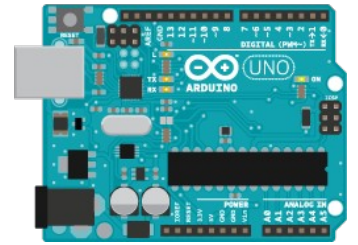
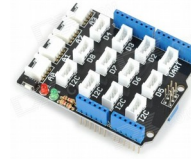


Programmer une carte Arduino avec mBlock



mBlock est un environnement de programmation graphique basée sur Scratch 2.0

NB : Un Shield des modules connexions GROVE sont utilisés ici.



1 – Téléchargement, Installation, Configuration

a) Téléchargement et Installation

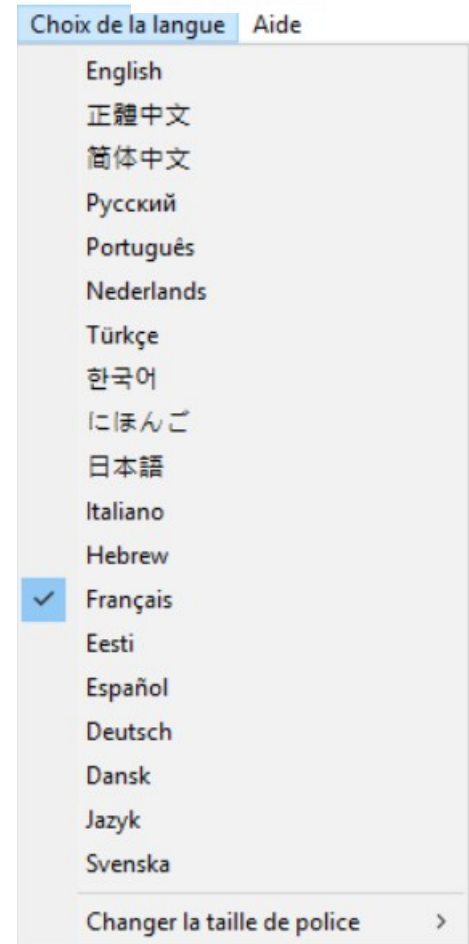
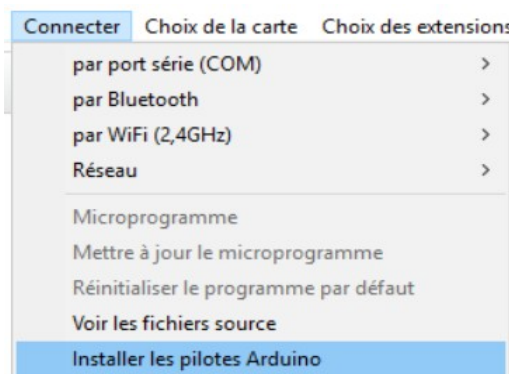
Télécharger le logiciel ici : <http://www.mblock.cc/download/>.

Puis installer sur c :

b) Configuration

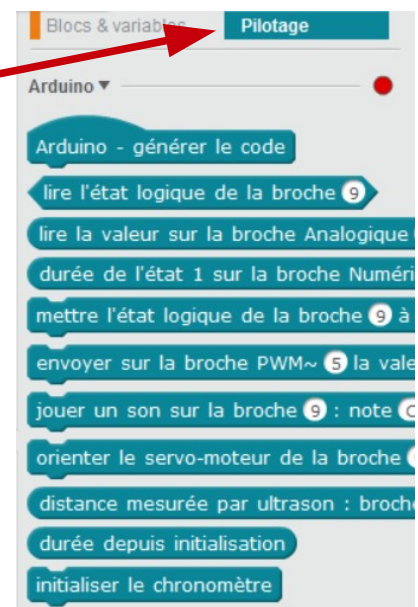
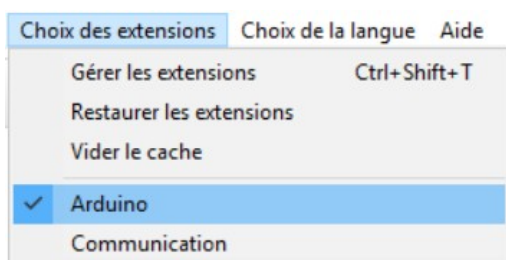
- choisir la langue : **Choix de la langue** > Français
- installer les pilotes Arduino pour reconnaître la carte :

