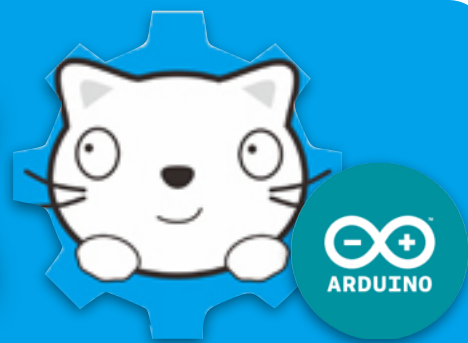


mBLOCK



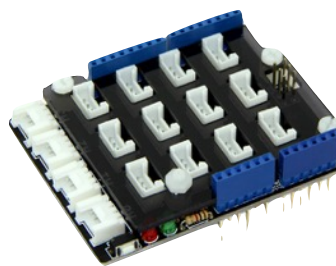
mBlock est un environnement de programmation graphique basée sur Scratch 2.0



+



=



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

TÉLÉCHARGEMENT, INSTALLATION ET CONFIGURATION



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

1

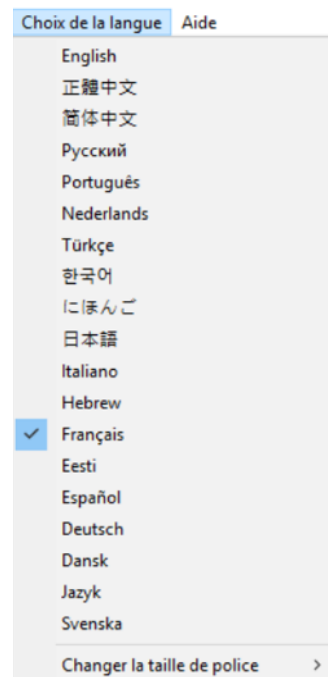
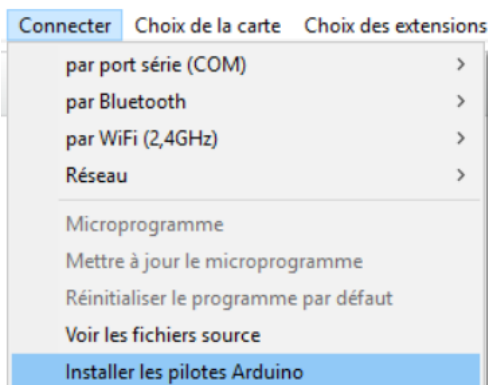
Puis installer sur c :

2

Choisir la langue : Choix de la langue > Français

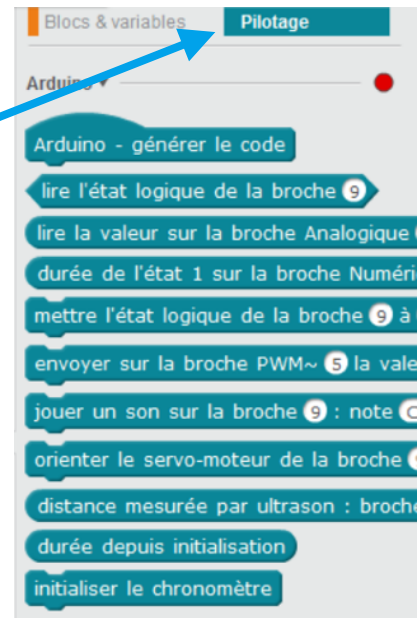
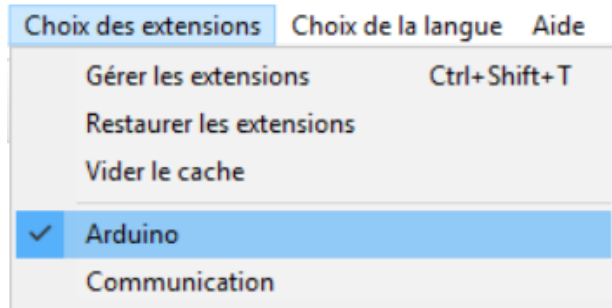
Installer les pilotes Arduino pour reconnaître la carte :

