

## Jukebox mit Klingeltönen

|                                                              |    |
|--------------------------------------------------------------|----|
| RTTTL Klingeltöne.....                                       | 1  |
| Jukebox im Seriellen Monitor.....                            | 1  |
| Verwendete Hardware.....                                     | 2  |
| Benötigte Bibliothek installieren.....                       | 2  |
| Definition der Variablen und Einbinden der Bibliothek.....   | 3  |
| Der setup-Teil.....                                          | 4  |
| Der loop-Teil.....                                           | 4  |
| Die Funktion ZeigeTitel.....                                 | 6  |
| Jukebox mit Tastenpad und LCD.....                           | 7  |
| Benötigte Bauteile:.....                                     | 7  |
| Der Schaltplan.....                                          | 7  |
| Zusätzlich benötigte Bibliothek installieren.....            | 8  |
| Definition der Variablen und Einbinden der Bibliotheken..... | 8  |
| Der setup-Teil.....                                          | 9  |
| Der loop-Teil.....                                           | 9  |
| Die Funktion ZeigeTitel.....                                 | 11 |

In dieser Anleitung wird beschrieben, wie du mit Lautsprecher, LCD und Tastenpad Klingeltöne abspielen kannst.

## RTTTL Klingeltöne

RTTTL (Ring Tone Text Transfer Language) sind Klingeltöne, die ursprünglich von Nokia für ihre Mobiltelefone entwickelt wurden. Ein Klingelton muss drei Bestandteile haben:

- ➡ den Namen des Klingeltons
- ➡ die Standardvorgaben: die Dauer der Noten und die verwendete Oktave
- ➡ die durch Kommata getrennten Noten (internationale Notenbezeichnung c, d, e, f, g, a, b)

alle Teile sind durch einen Doppelpunkt getrennt

### Beispiel Te Deum:

TeDeum:d=4,o=5,b=63:8c,8f,16f,16g,8a

d=4: Standardnotenlänge Viertelnote

o=5: Angabe der Oktave

b=63: Abspielgeschwindigkeit 63 Schläge pro Minute

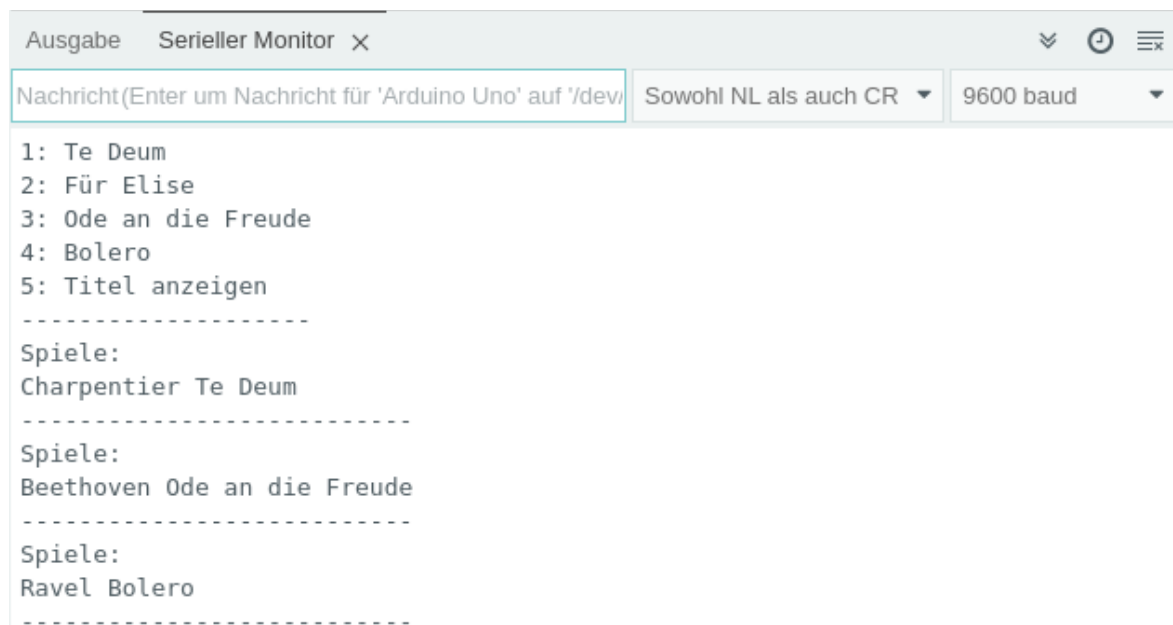
8c: Note C als Achtelnote gespielt

Klingeltöne zum Download bei [PICAXE](#) (abgerufen am 8.5.23)

## Jukebox im Seriellen Monitor

Das Programm verwendet im ersten Schritt den Seriellen Monitor um verschiedene Melodien abzuspielen.

So sieht es aus:

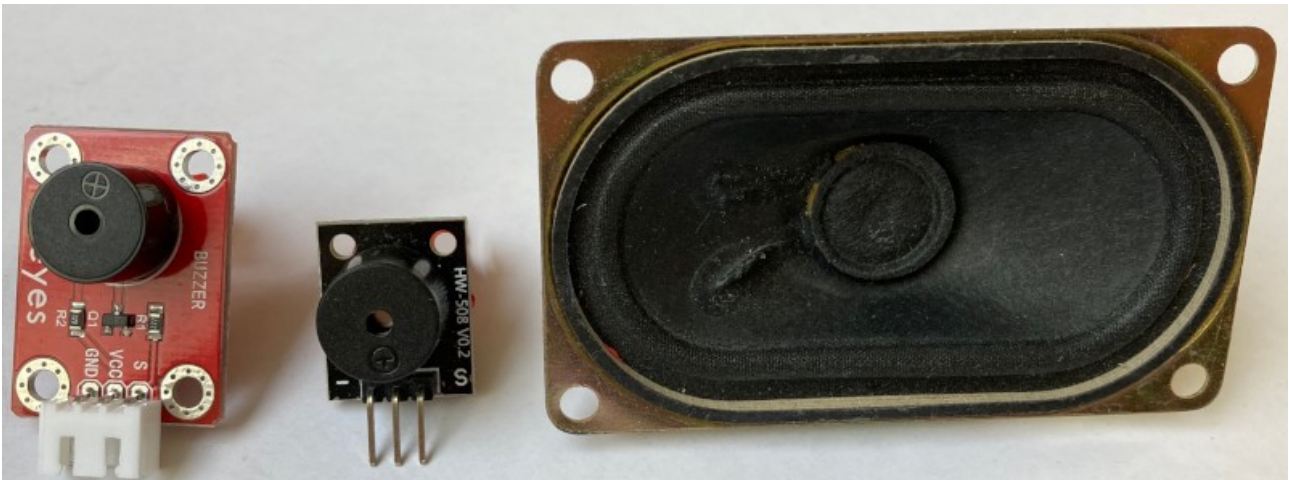


```

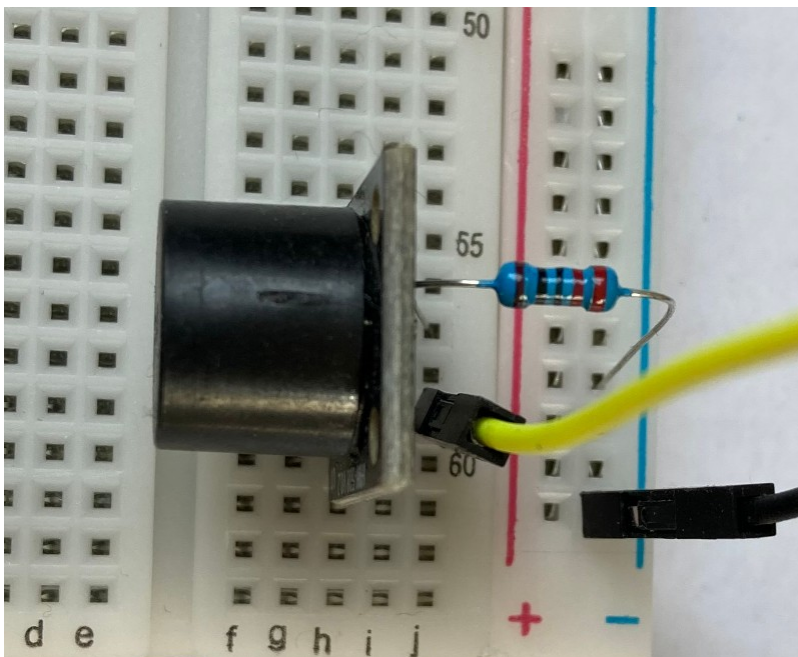
Ausgabe Serieller Monitor x
Nachricht(Enter um Nachricht für 'Arduino Uno' auf '/dev/... Sowohl NL als auch CR 9600 baud
1: Te Deum
2: Für Elise
3: Ode an die Freude
4: Bolero
5: Titel anzeigen
-----
Spiele:
Charpentier Te Deum
-----
Spiele:
Beethoven Ode an die Freude
-----
Spiele:
Ravel Bolero
-----

```

## Verwendete Hardware

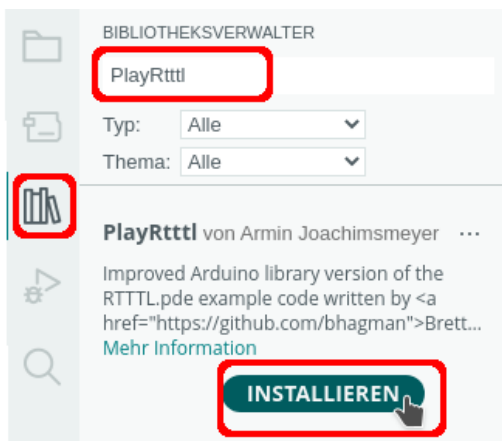


Beispiele für Lautsprecher



Der Lautsprecher wird mit dem Signalpin (gelbes Kabel) an den Pin 7 des Arduinos angeschlossen, das schwarze Kabel führt zu GND am Arduino. Der Widerstand dient nur dazu die Lautstärke zu reduzieren.

## Benötigte Bibliothek installieren



Klicke auf das Symbol Bibliothek und suche im nächsten Dialog nach der Bibliothek PlayRtttl.

## Definition der Variablen und Einbinden der Bibliothek

Die Klingeltöne müssen für den Befehl zum Abspielen als char-Array definiert werden. Die eckigen Klammern zeigen, dass es sich um ein Array handelt.

Titel und Interpret sollen aus dem Array extrahiert werden. # markiert den Beginn der Noten.

Der / dient als Trennung zwischen Interpret und Titel.

Leerzeichen sind in Variablen nicht erlaubt.

Bei Titel, die ein Leerzeichen enthalten, wird es zunächst mit einem Unterstrich „maskiert“. Der Unterstrich wird später durch ein Leerzeichen ersetzt.



**Beachte, dass die char-Arrays im Programm in eine Zeile geschrieben werden müssen**

```
char TeDeum[] =
"Charpentier/Te_Deum#:d=4,o=5,b=63:8c,8f,16f,16g,8a,8f,c6,8a,8a,8a#,16c6,16a#,16a,16a#,8c6,16g,16f,16g,16a,8g,8c,8f,16f,16g,8a,8f,c6,8a,8a,16a#,16c6,16a,16a#,g,16f,2f";

char FuerElise[] =
"Beethoven/Für_Elise#:d=8,o=5,b=125:32p,e6,d#6,e6,d#6,e6,b,d6,c6,4a.,32p,c,e,a,4b.,32p,e,g#,b,4c.6,32p,e,e6,d#6,e6,d#6,e6,b,d6,c6,4a.,32p,c,e,a,4b.,32p,d,c6,b,2a";

char OdeandieFreude[] =
"Beethoven/Ode_an_die_Freude#:d=4,o=6,b=100:a5,a5,a_5,c,c,a_5,a5,g5,f5,f5,g5,a5,a.5,8g5,2g5,";

char Bolero[] =
"Ravel/Bolero#:d=4,o=5,b=80:c6,8c6,16b,16c6,16d6,16c6,16b,16a,8c6,16c6,16a,c6,8c6,16b,16c6,16a,16g,16e,16f,2g,16g,16f,16e,16d,16e,16f,16g,16a,g,g,";
```

```
// Pin des Lautsprechers
# define Lautsprecher 7

// Interpret und Titel als String
String TitelSong;

// Rtttl-Codes der Titel
char TeDeum[] =
"Charpentier/
Te_Deum#:d=4,o=5,b=63:8c,8f,16f,16g,8a,8f,c6,8a,8a,8a#,16c6,16a#,16a,16a#,8c6,16g,16f,
16g,16a,8g,8c,8f,16f,16g,8a,8f,c6,8a,8a,16a#,16c6,16a,16a#,g,16f,2f";

char FuerElise[] =
"Beethoven/
Für_Elise#:d=8,o=5,b=125:32p,e6,d#6,e6,d#6,e6,b,d6,c6,4a.,32p,c,e,a,4b.,32p,e,g#,b,4c.6,
32p,e,e6,d#6,e6,d#6,e6,b,d6,c6,4a.,32p,c,e,a,4b.,32p,d,c6,b,2a";

char OdeandieFreude[] =
"Beethoven/
Ode_an_die_Freude#:d=4,o=6,b=100:a5,a5,a_5,c,c,a_5,a5,g5,f5,f5,g5,a5,a.5,8g5,2g5,";

char Bolero[] =
"Ravel/
Bolero#:d=4,o=5,b=80:c6,8c6,16b,16c6,16d6,16c6,16b,16a,8c6,16c6,16a,c6,8c6,16b,16c6,16a,
16g,16e,16f,2g,16g,16f,16e,16d,16e,16f,16g,16a,g,g,";
```

## Der setup-Teil

Hier wird der Serielle Monitor gestartet und das Menü angezeigt.

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

  // auf serielle Verbindung warten
  while (!Serial) { ; }
  delay(500);

  // Menü im Seriellen Monitor
  Serial.println("1: Te Deum");
  Serial.println("2: Für Elise");
  Serial.println("3: Ode an die Freude");
  Serial.println("4: Bolero");
  Serial.println("5: Titel anzeigen");
  Serial.println("-----");
}
```

## Der loop-Teil

Erläuterungen der Anweisungen:

**while (Serial.available());**

Die nächsten Anweisungen werden nur dann ausgeführt, wenn im Seriellen Monitor etwas eingegeben wird

**char Zeichen = Serial.read();**

Vom Seriellen Monitor wird ein Zeichen gelesen

**if (Zeichen == '1')**

Wenn 1 eingegeben wurde, werden die Anweisungen in den geschweiften Klammern ausgeführt. Entsprechende Anweisungen gelten für die übrigen Ziffern

**TitelSong = String(TeDeum);**

Für die „Weiterverarbeitung“ muss das char-Array in einen String umgewandelt werden

**ZeigeTitel(TitelSong);**

Aufruf der Funktion ZeigeTitel (Erläuterung im nächsten Abschnitt)

**playRtttlBlocking(Lautsprecher, TeDeum);**

Titel auf dem Pin des Lautsprechers abspielen

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

    if (Zeichen == '1')
    {
      // char-Array des Titels in String umwandeln
      TitelSong = String(TeDeum);

      // Funktion aufrufen, Interpret/Titel extrahieren
      ZeigeTitel();

      // Titel spielen
      playRtttlBlocking(Lautsprecher, TeDeum);
    }

    if (Zeichen == '2')
    {
      TitelSong = String(FuerElise);
      ZeigeTitel();
      playRtttlBlocking(Lautsprecher, FuerElise);
    }

    if (Zeichen == '3')
    {
      TitelSong = String(OdeandieFreude);
      ZeigeTitel();
      playRtttlBlocking(Lautsprecher, OdeandieFreude);
    }

    if (Zeichen == '4')
    {
      TitelSong = String(Bolero);
      ZeigeTitel();
      playRtttlBlocking(Lautsprecher, Bolero);
    }

    if (Zeichen == '5')
    {
      Serial.println("1: Te Deum");
      Serial.println("2: Für Elise");
      Serial.println("3: Ode an die Freude");
      Serial.println("4: Bolero");
      Serial.println("5: Titel anzeigen");
      Serial.println("-----");
    }
  }
}
```

## Die Funktion ZeigeTitel

Der loop-Teil ruft die Funktion Zeigetitel auf. Der String Titelsong muss jetzt nach Namen des Interpreten und Namen des Titels getrennt werden.



Mit Funktionen können Programmteile erstellt werden, die eine bestimmte Aufgabe ausführen und dann zu der Stelle im Programm zurückkehren, von dem aus die Funktion aufgerufen wurde. Der Vorteil ist, dass häufig verwendete Programmteile nur einmal erstellt werden müssen. Sie können dann an beliebiger Stelle im Programm aufgerufen werden. Eine Funktion kann einen Befehl ausführen und eine Variable an die Stelle zurückgeben (return), von der sie aufgerufen wurde oder lediglich eine Folge von Befehlen "abarbeiten". In diesem Fall arbeitet sie eine Folge von Befehlen ab.

```
TitelSong = TitelSong.substring(0, TitelSong.indexOf("#"));
```

Nach dem # suchen (indexOf) und mit substring den ersten Teil vom ersten Buchstaben (0) bis zum # extrahieren

```
TitelSong.replace("_", " ");
```

Alle \_ durch Leerzeichen ersetzen.

```
String Interpret = TitelSong.substring(0, TitelSong.indexOf("/"));
```

Die Buchstaben im String TitelSong von 0 bis zum / enthalten den Interpreten

```
String Titel = TitelSong.substring(TitelSong.indexOf("/") + 1, TitelSong.length());
```

Der zweite Teil des Strings TitelSong enthält den Titel. Er beginnt mit dem Zeichen hinter dem / (+1) und endet mit der Länge des Strings (length)

```
void ZeigeTitel()
{
    // Teil des Strings bis zum # kürzen,
    // enthält die Informationen zu Interpret und Titel
    TitelSong = TitelSong.substring(0, TitelSong.indexOf("#"));

    // _ durch leerzeichen ersetzen
    TitelSong.replace("_", " ");

    // erster Teil des Strings bis zum / -> Name des Interpreten
    String Interpret = TitelSong.substring(0, TitelSong.indexOf("/"));

    // zweiter Teil des Strings vom / + 1 bis zum Ende des Strings -> Name des Titels
    String Titel = TitelSong.substring(TitelSong.indexOf("/") + 1, TitelSong.length());

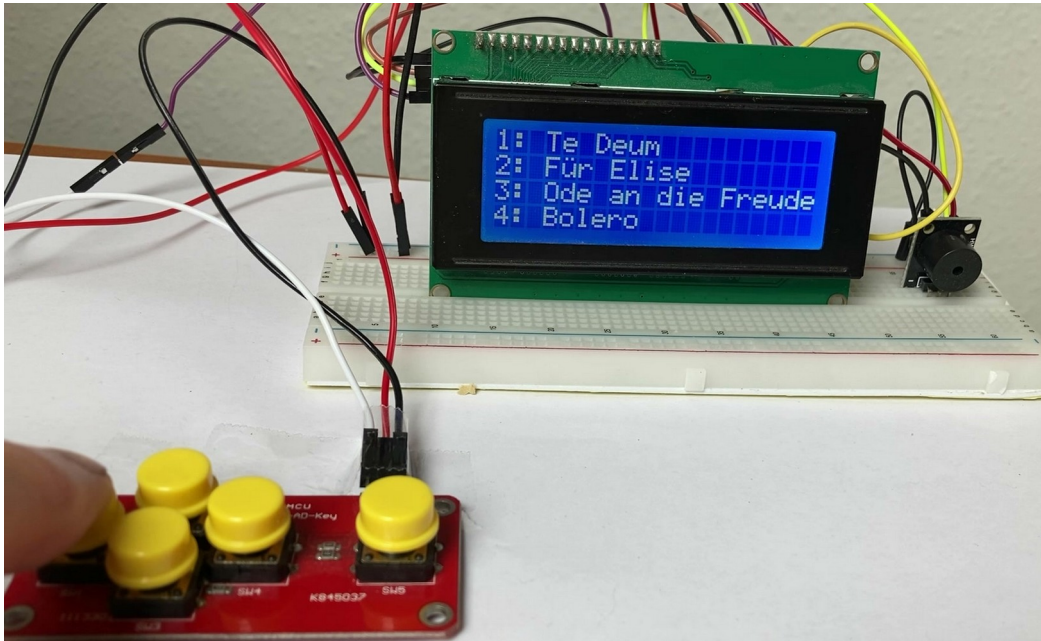
    Serial.println("Spiele:");

    // Interpret anzeigen
    Serial.print(Titel + " ");

    // Titel anzeigen
    Serial.println(Interpret);
    Serial.println("-----");
}
```

## Jukebox mit Tastenpad und LCD

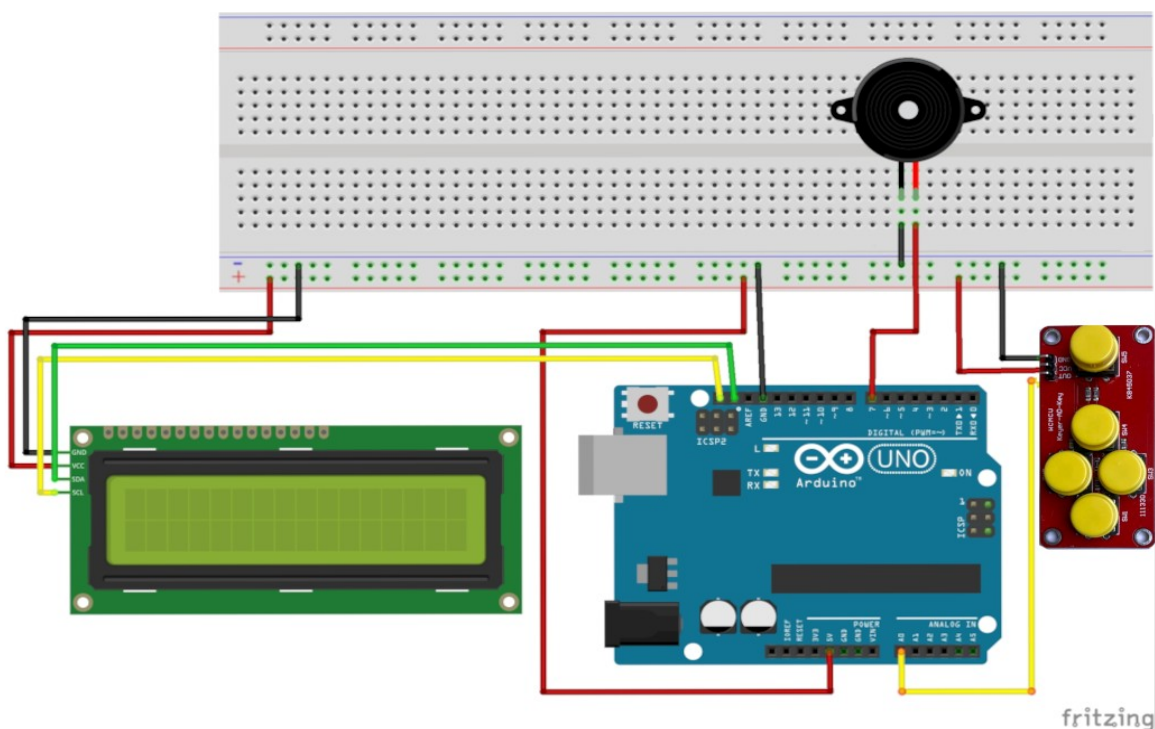
Die Tasten eines Tastenpads sollen jetzt die Titel auswählen.



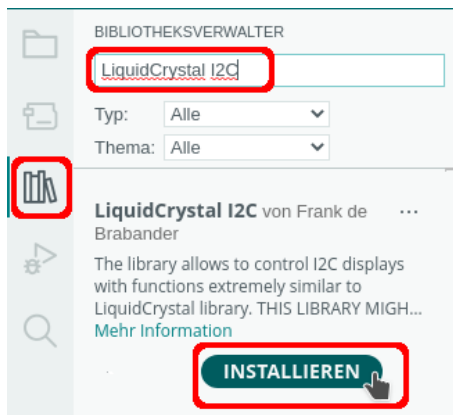
### Benötigte Bauteile:

- ➡ Lautsprecher
- ➡ Tastenpad
- ➡ 4-zeiliges LCD 1602 Modul
- ➡ Leitungsdrähte

### Der Schaltplan



## Zusätzlich benötigte Bibliothek installieren



Klicke auf das Symbol Bibliothek und suche im nächsten Dialog nach der Bibliothek LiquidCrystal I2C.

## Definition der Variablen und Einbinden der Bibliotheken

```
# include <PlayRtttl.hpp>
# include <LiquidCrystal_I2C.h>

// Name des LCDs (lcd) festlegen
LiquidCrystal_I2C lcd(0x27, 20, 4);

# define Lautsprecher 7

// Variable für die gedrückte Taste des Tastenpads
int GedrueckteTaste;

String TitelSong;

char TeDeum[] =
"Charpentier/
Te_Deum#:d=4,o=5,b=63:8c,8f,16f,16g,8a,8f,c6,8a,8a,8a#,16c6,16a#,16a,16a#,8c6,16g,16f,16g,16a,8g,8c,8f,16f,16g,8a,8f,c6,8a,8a,16a#,16c6,16a,16a#,g,16f,2f";

char FuerElise[] =
"Beethoven/
Für_Elise#:d=8,o=5,b=125:32p,e6,d#6,e6,d#6,e6,b,d6,c6,4a.,32p,c,e,a,4b.,32p,e,g#,b,4c.6,32p,e,e6,d#6,e6,d#6,e6,b,d6,c6,4a.,32p,c,e,a,4b.,32p,d,c6,b,2a";

char OdeandieFreude[] =
"Beethoven/
Ode_an_die_Freude#:d=4,o=6,b=100:a5,a5,a_5,c,c,a_5,a5,g5,f5,f5,g5,a5,a.5,8g5,2g5,";

char Bolero[] =
"Ravel/
Bolero#:d=4,o=5,b=80:c6,8c6,16b,16c6,16d6,16c6,16b,16a,8c6,16c6,16a,c6,8c6,16b,16c6,16a,16g,16e,16f,2g,16g,16f,16e,16d,16e,16f,16g,16a,g,g,";
```

## Der setup-Teil

Im setup-Teil wird das LCD gestartet.

```
void setup()
{
  // LCD starten
  lcd.init();

  // Hintergrundbeleuchtung einschalten
  lcd.backlight();
}
```

## Der loop-Teil

Das gedrückte Taste des Tastenpads wird mit analogRead am Eingang A0 abgefragt:

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

Die Analogen Eingänge liefern Werte zwischen 0 und 1023. Je nach gedrückter Taste ergeben sich diese Werte:

| Taste          | Ungefäher analoger Wert |
|----------------|-------------------------|
| links          | 0 - 5                   |
| oben           | 30 - 40                 |
| unten          | 85 - 95                 |
| rechts         | 160 - 170               |
| außen          | 330 - 380               |
| nicht gedrückt | 1023                    |

Eine switch case Abfrage weist je nach Wert die entsprechende Aktion zu.  
Beispiel Abfrage der linken Taste:

```
// linke Taste;
switch (Analogwert)
{
  case 0 ... 20:
    // char-Array des Titels in String umwandeln
    TitelSong = String(TeDeum);

    // Funktion aufrufen, Interpret/Titel extrahieren
    ZeigeTitel();

    // Titel spielen
    playRtttlBlocking(Lautsprecher, TeDeum);
    break;
```

case 0 ... 20 fragt den Wertebereich zwischen 0 und 20 ab. Die übrigen Wertebereiche werden entsprechend zugewiesen.



**Beachte das Leerzeichen vor und hinter den drei Punkten!**

Der vollständige loop-Teil:

```
void loop()
{
  // Tastenpad am Eingang A0 abfragen
  int Analogwert = analogRead(A0);

  // linke Taste;
  switch (Analogwert)
  {
    case 0 ... 20:
      // char-Array des Titels in String umwandeln
      TitelSong = String(TeDeum);

      // Funktion aufrufen, Interpret/Titel extrahieren
      ZeigeTitel();

      // Titel spielen
      playRtttlBlocking(Lautsprecher, TeDeum);
      break;

    // obere Taste
    case 30 ... 60:
      TitelSong = String(FuerElise);
      ZeigeTitel();
      playRtttlBlocking(Lautsprecher, FuerElise);
      break;

    // rechte Taste
    case 70 ... 120:
      TitelSong = String(OdeandieFreude);
      ZeigeTitel();
      playRtttlBlocking(Lautsprecher, OdeandieFreude);
      break;

    // untere Taste
    case 150 ... 200:
      TitelSong = String(Bolero);
      ZeigeTitel();
      playRtttlBlocking(Lautsprecher, Bolero);
      break;

    // rechte äußere Taste
    case 300 ... 400:
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("1: Te Deum");
      lcd.setCursor(0, 1);
      lcd.print("2: F\365r Elise");
      lcd.setCursor(0, 2);
      lcd.print("3: Ode an die Freude");
      lcd.setCursor(0, 3);
      lcd.print("4: Bolero");
      break;
  }
}
```

## Die Funktion ZeigeTitel

Die Funktion ZeigeTitel zeigt Interpret und Titel auf dem LCD an.

```
void ZeigeTitel(String TitelSong)
{
    // Teil des Strings bis zum # kürzen
    TitelSong = TitelSong.substring(0, TitelSong.indexOf("#"));

    // _ durch leerzeichen ersetzen
    TitelSong.replace("_", " ");

    // erster Teil des Strings bis zum / -> Name des Interpreten
    String Interpret = TitelSong.substring(0, TitelSong.indexOf("/"));

    // zweiter Teil des Strings von / + 1 bis zum Ende des Strings -> Name des Titels
    String Titel = TitelSong.substring(TitelSong.indexOf("/") + 1, TitelSong.length());

    // Titel auf dem LCD anzeigen
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Spiele:");
    lcd.setCursor(0, 1);
    lcd.print(Titel);
    lcd.setCursor(0, 2);
    lcd.print(Interpret);
}
```