Connecter > Installer les pilotes Arduino



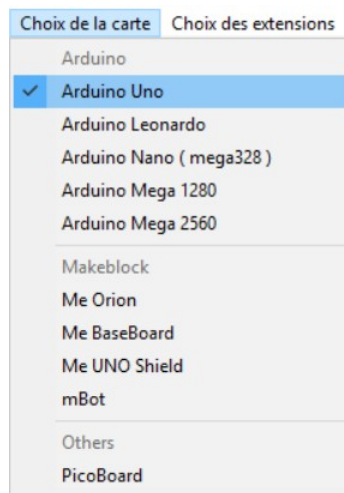
- choisir l'extension Arduino : **Choix des extensions** > Arduino

Les blocs spécifique Arduino sont accessible sous la rubrique
« Pilotage »

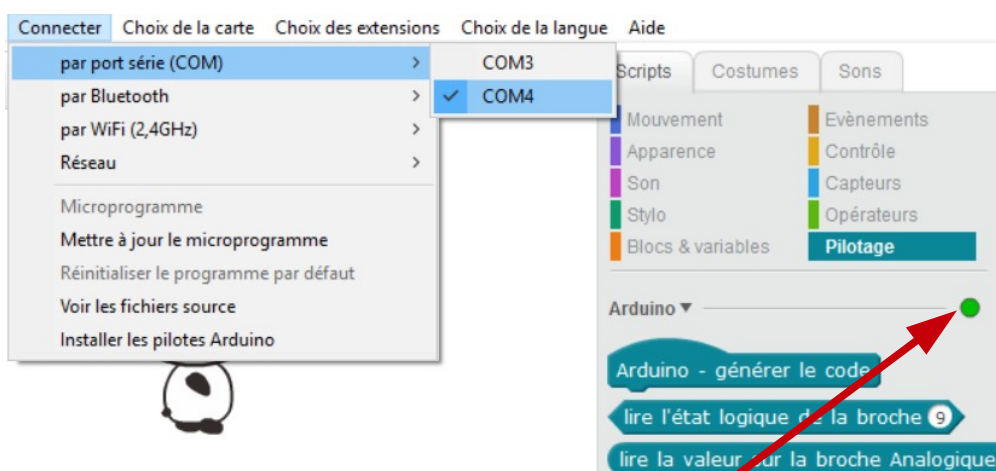


- connecter la carte :

Choix de la carte > Arduino Uno

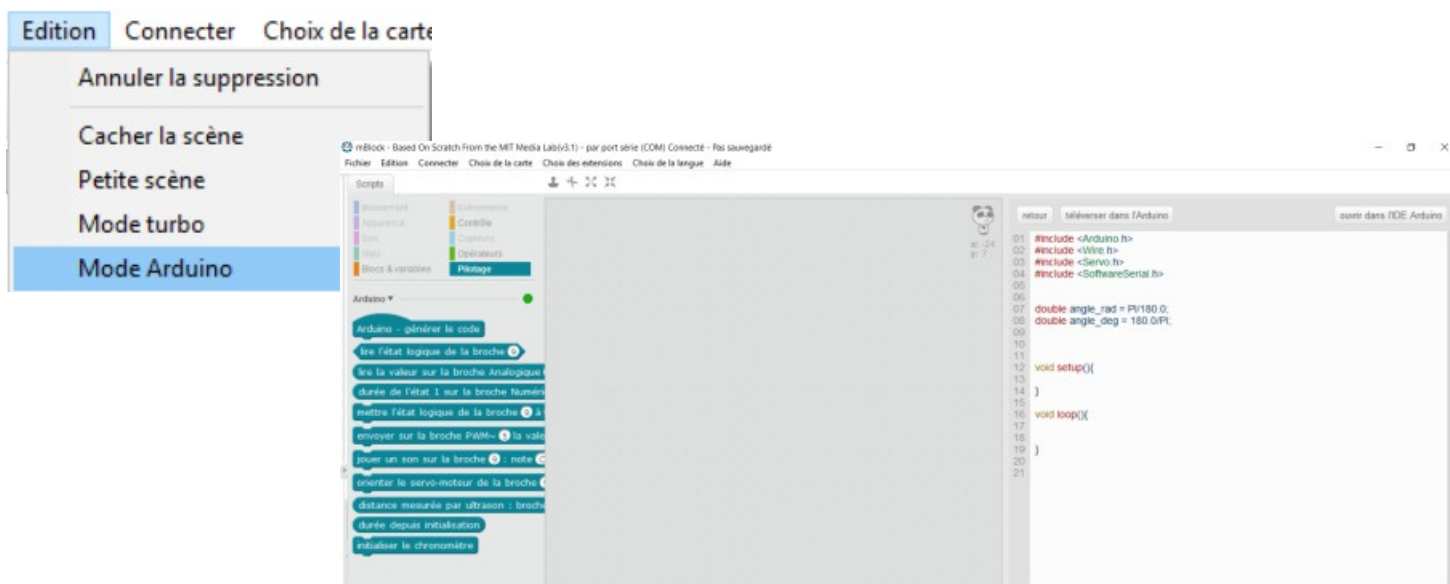


puis Connecter > par port série (COM) > COM x



Si la carte est bien reconnue et connectée, le point rouge passe au vert !

- passer en mode Arduino : Edition > Mode Arduino



2 – Programmation des Capteurs (en entrées)

a) Capteurs sur entrées Numériques

Bouton poussoir, Touch, Capteur de lumière, Détecteur de présence, Suiveur de ligne, ...



- Ajouter les blocs pour obtenir ce programme :

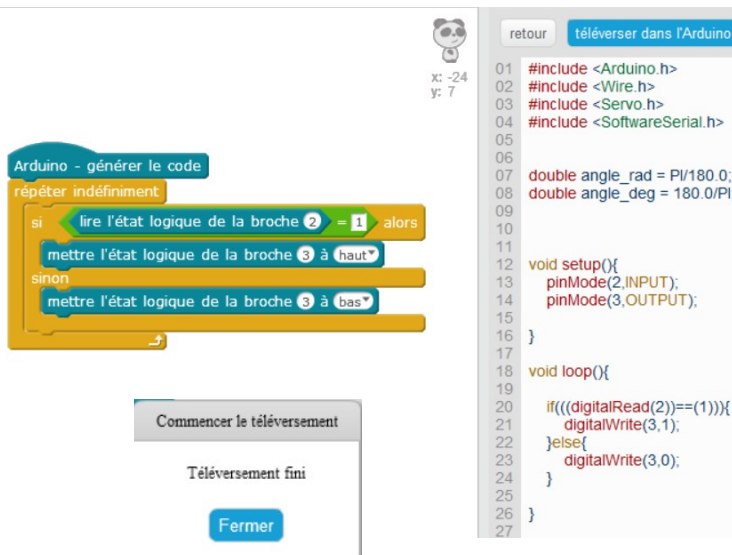


Test sur le port D2 Logique
Si D2 activé ou = « 1 » logique
Alors ... Sinon ...

SI OUI ALORS activer la Sortie D3

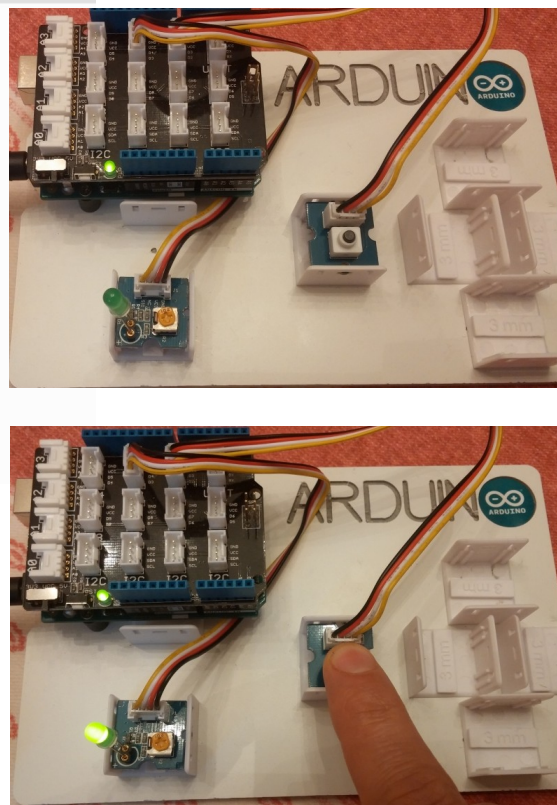
SINON ne pas activer la Sortie D3

- Transférer le programme dans la carte Arduino :



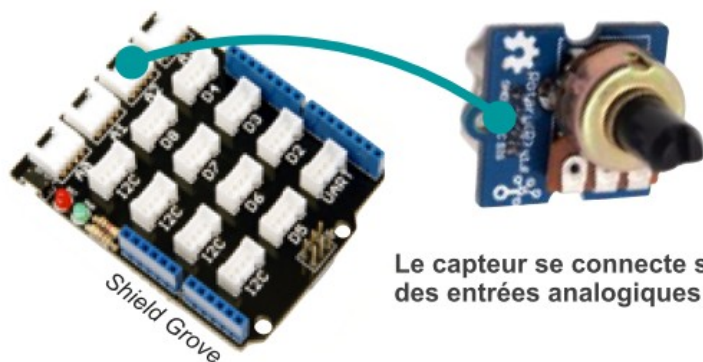
Ce programme permet d'allumer une DEL si le bouton poussoir est activé. Sinon, la DEL reste éteinte.

