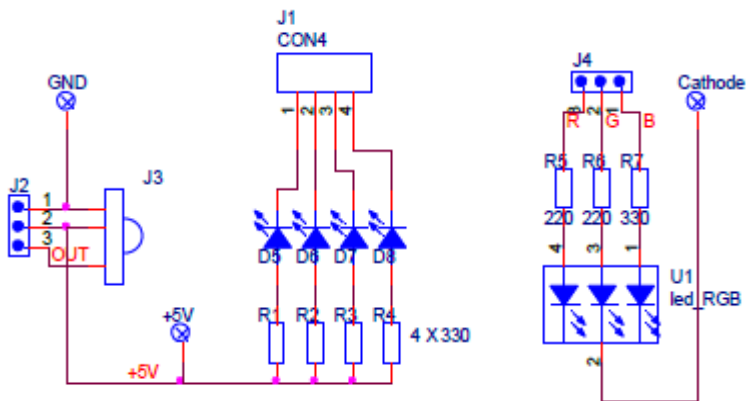
CR_Arduino.....	1
TP1.....	3
1 : Aspect matériel général.....	3
2 : Pilotage de 3 LEDS via télécommande Infra-Rouge.....	5
3 : Mise en œuvre d'une led RBG.....	9
4 : Mise en œuvre d'une led RBG par PWM.....	9
Commander la LED 1, 2, 3 avec les touches 1, 2, 3.....	13

# TP1

## 1 : Aspect matériel général

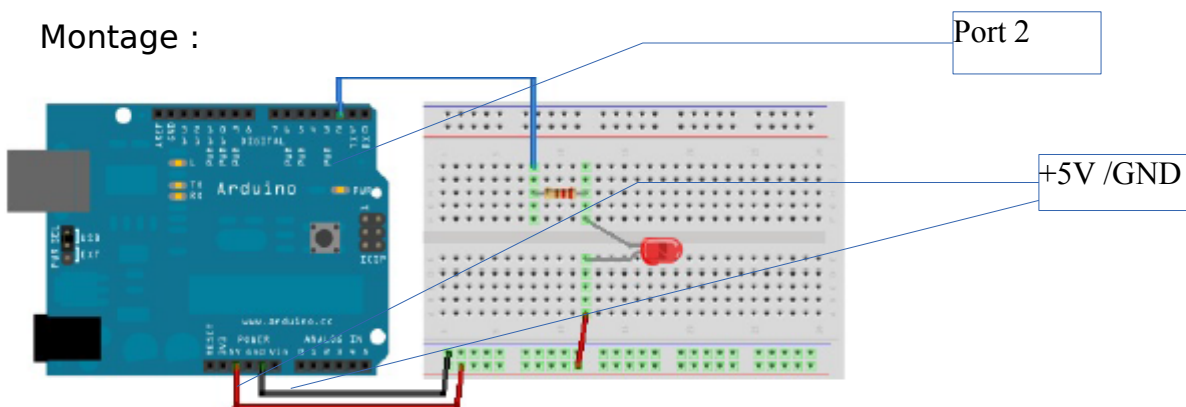
Faire clignoter une LED.

Schéma structurel :



