

車を動かす

Points of this section

学習パーツ :



Arduino IDE の使い方を学ぶ



プログラムをアップロードして車を動かす

準備 :



車（バッテリー付き）

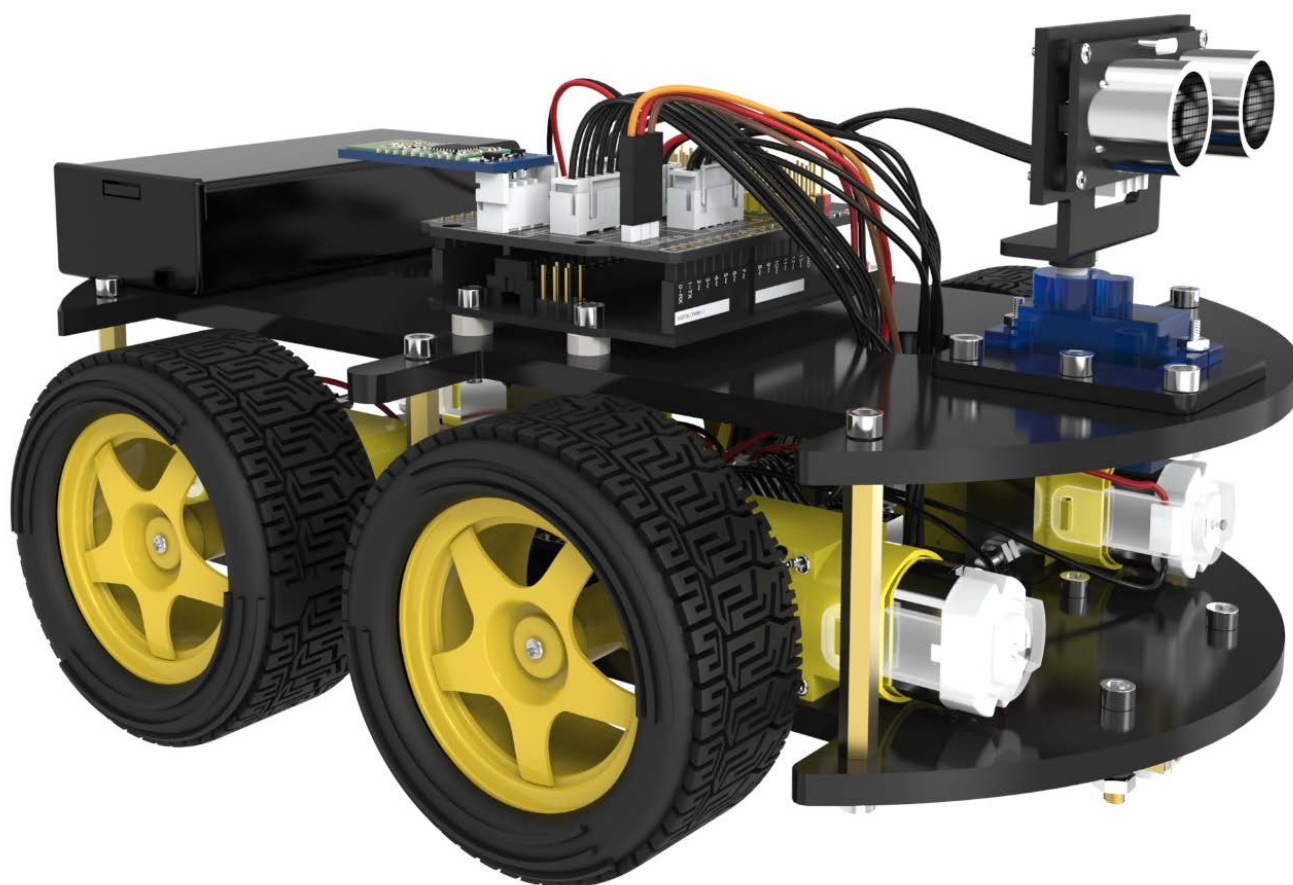


USB ケーブル

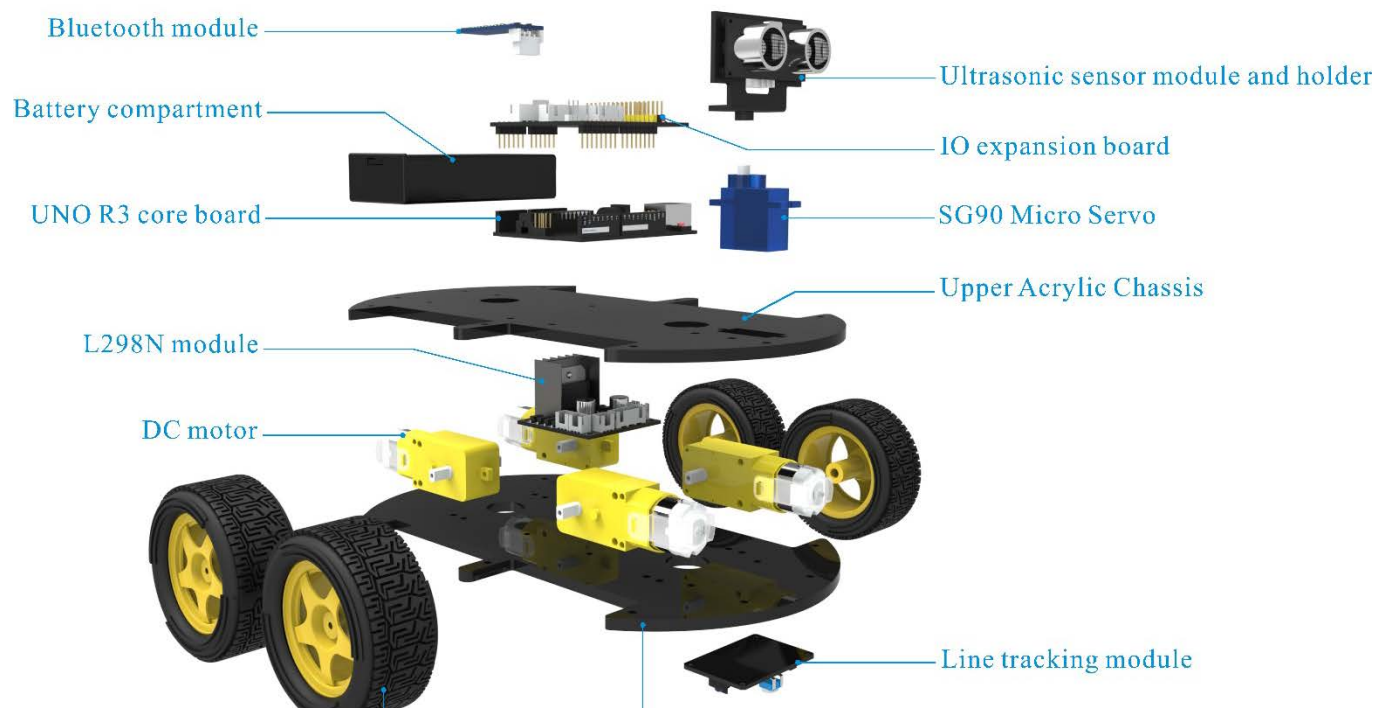
I. 車の紹介

このキットは、特に教育、競技、娯楽の目的で設計された非常に柔軟な車両用キットです。

キットの上部パネルは、9 グラムステアリングエンジンと直接互換性があります。また、超音速センサー、バッテリー、その他の固定穴を持ち、さまざまなセンサーの取り付けを容易にします。これは、学習や制作の目的に合う非常に面白く多用途のロボットです。これにより、Bluetooth や赤外線リモコン、障害の自動回避、回線検査など、さまざまな興味深いアイデアを実装できます。将来的に私たちと共にある小さな車両について説明しましょう。



Each parts of the car is as below:



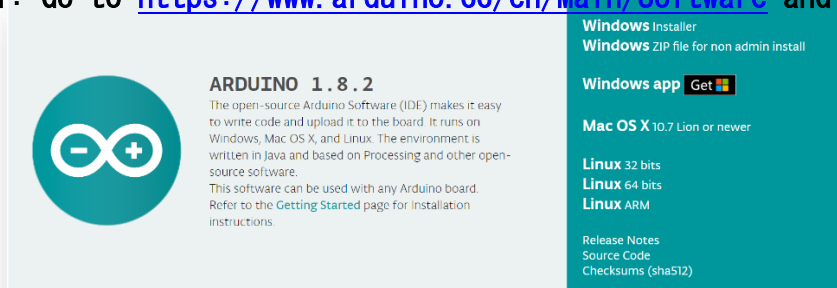
Function of each part :

1. スイッチ付きバッテリーホルダー：車両に電源を供給する
2. 電動モーター+ホイール：車両を動かす
3. アクリル板：車のフレーム
4. L298N モータ駆動基板：モータを回転させて動かす
5. UNO コントローラボード：車の脳、すべての部品を制御
6. V5 センサー拡張ボード：UNO と組み合わせると、接続がより簡単になります
7. サーボとクラウドプラットフォーム：GP2Y0A21 距離センサーを 180 度回転させる
8. 超音波センサモジュール：距離測定と障害物回避
9. ライン追跡モジュール：白と黒のレーンを認識するための白黒センサー
10. 赤外線受信機とリモコン：赤外線リモコン機能を提供する
11. Bluetooth モジュール：Bluetooth 制御機能を提供する

II. Upload program (プログラムをインストールする)

車両の各動きはプログラムによって制御されるので、プログラムをインストールして正しく設定する必要があります。Arduino Software IDE (統合開発環境) をプログラミングツールとして使用します。

STEP 1: Go to <https://www.arduino.cc/en/Main/Software> and find below page.



このウェブサイトで入手可能なバージョンは通常最新バージョンであり、実際のバージョンは画像のバージョンより新しい場合があります。

STEP2: お使いのコンピュータのオペレーティングシステムに適した開発ソフトウェアをダウンロードしてください。

ここで Windows を例に取る。

EXE インストールパッケージまたはグリーンパッケージを使用してインストールできます。

Windows Installer
Windows ZIP file for non admin install

以下は、インストール手順の exe 実装です。

“Windows Installer” を押す

Windows Installer

Press the button “JUST DOWNLOAD” to download the software.