Connecter > Installer les pilotes Arduino

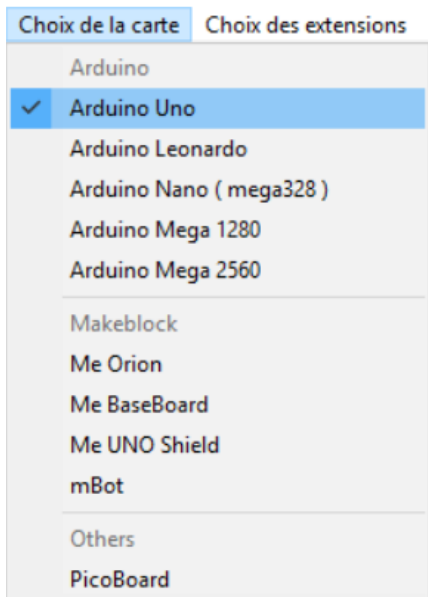


Choisir l'extension Arduino :
Choix des extensions > Arduino

Les blocs spécifiques Arduino sont accessibles
sous la rubrique « Pilotage »

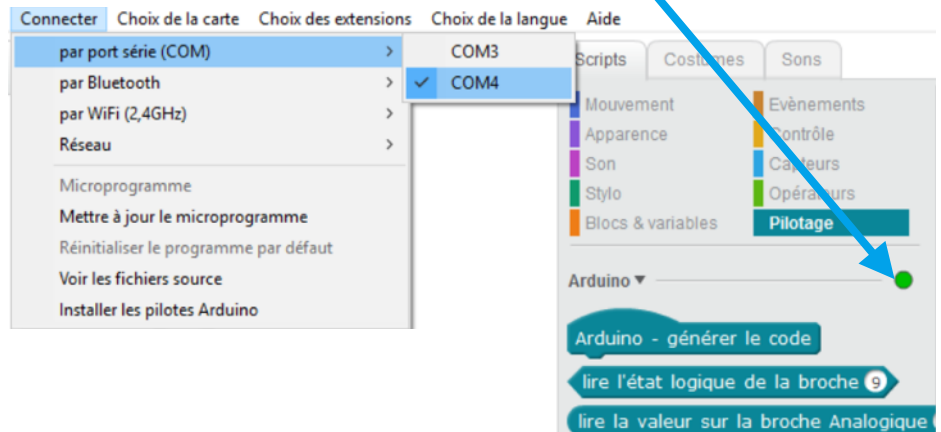


Connecter la carte :
Choix de la carte > Arduino Uno

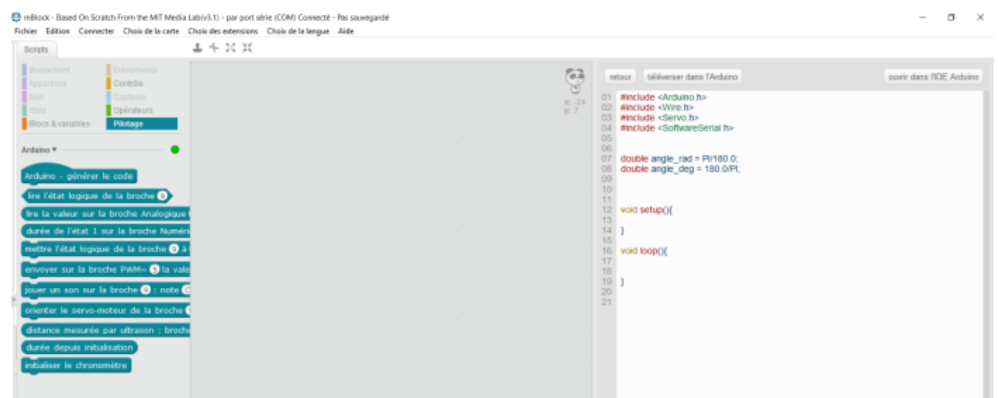
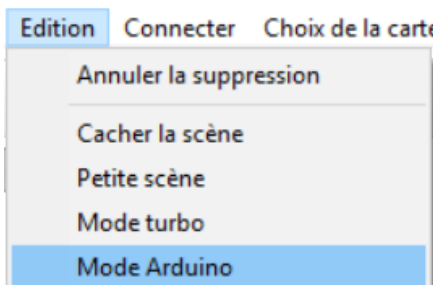


