

# *Leçon 1 – Faire bouger la voiture*

## *Point abordés dans cette section*

### *Apprentissage*

- *Apprendre comment utiliser l'interface de l'Arduino*
- *Faire bouger la voiture en chargeant le programme*

### *Préparation*

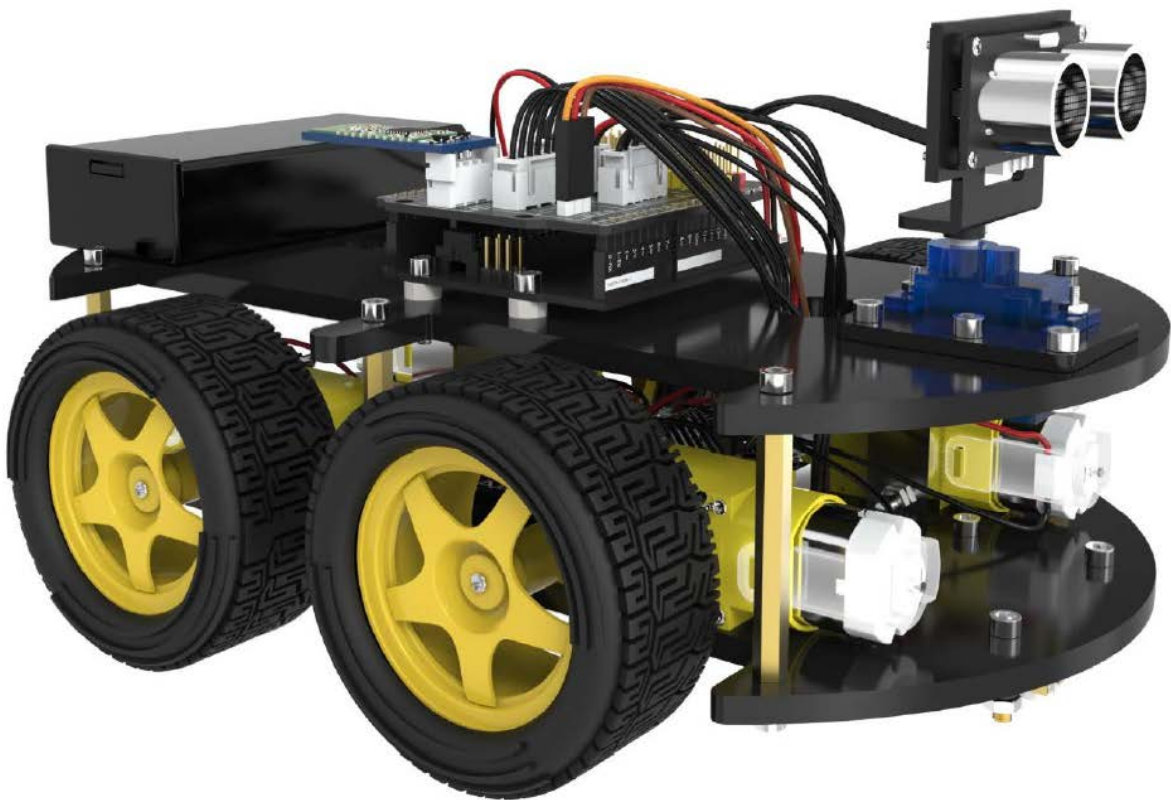
- *Une voiture avec les batteries*
- *Un câble USB*

## I – Introduction sur la voiture

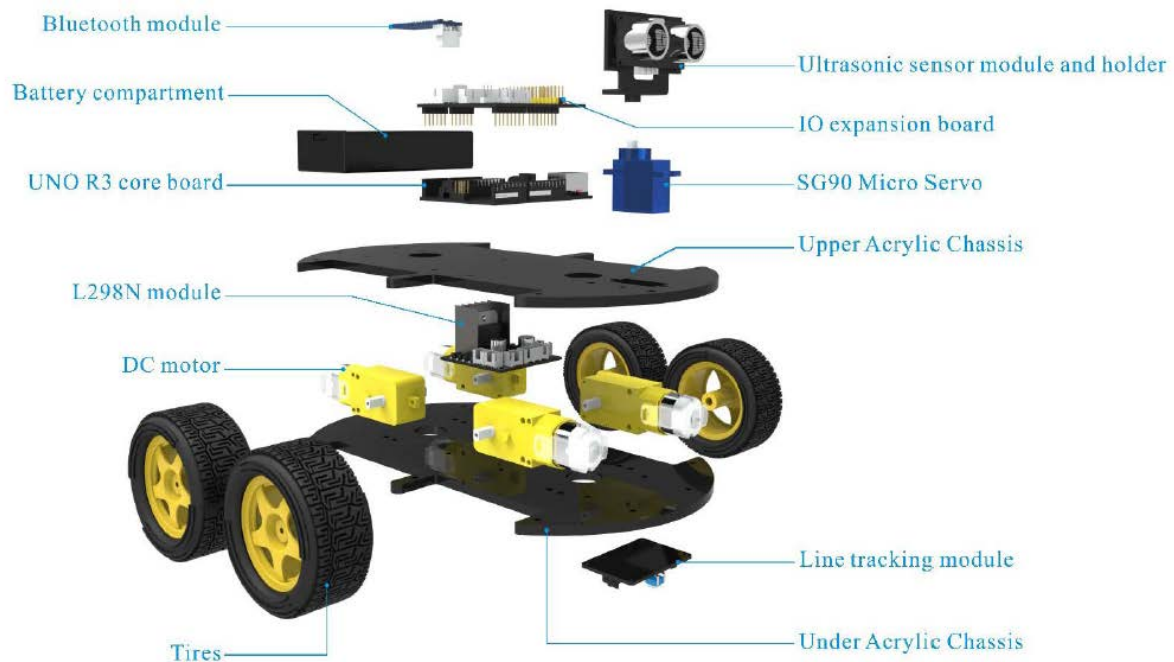
Ce kit est un kit véhicule extrêmement flexible spécialement conçu pour l'éducation, la compétition et à des fins de divertissement. Le panneau supérieur du kit est directement compatible avec un servo moteur de 9 grammes.

Il porte également un capteur supersonique, une batterie ainsi qu'un ensemble de trou permettant de fixer facilement d'autres capteurs. C'est un robot amusant et polyvalent qui répond aux objectifs d'apprentissage et de production. Avec lui, vous pouvez mettre en œuvre diverses idées intéressantes, telles que la télécommande Bluetooth et infrarouges, évitement automatique des obstacles et inspection des lignes.

Décrivons le petit véhicule qui nous accompagnera tout au long de notre apprentissage.



## Voici les différentes parties de notre voiture :



Bluetooth module = Module Bluetooth – Battery compartment = Compartiment batteries – UNO R3 core board = Carte Arduino UNO R3 – L298N module = Module L298N – DC MOTOR = Moteur DC – Tires = Pneus – Ultrasonic sensor module and holder = Module ultrason et support – IO Expansion Board = Carte d'extension – SG90 Micro Servo = SG90 servo moteur – Upper acrylic Chassis = Châssis acrylique supérieur – Line tracking module = Module de suivi de ligne – Under Acrylic Chassis = Châssis acrylique inférieur

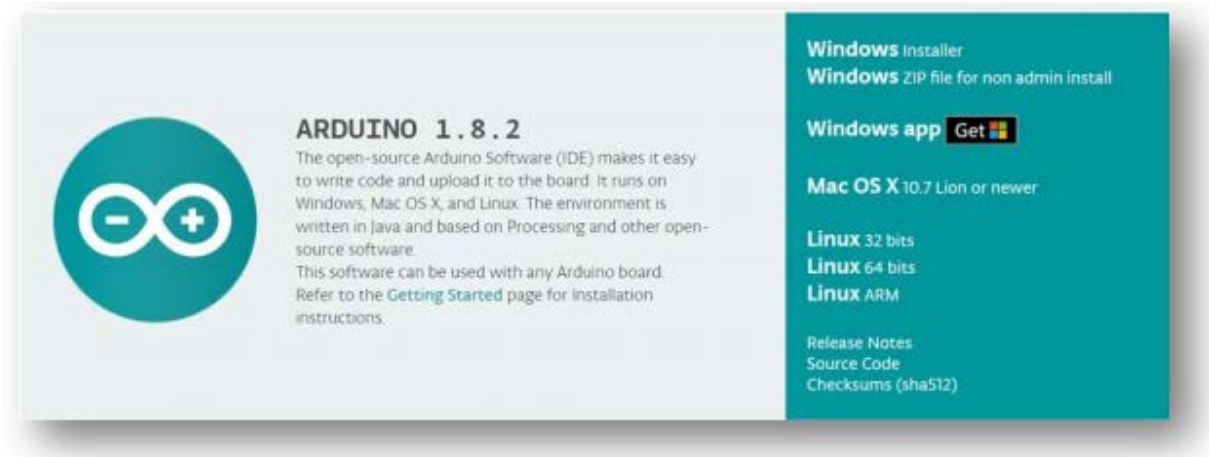
### Fonction de chaque partie:

1. Support de batterie avec un interrupteur: fournir une alimentation électrique pour le véhicule
2. Moteur électrique + roue: conduisez le véhicule pour vous déplacer
3. Plaque acrylique: le châssis de la voiture
4. Carte moteur L298N: entraînez le moteur à tourner les roues
5. La carte contrôleur UNO: le cerveau de la voiture, contrôle toutes les parties
6. Carte d'extension du capteur V5: combinée à l'UNO, rend la connexion plus facile
7. Plate-forme servo moteur : permet au capteur de distance GP2Y0A21 de faire pivoter 180 degrés
8. Module de capteur à ultrasons: mesure de distance afin d'éviter les obstacles
9. Module de suivi de ligne: capteur noir et blanc pour la reconnaissance des voies blanches et noires
10. Récepteur infrarouge et télécommande: fournissez la fonction de contrôle à distance infrarouge
11. Module Bluetooth: fournissez la fonction de contrôle Bluetooth

## II – Charger le programme

Chaque mouvement du véhicule est contrôlé par le programme. Il soit nécessaire d'installer et de configurer correctement le programme. Pour cela, nous utiliserons l'IDE du logiciel Arduino (Integrated Development Environment) comme outil de programmation.