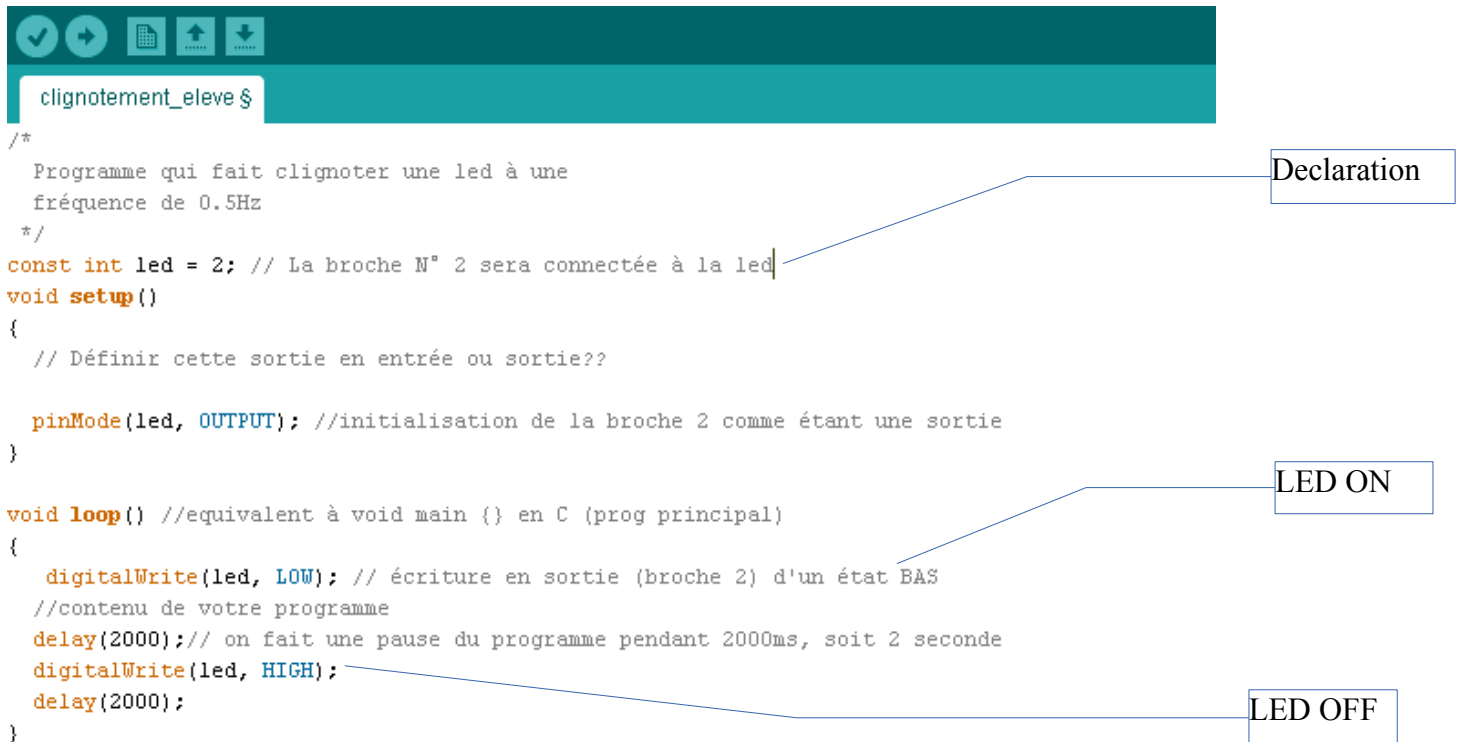
On remarque que pour qu'une LED soit allumée, il faut que CON4 soit à **0V** pour qu'il y ait une différence de potentiel.

Montage :



## Partie logicielle :

*Comme indiqué précédemment l'état de la LED doit être BAS pour qu'elle s'allume.*



```
/*
  Programme qui fait clignoter une led à une
  fréquence de 0.5Hz
*/
const int led = 2; // La broche N° 2 sera connectée à la led
void setup()
{
  // Définir cette sortie en entrée ou sortie??

  pinMode(led, OUTPUT); //initialisation de la broche 2 comme étant une sortie
}

void loop() //equivalent à void main {} en C (prog principal)
{
  digitalWrite(led, LOW); // écriture en sortie (broche 2) d'un état BAS
  //contenu de votre programme
  delay(2000); // on fait une pause du programme pendant 2000ms, soit 2 seconde
  digitalWrite(led, HIGH);
  delay(2000);
}
```

