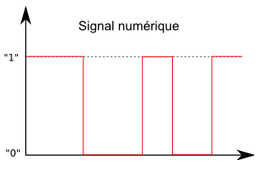
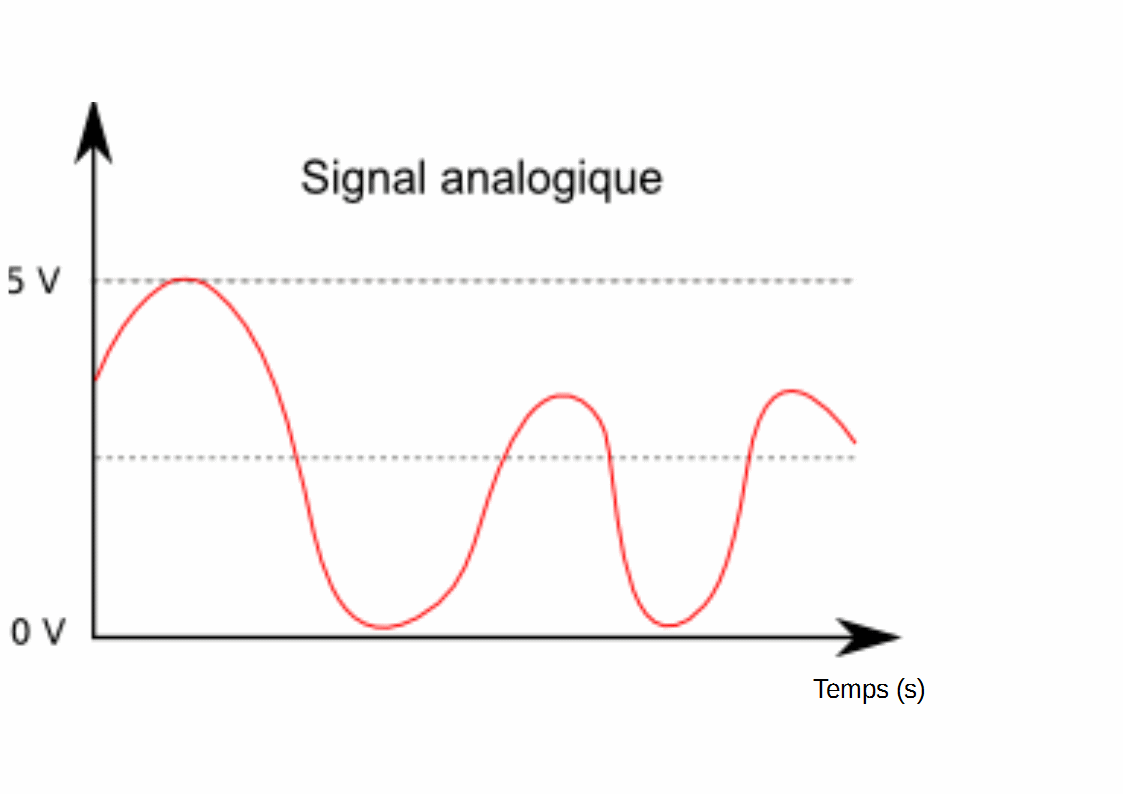
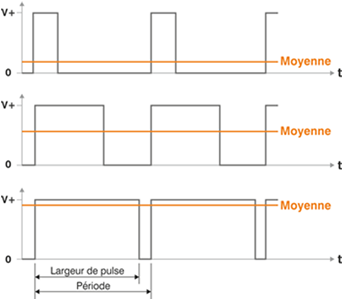
|  |  |  |
| --- | --- | --- |
| SI | Arduino Uno : Sorties analogiques | Seconde |



1. Le signal pseudo-analogiques des µC

En comparaison avec un signal logique qui a deux états possibles, un signal analogique en a une infinité (voir graphique ci-contre).

Le microcontrôleur traite uniquement des données numériques basées sur des signaux logiques. Il lui est donc impossible de générer directement un signal analogique (compris entre 0 et 5V) sur ses broches. C’est pour cela que l’on utilise un artifice, un signal pseudo-analogique crée par la technique de la Modulation de Largeur d’Impulsion (MLI ou PWM en anglais).

1. Le signal MLI

Si on fait varier une sortie numérique d’un µC de l’état haut (5V) à l’état bas (0V) très rapidement (500 à 1000 fois par seconde), la valeur moyenne de la tension sur une durée longue (1 seconde par exemple) sera inférieure à 5V.

En jouant sur la durée des états hauts et états bas (signal à motif en créneau) on modifie la valeur de cette tension moyenne à volonté (voir ci-contre).

C’est de cette façon que l’on génère un signal pseudo-analogique compris entre 0 et 5 V sur les µC.