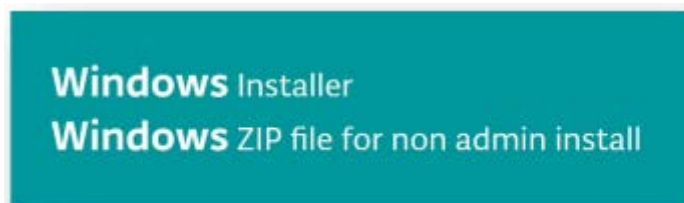
ÉTAPE 1: Passez à <https://www.arduino.cc/fr/Main/Software> et trouvez la page ci-dessous.



La version disponible sur ce site est habituellement la dernière version, et la version actuelle peut être plus récente que la version de l'image.

STEP2: Téléchargez le logiciel de développement adapté au système d'exploitation de votre ordinateur. Prenez Windows par exemple.

Vous pouvez l'installer à l'aide du package d'installation EXE.



Voici la mise en œuvre exe des procédures d'installation.

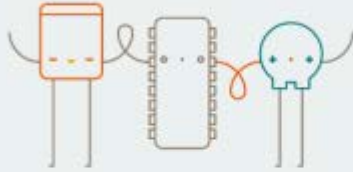
Appuyez sur "Windows Installer"



Appuyez sur le bouton "JUST DOWNLOAD" pour télécharger le logiciel.

## Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **8,808,272** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

\$25

\$50

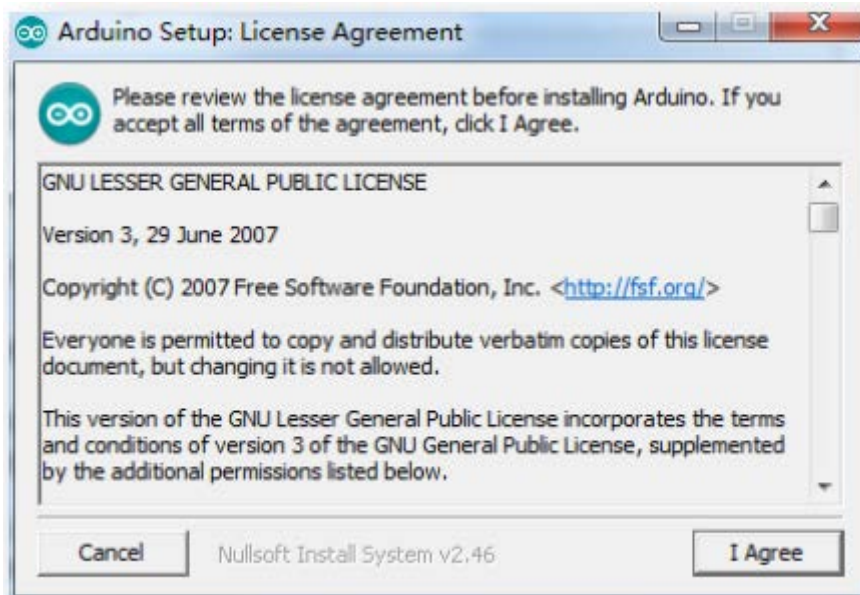
OTHER

JUST DOWNLOAD

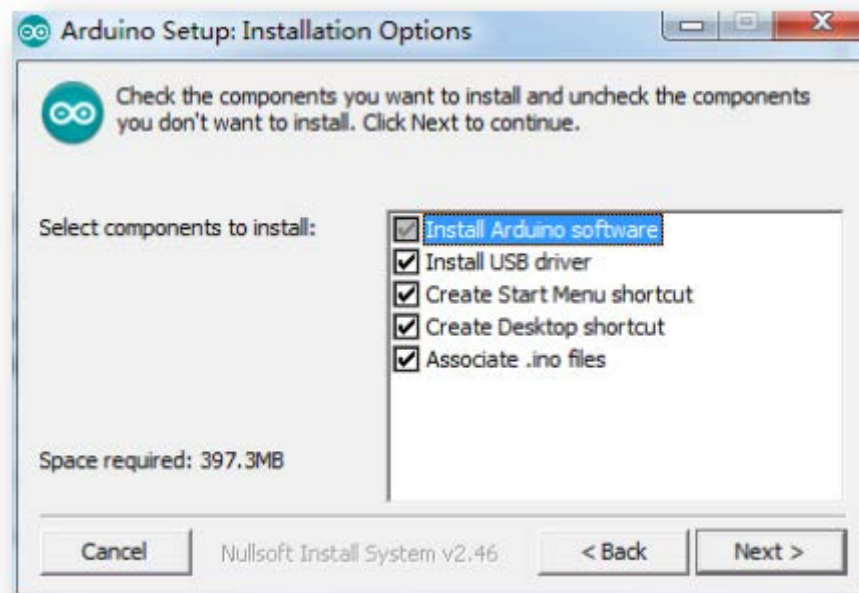
CONTRIBUTE &amp; DOWNLOAD

# JUST DOWNLOAD

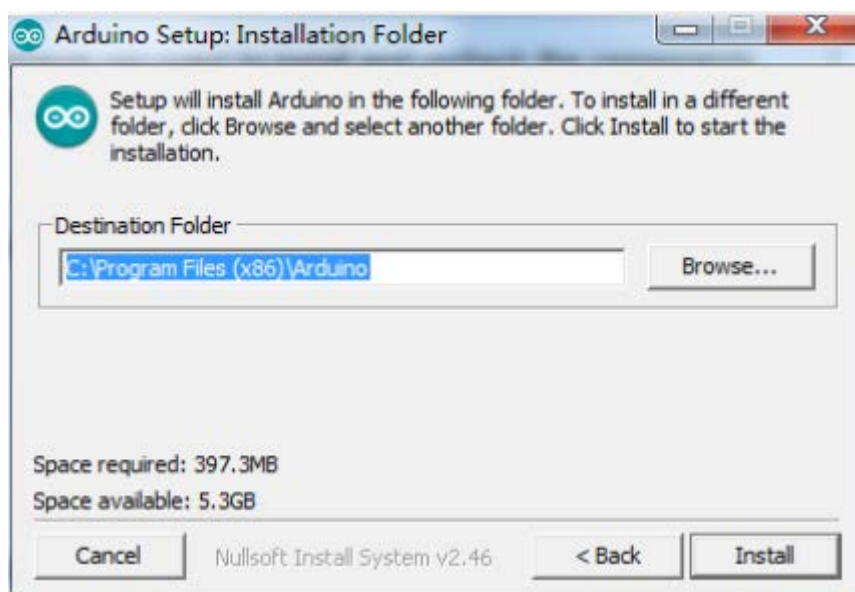
Ceux-ci sont disponibles dans les outils que nous fournissons, et les versions de nos outils sont les derniers en date lorsque ce cours a été réalisé.



Cliquer sur **I Agree**

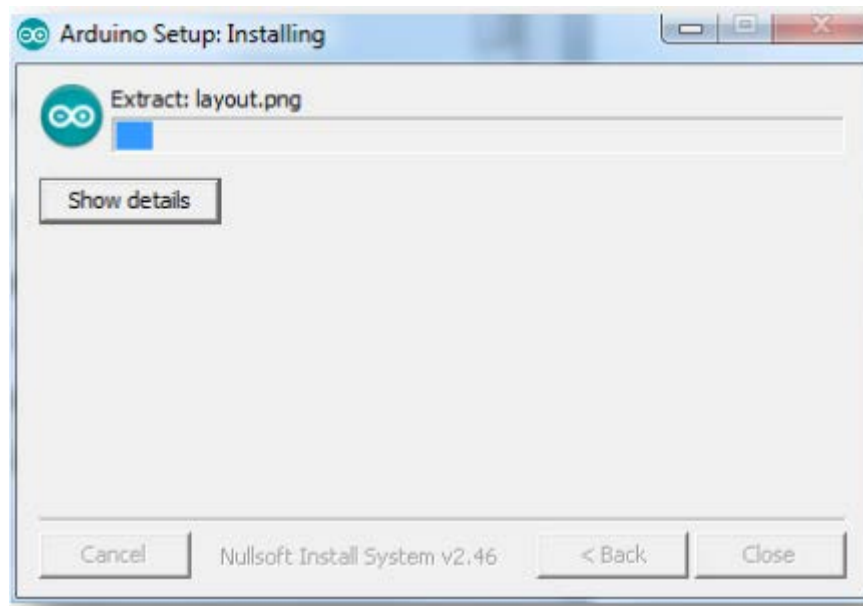


Cliquer sur **next**

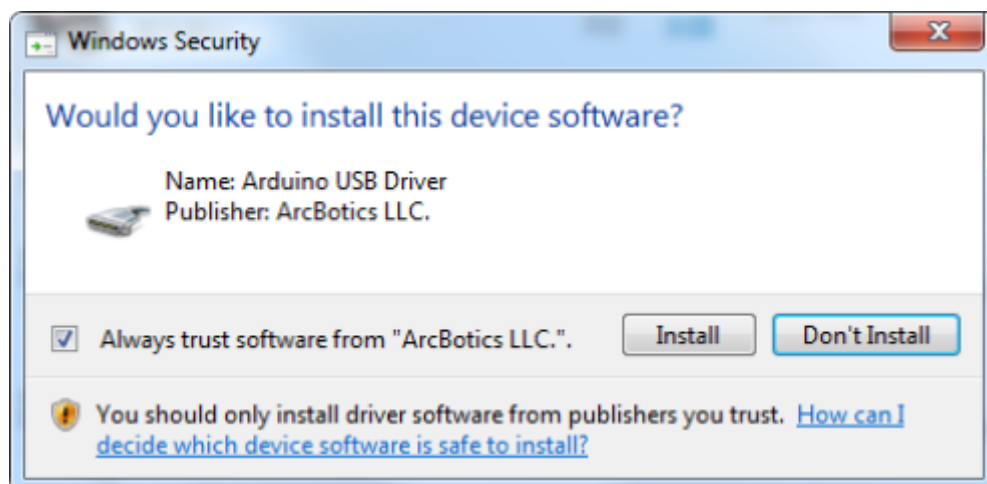


Cliquer sur **install**





A la fin, cliquer sur **Close**

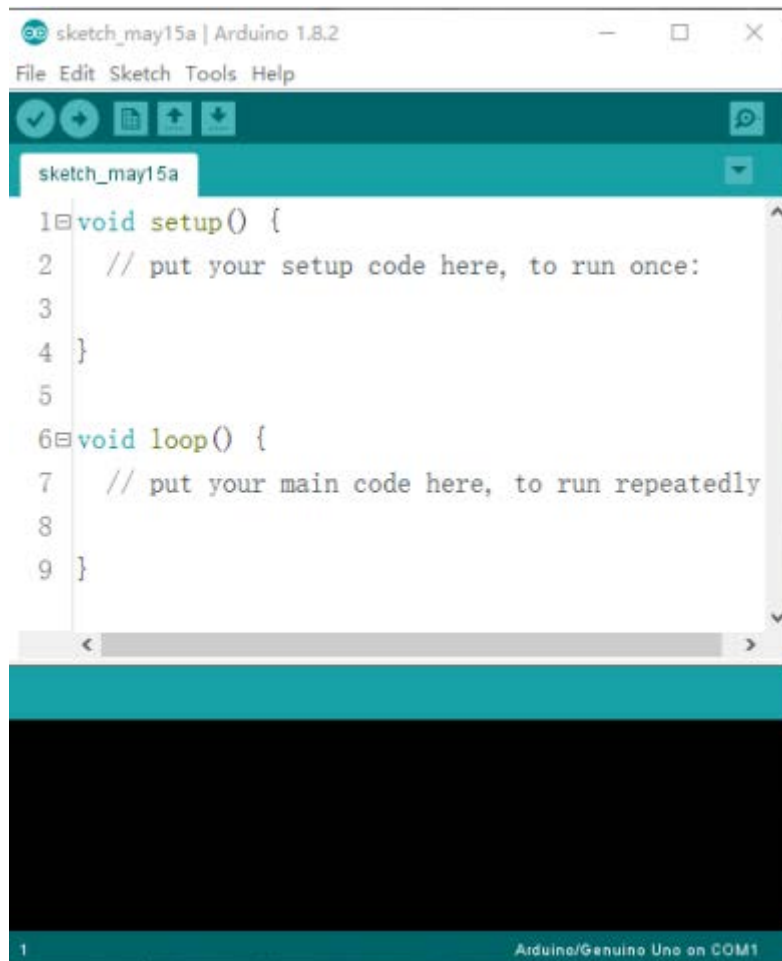


Installation des drivers Arduino : Cliquer sur **Install**

Ensuite, l'icône suivante apparaît sur le bureau



Double-cliquez pour entrer dans l'environnement de développement



```
sketch_may15a | Arduino 1.8.2
File Edit Sketch Tools Help

sketch_may15a

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly
8
9 }
```

1 Arduino/Genuino Uno on COM1

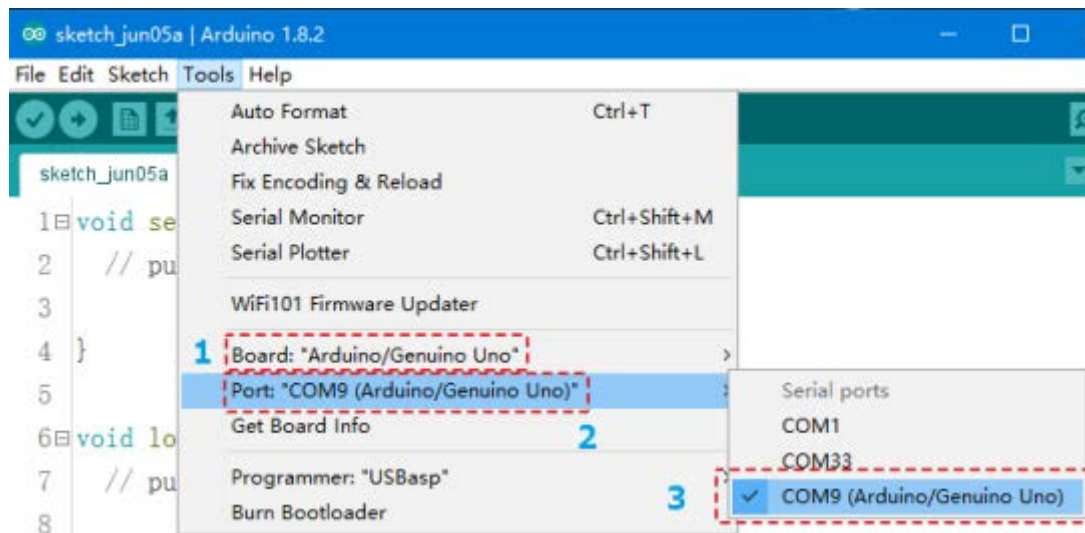
STEP3: Connectez la voiture à l'ordinateur.



ÉTAPE 4: ouvrez l'IDE Arduino. Sélectionnez "Outil" -> "Conseil:" -> "Arduino / Genuino Uno".  
Sélectionnez "Outil" -> "Port:" -> "COM (Arduino / Genuino Uno)".

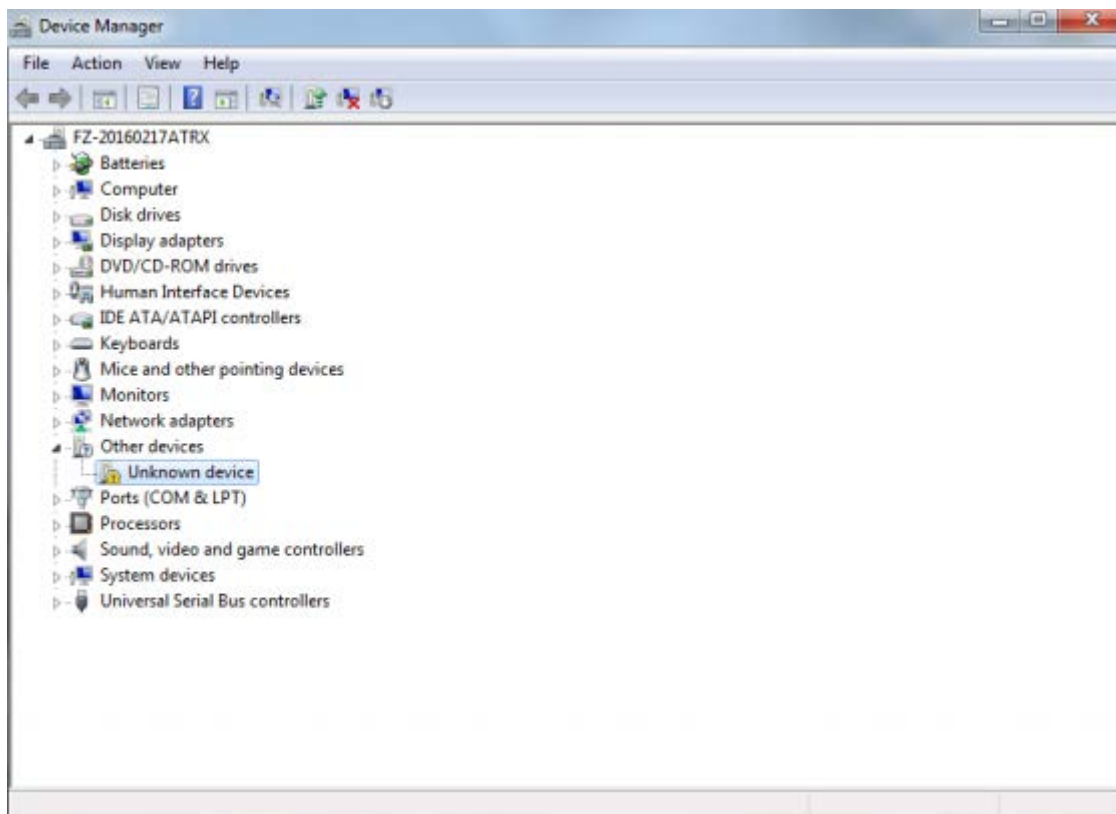


Chaque carte Arduino Uno a un port COM différent sur le même ordinateur et habituellement le numéro COM avec un nom suffixe "(Arduino / Genuino Uno)" dans Arduino 1.8.2. Vous devez choisir le numéro COM de l'affichage actuel.

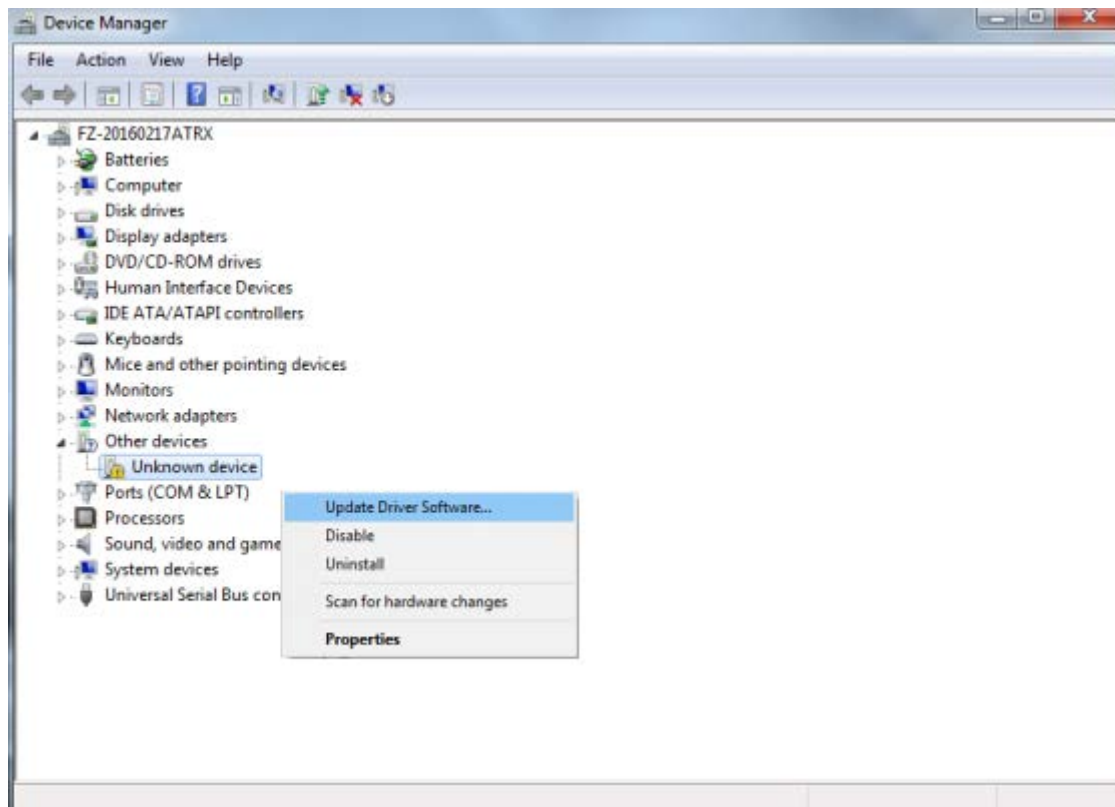


Si vous voyez le port "COM (Arduino / Genuino Uno)", cela signifie que le véhicule a été correctement connecté à l'ordinateur. Dans ce cas, vous pouvez passer directement à STEP 5. Sinon, vous devez installer le pilote de la manière suivante.

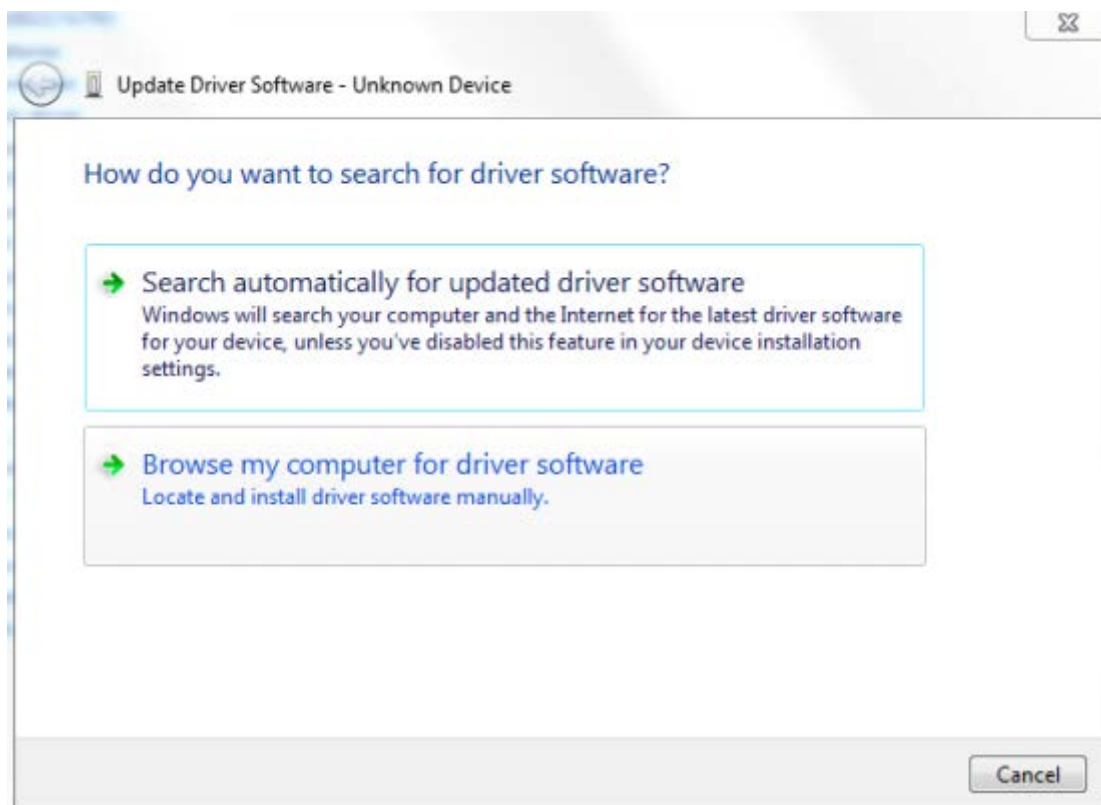
Ouvrez le Gestionnaire de périphériques en cliquant avec le bouton droit sur Mon ordinateur - Gestion - Gestionnaire de périphériques



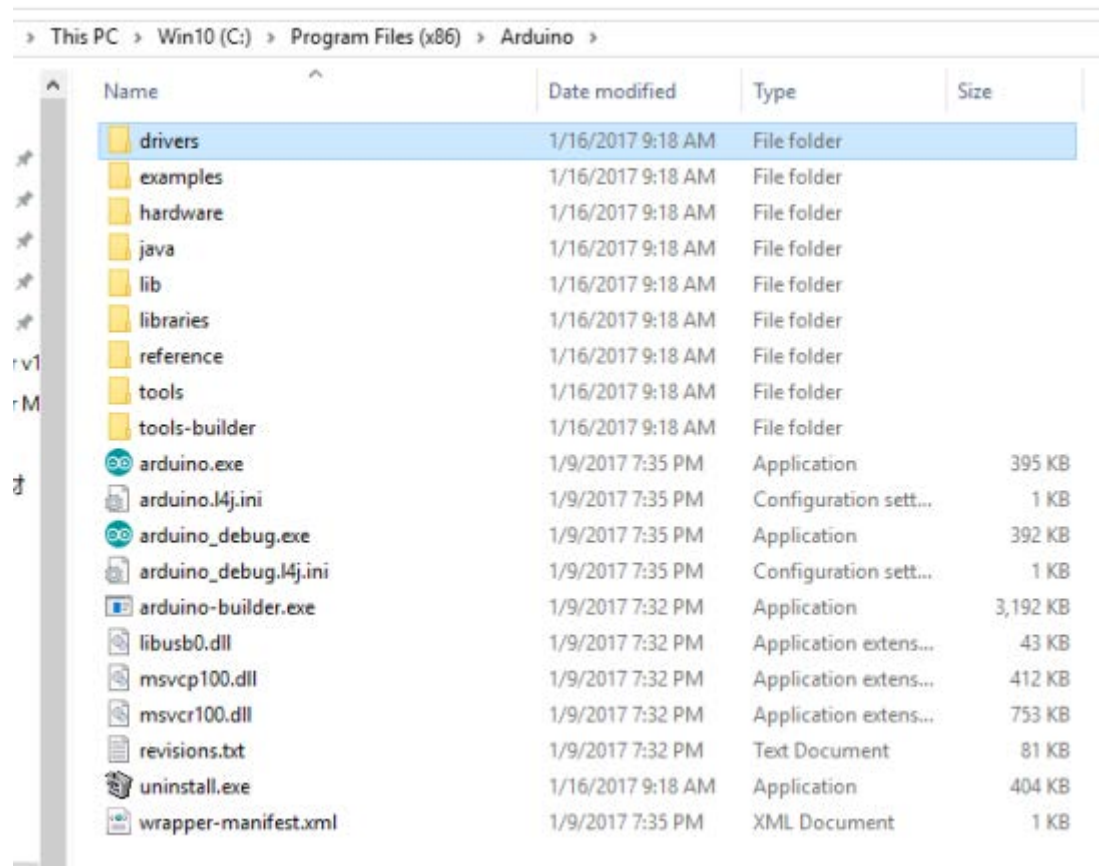
Clic droit sur un dispositif inconnu - mise à jour du logiciel de l'appareil