Puis Connecter > par port série (COM) > COM x

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

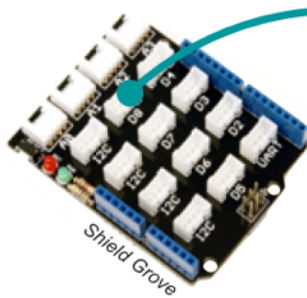


Passer en mode Arduino : Edition > Mode Arduino



PROGRAMMATION DES CAPTEURS (EN ENTRÉES)

Capteurs sur entrées Logiques/Numériques



Le capteur se connecte sur l'un des ports D2 à D8



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

Arduino - générer le code

répéter indéfiniment

si lire l'état logique de la broche 2 = 1 alors
mettre l'état logique de la broche 3 à haut
sinon
mettre l'état logique de la broche 3 à bas

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

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.

Arduino - générer le code
répéter indéfiniment
si lire l'état logique de la broche 2 = 1 alors
mettre l'état logique de la broche 3 à haut
sinon
mettre l'état logique de la broche 3 à bas

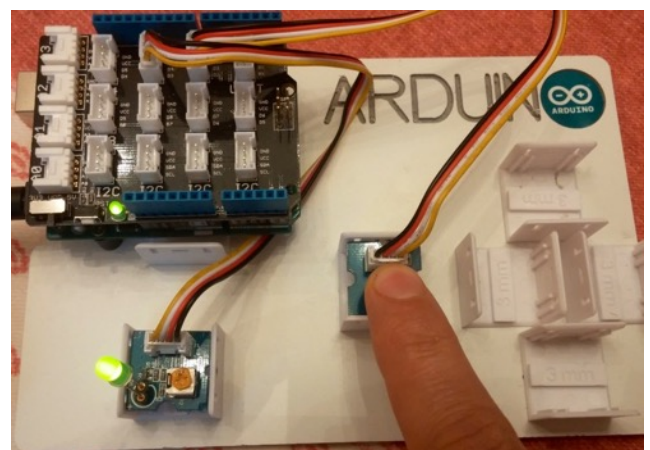
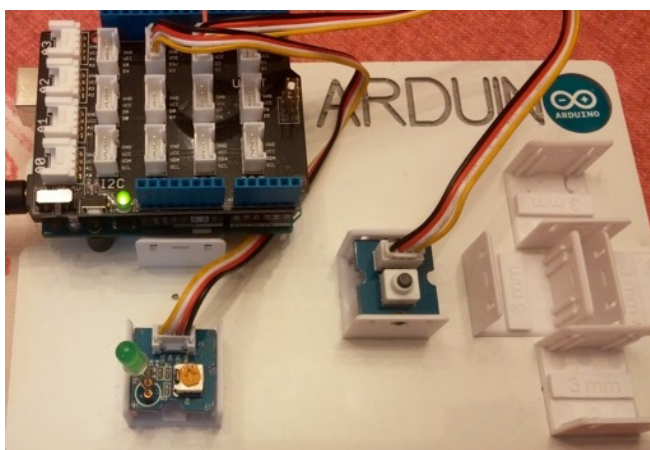
retour téléverser dans l'Arduino
01 #include <Arduino.h>
02 #include <Wire.h>
03 #include <Servo.h>
04 #include <SoftwareSerial.h>
05
06
07 double angle_rad = PI/180.0;
08 double angle_deg = 180.0/PI;
09
10
11
12 void setup(){
13 pinMode(2,INPUT);
14 pinMode(3,OUTPUT);
15 }
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Commencer le téléversement

Téléversement fini

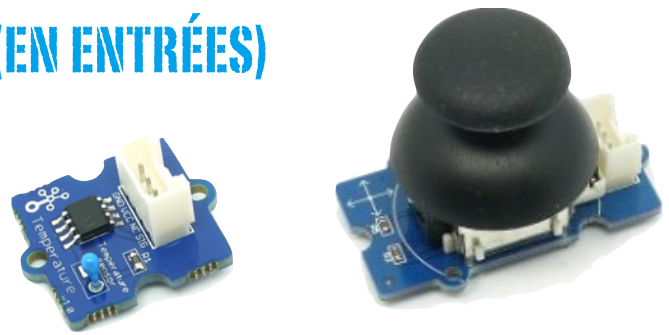
Fermer

Transférer le programme dans la carte Arduino :



PROGRAMMATION DES CAPTEURS (EN ENTRÉES)

Capteurs sur entrées Analogiques



Potentiomètre, Joystick, Capteur de Température, ...