La DEL est branchée sur le port D3 et le bouton poussoir sur le port D2.



b) Capteurs sur entrées Analogiques

Potentiomètre, Joystick, Capteur de Température, ...



Le capteur se connecte sur l'une des entrées analogiques de A0 à A3

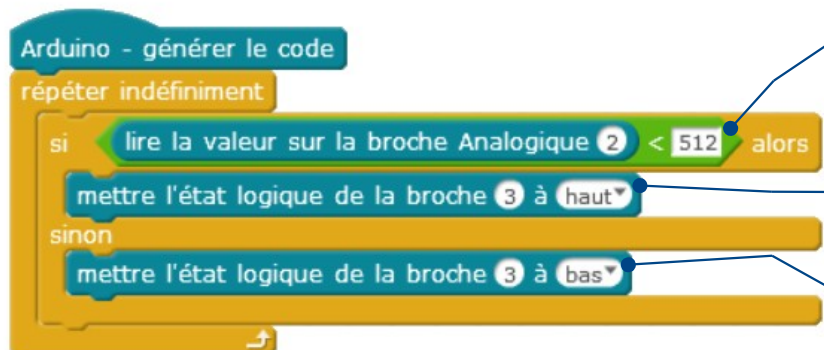


Les entrées analogiques convertissent la valeur d'entrée (une tension entre 0 et 5V) en valeur numérique sur 10 bits.

Soit 1024 valeurs possibles de 0 à 1023.

Décimal	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0
...
1023	1	1	1	1	1	1	1	1	1	1

- Ajouter les blocs pour obtenir ce programme :



Test sur le port A2 analogique
Si la valeur codée est inférieure à 512
Alors ... Sinon ...

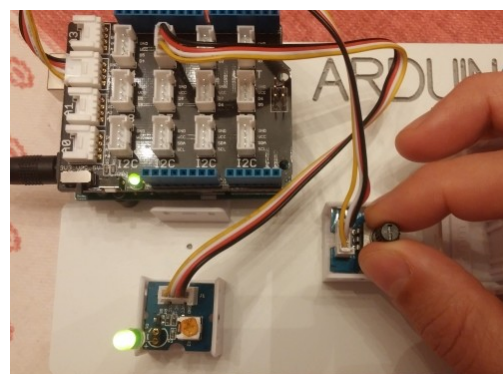
SI OUI ALORS activer la Sortie D3

SINON ne pas activer la Sortie D3

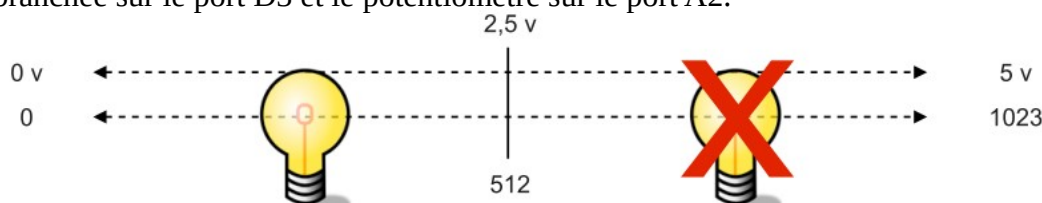
- Transférer le programme dans la carte Arduino :



Ce programme permet d'allumer une del si le capteur est dans la première moitié de sa plage d'utilisation, c'est à dire entre 0 et 2,5 volts, soit entre 0 et 512 en numérique.



