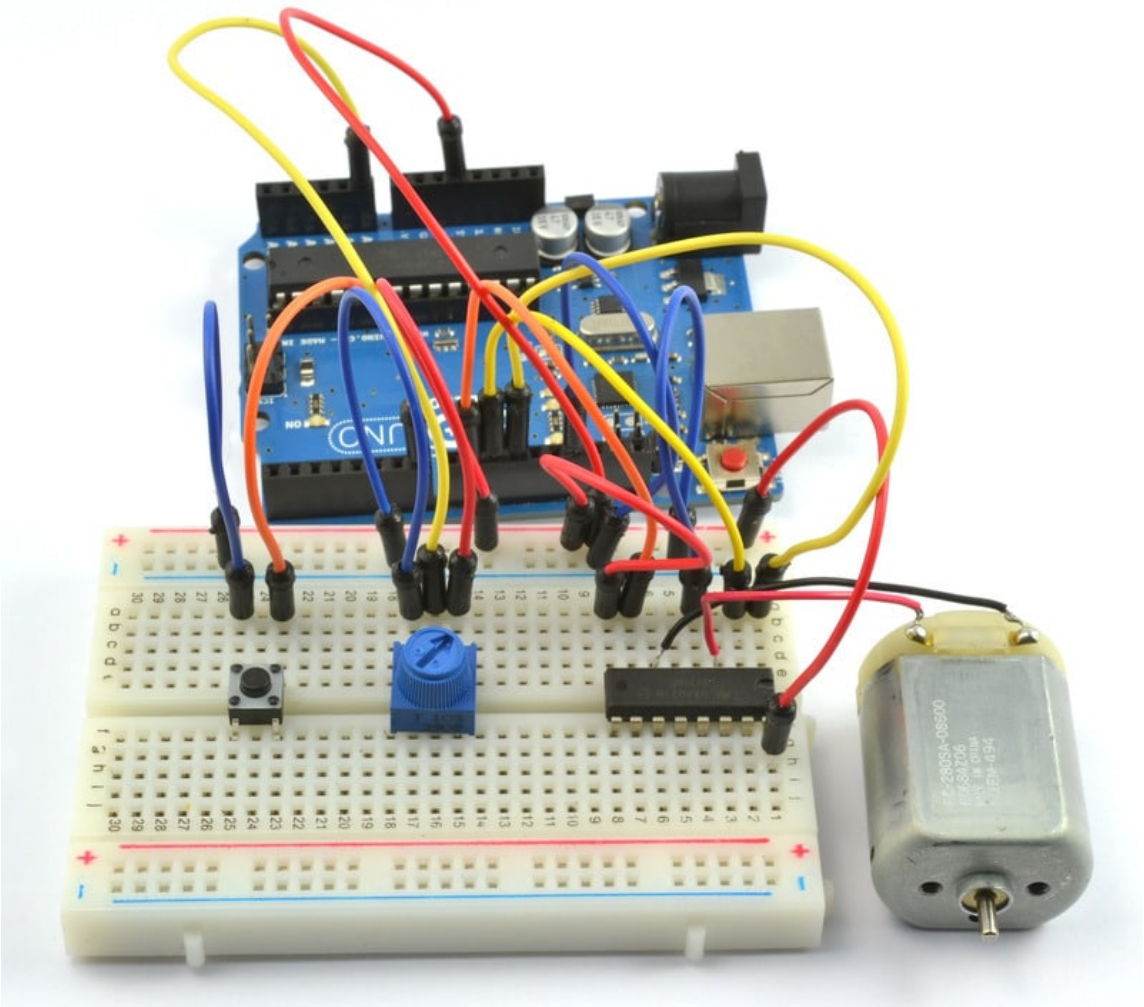
اردوينو – الدرس السادس عشر – التحكم باتجاه وسرعة دوران محرك DC Motor

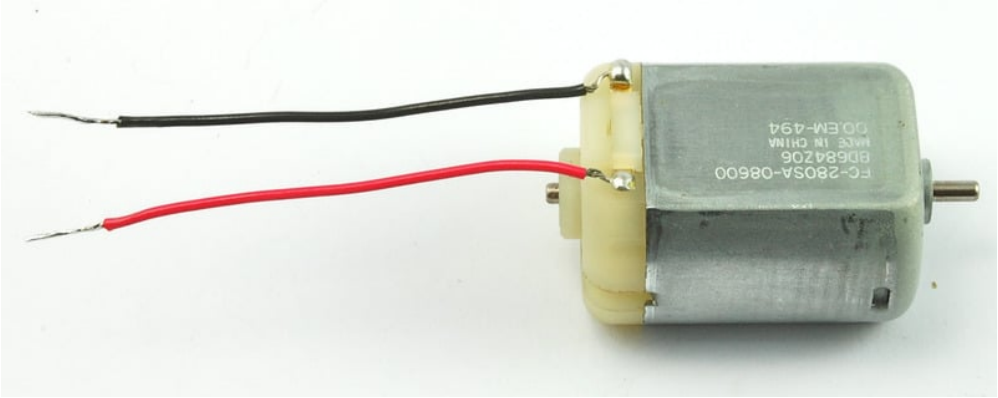
مقدمة

في هذا الدرس ستتعلم كيفية التحكم بسرعة واتجاه حركة محرك التيار المستمر الصغير DC Motor عن طريق الاردوينو ورقاقة التحكم بالمحرك L293D .

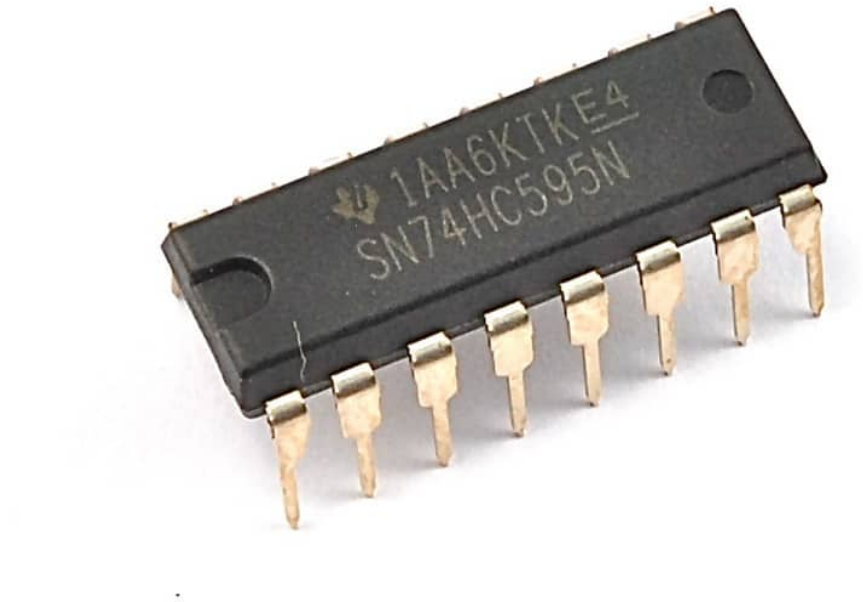
في هذه التجربة استخدمنا المقاوم المتغير 'pot' للتحكم بسرعة محرك DC Motor ،كما استخدمنا زر 'push button' للتحكم في اتجاه حركة محرك DC Motor.



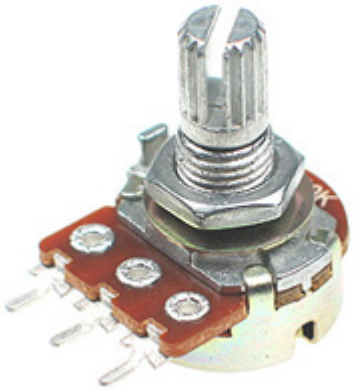
المواد والأدوات



1× محرك التيار المستمر الصغير (6V DC Motor)

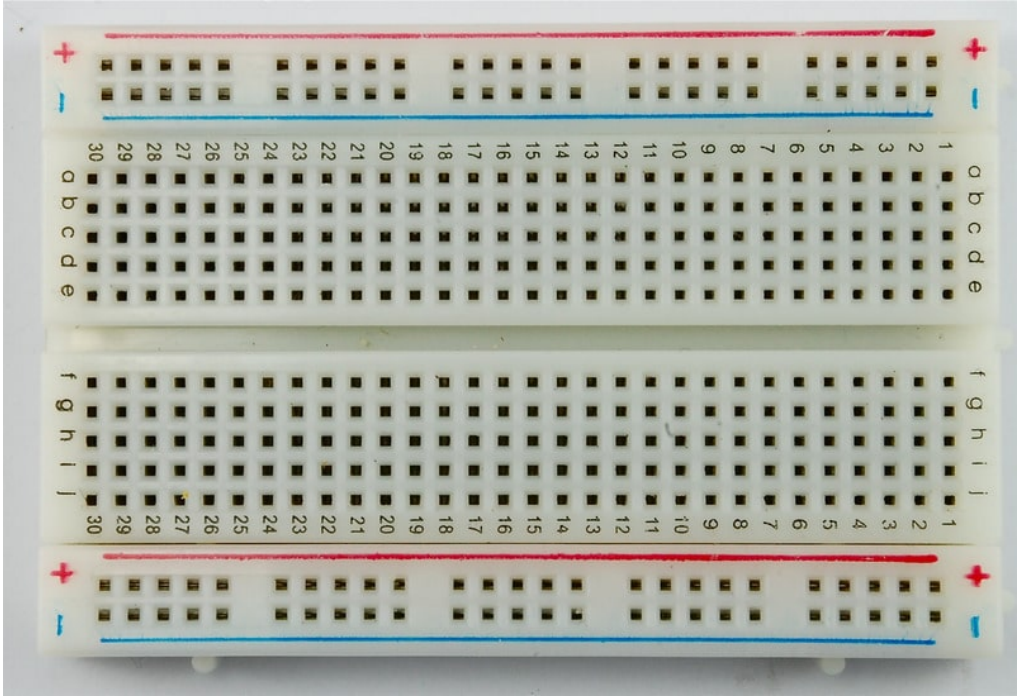


1× دائرة التحكم بالمحرك L293D

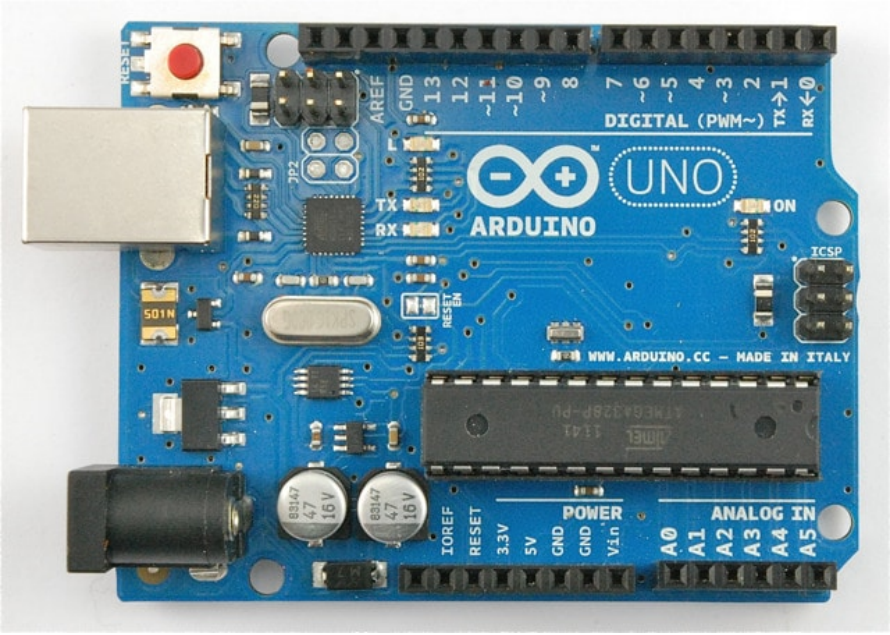


1× مقاوم متغير (10kΩ variable resistor (pot

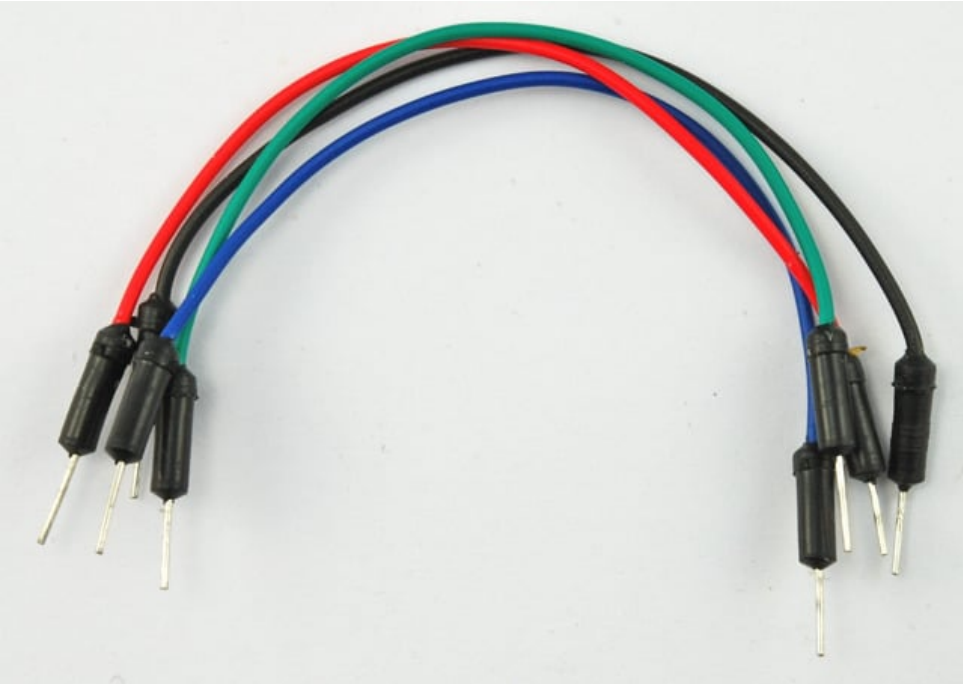
1× ازرار (Tactile push switch)



1× لوحة التجارب (Half-size Breadboard)



