



جامعة حلب في المناطق المحررة

كلية الهندسة المعلوماتية

المادة: معالج مصغر

السنة: الثالثة

الفئة: الثالثة

## مكنسة كهربائية ذكية تعتمد على الاردوينو

بإشراف

م. عبد الرحمن الاغا

د. م. عبد القادر غزال

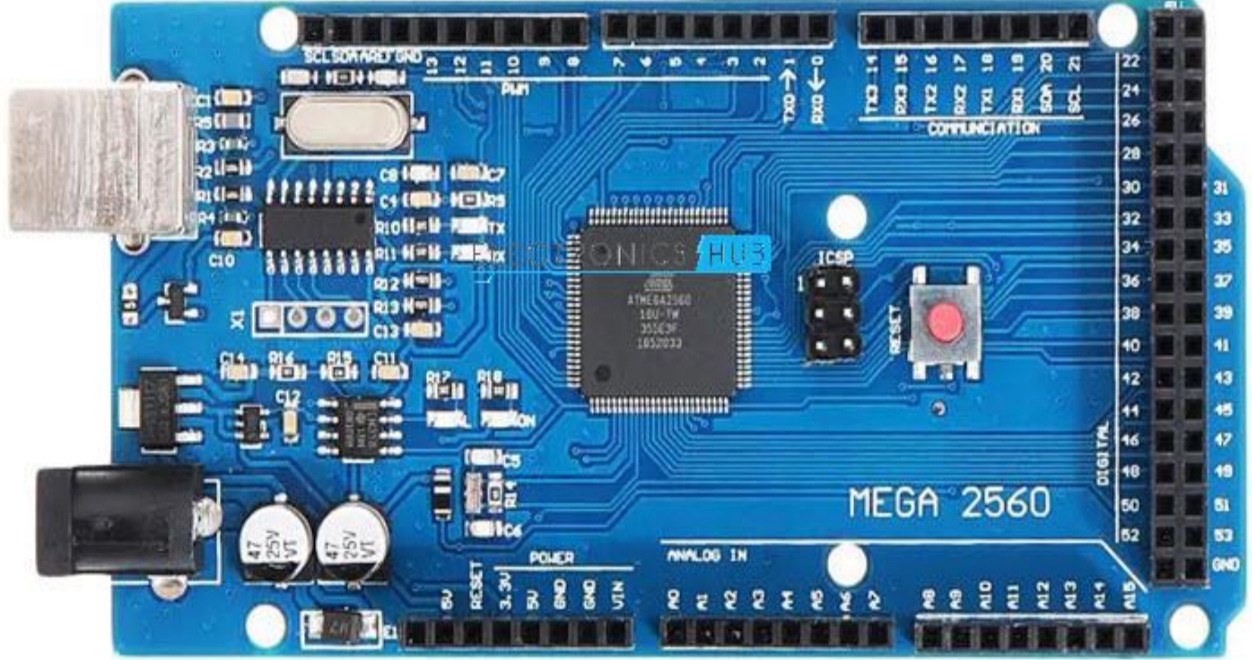
اعداد الطلاب

معاذ مصطفى الدرويش

جمعة محمد علي الدرويش



## ARDUINO MEGA 2560

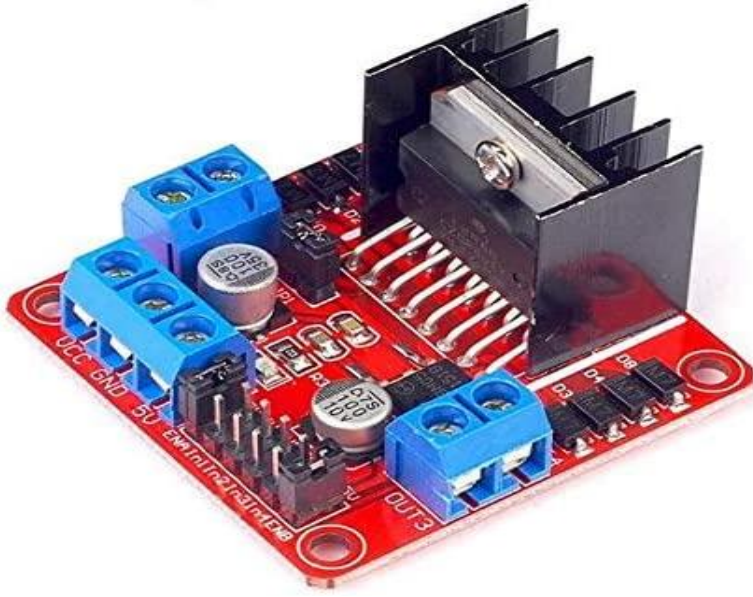


هو عبارة عن مشروع مفتوح المصدر يعتمد على دائرة الكترونية تضم متحكم مصغر وعناصر تغذية اتصال كما انها مجهزة للربط مع حساسات وتوسعات خارجية عديدة

يبرمج المتحكم بلغة ARDUINO Programing language وهي لغة مبسطة من

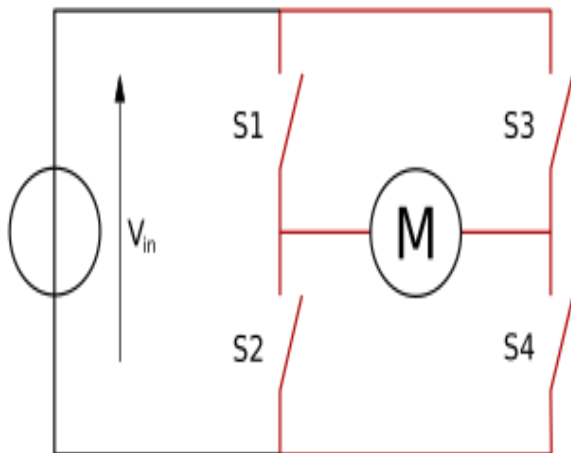
لغة C++ ضمن البيئة البرمجية Arduino IDE

## H-bridge



## H – bridge

تتكون الدائرة إلكترونية من أربع مفاتيح كهربائية. عندما يكون المفتاح الأول  $s_1$  والمفتاح الرابع  $s_4$  مغلقان ويكون المفتاح الثاني  $s_2$  والثالث  $s_3$  مفتوحان يتكون جهد موجب على المحرك. وعندما يكون المفتاح الثاني  $s_2$  والثالث  $s_3$  مغلقان والمفتاح الأول  $s_1$  والرابع  $s_4$  مفتوحان يتكون جهد معكوس. بإغلاق المفتاحين الأول  $s_1$  والثاني  $s_2$  معنا فان ذلك سوف يؤدي إلى صنع دائرة قصر (بالإنكليزية Short : circuit) وكذلك فان إغلاق المفتاحين الثالث  $s_3$  والرابع  $s_4$  سوف يعطي نفس النتيجة وتسمى هذه الحالة بي Shoot-through ويمكن استخدامها في عملية إيقاف تشغيل المحرك. الجدول التالي يتضمن آلية عمل الدائرة الإلكترونية لبعض الاحتمالات الممكنة.



s4	s3	s2	s1	النتيجة
0	0	0	0	المحرك يتحرك حركة حرة
1	0	1	0	المحرك يتحرك إلى اليمين
0	1	0	1	المحرك يتحرك إلى اليسار
0	0	1	1	المحرك يتوقف
1	1	0	0	المحرك يتوقف

## HC-SR04



### حساس المسافة عن طريق الأمواج فوق الصوتية HC-SR04:

يعتمد مبدأ عمل الحساس HC-SR04 على إرسال نبضات فوق صوتية تقع خارج مجال سمع الإنسان، لتنتشر على شكل مخروط وتنعكس مرة أخرى إلى الحساس من أي هدف يوجد في مسار هذه الأمواج، ليستقبلها ويولد لذلك تبعاً نبضة خرج عرضها يساوي زمن الذهاب وعودة الأمواج.

يتوقف الحساس لفترة قصيرة بعد أن ترسل الأمواج فوق الصوتية وخلال ذلك يتم انتظار عودة الأمواج.

بضرب زمن الذهاب أو الإياب بسرعة الصوت وبذلك نحصل على المسافة ما بين الحساس والحاجز الذي أمامه.

يمكن استخدام المتحكم الصغير لحساب عرض النبضة التي يولدها الحساس، وبالتالي حساب المسافة وإظهارها على شاشة إظهار أو الاستفادة منها باستخدامات أخرى.

### محرك تيار مستمر صغير بتروس (DC Gearbox Motor (TT Motor



عبارة عن آلة تحوّل الطاقة الكهربائية إلى طاقة ميكانيكية باستخدام التيار المستمر



## محرك السيرفر: Tower Pro SG90



أحد أنواع محركات التيار المستمر DC motor والذي يُستخدم للتحكم الدقيق، وهي متاحة بنوعين AC وDC، وكلُّ يُستخدم حسب التطبيق .

يتكوّن محرك السيرفر من:

محرك تيار مستمر صغير ومقاومة متغيرة تعمل كمقسم جهد potentiometer لتأمين التغذية الراجعة، بالإضافة لعبة سرعة لزيادة العزم، ونظام للتحكم، بالإضافة لذلك له ثلاثة أسلاك ملوّنة بالأحمر والبرتقالي والبنّي (قد يختلف الكود اللوني من نوع إلى آخر ) حيث أنّ السلك الأحمر يُستخدم للتغذية، والبنّي للأرضي، أما البرتقالي فيستخدم لإشارة التحكم التي ستقوم بتحديد موضع محور المحرك وغالباً ما تكون إشارة

**PWM.**

يتم تشغيل محرك التيار المستمر الموجود ضمن محرك السيرفر من خلال إشارة التحكم التي يستقبلها ووفقاً للتغذية الراجعة من نظام استشعار الموقع، وعندما يحقق محرك السيرفر الموضع المرغوب يتم إيقاف محرك التيار المستمر.

ومن الجدير ذكره أنّ تغذية المحرك الموجود ضمن السيرفر غير ثابتة، لكنّ الاستطاعة الإجمالية يجب أن تكون ثابتة، ويتحقق ذلك من خلال نظام التحكم بالاستطاعة.

السيرفر المُستخدم هنا هو من نوع Tower Pro SG90 البالغ عزمه 8. Kg/cm.

## بوردة شحن بطارية ليثيوم



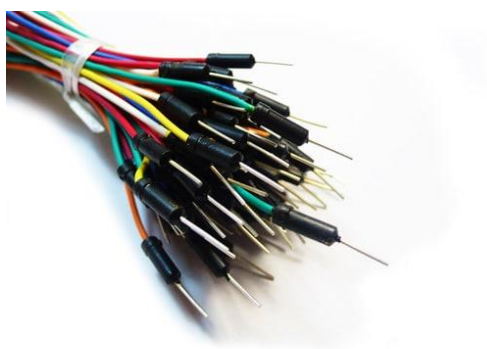
## BATTERY 3.3V



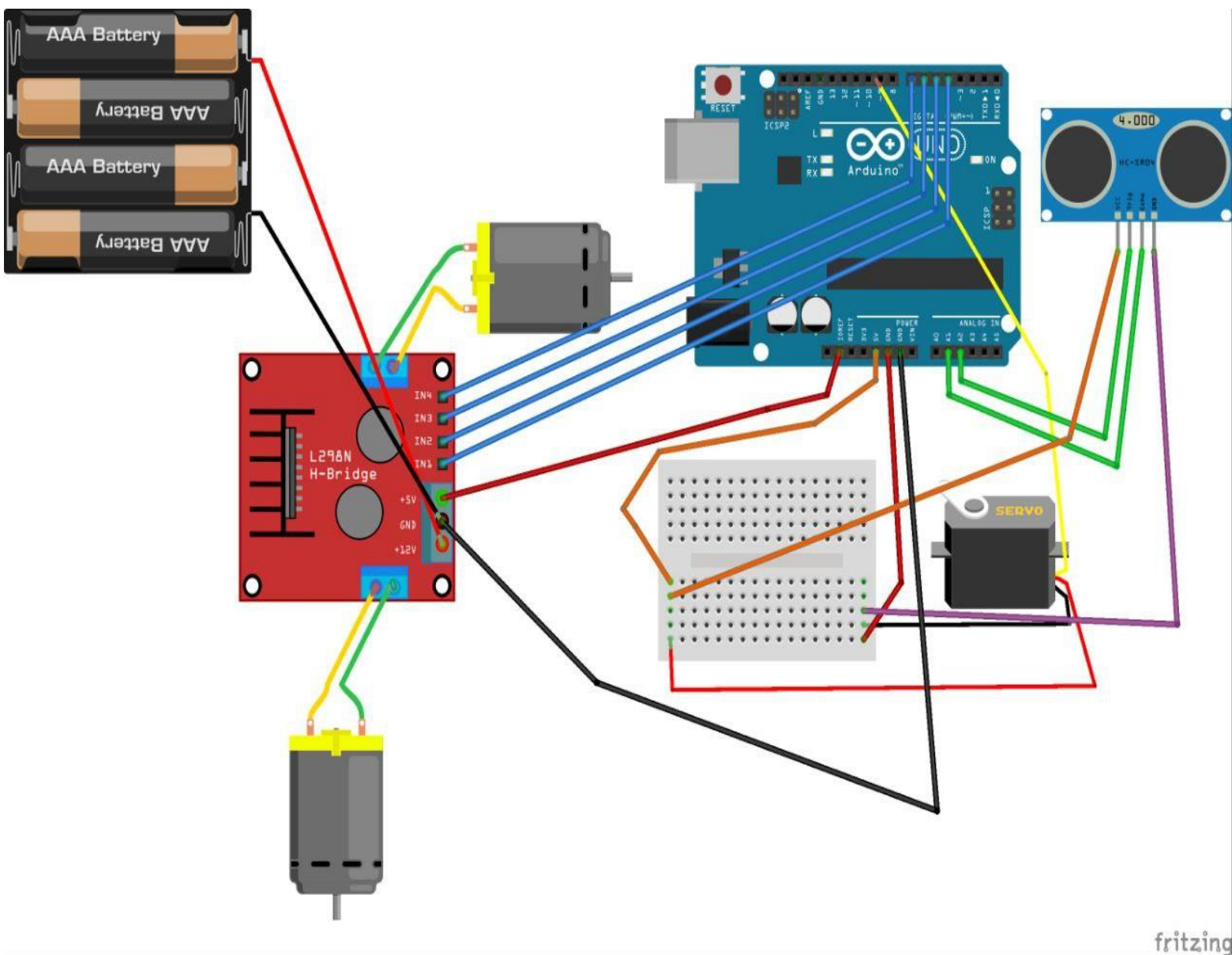
## مروحة شفط ١٢٧

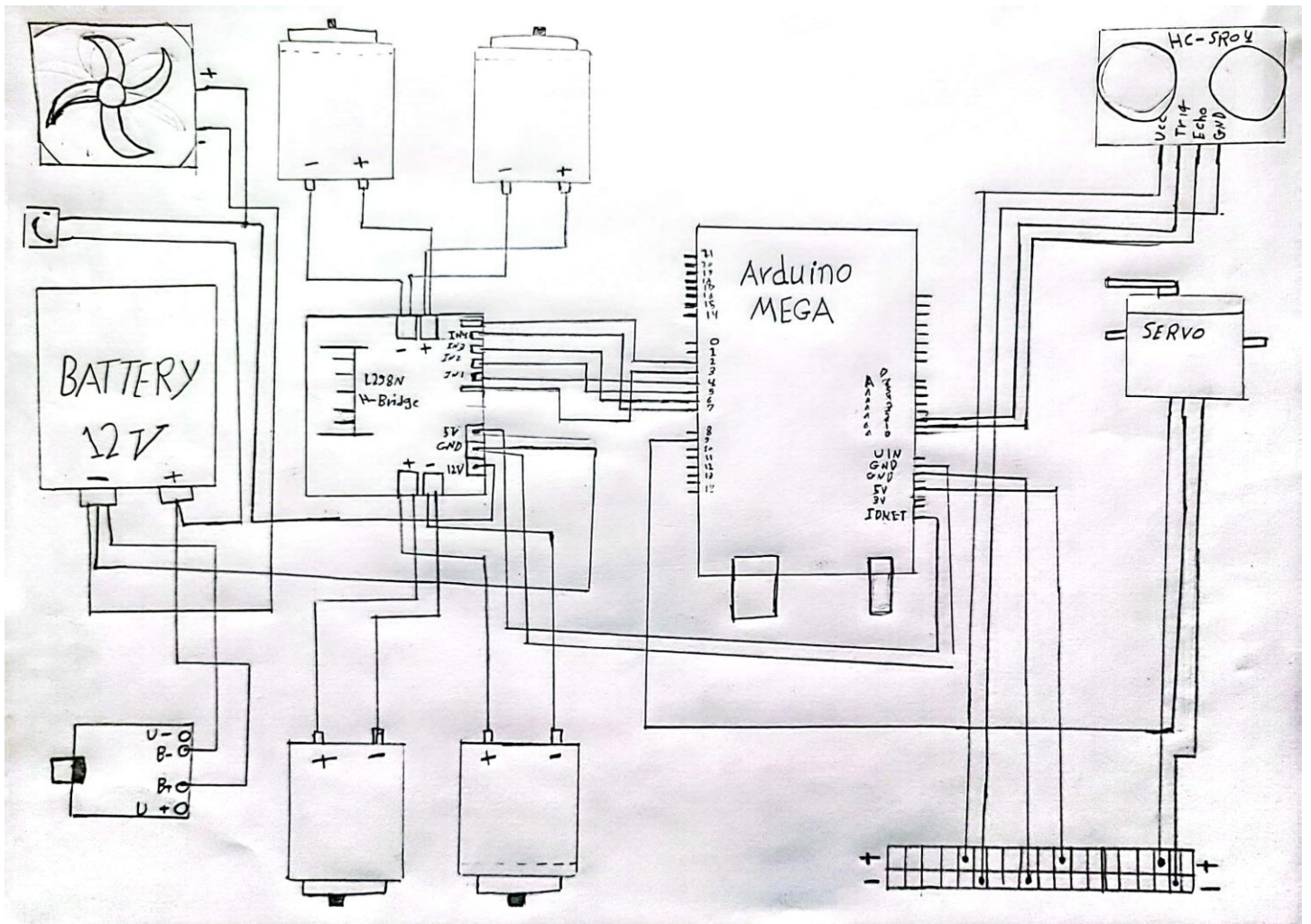


## connecting cable



## مخطط التوصيل





## code

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

```
Servo motor;
```

```
int duration=0;
```

```
int motor1 = 2;
```

```
int motor2 = 3;
```

```
int motor3 = 4;
```

```
int motor4 = 5;
```



```
int trig = 7;
int echo = 6;
int durattion;
int distance;
const int ENA = 2;
const int ENB = 3;

void setup(){
    motor.attach(8);
    Serial.begin(9600);

    pinMode(motor1,OUTPUT);
    pinMode(motor2,OUTPUT);
    pinMode(motor3,OUTPUT);
    pinMode(motor4,OUTPUT);
    pinMode(echo,INPUT);
    pinMode(trig,OUTPUT);
    analogWrite(ENA, 100);
    analogWrite(ENB, 100);
}

void loop(){

    digitalWrite(trig.LOW);
    delayMicroseconds(2);
    digitalWrite(trig.HIGH);
    delayMicroseconds(10);
```

```
digitalWrite(trig,LOW);  
duration= pulseIn(echo,HIGH);  
distance=(duration/2)*0.0324;  
Serial.print("distance= ");  
Serial.print(distance);  
Serial.print("cm ");
```

```
if (distance>20){  
    digitalWrite(motor1,HIGH);  
    digitalWrite(motor2,LOW);  
    digitalWrite(motor3,HIGH);  
    digitalWrite(motor4,LOW);  
}  
else if (distance<=20){  
    digitalWrite(motor1,LOW);  
    digitalWrite(motor2,LOW);  
    digitalWrite(motor3,LOW);  
    digitalWrite(motor4,LOW);
```

```
delay(1000);  
motor.write(0);  
delay(1000);  
motor.write(90);  
delay(1000);  
motor.write(180);  
delay(1000);
```

```
motor.write(90);
```

```
delay(1000);
```

```
digitalWrite(motor1,LOW);
```

```
digitalWrite(motor2,HIGH);
```

```
digitalWrite(motor3,LOW);
```

```
digitalWrite(motor4,HIGH);
```

```
delay(1500);
```

```
digitalWrite(motor1,LOW);
```

```
digitalWrite(motor2,HIGH);
```

```
digitalWrite(motor3,HIGH);
```

```
digitalWrite(motor4,LOW);
```

```
delay(1500);
```

```
digitalWrite(motor1,HIGH);
```

```
digitalWrite(motor2,LOW);
```

```
digitalWrite(motor3,HIGH);
```

```
digitalWrite(motor4,LOW);
```

```
}
```

```
}
```

<https://create.arduino.cc/projecthub/Techatronic/obstacle-avoiding-robot-using-arduino-873332>

[https://www.youtube.com/watch?v=d\\_eMlOelpD0&t=519s](https://www.youtube.com/watch?v=d_eMlOelpD0&t=519s)

<https://www.youtube.com/watch?v=hHlIWpZcYNs>

**the end**