

Lesson1ー車を動かす

このセクションのポイント

学習パート：

- ◆ 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

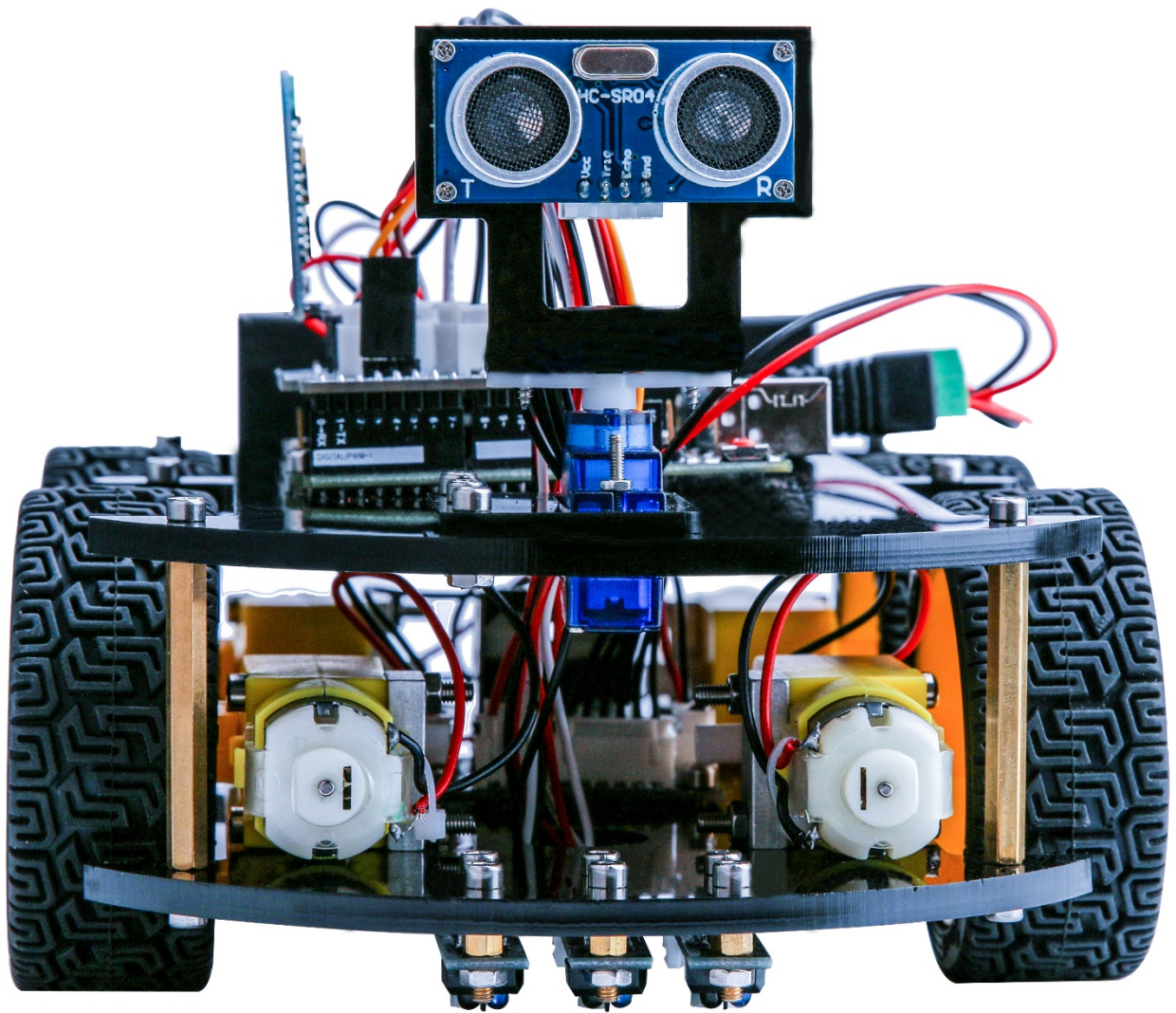
準備：

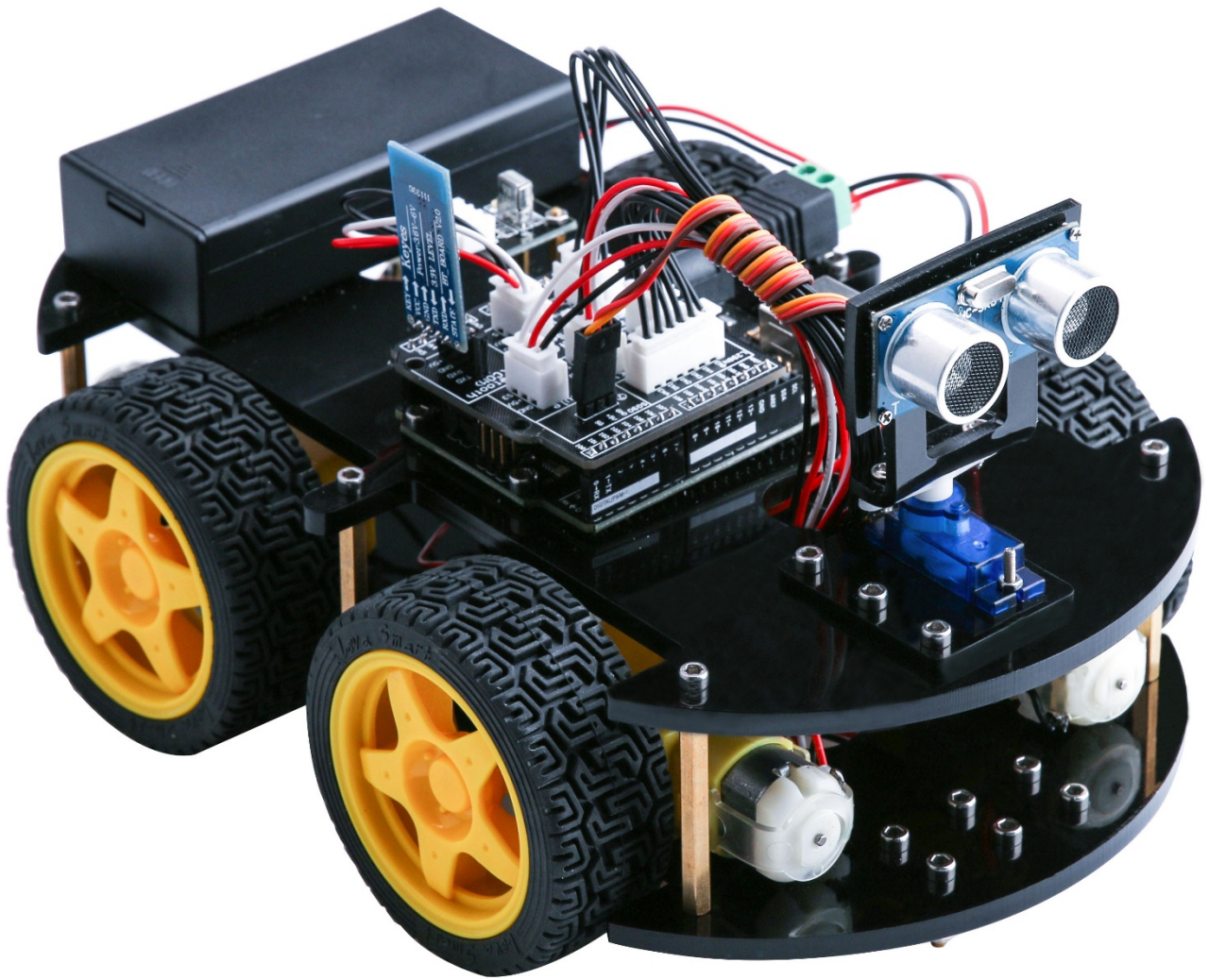
- ◆ 1 台の車（バッテリー付き）
- ◆ USB ケーブル 1 本

I. 車の紹介

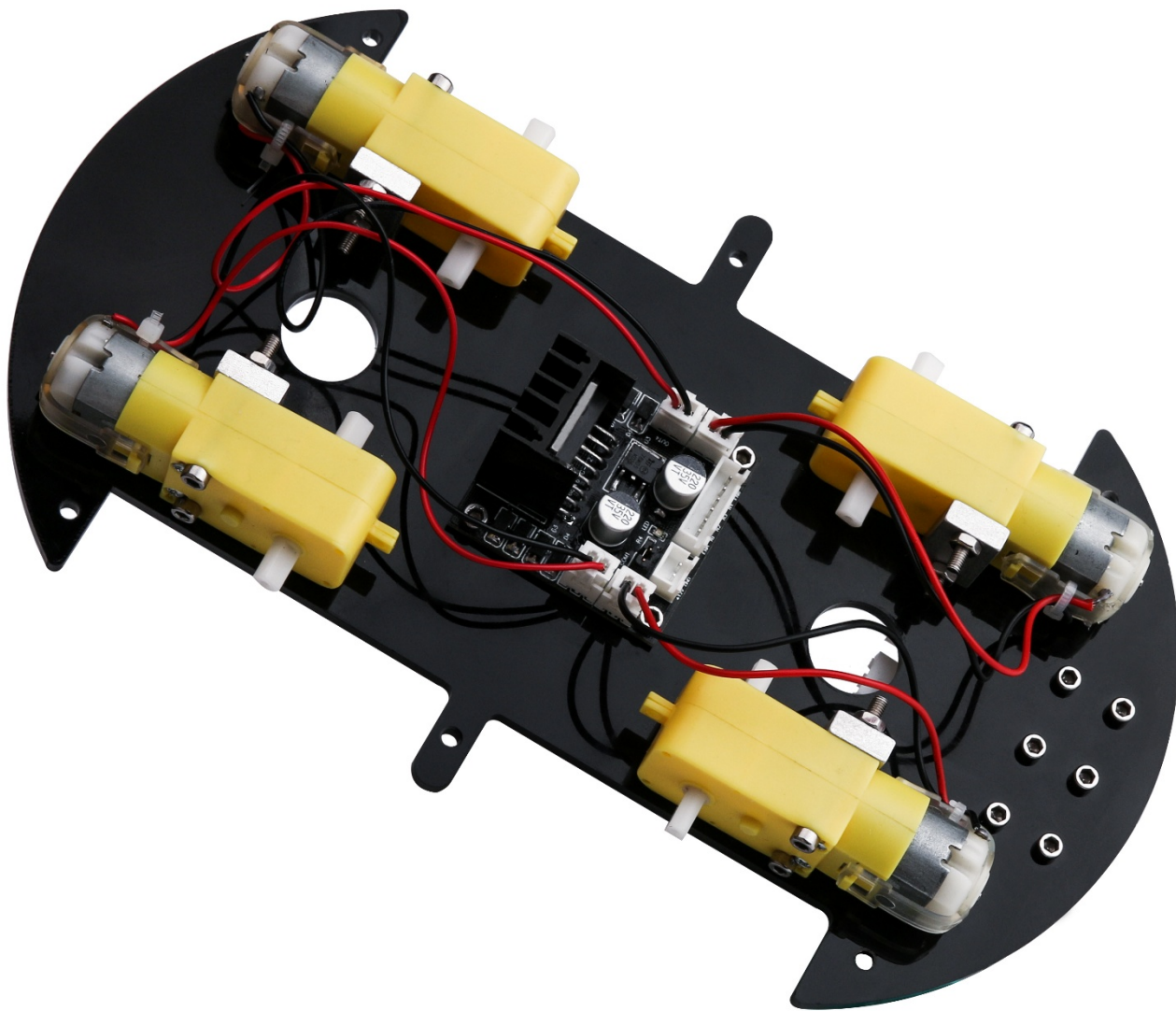
このキットは、特に教育、競技、娯楽の目的で設計された非常に柔軟な車両用キットです。キットの上部パネルは、9 グラムステアリングエンジンと直接互換性があります。また、超音速センサー、バッテリー、その他の固定穴を持ち、さまざまなセンサーの取り付けを容易にします。これは、学習や制作の目的に合う非常に面白く多用途のロボットです。これにより、Bluetooth や赤外線リモコン、障害の自動回避、回線検査など、さまざまな興味深いアイデアを実装できます。