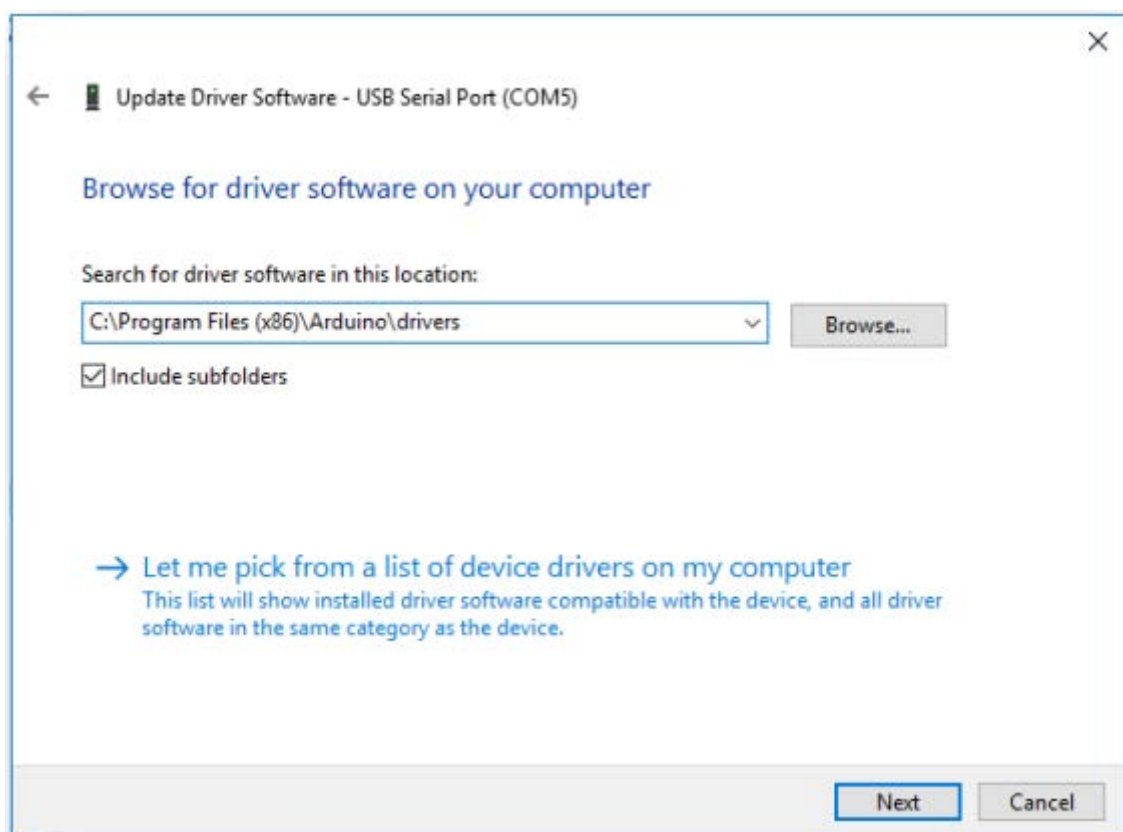
Cela montre que le pilote n'a pas été installé, et vous devez cliquer sur Parcourir mon ordinateur pour trouver les pilotes. Les lecteurs se trouvent dans le dossier Arduino. Normalement, vous installez le dossier dans C:\Program Files (x86)\Arduino.



Répertoire d'installation Arduino

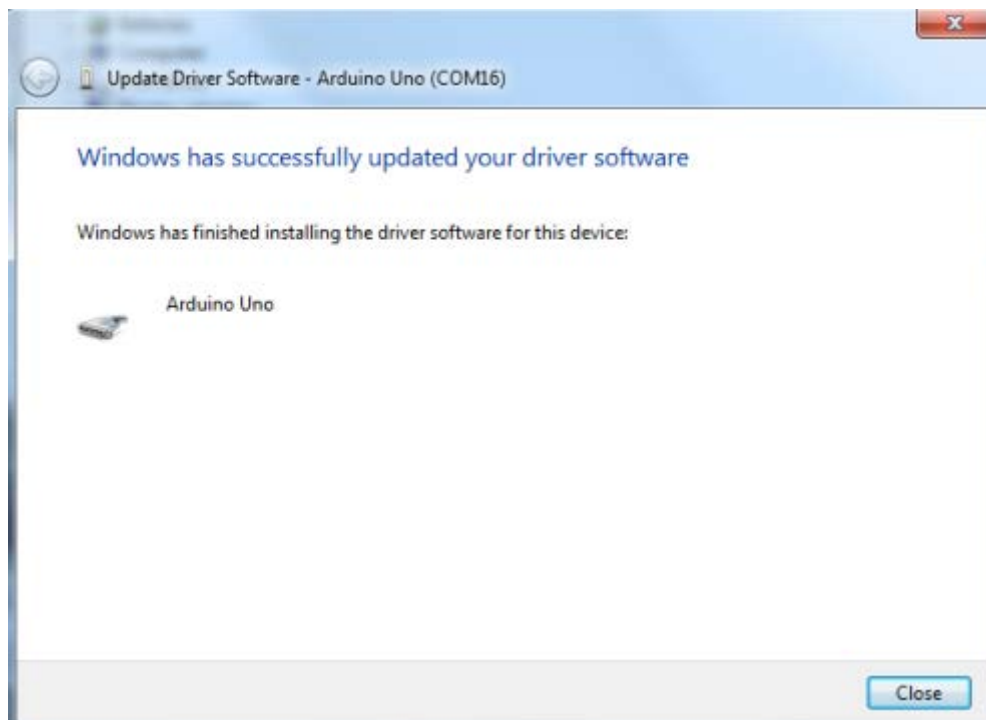
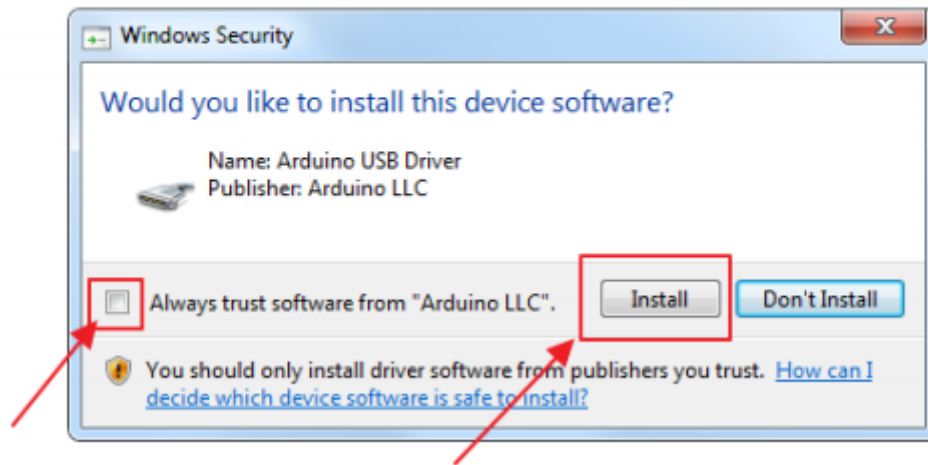


