كلية الهندسة
قسم هندسة الميكاترونيك

المتحكمات الصغرية و النظم المضمنة

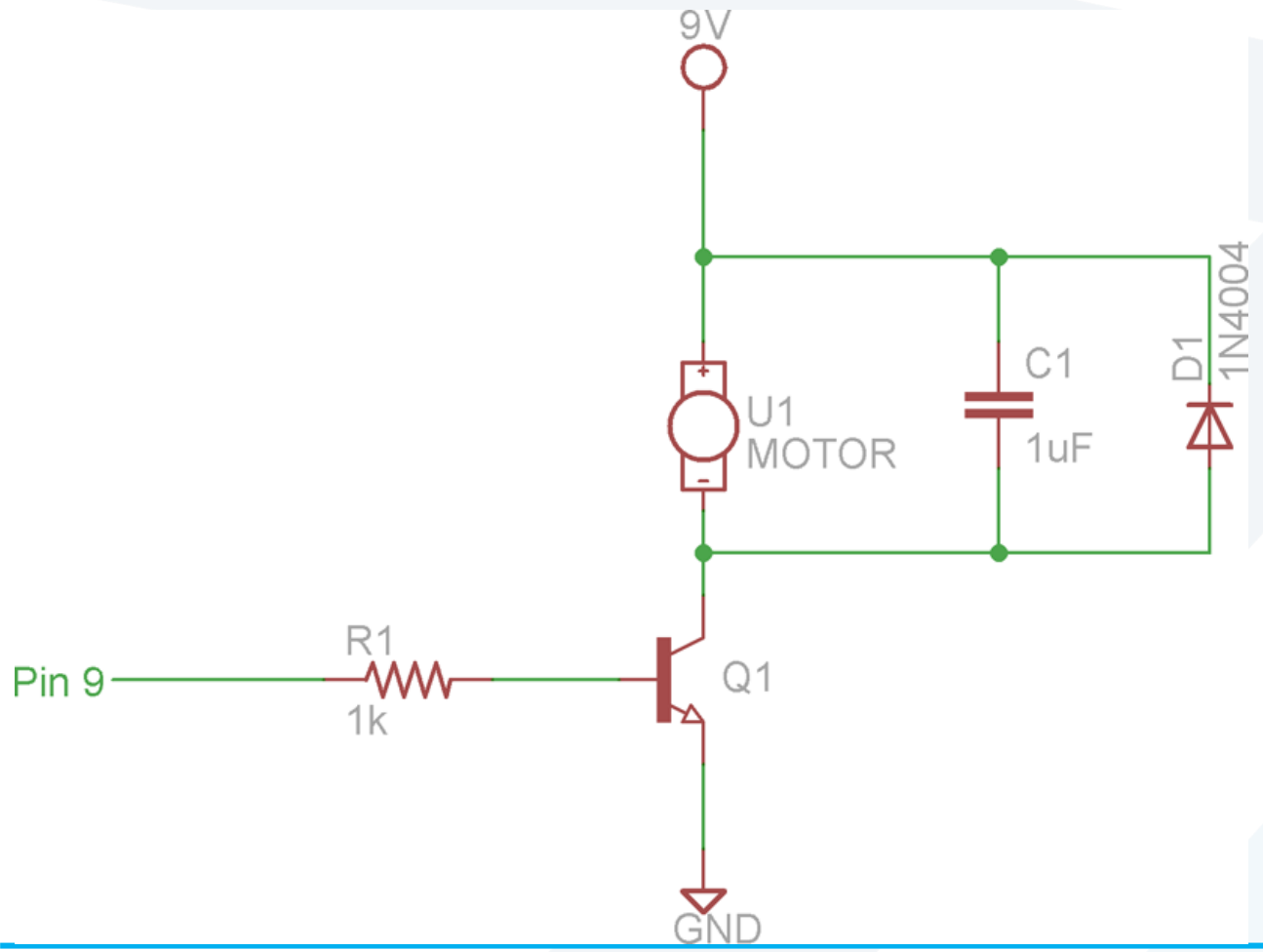
محاضرة 11

Arduino

د. فادي متوج

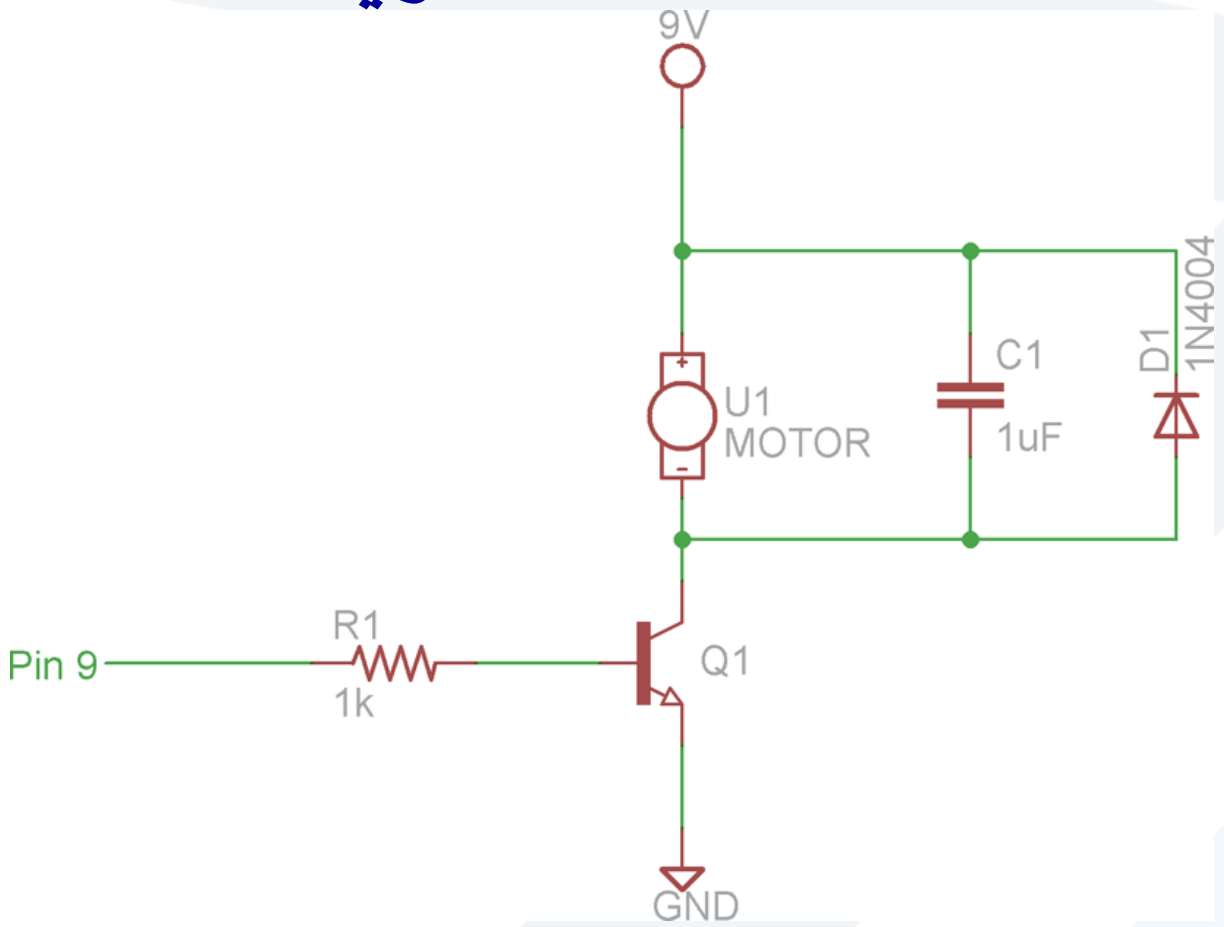
قيادة محركات التيار المستمر DC

- تدور محركات التيار المستمر عندما يتم تطبيق جهد تيار مستمر عليها.
- توجد مثل هذه المحركات بشكل شائع كمحركات القيادة في سيارات التحكم اللاسلكي (RC)، والمحركات التي تجعل الأقراص تدور في قارئ الـ DVD.
- تتوفرمحركات التيار المستمر بأحجام متعددة وهي رخيصة جدًا بشكل عام.
- من خلال ضبط الجهد الذي نطبقه عليها، يمكننا تغيير سرعة دورانها.
- من خلال عكس اتجاه الجهد المطبق عليها، يمكننا تغيير اتجاه دورانها أيضًا. يتم ذلك بشكل عام باستخدام دارة تسمى H-bridge



- Q1 هو ترانزستور من نوع NPN يستخدم لتوصيل و قطع جهد البطارية 9 فولت إلى المحرك.
- تستخدم المقاومة 1 kΩ لفصل قاعدة الترانزستور عن رجل التحكم في Arduino
- U1 هو محرك DC
- C1 مكثف مخصص لفلترة الضجيج الناتج عن المحرك.