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

Arduino - générer le code

répéter indéfiniment

```

si lire la valeur sur la broche Analogique 2 < 512 alors
  mettre l'état logique de la broche 3 à haut
sinon
  mettre l'état logique de la broche 3 à bas
    
```

Test sur le port A2 analogique

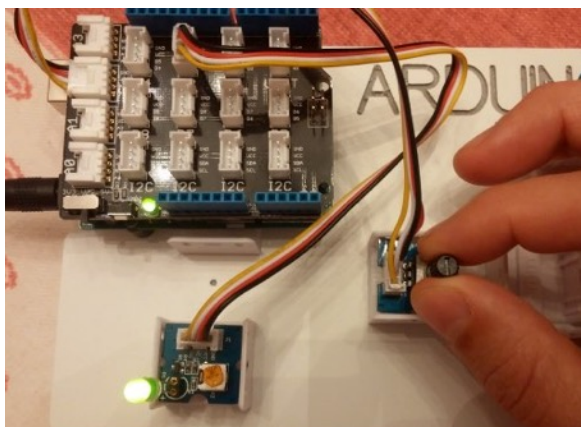
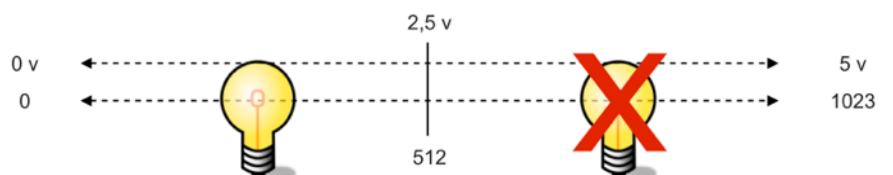
Si la valeur codée est inférieure à 512 Alors ... Sinon ...

Si OUI ALORS activer la Sortie D3

SINON ne pas activer la Sortie D3

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.



Arduino - générer le code

répéter indéfiniment

```

si lire la valeur sur la broche Analogique 2 < 512 alors
  mettre l'état logique de la broche 3 à haut
sinon
  mettre l'état logique de la broche 3 à bas
    
```

Commencer le téléversement

Téléversement fini

Fermer



retour téléverser dans l'Arduino

```

01 #include <Arduino.h>
02 #include <Wire.h>
03 #include <Servo.h>
04 #include <SoftwareSerial.h>
05
06
07 double angle_rad = PI/180.0;
08 double angle_deg = 180.0/PI;
09
10
11 void setup(){
12   pinMode(A0+2,INPUT);
13   pinMode(3,OUTPUT);
14 }
15
16
17 void loop(){
18   if((analogRead(A0+2)) < (512)){
19     digitalWrite(3,1);
20   }else{
21     digitalWrite(3,0);
22   }
23 }
24
25
26
27
    
```

PROGRAMMATION DES ACTIONNEURS (EN SORTIES)

Sorties Logiques/Numériques



Arduino - générer le code

répéter indéfiniment

mettre l'état logique de la broche 6 à haut

attendre 1 secondes

mettre l'état logique de la broche 6 à bas

attendre 1 secondes

Activer la Sortie D6

Pause de 1 seconde

Ne pas Activer la Sortie D6

Pause de 1 seconde

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

La DEL est branchée sur le port D6.