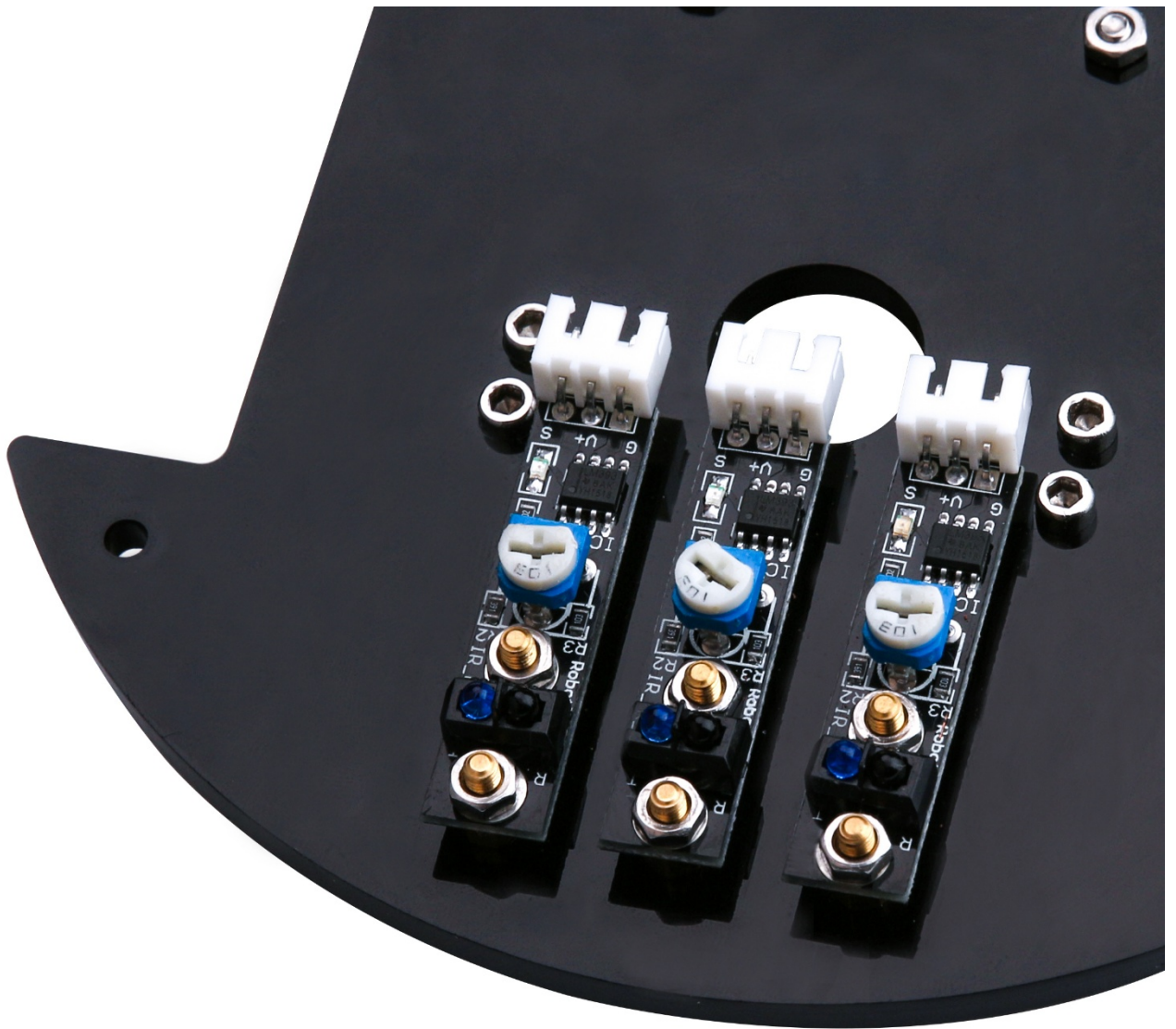
これから私たちと行動を共にする小さな車両について説明しましょう。

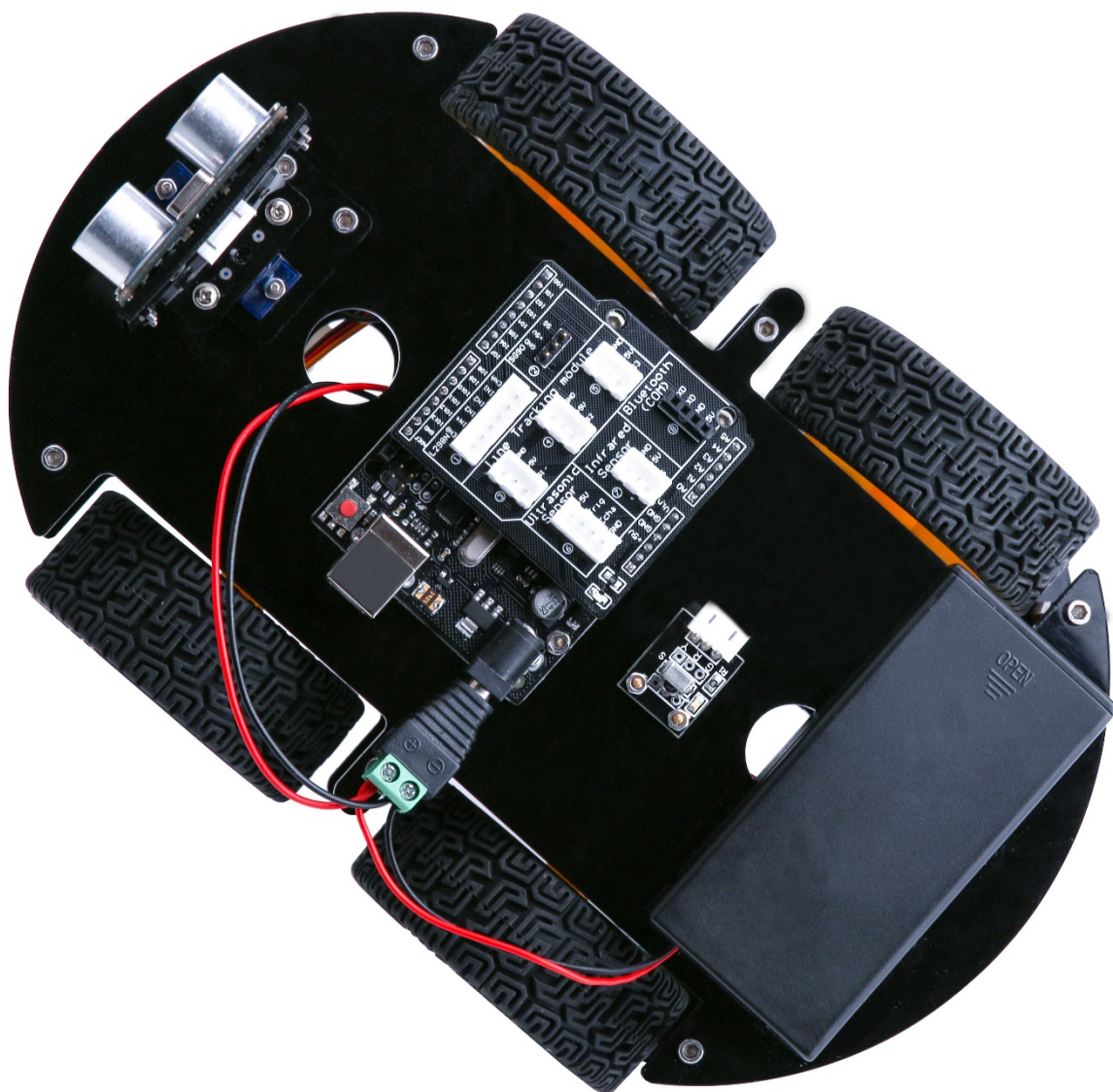




車の部品は下図のとおりです:







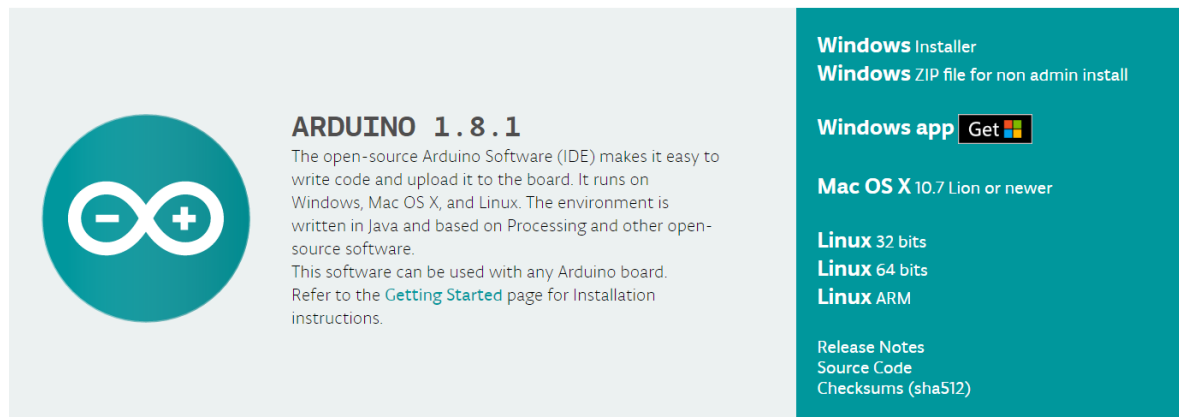
各部の機能：

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回転させます。
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
- 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. プログラムをアップロードする

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。

STEP1 :<https://www.arduino.cc/en/Main/Software> にアクセスし、以下のページを見つけます。



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: 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ます。ここで Windows を例に取る。

Windows Installer

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。

Support the Arduino Software

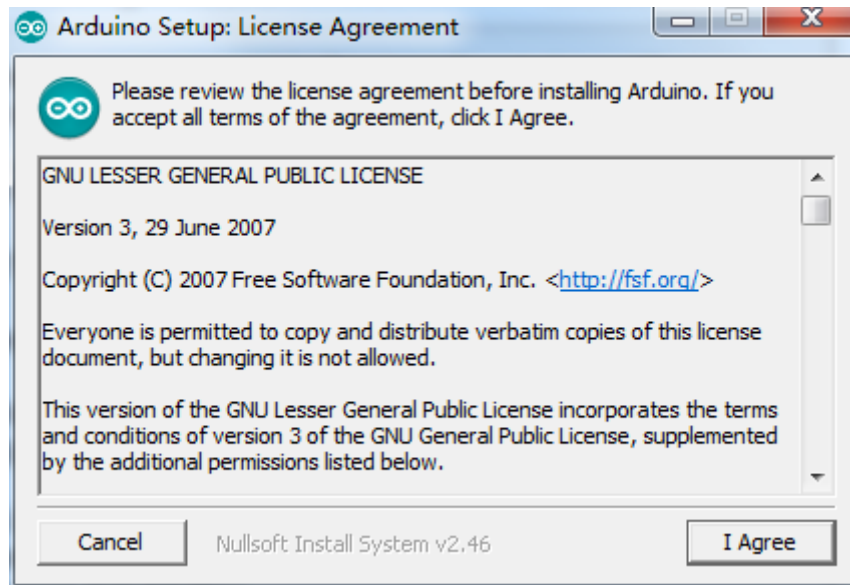
Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



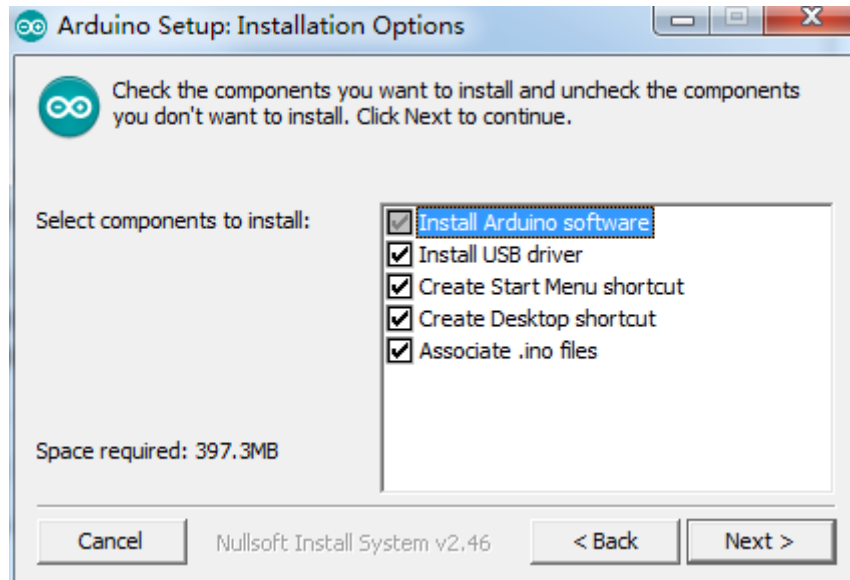
JUST DOWNLOAD

“JUST DOWNLOAD”ボタンを押してソフトウェアをダウンロードします。

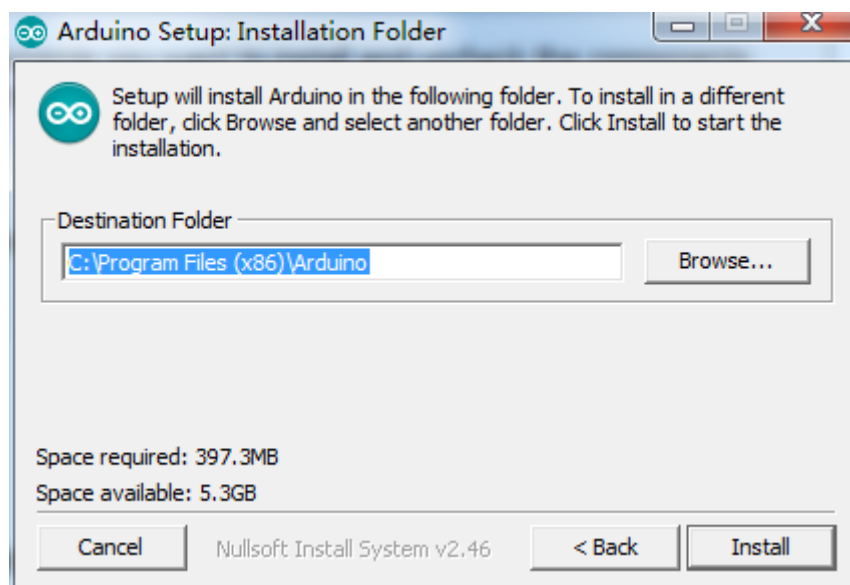
これらは私たちが提供している資料で入手できます。また、私たちの教材のバージョンは、このコースが作成されたときの最新バージョンです。



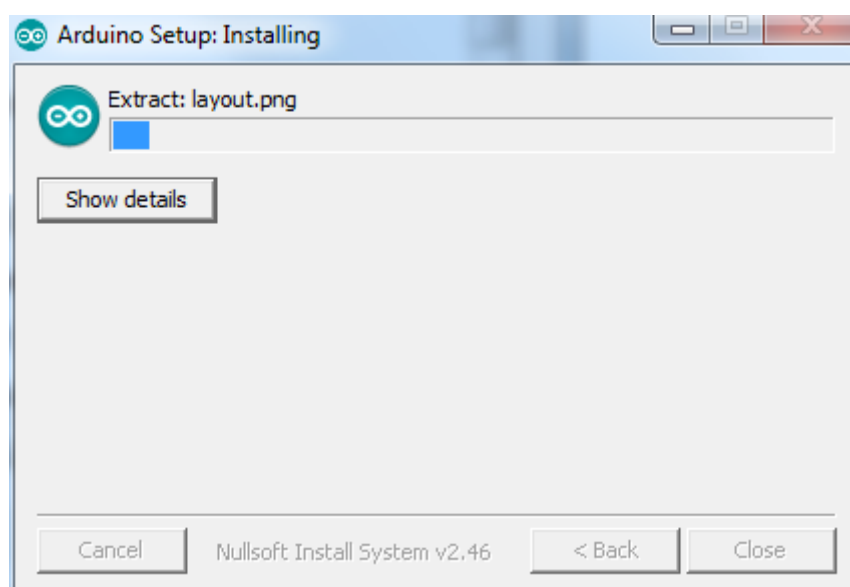
「同意する」を選択すると、次のインターフェースが表示されます。



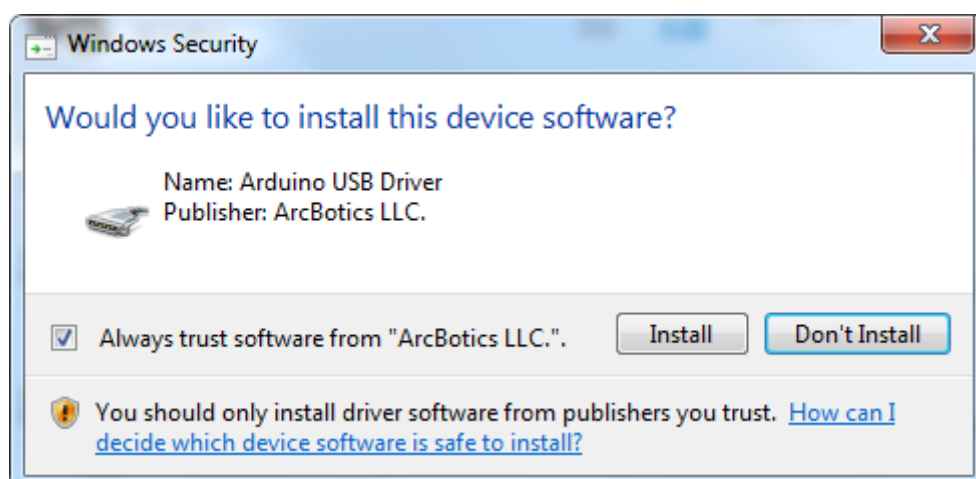
次を選択



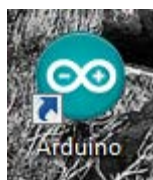
インストールを開始するには、[Install]を押します。



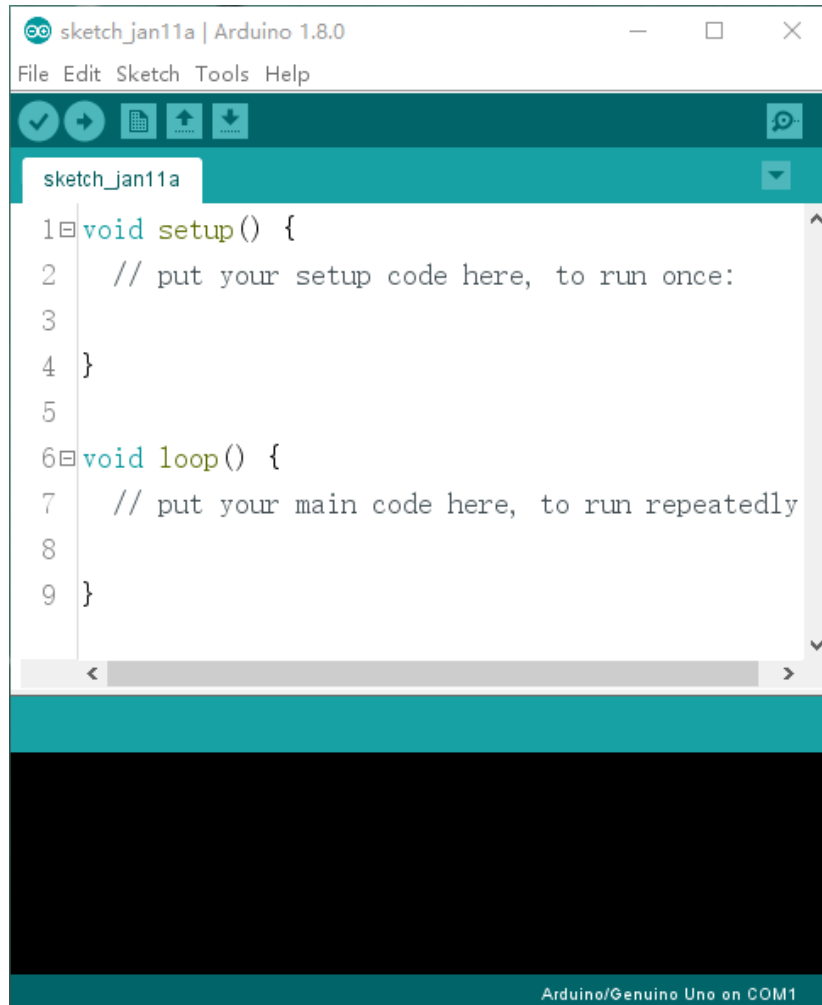
最後に、次のインターフェイスが表示されたら、[インストール]を選択して開発の正確性を確認します。