La génération de ce signal MLI permet notamment de faire varier :

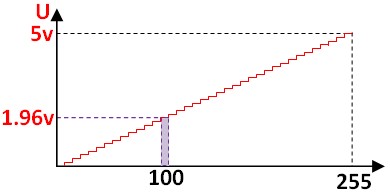
* La fréquence de rotation d’un moteur électrique à courant continu via une carte de puissance
* La fréquence des sons d’un buzzer (ou un piézoélectrique)
* La luminosité d’une LED

1. Sortie analogique sur la carte Arduino Uno

Sur la carte Arduino Uno, seuls les ports numériques précédés d’un tilde ’’ ~ ’’ sont capables de générer un signal MLI, cela concerne les broches 3, 5, 6, 9, 10 et 11.

Le signal MLI de la carte Arduino Uno est caractérisé par :

* La fréquence des créneaux : de 500 Hz pour les broches 3, 5, 10 et 11 à 1000 Hz pour les broches 5 et 6
* L’amplitude des créneaux : 5V
* La profondeur du convertisseur numérique/analogique (CNA) : 8 bits

Le nombre de bits du convertisseur numérique/analogique (CNA) correspond au nombre de valeurs pseudo-analogiques possibles. Un CNA de 8 bits permet une commande sur 256 valeurs (28), dans notre cas de 0 à 255 en décimal.

Conséquence : cela scinde la tension moyenne de ’’5V’’ de sortie en 256 pallier donc 255 valeurs (28-1).

A partir de ces informations on peut calculer, en fonction des besoins :

* Le quantum, qui correspond à la plus petite variation de tension du CNA (pour un bit) :
* La tension moyenne de sortie et le rapport cyclique en fonction de la valeur de commande.

Complétez le tableau ci-dessous sachant que le rapport cyclique et la tension moyenne de sortie sont proportionnels à la valeur de commande.

|  |  |  |
| --- | --- | --- |
| Valeur de commande (de 0 à 255) | Rapport cyclique (ton / T) | Tension moyenne de sortie |
| 0 | 0 % | 0 V |
| 1 |  |  |
| 64 |  |  |
|  | 50 % |  |
| 192 |  |  |
|  |  | 4,3 V |
| 255 | 100 % | 5 V |

Remarque : la valeur de commande doit être une variable de type entier notée ’’int’’ en programmation (integer).

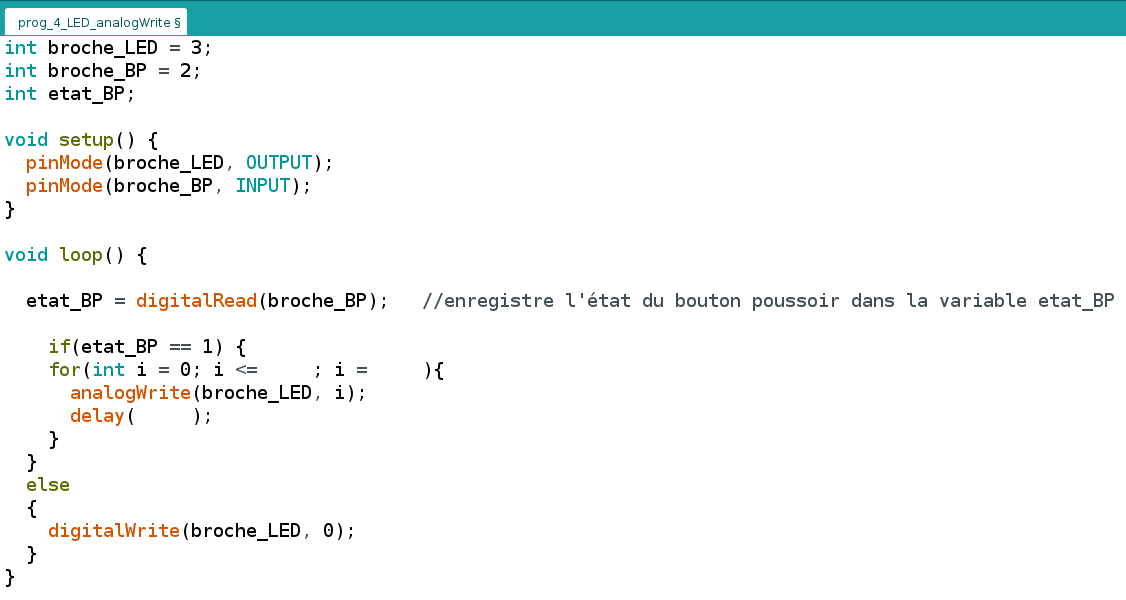
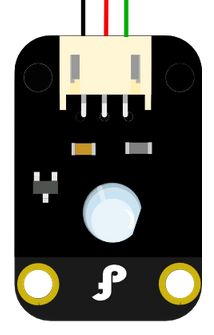
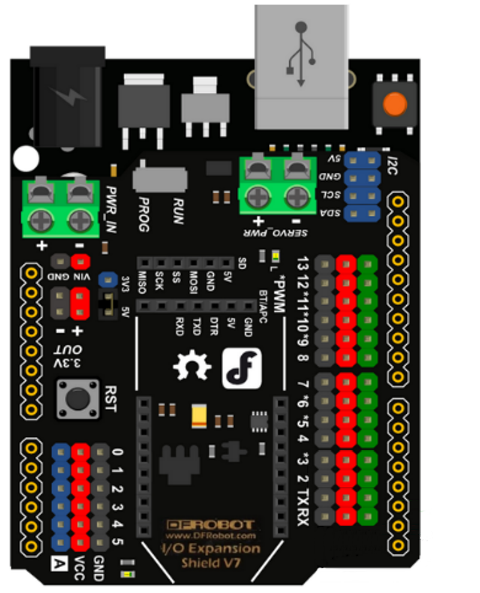
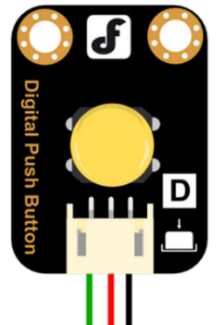
1. Programmation d’un MLI sous IDE Arduino



Du coté programmation, le signal MLI (ou PWM) est généré par la fonction :

A partir du montage du cours « Entrées numériques » et du code à compléter ci-dessous (voir fichier prog\_4\_LED\_analogWrite), réalisez un programme permettant d’allumer progressivement la led par paliers de 5 toutes les 200 ms sur la commande du MLI après pression du bouton poussoir.

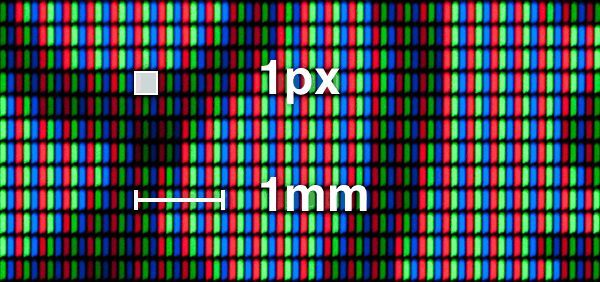
Evaluez le nombre de paliers puis le temps d’allumage de la led (et le vérifier).



1. La LED RGB (jouons un peu !-)
   1. Battements de cœur

Reprendre le programme précédent et ajouter une extinction progressive de la LED à la suite de la boucle « for ».

* 1. Afficher une couleur spécifique

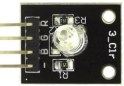
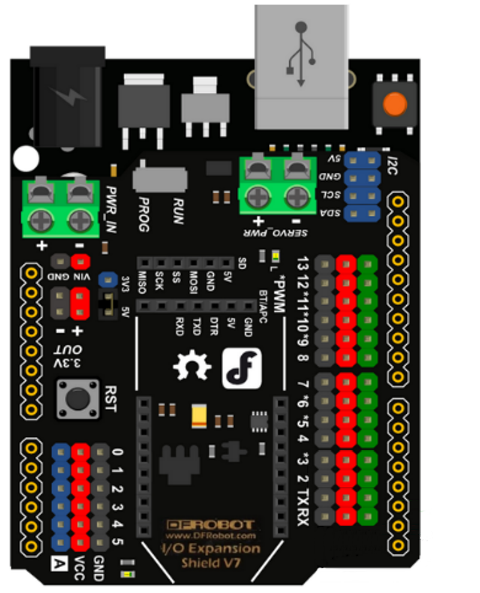
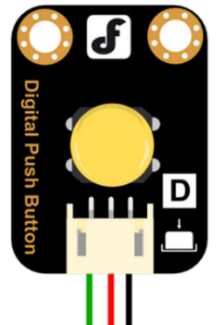
La LED RGB est constituée de 3 LED commandables indépendamment. De ce fait le nombre de variations de signaux donc de nuances de couleurs possibles sur 3 MLI est de 2553 = 16581375.

Remarque : votre écran d’ordinateur est constitué de pixels. Chaque pixel est une LED RGB miniature commandée en 3 x 8 bits.

A partir du montage du cours « LED RGB Tableau et boucle FOR » et du code à compléter ci-dessous (voir fichier prog\_5\_RGB\_analogWrite), réalisez un programme permettant d’afficher une des couleurs du pantone RVB du site suivant (lien à copier à partir de l’en tête du programme) : <https://www.toutes-les-couleurs.com/code-couleur-rvb.php>

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement



* 1. Séquence lumineuse

Réaliser une séquence lumineuse de 10 couleurs différentes sur une durée de 10 secondes en vous inspirant du programme ci-dessous (voir fichier prog\_6\_RGB\_analogWrite\_seq) :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement