

Jukebox mit Arduino



In dieser Anleitung zeige ich dir, wie du eine Jukebox mit Arduino bauen kannst. Hier geht es um die Elektronik und die Tonausgabe. Die Musik werden wir von einer SD-Karte abspielen. Doch eine Jukebox besteht natürlich aus mehr: Licht, Oberflächen, liebevoll gestalteten Musikauswahlschildern, Holz, Chrom, Kunststoff – diesen Teil überlasse ich aber dir :-)

Was brauchen wir für die Jukebox? Natürlich erstmal ein Arduino. Dazu eine Reihe von Tastern für die Auswahl und ein Modul, dass MP3s abspielen kann. Dafür verwende ich gern den [DFPlayer Mini](#), denn dieser ist nicht nur sehr günstig zu haben, sondern auch sehr einfach zu verwenden. Dann benötigst du entweder aktive Lautsprecher oder einen Vorverstärker. Zum Testen reicht aber auch erstmal ein normaler kleiner Lautsprecher. Fassen wir die Bauteile noch einmal zusammen:

<https://starthardware.org/jukebox-mit-arduino/>

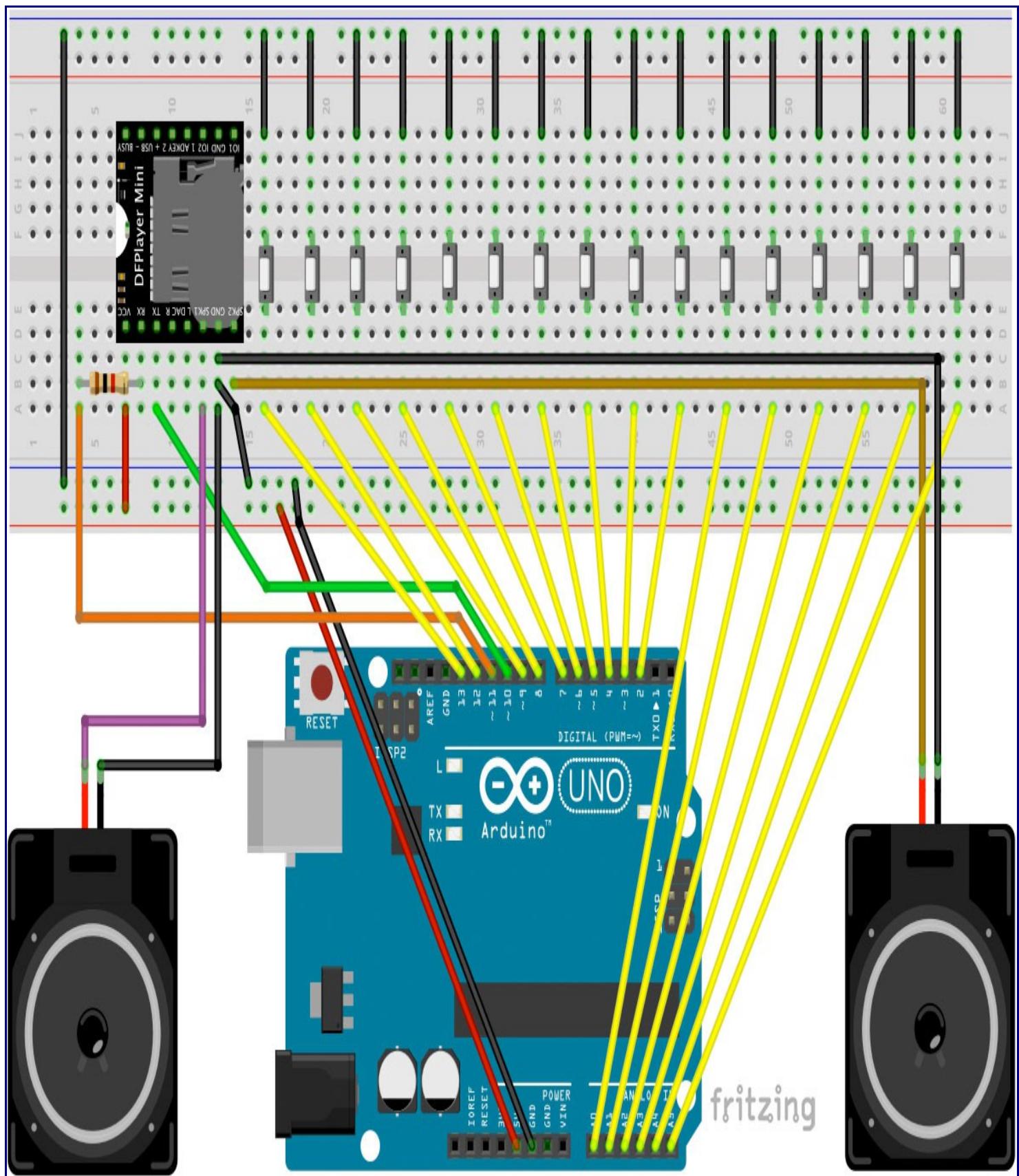
Bauteile

- [Arduino Set, Breadboard, Kabel, etc.*](#)
- 16x Taster
- 1x Widerstand 1kOhm
- [DFPlayer Mini*](#)
- [MicroSD-Karte*](#)
- [Lautsprecher*](#) – zum testen
- [Vorverstärker*](#) und Lautsprecher* oder
- [aktive Lautsprecher*](#) – je nach Geschmack und gewünschter Lautstärke

Du kannst auch gebrauchte Desktop-Speaker von Ebay kaufen oder im Freundeskreis rumfragen. Bei vielen Leuten liegen solche Sachen im Keller. Besser werden sie da nicht ;-)

Schaltung

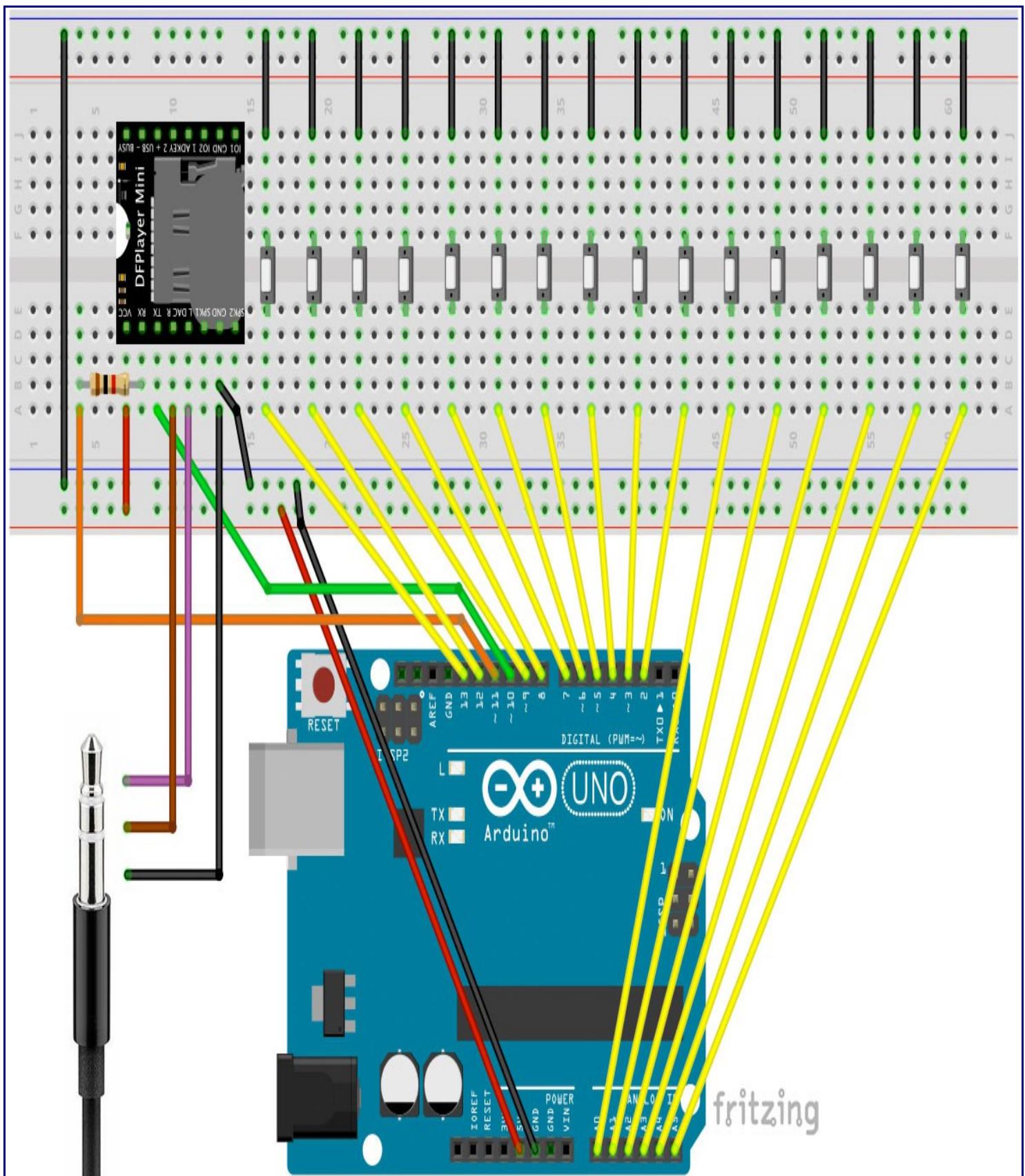
Die Schaltung unterscheidet sich kaum von der Schaltung für die [Sound Machine mit Arduino](#). Unserer Use-Case ist aber natürlich ein anderer und hier verwenden wir zwei Lautsprecher (Stereo).



Die Schaltung besteht aus dem DFPlayer Mini, zwei Lautsprechern und 16 Taster, wobei die Anzahl an Tastern der Musikauswahl entspricht. Du kannst auch weniger Taster verwenden oder mit Hilfe eines [Shift-Registers](#) noch mehr Taster anschließen.

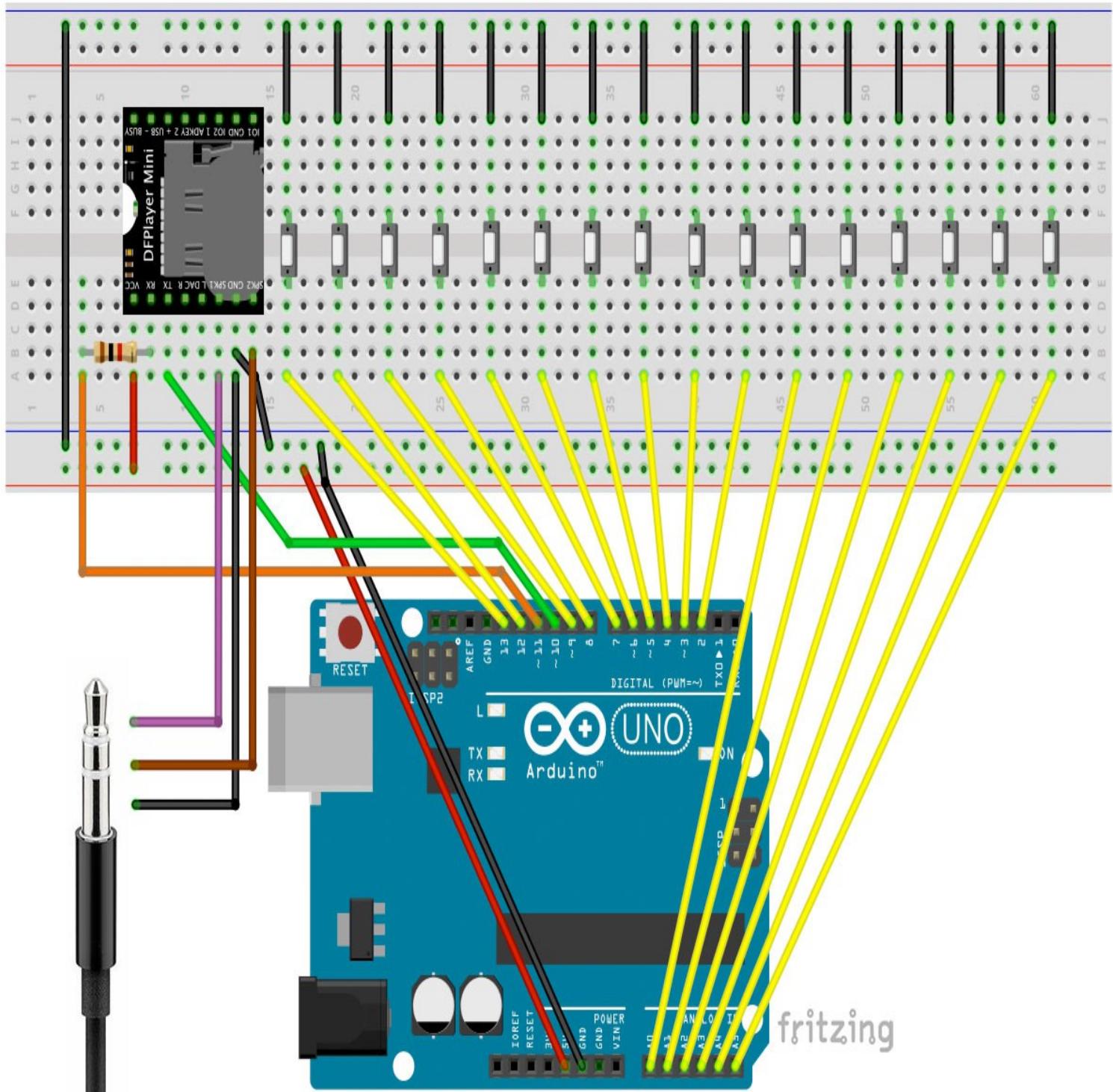
Der DFPlayer wird mit GND und 5V versorgt und an die SoftwareSerial-Pins 10 und 11 verbunden. Die Verbindung mit dem Pin 11 bekommt noch einen 1kOhm Widerstand. Die Lautsprecher werden direkt an den DFPlayer angeschlossen.

Wir verwenden für die Taster, die später die Musiktitel starten sollen, die internen Pullup-Widerstände des Arduino-Boards. Alle Taster sind mit dem GND und jeweils einem digitalen oder analogen Pin des Arduinos verbunden. (Die analogen Input-Pins lassen sich nämlich auch einfach als digitale Pins zum Auslesen von Tastern verwenden.)



Das Datenblatt des DFPlayers gibt an, dass die Pins DAC R und DAC L zur Verwendung von Kopfhörern oder Vorverstärkern genutzt werden sollen. Allerdings erzeugen diese Anschlüsse bei mir extremes Rauschen, das wahrscheinlich durch die Spannungswandlung des Arduinos kommt.

Besser funktioniert für mich die Verwendung der Speaker-Pins. Allerdings sollte die Lautstärke im Code sehr weit runtergeregt werden:



Wenn man das GND-Kabel so nah wie möglich am GND des DFPlayers anschließt, wird die Sound-Qualität erträglicher.

Ab hier wäre der weitere Weg also in die aktiven Lautsprecher oder den Vorverstärker. Das habe ich mal nicht extra aufgezeichnet.

SD-Karte

Kopiere jetzt deine Lieblings-Songs auf eine SD-Karte. Achte darauf, dass diese im FAT16 oder FAT32 Format formatiert ist. Die einzelnen MP3 Dateien müssen in einem Ordner mit dem Namen »MP3« im Hauptverzeichnis (root) der SD-Karte zu finden sein.

Mac-User sollten beim Formatieren darauf achten, dass sie auch Master Boot Record im Festplattendienstprogramm auswählen.

Code

Bind für die Verwendung des DFPlayers die **Programmbibliothek DFRobotDFPlayerMini** ein. Gehe dafür in der Arduino-Software ins Menü *Sketch>Programmbibliotheken einbinden>Programmbibliotheken verwalten ...* und gib im Suchfeld *DFPlayer* ein. Installiere die erwähnte Bibliothek *DFRobotDFPlayerMini*.

```
#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(10, 11); // RX, TX
DFRobotDFPlayerMini myDFPlayer;

int buttonPins[16] = {19,18,17,16,15,14,2,3,4,5,6,7,8,9,12,13}; // Pins, an denen die Taster angeschlossen sind

void setup() {
    mySoftwareSerial.begin(9600); // Start der SoftwareSerial Kommunikation
    Serial.begin(115200); // Start der "normalen" seriellen Kommunikation für den Serial Monitor

    if (!myDFPlayer.begin(mySoftwareSerial)) { // Verbindung der Software Serial Kommunikation mit dem DFPlayer
        Serial.println(F("Unable to begin:"));
        Serial.println(F("1. Please recheck the connection!"));
        Serial.println(F("2. Please insert the SD card!"));
        while(true){ // Programmabbruch -> while(true) ist Endlosschleife
            delay(0); // Code für den Watchdog des ESP8266, nicht relevant
    für Arduino UNO
        }
    }

    myDFPlayer.volume(10); // Lautstärke auf 10 ( Werte von 0 - 30 ist möglich)
    for (int i=0; i<16; i++){ // wiederhole 16 Mal (für jeden Taster)
        pinMode(buttonPins[i], INPUT_PULLUP); // Taster werden mit internen Pullup-Widerständen verwendet
    }

    void loop() {
        for (int i=0; i<16; i++){ // wiederhole 16 Mal (für jeden Taster)
            if (digitalRead(buttonPins[i]) == LOW){ // wenn ein Taster gedrückt wird
                Serial.println(i);
                myDFPlayer.playMp3Folder(i+1); // spielt das MP3 der Nummer des Tasters nach von der SD-Karte
                delay(200);
            }
        }
    }
}
```

```

delay(20);

/* Alles ab hier dient dem Debugging und kommt aus den Beispiel-Dateien der Library */
if (myDFPlayer.available()) {
    printDetail(myDFPlayer.readType(), myDFPlayer.read()); // Print the detail message from
DFPlayer to handle different errors and states.
}
}

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;
        }
    }
}

```

```
        break;
    default:
        break;
    }
    break;
default:
    break;
}
}
```