



# **DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS**

## **Sistemas Operativos Avanzados Sistemas Embebidos (IoT) + Android “Silla inteligente”**

Comisión: Martes Noche

Ciclo Lectivo: Segundo Cuatrimestre - Año 2018

Docentes:

- Lic. De Luca Graciela
- Ing. Valente Waldo
- Ing. Carnuccio Esteban
- Ing. Volker Mariano
- Ing. Garcia Gerardo

Alumnos:

- |                     |                 |
|---------------------|-----------------|
| • Andres, Martin    | DNI: 39.626.806 |
| • Ajaya, William    | DNI: 37.668.091 |
| • Serrano, Alex     | DNI: 37.860.875 |
| • Trejo Ian         | DNI: 38.700.240 |
| • Velazquez, Julian | DNI: 35.991.088 |

# Índice

## Contenido

<b>PROYECTO: SILLA INTELIGENTE “Smart Chair”</b>	<b>3</b>
<b>Objetivo:</b>	<b>3</b>
<b>Introducción:</b>	<b>3</b>
<b>Desarrollo:</b>	<b>3</b>
<b>Descripción del Entorno:</b>	<b>4</b>
<b>Funcionamiento general:</b>	<b>4</b>
<b>Pasos para la creación del script en arduino:</b>	<b>4</b>
<b>Funcionamiento del script Arduino:</b>	<b>5</b>
<b>Pasos para la creación de la App SillaInteligente con Android Studio:</b>	<b>6</b>
<b>Funcionamiento de la APK SillaInteligente:</b>	<b>6</b>
<b>Software empleado:</b>	<b>9</b>
<b>Detalles técnicos de los sensores utilizados:</b>	<b>9</b>
<b>Detalles técnicos de los actuadores utilizados:</b>	<b>15</b>
<b>Conexión de los sensores utilizados:</b>	<b>16</b>
<b>Conexión de los actuadores utilizados:</b>	<b>23</b>
<b>Alcance del sistema:</b>	<b>25</b>
<b>Arduino uno:</b>	<b>25</b>
<b>PWM:</b>	<b>27</b>
<b>Diseño:</b>	<b>28</b>
<b>Prototipo:</b>	<b>32</b>

# PROYECTO: SILLA INTELIGENTE

## “Smart Chair”

### Objetivo:

Durante la cursada se planteó como objetivo la creación de un sistema embebido que funcione en conjunto con un aplicativo en Android, tal como se implementaría en un sistema de internet de las cosas.

Para esto, desarrollamos un sistema embebido (SE) que se trata de una silla inteligente.

### Introducción:

En este informe se pretende explicar nuestra experiencia utilizando sistemas embebidos, el uso de sensores y actuadores, pruebas en protoboard, uso de la placa Arduino, su comunicación a través de una conexión Bluetooth al SO Android, entre otros. A lo largo de este informe expondremos cómo y con qué elementos desarrollamos nuestro sistema.

### Desarrollo:

El problema a resolver es la creación de una silla inteligente que pueda proveer al usuario de la siguiente información:

- Información sobre la postura del usuario al momento de estar sentado sobre la silla. Si el usuario se encuentra con una postura que lo pueda perjudicar, el sistema lo informará mediante los datos recibidos por un sensor de ultrasonido.
- Información sobre la temperatura y la humedad en el ambiente, comprobados por un sensor de temperatura y humedad.
- Información de cuánto tiempo está sentado el usuario en la silla. Cuando el usuario se siente, la silla tiene un pulsador que le va a indicar al sistema que el usuario tomó asiento.

Además, el sistema va contar con:

- Un cooler, que se va a activar si la temperatura del ambiente es demasiado alta.
- Un conjunto de luces led que se va a encender cuando el sensor de luminosidad detecte a través de una barrera óptica un nivel bajo por debajo del umbral.

## Descripción del entorno:

Esto se podrá utilizar con la tecnología adecuada. Para cada una de estas funcionalidades el sistema contará con los siguientes componentes:

### **Actuadores:**

- Cooler: 120mm PWM 2200rpm
- Led: 10 leds 5mm difuso azul.
- Buzzer: alarma sonora.

### **Sensores:**

- Sensor de temperatura: ds18b20 cable sumergible Arduino.
- Sensor de humedad y temperatura: Dht22 pcb.
- Sensor ultrasónico: hc-sr04 Arduino.
- Sensor de luz ambiental: breakout temt6000 Arduino.
- Pulsador: dip tact switch 6x6x5mm Arduino.
- Potenciometro: 5k.

### **Enlace:**

- Bluetooth: HC-05 maestro esclavo uart Arduino.

### **Otros componentes:**

- Cable USB-Serie.
- Cable macho/hembra y macho/macho.
- Placa Arduino Uno.
- Fuente 12v 0,5a.
- Resistencia 330 ohm, 1 Kohm, 4 Kohm.
- Transistor.
- Protoboard 830 puntos.

## Funcionamiento general:

La silla inteligente comenzará a funcionar y a analizar la información, cuando el usuario se siente, ya que se oprimira el pulsador y sonará el buzzer indicandolo. Se le comenzará a contabilizar el tiempo que esté sentado al usuario, además de corroborar que esté con una buena postura. Se medirá la temperatura y humedad del ambiente para, en caso de que haya una temperatura elevada se enciende el cooler para refrigerar al usuario. También contará con un sensor de luminosidad, que cuando haya poca luz, la silla automáticamente encienda los leds para mejor visión del usuario.

## Pasos para la creación del script en arduino:

Se desarrolló un programa en la computadora, estableciendo por código la forma de capturar los datos de los sensores y el accionar de los actuadores como describimos en la etapa de "FUNCIONAMIENTO." Usamos el concepto de cross/compiler; o sea, se compiló el

programa en la computadora y se pasó el script a la placa arduino. Para eso, previamente tuvimos que conectar la computadora al arduino mediante puerto USB-Serie, y también conectamos los sensores y actuadores mediante los pines del arduino.

## Funcionamiento del script Arduino:

Para nuestro script contamos con los sensores digitales como el pulsador, de temperatura, temperatura y humedad y ultrasonido. Por un lado el pulsador, cuando es presionado, indica el comienzo del script y activa un timer que va a contar el tiempo que lleve sentado el usuario. Luego tenemos el sensor de temperatura y humedad que nos va a indicar los valores de la temperatura y humedad del usuario. También está el sensor de temperatura que medirá la temperatura del ambiente. Y por último el sensor de ultrasonido que medirá la distancia entre la espalda del usuario y el respaldo de la silla, para indicar si este está sentado con una postura correcta o incorrecta.

Por otro lado tenemos los sensores analogicos los cuales son, el de luz ambiental y el potenciómetro. Por un lado el sensor de luz se encargará de informar si el nivel de luz es bajo o normal en el ambiente. Por otro el potenciómetro lo utilizaremos para regular el nivel de intensidad de la luz emitida por los diodo led.

Otros componentes que utilizaremos son los actuadores como el cooler, el buzzer y el diodo led. El cooler va a funcionar en conjunto con el sensor de temperatura, ya que cuando este detecte una temperatura muy elevada, el cooler se encenderá. El buzzer en cambio, funcionara en conjunto con el pulsador, cuando este sea presionado, luego de que el usuario se siente en la silla, este emitirá un pitido, indicando que el sistema ya comenzó a funcionar. Por último tenemos los diodo led, que funcionaran en conjunto con el sensor de luz ambiental, cuando el sensor detecte un bajo nivel de luz, los leds se encenderán automáticamente.

Además de esto contamos con un módulo bluetooth HC-05, configurandolo mediante comandos AT (el mismo tiene la posibilidad de hacerlo funcionar tanto en modo maestro como esclavo que es como queda configurado) conectado a la placa Arduino, que se queda esperando peticiones de conexiones de otros dispositivos. Cuando detecta que un dispositivo móvil con Android se conectó, dicha placa Arduino le envía los datos de temperatura, del ultrasonido la distancia de la espalda con la silla, de humedad previamente solicitado por el celular.

### **Protocolo:**

ATE1:ardu temp good  
ATE2:ardu temp reg  
ATE3:ardu temp bad

AUS1: ardu US good  
AUS2: ardu US Bad

ATH1: ardu temp hum good  
ATH2: ardu temp hum reg  
ATH3: ardu temp hum bad

ALU1 : ardu luz good  
ALU2 : ardu luz reg  
ALU3 : ardu luz bad

APO1 : ardu pote good  
APO2 : ardu pote reg  
APO3 : ardu pote bad

APU1 : ardu Pulsa High  
APU0 : ardu Pulsa Low

ALE1: ardu Led High  
ALE0: ardu Led Low

MSO1 : mobi sonido high  
MSO0 : mobi sonido Low

MLU1 : mobi luz high  
MLU0 : mobi luz low

MCO1 : mobi cooler high  
MCO0 : mobil cooler low

MTS1 : mobil tiempo high

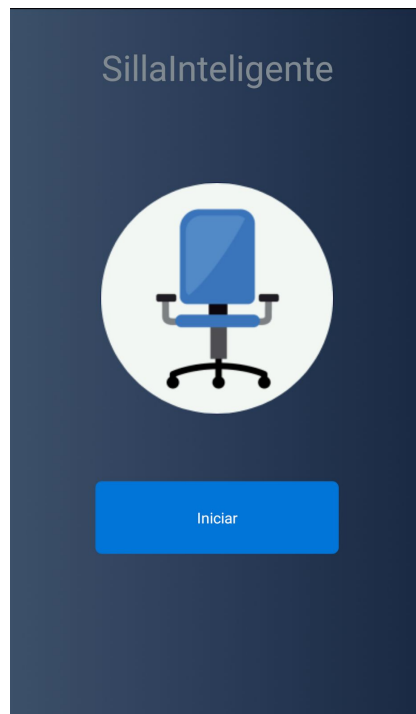
PW00-99 : intensidad luz

## Pasos para la creación de la App SillaInteligente con Android Studio:

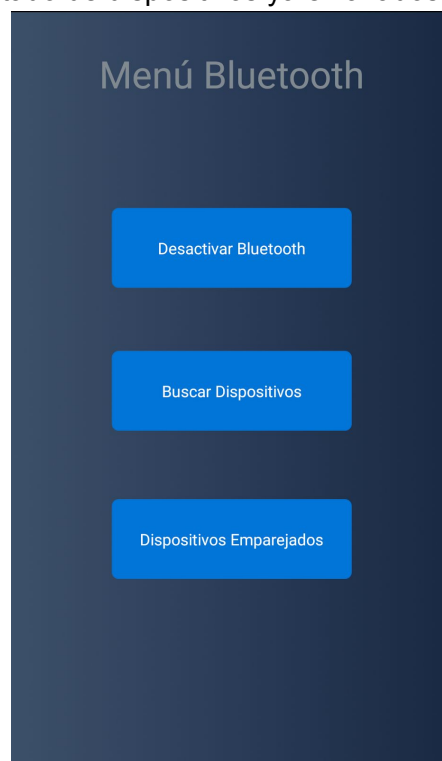
Se usó el Android Studio version 3.2.1 para crear la aplicación, donde se estableció las diferentes activities y las views que componen cada activity. Y por medio de cross/compiler se compiló el programa y se instaló la APK SillaInteligente en el dispositivo móvil para sistemas operativos android a partir de la versión 4.0.

## Funcionamiento de la APK SillaInteligente:

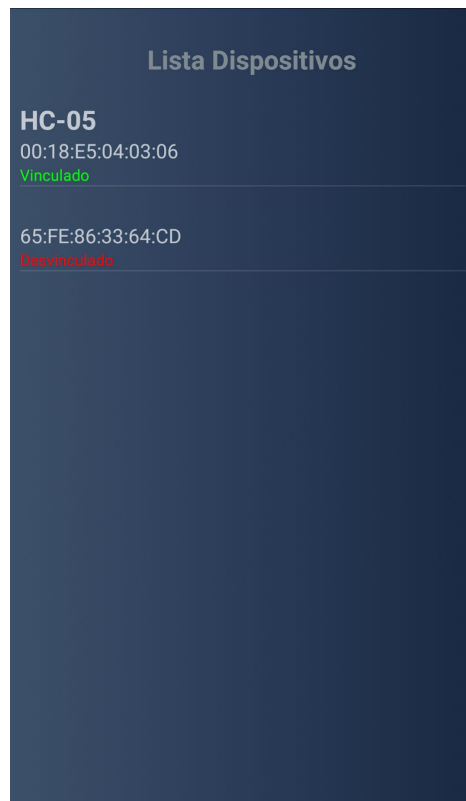
Al iniciar la aplicación se le brinda al usuario la opción de ingresar al menú de Bluetooth.



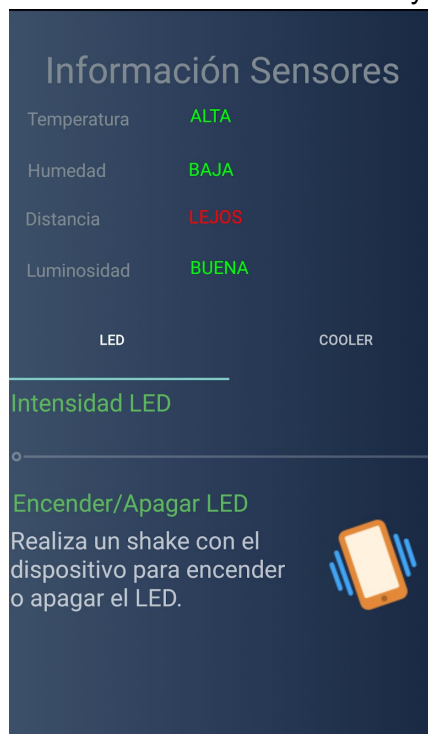
Allí se debe encender el Bluetooth del dispositivo para habilitar la búsqueda de nuevos dispositivos a enlazar o el listado de dispositivos ya enlazados.



En cualquiera de las dos opciones que se habilitan, aparece una pantalla con el listado de los dispositivos, al presionar sobre uno, comienza el enlace de dispositivos.



Automáticamente se abre una nueva actividad con los controles para el manejo de la silla, allí se muestra la información recibida y las acciones que se pueden tomar.





## **Bluetooth:**

La comunicación de entrada y salida es por medio de Bluetooth.

La comunicación por Bluetooth se realiza utilizando las clases BluetoothDevice, BluetoothManager, BluetoothAdapter y BluetoothSocket.

Los mensajes son enviados mediante dos Threads (uno que se encarga de enviar las acciones a la silla y otro que se encarga de escuchar la información proveniente de la silla).

## **Sensores:**

La función de control remoto se realiza tanto por medio de la interfaz gráfica como por medio de gestos/acciones realizadas en el teléfono.

Estos gestos/acciones especiales envían acciones a la silla, los gestos son:

- Shake: envía la acción de encender el LED
- Rotación: envía la acción de iniciar el funcionamiento del Cooler
- Sensor de Luz: lanza notificaciones en el dispositivo para incentivar la interacción con el embebido.

Los gestos son implementados haciendo uso de los sensores existentes en el dispositivo Android, los sensores utilizados son:

- Acelerómetro
- Giroscopio
- Sensor de Luz

Para lograr la implementación de gestos se utilizó las clases Sensor, SensorEvent, SensorManager y la interfaz SensorEventListener del paquete android.hardware.

- La aplicación de android toma datos de los sensores del dispositivo
- La aplicación no toma datos de sensores no utilizados del dispositivo

## **Software empleado:**

### **Sistema Embebido**

- IDE: Arduino

### **Aplicación Android**

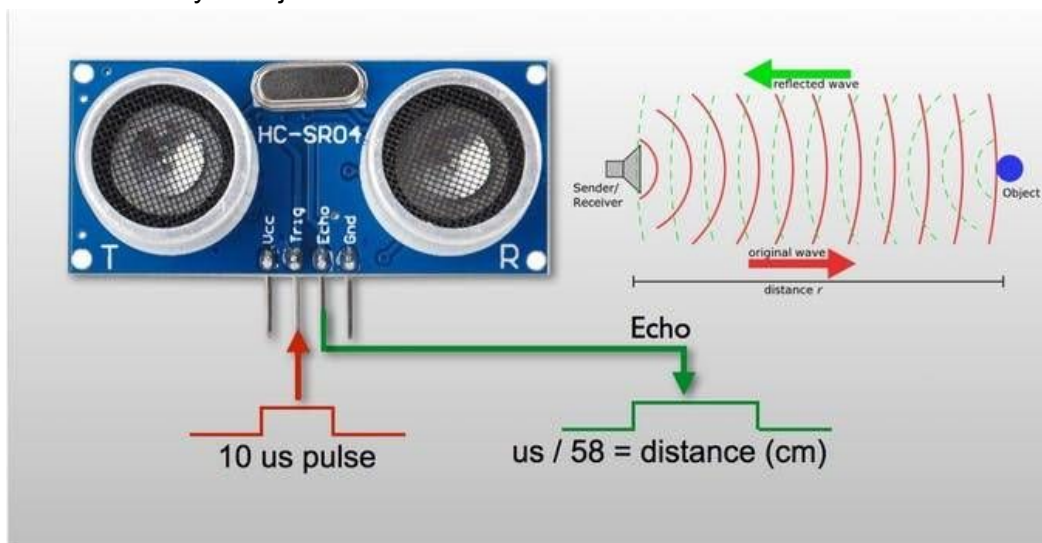
- IDE: Android Studio

## **Detalles técnicos de los sensores utilizados:**

- **Sensor ultrasónico HC-SR04**

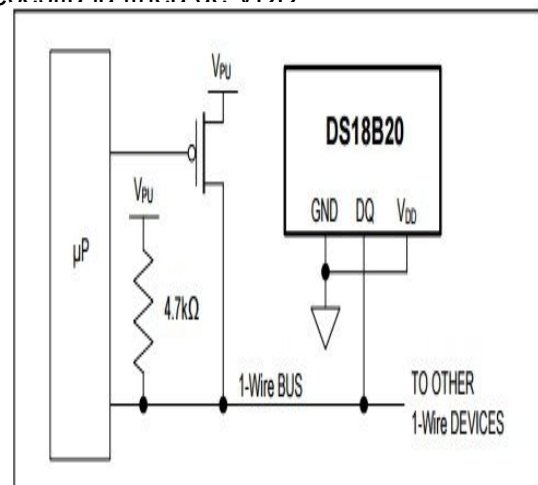
El sensor HC-SR04 es un módulo que incorpora un par de transductores de ultrasonido que se utilizan de manera conjunta para determinar la distancia del sensor con un objeto colocado enfrente de este. Un transductor emite una “ráfaga” de ultrasonido y el otro capta el rebote de dicha onda. El tiempo que tarda la onda sonora en ir y regresar a

un objeto puede utilizarse para conocer la distancia que existe entre el origen del sonido y el objeto.



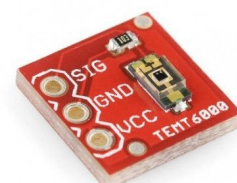
- **Sensor digital temperatura DS18B20**

El sensor internamente obtiene energía del pin de datos cuando este se encuentra en un estado alto y almacena carga en un condensador para cuando la línea de datos esté en un estado bajo, a esta forma de obtener energía se le llama “Parasite Power” y se usa cuando el sensor debe conectarse a grandes distancias o en donde el espacio es limitado, puesto que de esta forma no se necesita la línea de VDD



- **Sensor de luz ambiental TEMP6000**

El sensor luz ambiental TEMT6000 es sensible al espectro visible. La salida de este sensor es analógica por lo que lo que se puede medir con cualquier ADC de un Arduino o microcontrolador. Por lo tanto, el módulo puede ser utilizado como un excelente contacto controlado por luz solar. La domótica podría ser una opción para utilizarlo. En este caso la luz solar será la variable a medir con el



módulo, cuando el módulo no detecte la suficiente luz solar se emitirá una señal de al Arduino para encender el LED o en caso de lo contrario, que la luz solar sea suficientemente fuerte se emitirá otra señal que indique al Arduino que apague el LED con la medición del TMT6000.

## ● **Modulo bluetooth HC-05**

El módulo Bluetooth HC-05 es ideal para utilizar en todo tipo de proyectos donde necesites una conexión inalámbrica fiable y sencilla de utilizar. Se configura mediante comandos AT y tiene la posibilidad de hacerlo funcionar tanto en modo maestro como esclavo.

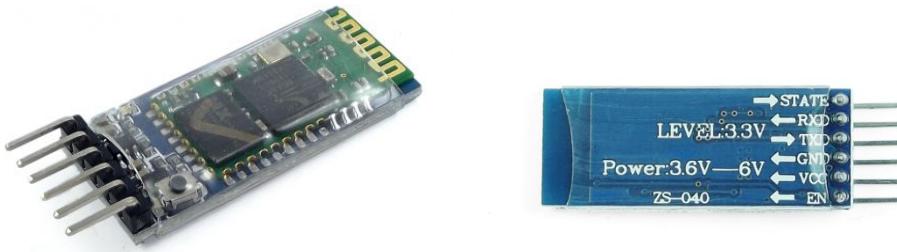
El módulo Bluetooth HC-05 puede alimentarse con una tensión de entre 3.3 y 6V (normalmente 5V), pero los pines TX y RX utilizan niveles de 3,3V por lo que no se puede conectar directamente a placas de 5V. Debes utilizar dos pequeñas resistencias como divisor de tensión para que el módulo no se estropee. En las imágenes del producto podrás ver el esquema de conexión. También dispone de un pulsador para entrar en modo comandos, aunque también lo puedes hacer por software utilizando el pin EN.

Tiene un LED incorporado que indica el estado de la conexión y si está emparejado o no en función de la velocidad del parpadeo.

### **Características:**

- Protocolo Bluetooth: v1.1 / 2.0.
- Frecuencia: banda ISM de 2,4 GHz.
- Modulación: GFSK
- Potencia de transmisión: menos de 4dBm, Clase 2.
- Sensibilidad: Menos de -84dBm en el 0,1% BER.
- Ratio asíncronos: 2.1Mbps (Max) / 160 kbps.
- Sincrónico: 1Mbps / 1Mbps.
- Perfiles de la ayuda: puerto serie Bluetooth (maestro y esclavo).
- Fuente de alimentación: + 3.3VDC 50mA. (soporta de 3.3 a 6V)
- Temperatura de trabajo: -5 ° C a 45 ° C.

Existen varios modelos y versiones para el módulo HC-05, el que usaremos es el que se muestra en las siguientes imágenes, que como vemos tiene un pulsador, el que nos servirá para entrar en Modo AT y configurarlo.



El módulo Bluetooth HC-05 viene configurado de fábrica como Esclavo, pero se puede cambiar para que trabaje como maestro, además al igual que el hc-06, se puede cambiar el nombre, código de vinculación velocidad y otros parámetros más.

Definamos primero que es un dispositivo bluetooth maestro y dispositivo esclavo:

#### Modulo bluetooth hc-05 como esclavo:

Cuando está configurado de esta forma, se comporta similar a un HC-06, espera que un dispositivo bluetooth maestro se conecte a este, generalmente se utiliza cuando se necesita comunicarse con una PC o Celular, pues estos se comportan como dispositivos maestros.

#### Modulo bluetooth hc-05 como Maestro:

En este modo, EL HC-05 es el que inicia la conexión. Un dispositivo maestro solo se puede conectarse con un dispositivo esclavo. Generalmente se utiliza este modo para comunicarse entre módulos bluetooth. Pero es necesario antes especificar con que dispositivo se tiene que comunicar, esto se explicará más adelante.

El módulo HC-05 viene por defecto configurado de la siguiente forma:

- Modo o role: Esclavo.
- Nombre por defecto: HC-05.
- Código de emparejamiento por defecto: 1234.
- La velocidad por defecto (baud rate): 9600.

El Módulo HC-05 tiene 4 estados los cuales es importante conocer:

#### Estado Desconectado:

- Entra a este estado tan pronto se alimenta el módulo, y cuando no se ha establecido una conexión bluetooth con ningún otro dispositivo.
  - El LED del módulo en este estado parpadea rápidamente.
- En este estado a diferencia del HC-06, el HC-05 no puede interpretar los comandos AT.

### Estado Conectado o de comunicación

- Entra a este estado cuando se establece una conexión con otro dispositivo bluetooth.
- El LED hace un doble parpadeo.
- Todos los datos que se ingresen al HC-05 por el Pin RX se transmite por bluetooth al dispositivo conectado, y los datos recibidos se devuelven por el pin TX. La comunicación es transparente.

### Modo AT 1

- Para entrar a este estado después de conectar y alimentar el módulo es necesario presionar el botón del HC-05.
- En este estado, podemos enviar comandos AT, pero a la misma velocidad con el que está configurado.
- El LED del módulo en este estado parpadea rápidamente igual que en el estado desconectado.

### Modo AT 2

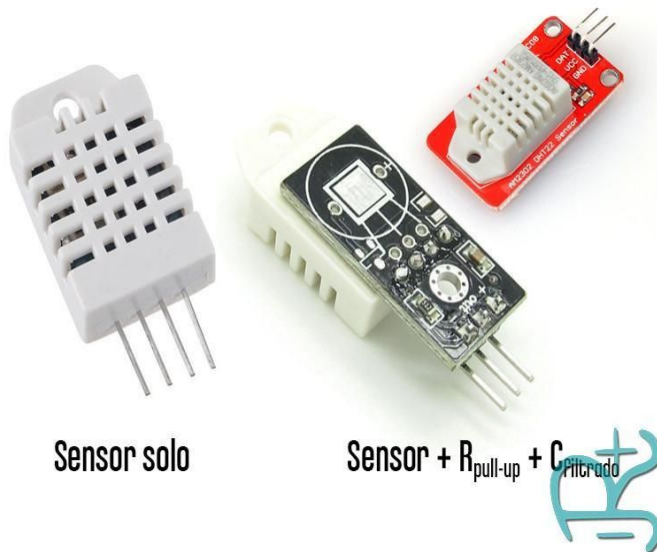
- Para entrar a este estado es necesario tener presionado el botón al momento de alimentar el módulo, es decir el módulo debe encender con el botón presionado, después de haber encendido se puede soltar y permanecerá en este estado.
- En este estado, para enviar comandos AT es necesario hacerlo a la velocidad de 38400 baudios, esto es muy útil cuando nos olvidamos la velocidad con la que hemos dejado configurado nuestro módulo.
- El LED del módulo en este estado parpadea lentamente.

### ● **Sensor DHT22 de temperatura y humedad**

Existen dos variantes del sensor en según que tienda, a saber:

1. El sensor suelto, con una “funda” blanca plástica y cuatro pines de conexión.
2. El sensor con la misma funda blanca que el anterior, pero esta vez soldado en una placa y con tres pines de conexión, además de una resistencia pull-up (entre 3-6 kΩ) y un condensador de filtrado (normalmente de 100 nF).

Diferentes encapsulados del DHT22



Sus principales características generales son:

- Alimentación: 3.3v – 5.5v, tomando como valor recomendado 5v.
- Resolución decimal, es decir, los valores tanto para humedad como para temperatura serán números con una cifra decimal.
- Tiempo de muestreo: 2 segundos, es decir, sólo nos puede ofrecer datos cada 2 segundos.

En cuanto a sus prestaciones leyendo temperatura:

- Rango de valores desde  $-40^{\circ}\text{C}$  hasta  $80^{\circ}\text{C}$  de temperatura.
- Precisión:  $\pm 0.5^{\circ}\text{C}$ ,  $\pm 1^{\circ}\text{C}$  como máximo en condiciones adversas.
- Tiempo de respuesta:  $< 10$  segundos, es decir, de media, tarda menos de 10 segundos en reflejar un cambio de temperatura real en el entorno.

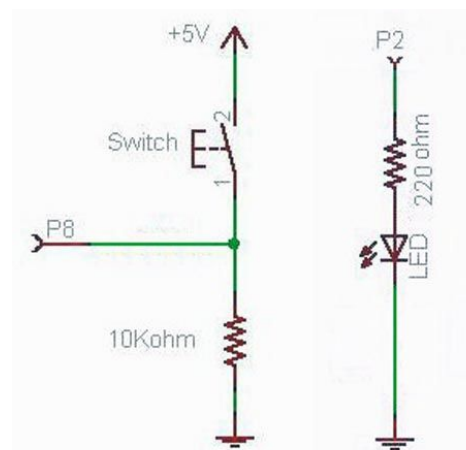
Si hablamos de sus prestaciones leyendo humedad relativa:

- Rango de valores desde 0% hasta 99.9% de Humedad Relativa.
- Precisión:  $\pm 2\% \text{RH}$ , a una temperatura de  $25^{\circ}\text{C}$ .
- Tiempo de respuesta:  $< 5$  segundos, es decir, de media, tarda menos de 5 segundos en reflejar un cambio de humedad relativa real en el entorno. Además, para darse esta afirmación, los test indicaron que la velocidad del aire debe ser de 1 m/s.

Nuevamente estamos ante un sensor modesto, pero que ya nos es válido para un montón de proyectos caseros y semi profesionales. Siempre y cuando no se requiera de una medición constante ni milimétrica, el sensor DHT22 es un fiel candidato.

### • Pulsador dip tact switch

Un interruptor es un dispositivo simple con dos posiciones, EN y AP (ENCendido y APagado). Una clase de interruptor que usted usa cada día es el interruptor de la luz. Cuando conecta, dentro del interruptor, dos cables son unidos, lo que permite fluir a la corriente que



enciende la luz o la tostadora se caliente. Cuando lo desconecta, los dos cables son desunidos y corta el flujo de la corriente.

- **Potenciómetro**



Un potenciómetro es un dispositivo conformado por 2 resistencias en serie, las cuales poseen valores que pueden ser modificados por el usuario. Existen múltiples tipos de potenciómetros, variando su forma y el método cómo modifican los valores de las resistencias. Un potenciómetro son 2 resistencias conectadas en serie. A partir del nodo que se forma entre estas dos resistencias tenemos un terminal, el cual normalmente será la pata del centro en un potenciómetro de 3 patas.

## Detalles técnicos de los actuadores utilizados:

- **Led – PWM**

Diodos emisores de luz utilizados como actuadoresl.

Se debe utilizar una resistencia para conectarlos ya que si la tensión es inferior a un valor determinado (Valor de polarización directa), el Led no se enciende y por el contrario si es superior a este valor el Led se rompe. El led es un diodo, por lo que tiene polaridad, entonces se debe conectar de una manera específica.

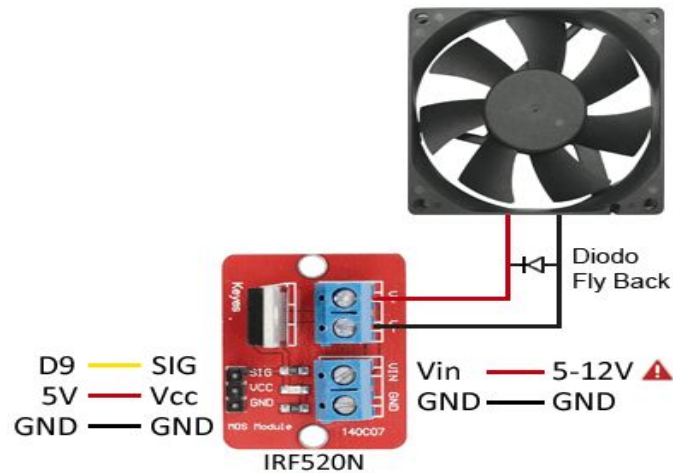


Para iluminar los Leds se utiliza PWM (Modulación por ancho de pulso), que es una técnica en la que se modifica el Duty Cycle, de la señal para controlar la energía que se envía.

- Corriente: 20mA
- Ground (-): 0v
- Voltaje: 3v – 3.6v

- **Cooler**

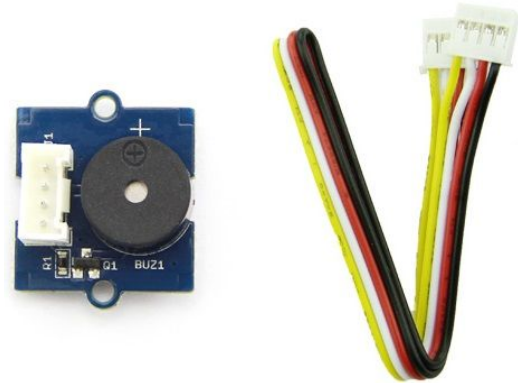
Este elemento se encarga de mantener la temperatura del equipo en el estado ideal, con el objetivo de que el extremo calor que se produce en los ordenadores no acabe con la estabilidad de los mismos.



Vin entre 5 y 12V. Al usar alimentación externa SIEMPRE poner con GND común.

- **Buzzer**

Buzzer tiene un zumbador piezoeléctrico como componente principal. El piezo puede conectarse a salidas digitales y emitirá un tono cuando la salida sea ALTA. Alternativamente, se puede conectar a una salida de modulación de ancho de pulso analógica para generar varios tonos y efectos.



## Conexión de los sensores con Arduino:

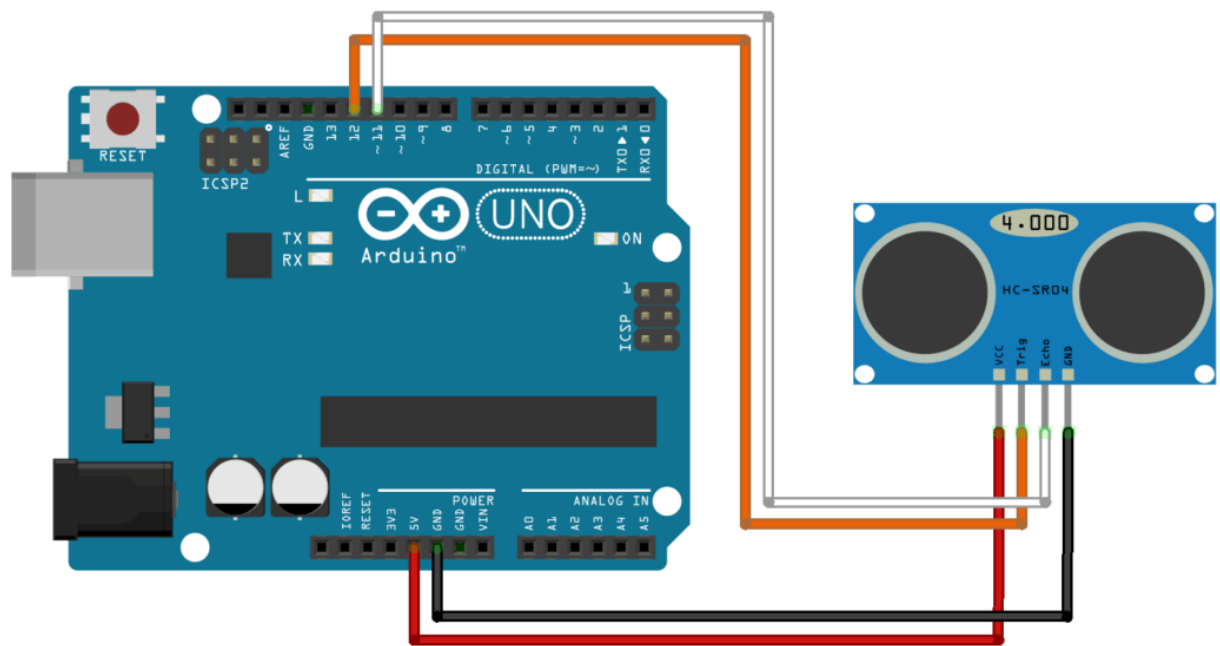
- **Sensor ultrasónico HC-SR04**

La conexión del sensor con Arduino es muy sencilla. Podemos realizarla utilizando un protoboard, o directamente con alambres. Para lograr que el sensor funcione, son necesarias 4 señales:

- Alimentación de 5 volts.
- Tierra o común del circuito.
- Señal de disparo (trig).
- Señal de eco (echo).

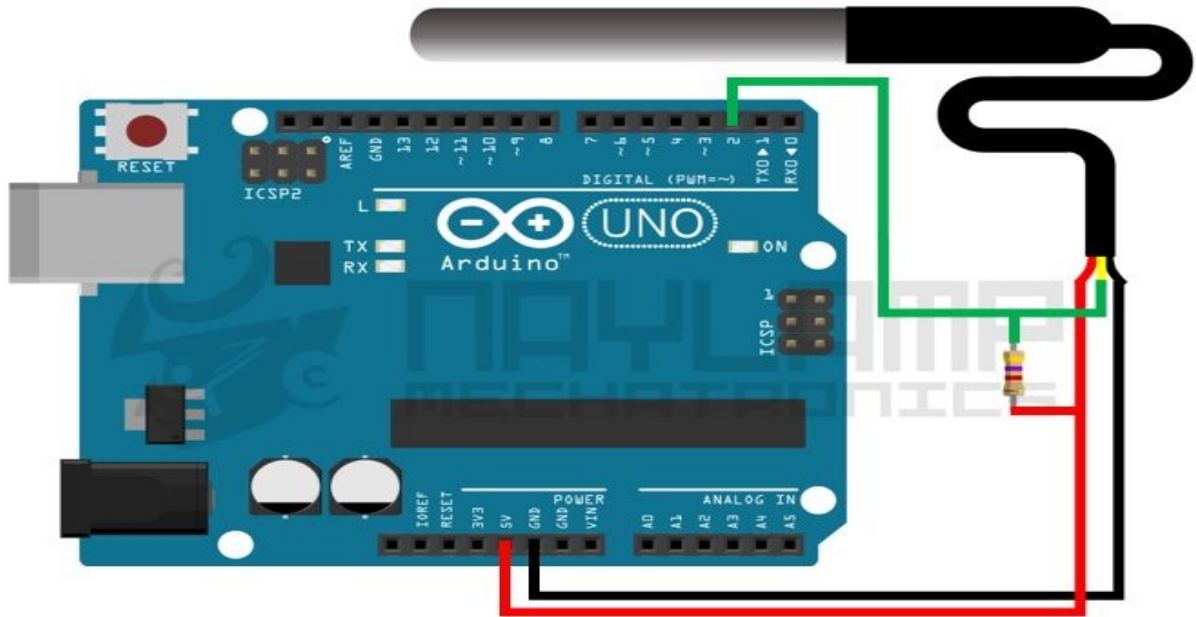
La siguiente imagen muestra como queda conectado el sensor utilizando solamente cables:





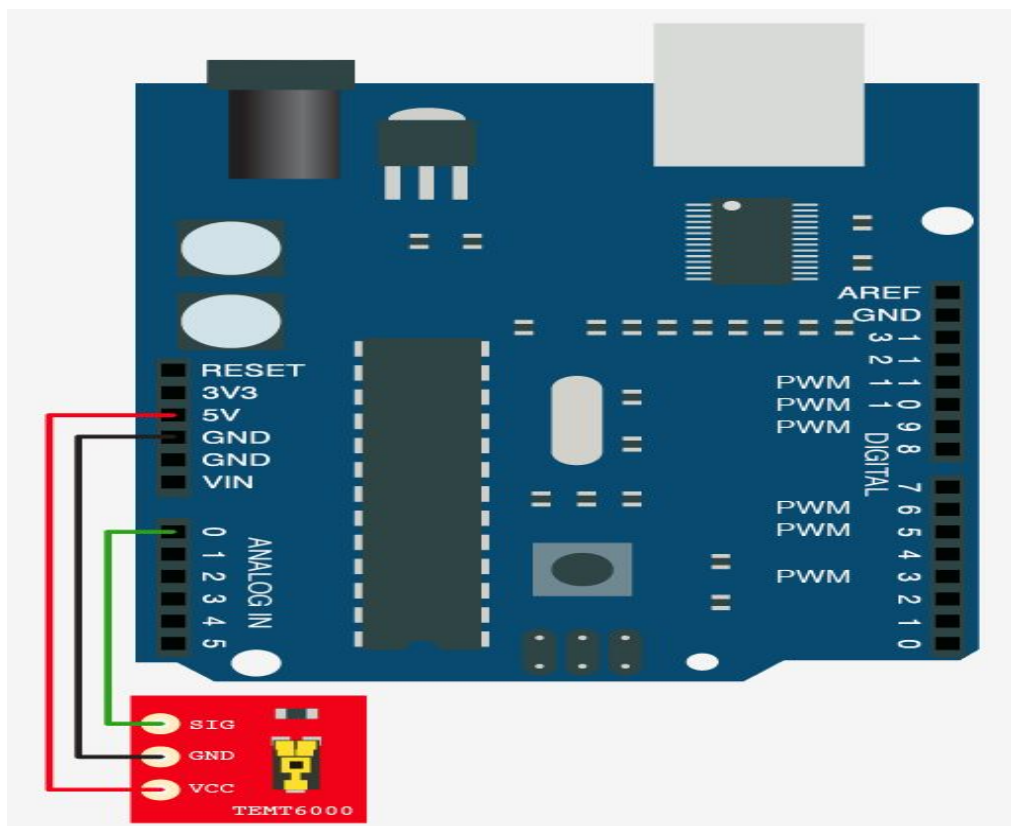
fritzing

- Sensor digital temperatura DS18B20



Notar que es necesario poner una resistencia Pull-Up de 4.7K, y pueden usar un protoboard para facilitar la conexión.

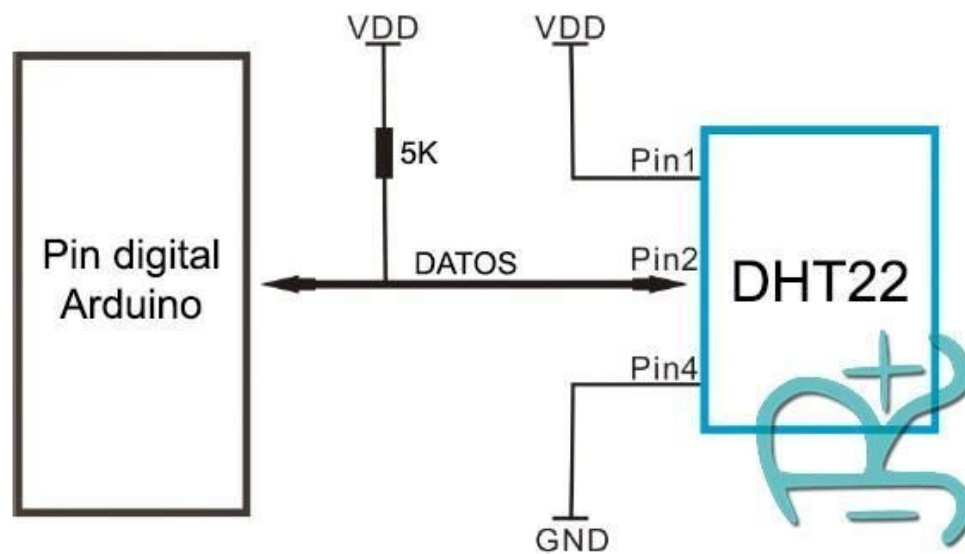
- Sensor de luz ambiental TEMA6000



Es muy simple, conecte la alimentación y la conexión a tierra, y el pin de señal a su entrada analógica favorita y listo. El sensor emitirá un voltaje analógico , que aumenta cuando hay más luz.

- **Sensor DHT22 de temperatura y humedad**

Como siempre que nos encontramos con componentes encapsulados o presoldados en una placa, una conexión va a estar condicionada por la cantidad de elementos -componentes- que ya forman parte del encapsulado que hemos adquirido. Una conexión totalmente funcional, contando con que nuestro DHT22 no trae ningún elemento extra es la siguiente:



Esquema conexión DHT22

De forma opcional podemos filtrar la alimentación colocando un condensador de, por ejemplo, 100 nF, entre VDD y GND.

- **Pulsador dip tact switch**

Para conectar un botón al arduino conviene tener en mente ciertos puntos:

- Un botón puede conectarse a cualquier pin de arduino (digital o analógico, ya que los analógicos usualmente funcionan también como digitales).
- Existen dos configuraciones posibles: con resistencia pull-up o resistencia pull-down.
- Configuraremos el pin seleccionado como entrada digital.
- Debemos tomar en cuenta los rebotes y el ruido eléctrico en el software.
- Es necesario utilizar una resistencia pull-up o pull-down, ya sea interna o externa.

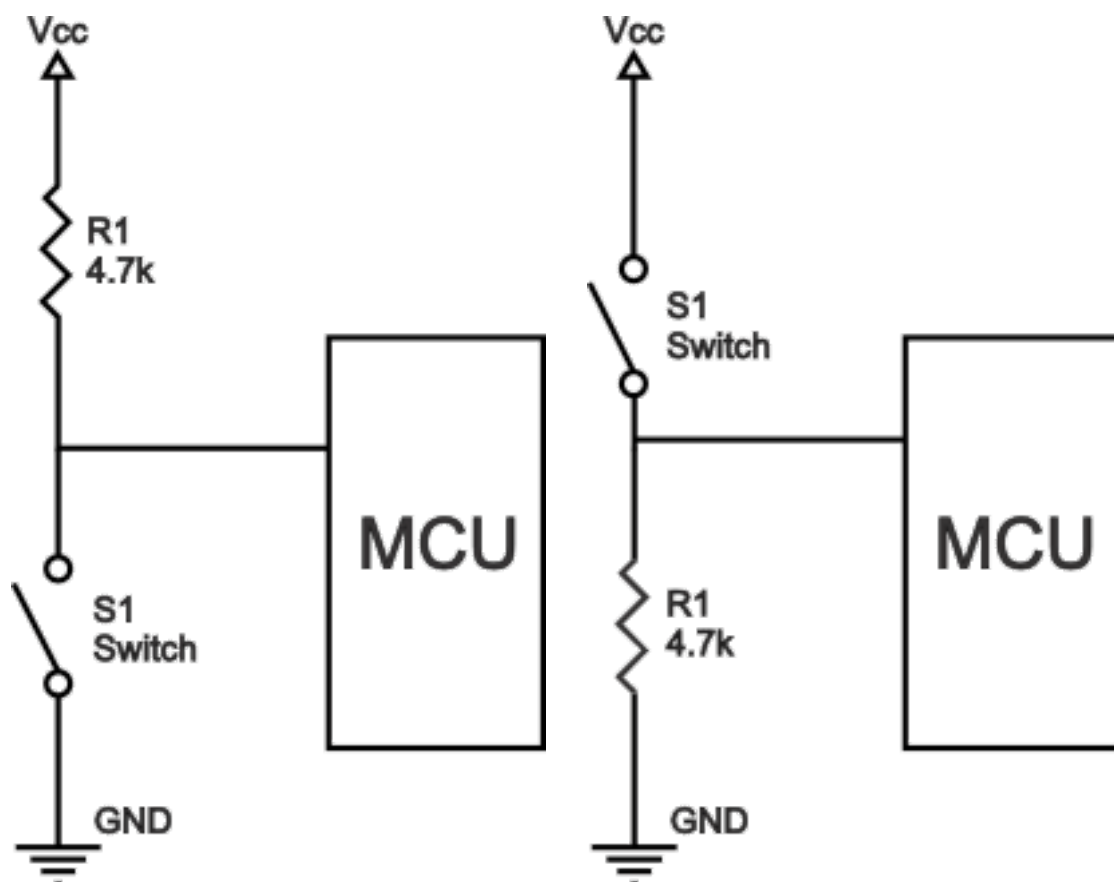
Las dos formas que existen para conectar un botón al arduino se visualizan en los siguientes diagramas:

---

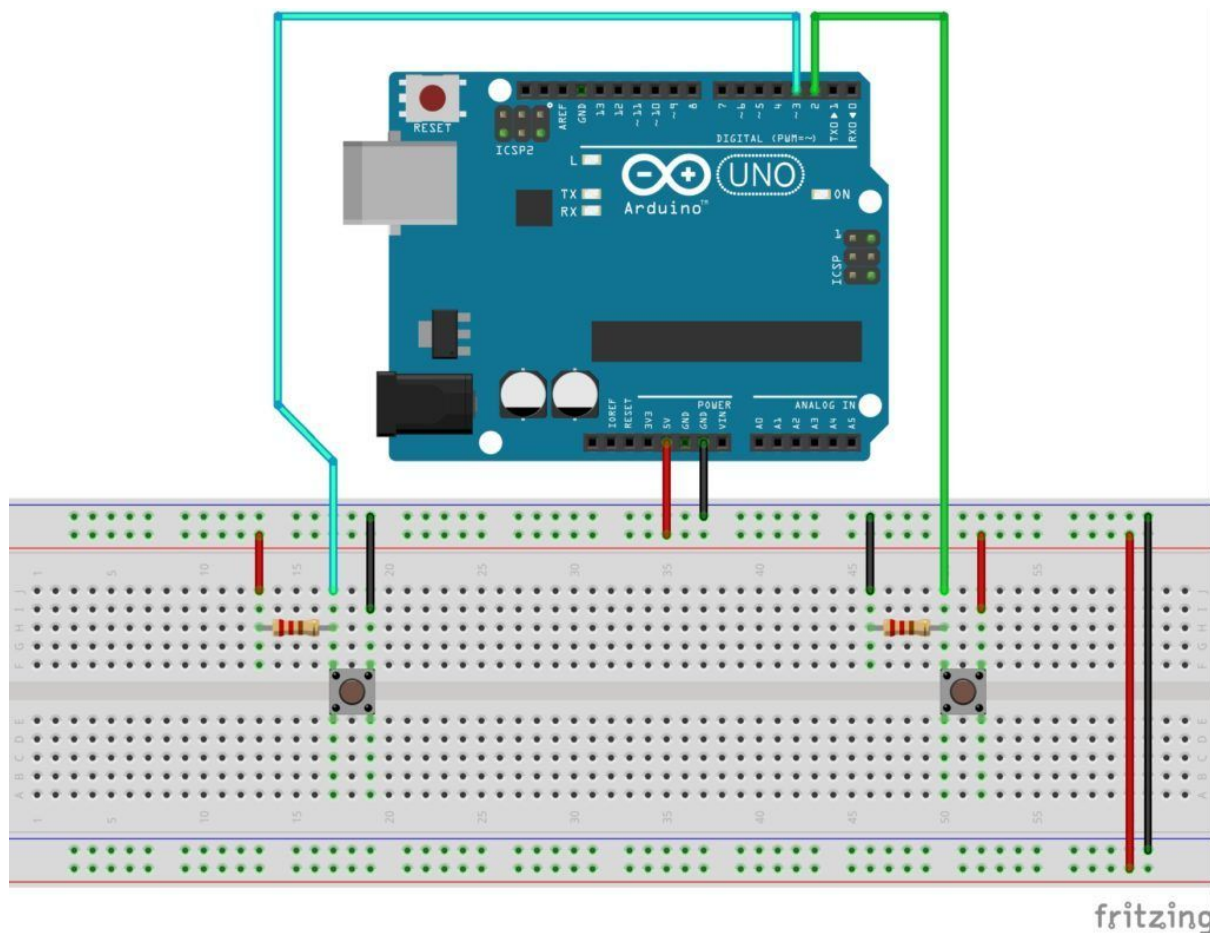
Botón con resistencia pull-up

Botón con resistencia pull-down

---



Visto en el protoboard quedaría como se muestra más abajo, siendo el botón colocado del lado izquierdo del protoboard el que está conectado con pull-up, mientras que el derecho está conectado con una resistencia pull-down.



El valor de las resistencias en este caso no es crítico, pero debemos considerar que valores menores causarán que circule una corriente mayor cuando se oprime el botón, mientras que los valores mayores pueden permitir que el ruido se introduzca al pin de entrada.

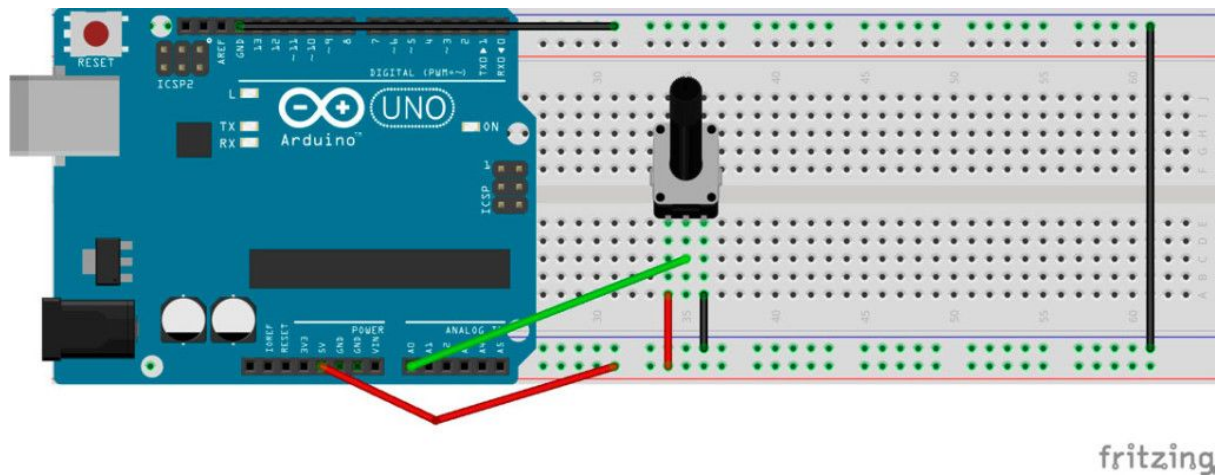
Los valores típicos para resistencia pull-up o pull-down son de 1 a 10 KOhms.

- **Potenciómetro 5k**

En la placa Arduino UNO tenemos 6 pines analógicos, desde A0 hasta A5 y su uso común es la lectura de datos de dispositivos analógicos como es el caso del potenciómetro. Tienen una resolución de 10 bits lo que implica que tenemos 1024 valores diferentes, es decir, podemos leer un rango de tensiones desde 0V hasta 5V detectando cambios de voltaje de 0.004V ( $5/1024$ ). Por lo que los valores que obtendremos irán desde 0 hasta 1023.

Y como la mejor manera de entender algo son los ejemplos, empezamos con uno que mediante el monitor serie podremos ir viendo qué valores vamos obteniendo en un pin analógico según vayamos modificando la posición del potenciómetro.

En el siguiente esquema mostramos cómo conectarlo.



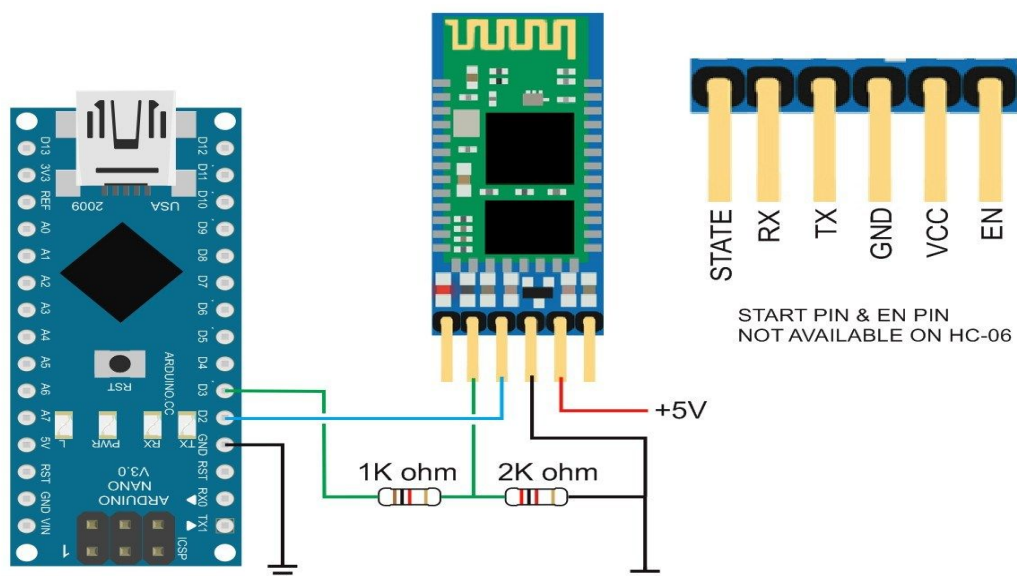
- **Bluetooth HC-05**

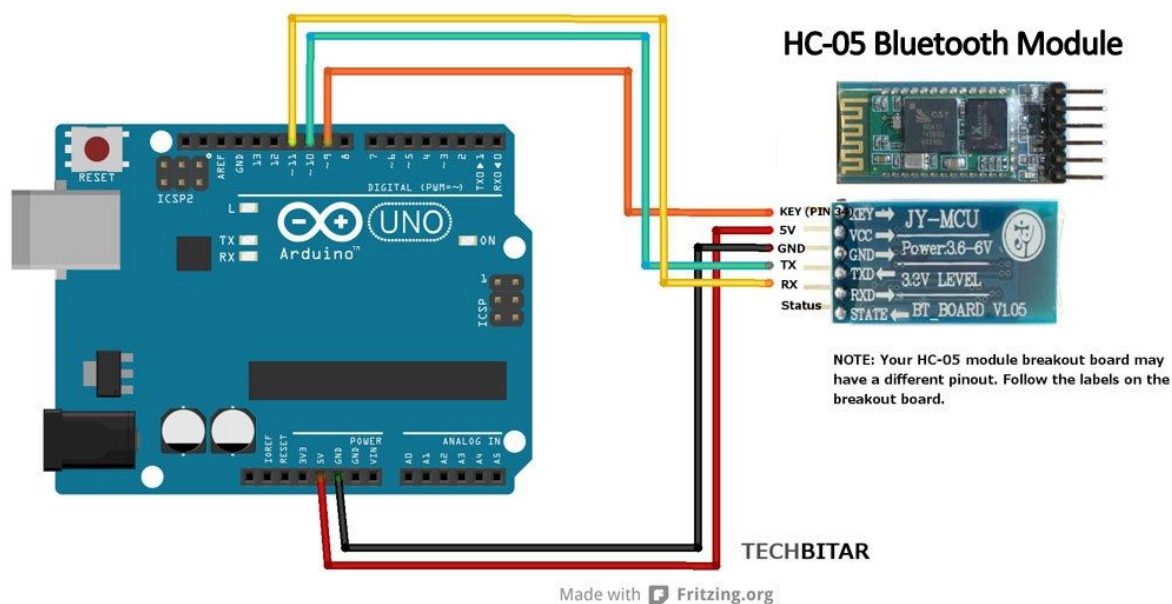
El módulo HC-05, que puede configurarse tanto como Master que como Slave, y que además dispone de bastante más parámetros de configuración y capacidades de interrogación.

Mientras que el HC-06 entra en modo de programación en cuanto lo enciendes y mientras no haya nadie conectado por Bluetooth, el HC-05 es ligeramente más complicado de colocar en modo comandos y requiere una cierta manera de arrancado, concretamente requiere que el pin KEY, (que no estaba conectado el caso del HC-06) esté en HIGH cuando encendemos el módulo.

Conexión a Arduino:

## HC-05 FC-114 & HC-06 FC-114 Connections to Arduino





Como el modulo HC-05 funciona a 3,3 V hay bastante controversia en internet respecto a si debemos o no, poner divisores de tensión para acceder a los pines del módulo con Arduino. Pero por las pruebas hechas y los tutoriales en principio no es necesario usar un divisor de tensión y los pines son tolerantes a 5V, pero hacerlo bajo vuestra responsabilidad.

## Conexión de los actuadores con Arduino:

- **Diodo led**

Se debe tener en cuenta el conocimiento de cuál es el cátodo y ánodo del LED. Se coloca el ánodo en el pin 13 y el cátodo a tierra (ground). Además se debe colocar una resistencia de 330 ohms, para evitar que este se queme, la cual se conecta con el cátodo del diodo led, como se muestra en la Figura 1:



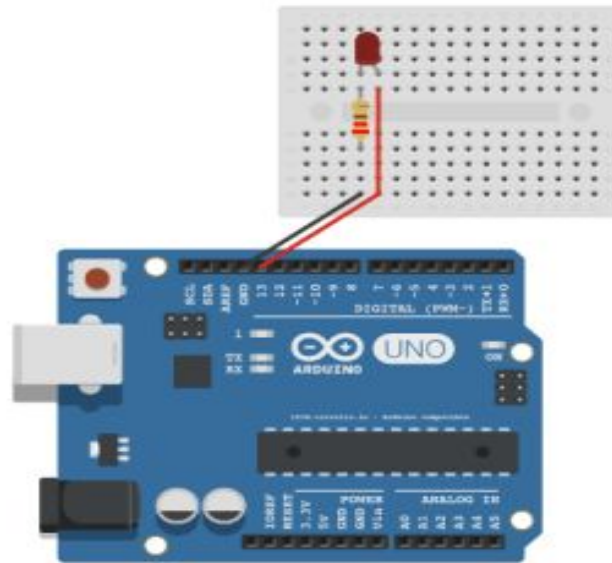
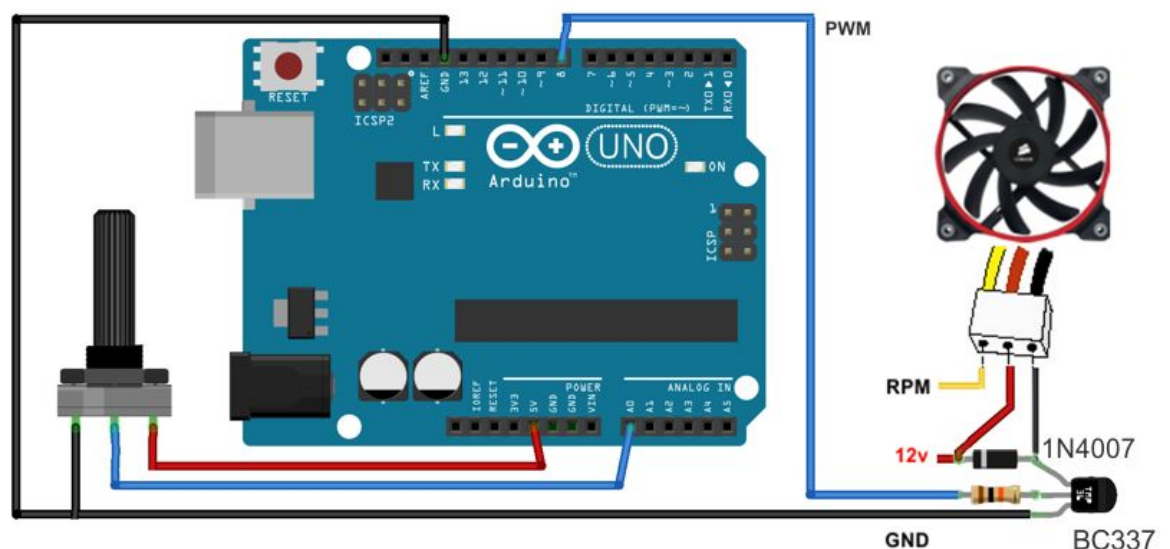


Figura 1 – Montaje del circuito Encender y apagar un LED con Arduino

- **Cooler**

Arduino dispone de salidas PWM pero funcionan a una frecuencia de 490 Hz, se pueden obtener múltiplos de esta frecuencia modificando los “timers” pero aun así tampoco hay forma sencilla de alcanzar los 25 KHz necesarios.

Esquema de conexiones del circuito para regular la velocidad de un ventilador de 3 pines y 12V:

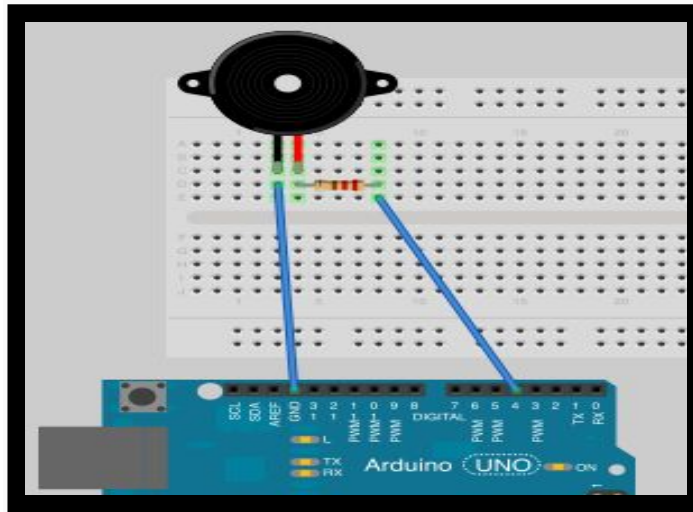


Y este programa permite controlar la velocidad de un ventilador de 2 ó 3 pines actuando sobre su alimentación, el valor de T se puede ajustar individualmente en cada caso particular.

- **Buzzer**



Antes de ir al circuito necesitaremos una resistencia de 100 ohms y el buzzer. Una vez obtenidos los materiales pasaremos al circuito.



## Alcance del Sistema:

### Sistema Embebido

- El embebido está incorporado a la silla siendo no externo a la misma
- El embebido proporciona un pulsador desde donde al sentarse comienza a tomar los valores de los sensores
- El embebido se conecta al dispositivo android a través de bluetooth enviándole datos

### Aplicación Android

La aplicación Android actúa como un control remoto para poder controlar los actuadores de la silla inteligente y mostrar la información obtenida de los sensores.

## Arduino uno:

### Características técnicas del ARDUINO UNO

Arduino es una placa con un microcontrolador de la marca Atmel y con toda la circuitería de soporte, que incluye, reguladores de tensión, un puerto USB (En los últimos modelos, aunque el original utilizaba un puerto serie) conectado a un módulo adaptador USB-Serie que permite programar el microcontrolador desde cualquier PC de manera cómoda y también hacer pruebas de comunicación con el propio chip.

Un arduino dispone de 14 pines que pueden configurarse como entrada o salida y a los que puede conectarse cualquier dispositivo que sea capaz de transmitir o recibir señales digitales de 0 y 5 V

También dispone de entradas y salidas analógicas. Mediante las entradas analógicas podemos obtener datos de sensores en forma de variaciones continuas de un voltaje. Las salidas analógicas suelen utilizarse para enviar señales de control en forma de señales PWM.

Nos permite programar el chip sobre la propia placa y después integrarlo en otros montajes.

Entradas y salidas:

Cada uno de los 14 pines digitales se puede usar como entrada o como salida. Funcionan a 5V, cada pin puede suministrar hasta 40 mA. La intensidad máxima de entrada también es de 40 mA.

Cada uno de los pines digitales dispone de una resistencia de pull-up interna de entre 20K $\Omega$  y 50 K $\Omega$  que está desconectada, salvo que nosotros indiquemos lo contrario.

Arduino también dispone de 6 pines de entrada analógicos que trasladan las señales a un conversor analógico/digital de 10 bits.

Pines especiales de entrada y salida:

RX y TX: Se usan para transmisiones serie de señales TTL.

Interrupciones externas: Los pines 2 y 3 están configurados para generar una interrupción. Las interrupciones pueden dispararse cuando se encuentra un valor bajo en estas entradas y con flancos de subida o bajada de la entrada.

PWM: Arduino dispone de 6 salidas destinadas a la generación de señales PWM de hasta 8 bits  
(0-255)

SPI: Los pines 10, 11, 12 y 13 pueden utilizarse para llevar a cabo comunicaciones SPI, que permiten trasladar información full dúplex en un entorno Maestro/Esclavo.

I<sup>2</sup>C: Permite establecer comunicaciones a través de un bus I<sup>2</sup>C. El bus I<sup>2</sup>C es un producto de Phillips para interconexión de sistemas embebidos. Actualmente se puede encontrar una gran diversidad de dispositivos que utilizan esta interfaz, desde pantallas LCD, memorias EEPROM, sensores...

Alimentación de un Arduino

Puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un pequeño transformador o, por ejemplo una pila de 9V. Los límites están entre los 6 y los 12 V. Como única restricción hay que saber que si la placa se alimenta con menos de 7V, la salida del regulador de tensión a 5V puede dar menos que este voltaje y si sobrepasamos los 12V, probablemente dañaremos la placa.

La alimentación puede conectarse mediante un conector de 2,1mm con el positivo en el centro o directamente a los pines Vin y GND marcados sobre la placa.

Hay que tener en cuenta que podemos medir el voltaje presente en el jack directamente desde Vin. En el caso de que el Arduino esté siendo alimentado mediante el cable USB, ese voltaje no podrá monitorizarse desde aquí.

#### Resumen de características Técnicas

Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje de entrada (Recomendado)	7 – 12V
Voltaje de entrada (Límite)	6 – 20V
Pines para entrada- salida digital.	14 (6 pueden usarse como salida de PWM)
Pines de entrada analógica.	6
Corriente continua por pin IO	40 mA
Corriente continua en el pin 3.3V	50 mA
Memoria Flash	32 KB (0,5 KB ocupados por el bootloader)
SRAM	2 KB
EEPROM	1 KB
Frecuencia de reloj	16 MHz

## PWM:

Un sistema de control (como un microcontrolador) no tiene capacidad alguna para trabajar con señales analógicas, de modo que necesita convertir las señales analógicas en señales digitales para poder trabajar con ellas. Para la conversión de analógico a digital, en el caso de un arduino Uno, el valor de 0 voltios analógico es expresado en digital como B0000000000 (0) y el valor de 5V analógico es expresado en digital como B1111111111 (1023). Arduino Uno tiene una resolución de 10 bits, es decir, unos valores entre 0 y 1023 para la conversión de analógico a digital.

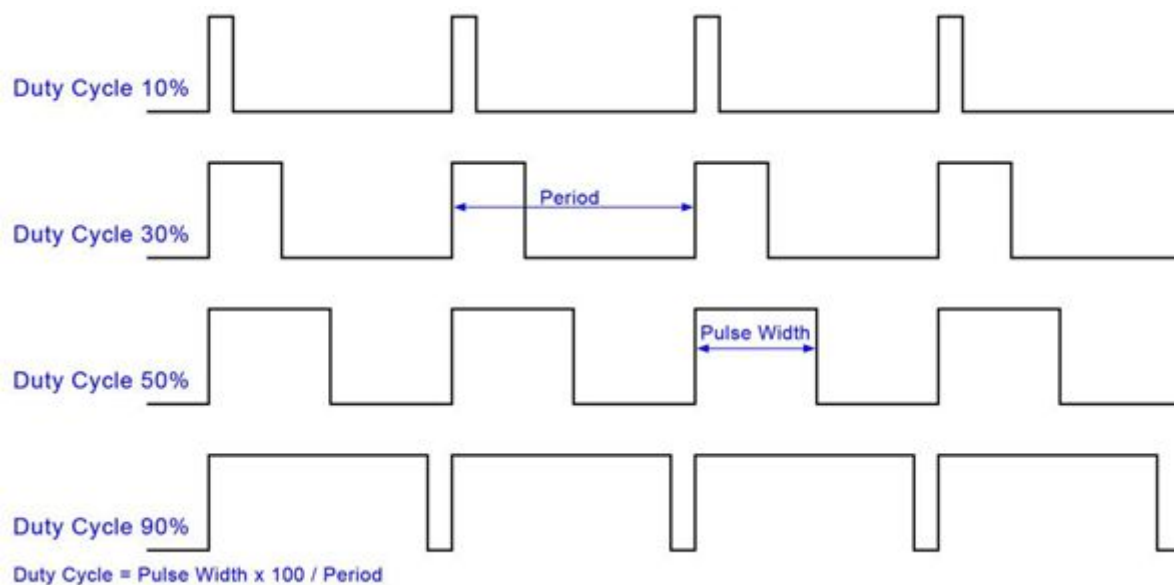
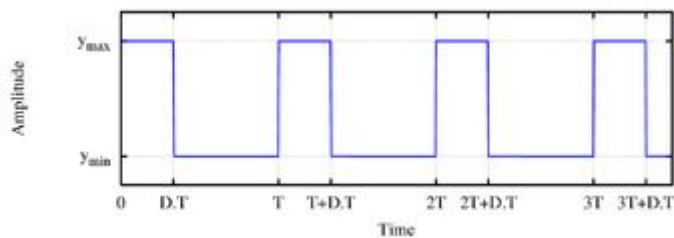
Los microcontroladores de Arduino contienen en la placa un conversor analógico a digital de 6 canales. El conversor tiene una resolución de 10 bits, devolviendo enteros entre 0 y 1023. La nomenclatura para los pines analógicos es A0, A1, etc...

Como hemos dicho Arduino Uno tiene entradas analógicas que gracias a los conversores analógico digital puede entender ese valor el microcontrolador, pero no tiene salidas analógicas puras y para solucionar esto, usa la técnica de PWM.

Las Salidas PWM (modulación por ancho de pulso) .

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica. ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período.



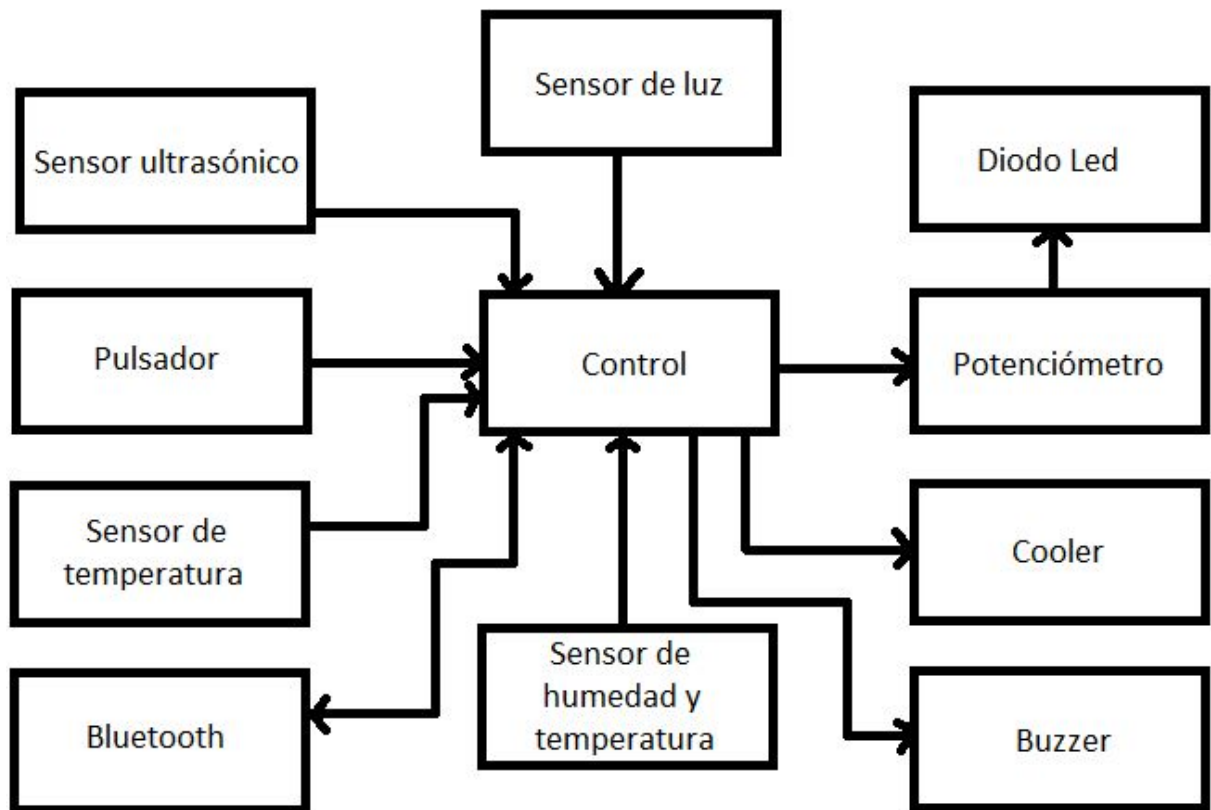
En Arduino la frecuencia de PWM es de 500Hz. Pero es un valor que puede modificarse en caso que lo necesitemos.

No confundir PWM con la función `tone()` que es utilizada para generar una onda cuadrada de ciclo de trabajo 50% y frecuencia variable, con el fin de emitir sonidos audibles, modificando la frecuencia.

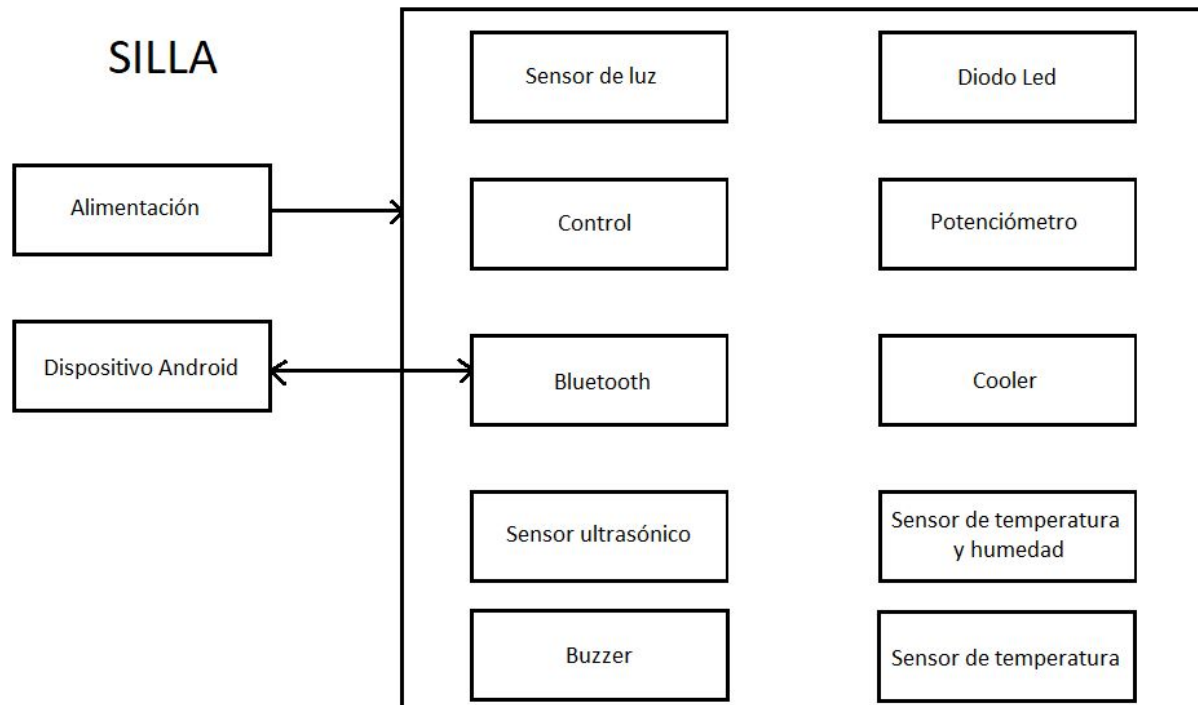
## Diseño:

Esquemas gráficos:

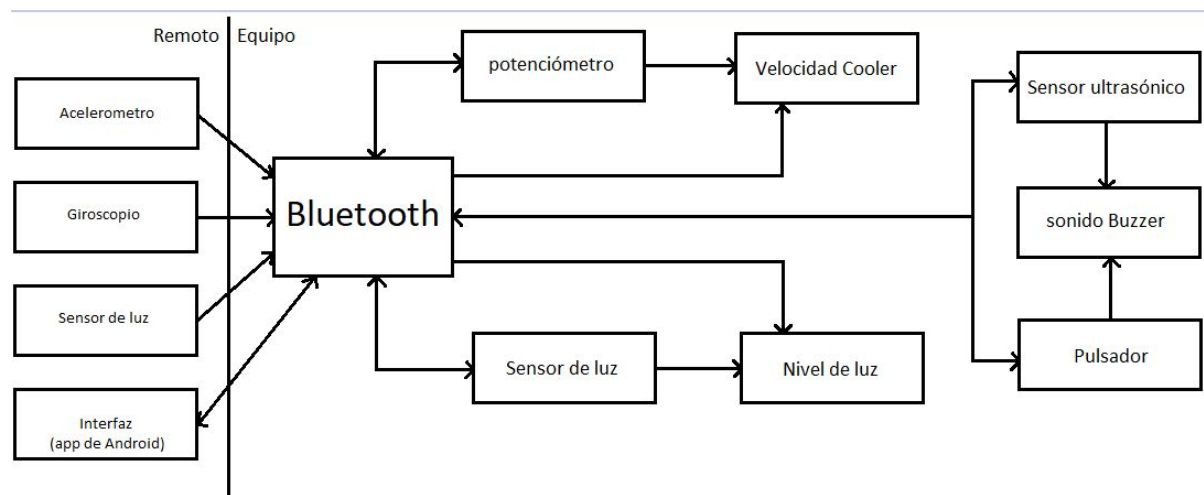
Diagrama funcional:



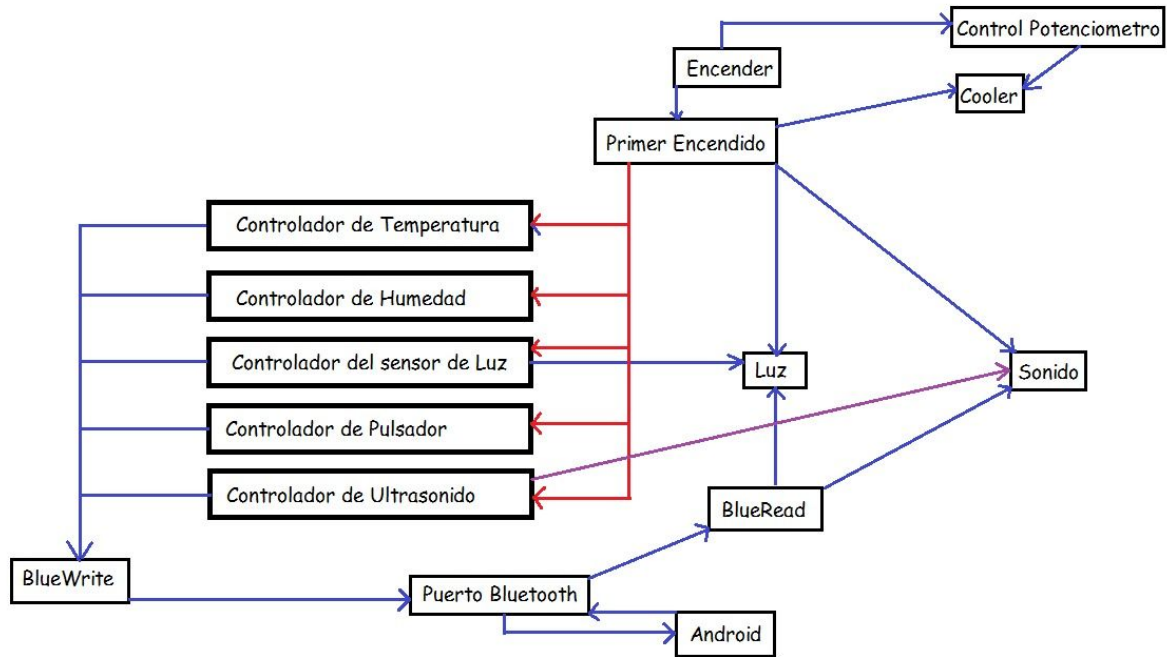
### Diagrama físico:



### Diagrama de lógico:



### Diagrama de lógico:



## Prototipo:

