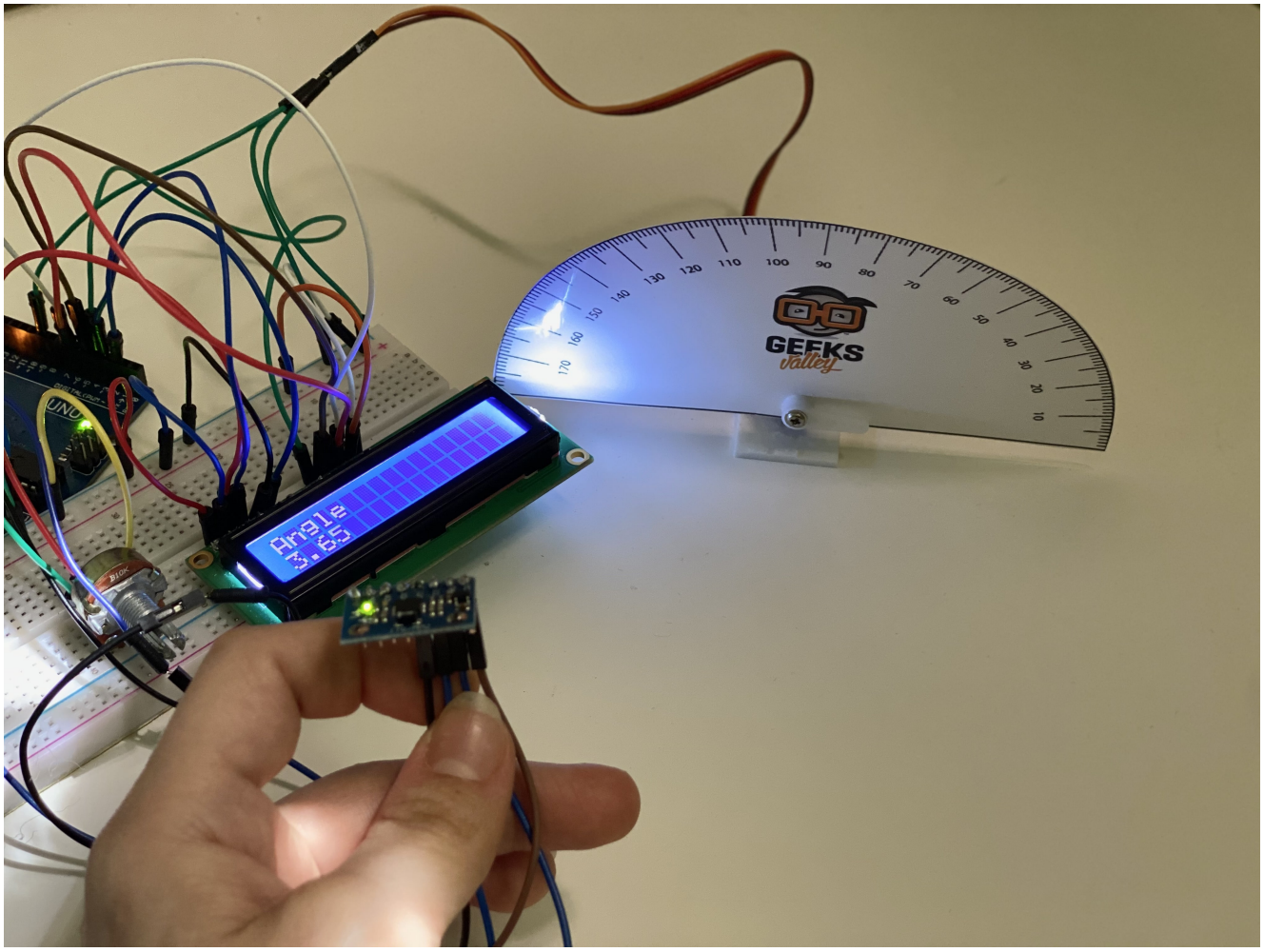
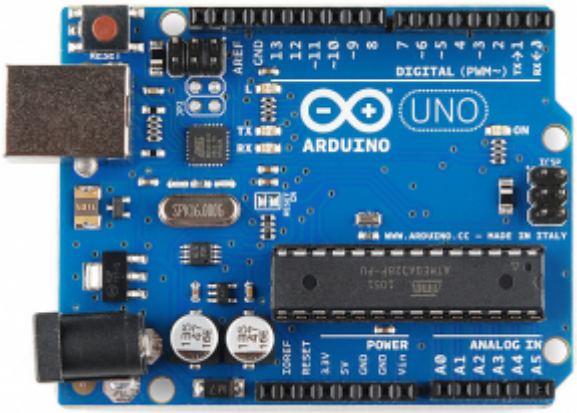
صناعة منقلة رقمية باستخدام حساس التسارع والاردوينو

مقدمة

المنقلة الرقمية واحدة من الأدوات المستخدمة لقياس الزوايا، ولها دورًا مهمًا في عملية الصناعة فكلما كانت أدوات القياس دقيقة كل ما كان المخرج النهائي أكثر دقة، في هذا الدرس ستتعلم كيفية صناعة منقلة رقمية باستخدام والاردوينو حساس الحركة والتسارع في ثلاث اتجاهات .



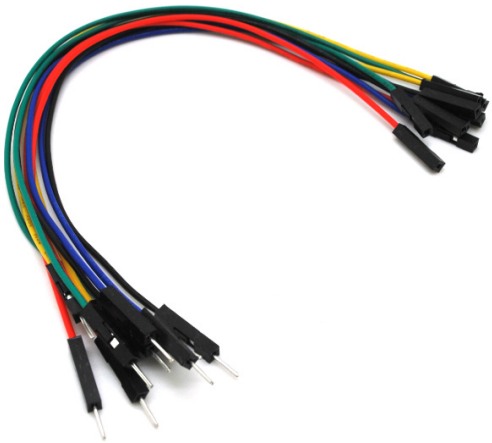
الأدوات والمواد



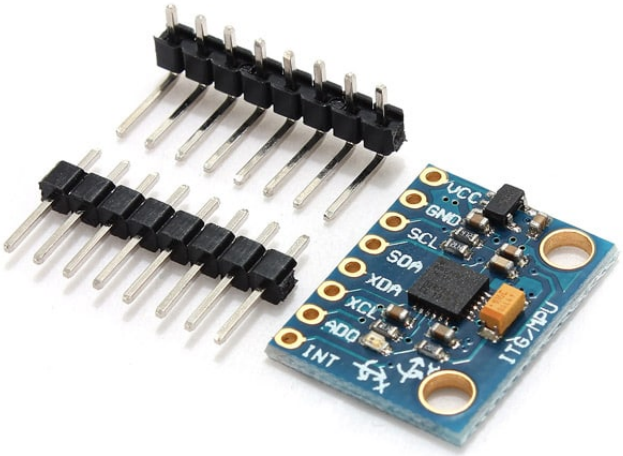
1× اردوينو اونو

1× سلك الاردوينو

حزمة أسلاك توصيل (ذكر- ذكر)



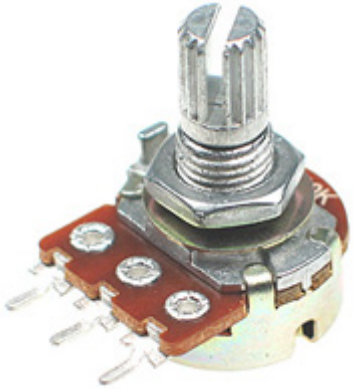
حزمة أسلاك توصيل (ذكر – أنثى)



