

Universidad Nacional de la Matanza
Departamento de Ingeniería e Investigaciones
Tecnológicas

Sistemas operativos

Avanzados

SISTEMAS EMBEBIDOS – IoT

Día de cursada: Miércoles

Turno: Noche

Docentes: Lic. Graciela de Luca
Ing Waldo Valiente
Ing Esteban Carnuccio
Ing Sebastián Barillaro

Integrantes	DNI
Alessandrini, Ernesto	33.458.573
Balart, Agustín	35.537.140
Cruz Rey, Juan	36.921.336
D'amico, Rodrigo	36.817.422
Famiglietti, Cristian	29.284.170

Contenido

PROYECTO: SAFEROOM	3
Introducción	3
Desarrollo:	3
Funcionamiento general:	3
Pasos para la creación del sketch en arduino:.....	4
Funcionamiento del sketch Arduino	4
Pasos para la creación de la App SafeRoom con Android Studio:	5
Funcionamiento de la APK SafeRoom:	5
Explicaciones de actuadores	5
Servomotor Tower Pro SG92R 9G	5
Conexión	7
DISPLAY LCD 16X2 BACKLIGHT AZUL (REES52 16X2 Hd44780)	7
BUZZER PIEZOELÉCTRICO PASIVO V1.1B	9
COOLER	12
Explicación de sensores	13
Sensor MQ	13
El calentador.....	13
Resistencia de carga	14
Sensor de Monóxido de Carbono MQ7	14
SENSOR DHT22	16
CICLO DE OPERACIÓN	17
BLUETOOTH	18
Modulo Bluetooth HC-05 y HC-06.....	18
ARDUINO UNO	20
PWM	24
DIAGRAMA EN BLOQUE	26
DIAGRAMA FUNCIONAL	26

DIAGRAMA FÍSICO	26
DIAGRAMA DE SW	27

PROYECTO: SAFEROOM

Introducción

A lo largo de este informe se pretende explicar nuestra experiencia utilizando Sistemas Embebidos, uso de sensores y actuadores, uso de placa Arduino, su comunicación a través de una conexión Bluetooth al SO Android, entre otros. El objetivo de este trabajo práctico es alejarse un poco de lo que veníamos acostumbrados a trabajar: solo con software; para poder dedicarnos también al hardware (electrónica). A lo largo de este informe expondremos cómo y con qué elementos desarrollamos en nuestro sistema.

Desarrollo:

El problema a resolver es la creación de un sistema de refrigeración y alertas en recintos de trabajos industriales e investigación, con la tecnología adecuada para la detección de cantidades peligrosas de monóxido de carbono y temperaturas, con el objetivo de alertar y actuar en consecuencia. Para ello se usaron los siguientes sensores y actuadores:

Actuadores:

- Servo motor: **Servomotor Tower Pro SG92R 9G**. (el mismo se usará para abrir una ventana).
- Led: **Placa Led Rgb Tricolor** (alarma visual).
- Cooler: **Intel E18764**.
- Buzzer pasivo: **v1.1b** (alarma sonora).
- Display: **REES52 16X2 Hd44780** (información visual).

Sensores:

- Sensor de temperatura **AM2302 / DHT22**.
- Sensor de Humedad **AM2302 / DHT22**.
- Sensor de monóxido **MQ7**.

Enlace:

- bluetooth: **HC-06**

Otras componentes:

- Cable USB-microUSB.
- Cable macho/hembra y macho/macho.
- Placa Arduino Uno.
- Fuente 12V, 2.5A.
- Fuente 5V, 350mA.
- Resistencia 220 ohm

Funcionamiento general:

Si el rango de monóxido y sensación térmica es menor a la cota inferior, el led permanece en color verde indicando que el ambiente es seguro. Si el rango de monóxido o de sensación térmica es mayor a su respectiva cota inferior y es menor a su respectiva cota superior, se muestra en el display "CUIDADO", en consecuencia el servo se acciona simulando la apertura de una ventana, y se acciona el cooler para ventilar el ambiente junto al led en color amarillo. Ahora si ambos superan sus respectivas cotas superiores, el display muestra "PELIGRO", en consecuencia se enciende un led a modo de alertar con su luz en color rojo y suena el buzzer a modo de alarma sonora.

Pasos para la creación del sketch 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 sketch a la placa arduino. Para eso, previamente tuvimos conectado la computadora al arduino mediante puerto usb, y también tenemos los sensores y actuadores mediante los pines del arduino.

Funcionamiento del sketch Arduino

Contamos con los sensores de monóxido, de temperatura y humedad que me indican el nivel de CO y sensación térmica que hay en un recinto. Mediante el display se reflejan dichos datos. Los actuadores se activarán en consecuencia a dichos valores. El accionar de los mismos está automatizado.

Si el rango de valores de CO o de sensación térmica es mayor a su respectiva cota inferior y es menor a su respectiva cota superior ponemos en funcionamiento algunos de nuestros actuadores de la siguiente manera:

- Apertura de ventana con el accionar del servomotor para permitir que el monóxido sea ventilado por la misma.
- La ventilación del ambiente se realiza mediante el accionar del cooler.
- Se genera un mensaje de alerta en el display: "CUIDADO" dependiendo de los valores procesados.
- Se enciende un led de color amarillo.

Ahora si los valores obtenidos de los sensores de monóxido o los de humedad y temperatura (sensación térmica) están por encima de su respectiva cota máxima, entonces pondremos en funcionamiento algunos de nuestros actuadores de la siguiente forma:

- El cooler se enciende.
- Se enciende un led de color rojo.
- Suena una alarma para advertir del problema.
- Se envía un mensaje de advertencia en el display que dice "PELIGRO".

A todo esto, contamos con un módulo bluetooth HC-06 (esclavo) conectado a la placa Arduino, que se queda esperando peticiones de conexión de otros dispositivos. Cuando detecta que un

dispositivo móvil con Android se conectó, dicha placa Arduino le envía los datos de monóxido, temperatura, humedad captados por dichos sensores y el cálculo de la sensación térmica, previamente solicitado por el celular.

También recibe el accionar de sus actuadores mediante la App SafeRoom instalada en el dispositivo móvil. De manera tal de poner hacer tareas programadas o pruebas de cada actuador. Todo esto mediante la conexión bluetooth.

Pasos para la creación de la App SafeRoom con Android Studio:

Se usó el Android Studio version 3.1.2 para crear la aplicación, donde se estableció las diferentes activities y las views que componente cada activity. Y por medio de cross/compiler se compiló el programa y se instaló la APK SafeRoom en el dispositivo móvil para sistemas operativos android a partir de la versión 4.2.0.

Funcionamiento de la APK SafeRoom:

Dicha aplicación necesita ser conectada mediante bluetooth a la placa arduino. Cuando se presiona el botón “Información”, el móvil le envía una solicitud de los datos necesitados al SE. Y la misma responde trayendo valores de monóxido, humedad, temperatura y sensación térmica. El usuario visualiza estos datos en la app. Si el mismo detecta un nivel elevado de impureza en el aire según las cotas establecidas, puede tomar decisiones que producirá el accionar de algunos actuadores conectados al arduino vía bluetooth:

- Si se desea ventilar la impureza del aire, el usuario presiona el botón “Encender ventilador”.
- Si el valor es crítico, el usuario optará por presionar el botón “Activar alarma”.

A su vez el usuario de la app SafeRoom, podrá activar cada actuador para hacer pruebas o porque lo considera necesario, siempre y cuando el sistema registre valores por debajo de las cotas inferiores de CO y ST. También podrá planificar un cronograma que en la fecha y hora que desee, se van a activar el cooler para bajar las temperaturas y el servomotor para abrir la ventana.

Explicaciones de actuadores

Servomotor Tower Pro SG92R 9G

El Servomotor Tower Pro SG92R 9G tiene un ángulo de giro de hasta 180 grados.

Los 3 cables en el conector están distribuidos de la siguiente forma: Rojo =Alimentación (+), Café = Alimentación (–) o tierra, Naranja= Señal PWM.

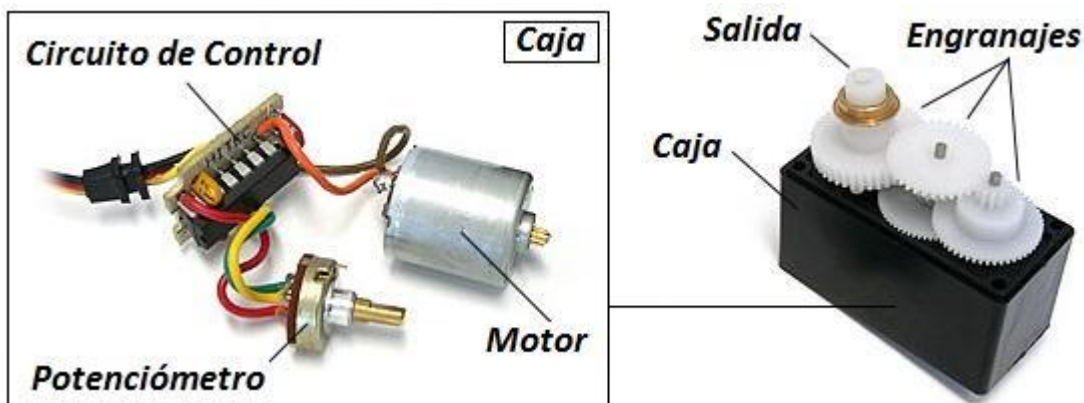
Tiene requerimientos de energía de un promedio de 0,335 Amp y 5V, se permite alimentarlo con la misma fuente de alimentación que el circuito de control, pero nosotros por problemas de tensión tuvimos que ponerlo en una fuente a parte, debido a que cuando un motor enciende consume de 6 a 9 veces su promedio y la suma de corriente de todos los dispositivos provocaba corte de energía. También por ese ruido provocado, con el capacitor que posee la nueva fuente se rectifica la corriente para que ande de forma correcta el circuito.

Características

- Micro Servo Tower-pro
- Velocidad: 0.10 sec/60° @ 4.8V
- Torque: 1.8 Kg-cm @ 4.8V
- Voltaje de funcionamiento: 3.0-7.2V
- Temperatura de funcionamiento: -30 °C ~ 60 °C
- Ángulo de rotación: 180°
- Ancho de pulso: 500-2400 μ s
- Longitud de cable de conector: 24.5cm

Los servos son motores de corriente continua (CC), pero en lugar tener un giro continuo que podamos aprovechar (para mover un molino por ejemplo), están preparados para moverse a un ángulo fijo en respuesta a una señal de control, y mantenerse fijos en dicha posición.

Un servo principalmente está formado por un conjunto reductor (engranajes), un motor de CC y por último por un circuito de control, aunque en la práctica se comporta como un bloque funcional que posiciona su eje en un ángulo preciso en función de la señal de control.

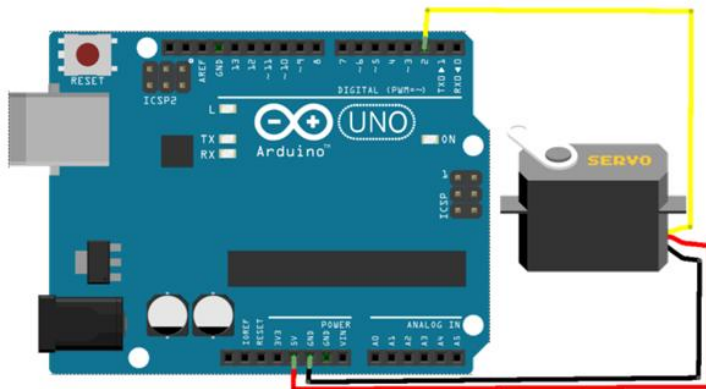


El control se realiza mediante una señal de control PWM, en la que el ancho de pulso indica el ángulo que deseamos que adopte el eje.

Tiene dos características especiales:

Por un lado, nos permite mantener una posición que indiquemos, siempre que esté dentro del rango de operación del propio dispositivo. Por otro lado nos permite controlar la velocidad de giro, podemos hacer que antes de que se mueva a la siguiente posición espere un tiempo.

Conexión



Led: Placa Led Rgb Tricolor (alarma visual)

Un led RGB es la unión de 3 leds de los colores rojo, verde, azul, en un encapsulado común, compartiendo la 'tierra' (cátodo es otro nombre que recibe el negativo). En función de la tensión que pongamos en cada pin podemos conseguir la mezcla de colores que deseemos.

DISPLAY LCD 16X2 BACKLIGHT AZUL (REES52 16X2 Hd44780)

Características

LCD alfanumérico de 2 filas por 16 columnas. Esto significa que puede mostrar 2 líneas x 16 caracteres cada una.

Tiene Controlador integrado LCD estándar HD44780 y un módulo REES52 que lo hace fácilmente adaptable al arduino.

Opera con 5V DC.

Consumo de corriente con luz de fondo total: 25 mA máx

Dimensión del módulo: 80mm x 35mm x 11mm.

Tamaño de la zona de display: 64.5mm x 16mm

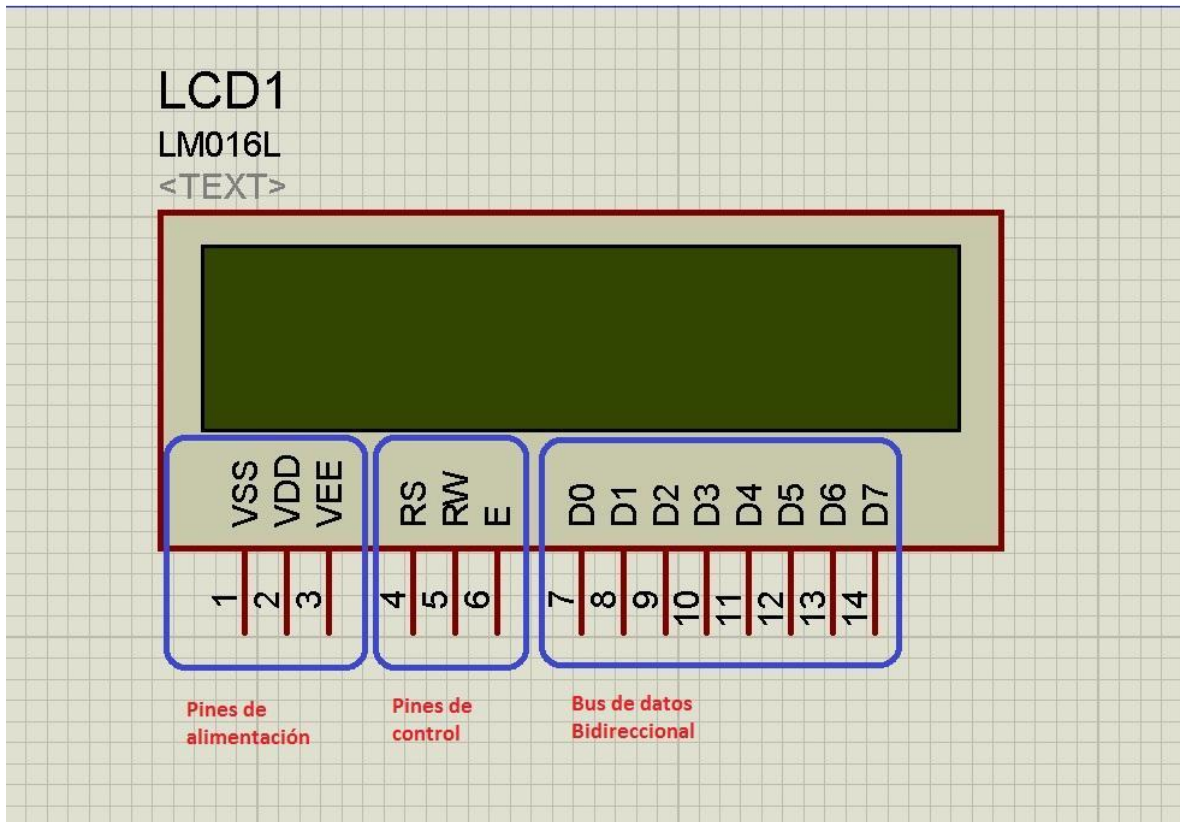
Puede mostrar letras, números, caracteres especiales, y hasta 8 caracteres creados por el usuario Interface paralela. Puede operar en modo de 8 bits, o de 4 bits para ahorrar pines del microcontrolador

En esta entrada vamos a explicar el funcionamiento y conexionado de un LCD con una resolución de 16x2.

El LCD (Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres y símbolos. Está gobernado por el microcontrolador HD44780, el cual dirige todo su funcionamiento.

¿Cómo es su conexión?

Lo podemos dividir en los Pines de alimentación, pines de control y los pines del bus de datos bidireccional.



Pines de alimentación:

Vss: Gnd

Vdd: +5

volts

Vee: corresponde al pin de contraste, lo realizaremos con un potenciómetro de 10K conectado a Vdd.

Pines de control:

RS: Corresponde al pin de selección de registro de control de datos (0) o registro de datos(1). Es decir el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de control/instrucción. y cuando RS es 1 el dato presente en el bus de datos pertenece a un registro de datos o un carácter.

RW: Corresponde al pin de Escritura(0) o de Lectura(1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla.

E: Corresponde al pin Enable o de habilitación. Si E (0) esto quiere decir que el LCD no está activado para recibir datos, pero si E (1) se encuentra activo y podemos escribir o leer desde el LCD.

Pines de Bus de datos:

El Bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos(D0 a D7) o empleando los 4 bits más significativos del bus de datos(D4 a D7).

DDRAM y CGROM?

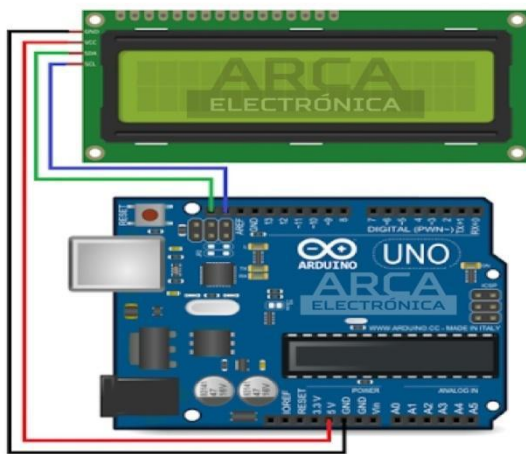
Son las dos zonas de memoria del LCD.

La memoria DDRAM (Data Display Ram): corresponde a una zona de memoria donde se almacenan los caracteres que se van a representar en pantalla. Es decir es la memoria donde se almacenan los caracteres a mostrar con su correspondiente posición.

La memoria CGROM es una memoria interna donde se almacena una tabla con los caracteres que podemos visualizar en el lcd.

La memoria CGRAM (Character Generator Ram): en ella se pueden almacenar nuestros propios caracteres.

Conexión de display al arduino



BUZZER PIEZOELÉCTRICO PASIVO V1.1B

Primero pasaremos a definir que significa piezoeléctrico. El mismo es un fenómeno que ocurre en determinados cristales que, al ser sometidos a tensiones mecánicas, adquieren una polarización eléctrica y aparece una diferencia de potencial y cargas eléctricas en su superficie que generan una tensión eléctrica.

Este fenómeno también ocurre a la inversa: se deforman bajo la acción de fuerzas internas al ser sometidos a un campo eléctrico. El efecto piezoeléctrico es normalmente reversible: al dejar de someter los cristales a un voltaje exterior o campo eléctrico, recuperan su forma.

Es decir, que son materiales (el cuarzo es el más conocido) que si los sometemos a una tensión eléctrica variable (como una señal PWM, que ya nos son familiares) vibran.

En los circuitos electrónicos digitales, suelen disponer de un reloj interno que vibra a una velocidad patrón, basados en cristales de cuarzo piezoeléctrico. El cristal de Arduino late a 16Mhz por segundo y la flecha de la foto de abajo indica la posición del mismo.

Si conectamos un piezo con una señal digital, vibran a una frecuencia sigue bastante fielmente la variación eléctrica con que los excita, y si vibran a la frecuencia audible, oiremos el sonido que

producen. A un componente que hace esto, le llamamos Buzzer o zumbador. Recordemos que el oído humano escucha en un rango de frecuencia audible, aproximadamente entre 20 Hz – 20 kHz.

Con esto, es suficiente para generar tonos audibles (como la típica alarma de los despertadores digitales) e incluso tonos musicales que podemos secuenciar.

La función Beep() de arduino me permite provocar un sonido. Lo único que beep () hace es poner una señal PWM por ejemplo, en el pin 9 de 20 sobre 255.

¿Qué pasa si quiero generar señales de tono variable para hacer una melodía? En tal caso Arduino Uno dispone de la función tone() que genera una señal de la frecuencia de audio y duración indicadas, y notone() que la corta.

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 analógica de ancho de pulso para generar varios tonos y efectos.

Especificaciones

Tensión de funcionamiento	4-8V
Salida de sonido	≥85dB
Frecuencia de resonancia	2300 ± 300Hz

Encontramos 2 tipos de buzzer:

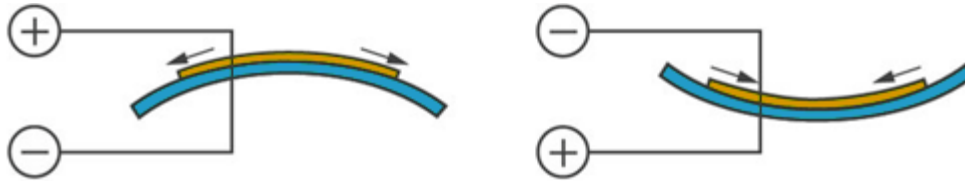
- **Buzzer ACTIVO:** Esta clase de zumbador, emite sonido cuando lo conectamos directamente a la fuente de alimentación.

- **Buzzer PASIVO:** En cambio, esta clase de zumbador, solo emite sonido si lo excitamos mediante un tren de pulsos.

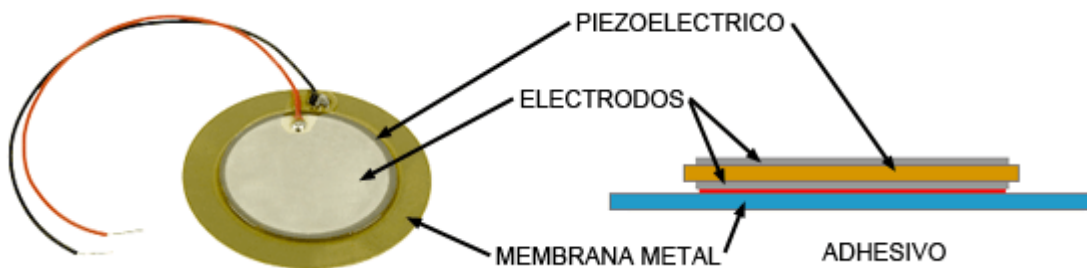
En nuestro proyecto decidimos usar buzzer pasivo. Entraremos a profundizar sobre estos mismos. Los buzzer pasivo son dispositivos que **permiten convertir una señal eléctrica en una onda de sonido**. Estos dispositivos no disponen de electrónica interna, por lo que tenemos que proporcionar una señal eléctrica para conseguir el sonido deseado. En oposición, los buzzer activos disponen de un oscilador interno, por lo que únicamente tenemos que alimentar el dispositivo para que se produzca el sonido. Pese a tener la complejidad de proporcionar y controlar nosotros la señal eléctrica, los buzzer pasivos **tienen la ventaja de que podemos variar el tono emitido** modificando la señal que aplicamos al mismo, lo que nos permite generar melodías.

Frecuentemente, el buzzer pasivo se acompaña en una placa para facilitar su conexión, que incorpora un transistor y resistencias necesarias para hacer funcionar el buzzer pasivo sin más que conectarlo.

Como hemos mencionado anteriormente, los buzzer son **dispositivos que convierten señales eléctricas en sonido**. Los buzzer son **transductores piezoeléctricos**. Los materiales piezoeléctricos tiene la propiedad especial de variar su volumen al ser atravesados por corrientes eléctricas.

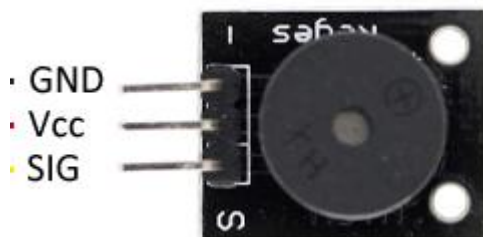


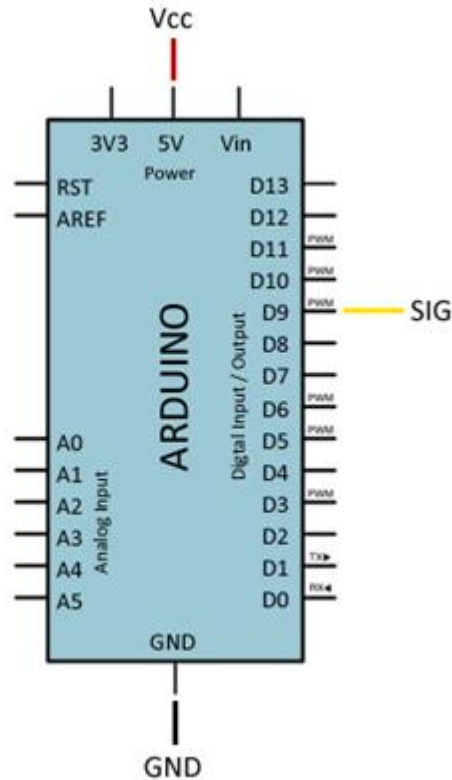
Un buzzer aprovecha este fenómeno para hacer vibrar una membrana al atravesar el material piezoeléctrico con una señal eléctrica.



Conexión de Buzzer con Arduino

La conexión es tan simple como conectar negativo a GND y positivo al pin 9. De todas maneras hay que tener cuidado. Los piezos tienen polaridad y hay que asegurarse de conectarlos correctamente. Si los conectáis al revés, simplemente no sonará, y tendréis que dar la vuelta. Ahora nos resta conectar el VCC al arduino.





COOLER

Introducción

Para nuestro proyecto, decidimos usar un cooler clásico de computadora de escritorio, ya que los mismos están estandarizados, lo que evita posibles errores de conexión. Esto es así salvo en marcas concretas que personalizan sus componentes, algo que tiende a desaparecer en la actualidad. Aun así tenemos cooler con distintas estandarización:

- Cooler de 2 cables
- Cooler de 3 cables
- Cooler de 4 cables

Cooler de 2 cables

Lo habitual es que sean de color rojo y negro, indicando el polo positivo y negativo respectivamente.

Cooler de 3 cables

Los ventiladores de tres pines suelen tener los cables de color negro, rojo y amarillo. Las líneas roja y negra siguen siendo las de alimentación, es decir positivo y negativo, y la tercera es el llamado "tacómetro". Esta línea, en las placas que dispongan de ella, indica las vueltas que las aspas del ventilador pueden dar durante un minuto. Se denomina principalmente con la abreviatura R.P.M.

Cooler de 4 cables

Esta es la última especificación introducida en ventiladores para informática. Los colores pueden tener cualquiera de estas combinaciones:

1. Negro - Rojo - Amarillo - Azul
2. Negro - Amarillo- Verde - Azul
3. Negro - Rojo - Verde - Azul

Los más comunes son el 1 y el 2. Hay que tener en cuenta que a veces el amarillo que está en tercer lugar puede ser de color blanco. Por lo demás sus funciones son siempre las mismas.

El negro será como siempre el negativo, el rojo en segundo lugar el positivo y el amarillo o verde en tercer lugar el tacómetro.

El cuarto cable es el controlador de velocidad de tecnología PWM, que por medio de pulsos digitales controla la velocidad de giro. Esto es particularmente útil cuando necesitamos controlar automáticamente su velocidad.

En nuestro proyecto usamos la siguiente especificación:

Negro - Amarillo- Verde - Azul

Explicación de sensores

Sensor MQ

Antes de explicar sensor de monóxido, vamos a aprender sobre los MQ en general.

Los sensores de gas de la serie MQ son sensores analógicos.

Estos sensores son electroquímicos y varían su resistencia cuando se exponen a determinados gases, internamente posee un calentador encargado de aumentar la temperatura interna y con esto el sensor pueda reaccionar con los gases provocando un cambio en el valor de la resistencia. El calentador dependiendo del modelo puede necesitar un voltaje entre 5 y 2 voltios, el sensor se comporta como una resistencia y necesita una resistencia de carga (RL) para cerrar el circuito y con este hacer un divisor de tensión y poder leerlo desde un microcontrolador.

Estos módulos también tienen una salida digital la cual internamente trabaja con un comparador y con la ayuda de un potenciómetro podemos calibrar el umbral y así poder interpretar la salida digital como presencia o ausencia del gas.

La diferencia entre los distintos tipos de sensores MQ es la sensibilidad a cierta gama de gases, más sensibles a algunos gases que a otros, pero siempre detectan a más de un gas.

El calentador

El voltaje para el calentador interno es muy importante. Algunos sensores usan 5V para el calentador, otros necesitan 2V. Este calentador es necesario para elevar la temperatura del sensor. . Mientras el calentador no alcance la temperatura de funcionamiento, la lectura del sensor no será fiable.

El tiempo de calentamiento depende de cada modelo de sensor. En la mayoría de modelos es suficiente para con unos pocos minutos.

Resistencia de carga

El sensor necesita una resistencia de carga en la salida a tierra. Su valor podría ser de 2kOhm a 47kOhm. Cuanto menor es el valor, menos sensible. Cuanto mayor sea el valor, menos precisa será para concentraciones más altas de gas.

Si se mide solo un gas específico, la resistencia de carga se puede calibrar aplicando una concentración conocida de ese gas.

Todos los modelos MQ disponen de un calentador necesario para elevar la temperatura del sensor, y que sus materiales adquieran la sensibilidad. Mientras el calentador no alcance la temperatura de funcionamiento, la lectura del sensor no será fiable.

El tiempo de calentamiento depende de cada modelo de sensor. En la mayoría de modelos es suficiente para con unos pocos minutos

Sensor de Monóxido de Carbono MQ7

Este sensor es de alta sensibilidad al monóxido de carbono (CO), pero también es sensible al H₂ (Dihidrógeno, de fórmula química H₂, gas cuya molécula se compone de 2 átomos de hidrógeno).

Este es un sensor fácil de usar para detección de Monóxido de Carbono (CO), ideal para detectar concentraciones de CO en el aire. El MQ-7 puede detectar concentraciones en el rango de 20 a 2000ppm.

Este sensor tiene una alta sensibilidad y un corto tiempo de respuesta.

El módulo posee una salida analógica que proviene del divisor de voltaje que forma el sensor y una resistencia de carga. También tiene una salida digital que se calibra con un potenciómetro, esta salida tiene un led indicador. Todo lo que se necesita es proporcionarle 5V, añadir una resistencia de carga y conectar la salida a una entrada ADC de un microcontrolador.

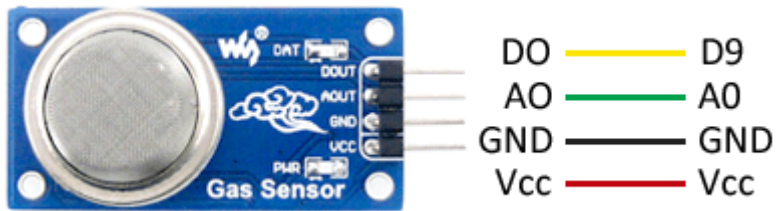
El calentador usa una tensión alterna de 5V y 1.4V. Y para q el sensor arroje los valores deseados, se necesita esperar a dicho calentador por 2 minutos.

Características:

- Detección de monóxido de carbono
- Alimentación 5V DC ó AC
- Temperatura de funcionamiento: -10 °C a 50 °C
- Consumo de potencia: menos de 750 mW
- Humedad relativa de operación: 65% ± 5%
- Salida analógica
- Voltaje de alimentación: 5 V ±0.1 V
- Resistencia del calefactor: 33 Ω ±5%
- Potencia consumida por el calefactor: 350 mW

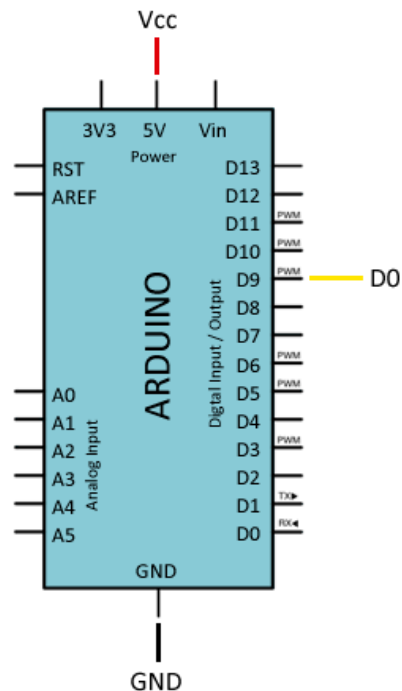
CONEXIÓN MQ7 CON ARDUINO

El esquema eléctrico es sencillo. Alimentamos el módulo conectando GND y 5V a los pines correspondientes de Arduino.

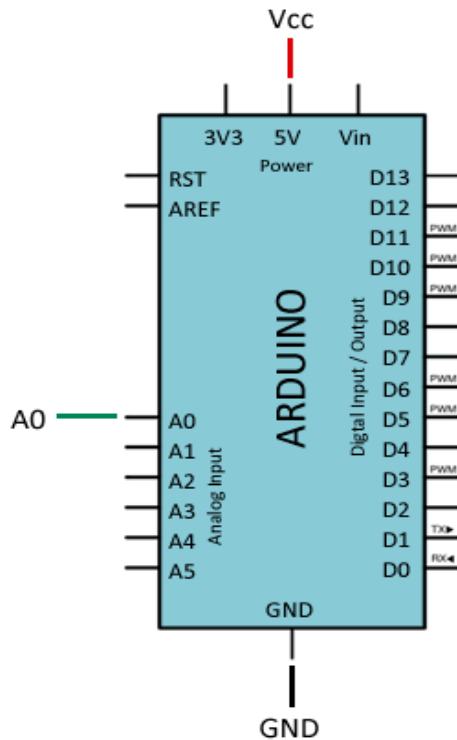


Ahora si queremos usar la lectura digital, conectamos la salida DO a una de las entradas digitales de Arduino.

La conexión vista desde Arduino quedaría así,



Si quisiéramos emplear el valor analógico, simplemente conectaremos la salida AO del sensor a una entrada analógica de Arduino. (En nuestro proyecto usamos salida analógica)



SENSOR DHT22

El DHT22 es un sensores digitales de Temperatura y Humedad, fáciles de implementar con cualquier microcontrolador. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante y solo un pin para la lectura de los datos. Tal vez la desventaja de estos es la velocidad de las lecturas y el tiempo que hay que esperar para tomar nuevas lecturas (nueva lectura después de 2 segundos), pero esto no es tan importante puesto que la Temperatura y Humedad son variables que no cambian muy rápido en el tiempo.

El rango de medición de temperatura es de -40°C a 80°C con precisión de $\pm 0.5^{\circ}\text{C}$ y rango de humedad de 0 a 100% RH (humedad relativa) con precisión de 2% RH, el tiempo entre lecturas debe ser de 2 segundos.



CARACTERÍSTICAS

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°C hasta 80°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°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 tests indicaron que la velocidad del aire debe ser de 1 m/s.

Funcionamiento

CICLO DE OPERACIÓN

Es el tiempo que el sensor tarda en ofrecer una respuesta desde que se le pide. Formato de datos de un solo bus para la comunicación y sincronización entre MCU y el sensor. El proceso de comunicación es de 4 ms aproximadamente.

Una transmisión de datos completa es de 40 bits. Donde obtenemos la temperatura y la humedad.

Ejemplo: Recibimos 40 bits:

- **0011 0101 0000 0000 0001 1000 0000 0000 0100 1101**

Los primeros 8 bit son de humedad en alto-high

Los segundos 8 bit son de (humedad en bajo-low

Los terceros 8 bit son de temperatura en alto-high

Los cuartos 8 bit son de temperatura en bajo-low

Los últimos 8 bit son de paridad

Tenemos un total de=40 bits

Calculando :

- **0011 0101+0000 0000+0001 1000+0000 0000= 0100 1101**

Datos correctos recibidos :

- Humedad : 0011 0101 = 35H = **53%RH**
- Temperatura : 0001 1000 = 18H = **24°C**

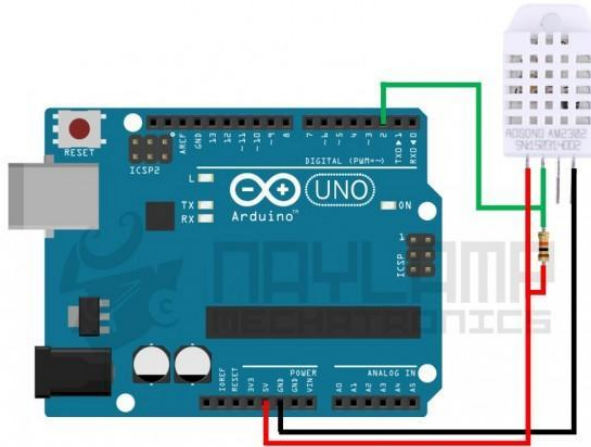
El microcontrolador externo y el microcontrolador que lleva integrado el sensor, se hablan entre sí de la siguiente manera:

- Se inicia la comunicación.
- El sensor responde estableciendo un nivel bajo de 80us y un nivel alto de 80us.
- El sensor envía 5 bytes con la información de temperatura y humedad.

Conexion

1	VCC
2	DATA
3	NC
4	GND

Tiene 4 pines. Pero para nuestro proyecto nos basta con usar el pin 1 (para voltaje), pin 2 (para lectura de datos digitales). Este pin se conecta al microcontrolador y el pin 4 (GND)



OTRAS COMPONENTES:

BLUETOOTH

Modulo Bluetooth HC-05 y HC-06

Los módulos de bluetooth HC-05 y HC-06 sirve para la comunicación entre distintos dispositivos. Además de recibir conexiones desde un dispositivo bluetooth, también es capaz de generar conexiones hacia otros dispositivos bluetooth. Ahora Un módulo Bluetooth HC-06 se comporta como esclavo, esperando peticiones de conexión. Presentan un modo de configuración y un modo de utilización y se configuran mediante comandos AT.

La tarjeta incluye un adaptador con 4 pines. Los pines de la board correspondientes son:

- VCC
- GND
- RX
- TX

Además posee un regulador interno que permite su alimentación de 3.6 a 6V.

- **Vcc**, Voltaje positivo de alimentación, aquí hay tener cuidado porque hay módulos que solo soportan voltajes de 3.3V, pero en su mayoría ya vienen acondicionados para q trabajen en el rango de 3.3V a 6V pero es bueno revisar los dato técnicos de nuestro módulo antes de hacer las conexiones

- **GND**, Voltaje negativo de alimentación, se tienen que conectar al GND del Arduino o al GND de la placa que se esté usando.

- **TX**, Pin de Transmisión de datos, por este pin el HC-06 transmite los datos que le llegan desde la PC o Móvil mediante bluetooth, este pin debe ir conectado al pin RX del Arduino

- **RX**, pin de Recepción, a través de este pin el HC-06 recibirá los datos del Arduino los cuales se transmitirán por Bluetooth, este pin va conectado al Pin TX del Arduino

Ahora desde parte del dispositivo móvil usamos el modo utilización de la siguiente manera:

1. Activar o desactivar Bluetooth
2. Buscar dispositivos cercanos al móvil de Bluetooth con BroadcastReceiver. En este caso se van a mostrar un listado con los bluetooth cercanos encontrados
3. Emparejar ambos el HC05 con el Bluetooth de Android. Con BroadcastReceiver se obtiene la MAC del HC-06
4. Realizar la transferencia de datos entre Arduino y Android a través del bluetooth. Esto se debe realizar en otro hilo de conexión. De esta manera tenemos 2 thread: el principal y el secundario, este último es quien recibe y envía datos a arduino.

Ahora el Arduino, recibe datos desde android y en base a ellos realiza acciones. Le envía a Android lo que fue ingresado en el Monitor serie.

Bluetooth Android necesita de 3 objetos para poder usar dicha componente:

BluetoothAdapter

- Permite realizar todo tipo de operaciones sobre el bluetooth del móvil.
- Activa o desactiva el bluetooth
- Realiza la búsqueda de dispositivos bluetooth cercanos y los almacena en una lista
- Me permite obtener datos de conectividad del bluetooth de Arduino, que se van a guardar en una objeto BluetoothDevice

BluetoothDevice

- Representa un dispositivo bluetooth con sus atributos y propiedades como nombre, clase, velocidad de transmisión, MAC, estado.
- Es utilizado para realizar el emparejamiento
- Nos da acceso a la transmisión de datos a través de BluetoothSocket

BluetoothSocket

- Me permite manejar un socket Bluetooth
- Establece la conexión y desconexión para la transmisión de datos (**connect y close**)
- Permite recibir datos del bluetooth remoto usando un InputStream y enviarlos usando un outputStream

La comunicación Bluetooth entre dos módulos debe realizarse entre un módulo configurado como maestro y otro como esclavo. Para la comunicación Bluetooth con computador, teléfono, PDA, tableta, etc., dicho módulo debe ser esclavo.

Configuración:

El módulo suele venir configurado con velocidad de transmisión serial de 9600 bps, 1 bit de parada, y sin bit de paridad, nombre: linvor, password: 1234.

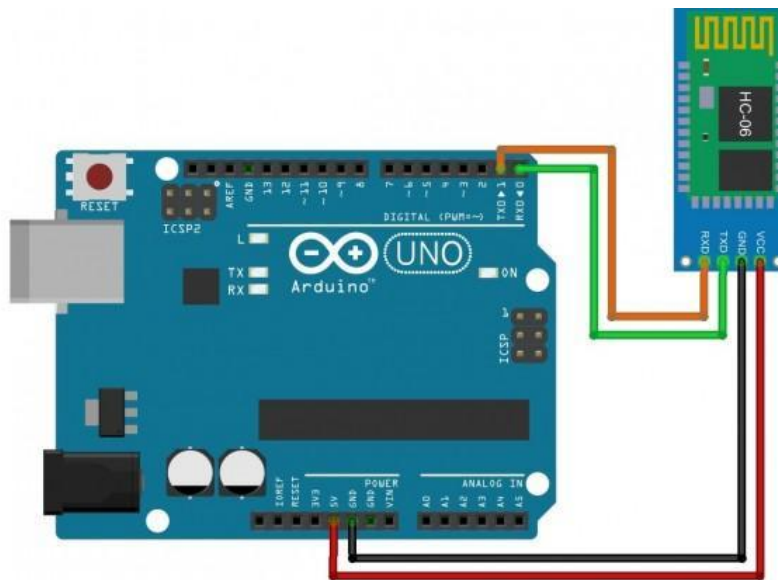
Usamos el monitor serial del IDE de Arduino para escribir los comandos AT. Como este monitor emplea la comunicación serial que el Arduino utiliza para comunicarse con la computadora en los pines 0 y 1 digitales, se crea un puerto serial por software para pasar los datos al módulo Bluetooth empleando otros pines digitales.

Una simple diferencia es que el módulo HC-06 funciona como Slave y el HC-05 como Master y Slave. Físicamente se diferencian por el número de pines. En el HC-06 tiene un conector de 4 pines mientras que el HC-05 trae uno de 6 pines.

Características

- Especificación bluetooth v2.0 + EDR (Enhanced Data Rate)
- Modo esclavo (Solo puede operar en este modo)
- Puede configurarse mediante comandos AT (Deben escribirse en mayúscula)
- Chip de radio: CSR BC417143
- Frecuencia: 2.4 GHz, banda ISM
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Antena de PCB incorporada
- Potencia de emisión: ≤ 6 dBm, Clase 2
- Alcance 5 m a 10 m
- Sensibilidad: ≤ -80 dBm a 0.1% BER
- Velocidad: Asíncrona: 2 Mbps (max.)/160 kbps, síncrona: 1 Mbps/1 Mbps
- Seguridad: Autenticación y encriptación (Password por defecto: 1234)
- Perfiles: Puerto serial Bluetooth
- Módulo montado en tarjeta con regulador de voltaje y 4 pines suministrando acceso a VCC, GND, TXD, y RXD
- Consumo de corriente: 30 mA a 40 mA
- Voltaje de operación: 3.6 V a 6 V
- Dimensiones totales: 1.7 cm x 4 cm aprox.
- Temperatura de operación: -25°C a $+75^{\circ}\text{C}$

CONEXIÓN



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 20KΩ y 50 KΩ 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.

I2C: Permite establecer comunicaciones a través de un bus I 2C. El bus I 2C 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

Entre los nombres más salientes que acabamos de mencionar encontramos:

- Tecnología TTL
- I2C
- **UART**
- SPI

Ahora pasaremos a explicar cada una de ellas.

TTL es la sigla en inglés de *transistor-transistor logic*, es decir, «**lógica transistor a transistor**». Es una tecnología de construcción de circuitos electrónicos digitales.

I2C es un bus de comunicaciones en serie. La principal característica de I2C es que utiliza dos líneas para transmitir la información: una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Como suelen comunicarse circuitos en una misma placa que comparten una misma masa esta tercera línea no suele ser necesaria.

Las líneas se llaman:

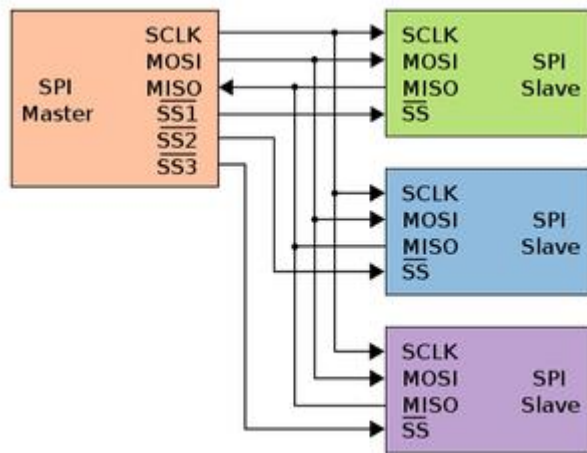
- SDA: datos
- SCL: reloj
- GND: tierra

Dentro la comunicación serie integrada en los microcontroladores de Arduino tenemos:

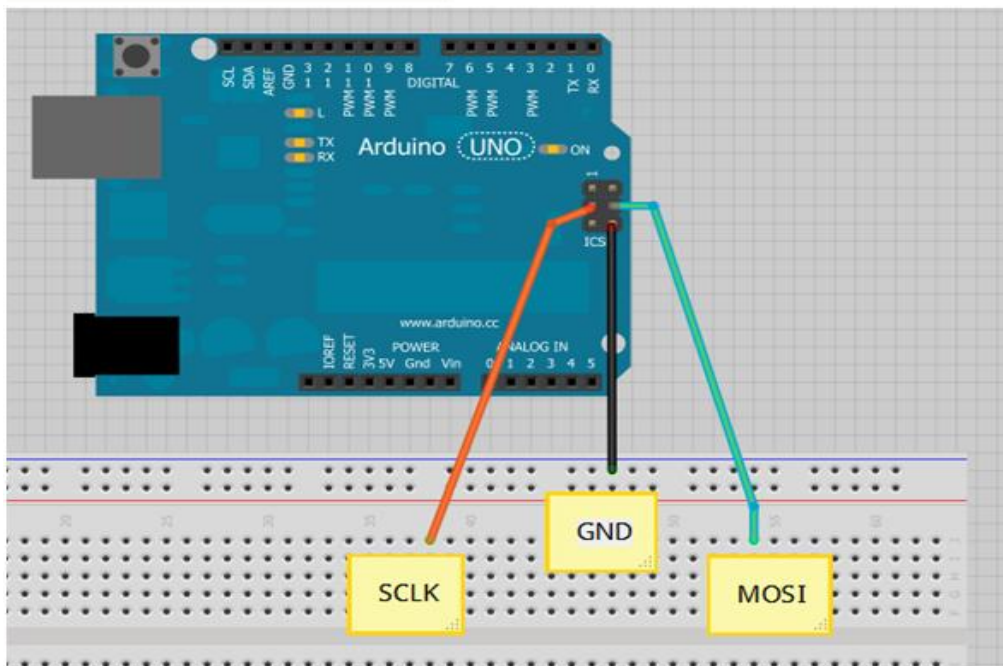
UART (recepción-transmisión asíncrona universal) es uno de los protocolos serie más utilizados. La mayoría de los microcontroladores disponen de hardware UART. Usa una línea de datos simple para transmitir y otra para recibir datos. Comúnmente, 8 bits de datos son transmitidos de la siguiente forma: un bit de inicio, a nivel bajo, 8 bits de datos y un bit de parada a nivel alto. UART se diferencia de SPI y I2C en que es asíncrono y los otros están sincronizados con señal de reloj. La velocidad de datos UART está limitado a 2Mbps

SPI es otro protocolo serie muy simple. Es un bus de comunicación a nivel de circuitos integrados. La transmisión de datos se realiza en serie, es decir un bit después de otro. El bus SPI se define mediante 4 pines:

- SCLK o SCK : Señal de reloj del bus. Esta señal rige la velocidad a la que se transmite cada bit.
- MISO(Master Input Slave Output – Entrada maestra, salida esclava): Es la señal de entrada a nuestro dispositivo, por aquí se reciben los datos desde el otro integrado.
- MOSI(Master Output Slave Input – Salida maestra, entrada esclava): Transmisión de datos hacia el otro integrado.
- SS o CS: Chip Select o Slave Select, habilita el integrado hacia el que se envían los datos. Esta señal es opcional y en algunos casos no se usa.



Hay que tener en cuenta que a la vez que el Master está enviando un dato también está recibiendo datos. Por ende permite la comunicación full duplex. Como desventaja, funciona a distancias cortas.



PWM

Antes de definir que es PWM debemos decir que un sistema de control (como un microcontrolador) no tiene capacidad para trabajar con señales analógicas, de modo que necesita convertir las señales analógicas en señales digitales para poder trabajar con ellas. Arduino Uno

tiene una resolución de 10 bits, es decir, unos valores entre 0 y 1023 para la conversión de analógico a digital. 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). Los microcontroladores de Arduino contienen en la placa un conversor analógico a digital de 6 canales. . La nomenclatura para los pines analógicos es A0, A1, etc... y sirven como entrada analógica; pero el mismo no tiene salidas analógicas. Para solucionar esto usa la técnica PWM

Las Salidas PWM (modulación por ancho de pulso) 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. El control digital se usa para crear una onda cuadrada, una señal conmutada entre encendido y apagado. Este patrón de encendido-apagado puede simular tensiones entre el encendido total (5 voltios) y el apagado (0 voltios) al cambiar la porción del tiempo que la señal pasa frente al tiempo que la señal se apaga. La duración de "encendido" se llama ancho de pulso. Para obtener valores pwm variables, cambia o modula ese ancho de pulso. Si repite este patrón de encendido-apagado lo suficientemente rápido con un LED, por ejemplo, el resultado es como si la señal fuera un voltaje constante entre 0 y 5 v que controla el brillo del LED.

En el siguiente gráfico, las líneas verdes representan un período de tiempo regular. Esta duración o período es el inverso de la frecuencia de PWM. En otras palabras, con la frecuencia de PWM de Arduino a aproximadamente 500Hz, las líneas verdes medirían 2 milisegundos cada una. Una llamada a `analogWrite()` está en una escala de 0 - 255, de modo que `analogWrite(255)` solicita un ciclo de trabajo del 100% (siempre activado), y `analogWrite(127)` es un ciclo de trabajo del 50% (en la mitad del tiempo) para ejemplo.

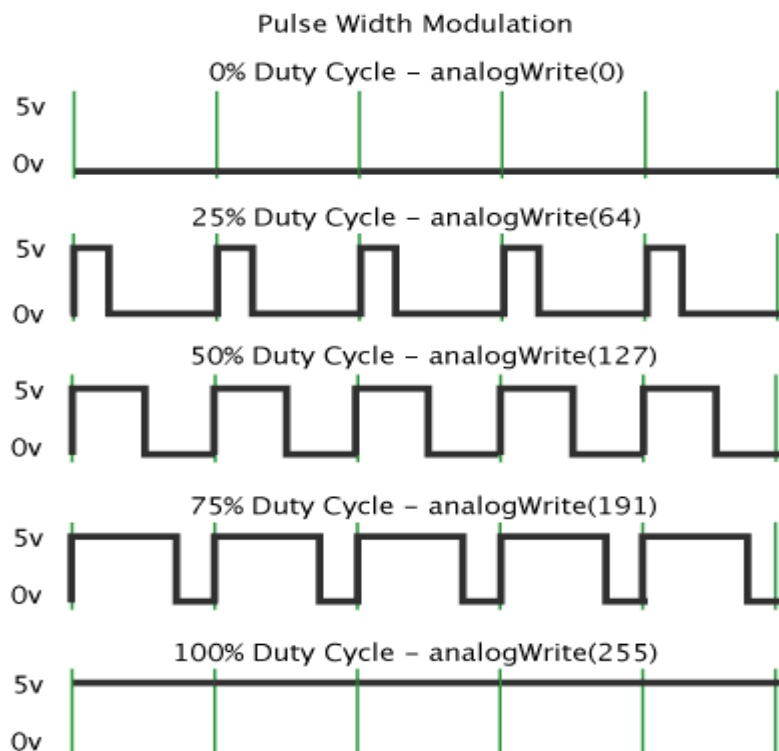


DIAGRAMA EN BLOQUE

DIAGRAMA FUNCIONAL

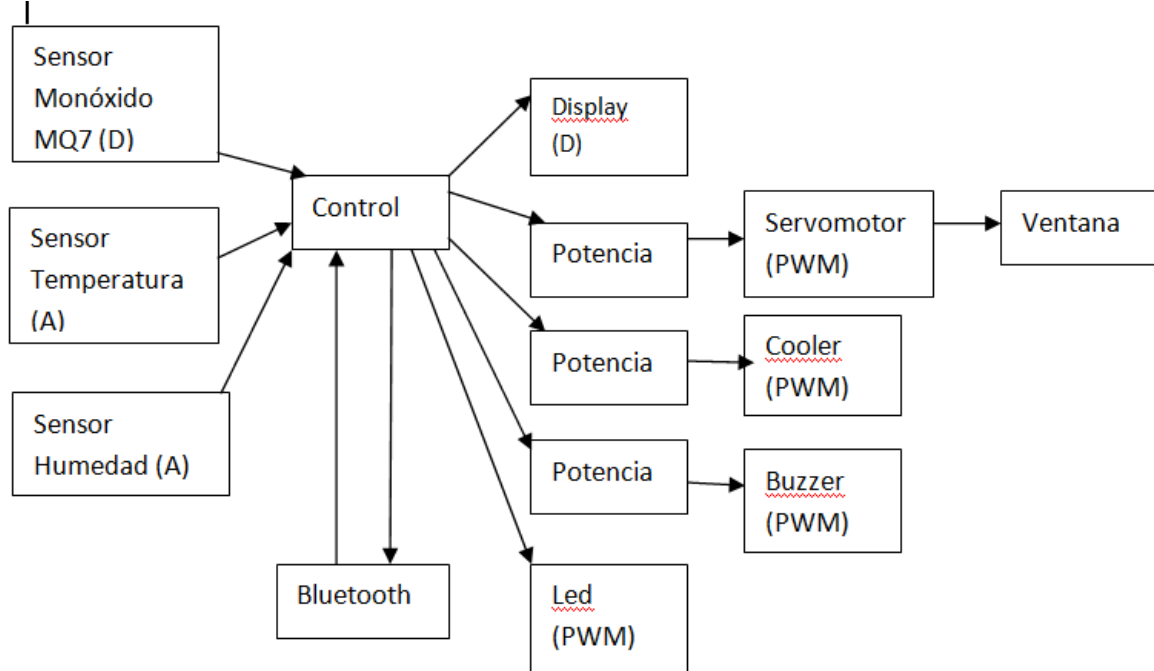


DIAGRAMA FÍSICO

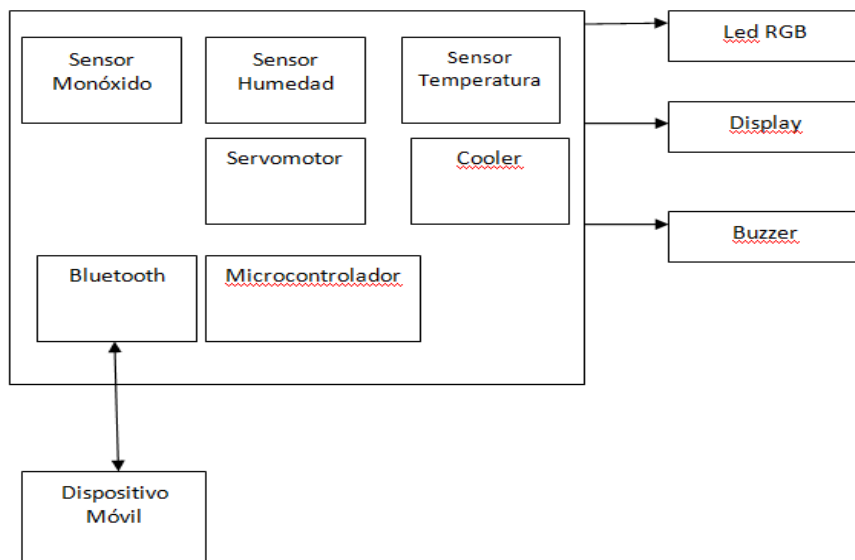


DIAGRAMA DE SW

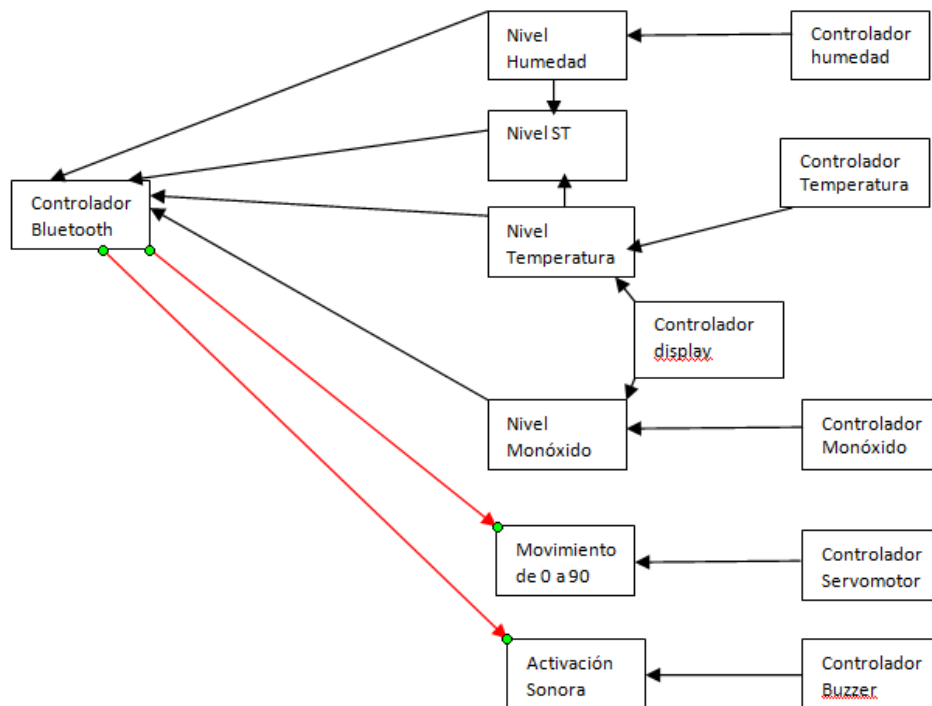


DIAGRAMA LÓGICO