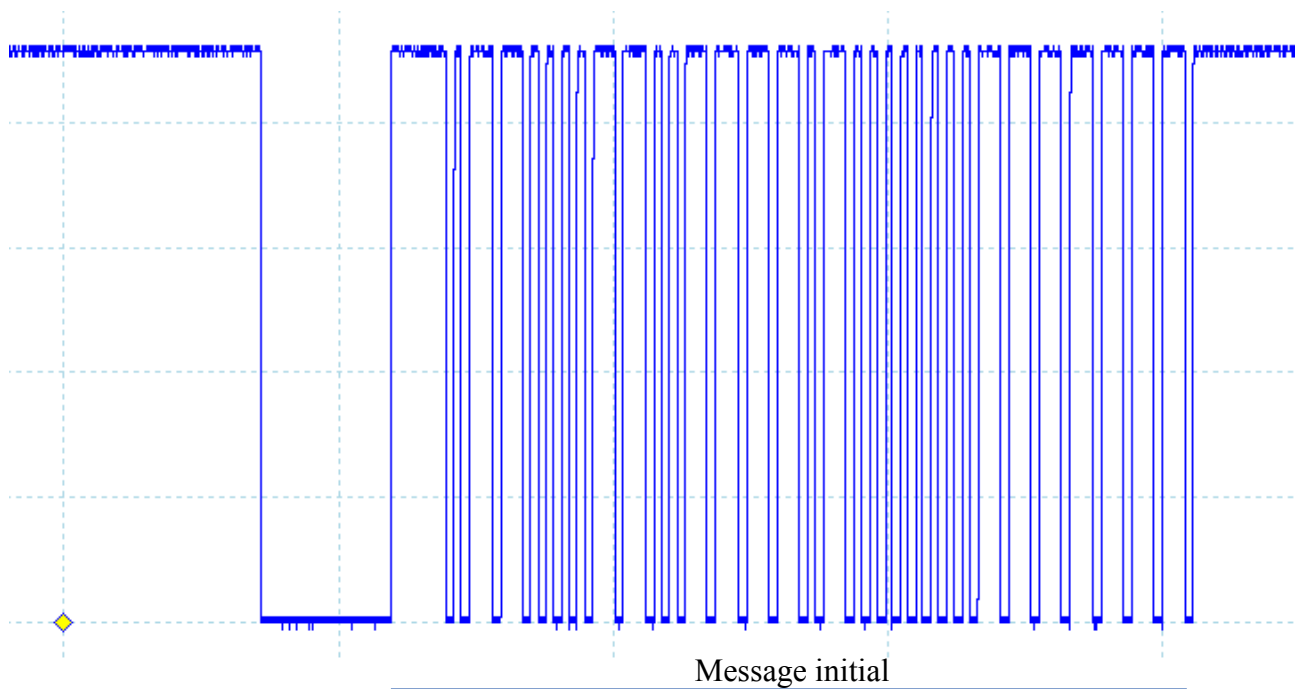
Declaration

LED ON

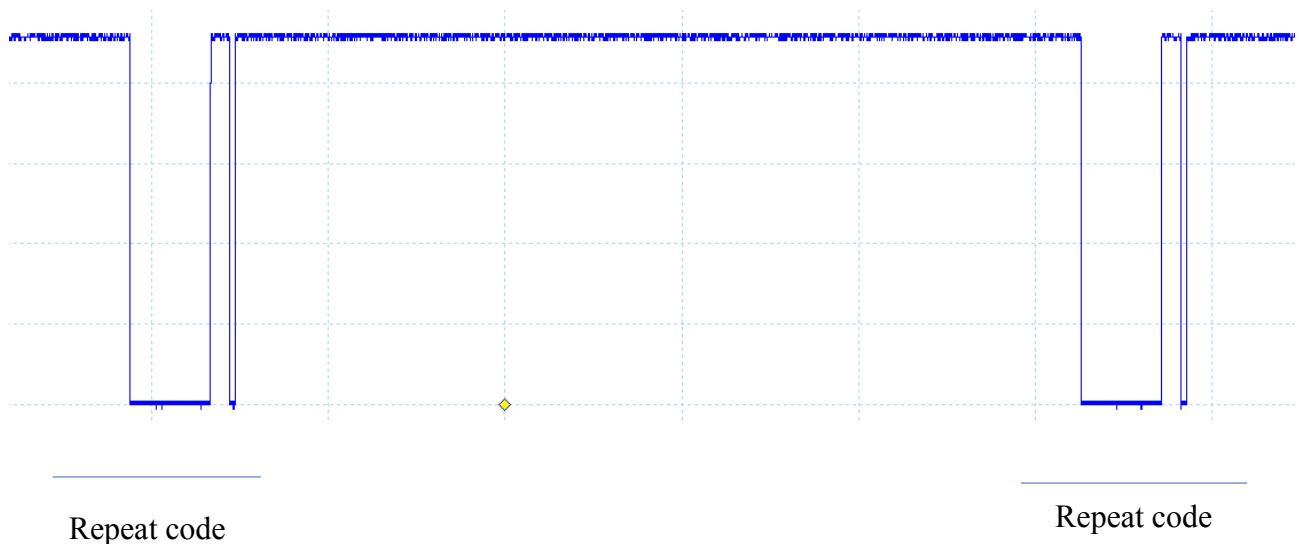
LED OFF

## 2 : Pilotage de 3 LEDS via télécommande Infra-Rouge

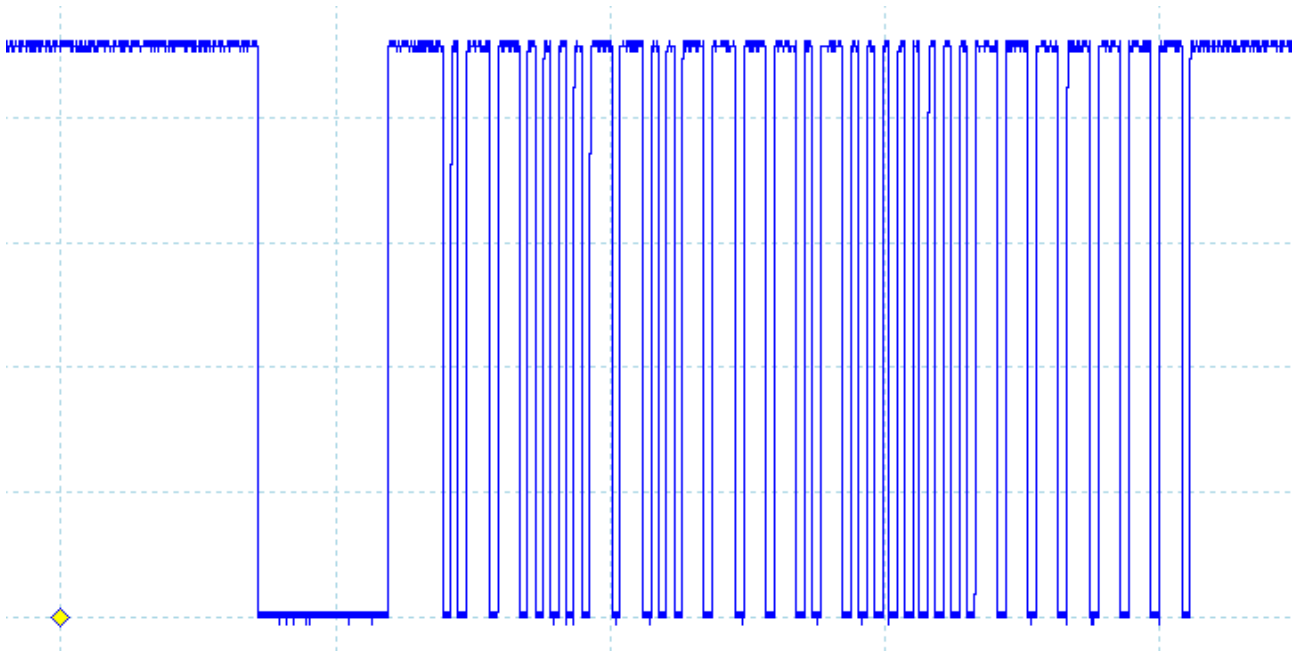
Relevé de la trame de la **touche « 1 »** de la télécommande. (signal de la sortie TSOP)



Puis (quand on maintient la touche)



Reprenons le code initial.



Base 10 :

01100001 10011110 10000000 01111111

de gauche a droite en hexadécimal : **619E8607F**

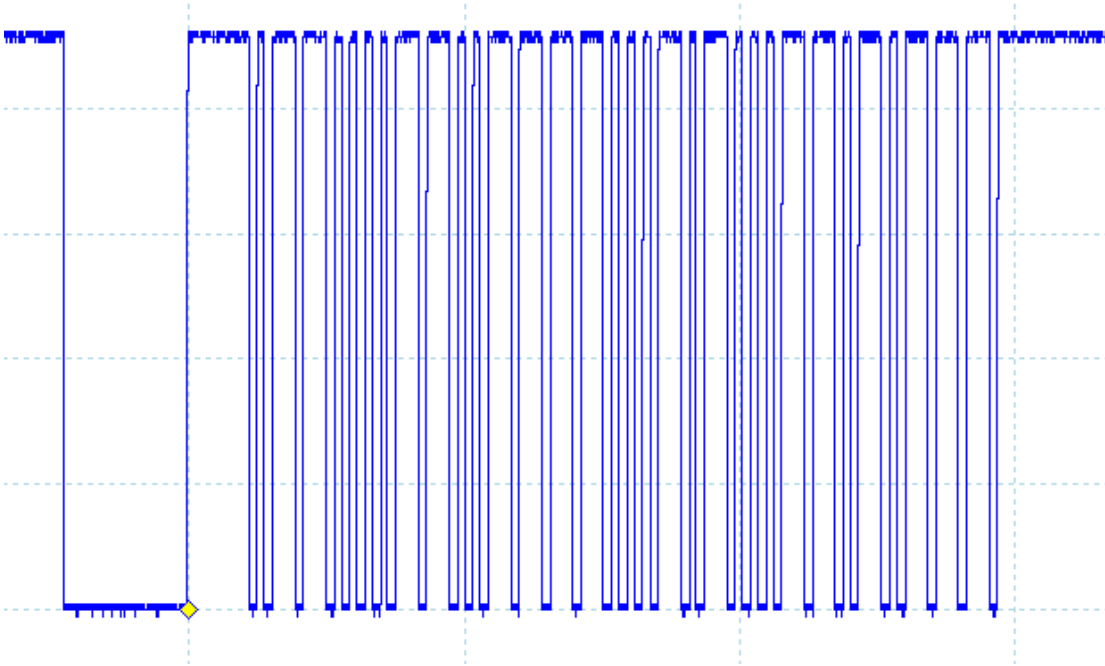
en décimal : **1637777535**

On voit bien qu'il y a 4 octets, 2 deux octets et 2 autres leurs contraires.

Sur le programme, on peut lire cela : #define touche **11637777535**

(Soit le motif code de la touche « 1 » trouvé précédemment)

**Touche « 2 » :**

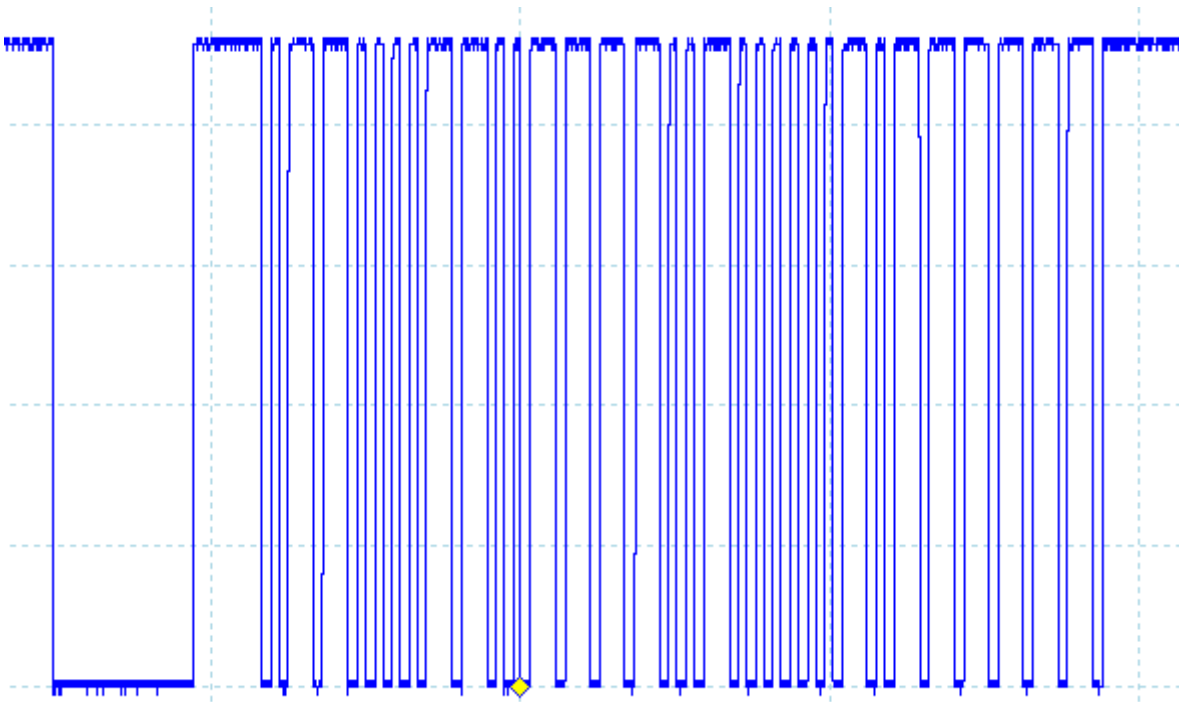


Base 10 : **01100001 10011110 00101000 11010111**

Hexadécimal : **619E28D7**

Décimal : **1637755095** (*à assigner a la LED 2*)

**Touche « 3 » :**



Base 10 : **01100001 10011110 01000000 10111111**

Hexadécimal : **619E40BF**

Décimal : **1637761215** (à assigner a la LED 3)

Nous avons maintenant les deux « motifs » décimaux assignés aux deux touches. Cette étape est laborieuse et longue si on aurait du décoder les motifs de toutes les touches.. Il existe donc une autre manière de les avoirs via le logiciel d'arduino :

Ajoutons cette ligne pour pouvoir lire la donnée reçue par le récepteur infrarouge :

