Programmation 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, 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 extension la voici servo.mext (cliquez dessus pour télécharger le fichier et faire un glisser-déposer du fichier sur Mblock pour importer l'extension).

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

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



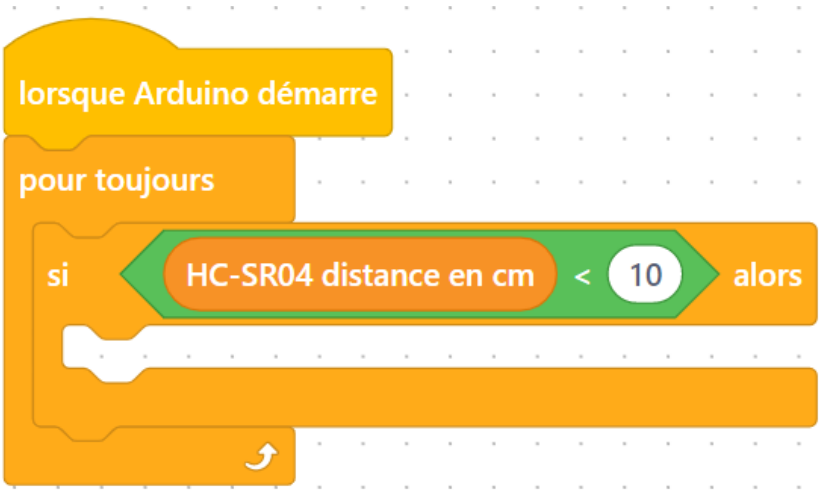
Ensuite mettre le bloc "pour toujours" pour faire une boucle infinie.



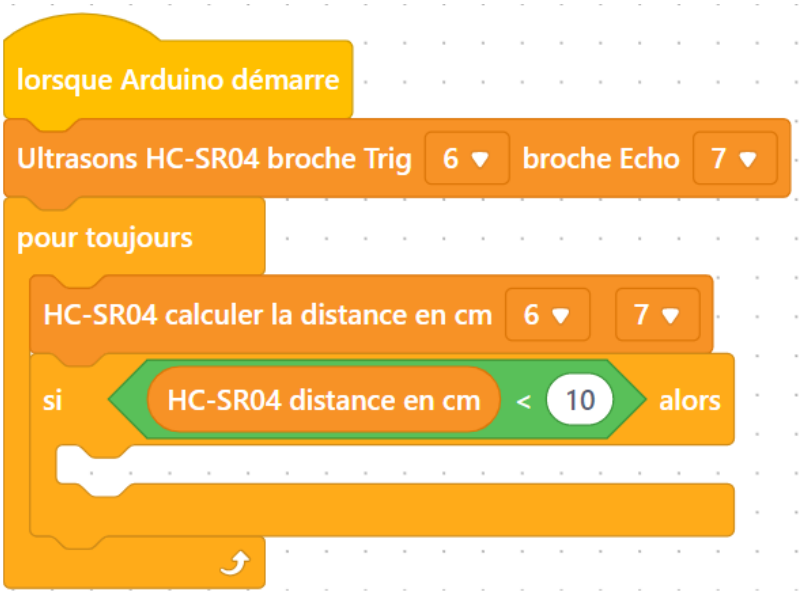
Mettre le bloc de conditions "si ... alors ...".



Maintenant on va réaliser la condition.

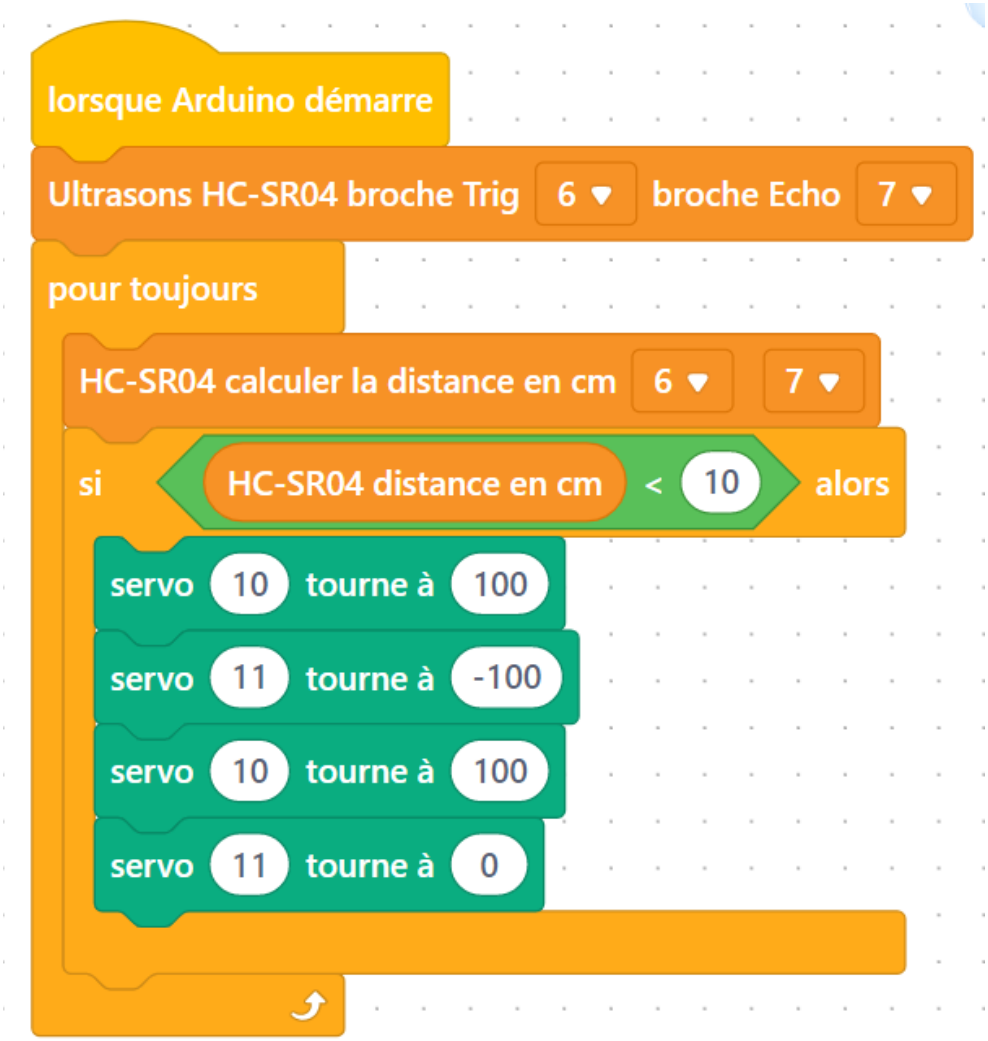


Ne pas oublier d'initialiser notre capteur de distance et de calculer la distance en continue.

