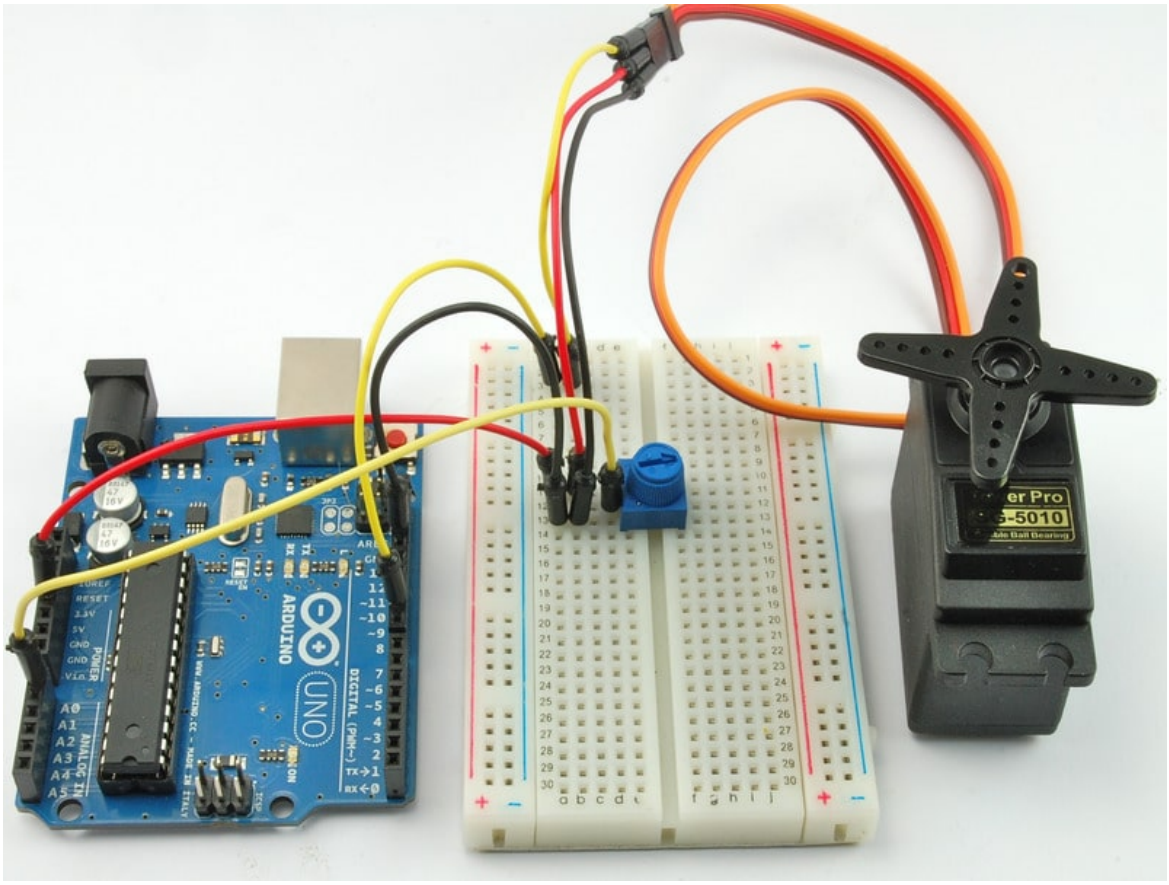
أردوينو – الدرس الخامس عشر – محرك السيرفو Servo Motor

مقدمة

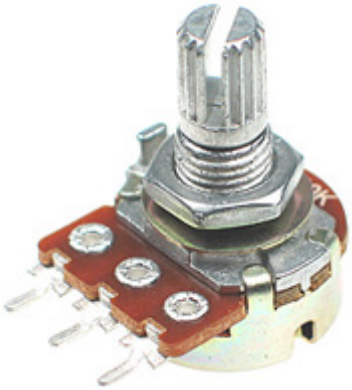
في هذا الدرس ستتعلم كيفية التحكم بدوران محرك السيرفو ‘servo motor’.

أولاً ، ستقوم بتعلم كيفية تحريك المحرك بمختلف الاتجاهات، ومن ثم ستقوم باضافة مقاوم متغير ‘pot’ للتحكم بوضعيته.

