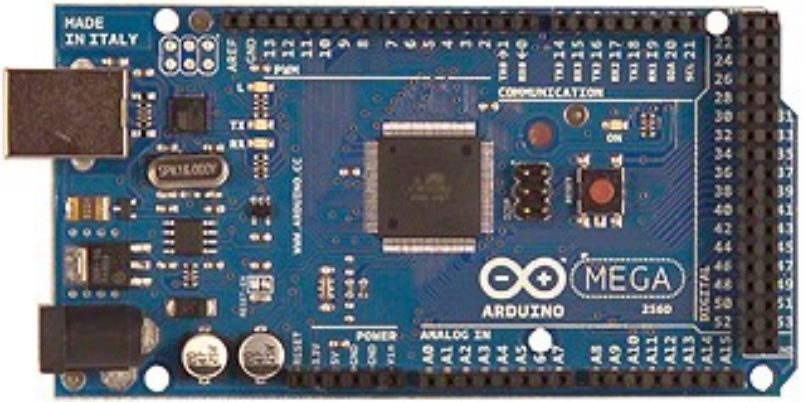
InSyTe
Sciences & Ingénierie des Systèmes Techniques

Présentation des cartes Arduino



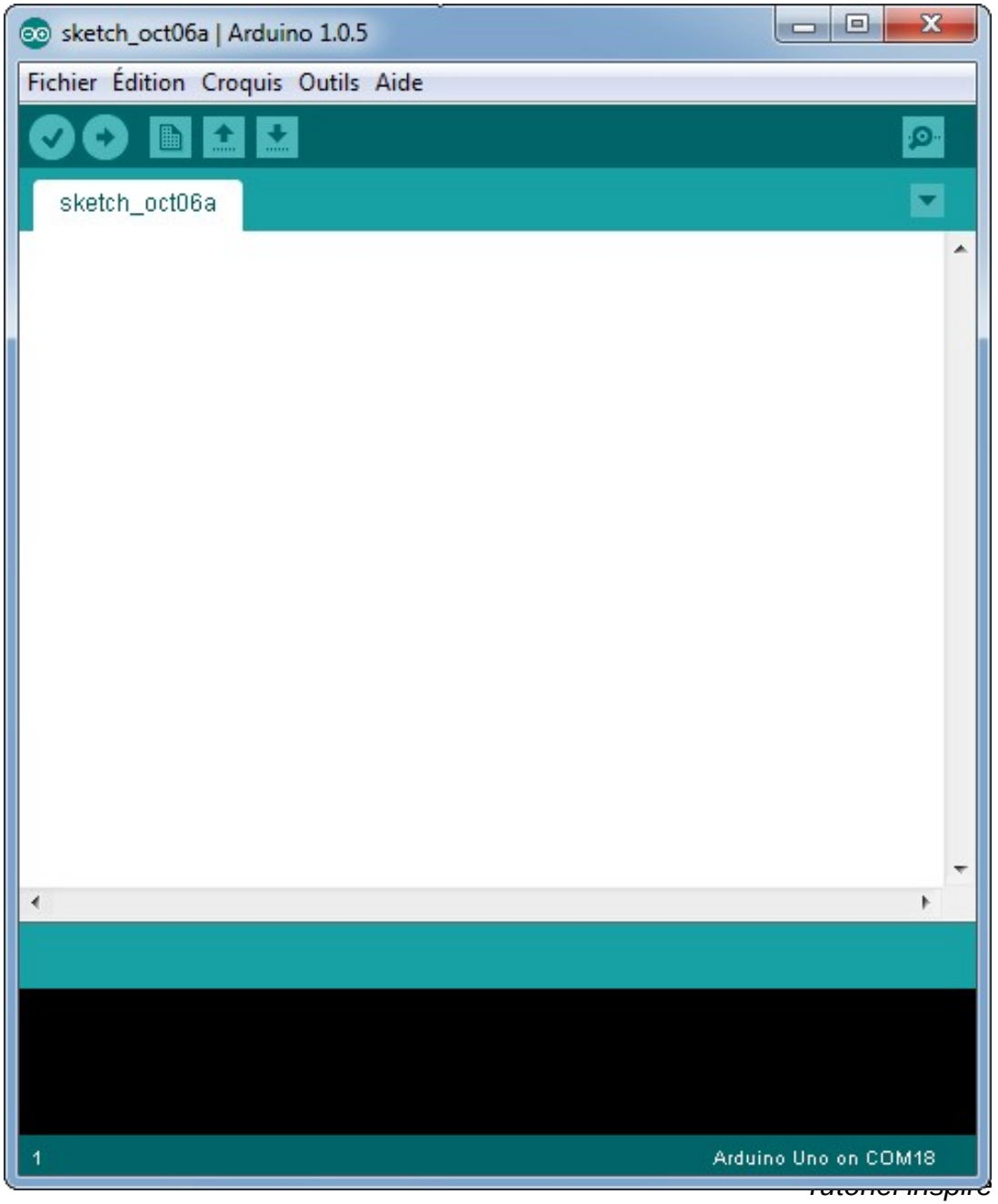
Tutoriel inspiré de l'excellent site http://eskimon.fr

Une Arduino Uno ou Duemilanove	Un câble USB A mâle/B mâle
	

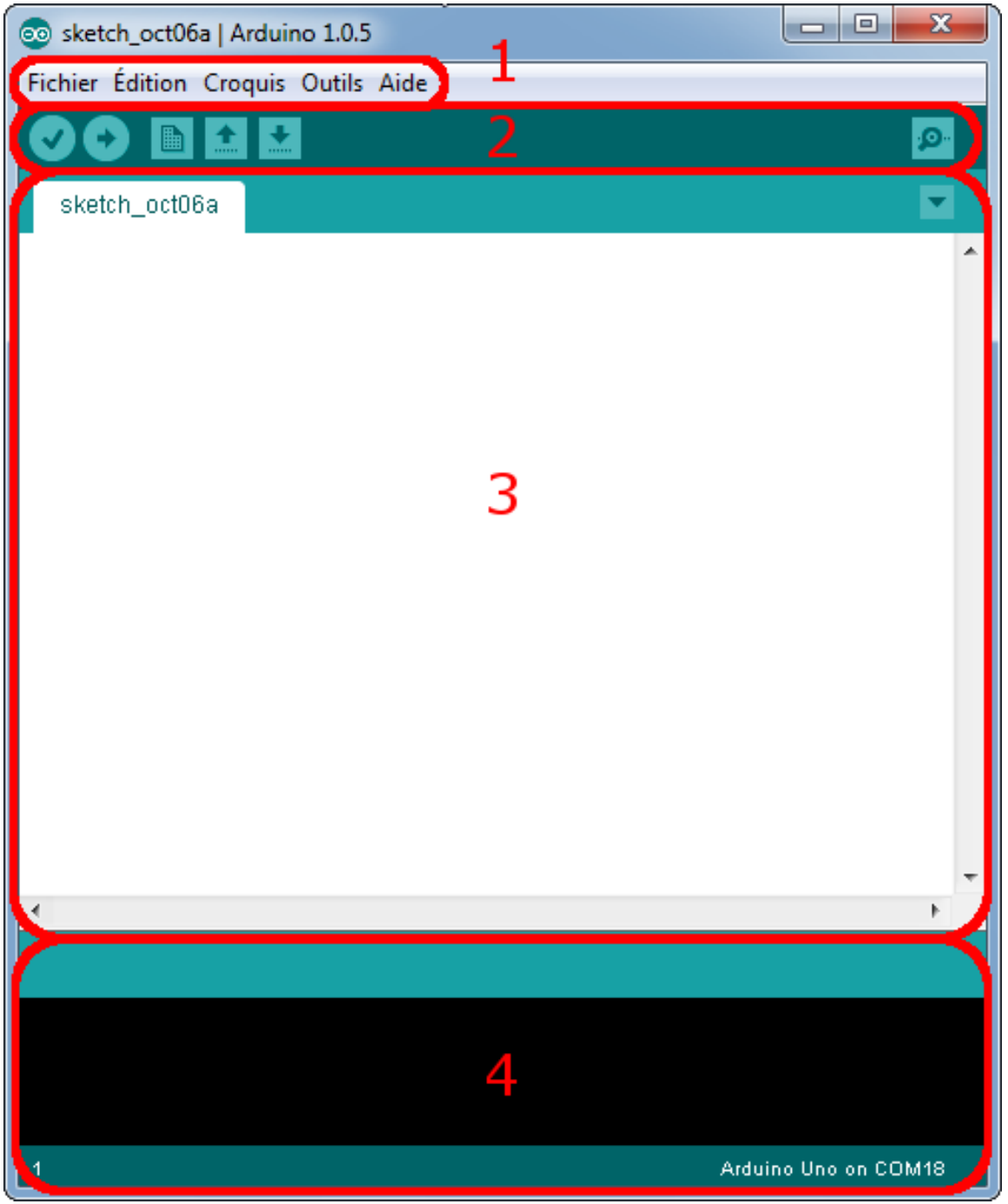
Une BreadBoard (plaque d'essai)	Un lot de fils pour brancher le tout !
	

Tutoriel inspiré de l'excellent site http://eskimon.fr

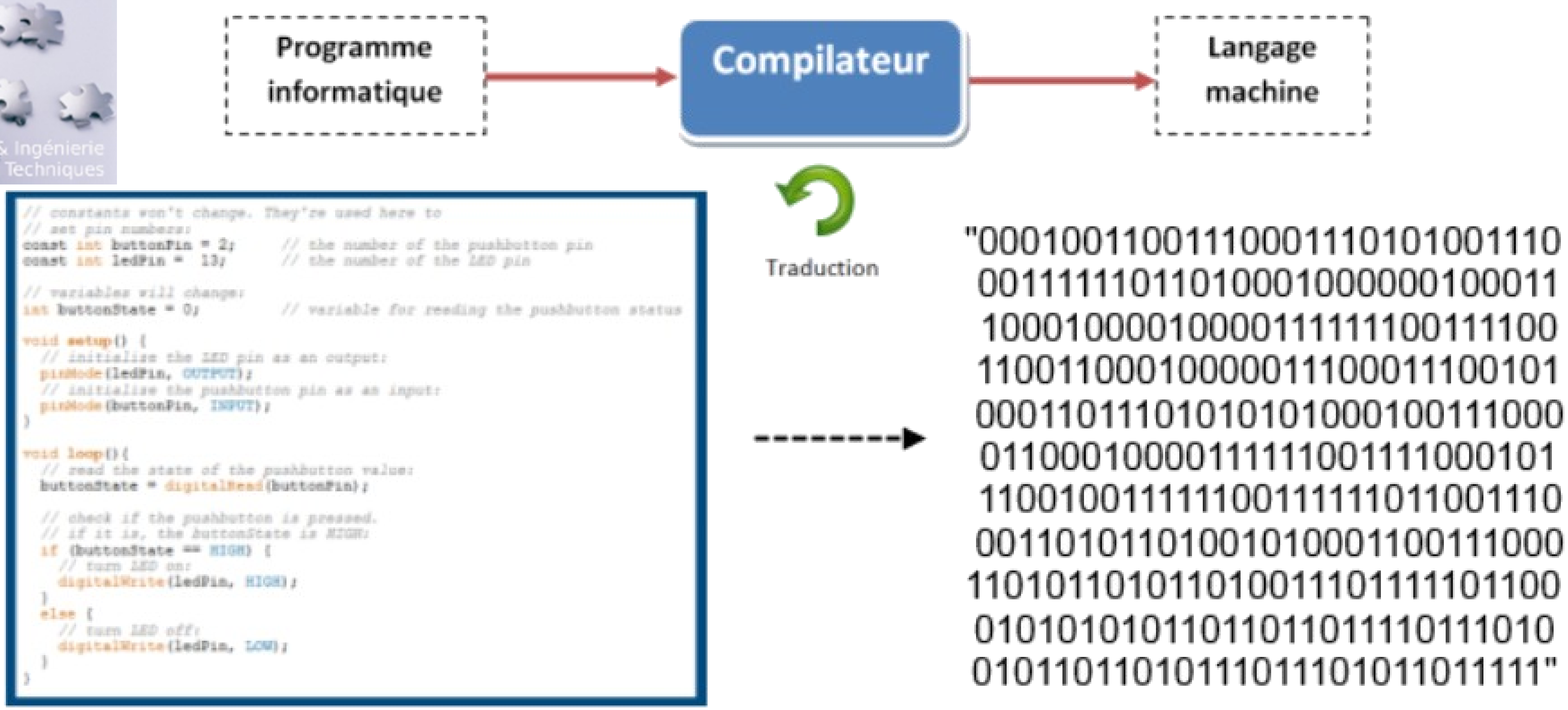
L'IDE pour Arduino

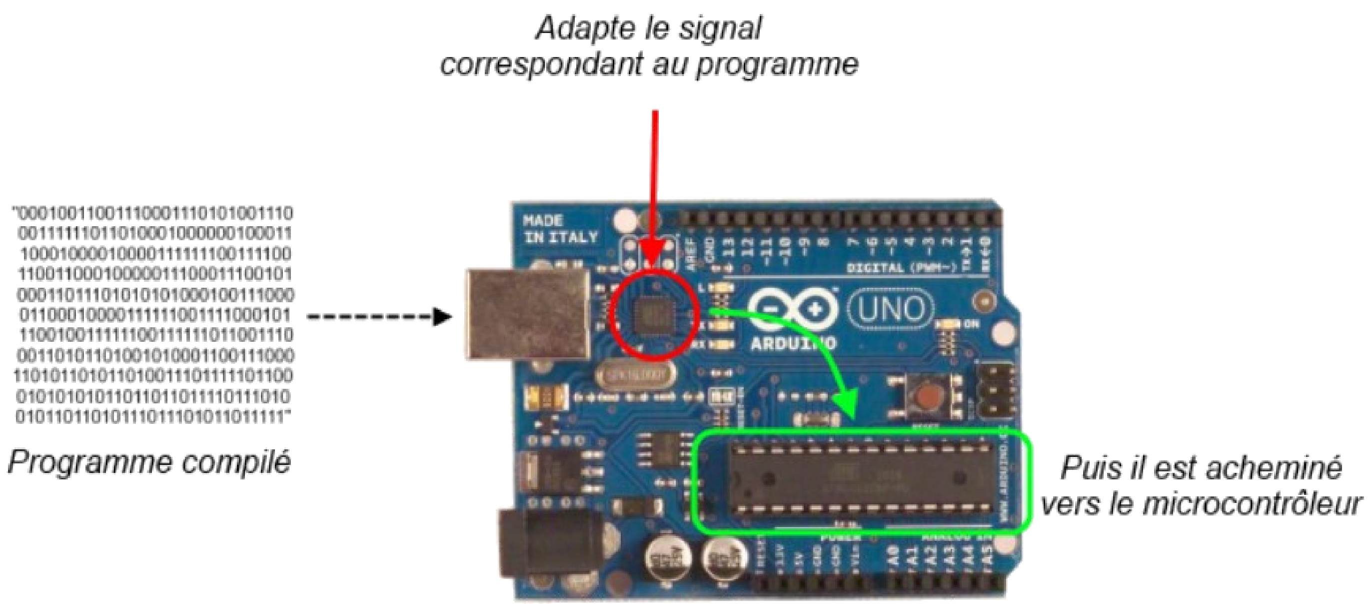


Tutoriel inspiré de l'excellent site http://eskimon.fr



Tutoriel inspiré de l'excellent site http://eskimon.fr





Tutoriel inspiré de l'excellent site http://eskimon.fr

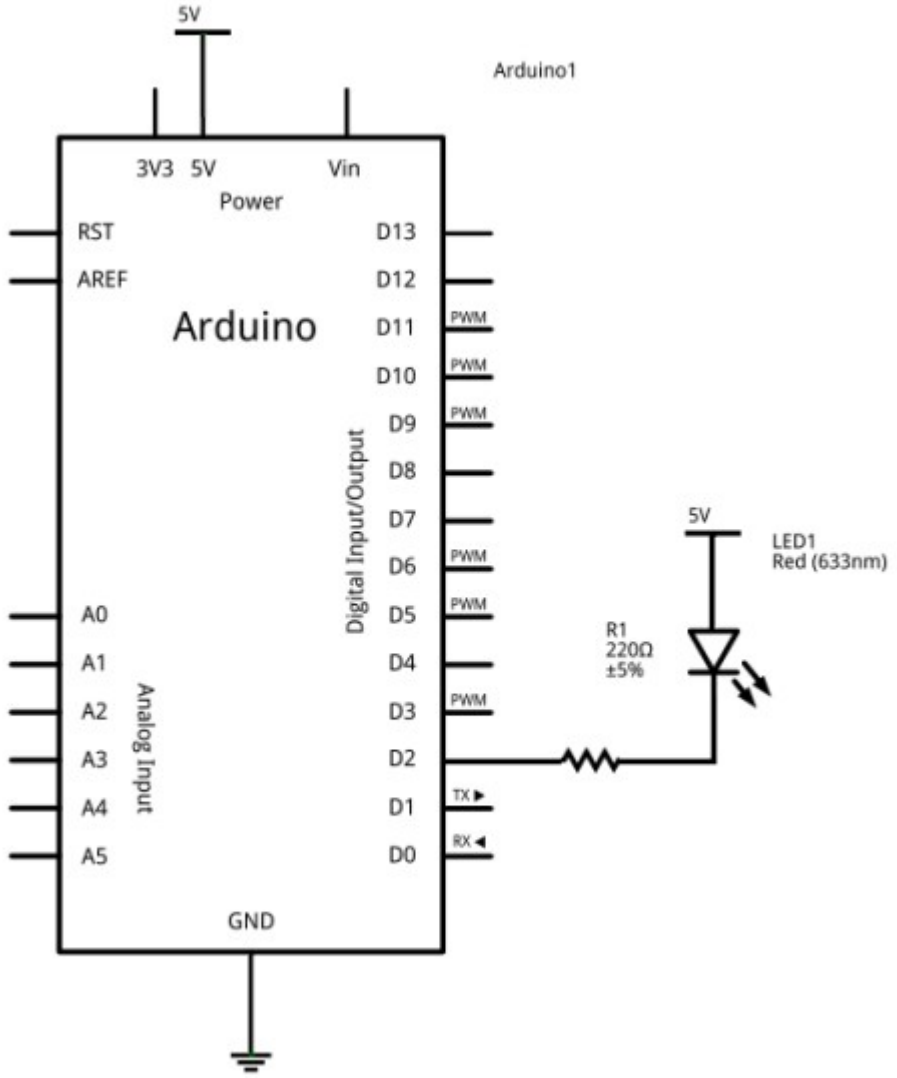
Le code minimal

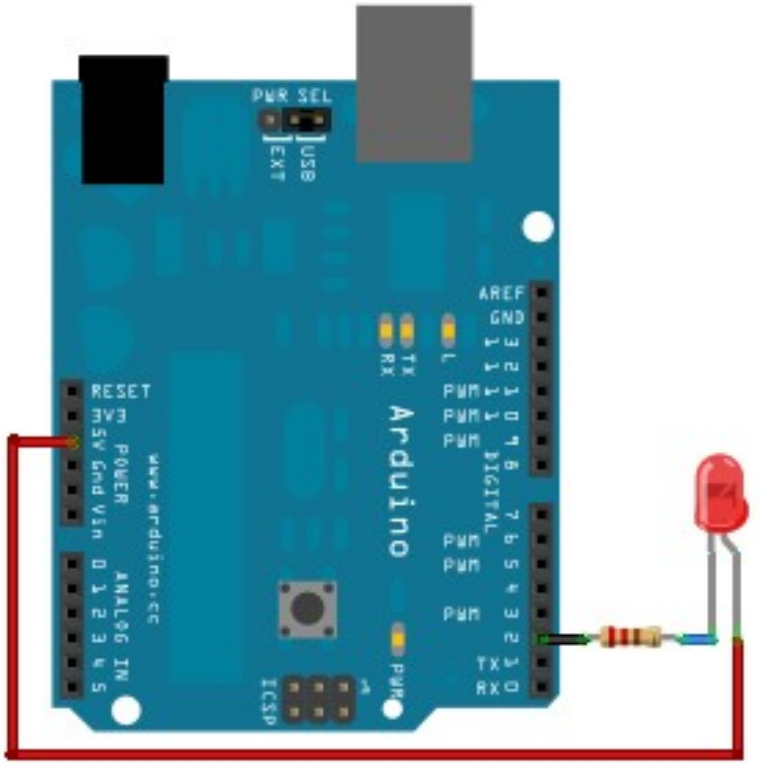
```
1  //fonction d'initialisation de la carte
2  void setup()
3  {
4      //contenu de l'initialisation
5  }
6
7  //fonction principale, elle se répète (s'exécute) à l'infini
8  void loop()
9  {
10     //contenu de votre programme
11 }
```