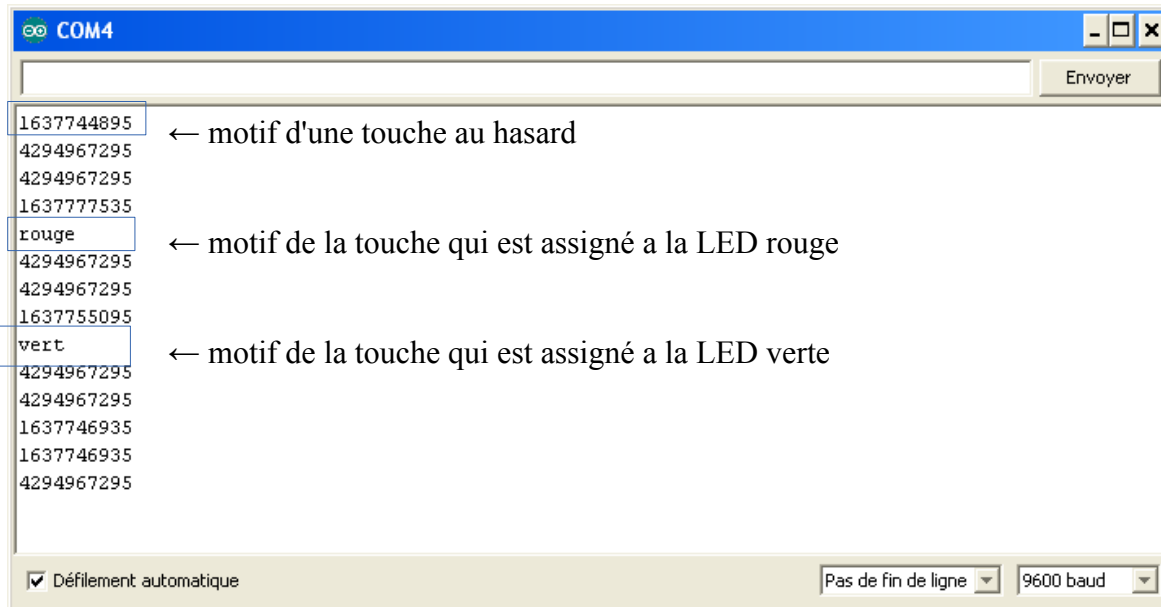
**`Serial.println(results.value,DEC);`**

**`switch (results.value)`**

Programme permettant de commander la LED 1, 2, 3 avec les touches 1, 2, 3 de la télécommande [ICI](#)



Ensuite, aller dans outils → moniteur série →



Nb: Les motifs qui commencent par [42949.....] sont des motifs qui apparaissent lorsque l'on reste appuyé sur la touche. (Se sont les codes de répétition)

### 3 : Mise en œuvre d'une led RGB

Pour la led RGB, il faut relier la cathode commune (Kc) au GND du arduino et en commandant par des niveaux hauts ou des PWM sur J4 .

(changer la ligne :

```
etatrouge=0; // Initialiser la variable pour éteindre la led
```

```
etatvert=0;
```

```
etatjaune=0;)
```

### 4 : Mise en œuvre d'une led RGB par PWM

À partir de ces trois couleurs (rouge/vert/bleu), il est possible de créer n'importe quelle autre couleur du spectre lumineux visible en mélangeant ces trois couleurs primaires entre elles.

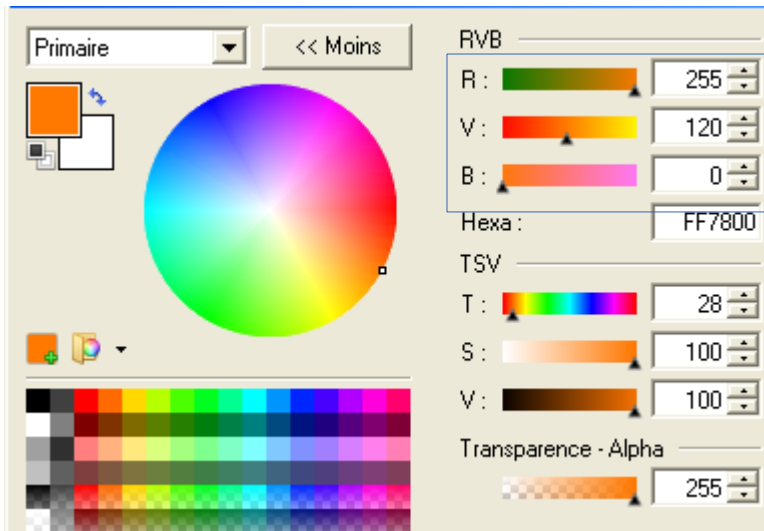
Par exemple, pour faire de l'orange on va mélanger du rouge (2/3 du volume final) et du vert (à 1/3 du volume final).