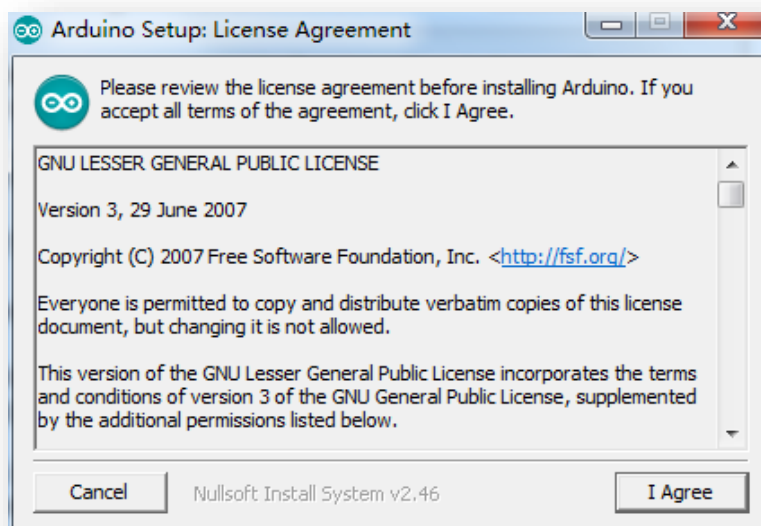


JUST DOWNLOAD

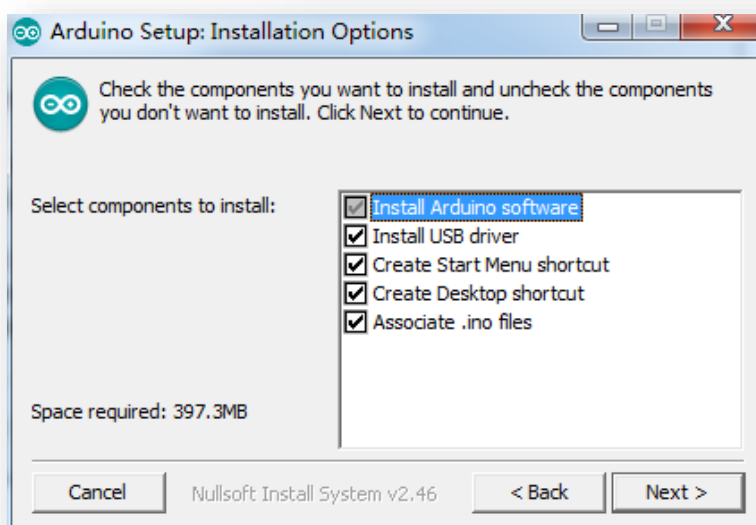
The download file

arduino-1.8.2-windows.exe

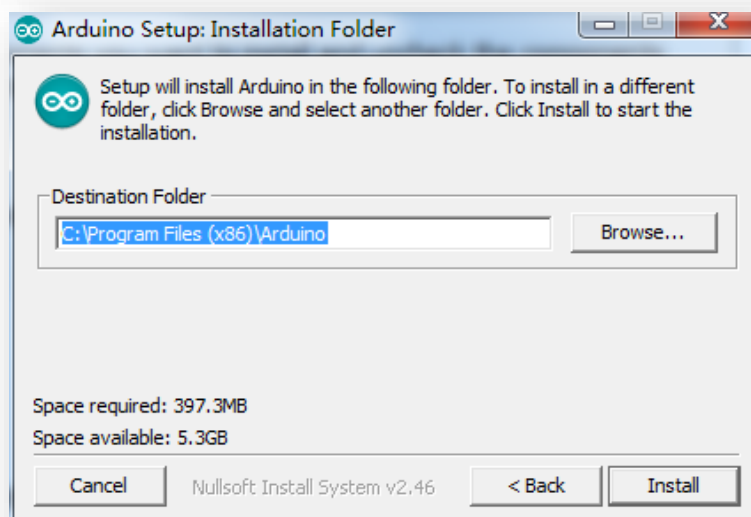
これらは私たちが提供している資料で入手できます。また、私たちの教材のバージョンは、このコースが作成されたときの最新バージョンです。



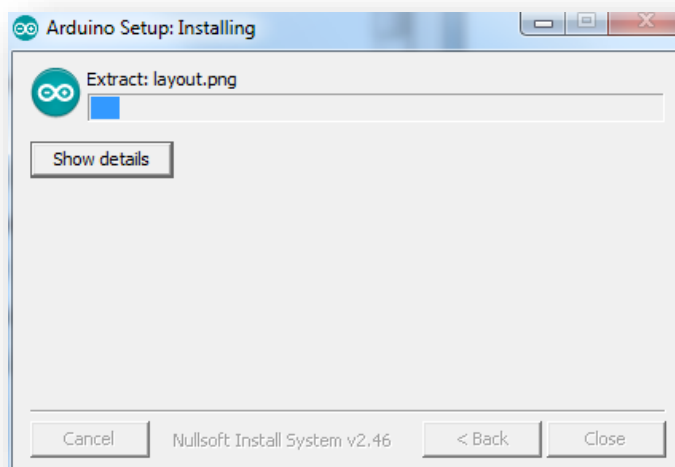
「同意する」を選択すると、次のインターフェースが表示されます。



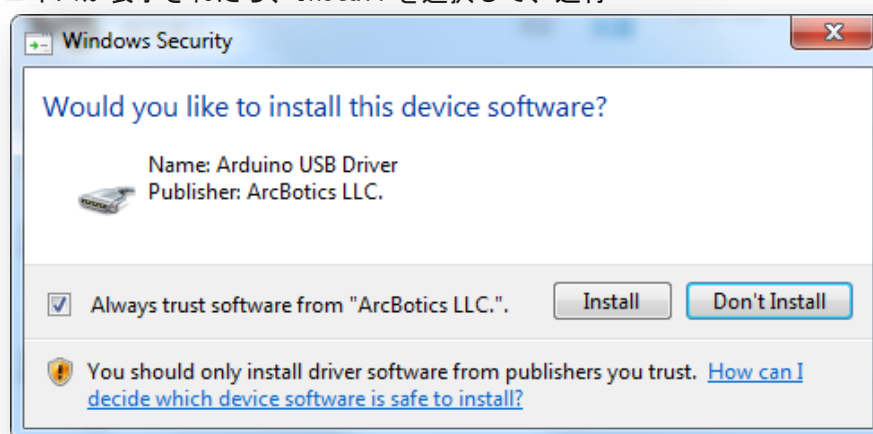
Next を選択



インストールを開始するには、[Install]を押します。



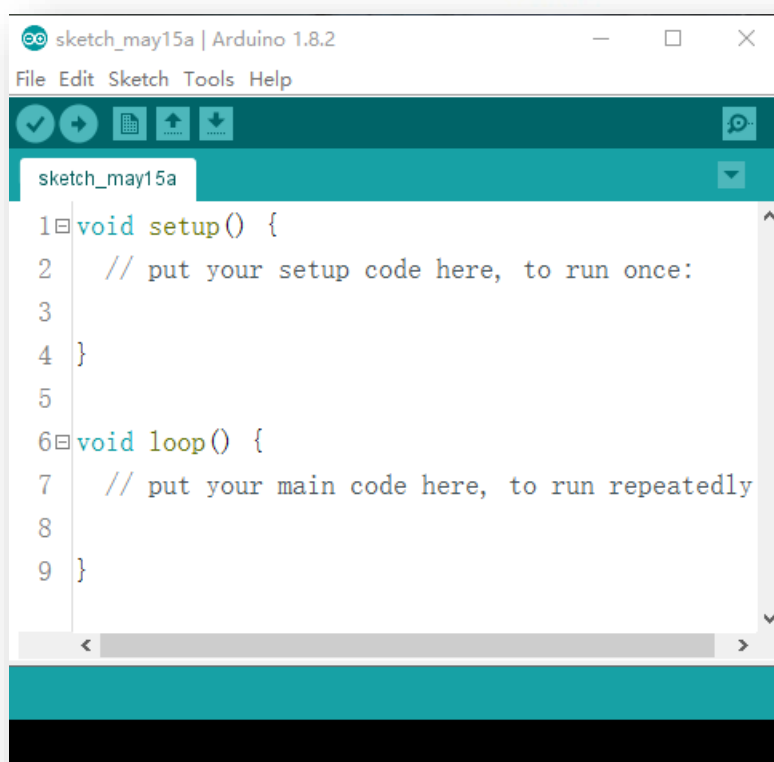
最後に、次のインターフェイスが表示されたら、Installを選択して、進行



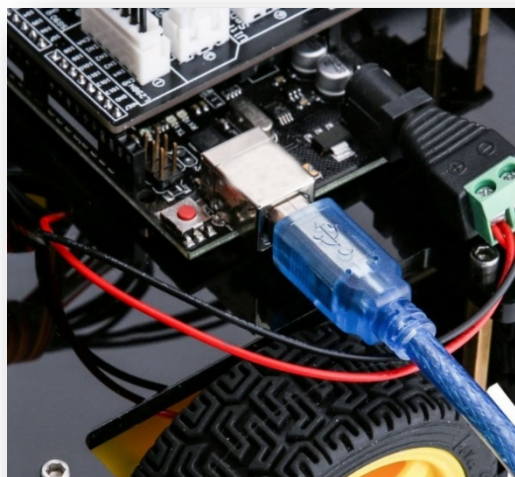
次に、デスクトップに次のアイコンが表示されます



ダブルクリックして希望の開発環境に入ります

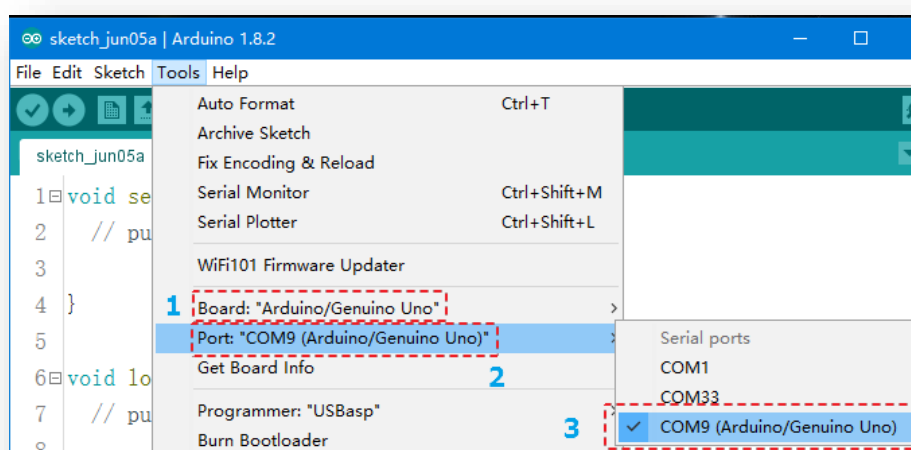


STEP3 : 車をコンピュータに接続します。

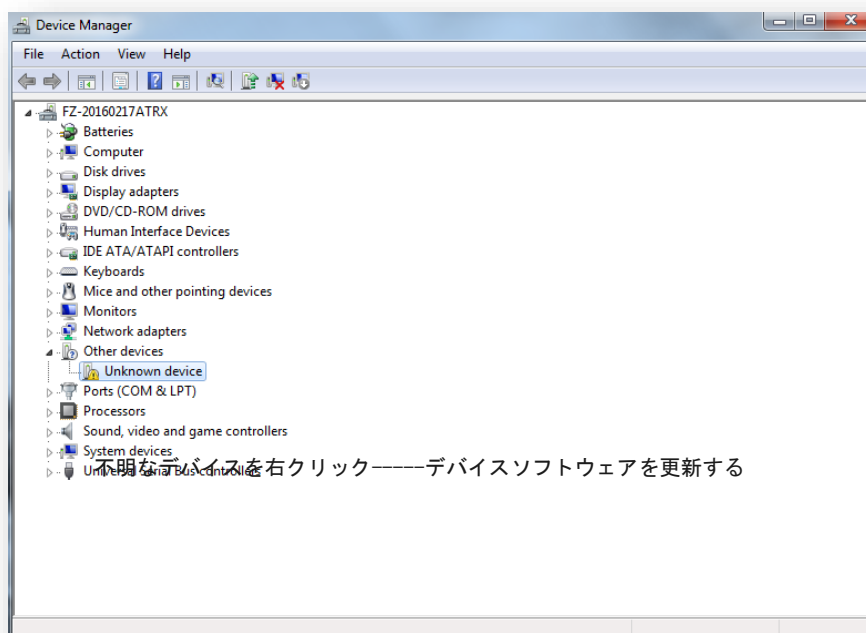


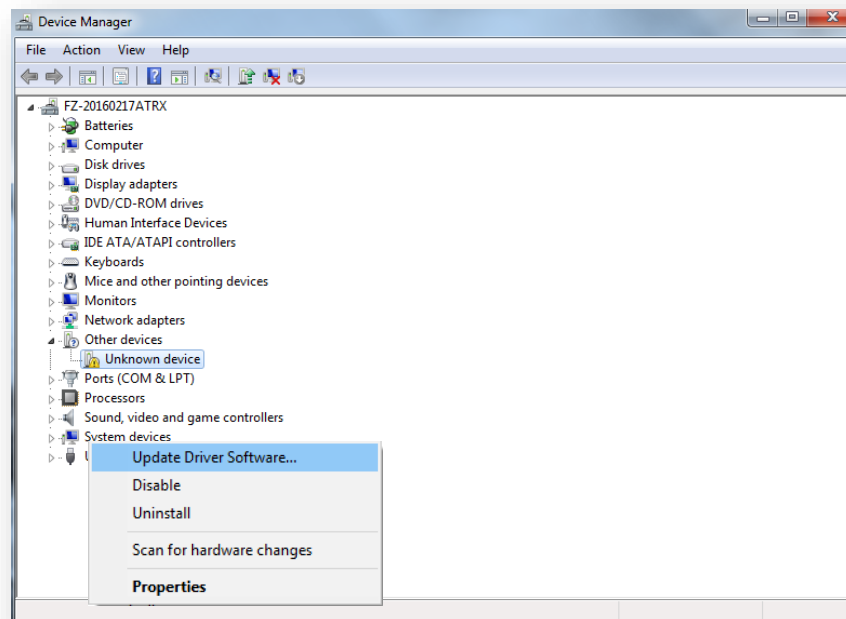
STEP 4: Arduino IDE を開きます。“Tool”→“Board :”→“Arduino / Genuino Uno”を選択します。“Tool”→ “Port :”→ “COM (Arduino / Genuino Uno) ”を選択します

各 Arduino Uno ボードは同じコンピュータ上に異なる COM 番号を持ち、通常は Arduino 1.8.2 の接尾辞名「(Arduino / Genuino Uno) 」を持つ COM 番号を持ちます。 実際のディスプレイの COM 番号を選択する必要があります。



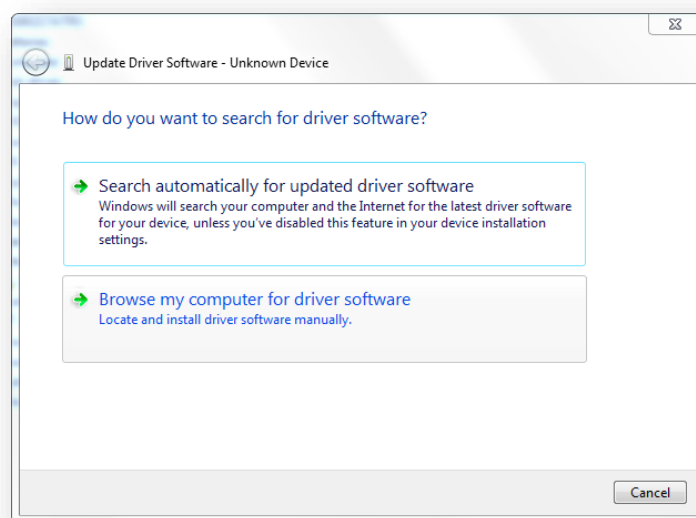
ポート「COM (Arduino / Genuino Uno) 」が表示されている場合は、車両がコンピュータに正しく接続されていることを意味します。 この場合、STEP 5 に直接ジャンプすることができます。 それ以外の場合は、次の方法でドライバをインストールする必要があります。Open マイコンピュータ - 管理 - デバイスマネージャを右クリックしてデバイスマネージャを開きます。



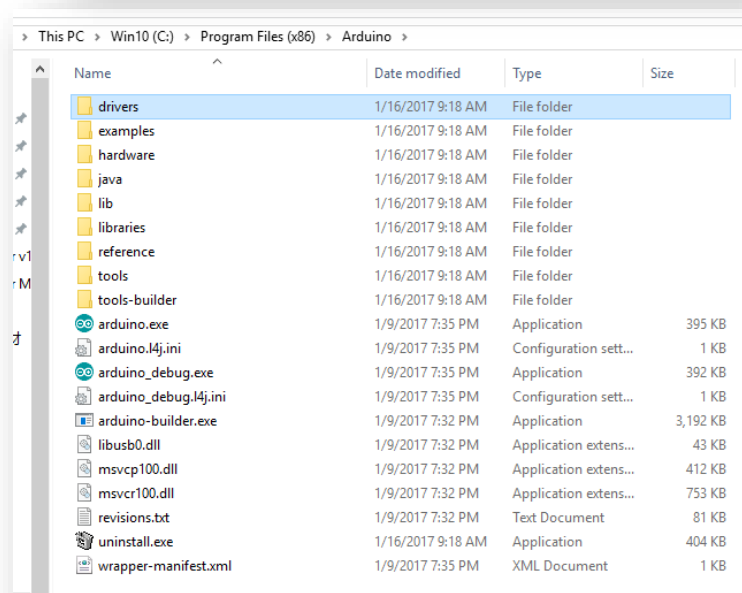


ドライバがインストールされていないことが表示され、[コンピュータを参照してドライバソフトウェアを検索する]をクリックしてドライバを検索する必要があります。ドライバはArduino フォルダにあります。通常、このフォルダは次の場所にインストールされます。

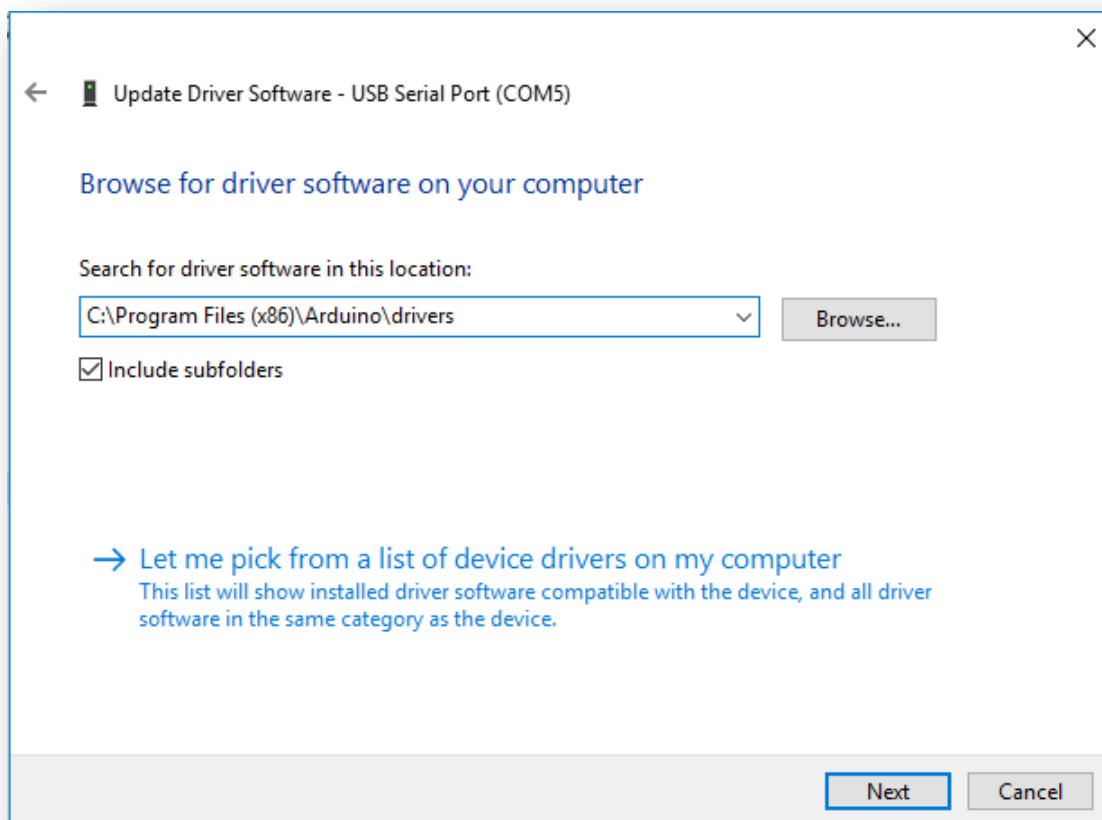
C:\Program Files (x86)\Arduino.



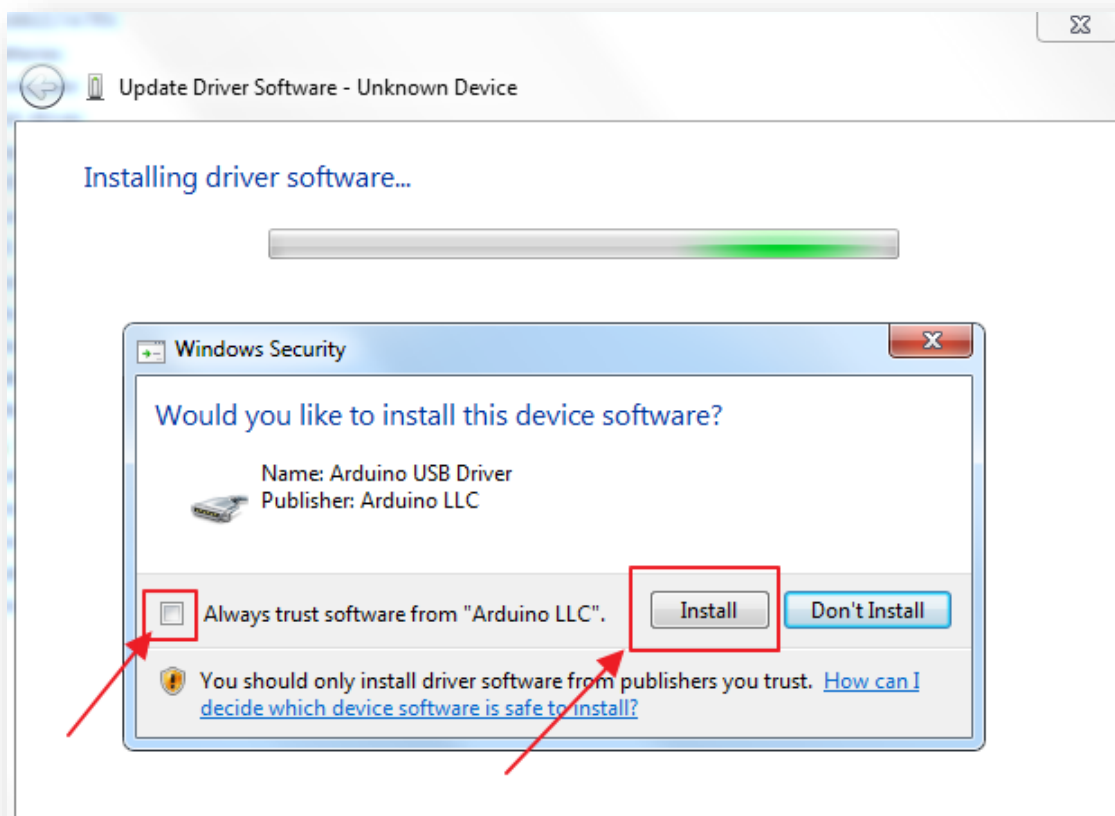
Arduino install folder

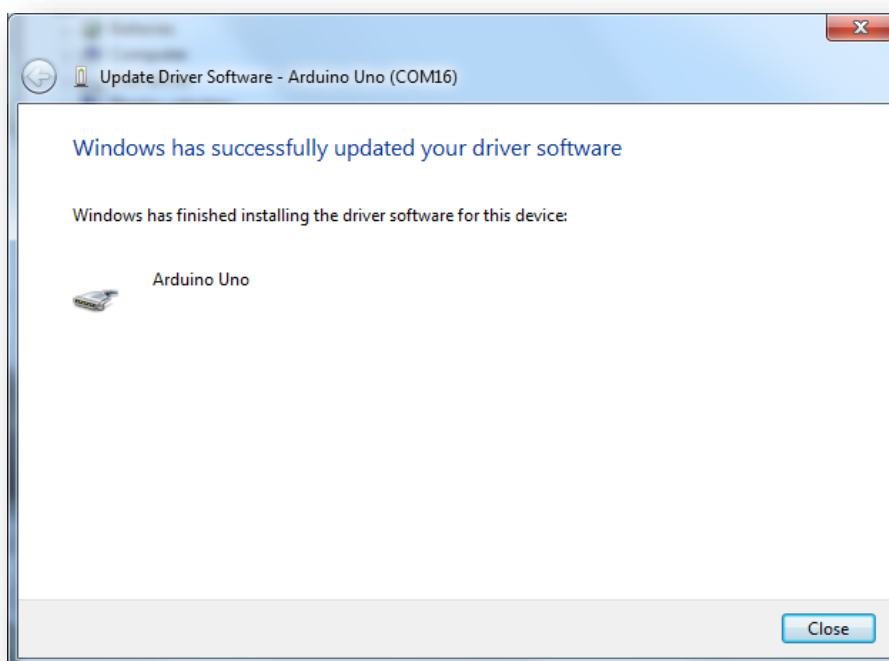


Select the Arduino driver folder

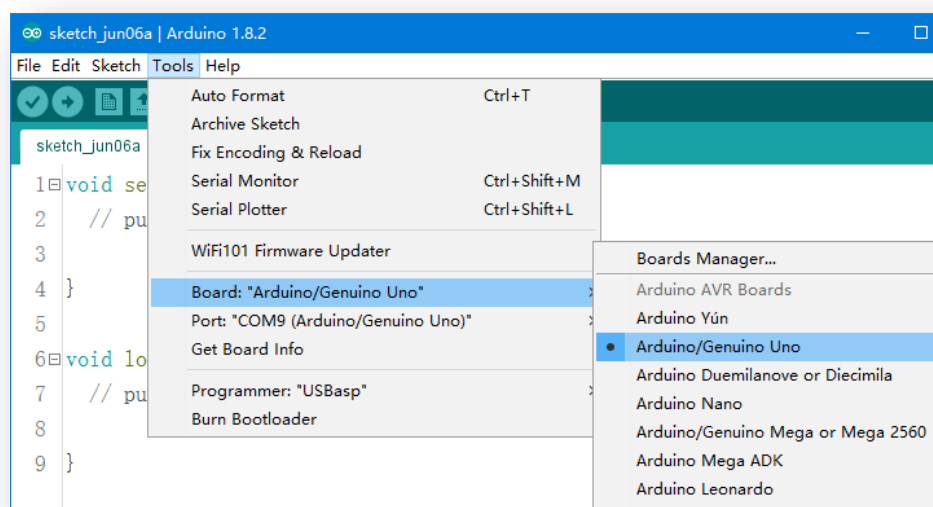


Install Arduino USB device

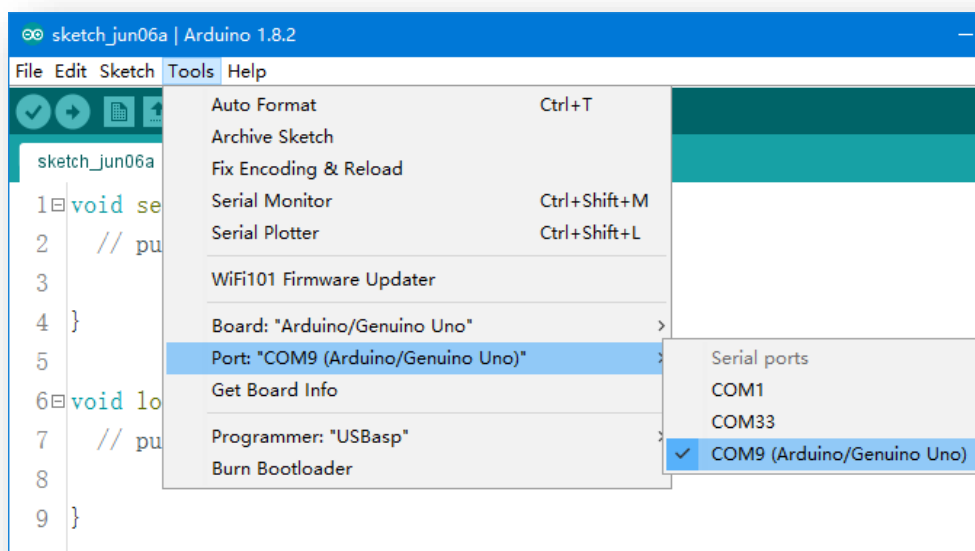




STEP5: ドライバをインストールしたら、IDE を開き、クリック “Tools” →” Board” → “Arduino/Genuino Uno” .

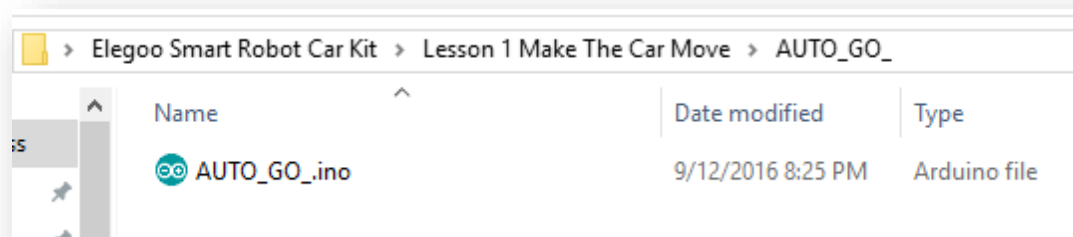


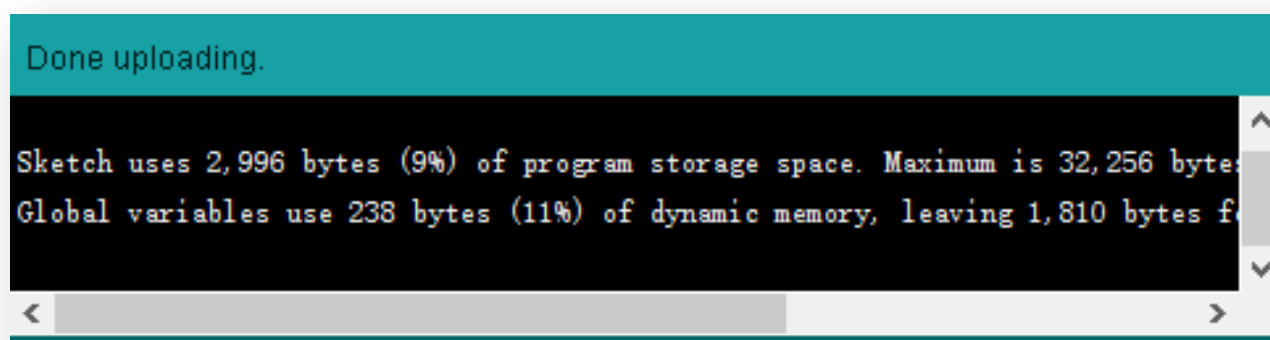
STEP6: Click “Tools” →” Port” →COM.



STEP7: ディレクトリ内のコードファイルを開きます。 “¥Lesson 1 Make The Car Move ¥AUTO_GO_¥AUTO_GO_.ino”

UNO コントローラボードへのアップロード (TIPS: 毎回プログラムをアップロードするときに Bluetooth モジュールを抜かなければなりません。そうしないと、プログラムのアップロードに失敗します。)

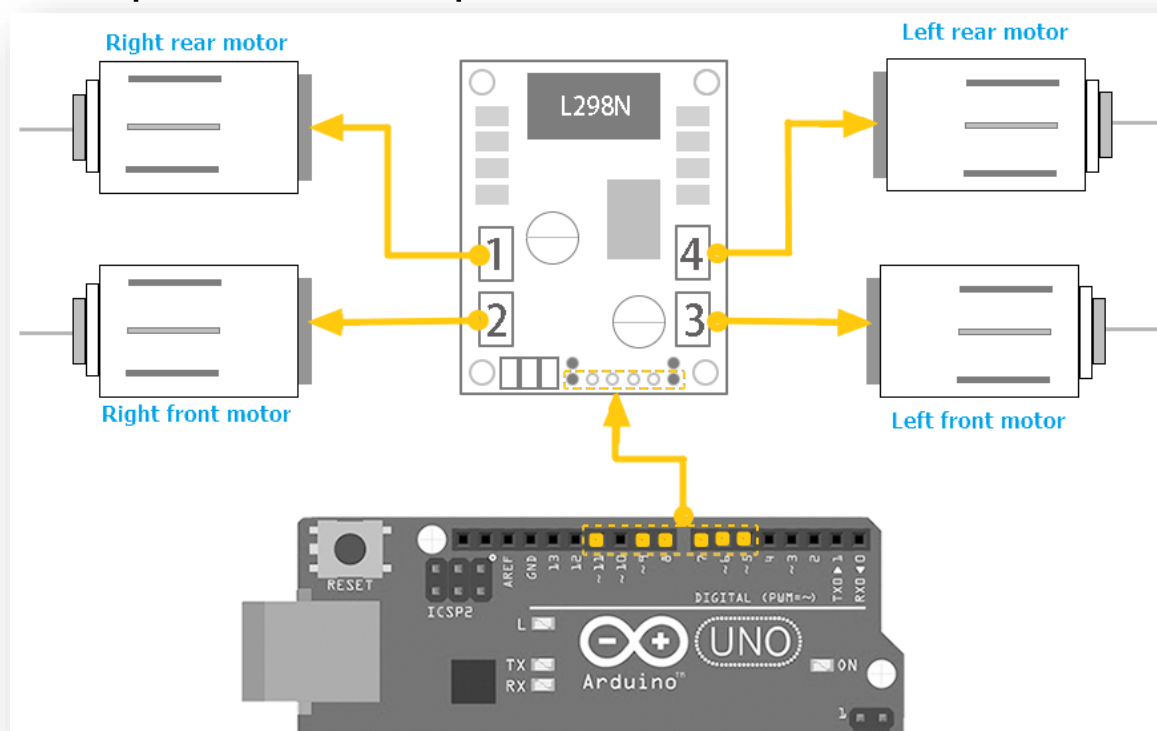




上の写真はアップロードが成功したことを示しています。

STEP8 : 結果を見てみましょう。 プログラムを UNO コントローラボードにアップロードします。 車をコンピュータに接続した後、電源スイッチをオンにして車を地上に置くことができます。 それからあなたは車が動いているのを見るでしょう。

Ⅲ. Description of Principles (原則の説明)



L298N motor driver board の使用方法

L298N ボード上の接続ポートの定義は上にマークされています。上記の図のようにモータを L298N 基板に接続してください。一方のモータの回転方向が反対の場合は、黒と赤のワイヤの接続位置を変更してください。

L298N GND は battery box GND に接続;

L298N VCC is は battery box VCC に接続;

