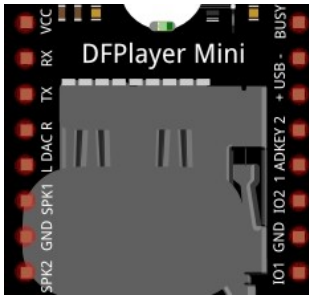
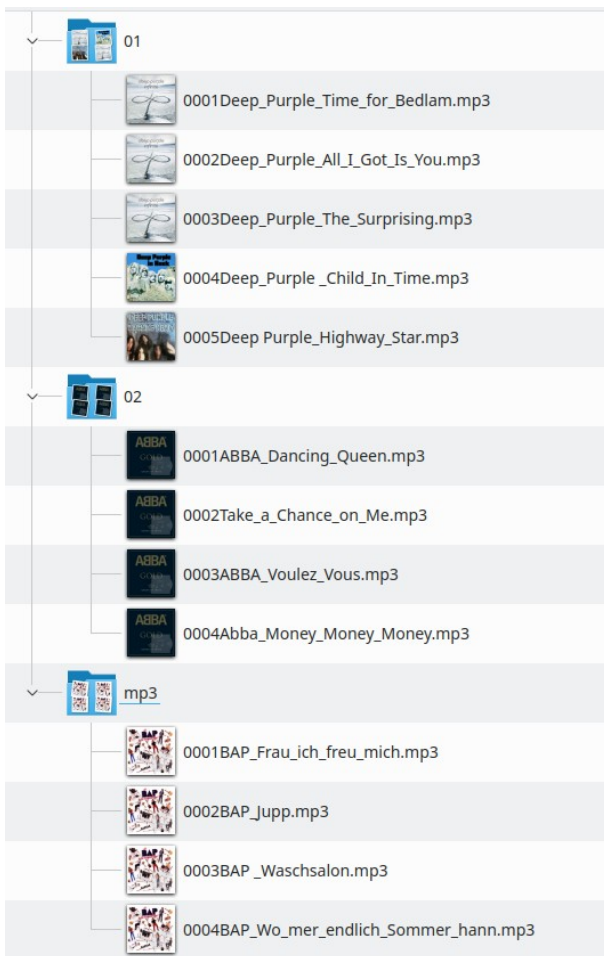


Der DFPlayerMini ist ein Modul, das mp3- und wav-Dateien abspielen kann. Der Lautsprecher wird über Steckverbindungen mit dem Modul verbunden.



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 Dateierdung mp3 oder wav) entsprechen und durchnummeriert 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 Verzeichnis 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;

Schlüsselwort	Aktion	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

mp3-Player im Seriellen Monitor

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

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

// Name des Players
DFRobotDFPlayerMini mp3Player;

int Lautstaerke = 15;
/*
  Schalter für die Pausenfunktion
  Pause == true -> Wiedergabe gestoppt
  Pause == false -> Wiedergabe neu starten
*/
bool Pause = false;

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

  delay(1000);

  // Player starten
  // falls die Parameter true, true nicht funktionieren
  // mp3Player.begin(DFPlayerSoftwareSerial, false, true);
  mp3Player.begin(DFPlayerSoftwareSerial, true, true);
  /*
  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);

  // zufällige Wiedergabe starten
  mp3Player.randomAll();

  // Menü
  Serial.println("-----");
  Serial.println("n = nächster Titel");
  Serial.println("v = vorheriger Titel");
  Serial.println("+ = Lautstärke +");
  Serial.println("- = Lautstärke -");
  Serial.println("s = Stopp/Weiter");
```

```
Serial.println("z  = zufällige Wiedergabe");
Serial.println("m  = Menü anzeigen");
Serial.println("1-9 = Verzeichnis 1-9 spielen");
Serial.println("-----");
}

void loop()
{
  // auf serielle Eingabe warten
  while (Serial.available() > 0)
  {
    // Eingabe im Seriellen Monitor lesen
    char Zeichen = Serial.read();

    // wenn die Eingabe eine Zahl ist
    if (isdigit(Zeichen))
    {
      // char in String umwandeln, anschließend zu int konvertieren
      String Zahl = String(Zeichen);
      int Nummer = Zahl.toInt();
      Serial.println("Spiele Verzeichnis " + String(Nummer));
      mp3Player.loopFolder(Nummer);
      Pause = false;
    }

    // - -> Lautstärke -
    if (Zeichen == '-')
    {
      if (Lautstaerke > 2) Lautstaerke = Lautstaerke - 2;
      mp3Player.volume(Lautstaerke);
      Serial.println("Lautst\u00e4rke: " + String(Lautstaerke));
    }

    // + -> Lautstärke +
    if (Zeichen == '+')
    {
      if (Lautstaerke <= 28) Lautstaerke = Lautstaerke + 2;
      mp3Player.volume(Lautstaerke);
      Serial.println("Lautst\u00e4rke: " + String(Lautstaerke));
    }

    // n -> nächster Titel
    if (Zeichen == 'n')
    {
      mp3Player.next();
      Serial.println("Nächster Titel");
    }

    // v -> vorheriger Titel
    if (Zeichen == 'v')
    {
      mp3Player.previous();
      Serial.println("Vorheriger Titel");
    }
  }
}
```

```
// s -> Pause/Weiter
if (Zeichen == 's')
{
  if (!Pause)
  {
    mp3Player.stop();
  }
  if (Pause)
  {
    mp3Player.start();
  }
  Pause = !Pause;
  if (Pause) Serial.println("Wiedergabe angehalten");
  if (!Pause) Serial.println("Wiedergabe fortgesetzt");
}

// z -> zufällige Wiedergabe Titel
if (Zeichen == 'z')
{
  mp3Player.randomAll();
  Serial.println("Zufällige Wiedergabe gestartet");
  Pause = false;
}

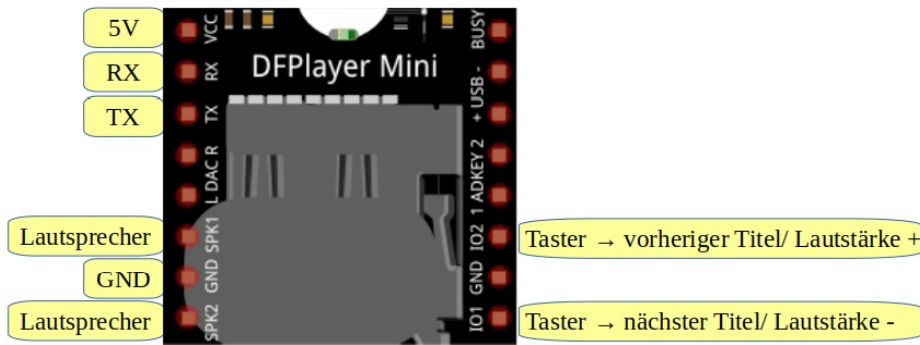
// m -> Menü anzeigen
if (Zeichen == 'm')
{
  Serial.println("-----");
  Serial.println("n  = nächster Titel");
  Serial.println("v  = vorheriger Titel");
  Serial.println("+  = Lautstärke +");
  Serial.println("-  = Lautstärke -");
  Serial.println("s  = Stopp/Weiter");
  Serial.println("z  = zufällige Wiedergabe");
  Serial.println("m  = Menü anzeigen");
  Serial.println("1-9 = Verzeichnis 1-9 spielen");
  Serial.println("-----");
}
}
}
```

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:

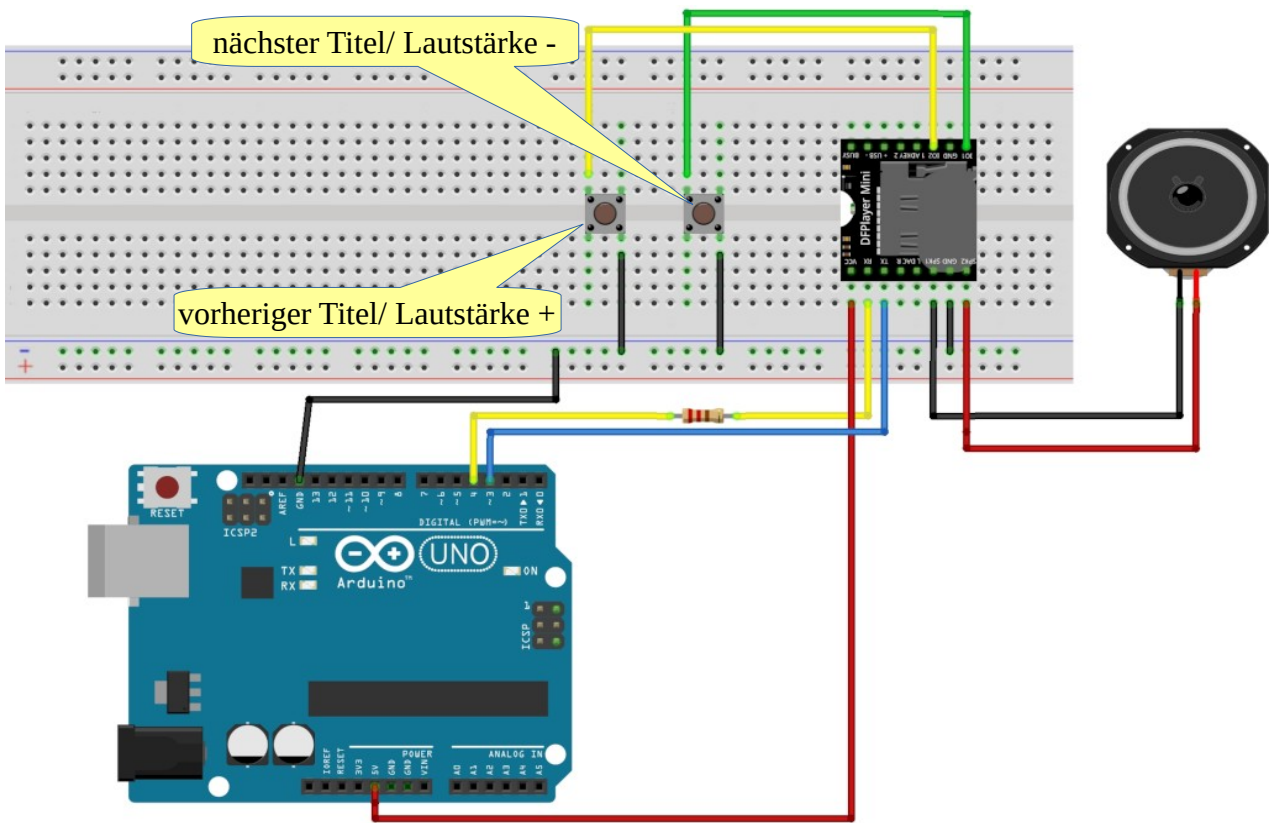
Die Pins RX und TX können an beliebigen Ports außer 0 und 1 angeschlossen werden. Sie werden mit der Bibliothek SoftwareSerial angesteuert.



Benötigte Bauteile:

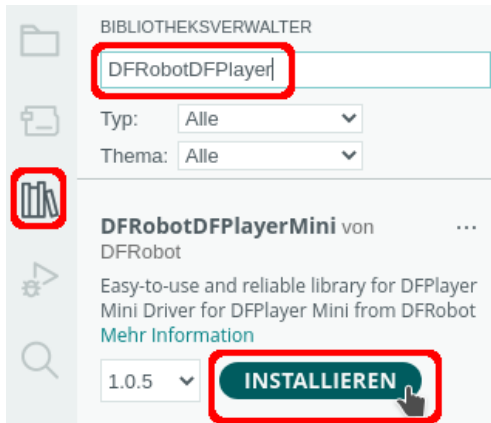
- ➔ Lautsprecher
- ➔ 2 Taster
- ➔ DFPlayer Mini
- ➔ Leitungsdrähte

Baue die Schaltung auf.



fritzing

Bevor du starten kannst, musst du eine Bibliothek installieren:



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

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

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

// Name des Players
DFRobotDFPlayerMini mp3Player;

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

    // Player starten
    // mp3Player.begin(DFPlayerSoftwareSerial, false, true);
    mp3Player.begin(DFPlayerSoftwareSerial, true, true);

    /*
    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);

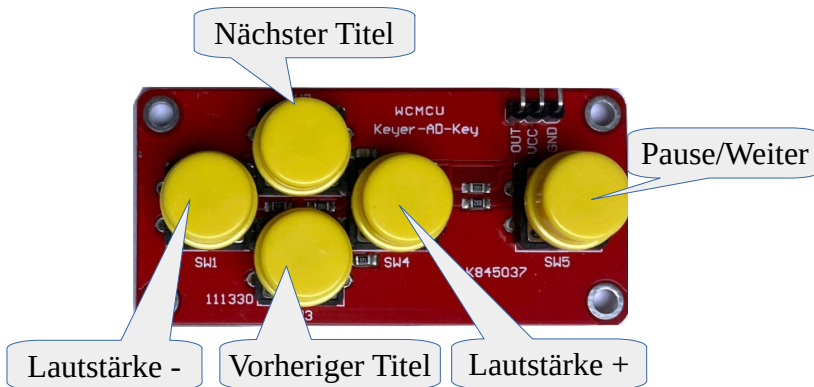
    // alle Titel in zufälliger Reihenfolge spielen
    mp3Player.randomAll();
}
```

```
void loop()
{
  // bleibt leer, Programm läuft nur einmal
}
```

mp3-Player mit Tastenpad

Der mp3-Player soll über das Tastenpad gesteuert werden.

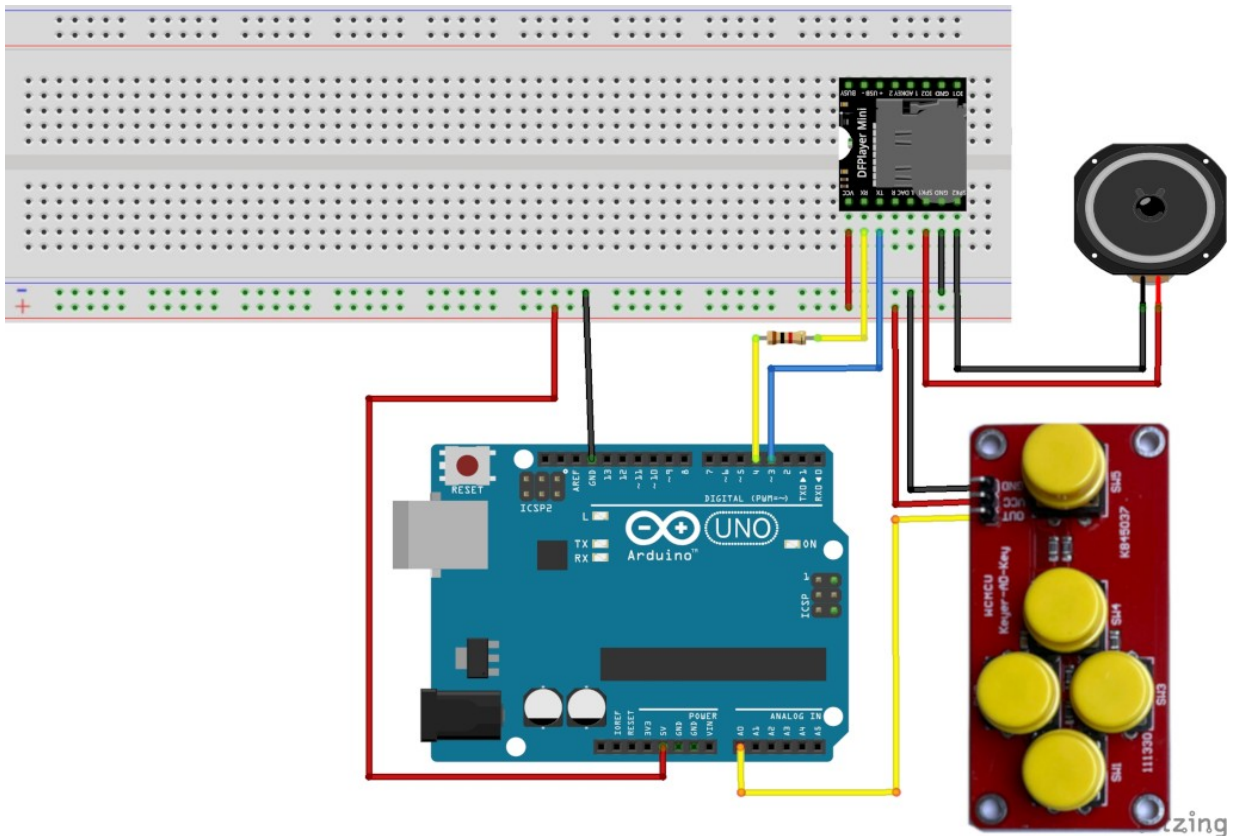
Die Tastenbelegung:



Benötigte Bauteile:

- ➔ Lautsprecher
- ➔ Tastenpad
- ➔ DFPlayer Mini
- ➔ Leitungsdrähte

Baue die Schaltung auf.



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

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

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

// 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, false, true);

  /*
   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);

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



Verwende die Funktion Tasterabfrage.

```
int Tasterabfrage()
{
  int Analogwert = analogRead(A0);

  // kurzes delay() -> doppelten Tastendruck so weit wie möglich verhindern
  delay(200);

  // Serial.println(Analogwert);

  /*
   A0 gibt je nach Taster einen Wert aus
   über den Seriellen Monitor wird dieser Wert angezeigt
   und kann dann eventuell angepasst werden
  */
  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);
    Serial.println("Lautstärke: " + String(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("Lautstärke: " + String(Lautstaerke));
  }

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