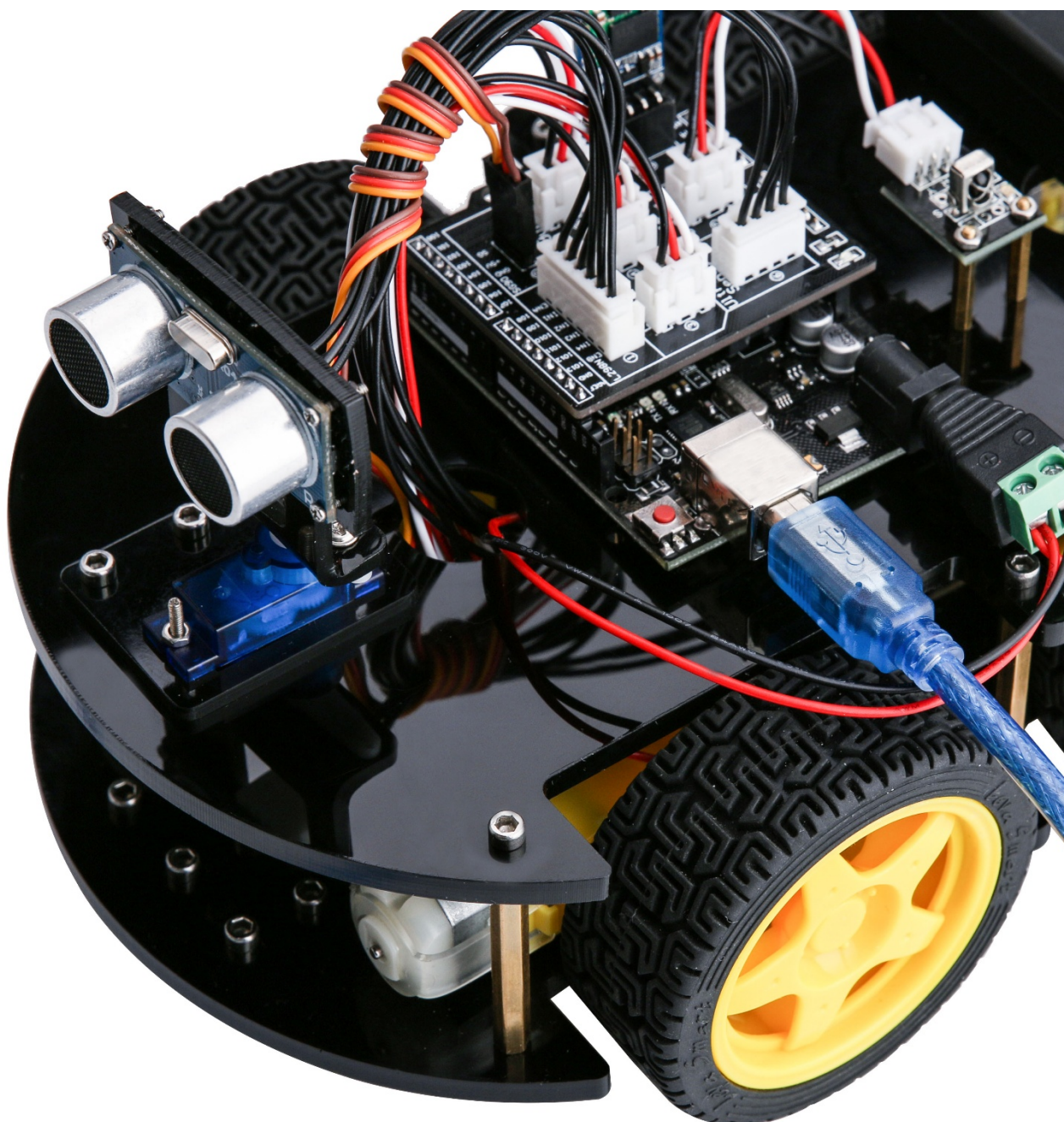
次に、デスクトップに次のアイコンが表示されます。



ダブルクリックして希望の開発環境に入ります。



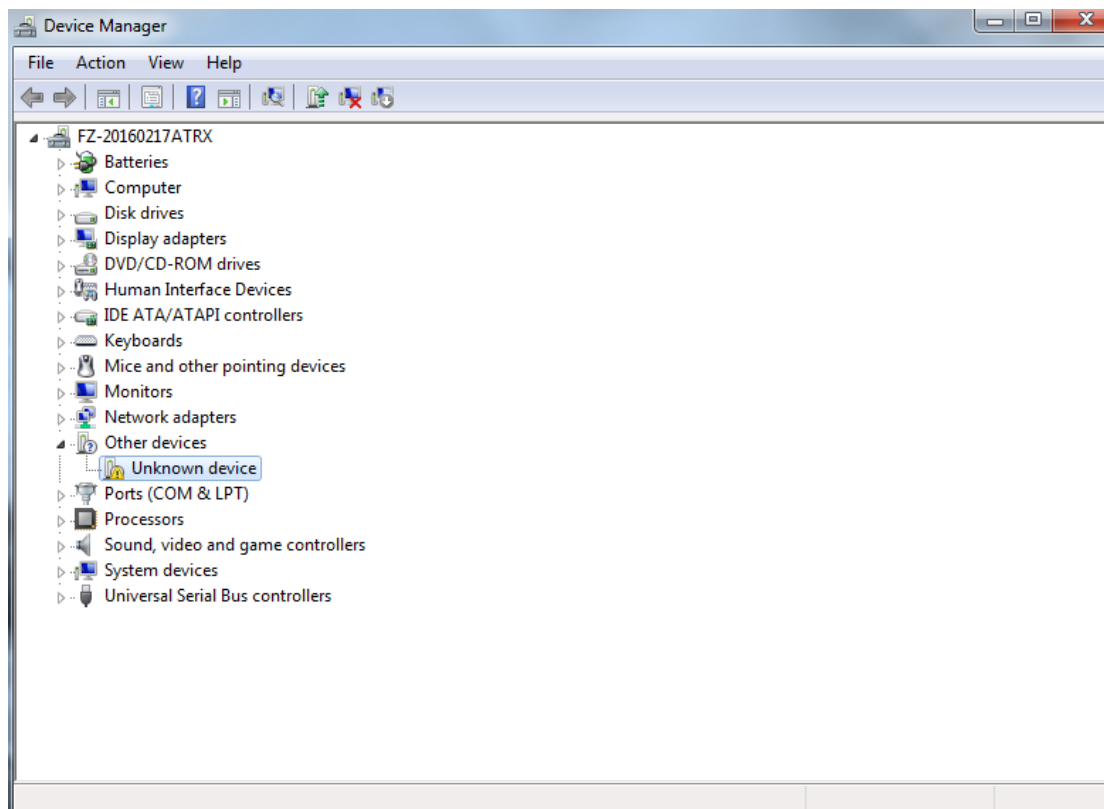
STEP3 : コンピュータに車を接続します。



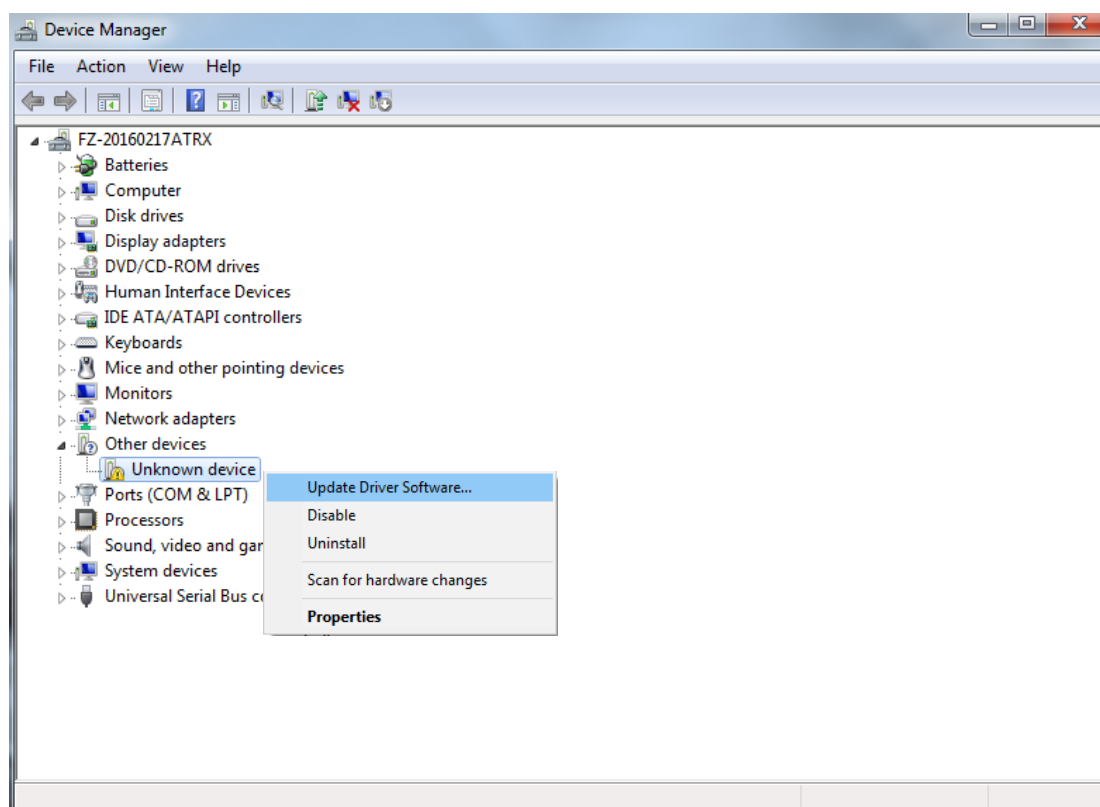
プログラムを UNO ボードにアップロードし、コンピュータから切り離してから、車の電源を入れます。（ヒント：毎回プログラムをアップロードするときに Bluetooth モジュールを取り外す必要があります。そうしないと、プログラムのアップロードに失敗します）。

STEP 4 : IDE - ツール - ポートを開きます。正しいポート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。