Lien vers page Eskimon

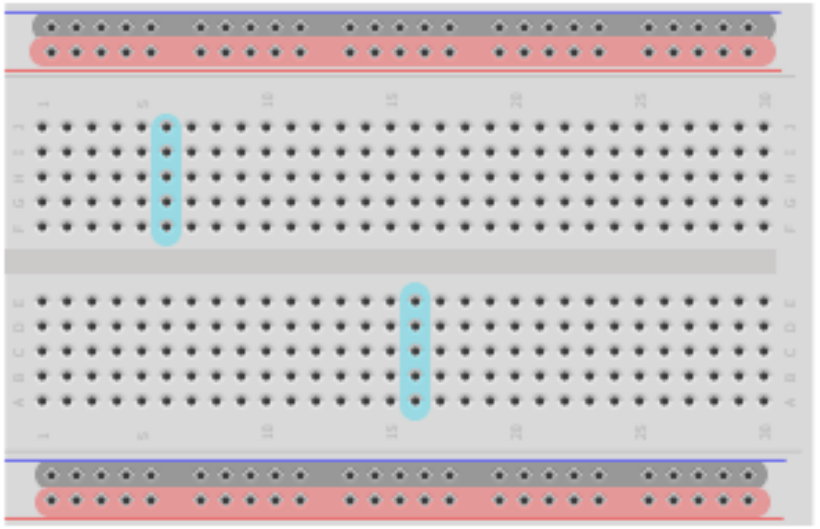
Tutoriel inspiré de l'excellent site http://eskimon.fr

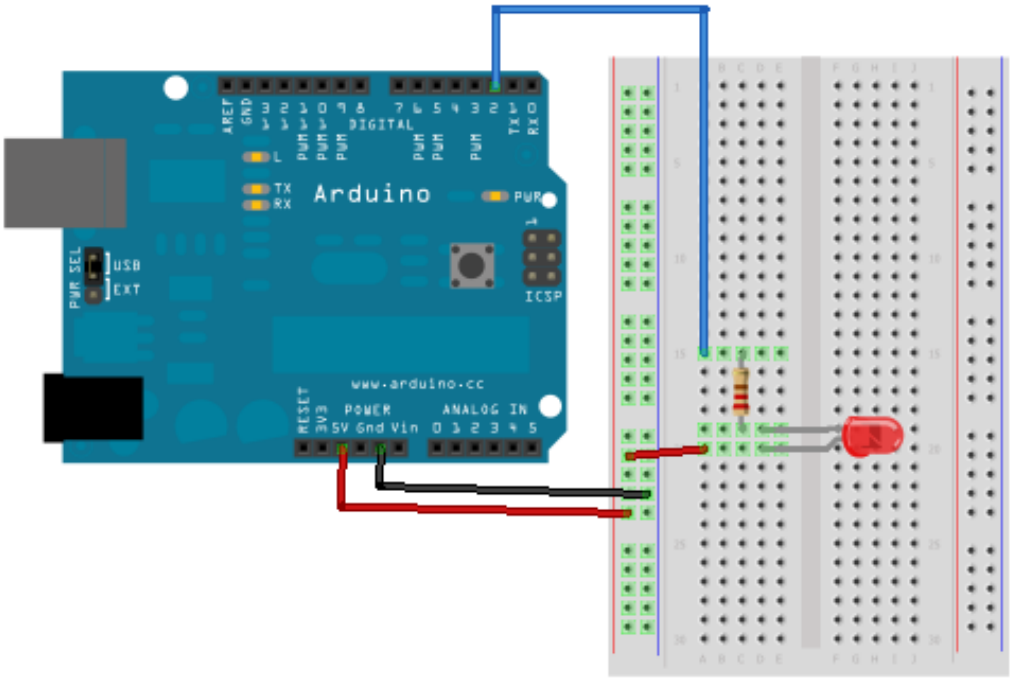
Premier programme : Allumer un LED





Tutoriel inspiré de l'excellent site http://eskimon.fr





Tutoriel inspiré de l'excellent site http://eskimon.fr

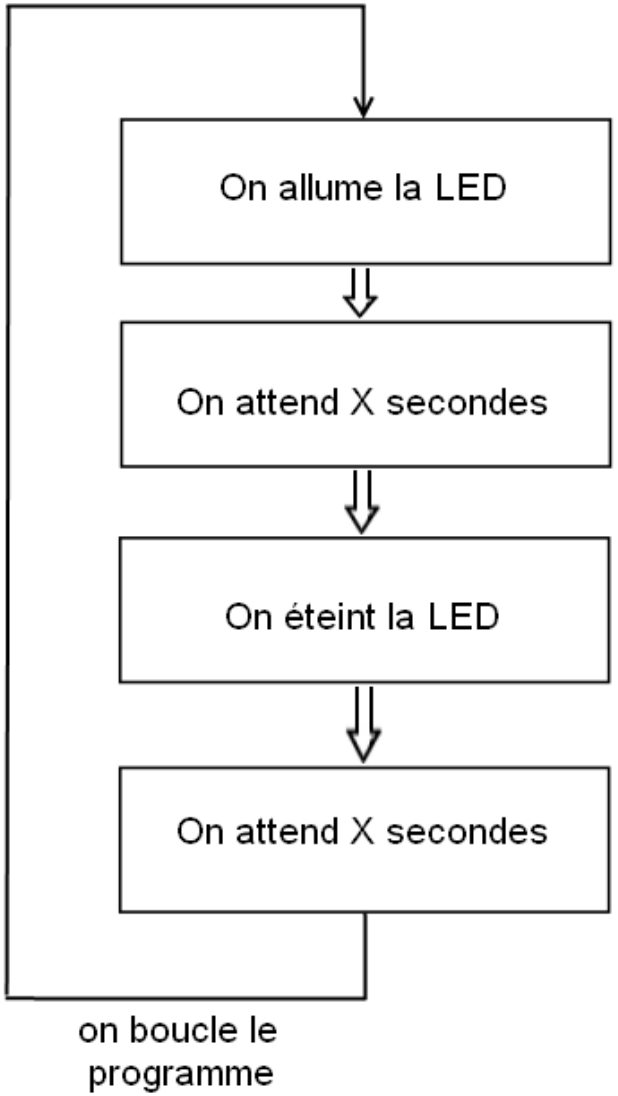
```
1  //définition de la broche 2 de la carte en tant que variable
2  const int led_rouge = 2;
3
4  //fonction d'initialisation de la carte
5  void setup()
6  {
7      //initialisation de la broche 2 comme étant une sortie
8      pinMode(led_rouge, OUTPUT);
9  }
10
11 //fonction principale, elle se répète (s'exécute) à l'infini
12 void loop()
13 {
14     // écriture en sortie (broche 2) d'un état BAS
15     digitalWrite(led_rouge, LOW);
16 }
```

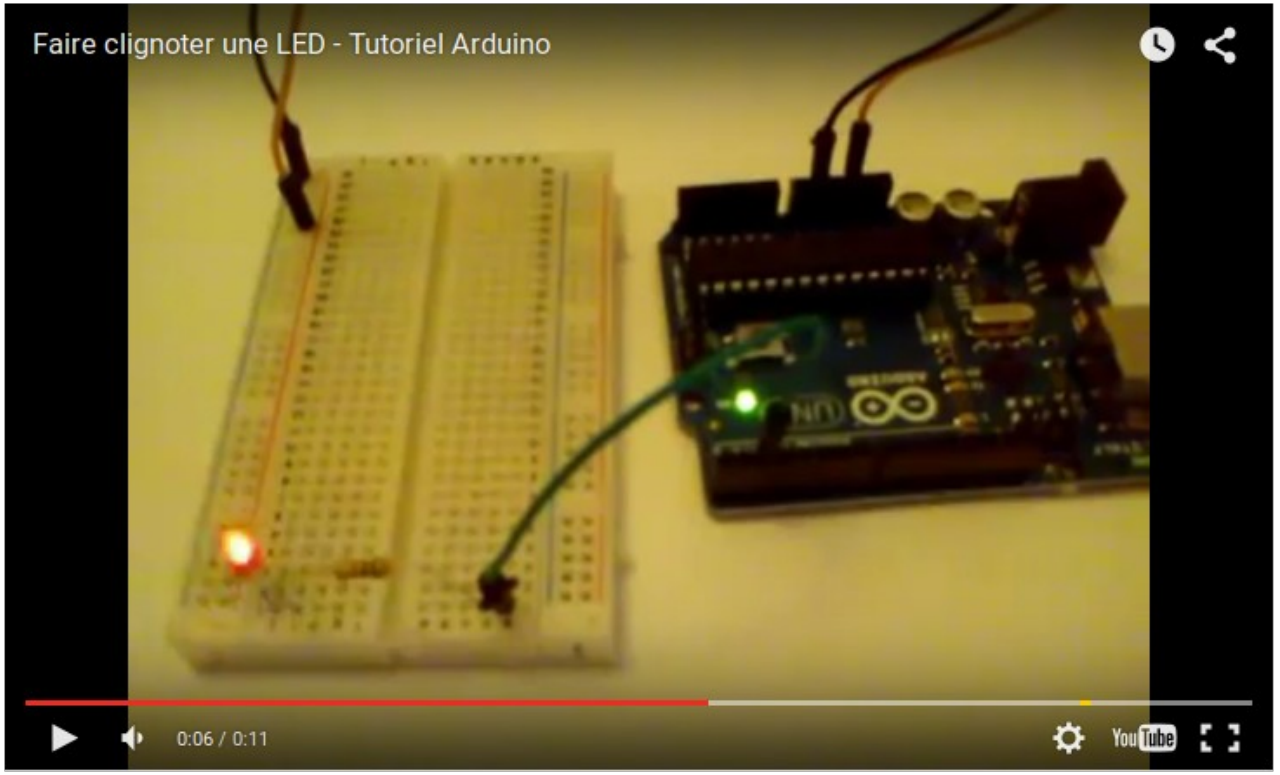
Lien vers la page Eskimon

Tutoriel inspiré de l'excellent site http://eskimon.fr

C'est à vous ! Réalisez le programme.

Introduction à la fonction « temps »





Tutoriel inspiré de l'excellent site http://eskimon.fr


```
1  //définition de la broche 2 de la carte en tant que variable
2  const int led_rouge = 2;
3
4  //fonction d'initialisation de la carte
5  void setup()
6  {
7      //initialisation de la broche 2 comme étant une sortie
8      pinMode(led_rouge, OUTPUT);
9  }
10
11 void loop()
12 {
13     // allume la LED
14     digitalWrite(led_rouge, LOW);
15     // fait une pause de 600 ms
16     delay(600);
17     // éteint la LED
18     digitalWrite(led_rouge, HIGH);
19     // fait une pause de 40 ms
20     delay(40);
21 }
```

Lien vers la page Eskimon

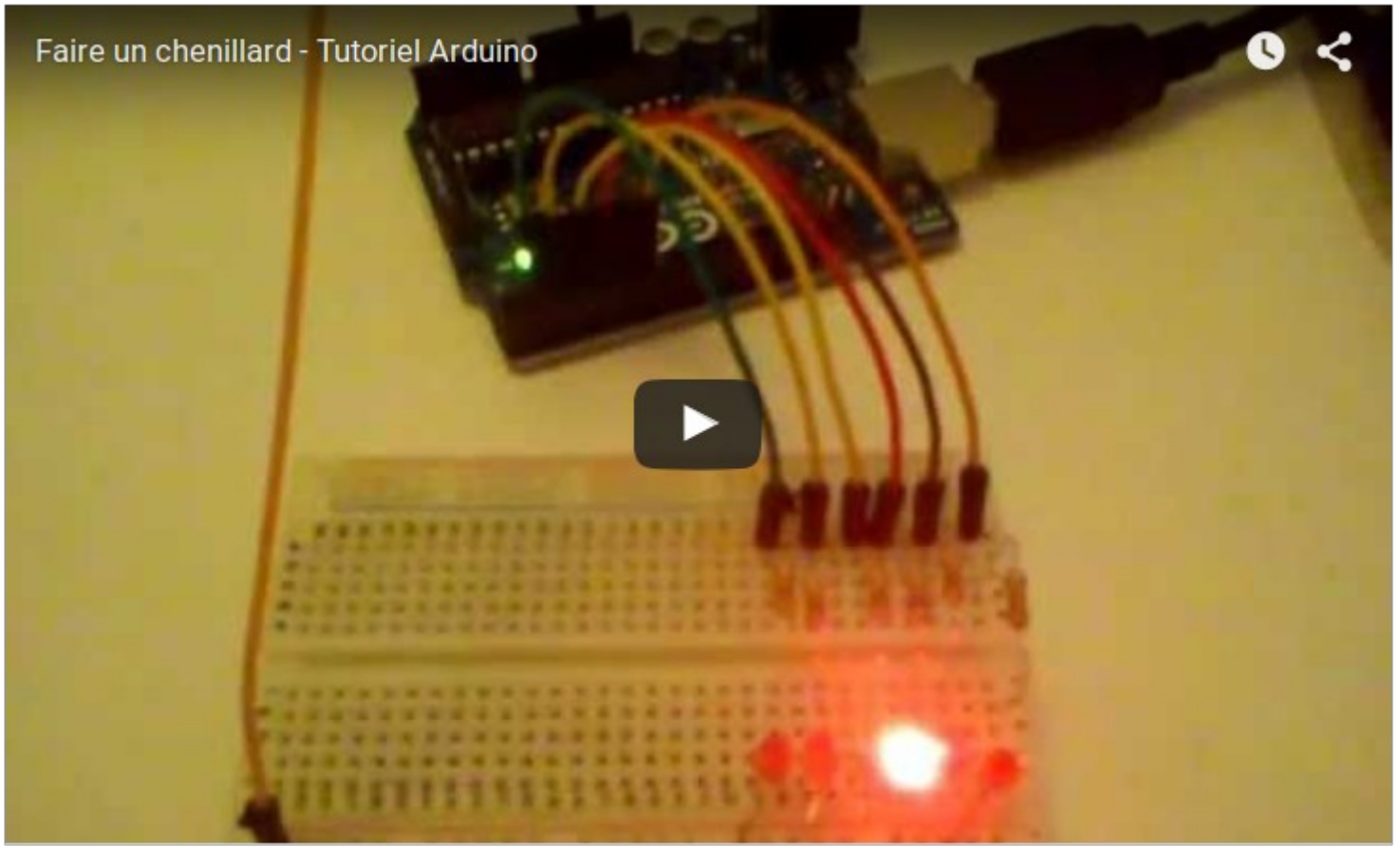
Tutoriel inspiré de l'excellent site http://eskimon.fr

C'est à vous ! Réalisez le programme.

Réaliser un chenillard

Le but du programme

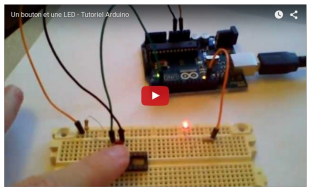
Le but du programme que nous allons créer va consister à réaliser un chenillard. Pour ceux qui ne savent pas ce qu'est un chenillard :

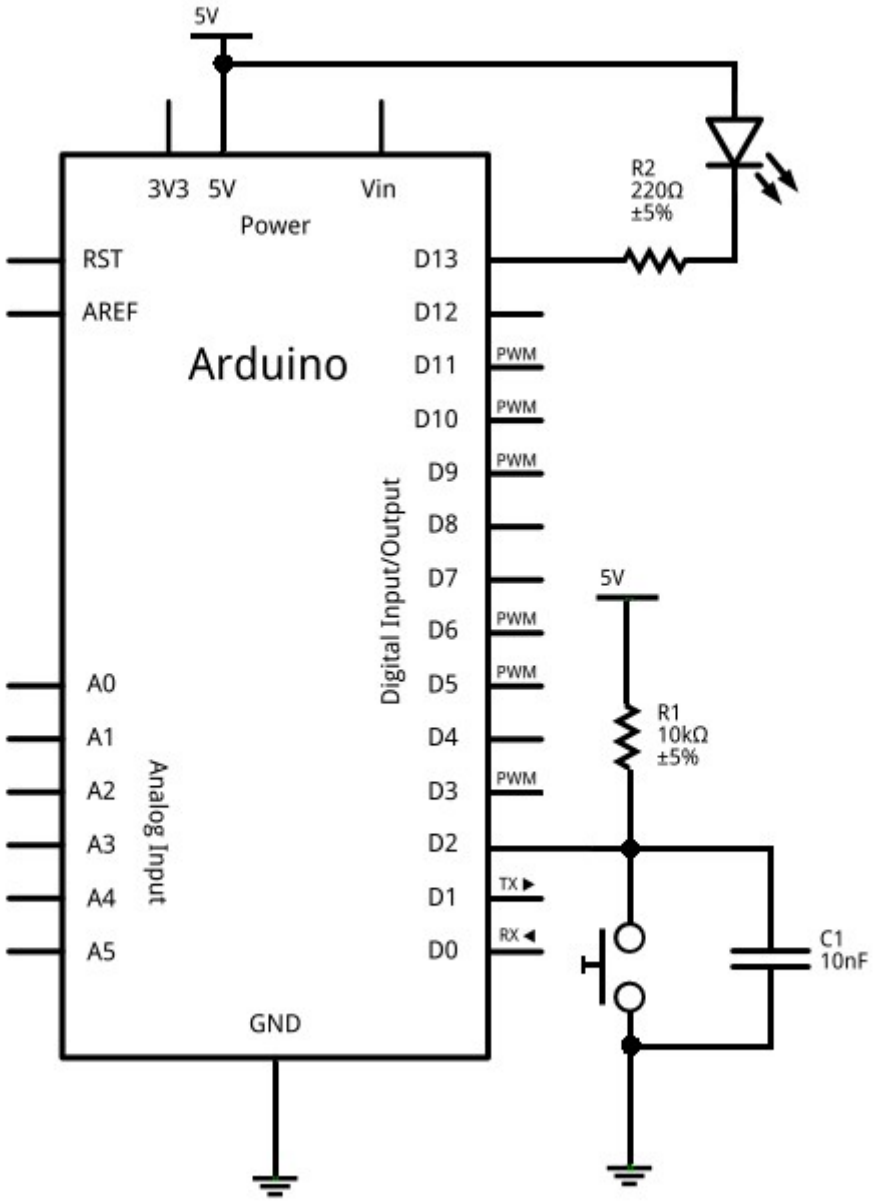


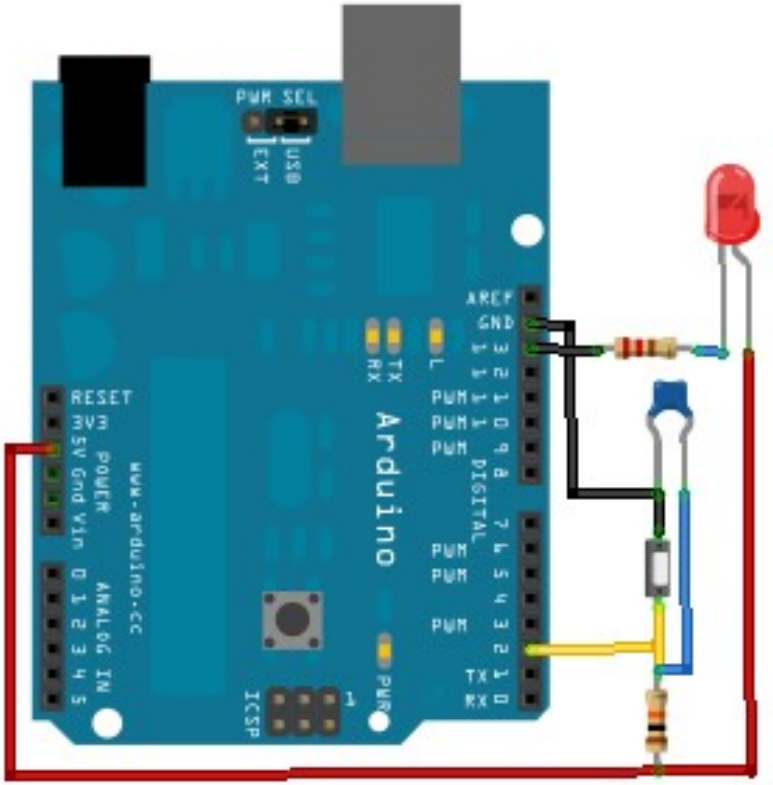
Tutoriel inspiré de l'excellent site http://eskimon.fr

C'est à vous ! Réalisez le programme.