1× حساس الحركة والتسارع في ثلاث اتجاهات

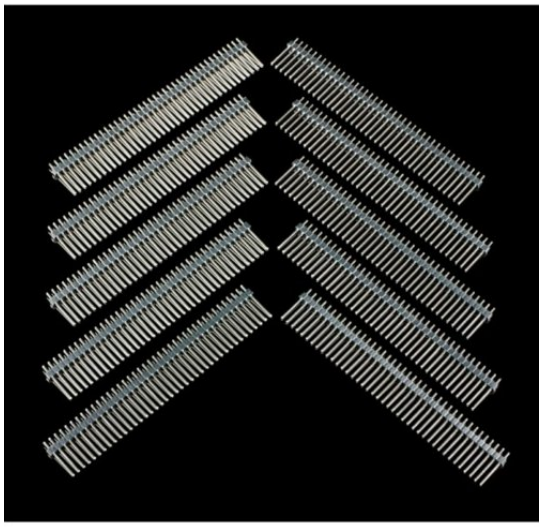
1× لوحة تجارب – حجم كبير

1× شاشة كرستالية



1× مقاومة متغيرة

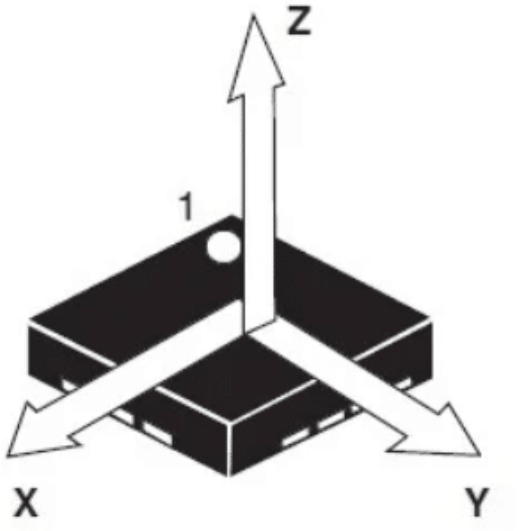
1× محرك سيرفو



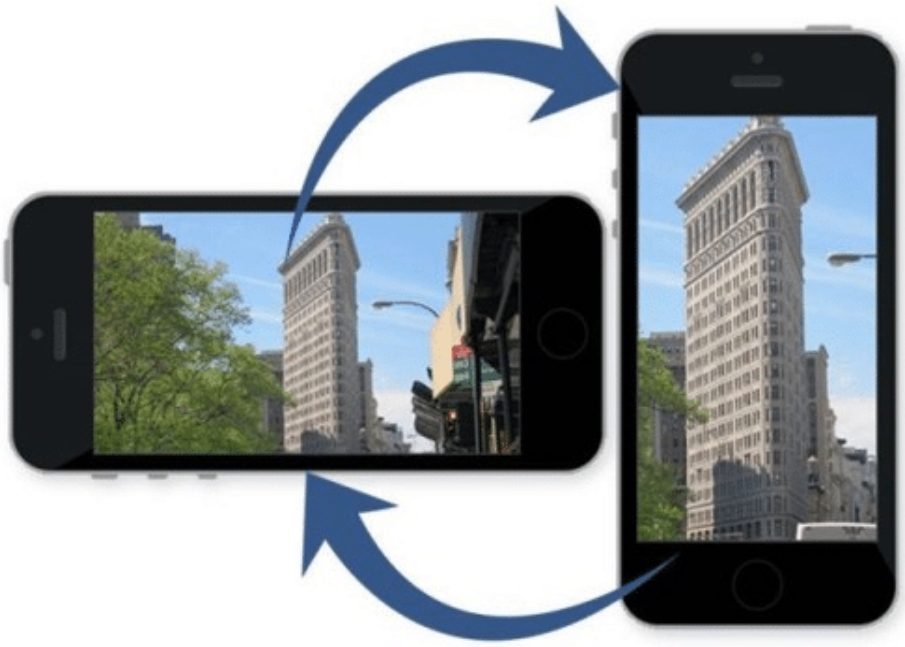
1× 40 رأس دبوس

حساس الحركة والتسارع في ثلاث اتجاهات

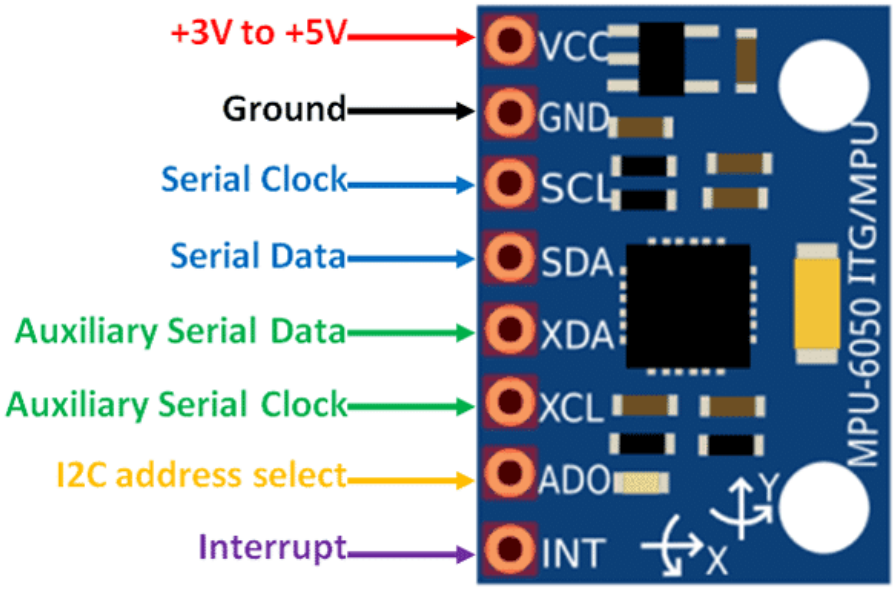