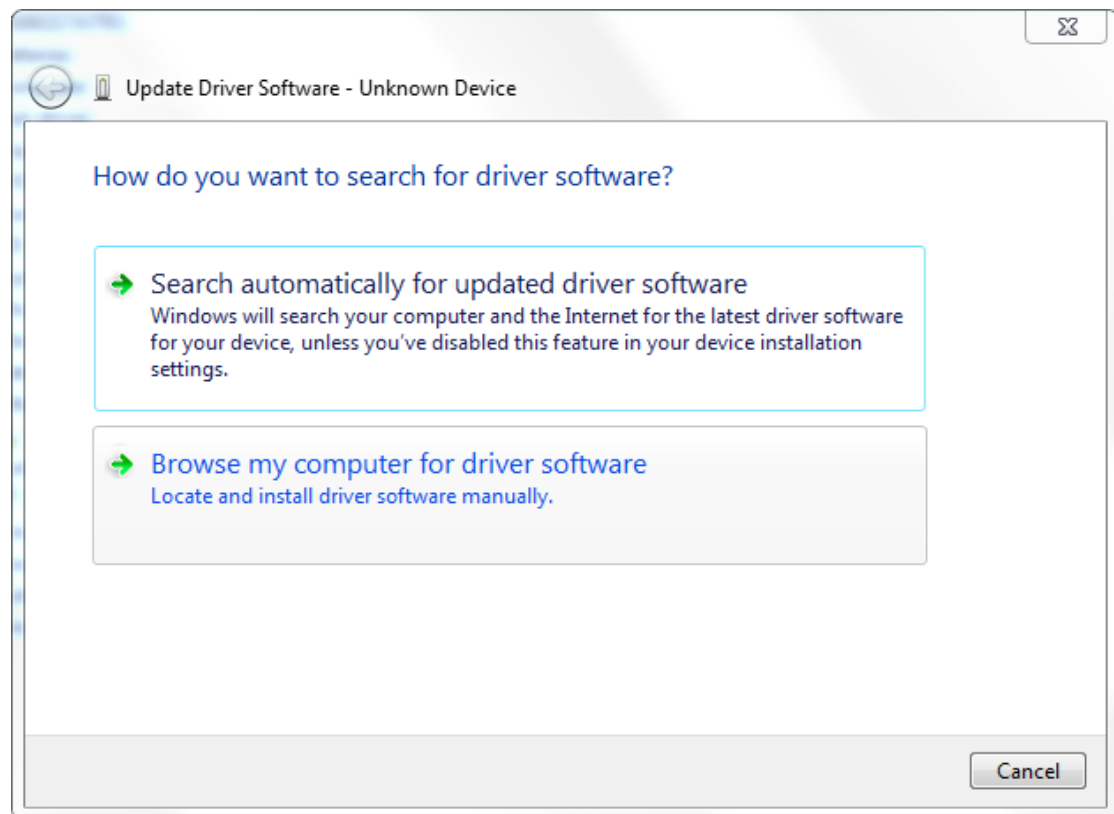
[マイコンピュータ] - [管理] - [デバイスマネージャ] を右クリックして [デバイスマネージャ] を開きます。



不明なデバイスを右クリック-----デバイスソフトウェアを更新する。



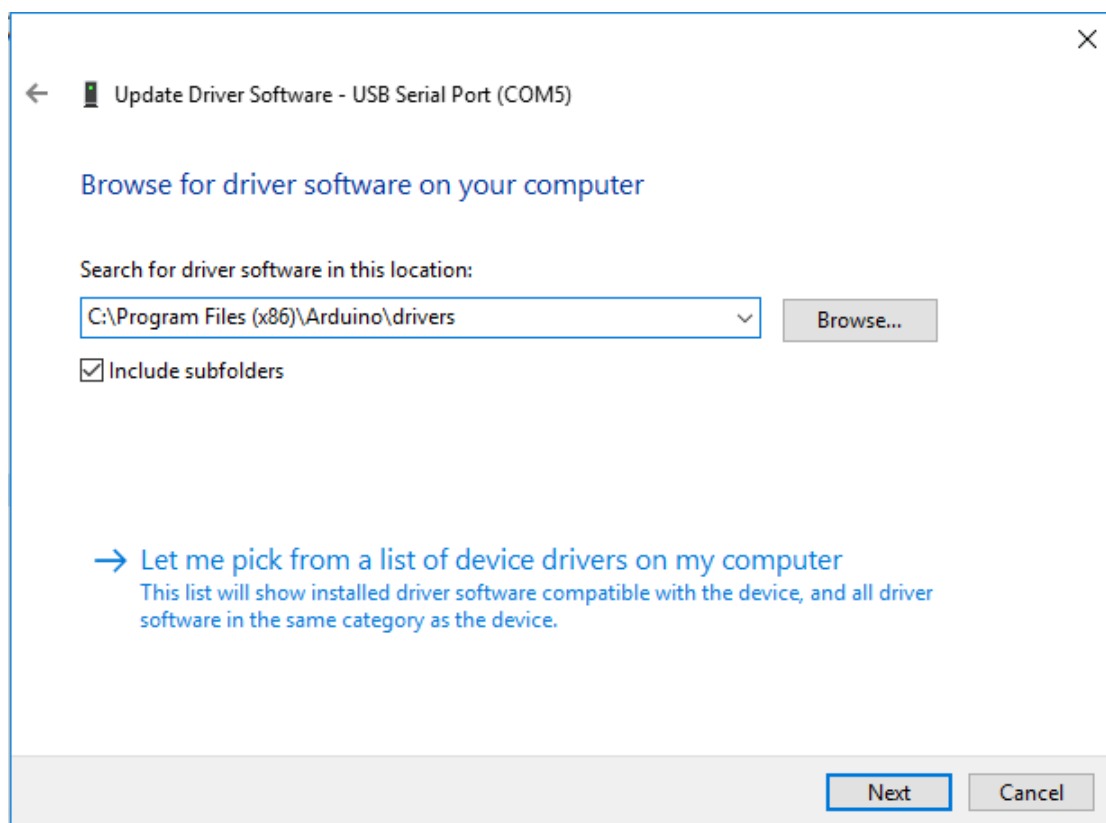
ドライバがインストールされていないことが表示され、[コンピュータを参照してドライバソフトウェアを検索する]をクリックしてドライバを検索する必要があります。ドライブはArduino フォルダにあります。通常、C:\Program Files (x86)\Arduino にフォルダをインストールします。