UNO board は battery box に接続

ここで L298N 5V は UNO 5V に接続できません。

ENA と ENB は、PWM によって左のモータの速度と右のモータの速度を別々に制御します。

IN1、IN2、IN3、IN4 : IN1 と IN2 は左のモータを制御するために使用され、IN3 と IN4 は右のモータを制御するために使用されます。 原理については、以下のシートをご覧ください：（例えば、左のモーターを取る）

| 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. Make the Car Move(車を動かす)

The first step: モーターを駆動する

私たちはスピードをコントロールすることなくモーターを動かそうとします。速度制御なしでプログラムを書くのは簡単だからです。

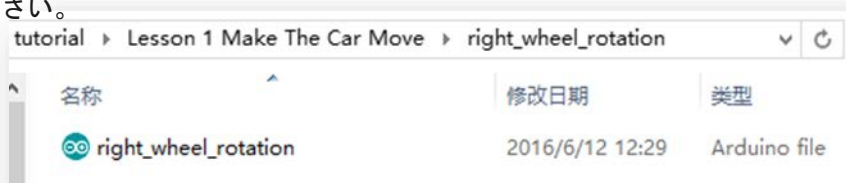
まず、L298N ボードのモーター接続を見てみましょう。私たちは Arduino 5, 6, 7, 8, 9, 11 ピンを使って車を制御します。9 と 11 ピンが右のホイールを制御します。7, 8 ピンは左のホイールを制御します。5 ピンと 6 ピンは ENA と ENB を制御します。

接続は以下のようになります：

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

上のシートに基づいて、最初に、右の車輪を正の方向に 0.5s、0.5s を停止、負の方向に 0.5s、0.5s を停止する単純なプログラムを設計します。そしてホイールは反応を繰り返すでしょう。

UNO コントローラボードをコンピュータに接続し、パス “¥レッスン 1 車の動きを確認する ¥ right_wheel_rotation ¥ right_wheel_rotation.ino” でコードファイルを開きます。UNO ボードにプログラムをアップロードしてください。



Code preview:

```
//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
//  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 stopped
  delay(500);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW); //Right wheel turning backwards
  delay(500);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH); //Right wheel stopped
  delay(500);
}

```

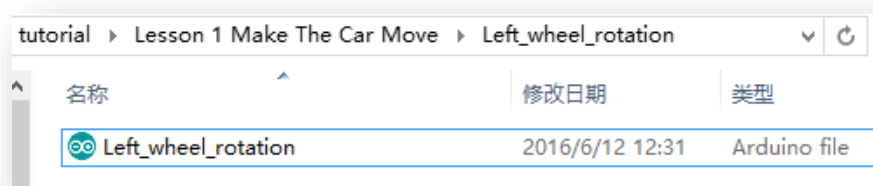
それをコンピュータから取り外し、車の電源を入れます。あなたは、あなたが期待どおりに右の車輪が動くのを見るでしょう。

車が動いていない場合は、UNO ボードのリセットボタンを押してください。

モーターの移動方向が設定した方向と異なる場合は、モータから L298N ボードへの黒線と赤線の接続を変更することができます。

次に、同じ方法で左のホイールを回転させます。

UNO コントローラボードをコンピュータに接続し、コードファイルを “Lesson 1 Make The Car Move ¥ Left_wheel_rotation ¥ Left_wheel_rotation.ino” パスで開きます。プログラムを UNO ボードにアップロードします。



Code preview:

```
//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);
}
```

それをコンピュータから取り外し、車の電源を入れます。 あなたは、あなたが期待どおりに右の車輪が動くのを見るでしょう。

he second step: 前後に移動

車のデバッグを終了したら、車を動かすプログラムを書くことができます。

以下は車の動き方です：

| 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 |

次に、車を 0.5 秒進ませてから 0.5 秒を止める、そして車を 0.5 秒後退させてから 0.5 秒を止める簡単なプログラムを書く。

UNO コントローラボードをコンピュータに接続し、“Lesson 1 Make The Car Move ¥ forward_back ¥ forward_back.ino”というパスでコードファイルを開きます。 プログラムを UNO ボードにアップロードします。

| tutorial | Lesson 1 Make The Car Move | forward_back | ▼ | 🔄 |
|--------------|----------------------------|--------------|---|---|
| 名称 | 修改日期 | 类型 | | |
| forward_back | 2016/6/12 12:30 | Arduino file | | |

Code preview:

```
//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);
}
```

プログラムを UNO ボードにアップロードし、コンピュータから切り離してから、車の電源を入れます。あなたは、あなたが期待どおりに右の車輪が動くのを見るでしょう。

The third step: Write the program

自動車を自動的に動かすためにプログラム全体を書くのは難しいかもしれません。したがって、動きを別々の機能に分けることができます（たとえば、前進と左回り）。最後のステップでプログラムを書くときに、関数を呼び出すことができます。

次に、各ムーブメントのプログラムを作成します。

Code preview:

```
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");
}

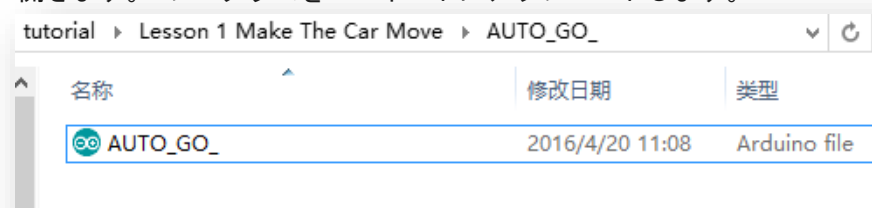
void right() {
  digitalWrite(ENA, HIGH);
  digitalWrite(ENB, HIGH);
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  Serial.println("Right");
}

```

The fourth step: Move automatically(自動的に動く)

我々は、車を自動的に移動させるためのプログラムを書き始めます :前進 0.4 秒 - 後退 0.4 秒 - 左 0.4 秒 - 右 0.4 秒進みます。

UNO コントローラボードをコンピュータに接続し、コードファイルを "Lesson 1 Make The Car Move ¥ AUTO_GO_ ¥ AUTO_GO_.ino"ディレクトリに開きます。 プログラムを UNO ボードにアップロードします。



Code preview:

```

//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 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");
}
```

```

//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);
}


```

それをコンピュータから取り外し、車の電源を入れます。あなたは、あなたが期待どおりに右の車輪が動くのを見るでしょう。

The fifth step: speed_control

機能を達成するためのコードは、車の速度を制御することです：前方に行くと速度を減らす→ stop 1s →走って加速する→ stop 2s.

UNO コントローラボードをコンピュータに接続し、“Lesson 1 Car Move ¥ speed_control ¥ speed_control.ino”ディレクトリにコードファイルを開きます。プログラムを UNO ボードにアップロードします。

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

Code preview:

```

//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);  
}
```