Exemple : Je veut une couleur orange :

Rouge: 255 (allumée à 100%)

Vert:120 (allumée à 47%)

Bleu:0 (allumée à 0%, soit éteinte)



```
#define led_rouge 5// préciser sur quelle pin la connecter
#define led_verte 3
#define led_bleu 6
```

```
void setup()
{
  pinMode(led_rouge, OUTPUT); // Positionner la broche en sortie pour la led rouge
  pinMode(led_verte, OUTPUT);
  pinMode (led_bleu, OUTPUT);
```

```
  analogWrite(led_rouge, 255);
  analogWrite(led_verte, 120);
  analogWrite(led_bleu, 0);
}
```

Attention, il faut connecter les PIN sur des broches « Pulse width modulation » notées comme cela sur l'arduino « ~ »

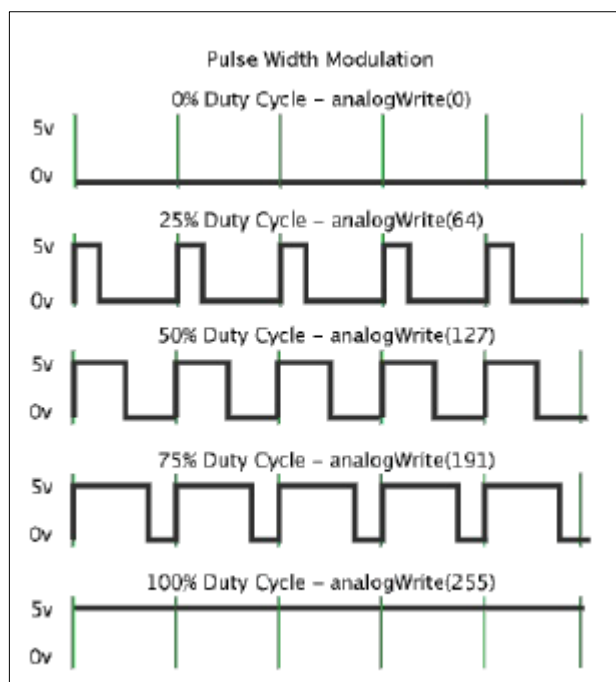
```
void loop()
{
  |
}
```

Pour qu'une LED ne s'allume pas totalement il faut changer la PWM, qui est un signal de fréquence fixe

Elle a un rapport cyclique qui varie avec le temps suivant "les ordres qu'elle reçoit".

Le rapport cyclique est mesuré en pour cent (%). Plus le pourcentage est élevé, plus le niveau logique 1 est présent dans la période et moins le niveau logique 0 l'est. Et inversement. Le rapport cyclique du signal est donc le pourcentage de temps de la période durant lequel le signal est au niveau logique 1.

Exemple pour 0,  $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{3}{4}$  et 1 :

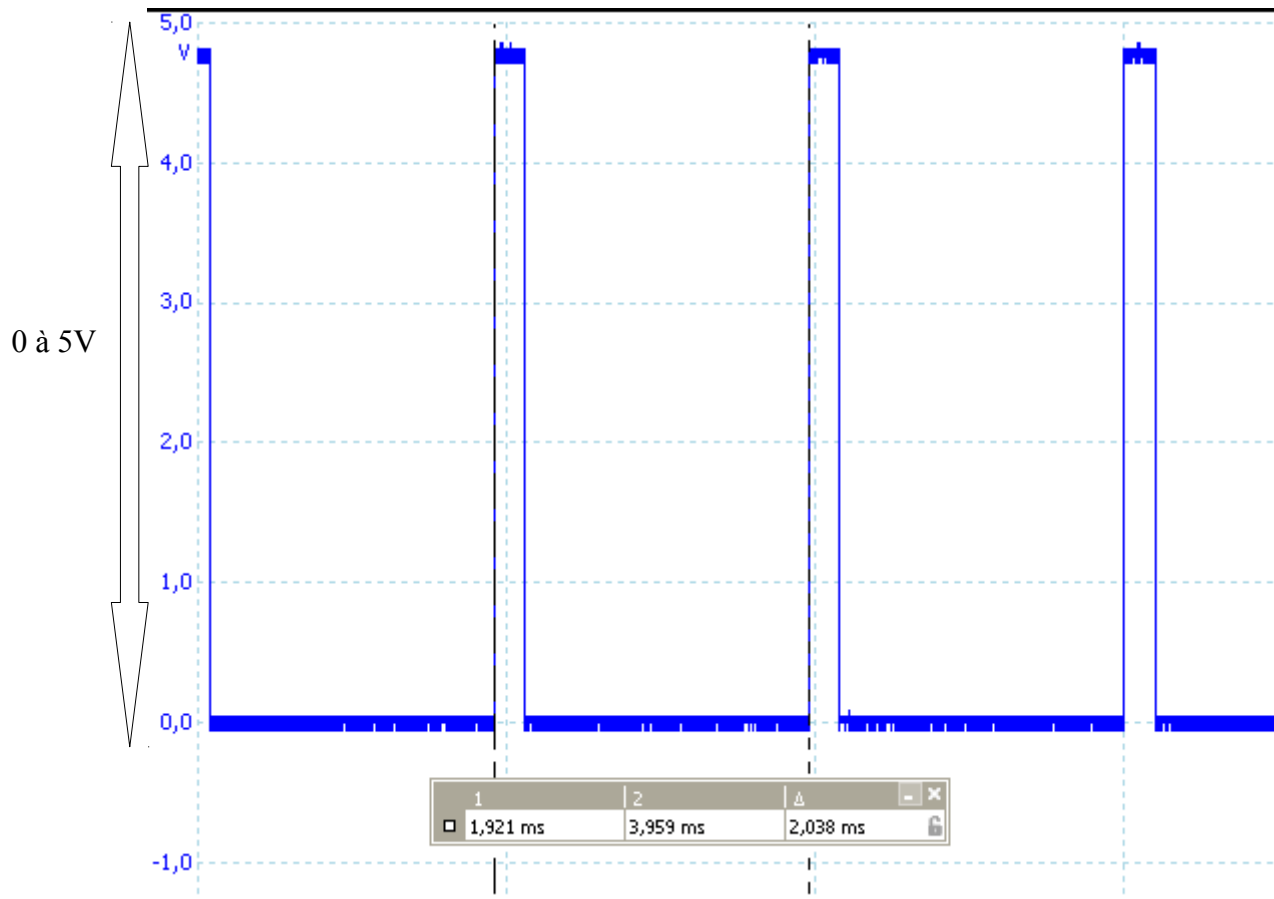


Nous allons essayer de voir cette modulation pour une led qui est allumée à **10%**. (environs 25 en donnée numérique.) `analogWrite(led_verte, 25);`

(10 % trouvé car :  $(100/255)*25 \approx 10$ )

**Résultat sur picoscope page suivante → → →**

Voici la mesure, elle correspond avec l'exemple précédent.



$T = 2,038 \text{ ms}$

Nous avons du 5V pendant  $194 \mu\text{s}$

Le rapport cyclique est donc de  $2,038 \text{ ms} / 194 \mu\text{s} = \mathbf{10,5 \%}$

Cela concorde avec ce que nous avons prévu.

**Commander la LED 1, 2, 3 avec les touches 1, 2, 3**

```
#include <IRremote.h>
// definition des touche télécommandes Oxygen
//les codes ont été relevés par hyperterm

#define touche1 1637777535
#define touche2 1637755095 //01100001 10011110 00101000 11010111
#define touche3 1637761215 //01100001 10011110 01000000 10111111

#define led_rouge 2// préciser sur quelle pin la connecter
#define led_verte 3
#define led_jaune 4

int RECV_PIN = 11;
int etatrouge ;
int etatvert ;
int etatjaune ;
IRrecv irrecv(RECV_PIN);

decode_results results;

void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn(); // start le récepteur
  pinMode(led_rouge, OUTPUT); // Positionner la broche en sortie pour la led
rouge
  pinMode(led_verte, OUTPUT);
  pinMode (led_jaune, OUTPUT);
  etatrouge=1; // Initialiser la variable pour éteindre la led
  etatvert=1;
  etatjaune=1;

  digitalWrite(led_rouge, etatrouge);
  digitalWrite(led_verte, etatvert);
  digitalWrite(led_jaune, etatjaune);

}

void loop() {
  if (irrecv.decode(&results))
  {
    Serial.println(results.value,BIN);
    switch (results.value)
    {
      case touche1:
      {
        Serial.println("rouge");
        etatrouge= !etatrouge; // Quelle est le role de cette ligne?
        digitalWrite(led_rouge, etatrouge);
      }
      break;
    }
  }
}
```

```
case touche2:
{
    Serial.println("vert");
    etatvert= !etatvert; // Quelle est le role de cette ligne?
    digitalWrite(led_verte, etatvert);
}
break;
case touche3:
{
    Serial.println("jaune");
    etatjaune= !etatjaune; // Quelle est le role de cette ligne?
    digitalWrite(led_jaune, etatjaune);
}
break;

}
irrcv.resume(); // Receive the next value
}
}
```