Arduino - générer le code

répéter indéfiniment

mettre l'état logique de la broche 6 à haut

attendre 1 secondes

mettre l'état logique de la broche 6 à bas

attendre 1 secondes

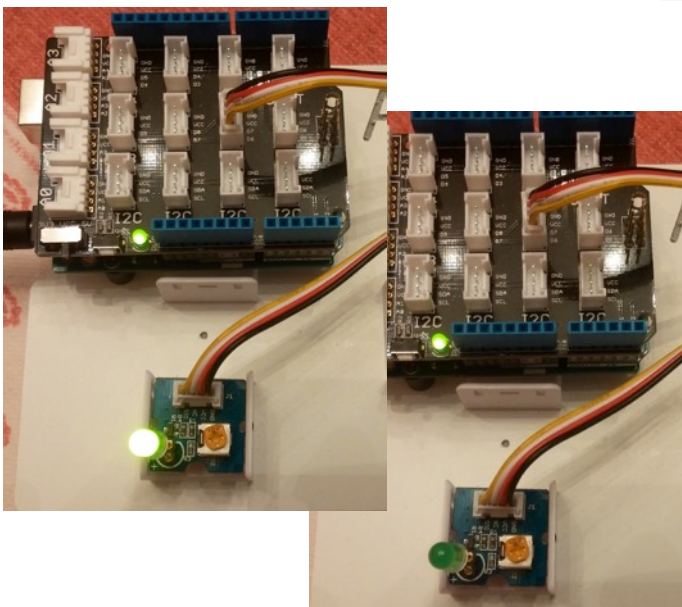
Commencer le téléversement

Téléversement fini

Fermer

retour téléverser dans l'Arduino

```
01 #include <Arduino.h>
02 #include <Wire.h>
03 #include <Servo.h>
04 #include <SoftwareSerial.h>
05
06
07 double angle_rad = PI/180.0;
08 double angle_deg = 180.0/PI;
09
10
11 void setup(){
12   pinMode(6,OUTPUT);
13 }
14
15 void loop(){
16   digitalWrite(6,1);
17   delay(1000*1);
18   digitalWrite(6,0);
19   delay(1000*1);
20 }
21
22
23
24
25
```



PROGRAMMATION DES ACTIONNEURS (EN SORTIES)

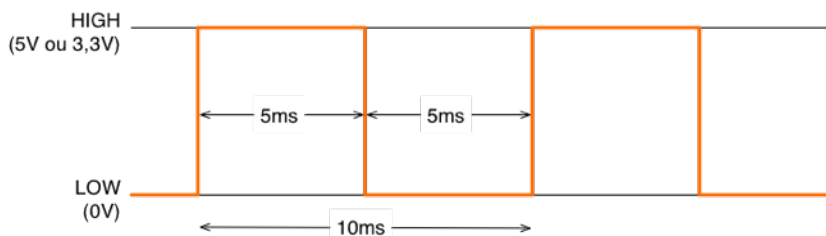
Sorties PWM (Pseudo Analogiques)

Une sortie PWM sur une carte Arduino est une sortie Numérique dont les signaux ont toujours une valeur LOW (0 logique) ou HIGH (1 logique). 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-contre, l'impression sera une **luminosité** de 50% de la luminosité maximum.

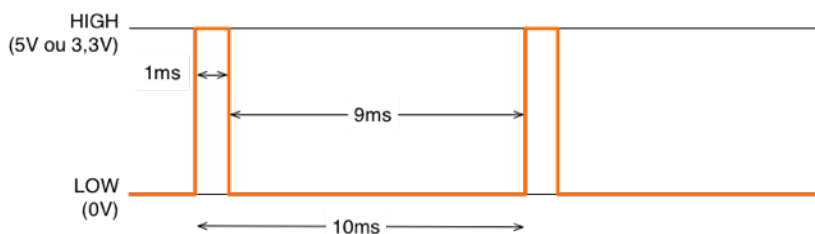
Exemple avec PMW à 50%

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



Exemple avec PMW à 10%

La fréquence est de 100Hz, le rapport cyclique de 10%



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

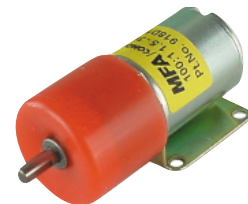
Le pourcentage de temps passé à l'état HIGH sur la période du signal est appelé le *rapport cyclique*. Il varie donc de 0%, le signal est tout le temps LOW, à 100%, le signal est tout le temps HIGH.

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



Seuls les ports **3, 5, 6, 9, 10 et 11** peuvent fournir une sortie analogique (PWM).

Ils sont repérés par le symbole :



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

Arduino - générer le code

répéter indéfiniment

envoyer sur la broche PWM~ 3 la valeur 255

attendre 1 secondes

envoyer sur la broche PWM~ 3 la valeur 100

attendre 1 secondes

envoyer sur la broche PWM~ 3 la valeur 0

attendre 1 secondes

Allumer la DEL connectée sur le port D3 à 100% de luminosité

Allumer la DEL connectée sur le port D3 à 50% de luminosité

Eteindre la DEL

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

Arduino - générer le code

répéter indéfiniment

envoyer sur la broche PWM~ 3 la valeur arrondi de lire la valeur sur la broche Analogique 2 / 4

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

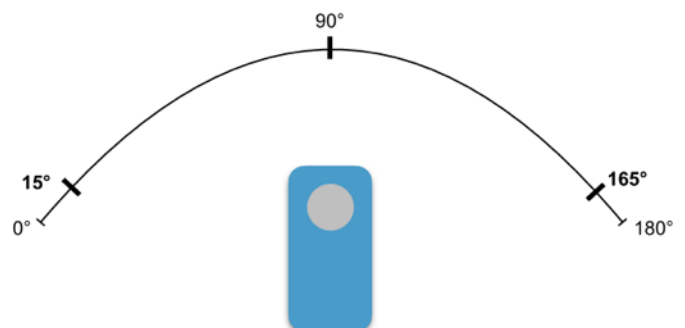


Comme déjà vu précédemment, la valeur en entrée du potentiomètre est comprise entre 0 et 1024 soit 4 fois plus que valeur attendue pour le rapport cyclique en sortie PWM (d'où la division par 4).

PILOTER UN SERVOMOTEUR



Le servomoteur se connecte sur l'un des ports de D2 à D8.



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

Arduino - générer le code

répéter indéfiniment

orienter le servo-moteur de la broche 7 à un angle de 0°

attendre 1 secondes

orienter le servo-moteur de la broche 7 à un angle de 90°

attendre 1 secondes

orienter le servo-moteur de la broche 7 à un angle de 180°

attendre 1 secondes

Servomoteur en position 0° sur le port D7

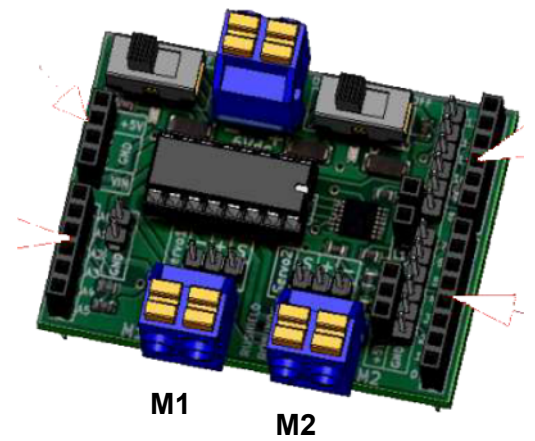
Servomoteur en position 90° sur le port D7

Servomoteur en position 180° sur le port D7

PILOTER 2 MOTEURS (DANS LES 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



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 en vitesse maximum

M1 et M2 marche avant

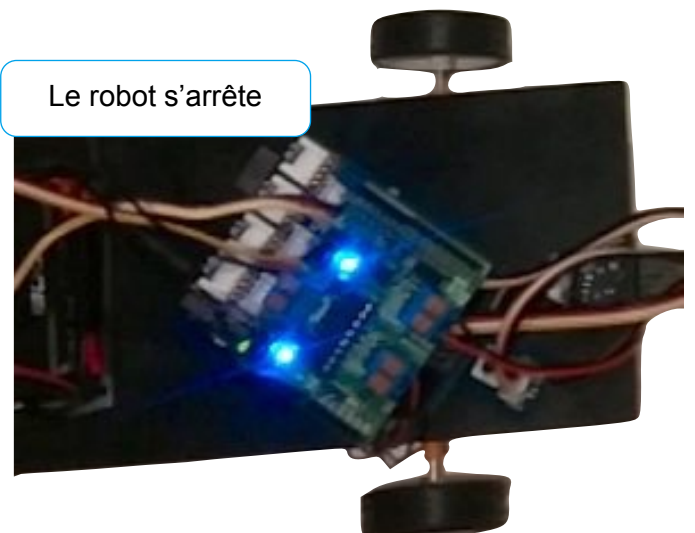
Attendre 5 secondes

Le robot tourne à droite pendant 1 seconde

Le robot recule pendant 3 secondes

Le robot s'arrête

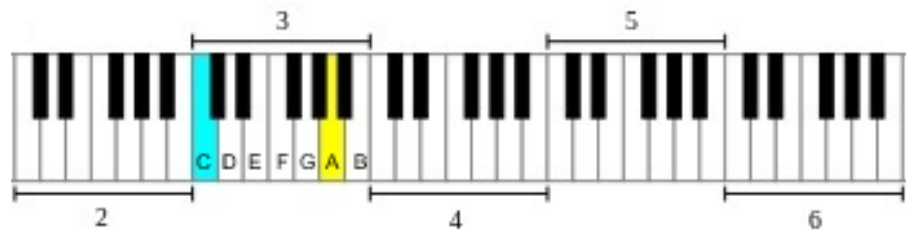
Le robot avance pendant 5 secondes



PILOTER UN 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)



Arduino - générer le code

répéter indéfiniment

jouer un son sur la broche 3 : note C3 fréquence Tout

jouer un son sur la broche 3 : note D3 fréquence un demi

jouer un son sur la broche 3 : note E3 fréquence Tout

jouer un son sur la broche 3 : note C3 fréquence Double

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

Jouer un RE sur 1/2 temps

Jouer un MI sur 1 temps

Jouer un DO sur 2 temps

UTILISER LE CAPTEUR À ULTRASON

Arduino - générer le code

répéter indéfiniment

mettre distance à distance mesurée par ultrason : broche TRIG 2 , broche ECHO 3

si distance < 20 alors

mettre l'état logique de la broche 7 à haut

sinon

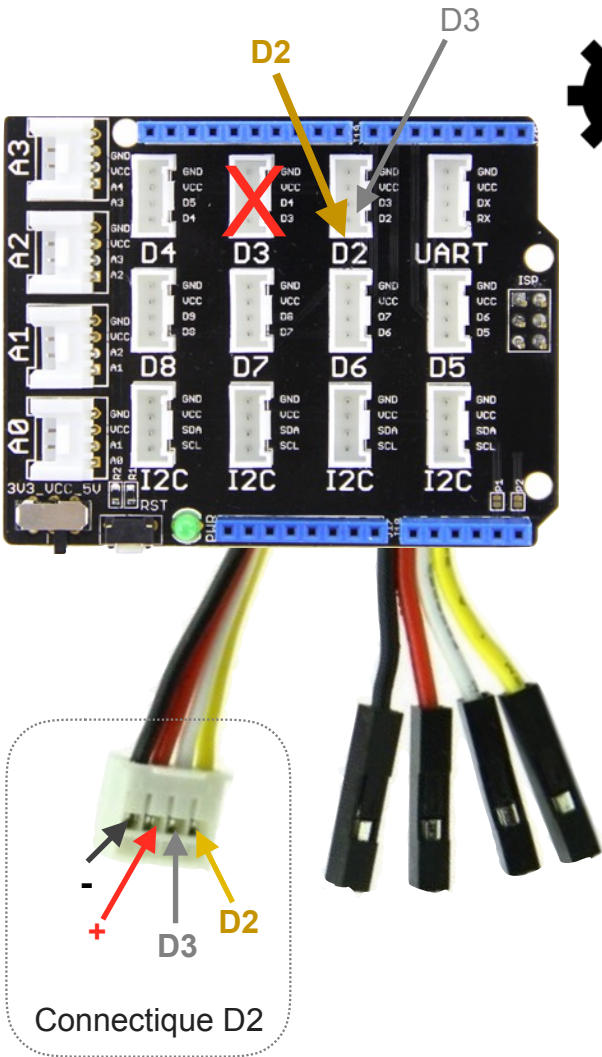
mettre l'état logique de la broche 7 à haut

Variable distance contenant la mesure du capteur

Test si la variable distance est inférieur à 20 cm

Ce programme permet de désactiver une del sur le port 7 dès que le capteur à ultrasons détecte quelque chose devant lui à moins de 20cm.

Dans le cas contraire la del reste allumée.



Prise		Capteur Ultrason
Prise D2	Noir	Gnd -
	Rouge	Vcc +
	Blanc	Echo
	Jaune	Trigger
Prise D3	A ne pas utiliser !	