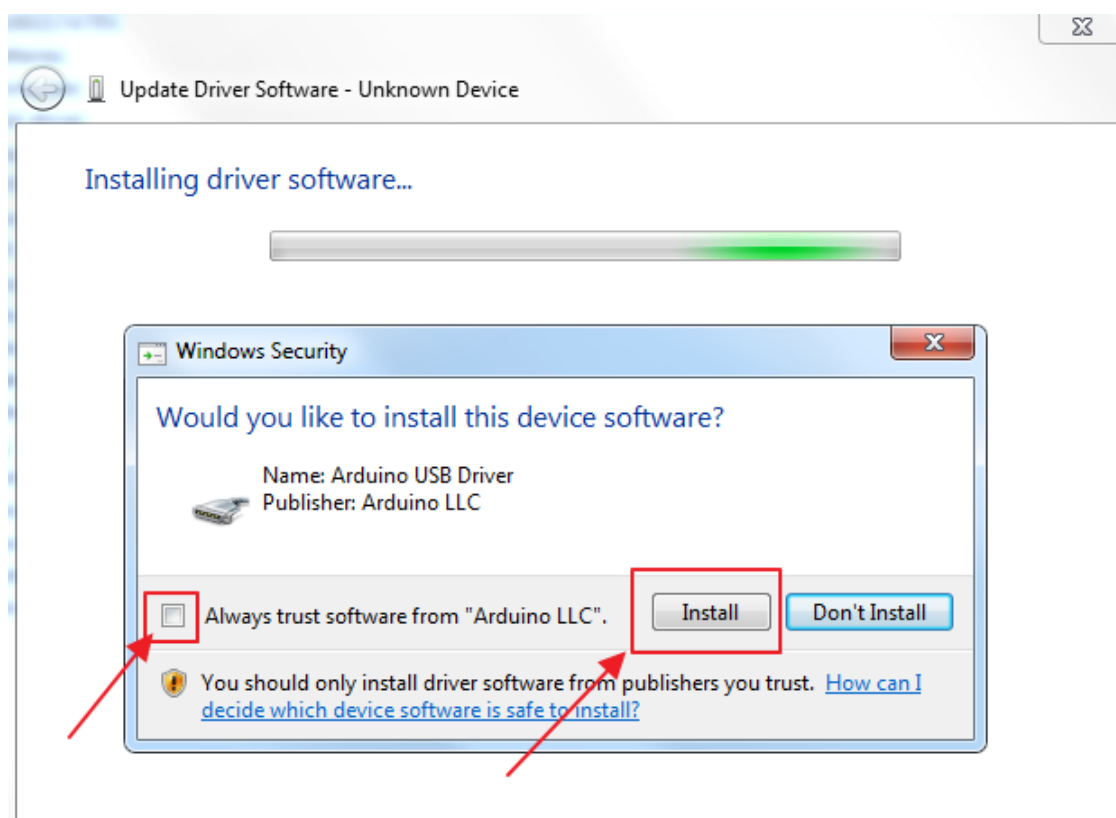
Arduino インストールフォルダ

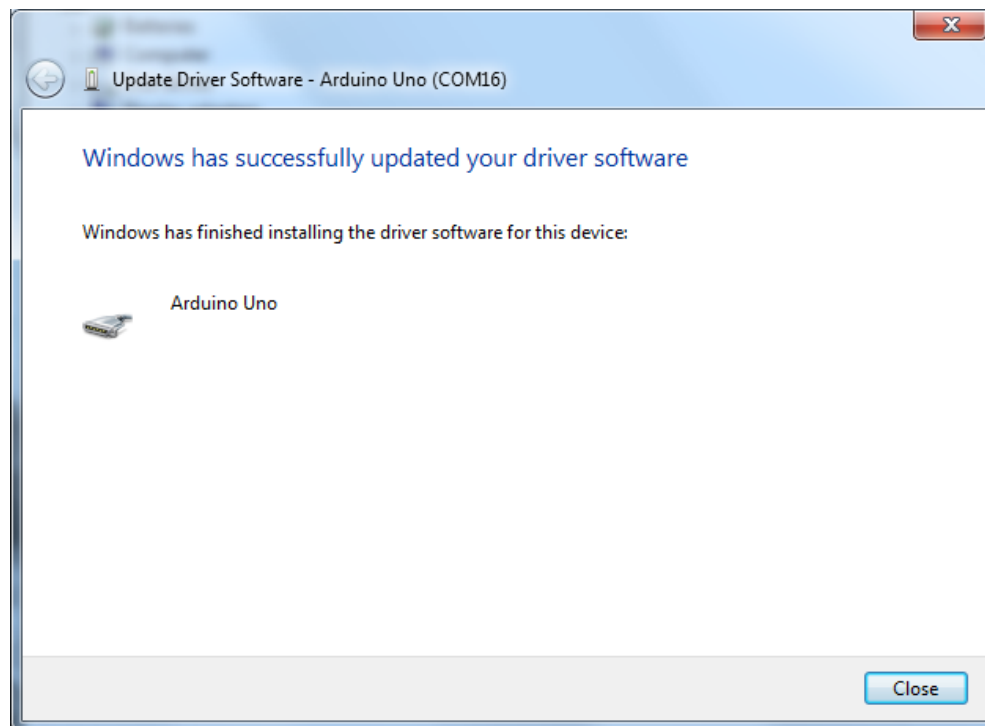
> This PC > Win10 (C:) > Program Files (x86) > Arduino >				
Name	Date modified	Type	Size	
drivers	1/16/2017 9:18 AM	File folder		
examples	1/16/2017 9:18 AM	File folder		
hardware	1/16/2017 9:18 AM	File folder		
java	1/16/2017 9:18 AM	File folder		
lib	1/16/2017 9:18 AM	File folder		
libraries	1/16/2017 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	

Arduino ドライバフォルダを選択します。

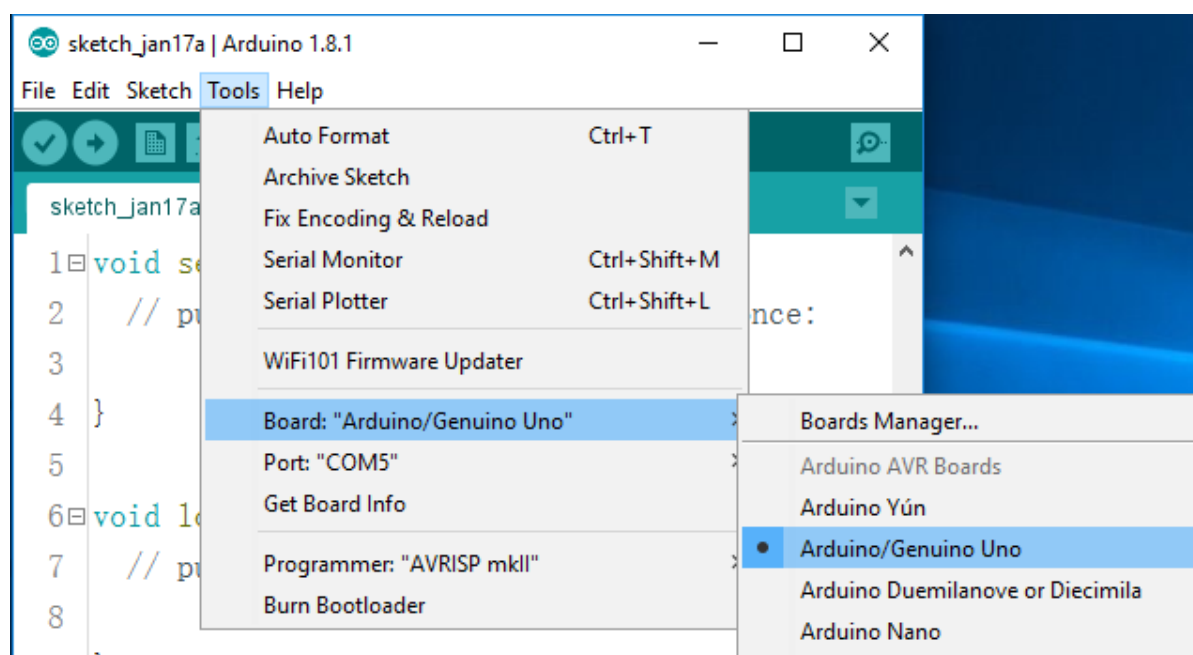


Arduino USB デバイスをインストールする。

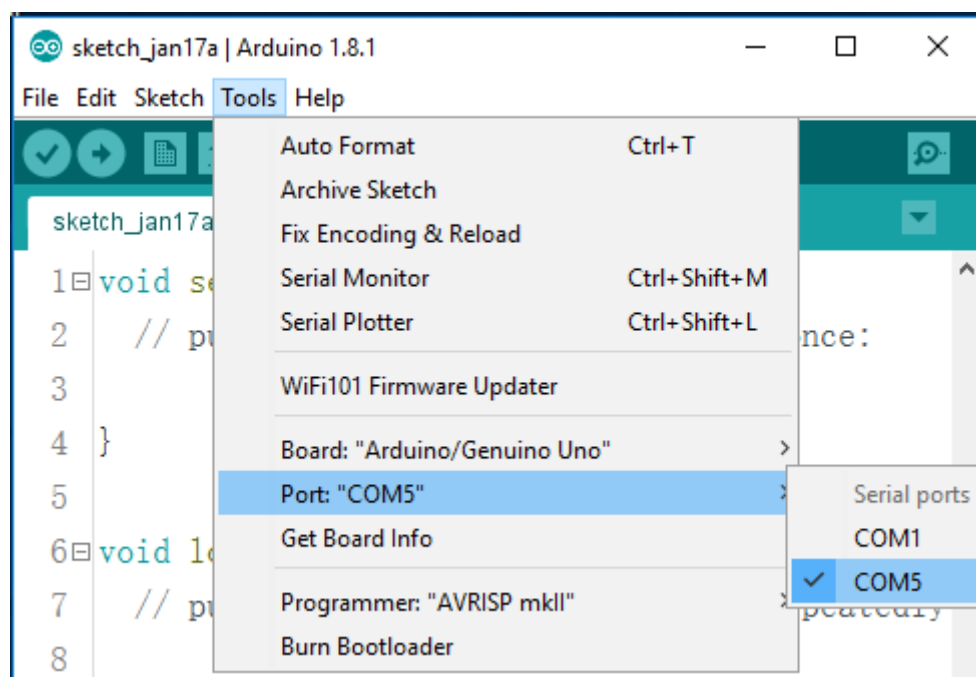




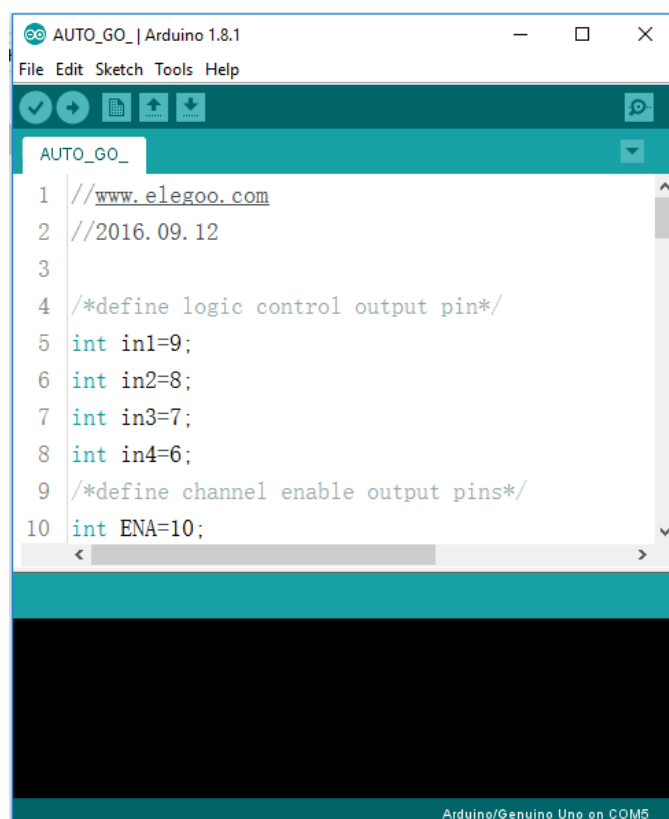
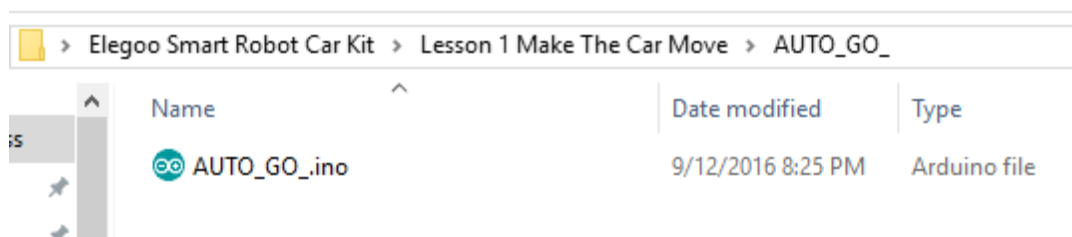
STEP5 : ドライバがインストールされたら、IDE を開き、ツール—ボード— Arduino / Genuino Uno をクリックしてください。

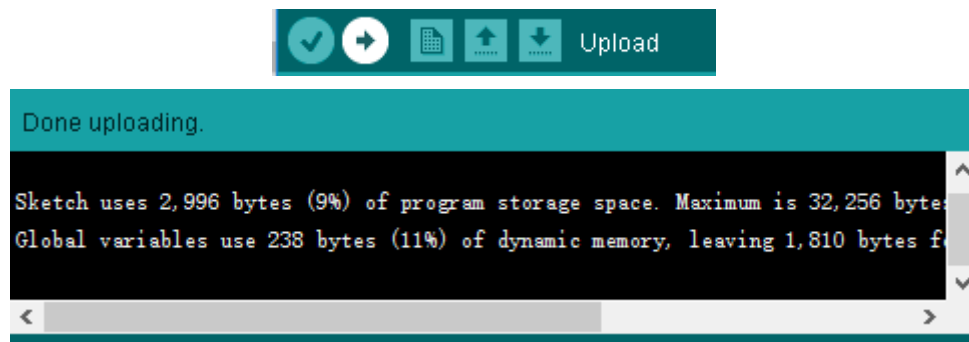


STEP6 : ツール——ポート—— COMx (Arduino / Genuino Uno) をクリックします。



STEP7 : AUTO_GO_ ¥ AUTO_GO_.ino ファイルを開き、UNO コントローラボードにアップロードします

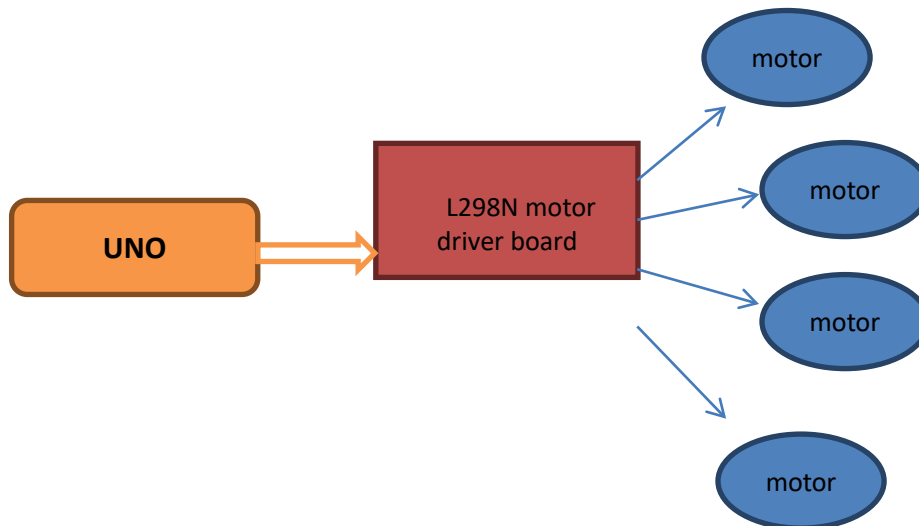




上の写真はアップロードが成功したことを示しています。

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。 その後車が動いているのが確認できるはずです。

Ⅲ. 原則の説明



・ L298N motor driver board のインストール方法

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 はバッテリーボックス GND に接続されています。

L298N VCC はバッテリーボックス VCC に接続されています。

UNO ボードはバッテリーボックスにも接続されています。

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 は右モータを制御するために使用され、IN3 と IN4 は左モータを制御するために使用されます。 原則については、下記のシートをご覧ください：（例えば右モータを取る）

ENA	IN1	IN2	DC モータの状態
0	X	X	停止
1	0	0	制動
1	0	1	前進
1	1	0	後進
1	1	1	制動

IV. 車を動かす

第1ステップ：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、Arduino を使って 5, 6, 7, 8, 9, 10 ピンを使って車を制御します。 6 と 7 ピンは右のホイールを制御します。 8 ピンと 9 ピンは左のホイールを制御します。 5 ピンと 10 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 拡張ボード
ENA	5
IN1	6
IN2	7
IN3	8
IN4	9
ENB	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、逆方向に 0.5s、0.5s を停止する単純なプログラムを設計します。 ホイールはこの動作を繰り返します。

UNO コントローラ ボードをコンピュータに接続し、ファイルを開きます right_wheel_rotation ¥ right_wheel_rotation.ino



コードは次のとおりです：

```
/*In1 connected to the 9 pin,  
In2 connected to the 8 pin, ENA pin 10,*/  
int ENA=5;  
int IN1=6;  
int IN2=7;  
void setup()  
{  
  pinMode(IN1,OUTPUT);  
  pinMode(IN2,OUTPUT);  
  pinMode(ENA,OUTPUT);  
  digitalWrite(ENA,HIGH);  
}
```

```

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

```

プログラムを UN0 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。

車が動いていない場合は、UN0 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。

UN0 コントローラボードをコンピュータに接続し、ファイルを開きます。Left_wheel_rotation ¥ Left_wheel_rotation.ino



コードは次のとおりです:

```

/*In3 connected to the 7 pin,
  In4 connected to the 6 pin, ENB pin 5, */
int ENB=11;
int IN3=8;
int IN4=9;
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);
}

```

プログラムを UN0 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。 期待どおりに右の車輪が動くはずですが。

第 2 ステップ：前後に移動する

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。

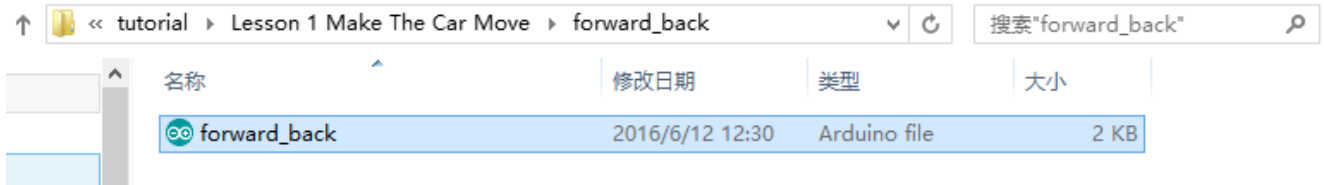
以下は車の動きの仕方です：

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、0.5 秒後に 0.5 秒後に 0.5 秒後に 0.5 秒停止するという簡単なプログラムを作成します。

UNO コントローラボードをコンピュータに接続し、ファイル forward_back ¥ forward_back.ino を開きます



コードは次のとおりです:

```
int ENA=5;
int IN1=6;
int IN2=7;
int ENB=11;
int IN3=8;
int IN4=9;
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);          // 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,LOW);
```



```

digitalWrite(IN2,HIGH);          //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);
}

```

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。期待どおりに右の車輪が動くはずです。

第3ステップ：プログラムを書く

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：

コードは次のとおりです：

```

/*****
**** 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  
}
```

第4ステップ：自動的に移動する

最後に、車を自動的に動かすためのプログラムを書いていきます:0.4s 前進 - 0.4s 後退 - 0.4s 左折 - 0.4s 右折。

UNO コントローラボードをコンピュータに接続し、ファイル AUTO_GO_ ¥ AUTO_GO_.ino を開きます。



コードは次のとおりです:

```
/*define logic control output pin*/
int in1=6;
int in2=7;
int in3=8;
int in4=9;
/*define channel enable output pins*/
int ENA=5;
int ENB=11;
/*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);
}
```

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。 期待どおりに右の車輪が動くはずです。