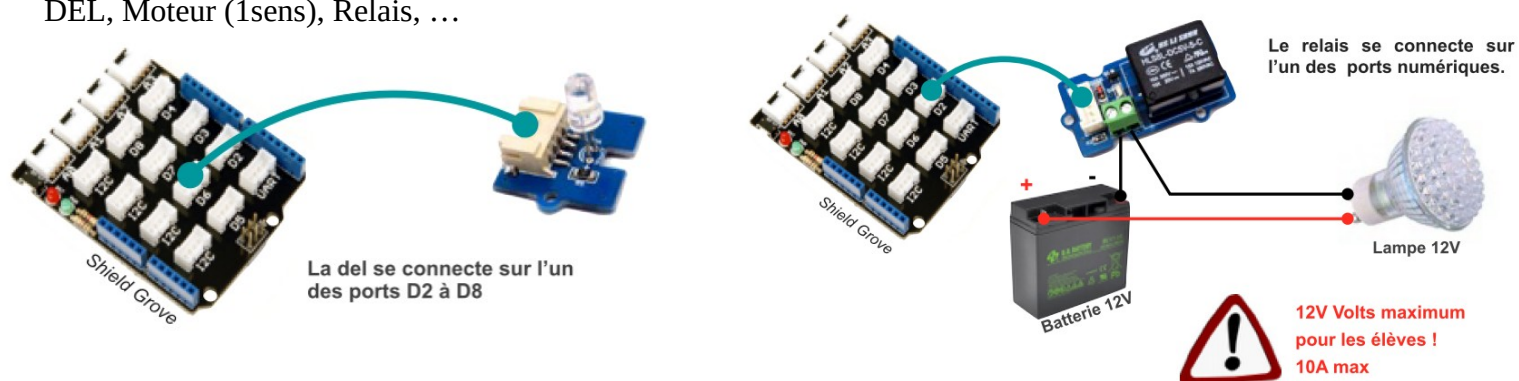
La DEL est branchée sur le port D3 et le potentiomètre sur le port A2.



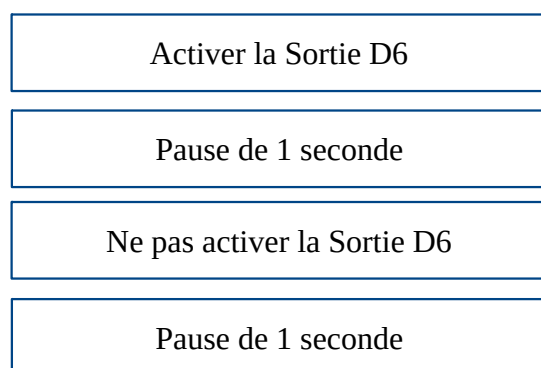
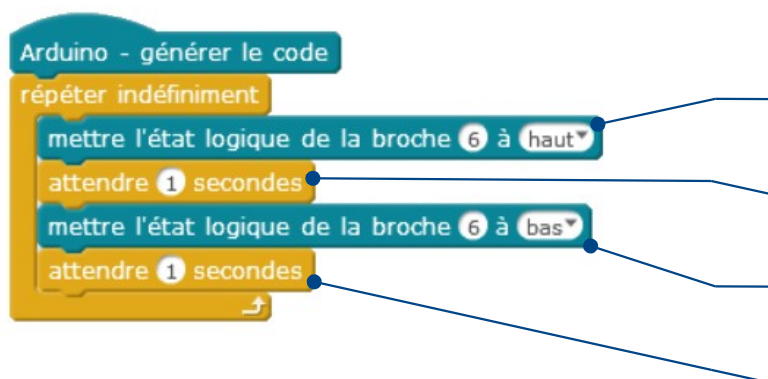
3 – Programmation des Actionneurs (en sortie)

