



Arduino

بسم الله الرحمن الرحيم السلام عليكم

الموضوع: كيفية انشاء مشروع بسيط لسيارة ذاتية القيادة بالآردوينو.

~ تمهيد ~

ما هي لوحة الآردوينو؟ (Arduino Board):

هي لوحة إلكترونية شبه ذكية مفتوحة المصدر، حيث لا توجد لها وظيفة معينة صنع لأجلها حيث تمكن للمستخدم من إنشاء العديد من المشاريع الصغيرة البسيطة أو الكبيرة المعقدة وله العديد من الاستخدامات، وللتحكم به نستخدم البرمجة وذلك بلغة «C» .

ما هي البرمجة ولغاتها؟ (Programming and its languages):

هي وسيلة نستعملها للتواصل مع الآلة بسهولة حيث نقوم بإعطاء أوامر للآلة ليقوم بتنفيذها ثم تعرض لنا النتائج بعدة طرق، وللمبرمجة عدة لغات وكل لغة ما يميزها عن الأخرى، ولكل لغة استعمال خاص وقواعد خاصة بها.

واللغة التي سنستعملها في هذا المشروع هي: «C».

ملاحظة: لغة «C» ليست مخصصة للآردوينو فقط فهي لغة عامة والآردوينو حالة خاصة منها حيث يتم الاستعانة ببرنامج خاص بالآردوينو لكتابة النص البرمجي (الأوامر) وهو برنامج «Arduino IDE» وهذا البرنامج مهم فهو يمكنك من التواصل بالآردوينو.

~ خطوات إنشاء المشروع ~

اللوازم هي:

رقم	الاسم	الشكل	كيفية الربط	الدور
01	لوحة أردوينو (Arduino Board)			العنصر الرئيسي والوحيد الذي سيتحكم بالسيارة
02	أسلاك حسب الحاجة (Cabels)			أسلاك للتوصيل والربط بين العناصر
03	لوحة التجارب (BreadBoard)			لتثبيت العناصر والتوصيل
04	عجلتين بمحرك 2 Wheels with DC) (Motor			لتحريك السيارة
05	عجلة محورية (Caster wheel)			موازنة السيارة
06	بطارية 9 فولط (Battery 9V)			تغذية العناصر
07	هيكل السيارة			لحمل العناصر
08	مقاومة متغيرة (Variabel Resistance)			لتعيين مسافة التوقف
09	لوحة التحكم بالمحركات (L298N Board)			للتحكم باتجاه دوران المحركات (العجلات)
10	جهاز قياس المسافة (Ultra Sonic 4 pin)			قياس المسافة الفاصلة بين السيارة والجدار

تدوير جهاز قياس المسافة في جميع الاتجاهات			محرك 180 درجة (Servo Motor 180°)	11
---	---	--	-------------------------------------	----

المشروع ينقسم لقسمين هما:

1. الجانب البرمجي (النص البرمجي).
2. الجانب الهيكلي (هيكل السيارة).

الجانب البرمجي:

- نقوم بتشغيل برنامج **Arduino IDE** لكتابة النص البرمجي وأول واجهة ستقابلنا:

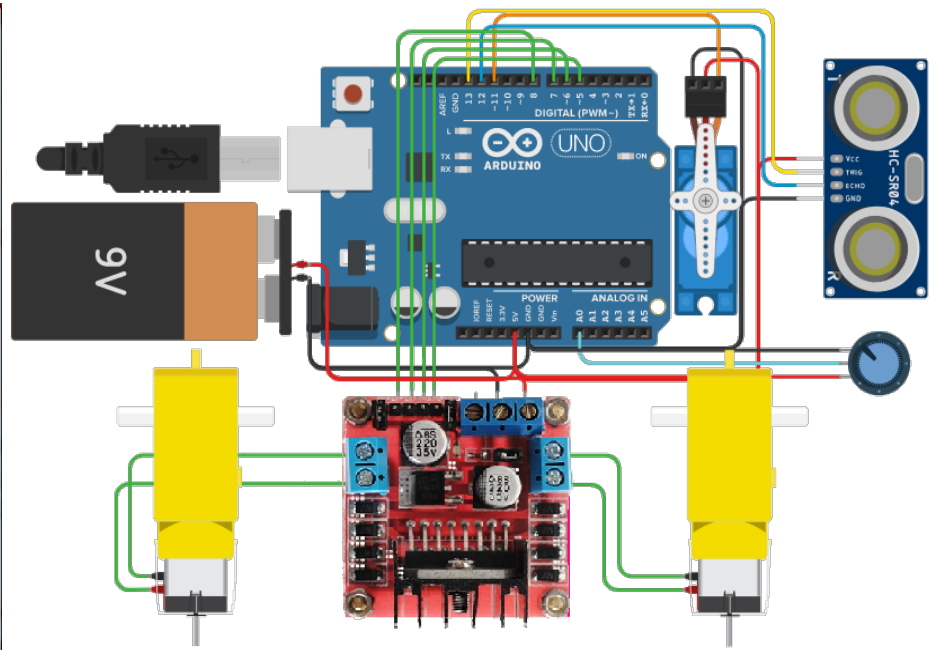
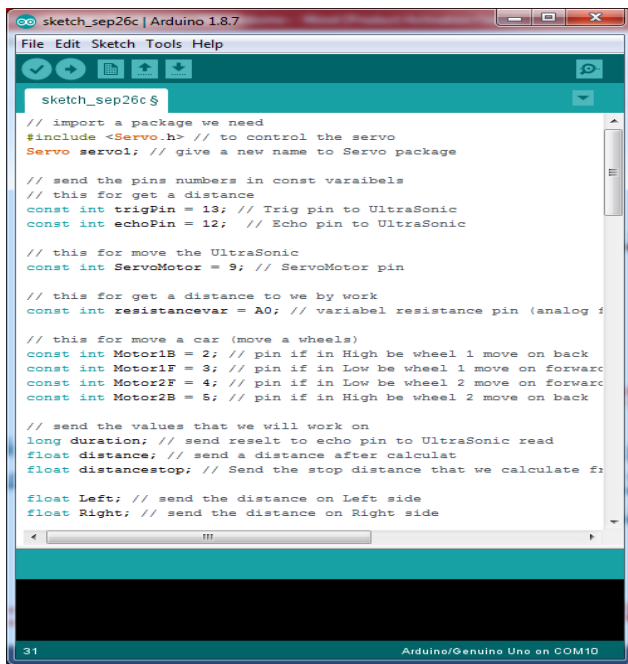


- كما نلاحظ بالصورة توجد دالتين أو قسمين: (void Setup, void loop)

Void setup: وهو المكان المخصص للأوامر أو الإعدادات الأولية مثل تحديد المخارج والمداخل التي نحتاجها بالأردوينو وهي تكون مرقمة من 0 إلى 13 أو أكثر على حسب نوع الأردوينو، وغيره من الإعدادات.

Void loop: وهو المكان المخصص للأوامر الأساسية التي سيطبقها الأردوينو باستمرار وتكرار.

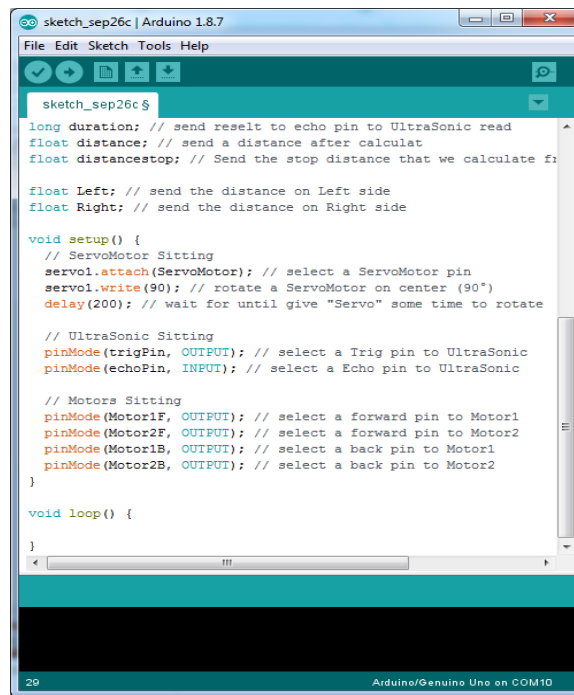
1. نقوم بتسمية المداخل التي نحتاجها في متغيرات ثابتة (أي لا يمكن تغيير قيمتها أي ذاكرته ثابتة) ونقوم بحجز متغيرات لتخزين فيها المعلومات كما في المخطط:



ملاحظات:

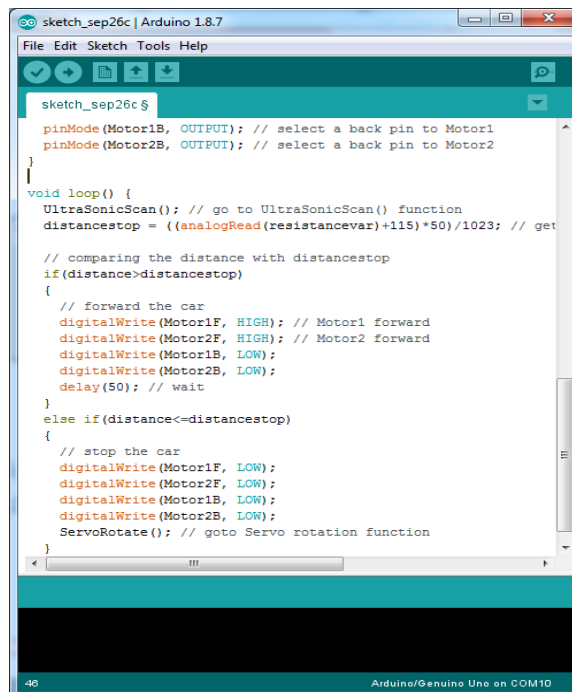
- تحديد المتغيرات لا يهم أن يكون داخل **void loop** أو **void setup** يمكنك وضعها في أي مكان شريطة أن يكون تعيين المتغير قبل استعماله (يكون ترتيبه هو الأول) فلا يمكن استعماله قبل تعيينه، والأحسن تعيينها خارج الدوال أو في الدالة (**void setup**).
- الأمر (**#include <Servo.h>**) دوره استدعاء ملف (مكتبة جاهزة) به مجموعة من أوامر المعدة مسبقاً فهي تساعد باختصار الكود وهي خاصة بـ «**Servo Motor**» .
- الأمر (**Servo servo1**) دوره إعطاء اسم جديد للمكتبة.
- الكلمات (**float, long, int, const int**) هي لتحديد نوع المتغير وهي على الترتيب (رقم حقيقي، ثابت، رقم حقيقي، رقم طويل، رقم عشري).
- الكتابة التي تأتي بعد (**// text**) أو التي بين الرمزین (**/* text */**) تعتبر تعليقاً ولا تؤثر في الكود حيث يتم تخطيها عند تنفيذه.
- الفرق بين الرمزین (**//text** و **/* text */**) هو أن الأول ينتهي التعليق بانتهاء السطر، أم الثاني ينتهي التعليق قبل الرمز (***/) حيث يمكن كتابة العديد من أسطر التعليقات شريطة أن تكون داخل الرمز.**
- الحرص على عدم الخلط بين الأحرف الصغيرة والكبيرة فهي تؤثر بالكود.
- الرمز (**;**) يدل على إنتهاء الأمر فبها يتم غلق الأمر وهو ضروري ويمكن أن يكون لأمر واحد أكثر من سطر.