Sélectionnez le dossier du pilote Arduino

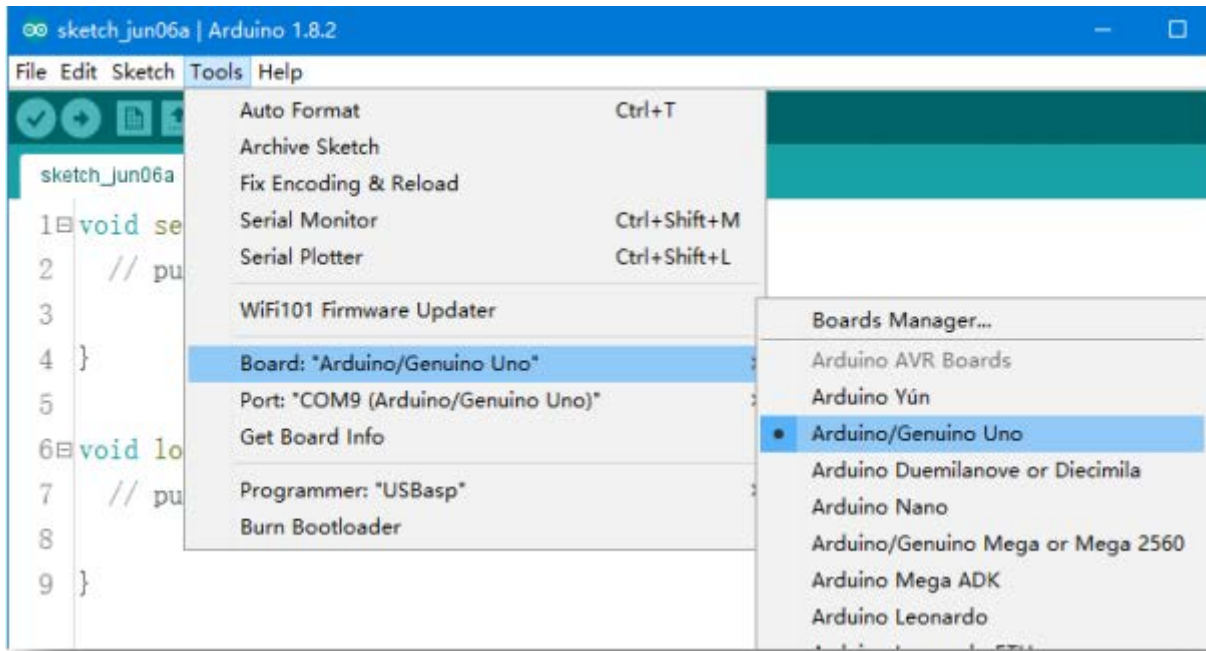


Installer le périphérique USB Arduino

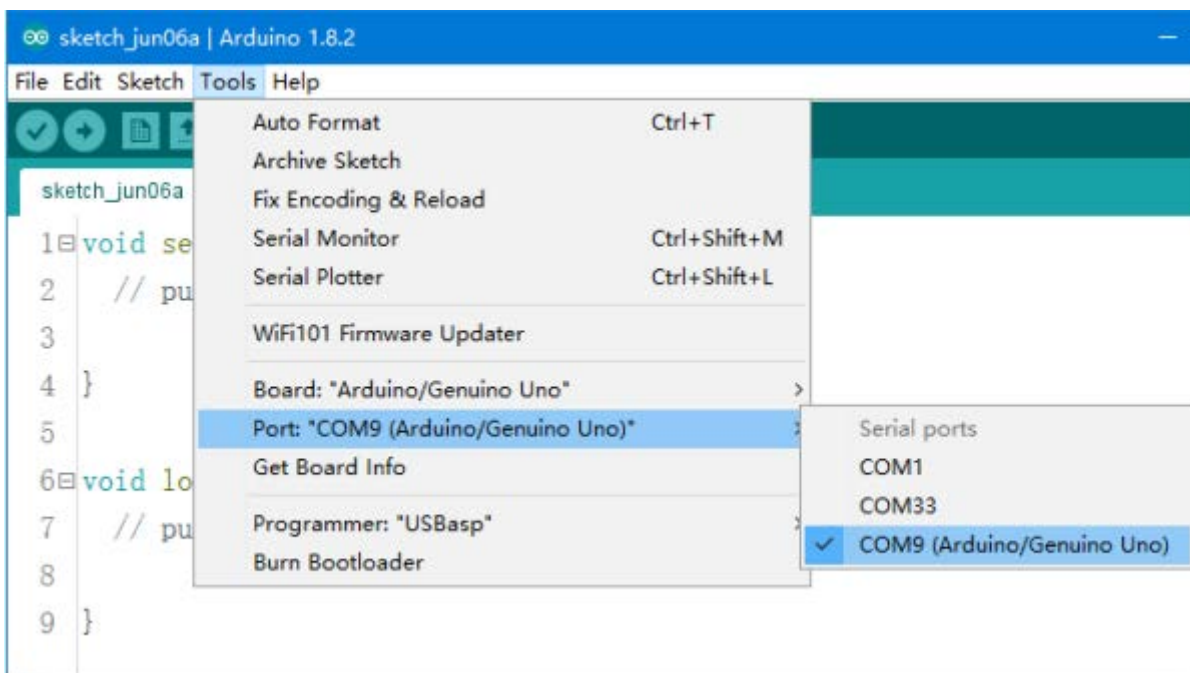
Installing driver software...



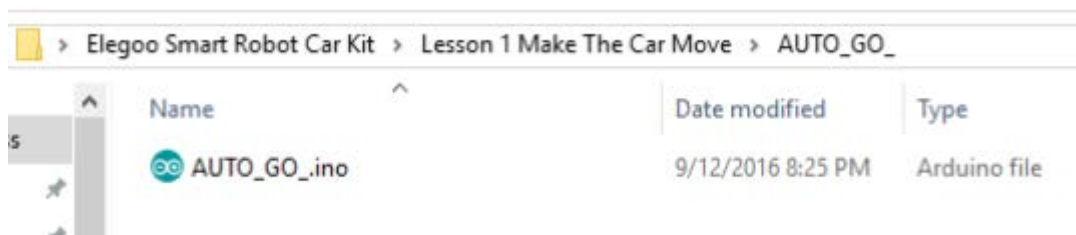
STEP5: Une fois le pilote installé, ouvrez l'IDE, puis cliquez sur "Outils" -> "Conseil" -> "Arduino / Genuino Uno".

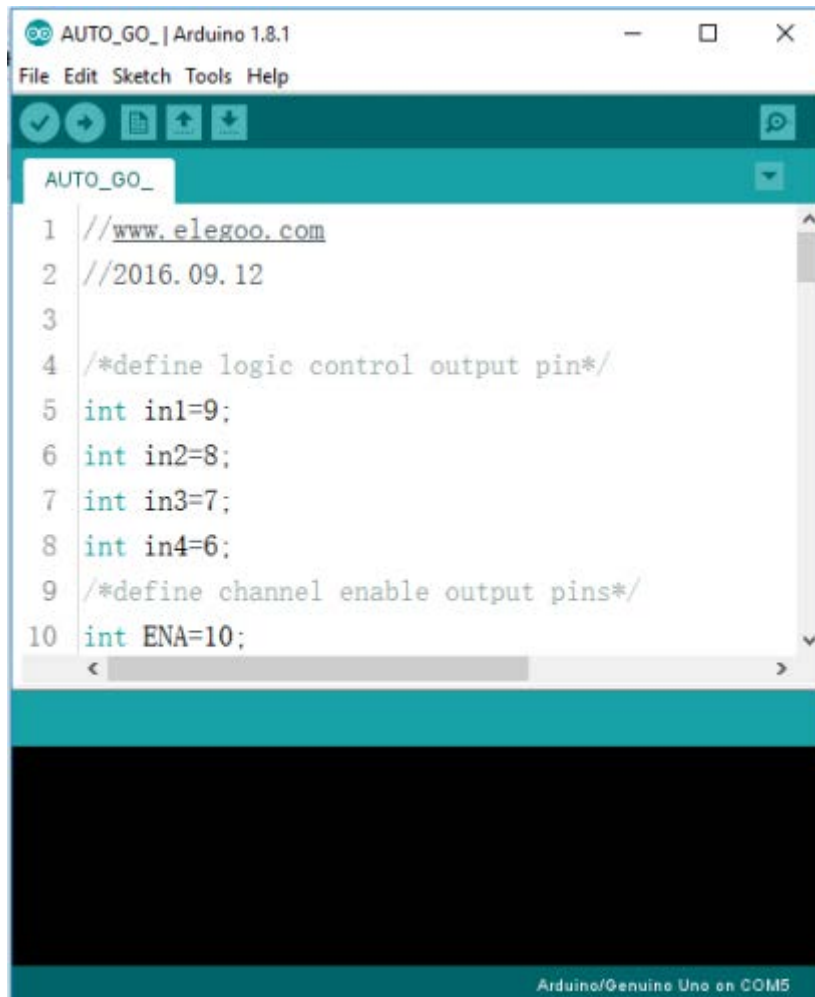


STEP6: Click "Tools" -> "Port" -> COM.



STEP7: Ouvrez le fichier de code dans le répertoire "\\ Lesson 1 Faire bouger la Voiture\\AUTO\_GO\_\\AUTO\_GO\_.ino" et téléchargez-le sur la carte contrôleur UNO. (CONSEILS: Le module bluetooth doit être retiré à chaque fois que vous téléchargez le programme, ou il sera impossible de télécharger le programme.)

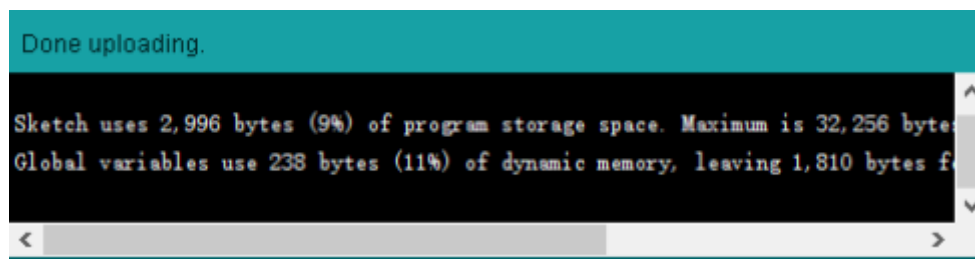




The screenshot shows the Arduino IDE interface with the sketch named "AUTO\_GO\_". The code is as follows:

```
1 //www.elegoo.com
2 //2016.09.12
3
4 /*define logic control output pin*/
5 int in1=9;
6 int in2=8;
7 int in3=7;
8 int in4=6;
9 /*define channel enable output pins*/
10 int ENA=10;
```

The status bar at the bottom indicates "Arduino/Genuino Uno en COM5".