a) Actionneurs sur sorties Numériques

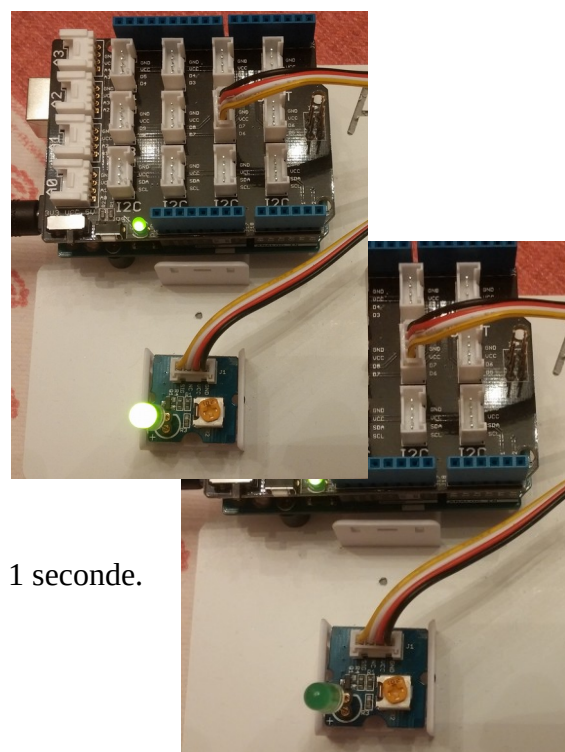
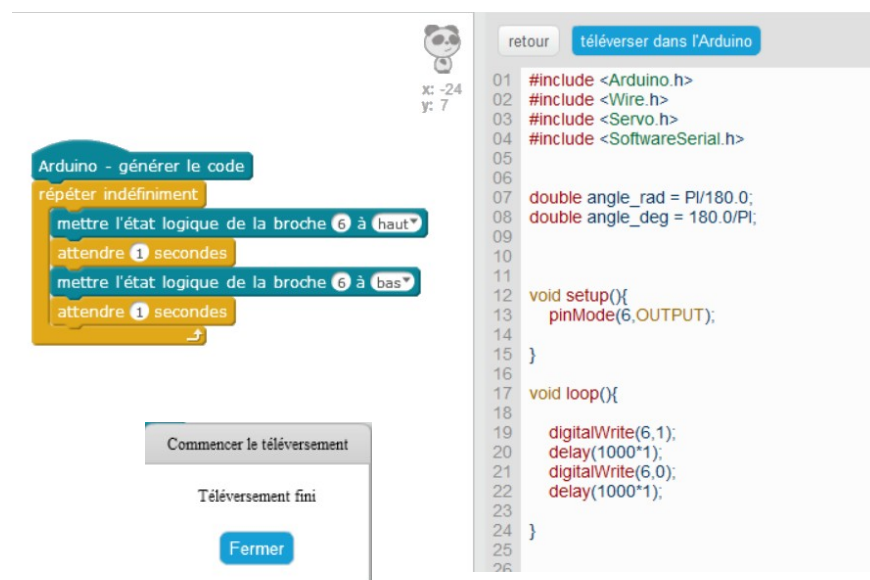
DEL, Moteur (1sens), Relais, ...



- Ajouter les blocs pour obtenir ce programme :



- Transférer le programme dans la carte Arduino :



Ce programme permet d'allumer une DEL alternativement toutes les 1 seconde.

La DEL est branchée sur le port D6.

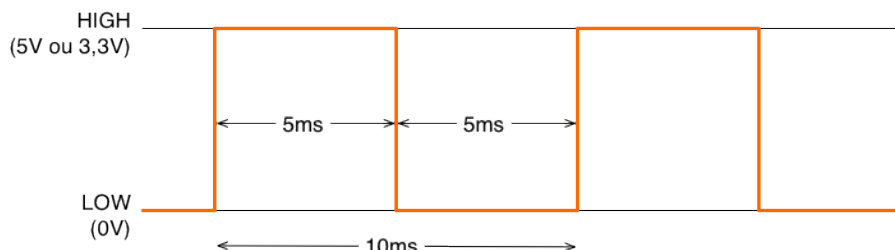
b) Sorties PWM

- Une sortie PWM sur une carte Arduino est une sortie Numérique dont les signaux ont toujours une valeur LOW ou HIGH. Mais le principe est de construire un signal qui est alternativement LOW et HIGH et de répéter très vite cette alternance en faisant varier la fréquence du signal.

Dans le cas d'une **DEL**, elle est alternativement allumée et éteinte mais le cycle est tellement rapide que la persistance rétinienne nous donne l'illusion d'une DEL allumée en permanence. Prenons par exemple une période de 10ms, soit une fréquence de 100Hz. Si la DEL est allumée pendant 5ms et éteinte pendant 5ms, comme sur la figure ci-dessous, l'impression sera une **luminosité** de 50% de la luminosité maximum.

PMW à 50%

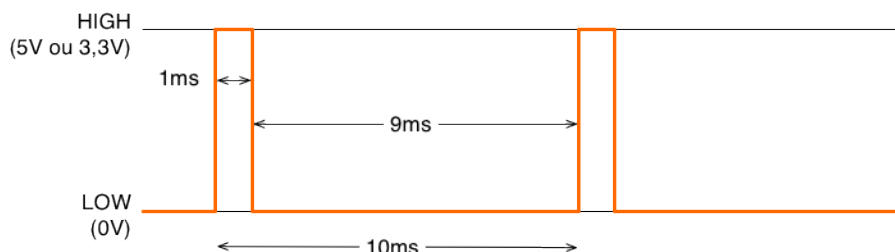
La fréquence est de 100Hz, le rapport cyclique de 50%



Si la DEL est allumée pendant 1ms et éteinte pendant 9ms, l'impression sera une luminosité de 10% comme sur la figure ci-dessous.