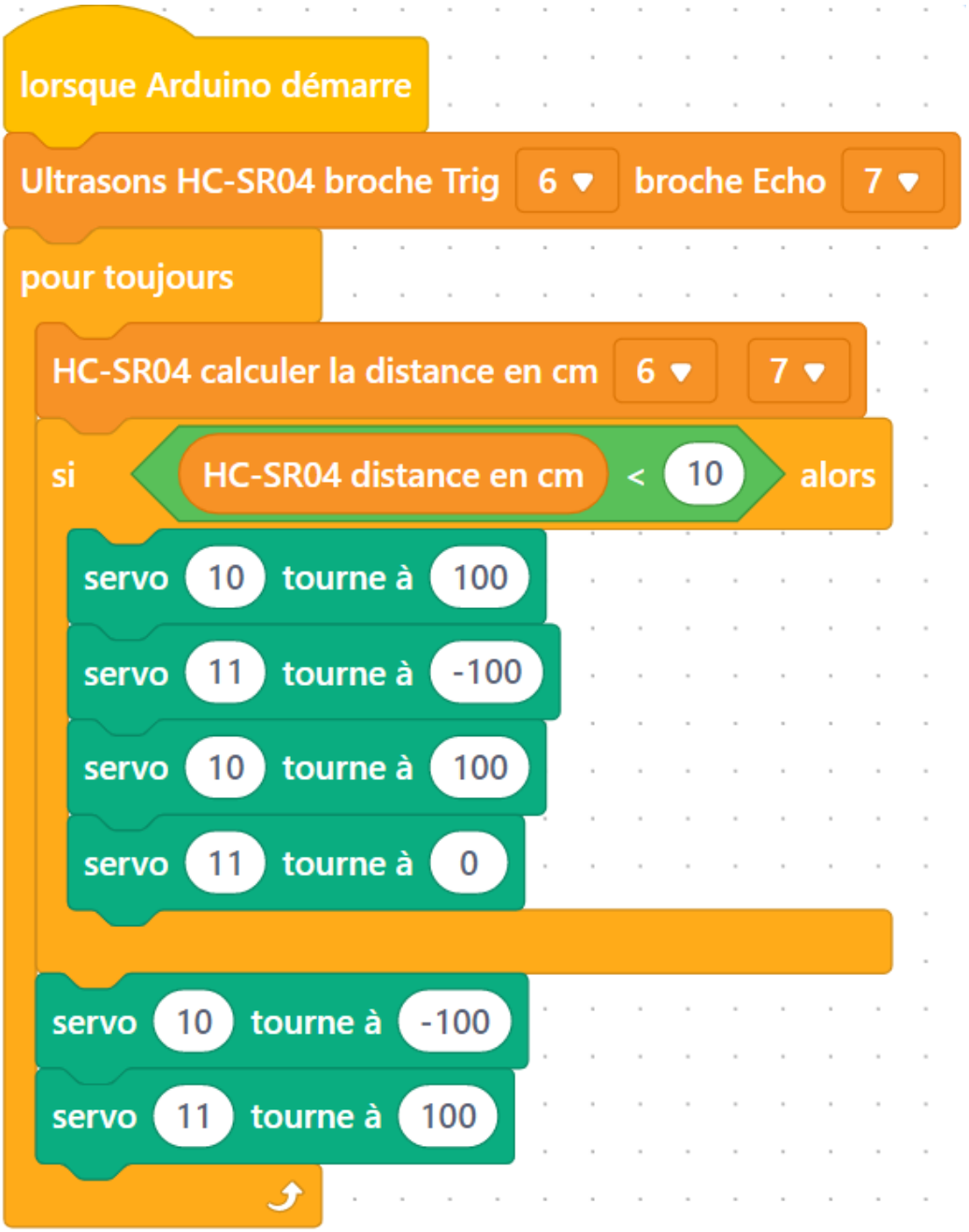


Maintenant si la condition est correct alors il doit d'abord reculer puis tourner à gauche.

Les valeurs 100 et -100 sont les vitesse maximale dans un sens différent.



Rajouter les 2 dernier blocs pour que si la condition n'est pas réalisée le robot avance.



Programmation sur Arduino IDE :

Ici nous décomposerons notre programme pour bien l'écrire.

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

```
#include <Servo.h> // Inclusion de la bibliothèque Servo pour contrôler les servomoteurs
#define trigPin 6 // Attribution des pins du capteur à ultrasons
#define echoPin 7 // Attribution des pins du capteur à ultrasons
Servo servo1;  // Déclaration du premier servomoteur
```

```
Servo servo2;  // Déclaration du second servomoteur
```

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

```
void setup() {
pinMode(trigPin, OUTPUT); // Configure la broche trigPin (6) en sortie
pinMode(echoPin, INPUT); // Configure la broche echoPin (7) en entrée
servo1.attach(11); // Attribution de la broche 11 au premier servomoteur
servo2.attach(10); // Attribution de la broche 10 au deuxieme servomoteur
}
```

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 duréé
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; // Nous déclarons notre variable que nous retrouverons plus tard
// Envoie une impulsion courte pour déclencher le capteur ultrasonique
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);  // Début de l'impulsion
delayMicroseconds(10);        // Durée de l'impulsion (10µs)
digitalWrite(trigPin, LOW);   // Fin de l'impulsion
// Mesure du temps entre l'émission et la réception de l'onde sonore
duration = pulseIn(echoPin, HIGH);
// Conversion de la durée en distance (en cm)
distance = (duration*0.034) / 2;
// Si un obstacle est détecté à moins de 20 cm
if (distance < 20) {
  // Mouvement 1 : envoie d'un signal d'une durée 1000 µs à servo1, envoie d'un signal d'une
durée 2000 µs à servo2
  servo1.writeMicroseconds(1000);
  servo2.writeMicroseconds(2000);
  delay (2000); // Attend 2 secondes
  // Mouvement 2 : servo1 reste dans sa position, envoie d'un signal d'une durée de 1500 us à
servo2
  servo1.writeMicroseconds(1000);
  servo2.writeMicroseconds(1500);
  delay (2000);// Attend 2 secondes
```

```
}

else {
 // Si aucun obstacle détecté on avance normalement
 servo1.writeMicroseconds(2000);
 servo2.writeMicroseconds(1000);
 delay (2000);
}
}
}
```

Revision #3
Created 1 July 2025 00:34:52 by Gaëtan Carron
Updated 25 November 2025 13:29:22 by Quentin Petrazoller