The screenshot shows the "Done uploading." message in the Arduino IDE. Below the message, the following information is displayed:

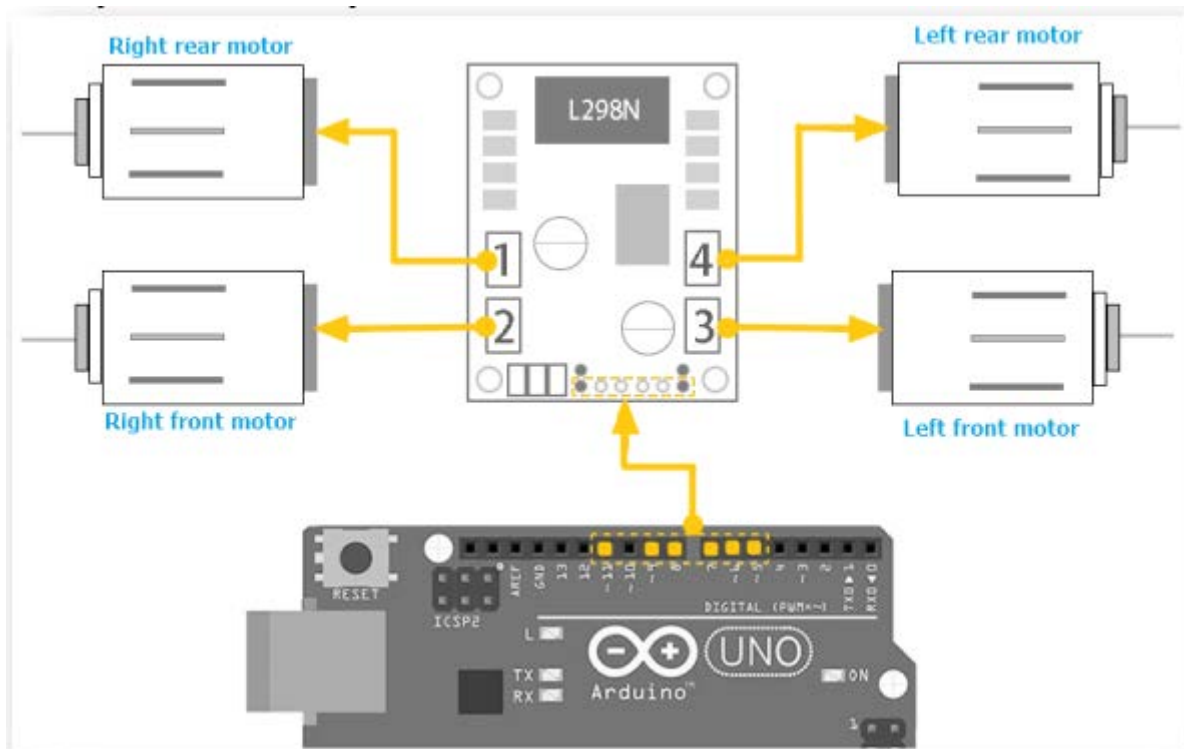
```
Sketch uses 2,996 bytes (9%) of program storage space. Maximum is 32,256 bytes.
Global variables use 238 bytes (11%) of dynamic memory, leaving 1,810 bytes free.
```

L'image ci-dessus montre qu'elle est téléchargée avec succès.

ÉTAPE 8: Voyons les résultats. Téléchargez le programme sur la carte contrôleur UNO. Après avoir débranché la voiture sur l'ordinateur, vous pouvez allumer l'interrupteur d'alimentation et mettre la voiture sur le sol. Ensuite, vous verrez le déplacement de la voiture.



### III – Description des principes utilisés



#### Comment utiliser le panneau de conduite du moteur L298N

La définition des ports de connexion sur la carte L298N a été indiquée ci-dessus. Les moteurs doivent être connectés à la carte L298N comme l'image ci-dessus, et si vous trouvez que la direction de rotation d'un des moteurs est opposée, changez la position de connexion de ses fils noirs et rouges.

L298N GND est connecté à la batterie GND;

L298N VCC est connecté au boîtier de batterie VCC;

La carte UNO est également connectée au boîtier de la batterie.

L298N 5V ici ne peut pas être connecté à UNO 5V;

ENA et ENB contrôlent la vitesse du moteur droit et la vitesse du moteur gauche séparément par PWM.

IN1, IN2, IN3, IN4: IN1 et IN2 sont utilisés pour contrôler le moteur gauche, IN3 et IN4 sont utilisés pour contrôler le moteur droit. À propos du principe, regardez la fiche ci-dessous: (Nous prenons le moteur gauche par exemple)

ENA	IN1	IN2	DC MOTOR STATUS
0	X	X	STOP
1	0	0	BRAKING
1	1	0	FORWARD
1	0	1	BACKWARD
1	1	1	BARKING

## IV – Faire bouger la voiture

Nous allons essayer de déplacer le moteur sans contrôler la vitesse. Parce qu'il est facile d'écrire un programme sans contrôle de la vitesse.

Tout d'abord, voyons la connexion du moteur à la carte L298N, nous utiliserons Arduino 5, 6, 7, 8, 9, 11 broches pour contrôler la voiture. 9 et 11 broches contrôlent la roue droite. 7 et 8 broches contrôlent la roue gauche. 5 et 6 broches contrôlent ENA et ENB.

La connexion est donc ci-dessous:

L298N	V5 expansion board
ENA	5
ENB	6
IN1	7
IN2	8
IN3	9
IN4	11

Sur la base de la feuille ci-dessus, nous concevons tout d'abord un programme simple pour que la rotation de la roue droite soit 0.5s en sens positif, arrête 0.5s, tourne 0.5s dans le sens négatif et arrête 0.5s. Et la roue répétera la réaction.

Connectez la carte contrôleur UNO à l'ordinateur, ouvrez le fichier de code dans le chemin "\ Lesson 1 Faire bouger la voiture \ right\_wheel\_rotation \ right\_wheel\_rotation.ino". Téléchargez le programme dans l'IDE UNO.

**Voyons ce code :**

```
//www.elegoo.com
// Right motor truth table
//Here are some handy tables to show the various modes of operation.
// ENB IN3 IN4 Description
// LOW Not Applicable Not Applicable Motor is off
// HIGH LOW LOW Motor is stopped (brakes)
// HIGH LOW HIGH Motor is on and turning forwards Http://www.elegoo.com
```

```
// HIGH HIGH LOW Motor is on and turning backwards
// HIGH HIGH HIGH Motor is stopped (brakes)
// define IO pin
#define ENB 6
#define IN3 9
#define IN4 11
//init the car
void setup() {
  pinMode(IN3, OUTPUT); //set IO pin mode OUTPUT
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
  digitalWrite(ENB, HIGH); //Enable right motor
}
//main loop
void loop() {
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH); //Right wheel turning forwards
  delay(500); //delay 500ms
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, LOW); //Right wheel stoped
  delay(500);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW); //Right wheel turning backwards
  delay(500);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH); //Right wheel stoped
  delay(500);
}
```

Débranchez-le de l'ordinateur, puis allumez l'alimentation de la voiture. Vous verrez que la roue droite se déplace comme prévu.

Si la voiture ne bouge pas, appuyez sur le bouton de réinitialisation sur la carte UNO.

Si la direction de déplacement du moteur est différente de celle que vous avez définie, vous pouvez changer la connexion des lignes noires et rouges du moteur au panneau L298N.

Ensuite, nous faisons tourner la roue gauche de la même manière.

Connectez la carte contrôleur UNO à l'ordinateur, ouvrez le fichier de code dans le chemin d'accès "Leçon 1 Faire bouger la voiture \ Left\_wheel\_rotation \ Left\_wheel\_rotation.ino". Téléchargez le programme dans le tableau UNO.

```
//www.elegoo.com
// Left motor truth table
//Here are some handy tables to show the various modes of operation.
// ENA IN1 IN2 Description
// LOW Not Applicable Not Applicable Motor is off
// HIGH LOW LOW Motor is stopped (brakes)
// HIGH HIGH LOW Motor is on and turning forwards
// HIGH LOW HIGH Motor is on and turning backwards
// HIGH HIGH HIGH Motor is stopped (brakes)
// define IO pin
#define ENA 5
#define IN1 7
#define IN2 8
//init the car
void setup() {
  pinMode(IN1, OUTPUT); //set IO pin mode OUTPUT
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);
  digitalWrite(ENA, HIGH); //Enable left motor
}
//main loop
void loop() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW); //Right wheel turning forwards
  delay(500); //delay 500ms
  digitalWrite(IN1, LOW);
```

```
digitalWrite(IN2, LOW); //Right wheel stoped
delay(500);
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH); //Right wheel turning backwards
delay(500);
digitalWrite(IN1, HIGH);
digitalWrite(IN2, HIGH); //Right wheel stoped
delay(500);
}
```

Débranchez-le de l'ordinateur, puis allumez l'alimentation de la voiture. Vous verrez que la roue droite se déplace comme prévu.

Après avoir terminé le débogage de la voiture, vous pouvez écrire des programmes pour faire bouger la voiture.

Voici la façon dont la voiture se déplace:

CAR	forward	back	stop
Left wheel	Forward	back	stop
Right wheel	Forward	back	stop

CAR	Turn left	Turn right	stop
Left wheel	back	Forward	Stop
Right wheel	forward	back	stop

Ensuite, nous écrirons un programme simple pour faire avancer la voiture 0.5s, puis arrêter 0.5s, puis sauvegarder 0.5s puis arrêter 0.5s.

```
//www.elegoo.com
// Left motor truth table
// ENA IN1 IN2 Description
// LOW Not Applicable Not Applicable Motor is off
// HIGH LOW LOW Motor is stopped (brakes)
// HIGH HIGH LOW Motor is on and turning forwards
// HIGH LOW HIGH Motor is on and turning backwards
// HIGH HIGH HIGH Motor is stopped (brakes)
// Right motor truth table
// ENB IN3 IN4 Description
// LOW Not Applicable Not Applicable Motor is off
```

```
// HIGH LOW LOW Motor is stopped (brakes)
// HIGH LOW HIGH Motor is on and turning forwards
// HIGH HIGH LOW Motor is on and turning backwards
// HIGH HIGH HIGH Motor is stopped (brakes)
// The direction of the car's movement
// Left motor Right motor Description
// stop(off) stop(off) Car is stopped
// forward forward Car is running forwards
// forward backward Car is turning right
// backward forward Car is turning left
// backward backward Car is running backwards
//define the L298n IO pin
#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11
void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
}
void loop() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH); //go forward
```



```
delay(500);
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW); //stop
delay(500);
digitalWrite(IN1, LOW);
digitalWrite(IN2, HIGH);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW); //go back
delay(500);
digitalWrite(IN1, LOW);
digitalWrite(IN2, LOW);
digitalWrite(IN3, HIGH);
digitalWrite(IN4, HIGH); //stop
delay(500);
}
```

Téléchargez le programme sur l'arduino UNO, déconnectez-le de l'ordinateur, puis allumez l'alimentation de la voiture. Vous verrez que la roue droite se déplace comme prévu.

### La troisième étape: Écrivez le programme

Il peut être difficile pour vous d'écrire l'ensemble du programme pour que la voiture se déplace automatiquement. Nous séparons donc les mouvements en différentes fonctions, par exemple en avançant et en tournant à gauche. Et lorsque nous écrivons le programme en dernière étape, nous pouvons appeler la fonction. Ensuite, nous commençons à écrire des programmes pour chaque mouvement:

```
void forward(){
digitalWrite(ENA,HIGH); //enable L298n A channel
digitalWrite(ENB,HIGH); //enable L298n B channel
digitalWrite(IN1,HIGH); //set IN1 high level
digitalWrite(IN2,LOW); //set IN2 low level
digitalWrite(IN3,LOW); //set IN3 low level
digitalWrite(IN4,HIGH); //set IN4 high level
Serial.println("Forward");//send message to serial monitor
}
```

```
void back(){
digitalWrite(ENA,HIGH);
digitalWrite(ENB,HIGH);
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
Serial.println("Back");
}

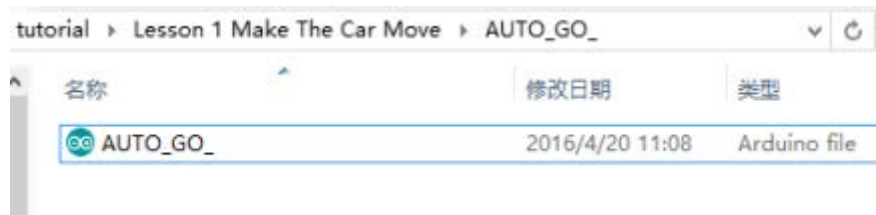
void left(){
digitalWrite(ENA,HIGH);
digitalWrite(ENB,HIGH);
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
Serial.println("Left");
}

void right(){
digitalWrite(ENA,HIGH);
digitalWrite(ENB,HIGH);
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
Serial.println("Right");
}
```

### **La quatrième étape: Déplacez-vous automatiquement**

Nous commençons à écrire un programme pour faire avancer la voiture automatiquement: avancer 0.4s – attendre 0.4s - tourner à gauche 0.4s - tourner à droite 0.4s.

Connectez la carte contrôleur UNO à l'ordinateur, ouvrez le fichier de code dans le répertoire "Leçon 1 Faire bouger la voiture \ AUTO\_GO\_ \ AUTO\_GO\_.ino". Téléchargez le programme dans l'Arduino UNO.



```
//www.elegoo.com
// The direction of the car's movement
// ENA ENB IN1 IN2 IN3 IN4 Description
// HIGH HIGH HIGH LOW LOW HIGH Car is runing forward
// HIGH HIGH LOW HIGH HIGH LOW Car is runing back
// HIGH HIGH LOW HIGH LOW HIGH Car is turning left
// HIGH HIGH HIGH LOW HIGH LOW Car is turning right
// HIGH HIGH LOW LOW LOW LOW Car is stoped
// HIGH HIGH HIGH HIGH HIGH HIGH Car is stoped
// LOW LOW N/A N/A N/A N/A Car is stoped

//define L298n module IO Pin
#define ENA 5
#define ENB 6
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11
void forward(){
digitalWrite(ENA,HIGH); //enable L298n A channel
digitalWrite(ENB,HIGH); //enable L298n B channel
digitalWrite(IN1,HIGH); //set IN1 hight level
digitalWrite(IN2,LOW); //set IN2 low level
digitalWrite(IN3,LOW); //set IN3 low level
digitalWrite(IN4,HIGH); //set IN4 hight level
Serial.println("Forward");//send message to serial monitor
```

```
}  
void back(){  
  digitalWrite(ENA,HIGH);  
  digitalWrite(ENB,HIGH);  
  digitalWrite(IN1,LOW);  
  digitalWrite(IN2,HIGH);  
  digitalWrite(IN3,HIGH);  
  digitalWrite(IN4,LOW);  
  Serial.println("Back");  
}  
void left(){  
  digitalWrite(ENA,HIGH);  
  digitalWrite(ENB,HIGH);  
  digitalWrite(IN1,LOW);  
  digitalWrite(IN2,HIGH);  
  digitalWrite(IN3,LOW);  
  digitalWrite(IN4,HIGH);  
  Serial.println("Left");  
}  
void right(){  
  digitalWrite(ENA,HIGH);  
  digitalWrite(ENB,HIGH);  
  digitalWrite(IN1,HIGH);  
  digitalWrite(IN2,LOW);  
  digitalWrite(IN3,HIGH);  
  digitalWrite(IN4,LOW);  
  Serial.println("Right");  
}  
//before execute loop() function,  
//setup() function will execute first and only execute once  
void setup() {  
  Serial.begin(9600);//open serial and set the baudrate  
  pinMode(IN1,OUTPUT);//before using io pin, pin mode must be set first
```

```

pinMode(IN2,OUTPUT);
pinMode(IN3,OUTPUT);
pinMode(IN4,OUTPUT);
pinMode(ENA,OUTPUT);
pinMode(ENB,OUTPUT);
}
//Repeat execution
void loop() {
forward(); //go forward
delay(1000); //delay 1000 ms
back(); //go back
delay(1000);
left(); //turning left
delay(1000);
right(); //turning right
delay(1000);
}


```

Débranchez-le de l'ordinateur, puis allumez l'alimentation de la voiture. Vous verrez que la roue droite se déplace comme prévu.

### La cinquième étape: contrôle de vitesse

Le code a pour but de contrôler la vitesse de la voiture: avancer et réduire la vitesse -> arrêter 1s -> avancer et accélérer -> arrêter 2s.

Connectez la carte contrôleur UNO à l'ordinateur, ouvrez le fichier de code dans le répertoire "Leçon 1 Faites bouger la voiture \ speed\_control \ speed\_control.ino". Téléchargez le programme dans le tableau UNO.

Elegoo Smart Robot Car Kit V3.0 > Lesson 1 Make The Car Move > speed_control			
	名称	修改日期	类型
材	 speed_control.ino	2017/5/11 15:35	Arduino file

```
//www.elegoo.com
```

```
#define ENA 5
```

```
#define ENB 6
```

```
#define IN1 7
#define IN2 8
#define IN3 9
#define IN4 11
void setup() {
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
}
void loop() {
  //go forward
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
  //reduce the speed
  for(int i = 255; i >= 0; i--){
    analogWrite(ENB,i);
    analogWrite(ENA,i);
    delay(20);
  }
  //stop
  analogWrite(ENB,0); //speed = 0
  analogWrite(ENA,0);
  delay(1000);
  //runing back
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
```



```
//accelerate
for(int i = 0; i <= 255; i++){
  analogWrite(ENB,i);
  analogWrite(ENA,i);
  delay(20);
}

//stop
digitalWrite(ENB,LOW); //Motor is off
digitalWrite(ENA,LOW);
delay(2000);
}
```