PWM à 10%

La fréquence est de 100Hz et le rapport cyclique de 10%.



Le pourcentage de temps passé à l'état HIGH sur la période du signal est appelé le *rapport cyclique*. Il varie donc de 0%, le signal est tout le temps LOW, à 100%, le signal est tout le temps HIGH.

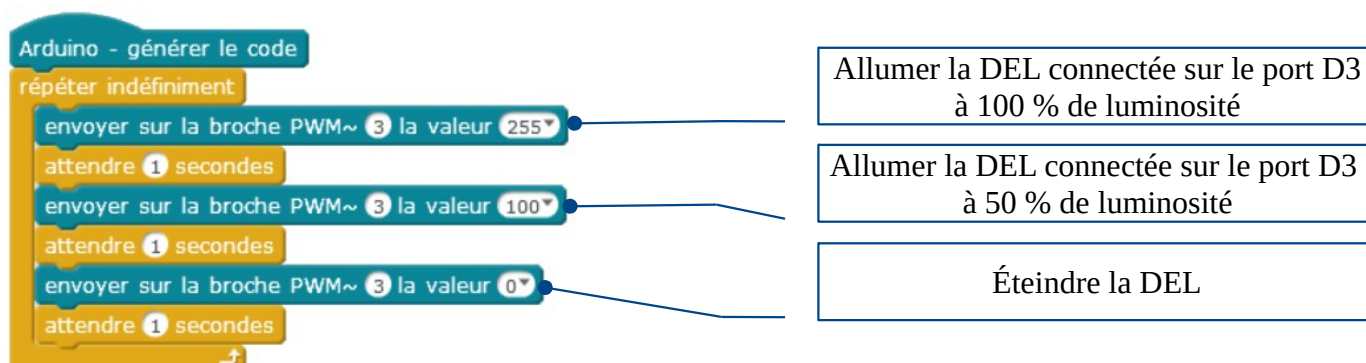
Dans le cas d'un **moteur**, c'est la **vitesse de rotation** qui sera variable en fonction de la fréquence paramétrée...

- Sortie PWM sur une Arduino UNO : **3, 5, 6, 9, 10 et 11**

- Tableau de correspondance :

Rapport Cyclique %	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
Valeur sur 8 bits	0	13	26	38	51	64	77	89	102	115	128	140	153	166	179	191	204	217	230	242	255

- Programme pour une DEL qui s'éclaire à différents niveaux de luminosité :

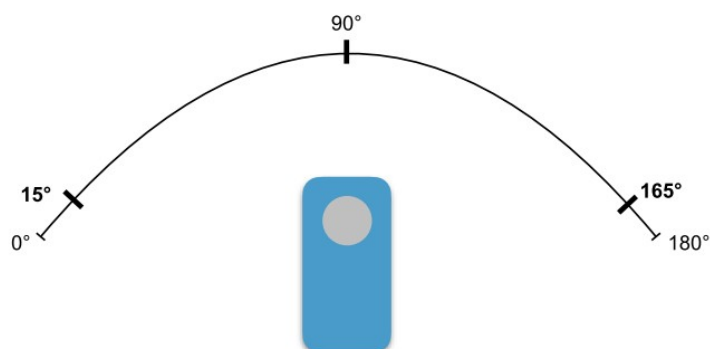


- Variation de l'intensité lumineuse d'une DEL à partir de la valeur d'un potentiomètre :



Comme déjà vu au 2.b, la valeur en entrée du potentiomètre est comprise entre 0 et 1024 soit 4 fois plus que valeur attendue pour le rapport cyclique en sortie PWM (d'où la division par 4).

c) Le servomoteur



Amplitude angulaire théorique : 0° à 180°
Amplitude angulaire réelle : 15° à 165°

Ajouter les blocs pour obtenir ce programme :



Mettre le servomoteur connecté sur le port D7 en position 0°

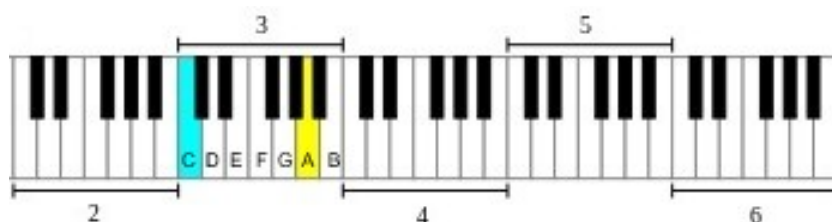
Mettre le servomoteur connecté sur le port D7 en position 90°

