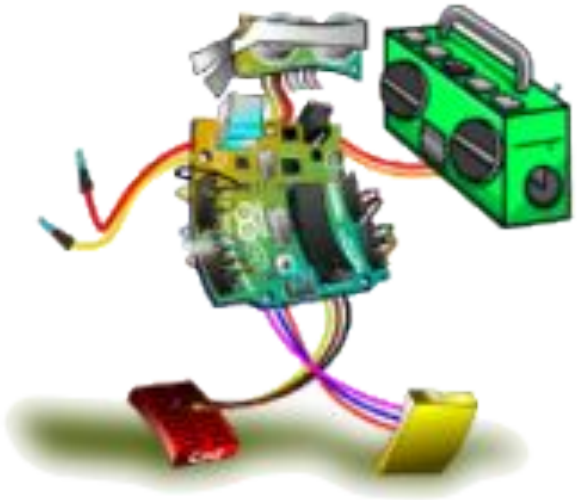
Proyecto A5: audio con DFPlayerMini

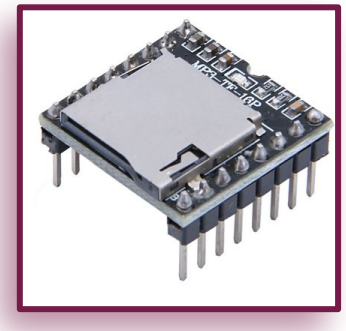
Roby se nos va de marcha con Arduino y su "loro" (DFPlayerMini).

Tampoco hace falta muchas explicaciones más...



por Aurelio Gallardo

El módulo DFPlayerMini



Su precio en aliexpress es de 1.45€ + gastos de envío , dependiendo de la tienda.

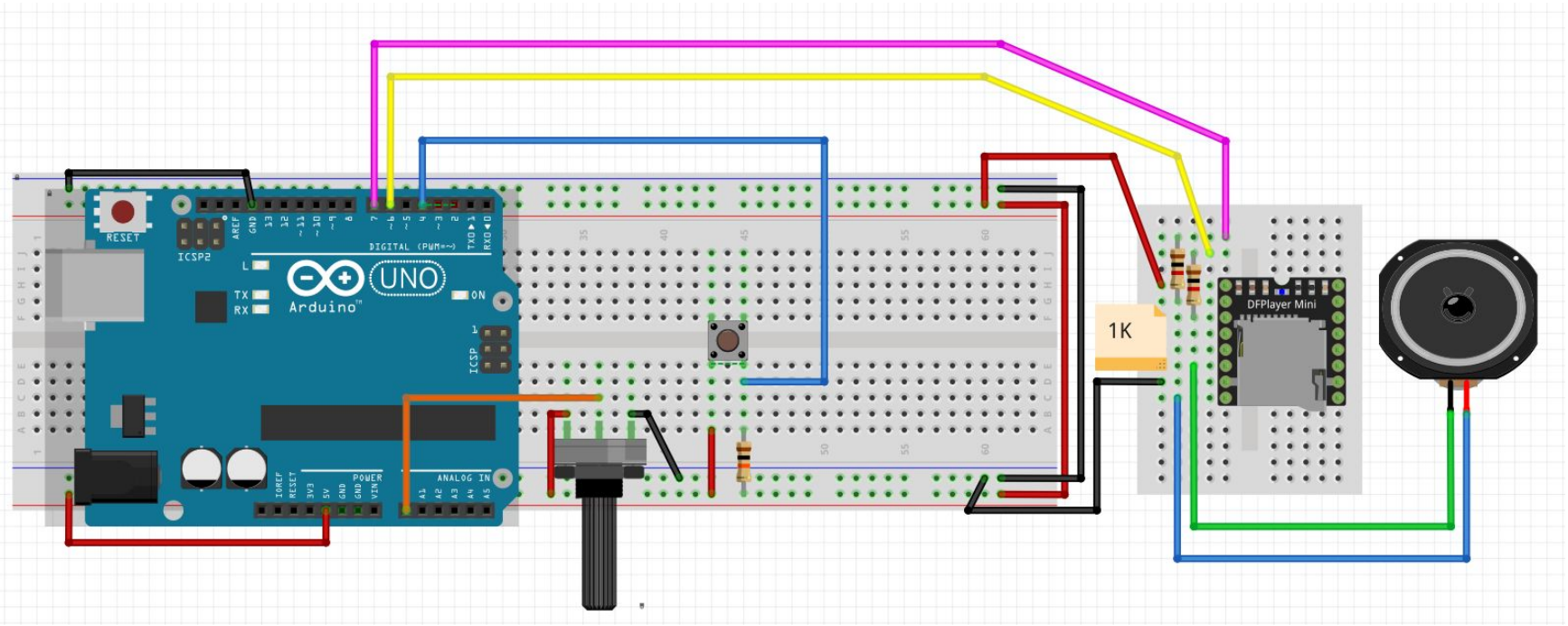
Este pequeño módulo es una pequeña maravilla. Básicamente la idea es reproducir contenido de audio (mp3, principalmente) desde una tarjeta microSD y controlado desde una placa de arduino.

Vamos a prepararnos. Por ejemplo, baja cuatro ficheros mp3 de la página http://soundbible.com/

Formatea una tarjeta microSD con formato fat32 e introduce las 4 pistas (en una carpeta llamada mp3). Renómbralas como 0001.mp3, 0002.mp3 , 0003.mp3 y 0004.mp3

Necesitarás un pen usb adaptador a tarjetas microSD, para poder grabar desde el ordenador. Además los siguientes materiales: arduino Uno, ordenador con IDE de Arduino, placa protoboard, cable USB de conexión, cables jumper dupont 2.54 macho-macho y macho hembra, DFPlayermini, pulsador, potenciómetro (10K~100K), tarjeta microSD formateada a fat32, resistencias de 1K y 10K y speaker.

El módulo DFPlayerMini (conexiones)



Monta el circuito de la figura. El potenciómetro nos servirá para controlar el volumen (A0) y el botón irá alternando las pistas (Pin 4). *El altavoz (speaker) que he usado es de 4 Ohm (*)* . Las resistencias al RX y TX del DFPlayermini de 1K cada una (pines 6 y 7).

(*) NOTA sobre el altavoz (speaker): según la documentación, se puede usar un altavoz de 4 a 8 Ohm pero siempre menor de 3W de potencia. Ver manual: http://image.dfrobot.com/image/data/DFR0299/DFPlayer%20Mini%20Manul.pdf

El módulo DFPlayerMini. Página del proyecto y librería.

La biblioteca que usaremos es la desarrollada por DFRobot (Wiki del proyecto: https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299). La encontraremos en DFRobotDFPlayerMini o podemos descargarla en formato ZIP directamente desde GITHUB.

Instalamos la biblioteca en Arduino, siguiendo la ruta: PROGRAMA >> INCLUIR LIBRERíA >> AÑADIR BIBLIOTECA ZIP. Esta biblioteca no está disponible desde los repositorios de Arduino y hay que añadirla de forma manual.

De acuerdo con la wiki del proyecto, la reproducción de las pistas se realiza desde una carpeta llamada mp3, localizada en la raíz de la tarjeta. Además las pistas deben ser renombradas y, al menos, empezar con una numeración formateada a 4 dígitos: la primera sería 0001.mp3, la segunda 0002.mp3 , etc… Aunque es factible renombrarlas como 0001hola.mp3 , etc..

El módulo DFPlayerMini (programa básico)

```
// Bibliotecas necesarias
#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

// La biblioteca SoftwareSerial se usa cuando se necesita instanciar una segunda
// comunicación serie. En este caso lo haremos con nuestro DFPlayerMini
// La comunicación serie "normal" la mantendremos con el PC
// y así veremos información en la pantalla.
SoftwareSerial miSegundaSerial(6, 7); // RX, TX
// Instancio objeto de la clase DFRobotDFPlayerMini
DFRobotDFPlayerMini miMP3;

// *********
// Variables
// *********
int volumen = 10; // valor de volumen, de 0 a 30
unsigned long tiempo=0; // tiempo en ms
int pista=0; // nº de pista a reproducir
int estado = LOW; // estados pulsación de botón
int estadoAnterior = LOW;
```

El programa es una adaptación del que podéis encontrar como ejemplo (GetStarted) al instalar la librería. Pero ¡ojo! No es exactamente igual.

Se usará una función denominada waitMilliseconds que no hará nada durante una cantidad prefijada. Durante ese tiempo DFPlayerMini reproducirá el archivo de sonido y, si activamos dicha función, no podrá actuar sobre el DFPlayerMini. Puede definirse un array con la duración de cada archivo, de manera que justo cuando acabe de reproducir lo volvamos a tener listo para su uso.

El módulo DFPlayerMini (programa básico)

```
// *********
// SETUP
// *********
void setup()
{
  miSegundaSerial.begin(9600); // a 9600
  Serial.begin(115200); // a 115200

  Serial.println();
  Serial.println(F("Demostración DFRobot DFPlayer Mini")); // Serial.println
  // con función macro F() https://heli.xbot.es/?p=519
  Serial.println(F("Inicializar DFPlayer Mini puede llevar varios segundos"));

  if (!miMP3.begin(miSegundaSerial)) {  //Use softwareSerial to communicate with mp3.
    Serial.println(F("Incapaz de arrancar"));
    Serial.println(F("1.Comprueba las conexiones"));
    Serial.println(F("2. Comprueba la tarjeta"));
    while(true){
      delay(0); // Código compatible con la vigilancia ESP8266
    }
  }
  Serial.println(F("DFPlayer Mini Activado"));

}
```

El módulo DFPlayerMini (programa básico)

```
// *********
// Principal
// *********
void loop()
{

  tiempo = millis(); // tiempo transcurrido desde el inicio del programa

  estado = digitalRead(4); // lee el pin 4

  // Si detecta una subida por flanco en el botón
  if (estado==HIGH && estadoAnterior==LOW) {
    pista = (pista+1)*(pista<4)+1*(pista>=4); // de la 1 a la 4
    Serial.println("Pista "+(String) pista);
    tocaMusica(pista,10000); // Función que toca la pista
    // y el programa "no hace nada" en el tiempo estipulado
  }

  if (miMP3.available()) {
    // Cadena que devuelve DFPlayerMini. Podemos
    // detectar estados y errores.
    printDetail(miMP3.readType(), miMP3.read());
  }

  // Cambio de volumen con el potenciómetro
  volumen = map(analogRead(A0),0,1023,1,30);
  miMP3.volume(volumen);
  Serial.println("Volumen "+ (String) volumen);

  estadoAnterior = estado; // refresco del estado

} // Fin del Loop
// *************************************************
```

El módulo DFPlayerMini (programa básico)

```
// *********
// Funciones
// *********
// toca una de las cuatro pistas almacenadas
void tocaMusica(int pista, int tiempo) {
  miMP3.play(pista);  // sd:/mp3/0001.mp3
  // miMP3.next(); // Otra posible orden para tocar la pista "siguiente"
  waitMilliseconds(tiempo); // Espera el tiempo estipulado
  // comentar esta función para observar otro modo de reproducción
}

// Función que espera un tiempo
void waitMilliseconds(uint16_t msWait)
{
  uint32_t start = millis();
  while ((millis() - start) < msWait)
  {
    delay(1); // El programa no hace nada
    // mientras dura el sonido activado
  }
}
```

Un compendio de todos los métodos y atributos de la clase DFRobotDFPlayerMini lo encontraréis en el programa de ejemplo FullFunction

Al iniciar el programa, el DFPlayerMini está desactivado. Cuando pulso el botón, pista=1 y se activa la función tocaMusica reproduciendo el primer archivo.

Si usamos la función waitMilliseconds, el programa no responderá hasta pasado el tiempo que se le pase. Esto permitiría reproducir el archivo sin interrupciones. ¿Qué ocurriría si no uso la función?

El módulo DFPlayerMini (programa básico)

```
// Función detallada en la página del proyecto
// Detecta estados y errores
void printDetail(uint8_t type, int value){
  switch (type) {
    case TimeOut:
      Serial.println(F("Time Out!"));
      break;
    case WrongStack:
      Serial.println(F("Stack Wrong!"));
      break;
    case DFPlayerCardInserted:
      Serial.println(F("Card Inserted!"));
      break;
    case DFPlayerCardRemoved:
      Serial.println(F("Card Removed!"));
      break;
    case DFPlayerCardOnline:
      Serial.println(F("Card Online!"));
      break;
    case DFPlayerUSBInserted:
      Serial.println("USB Inserted!");
      break;
    case DFPlayerUSBRemoved:
      Serial.println("USB Removed!");
      break;
    case DFPlayerPlayFinished:
      Serial.print(F("Number:"));
      Serial.print(value);
      Serial.println(F(" Play Finished!"));
      break;
    case DFPlayerError:
      Serial.print(F("DFPlayerError:"));
      switch (value) {
        case Busy:
          Serial.println(F("Card not found"));
          break;
        case Sleeping:
          Serial.println(F("Sleeping"));
          break;
        case SerialWrongStack:
          Serial.println(F("Get Wrong Stack"));
          break;
        case CheckSumNotMatch:
          Serial.println(F("Check Sum Not Match"));
          break;
        case FileIndexOut:
          Serial.println(F("File Index Out of Bound"));
          break;
        case FileMismatch:
          Serial.println(F("Cannot Find File"));
          break;
        case Advertise:
          Serial.println(F("In Advertise"));
          break;
        default:
          break;
      }
      break;
    default:
      break;
  }

}
```

Ejercicio A: reclamo.

Se pueden descargar diferentes audios de animales desde la web https://www.freeaudiolibrary.com/es/efectos-gratuitos-animales/ , simplemente registrándonos gratuitamente (si usas Firefox y tienes el complemento videodownloadhelper instalado, en el botón de reproducir de cada animal, pulsa con el botón derecho y descárgalo usando el complemento).

El proyecto consistiría en un generador de sonidos de animales (reclamo). Somos amantes de la fotografía de naturaleza y lo necesitamos para atraer a especímenes que queremos fotografiar.

- **El programa debe tener algún sistema de elección de la pista, para saber a qué animal estamos imitando.**
- **Un pulsado iniciará el archivo de sonido.**
- **Algún otro sistema podrá dejarlo en modo loop, para que se reproduzca continuamente.**