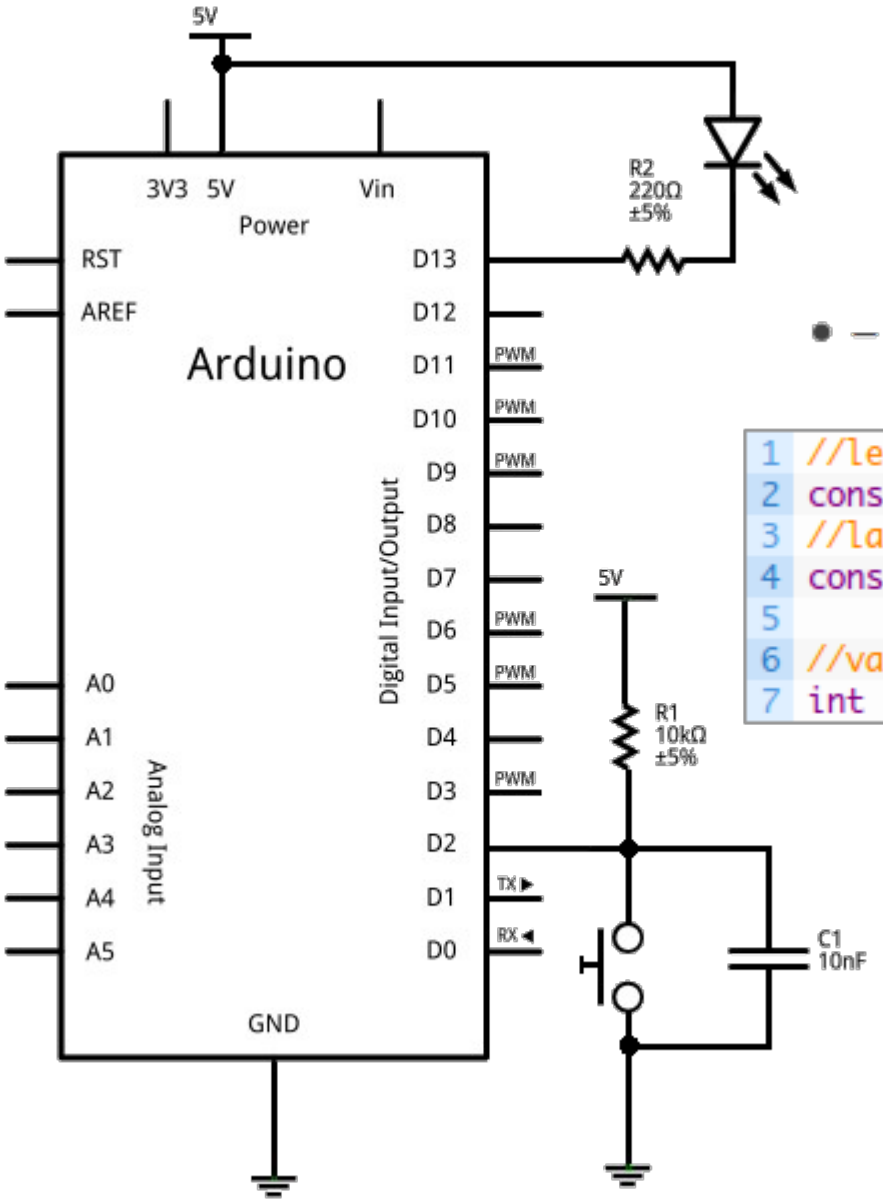
Allumer une Led si le bouton poussoir est actionné !







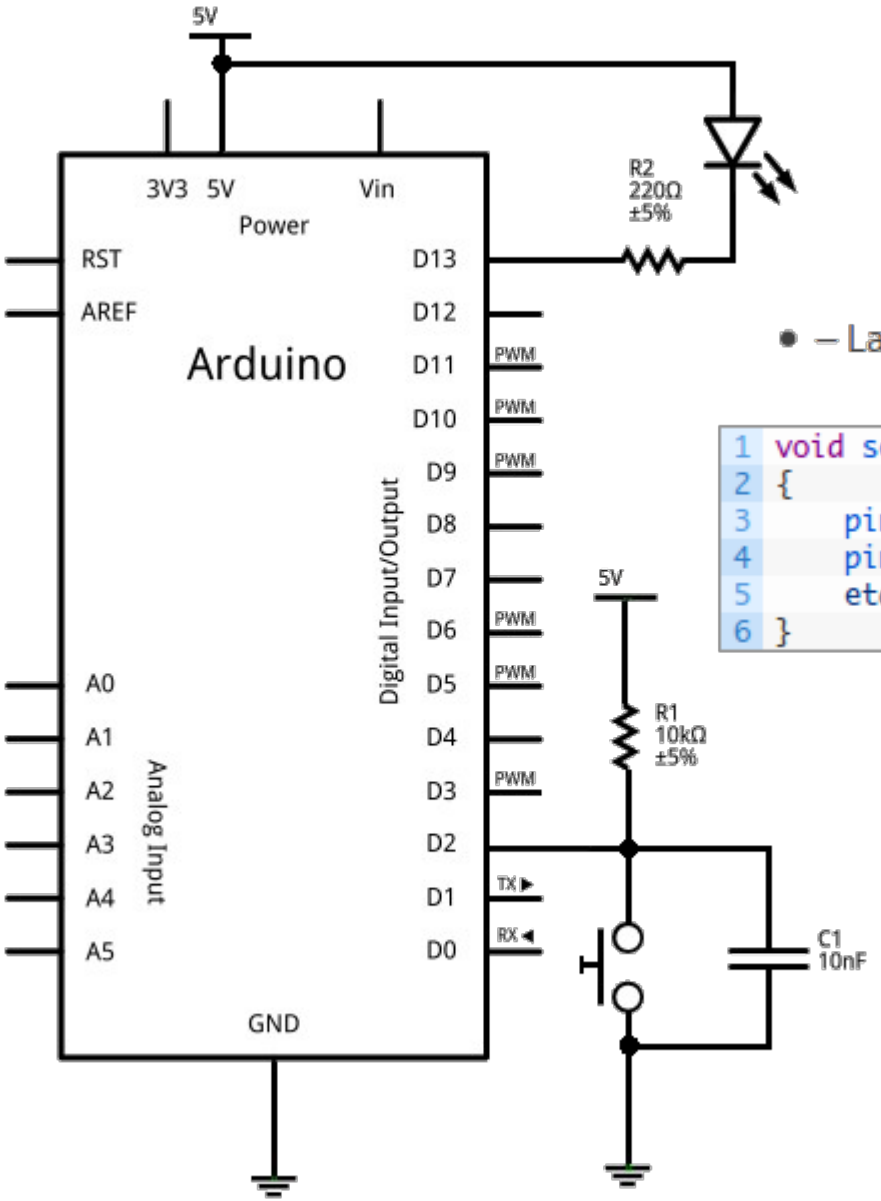
Tutoriel inspiré de l'excellent site http://eskimon.fr



- – Les variables globales

```
1 //le bouton est connecté à la broche 2 de la carte Adruino
2 const int bouton = 2;
3 //la LED à la broche 13
4 const int led = 13;
5
6 //variable qui enregistre l'état du bouton
7 int etatBouton;
```

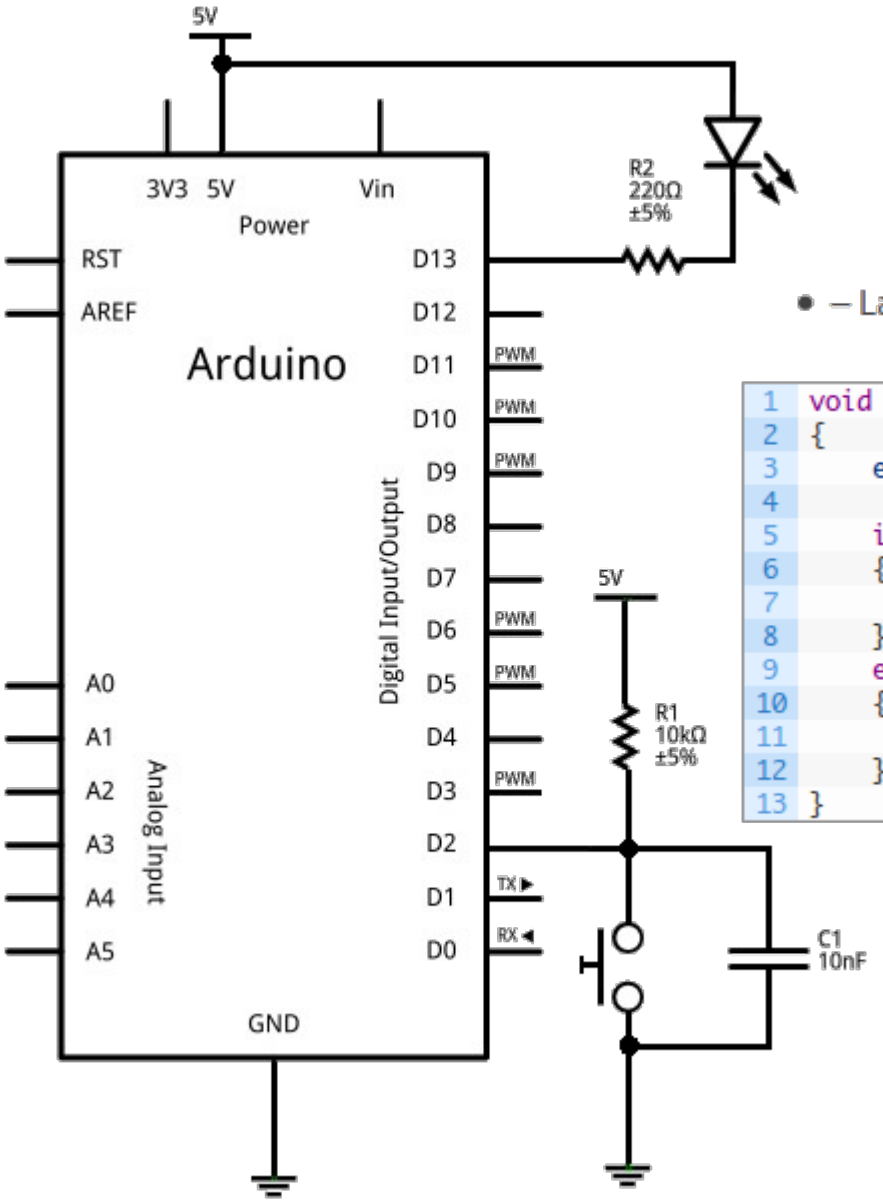
Tutoriel inspiré de l'excellent site http://eskimon.fr



- La fonction setup()

```
1 void setup()
2 {
3     pinMode(led, OUTPUT); //la led est une sortie
4     pinMode(bouton, INPUT); //le bouton est une entrée
5     etatBouton = HIGH; //on initialise l'état du bouton comme "relaché"
6 }
```

Tutoriel inspiré de l'excellent site http://eskimon.fr



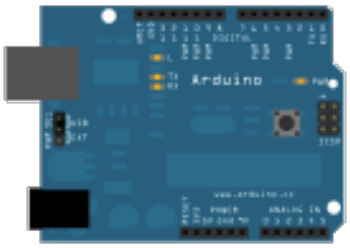
- – La fonction loop()

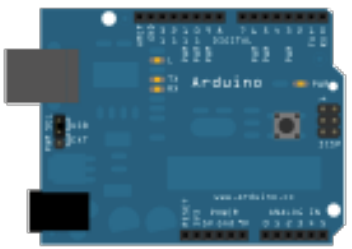
```
1  void loop()
2  {
3      etatBouton = digitalRead(bouton); //Rappel : bouton = 2
4
5      if(etatBouton == HIGH) //test si le bouton a un niveau logique HAUT
6      {
7          digitalWrite(led,LOW); //le bouton est relâché, la LED est allumée
8      }
9      else  //test si le bouton a un niveau logique différent de HAUT (donc BAS)
10     {
11         digitalWrite(led,HIGH); //la LED reste éteinte
12     }
13 }
```

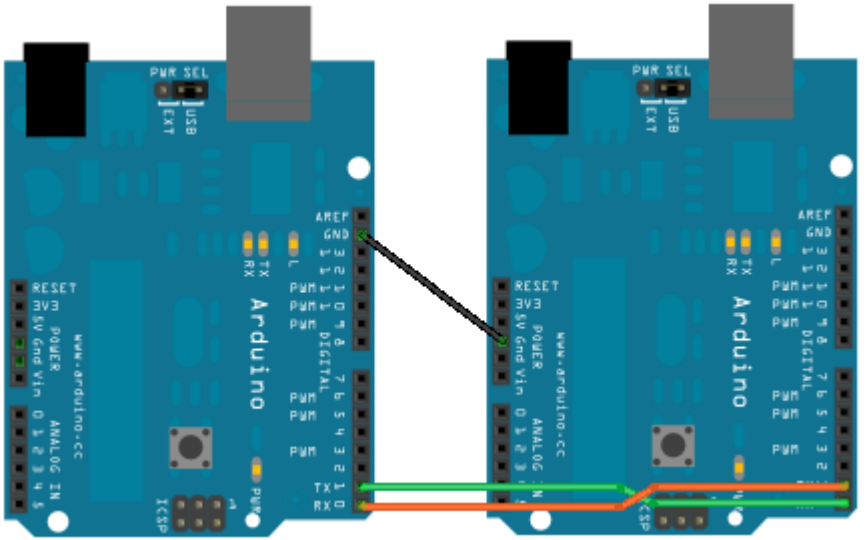
Tutoriel inspiré de l'excellent site http://eskimon.fr

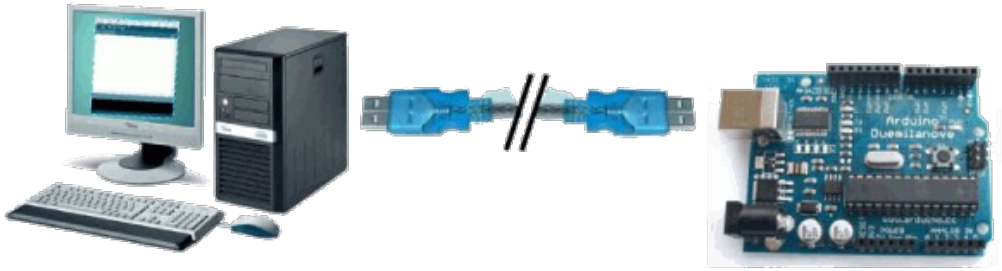
C'est à vous ! Réalisez le programme.

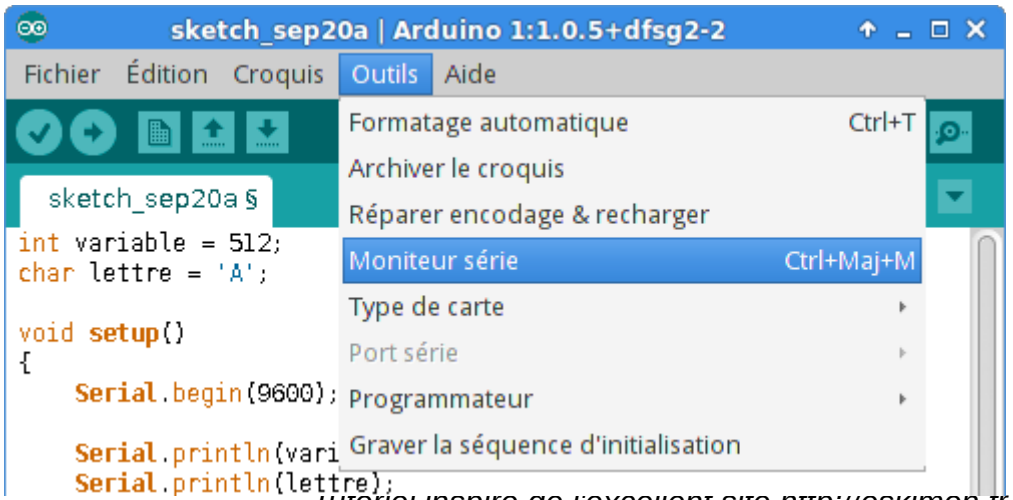
Communication par la voie série



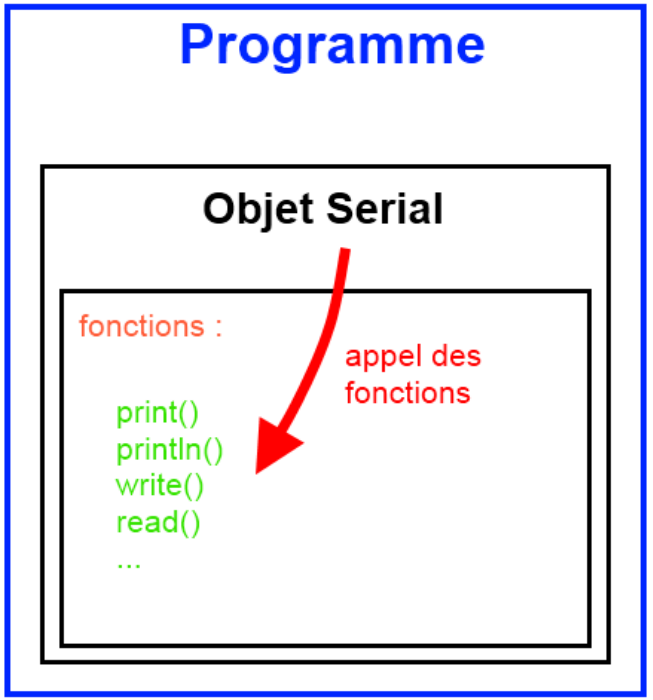


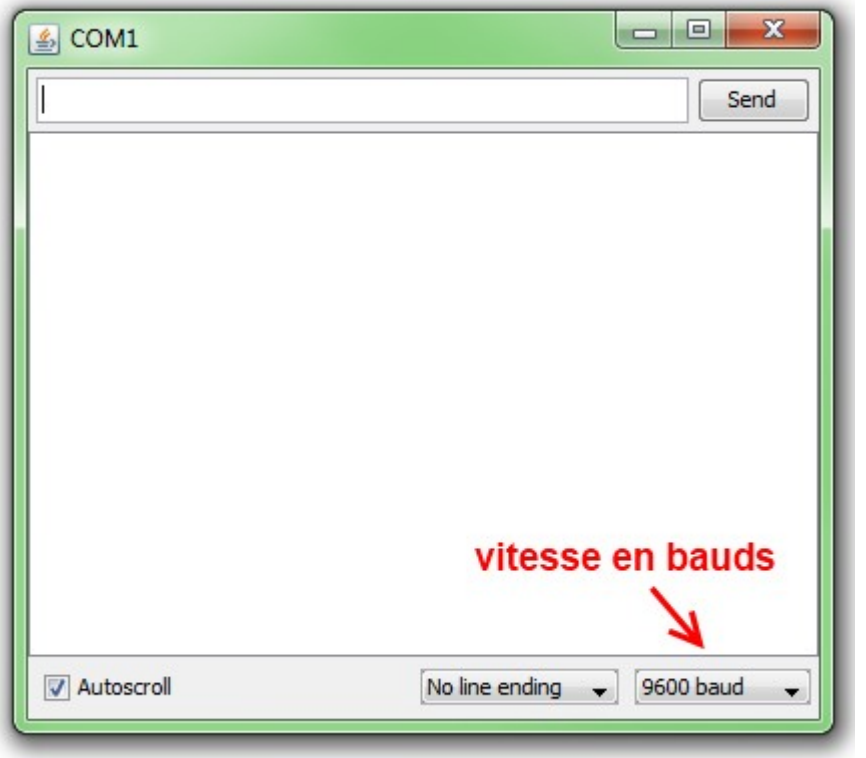






```arduino
void setup()
{
    //création de l'objet Serial (=établissement d'une nouvelle communication série)
    Serial.begin(9600);
    //envoie de la chaine "Salut les zéros !" sur la voie série
    Serial.print("Salut les zéros !");
}
```





Tutoriel inspiré de l'excellent site http://eskimon.fr

Envoyé des données vers la carte Arduino

```
1  int variable = 512;
2  char lettre = 'a';
3
4  void setup()
5  {
6      Serial.begin(9600);
7
8      Serial.println(variable);
9      Serial.print(lettre);
10 }
```

```
1 512
2 a
```

Recevoir des données depuis la carte Arduino

Récupérer les données présente dans le « buffer » et s'il y en a on fera quelque chose !