Mettre le servomoteur connecté sur le port D7 en position 180°

d) Le buzzer

Pour émettre un son, le buzzer a besoin de 3 paramètres :

- le port de connexion
- la note : C2, D2, ... (voir correspondance ci-contre)
- la durée de la note (1/8, 1/4, 1/2, 1, 2 temps)

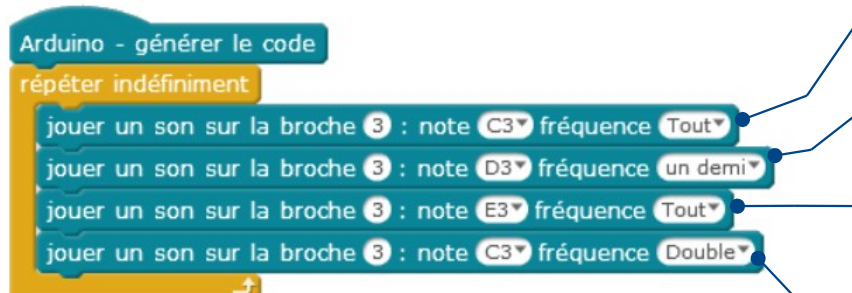


Jouer sur le buzzer connecté sur le port D3 un DO sur 1 temps

Jouer sur le buzzer connecté sur le port D3 un RÉ sur 1/2 temps

Jouer sur le buzzer connecté sur le port D3 un MI sur 1 temps

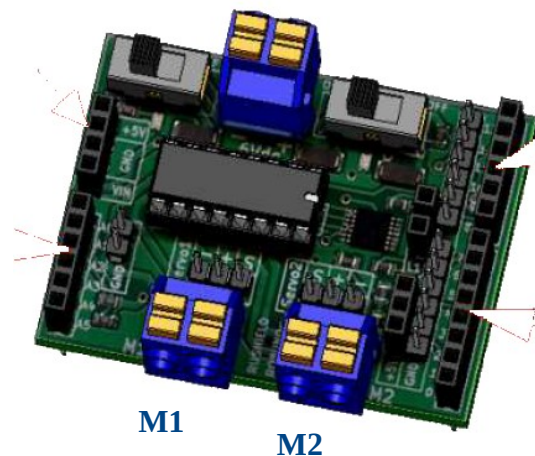
Jouer sur le buzzer connecté sur le port D3 un DO sur 2 temps



e) 2 Moteurs – 2 Sens

Pour piloter 2 moteurs dans le 2 sens, un shield supplémentaire est nécessaire. Il s'agit ici du RUSHIELD (vendu par [Technologie Services](#))

	Sortie 11	Sortie 9 PWM
Moteur 1 (M1)	0 = marche arrière	0 = arrêt
	1 = marche avant	255 = vitesse max
	Sortie 12	Sortie 10 PWM
Moteur 2 (M2)	0 = marche arrière	0 = arrêt
	1 = marche avant	255 = vitesse max



Ajouter les blocs pour obtenir ce programme :

```

Arduino - générer le code
répéter indéfiniment
  envoyer sur la broche PWM~ 9 la valeur 255
  envoyer sur la broche PWM~ 10 la valeur 255
  mettre l'état logique de la broche 11 à haut
  mettre l'état logique de la broche 12 à haut
  attendre 5 secondes
  envoyer sur la broche PWM~ 9 la valeur 255
  envoyer sur la broche PWM~ 10 la valeur 0
  mettre l'état logique de la broche 11 à haut
  mettre l'état logique de la broche 12 à bas
  attendre 1 secondes
  envoyer sur la broche PWM~ 9 la valeur 255
  envoyer sur la broche PWM~ 10 la valeur 255
  mettre l'état logique de la broche 11 à bas
  mettre l'état logique de la broche 12 à bas
  attendre 3 secondes
  envoyer sur la broche PWM~ 9 la valeur 0
  envoyer sur la broche PWM~ 10 la valeur 0
  
```

M1 et M2 vitesse maximum
M1 et M2 marche avant
Attendre 5 secondes

Le robot avance pendant 5 secondes

Le robot tourne à droite pendant 1 seconde

Le robot recule pendant 3 secondes

Le robot s'arrête

