

# Programmation du LittleBot

## Le Programme en Mblock :

### Objectifs :

Le robot doit tout le temps avancer sauf si il rencontre un obstacle à moins de 10 cm il doit reculer puis tourner vers la gauche.

Tout d'abord commençons par ajouter les extensions que nous auront besoin, dans la barre des extensions taper "ultrasons" et ajouter l'extension (il y en aura que une) et pour la 2ème extensions la voici [servo.next](#).

Dès que les 2 extensions sont installées on peut commencer à programmer.

Dans un premier temps nous allons faire la condition "si il voit un obstacle à moins de 10 cm".

Mettre le bloc d'évènement pour pouvoir jouer le code.



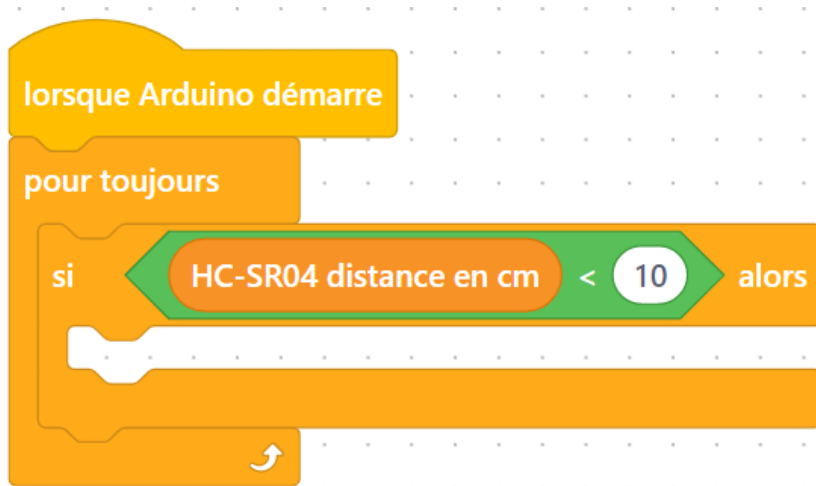
Ensuite mettre le bloc "pour toujours" pour faire une boucle infinie.



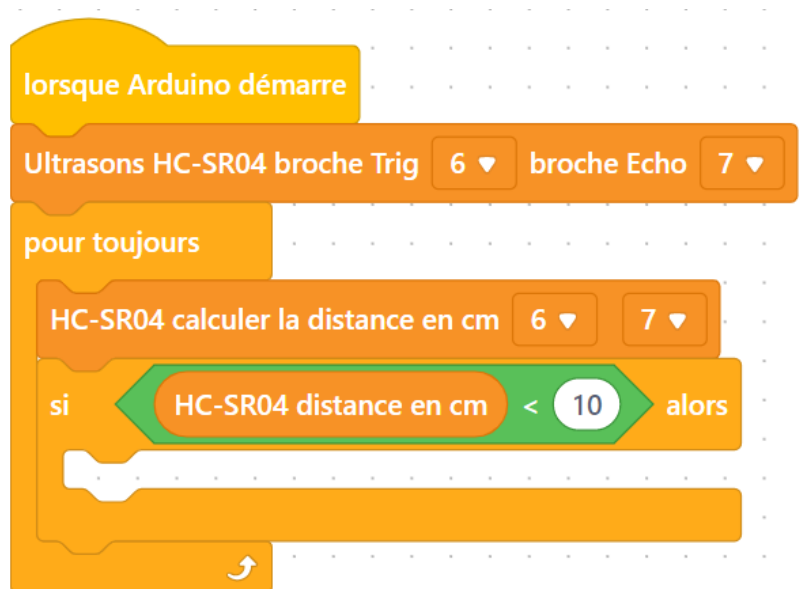
Mettre le bloc de conditions "si ... alors ...".



Maintenant on va réaliser la condition.



Ne pas oublier d'initialiser notre capteur de distance et de calculer la distance en continue.



Maintenant si la condition est correcte alors il doit d'abord reculer pendant 2 secondes puis tourner à gauche pendant 2 sec.

Les valeurs 100 et -100 sont les vitesses maximales dans un sens différents, c'est tout à fait normal que les valeurs soient égales à ces nombres car les servomoteurs ne sont pas fixés dans le même sens.



Il manque juste que si la conditions n'est pas réalisé les servomoteurs avance tout le temps.



## Programmation sur Arduino IDE :

Ici nous décomposerons notre programme pour bien l'écrire.

Tout d'abord, nous déclarons la librairie et les servo que nous utiliserons :

```
#include <Servo.h>

#define trigPin 6
#define echoPin 7

Servo servo1;
Servo servo2;
```

Puis nous déclarons sur quelles pins sont branchés notre capteur et nos servos :

```
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  servo1.attach(11);
  servo2.attach(10);
}
```

Rentrons dans le vif du sujet :

```
void loop() {
  long duration, distance;      // Nous déclarons notre variable que nous retrouverons plus tard
  digitalWrite(trigPin, LOW);   // Ici notre capteur à ultrason est en "position 0"
  delayMicroseconds(2);        // Pendant 2 Microsecondes
  digitalWrite(trigPin, HIGH);  // Ici notre capteur à ultrason est "activé"
  delayMicroseconds(10);       // Pendant 10 Microsecondes
  digitalWrite(trigPin, LOW);   // Puis nous le retournons en position "0"
  duration = pulseIn(echoPin, HIGH); // Nous déclarons notre variable "duration" qui est la
  // durée du trajet du son.
  distance = (duration*0.034) / 2; // Nous déclarons notre variables "distance" par la durée
  // multiplié par la vitesse du son le tout divisé par 2.
  if (distance < 20) {          // Nos déplacement commence ici, "Si la distance est
  // inférieur à 20cm alors..."
    servo1.writeMicroseconds(1000); // Servo Gauche tourne à l'envers
    servo2.writeMicroseconds(2000); // Servo Droit tourne à l'envers
    delay(2000); // pendant 2 sec
    servo1.writeMicroseconds(1000); // Servo Gauche tourne à l'envers
    servo2.writeMicroseconds(1500); // Arrêt du Servo Droit
    delay(2000); // pendant 2 sec
  }

  else {                        // Sinon...
    servo1.writeMicroseconds(2000); // Servo Gauche tourne
```

```
servo2.writeMicroseconds(1000); //Servo Droit tourne  
delay (2000); // pendant 2 sec  
}  
}
```

Puis nous assemblons le tout, voici à quoi cela devrait ressembler :

```
#include <Servo.h>  
#define trigPin 6  
#define echoPin 7  
Servo servo1;  
Servo servo2;  
  
void setup() {  
  pinMode(trigPin, OUTPUT);  
  pinMode(echoPin, INPUT);  
  servo1.attach(11);  
  servo2.attach(10);  
}  
  
void loop() {  
  long duration, distance;  
  digitalWrite(trigPin, LOW);  
  delayMicroseconds(2);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  duration = pulseIn(echoPin, HIGH);  
  distance = (duration*0.034) / 2;  
  if (distance < 20) {  
    servo1.writeMicroseconds(1000);  
    servo2.writeMicroseconds(2000);  
    delay (2000);  
    servo1.writeMicroseconds(1000);  
    servo2.writeMicroseconds(1500);  
    delay (2000);  
  }  
  
  else {
```

```
servo1.writeMicroseconds(2000);  
servo2.writeMicroseconds(1000);  
  delay (2000);  
}  
}
```

---

Revision #4

Created 9 April 2025 14:07:12 by Quentin Petrazoller

Updated 10 April 2025 14:32:40 by Quentin Petrazoller