- D1 هو ديود يستخدم لحماية البطارية من الجهد العكسي الناتج عن المحرك الذي يعمل كحمل تحريضي.

توصيل محرك DC إلى Arduino بمصدر طاقة ثانوي



- تتطلب محركات التيار المستمر عمومًا تيارًا أكبر مما يمكن أن يوفره Arduino.
- لمعالجة هذه المشكلة، سنتعلم أولاً كيفية عزل محرك التيار المستمر عن Arduino، ثم كيفية تزويده بالطاقة من خلال مصدر طاقة ثانوي.
- سيسمح الترانزستور لـ Arduino بتشغيل وإيقاف المحرك بأمان، بالإضافة إلى التحكم في السرعة باستخدام تقنية تعديل عرض النبض (PWM).

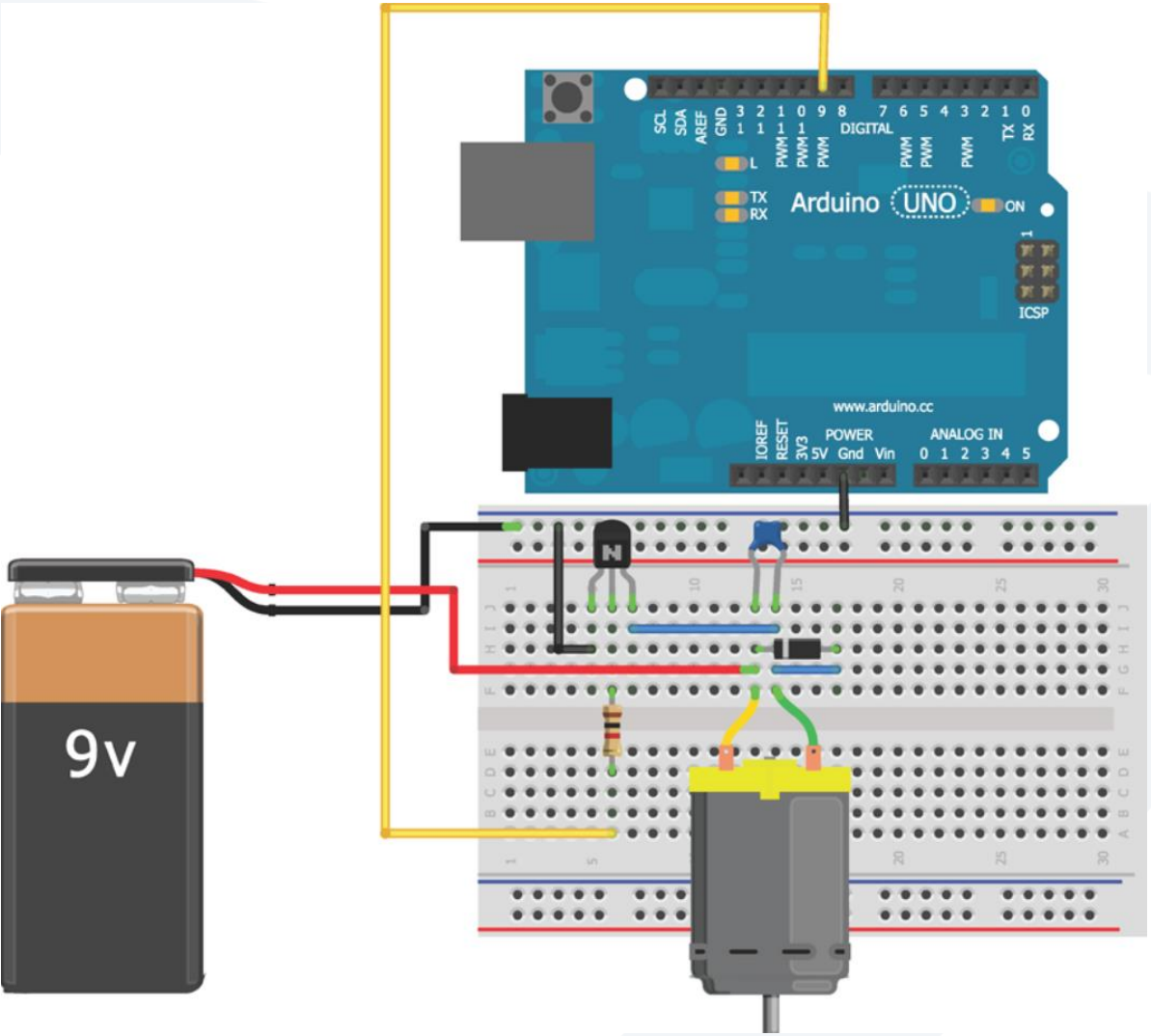
توصيل محرك DC إلى Arduino بمصدر طاقة ثانوي

- يمكن لبعض محركات التيار المستمر أن تستهلك تيارًا أكبر مما يمكن أن يعطيه المنبع5V في Arduino
- العديد من المحركات تعمل بجهد أعلى من 5 فولت. على الرغم من أنها قد تدور عند 5 فولت، إلا أنها تصل إلى سرعتها القصوى عند 9 فولت أو 12 فولت فقط.
- يجب علينا توصيل الأرضي لكل من مصدر الطاقة الثانوي وأرضي Arduino مع بعضهما.
- يضمن هذا الاتصال نقطة مرجعية مشتركة بين مستويات الجهد في جزئي الدارة.

استخدام الترانزستور كمفتاح

- من خلال تغيير الجهد على قاعدة الترانزستور، يمكننا التحكم في السماح أو عدم السماح للتيار بالتدفق.
- عندما يتم تطبيق جهد عالٍ بدرجة كافية على القاعدة، يُسمح للتيار بالتدفق عبر الترانزستور، ونتيجة لذلك يدور المحرك.
- إن 5فولت المولدة من قبل الأرجل الرقمية لبورد Arduino أكثر من كافية لجعل الترانزستور بحالة ON
- من خلال الاستفادة من PWM، يمكننا التحكم في سرعة المحرك عن طريق تشغيل وإيقاف الترانزستور بسرعة.
- نظرًا لأن المحرك يمكنه الحفاظ على الزخم، فإن دورة العمل (duty cycle)لإشارة PWM تحدد سرعة المحرك.
- يقوم الترانزستور بشكل أساسي بتوصيل وفصل أحد طرفي المحرك عن الأرضي و بالتالي التحكم بوصل و فصل دارة المحرك.

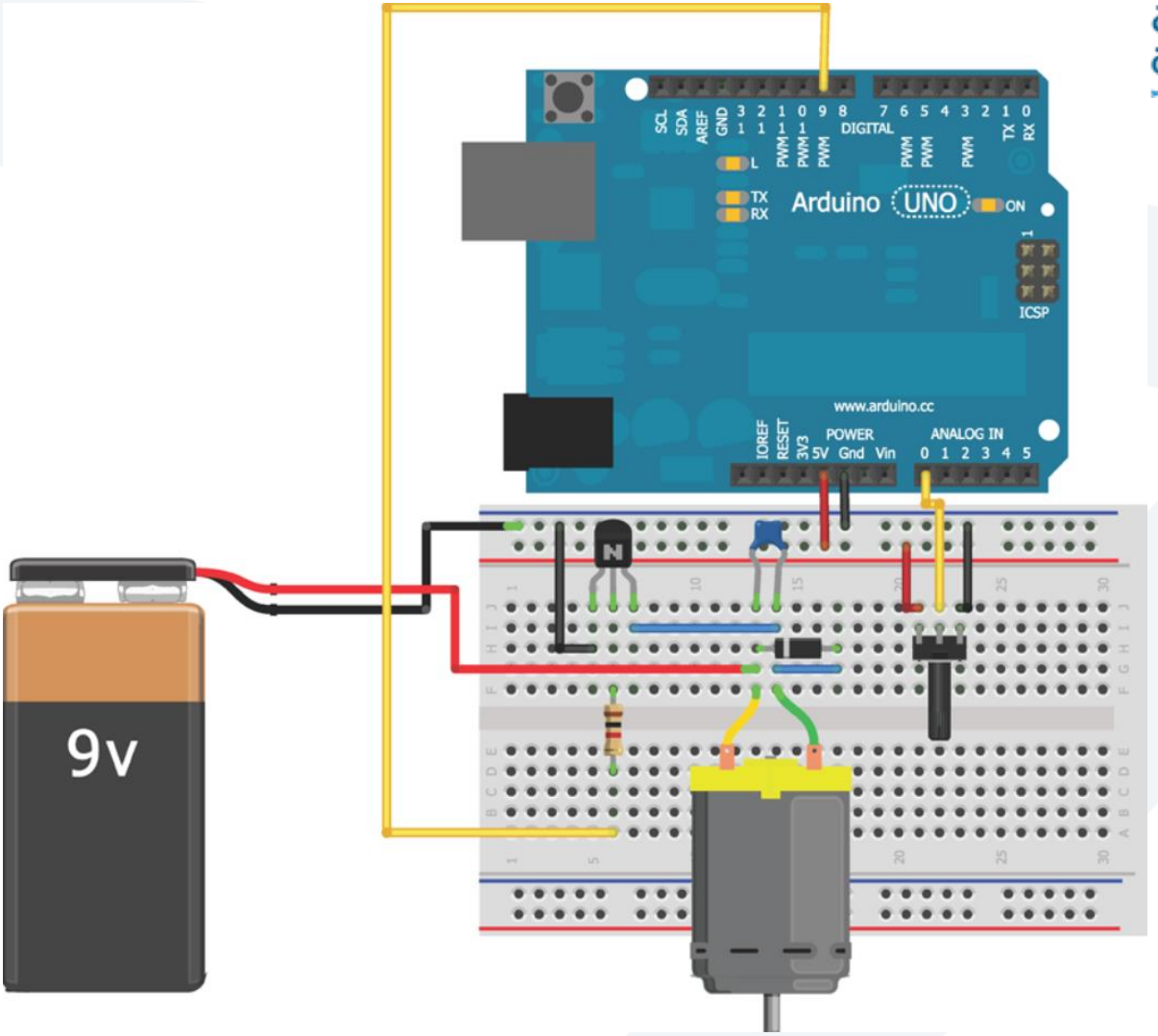
التحكم في سرعة المحرك باستخدام PWM



```
const int MOTOR=9;  //Motor on Digital Pin 9
void setup() {
pinMode (MOTOR, OUTPUT);
}
void loop() {
for (int i=0; i<256; i++)
{
analogWrite(MOTOR, i);
delay(10);
}
delay(2000);
for (int i=255; i>=0; i--)
{
analogWrite(MOTOR, i);
delay(10);
}
delay(2000); }
```

- من خلال إرسال إشارات PWM بدورة عمل متغيرة إلى الترانزستور، فإن التيار يمر عبر المحرك و يتوقف و تتكرر هذه العملية بسرعة (حسب تردد إشارة PWM) مما يؤدي إلى تغيير في سرعة المحرك.
- إذا تم توصيل كل شيء بشكل صحيح، يجب أن يؤدي الكود السابق إلى زيادة سرعة المحرك ببطء، ثم تتناقص سرعته مرة أخرى و تتكرر هذه العملية ضمن تابع الـ loop.

التحكم بسرعة المحرك باستخدام مقاومة متغيرة

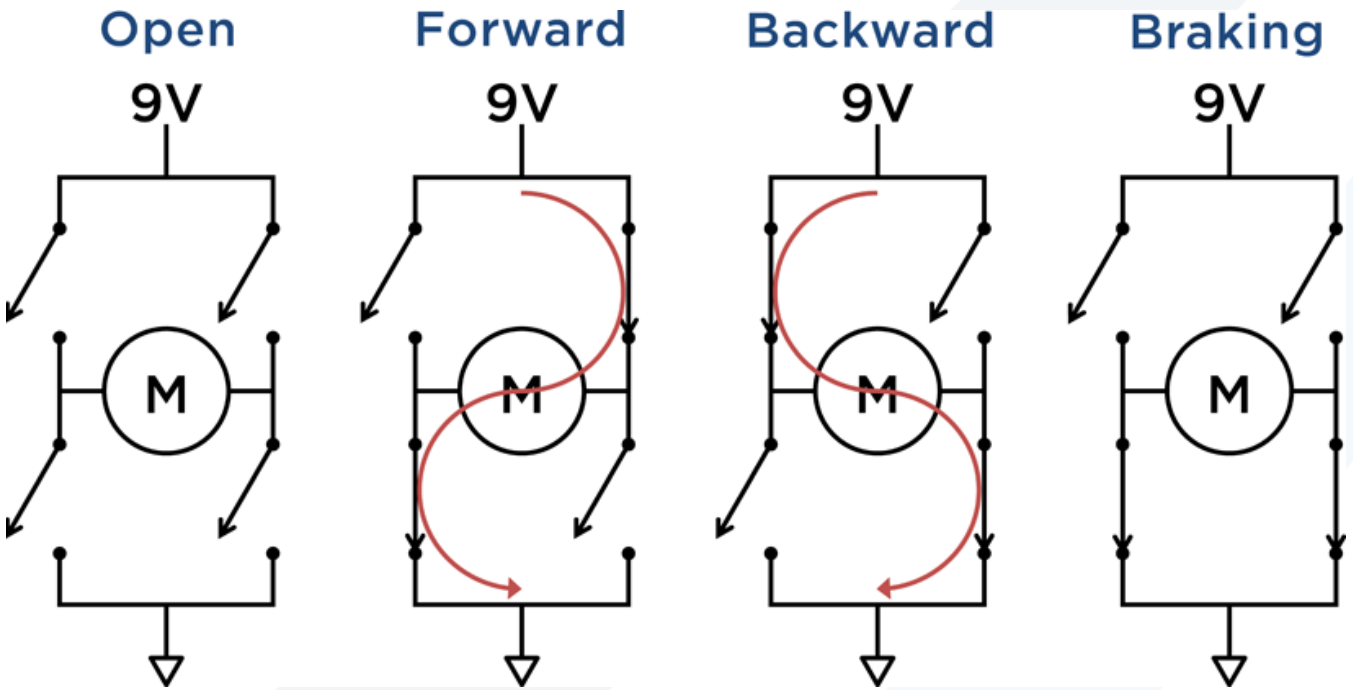


- بإضافة مقاومة متغيرة إلى الرجل A0 التشابهية، يمكننا ضبط سرعة المحرك يدويًا.
- يمكننا الآن التحكم في سرعة المحرك بناءً على القيمة الحالية للمقاومة المتغيرة.
- عند ضبط قيمة المقاومة المتغيرة على الصفر، يتوقف المحرك.
- مع تدوير المقاومة المتغيرة بحيث تصل إلى أقصى قيمة ، يدور المحرك بأقصى سرعة.
- أردوينو يعمل بسرعة كبيرة حيث أنه يمر عبر الحلقة loop عدة آلاف من المرات كل ثانية. لذلك فإن سرعة المحرك يتم ضبطها في الزمن الحقيقي باستخدام المقاومة المتغيرة.

```
//Motor Speed Control with a Potentiometer
const int MOTOR=9;      //Motor on Digital Pin 9
const int POT=0;        //POT on Analog Pin 0
int val = 0;
void setup()
{
pinMode (MOTOR, OUTPUT);
}
void loop()
{
val = analogRead(POT);
val = map(val, 0, 1023, 0, 255);
analogWrite(MOTOR, val);
}
```

استخدام H-bridge للتحكم في اتجاه دوران محرك التيار المستمر

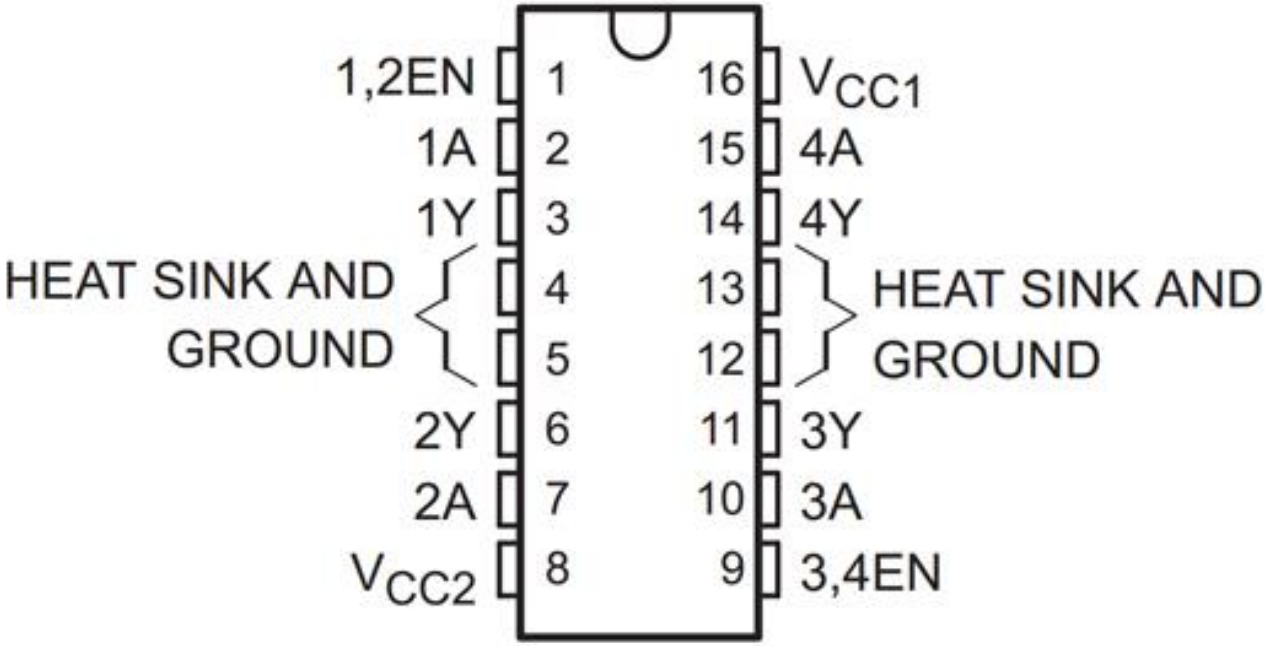
- يجب على أي محرك تيار مستمر أن يكون قادرًا على الدوران في اتجاهين. لتحقيق ذلك ، يمكننا استخدام دارة تدعى H-bridge .



استخدام الترانزستور كمفتاح

- يشكل المحرك مع السويتشات الأربعة حرف H كبير.
- السويتشات في هذه الدارة هي في الواقع عبارة عن ترانزستورات.
- يحتوي H-bridge على أربع حالات عمل رئيسية: الفتح، والفرملة، والأمام، والخلف.
- في حالة الفتح، تكون جميع السويتشات مفتوحة ولا يدور المحرك.
- في حالة الدوران الأمامي، يتم وصل سويتشين متعاكسين قطريًا، مما يؤدي لمرور التيار من المنبع 9 فولت، عبر المحرك، وصولاً إلى الأرضي.
- عندما يتم وصل السويتشات المعاكسة، يمر التيار عبر المحرك في الاتجاه المعاكس، مما يتسبب في دورانه في الاتجاه المعاكس.
- في حالة الكبح، تتوقف كل الحركة المتبقية الناتجة عن الزخم ويتوقف المحرك.

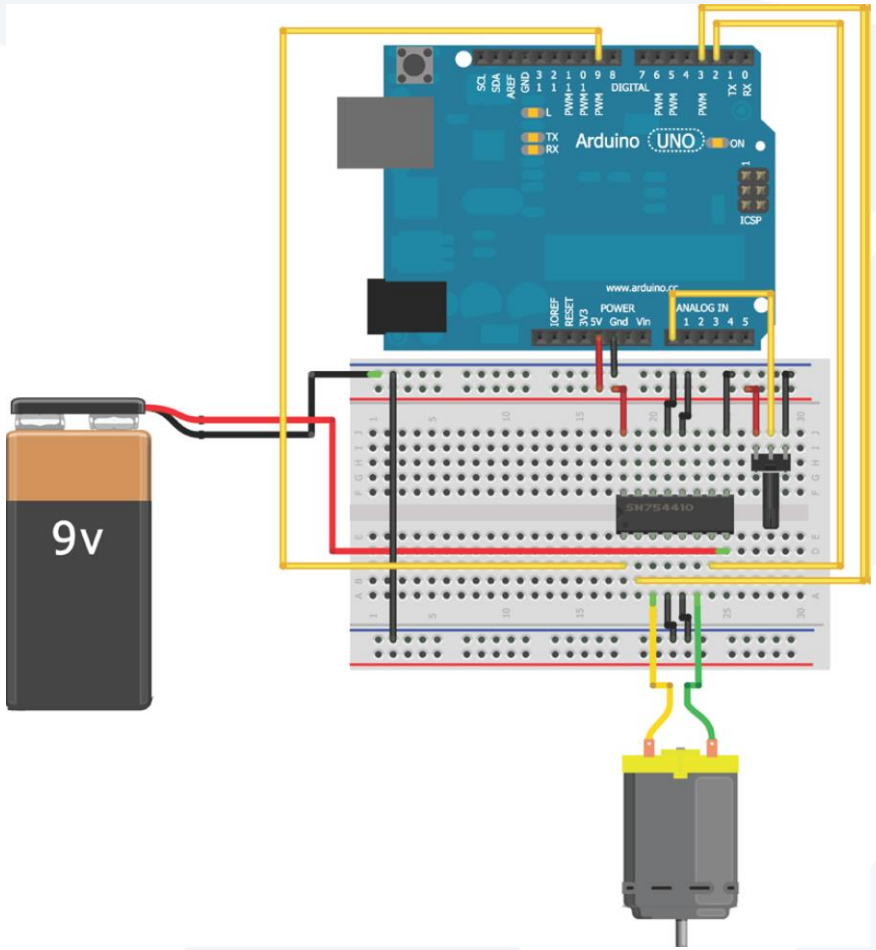
L293 H-bridge

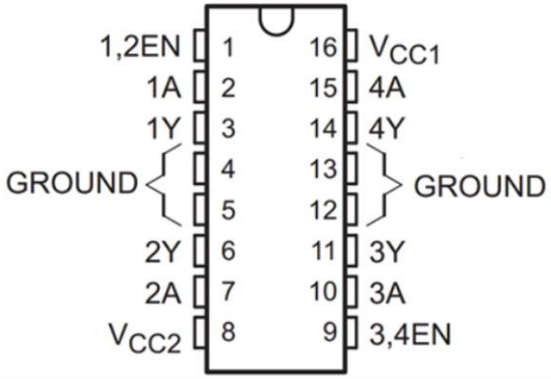


L293 H-bridge

- GND (الأرجل 4 و 5 و 12 و 13): تتصل الأرجل الأربعة بالأرضي المشترك بين المنبعين 9 فولت و 5 فولت.
- VCC2 (الرجل 8): تؤمن الرجل VCC2تيار المحرك، لذلك نقوم بتوصيلها بالمنبع 9 فولت.
- VCC1 (الرجل 16): تؤمن VCC1 الحالة المنطقية المطلوبة لعمل الدارة، لذلك نقوم بتوصيلها بجهد 5 فولت.
- 1Yو 2Y (الأرجل 3 و 6): تتصل أسلاك المحرك الأول بهذه الأرجل.
- 1Aو 2A (الأرجل 2 و 7): يتم التحكم في حالة السويتشات الموجودة على اليسار بواسطة هذه الأرجل ، لذا فهي متصلة بأرجل الإدخال / الإخراج على Arduino للتبديل.
- 1,2EN (الرجل 1): تستخدم هذه الرجل لتمكين أو تعطيل الـH-bridge اليسارية. وهي متصلة برجل PWM على بورد Arduino، بحيث يمكن التحكم في السرعة.
- 3Y و 4Y (الأرجل 11 و 14): تتصل أسلاك المحرك الثاني بهذه الأرجل. نظرًا لأننا نستخدم محرك واحد فقط، يمكننا ترك هذه الأرجل غير متصلة.
- 3A و 4A (الأرجل 10 و 15): يتم التحكم في حالة السويتشات الموجودة على اليمين بواسطة هذه الأرجل ، لكننا نستخدم فقط المحرك الأيسر في هذا المثال، لذلك يمكننا تركها بدون توصيل.
- 3,4EN (الرجل 9): تستخدم هذه الرجل لتمكين أو تعطيل الـ H-bridge اليمينية. ، لكننا نستخدم فقط المحرك الأيسر في هذا المثال، لذلك يمكننا تعطيلها عن طريق توصيل هذه الرجل مباشرة بـ GND

التحكم في اتجاه وسرعة المحرك باستخدام مقاومة متغيرة و دارة H-bridge





- سنتحكم في المحرك على النحو التالي:
- ضبط قيمة المقاومة المتغيرة في مجال محدد يتوسط مجال قيم المقاومة المتغيرة يوقف المحرك.
- يؤدي ضبط قيمة المقاومة المتغيرة في مجال أعلى من الوسط إلى زيادة السرعة إلى الأمام.
- يؤدي ضبط قيمة المقاومة المتغيرة في نطاق أقل من الوسط إلى زيادة السرعة للخلف.
- سنستخدم توابع في برنامج Arduino الخاص بنا:
  - brake لإيقاف المحرك.
  - forward يجعله يدور للأمام بسرعة محددة.
  - reverse يجعله يدور للخلف بسرعة محددة.

```
//Motor goes forward at given rate (from 0-255)
void forward (int rate)
{
digitalWrite(EN, LOW);
digitalWrite(MC1, HIGH);
digitalWrite(MC2, LOW);
analogWrite(EN, rate);
}
//Motor goes backward at given rate (from 0-255)
void reverse (int rate)
{
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, HIGH);
analogWrite(EN, rate);
}
//Stops motor
void brake ()
{
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, LOW);
digitalWrite(EN, HIGH);
}
```

Functions

البرنامج الكامل (1/3)

```
const int EN=9;     //Half Bridge 1 Enable
const int MC1=3;    //Motor Control 1
const int MC2=2;    //Motor Control 2
const int POT=0;    //POT on Analog Pin 0
int val = 0;        //for storing the reading from the POT
int velocity = 0;       //For storing the desired velocity (from 0-255)
void setup()
{
pinMode(EN, OUTPUT);
pinMode(MC1, OUTPUT);
pinMode(MC2, OUTPUT);
brake();        //Initialize with motor stopped
}
```

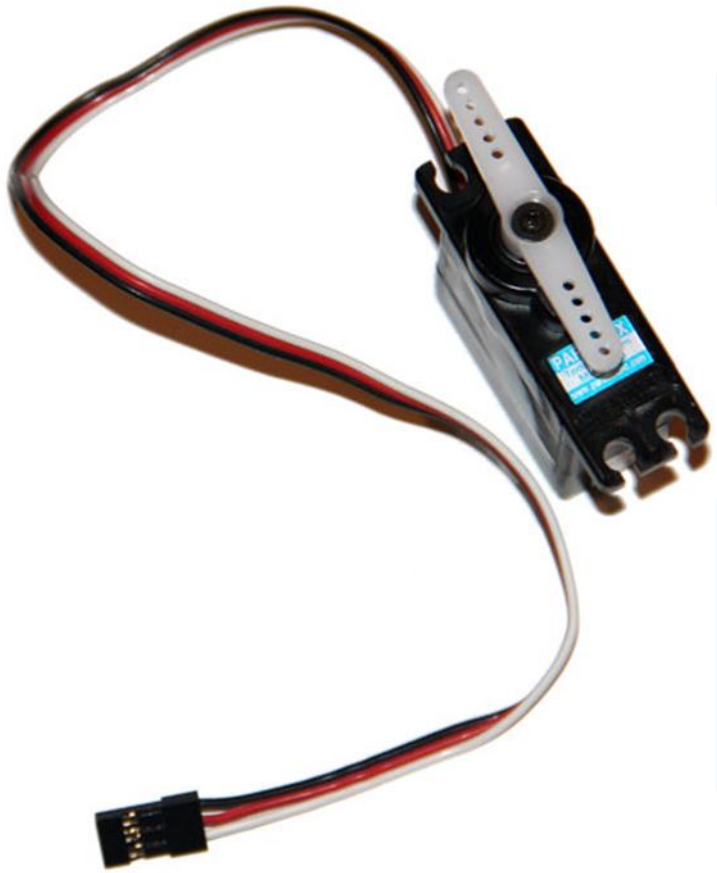
البرنامج الكامل (2/3)

```
void loop()
{
val = analogRead(POT);
//go forward
if (val > 562)
{
velocity = map(val, 563, 1023, 0, 255);
forward(velocity);
}
//go backward
else if (val < 462)
{
velocity = map(val, 461, 0, 0, 255);
reverse(velocity);
}
//brake
else
{brake(); } }
```

البرنامج الكامل (3/3)

```
//Motor goes forward at given rate (from 0-255)
void forward (int rate) {
digitalWrite(EN, LOW);
digitalWrite(MC1, HIGH);
digitalWrite(MC2, LOW);
analogWrite(EN, rate); }
//Motor goes backward at given rate (from 0-255)
void reverse (int rate) {
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, HIGH);
analogWrite(EN, rate); }
//Stops motor
void brake () {
digitalWrite(EN, LOW);
digitalWrite(MC1, LOW);
digitalWrite(MC2, LOW);
digitalWrite(EN, HIGH); }
```

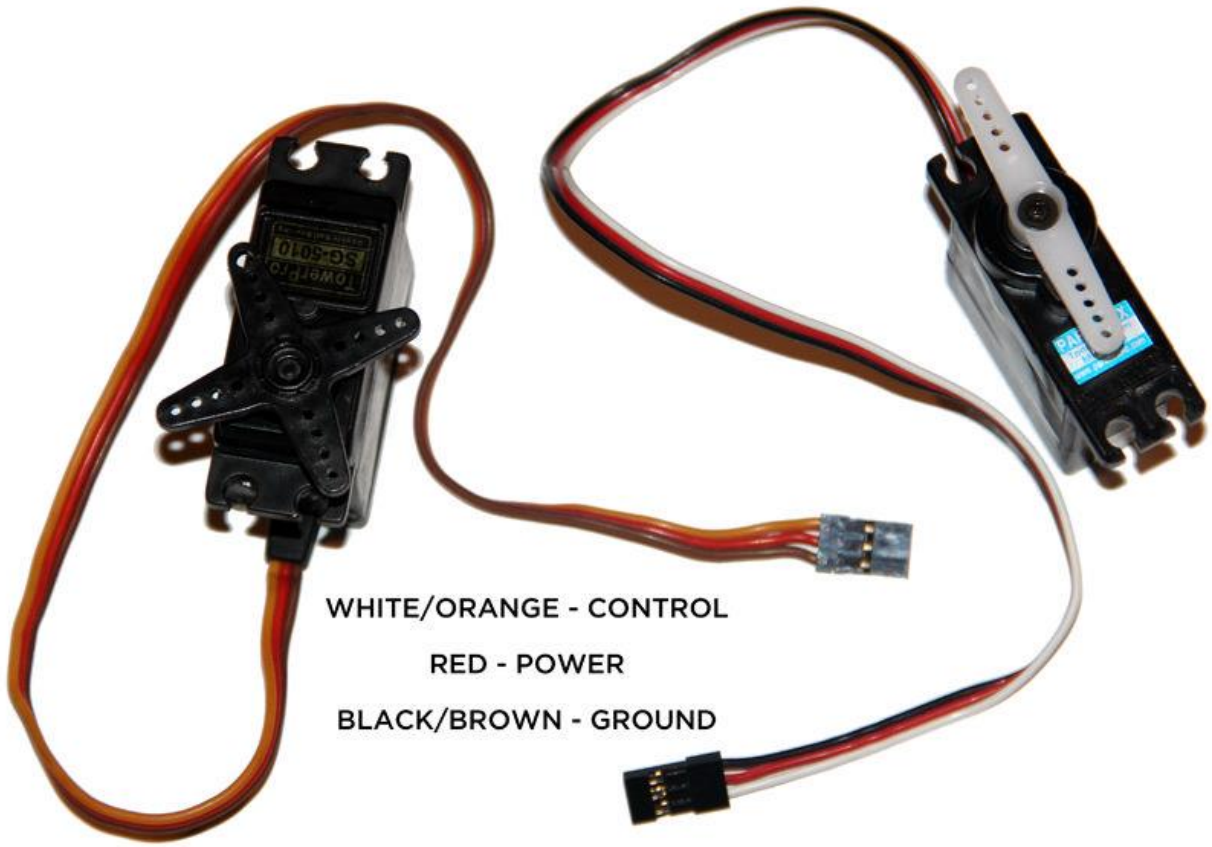
قيادة محركات الـ servo



محركات الـ servo

- تعمل محركات التيار المستمر كمحركات قيادة ممتازة في كثير من المشاريع و التطبيقات، ولكنها ليست مثالية للعمل الدقيق نظرًا لعدم قدرتها على تحديد موضعها الدقيق. بمعنى آخر، بدون استخدام مشفر خارجي من نوع ما، لن نعرف أبدًا الموضع الدقيق لمحرك الـ DC.
- على النقيض من ذلك، تعد محركات السيرفو فريدة من نوعها من حيث قدرتها على الدوران إلى موضع زاوي معين وتبقى هناك حتى يطلب منها الانتقال إلى موضع جديد.
- تشمل أمثلة استخدام محركات السيرفو أنظمة أقفال الأبواب، والتحكم بدقة في فتح عدسة كاميرا الخ .....

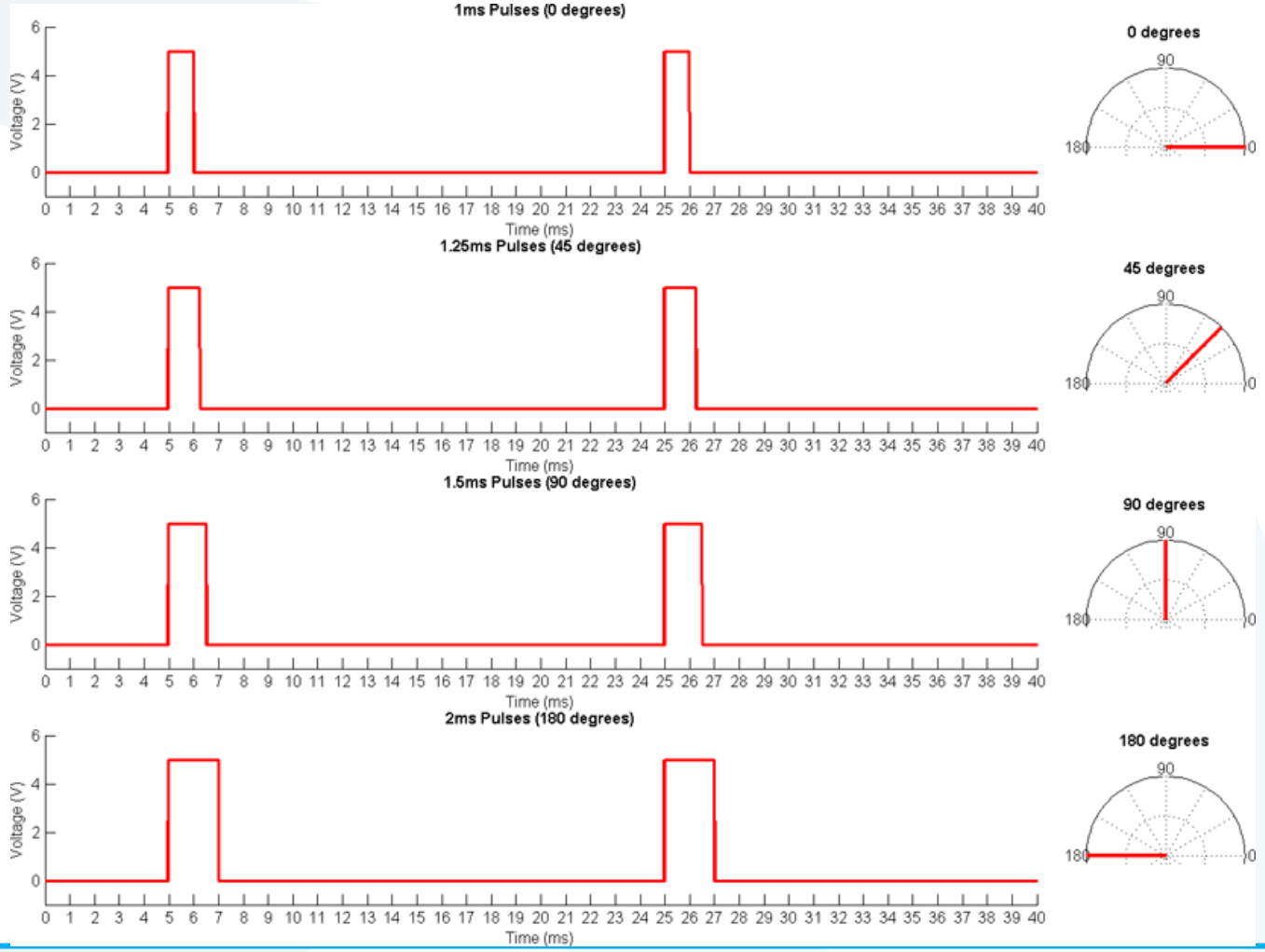
التحكم بمحركات الـ servo

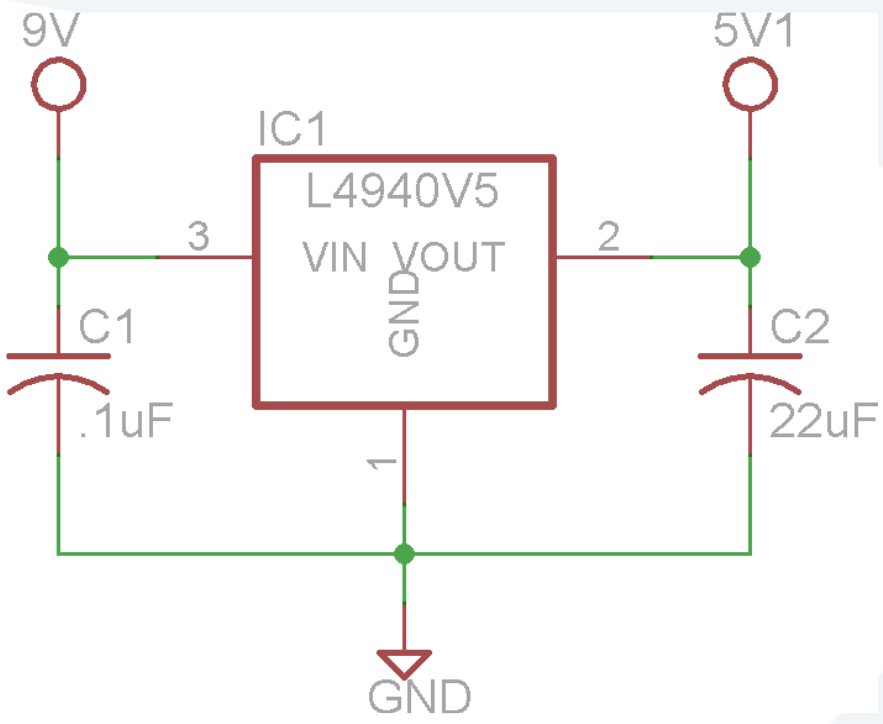


- على عكس نظيراتها من محركات التيار المستمر، تحتوي محركات الـ servo على ثلاثة أرجل: رجل للتزود بالطاقة (عادةً حمراء)، والأرضي (عادةً ما تكون بنية أو سوداء) ، و رجل تستقبل إشارة التحكم (عادةً ما تكون بيضاء أو برتقالية).

التحكم بمحركات الـ servo

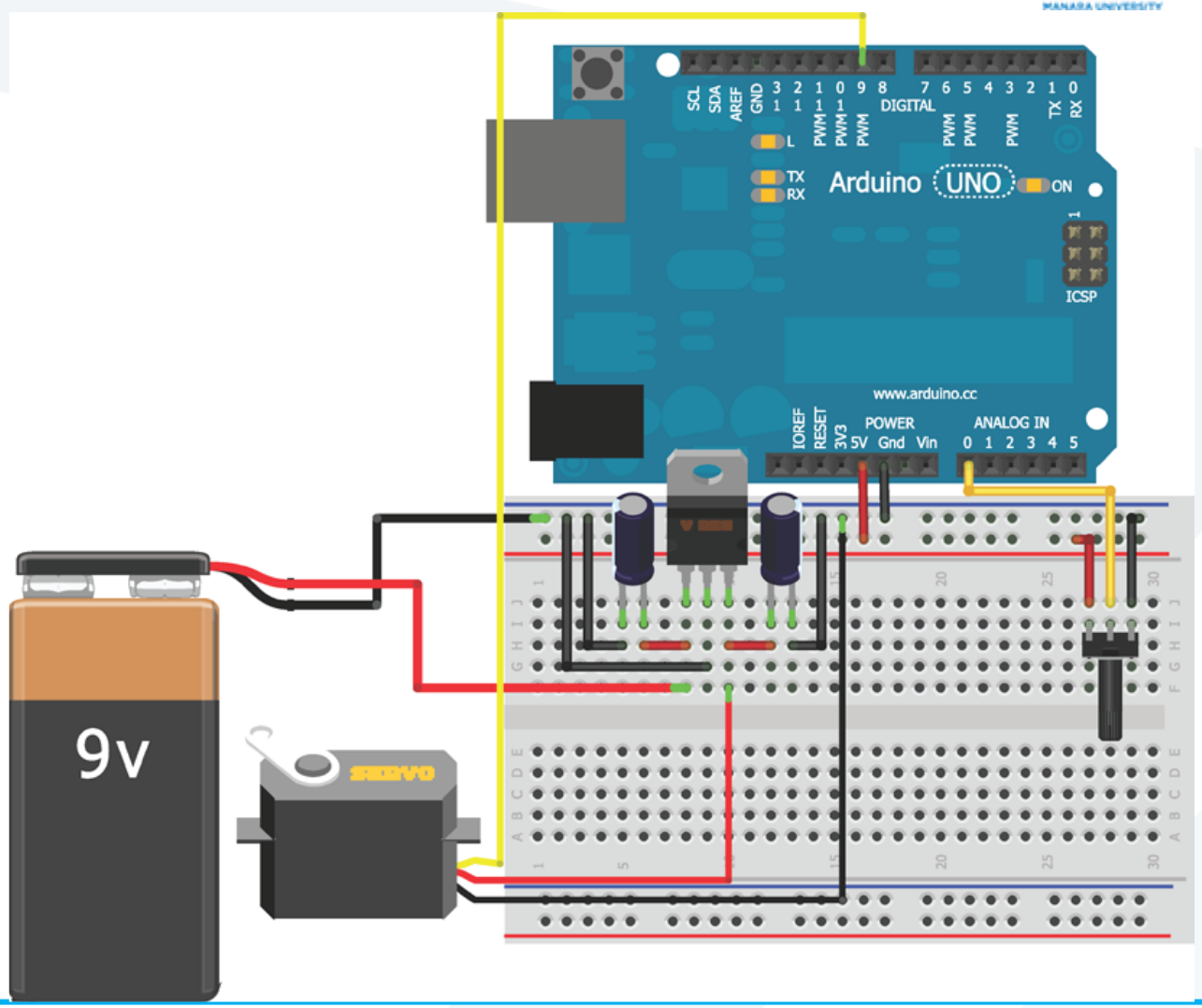
- يمكن أن تسحب محركات السيرفو تيار أكثر مما يمكن لـ Arduino توفيره.
- يجب دائمًا توصيل أرجل الطاقة والأرضي لمحرك السيرفو بمصدر طاقة ثابت.
- يتم التحكم في محركات السيرفو باستخدام عرض النبضة القابل للتعديل على رجل استقبال الإشارة.
- يؤدي إرسال نبضة 5 فولت لمدة 1ms إلى دوران المحرك إلى الزاوية 0 درجة.
- يؤدي إرسال نبضة 5 فولت لمدة 2ms إلى دوران المحرك إلى الزاوية 180 درجة
- يؤدي إرسال نبضة 5 فولت لمدة 1.5ms إلى دوران المحرك إلى الزاوية 90 درجة.
- بمجرد إرسال نبضة تحكم، يدور السيرفو إلى هذا الموضع المحدد ويبقى هناك حتى يتم تلقي نبضة تحكم أخرى.
- ومع ذلك، إذا أردنا أن يحتفظ محرك السيرفو بموقعه (يمانع الضغط عليه لتغيير موضعه ومحاولة الحفاظ على الموضع المحدد) ، عندئذ نعيد إرسال الأمر مرة كل 20 مللي ثانية.


1ms Pulses (0 degrees)
0 degrees
1.25ms Pulses (45 degrees)
45 degrees
1.5ms Pulses (90 degrees)
90 degrees
2ms Pulses (180 degrees)
180 degrees
Voltage (V)
Time (ms)
90
180
0



- يمكن أن تسحب محركات السيرفو تيار أكثر مما يمكن لـ Arduino توفيره. ومع ذلك، فإن معظم محركات السيرفو مصممة لتعمل بجهد 5 فولت، وليس 9 فولت أو 12 فولت مثل محرك التيار المستمر.
- على الرغم من أن الجهد هو نفسه الذي يعطيه Arduino، فإننا نستخدم مصدر طاقة منفصل يمكنه توفير تيار أكبر.
- للقيام بذلك، نستخدم بطارية 9 فولت ومنظم جهد لتوليد 5 فولت من بطارية 9 فولت.
- منظم الجهد عبارة عن عنصر الكتروني يحتوي بشكل عام على ثلاثة أرجل: جهد الدخل، جهد الخرج، والأرضي.
- يتم توصيل رجل الأرضي للمنظم بكل من أرضي جهد الدخل وأرضي جهد الخرج.
- في منظمات الجهد، يجب أن يكون جهد الدخل دائمًا أعلى من جهد الخرج، ويتم ضبط جهد الخرج على قيمة ثابتة اعتمادًا على المنظم الذي نستخدمه.
- يتحول انخفاض الجهد بين الدخل و الخرج على شكل حرارة، ويهتم المنظم بضمان بقاء الخرج دائمًا كما هو، حتى مع انخفاض جهد الدخل (في حالة تفريغ البطارية بمرور الوقت).
- L4940V5 عبارة عن منظم جهد بجهد 5 فولت قادر على توفير ما يصل إلى 1.5 أمبير عند 5 فولت.

مثال : التحكم بمحرك servo



- يشتمل Arduino IDE على مكتبة مدمجة تجعل التحكم في محركات السيرفو أمرًا سهلاً.
- المكتبة عبارة عن مجموعة من التعليمات البرمجية المفيدة.
- كل ما يتعين علينا القيام به هو وصل محرك سيرفو برجل معينة وإعطائه زاوية ليدور إليها.
- تهتم المكتبة بالباقي، حتى أن المكتبة تقوم بتعيين رجل التوصيل كخرج.
- إن أبسط طريقة لاختبار وظائف محرك السيرفو لدينا هي التحكم بمحرك السيرفو من خلال مقاومة متغيرة كما فعلنا في مثال سابق مع محرك DC .
- يؤدي وضع المقاومة المتغيرة على القيمة 0 إلى دوران المحرك إلى الزاوية 0 درجة.
- يؤدي وضع المقاومة المتغيرة على القيمة 1023 إلى دوران المحرك إلى الزاوية 180 درجة.

الكود البرمجي للمثال

```
#include <Servo.h>
const int SERVO=9;  //Servo on Pin 9
const int POT=0;    //POT on Analog Pin 0
Servo myServo;
int val = 0;      //for storing the reading from the POT
void setup() {
myServo.attach(SERVO);
}
void loop() {
val = analogRead(POT);  //Read Pot
val = map(val, 0, 1023, 0, 179);   //scale it to servo range
myServo.write(val); //sets the servo
delay(15);  //waits for the servo
}
```