1× اردوينو اونو



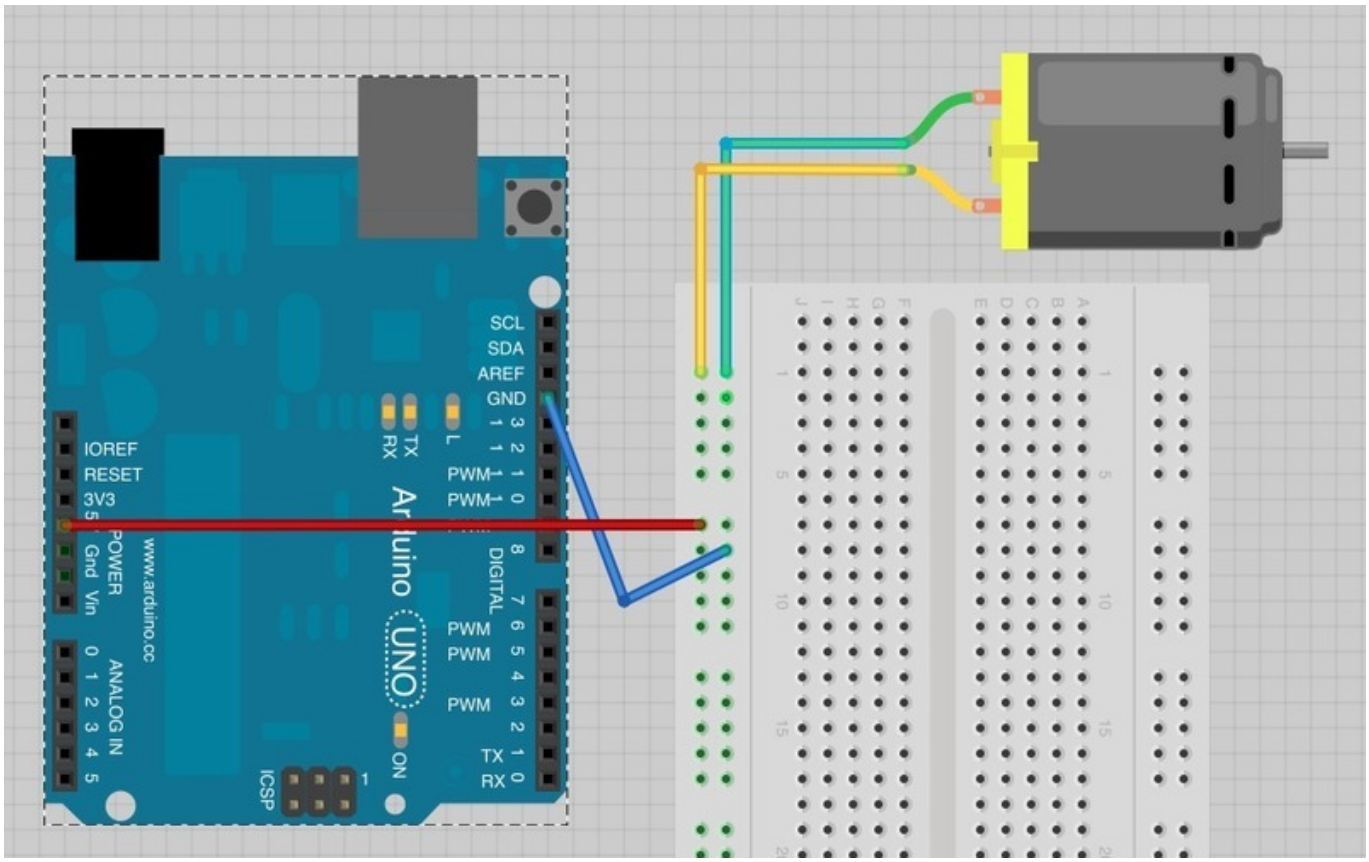
حزمة أسلاك توصيل (ذكر_ذكر)

×1 سلك اردوينو

توصيل الدائرة

قبل ان تحكم بالمحرك الصغير DC Motor، علينا اجراء التجربة مع رقاقة التحكم بالمحرك L293D للتعرف على طريقة عملها.

كل ماعلينا فعله هو توفير طاقة 5V من الاردوينو للمحرك الصغير DC Motor.

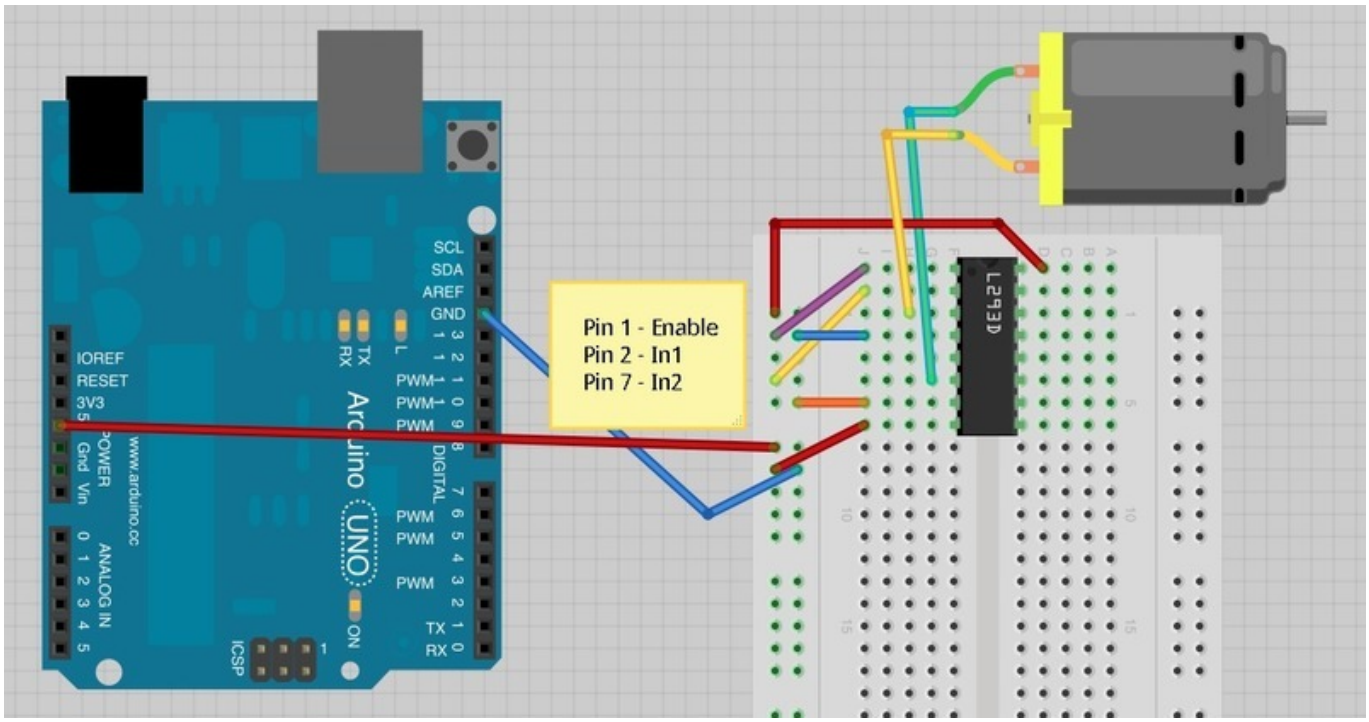


لاحظ اتجاه دوران محرك DC Motor، يمكنك لمس محرك DC Motor بأصابعك لمعرفة اتجها الحركه . يمكنك عكس اتجاه حركة محرك DC Motor عبر عكس توصيلات المحرك (قم بتبديل توصيلة الـ5V بـGND ، كذلك الحال مع التوصيله الأخرى) بذلك سترى انعكاس لاتجاه حركة الدوران.

وهذا هو دور رقاقة التحكم بالمحرك L293D، تقوم بالتحكم بالمنافذ التي توفر الطاقة والمجال الأرضي GND وبذلك تتحكم باتجاه حركة

الدوران.

قم ببناء لوح التجارب كالتالي، لاحظ ان دور متحكم الاردوينو هنا هو توفير الطاقه فقط.



المنافذ التي تهمنا بالنسبة لرقاقة التحكم بالمحرك L293D هي ثلاث منافذ ، منفذ 1 (Enable Pin 1)، منفذ 2 (In1 Pin 2) ، ومنفذ 7 (In2 Pin 7) .

وهذه تكن موصولة إما بـ5V أو GND باستخدام السلك البنفسجي ، أو الأصفر أو البرتقالي.

اذا قمت بربط (Enable Pin 1) للـGND سيتوقف المحرك، مهما فعلت بـpin In1 و pin In2 . يقوم Enable بالتشغيل أو الإطفاء. هذا يجعله مفيد لاستخدام 'PWM output' للتحكم في سرعة المحرك. قم بتوصيل Pin 1 إلى 5V لتشغيل المحرك مره أخرى.

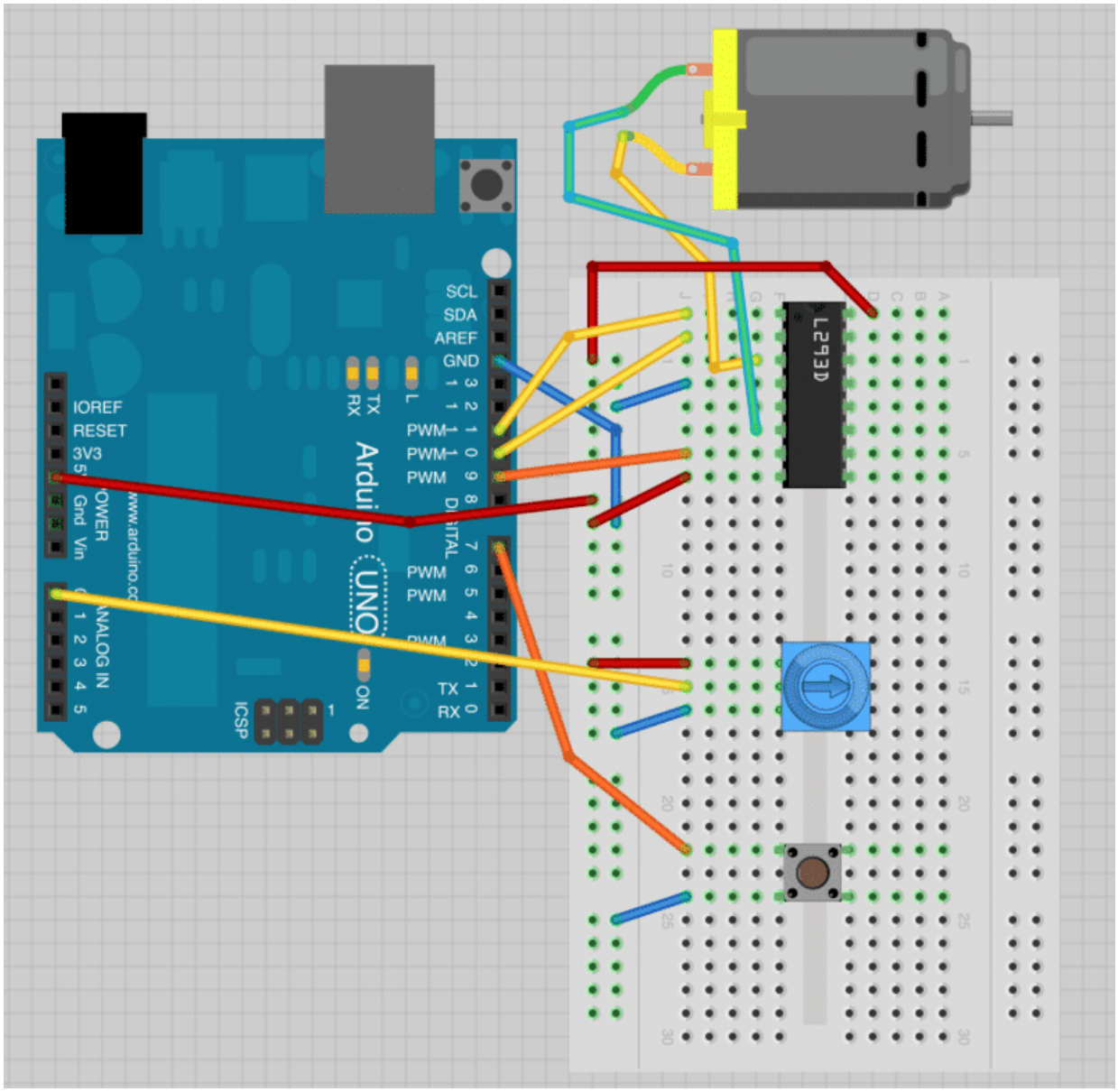
والان قم بتغيير (pin 2 In1,أصفر) من 5V إلى GND.
In1 و In2 جميعها مرتبطه بـGND ، وبالتالي سيتوقف المحرك مره أخرى.

تغيير In2 من GND إلى 5V سيعكس حركة دوران المحرك DC Motor (الاتجاه الآخر).

أخيراً ، بتغيير In1 وربطه مره أخرى بـ 5V بالتالي سيكون In1 و In2 مرتبطه بـ5V سيوقف محرك DC Motor .

