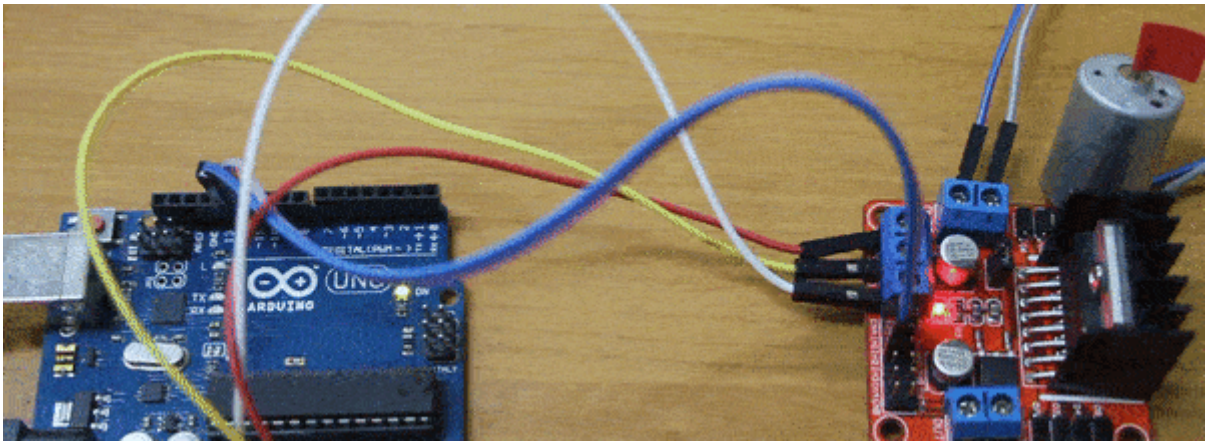
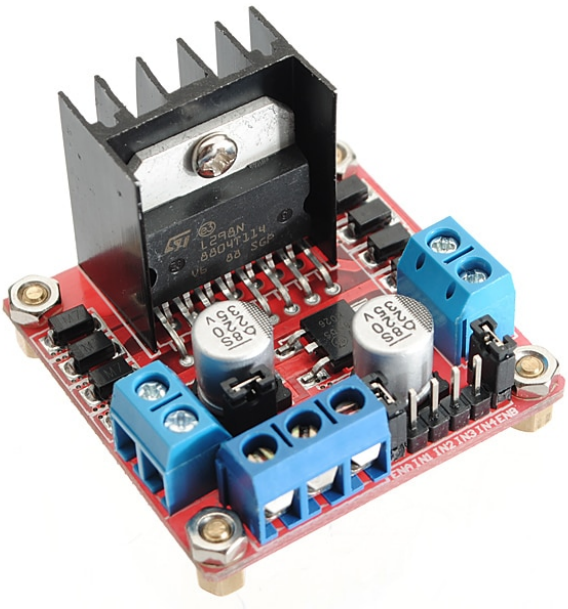
التحكم في محرك التيار المستمر بإستخدام L298

في هذا المشروع، سنتعلم فكرة عمل الدارة المتكاملة L298. ونتعرف على كيفية استعمالها للتحكم في تشغيل وإيقاف محرك التيار المستمر، وعكس اتجاه حركته. تستخدم هذه في مشاريع الروبوت مثل، متتبع المسار.



المكونات المطلوبة

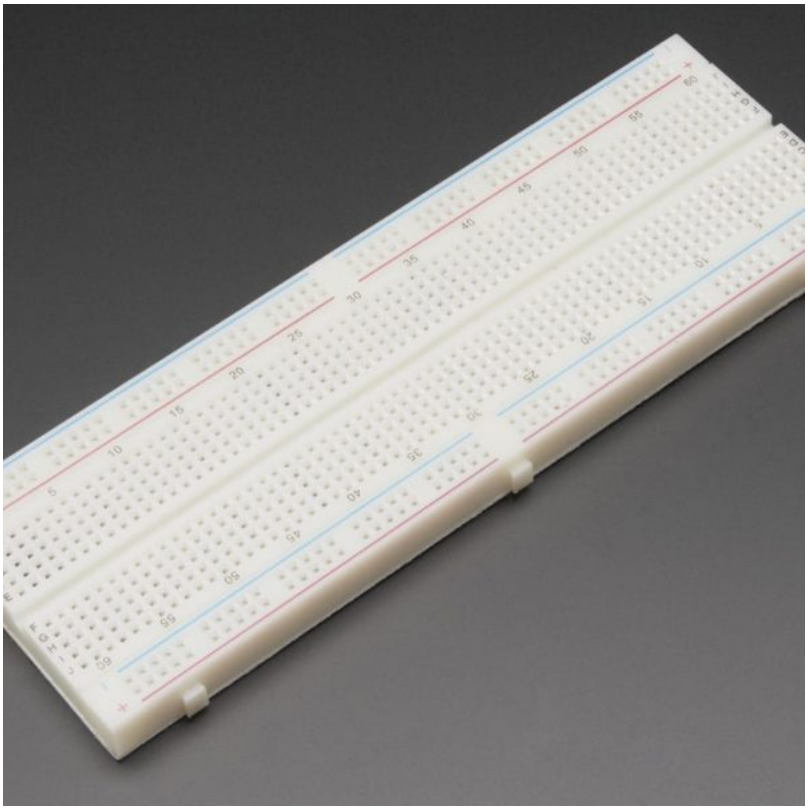
Arduino Uno



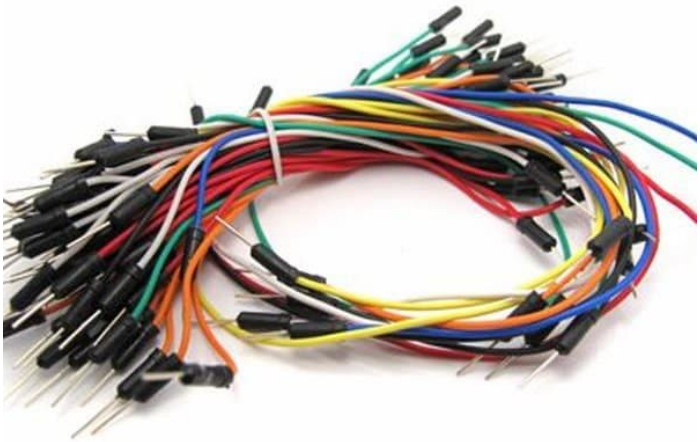
L298 Motor Driver

DC Motor

DC Power Supply



Breadboard



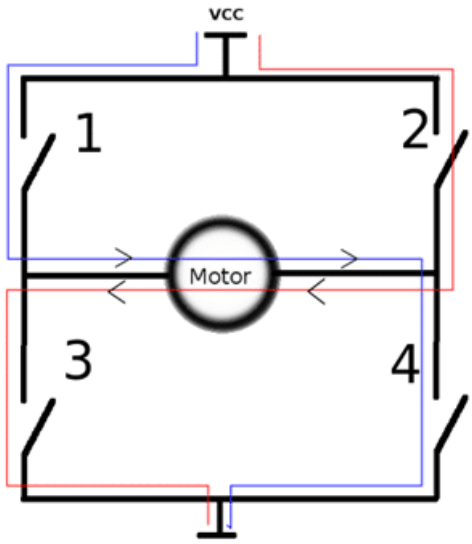
Wires

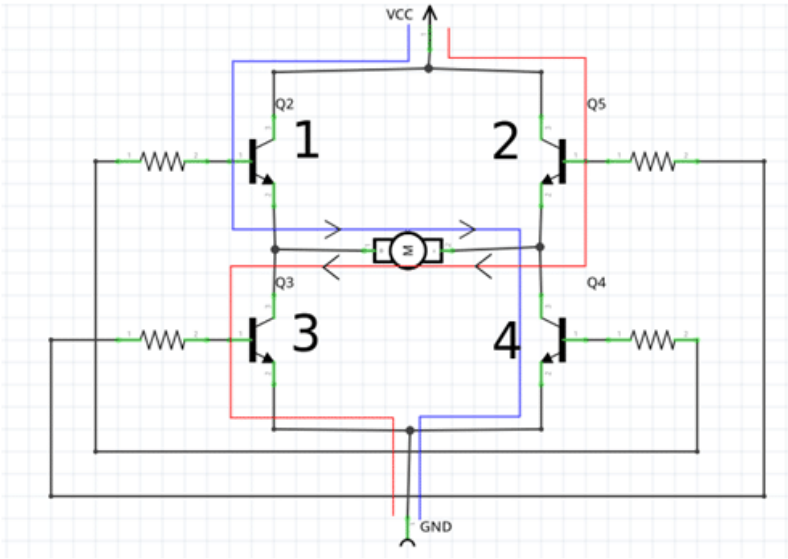
H-Bridge :

هو عبارة عن أربعة ترانزستور موصلين معا بشكل معين لتمكين المحرك من الدوران في إتجاهين مختلفين.

فكرة عمله :

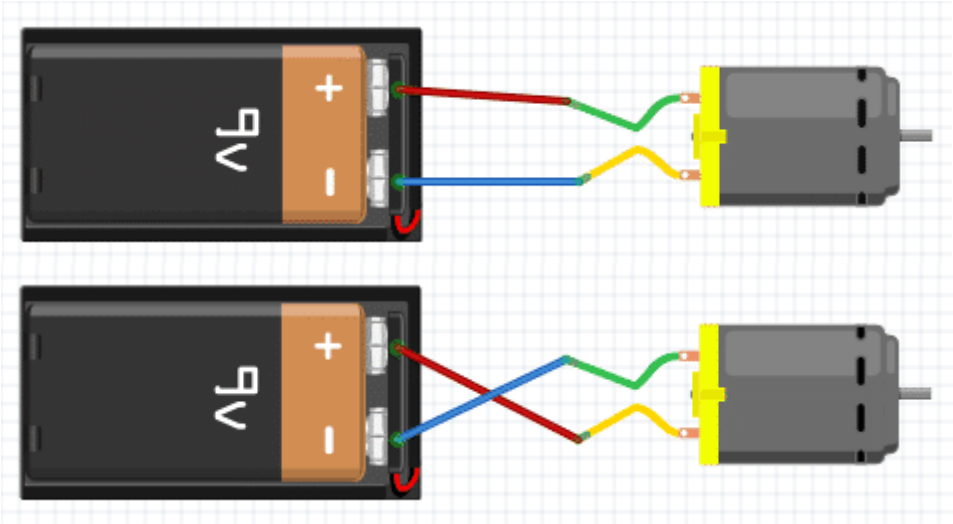
الصورة التالية توضح كيف يتركب الـ H-Bridge:





عند تشغيل كلا الترانزستور 1 و 4 يعمل المحرك نحو الإتجاه الأول وعند تشغيل الترانزستور 2 و 3 يعمل المحرك في الإتجاه المعاكس للإتجاه السابق.

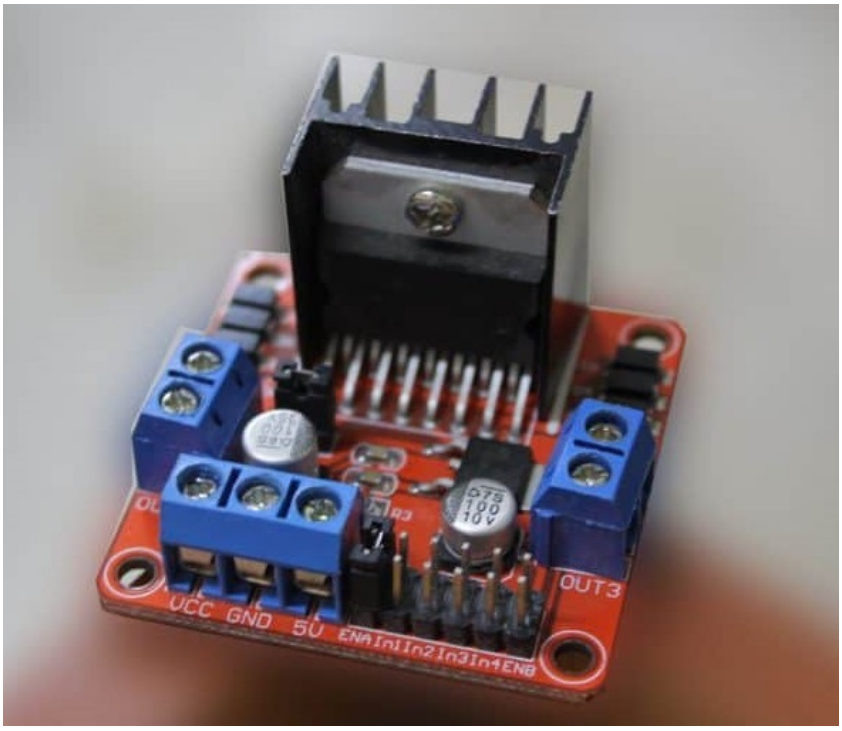
وبذلك، يتم عكس الدوران اتوماتيكيا دون الحاجة إلى تغيير التوصيل يدويا. على عكس محرك التيار المستمر، الذي يتم تعكس اتجاهه عن طريق عكس اطراف المحرك مع مصدر التيار الكهربائي.



في الحالة الأولى، سيدور المحرك مع إتجاه عقارب الساعة. وبعكس توصيل الأطراف (كما هو موضح بالحالة الثانية) سينعكس إتجاه الدوران ليصبح عكس عقارب الساعة.

L298 Motor Driver

هو عبارة عن H-Bridge تم تجميعها ووضعها معا في IC ، ليتم استخدامها بسهولة. يمكن إستخدام واحد من IC للتحكم في محركين في نفس الوقت. لكل محرك مخرج خاص به وأطراف تحكم خاصة.



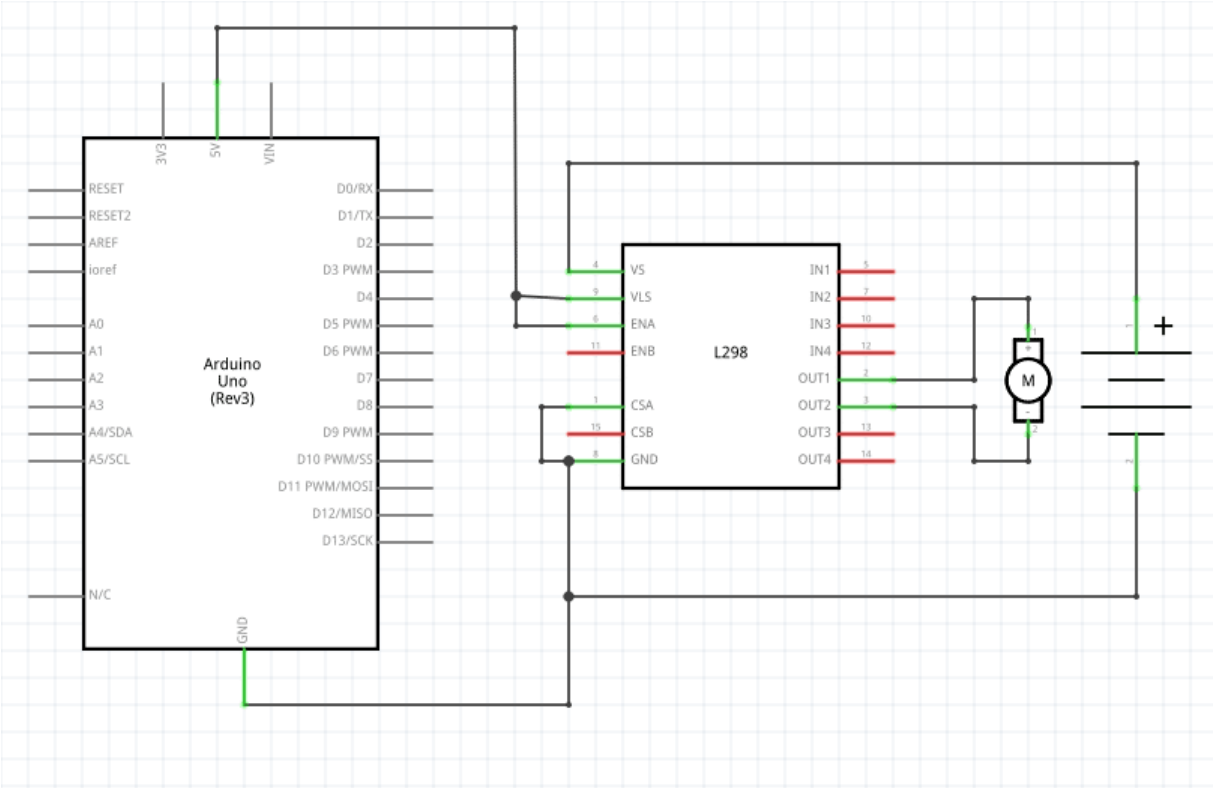
الاطراف المستخدمة

Output A	طرفى المحرك
VCC	طرف البطارية الموجب
5v	5v فى الاردوينو
GND	gnd فى الاردوينو
IN1	12 فى الاردوينو
IN2	13 فى الاردوينو

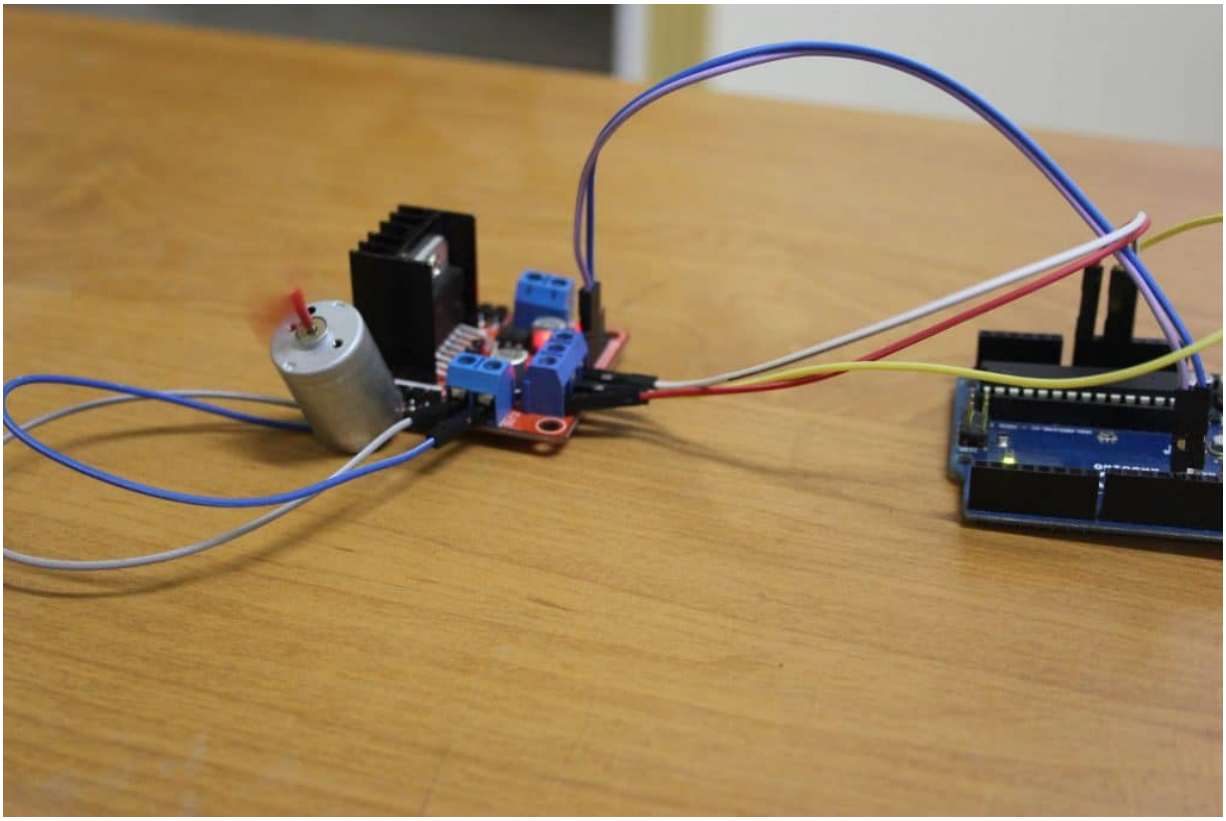
شرح الدارة

لا يمكن توصيل محرك التيار المستمر مباشرة مع الأردوينو. وذلك لأن المحرك يحتاج إلى تيار عالي لا يستطيع الأردوينو إعطاءه له. لذلك، سنقوم بإستخدام الـ L298 كدائرة بين الاردوينو الذي يعمل مع تيار صغيرة وبين المحرك الذي يحتاج إلى تيار عالي.

قم بتوصيل الدارة كما هو موضح بالصورة :



ولنتمكن من تشغيل المحرك في إتجاهين، يتم تشغيل الطرف الأول من الـ L298 للأردوينو في المحرك، فيدور المحرك في الإتجاه الأول. وعند تشغيل الطرف الثاني سيدور بالإتجاه المعاكس.



الكود البرمجي :

في هذا المشروع، سيدور المحرك في الإتجاه الأول لمدة ثلاث ثوان، ثم يتوقف لمدة ثلاث ثوان. ثم يدور المحرك بالإتجاه المعاكس لمدرة ثلاث ثوان، ثم يتوقف لمدة ثلاث ثوان أخرى. وهكذا حتى يتم فصل التيار الكهربائي.

قم بتحميل الكود التالي إلى الأردوينو :

```
#define MOTOR_IN1 12
#define MOTOR_IN2 13

void motor_forward(void);  // a function that will be called to rotate it clockwise
void motor_reverse(void);  // a function that will be called to totate it counter-
clockwise
void motor_stop(void);     // a function that will be called to stop the rotation

void setup() {
  pinMode(MOTOR_IN1, OUTPUT);  // set the first pin of the relay as output
  pinMode(MOTOR_IN2, OUTPUT);  // set the 2nd pin of the relay as output
}

void loop() {
  motor_forward();             // move forward/clockwise
  delay(3000);                 // keep rotating cw for 3 seconds
  motor_stop();                // stop rotating
  delay(3000);                 // stand still for 3 seconds
  motor_reverse();             // reverse the rotation direction/ccw
  delay(3000);                 // keep rotating ccw for 3 seconds
  motor_stop();                // stop rotating
  delay(3000);                 // stand still for 3 seconds
}

void motor_forward(void)       // the function that will cause the motor to rotate
cw
{
  digitalWrite(MOTOR_IN1, HIGH);
  digitalWrite(MOTOR_IN2, LOW);
}

void motor_reverse(void)       // the function that will cause the motor to rotate
ccw
{
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, HIGH);
}

void motor_stop(void)          // the function that will cause the motor to stop
rotating
{
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, LOW);
}
```

شرح الكود :

قمنا سابقا بتوصيل طرفى كلا من الترانزستور (IN1,IN2) بمنفذ 12 و 13 للأردوينو . لذلك قمنا بتسمية كلا المنفذين للأردونو تبعا لما تم توصيله بالدارة.

نقوم بتعرف المتغيرات IN1 و IN2 (أطراف الـ H-bridge الموصله بالاردوينو) كمخرج.

```
void setup() {
  pinMode(MOTOR_IN1, OUTPUT);  // set the first pin of the relay as output
  pinMode(MOTOR_IN2, OUTPUT);  // set the 2nd pin of the relay as output
}
```

في دالة ()loop، نقوم أولا بإستدعاء الدالة ()motor_forward . تقوم هذه الدالة بتشغيل المحرك مع اتجاه عقارب الساعة لمدة 3 ثوان ((delay(3000) . ثم نقوم باستخدام الدالة ()motor_stop ، لإيقاف المحرك عن العمل لمدة 3 ثوان. ثم يتم عكس اتجاه حركة المحرك باستخدام الدالة ()motor_reverse لمدة 3 ثوان. ومن ثم يعود ليكرر نفس هذه المهمة من البداية مرة أخرى.

```
void loop() {
  motor_forward();             // move forward/clockwise
  delay(3000);                 // keep rotating cw for 3 seconds
  motor_stop();                // stop rotating
  delay(3000);                 // stand still for 3 seconds
  motor_reverse();             // reverse the rotation direction/ccw
  delay(3000);                 // keep rotating ccw for 3 seconds
  motor_stop();                // stop rotating
  delay(3000);                 // stand still for 3 seconds
}
```

الدلة ()motor_forward، تقوم بتحريك المحرك بإتجاه عقارب الساعة. تتم هذه العملية عن طريق جعل قيمة IN1 للمرحل HIGH والطرف الآخر LOW .

```
void motor_forward(void)       // the function that will cause the motor to rotate
cw
{
  digitalWrite(MOTOR_IN1, HIGH);
  digitalWrite(MOTOR_IN2, LOW);
}
```

تعمل هذه الدالة ()motor_reverse بشكل مشابه للدالة السابقة، إلا أنها تعكس اتجاه دوران المحرك. تتم هذه العملية عن طريق جعل قيمة IN2 للمرحل HIGH، وIN1 قيمة LOW .

```
void motor_reverse(void)       // the function that will cause the motor to rotate
ccw
{
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, HIGH);
}
```

دالة ()motor_stop، تقوم بإيقاف المحرك تماما عن الحركة، عن طريق جعل قيمة كلا الطرفين LOW فلا يصل التيار للمحرك فيتوقف.

```
void motor_stop(void)          // the function that will cause the motor to stop
rotating
{
  digitalWrite(MOTOR_IN1, LOW);
  digitalWrite(MOTOR_IN2, LOW);
}
```