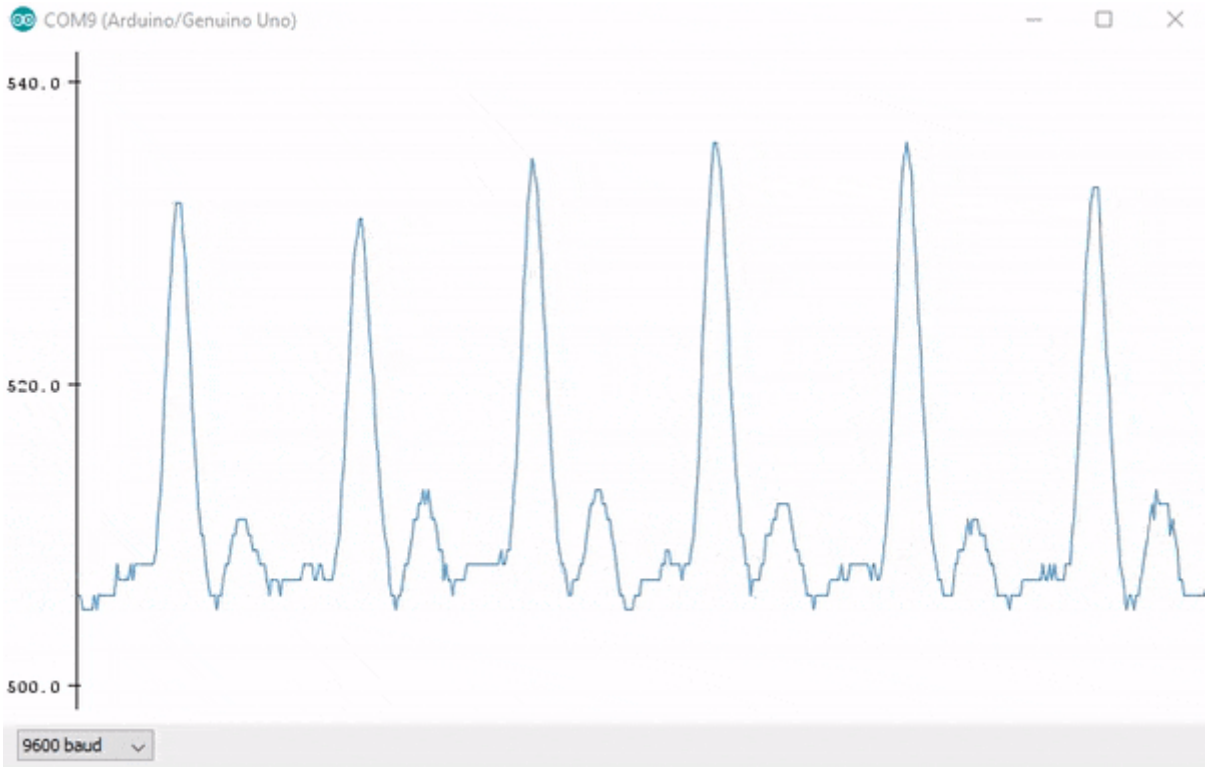
تخطيط نبض القلب

مع تطور التقنية والصناعة الإلكترونية، أصبح جمع البيانات الطبية أسهل بشكل كبير. فقد أصبحت أجهز قياس وتخطيط القلب الكهربائي أصغر حجما، وذات أوزان أقل، وأيضا ذات استهلاك طاقة أقل، مما جعلها أحد بديهيات أنظمة التشخيص الطبي في أي مركز رعاية صحية أو مستشفى.

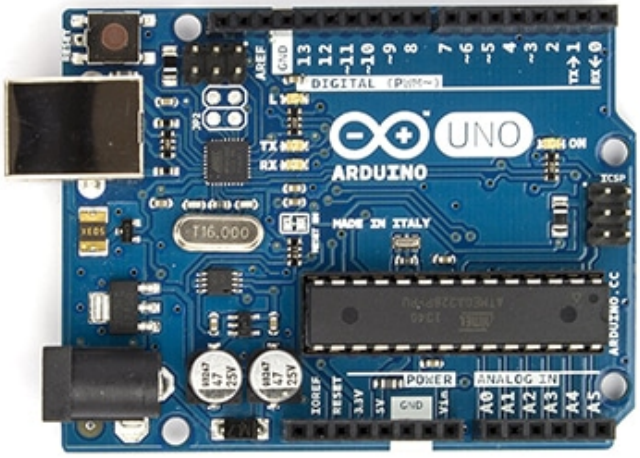
لذلك قمنا بعمل مشروع قياس ومراقبة نبض القلب، من خلاله يمكنك مراقبة معدل ضربات القلب الخاصة بك في المنزل.



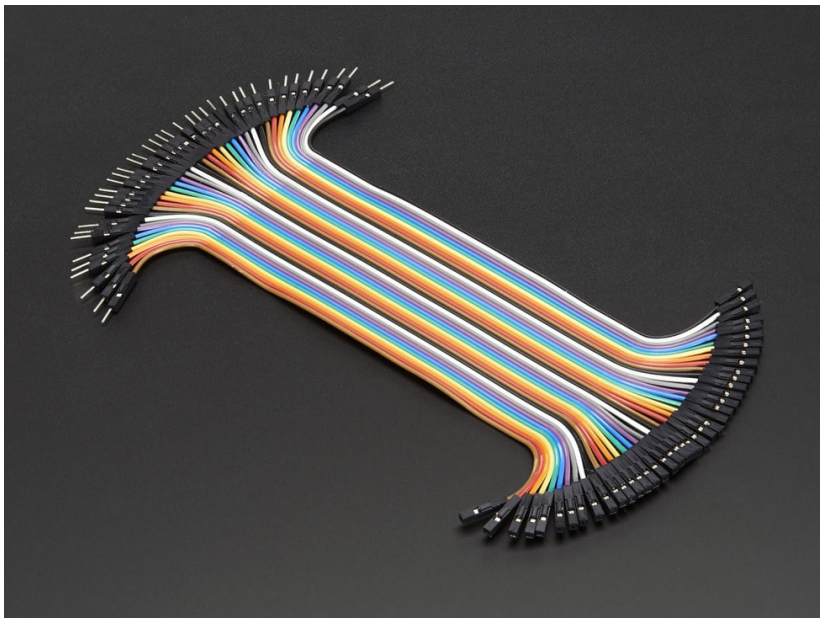
القطع المطلوبة

في هذا المشروع سنقوم بإستخدام القطع التالية :

مستشعر لقياس معدل نبضات القلب



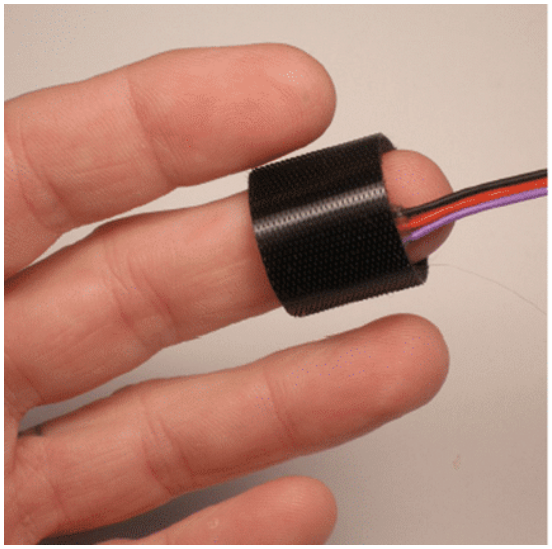
Arduino Uno R3



مستشعر نبض القلب :

جهاز استشعار النبض هو جهاز مصمم بشكل جيد لتوصيله وتشغيله مع الاردوينو لغرض قياس معدل نبضات القلب. يمكن استخدامه في العديد من المشاريع الإلكترونية، من قبل الطلاب والفنانين والرياضيين والصناع ومطوري الألعاب الذي يرغبون في دمج بيانات معدل ضربات القلب في مشاريعهم. فهو يتكون من جهاز استشعار معدل ضربات القلب البصرية مع مضخم ودوائر لإلغاء الضوضاء مما يجعلها سريعة وسهله للحصول على قراءات نبض موثوق بها.

الجزء الأمامي من جهاز الاستشعار (الجانب مع شعار القلب) هذا الجانب هو الذي سيتم اتصاله مع الجلد. على هذه الجهة سترى حفرة مستديرة صغيرة، وهي عبارة عن LED يضئ، وهناك أيضا مربع تحت الـ LED قليلا. هذا المربع عبارة عن جهاز استشعار للضوء المحيط، تماما مثل تلك المستخدمة في الهواتف المحمولة، والأقراص، وأجهزة الكمبيوتر، لضبط سطوع الشاشة في ظروف الإضاءة المختلفة.

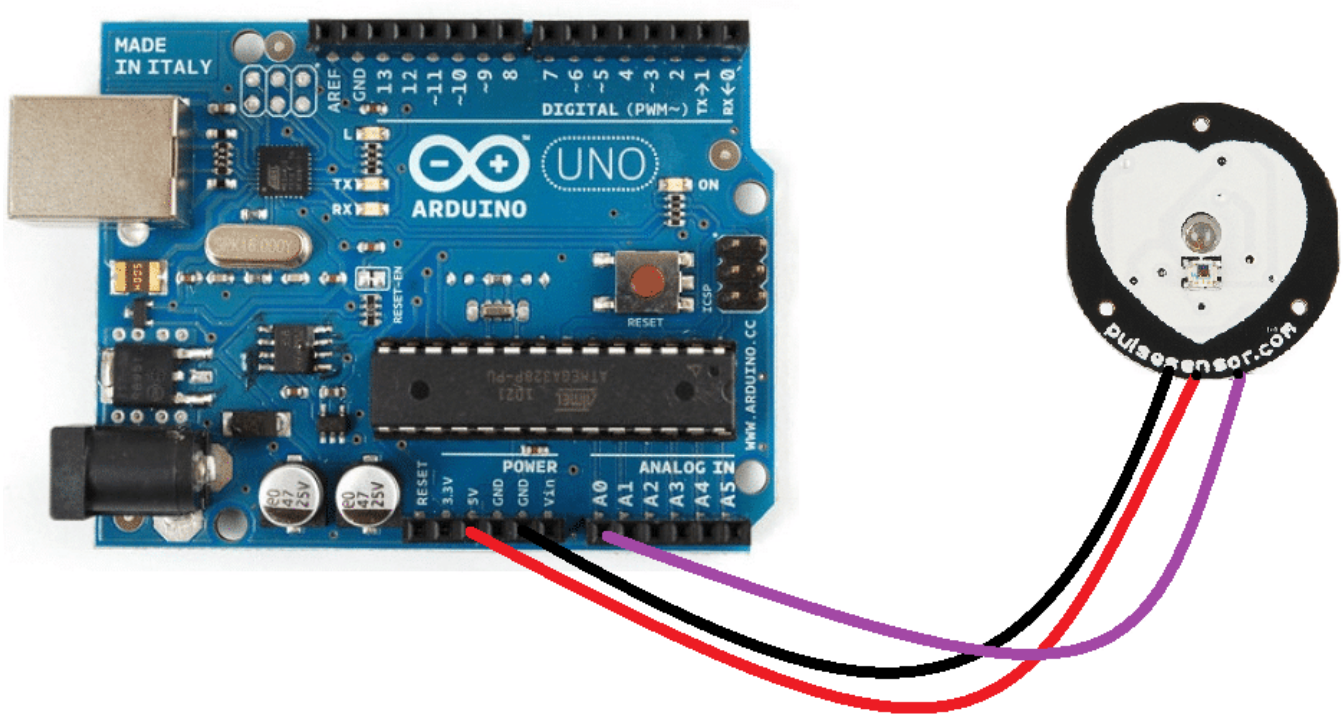


يضئ الـ LED في الإصبع أو شحمة الأذن، أو غيرها من الأنسجة الشعرية، ويقوم جهاز الاستشعار بقراءة الضوء الذي يرتد مرة أخرى.

الجزء الخلفي من أجهزة الاستشعار هو المكان الذي يتم تركيب بقية الأجزاء.

تصميم لوح التجارب :

قم بتوصيل الدائرة كما هو موضح بالصورة التالية :



البرمجة :

قم بتحميل الكود التالي إلى الأردوينو :

```
//  Variables
int PulseSensorPurplePin = 0;        // Pulse Sensor PURPLE WIRE connected to ANALOG
PIN 0
int LED13 = 13;   //  The on-board Arduion LED
```

```
int Signal;               // holds the incoming raw data. Signal value can range
from 0-1024
int Threshold = 510;      // Determine which Signal to "count as a beat", and which
to ingore.


// The SetUp Function:
void setup() {
  pinMode(LED13,OUTPUT);       // pin that will blink to your heartbeat!
   Serial.begin(9600);         // Set's up Serial Communication at certain speed.
}

// The Main Loop Function
void loop() {
  Signal = analogRead(PulseSensorPurplePin);  // Read the PulseSensor's value.
                                              // Assign this value to the "Signal"
variable.
   Serial.println(Signal);                    // Send the Signal value to Serial
Plotter.
   if(Signal > Threshold){                    // If the signal is above "510", then
"turn-on" Arduino's on-Board LED.
     digitalWrite(LED13,HIGH);
   } else {
     digitalWrite(LED13,LOW);    //  Else, the sigal must be below "510", so "turn-
off" this LED.
   }

delay(10);
}
```

لمحة عن الكود :

أولا نقوم بتعريف المتغيرPulseSensorPurplePin إعتمادا على ما تم توصيله بين المستشعر والأردوينو، ومتغير LED13 وهو الضوء الموجود على لوحة الأردوينو

```
int PulseSensorPurplePin = 0;
int LED13 = 13;
```

ومن ثم تعريف المتغير Signal الذي سيتم استخدامه لقراءة المستشعر، ومتغير Threshold لحديد حد للإشارات التي سيتم احتسابها كنبضة، والإشارات التي سيتم تجاهلها.

```
int Signal;
int Threshold = 510;
```

في دالة ()setup ، نقوم بتهيئة المتغيرات كمخرج أو مدخل.

```
void setup() {
  pinMode(LED13,OUTPUT);
   Serial.begin(9600);
}
```

في دالة ()loop، نقوم بقراءة المستشعر ومن ثم طباعة القيمة.

```
 Signal = analogRead(PulseSensorPurplePin);
  Serial.println(Signal);
```

ثم نقوم بمقارنة قيمة المستشعر بالحد Thresholdالذي تم تعريفه مسبقا، إذا كانت القيمة أكبر من Threshold يتم إضاءة الـ LED 13 على لوحة الأردوينو، وإذا كانت أقل يتم إطفاءه.

```
if(Signal > Threshold){
    digitalWrite(LED13,HIGH);
  } else {
    digitalWrite(LED13,LOW);
  }
```

وأخيرا قم بفتح نافذة Serial Plotter وراقب نبضك.