Leçon 4 – Eviter les obstacles

Point abordés dans cette section

La joie de l'apprentissage, n'est pas seulement savoir comment contrôler votre voiture, mais aussi savoir comment protéger votre voiture.

Donc, rendez votre voiture loin de la collision.

Apprentissage

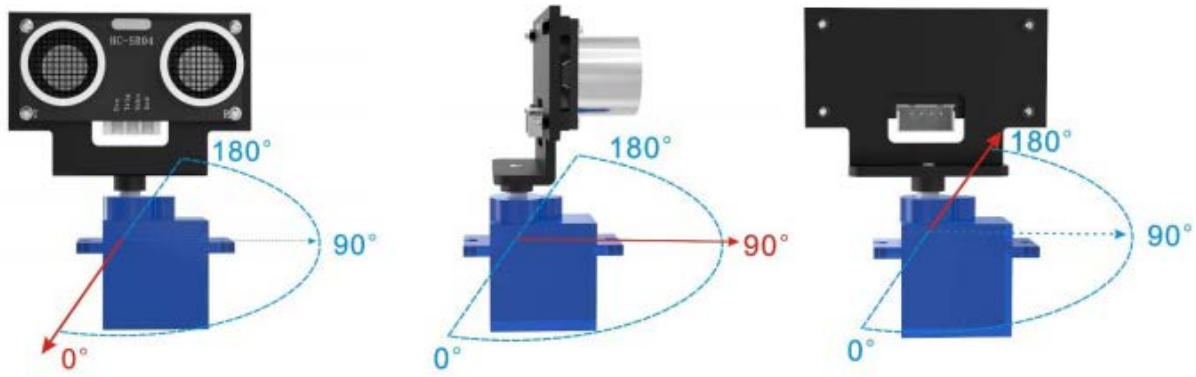
- *Apprenez comment assembler le module ultrasonique*
- *Familiarisez-vous avec l'utilisation de la direction*
- *Découvrez le principe de l'évitement de la voiture*
- *Utilisez le programme pour que la voiture évite les obstacles*

Préparation

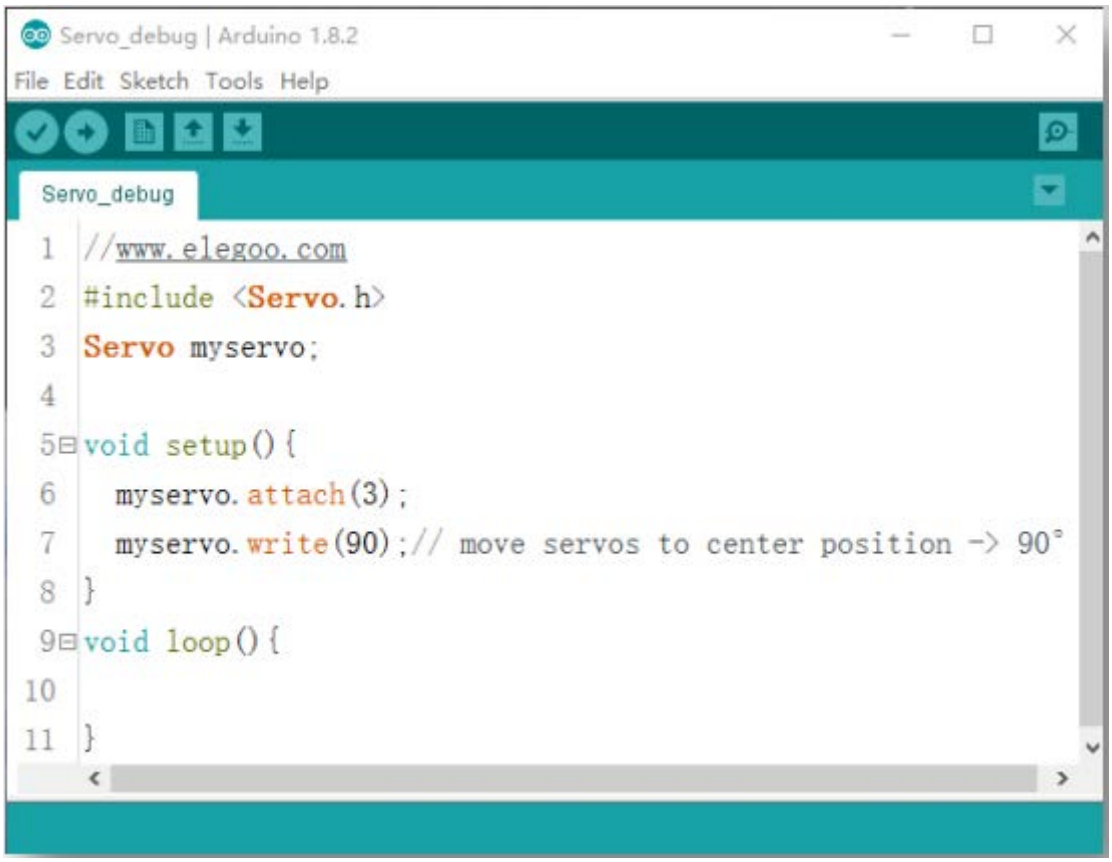
- *Un véhicule (équipé d'une batterie)*
- *Un câble USB*
- *Module de détection de distance*

I – Connexion

Lors de l'assemblage du support de module de capteur à ultrasons, le servo doit configurer afin de s'assurer qu'il puisse tourner à 180 degrés.

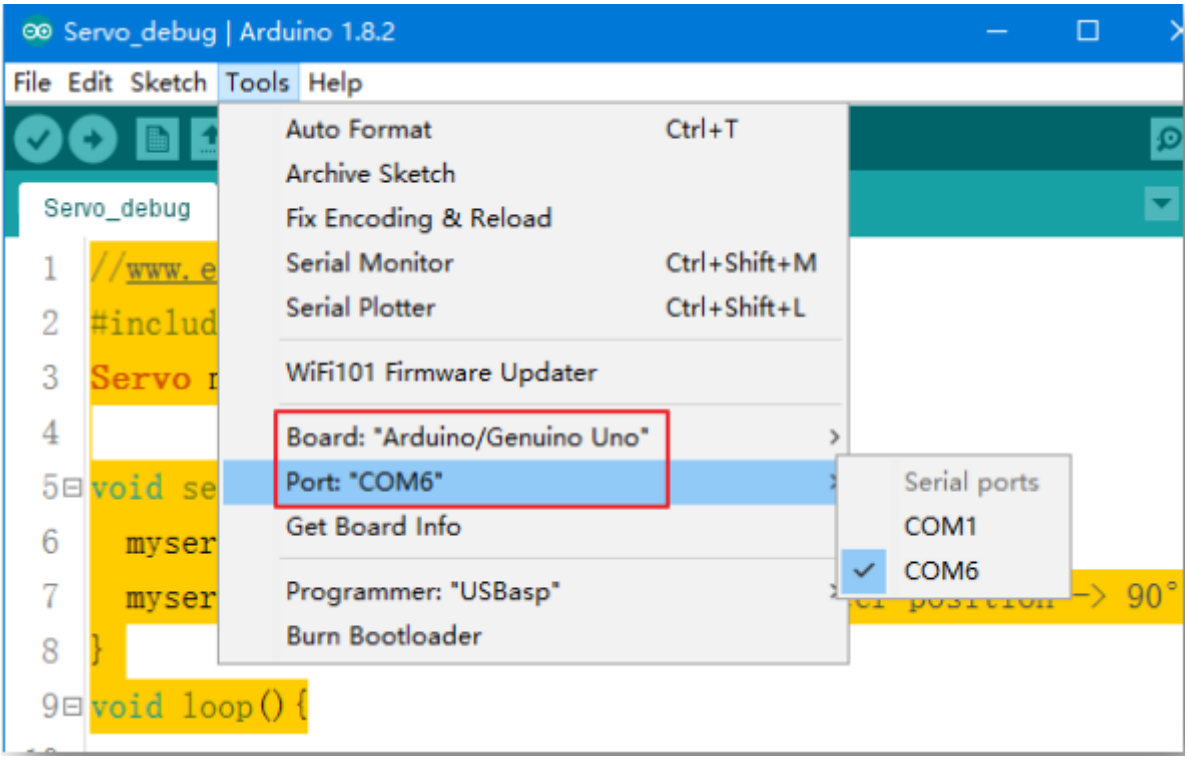


STEP1: Connectez l'UNO à l'ordinateur et ouvrez le fichier de code Servo_debug dans le chemin "\ Leçon 4 Voiture à évitement d'obstacles \ Servo_debug \ Servo_debug.ino".

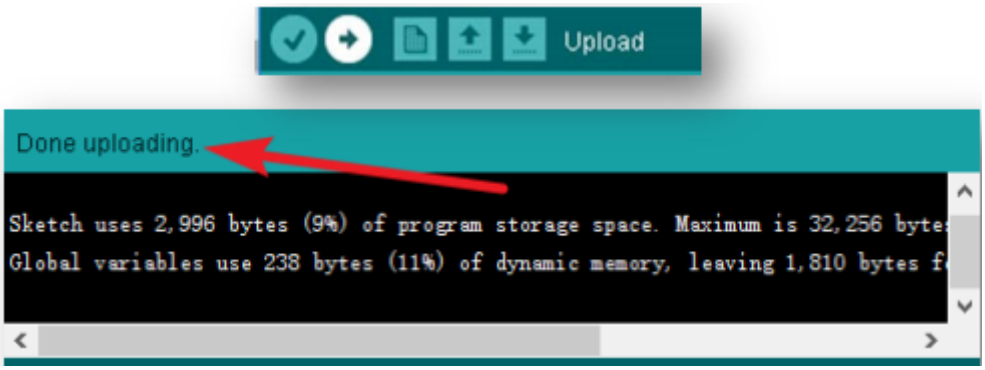


```
//www.elegoo.com
#include <Servo.h>
Servo myservo;
void setup(){
myservo.attach(3);
myservo.write(90);// move servos to center position -> 90°
}
void loop(){
}
```

ÉTAPE 2: Sélectionnez "Outil" -> "Port" et "Carte" dans l'IDE Arduino.