Buffer = « salle d'attente des données »

```
1 void loop()
2 {
3     int donneesALire = Serial.available(); //lecture du nombre de caractères disponibles dans le buff
4     if(donneesALire > 0) //si le buffer n'est pas vide
5     {
6         //Il y a des données, on les lit et on fait du traitement
7     }
8     //on a fini de traiter la réception ou il n'y a rien à lire
9 }
```

Il est aussi possible de déclencher une action si le buffer est vide...

```
1  void loop()
2  {
3      //on lit le premier caractère non traité du buffer
4      char choseLue = Serial.read();
5
6      if(choseLue == -1) //si le buffer est vide
7      {
8          //Rien à lire, rien lu
9      }
10     else //le buffer n'est pas vide
11     {
12         //On a lu un caractère
13     }
14 }
```

Tutoriel inspiré de l'excellent site http://eskimon.fr

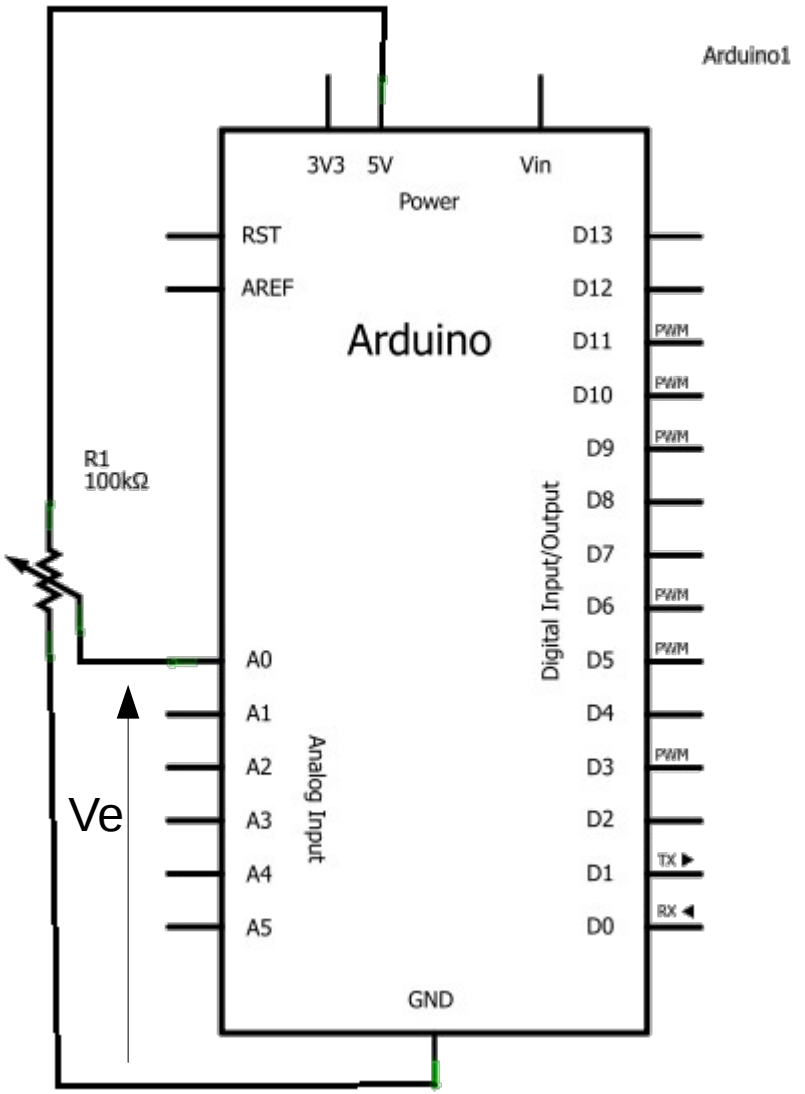
Exemple de code complet

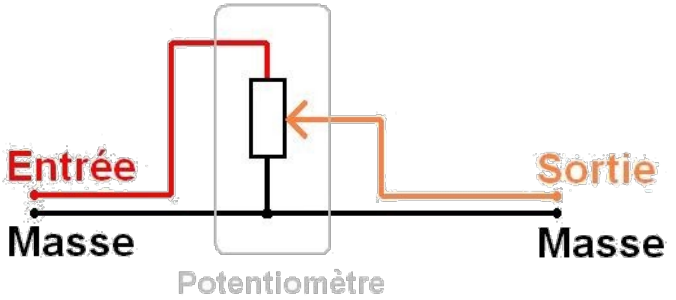
```
1  void setup()
2  {
3      Serial.begin(9600);
4  }
5
6  void loop()
7  {
8      char carlu = 0; //variable contenant le caractère à lire
9      int cardispo = 0; //variable contenant le nombre de caractère disponibles dans le buffer
10
11     cardispo = Serial.available();
12
13     while(cardispo > 0) //tant qu'il y a des caractères à lire
14     {
15         carlu = Serial.read(); //on lit le caractère
16         Serial.print(carlu); //puis on le renvoi à l'expéditeur tel quel
17         cardispo = Serial.available(); //on relit le nombre de caractères dispo
18     }
19     //fin du programme
20 }
```

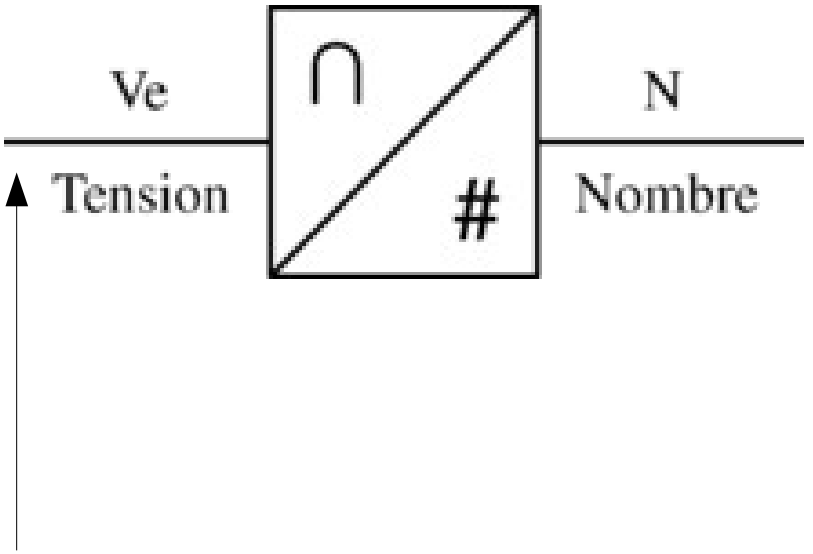
Lien vers site Eskimon

Tutoriel inspiré de l'excellent site http://eskimon.fr

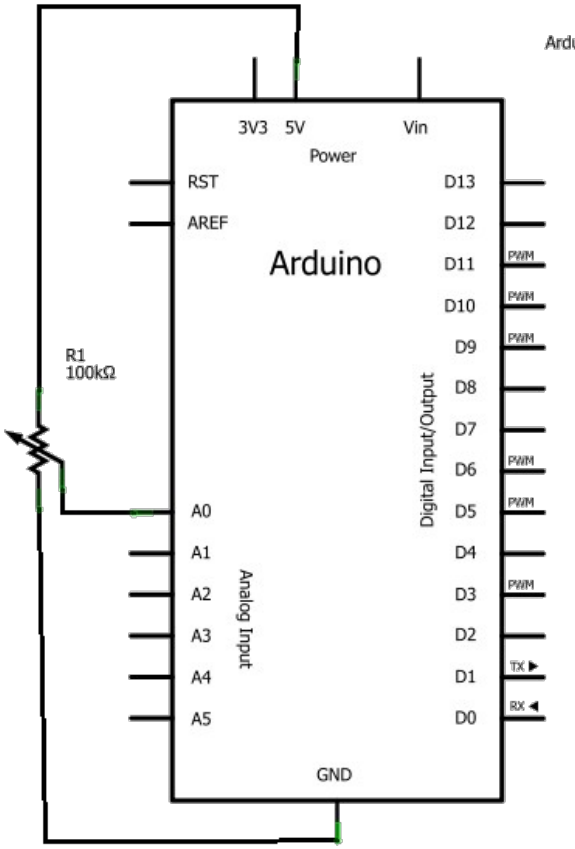
Les entrées analogiques de l'Arduino

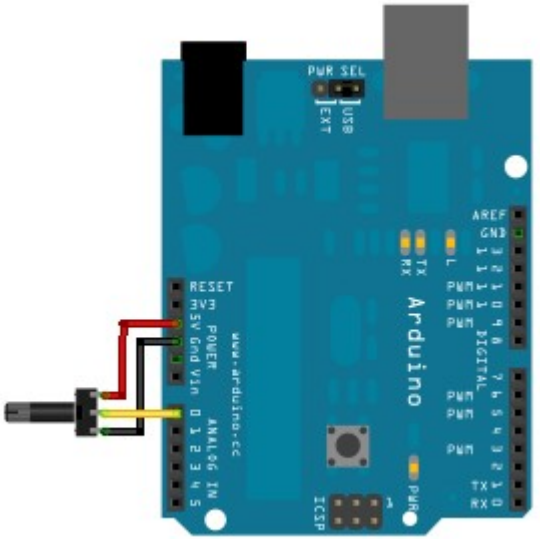


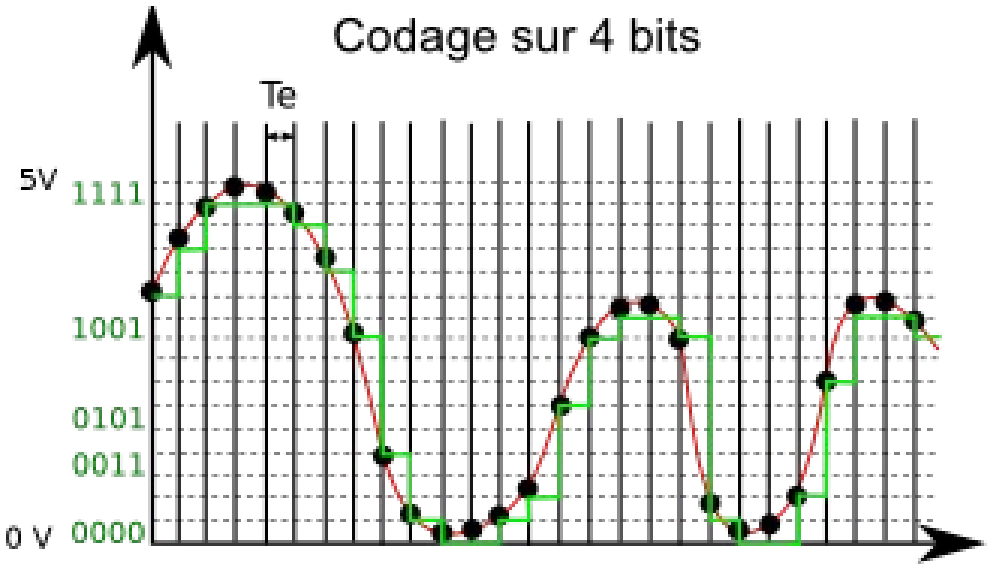


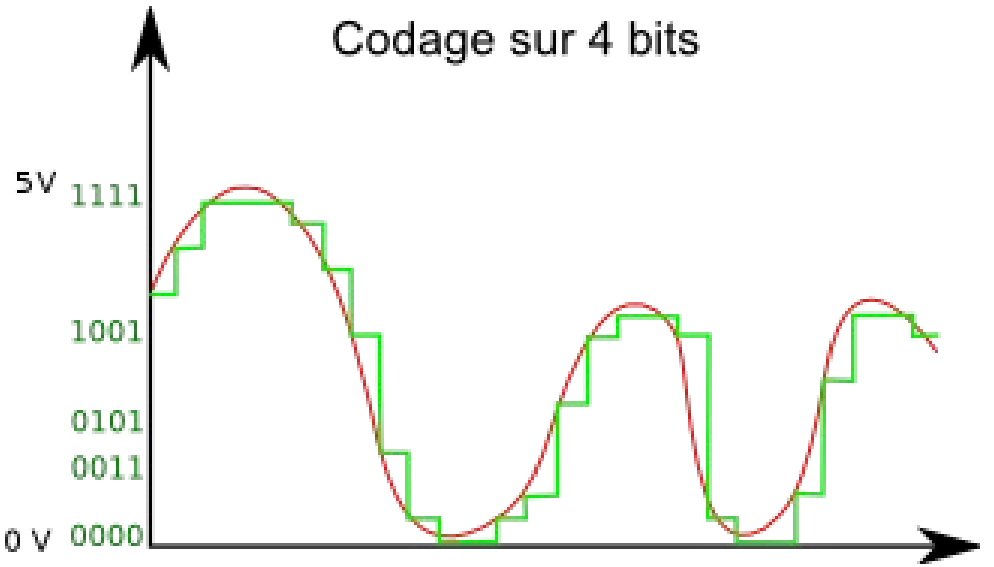


Tutoriel inspiré de l'excellent site http://eskimon.fr





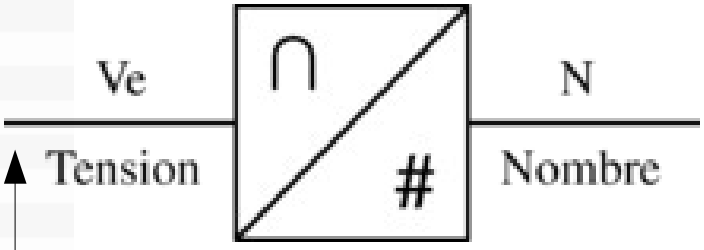




Tutoriel inspiré de l'excellent site http://eskimon.fr

On souhaite afficher sur le terminal série la tension lue sur l'entrée analogique A0


```ino
// le potentiomètre, branché sur la broche analogique 0
const int potar = 0;
//variable pour stocker la valeur lue après conversion
int valeurLue;
//on convertit cette valeur en une tension
float tension;

void setup()
{
    //on se contente de démarrer la liaison série
    Serial.begin(9600);
}

void loop()
{
    //on convertit en nombre binaire la tension lue en sortie du potentiomètre
    valeurLue = analogRead(potar);

    //on traduit la valeur brute en tension (produit en croix)
    tension = valeurLue * 5.0 / 1023;

    //on affiche la valeur lue sur la liaison série
    Serial.print("valeurLue = ");
    Serial.println(valeurLue);

    //on affiche la tension calculée
    Serial.print("Tension = ");
    Serial.print(tension,2);
    Serial.println(" V");

    //on saute une ligne entre deux affichages
    Serial.println();
    //on attend une demi-seconde pour que l'affichage ne soit pas trop rapide
    delay(500);
}
```

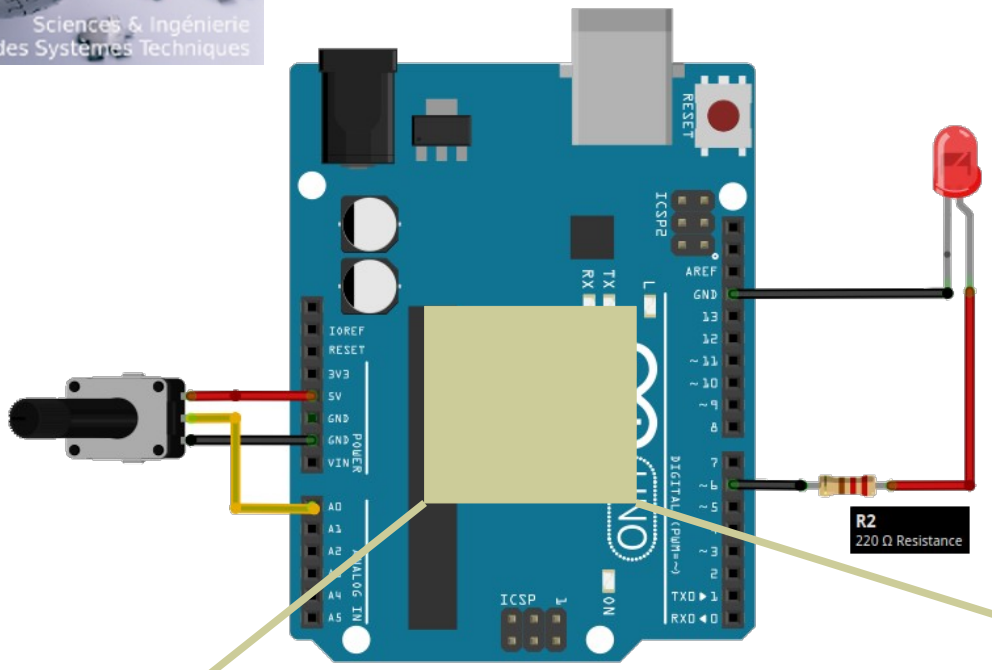


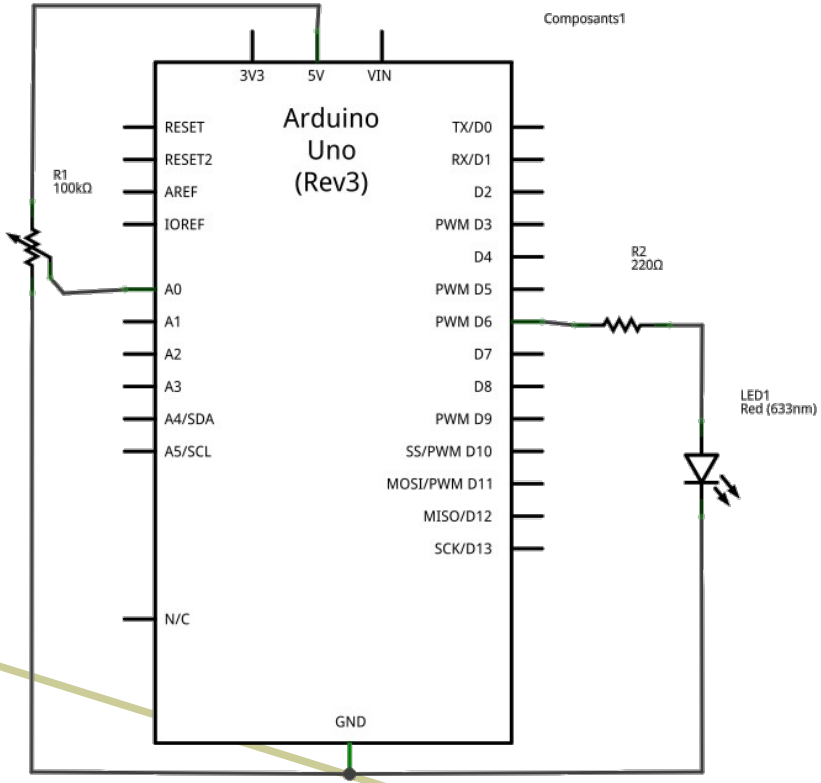
10 bits -->0 à 1023

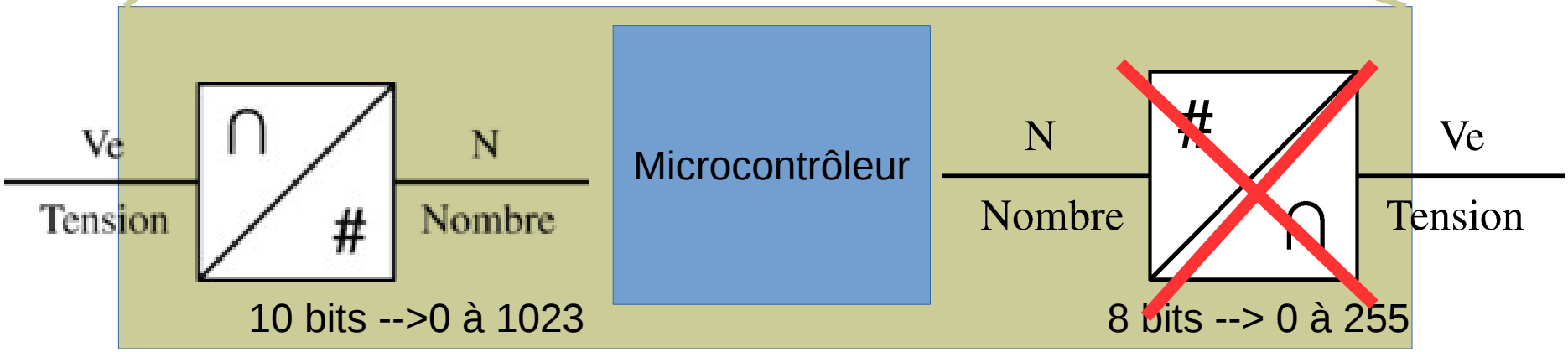
0V-->00 0000 0000b = 0dec
5V-->11 1111 1111b = 1023dec

Tutoriel inspiré de l'excellent site http://eskimon.fr

Les sorties analogiques de l'Arduino

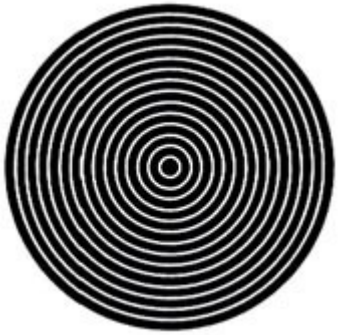


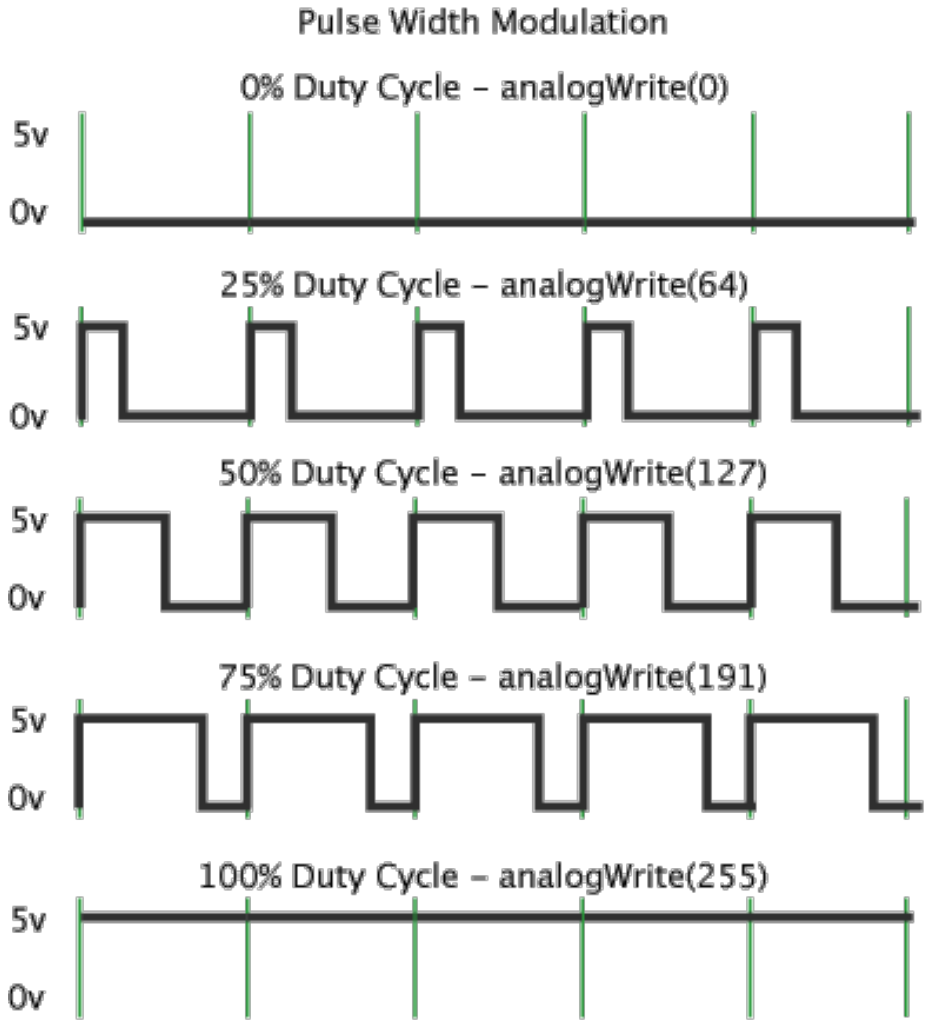




0V-->00 0000 0000b = 0dec
5V-->11 1111 1111b = 1023dec

0000 0000b=0dec-->0V
1111 1111b=255dec-->5V





Eteinte !

Un peu allumée..

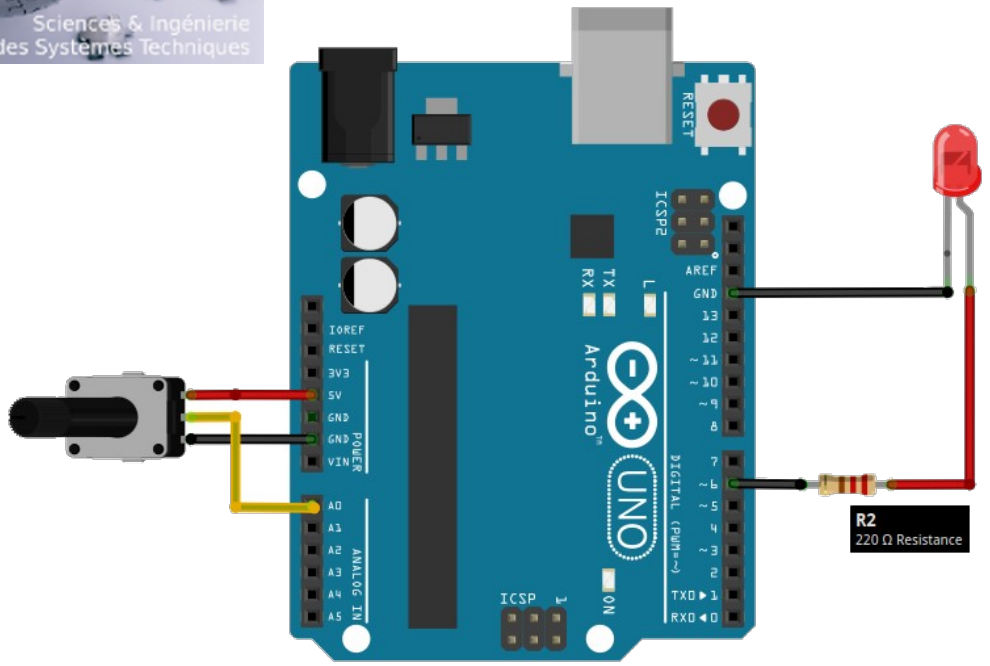
Un peu plus allumée..

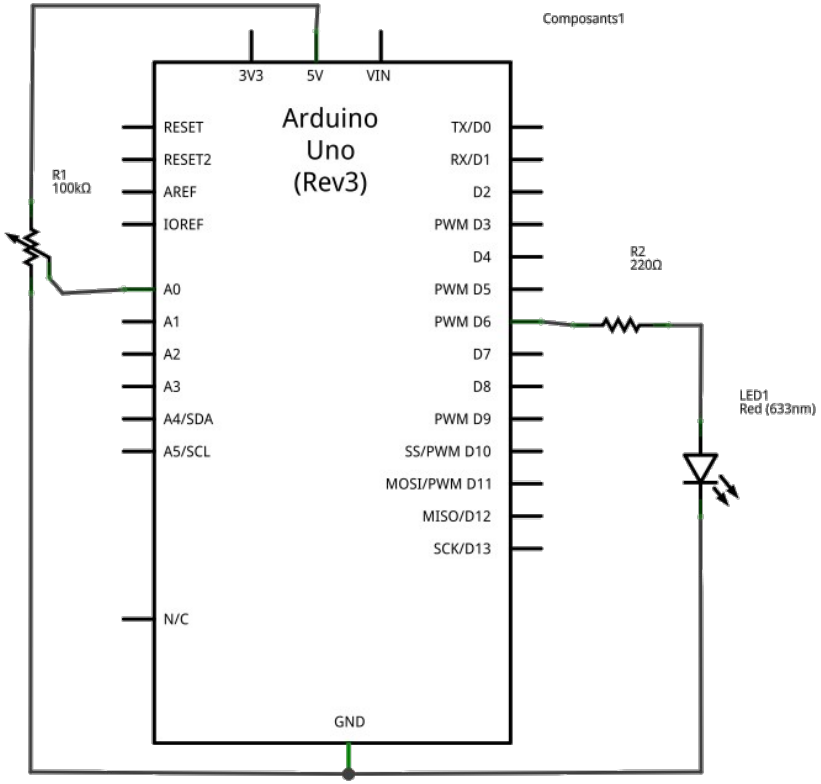
Encore un peu plus allumée..

Carrément allumée !

Tutoriel inspiré de l'excellent site http://eskimon.fr

Faire varier la luminosité d'une LED en agissant sur le potentiomètre :

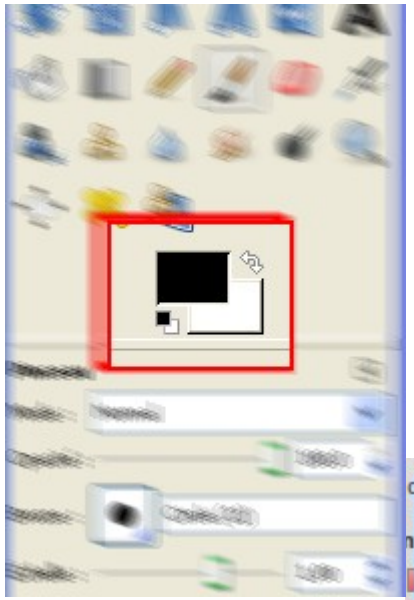




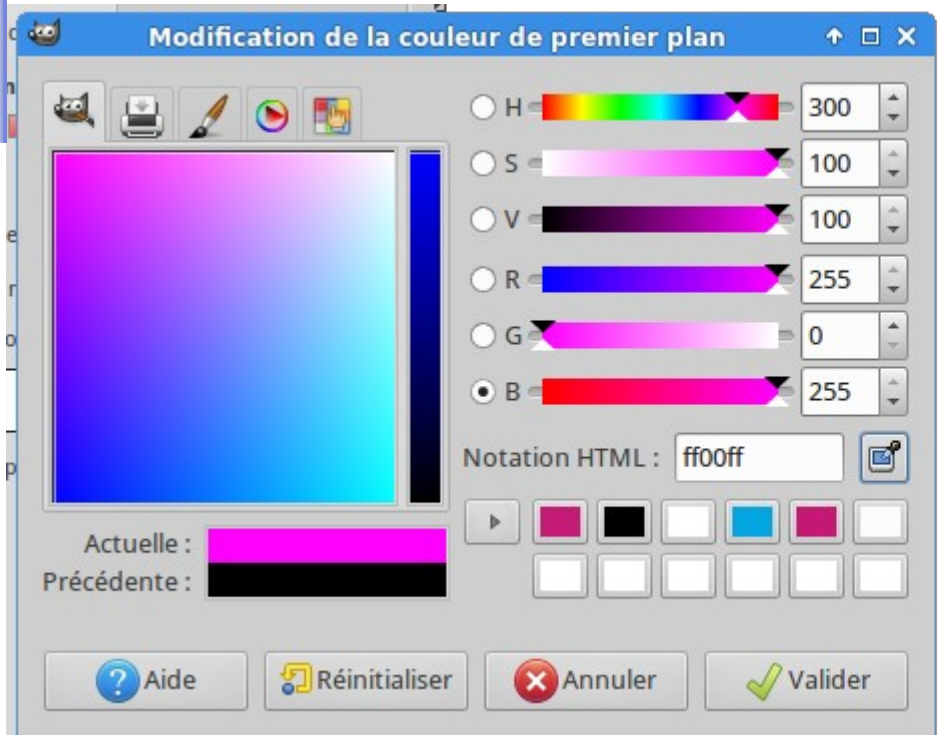

```
1  //une sortie analogique sur la broche 6
2  const int sortieAnalogique = 6;
3
4  void setup()
5  {
6      pinMode(sortieAnalogique, OUTPUT);
7  }
8
9  void loop()
10 {
11     //on met un rapport cyclique de 107/255 = 42 %
12     analogWrite(sortieAnalogique, 107);
13 }
```

C'est à vous ! Réalisez le programme.

Une LED multi-couleurs :
La led RGB



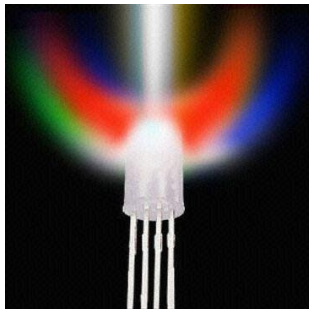
GIMP

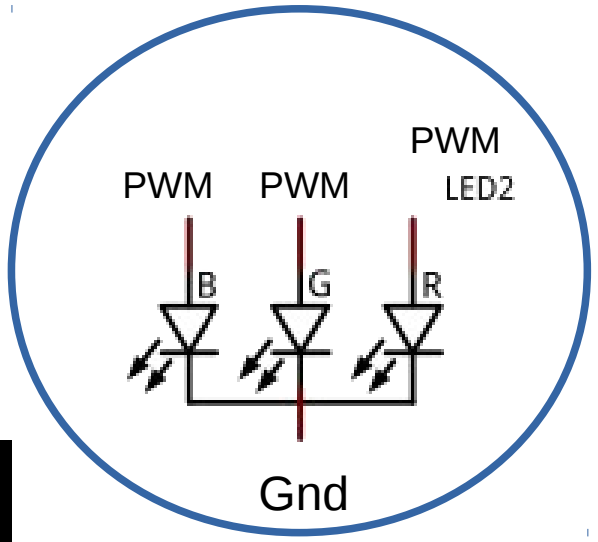


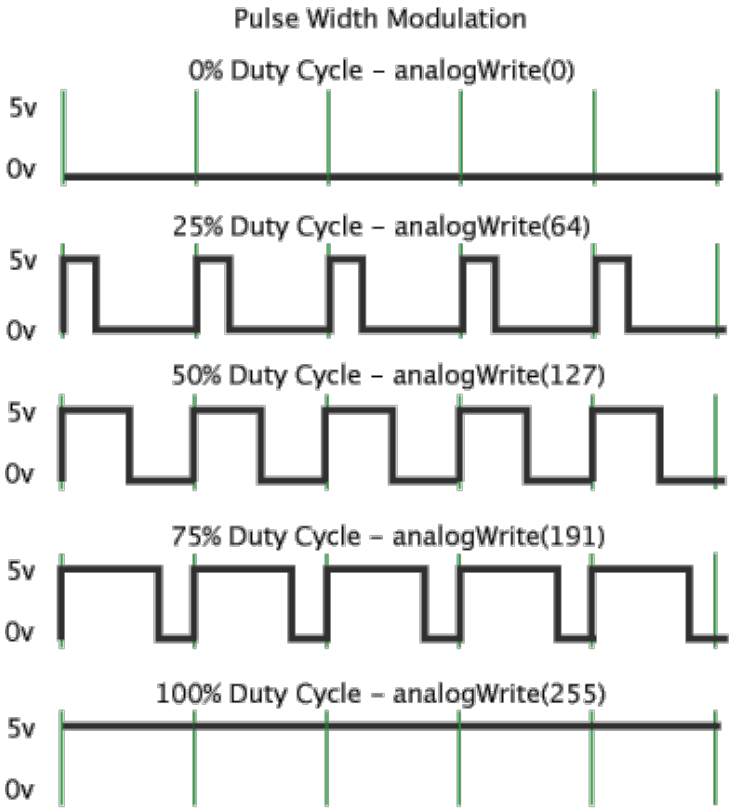
Tutoriel inspiré de l'excellent site http://eskimon.fr

InSyTe
Sciences & Ingénierie des Systèmes Techniques

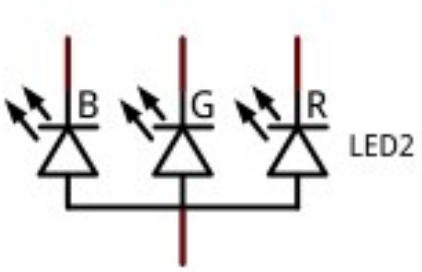
Led RGB à anode commune

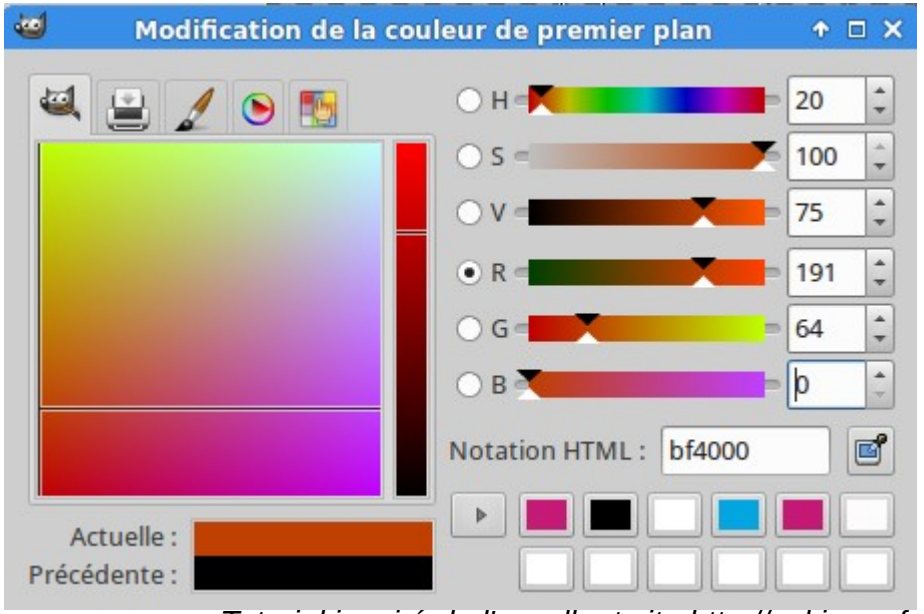




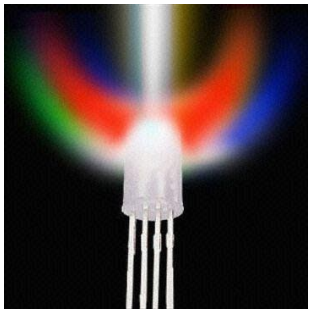


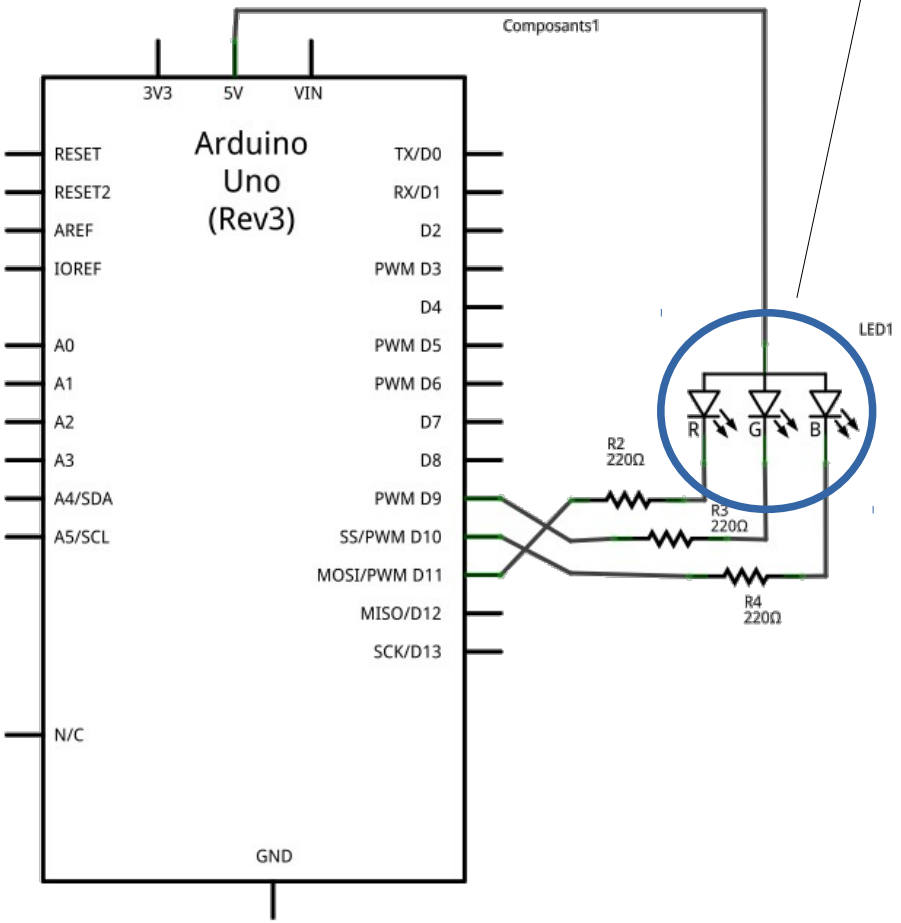
Led RGB à cathode commune

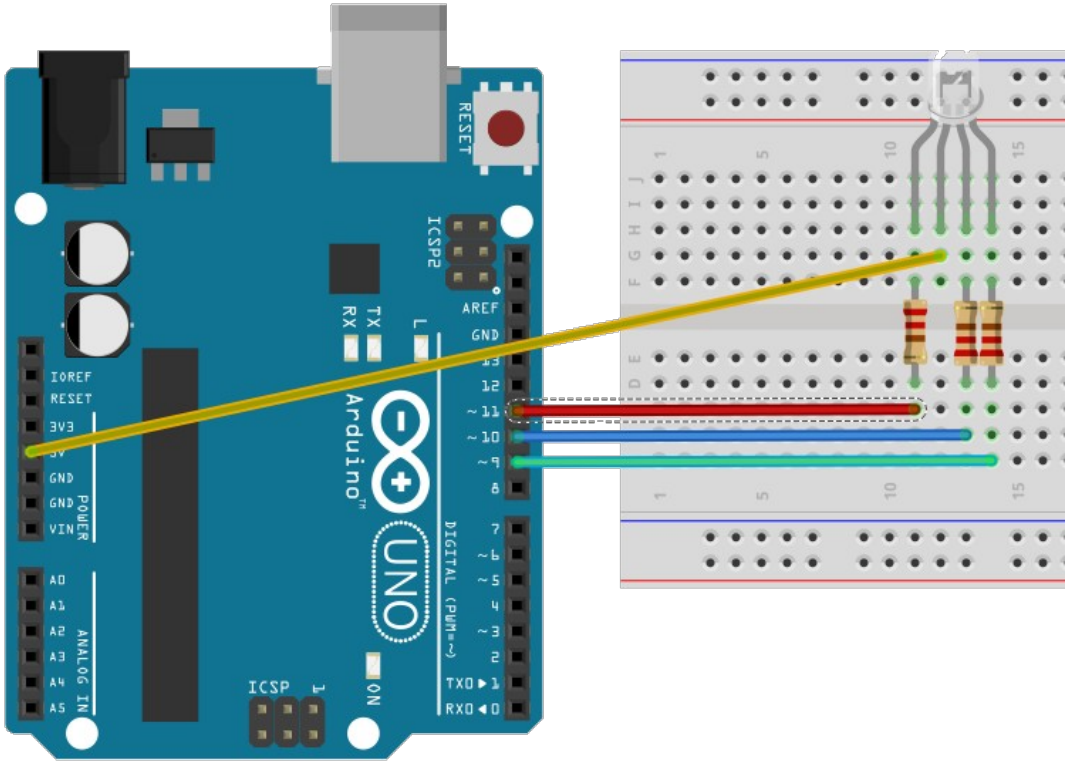




Tutoriel inspiré de l'excellent site http://eskimon.fr



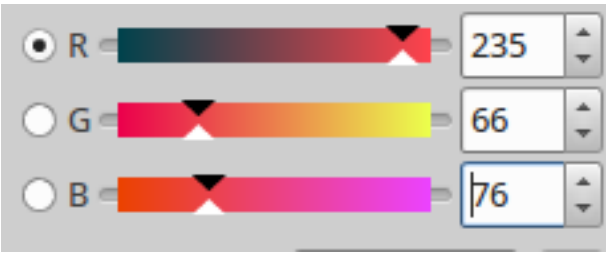




Tutoriel inspiré de l'excellent site http://eskimon.fr

C'est à vous !

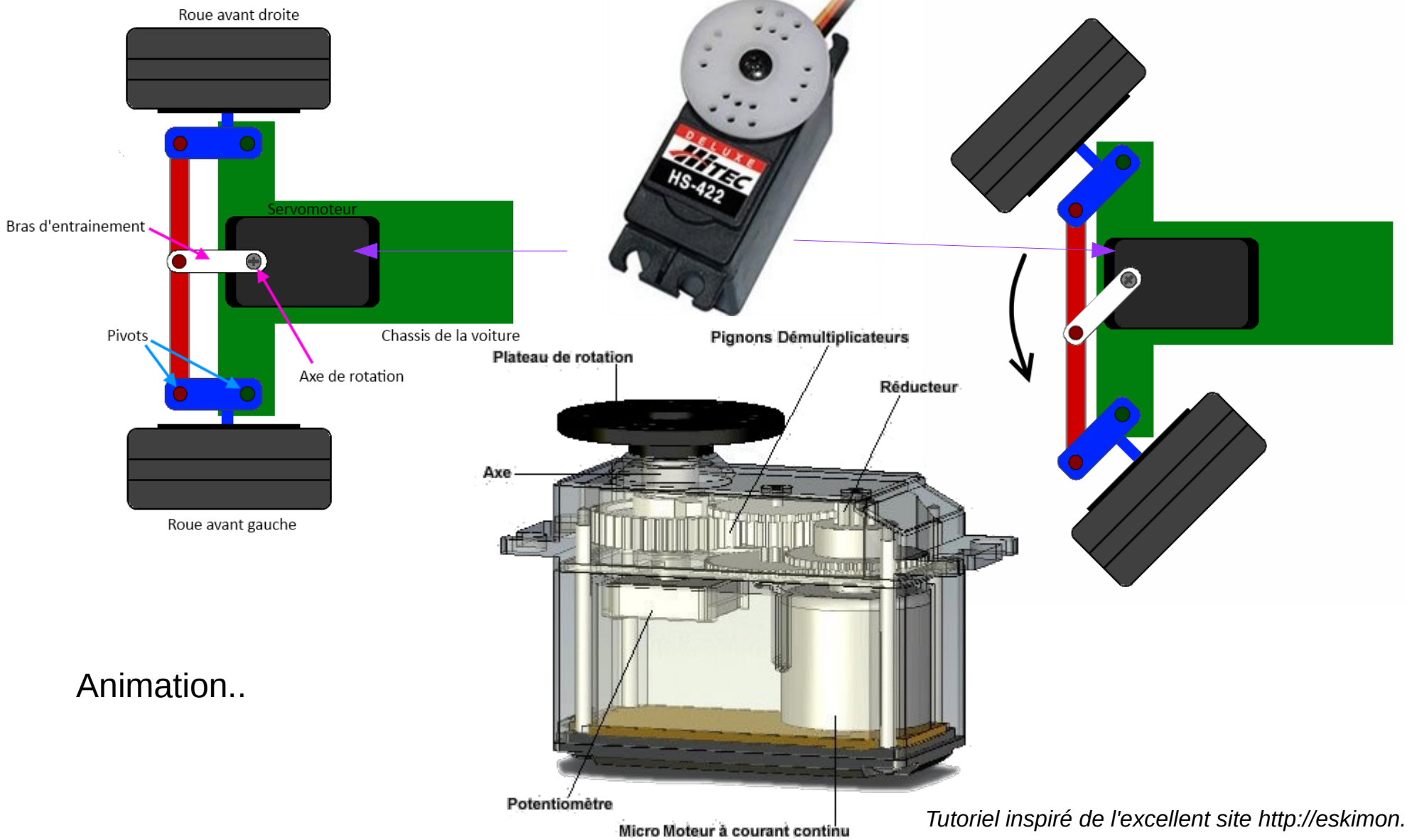
Réalisez le câblage de la diapositive précédente.
En vous inspirant du programme ci-après, réaliser un programme permettant d'obtenir la couleur « Actuelle »




```
1  const int ledRouge = 11;
2  const int ledVerte =  9;
3  const int ledBleue = 10;
4
5  void setup()
6  {
7      //on déclare les broches en sorties
8      pinMode(ledRouge, OUTPUT);
9      pinMode(ledVerte, OUTPUT);
10     pinMode(ledBleue, OUTPUT);
11
12     //on met la valeur de chaque couleur
13     analogWrite(ledRouge,    );
14     analogWrite(ledVerte,    );
15     analogWrite(ledBleue,  );
16 }
17
18 void loop()
19 {
20     //on ne change pas la couleur donc rien à faire dans la boucle principale
21 }
```

Tutoriel inspiré de l'excellent site http://eskimon.fr

Les servo-moteurs avec Arduino



Animation..

Tutoriel inspiré de l'excellent site http://eskimon.fr

Caractéristiques Minimun:

Dimensions : 40.4x19.8x36mm

Poids : 37.2g

Couple : 3.2kg.cm sous 4.8V

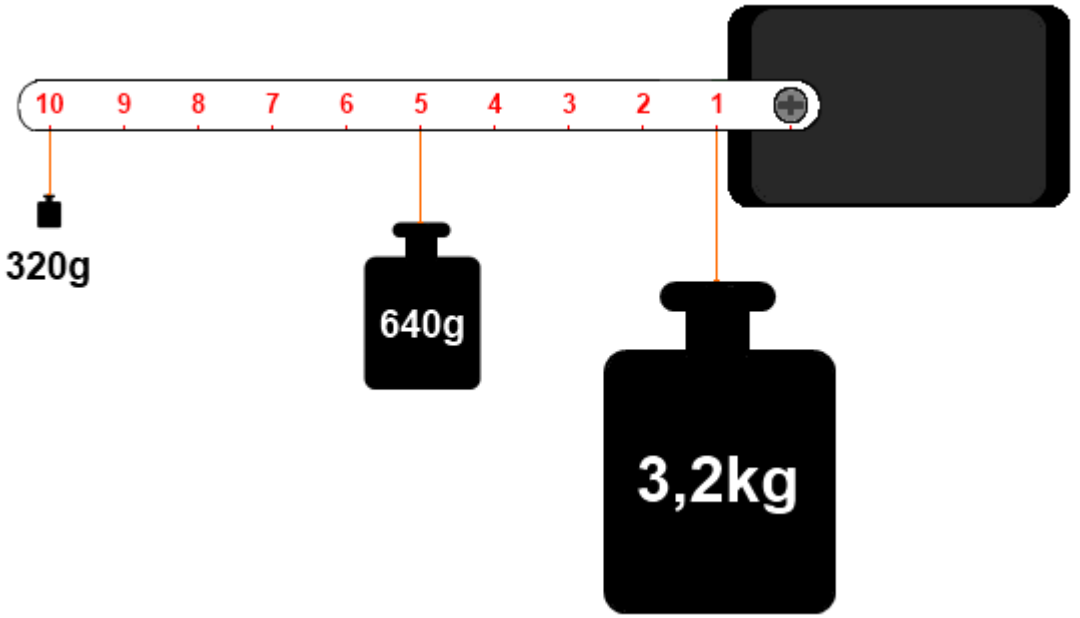
Couple : 4.1kg.cm sous 6V

Vitesse : 0.23s/60° sous 4.8V

Vitesse : 0.19s/60° sous 6V

Pignons en nylon

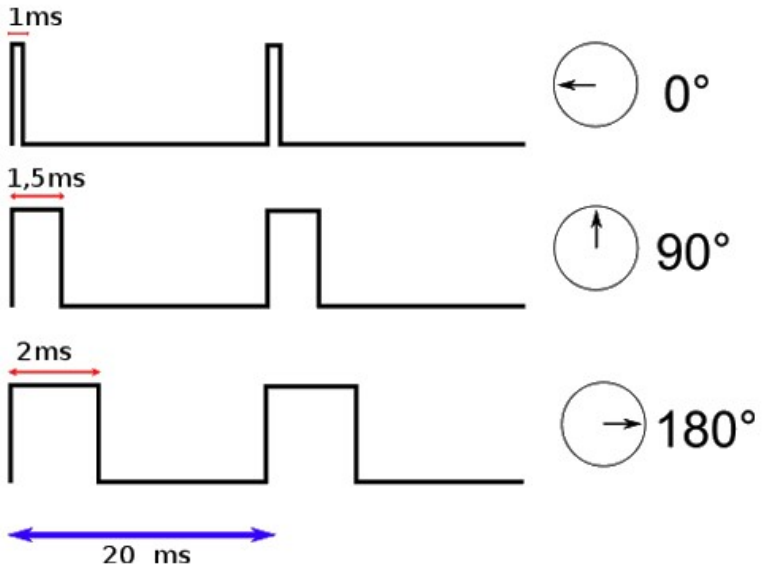
Alimentation : 4.8 à 6V

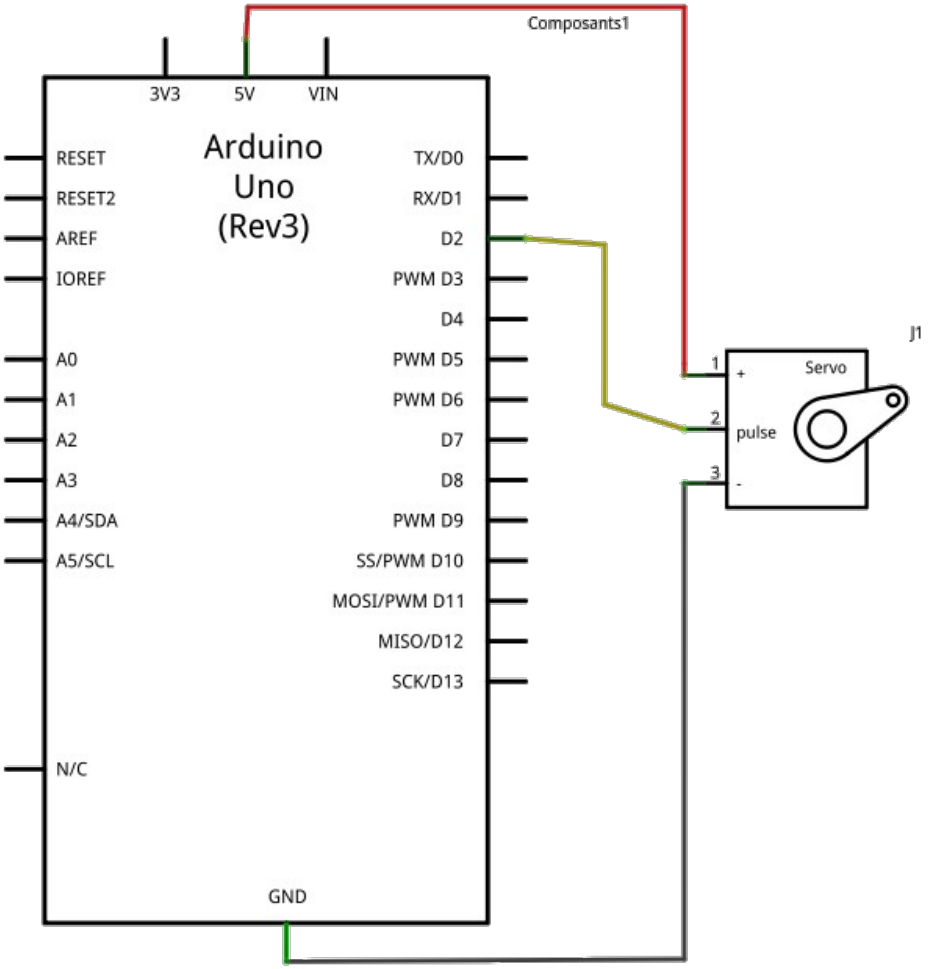


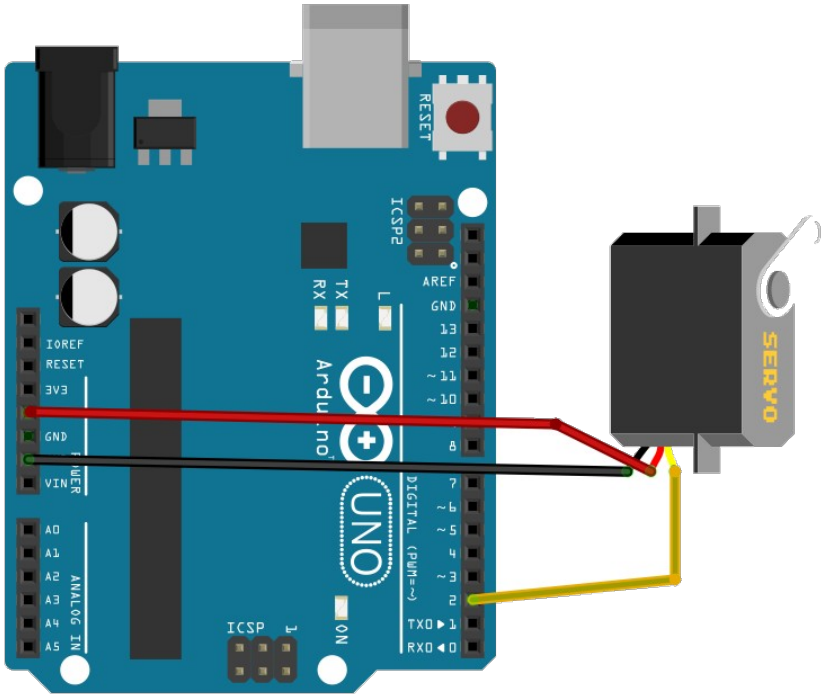
Tutoriel inspiré de l'excellent site http://eskimon.fr

InSyTe
Sciences & Ingénierie des Systèmes Techniques

```arduino
#include <Servo.h>

Servo monServo;

void setup()
{
    monServo.attach(2, 1000, 2000);
    monServo.write(90);
}

void loop()
{
}
```

