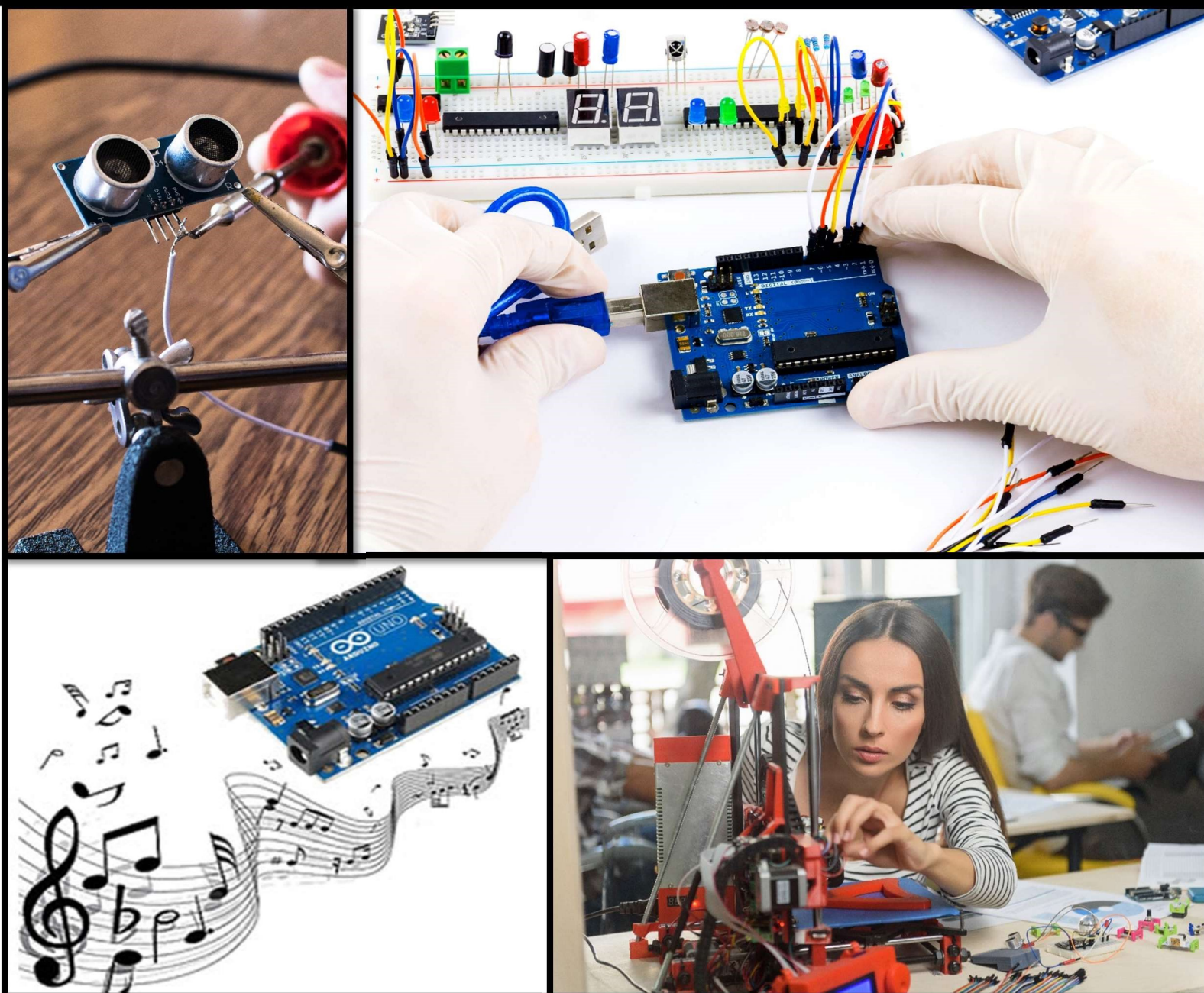


Arduino II – Reproductor MP3



Índice

1. Datos generales	1
2. Introducción	1
2.1. Objetivos	1
2.2. Descripción del curso	1
3. Sonido con Arduino	2
3.1. Aplicaciones.....	2
3.2. El altavoz	3
3.3. El amplificador de audio PAM8406.....	3
3.4. Mini reproductor MP3 DFPlayer	5
3.5. Circuito 1: Reproductor MP3 DFPlayer Mini modo autónomo.....	7
3.6. Comunicación por puerto serie.....	8
3.7. Sentencias para comunicación serie – IDE Arduino.....	11
3.8. Sentencias para comunicación serie – mBlock	12
3.9. Circuito 2: Enviando un mensaje por puerto serie	12
3.10. Librería para el reproductor DFPlayer Mini.....	12
3.11. Circuito 3: Reproductor con Arduino	14
4. Pantalla LCD.....	15
4.1. Pantalla LCD 16X2.....	15
4.2. Módulo I2C para pantallas LCD	16
4.3. Bus de datos I2C.....	18
4.4. Librerías para pantallas LCD y comunicación I2C.....	19
5. Proyecto: Reproductor de música MP3	21
5.1. Lista de materiales	21
5.2. Esquema electrónico.....	23
5.3. Programación del reproductor de música MP3.....	24

1. Datos generales

- Título: AUDIO CON ARDUINO. ARDUINO (II)
- Proyecto: REPRODUCTOR DE MÚSICA MP3
- Organiza: DIGITAL CODESIGN – KREITEK – COLEGIO NURYANA
- Fecha: 06/09/19 – 09/09/19
- Horario: 17:00 – 19:30
- Horas lectivas: 5
- Plazas: 20
- Lugar: MAKERSPACE KREITEK – COLEGIO NURYANA
- Ponente: MIQUEAS FORTES GONZÁLEZ
DANIEL ÁLVAREZ EUSEBIO
- Categoría: ELECTRÓNICA Y PROGRAMACIÓN

2. Introducción

2.1. Objetivos

El presente curso tiene por objeto el fomento de las vocaciones científicas, la creatividad y la actitud emprendedora entre alumnos, profesores y emprendedores, promoviendo un nuevo modelo de trabajo y aprendizaje basado en proyectos, también denominado Educación STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas) o Educación 4.0, caracterizado por el empleo de las TIC (Tecnologías de la Información y la Comunicación) como recurso para la formación y adquisición de nuevas competencias tecnológicas.

2.2. Descripción del curso

Acción formativa de Arduino Unidad 2: Audio con Arduino, es el segundo de una serie de cursos de electrónica, robótica y programación con Arduino, una placa de desarrollo de hardware y software libre muy popular por su facilidad de uso y gran versatilidad, gracias a la cual resulta muy sencillo desarrollar infinidad de proyectos, tanto de propósito educativo como profesional, convirtiendo en fácil, accesible y barato, lo que antes era caro y complejo.

En esta segunda acción puntual introduciremos un nuevo concepto con el que trabajar, el sonido, fundamental en muchos proyectos tanto de robótica como de electrónica en general. Aprenderemos a utilizar altavoces, módulos amplificadores de audio y módulos reproductores MP3 para ser capaces de añadir sonido a cualquier tipo de proyecto. También veremos cómo manejar y controlar con Arduino pantallas LCD, uno de los módulos más utilizados en proyectos cuando se requiere mostrar información, por ejemplo para disponer de un menú o para visualizar datos obtenidos por sensores.

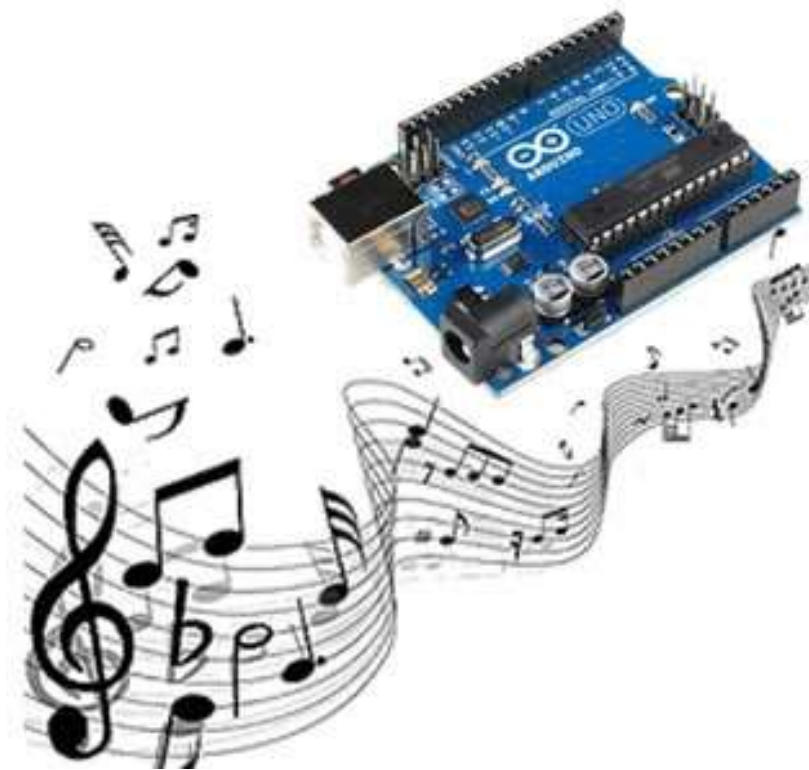
Realizaremos varios ejemplos para trabajar todos estos nuevos conceptos por separado, así como un reto final que nos permita aplicar todo lo adquirido en un único proyecto, el cual consistirá en la fabricación de un REPRODUCTOR DE MÚSICA MP3 con Arduino que incluya sonido con amplificador y altavoces, lector de tarjetas micro SD, pantalla LCD que muestre la pista actual y el nivel de volumen, seis botones (play, stop, canción siguiente, canción anterior, subir volumen, bajar volumen).

Respecto a anteriores acciones formativas avanzaremos en lenguaje de programación, aprendiendo a manejar nuevas librerías e instrucciones. Serán necesarias algunas piezas adicionales para construir el reproductor, que serán fabricadas previamente mediante impresión 3D.

3. Sonido con Arduino

3.1. Aplicaciones

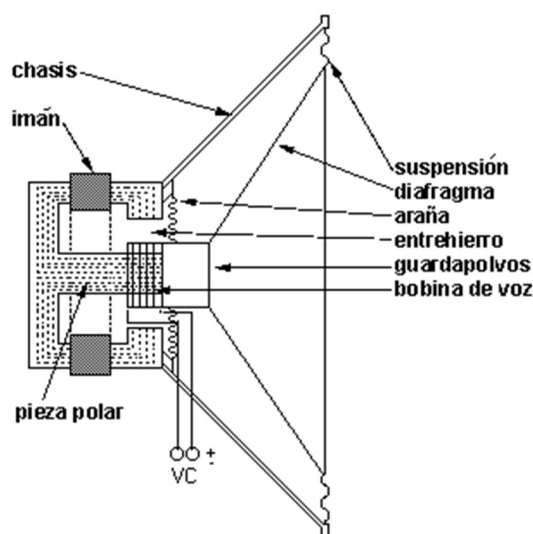
El sonido es uno de los recursos de mayor importancia y con el que tendremos que trabajar muy a menudo, en los diversos tipos de proyectos de electrónica con Arduino y placas similares que podamos realizar.



Conocer lo básico del sonido no solo resulta útil para trabajar con proyectos musicales o fabricar instrumentos, sino que las posibles aplicaciones van mucho más allá, como por ejemplo las alarmas, sonidos vinculados a la pulsación de botones o navegación de menús, grabaciones automatizadas para guiar la experiencia de usuario, efectos de sonido, etc...

3.2. El altavoz

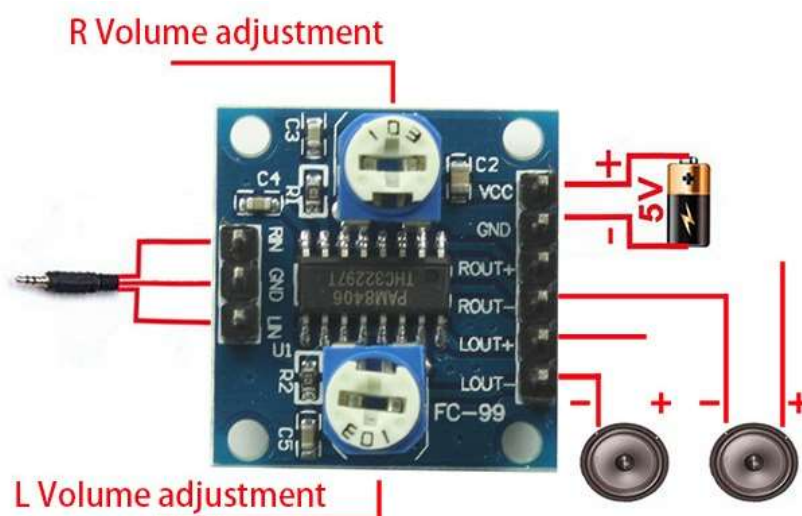
Un altavoz consta de un imán fijo que genera un campo magnético constante, y de un electroimán que genera un campo magnético variable dependiente de la señal eléctrica aplicada. La acción de estos dos campos magnéticos produce fuerzas de atracción y repulsión capaces de mover una estructura móvil denominada diafragma. El movimiento o vibración del diafragma es lo que produce las ondas de sonido que se transmiten por el aire.



3.3. El amplificador de audio PAM8406

Un amplificador de audio no es más que un tipo de amplificador electrónico, es decir, un circuito electrónico capaz de entregar a su salida una señal de mayor amplitud que la señal de entrada, pero de igual forma y frecuencia.

El módulo amplificador de audio que utilizaremos para construir el reproductor emplea el circuito integrado PAM8406:



Se trata de un amplificador de audio estéreo (estereofónico), lo que significa que dispone de dos canales independientes para reproducir el audio simultáneamente, un canal izquierdo y un canal derecho. El audio estéreo es aquel que se graba a partir de una pareja de dos micrófonos iguales (par estéreo), con lo que se obtiene un audio mucho más realista respecto al audio “mono”, ya que los humanos lo escuchamos todo por dos oídos.

PINES MÓDULO AMPLIFICADOR PAM8406	
PIN	Descripción
RIN	Canal derecho entrada
LIN	Canal izquierdo entrada
GND	Tierra común para los canales derecho e izquierdo
VCC	Polo positivo fuente de alimentación
GND	Polo negativo fuente de alimentación (Igual GND anterior)
ROUT+	Positivo canal derecho salida para altavoz 1
ROUT-	Negativo canal derecho salida para altavoz 1

LOUT+	Positivo canal izquierdo salida para altavoz 2
LOUT-	Negativo canal izquierdo salida para altavoz 2

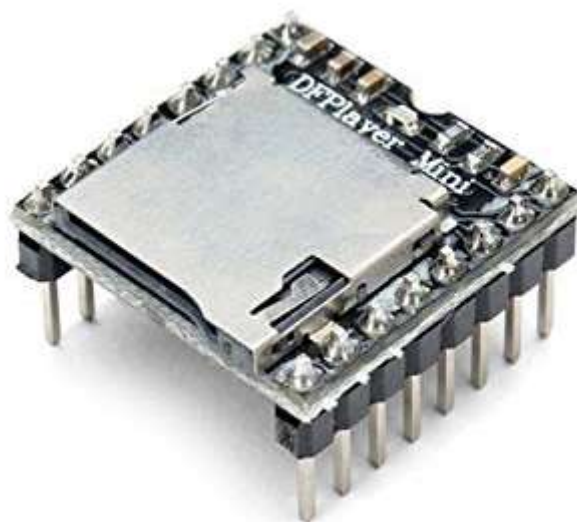
El funcionamiento es sencillo:

- La señal de audio de entrada a amplificar (estéreo) lleva tres cables que se deben conectar a los pines correspondientes del amplificador: canal derecho, canal izquierdo y común (GND).
- El módulo debe ser alimentado a través de los pines de alimentación VCC y GND.
- A la salida podemos conectar hasta dos altavoces para escuchar el audio en modo estéreo. Hay dos pines para cada altavoz.

Asimismo indicar que el módulo amplificador de audio PAM8406 consta de dos potenciómetros para regular la intensidad de la señal de forma independiente para cada canal, pudiendo controlar así el volumen.

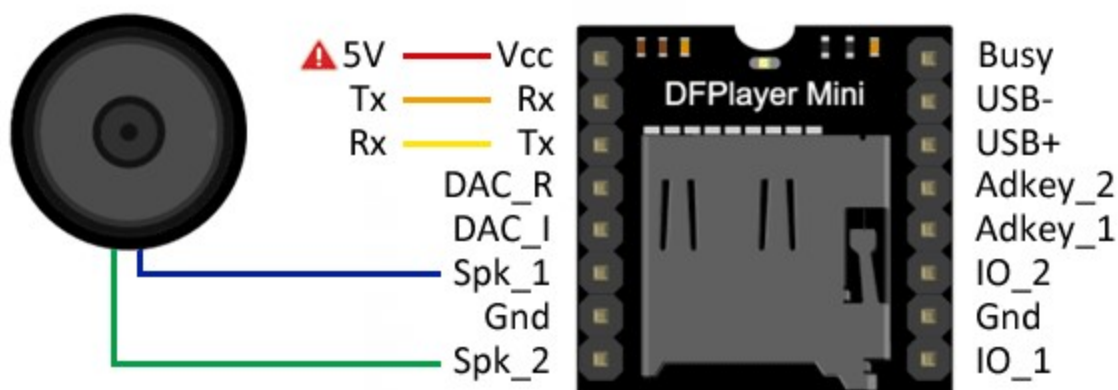
3.4. Mini reproductor MP3 DFPlayer

El módulo DFPlayer Mini es un reproductor de audio completo capaz de decodificar y reproducir formatos de fichero MP3, WMA y WAV.



Dispone de un lector integrado micro SD y de dos pines de entrada para comunicación serie por puerto USB (USB+ y USB-) a los que podemos conectar un conector USB hembra para poder leer las pistas de audio desde un pendrive por ejemplo. Tiene

capacidad para trabajar con hasta 100 carpetas y 255 canciones. Dispone también de 30 niveles de volumen ajustable.



Vcc externo 5V y 1A. Al usar alimentación externa SIEMPRE poner con GND común.

A continuación se indica la descripción de los pines del DFPlayer Mini:

VCC	Tensión de entrada (3.3 – 5 V)
RX	UART puerto receptor (desde donde se recibe)
TX	UART puerto transmisor (desde donde se envía)
DAC_R	Salida de audio canal derecho
DAC_I	Salida de audio canal izquierdo
SPK_1	Terminal 1 para altavoz
GND	Masa o común de la alimentación
SPK_2	Terminal 2 para altavoz
IO_1	P. Corta (canción previa) – P. Larga (bajar volumen)
IO_2	P. Corta (canción siguiente) – P. Larga (subir volumen)
ADKEY_1	Reproducir primer segmento
ADKEY_2	Reproducir último segmento
USB + / -	Para conectar un puerto USB
BUSY	El módulo indica si está reproduciendo (ocupado) o no

El módulo DFPlayer Mini es un reproductor de audio que puede funcionar de manera totalmente autónoma, ya que dispone de su propio microcontrolador y entradas específicas preprogramadas para una función concreta (selección de pista y volumen).

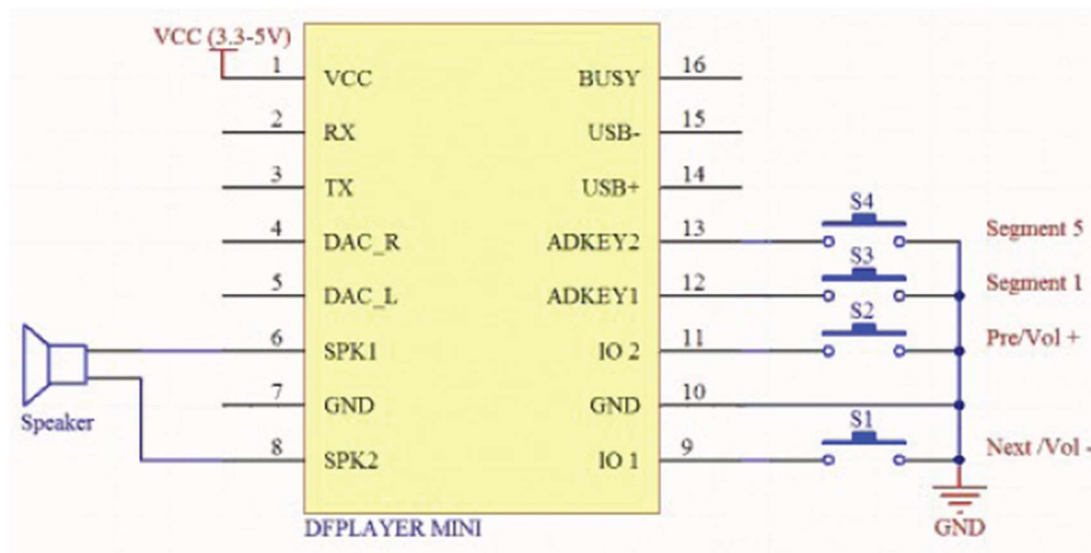
También consta de un pequeño amplificador de audio integrado, por lo que le podemos conectar un altavoz directamente (SPK_1 y SPK_2) y escuchar música, aunque evidentemente el volumen es bajo y conviene más llevar la señal de audio (DAC_R y DAC_L) a un amplificador externo para obtener mayor potencia, que es para lo que queremos el amplificador PAM8406.

Las posibilidades son aún mayor si en lugar de utilizar el DFPlayer Mini como reproductor autónomo, independizamos el control a través de un microcontrolador externo Arduino que envíe comandos a través del puerto serie (RX y TX).

Más adelante veremos la librería del reproductor DFPlayer Mini para Arduino IDE y mBlock y aprenderemos las instrucciones específicas para controlar el reproductor desde Arduino.

3.5. Circuito 1: Reproductor MP3 DFPlayer Mini modo autónomo

El primer circuito práctico que vamos a realizar será para hacer funcionar el reproductor DFPlayer Mini en modo autónomo. El siguiente esquema muestra la configuración más sencilla para lograr esto:



Nosotros haremos lo mismo pero en lugar de conectar un altavoz directamente a la salida de audio amplificada, aprovecharemos la salida de audio estéreo (DAC_R y DAC_L)

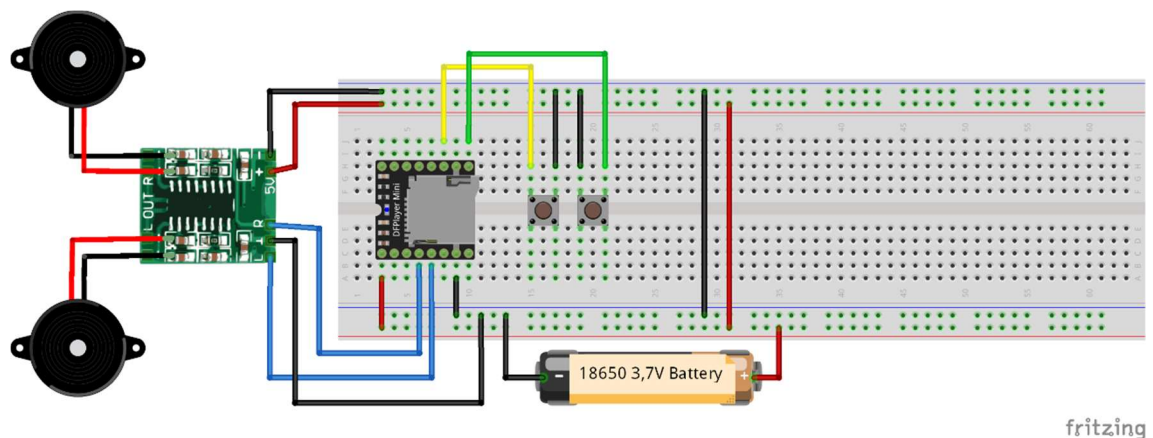
sin amplificar para conectar el amplificador de audio externo PAM8406 con dos altavoces, para lograr una mayor potencia de audio y calidad de sonido.

Los pulsadores S3 y S4 no serán necesarios para esta prueba, ya que los pines ADKEY1 y ADKEY2 son para cambiar la lista de reproducción, y en nuestro caso trabajaremos con una única lista de reproducción, es decir, que en la tarjeta micro SD tendremos una sola carpeta para archivos MP3.

El pulsador S1 conectado al pin IO1 nos servirá para iniciar la reproducción y seleccionar la siguiente pista de audio (pulsación corta), así como para bajar el volumen (pulsación larga).

El pulsador S2 conectado al pin IO2 nos servirá para seleccionar la pista de audio anterior (pulsación corta), así como para subir el volumen (pulsación larga).

La siguiente imagen muestra cómo quedaría el circuito en este primer ejercicio:

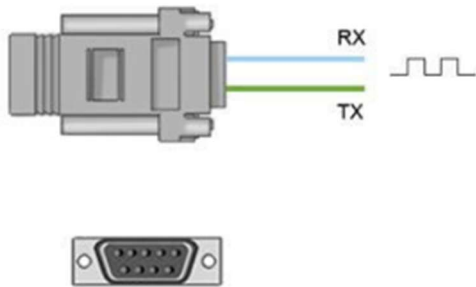


3.6. Comunicación por puerto serie

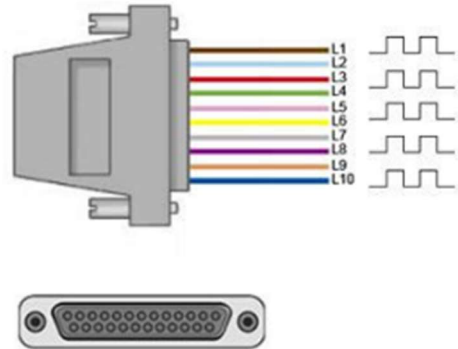
La comunicación por puerto serie es la forma más común de comunicación entre dispositivos, y la principal manera en que el Arduino se comunica con el ordenador y otros componentes.

Un puerto serie envía y recibe información mediante una secuencia de bits de uno a uno a través de un canal de comunicación. A diferencia de este, un puerto paralelo envía la información a través de múltiples canales.

COMUNICACIÓN SERIE



COMUNICACIÓN PARALELO



Existen muchos tipos de puertos serie. El más conocido sin duda es el puerto USB (Universal Serial Bus), aunque también podemos encontrar el antiguo puerto serie RS-232, y otros muchos protocolos de comunicación serie como el RS-485, I2C, SPI, etc...

La comunicación serie es la base de las comunicaciones en los microcontroladores, y todos disponen de al menos una unidad UART (Universally Asynchronous Receiver/Transmitter), un tipo de puerto serie que consta de dos canales, uno de transmisión (TX) y otro de recepción (RX).

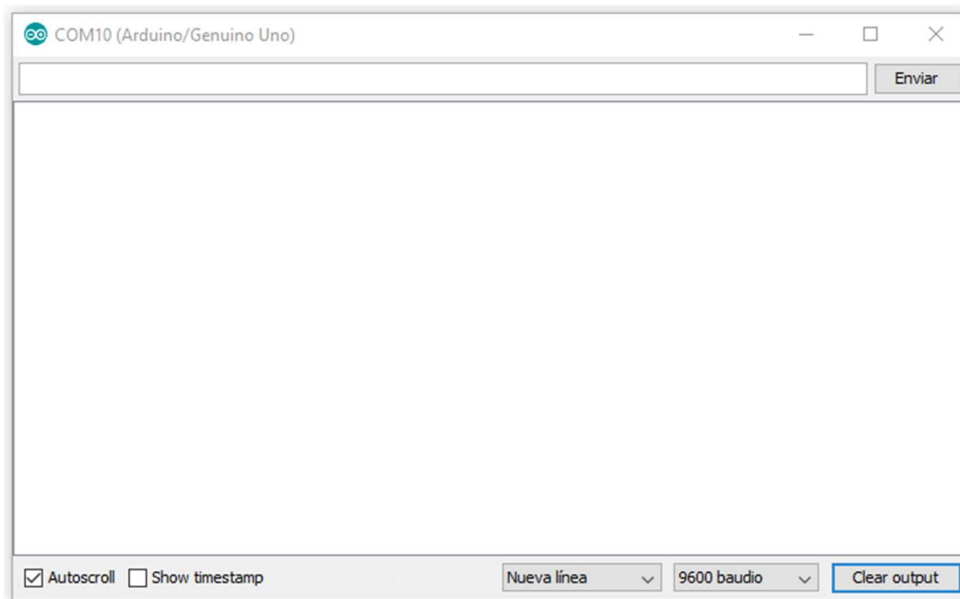
Las placas Arduino disponen de al menos una unidad UART. En el caso de Arduino UNO, los puertos serie están físicamente conectados a los pines digitales 0 (RX) y 1 (TX), que no podremos utilizar como entradas y salidas digitales corrientes mientras el puerto serie esté en uso.

Podemos comunicar la placa Arduino con el ordenador a través del bus serie utilizando el puerto USB, el mismo puerto que utilizamos para programarlo, estableciendo así un canal de comunicación bidireccional que nos permita enviar y recibir datos.

La interfaz que nos permite visualizar los datos enviados desde el Arduino, así como enviar datos desde el PC al Arduino, es el **Monitor Serial** que viene integrado en Arduino IDE, y que podremos encontrar en la esquina superior derecha del software con el siguiente símbolo:



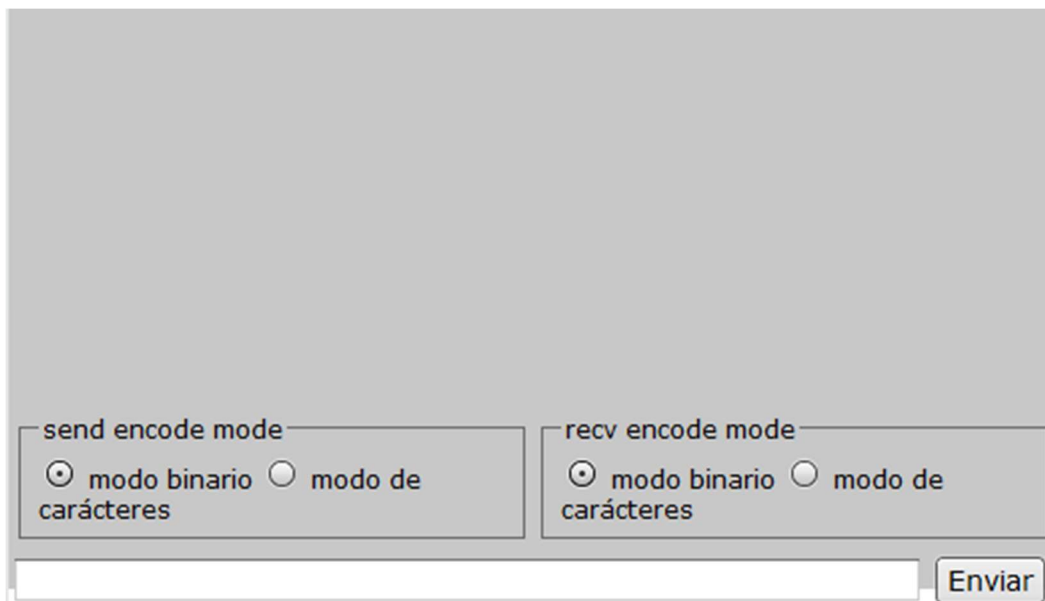
Una vez abierto, el Monitor Serial de Arduino IDE tiene el siguiente aspecto:



Como puedes ver se trata de una interfaz muy básica y sencilla de utilizar, con los siguientes elementos a destacar:

- **Envío de datos:** Desde el espacio en blanco superior podemos introducir y enviar datos desde el PC al Arduino.
- **Recepción de datos:** Desde el espacio en blanco que ocupa la gran parte de la ventana se muestran los datos recibidos.
- **Autoscroll:** Para efectuar o no el scroll automático de la ventana de recepción de datos.
- **Show timestamp:** Para mostrar la hora exacta de envío de cada dato.
- **Ajuste de línea:** Para añadir un tipo de ajuste de línea al final del mensaje que se desea enviar.
- **Baud Rate o Tasa de baudios:** Velocidad de transmisión medida en tasa de baudios, lo que significa número de baudios por segundo. Un baudio es un símbolo que puede contener varios bits de información. Es muy importante que la tasa de baudios del receptor coincida con la del emisor.

En el caso de mBlock también tenemos un monitor serie que siempre se muestra abierto por defecto. Se encuentra en la esquina inferior derecha de la interfaz de programa:



A través de esta pantalla podemos enviar datos desde mBlock al Arduino y visualizar los datos enviados desde el Arduino al PC. Asimismo podemos seleccionar el tipo de dato, modo binario o modo de caracteres, tanto para la recepción como para el envío.

3.7. Sentencias para comunicación serie – IDE Arduino

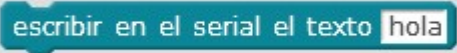
A continuación se indican los nuevos tipos de sentencias que necesitaremos para trabajar la comunicación por puerto serie en Arduino IDE a nivel básico:

Sentencia	Descripción
Serial.begin(9600)	Línea de configuración de la comunicación serie indicando el baudrate o tasa de baudios. Como ejemplo se ha puesto la velocidad típica de 9600 baudios/seg.
Serial.print()	Se envía por puerto serie el contenido del “print” sin salto de línea.
Serial.println()	Se envía por puerto serie el contenido del “print” con salto de línea.
#include "SoftwareSerial.h"	También podemos utilizar esta librería para cuando necesitamos agregar por software un puerto serie adicional haciendo uso de dos pines digitales cualesquiera.

SoftwareSerial miPuertoSerie(2, 4);	Después de declarar la librería, lo primero es crear un objeto de tipo SoftwareSerial con cualquier nombre, indicando los pines digitales a utilizar para el puerto serie. En el ejemplo mostrado el pin 2 es para RX y el pin 4 para TX.
miPuertoSerie.begin(9600)	Para configurar la tasa de baudios cuando usamos la librería SoftwareSerial.h. En el ejemplo se indica la velocidad típica de 9600.

3.8. Sentencias para comunicación serie – mBlock

A continuación se indican los nuevos tipos de sentencias que necesitaremos para trabajar la comunicación por puerto serie en mBlock a nivel básico:

Sentencia	Descripción
	Se envía por puerto serie el texto contenido en el espacio en blanco con salto de línea.

3.9. Circuito 2: Enviando un mensaje por puerto serie

Para poner en práctica lo aprendido anteriormente sobre comunicación por puerto serie, vamos a realizar un programa capaz de ejecutar un contador de números enteros que comience desde cero e incremente la cuenta de uno en uno con periodo de 1 segundo. El programa será capaz de enviar los datos desde el Arduino hasta el ordenador para que sea posible visualizar la cuenta a través del monitor serial.

Realizaremos el programa desde Arduino IDE y también desde mBlock. Para este ejercicio no será necesario montar ningún circuito electrónico, bastará con conectar el Arduino UNO al ordenador.

3.10. Librería para el reproductor DFPlayer Mini

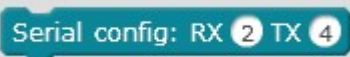

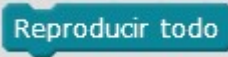

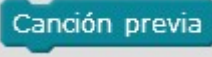
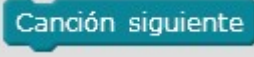
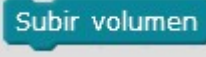
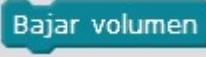
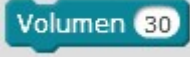
Anteriormente vimos cómo hacer funcionar el reproductor DFPlayer Mini de forma autónoma sin necesidad de depender de un microcontrolador externo para comandar el reproductor.

A partir de aquí veremos cómo utilizar Arduino para controlar el reproductor de música, para lo cual necesitaremos de una nueva librería de funciones específicas para el DFPlayer, tanto para Arduino IDE como para mBlock.

En Arduino IDE, la librería que vamos a utilizar se llama "DFRobotDFPlayerMini.h", y las nuevas funciones que vamos a aprender son las siguientes:

Sentencia	Descripción
DFRobotDFPlayerMini miMP3	Declaramos un objeto de tipo DFRobotDFPlayerMini con el nombre que queramos darle (ej: miMP3).
miMP3.begin(miPuertoSerie)	Para pasar la configuración del puerto serie al objeto reproductor creado.
miMP3.play(1)	Haciendo referencia al objeto creado con el nombre asignado previamente, la función "play()" permite iniciar la reproducción de la pista indicada entre paréntesis.
miMP3.enableLoopAll()	Reproduce en orden ascendente y en bucle permanente todas las pistas de audio de la lista de reproducción.
miMP3.readCurrentFileNumber(1)	Lee el número de la canción en reproducción.
miMP3.volume(30)	Establece el nivel de volumen al valor indicado (0 – 30).
miMP3.readVolume()	Lee el nivel de volumen actual.
miMP3.pause()	Detiene la reproducción.
miMP3.previous();	Reproduce la pista de audio anterior.
miMP3.next();	Reproduce la pista de audio siguiente.
miMP3.volumeUp();	Incrementa el nivel de volumen en una unidad.
miMP3.volumeDown();	Reduce el nivel de volumen en una unidad.

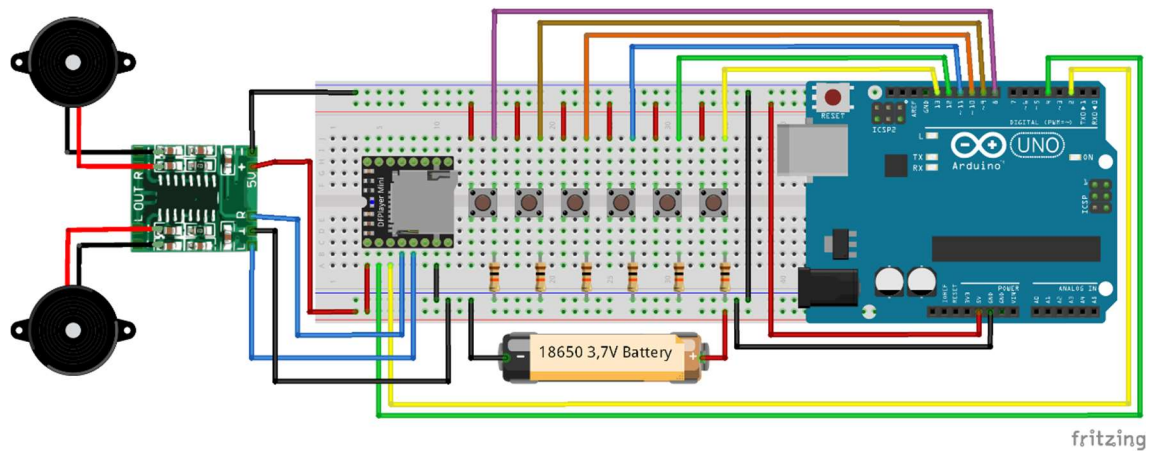
En mBlock, la Extensión que vamos a utilizar se llama "DFPlayerMini", y las nuevas funciones que vamos a aprender son las siguientes:

Sentencia	Descripción
	Se configura el puerto serie indicando los pines digitales destinados para RX y TX.
	Reproduce la pista de audio indicada de la lista de reproducción.
	Reproduce en orden ascendente y en bucle permanente todas las pistas de audio de la lista de reproducción.
	Detiene la reproducción.
	Reproduce la pista de audio anterior.
	Reproduce la pista de audio siguiente.
	Incrementa el nivel de volumen en una unidad.
	Reduce el nivel de volumen en una unidad.
	Establece el nivel de volumen al valor indicado (0 – 30).

3.11. Circuito 3: Reproductor con Arduino

Ahora que ya conocemos los componentes necesarios para fabricar un reproductor de música, así como las nuevas librerías específicas del reproductor DFPlayer Mini para programar un control externo del módulo MP3, podemos utilizar Arduino para controlar todas las funciones del reproductor, lo que amplía mucho las posibilidades.

En este nuevo caso práctico comenzaremos por realizar el circuito que se muestra a continuación:



A continuación realizaremos un programa, tanto con Arduino IDE como con mBlock, que nos permita controlar desde Arduino el reproductor DFPlayer Mini, añadiendo seis pulsadores con los que implementaremos las siguientes funciones:

- Play: para iniciar la reproducción
- Stop: para detener la reproducción
- Prev: para seleccionar la canción anterior
- Next: para seleccionar la canción siguiente
- Up: para subir volumen
- Down: para bajar volumen

4. Pantalla LCD

4.1. Pantalla LCD 16X2

Las pantallas LCD (Liquid Crystal Display) son de gran utilidad en multitud de proyectos de electrónica y Arduino, ya que nos permiten de manera muy sencilla mostrar datos, lecturas de sensores, menús, símbolos y cualquier otro tipo de mensaje que requiera nuestro proyecto.

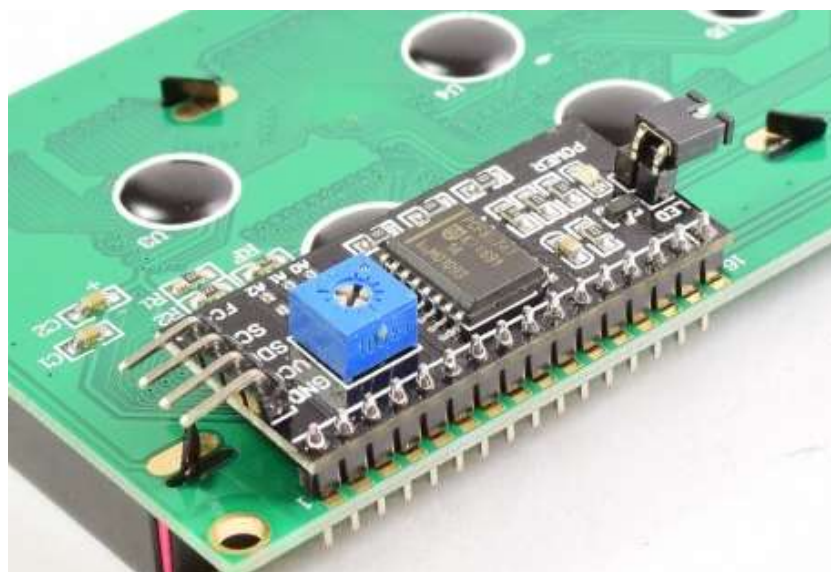
En nuestro caso vamos a utilizar una pantalla LCD de 16X2, lo que significa que dispone de 2 filas y 16 caracteres por fila:



Con esta pantalla podremos completar nuestro reproductor de música dándole un toque profesional, mostrando mensajes que indiquen la función en ejecución del reproductor, así como otros datos útiles como el número de la canción en reproducción y el nivel de volumen.

4.2. Módulo I2C para pantallas LCD

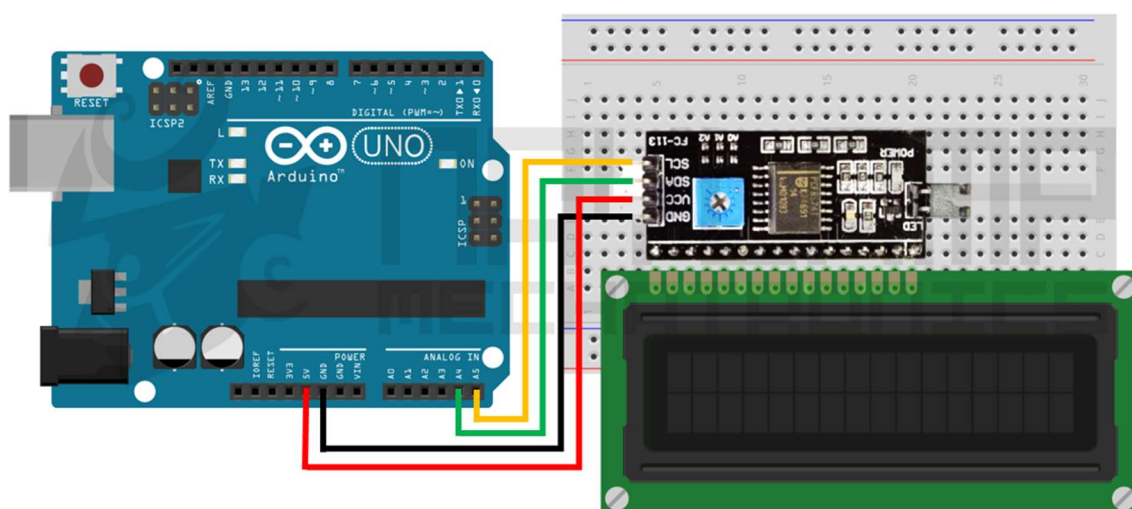
Se trata de un módulo adaptador de bus de datos I2C para pantallas LCD, que permite controlar pantallas LCD desde Arduino o cualquier otro microcontrolador haciendo uso de tan solo 2 pines, simplificando muchísimo el conexionado entre componentes.



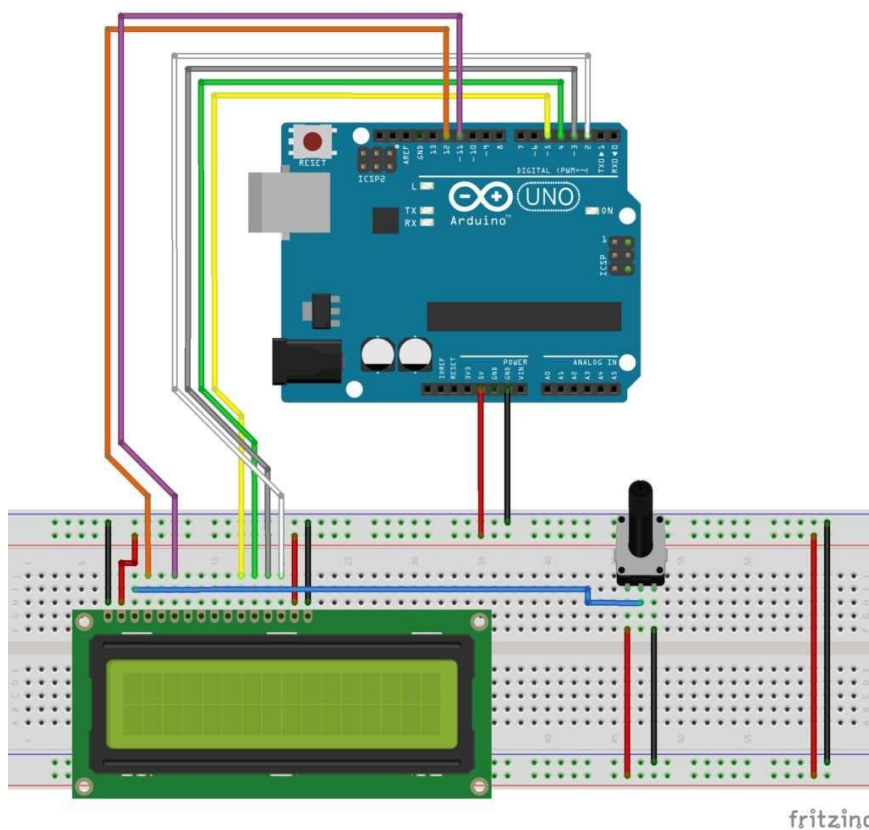
Los pines del módulo adaptador I2C están ordenados y distribuidos de tal manera que coinciden perfectamente con los pines de las pantallas LCD que vamos a utilizar.

El módulo dispone asimismo de un potenciómetro que sirve para regular el contraste de la pantalla LCD.

La siguiente imagen muestra cómo quedan las conexiones entre el Arduino UNO y la pantalla LCD 16X2 con módulo adaptador I2C:



Compárese el esquema anterior con el siguiente que nos muestra el circuito de conexión entre una pantalla LCD y el Arduino UNO sin usar módulo adaptador I2C:



No cabe duda de que la principal ventaja de utilizar una pantalla LCD con módulo adaptador I2C es simplificar el circuito y ahorrar pines del Arduino, de forma que puedan quedar más recursos disponibles para otros componentes de nuestros proyectos.

No obstante, queda una cuestión muy importante por tratar: ¿Qué es y cómo funciona el bus de datos I2C? Esto lo veremos en el siguiente apartado.

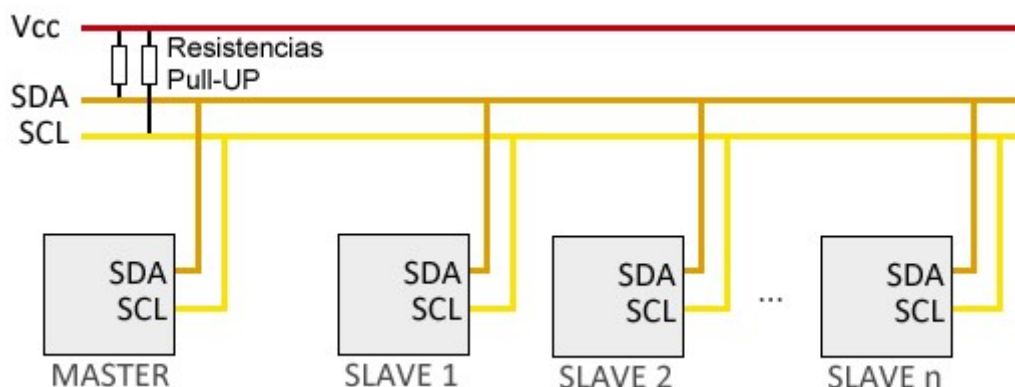
4.3. Bus de datos I2C

El Bus de datos I2C (Inter-Integrated Circuit) es un tipo de bus serie desarrollado por Philips en 1982 para comunicación entre microcontroladores y dispositivos electrónicos, que llegó a convertirse en un estándar de comunicación para muchos fabricantes del mercado.

Este bus de datos requiere solo de dos cables para su funcionamiento, uno para el envío y recepción de datos (SDA) y otro para la señal de reloj (SCL), lo que le confiere la característica de ser un bus de datos síncrono.

El bus I2C presenta una arquitectura de tipo maestro-esclavo, lo que significa que existe un maestro (también se puede configurar para que haya más de uno), que en nuestro caso será el Arduino UNO, así como uno o varios esclavos conectados al bus. El maestro

es el que envía información a través del canal SDA, y quién la solicita a los esclavos. Los esclavos no pueden comunicarse ni con el maestro ni con otros esclavos por sí solos.



Los dispositivos conectados al bus de datos como esclavos deben tener una dirección única. Es la forma que tiene el maestro de distinguir a los esclavos conectados.

Como puede verse en el esquema anterior, el bus I2C necesita de dos resistencias de Pull-Up entre alimentación y los dos hilos SDA y SCL, pero en caso de usar Arduino no suele ser necesario añadir estas resistencias gracias a la librería “Wire.h” que activa las resistencias de Pull-Up internas del propio Arduino, además de disponer del resto de funciones necesarias para la comunicación por bus I2C con Arduino.

En el caso de Arduino UNO y Arduino NANO, los pines preestablecidos para trabajar con el bus de datos I2C son las entradas analógicas A4 para SDA y A5 para SCL.

Los módulos I2C para las pantallas LCD que vamos a utilizar, disponen por defecto de la dirección 0x3F para el bus I2C en la mayoría de los casos, aunque también los podemos encontrar con la dirección 0x27. Si se diera el caso de que en un mismo proyecto necesitamos usar dos módulos adaptadores I2C para dos pantallas LCD diferentes, lo más probable es que la dirección por defecto sea la misma, lo cual no puede suceder para una correcta configuración del bus I2C. En dicho supuesto la solución sería modificar la dirección por defecto del módulo o módulos adicionales. La forma de hacerlo es soldando los puentes A0, A1, A2 presentes en el módulo, con lo cual podemos disponer de hasta 7 secuencias binarias diferentes para cambiar la dirección del dispositivo en el bus I2C.

4.4. Librerías para pantallas LCD y comunicación I2C

Para trabajar con pantallas LCD con protocolo de comunicación I2C necesitaremos las siguientes dos librerías:

- **LiquidCrystal_I2C.h**: Librería para pantallas LCD adaptada a comunicación I2C.
- **Wire.h**: Librería que permite la comunicación I2C entre Arduino y otros dispositivos.

La primera librería deberemos instalarla manualmente. Podrás descargarla fácilmente de Internet, aunque también la podrás encontrar en la carpeta de archivos del curso. Por otro lado, la segunda librería no necesitas buscarla en ningún lado, dado que se encuentra preinstalada en el software IDE de Arduino, y bastará con que la declares directamente en el editor de texto.

A continuación se indican las nuevas sentencias específicas de la librería LiquidCrystal_I2C.h que aprenderemos a manejar:


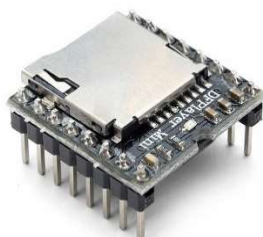

Sentencia	Descripción
#include <LiquidCrystal_I2C.h>	Declaración de la librería LiquidCrystal_I2C.h
#include <Wire.h>	Declaración de la librería Wire.h
LiquidCrystal_I2C lcd(0x3F, 16, 2);	Instanciamos un objeto de tipo LiquidCrystal_I2C al que llamamos "lcd" (puede tener cualquier nombre), cuyos atributos quedan definidos por medio de tres variables, que en nuestro caso son: "0x3F" es la dirección para el bus I2C (también podría ser 0x27), "16" es el número de caracteres por línea de la pantalla y "2" es el número de líneas de la pantalla.
lcd.init()	Inicializa el modulo adaptador LCD a I2C, esta función internamente configura e inicializa el I2C y el LCD.
lcd.backlight()	Enciende la luz de fondo del LCD.
lcd.noBacklight()	Apaga la luz de fondo del LCD.
lcd.clear()	Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda (posición (0,0)).
lcd.setCursor(col, row)	Sitúa el cursor de la pantalla en la posición indicada por la columna (col) y la fila (row).
lcd.print()	Muestra por pantalla el mensaje indicado entre paréntesis. El mensaje se imprime a partir de la posición del cursor.

lcd.createChar (num, datos)	Sirve para crear un carácter personalizado de 5x8 píxeles, donde “num” es el número del carácter (de 0 a 7) y “datos” es una matriz que contiene los píxeles del carácter.
lcd.scrollDisplayLeft()	Se desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la izquierda
lcd.scrollDisplayRight()	Se desplaza el contenido de la pantalla (texto y el cursor) un espacio a la derecha.



5. Proyecto: Reproductor de música MP3

5.1. Lista de materiales

En la siguiente lista se muestra el total de componentes electrónicos necesarios para fabricar el reproductor de música MP3 objeto del curso:

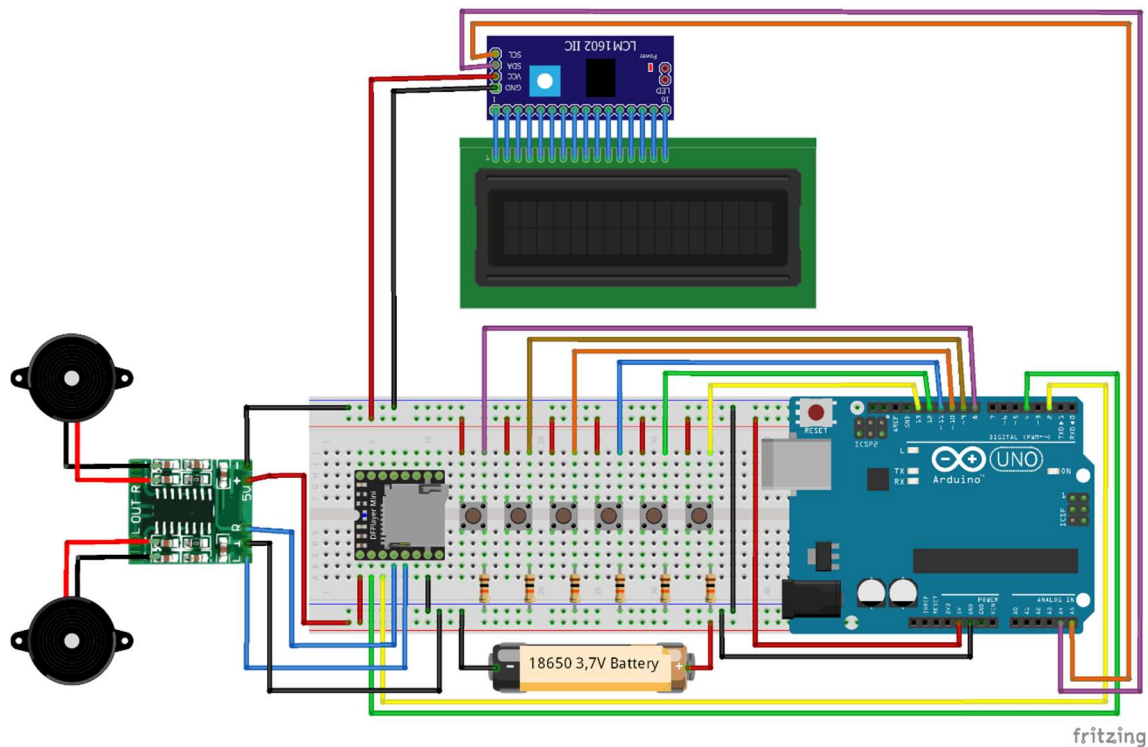
Imagen	Ud.	Descripción
	1	Arduino UNO
	1	Reproductor DFPlayer Mini
	1	Tarjeta micro SD

	1	Amplificador de audio PAM8406
	2	Altavoz 3W 4 Ohm
	1	Pantalla LCD 16x2 + Módulo adaptador I2C
	6	Resistencia 10 Kohm
	6	Pulsador Omron B3F con capuchón de colores
	1	Set cables Dupont macho-macho y macho-hembra

	1	Portapilas para batería 18650 3,7V
	1	Batería 18650 3,7V

5.2. Esquema electrónico

La siguiente imagen muestra el esquema electrónico del reproductor de música completo, que integra todos los componentes que hemos trabajado a lo largo del curso, que son el módulo reproductor con etapa de amplificación y altavoces, microcontrolador externo con Arduino UNO para control mediante programación del módulo reproductor, conjunto de pulsadores para permitir al usuario interactuar con el reproductor teniendo el control manual de todas las funciones básicas (play, stop, prev, next, vol_up, vol_down) y finalmente una pantalla LCD para visualizar los diferentes estados del reproductor (número de pista en reproducción, nivel actual de volumen y mensajes adicionales):



El siguiente objetivo por tanto será montar el circuito anterior, dejando preparada toda la electrónica antes del paso final que será la programación.

5.3. Programación del reproductor de música MP3

Llegado a este punto ya deberíamos tener todo el circuito montado, incluso la programación media hecha ya que para realizar el programa final partiremos del último código realizado, ya que realmente el reproductor es el mismo, solo que ahora hemos agregado una pantalla LCD, de modo que para finalizar el proyecto solo tendremos que centrarnos en añadir la programación necesaria para que a través de la pantalla podamos visualizar como mínimo la siguiente información:

- Un mensaje de bienvenida al encender el reproductor
- El número de la pista de audio que se está reproduciendo
- El nivel actual de volumen

Elige el software que más te guste para hacer el programa, Arduino IDE o mBlock, o incluso los dos!!! Si ves que te ha sobrado tiempo.