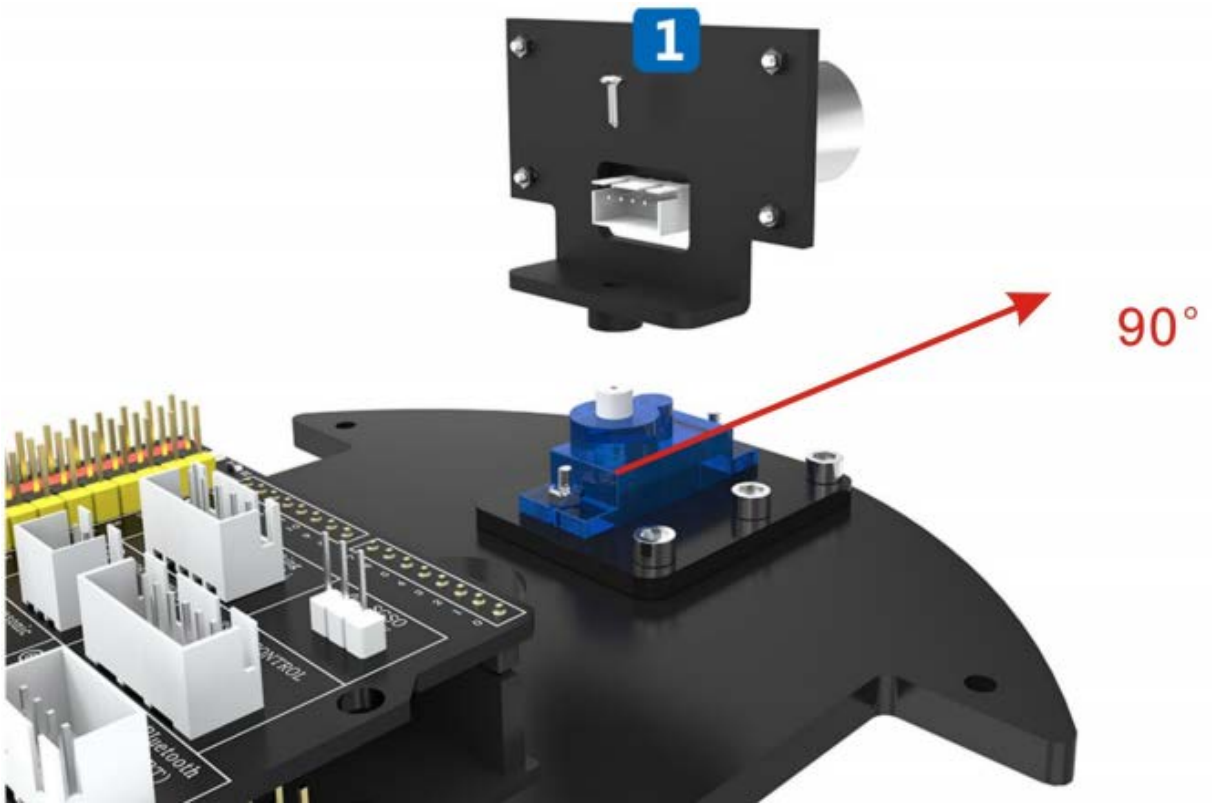
STEP3: cliquez sur le bouton flèches pour télécharger le code sur la carte contrôleur UNO.



Après avoir effectué le chargement, le servo tourne à 90 degrés et s'immobilise.

ÉTAPE 4: Assembler le module de capteur à ultrasons à 90 degrés.

L'angle de chaque dent sur micro servo est de 15 degrés et si vous l'installez au milieu de la direction de 90 degrés, il tourne à gauche ou à droite de 15 degrés, ce qui signifie que le degré réel d'installation du micro servo est de 15 degrés ou 105 degrés.



FAQ sur le servomoteur.

1 - Pourquoi le micro servo tourne-t-il dans le sens inverse des aiguilles d'une montre de 15 degrés chaque fois que j'allume le courant?

Ceci est normal pour le servomoteur SG90 et cela n'affectera pas l'utilisation normale avec les programmes.

Si vous ne l'avez pas contrôlé avec des programmes, vous pouvez le faire tourner à la normale avec votre main ou bien brancher les fils connectés avec un micro servo avant d'allumer l'alimentation.