2. ضبط الإعدادات في **void setup**: نقوم بإعداد المداخل التي نحتاجها حيث بدلاً من وضع الأرقام نقوم بوضع أسماء المتغيرات الثابتة التي تم تعيينها سابقاً



ملاحظات:

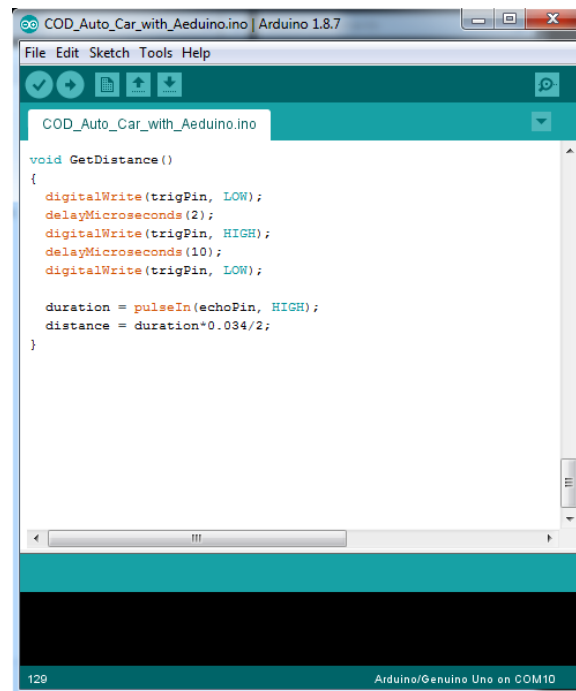
- يجب الحرص أن تكون الأوامر داخل ({Commands}).
- الأمر (servo1.attach(ServoMotor)) دوره تحديد المخرج الخاص بـ (Servo Motor) الذي من خلاله يتحكم به.
- الأمر (servo1.write(90)) دوره إعطاء الأمر لـ (Servo Motor) بالدوران للزاوية التي نريد (في هذا المثال الزاوية هي 90).
- الأمر (delay(200)) دوره التأخير (الانتظار) ويكون اختياريًا (في هذا مثال قدر التأخير هو 200ms).
- الأوامر (pinMode(Pin, OUTPUT)) دوره تحديد المداخل والمخارج، حيث (INPUT, OUTPUT, Pin) هي على الترتيب (رقم المدخل أو المخرج، في حالة المخرج، في حالة المدخل).
- الرمز (,) مهم للفصل بين القيم في الأوامر.
- 3. الأوامر الأساسية void loop: نقوم بتحديد الأوامر الأساسية التي سيطبقها الأردوينو بصفة دورية (بالتكرار) كما في الشكل:



ملاحظات:

- الأمر **(GetDistance())** دوره تطبيق دالة (مجموعة من الأوامر) جاهزة هذه المرة ليست بملف (مكتبة) أي أنه يتم إعدادها داخل الكود وهي مشابهة لـ **(void setup, void loop)**، الاختلاف أنها لا يتم تطبيقها إلا إن تم استدعائها.
- **(distancestop = ((analogRead(resistancevar)+115)*50)/1023)** دوره حساب مسافة التوقف (المسافة بين الجدار والسيارة) حيث القيمة الأعظمية للمقاومة المتغيرة هي **(1023)**، أما المسافة الأعظمية فهي اختيارية (سبب زيادة العدد 115 لكي لا تكون النتيجة معدومة).
- الأمر **(if(distance>distancestop))** دوره المقارنة بين المسافة الحالية بمسافة التوقف، إن تحقق الشرط يطبق الأوامر التي بداخله، نفسه مع الأوامر المماثلة التي بها كلمة **(if)**.
- العبارة **(else if)** تعني إن لم يتحقق الشرط السابق تحقق **(جرب)** من هذا الشرط.
- الأمر **(digitalWrite(pin, HIGH))** دوره تعيين إشارة الخروج في المدخل أو المخرج المحدد، حيث **(HIGH, LOW)** هي على الترتيب **(1, 0)**.
- قيم الـ **(HIGH, LOW)** كهربائيا هي على الترتيب **(3.3v, 0v)**.

4. دالة حساب المسافة **GetDistance**: نقوم بتحديد الأوامر التي سيطبقها الأردوينو عند استدعاء الدالة، حيث ستكون الأوامر متعلقة بحساب المسافة باستخدام جهاز **(UltraSonic)**.



```
void GetDistance ()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);
  distance = duration*0.034/2;
}
```

ملاحظات:

- الأمر **(delayMicroseconds(2))** دوره التأخير (الإنظار) بالميكرو ثانية حيث **(1μs = 1.10⁻⁶s)** ويكون اختياريا (في هذا مثال قدر التأخير هو **2μs**).

- الأمر `((duration = pulseIn(echoPin, HIGH))` دوره قراءة وقت النبضة وحفظها في المتغير `(duration)`.
- الأمر `(distance = duration*0.034/2)` دوره حساب المسافة، وهو يوافق العبارة `(المسافة = (وقت النبضة / 2) × (سرعة الصوت في الهواء [340 م / ث]))` وحفظها في المتغير `(distance)`.

5. دالة مقارنة المسافات في الجهتين **ScanAllSides**: نقوم بتحديد الأوامر التي سيطبقها الأردوينو عند استدعاء الدالة، حيث ستكون الأوامر متعلقة بتوجيه السيارة للمسافة الأبعد مع الأخذ بعين الاعتبار المسافة المحددة من المقاومة المتغيرة حيث يجب أن تكون المسافة الأكبر في الجهتين أكبر من المسافة المحددة.

```

COD_Auto_Car_with_Aeduno.ino | Arduino 1.8.7
File Edit Sketch Tools Help
COD_Auto_Car_with_Aeduno.ino $
void ScanAllSides()
{
  // get the distance on Right side
  servol.write(45); // rotate the Servo on Right side (45°)
  delay(1000); // wait 1min
  GetDistance(); // start scan for get a distance
  Right = distance; // send the resulting distance on the "Right"
  delay(1000); // wait 1min
  // get the distance on Left side
  servol.write(135); // rotate the Servo on Left side (135°)
  delay(1000); // wait 1min
  GetDistance(); // start scan for get a distance
  Left = distance; // send the resulting distance on the "Left"
  delay(1000); // wait 1min
  servol.write(90); // return to the center by rotate the Servo
  delay(1000);
  // compare distances
  if (Right > Left > distancestop) // Move to Right Side
  {
    digitalWrite(Motor1F, LOW);
    digitalWrite(Motor1B, HIGH);
    digitalWrite(Motor2F, HIGH);
    digitalWrite(Motor2B, LOW);
    delay(2000);
    digitalWrite(Motor1F, HIGH);
    digitalWrite(Motor2F, HIGH);
    digitalWrite(Motor1B, LOW);
    digitalWrite(Motor2B, LOW);
  }
}

```

```

COD_Auto_Car_with_Aeduno.ino | Arduino 1.8.7
File Edit Sketch Tools Help
COD_Auto_Car_with_Aeduno.ino $
digitalWrite(Motor2F, HIGH);
digitalWrite(Motor2B, LOW);
delay(2000);
digitalWrite(Motor1F, HIGH);
digitalWrite(Motor2F, HIGH);
digitalWrite(Motor1B, LOW);
digitalWrite(Motor2B, LOW);
}
else if (Left > Right > distancestop) // Move to Left Side
{
  digitalWrite(Motor1F, HIGH);
  digitalWrite(Motor1B, LOW);
  digitalWrite(Motor2F, LOW);
  digitalWrite(Motor2B, HIGH);
  delay(2000);
  digitalWrite(Motor1F, HIGH);
  digitalWrite(Motor2F, HIGH);
  digitalWrite(Motor1B, LOW);
  digitalWrite(Motor2B, LOW);
}
else
{
  digitalWrite(Motor1F, LOW);
  digitalWrite(Motor1B, HIGH);
  digitalWrite(Motor2F, LOW);
  digitalWrite(Motor2B, HIGH);
  delay(2000);
}
}
}

```

الجانب الهيكلي:

1. تحضير الهيكل (الشكل 07):

- يجب أن يكون الهيكل ملئاً لحمل جميع العناصر.
- يمكن صناعة هيكل يدويا شرط أن يكون قادر على حمل العناصر.
- ثقب الأماكن التي سيتم تثبيت فيها العناصر.

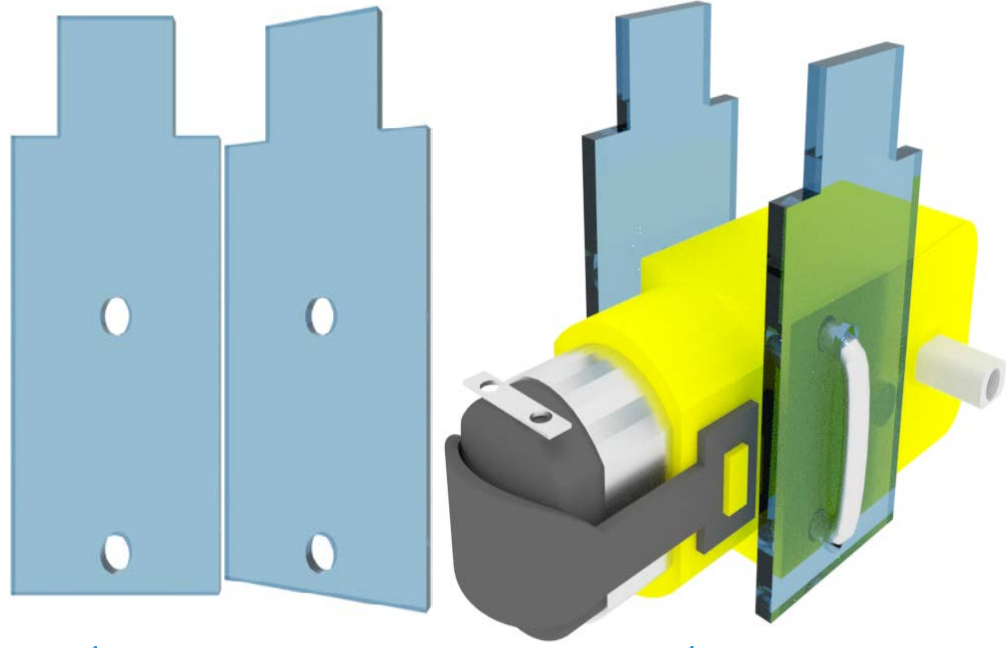
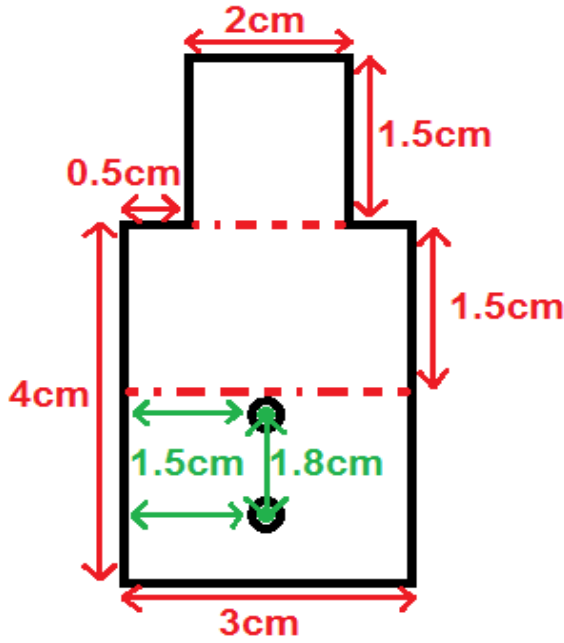
2. تثبيت لوحة التجارب (الشكل 03) في الهيكل:

- يستحسن أن تكون في مقدمة السيارة.
- يمكن الاستغناء عنها وتثبيت العناصر في الهيكل والتوصيل بالأسلاك.

