

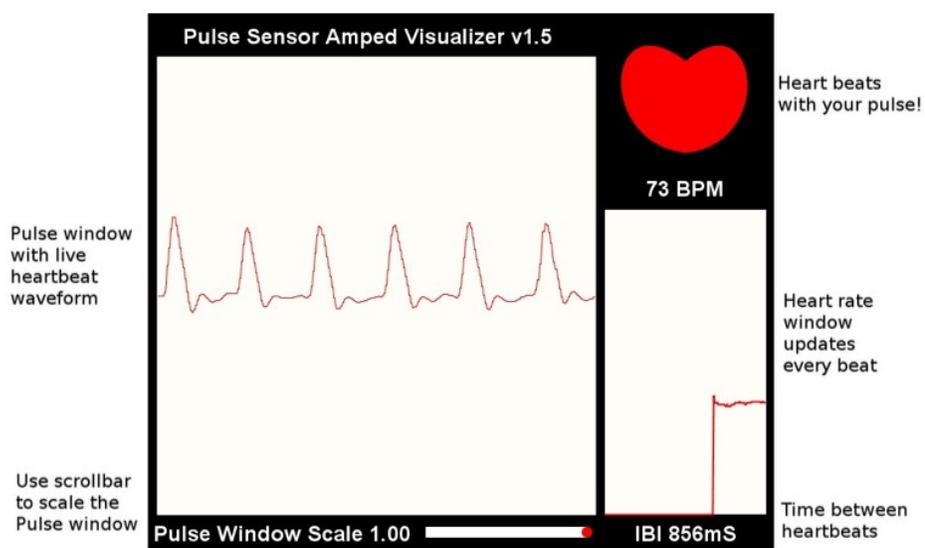
Sensor de pulso cardiaco

(Versión 29-6-18)



La siguiente figura corresponde a un proyecto desarrollado en Processing.

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Processing es la base del código de Arduino.



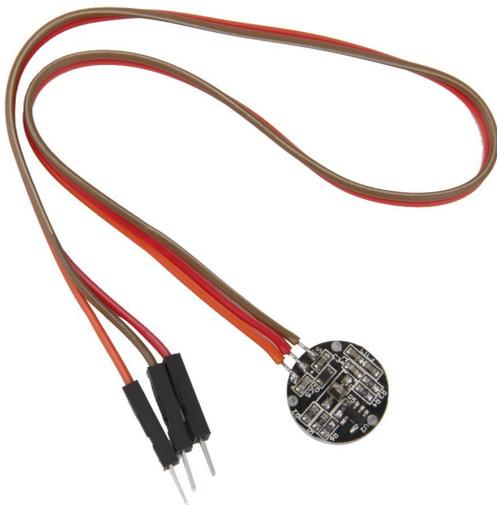
Descripción del Producto

Los datos de la frecuencia cardíaca pueden ser muy útiles cuando se esté diseñando una rutina de ejercicios, estudiando la actividad física o niveles de ansiedad o simplemente si quiere que su camisa parpadee con los latidos de su corazón. El problema es que la frecuencia cardíaca puede ser difícil de medir. Afortunadamente, el sensor de pulso puede resolver ese problema.

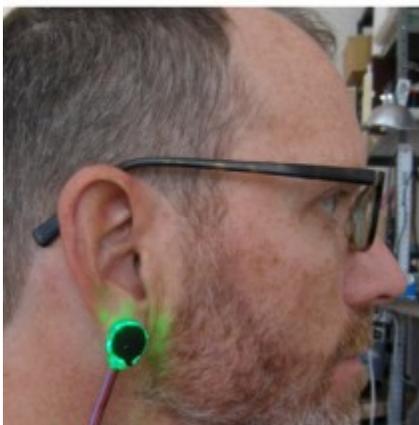
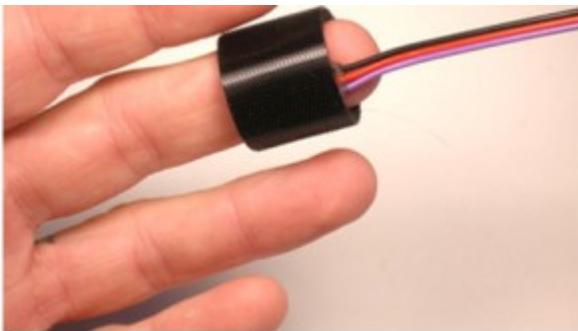
El Sensor de pulso es un sensor de frecuencia cardíaca plug-and-play para Arduino. Puede ser utilizado por estudiantes, artistas, atletas, fabricantes y desarrolladores de juegos y dispositivos móviles que deseen incorporar fácilmente datos de frecuencia cardíaca en vivo en sus proyectos. Combina esencialmente un sensor de frecuencia cardíaca óptico simple con circuitos de cancelación de ruido y amplificación que lo hacen rápido y fácil para obtener lecturas de pulso confiables. Además, consume solo 4mA de corriente a 5V, por lo que es ideal para aplicaciones móviles.

¡Simplemente coloque el Sensor de pulso en el lóbulo de su oreja o en su dedo y conéctelo a su Arduino de 3.3 o 5 V y estará listo para leer la frecuencia cardíaca! El cable del Sensor de pulso termina con unos pines macho estándar, por lo que no se requiere soldadura. Por supuesto, el código de ejemplo de Arduino está disponible, así como un programa de procesamiento para visualizar datos de frecuencia cardíaca.

La esencia de estos circuitos es un sensor integrado de circuito de amplificación óptica y con un circuito de eliminación de ruido de la frecuencia cardíaca todo ello alimentado con una tensión de alimentación: 3.3V ~ 5 V. O sea esta optimizado a la señal que va a percibir.



Lo ideal para medir el pulso es poner el sensor de pulso en el dedo o lóbulo de la oreja, directamente o bien mediante algún sistema mecánico que lo deje fijo como por ejemplo alguno de los sistemas que mostramos a continuación:



Estos sensores cuentan con una salida analógica que se puede conectar por ejemplo a una entrada analógica de un Arduino, para probar la frecuencia cardíaca. Estudiantes, artistas, deportistas, creadores, desarrolladores de juegos, o terminales móviles puedan desarrollar software o interactivos relacionado con el ritmo cardíaco.

+++++

CONCEPTOS FISIOLÓGICOS INVOLUCRADOS

El sensor de pulso cardíaco es esencialmente un **fotopleletismógrafo** que es un dispositivo médico conocido que se usa para controlar la frecuencia cardíaca de forma no invasiva. Asimismo los fotopleletismógrafos miden los niveles de oxígeno en la sangre (SpO2) pero no siempre lo soportan.

La señal de pulso cardíaco que sale de un fotopleletismograma **es una fluctuación analógica de voltaje**, y tiene una *forma de onda predecible*, tal como estamos acostumbrados a ver (la representación de la onda de pulso se denomina fotopleletismograma o **PPG**).

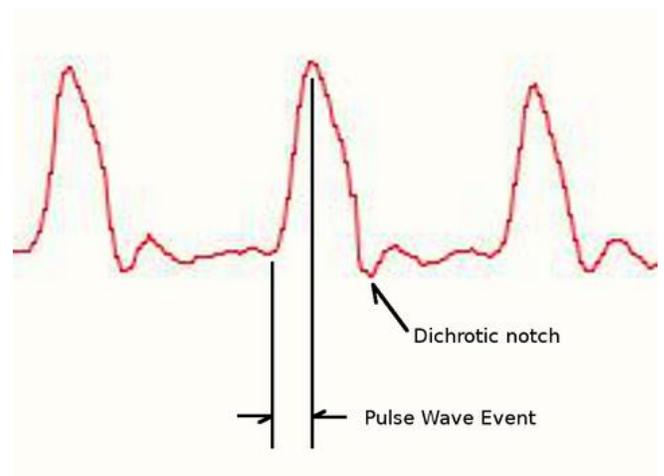
El Sensor de pulso amplifica la señal bruta del Sensor de pulso anterior y normaliza la onda de pulso alrededor de $V / 2$ (punto medio en voltaje) respondiendo a los cambios relativos en la intensidad de la luz. Tal y como esta construido, veremos que la luz interna del LED verde del sensor se refleja de nuevo en el sensor cambiando durante cada impulso, ocurriendo lo siguientes:

- Si la cantidad de luz incidente en el sensor permanece constante, el valor de la señal permanecerá en (o cerca de) 512 (punto medio del rango de ADC).
- Más luz y la señal aumentará.
- Menos luz, todo lo contrario: el valor de la señal analógica disminuirá



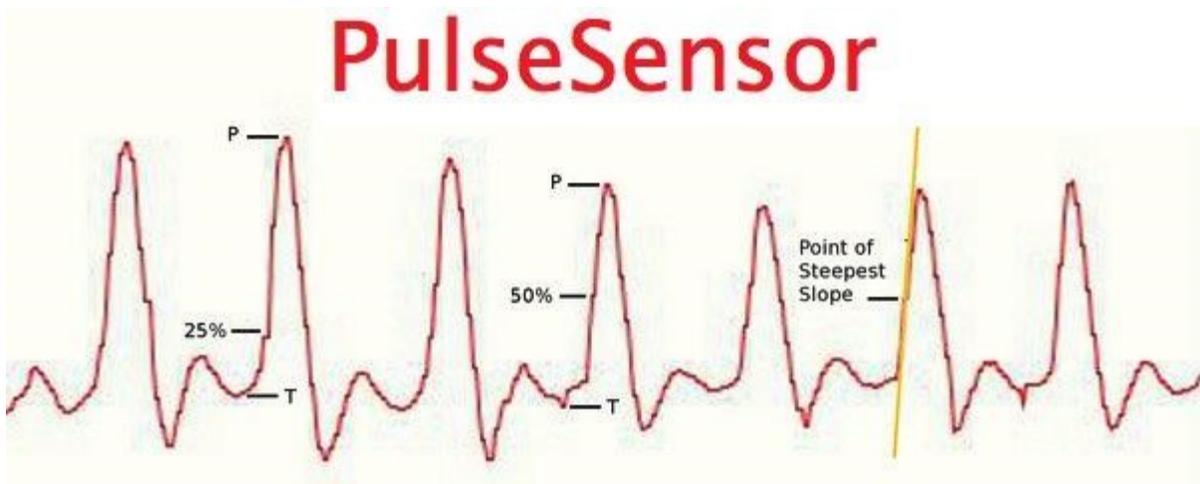
El objetivo es encontrar **momentos sucesivos de latido instantáneo del corazón y medir el tiempo transcurrido entre ellos**, llamado intervalo **Inter Beat (IBI)** pues al seguir la forma y el patrón predecibles de la onda PPG, podemos hacer exactamente eso.

Cuando el corazón bombea sangre por el cuerpo, con cada latido hay una onda de pulso (una especie de onda de choque) que viaja a lo largo de todas las arterias hasta las mismas extremidades del tejido capilar donde está conectado el sensor de pulso. La sangre real circula en el cuerpo mucho más lentamente de lo que viaja la onda de pulso.



Sigamos los eventos a medida que progresan desde el punto **'T' en el PPG** a continuación. Se produce un aumento rápido en el valor de la señal a medida que la onda de pulso pasa por debajo del sensor, luego la señal vuelve a descender hacia el punto normal. A veces, la muesca dicróica (pico descendente) es más pronunciada que otras, pero, en general, la señal se establece en el ruido de fondo antes de que la siguiente onda de pulso se filtre.

Como la onda se repite y es predecible, podríamos elegir casi cualquier característica reconocible como punto de referencia, por ejemplo, el pico, y medir la frecuencia cardíaca haciendo cálculos matemáticos sobre el tiempo entre cada pico, pero sin embargo, esto puede dar lugar a lecturas falsas.

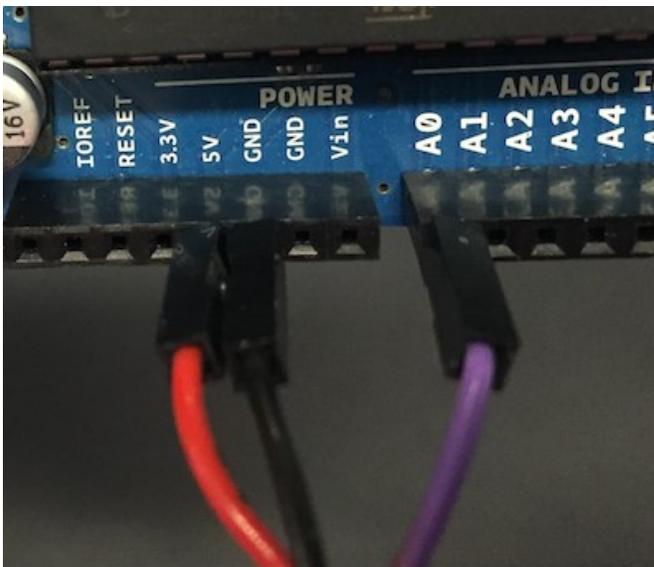


Algunos investigadores del corazón dicen que es cuando la señal alcanza el 25% de la amplitud, algunos dicen que es el 50% de la amplitud, y algunos dicen que es el momento en que la pendiente es más pronunciada durante el evento ascendente.

CIRCUITO EMPLEADO CON ESTE SENSOR

El circuito que vamos a ver es muy simple pues solo se precisa conectar un buzzer y el sensor de pulsos cardíacos.

Como podemos ver en el video, el sensor de pulsos cardíacos se conecta a la alimentación de +5V entre el hilo rojo (+5v) y el naranja (GND) y del hilo marrón obtenemos la salida analógica que conectaremos a la primera entrada analógica (A0) de cualquier placa que soporte entradas analógicas como pueden ser **Arduino** o **Netduino**.



Arduino



Netduino

Para complementar el circuito puede ser interesante reflejar el punto máximo de nivel que reproduciremos mediante un buzzer conectado al pin 11(u otro) de salida binaria.

A continuación en este breve ejemplo para Arduino se puede mostrar un pulso de latido del corazón humano en directo ayudándonos por medio de **“Serial Plotter”** de arduino o por ejemplo con una aplicación móvil usando un modulo Bluetooth conectado a nuestro Arduino

En este pequeño programa para Arduino que vamos a ver, sonará un buzzer con cada latido de tu corazón al mismo tiempo que se envía el valor de la señal de forma serie (esta es la señal directa del sensor de pulso) el cual podemos visualizar en un ordenador o si tenemos conectado un modulo Bluetooth a nuestro arduino mediante un smartphone usando una app .

ENSAYOS DEL DOCENTE

En siguiente programa se ensayo y visualizo mediante el Serial Plotter del IDE de Arduino.
En lugar del buzzer se utilizo un LED verde. NO SE TRABAJO CON BLUETTOOTH.

```
//Programa para capturar el pulso cardiaco- Version 1.0
//Revisado Prof: Bolaños 2018- Programa base tomado de Internet
// Variable para fijar el puerto donde conectaremos el buzzer o LED
int buzzer = 7; //Buzzer activo
// la variable pulso contiene los datos brutos entrantes pudiendo variar entre 0-1024
int pulso;

// Determina qué señal “se contará como un latido” y qué señal ignorar.
int limite = 522;//original 550- Ver los picos en Ploter y reprogramar

void setup() {
//definimos donde conectamos el buzzer , que sonará al ritmo de su corazón
pinMode(buzzer,OUTPUT);

// Configura la comunicación serial a 9600 dependiendo de su adaptador bluetooth como esté
configurado
Serial.begin(9600);
}
void loop() {
// Lee el valor del pin analógico 0, y Asigna este valor a la variable “pulso”.
pulso = analogRead(A0);

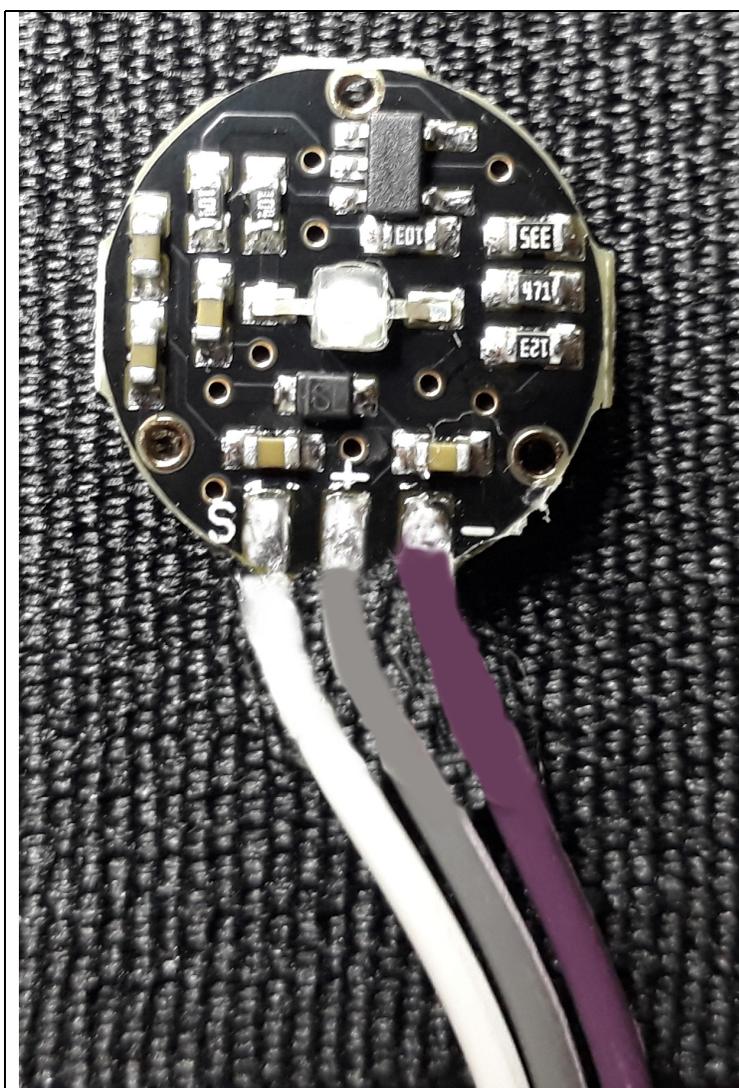
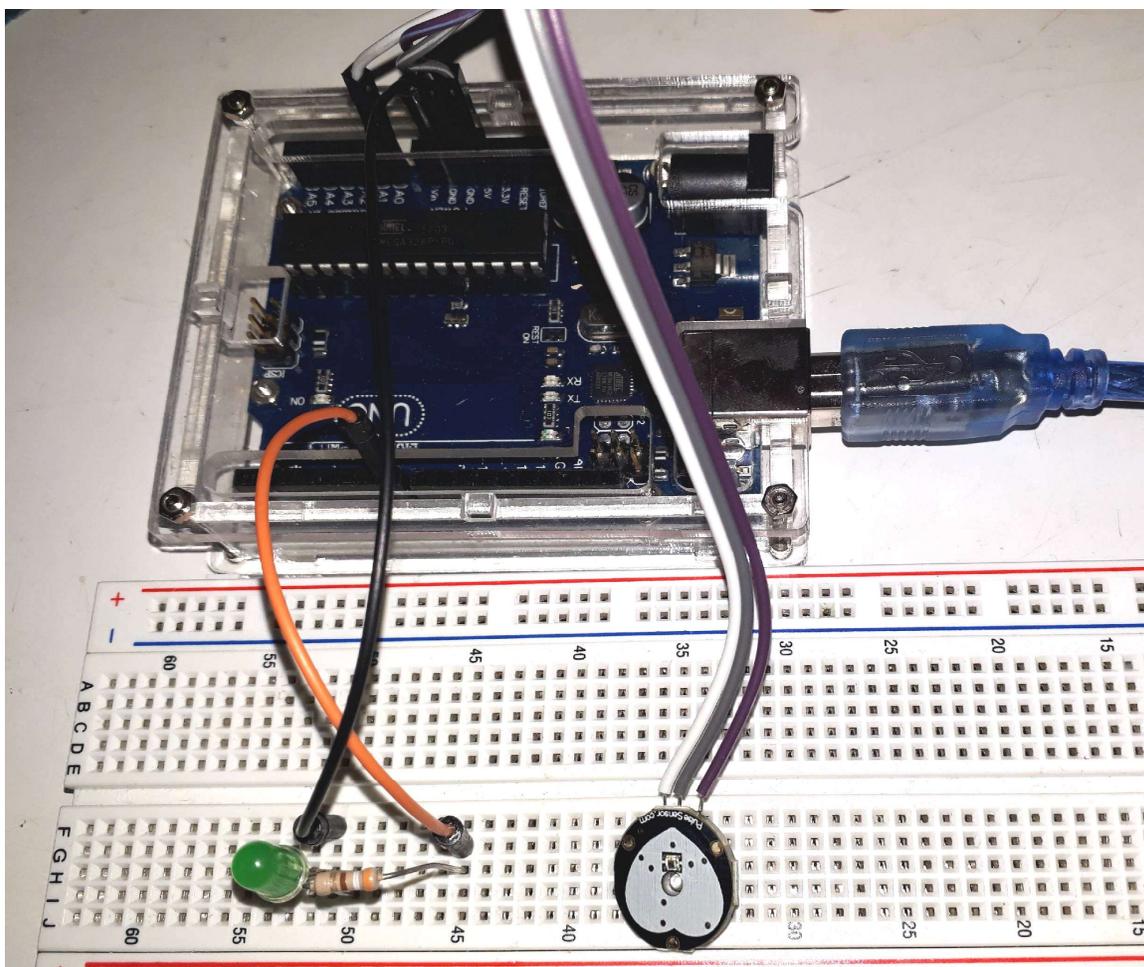
//Este caracter lo filtra la aplicación en APP inventor

//Serial.print(""); // NOTA DJB: no usada en esta version

// Envíe el valor de pulso al Plotter serial. Comentar si queremos visualizar en “serial ploter”
Serial.println(pulso);
if(pulso > limite){
// Si la señal es superior a “550”, entonces suena el buzzer.
digitalWrite(buzzer,HIGH);
}
else
{
// De lo contrario, deja de sonar el buzzer.
digitalWrite(buzzer,LOW);
}
//Retardo de 35ms
delay(35);
}
```

PROGRAMA BASADO EN LA FUENTE:

<http://soloelectronicos.com/tag/sensor-cardiaco-arduino/>



Detalle del sensor
utilizado en los ensayos:

VIOLETA = GND

GRIS = +5 V

BLANCO = SEÑAL a (A0)

Observándose:

