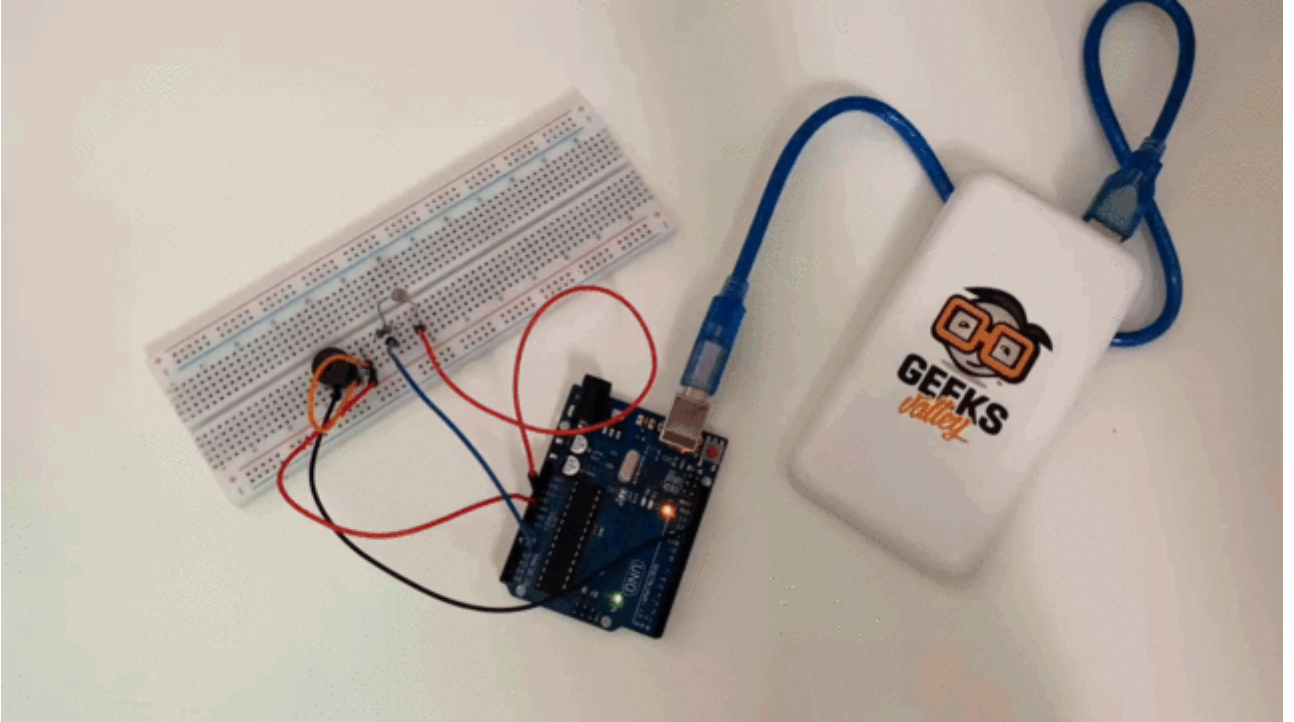


اردوينو – الدرس الحادي عشر – اصدار الأصوات (Piezo sounder)

مقدمة

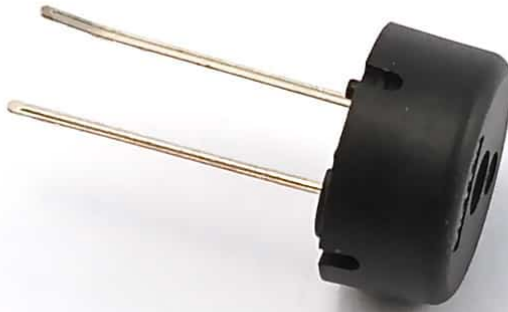
في هذا الدرس سنتعلم كيفية اصدار الأصوات بمتحكم الأردوينو.
أولاً ستقوم بتشغيل مقطع صوتي، ثم القيام بالتلاعب بمستوى الصوت عبر المستشعر الضوئي 'photocell'.



المواد والأدوات



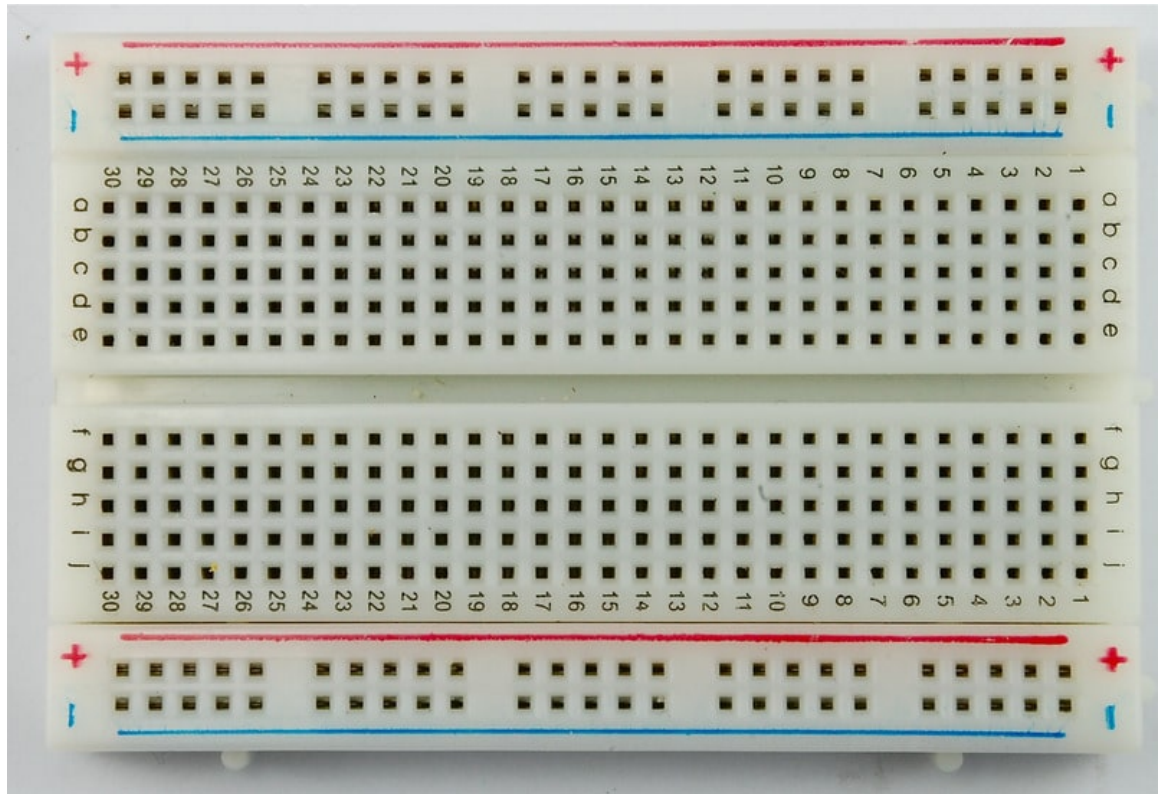
(k Ω Resistor 1) x1



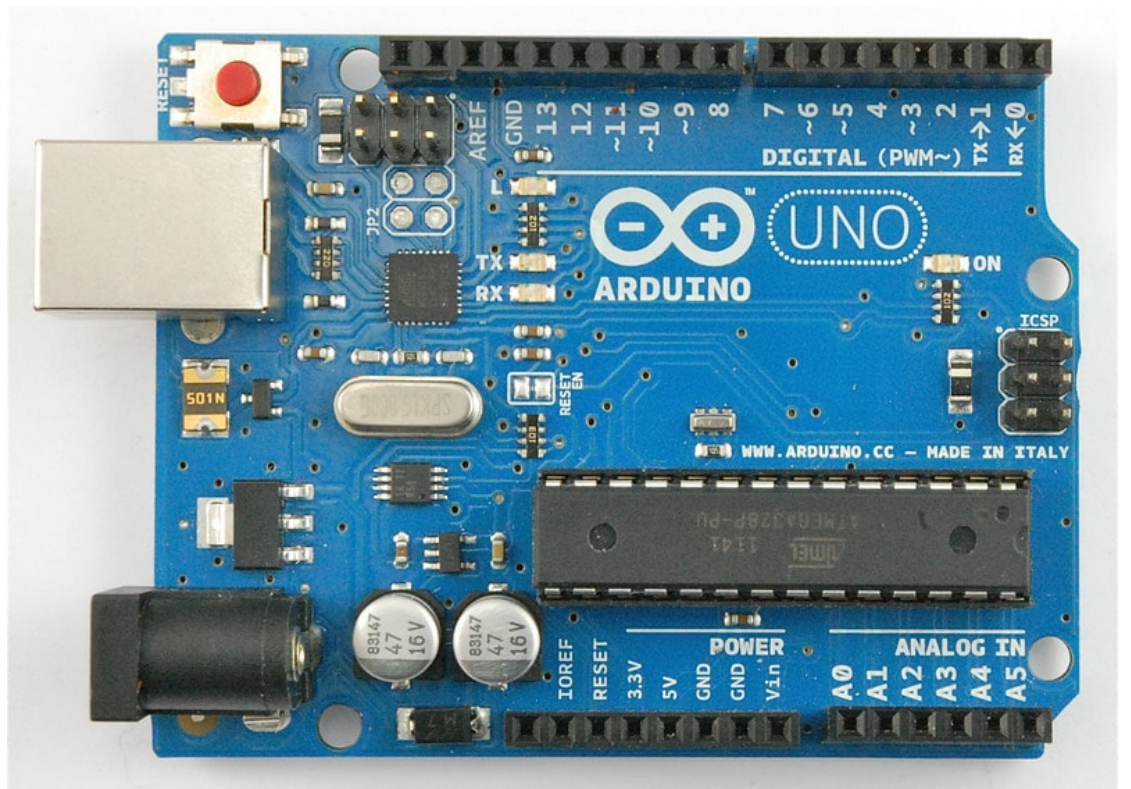
x1 سماعة البيزو (Piezo sounder)



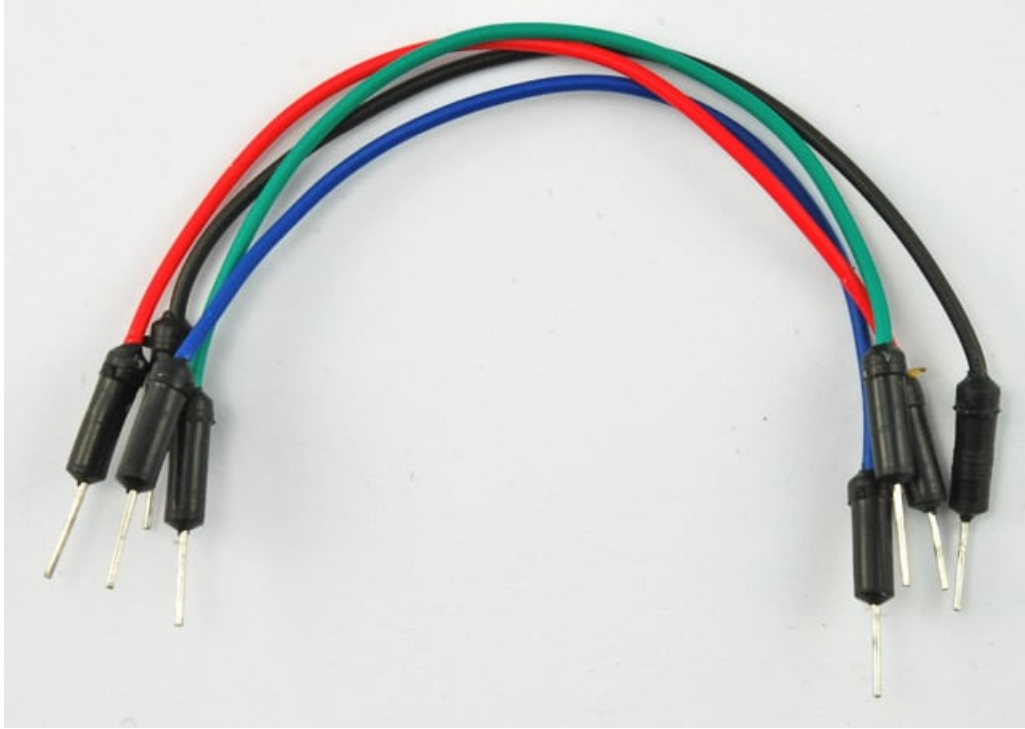
x1 المستشعر الضوئي (Photocell)



×1 لوحة التجارب (Half-size Breadboard)



×1 اردوينو اونو



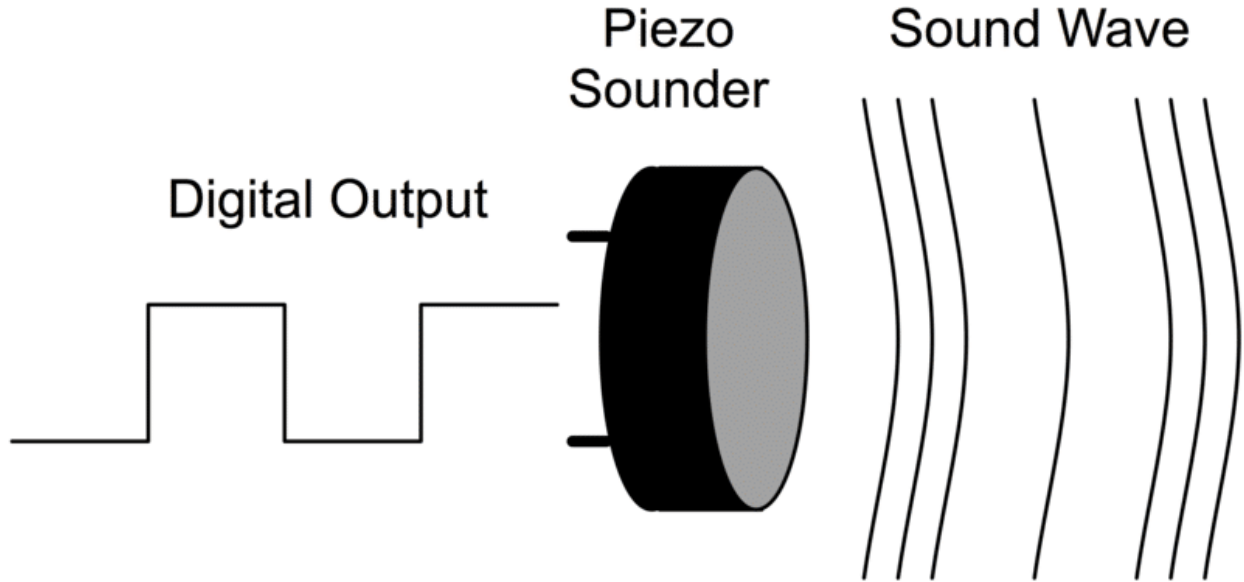
حزمة أسلاك توصيل (ذكر-ذكر)



1 × سلك اردوينو

إصدار الأصوات

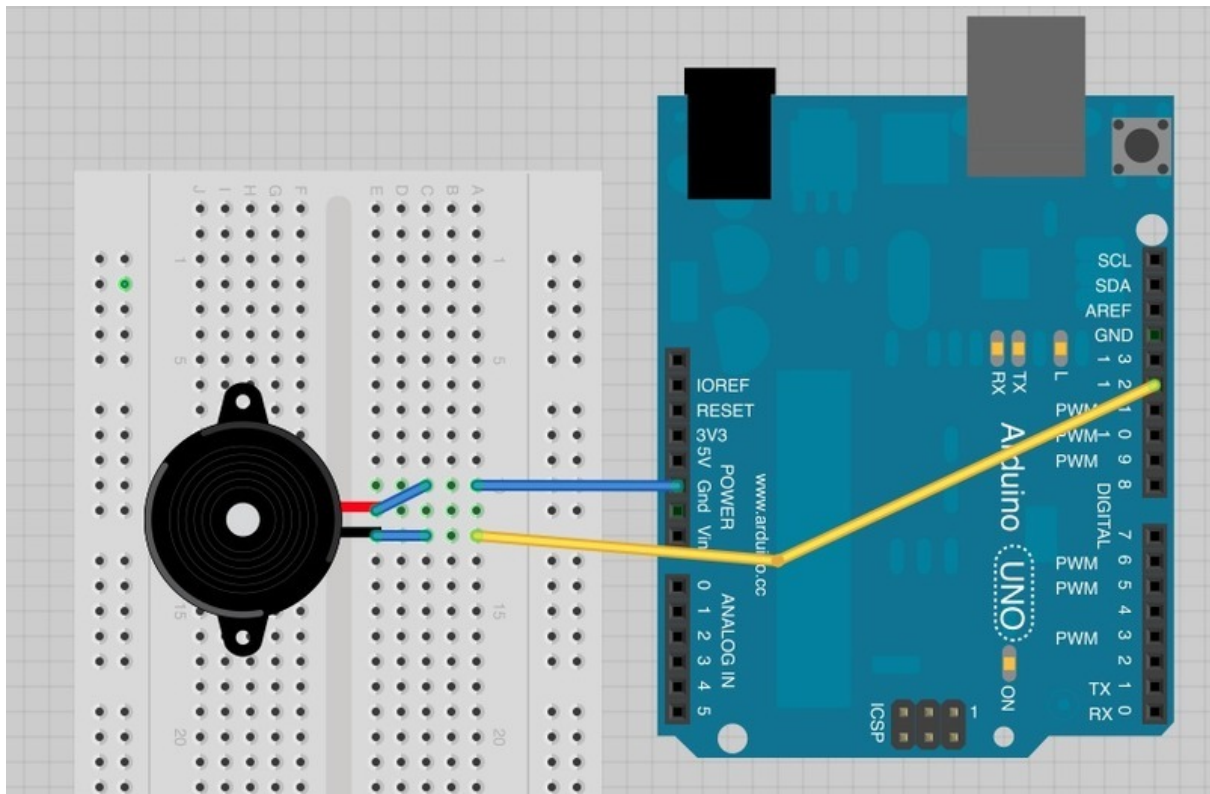
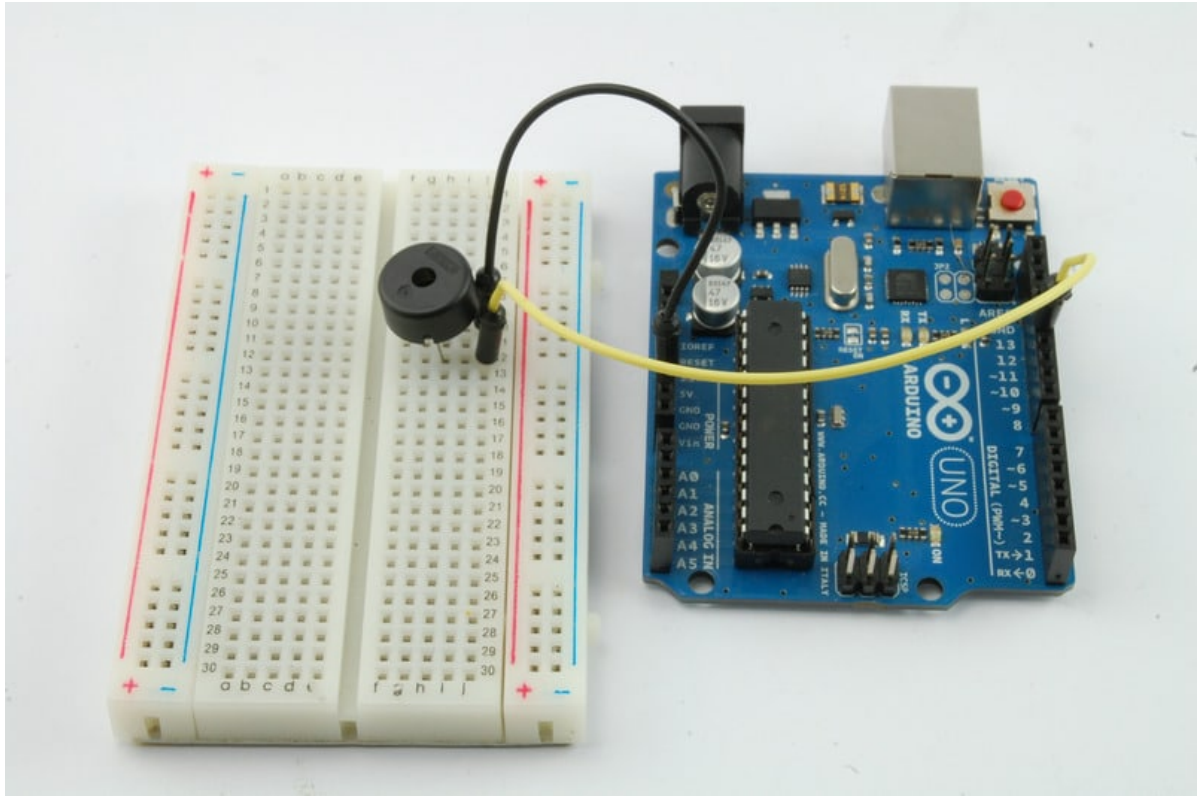
الصوت هو عبارة عن نبضة في ضغط الهواء سرعة النبضة (دورات بالثانية أو هيرتز) هي ما يصدر الصوت. كلما زادت قوة النبضة كلما زاد علو الصوت.



المتوسط C عادة يعرف بالتردد 261Hz. وكأنك قمت بتشغيل وإطفاء منفذ رقمي 'digital pin' لأكثر من 261 مرة بالثانية .
 لسماع النتيجة، عليك القيام بتركيب قطعه تقوم بترجمة الترددات الالكترونية إلى صوت هذا يمكن عن طريق تركيب سماعات كبيره او
 عبر استخدامك لسماعة البيزو 'piezo sounder'.
 سماعة piezo تستخدم نوع خاص من الكريستال تتمدد وتنكمش كتردد الكتروني والتي ينتج عنها الصوت.

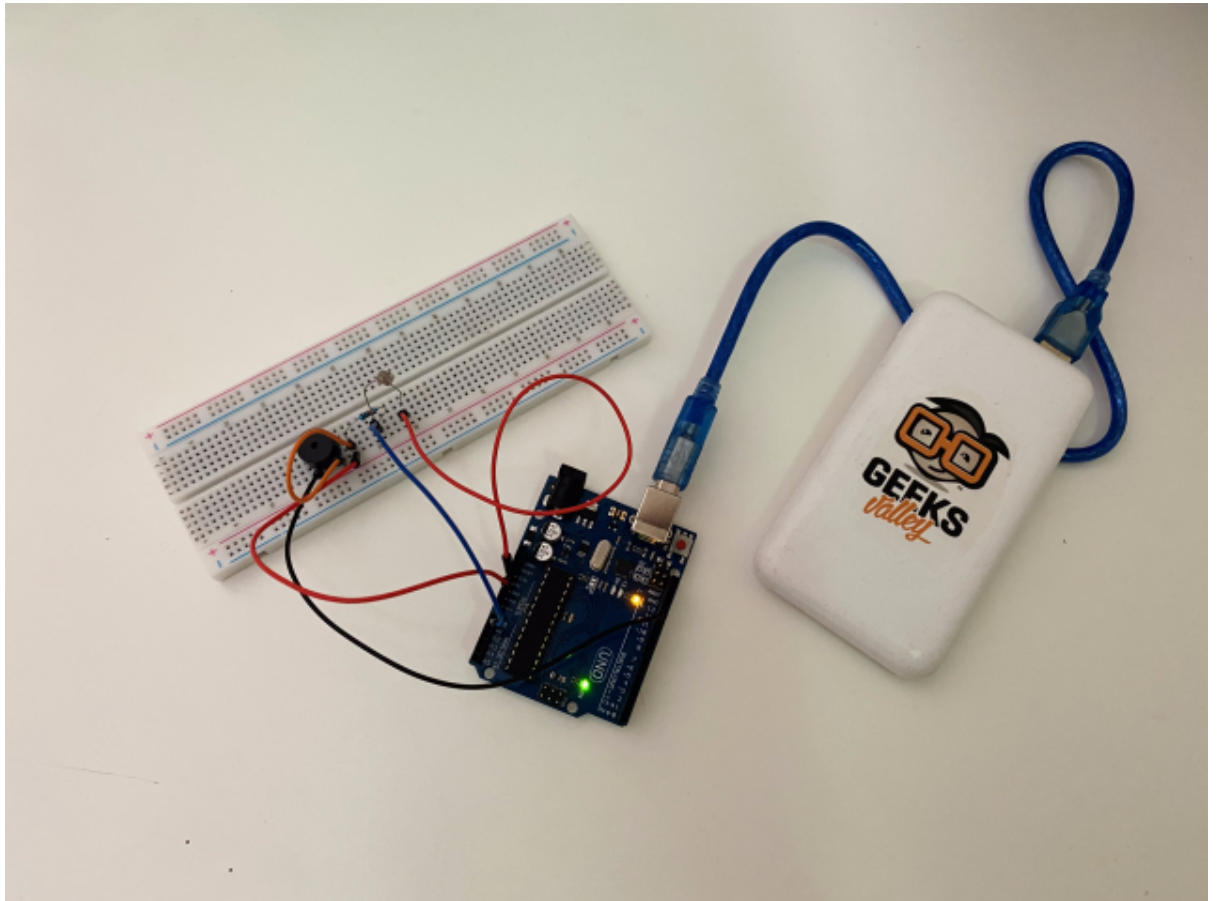
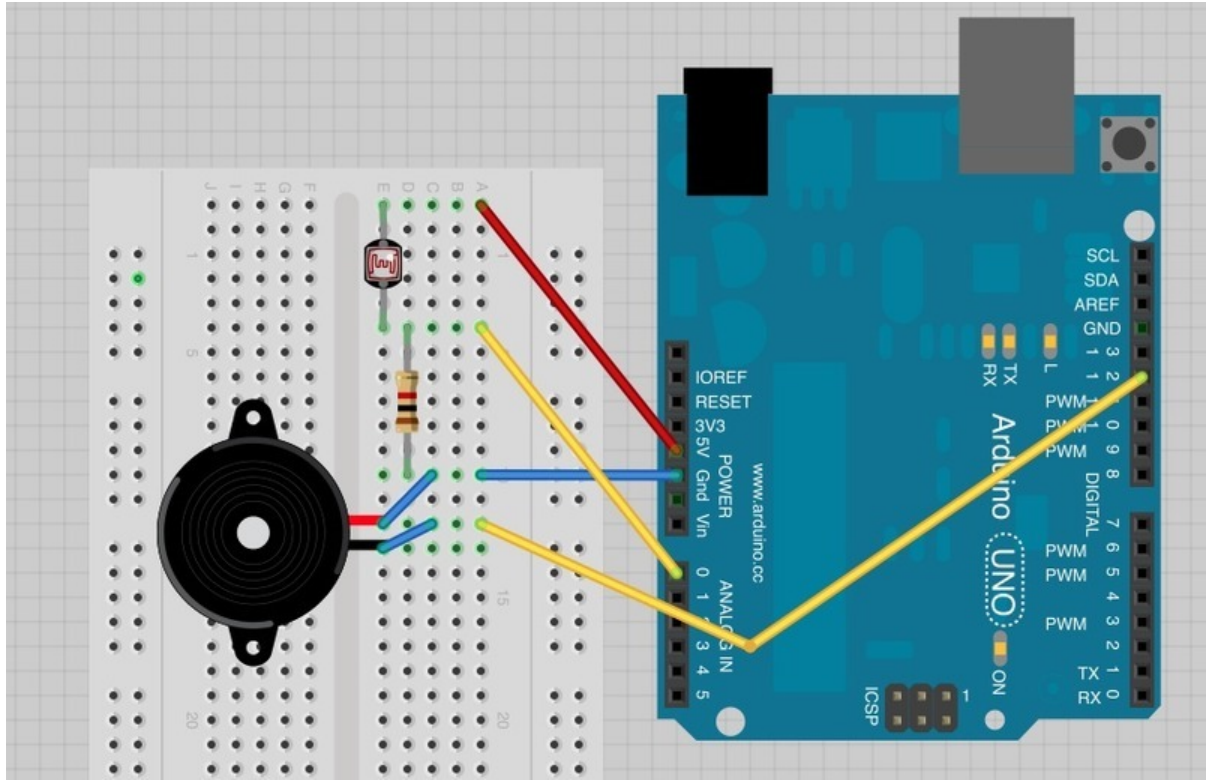
توصيل الدائرة

في هذا الجزء ستجد ان سماعة البيزو 'piezo buzzer' موضوعة على لوح التجارب.
 الرجل القصيرة موصولة بالمجال الأرضي GND والرجل الأخرى مربوطة بالمنفذ الرقمي 12 digital pin على الاردوينو.
 الدائرة رقم (1):



آلة Pseudo-Theremin الموسيقية الدائرة رقم (2):

يمكن اصدار الأصوات عند تمرير يدك امامها ،
 ستصنع آلة مشابهه في الوظيفة للتحكم والتأثير على مستوى الصوت بمجرد تمرير يدك فوق المستشعر الضوئي 'photocell'.
 اضعف المستشعر الضوئي 'photocell' و المقاوم resistor للوح التجارب.



الكود البرمجي

ارفع الكود البرمجي التالي للدائرة رقم (1) على متحكم الاردوينو:

```

int speakerPin = 12;

int numTones = 10;
int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//          mid C  C#   D    D#   E    F    F#   G    G#   A

void setup()
{
  for (int i = 0; i < numTones; i++)
  {
    tone(speakerPin, tones[i]);
    delay(500);
  }
  noTone(speakerPin);
}

void loop()
{
}

```

لتشغيل مقطع صوتي، حدد المجال الترددي. انظر للجزء التالي من الدرس المخصص للصوت . كل مجال ترددي لكل نوته صوتية تم الاحتفاظ به في مصفوفه 'array' . المصفوفه 'array' هي مثل القائمة، وبذلك يتم تشغيل المقطع عبر الانتقال لكل نوته صوتية بالقائمة.

في حلقة 'for' loop سيبدأ العد من 0 وحتى 9 باستخدام القيمة i . للحصول على المجال الترددي للنوته الصوتية لتشغيلها بكل خطوه نقوم باستخدام 'tones[i]' . هذا يعني ان القيمة الموجودة في مصفوفة 'tones' في موضع 'i' كمثال ، قيمة 'tones[0]' هي 261 ، وقيمة 'tones[1]' هي 277 .. وهكذا ..

الأمر 'tone' في الاردوينو يقوم بأخذ متغيرين اثنين، الأول هو المنفذ الذي يقوم باصدار الأصوات والثاني هي التردد الصوتي للمقطع لتشغيله.

عند الانتهاء من تشغيل من جميع النوتات الصوتية ، أمر 'noTone' يقوم بإيقاف اصدار الأصوات.

كان بإمكانك ان تضع الكود البرمجي لتشغيل الاصوات داخل دالة loop عوضاً عن دالة setup وذلك لأنها ستكرر المقطع الصوتي مره بعد مره دون توقف مما سيتسبب بالإزعاج لذلك تم وضعه داخل دالة setup لذلك دالة loop فارغة .

لإعادة تشغيل المقطع الصوتي كل ما عليك فعله هو الضغط على زر reset الموجود بمتحكم الاردوينو .

ارفع الكود للدائرة (2) على متحكم الاردوينو:

```

int speakerPin = 12;
int photocellPin = 0;

void setup()
{
}

void loop()
{
}

```



```
int reading = analogRead(photocellPin);  
int pitch = 200 + reading / 4;  
tone(speakerPin, pitch);  
}
```

الكود واضح، استخدم المنفذ التناظري 'analog pin' للقراءة من A0 لقياس الضوء. وستكون القيمة ما بين 0 و 700.

اضفنا القيمة 200 لجعل التردد 200Hz كأقل تردد، وببساطه يتم اضافة القراءة عبر قسمة القيمة على 4 ، لاصدار تردد ما بين 200Hz و 370Hz.

أنشطة أخرى

حاول تغيير القيمة 4 بالسطر التالي لخفض القيم المرتفعة:

```
int pitch = 200 + reading / 4;
```

تغيير القيمة سيرفع أو يقلل من التردد اعتماداً على الرقم الذي قمت بوضعه.

يمكنك التلاعب بقيم النوتات الصوتية لترى ماذا سيحدث.