



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Práctica 6: Pulsos por minuto con
Fotopletismógrafo

Unidad de aprendizaje: Instrumentación

Grupo: 3CM4

Integrantes:

Aguilar Herrera Arianna Itzamina

Nicolás Sayago Abigail

Ramos Diaz Enrique

Profesor(a):

Tellez Barrera Juan Carlos

Fecha de entrega: 17 de noviembre de 2018

Índice

1 Objetivo	2
2 Listado de materiales	2
3 Listado de equipo	2
4 Introducción	3
4.1 Fotopletismógrafo	3
4.2 Arduino	4
4.3 Bluetooth	5
5 Desarrollo práctico	6
5.1 Recordando la práctica 5	6
5.2 PE 1 - Transistor como interruptor	6
5.2.1 Esquema de conexión	6
5.2.2 Circuito cableado	6
5.2.3 Funcionamiento	6
5.3 PE 2 - Módulo Bluetooth HC-05 y placa Arduino	8
5.3.1 Esquema de conexión	8
5.3.2 Circuito cableado	8
5.3.3 Funcionamiento	9
5.4 PE 3 - Programa en Arduino	10
5.4.1 Código	10
5.4.2 Explicación del código	10
5.5 PE 4 - Aplicación Android desarrollada para móviles	12
5.5.1 Código e interfaz gráfica	12
5.5.2 Explicación del código	12
5.5.3 Instalación	14
5.6 Funcionamiento completo	15
6 Observaciones	19
6.1 Aguilar Herrera Arianna Itzamina	19
6.2 Ramos Diaz Enrique	19
6.3 Nicolas Sayago Abigail	20
7 Anexos	20
7.1 BC547	20
7.2 LM741	21
7.3 Arduino Mega	22
7.4 HC-05	22
7.5 Sensor TCRT5000L	23
7.6 Página Web: MIT App Inventor	23
7.7 Arduino IDE	24

Referencias	24
-----------------------	----

1. Objetivo

- ✓ Aprender el uso y características del sensor de reflexión infrarrojo TCRT5000L
- ✓ Graficar los pulsos del corazón con ayuda de un instrumento, el fotopletismógrafo.
- ✓ Programar una aplicación en Arduino para interpretar las señales de salida de un fotopletismógrafo.
- ✓ Enviar las mediciones de los pulsos por minuto a un dispositivo móvil, por medio de conexión Bluetooth.

2. Listado de materiales

Resistores:

- 1 de $150\ \Omega$
- 3 de $1\ k\Omega$
- 2 de $10\ k\Omega$
- 2 de $220\ k\Omega$
- 3 de $330\ k\Omega$

Preset ó Potenciómetro:

- 1 de $10\ k\Omega$

Capacitores:

- 5 de $0.47\ \text{microF}$ o $470\ \text{nanoF}$
- 1 de $100\ nF$

Semiconductores:

- 1 Sensor de reflexión infrarrojo TCRT5000L
- 1 Diodo LED

Circuitos integrados:

- 1 LM741 OPAMP (alimentación $\pm 12V$)
- Arduino Mega o Uno
- Módulo Bluetooth HC-05

Opcionales:

- 1 Transistor BC547
- 1 Resistor $1K\Omega$

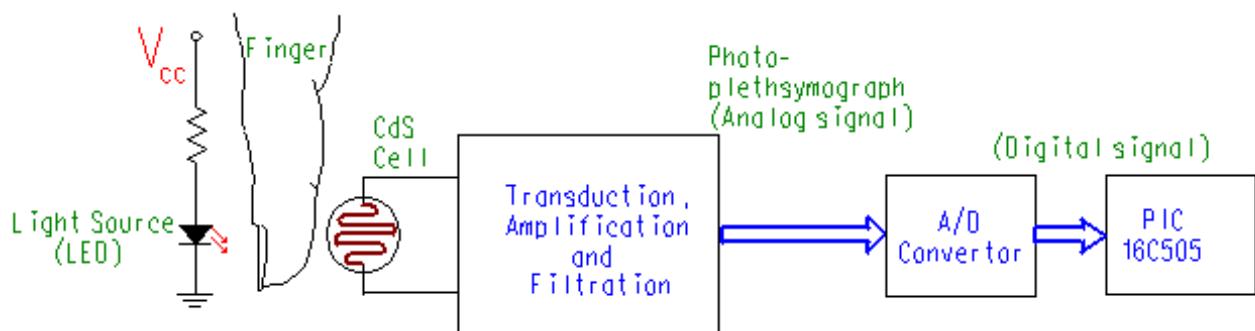
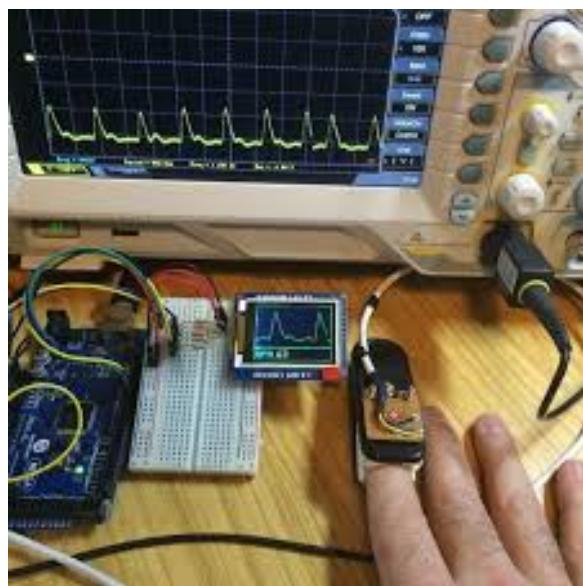
3. Listado de equipo

- 1 Fuente de alimentación.
- 1 Osciloscopio y voltímetro digital
- Celular smartphone con Bluetooth superior a la versión 5 y Android superior a la versión 2.1

4. Introducción

4.1. Fotopletismógrafo

En teoría, el fotopletismógrafo es simple: mide la variación en la cantidad de luz que pasa a través de su dedo causada por la naturaleza pulsátil del flujo sanguíneo.



Un fuente de luz infrarroja (LED) emite un haz sobre la piel para iluminar los vasos subcutáneos, estos reflejan parte de dicho haz dependiendo la cantidad de hematíes que contienen. La luz reflejada incide en un fotosensor (usualmente de cadmio-selenio) que la convierte en un voltaje equivalente.

Debido a que la piel absorbe más del 90 % de la luz, el par diodo-fotosensor se acompaña de amplificadores y filtros que garantizan un voltaje adecuado. El ciclo cardíaco puede obtenerse midiendo el intervalo que existe entre cada pico de voltaje.

La calidad de las mediciones realizadas utilizando esta técnica dependen de la ubicación del emisor y receptor de luz, las características de los tejidos de cada paciente, la calidad de los amplificadores y filtros, entre otros. No obstante, su uso es generalizado teniendo en cuenta su bajo costo y su carácter no invasivo

La señal que se trata de observar consiste en variaciones muy pequeñas (cambios en la intensidad de la luz debidos al flujo de sangre hacia o desde el dedo), superpuestas a una señal constante grande (luz promedio que fluye a través del dedo).

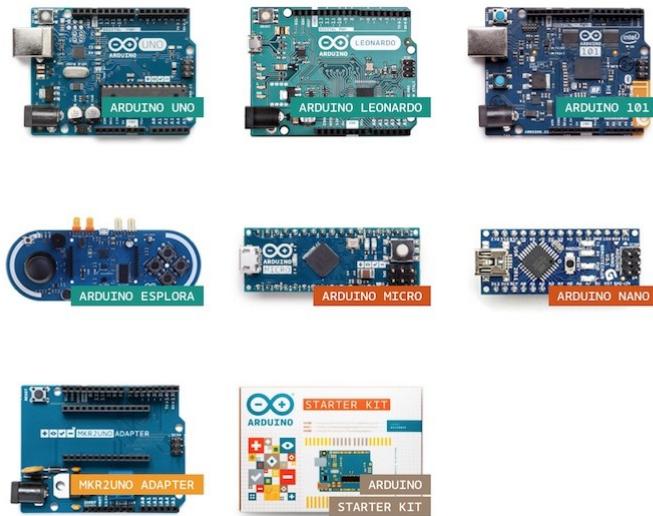
Es solo la parte de la señal que varía con el tiempo la que se debe amplificar. Para deshacerse de la señal de CC se puede utilizar un filtro de pasa altas.

Para asegurarse de que la señal de interés (el fotopletismógrafo) no sea eliminada por el filtro, la frecuencia de corte del filtro debe ser inferior a aproximadamente 1 Hz (una frecuencia cardíaca típica).

4.2. Arduino

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

El Arduino es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.



El microcontrolador de Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que conectes se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos.

El tipo de periféricos que puedas utilizar para enviar datos al microcontrolador depende en gran medida de qué uso le estés pensando dar. Pueden ser cámaras para obtener imágenes, teclados para introducir datos, o diferentes tipos de sensores.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.

4.3. Bluetooth

El Bluetooth Special Interest Group (SIG), una asociación comercial formada por líderes en telecomunicación, informática e industrias de red, está conduciendo el desarrollo de la tecnología inalámbrica Bluetooth y llevándola al mercado.

La tecnología inalámbrica Bluetooth es una tecnología de ondas de radio de corto alcance (2.4 gigahertzios de frecuencia) cuyo objetivo es el simplificar las comunicaciones entre dispositivos informáticos, como ordenadores móviles, teléfonos móviles, otros dispositivos de mano y entre estos dispositivos e Internet. También pretende simplificar la sincronización de datos entre los dispositivos y otros ordenadores.

Permite comunicaciones, incluso a través de obstáculos, a distancias de hasta unos 10 metros. Esto significa que, por ejemplo, puedes oír tus mp3 desde tu comedor, cocina, cuarto de baño, etc. También sirve para crear una conexión a Internet inalámbrica desde tu portátil usando tu teléfono móvil. Un caso aún más práctico es el poder sincronizar libretas de direcciones, calendarios etc en tu PDA, teléfono móvil, ordenador de sobremesa y portátil automáticamente y al mismo tiempo.



5. Desarrollo práctico

5.1. Recordando la práctica 5

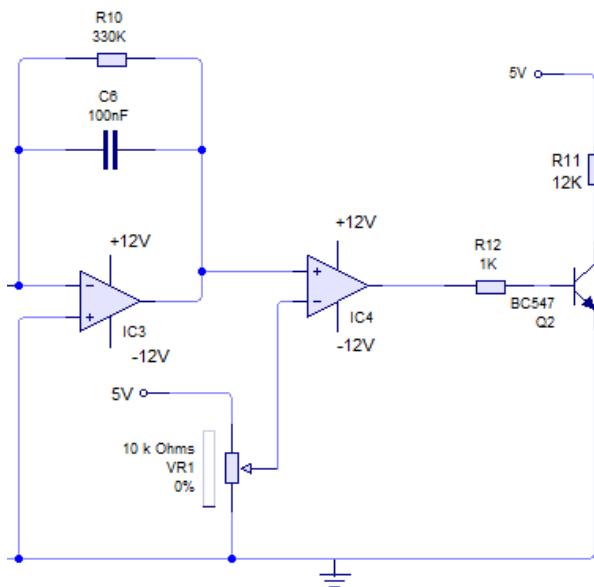
En la práctica anterior acondicionamos un sensor TCRT5000L para realizar un fotopletismógrafo, un circuito que mide las pulsaciones cardiacas de un individuo por medio de los cambios de luz que genera la sangre en los dedos de la mano.

Ahora bien, en aquella práctica únicamente llegamos hasta la etapa de un comparador. En ésta práctica vamos a interpretar esta señal obtenida como el número de pulsos del corazón por minuto, por medio de una aplicación móvil y Arduino.

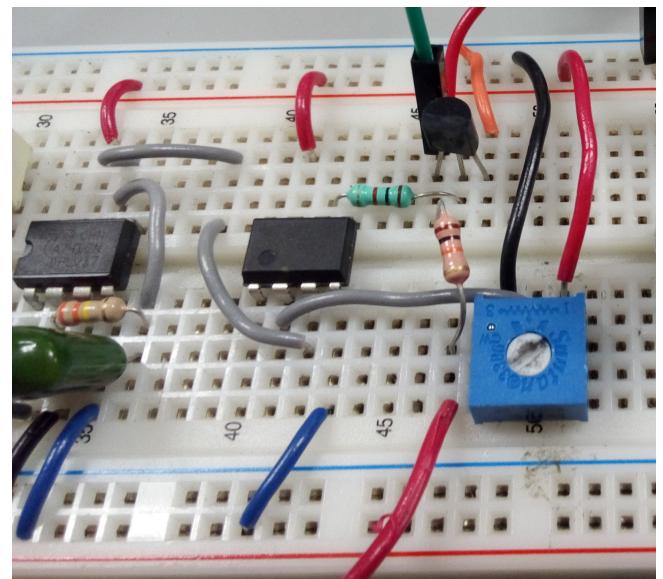
Antes de entrar de lleno a la parte de Arduino, Bluetooth y Android, primero debemos realizar un ligero ajuste.

5.2. PE 1 - Transistor como interruptor

5.2.1. Esquema de conexión



5.2.2. Circuito cableado



5.2.3. Funcionamiento

Tomando el circuito anterior, debemos de quitar el LED y su resistencia que encendía a cada pulso cardíaco, y en su lugar colocar un transistor BC547, que va a funcionar como interruptor que limita la salida del comparador de voltaje a 5V, entregándonos una señal cuadrada (quitamos el LED para no reducir la corriente que debe ir hacia el transistor).

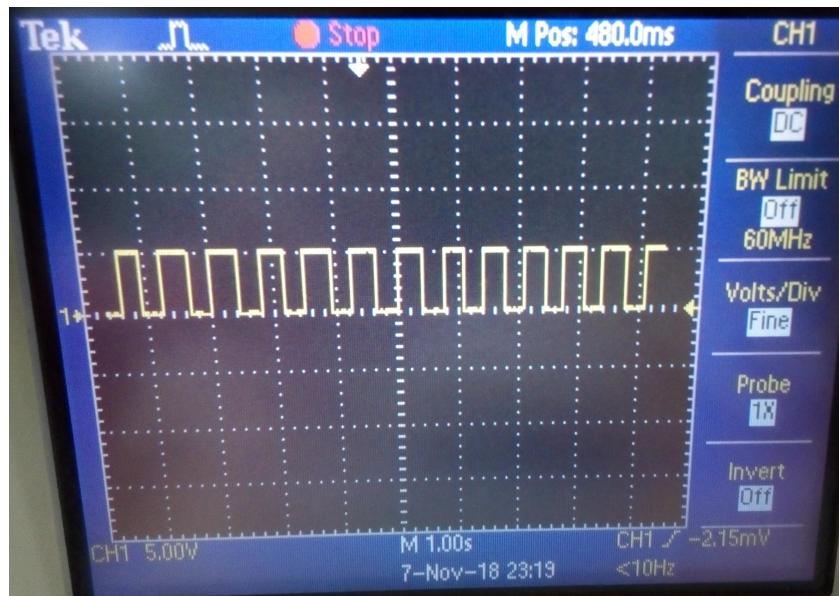
Específicamente, cuando el sensor no detecta ningún cambio de luz, es decir, no existe

pulso o no se ha colocado el dedo en él, la señal se mantendrá en 5 V constantes; cada vez que el sensor detecte un pulso, habrá un flanco de bajada hasta los 0 V, y cuando no se detecte nada, habrá un flanco de subida a 5 V nuevamente.

Parámetros de medición (Vpp)

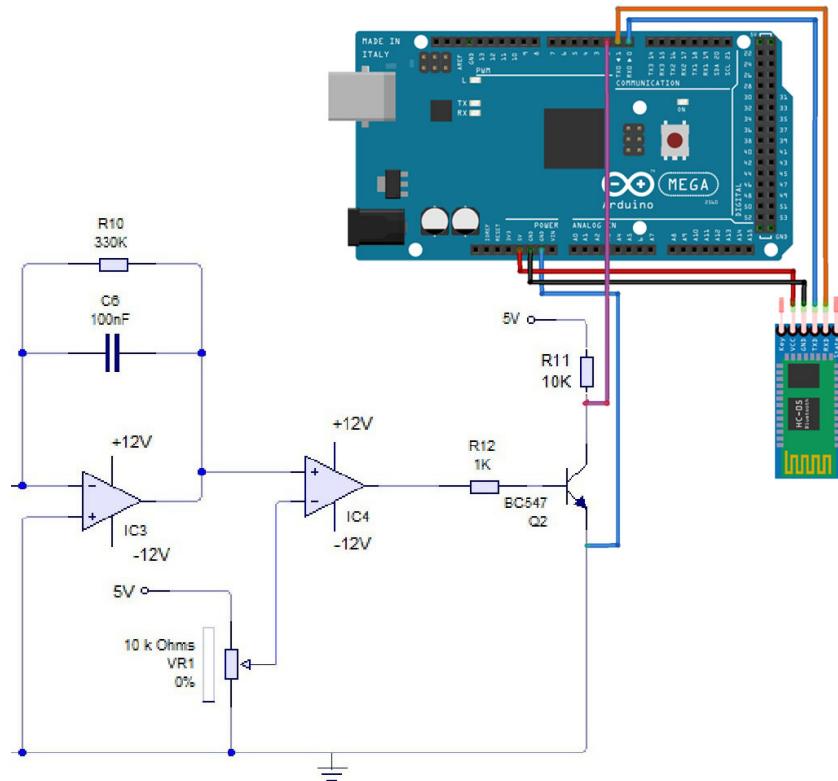
- ✓ **Amplitud máxima: 5 V**
- ✓ **Volts por división: 5 V**
- ✓ **Base de tiempo: 1 s**
- ✓ **Acoplamiento: Corriente Continua**

Nota: Nos interesa obtener una señal digital, que arroje valores lógicos de 1 (5V) y 0 (0V) para poder interpretarlas en Arduino.

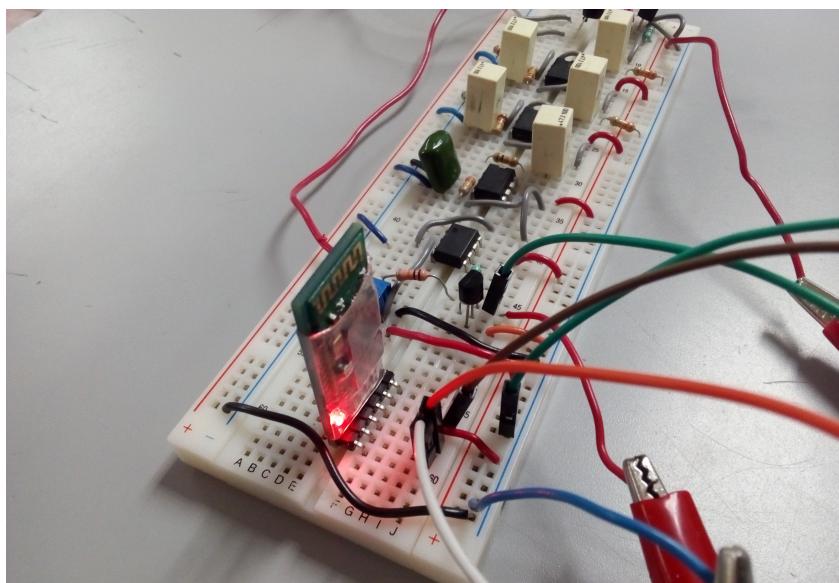


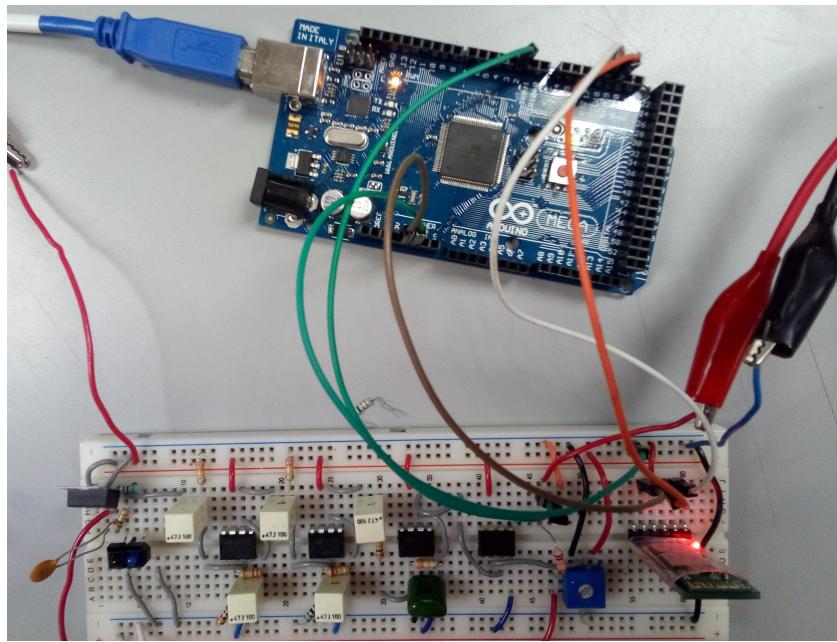
5.3. PE 2 - Módulo Bluetooth HC-05 y placa Arduino

5.3.1. Esquema de conexión



5.3.2. Circuito cableado





5.3.3. Funcionamiento

La señal que sale de la pata colectora del transistor va ir directamente al pin 2 de la placa de Arduino.

A su vez, enviamos el pin que entrega 5V al pin VCC del módulo Bluetooth HC-05, quien va a ser el encargado de transmitir la información codificada desde la placa de Arduino hacia nuestra aplicación Android en nuestro dispositivo móvil.

Es de suma importancia cablear la salida del pin RX1 del Arduino hacia el pin TxD del HC-05, y de forma similar cablear la salida del pin TX1 del Arduino hacia el pin RxD del HC-05.

Se colocan de forma cruzada, pues el pin RX1 de Arduino va a recibir datos por medio del pin TxD del HC-05 y viceversa, el pin RxD del HC-05 va a recibir datos por medio del pin TX1 de Arduino. Esta última línea de comunicación es la que nos interesa por ahora, pues la comunicación será unidireccional, de Arduino al módulo Bluetooth, y éste va a retransmitir los datos hacia el dispositivo móvil.

No debemos olvidar colocar las tierras de ambos componentes en la tierra común del circuito fotopletismógrafo.

5.4. PE 3 - Programa en Arduino

5.4.1. Código

```

1 //BT RxD = TX1 Arduino
2 //BT TxD = RX1 Arduino
3 //BT GND = GND Arduino
4 //BT VCC = 5V Arduino
5 /*VARIABLES PARA LA LECTURA DE PULSOS*/
6 unsigned long tiempoAnt=0;
7 int tiempo=0;
8 int iniPeriodo=0;
9 int finPeriodo=0;
10 float periodo=0;
11 float pulsos;
12
13 void setup(){
14     // put your setup code here, to run once:
15     Serial1.begin(9600); //Bluetooth
16     Serial.begin(9600); //PC
17     pinMode(2, INPUT);
18     attachInterrupt(digitalPinToInterrupt(2), contar,RISING);
19 }
20
21 void loop(){
22     // put your main code here, to run repeatedly:
23     if(millis()-tiempoAnt>=1000) {
24         tiempo++;
25         tiempoAnt=millis();
26     }
27
28     if(tiempo == 2){
29         if(pulsos<120) {
30             Serial.print("Pulsaciones por minuto: ");
31             Serial.println((int)pulsos);
32             Serial1.println((int)pulsos);
33         }
34         tiempo=0;
35     }
36 }
37
38 void contar(){
39     finPeriodo=millis();
40     periodo=(float)(finPeriodo-iniPeriodo);
41     iniPeriodo=millis();
42     pulsos=60000/periodo; //1 minuto
43 }
```

5.4.2. Explicación del código

Según la señal que esta continuamente entrando en el Arduino, debemos ser capaces de contar los cambios de flanco de éste entre 0 y 5 V, pues eso representa un pulso cardíaco obtenido en la lectura, en un periodo de tiempo de un minuto.

Para ellos utilizamos interrupciones en Arduino, que son un mecanismo que permiten asociar una función a la ocurrencia de un determinado evento, en este caso, un pulso.

Cuando ocurre el evento el procesador “sale” inmediatamente del flujo normal del programa y ejecuta la función de interrupción asociada ignorando por completo cualquier otra tarea

(por esto se llama interrupción). Al finalizar la función de interrupción asociada, el procesador vuelve al flujo principal, en el mismo punto donde había sido interrumpido.

En esta caso, nuestra función de interrupción esta definida como *contar()*, y se ejecutará cada vez que hay un flanco de subida: **RISING** en el pin 2, que es la señal que sale del transistor.

Caqui leemos el tiempo anterior de que exista un flanco de subida, y leemos el tiempo posterior de este, que iremos guardando en variables globales. De éste modo, podemos calcular la duración del periodo total de cada flanco de subida, dividimos 60 segundos entre esta dato para obtener el número de pulsaciones.

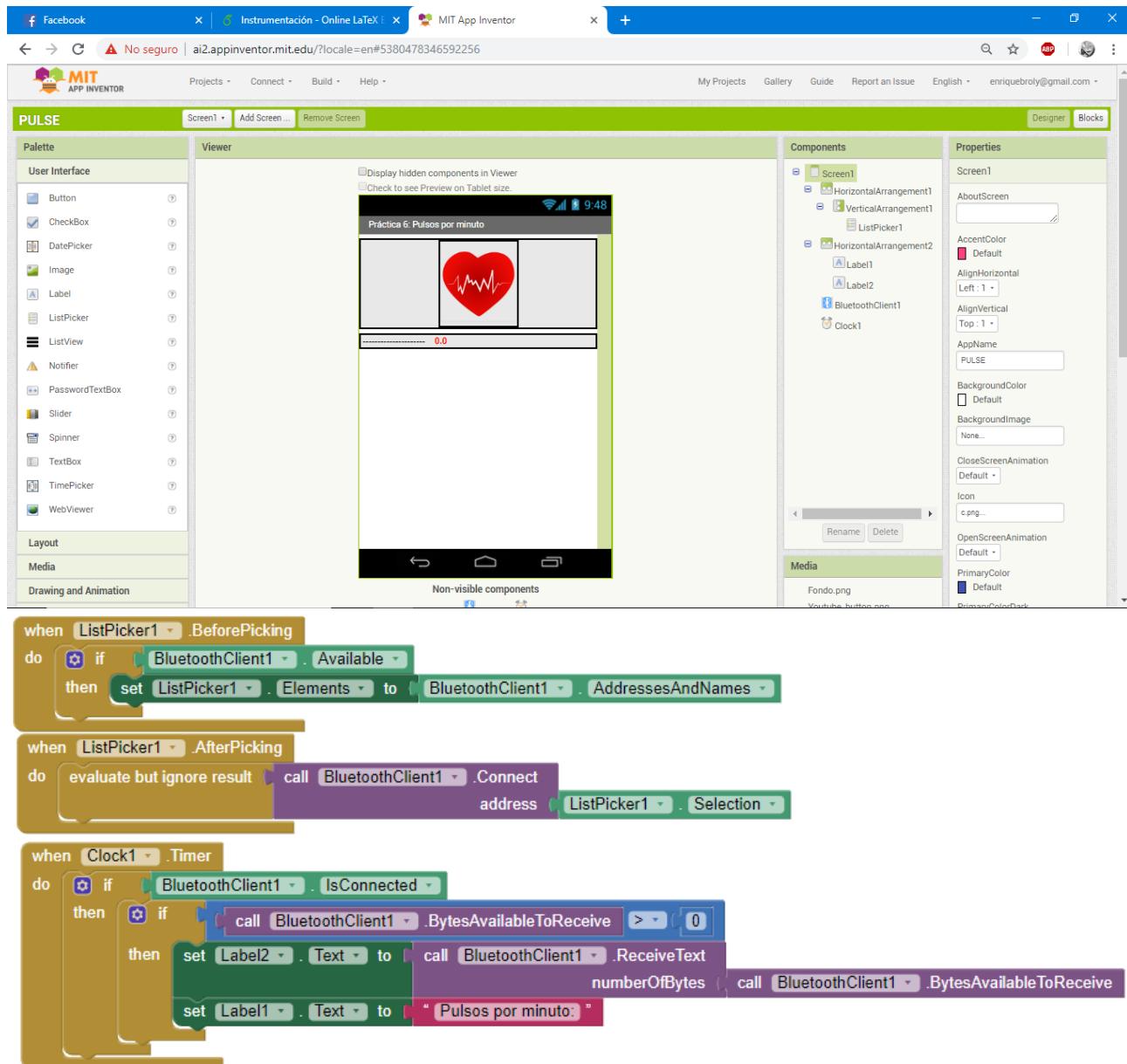
En el bloque *setup()* inicializamos dos canales de comunicación seriales: **Serial** para visualizar los datos en la computadora, y **Serial1**, que corresponde al canal que maneja el Arduino para la comunicación Bluetooth, **este es el que nos interesa**.

Y finalmente, definimos que el pin 2 es una entrada y que usaremos interrupciones con *attachInterrupt()*.

A su vez, en el bloque *loop()*, tenemos una condición que posee un contador de tiempo y nos ayuda a saber cuándo ha pasado un segundo. Debajo de ésta, hay otra más que verifica que hayan transcurrido dos segundos en la ejecución del programa para imprimir los pulsos por minuto; debemos imprimir éste dato únicamente cuando sea menor a 120, pues el promedio de pulsos cardíacos por minuto en el ser humano esta entre 50 y 100 pulsos.

5.5. PE 4 - Aplicación Android desarrollada para móviles

5.5.1. Código e interfaz gráfica



5.5.2. Explicación del código

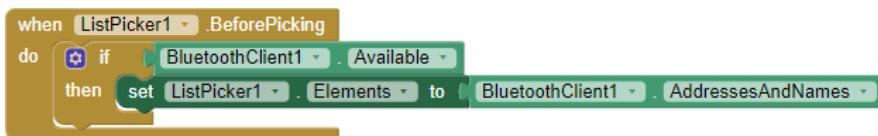
Para crear esta aplicación, utilizamos la página Web **MIT App Inventor**.

Primero creamos la interfaz gráfica, que consiste en un botón con una lista que contendrá todos los dispositivos Bluetooth emparejados que el dispositivo móvil posee (en donde se

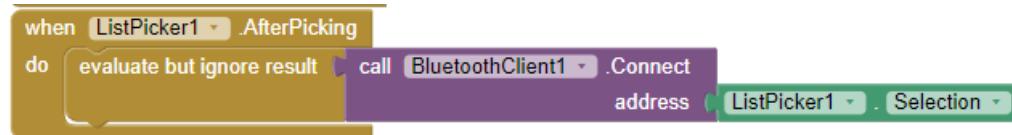
encuentra el ícono del corazón), y un cuadro de texto en donde se mostrarán los pulsos por minuto.

Además, contamos con dos elementos no visibles: un reloj para que nuestra aplicación esté sincronizada con el temporizador del Arduino y los datos se muestren en tiempo real, **este debe ser inicializado en un valor de 1000 milisegundos, o un segundo**, y el cliente de conexión Bluetooth.

Pasando a los bloques de código, primero le decimos a la aplicación que cada vez que se inicie, cargue todos los dispositivos Bluetooth emparejados al dispositivo móvil junto a su MAC.



Luego, decimos que al momento de seleccionar un dispositivo de la lista, la aplicación se conecte a él enviándole su dirección MAC por medio de la función *call BluetoothClient1.Connectaddress()*.



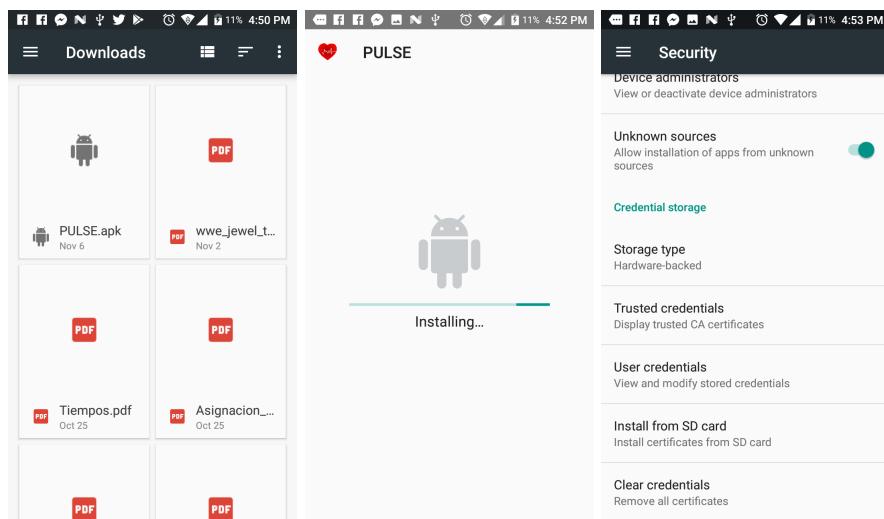
Por último, cada vez que haya un pulso de reloj en la aplicación de un segundo, verificar que el cliente Bluetooth este conectado para así llamar a la función *BytesAvailableToReceive()* y revisar que sí existan bytes que están siendo transmitidos. Si lo están, colocamos estos bytes en nuestros campos de texto con ayuda de la función *ReceiveTextnumberOfBytes()*, para ir leyendo la información desde el Arduino.



Ésta información será la variable pulsos que enviamos por el canal **Serial1** en el código de Arduino.

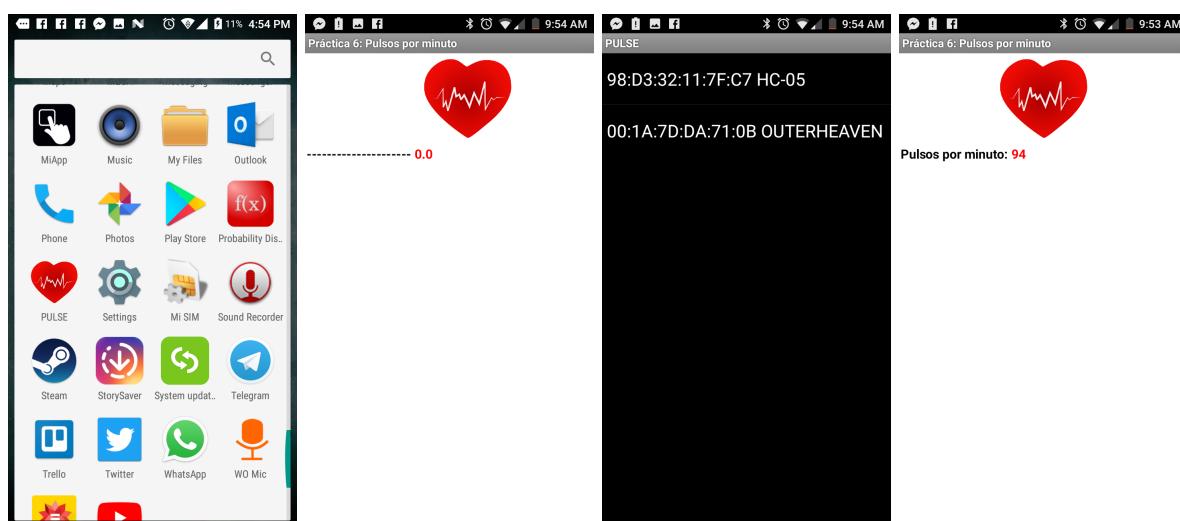
5.5.3. Instalación

1. Descargar e instalar el archivo "PULSE.apk" desde el móvil.

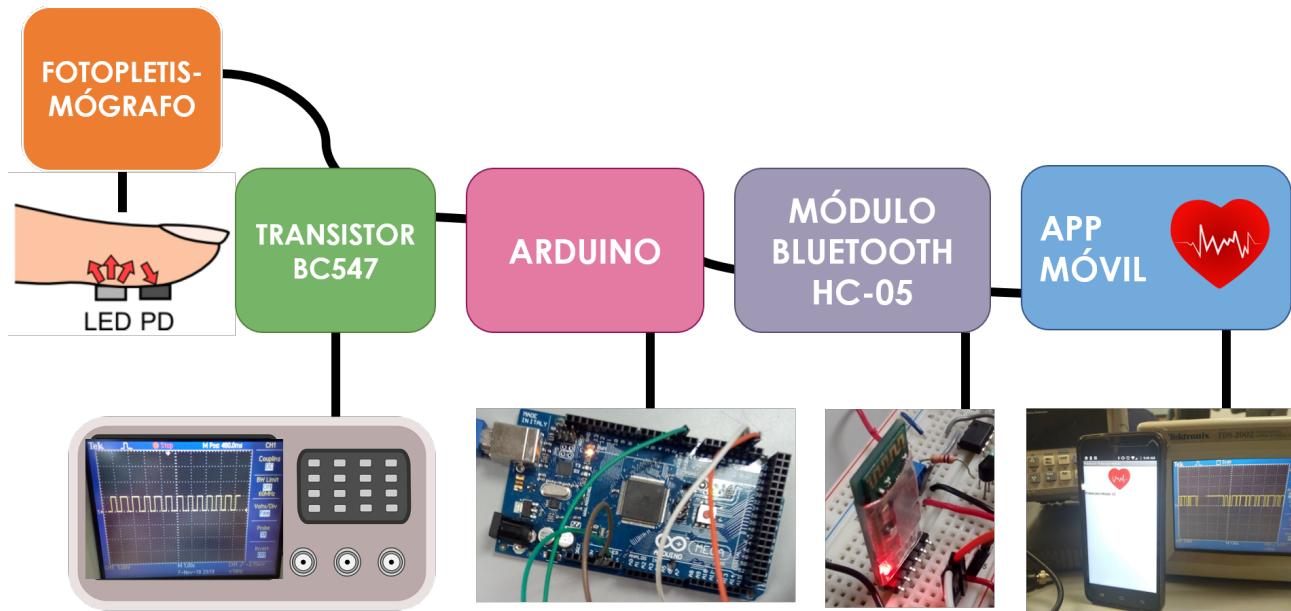


Importante: Activar la opción de instalas aplicaciones de origen desconocido desde Ajustes.

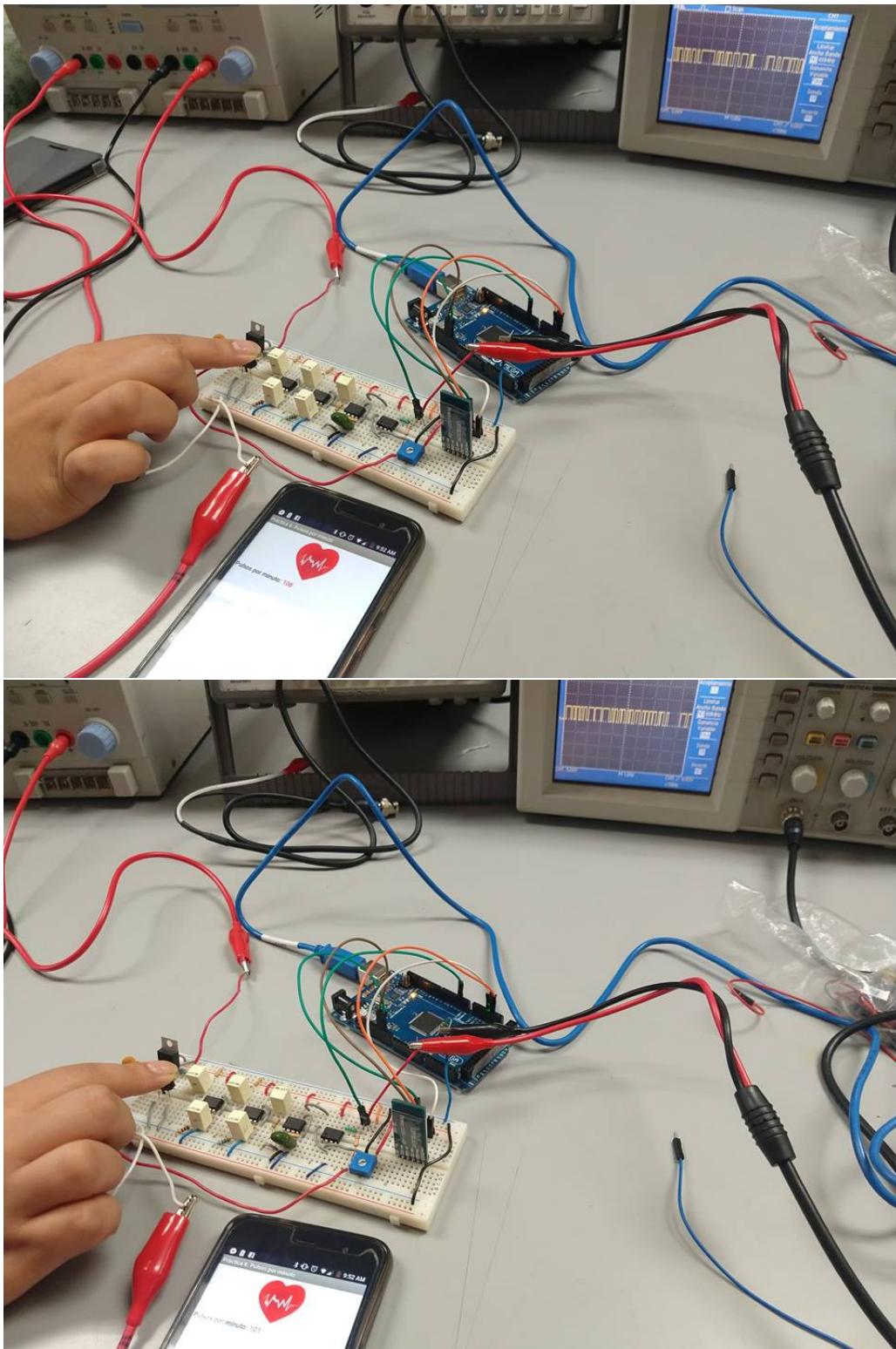
2. Activar el Bluetooth en el celular y emparejar el módulo HC-05 con el código "1234".
3. Entrar a la app.
4. Pulsar el ícono del corazón, aparecerá el listado de dispositivos Bluetooth, seleccionar el HC-05
5. La app comenzará a desplegar las pulsaciones por segundo.

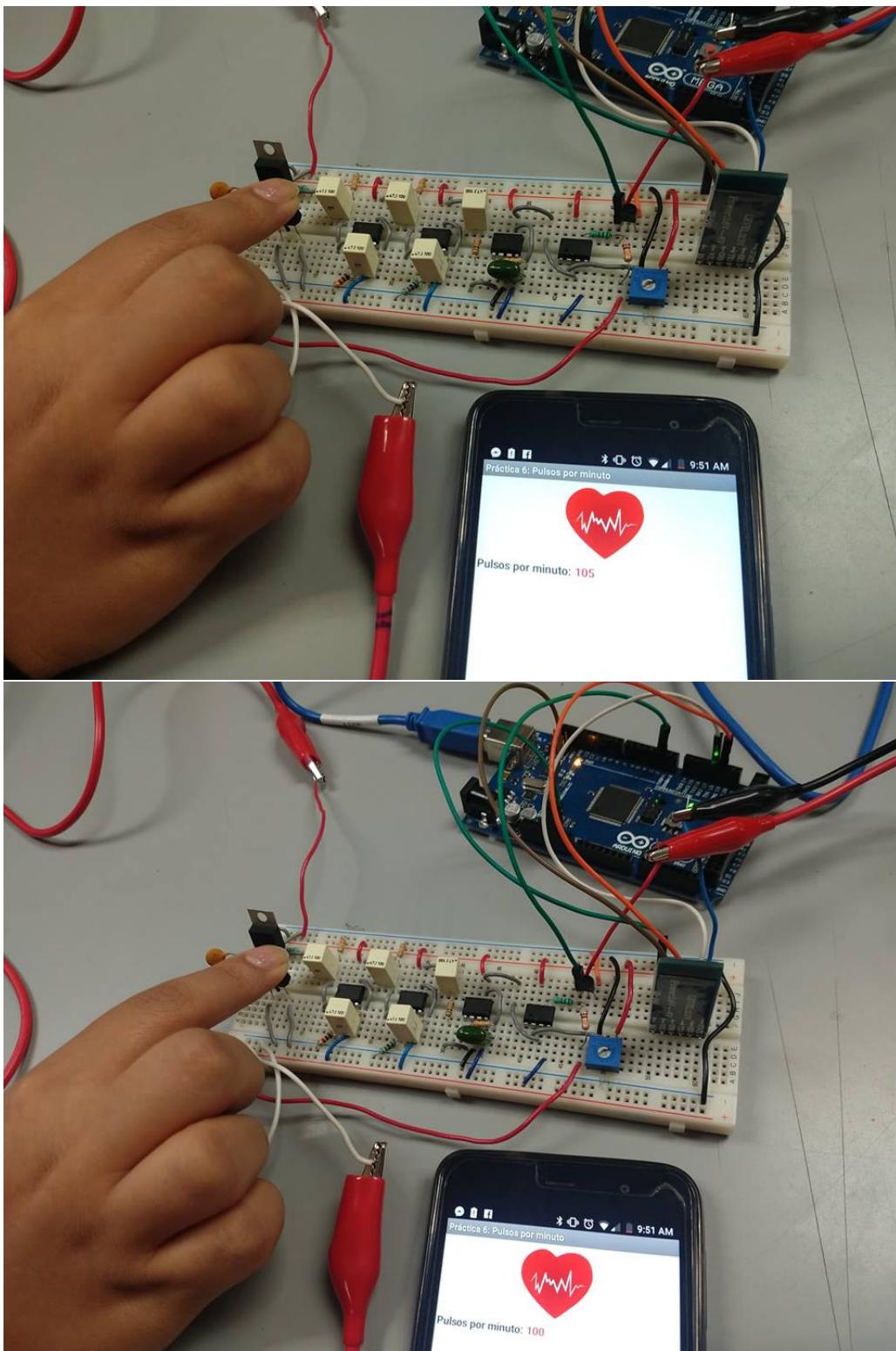


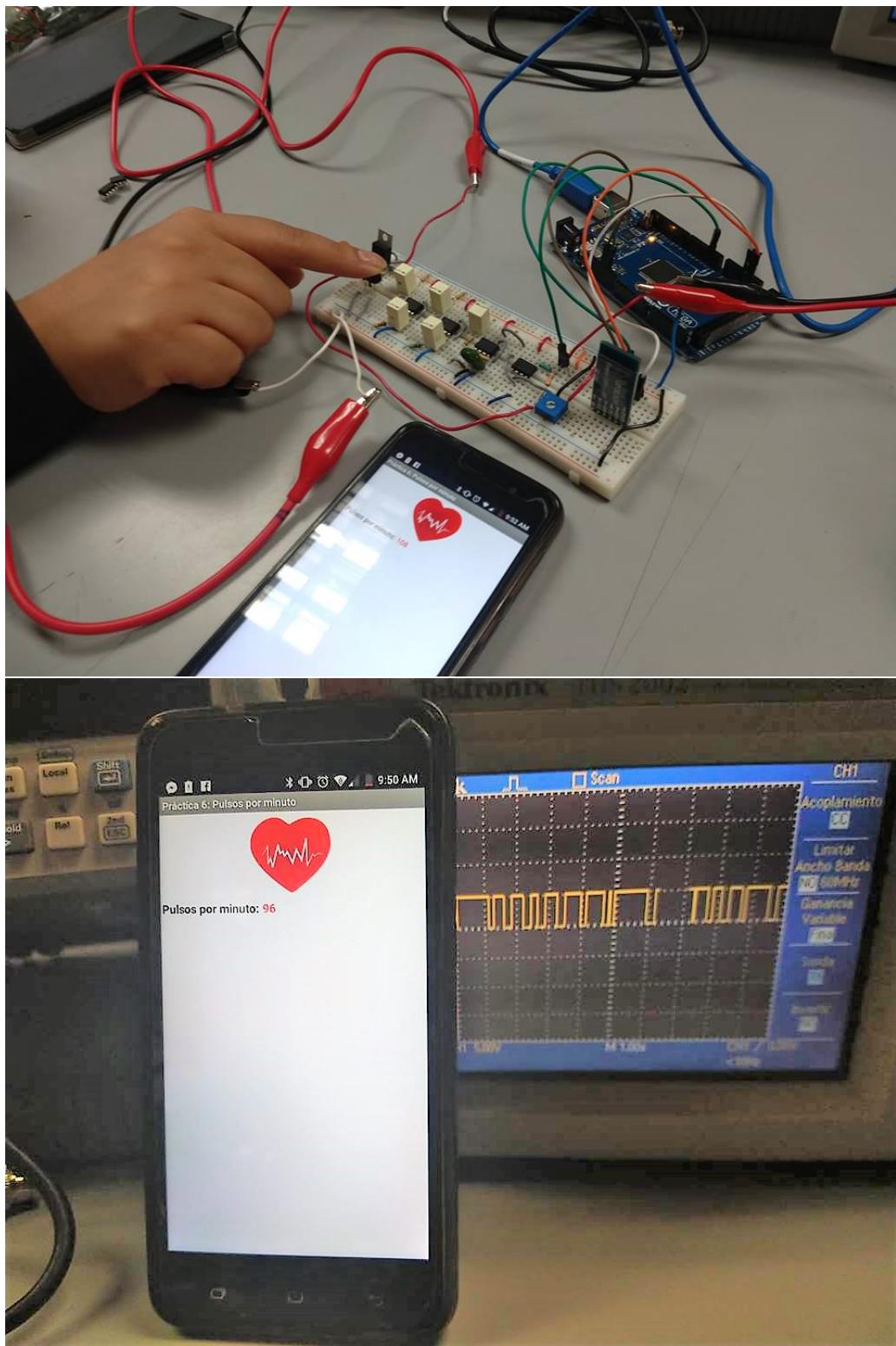
5.6. Funcionamiento completo



- Fotopletismografo: Entrega una señal que corresponde a los latidos cardíacos de una persona.
- Transistor BC547: Funciona como interruptor y limita la señal dentro de un rango de 0 a 5 V.
- Arduino: Calcula el periodo de cada pulso por minuto con ayuda de interrupciones y envía la cantidad de pulsos por minuto vía Bluetooth.
- Módulo Bluetooth HC-05: Retransmite la información recibida a un dispositivo móvil.
- App Móvil: Se conecta al HC-05 y despliega el número de pulsos por minuto en tiempo real.







6. Observaciones

- ✓ En los 2 vídeos anexados a este reporte se puede visualizar a mayor detalle el comportamiento y funcionamiento del circuito. Link: <https://drive.google.com/drive/folders/1JuevvKissNiwqsKaR3CjCS-S7fGidH1D?usp=sharing>
- ✓ También anexamos los códigos tanto del programa de Arduino como de la aplicación desarrollada en MIT App Inventor y su APK. Link: <https://drive.google.com/drive/folders/1RQ2Xoht1R3sdPO-IiZWNO9LnZcn1Rg9G?usp=sharing>
- ✓ A veces los valores de pulsaciones por segundo están por debajo de los 50 pulsos o encima de los 100, esto es debido a una mala colocación del dedo en el sensor TCRT5000L, provocando que entre ruido indeseado en la señal, debido a los cambios de luz y sombras, y el Arduino los detecte como pulsos. Basta con colocar el dedo firmemente y relajarlo en el sensor, sin presionar mucho pero también evitando levantarla.
- ✓ Al momento de cargar el programa de Arduino a la placa, es preferible que ningún cable o pines estén conectados al circuito o al HC05, sino hasta que finalice la carga.

6.1. Aguilar Herrera Arianna Itzamina

Esta práctica fue, realmente, muy corta, ya que se tomó el funcionamiento de la práctica anterior (práctica 5) y se combinó con el módulo de Bluetooth, para poder mandar la frecuencia cardíaca obtenida mediante el fotopletismógrafo, y enviarla a una aplicación a, en este caso, un teléfono celular, por lo que, podemos concluir que el uso del módulo Bluetooth es muy útil para darle una mejor aplicación a nuestros instrumentos, ya que, mediante Arduino (en este caso se usó Arduino, pero hay otros lenguajes con los que podría funcionar, aunque Arduino es recomendable por su similitud con C) y Bluetooth podemos enviar nuestra variable medida y manipularla a nuestra conveniencia.

6.2. Ramos Diaz Enrique

El principal reto de ésta práctica no fue el armado o acondicionamiento del sensor y la variable que queríamos medir, pues eso se realizó en la práctica anterior, sino la correcta interpretación de ésta, así como la comunicación entre Arduino y una aplicación móvil Android por medio de Bluetooth.

Al principio se intentó, por medio de Arduino, leer la cantidad de ceros lógicos (que corresponden a cuando se detecta un pulso) en un periodo de diez segundos y luego multiplicarlo por seis para obtener los pulsos por minuto, pero no tomamos en cuenta lo que realmente importaban era la duración del periodo completo los flancos de subida de la señal cuadrada. Utilizando interrupciones en Arduino, logramos calcular la frecuencia de cada pulso en un minuto.

El siguiente reto era desarrollar la aplicación móvil en donde desplegaremos los resultados. Con ayuda de la página web MIT App Inventor, logramos realizarla de forma muy fácil.

Finalmente quedaba conectar todo y sincronizar el dispositivo móvil con el módulo Bluetooth HC-05, para desplegar los pulsos por minuto que entregaba el fotopletismógrafo, interpretados y enviados por el Arduino.

6.3. Nicolas Sayago Abigail

Debido a que el circuito que construimos en la práctica anterior se utilizó, esta práctica fue más sencilla, el problema aquí fue comunicar la señal que nos generaba el sensor con la computadora.

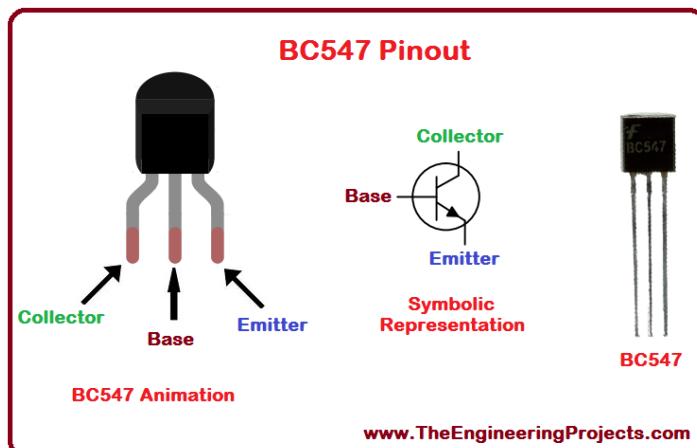
En Arduino existen dos formas de comunicar dependiendo la señal, ya sea analógica o digital, primero nos equivocamos e inicialmente lo hicimos con señal analógica pero a parte de que lleva más líneas de código, si no es conectado correctamente puede causar fallas, corregimos y usamos el código para la señal digital y así fue más fácil y correcto.

La implementación del código paso por varias partes, primero lo intentamos con el delay de Arduino, pero se mejoró con Interrupciones.

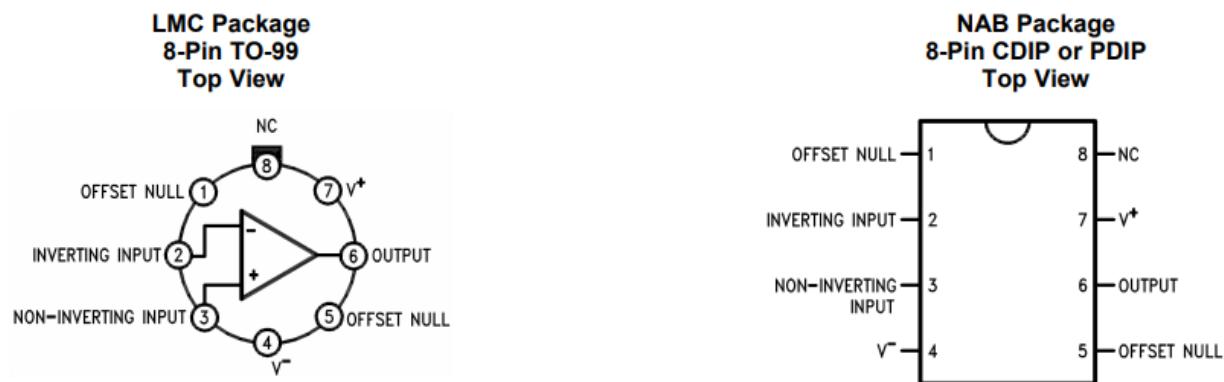
7. Anexos

7.1. BC547

Es un transistor amplificador de audio y VHF Freq. Driver con una corriente máxima de colector de 0.6 A, en su composición posee una placa de semiconductor con tres regiones consecutivas de diferente conductibilidad eléctrica las cuales forman dos uniones NPN, las dos regiones extremas tienen un mismo tipo de conductibilidad, la intermedia, conductibilidad de otro tipo, estas son llamadas emisor, colector y base.



7.2. LM741

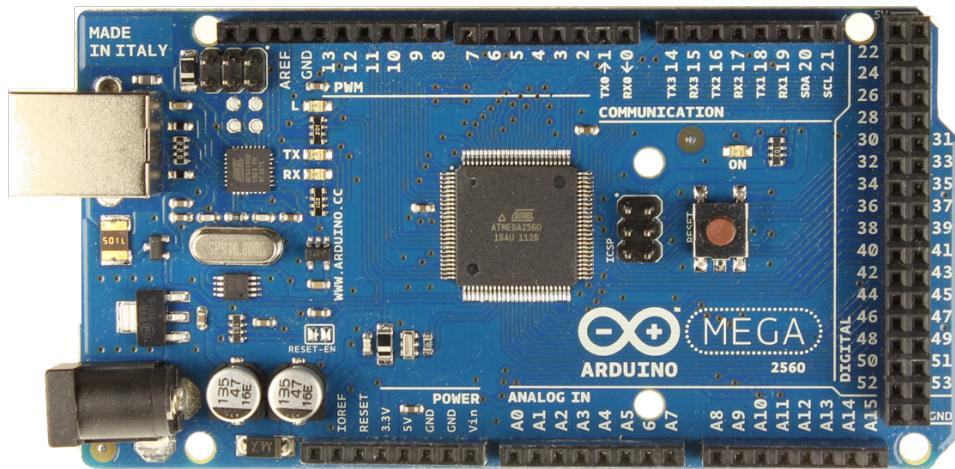


LM741H is available per JM38510/10101

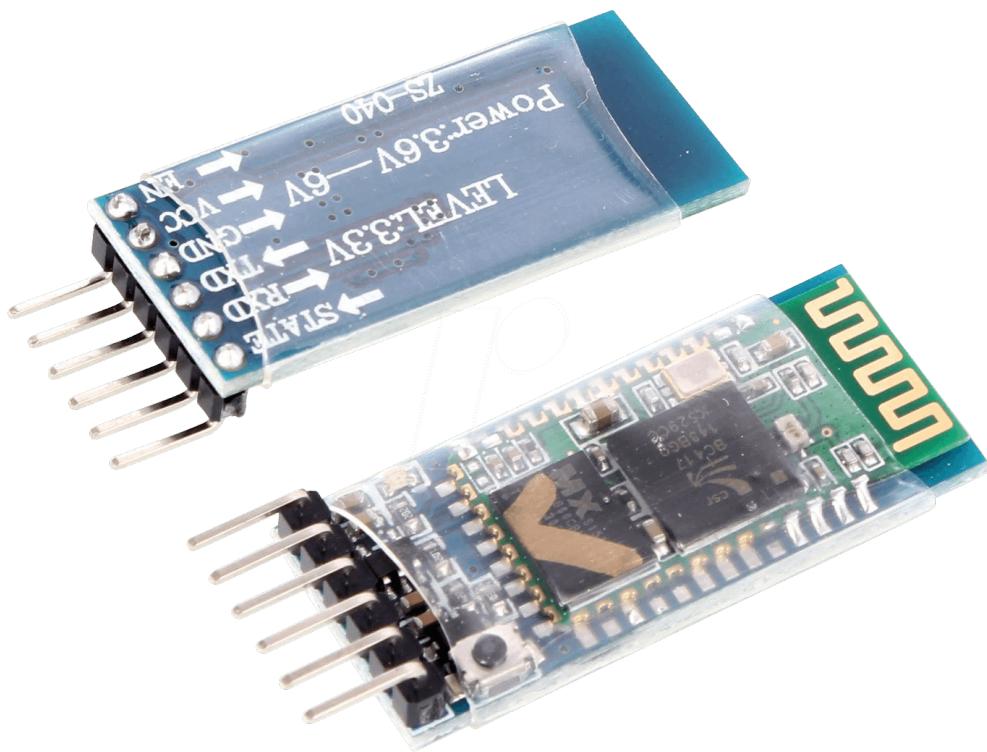
El amplificador operacional LM741 debe estar alimentados por una fuente de +12 V y -12 V en serie. [Consultar hoja de especificaciones del LM741].



7.3. Arduino Mega



7.4. HC-05

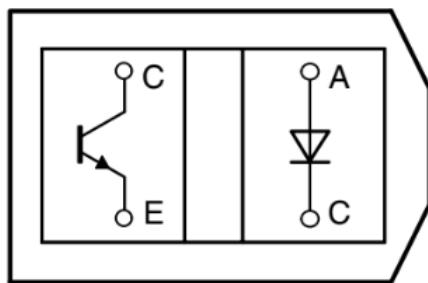


7.5. Sensor TCRT5000L

El TCRT5000 y TCRT5000L son sensores reflectivos que incluyen un emisor de infrarrojos y un fototransistor en un paquete de plomo que bloquea la luz visible. El paquete incluye dos clips de montaje. TCRT5000L es la versión de plomo más larga. Debe alimentarse con 5 V.



19156_2



Top view

19156_1

7.6. Página Web: MIT App Inventor

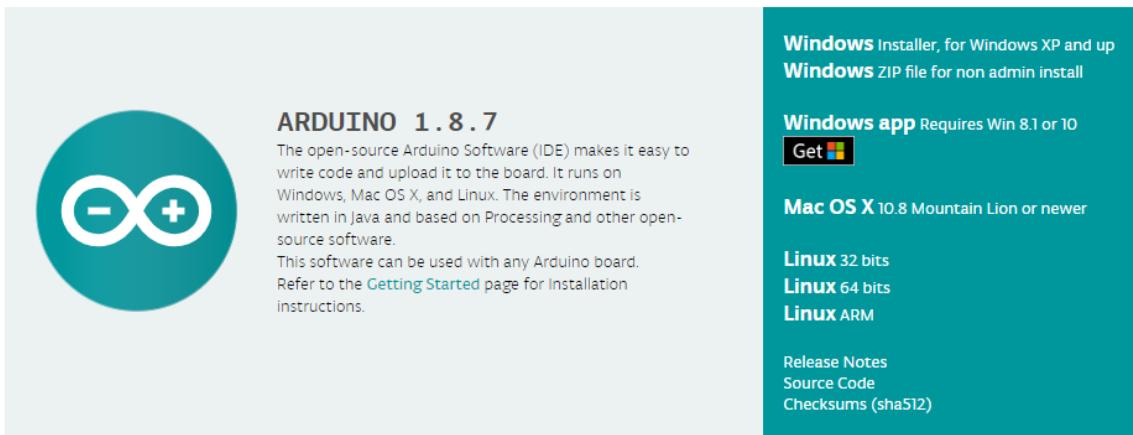


Para registrarnos y comenzar a crear aplicaciones, únicamente necesitamos una cuenta de Google.

Link: <http://ai2.appinventor.mit.edu/>

7.7. Arduino IDE

Download the Arduino IDE



Link de descarga: <https://www.arduino.cc/en/Main/Software>

Referencias

- [1] Y. FM, *Qué es Arduino, cómo funciona y qué puedes hacer con uno*. [Online]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- [2] masadelante, *¿Qué significa Bluetooth? - Definición de Bluetooth*. [Online]. Available: <http://www.masadelante.com/faqs/que-es-bluetooth>
- [3] M. Capacitación, *Señales Tx y Rx*. [Online]. Available: <http://cursos.mcilelectronics.cl/señales-tx-y-rx-1/>
- [4] Inventable, *EL TRANSISTOR BC548*. [Online]. Available: <https://www.inventable.eu/2017/07/09/transistor-bc548/>
- [5] L. Llamas, *QUÉ SON Y CÓMO USAR INTERRUPCIONES EN ARDUINO*. [Online]. Available: <https://www.luisllamas.es/que-son-y-como-usar-interrupciones-en-arduino/>