

Den Tasten 1 bis 8 sollen die Töne der C-Dur Tonleiter zugeordnet werden.

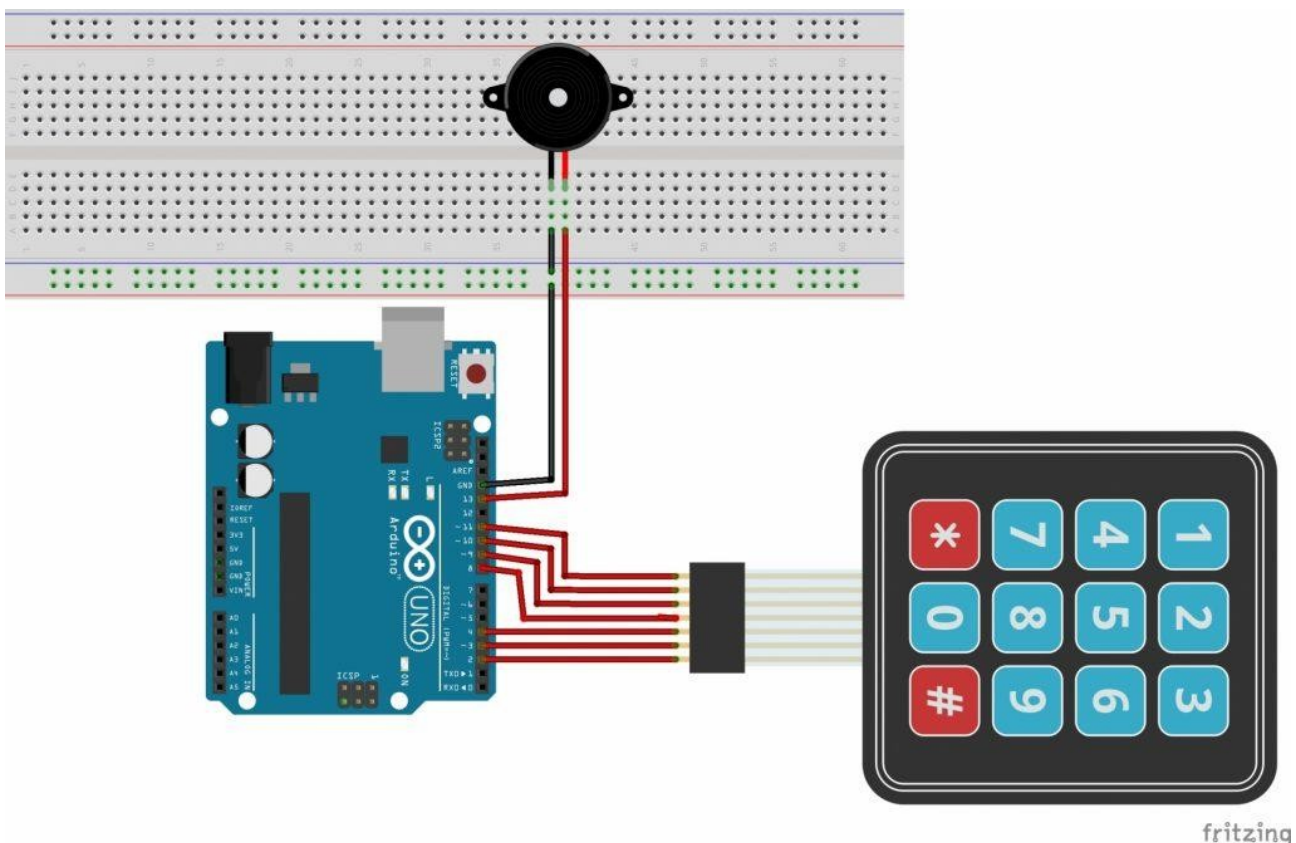
Note	Frequenz	Taste auf dem Tastenpad
c'	261	Taste 1
d'	293	Taste 2
e'	339	Taste 3
f'	349	Taste 4
g'	391	Taste 5
a'	440	Taste 6
h'	493	Taste 7
c''	523	Taste 8

Hier findest du eine **Übersicht** über die Frequenzen.

Benötigte Bauteile:

- ➔ Lautsprecher
- ➔ Tastenfeld 3×4
- ➔ Leitungsdrähte

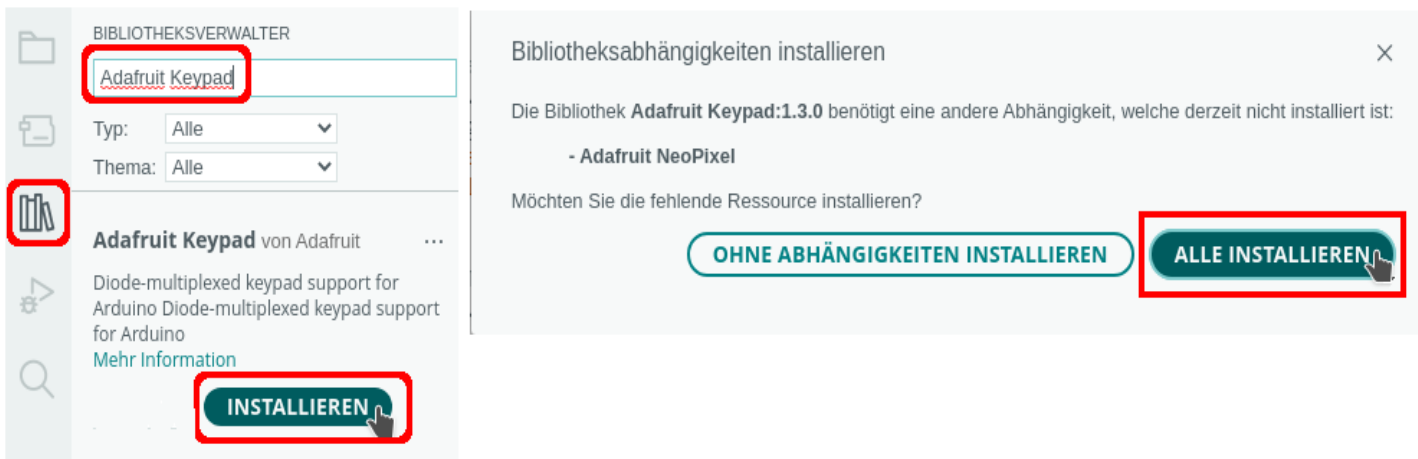
Baue die Schaltung auf.



fritzing

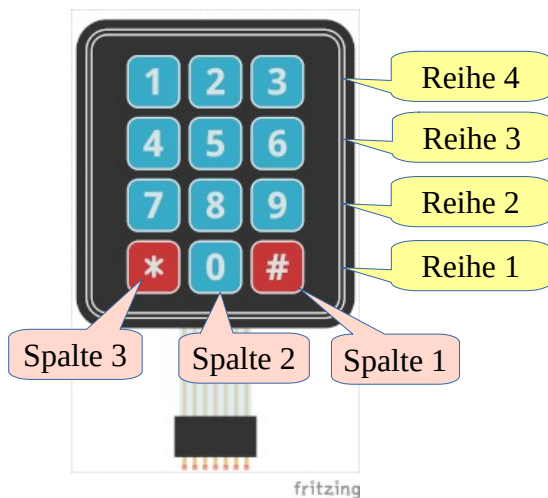
Bei einem Tastendruck wird der Ton wiedergegeben, die Tasten * und # schalten ihn wieder aus.

Benötigte Bibliothek:



Das Tastenfeld besteht aus Reihen und Spalten, die von unten nach oben in einem **Array** angeordnet werden:

```
// Array 3 x 4
char Tasten[REIHEN][SPALTEN] = {
  Spalte 1 Spalte 2 Spalte 3
  {'#', '0', '*'}, Reihe 1
  {'9', '8', '7'}, Reihe 2
  {'6', '5', '4'}, Reihe 3
  {'3', '2', '1'} Reihe 4
};
```



Binde die benötigte Bibliothek ein und definiere die Variablen.

```
# include <Adafruit_Keypad.h>

int LAUTSPRECHER = 13;

// Größe des Tastenfeldes
// 3 Spalten
const byte SPALTEN = 3;

// 4 Zeilen
const byte REIHEN = 4;

// die Ziffern/Zeichen:
// Array 3 x 4
char Tasten[REIHEN][SPALTEN] =
{
  { '#', '0', '*' },
  { '9', '8', '7' },
  { '6', '5', '4' },
  { '3', '2', '1' }
};

// die Pins für die 3 Spalten
byte SpaltenPins[SPALTEN] = { 3, 4, 5 };

// die Pins für die 4 Zeilen
byte ReihenPins[REIHEN] = { 6, 7, 8, 9 };

// TastenFeld → Name des Keypads
// -> Zuordnung der Pins zu den REIHEN und SPALTEN des Arrays
Adafruit_Keypad TastenFeld = Adafruit_Keypad(makeKeymap(Tasten), ReihenPins,
SpaltenPins, REIHEN, SPALTEN);
/*
  C-Dur Tonleiter Frequenzen
  c' -> 264
  d' -> 297
  e' -> 330
  f' -> 352
  g' -> 396
  a' -> 440
  h' -> 495
  c'' -> 528
*/
```

Der setup-Teil:

```
void setup()
{
  // Tastenfeld starten
  Tastenfeld.begin();
  pinMode(LAUTSPRECHER, OUTPUT);
  Serial.begin(9600);

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

Der loop-Teil. Beachte die Kommentare.

```
void loop()
{
  // gedrückte Taste lesen
  Tastenfeld.tick();
  while (Tastenfeld.available())
  {
    keypadEvent Taste = Tastenfeld.read();

    // wenn die Taste losgelassen wurde -> Lautsprecher ausschalten
    if (Taste.bit.EVENT == KEY_JUST_RELEASED) noTone(LAUTSPRECHER);

    // solange die Taste gedrückt wird -> Ton abspielen
    if (Taste.bit.EVENT == KEY_JUST_PRESSED)
    {
      // gedrückte Taste anzeigen
      Serial.print("ASCII: " +String(Taste.bit.KEY) + "\t");
      Serial.print("Taste: ");
      Serial.println((char)Taste.bit.KEY);
      /*
       * Tastencode entspricht dem ASCII-Code:
       * 49 = 1
       * 50 = 2
       * . . .
       * 56 = 8
       */
      switch (Taste.bit.KEY)
      {
        case 49:
          tone(LAUTSPRECHER, 264);
          break;

        case 50:
          tone(LAUTSPRECHER, 297);
          break;
      }
    }
  }
}
```

```
case 51:
  tone(LAUTSPRECHER, 330);
  break;

case 52:
  tone(LAUTSPRECHER, 352);
  break;

case 53:
  tone(LAUTSPRECHER, 396);
  break;

case 54:
  tone(LAUTSPRECHER, 440);
  break;

case 55:
  tone(LAUTSPRECHER, 495);
  break;

case 56:
  tone(LAUTSPRECHER, 520);
  break;
}
}
}
}
```