

# Leçon 1 – Mettre le robot en mouvement

## Points clés de la leçon

Sommaire:

- ◆ Apprendre à utiliser l'IDE arduino (Integrated Development Environment)
- ◆ Transférer le programme pour mettre le robot en mouvement

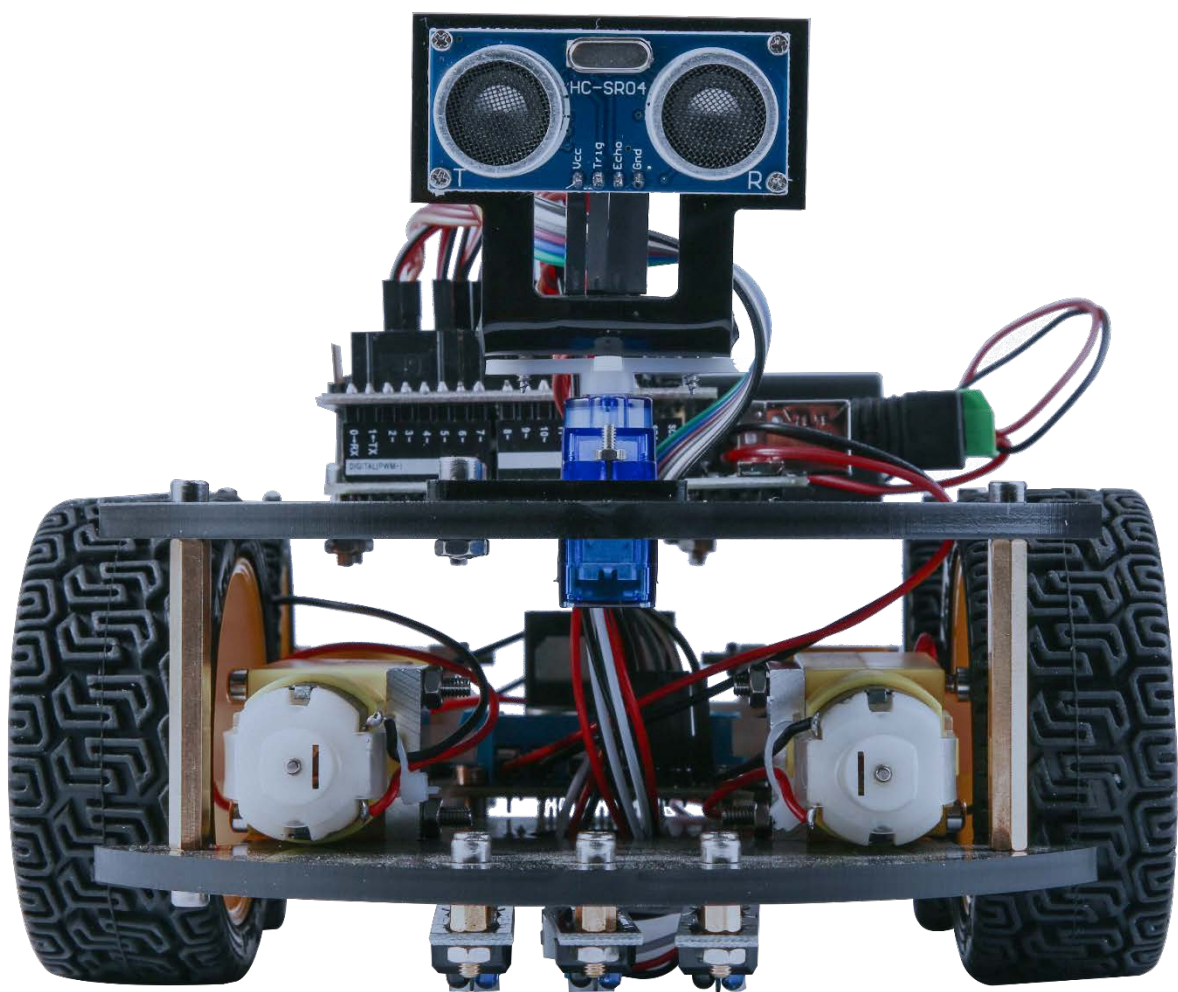
Matériel :

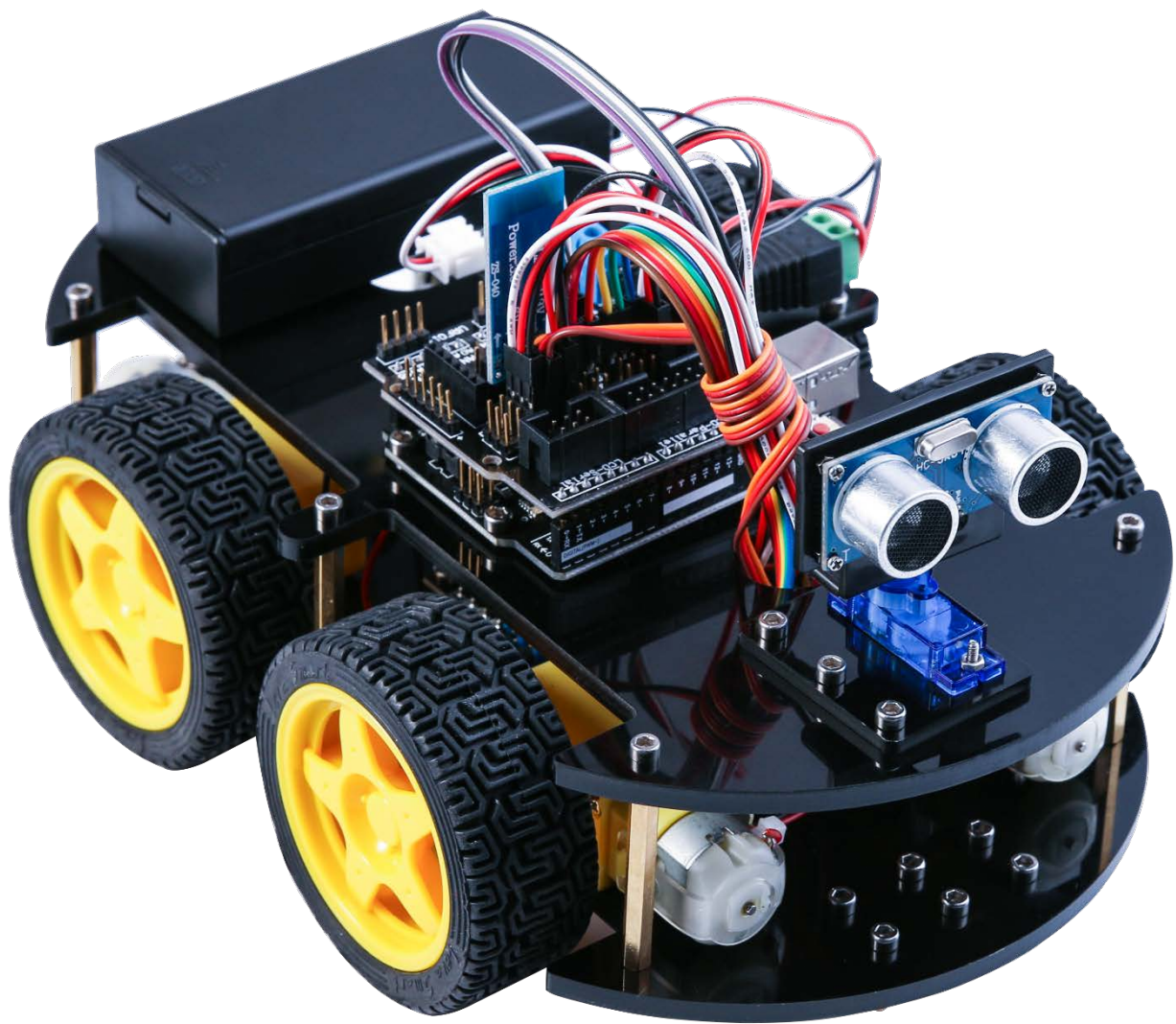
- ◆ Le robot (avec ses piles)
- ◆ Un cable usb

## I. Introduction

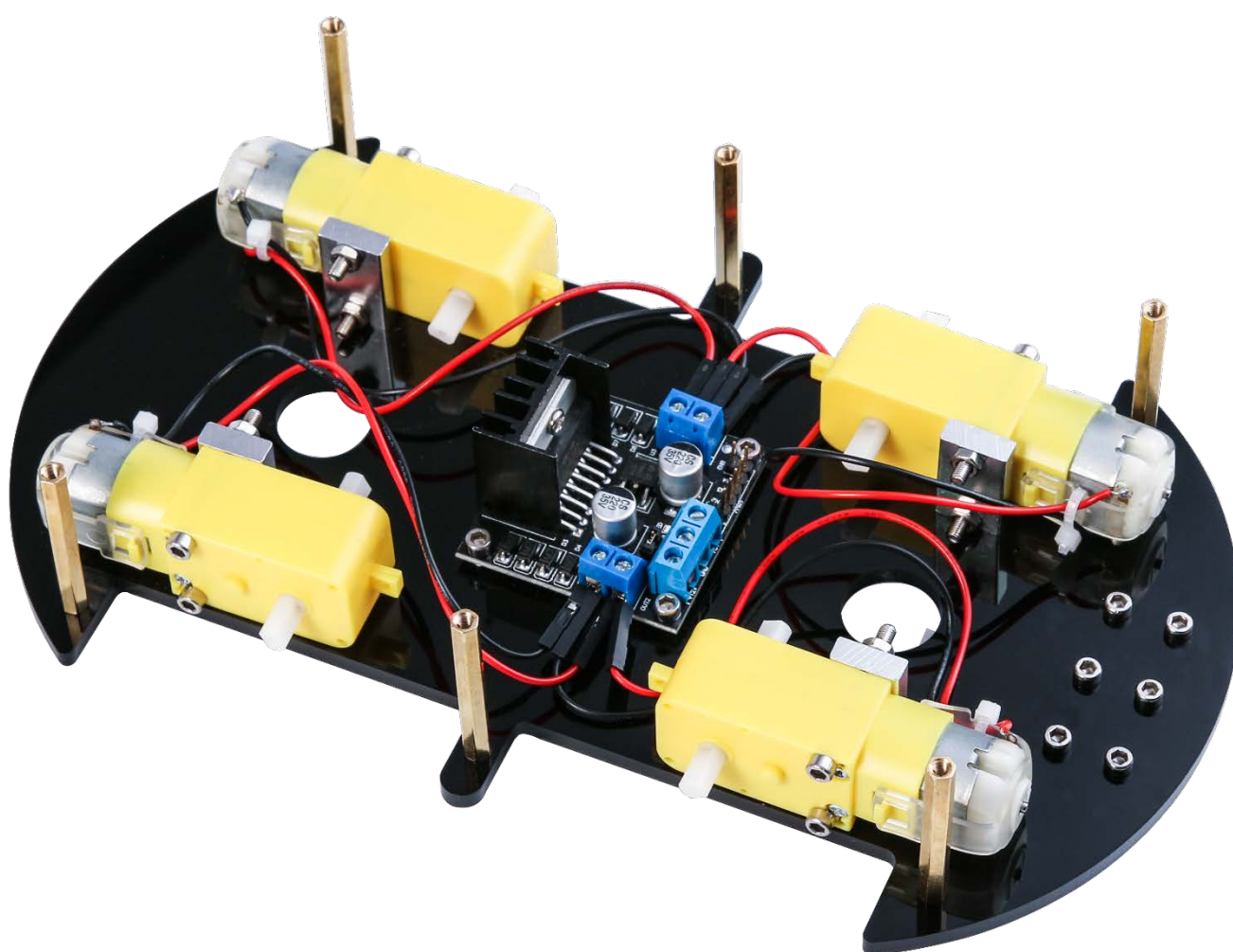
Vous allez pouvoir expérimenter des phénomènes physiques et technologiques comme l'infrarouge, les ultrasons, le Bluetooth dans le but de faire de la commande à distance ou du déplacement automatique en évitement d'obstacle ou suivi de parcours.

Commençons par une présentation des différentes parties du robot, qui va nous en sommes sûrs, vous tenir compagnie un long moment.



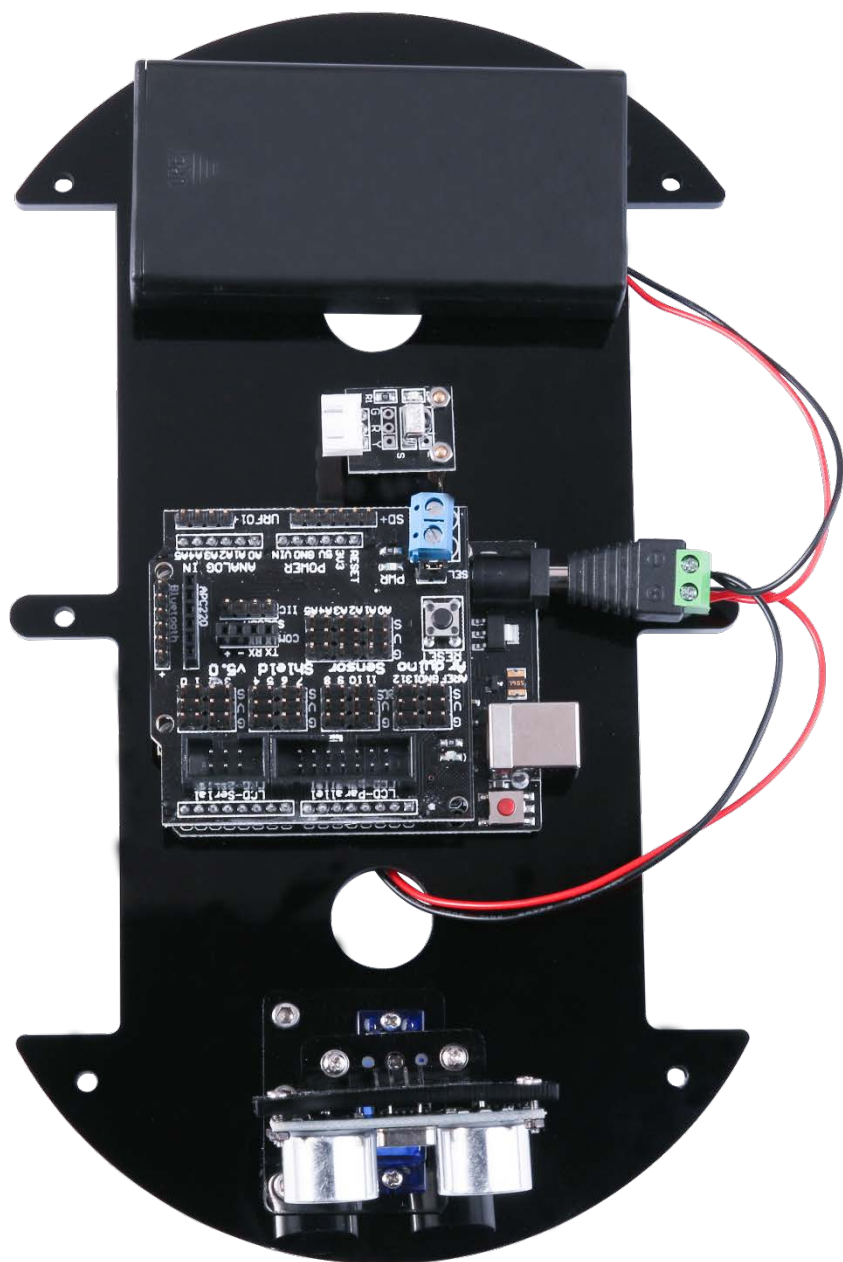


Chaque élément du robot est représenté ci dessous :









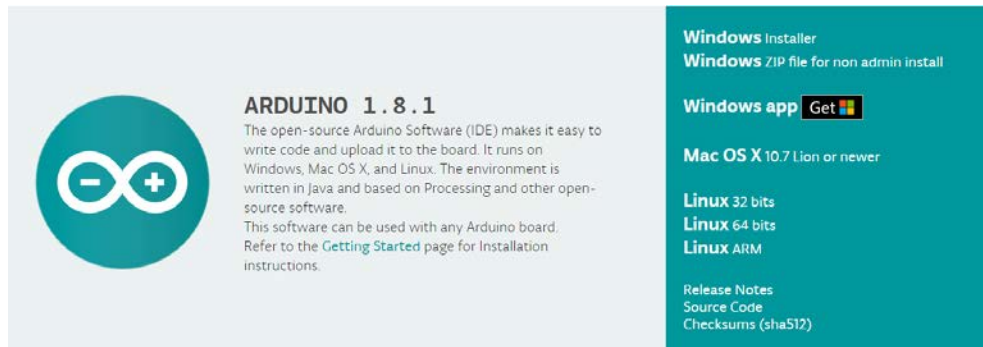
## Fonction de chaque élément:

1. Bloc batterie avec l'interrupteur Marche/Arrêt du robot
2. Les moteurs et roues : permettent au véhicule de se déplacer
3. Les platines acryliques : châssis de la voiture
4. L298N carte de contrôle pour le pilotage des moteurs
5. Carte de contrôle UNO : le cerveau
6. Carte d'extension V5: permet de connecter aisément tous les capteurs
7. Servomoteur SG90 : permet d'orienter le module Ultrasons à 180 degrés
8. Module Ultrasons : permet de faire des mesures de distances
9. Capteurs "Line tracking module": utilisés pour le suivi de lignes noires
10. Récepteur infrarouge et sa télécommande pour un contrôle à distance
11. Module Bluetooth: permet un contrôle du robot avec les appareils mobiles

## II. Transférer le programme

Chaque mouvement du véhicule est contrôlé par le programme. Il est donc nécessaire d'installer correctement le logiciel Arduino sur votre ordinateur pour pouvoir transférer les programmes.

Etape 1: Rendez-vous à l'adresse <https://www.arduino.cc/en/Main/Software> et recherchez la page suivante :



Vous y trouverez la toute dernière version du logiciel Arduino.

Etape 2: Télécharger l'exécutable permettant l'installation du logiciel

# Windows Installer

Procéder à l'installation :


### 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.



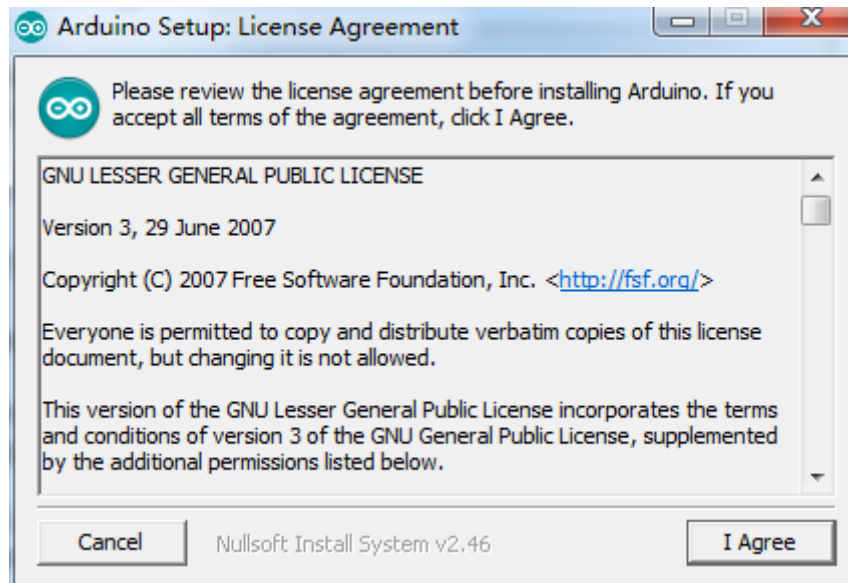
JUST DOWNLOAD

Cliquer sur "JUST DOWNLOAD" pour télécharger le logiciel

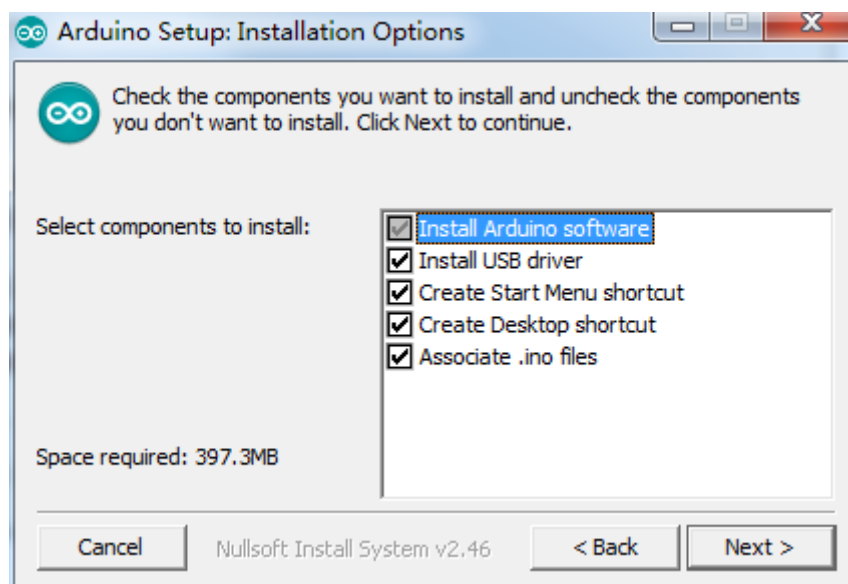
 **arduino-1.8.0-windows.exe**



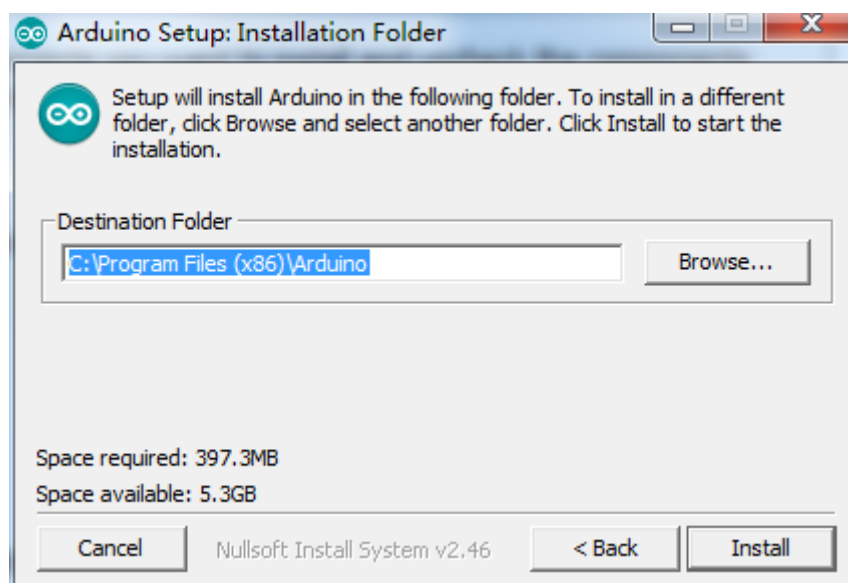
Nous proposons aussi le logiciel à la dernière version sur nos CD fournis avec les kits.



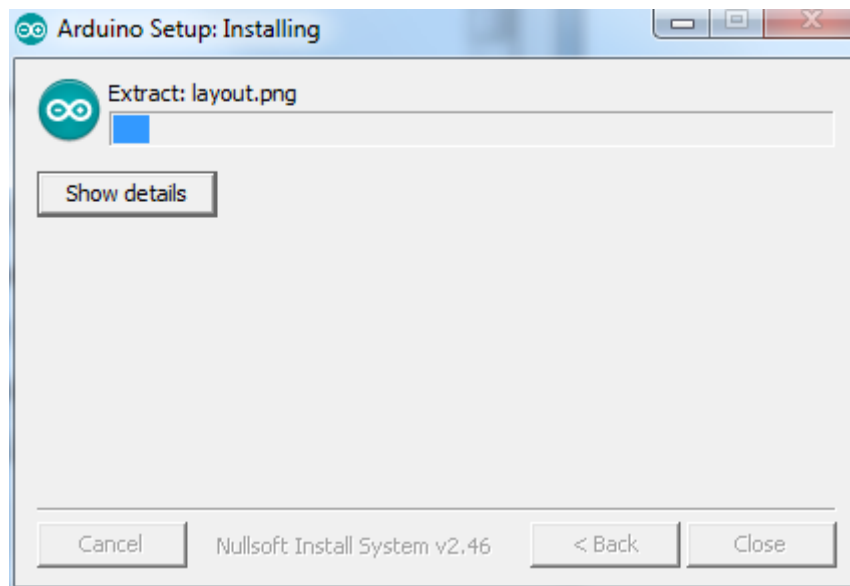
Choisir “I agree”



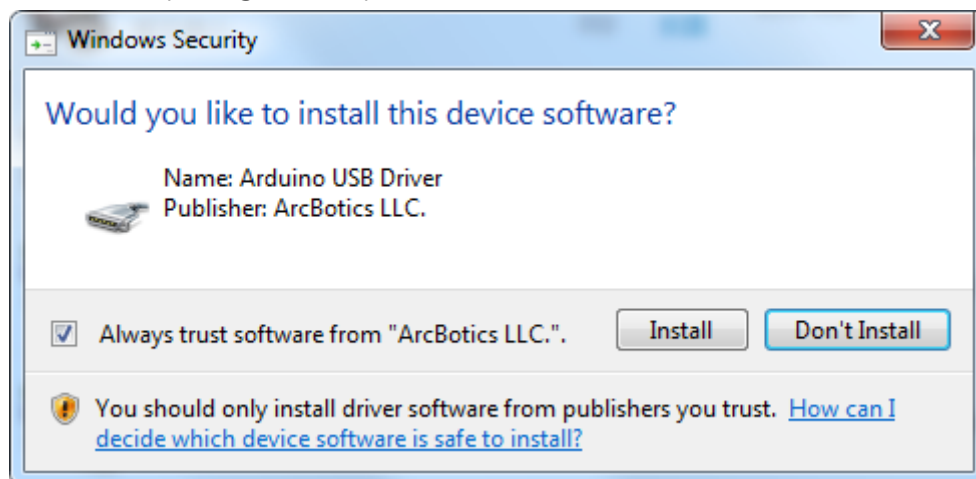
Choisir “Next”



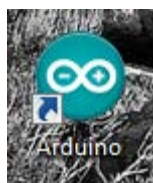
Cliquer sur Install pour démarrer l'installation



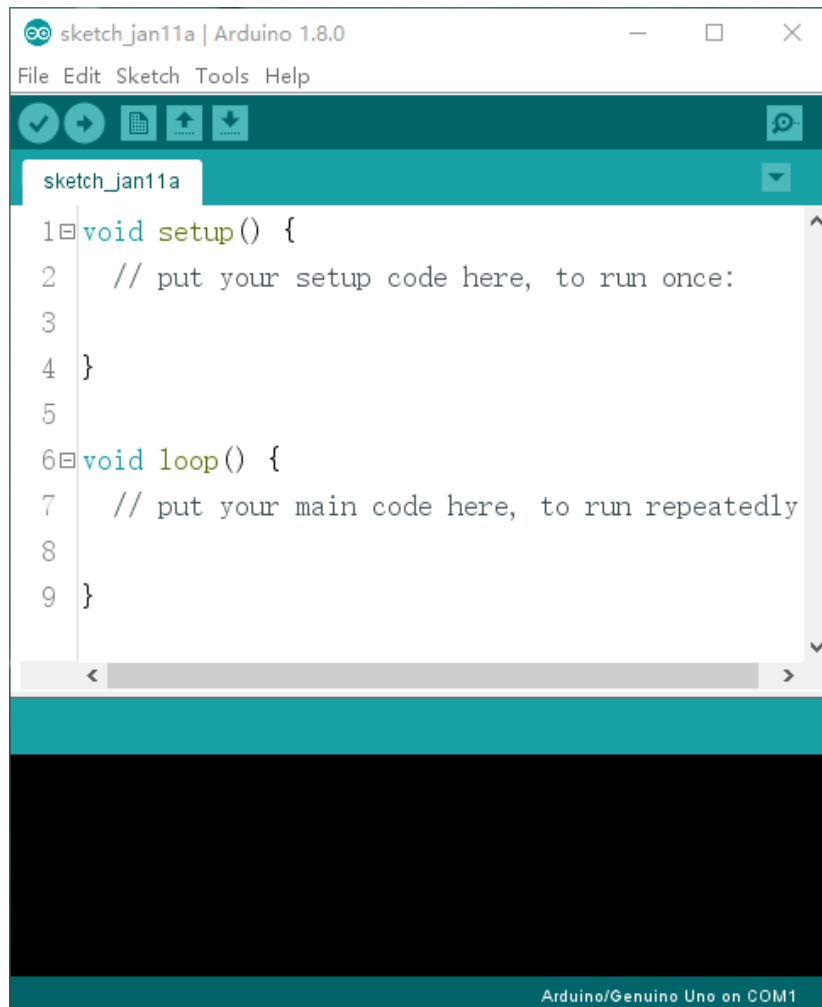
Installer le logiciel suivant pour garantir que l'ordinateur reconnaitra la carte



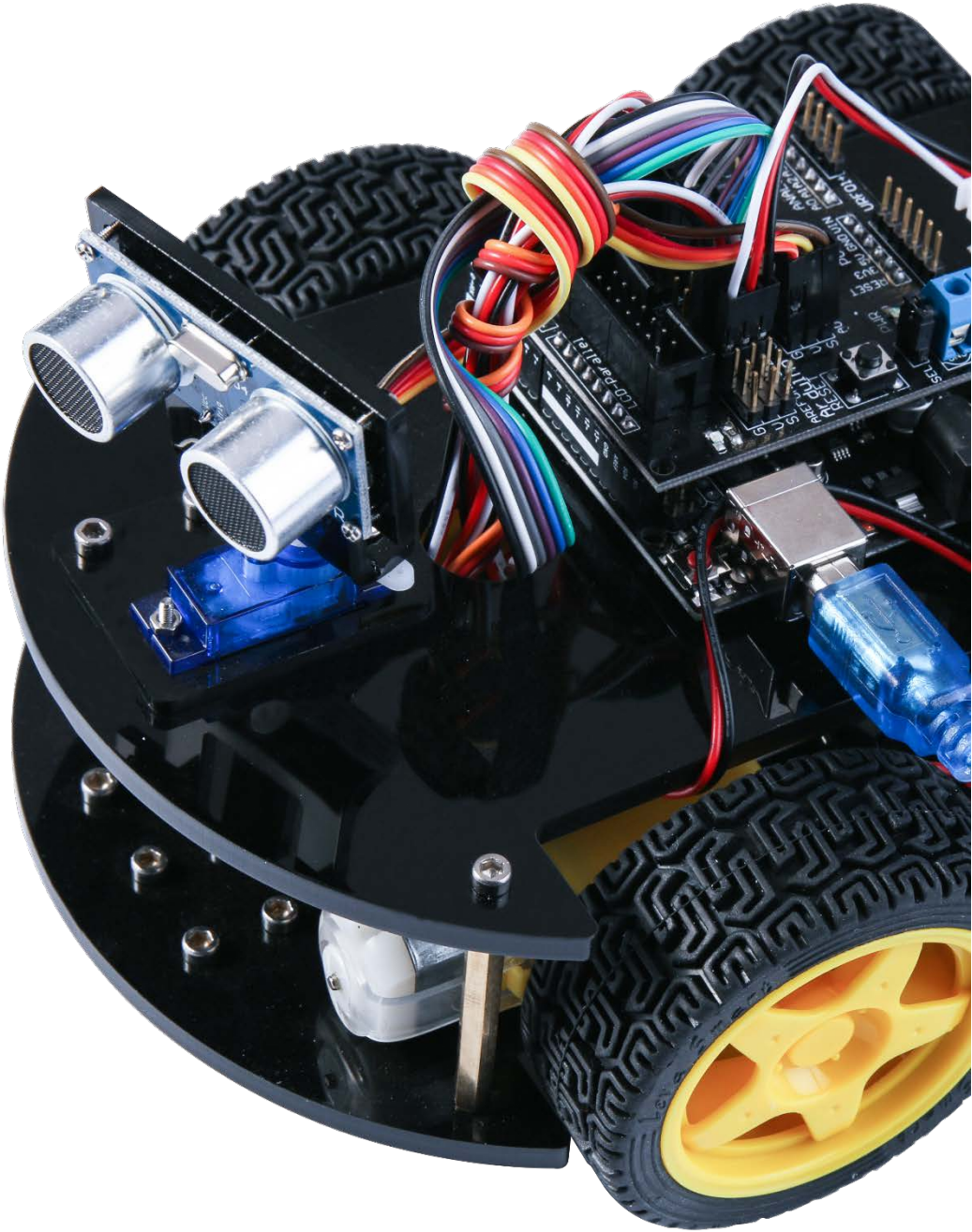
Vous avez maintenant cet icône sur votre bureau



Double cliquer sur l'icône pour lancer l'interface de programmation (IDE)

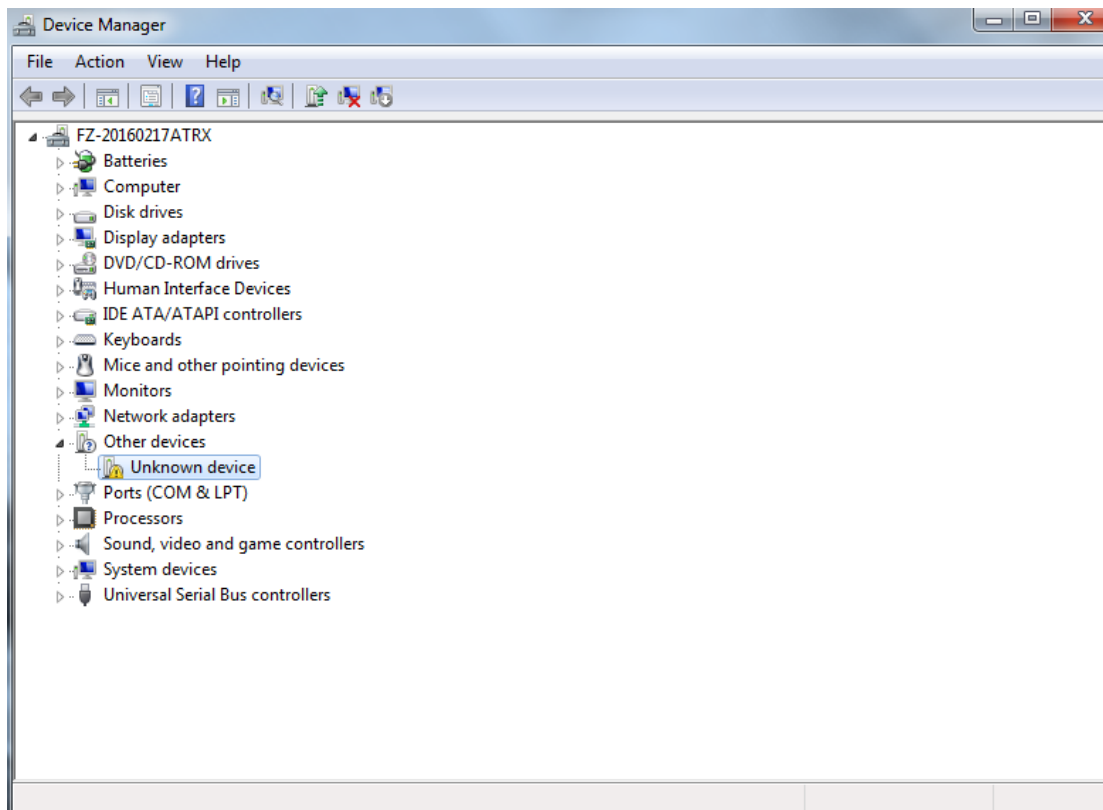


**Etape 3: Connecter le robot à l'ordinateur.**

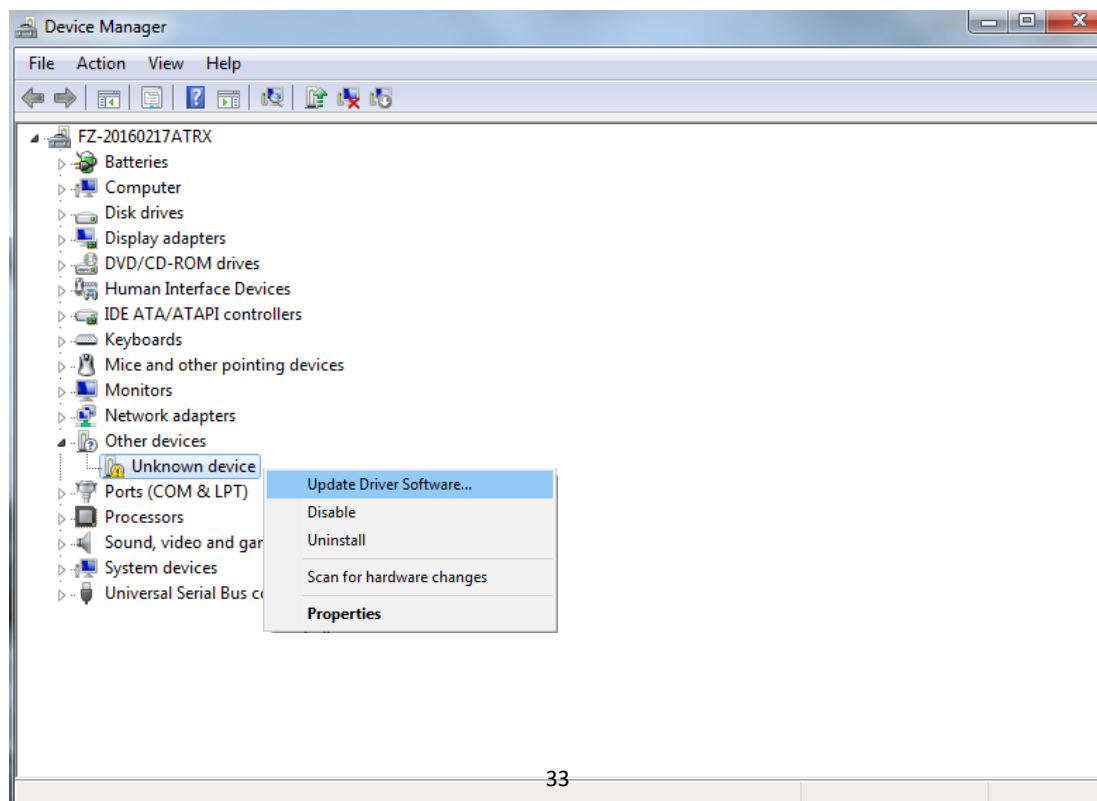


**Etape 4: Ouvrir IDE—Tool—Port. Si vous voyez le bon port, cela signifie que le véhicule a bien été reconnu par votre ordinateur. Dans ce cas, vous pouvez sauter directement à l'étape 5. Dans le cas contraire, vous devez installer le pilote en suivant la procédure suivante**

Ouvrir le gestionnaire de périphériques (mon ordinateur —système—gestionnaire de périphériques)

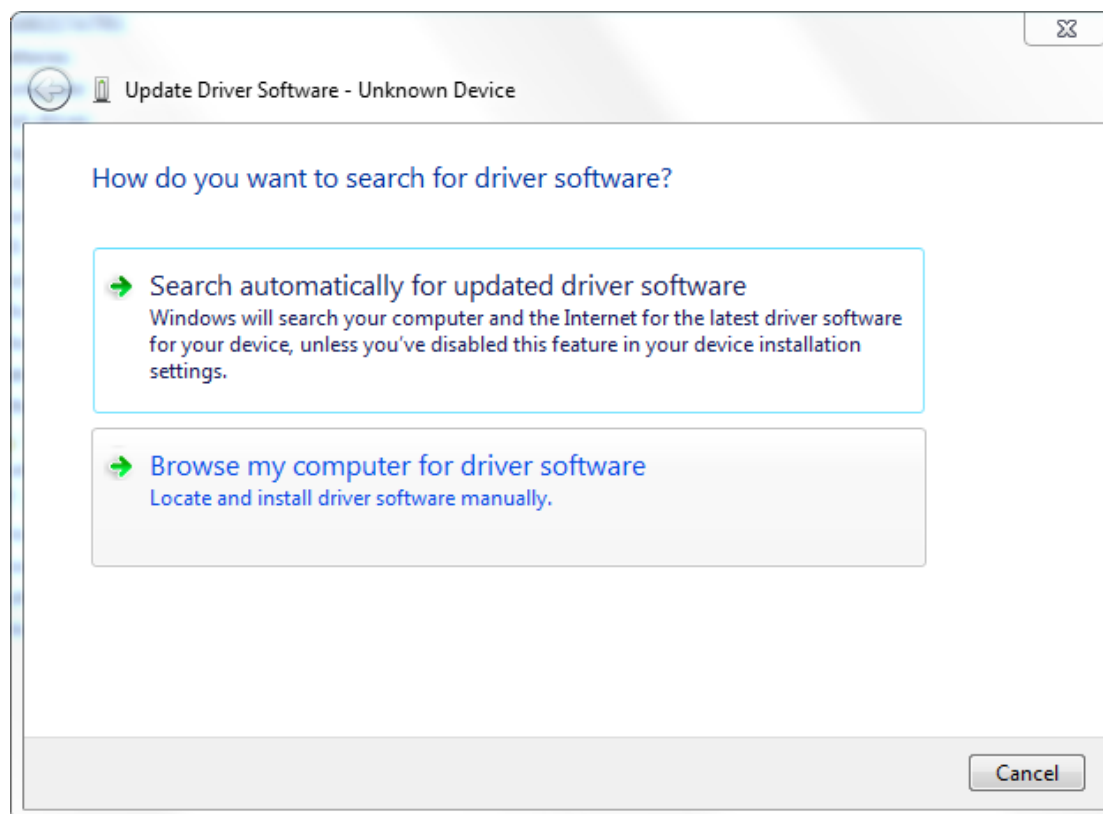


Clic droit sur le matériel inconnu





Cela montre que le pilote n'a pas été installé, vous devez faire un click sur "browse my computer" (parcourir mon ordinateur) pour trouver les pilotes. Les pilotes se trouvent dans le dossier Arduino que vous trouverez normalement dans C:\Program Files (x86)\Arduino.

