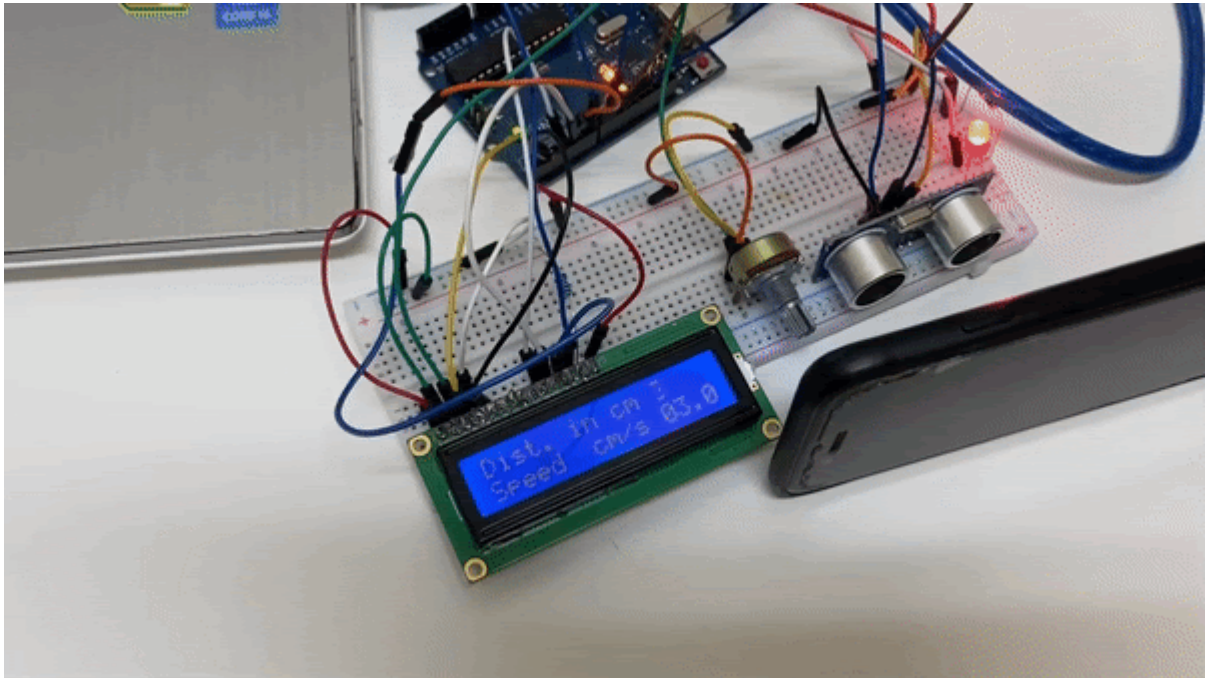


قياس السرعة باستخدام الاردوينو وحساس الموجات الصوتية

مقدمة

يمكن أن يكون تحديد سرعة حركة الأجسام مفيداً وضرورياً في العديد من التطبيقات، ويمكن استخدام وسائل متنوعة لتحديد السرعة، في هذا الدرس سنتعلم طريقة قياس السرعة والمسافة باستخدام الاردوينو وحساس الموجات فوق الصوتية.



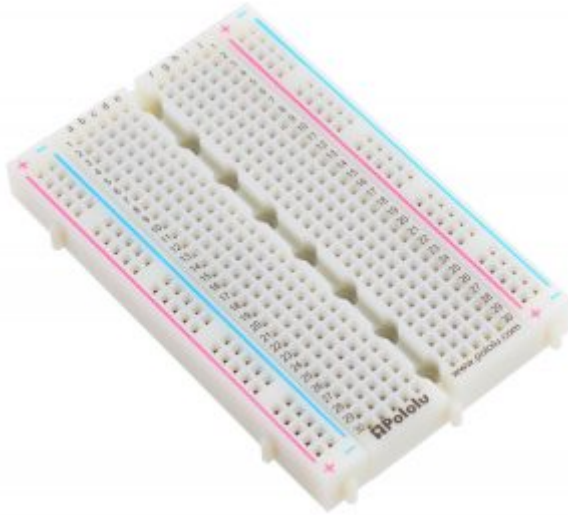
المواد والأدوات



×1 اردوينو اونو



×1 سلك الاردوينو



×1 لوحة تجارب - حجم كبير



×1 شاشة كرسالية (LCD 2×16)



×1 مقاومة متغيرة



حزمة أسلاك توصيل (ذكر - ذكر)



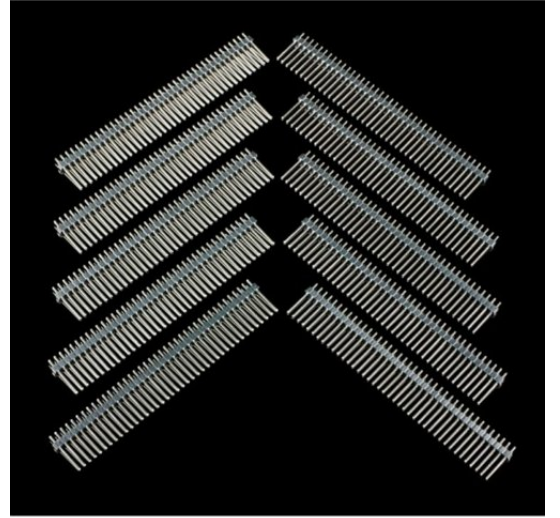
×1 حساس المسافة (HC-SR04)



×1 مقاومة 220 Ω



×1 ثنائي مشع للضوء أحمر (LED)



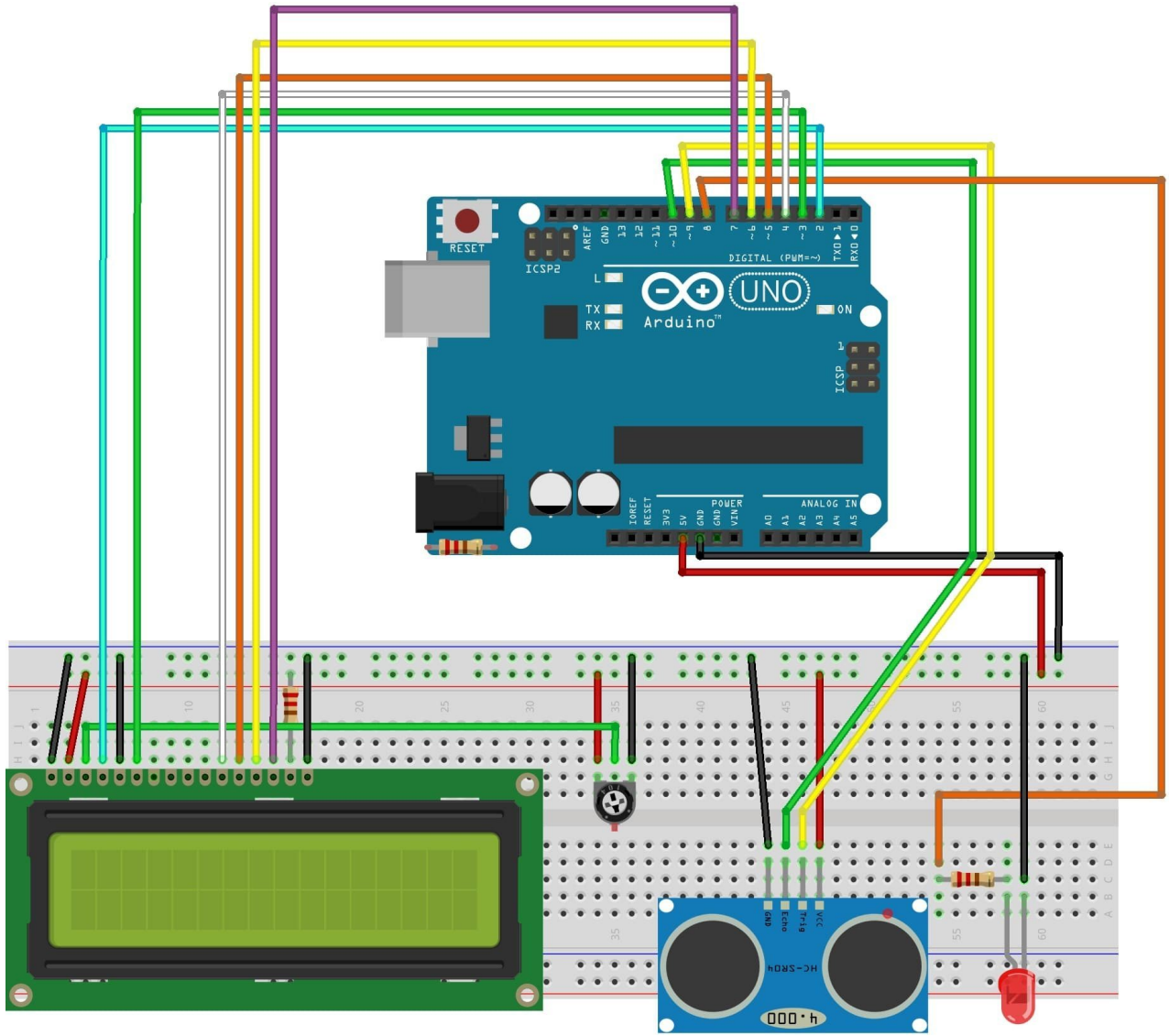
×1 رأس دبوس 40

توصيل الدائرة

للمزيد حول حساس الموجات الفوق صوتية يمكنك الرجوع للدرس التالي حساس الموجات فوق الصوتية.

لمعرفة المزيد حول الشاشة الكرسطالية يمكنك الرجوع للدرس التحكم بالشاشة الكرسطالية LCD

لا بد من تلحيم المنافذ مع الشاشة الكرسطالية، للمزيد حول اللحام يمكنك الرجوع للدرس تعلم كيفية التلحيم – تلحيم القطع باللوحة الإلكترونية



الكود البرمجي

ارفع الكود البرمجي التالي على لوحة الاردوينو باستخدام برنامج اردوينو (IDE).

```
#include <LiquidCrystal.h>
const int RS = 2, EN = 3, D4 = 4, D5 = 5, D6 = 6, D7 = 7;
LiquidCrystal lcd(RS,EN,D4,D5,D6,D7);

// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;

// defines variables
long duration;
int distance1=0;
int distance2=0;
double Speed=0;
int distance=0;

void setup()
```

```

{
lcd.begin(16, 2); // LCD 16X2
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
pinMode(8 , OUTPUT);
Serial.begin(9600); // Starts the serial communication
}

void loop()
{
//calculating Speed
distance1 = ultrasonicRead(); //calls ultrasonicRead() function below

delay(1000); //giving a time gap of 1 sec

distance2 = ultrasonicRead(); //calls ultrasonicRead() function below

//formula change in distance divided by change in time
Speed = (distance2 - distance1)/1.0; //as the time gap is 1 sec we divide it by 1.

//Displaying Speed
Serial.print("Speed in cm/s :");
Serial.println(Speed);
lcd.setCursor(0,1);
if (Speed<0)
{
lcd.print("Speed cm/s ");
lcd.print("0");
}
if (Speed>0)
{
lcd.print("Speed cm/s ");
lcd.print(Speed);
}
// LED indicator
if (distance >0 && distance <5)
{
digitalWrite( 8 , HIGH);
delay(50); // waits for a second
}
if (distance > 5 && distance <10 )
{
digitalWrite( 8 , HIGH);
delay(50); // waits for a second
digitalWrite( 8 , LOW); // sets the LED off
delay(50); // waits for a second
}
if (distance >10 && distance < 20)
{
digitalWrite( 8 , HIGH);
delay(210); // waits for a second
digitalWrite( 8 , LOW); // sets the LED off
}
}

```



```

delay(210); // waits for a second
}
if (distance >20 && distance < 35)
{
digitalWrite( 8 , HIGH);
delay(610); // waits for a second
digitalWrite( 8 , LOW); // sets the LED off
delay(610); // waits for a second
}
}
float ultrasonicRead ()
{
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
//calculating distance
distance= duration*0.0340 / 2;
// Prints the distance on the Serial Monitor
Serial.print("Distance in cm : ");
Serial.println(distance);
lcd.setCursor(0,0);
lcd.print("Dist. in cm ");
lcd.print(distance);
lcd.print(" ");
return distance;
}

```

شرح الكود البرمجي

هذا السطر يستدعي مكتبة الشاشة الكرسطالية.

هذا السطر يستدعي مكتبة الشاشة الكرسطالية.

نستطيع تحميلها بتتبع المسار التالي:

Sketch > Include libraries > Manage libraries

ثم نكتب بخانة البحث Liquid crystal by Arduino

ثم نضغط على Install.

```
#include <LiquidCrystal.h>
```

بعد ذلك أعلننا عن المتغيرات اللازمة مثل المتغيرات الخاصة بالشاشة الكرسطالية وتم توضيح المداخل التي استخدمها في لوحة الاردوينو لربط الشاشة الكرسطالية.

```
const int RS = 2, EN = 3, D4 = 4, D5 = 5, D6 = 6, D7 = 7;
LiquidCrystal lcd(RS,EN,D4,D5,D6,D7);
```

هذه الأسطر توضح منافذ الـ Arduino التي ستستخدمها لربط حساس المسافة في هذا المشروع المنفذ رقم 9 مع triqPin في الحساس والمنفذ رقم 10 مع echoPin.

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
```

هنا يتم تهيئة المتغيرات التي سيتم تخزين فيها قيم المسافة والسرعة وسيتم وضع القيمة الابتدائية لكل من المسافة والسرعة=0.

```
// defines variables
long duration;
int distance1=0;
int distance2=0;
double Speed=0;
int distance=0;
```

في الدالة setup() يتم تهيئة الشاشة الكرسطالية وحساس المسافة وتهيئة المنفذ 8 مع الثنائي المشع للضوء الأحمر.

```
void setup()
{
  lcd.begin(16, 2); // LCD 16X2
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  pinMode( 8 , OUTPUT);
  Serial.begin(9600); // Starts the serial communication
}
```

في الدالة loop() يتم قياس السرعة والمسافة بشكل مستمر وبعد كل قراءة سيتم طباعة القيم المقروءة من الحساس على الشاشة الكرسطالية .

إذا كانت المسافة:

أكبر من الصفر وأصغر من 5 فسيعمل الثنائي المشع للضوء.

وإذا كانت المسافة أكبر من 5 وأصغر من 10 أو أكبر من 10 وأصغر من 20 أو كانت أكبر من 20 وأصغر من 35 فسيعمل الثنائي المشع أقل من ثانية ثم ينطفئ لمدة أقل من ثانية وهكذا.

```
void loop()
{
  //calculating Speed
  distance1 = ultrasonicRead(); //calls ultrasonicRead() function below

  delay(1000); //giving a time gap of 1 sec

  distance2 = ultrasonicRead(); //calls ultrasonicRead() function below

  //formula change in distance divided by change in time
```



```
// Reads the echoPin, returns the sound wave travel time in microseconds
duration = pulseIn(echoPin, HIGH);
//calculating distance
distance= duration*0.0340 / 2;
// Prints the distance on the Serial Monitor
Serial.print("Distance in cm : ");
Serial.println(distance);
lcd.setCursor(0,0);
lcd.print("Dist. in cm ");
lcd.print(distance);
lcd.print(" ");
return distance;
}
```

يمكنك اختبار نظام قياس السرعة والمسافة بعد رفع الكود البرمجي على لوحة الاردوينو.

لا تنسَ فصل مصدر الطاقة بعد الانتهاء من استخدام النظام.