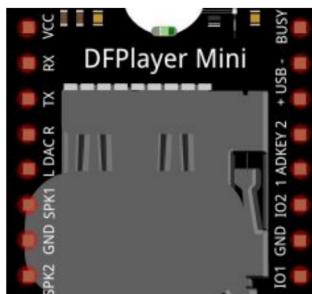


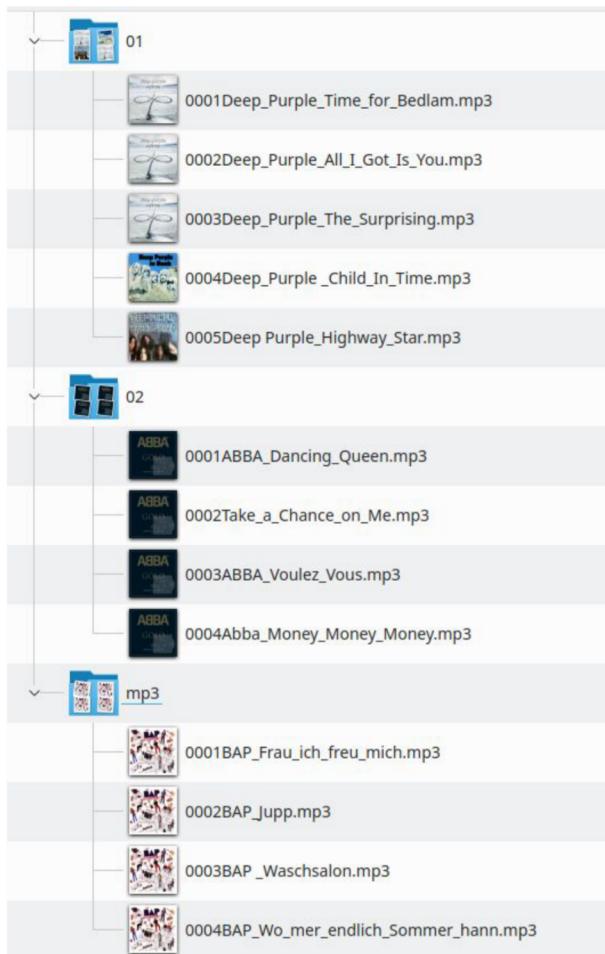
mp3-Player mit DFPlayer Mini



Der DFPlayerMini ist ein Modul, das mp3- und wav-Dateien abspielen kann. Der Lautsprecher wird über Steckverbindungen mit dem Modul verbunden.

Außerdem können noch zwei Taster angeschlossen werden, mit denen die Lautstärke verändert und ein Vor- und Rücksprung der Musikdateien ausgeführt werden kann.

Die Musikdateien werden auf eine maximal 32 GB große Mini-SD-Karte kopiert, die Formatierung muss FAT16 oder FAT32 sein.



Es können verschiedene Ordner angelegt werden. Zunächst können alle Dateien natürlich in das Stammverzeichnis kopiert werden. Die Namen müssen der 4+3-Konvention (4 Ziffern, Punkt und die Dateiendung mp3 oder wav) entsprechen und durchnummierter werden.

0001.mp3 0002.mp3 ... 3000.mp3

Nach den vier Ziffern dürfen auch Buchstaben verwendet werden, die eine Identifizierung des Titels ermöglichen (0001_Deep_Purple_Child_in_Time.mp3).

Außerdem können auch weitere Verzeichnisse angelegt werden, die von 01 bis maximal 99 bezeichnet werden können. In jedes dieser Verzeichnisse können wiederum höchstens 3000 Dateien abgelegt werden.

Weiterhin gibt es das Sonderverzeichnis mp3. Dateien in diesem Verzeichnis werden mit einem besonderen Befehl (playMp3Folder) abgespielt.

Überblick über die Methoden. Jeder Methode muss der Name des Players (mp3Player) vorangestellt werden:

DFRobotDFPlayerMini mp3Player;

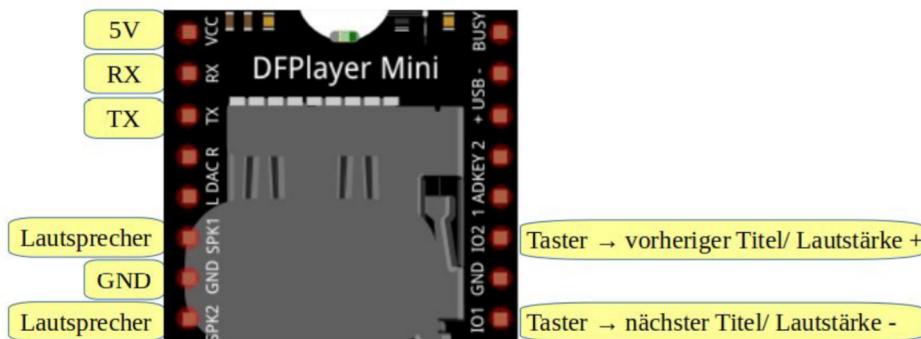
Befehl	Funktion	Beispiel
begin()	Player starten	SoftwareSerial DFPlayerSoftwareSerial(RX, TX); mp3Player.begin(DFPlayerSoftwareSerial);
volume(Wert)	Lautstärke mögliche Werte 1 -30	volume(20);
EQ()	Equalizer	EQ(DFPLAYER_EQ_NORMAL); EQ(DFPLAYER_EQ_POP); EQ(DFPLAYER_EQ_ROCK); EQ(DFPLAYER_EQ_JAZZ); EQ(DFPLAYER_EQ_CLASSIC); EQ(DFPLAYER_EQ_BASS);
play(Nummer)	spielt eine Datei aus dem Stammverzeichnis	play(1);
pause()	pausiert die Wiedergabe	
start()	startet die Wiedergabe nach einer Pause	
previous()	vorherigen Titel spielen	
next()	nächsten Titel spielen	
loopFolder(Ordner)	wiederholt alle Titel im Ordner	loopFolder(1) alle Titel im Ordner 01 werden wiederholt
playFolder(Ordner, Nummer)	spielt den angegebenen Titel aus dem Ordner	playFolder(2, 1); spielt den ersten Titel aus dem Ordner 02 der Ordner darf maximal 255 Dateien enthalten
playLargeFolder(Ordner, Nummer)	spielt den angegebenen Titel aus dem Ordner	playLargeFolder(2, 1); spielt den ersten Titel aus dem Ordner 02 der Ordner darf bis zu 3000 Dateien enthalten
playMp3Folder(Nummer)	spielt die Titel aus dem Ordner mp3	playMp3Folder(1); spielt den ersten Titel aus dem Ordner mp3 der Ordner darf bis zu 3000 Dateien enthalten
enableLoopAll()	alle Dateien wiederholen	
randomAll()	Dateien zufällig abspielen	
readFileCounts()	liest die Anzahl aller Dateien auf der SD-Karte	
readFileCountsInFolder(Ordner)	liest die Anzahl der Dateien im Ordner	int Dateien = mp3Player.readFileCountsInFolder(1); Serial.print(String(Dateien)); listet die Anzahl der Dateien im Ordner 01 auf

Das Programm soll alle Musikdateien in einem bestimmten Ordner in einer Schleife wiederholen.

Zwei Taster dienen der Navigation und der Änderung der Lautstärke. Ein kurzer Tastendruck auf den jeweiligen Taster springt vor oder zurück, ein langer Tastendruck verringert oder vergrößert die Lautstärke.

Die Pinbelegung des DFPlayers:

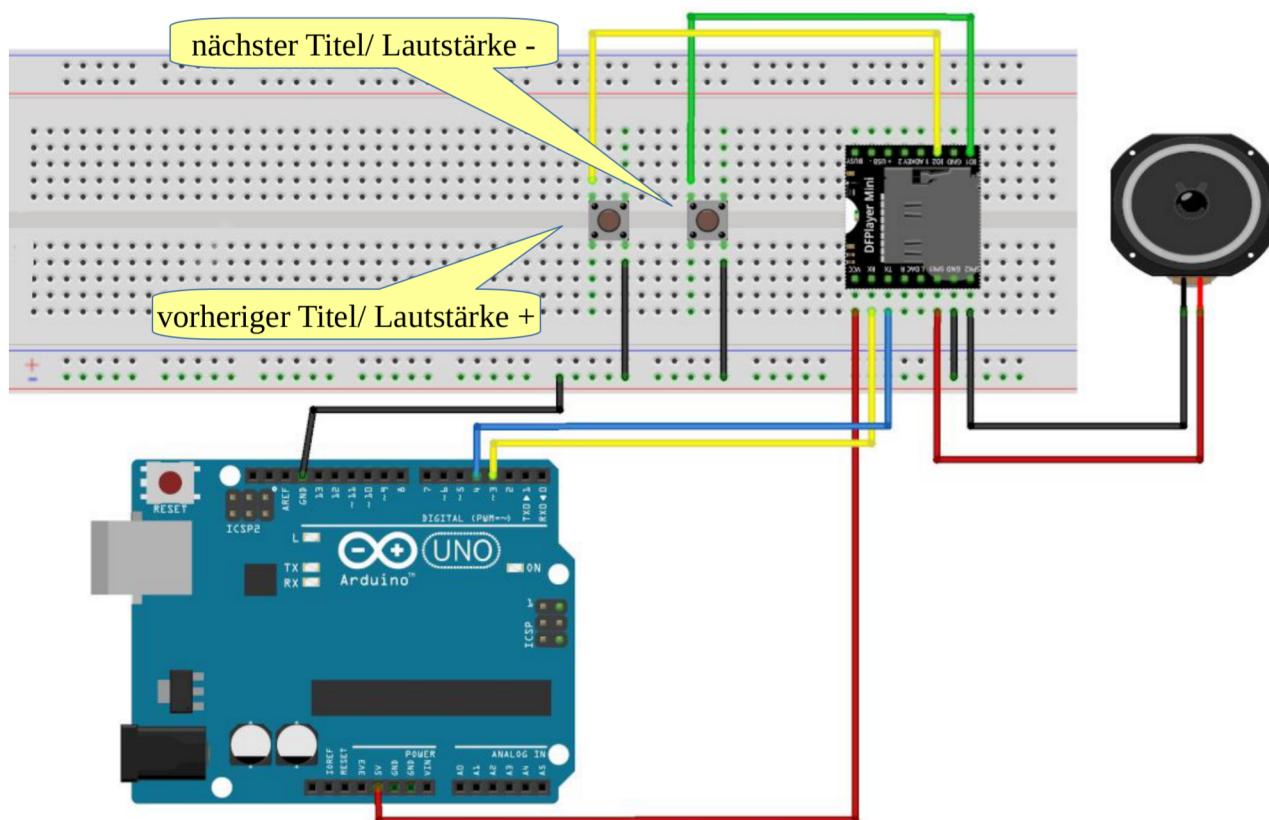
Die Pins RX und TX können an beliebigen Ports außer 0 und 1 angeschlossen werden. Sie werden mit der Bibliothek SoftwareSerial angesteuert.



Benötigte Bauteile:

- Lautsprecher
- 2 Taster
- DFPlayer Mini
- Leitungsdrähte

Baue die Schaltung auf.





Sollte das Programm nicht funktionieren, vertausche die Pins für RX und TX.

Bevor du starten kannst, musst du eine Bibliothek installieren:

Sketch → Bibliothek einbinden → Bibliotheken verwalten



Binde die benötigten Bibliotheken ein und definiere die Variablen.

```
# include <SoftwareSerial.h>
# include <DFRobotDFPlayerMini.h>

// RX/TX Port definieren
SoftwareSerial DFPlayerSoftwareSerial(4, 3);

// Name des Players
DFRobotDFPlayerMini mp3Player;

// Nummer des Ordners, der Ordner muss Daten enthalten
int Ordner = 1;
```

Das Programm soll nur einmal laufen und benötigt deshalb nur den setup-Teil.
Beachte die Kommentare.

```
void setup()
{
    DFPlayerSoftwareSerial.begin(9600);
    Serial.begin(9600);

    mp3Player.begin(DFPlayerSoftwareSerial);
    Serial.println(mp3Player.readFileCounts());

    /*
        Equalizer: mögliche Werte
        mp3Player.EQ(DFPLAYER_EQ_NORMAL);
        mp3Player.EQ(DFPLAYER_EQ_POP);
        mp3Player.EQ(DFPLAYER_EQ_ROCK);
        mp3Player.EQ(DFPLAYER_EQ_JAZZ);
        mp3Player.EQ(DFPLAYER_EQ_CLASSIC);
        mp3Player.EQ(DFPLAYER_EQ_BASS);
    */
    mp3Player.EQ(DFPLAYER_EQ_ROCK);
    // Lautstärke 1 .. 30
    mp3Player.volume(15);
```

```
// Anzahl der Dateien im Ordner feststellen und anzeigen
int AnzahlDateien = mp3Player.readfileCountsInFolder(Ordner);

// wenn sich im Ordner keine Daten befinden
if (AnzahlDateien == -1)
{
    Serial.println("Der Ordner enth\u00fclt keine Musikdateien!");
    Serial.println("Das Programm wird beendet!");
    while(1);
}

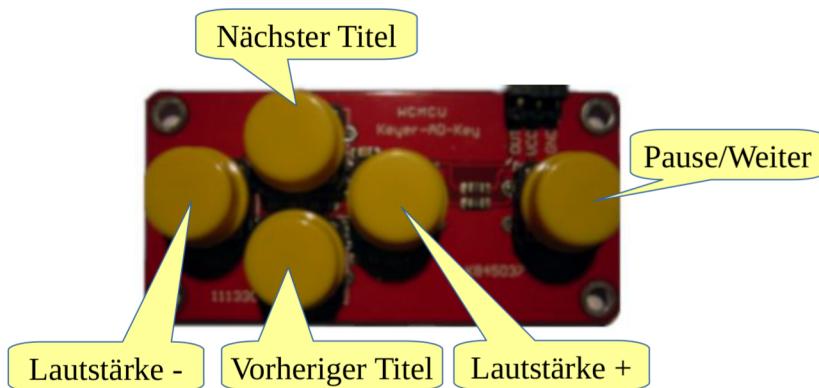
else
{
    Serial.print(String(AnzahlDateien) + " Dateien im Ordner " + String(Ordner));
    mp3Player.loopFolder(Ordner);
}
}

void loop()
{
    // bleibt leer, Programm l\u00e4uft nur einmal
}
```

mp3-Player mit Tastenpad

Der mp3-Player soll \u00fcber das Tastenpad gesteuert werden.

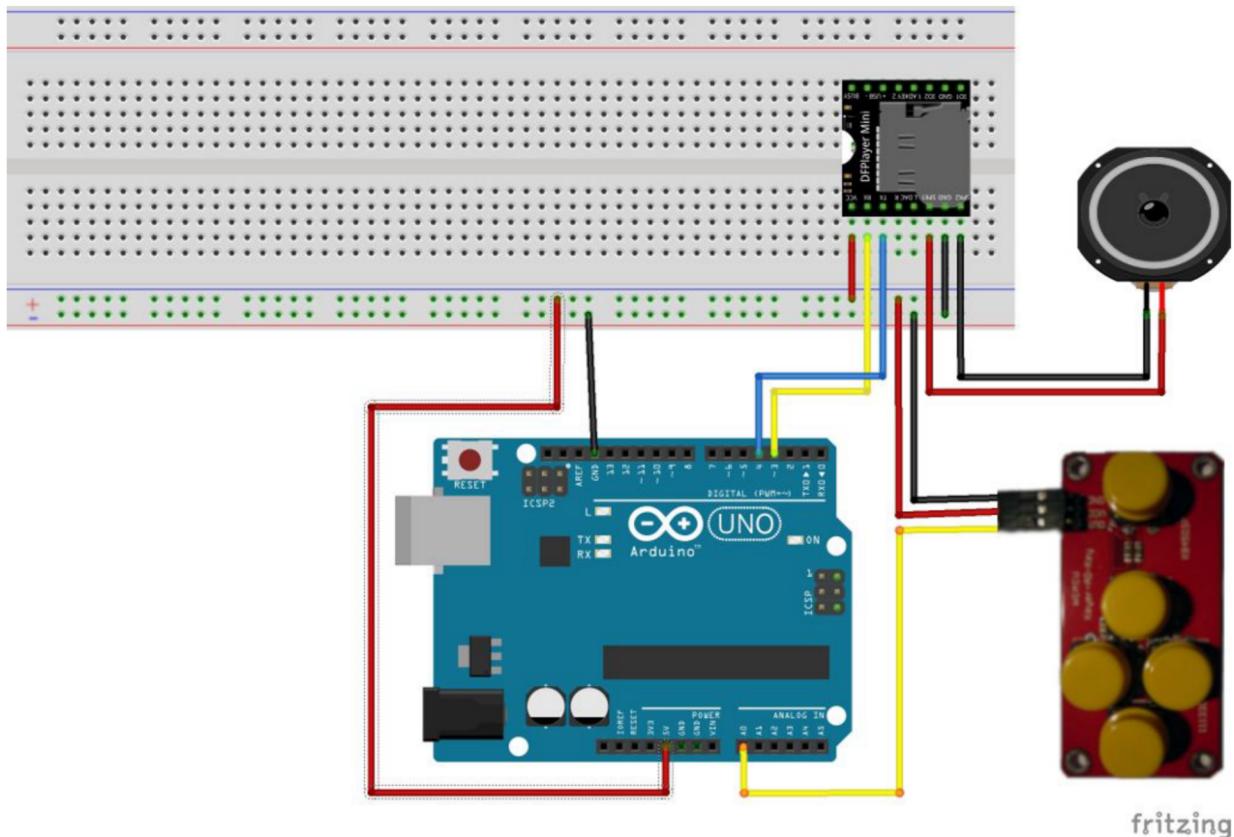
Die Tastenbelegung:



Ben\u00f6tigte Bauteile:

- Lautsprecher
- Tastenpad
- DFPlayer Mini
- Leitungsdr\u00e4hte

Baue die Schaltung auf.



Binde die benötigten Bibliotheken ein und definiere die Variablen.
Beachte die Kommentare

```
# include <SoftwareSerial.h>
# include <DFRobotDFPlayerMini.h>

// wird für die Abfrage von Wertebereichen mit switch benötigt
# include <stdio.h>

// RX/TX Port definieren
SoftwareSerial DFPlayerSoftwareSerial(4, 3);

// Name des Players
DFRobotDFPlayerMini mp3Player;
int Taster;
int Analogwert = 0;
int Lautstaerke = 10;

/*
Schalter für die Pausenfunktion
Pause == true -> Wiedergabe gestoppt
Pause == false -> Wiedergabe neu starten
*/
bool Pause = false;
```

Der setup-Teil:

```
void setup()
{
    DFPlayerSoftwareSerial.begin(9600);
    Serial.begin(9600);

    // Player starten
    mp3Player.begin(DFPlayerSoftwareSerial);

    /*
        Equalizer:
        mp3Player.EQ(DFPLAYER_EQ_NORMAL);
        mp3Player.EQ(DFPLAYER_EQ_POP);
        mp3Player.EQ(DFPLAYER_EQ_ROCK);
        mp3Player.EQ(DFPLAYER_EQ_JAZZ);
        mp3Player.EQ(DFPLAYER_EQ_CLASSIC);
        mp3Player.EQ(DFPLAYER_EQ_BASS);
    */
    mp3Player.EQ(DFPLAYER_EQ_ROCK);

    // Lautstärke 1 .. 30
    mp3Player.volume(Lautstaerke);

    // Anzahl der Dateien auf der SD-Karte feststellen und anzeigen
    int AnzahlDateien = mp3Player.readFileCounts();

    // wenn sich auf der SD-Karte keine Daten befinden
    if (AnzahlDateien == -1)
    {
        Serial.println("Der Ordner enth\u00fclt keine Musikdateien!");
        Serial.println("das Programm wird beendet!");
        while (1);
    }

    // die SD-Karte enth\u00e4lt Daten
    else
    {
        Serial.println(String(AnzahlDateien) + " Dateien auf der SD-Karte ");
        Serial.println("Lautst\u00e4rke: " + String(mp3Player.readVolume()));

        // Wiedergabe starten
        mp3Player.enableLoopAll();
    }
}
```



Verwende die Funktion Tasterabfrage.

```
int Tasterabfrage()
{
    Analogwert = analogRead(A0);
    delay(200);
    // Serial.println(Analogwert);

    /*
        für die Abfrage von Wertebereichen muss
        # include <stdio.h>
        im Kopf eingebunden werden
        A0 gibt je nach gedrückten Taster einen Wert aus
        über den Seriellen Monitor wird dieser Wert angezeigt
        und kann dann eventuell angepasst werden
        die Zeile Serial.println ... kann dann auskommentiert werden
        Taster 1 → links
        Taster 2 → oben Mitte
        Taster 3 → unten Mitte
        Taster 4 → rechts
        Taster 5 → außen
    */
    switch (Analogwert)
    {
        case 0 ... 20:
            Taster = 1;
            break;
        case 30 ... 60:
            Taster = 2;
            break;
        case 70 ... 120:
            Taster = 3;
            break;
        case 150 ... 200:
            Taster = 4;
            break;
        case 300 ... 400:
            Taster = 5;
            break;
        default:
            return 0;
    }

    // gedrückten Taster zurückgeben
    return Taster;
}
```

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
    Taster = Tasterabfrage();

    // Taster links -> Lautstärke -
    if (Taster == 1)
    {
        if (Lautstaerke > 0) Lautstaerke = Lautstaerke - 2;
        mp3Player.volume(Lautstaerke);
    }

    // Taster oben Mitte -> nächster Titel
    if (Taster == 2)
    {
        mp3Player.next();
    }

    // Taster unten Mitte -> vorheriger Titel
    if (Taster == 3)
    {
        mp3Player.previous();
    }

    // Taster rechts -> Lautstärke +
    if (Taster == 4)
    {
        if (Lautstaerke < 30) Lautstaerke = Lautstaerke + 2;
        mp3Player.volume(Lautstaerke);
        Serial.println("Lautst\u00e4rke: " + String(Lautstaerke));
    }

    // Taster außen -> Pause/Weiter
    if (Taster == 5)
    {
        if (!Pause)
        {
            Pause = !Pause;
            mp3Player.stop();
        }

        if (Pause)
        {
            Pause = !Pause;
            mp3Player.start();
        }
    }
}
```