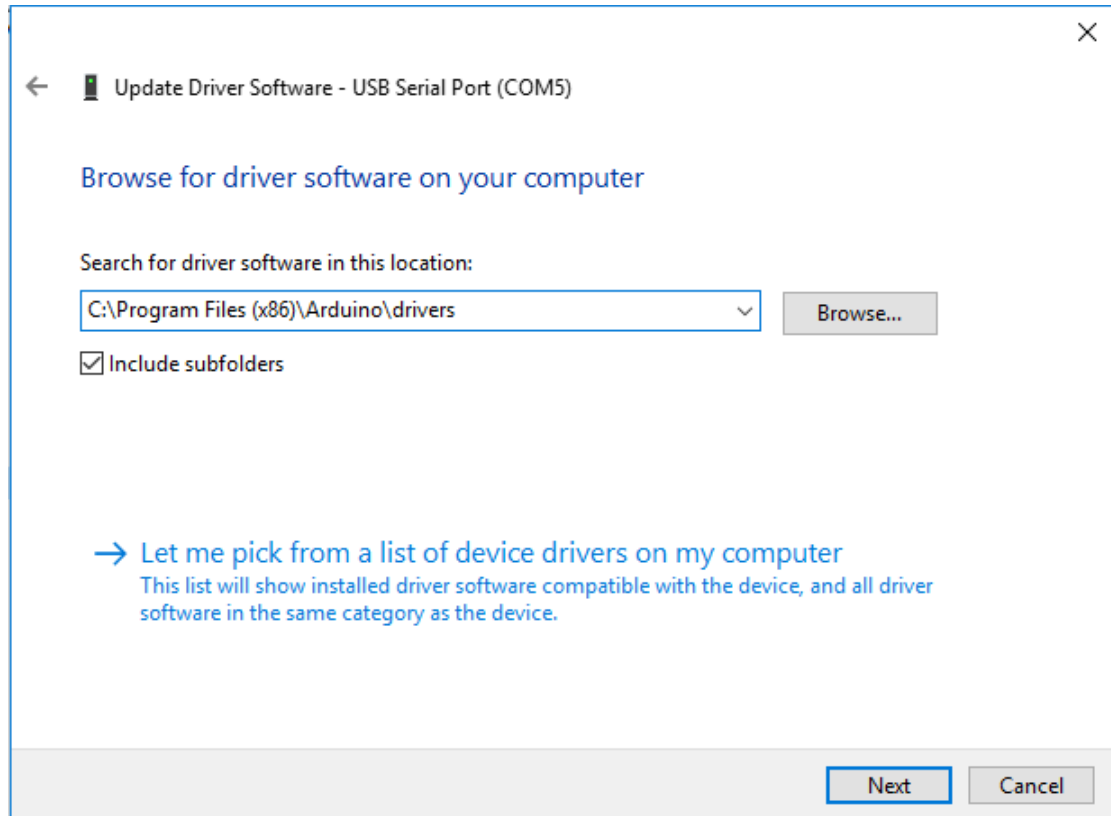


Dossier d'installation Arduino :

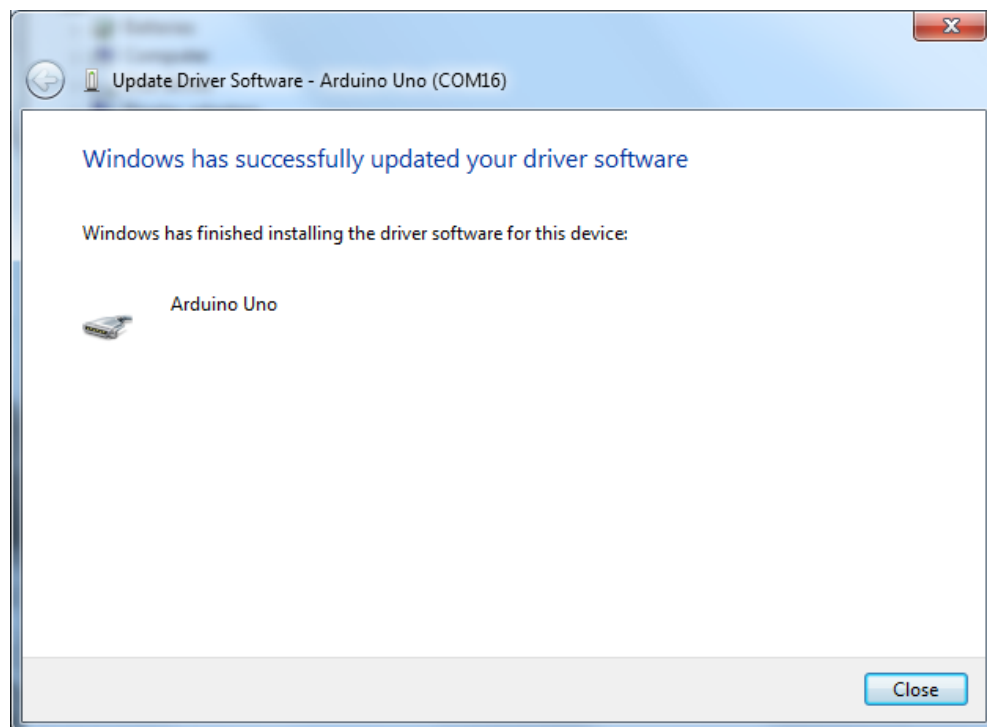
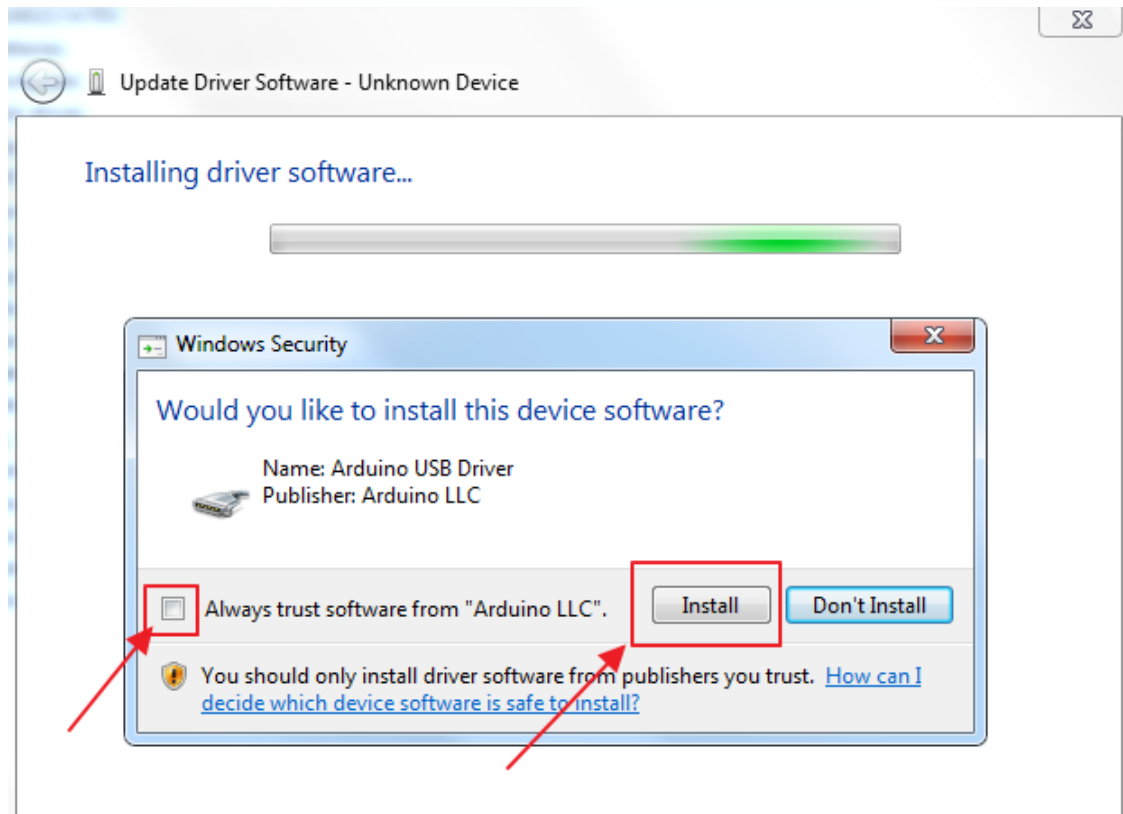
» This PC » Win10 (C:) » Program Files (x86) » Arduino »

Name	Date modified	Type	Size
drivers	1/16/2017 9:18 AM	File folder	
examples	1/16/2017 9:18 AM	File folder	
hardware	1/16/2017 9:18 AM	File folder	
java	1/16/2017 9:18 AM	File folder	
lib	1/16/2017 9:18 AM	File folder	
libraries	1/16/2017 9:18 AM	File folder	
reference	1/16/2017 9:18 AM	File folder	
tools	1/16/2017 9:18 AM	File folder	
tools-builder	1/16/2017 9:18 AM	File folder	
arduino.exe	1/9/2017 7:35 PM	Application	395 KB
arduino.l4j.ini	1/9/2017 7:35 PM	Configuration sett...	1 KB
arduino_debug.exe	1/9/2017 7:35 PM	Application	392 KB
arduino_debug.l4j.ini	1/9/2017 7:35 PM	Configuration sett...	1 KB
arduino-builder.exe	1/9/2017 7:32 PM	Application	3,192 KB
libusb0.dll	1/9/2017 7:32 PM	Application extens...	43 KB
msvcp100.dll	1/9/2017 7:32 PM	Application extens...	412 KB
msvcr100.dll	1/9/2017 7:32 PM	Application extens...	753 KB
revisions.txt	1/9/2017 7:32 PM	Text Document	81 KB
uninstall.exe	1/16/2017 9:18 AM	Application	404 KB
wrapper-manifest.xml	1/9/2017 7:35 PM	XML Document	1 KB

