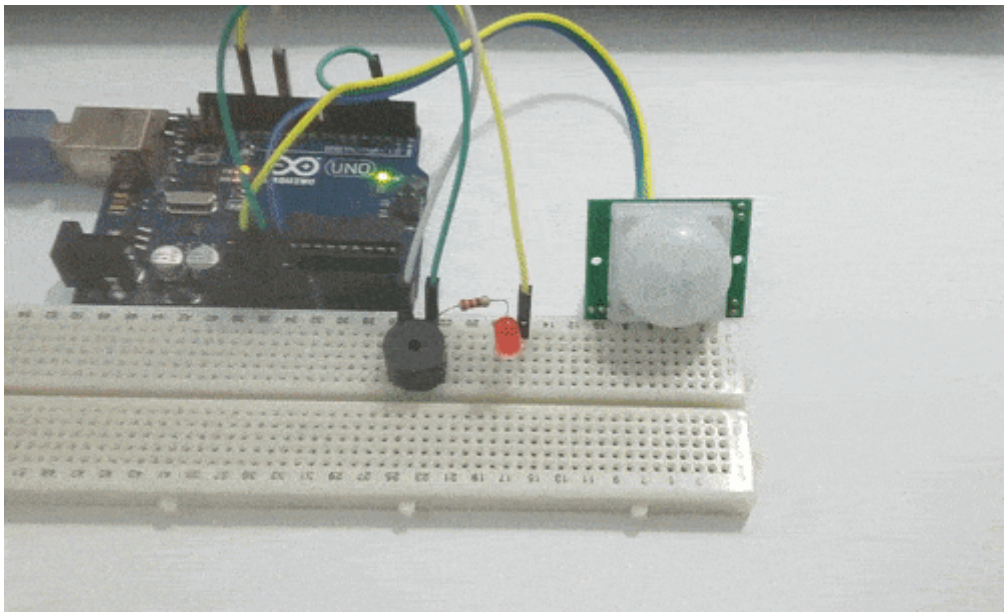
نظام الحماية ضد السرقة

في هذا المشروع سنتعلم كيفية استخدام حساس الحركة PIR Sensor مع الاردوينو لعمل نظام انذار ضد السرقة.



المكونات المطلوبة



Arduino Uno



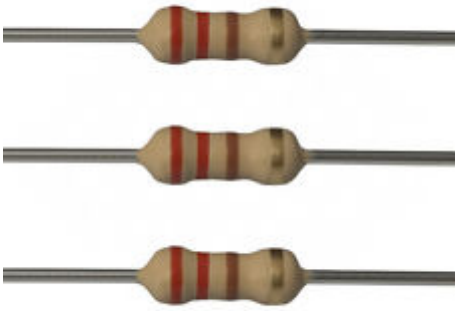
PIR Sensor



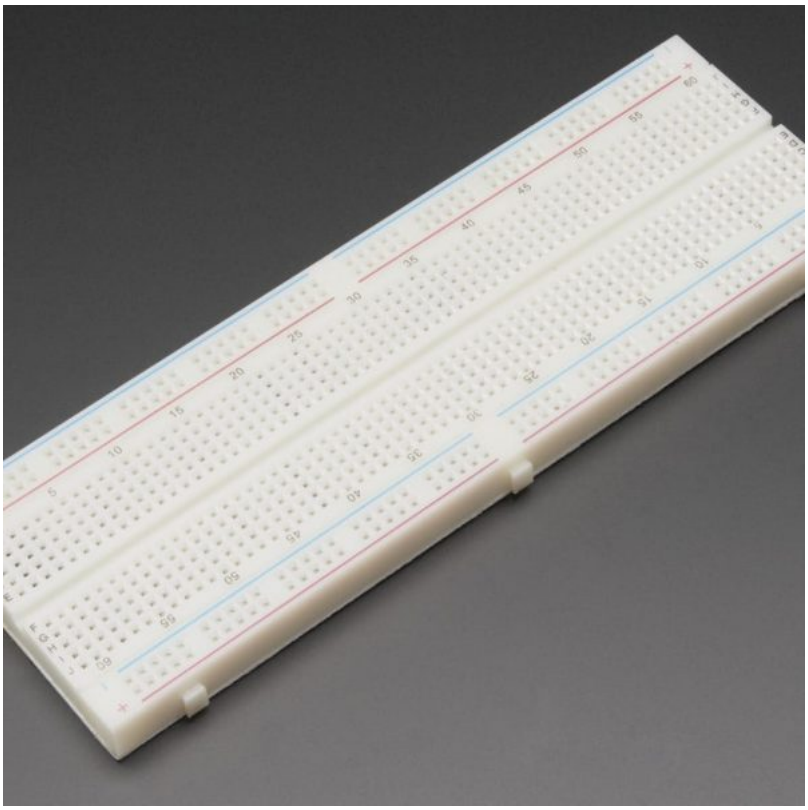
Buzzer



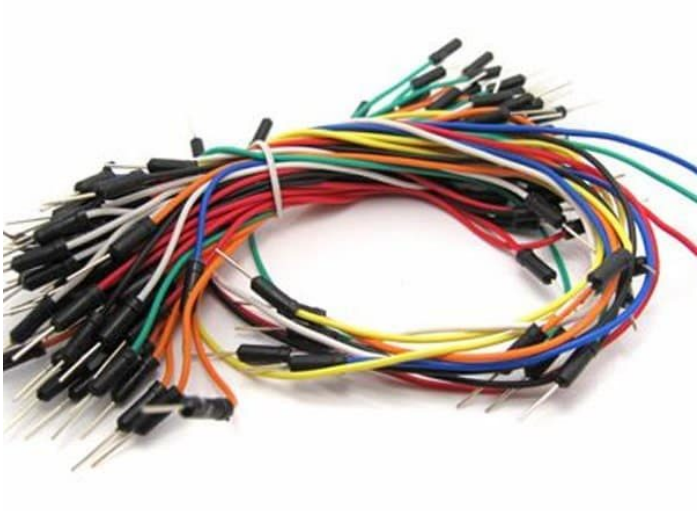
LED



Ohm Resistor 220



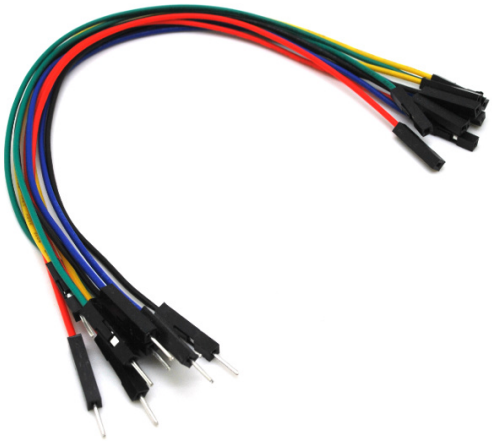
Breadboard



Wires



1× سلك الاردوينو



حزمة أسلاك توصيل (ذكر – أنثى)

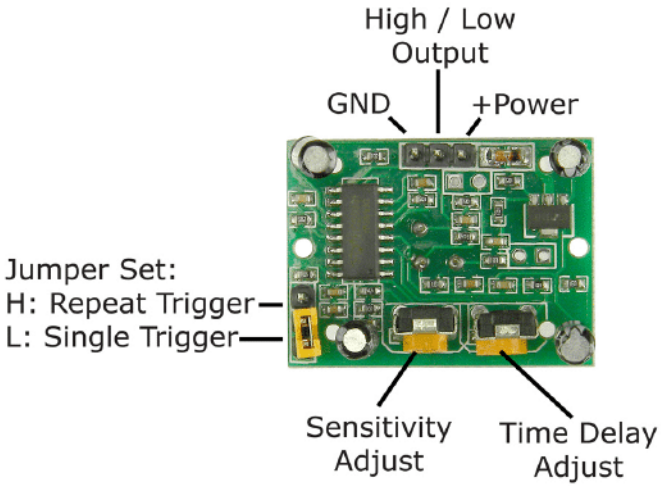
حساس الحركة PIR Sensor

يعمل الحساس على قياس مقدار التغير في الاشعه تحت الحمراء الصادرة عن الاجسام و في حالتنا الانسان. ولكن لا يقوم الحساس بقياس كمية الأشعة الصادرة من الإنسان بل التغير الحادث لهذه الأشعة، وهكذا يشعر الحساس بوجود حركة.



عند تحرك الشخص امام الحساس يحدث تغيير في كمية الأشعة تحت الحمراء التي يستقبلها الحساس، فيعطى إشارة بأن هناك شخص امامه.

عناصر الحساس :

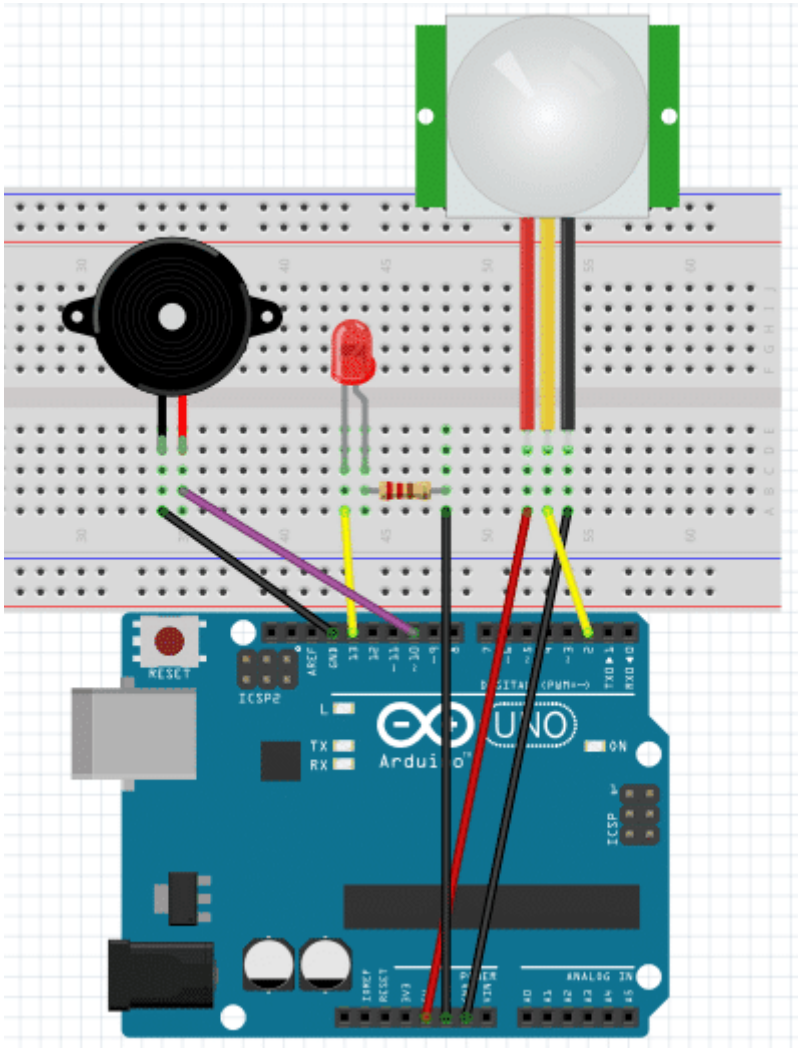


يتم توصيله إلى الأردوينو كما هو موضح بالجدول :

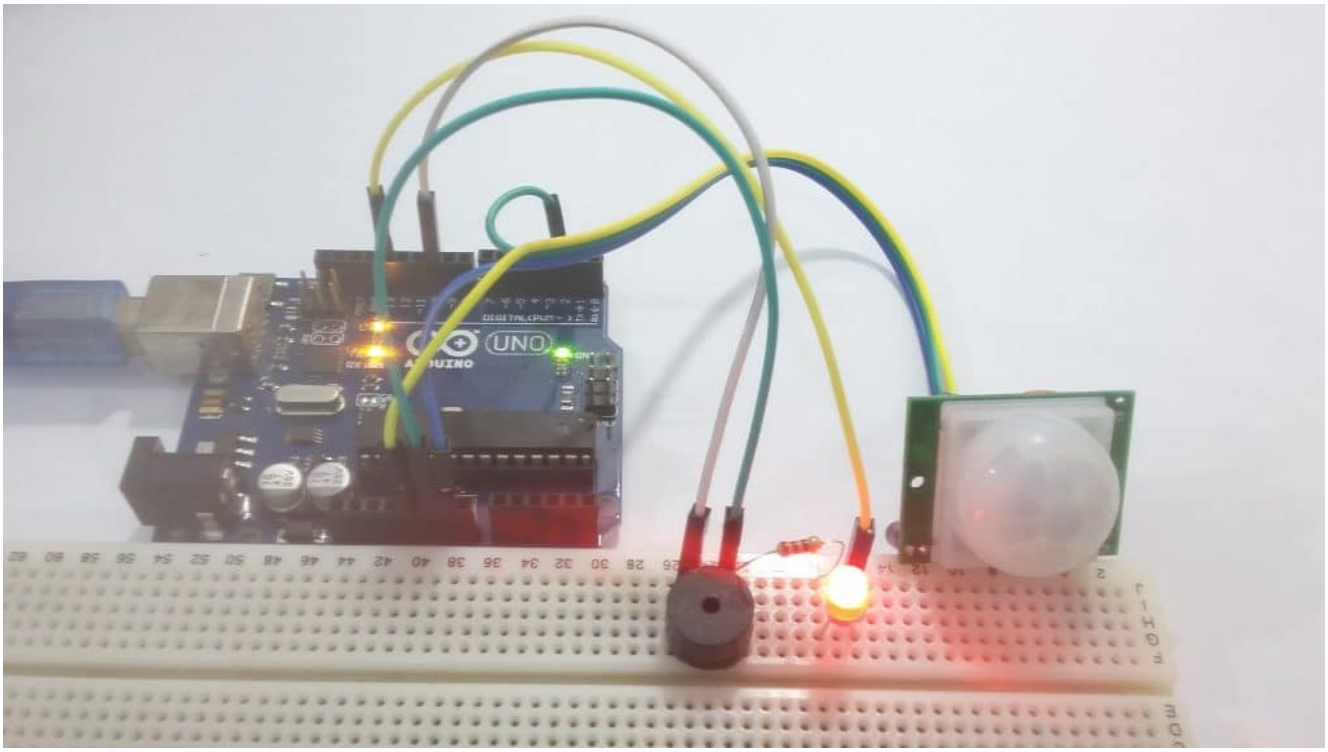
الطرف (بداية من اليسار)	التوصيل
1	GND
2	Output To Arduino
3	VCC

توصيل الدارة

قم بتوصيل الدارة كما هو موضح بالصورة التالية :



سنقوم بكتابه برنامج، بحيث يقوم الحساس بإلتقاط ما إذا كان هناك حركة ام لا، , وعند الكشف عن وجود حركة يقوم بإرسال اشارة إلى الأردوينو الذي يقوم بدوره بتشغيل الـ Buzzer وإضاءة الـ LED للتنبيه.



الكود البرمجي

```
// led
#define LED 13
// pir
#define INPUT 2
// buzzer
#define SPEAKER 10

int pirFlag = 0;
int val = 0;


void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(INPUT, INPUT);
  pinMode(SPEAKER, OUTPUT);
  Serial.begin(9600);
}


void loop()
{
  val = digitalRead(INPUT);
  // test this val
  if (val == HIGH){
    digitalWrite(LED, HIGH);
    tone(SPEAKER, 160, 300);
```

```
    delay(300);
    if (pirFlag == 0){
        Serial.println("Motion detected!");
        pirFlag = 1;
    }
  }
  else{
    digitalWrite(LED, LOW);
    noTone(SPEAKER);
    if (pirFlag == 1){
        Serial.println("Motion ended!");
        pirFlag = 0;
    }
  }
}
```

شرح الكود :

في البداية، نقوم بتسمية منافذ الأردوينو المستخدمة في المشروع، من أجل تسهيل عملية التعامل معها. ثم نقوم بالإعلان عن المتغيرات التي سنحتاج استخدامها في البرنامج.

سيتم استخدام المتغير pirFlag لتسجيل حالة الحساس. نقوم بوضع قيمة ابتدائية للحالة وهي 0. والمتغير val يستخدم لتسجيل الإشارة القادمة من الحساس إلى الأردوينو حسب وجود حركة ام لا.

```
// led
#define LED 13
// pir
#define INPUT 2
// buzzer
#define SPEAKER 10

int pirFlag = 0;
int val = 0;
```

في الدالة ()setup، قمنا بضبط المنافذ المستخدمة إما مدخله أو مخرجة. يتم ضبط الـ LED والـ Buzzer كمخرج، والمنفذ الموصل مع الحساس كمدخل.

ويتم تفعيل الاتصال التسلسلي من أجل الطباعة على الشاشة التسلسلية Serial Monitor عند الكشف عن وجود حركة.

```
void setup()
{
  pinMode(LED, OUTPUT);
  pinMode(INPUT, INPUT);
  pinMode(SPEAKER, OUTPUT);
  Serial.begin(9600);
}
```

في دالة الـ ()loop، نقوم بقراءة الإشارات القادمة من الحساس واختبارها. إذا كانت الإشارة High اي انه تم الكشف عن وجود حركة، نقوم بتشغيل ال LED والـ Buzzer . ثم نقوم بتحقق من الحالة المسجله في المتغير PirFlag إذا كانت 0 نقوم بطباعة رسالة تدل على

وجود حركة على الشاشة التسلسلية وتغير قيمة المتغير pirFlag إلى 1 .

```
val = digitalRead(INPUT);
  // test this val
  if (val == HIGH){
    digitalWrite(LED, HIGH);
    tone(SPEAKER, 160, 300);
    delay(300);
    if (pirFlag == 0){
        Serial.println("Motion detected!");
        pirFlag = 1;
    }
  }
```

في حال كانت الإشاءة القادمة من الحساس LOW اي انه لم يتم الكشف عن وجود حركة، نقوم بإيقاف تشغيل الـ LED و الـ Buzzer . والتحقق من الحالة المسجله لدى الـ pirFlag إذا كانت 1 اي انه كانت هناك حركة وتوقفت فنقوم بالطباعة على الشاشة التسلسلية رساله تدل على وقف الحركة، وتغير قيمة المتغير pirFlag إلى 0 .

```
  else{
    digitalWrite(LED, LOW);
    noTone(SPEAKER);
    if (pirFlag == 1){
        Serial.println("Motion ended!");
        pirFlag = 0;
    }
  }
```

تستخدم الدالة tone() لتوليد اشارات يكمننا من سماعها عن طريق سماعه او Buzzer

```
tone(SPEAKER, 160, 300);

noTone(SPEAKER);
```

```
tone(اسم الرجل, التردد , المدة الزمنية)
noTone(اسم الرجل)
```