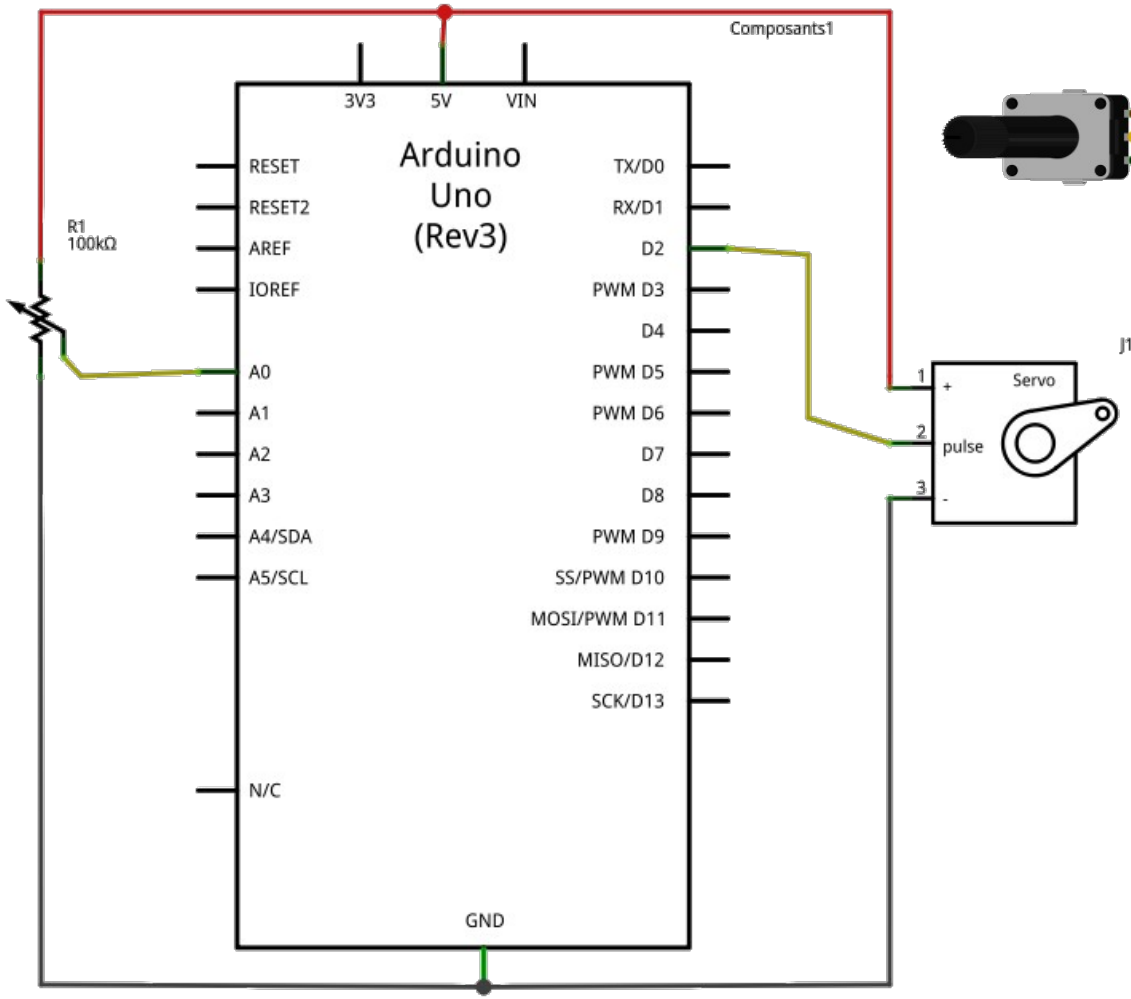


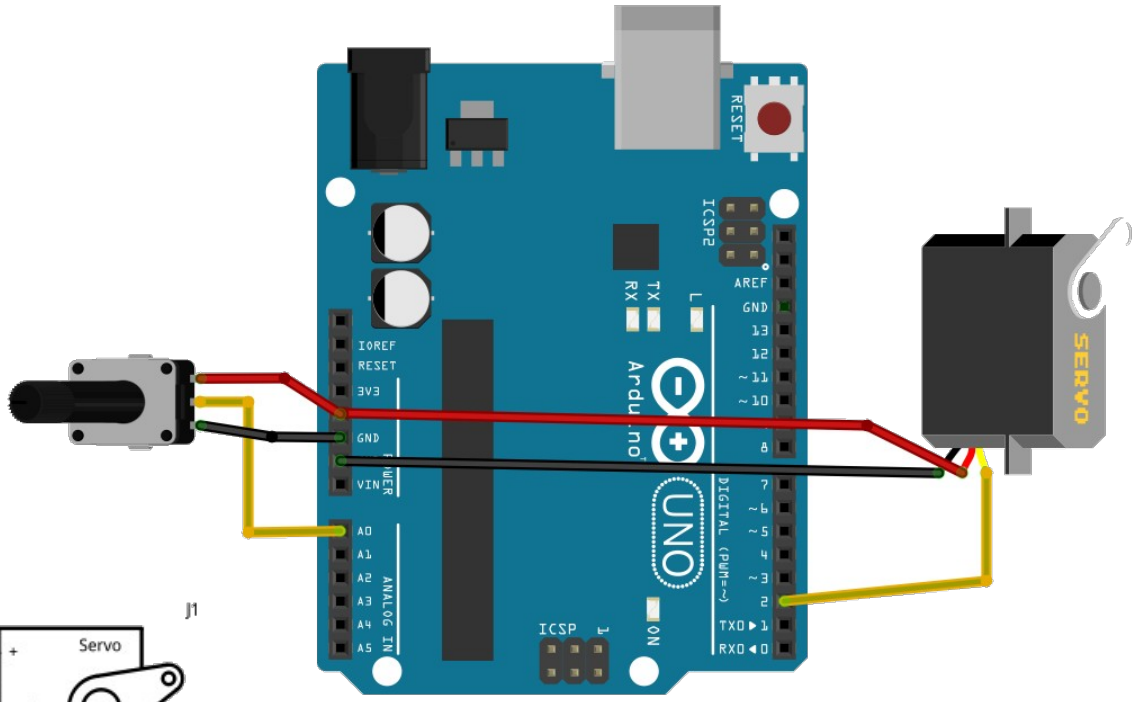




Tutoriel inspiré de l'excellent site http://eskimon.fr

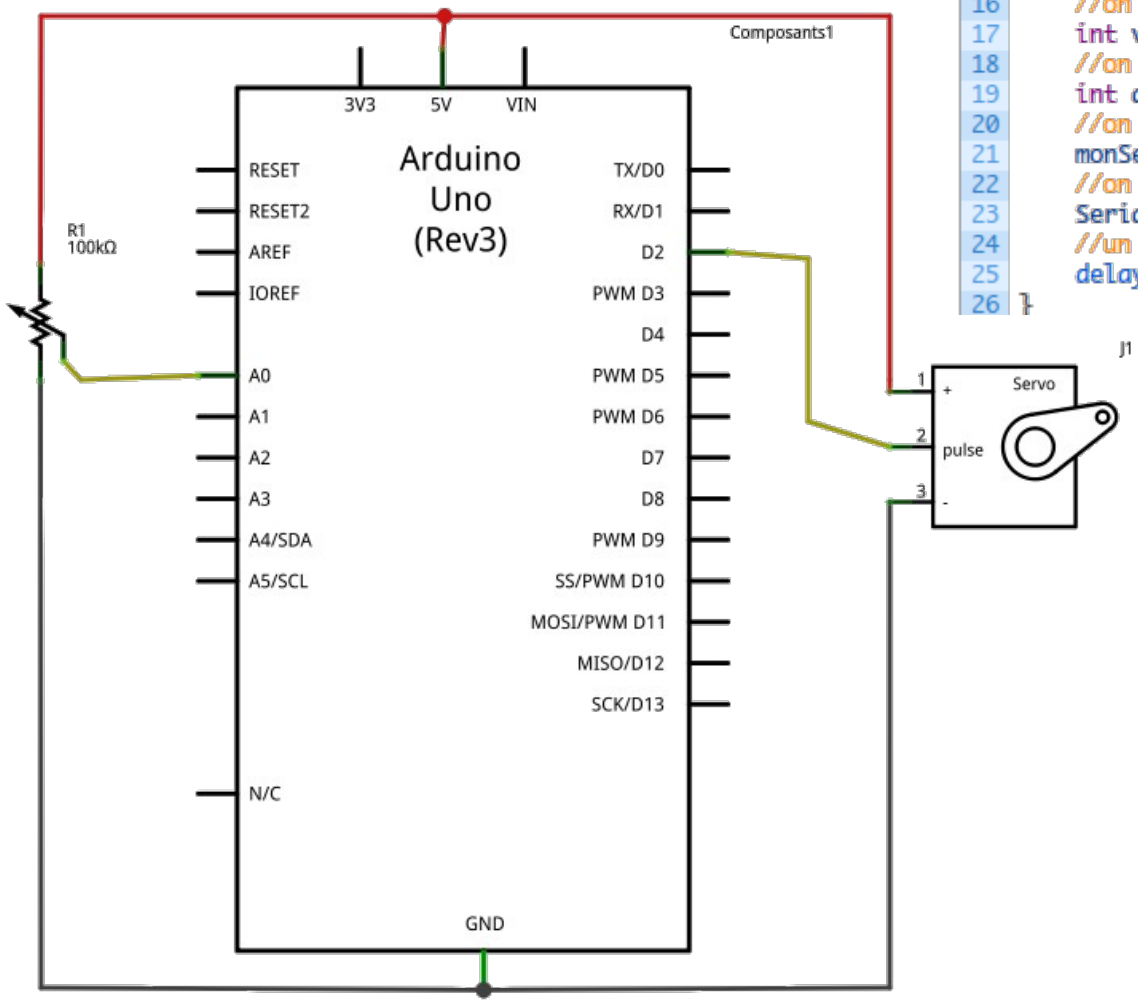
Faire varier la position du servo-moteur en agissant sur le potentiomètre :





Tutoriel inspiré de l'excellent site http://eskimon.fr

```
 #include  //On oublie pas d'ajouter la bibliothèque !

const int potar = 0; //notre potentiomètre
Servo monServo;

void setup()
{
    //on déclare le servo sur la broche 2 (éventuellement régler les bornes)
    monServo.attach(2);
    //on oublie pas de démarrer la voie série
    Serial.begin(9600);
}

void loop()
{
    //on lit la valeur du potentiomètre
    int val = analogRead(potar);
    //on convertit la valeur lue en angle compris dans l'intervalle [0;180]
    int angle = val / 5.7;
    //on met à jour l'angle sur le servo
    monServo.write(angle);
    //on renvoie l'angle par la voie série pour superviser
    Serial.println(angle);
    //un petit temps de pause
    delay(100);
}
```

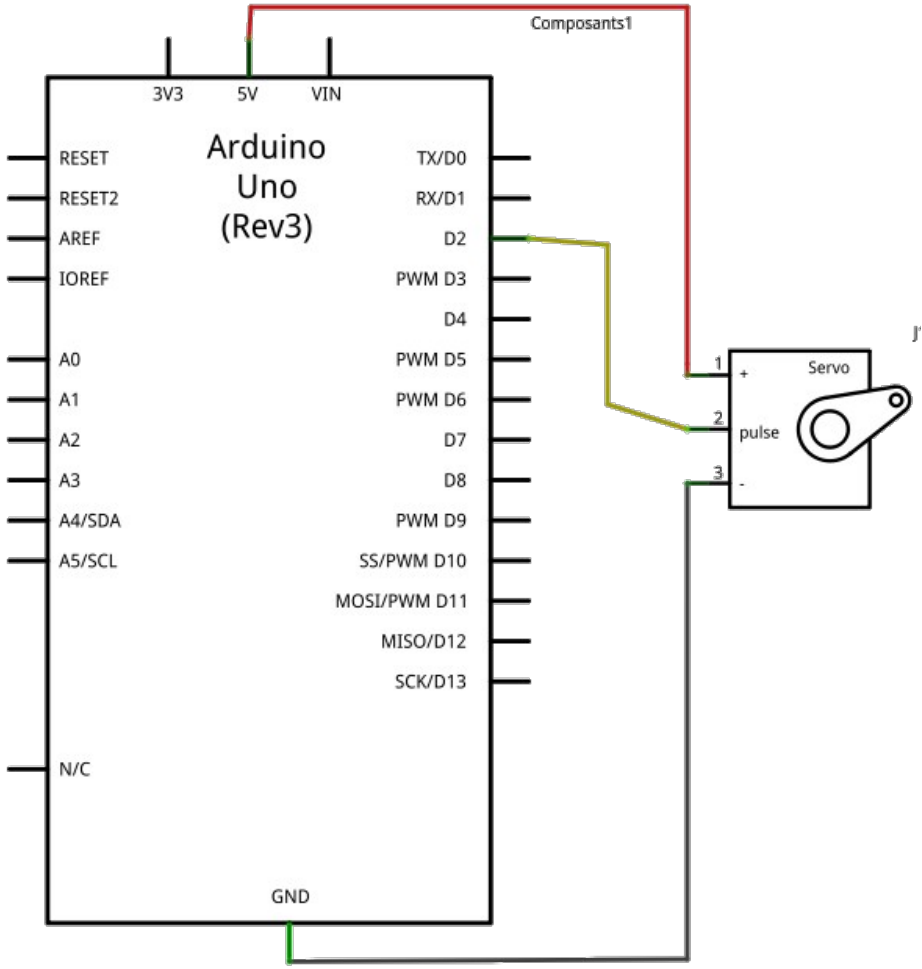


Tutoriel inspiré de l'excellent site http://eskimon.fr

Trouver les limites d'un servo-moteur non standard !

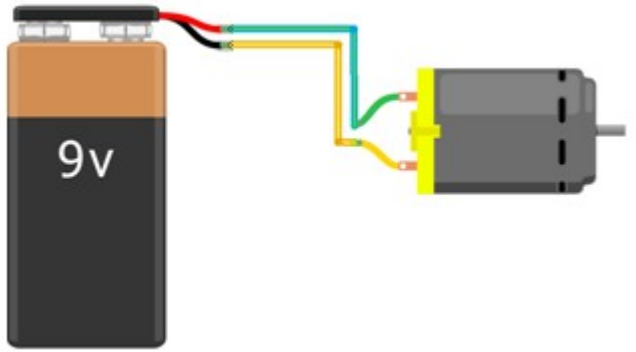
```
monServo.attach(2, 1000, 2000);
```

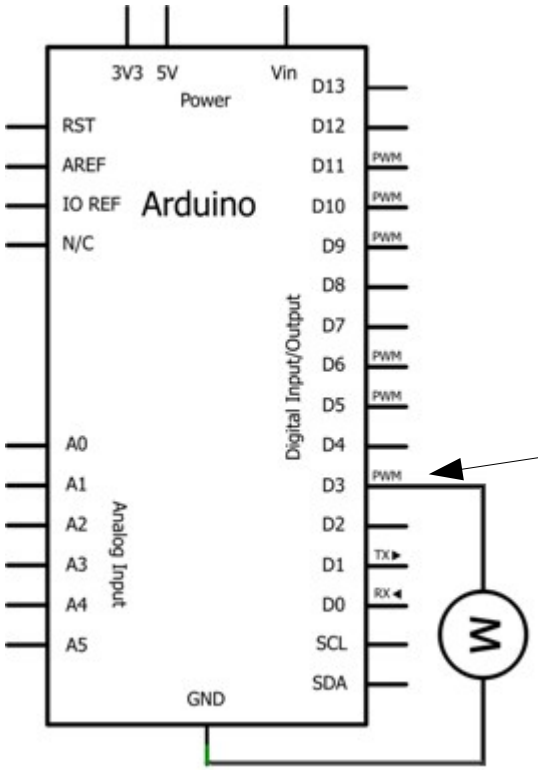
```
#include

int temps = 1500; //censé être à mi-chemin entre 1000 et 2000, un bon point de départ

Servo monServo;

void setup()
{
    Serial.begin(115200);
    Serial.println("Hello World");

    monServo.attach(2);
    //on démarre à une valeur censé être la moitié de
    //l'excursion totale de l'angle réalisé par le servomoteur
    monServo.writeMicroseconds(temps);
}

void loop()
{le
    //des données sur la liaison série ? (lorsque l'on appuie sur 'a' ou 'd')
    if(Serial.available())
    {
        char commande = Serial.read(); //on lit

        //on modifie la consigne si c'est un caractère qui nous intéresse
        if(commande == 'a')
            temps += 10;  //ajout de 10µs au temps HAUT
        else if(commande == 'd')
            temps -= 10;  //retrait de 10µs au temps HAUT

        //on modifie la consigne du servo
        monServo.writeMicroseconds(temps);

        //et on fait un retour sur la console pour savoir où on est rendu
        Serial.println(temps, DEC);
    }
}
```

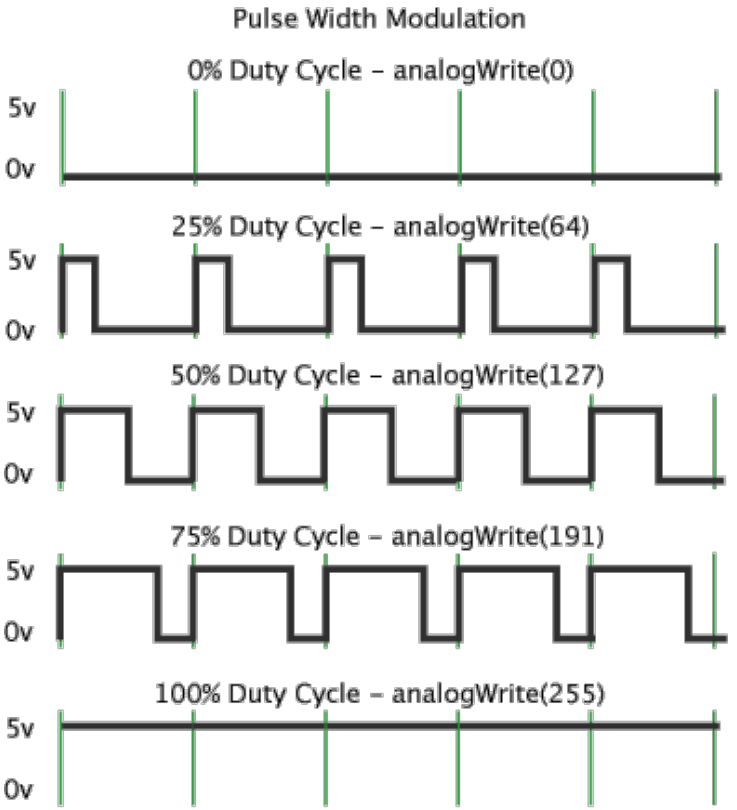


Tutoriel inspiré de l'excellent site http://eskimon.fr

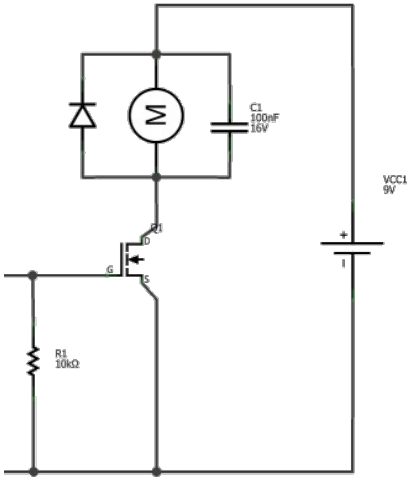
Les moteurs à courant continu avec Arduino