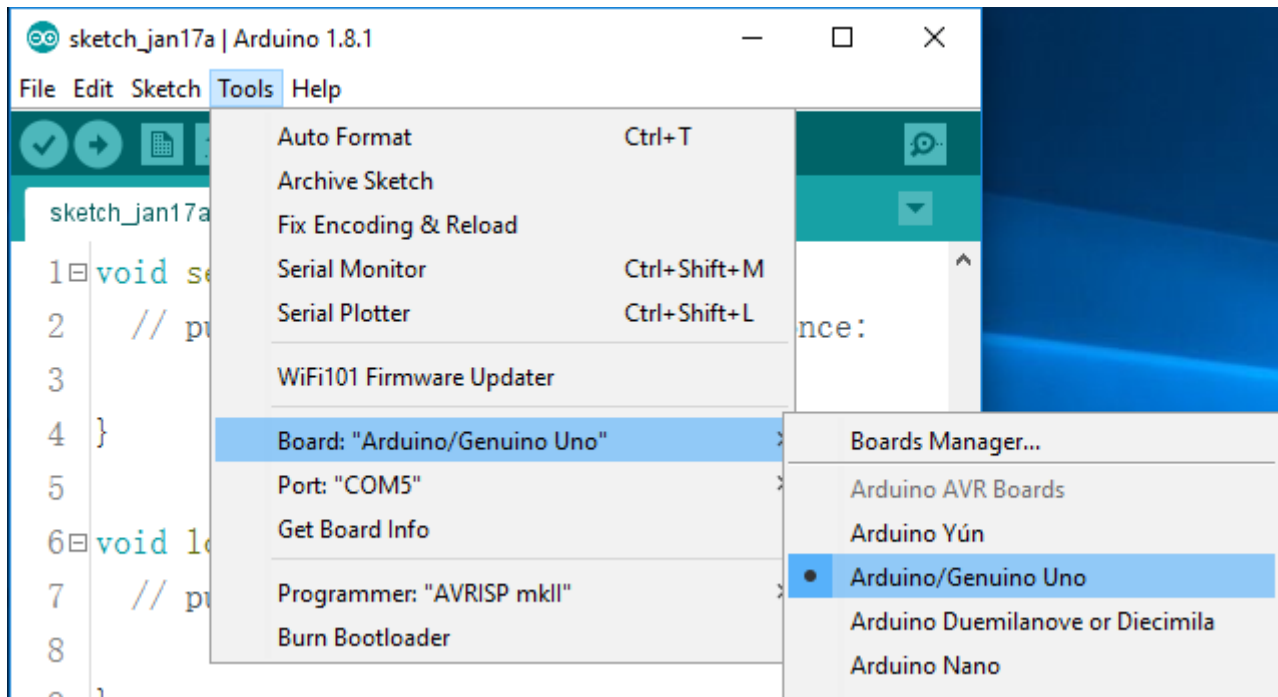
Selectionner le dossier Arduino



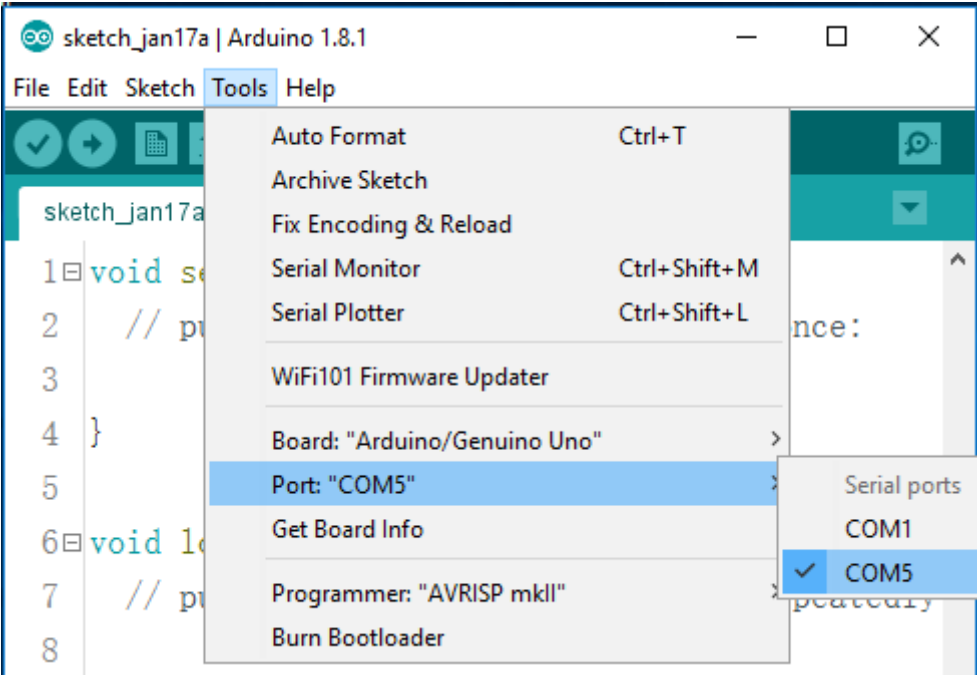
Installer le dispositif usb Arduino



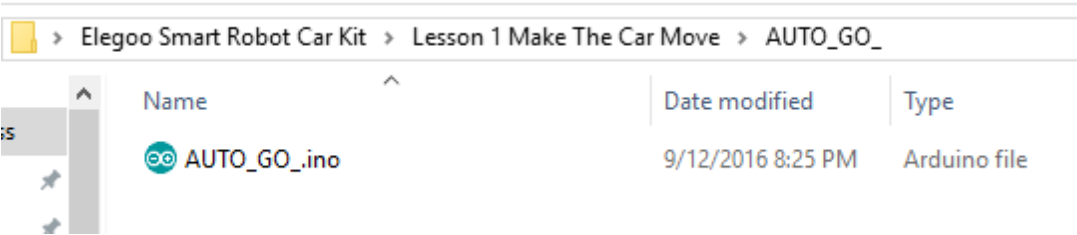
**Etape 5: Après avoir installé les pilotes, ouvrir l'IDE et cliquer sur Tools---Board---Arduino/GenuinoUno**



**Etape 6: Cliquer sur Tools---Port---COMx (Arduino/Genuino Uno)**



**Etape 7: Ouvrir le fichier AUTO\_GO\_\AUTO\_GO\_.ino et transférer dans la carte de contrôle UNO**





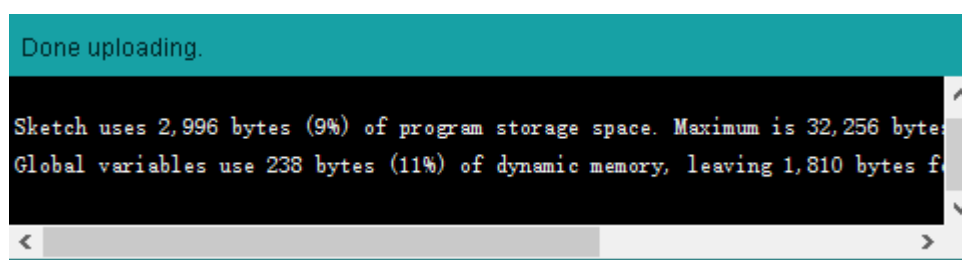


Cliquer sur le bouton avec la fleche pour debuter le transfert

**(ASTUCE: le module BLUETOOTH doit être retiré du robot avant le transfert du programme. Dans le cas contraire, une erreur surviendra durant le transfert.)**

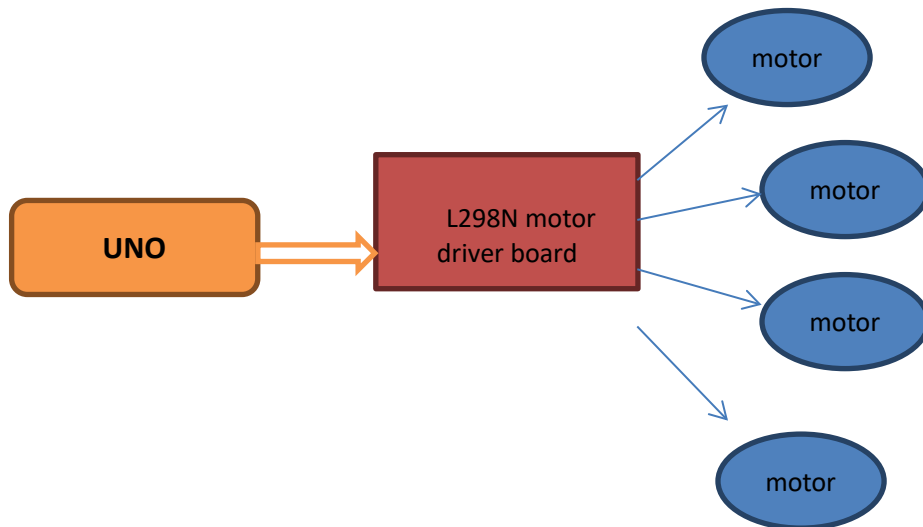


L'image ci dessous montre que le transfer test réussi.



**Etape 8:** Regardons ce qui se passe. Après avoir transféré le programme dans la carte contrôleur UNO, vous pouvez déconnecter la voiture de l'ordinateur. Vous pouvez actionner l'interrupteur d'alimentation et poser le robot sur le sol. Vous voyez à présent qu'il se déplace.

### III. Principes



#### Comment brancher la carte L298N

Les moteurs doivent être connectés à la carte L298N comme sur l'image ci-dessus, si vous constatez qu'un moteur tourne dans le sens opposé au sens attendu, échangez la position des fils noirs et rouges.

La carte L298N est branchée de la manière suivante

L298N GND est reliée à la masse de la batterie du robot;

L298N VCC est reliée au + de la batterie du robot;

La carte UNO est aussi connectée à la batterie.

L298N 5V ne peut pas être connecté au UNO 5V;

ENA et ENB contrôlent la vitesse des moteurs gauches et droites de manière séparée par une sortie PWM.

IN1, IN2, IN3, IN4 : IN1 et IN2 sont utilisés pour contrôler les moteurs de droites, IN3 et IN4 sont utilisés pour contrôler les moteurs de gauches. Voici ci-dessous un exemple de fonctionnement (moteurs de droites)

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

## IV. Mettre le robot en mouvement

### Première étape: piloter le moteur

Tout d'abord, nous allons mettre le moteur en mouvement sans nous soucier de la vitesse de rotation.

La carte L298N est reliée à la carte UNO comme suit:

L298N	V5 expansion board
ENA	10
IN1	9
IN2	8
IN3	7
IN4	6
ENB	5

Nous allons réaliser un programme simple, permettant de mettre les roues de droite en mouvement 0.5s dans un sens, puis 0.5s dans le sens inverse après une pause de 0.5s.

Connecter la carte contrôleur UNO à l'ordinateur, ouvrir le fichier right\_wheel\_rotation\right\_wheel\_rotation.ino



Après avoir ouvert le fichier .ino, le code suivant s'affiche dans l'IDE Arduino :

```
/*In1 connected to the 9 pin,  
In2 connected to the 8 pin, ENA pin 10,*/  
int ENA=10;  
int IN1=9;  
int IN2=8;  
void setup(){  
  pinMode(IN1,OUTPUT);  
  pinMode(IN2,OUTPUT);  
  pinMode(ENA,OUTPUT);  
  digitalWrite(ENA,HIGH);
```

```

}
void loop(){
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);      //Right wheel forward
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);      //Right wheel stop
  delay(500);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);      //Right wheel back
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);      //Right wheel stop
  delay(500);
}

```

Transférer le code sur la carte UNO, débranchez le câble USB, posez le robot au sol et mettez-le en marche. Vous pourrez constater que le robot se comporte comme exposé.

Si la voiture ne bouge pas, appuyer sur le bouton "RESET" de la carte UNO.

Si la direction est différente de celle attendue, inverser la position des bables noirs et rouges du moteur sur la carte L298N.

De la même manière, vous pouvez mettre en mouvement les roues de gauche :  
"Left\_wheel\_rotation\Left\_wheel\_rotation.ino"



**Code is as follow:**

```
/*In3 connected to the 7 pin,  
In4 connected to the 6 pin, ENB pin 5,*/  
int ENB=5;  
int IN3=7;  
int IN4=6;  
void setup(){  
  pinMode(IN3,OUTPUT);  
  pinMode(IN4,OUTPUT);  
  pinMode(ENB,OUTPUT);  
  digitalWrite(ENB,HIGH);  
}  
void loop(){  
  digitalWrite(IN3,LOW);  
  digitalWrite(IN4,HIGH);          //Left wheel forward  
  delay(500);  
  digitalWrite(IN3,LOW);  
  digitalWrite(IN4,LOW);          //Left wheel stop  
  delay(500);  
  digitalWrite(IN3,HIGH);  
  digitalWrite(IN4,LOW);          //Left wheel back  
  delay(500);  
  digitalWrite(IN3,LOW);  
  digitalWrite(IN4,LOW);          //Left wheel stop  
  delay(500);  
}
```

Transférer le code sur la carte UNO, débranchez le câble USB, posez le robot au sol et mettez-le en marche. Vous pourrez constater que le robot se comporte comme exposé.



## Deuxième étape : bouger en avant et en arrière

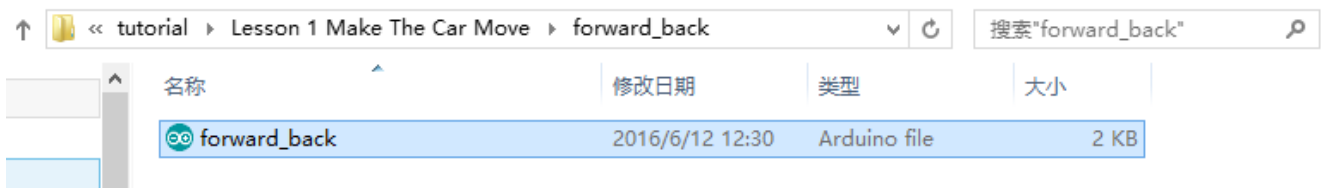
Vous pouvez maintenant écrire des programmes pour mettre le robot en mouvement

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

Nous allons écrire un programme simple permettant de mettre le robot en mouvement  
En avant 0.5s, Arrêt 0.5s, En arrière 0.5s, Arrêt 0.5s.

Connecter la carte UNO à l'ordinateur et ouvrir le fichier "forward\_back\forward\_back.ino"



**Code is as follow:**

```
int ENA=10;
int IN1=9;
int IN2=8;
int ENB=5;
int IN3=7;
int IN4=6;
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,LOW);
  digitalWrite(IN2,HIGH);          // left wheel goes forward
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);          // right wheel goes forward
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);          //left wheel holds still
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);          // right wheel holds still
  delay(500);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);          //left wheel is back up
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);          // right wheel is back up
  delay(500);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,LOW);          // left wheel holds still
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,LOW);          // right wheel holds still
  delay(500);
}

```

Transférer le programme sur la carte UNO du robot après avoir branché celui-ci à l'ordinateur via le câble USB.

Débranchez le câble, posez le robot au sol, mettez-le en marche et observez son comportement.

### Troisième étape : écrire un programme

Il peut sembler difficile de créer un code complet permettant de mettre le robot en mouvement.

Nous vous conseillons donc de bien séparer les mouvements dans différentes fonctions : Aller en avant, tourner à gauche .... Le programme final permettra d'exécuter ces différentes fonctions

Next, we begin to write programs for each movement:

*Le code est le suivant:*

```

/*****
Forward sub function
functions: Move forward
*****/
void forward( ){
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);          //Left wheel forward
    digitalWrite(IN3,LOW);
    digitalWrite(IN4,HIGH);        //Right wheel forward
}

/*****
Forward sub function
functions: Move backward
*****/
void back( ){
    digitalWrite(IN1,LOW);
    digitalWrite(IN2,HIGH);        //Left wheel back
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);        //Right wheel back
}

/*****
turnLeftsub function
functions: Turn left
*****/
void turnLeft( ){
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);        //Left wheel back

```

```

digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);          //Right wheel forward
|
|
|
/*****
turn Right sub function
functions: Turn right
*****/
void turnRight( ){
|
|
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);        //Left wheel forward
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);        //Right wheel back
|
|
|
/*****
stop sub function
functions: Stop
*****/
void _stop(){
|
|
digitalWrite(IN1,LOW);
digitalWrite(IN2,LOW);        //Left wheel stop
digitalWrite(IN3,LOW);
digitalWrite(IN4,LOW);        //Right wheel stop
|
|

```

## Quatrième étape : mouvements automatiques

Pour finir, nous commençons à mettre le robot en mouvement automatiquement : en avant 0.4s, en arrière 0.4s, à gauche 0.4s, à droite 0.4s

Vous pouvez accéder à ce code via le fichier "AUTO\_GO\_ \ AUTO\_GO\_.ino"



The code is as below:

```

/*define logic control output pin*/
int in1=9;
int in2=8;
int in3=7;
int in4=6;
/*define channel enable output pins*/
int ENA=5;
int ENB=10;
/*define forward function*/
void _mForward()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,HIGH);//digital output
    digitalWrite(in2,LOW);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    Serial.println("Forward");
}
/*define back function*/

```



```
void _mBack()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("Back");
}

/*define left function*/
void _mleft()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("Left");
}

/*define right function*/
void _mright()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    Serial.println("Right");
}

/*put your setup code here, to run once*/
```

```

void setup() {
  Serial.begin(9600); //Open the serial port and set the baud rate to 9600
  /*Set the defined pins to the output*/
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
}
/*put your main code here, to run repeatedly*/
void loop() {
  _mForward();
  delay(1000);
  _mBack();
  delay(1000);
  _mleft();
  delay(1000);
  _mright();
  delay(1000);
}

```

Transférer le code sur la carte UNO, débranchez le câble USB, posez le robot au sol et mettez-le en marche. Vous pourrez constater que le robot se comporte comme exposé.