

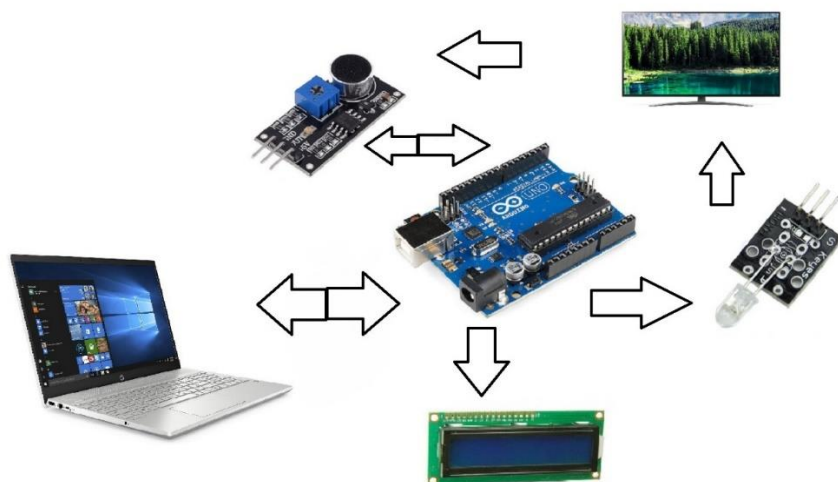
DATOS DE LOS MIEMBROS DEL GRUPO.

NOMBRE	APELLIDOS	Nº MATRÍCULA	EMAIL
Rafael	Rodríguez Palomo	55435	rafael.rodriquezp@alumnos.upm.es

TÍTULO Y RESUMEN.

CONTROLADOR VOLUMEN DE AUDIO

Dispositivo electrónico basado en un microcontrolador capaz de monitorizar el volumen de audio de un determinado reproductor a través de un sensor de sonido. La intensidad del sonido será monitorizada por el ordenador, si el volumen supera un determinado nivel establecido por el usuario un emisor de infrarrojos regulará el volumen de forma automática. Cuando se produzca una incidencia quedará registrada en el ordenador. A su vez una pantalla OLED o LCD mostrará un gráfico dinámico que muestre las fluctuaciones del volumen de manera que sea fácil de interpretar por el usuario.



REQUISITOS FUNCIONALES

1. Es necesario un ordenador para el funcionamiento del dispositivo.
2. Al iniciar el dispositivo es necesario establecer un máximo de volumen.
3. Cada cierto intervalo de tiempo el sensor de audio recibe la intensidad de sonido.
4. Las incidencias aparecen en la pantalla del ordenador.

HARDWARE Y FUNDAMENTOS TÉCNICOS.

MEDIR INTENSIDAD DEL SONIDO – SENSOR DE SONIDO CON MICRÓFONO MAX9812

El MAX9812 es un amplificador de sonido especialmente diseñado para tratar la señal proveniente de micrófonos. Podemos emplearlo para medir sonido desde un procesador como Arduino.

En los módulos con micrófono la señal generada es demasiado baja para ser registrada correctamente. Si queremos poder medir sonidos, deberemos amplificar la señal.

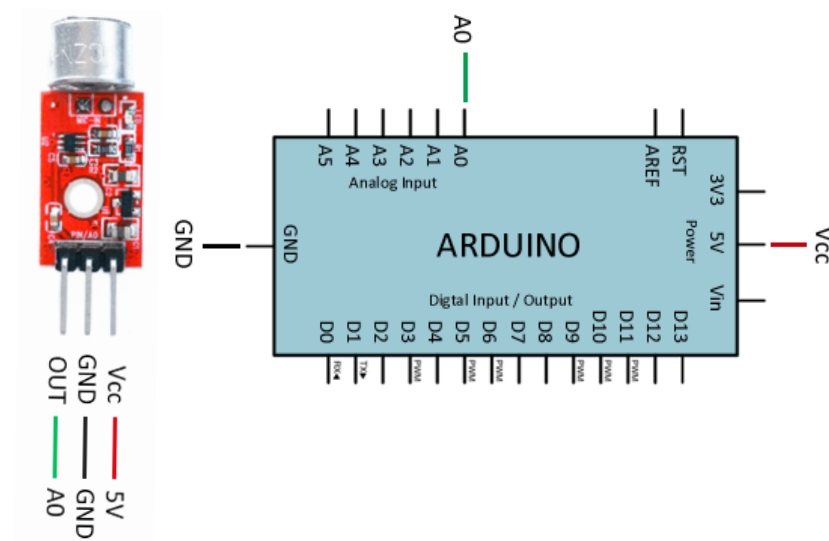
El MAX9812 dispone de entradas single/dual, ganancia fija de 20dB, ancho de banda de 500 kHz, bajo ruido y baja distorsión de señal 0.015% THD.

Podemos emplear los sensores con MAX9812 para aplicaciones que requieran medir la intensidad o frecuencia del sonido. Por ejemplo, podemos variar la iluminación de una tira LED en función del sonido, o mostrar un ecualizador en una pantalla TFT.

El esquema eléctrico es sencillo. Alimentamos el módulo conectando GND y 5V a los pines correspondientes de Arduino. Por otro lado, conectamos la salida analógica del sensor a una entrada analógica de Arduino.

Cuando manejemos sonidos lo primero que debemos entender es que la señal que obtenemos varía rápidamente y además tiene mucho ruido. De esta forma, el valor puntual de una medición no suele tener importancia, y deberemos realizar la integración en un determinado periodo. Para ello lo habitual es trabajar con ventanas temporales, es decir, un intervalo en el que realizamos la medición.

Para ello, definimos una ventana temporal de 50 ms, equivalente a una frecuencia de 20 Hz, y calculamos el máximo y mínimo registrado dentro de la ventana. A continuación, se muestra el valor registrado por puerto serie.



Ejemplo de código que puede servir como referencia.

```
1  const int sensorPIN = A0;
2  const int sampleWindow = 50; // Ancho ventana en ms (50 ms = 20Hz)
3
4  void setup()
5  {
6    Serial.begin(9600);
7  }
8
9  void loop()
10 {
11   unsigned long startMillis= millis();
12
13   unsigned int signalMax = 0;
14   unsigned int signalMin = 1024;
15
16   // Recopilar durante la ventana
17   unsigned int sample;
18   while (millis() - startMillis < sampleWindow)
19   {
20     sample = analogRead(sensorPIN);
21     if (sample < 1024)
22     {
23       if (sample > signalMax)
24       {
25         signalMax = sample; // Actualizar máximo
26       }
27       else if (sample < signalMin)
28       {
29         signalMin = sample; // Actualizar mínimo
30       }
31     }
32   }
33   unsigned int peakToPeak = signalMax - signalMin; // Amplitud del sonido
34   double volts = (peakToPeak * 5.0) / 1024; // Convertir a tensión
35   Serial.println(volts);
36 }
```

REGULAR VOLUMEN – EMISOR DE INFRARROJOS KY-005

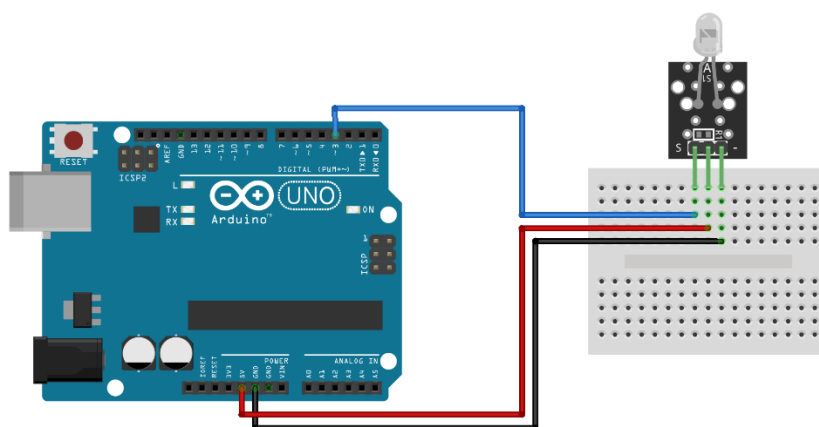
Este módulo de Keyes contiene un led emisor de luz infrarroja y una resistencia limitadora de corriente. La línea central del módulo es la alimentación, y se conecta a los +5V del Arduino. La línea marcada con el signo - va conectada a GND del Arduino. La línea marcada con la letra S va conectada a la línea digital 2.

Utiliza la biblioteca IRremote para enviar señales de infrarrojos en serie con el KY-005.

La conexión del pin de salida la determina la biblioteca: la entrada digital 3 en Arduino Uno. Depende de la placa que se esté utilizando, de modo que si utiliza otra se debe verificar la documentación de la biblioteca de IRremote.

La biblioteca remota IRremote permite enviar y recibir códigos remotos de IR en múltiples protocolos. Es compatible con **NEC**, **Sony SIRC**, **Philips RC5**, **Philips RC6** (es recomendable buscar información sobre estos protocolos para entender mejor su funcionamiento) y protocolos sin formato. Si se necesitan protocolos adicionales, son fáciles de agregar. Incluso la biblioteca puede utilizarse para grabar códigos desde su control remoto y retransmitirlos, como un control remoto universal mínimo.

Es recomendable implementar un KY-022 para probar el KY-005.



Ejemplo de código:

Este programa envía un código de encendido/apagado de Sony TV cada vez que se envía un carácter al puerto serie, lo que permite que Arduino encienda o apague el televisor. (Los códigos de Sony deben enviarse 3 veces de acuerdo con el diseño del protocolo).

```
Prueba de modulo de led infrarrojo KY-005
1  /*
2   * Prueba de modulo de led infrarrojo KY-005
3   */
4  #include <IRremote.h>
5  IRsend irsend; // iniciar el objeto infrarrojo
6
7  void setup()
8  {
9    Serial.begin(9600); // iniciar comunicacion serie
10 }
11 void loop()
12 {
13   if (Serial.read() != -1) {
14     for (int i=0; i<3; i++) {
15       irsend.sendSony(0xa40, 17); // codigo de encendido de Sony TV
16       delay(1000);
17     }
18   }
19 }
```

INTERFAZ – PANTALLA LCD

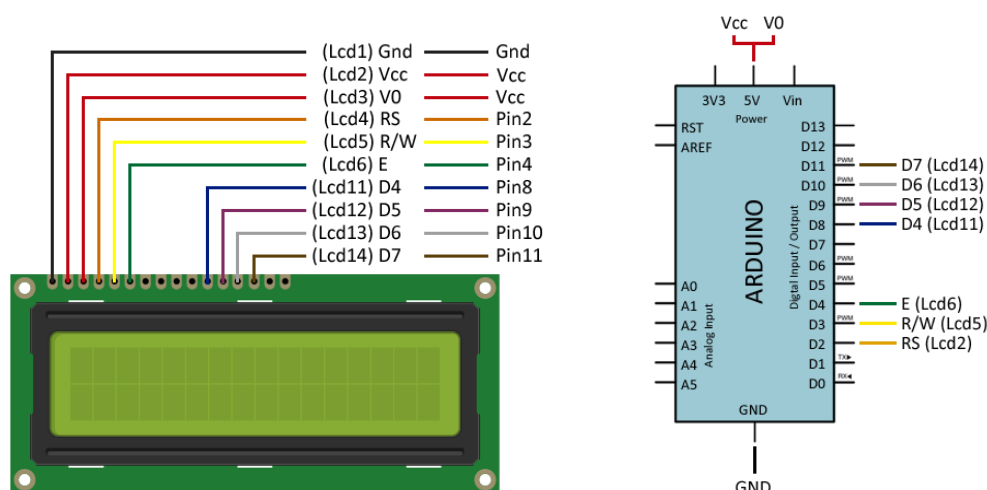
Las pantallas LCD (Liquid Cristal Display) son una de las formas más sencillas y económicas de dotar de un display a un autómata.

El Hitachi HD44780 es uno de los controladores de LCDs más ampliamente extendidos por su sencillez y bajo precio. Está diseñado para controlar LCDs monocromos de hasta 80 caracteres alfanuméricos y símbolos. También dispone de una pequeña memoria RAM para configurar nuestros propios caracteres o dibujos.

Las pantallas LCD se fabrican en distintos tamaños, siendo comunes 16x02 (2 líneas de 16 caracteres).

Conectar directamente un LCD a Arduino requiere una gran cantidad de pines. Suele ser aconsejable emplear un adaptador a bus I2C. Por este motivo es posible que en el controlador de audio no sea posible incluir una pantalla LCD por el número de pines disponibles en la placa Arduino.

El IDE de Arduino incorpora de serie la librería LiquidCrystal, para controlar pantallas LCD HD44770 de forma sencilla. La librería LiquidCrystal varios ejemplos sobre su uso. Resulta aconsejable examinarlos, en general es sencilla de utilizar.



Ejemplo de código:

Por ejemplo, el siguiente sketch muestra el uso de las funciones de la librería para mostrar los textos "Linea 1" y "Linea 2" en una pantalla de 16x02.

```
1  /* Conexión
2  PIN2 - LCD4 (RS)
3  PIN3 - LCD5 (R/W)
4  PIN4 - LCD6 (E)
5
6  PIN8 - LCD11 (D4)
7  PIN9 - LCD12 (D5)
8  PIN10 - LCD13 (D6)
9  PIN11 - LCD14 (D7)
10 */
11
12 #include <LiquidCrystal.h>
13
14 LiquidCrystal lcd(2, 3, 4, 0, 9, 10, 11); // (RS, RW, E, D4, D5, D6, D7)
15
16 void setup()
17 {
18   lcd.begin(16, 2); // Inicia un LCD 16x02 (columnas, filas)
19   lcd.setCursor(0, 0); // Pone el cursor en las coordenadas (0,0)
20   lcd.print("Linea 1"); // Escribe el LCD
21   lcd.setCursor(0, 1); // Ponemos el cursor en la segunda fila
22   lcd.print("Linea 2"); // Escribe el LCD
23 }
24
25 void loop()
26 {
27   //Apagar cursor
28   lcd.noCursor();
29   delay(500);
30
31   //Apagar encender cursor
32   lcd.cursor();
33   delay(500);
34 }
```

POTENCIÓMETRO DE 10K

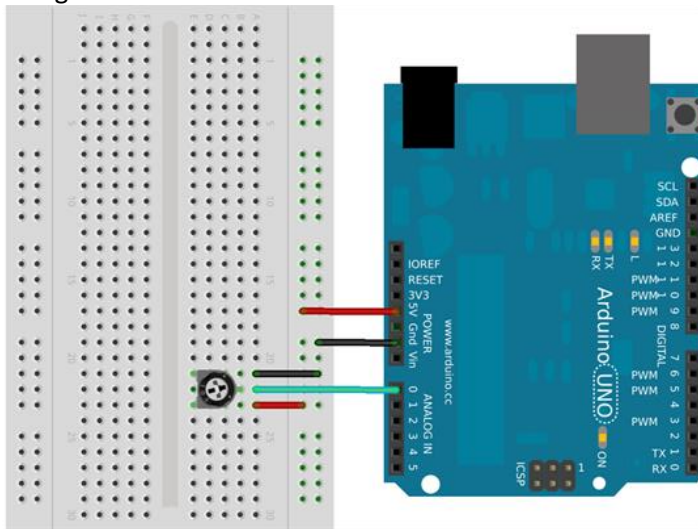
Los potenciómetros son dispositivos que se usa frecuentemente para introducir valores analógicos en nuestro Arduino, por ejemplo, para regular la intensidad de la luz de un Led, la velocidad de un motor DC, o la posición de un servo.

La medición del valor de un potenciómetro es similar (aunque no igual) a la medición de la resistencia desconocida.

Un potenciómetro es un dispositivo que permite variar su resistencia de forma manual, entre un valor mínimo R_{min} , (normalmente 0 ohmios) y un valor máximo R_{max} . Valores habituales de R_{max} son 5k, 10k o 20k ohmios.

Internamente un potenciómetro está constituido por un contacto móvil que se desplaza a lo largo de una pista resistiva. De esta forma, al mover el potenciómetro se mueve el contacto a lo largo de la pista, y variando la longitud del tramo de pista con el que estamos en contacto, y por tanto variando su resistencia.

Normalmente un potenciómetro tiene tres terminales. Los dos extremos están unidos a ambos lados de la pista, por lo que siempre registrarán la resistencia máxima R_{max} . El terminal restante corresponde con el contacto móvil. Esta terminal varía su resistencia respecto a los otros dos terminales a medida que accionamos el potenciómetro, siendo la suma de la resistencia a los otros terminales igual a R_{max} .



Ejemplo de código del potenciómetro

```
1  const int analogPin = A0;
2  int value;          //variable que almacena la lectura analógica raw
3  int position;       //posicion del potenciómetro en tanto por ciento
4
5  void setup() {
6  }
7
8  void loop() {
9      value = analogRead(analogPin);          // realizar la lectura analógica raw
10     position = map(value, 0, 1023, 0, 100); // convertir a porcentaje
11
12     //...hacer lo que se quiera, con el valor de posición medido
13
14     delay(1000);
15 }
```

