Smart Air

# Objetivo

Desarrollar un Sistema embebido (Smart Air) aplicando los conocimientos adquiridos durante el cuatrimestre. SmartAir es un aire inverter pequeño y transportable controlado desde una aplicación Android desde tu celular.

Se tendrá la posibilidad de controlar la temperatura interna, la temperatura externa, entre otras funciones más detalladas a continuación.

# Descripción general del sistema – Manual de Uso

Smart Air es un producto diseñado especialmente para poder ser utilizado en cualquier lugar donde puede enchufarse, es perfecto para refrescarse en un día de verano o para calentarse en un día frío de invierno, gracias a que es un aire inverter. Brinda la posibilidad de prender y apagar el aparato, entre otras funciones, a través de una aplicación Android utilizando Bluetooth.

El sistema embebido (SE) tomará la temperatura exterior de la habitación en la que es colocado y la temperatura interna por donde el aire (cálido o frío) es expulsado. De esta forma al seleccionar en la aplicación el modo elegido, sea frío, calor o automático (24°C). Se encenderán en el modo correcto las placas Peltier que tiene internamente para generar frío o calor según corresponda.

Además, se podrá elegir desde la aplicación con qué velocidad queremos que el aire sea expulsado en caso de requerir enfriar o calentar la habitación más rápido.

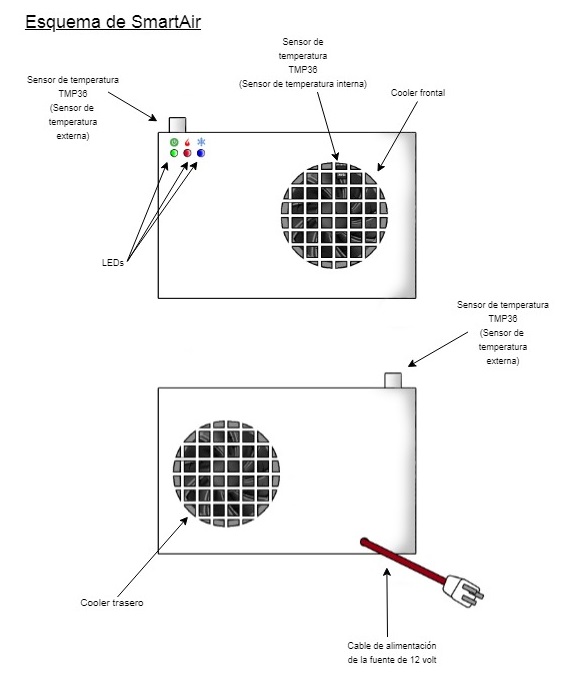
Para un mejor uso es recomendable que la parte trasera del dispositivo donde se elimina el aire se apunte al exterior, al igual que los aires convencionales.

El dispositivo también cuenta con un sensor de inclinación el cual permite, cuando el dispositivo está encendido y es transportado, que este se apague para no generar daños internos.

Por último, cuenta con un buzzer que indica cuando el dispositivo se enciende, apaga, cambia de temperatura, velocidad y modo, además de tres leds frontales que indican el estado del dispositivo:

* Al encender el dispositivo se prenderá el led verde.
* Al estar en “modo frío”, además de mantener el led verde de encendido, se prenderá el led azul.
* Al estar en “modo calor”, además de mantener el led verde de encendido, se prenderá el led rojo.
* Al estar en “modo automático”, además de mantener el led verde de encendido, se prenderán los led azul y rojo.
* Por último, el dispositivo cuando con un “modo seguro”, este se activa cuando el sensor de inclinación detecta que el dispositivo se encuentra en una posición que compromete su integridad, cuando esto sucede se apagarán los led azul y rojo (según corresponda), el buzzer producirá un pitido constante y se apagarán las placas Peltier al igual que los ventiladores internos.

En la siguiente imagen se puede apreciar el dispositivo y como debe ser conectado:



# Componentes

## Hardware:

* 1 Placa Arduino UNO SMD CH340.
* 1 Cable USB
* 2 Sensores de temperatura TMP-36.
* 1 Buzzer.
* 1 Sensor de inclinación.
* 1 Led Verde.
* 1 Led Rojo.
* 1 Led Azul.
* 2 Celdas Peltier TEC1-12706.
* Protoboard de 400 puntos.
* 2 Coolers con disipador de 8x8cm.
* 1 Fuente de computadora 12v.
* Cables macho-macho.
* Cables macho-hembra.
* Pines.
* Modulo Bluetooth HC-05.
* Resistencias 4k7.
* Transistores BC337.
* 2 Transistores IRF 540.
* 2 Transistores BC558.
* 2 Transistores BC548.
* Puente h (opcional para aire inverter).

# Implementación del Embebido

## Implementación Leds

*Objetivo:* Se colocan los leds para indicar los distintos estados por los que pasa el dispositivo:

* Encendido: Led verde encendida.
* Modo frío: Led Azul y Verde encendidos.
* Modo calos: Led Rojo y Verde encendidos.
* Modo automático: Led Azul, Rojo y Verde encendidos.

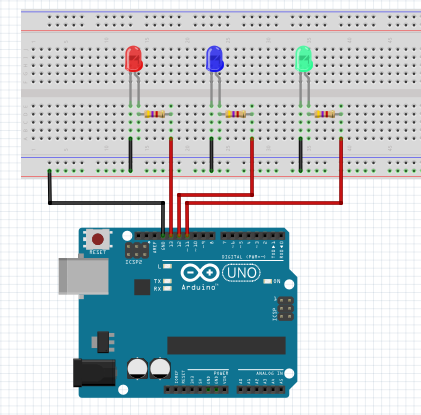
*Señal:* digital.

*Estados*: LOW (apagar) y HIGH (encender)

*Pin Arduino:*

* Pin 11: Led verde (encendido) configurado como salida.
* Pin 12: Led Azul (modo frío) configurado como salida.
* Pin 13: Led Rojo (modo calor) configurado como salida.

*Circuito:* Como se muestra en la imagen se conecta una resistencia de 4K7Ω al positivo del led y el otro extremo de la resistencia se conecta al pin del Arduino que controla el led. El extremo negativo del led se conecta al pin ground del Arduino.



## Implementación de Sensor de inclinación

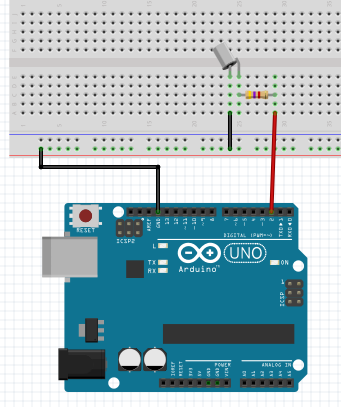
*Objetivo:* Se coloca un sensor de inclinación para proteger las partes delicadas del dispositivo cuando está encendido y es transportado. Cuando se detecta movimiento, las partes internar que generar frío o calor se apagan y se activa el buzzer para indicar que el dispositivo está en “modo seguridad”.

*Señal:* digital.

*Estados*: LOW (inclinado).

*Pin Arduino:* Pin 2 configurado como INPUT\_PULLUP.

*Circuito:* Como se muestra en la imagen se conecta una resistencia de 4K7Ω al positivo del sensor de inclinación y el otro extremo de la resistencia se conecta al pin del Arduino que controla el led. El extremo negativo del sensor se conecta al pin ground del Arduino.



## Implementación de Buzzer

*Objetivo:* Se coloca un buzzer para indicar de forma sonora las acciones realizadas por el usuario. El buzzer sonará al encerder, cambiar de modo, subir o bajar temperatura y velocidad de los cooler. También indica el “modo seguridad” cuando se inclina el dispositivo sonando de forma constante hasta que el dispositivo vuelva a la posición correcta.

*Señal:* digital.

*Estados*: tone(buzzer,<tiempo>) , notone(buzzer).

*Pin Arduino:* Pin 8 configurado como salida.

*Circuito:* Como se muestra en la imagen se conecta el buzzer al pin del Arduino que lo controla de forma directa. El extremo negativo del buzzer se conecta al pin ground del Arduino.

## 

## Implementación de sensores de temperatura

*Objetivo:* Se colocan dos sensores de temperatura (uno interno al dispositivo y otro externo) para medir la temperatura y eventualmente controlarla con las celdas Peltier y los coolers.

*Señal:* analógica. Es un sensor que se puede alimentar entre un rango de voltaje que va desde los 2.7V hasta los 5.5V, viene calibrado directamente en grados centígrados (ºC), presenta un factor de escala lineal de 10 mV/ºC (esto es la relación entre el cambio en la señal de salida y el cambio en la señal de la medida, es decir, cada 10 mV aumenta 1 ºC).

*Pin Arduino:*

* Pin Analógico 0 (entrada): Sensor de temperatura TMP36 (temperatura interna)
* Pin Analógico 1 (entrada): Sensor de temperatura TMP36 (temperatura externa)

*Circuito:* Como se muestra en la imagen se conectan los conectores del medio de los sensores de temperatura a los pines analógicos que serán configurados como entradas, luego se conecta el conector izquierdo a 5V y el derecho al ground del Arduino.