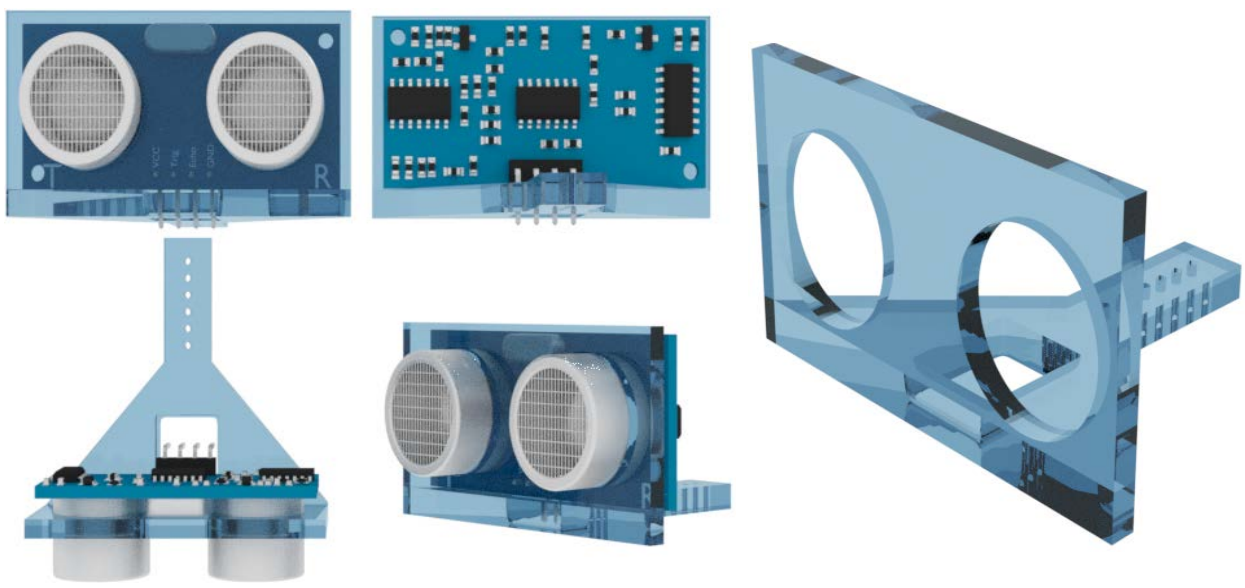
3. تثبيت العجلة المحورية (الشكل 05) في الهيكل:

- يجب أن تكون في خلف السيارة وفي المنتصف لتحافظ على التوازن.

- يجب تثبيتها جيدا بالبراغي.
- لا يهم أن تكون بطول العجلات الأخرى.
- 4. تثبيت العجلات بالمحرك (الشكل 04) في الهيكل:
- ستحتاج لقطعتين لوحيتين لكل عجلة من أجل تثبيتها في الهيكل.



- يجب أن تكون العجلات بنفس الارتفاع عن الأرض.
- 5. تثبيت لوحة الأردوينو (الشكل 01) في الهيكل:
- يجب تثبيت اللوحة جيدا في الهيكل.
- يجب تثبيت الكود البرمجي في اللوحة.
- 6. وضع لوحة التحكم بالمحركات (الشكل 09) فوق لوحة التجارب:
- لا يهم أن توضع فوق لوحة التجارب.
- يستحسن أن تكون تغذيته مستقلة أي من البطارية مباشرة.
- 7. وضع المحرك 180° (الشكل 11) فوق لوحة التجارب:
- يجب أن يكون موضعه في مقدمة السيارة.
- يجب أن يرفق بالحامل لخاص به.
- 8. تثبيت جهاز قياس المسافة (الشكل 10) في المحرك 180° :
- يجب أن يكون موجهها للأمام عندما يكون المحرك في 90° .
- ستحتاج لحامل لحمل العنصر وتثبيته في المحرك 180° (يمكن شراؤه أو إنشائه).



9. تثبيت المقاومة المتغيرة (الشكل 08) في لوحة التجارب:

○ يجب تثبيتها في 3 أعمدة مستقلة أي كل عمود غير متصل بالآخر.

10. تثبيت البطارية (الشكل 60) في الهيكل:

○ من المستحسن أن تكون البطارية 9 فولط فما فوق.

11. توصيل العناصر بالأسلاك (الشكل 02):

○ لكل عنصر طريقة ربط خاصة، وطريقة ربط كل عنصر موجودة في الجدول.

○ من المستحسن أن تعدد الألوان لكي يسهل التمييز بينها، مثل (الأسود للموجب، الأحمر للسلبي).

12. الشكل الافتراضي النهائي للسيارة (بدون التوصيل):