والآن وبعد ان اعتدت على التحكم بمحرك DC Motor ورحة دورانه ، يمكننا الآن جعل متحكم الاردوينو يتحكم بمنافذ Enable, In1, In2

عند بناء لوح التجارب عليك التأكد من موضع واتجاه الرقاقة كما هي في التصميم (الطرف الذي حيتوي على نتوء يجب أن يكون باتجاه الأعلى)



الكود البرمجي

ارفع الكود التالي على متحكم الاردوينو:

```
int enablePin = 11;
int in1Pin = 10;
int in2Pin = 9;
int switchPin = 7;
int potPin = 0;

void setup()
{
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(switchPin, INPUT_PULLUP);
}
```

```
void loop()
{
  int speed = analogRead(potPin) / 4;
  boolean reverse = digitalRead(switchPin);
  setMotor(speed, reverse);
}

void setMotor(int speed, boolean reverse)
{
  analogWrite(enablePin, speed);
  digitalWrite(in1Pin, ! reverse);
  digitalWrite(in2Pin, reverse);
}
```

داخل دالة setup تم تعريف المنافذ وحالاتها .

داخل دالة loop ، يتم تحديد قيمة السرعة للمحرك DC Motor DC Motor عبر اخذ قراءة القيمة التناظرية 'analogRead' من المقاوم المتغير 'pot' وقسمته على 4.

السبب وراء القسمة على 4 هي لأن القيمة المستخرجة من القراءة التناظرية 'analogRead' تكون مابين 0 و 1023 ولكنها يجب أن تكون القيمة بي 0 و 255.

اذا تم الضغط على الزر ، فإن محرك DC Motor سيتحرك للأمام ، وبالضغط مره أخرى سيتحرك بعكس الاتجاه.
القيمة لمتغير 'reverse' يتم أخذه من متغير 'switchPin' ،لذا عند الضغط على الزر ستكون القيمة 0،وعند الضغط مره أخى ستكون القيمة 1 .

قيمة السرعة و الانعكاس 'reverse' يتم تمريرها إلى الدالة 'setMotor' والتي ستحدد المنافذ لرقاقة التحكم بمحرك للتحكم .

```
void setMotor(int speed, boolean reverse)
{
  analogWrite(enablePin, speed);
  digitalWrite(in1Pin, ! reverse);
  digitalWrite(in2Pin, reverse);
}
```

أولاً ، السرعة تم تحديدها عبر استخدام 'analogWrite' لمنفذ enable pin . منفذ enable pin يقوم بتشغيل أو اطفاء المحرك بغض النظر عن قيم منافذ in1 و in2.

للتحكم باتجاه حركة الدوران للمحرك DC Motor علينا عكس قيم منافذ in1 و in2.

اذا كانت قيمة in1 عالية (اي تساوي 1) و قيمة in2 منخفضة (اي تساوي 0)، سيقوم محرك DC Motor بالدوران باتجاه واحد ، واذا تم عكس القيم لمنافذ in1 و in2 فسيقوم محرك DC Motor بالدوران بالاتجاه المعاكس.

الأمر ' ! ' يعني ' ليس – not' لذلك أول امر 'digitalWrite' لـin1 يقوم بعكس القيمة المأخوذه من 'reverse' فمثلاً لو كانت القيمة عالية (تساوي 1) فسيوقم بعكسها إلى قيمة منخفضة (تساوي 0) والعكس صحيح.

الأمر الثاني 'digitalWrite' لـin2 يقوم بأخذ القيمة من 'reverse' وهذا يعني بأنها ستكون دائما عكس القيمة لـin1 .

أنشطة أخرى

تستطيع تغيير الكود البرمجي بحيث يمكنك التحكم بمحرك DC Motor دون الحاجة للمقاوم المتغير و الزر . بحيث تبدأ حركة المحرك بشكل بطيء باتجاه واحد ، ثم تزداد السرعة بشكل تدريجي ، ثم تبطئ مره أخرى ويتم عكس الاتجاه وهكذا..