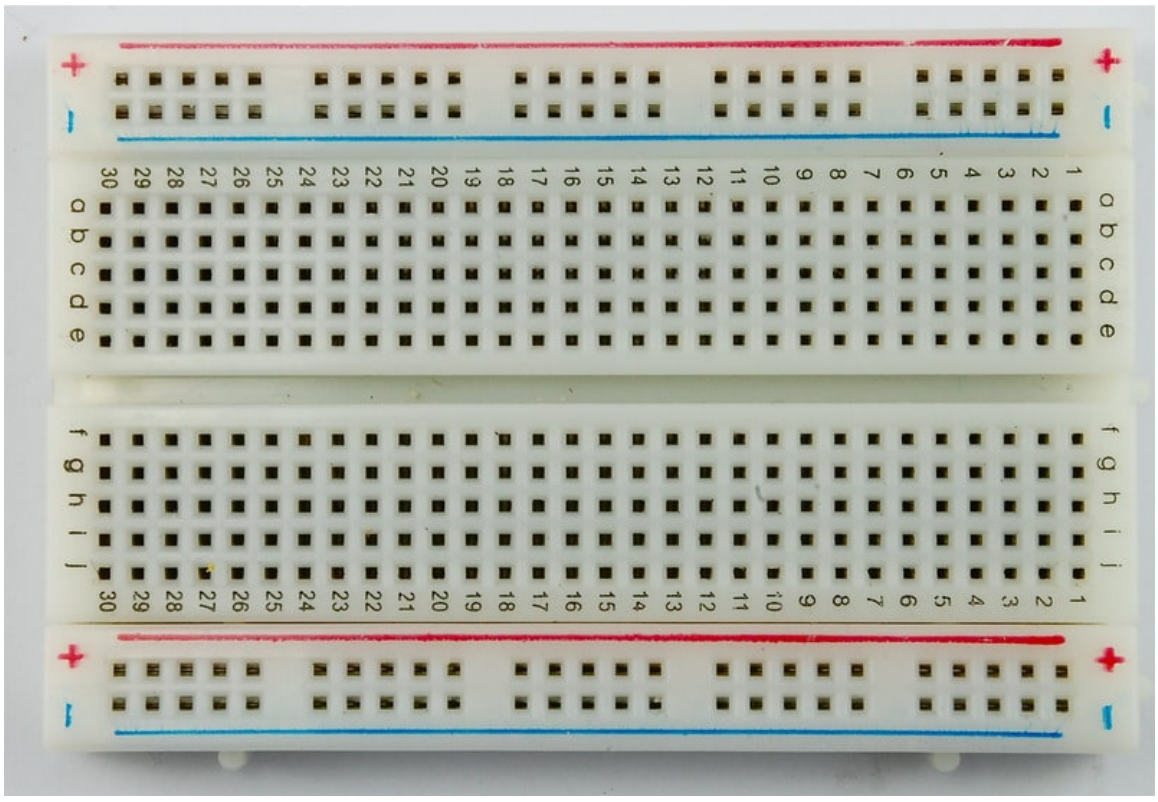


المواد والأدوات

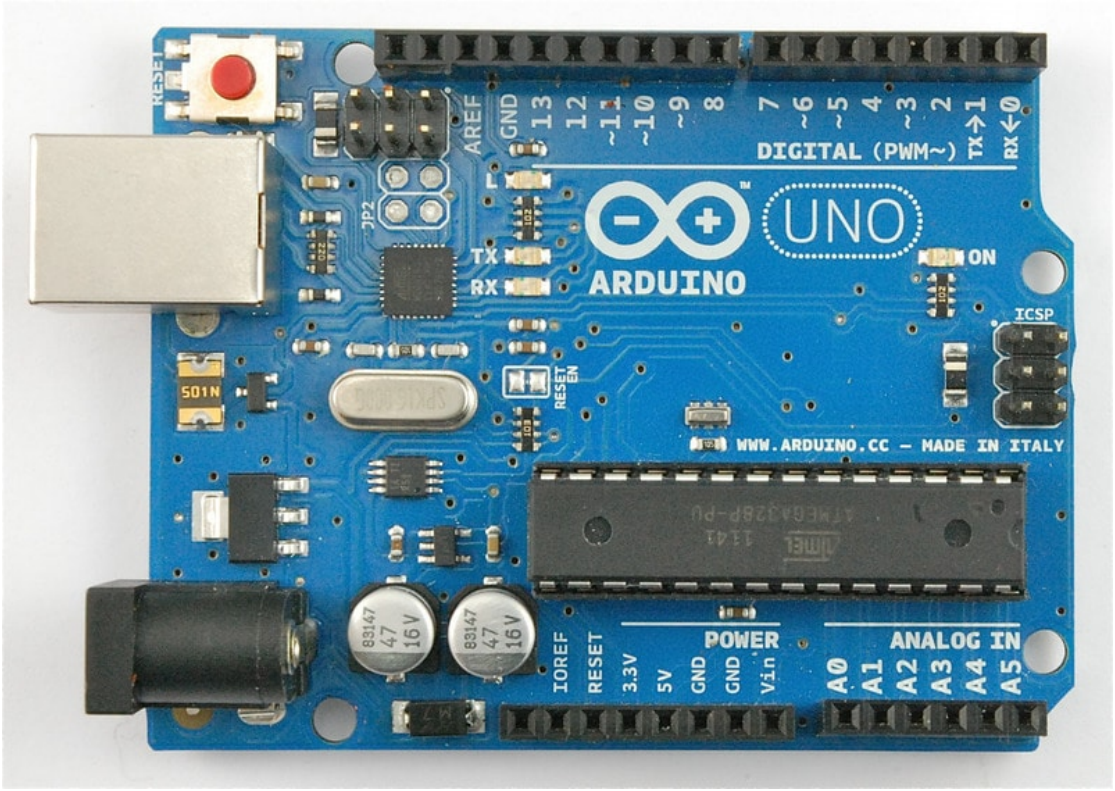
1× محرك سيرفو (TowerPro MG946R Servo)



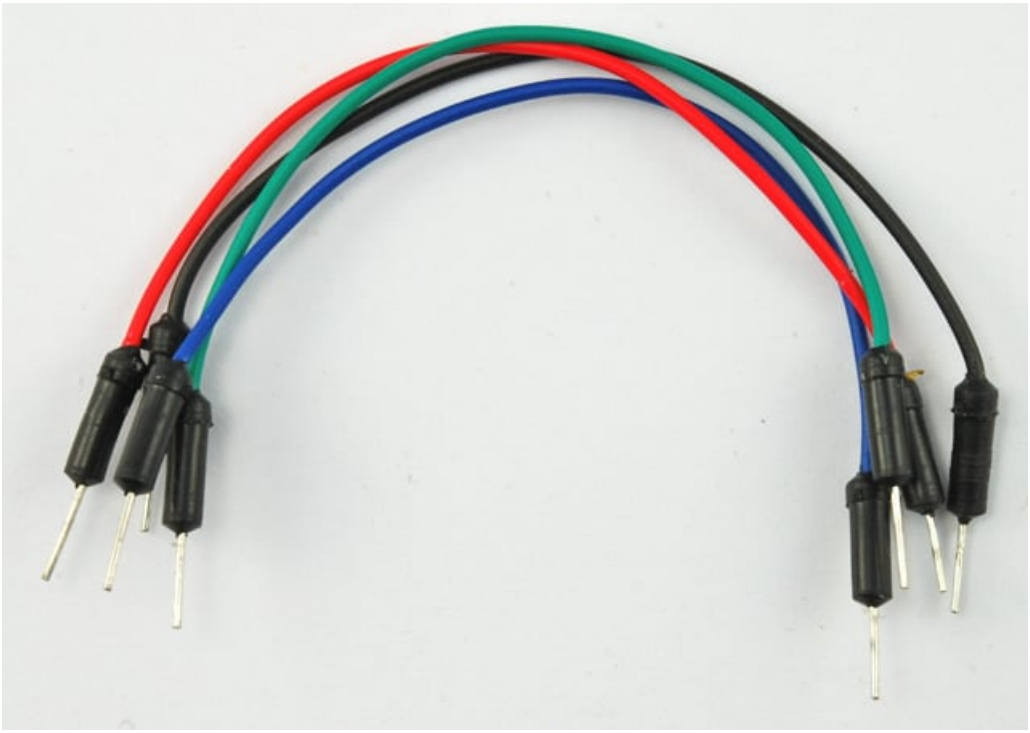
1× مقاوم متغير (pot) (10kΩ variable resistor)



1× لوحة التجارب (Half-size Breadboard)

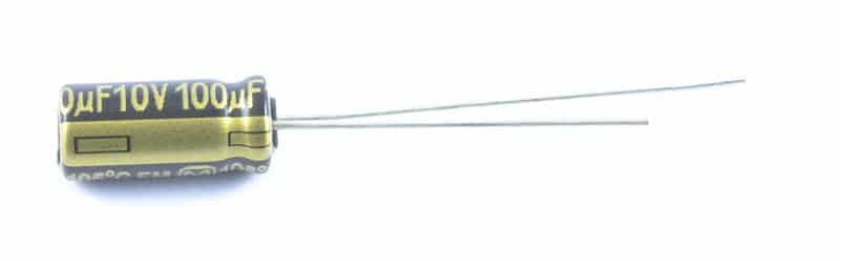


1× اردوينو اونو



حزمة أسلاك توصيل (ذكر–ذكر)

1× سلك اردوينو

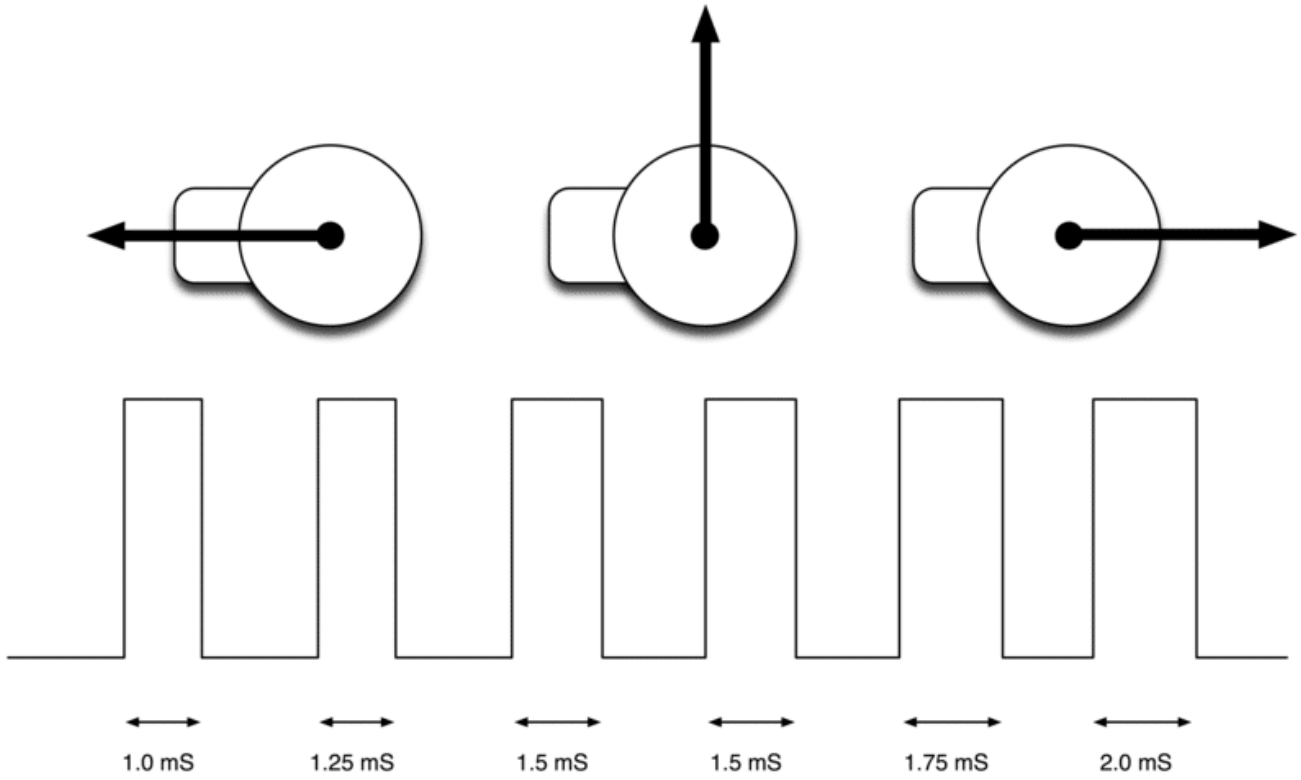


1× مكثف (100µF capacitor)

اختياري

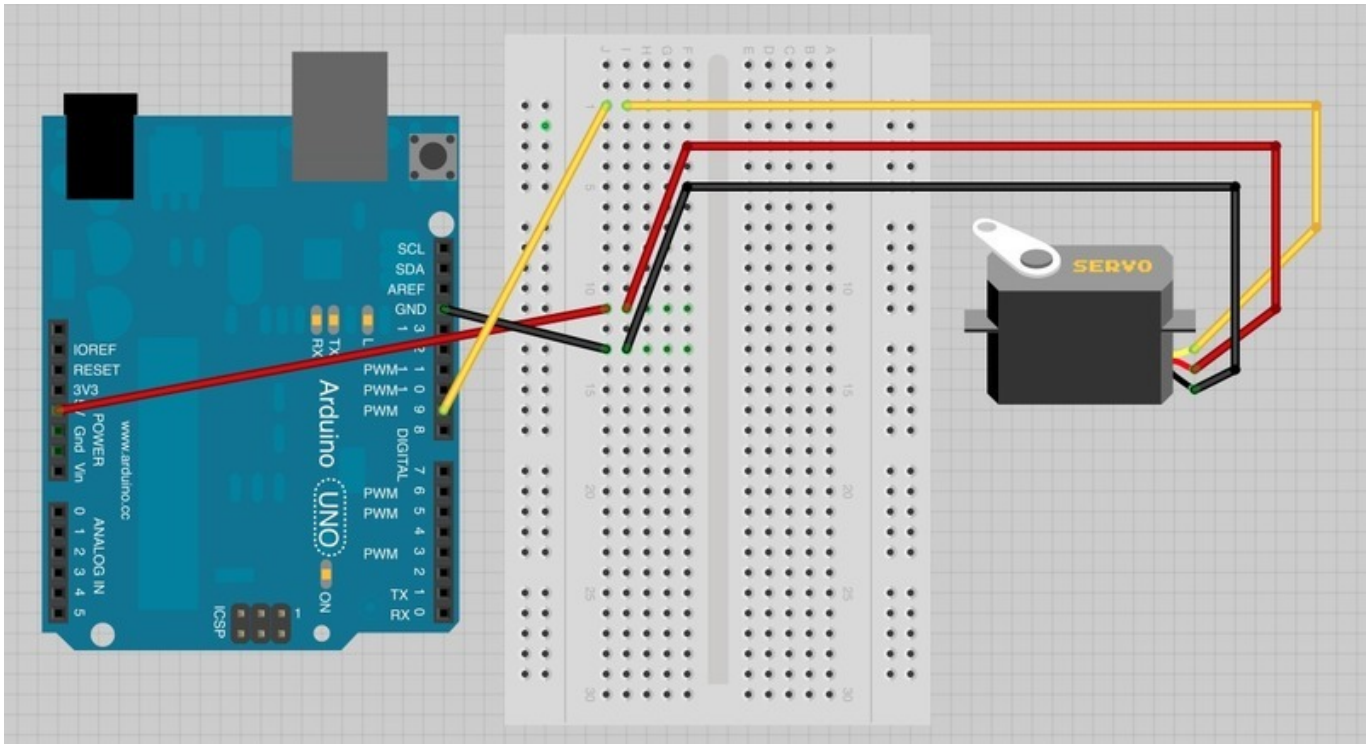
محرك السيرفو ‘servo motors’

موضع محرك السيرفو يعتمد على طول الذبذبة. حيث يستقبل ذبذبة كل مايقارب 20 جزء من الثانية . اذا كانت الذبذبة بجزء واحد من الثانية فإن زاوية السيرفو تساوي صفر، واذا كانت 1.5 جزء من الثانية فإنها ستكون بالمنتصف ، واذا كانت جزئين من الثانية فستكون بزاوية 180 درجة.



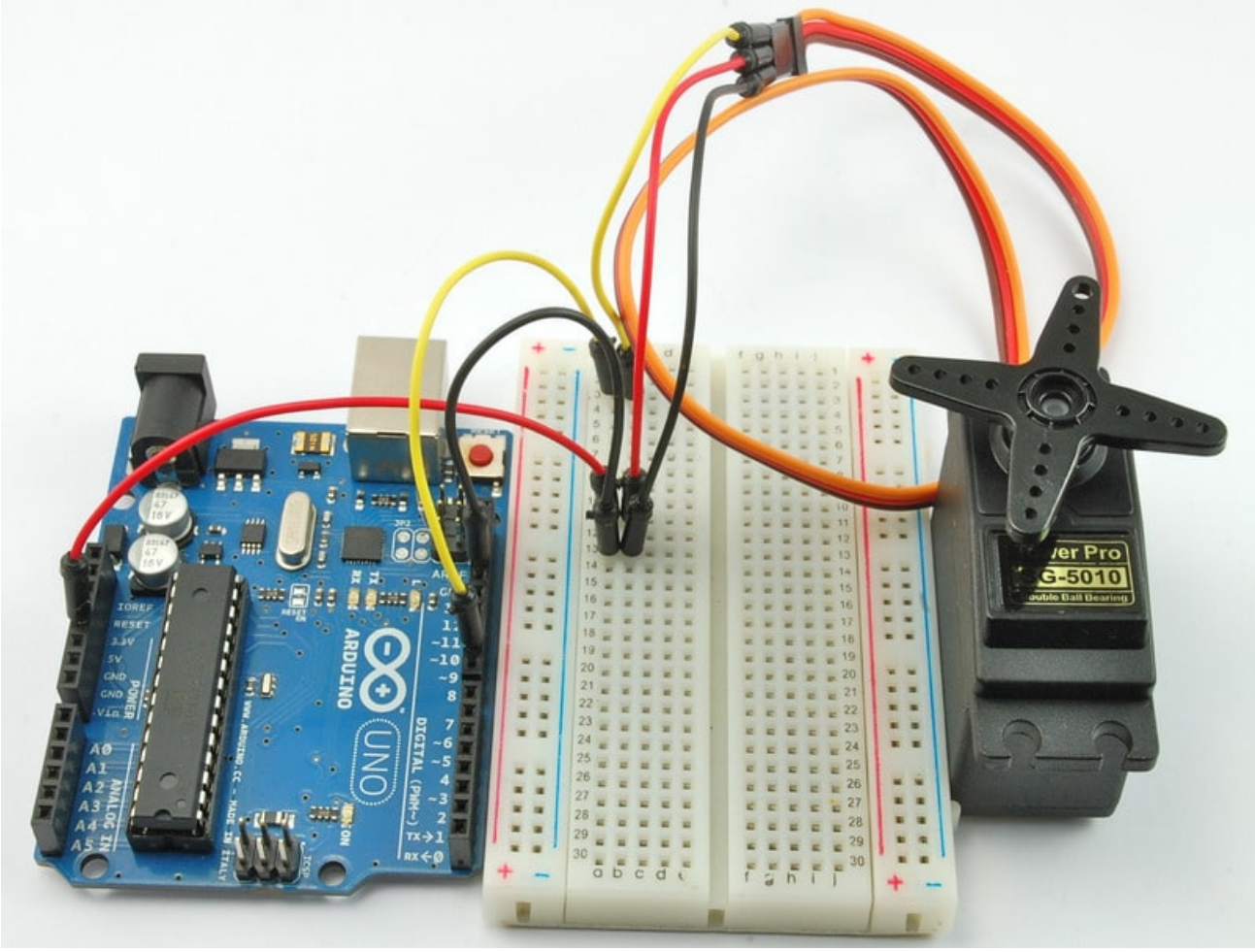
توصيل الدائرة محرك السيرفو

في هذه التجربة، اربط محرك السيرفو 'servo motor' بالاردوينو.



محرك السيرفو 'servo motor' لديه ثلاثة رؤوس ، اللون الأحمر لطاقة 5V ، اللون الأسود/أو البني للمخرج الأرضي GND ، واللون الأخير عادة مايكون برتقالي/أو أصفر يتم ربطه بالمنفذ الرقمي 9 'digital pin'.

يوجد مقبس بنهاة اسلاك محرك السيرفو 'servo motor' تسمح بوضع وربط اسلاك الـ jumper wires بها للربط مع لوح التجارب والأردوينو.



الكود البرمجي محرك السيرفو

قم برفع الكود التالي على متحكم الاردوينو وستجد أن محرك السيرفو 'servo motor' يبدأ بالحركة مباشرة باتجاه واحد ثم العوده للاتجاه الآخر.

الكود يعتمد على المثال البرمجي 'sweep' الموجود بمكتبة أمثلة الأردوينو
File> Examples> Servo> Sweep

```
#include <Servo.h>

int servoPin = 9;

Servo servo;

int angle = 0;   // servo position in degrees

void setup()
{
  servo.attach(servoPin);
}


void loop()
{
```

```
  // scan from 0 to 180 degrees
  for(angle = 0; angle < 180; angle++)
  {
    servo.write(angle);
    delay(15);
  }
  // now scan back from 180 to 0 degrees
  for(angle = 180; angle > 0; angle--)
  {
    servo.write(angle);
    delay(15);
  }
}
```

يتم التحكم بمحرك السيرفو 'servo motor' عبر سلسلة من الذبذبات ، ولجعل العملية سهله تم توفير مكتبة كود للاردوينو ليسهل عليك عملية توجيه محرك السيرفو 'servo motor' للتوجيه لزاوية معينة.

الاوامر البرمجية للتحكم بمحرك السيرفو تماماً كالاوامر الأخرى في برنامج الأردوينو ولكن لأننا لانستخدمها بشكل دائم في كل المشاريع ، لذلك عليك استدعاء مكتبة اوامر التحكم بمحرك السيرفو 'servo motor' وذلك عبر الأمر التالي

```
#include <Servo.h>
```

وكالعادة نقوم بتحديد وتعريف المنفذ الذي يتم ربطه بمحرك السيرفو عبر المتغير 'servoPin' .

```
Servo servo;
```

لنقم برمجياً بتعريف محرك سيرفو من نوع سيرفو . المكتبة البرمجية وفرت لنا نوع للتعريف وهو 'servo' تماماً كالتعريفات الاخرى مثل 'int' و 'float' ..الخ

لذلك باستخدام تعريف النوع 'servo' يمكنك من أن تعرف حتى 8 محركات سيرفو مرتبطه بالأردوينو . التعريف يتم كالتالي

```
Servo servo1;
Servo servo2;
```

في دالة setup علينا ربط متغير 'servo' بالمنفذ الذي سيتحكم بمحرك السيرفو ، وذلك عبر الأمر التالي

```
  servo.attach(servoPin);
```

المتغير 'angle' يستخدم لتحديد الزاوية الحالية لمحرك السيرفو . بداخل دالة loop استخدمنا حلقتين 'for loop' وذلك لزيادة الزاوية في اتجاه واحد حتى 180 درجة ثم العودة والذهاب للجهة الأخرى.

الأمر

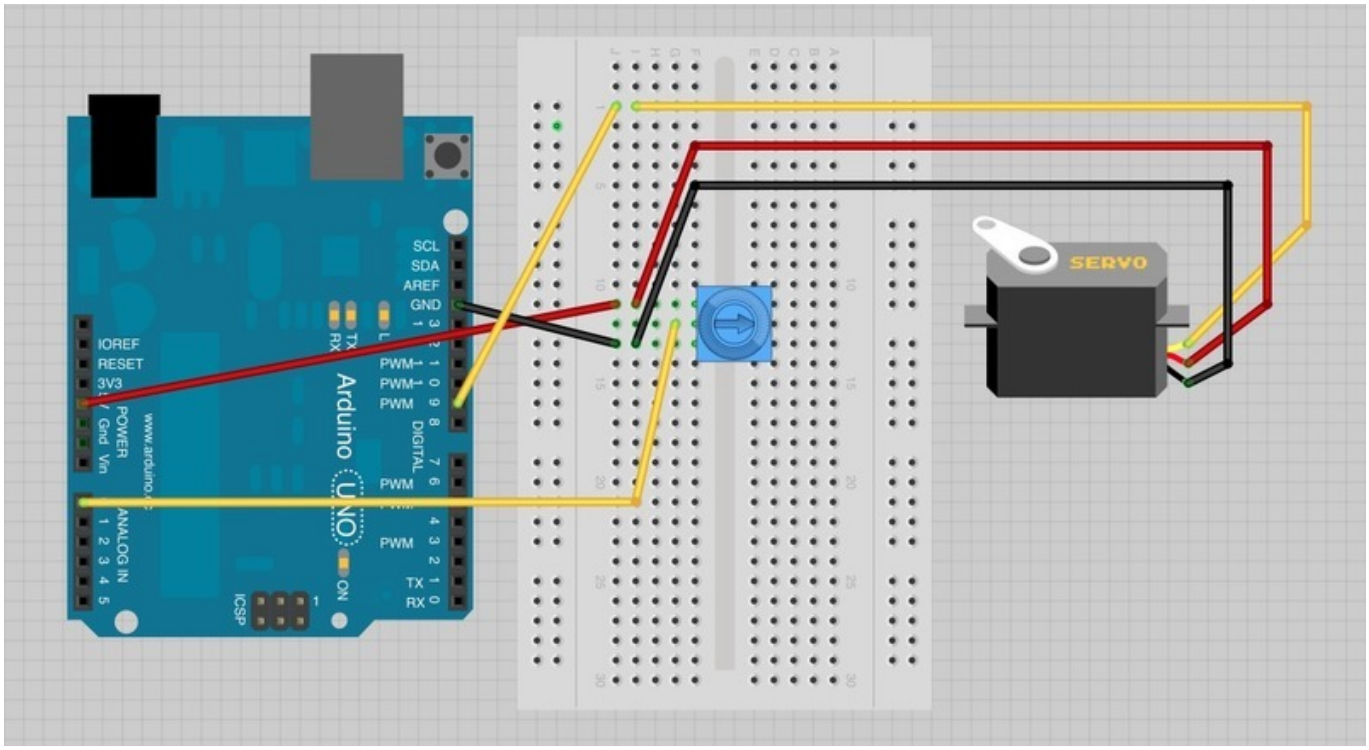
```
  servo.write(angle);
```

يطلب من السيرفو لتحديث موضعه من الزاوية التي تم تحديدها له.

توصيل الدائرة للموجه

الخطوة التالية هي وضع المقاوم المتغير واستخدامه كموجه يسمح لنا بتغيير زاوية محرك السيرفو عبر توجيهه.

كل ماعلينا فعله هو وضع المقاوم المتغير 'pot' وربطه بالمنفذ A0 بالأردوينو.



الكود البرمجي للموجه

الكود البرمجي لتوجيه محرك السيرفو عبر المقبض أسهل من الكود السابق

```
#include <Servo.h>

int potPin = 0;
int servoPin = 9;
Servo servo;

void setup()
{
  servo.attach(servoPin);
}

void loop()
{
  int reading = analogRead(potPin);     // 0 to 1023
  int angle = reading / 6;              // 0 to 180
```

```
  servo.write(angle);
}
```

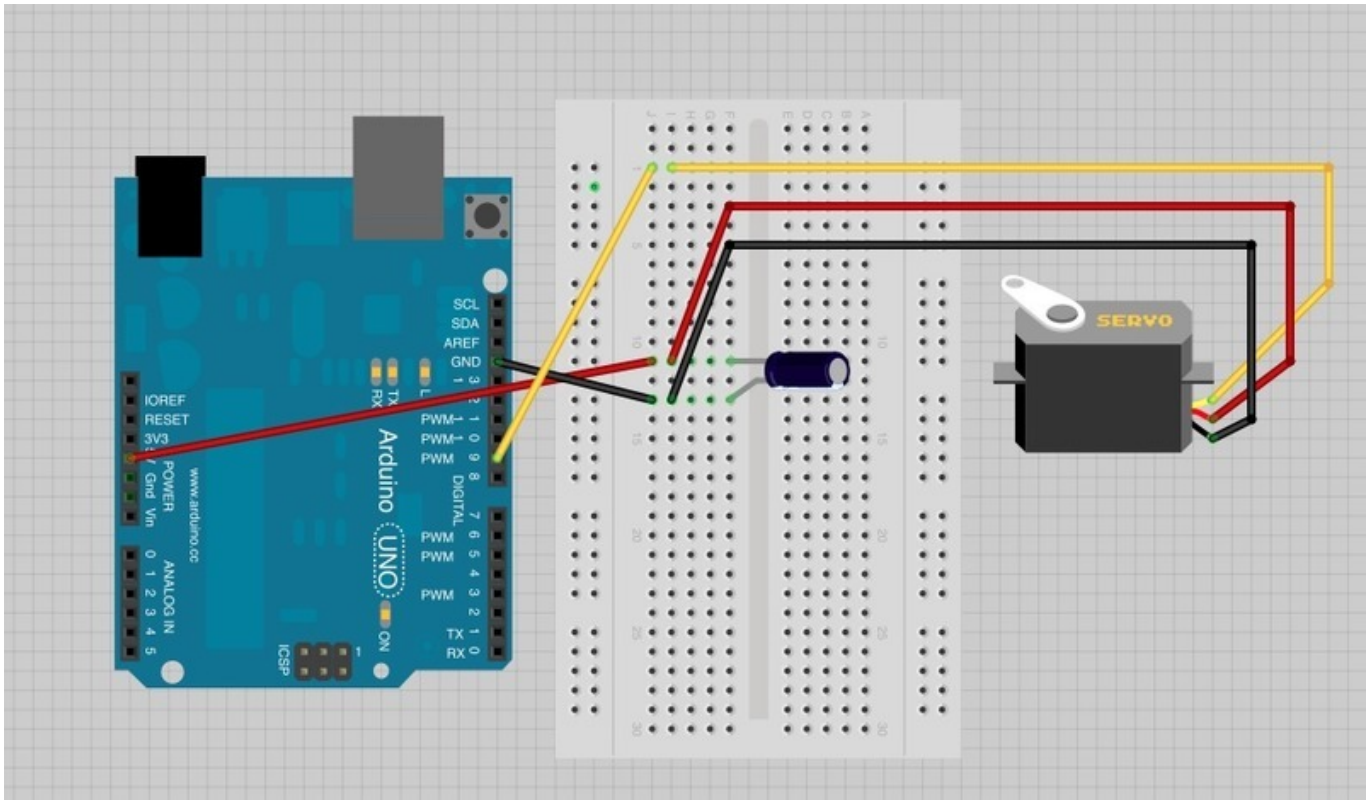
هنالك متغير آخر يدعى 'potPin' .

لتحديد موضع السيرفو ، علينا تحويل القراءة التناظرية 'analogRead' من المنفذ A0. هذا سيعطينا قيمة مابين 0 و 1023 . وبما أن السيرفو يستطيع التحرك حتى زاوية 180 درجة كحد أقصى لذا علينا خفض الحد الأقصى من القيمة المقروءة. وذلك عبر قسمة الرقم على 6 وذلك سيعطينا قيمة مابين 0 و 170 والتي ستكون مناسبة.

المشاكل والإصلاحات

قد يحدث خلل أثناء عمل السيرفو 'servo motor'. وقد يحصل هذا عندما يكون متحكم الأردوينو مرتبط "ببعض" منافذ الـUSB . هذا لأن المحرك يوجه الكثير من الطاقه، خصوصاً عند بدء عمل المحرك،مما قد يخفض الجهد Voltage بالأردوينو،فيتم اعادة تشغيله

يتم حل هذه المشكلة عبر اضافة مكثف عالي '470uF) 'capacitor أو أعلى) يتم وضعه مابين 5V و GND.



المكثف 'capacitor' يعمل كخزان للتيار الكهربائي حيث يسمح للمحرك عند بدء تشغيله بأخذ التيار المخزن بالمكثف اضافة للتيار الذي يقوم بأخذه من الأردوينو.

النهاية الأطول للمكثف 'capacitor' هي النهاية الموجبة والتي يجب ربطها بـ5V. النهاية السالبة عادة يكون علامة ' – ' عليها.

أنشطة أخرى

حاول تقليل معدل التأخير 'delay' من 15 جزء من الثانية إلى 5 جزء من الثانية .
لاحظ اختلاف سرعة السيرفو.

حاول التحكم بمحرك السيرفو عن طريق شاشة الاتصال التسلسلي عوضاً عن المقاوم المتغير.

ملاحظة: لتتمكن من قراءة الرقم من شاشة الاتصال التسلسلي serial monitor تستطيع استخدام دالة Serial.parseInt() لقراءة الأرقام من شاشة الاتصال التسلسلي serial monitor