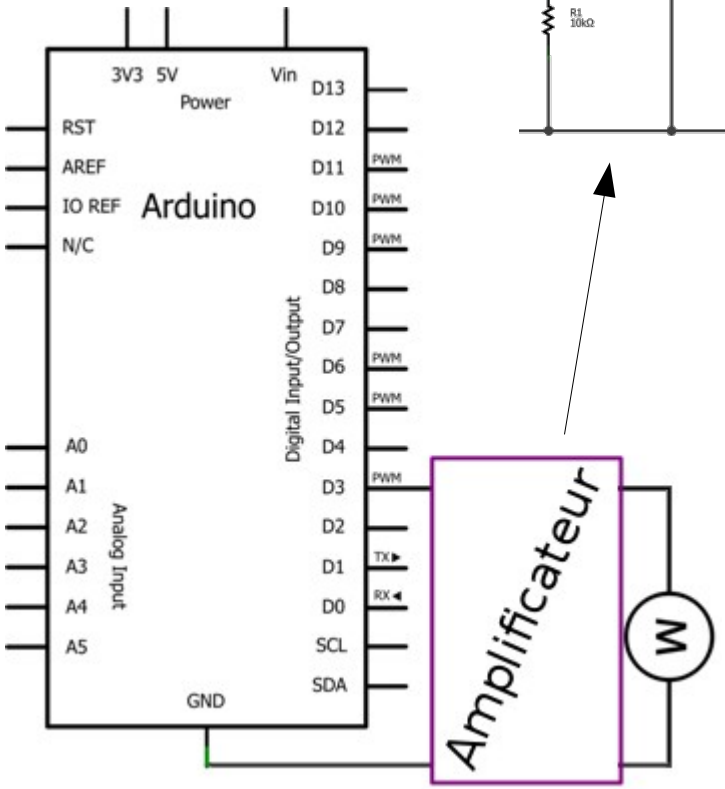


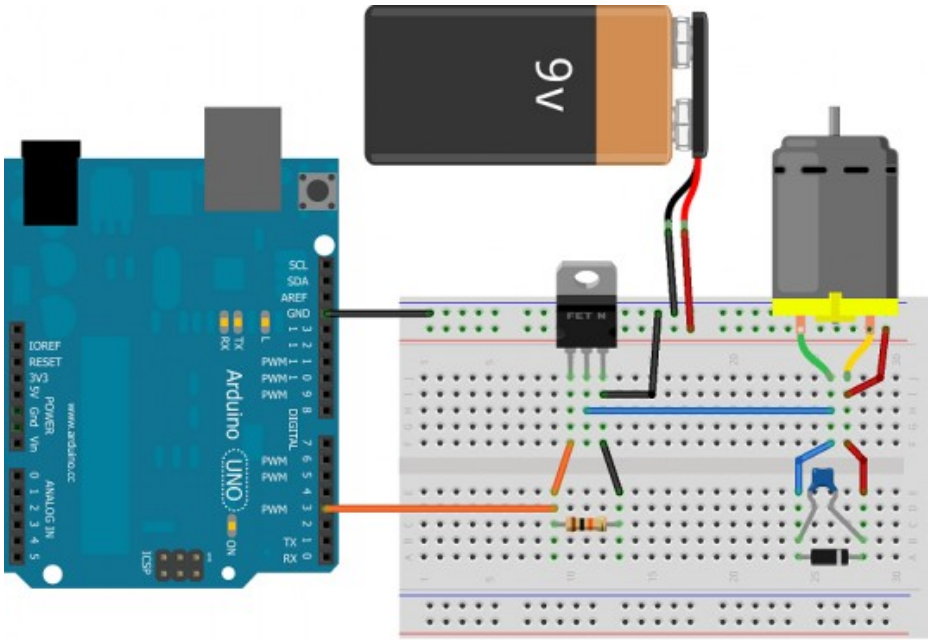




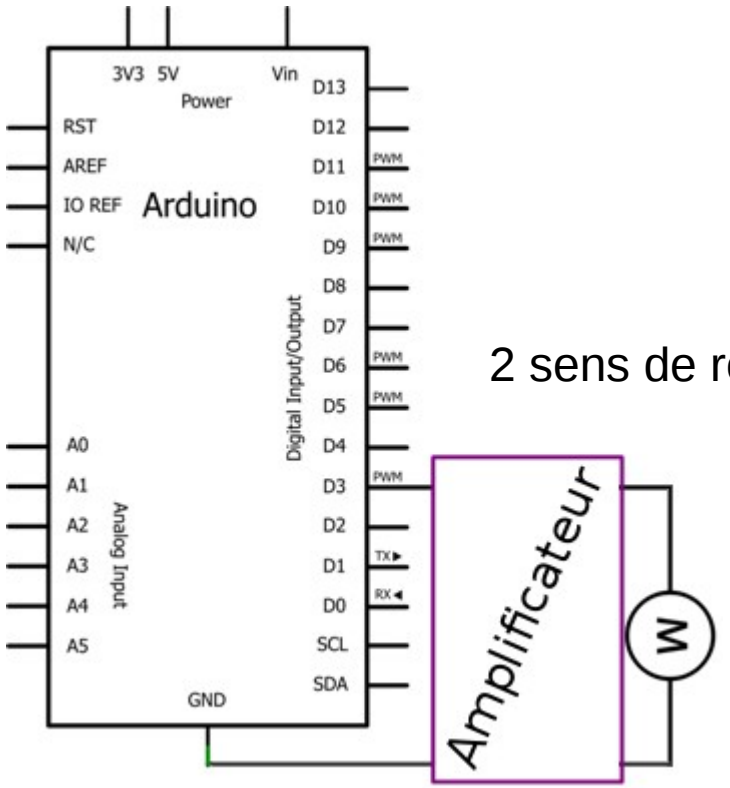
Tutoriel inspiré de l'excellent site http://eskimon.fr



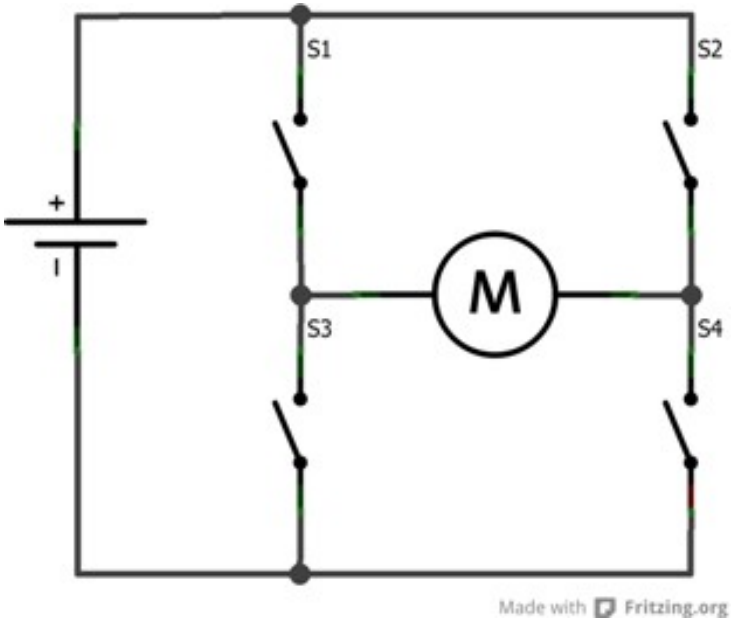


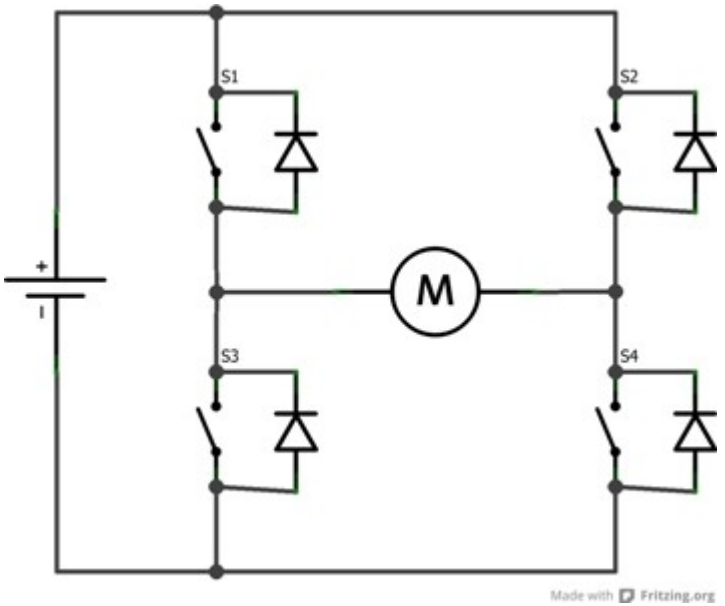


Tutoriel inspiré de l'excellent site http://eskimon.fr

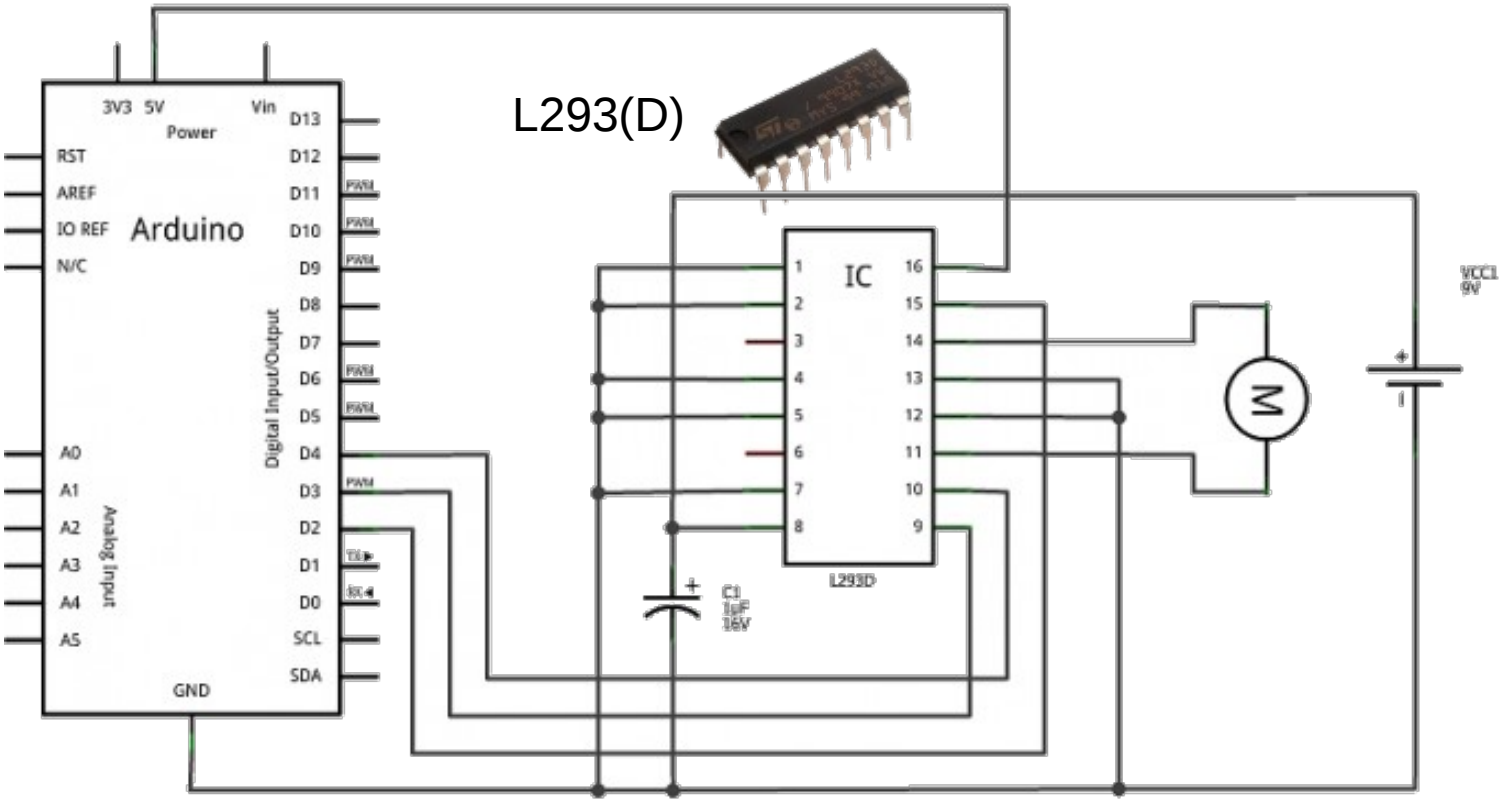


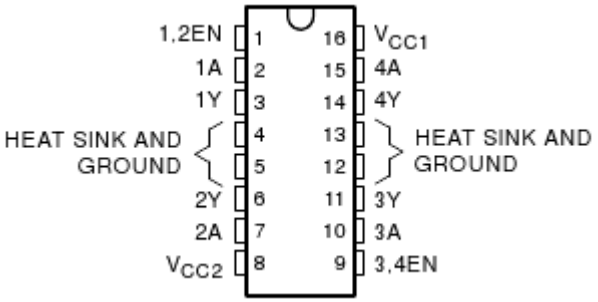
2 sens de rotation ?

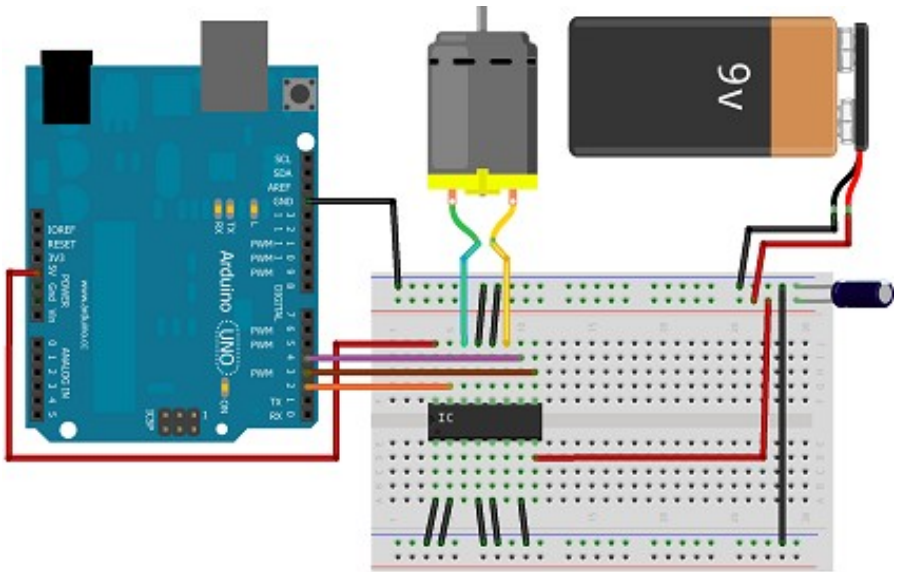




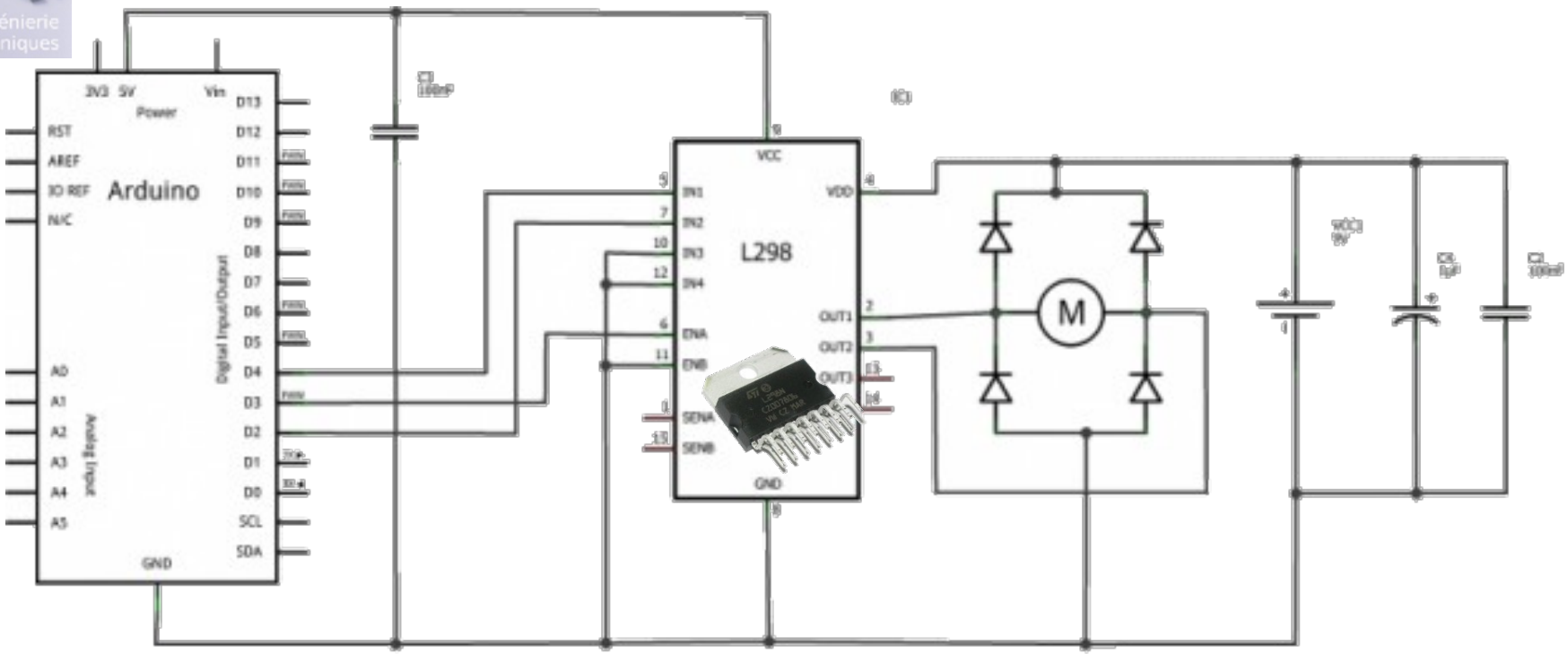
Tutoriel inspiré de l'excellent site http://eskimon.fr

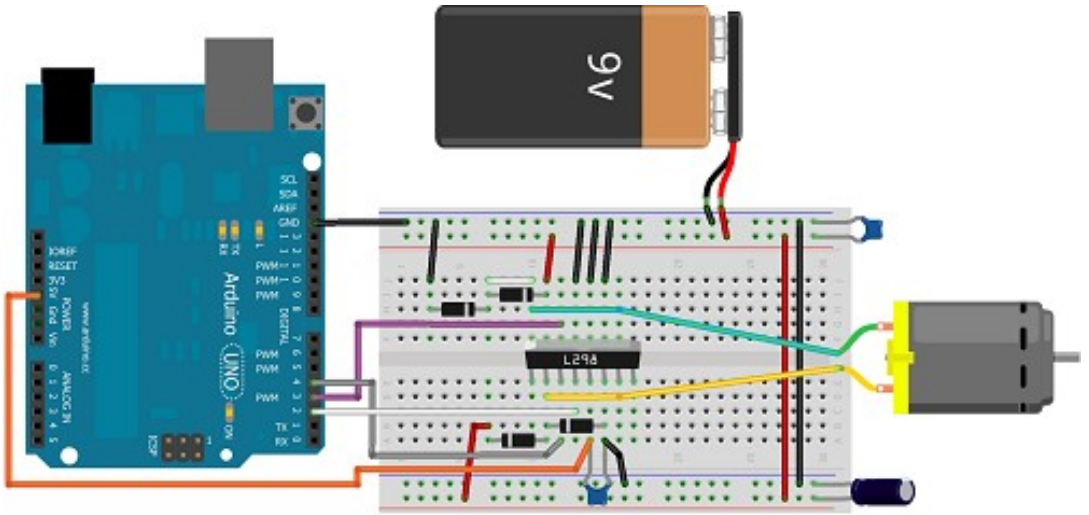






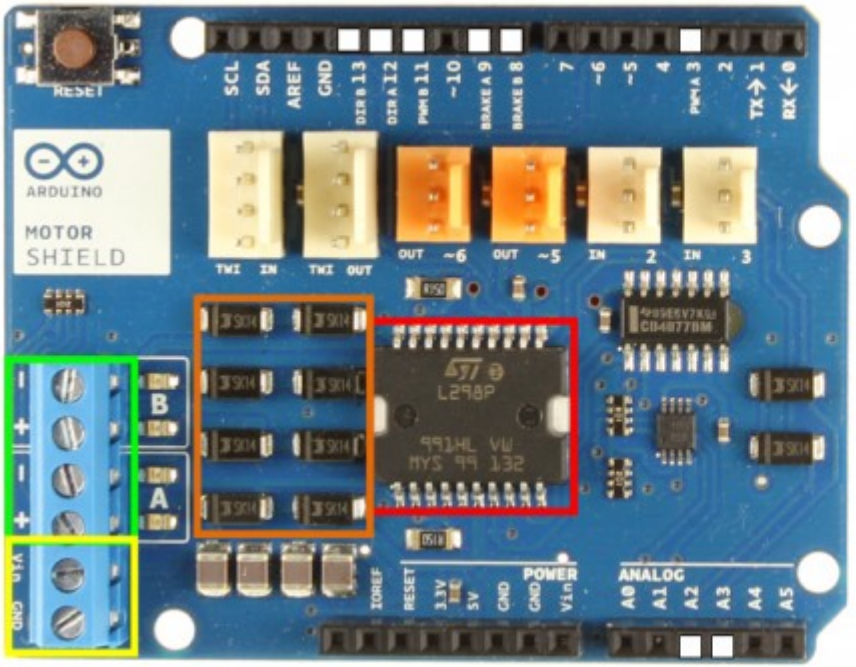
Input 1 (broche 2 et 10)	Input 2 (broche 7 et 15)	Effet
0	1	Tourne dans le sens horaire
1	0	Tourne dans le sens anti-horaire
0	0	Frein
1	1	Frein





Tutoriel inspiré de l'excellent site http://eskimon.fr

Utiliser un Shield Arduino pour piloter les moteurs à courant continu :



- Le cœur du shield : le L298
- Les diodes de roue libre (4 par moteurs)
- Les sorties pour brancher les moteurs
- L'entrée de l'alimentation du shield
- Les broches d'interface entre Arduino et le shield

A terminer...

Tutoriel inspiré de l'excellent site http://eskimon.fr