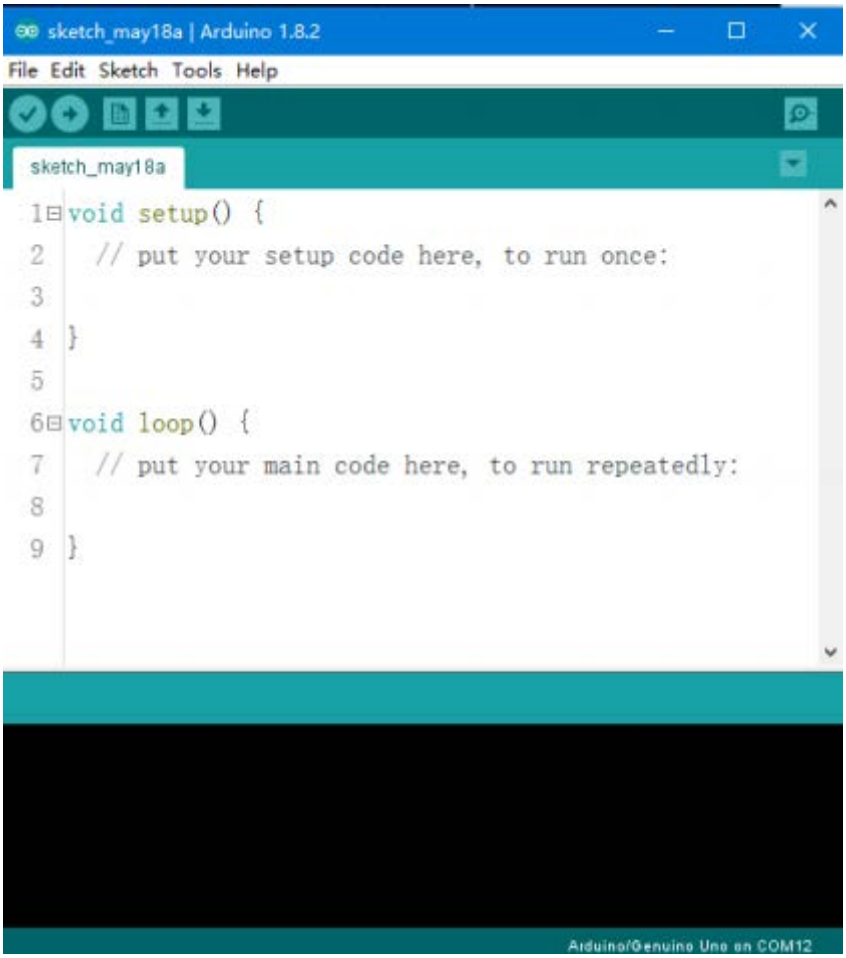
2 - Le micro servo est hors de contrôle et continue de tourner.

Utilisez "myservo.write (angle)" pour commander le servo micron au degré d'angle qui a une plage de 0 à 180. Si elle dépasse la portée, le micro servo ne reconnaîtra pas cet angle et continuera à tourner.

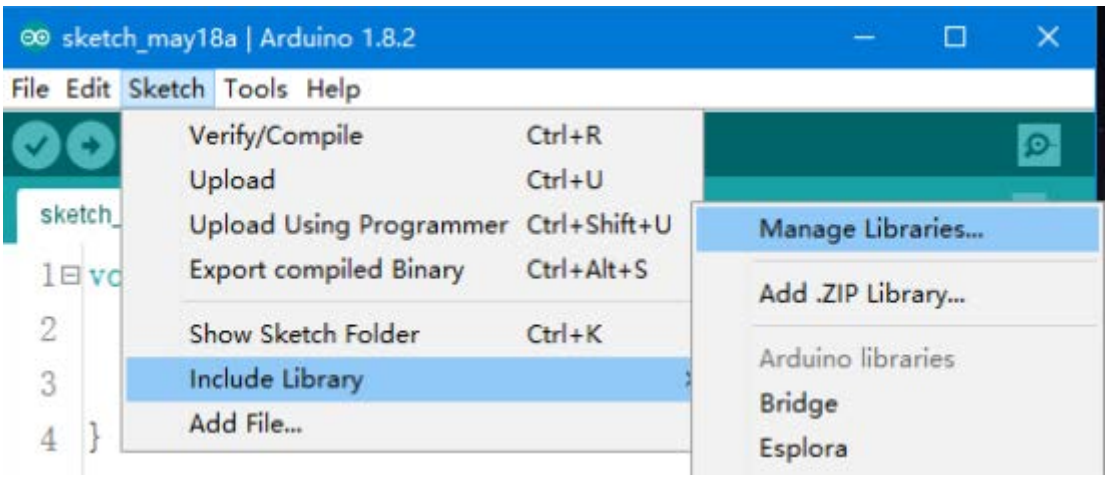
II – Charger le programme

Parce que le programme utilise la bibliothèque <servo.h>, nous devons d'abord installer la bibliothèque.

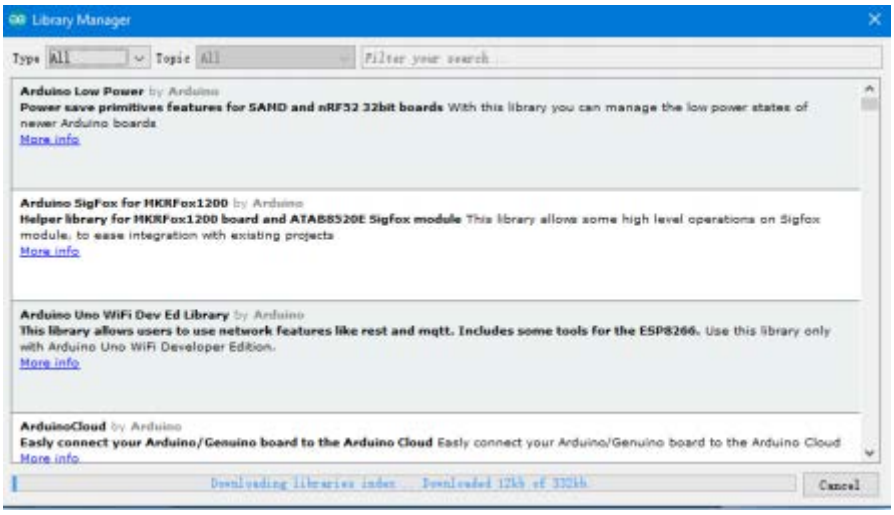
Ouvrez le logiciel Arduino



Select Sketch -> Include Library -> Manage Libraries...

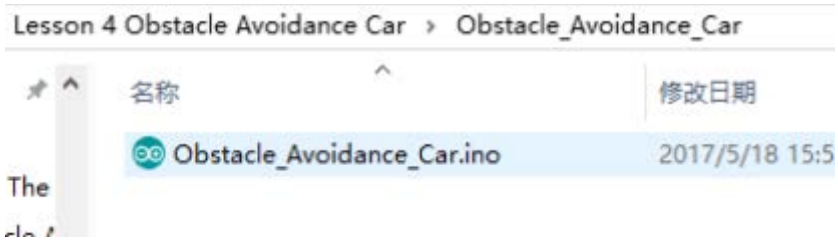


Attendez " Downloading libraries index" pour terminer.



Recherchez le servo, puis installez la version la plus récente. L'image suivante montre que la bibliothèque Servo est déjà installée.

Connectez la carte contrôleur UNO à l'ordinateur, ouvrez le fichier de code dans le chemin "\ Leçon 4 Voiture à evitement d'obstacles \ Obstacle_Avoidance_Car \ Obstacle_Avoidance_Car.ino". Téléchargez le programme dans le tableau UNO.




```
//www.elegoo.com

#include <Servo.h> //servo library

Servo myservo; // create servo object to control servo

int Echo = A4;

int Trig = A5;

#define ENA 5

#define ENB 6
```

```
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11
#define carSpeed 150
int rightDistance = 0, leftDistance = 0, middleDistance = 0;
void forward(){
analogWrite(ENA, carSpeed);
analogWrite(ENB, carSpeed);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
Serial.println("Forward");
}
void back() {
analogWrite(ENA, carSpeed);
analogWrite(ENB, carSpeed);
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
Serial.println("Back");
}
void left() {
analogWrite(ENA, carSpeed);
analogWrite(ENB, carSpeed);
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
```

```
Serial.println("Left");
}
void right() {
analogWrite(ENA, carSpeed);
analogWrite(ENB, carSpeed);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
Serial.println("Right");
}
void stop() {
digitalWrite(ENA, LOW);
digitalWrite(ENB, LOW);
Serial.println("Stop!");
}
//Ultrasonic distance measurement Sub function
int Distance_test() {
digitalWrite(Trig, LOW);
delayMicroseconds(2);
digitalWrite(Trig, HIGH);
delayMicroseconds(20);
digitalWrite(Trig, LOW);
float Fdistance = pulseIn(Echo, HIGH);
Fdistance= Fdistance / 58;
return (int)Fdistance;
}
void setup() {
myservo.attach(3); // attach servo on pin 3 to servo object
Serial.begin(9600);
pinMode(Echo, INPUT);
```

```
pinMode(Trig, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
pinMode(ENA, OUTPUT);
pinMode(ENB, OUTPUT);
stop();
}
void loop() {
myservo.write(90); //setservo position according to scaled value
delay(500);
middleDistance = Distance_test();
if(middleDistance <= 20) {
stop();
delay(500);
myservo.write(10);

delay(1000);
rightDistance = Distance_test();

delay(500);
myservo.write(90);
delay(1000);
myservo.write(180);
delay(1000);
leftDistance = Distance_test();

delay(500);
myservo.write(90);
delay(1000);
```

```
if(rightDistance > leftDistance) {
right();
delay(360);
}
else if(rightDistance < leftDistance) {
left();
delay(360);
}
else if((rightDistance <= 20) || (leftDistance <= 20)) {
back();
delay(180);
}
else {
forward();
}
}
else {
forward();
}
}
```

Après avoir téléchargé le programme sur le panneau de contrôle UNO, débranchez le câble, mettez le véhicule sur le sol et allumer l'alimentation électrique.

Vous verrez que le véhicule avance et que la plate-forme du détecteur continue à tourner pour que les capteurs de mesure de distance fonctionnent en permanence. S'il y a des obstacles à venir, la plate-forme cloud s'arrêtera et le véhicule changera sa direction pour contourner l'obstacle. Après avoir contourné l'obstacle, la plate-forme du détecteur continuera à tourner et le véhicule se déplacera également.

III – Introduction au principe de fonctionnement

Tout d'abord, apprenons le fonctionnement du servo SG90:

Classification: 180 appareils de direction

Normalement, le servo dispose de 3 lignes de commande: alimentation, masse et panneau.

Définition des servo-broches: ligne marron = GND, ligne rouge = 5V, signal = orange.

Comment fonctionne le servo:

La puce de modulation de signal dans le servo reçoit des signaux de la carte contrôleur puis le servo obtiendra la tension DC de base. Il existe également un circuit de référence à l'intérieur du servo qui produira une tension standard. Ces deux tensions se comparent entre elles et la différence sera extraite. Ensuite, la puce du moteur recevra la différence et décidera de la vitesse de rotation, de la direction et de l'ange. Lorsqu'il n'y a pas de différence entre les deux tensions, le servo s'arrêtera.

Comment contrôler le servo:

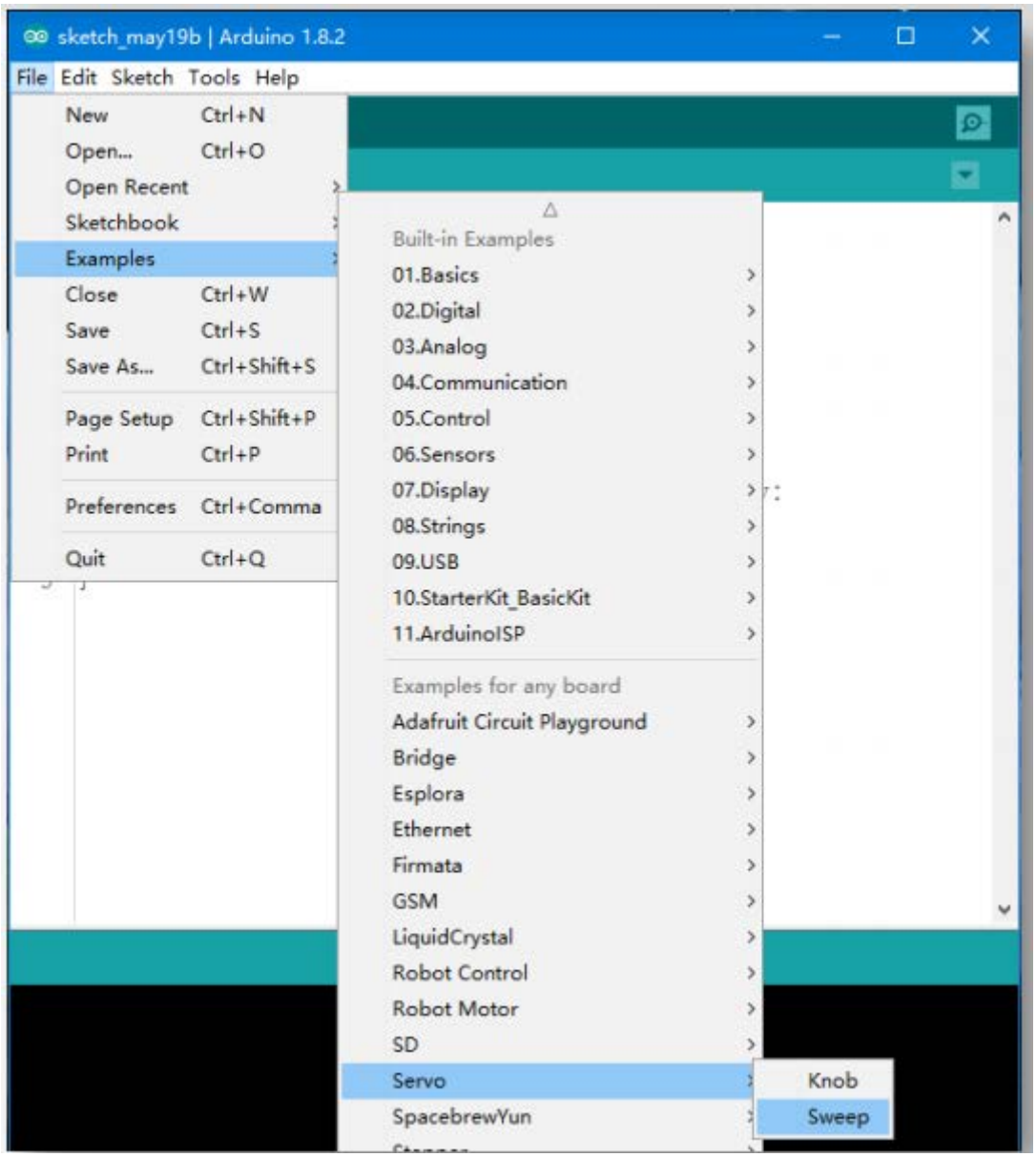
Pour contrôler la rotation de l'asservissement, il faut que l'impulsion du temps atteigne environ 20 ms et que la largeur d'impulsion de niveau élevé soit d'environ 0,5 ms ~ 2,5 ms, ce qui correspond à l'angle limite du servo.

Prenons un servo permettant un angle 180, la relation de contrôle correspondante est la suivante:

0.5ms	0 degree
1.0ms	45 degree
1.5ms	90 degree
2.0ms	135 degree
2.5ms	180 degree

Le programme d'exemple:

Ouvrez Arduino IDE et sélectionnez "Fichier-> Exemples-> Servo-> Balayage"



Ensuite, regardons le module capteur à ultrasons.

Caractéristique du module: distance de test, module haute précision.

Application des produits: évitement de l'obstacle du robot, distance d'essai de l'objet, essai des liquides, sécurité publique, test du stationnement.

Principaux paramètres techniques

(1): tension utilisée: DC --- 5V

(2): courant statique: moins de 2 mA

(3): niveau de production: supérieur à 5V

(4): niveau de sortie: inférieur à 0

(5): angle de détection: pas plus grand que 15 degrés

(6): distance de détection: 2cm-450cm

(7): haute précision: jusqu'à 0.2cm Méthode de connexion des lignes: VCC, trig (la fin de la commande), écho (fin de réception), GND

Comment fonctionne le module:

(1) Appliquer le port IO de TRIG pour déclencher la portée, donner un signal de haut niveau, au moins 10us;

(2) Le module envoie 8 ondes carrées de 40kz automatiquement, teste s'il y a des signaux retournés automatiquement;

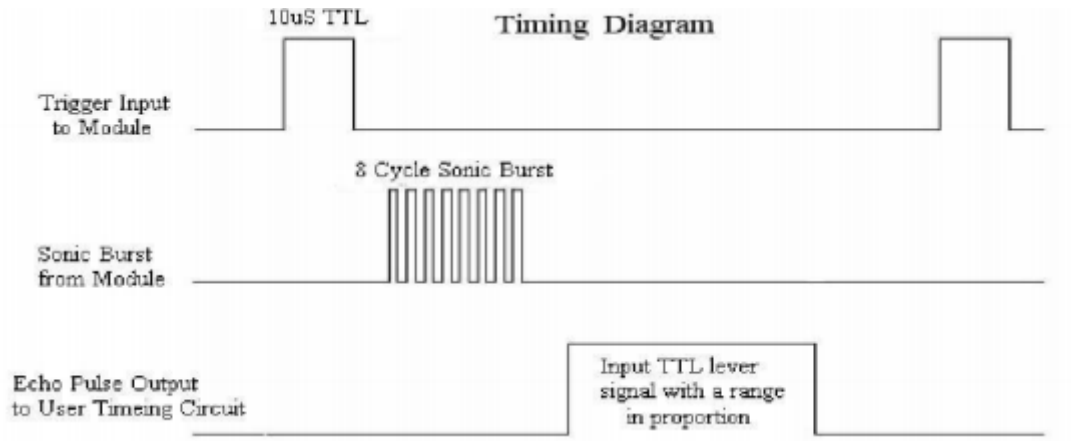
(3) S'il y a des signaux reçus, le module émettra une impulsion de niveau élevé via le port IO d'ECHO, le temps de durée de l'impulsion de haut niveau étant le temps entre l'envoi et la réception des ondes. Ainsi, le module peut connaître la distance en fonction de l'heure.

Distance de test = (temps de niveau * vitesse du son (340M / S)) / 2);

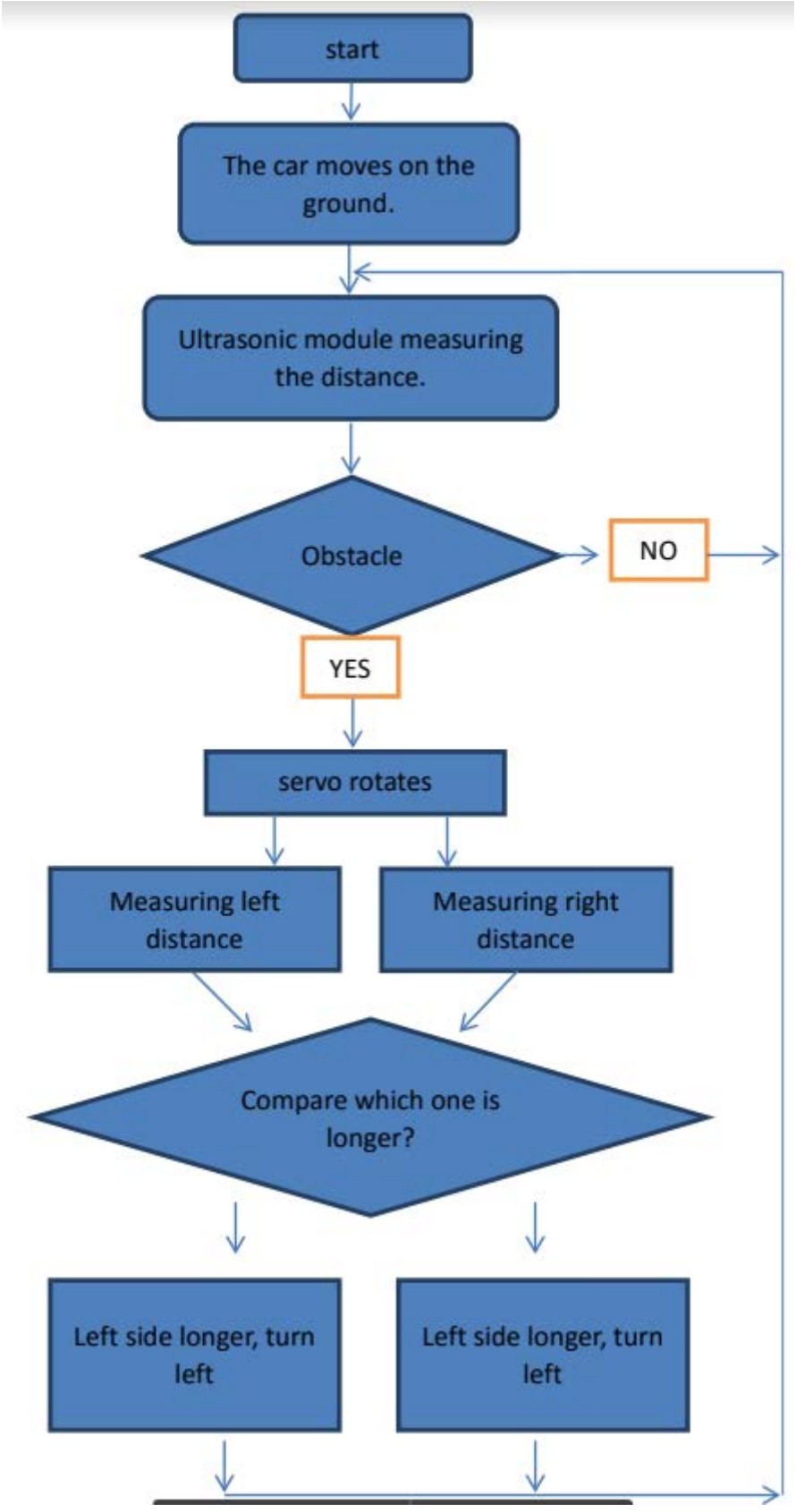
Fonctionnement réel:

Le diagramme de synchronisation est illustré ci-dessous. Il vous suffit de fournir une impulsion courte de 10uS à l'entrée de déclenchement pour commencer la portée, puis le module enverra une suite de 8 cycles d'échographie à 40 kHz et augmentera son écho. L'Echo est un objet à distance qui est la largeur d'impulsion et la plage proportionnelle. Vous pouvez calculer la plage à travers l'intervalle de temps entre l'envoi du signal de déclenchement et le signal d'écho de réception.

Formule: uS / 58 = centimètres ou uS / 148 = pouces; ou: la plage = temps de niveau élevé * vitesse (340M / S) / 2; nous vous suggérons d'utiliser un cycle de mesure de plus de 60 ms, afin d'éviter le signal de déclenchement du signal d'écho.




```
/*Ultrasonic distance measurement Sub function*/
int Distance_test()
{
digitalWrite(Trig, LOW);
delayMicroseconds(2);
digitalWrite(Trig, HIGH);
delayMicroseconds(20);
digitalWrite(Trig, LOW);
float Fdistance = pulseIn(Echo, HIGH);
Fdistance= Fdistance/58;
return (int)Fdistance;
}
```



Du diagramme ci-dessus, nous pouvons voir que le principe de la voiture à évitement d'obstacles est très simple. Le module de capteur à ultrasons détectera de plus en plus la distance entre la voiture et

les obstacles et envoie les données à la carte contrôleur, puis la voiture arrête et tourne le servo pour effectuer une détection à gauche puis à droite. Après avoir comparé la distance du côté différent, la voiture tourne vers le côté qui a plus de distance et avance. Ensuite, le module capteur à ultrasons détecte à nouveau la distance.

```
if(rightDistance > leftDistance) {
right();
delay(360);
}
else if(rightDistance < leftDistance) {
left();
delay(360);
}
else if((rightDistance <= 20) || (leftDistance <= 20)) {
back();
delay(180);
}
else {
forward();
}
```