حساس التسارع والحركة مصمم لقياس أبسط وأدق التغيرات في معدل الحركة في ثلاث اتجاهات.



يستخدم حساس التسارع والحركة في مجالات كثيرة، أبسط مثال هو شاشة هاتفك المحمول عند إمالة الهاتف يتغير اتجاه الشاشة إما بشكل عرضي أو بشكل طولي.



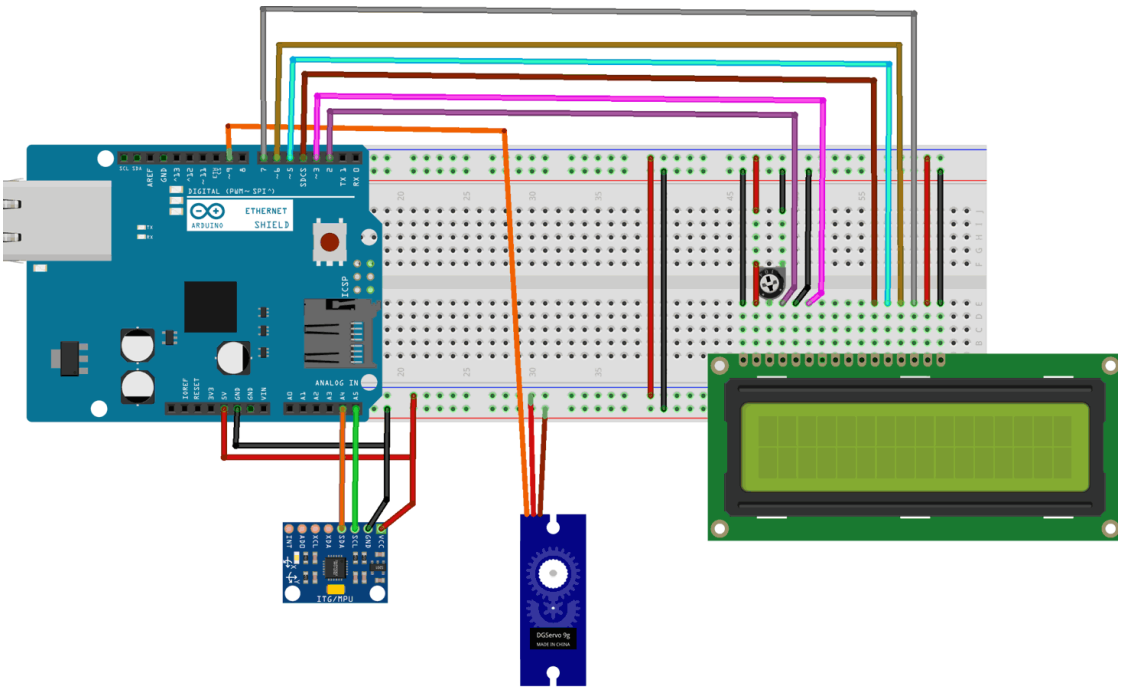
يحتوي الحساس على 8 مداخل وهي موضحة بالشكل التالي:



توصيل الدائرة

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

لابد من تلحيم المنافذ مع الشاشة الكرستالية، للمزيد حول اللحام يمكنك الرجوع للدرس تعلم كيفية التلحيم – تلحيم القطع باللوحة الإلكترونية



الكود البرمجي

```
#include <Servo.h>                  //Include Servo Motor library for using Servo
#include <LiquidCrystal.h>          //Include LCD library for using LCD
#include <Wire.h>                   //Include WIre library for using I2C
LiquidCrystal lcd(2,3,4,5,6,7);   //Define LCD display pins RS,E,D4,D5,D6,D7
const int MPU_addr=0x68;          //I2C MPU6050 Address
Servo myservo;                   //myservo object for class servo
int16_t axis_X,axis_Y,axis_Z;
int minVal=265;
int maxVal=402;
double x;
double y;
double z;
int pos = 0;
void setup()
{
  Wire.begin();                          //Begins I2C communication
  Wire.beginTransmission(MPU_addr);     //Begins Transmission with MPU6050
  Wire.write(0x6B);                      //Puts MPU6050 in Sleep Mode
  Wire.write(0);                         //Puts MPU6050 in power mode
  Wire.endTransmission(true);           //Ends Trasmission
  myservo.attach(9);                   //Servo PWM pin as 9 in UNO
  lcd.begin(16,2);                     //Sets LCD in 16X2 Mode
}
void loop()
{
  Wire.beginTransmission(MPU_addr); //Begins I2C transmission
  Wire.write(0x3B);                  //Start with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true); //Request 14 Registers from MPU6050
  axis_X=Wire.read()<<8|Wire.read(); //Obtain 0x3B (ACCEL_XOUT_H) & 0x3C
```

```
(ACCEL_XOUT_L)
  axis_Y=Wire.read()<<8|Wire.read(); //0x3B (ACCEL_YOUT_H) & 0x3C (ACCEL_YOUT_L)
  axis_Z=Wire.read()<<8|Wire.read(); //0x3B (ACCEL_ZOUT_H) & 0x3C (ACCEL_ZOUT_L)
    int xAng = map(axis_X,minVal,maxVal,-90,90);
    int yAng = map(axis_Y,minVal,maxVal,-90,90);
    int zAng = map(axis_Z,minVal,maxVal,-90,90);
   x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);     //Formula to calculate x values in
degree
     int pos = map(x,0,180,0,180); // As X value is from 0 to 360 deg
     myservo.write(pos);           // Write angle obtained 0 to 180 to servo
     lcd.setCursor(0,0);
     lcd.print("Angle");
     lcd.setCursor(0,1);
     lcd.print(x);
     delay(500);
     lcd.clear();
}
```

شرح الكود البرمجي

هنا نستدعي مكتبة الشاشة الكرستالية ومكتبة Wire.h التي تحتوى على الدوال اللازمة للتواصل بين الاردوينو وحساس التسارع ومكتبة محرك السيرفو.

نستطيع تحميل مكتبة الشاشة الكرستالية بتتبع المسار التالي:

Sketch > Include libraries > Manage libraries

ثم نكتب بخانة البحث Liquid crystal by Arduino

ثم نضغط على Install.

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

بعد ذلك أعلنا عن المتغيرات اللازمة مثل المتغيرات الخاصة بالشاشة الكرستالية.

```
LiquidCrystal lcd(2,3,4,5,6,7);
```

بعد ذلك يتم تحديد بروتوكول الاتصال التسلسلي I2c الخاص بحساس التسارع.

```
const int MPU_addr=0x68;
```

هذا السطر يعرف myservo الخاص بمحرك السيرفو ثم يتم تعريف القيم الخاصة بحساس التسارع على ثلاثة محاور x و y و z.

```
Servo myservo;
int16_t axis_X,axis_Y,axis_Z;
```

يتم تحدد أقصى قيمة وأقل قيمة يقرأها الحساس بين (265–402) لقياس الزوايا من 0 إلى 360.

```
int minVal=265;
int maxVal=402;
```

في دالة ()void يبدأ الاتصال التسلسلي والنقل بين حساس التسارع والعنوان 0x68.

```
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
```

في هذا السطر يدخل حساس التسارع في وضع السكون بعد توصيله مع العنوان 0x6B, يستأنف العمل بعد ادخال القيمة صفر.

```
Wire.write(0x6B);
Wire.write(0);
```

بعد تفعيل حساس التسارع انهِ النقل.

```
Wire.endTransmission(true);
```

محرك السيرفو تم ربطه مسبقًا مع المدخل الرقمي 9 على لوحة الاردوينو.

```
  myservo.attach(9);
```

في دالة ()loop يستأنف الاتصال من جديد.

```
 Wire.beginTransmission(MPU_addr);
```

ويبدأ مع المسجل (ACCEL_XOUT_H) 0x3B

```
Wire.write(0x3B);
```

يستأنف الاتصال من جديد لكن بإضافة False لكن الاتصال هنا مفعل.

```
Wire.endTransmission(false);
```

هذا السطر يطلب بيانات حساس التسارع (x و y و z) من المسجل 14.

```
Wire.requestFrom(MPU_addr,14,true);
```

بعد أخذ البيانات المسجلة لكلًا من (x و y و z) يتم تخزينها في axis_X,axis_Y,axis_Z.

```
 axis_X=Wire.read()<<8|Wire.read();
  axis_Y=Wire.read()<<8|Wire.read();
  axis_Z=Wire.read()<<8|Wire.read();
```

اجعل القيم لكل الثلاث محاور (x و y و z) بين (265–402) يتم تمثيلها كـ90 و ـ90

```
 int xAng = map(axis_X,minVal,maxVal,-90,90);
    int yAng = map(axis_Y,minVal,maxVal,-90,90);
```

```
    int zAng = map(axis_Z,minVal,maxVal,-90,90);
```

هنا قيمة x لحساب الزاوية من 0 – 360 نقوم بتحويل فقط قيمة x لأن دوران محرك السيرفو يعتمد عليها.

```
x= RAD_TO_DEG * (atan2(-yAng, -zAng)+PI);
```

قيمة الزاوية x التي بين 0 ـ360 يتم تحويلها لـ0–180.

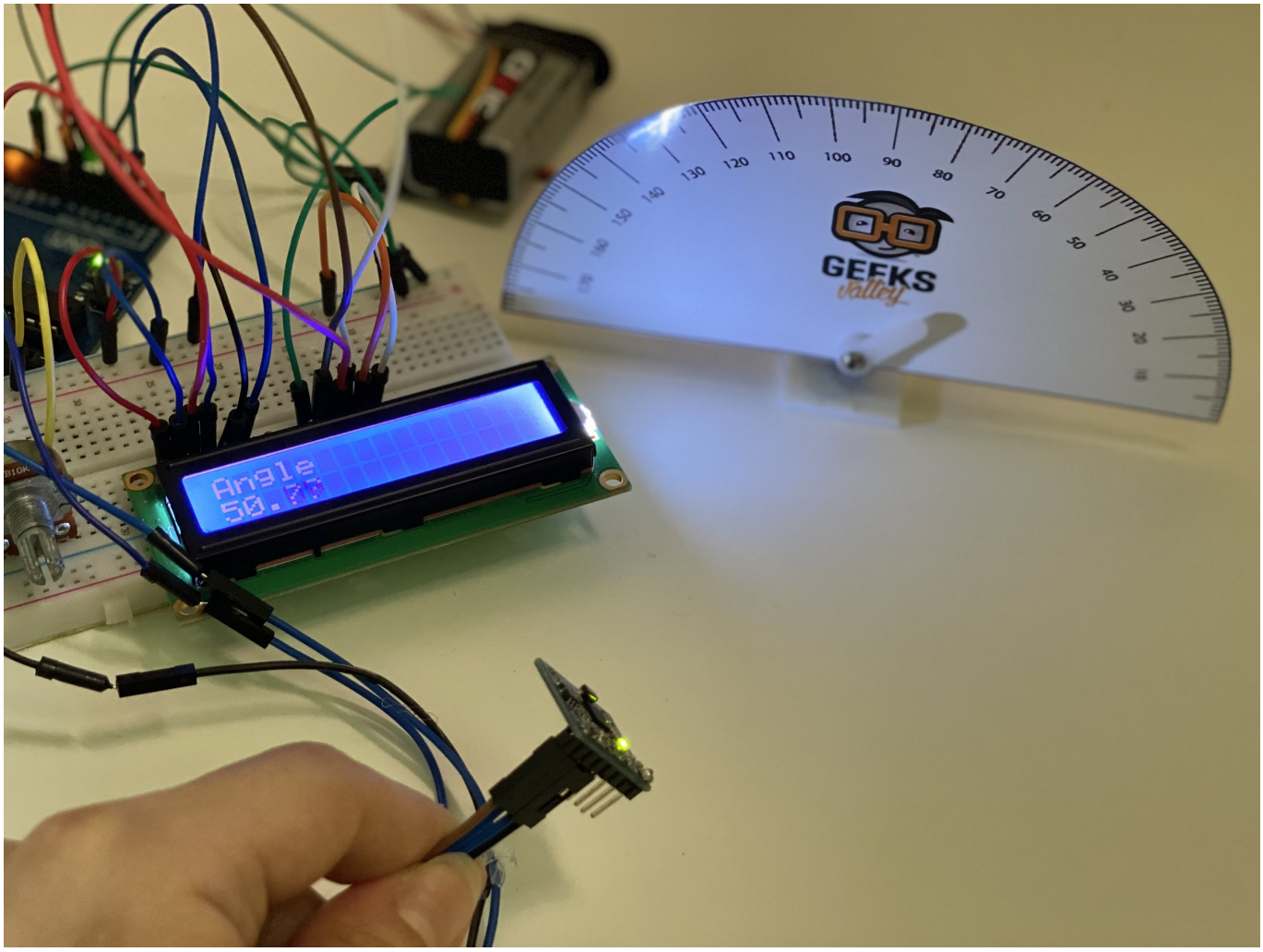
```
 int pos = map(x,0,180,0,180);
```

هذا السطر يأخذ قيمة الزاوية من حساس التسارع وعلى أساس تلك القيم يدور محرك السيرفو ويتم طباعة قيمة الزاوية على الشاشة الكرستالية.

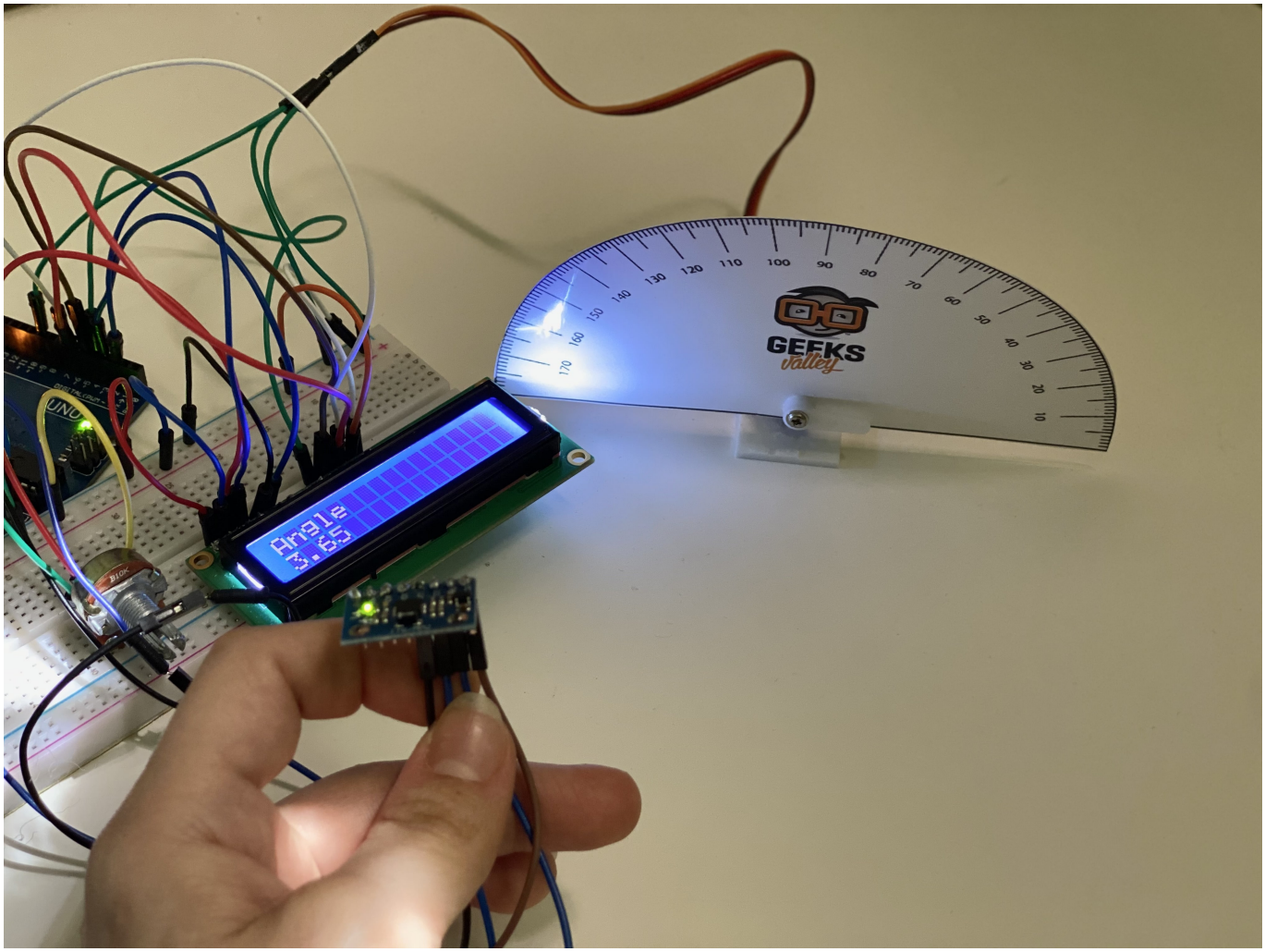
```
 myservo.write(pos);
lcd.setCursor(0,0);
lcd.print("Angle");
lcd.setCursor(0,1);
lcd.print(x);
delay(500);
lcd.clear();
```

بعد رفع الكود على لوحة الاردوينو ستكون هناك عدة قيم يتم تسجيلها من حساس التسارع والحركة وبناء على تلك القيم يتم دوران محرك السيرفو وطباعة قيمة الزاوية على الشاشة الكرستالية.

زاوية تقريبًا 50.



زاوية تقريبًا 3.


Angle
3.65
GEEKS
valley