

Alexander Steiger
Erlenweg 21
36110 Schlitz-Hutzdorf
steigeralex@gmx.de

Elektor-Verlag GmbH
Internationaler
R8C-Design-Wettbewerb



Elektronik-Projekt

Alexander Steiger

MusicTree

August 2006

Zusammenfassung

Elektronik muss nicht immer so kompliziert sein. Besonders für Anfänger ist der Einstieg in die Mikrocontroller-Welt sehr schwer. Ohne Spass an der Sache wird sich der Einstieg in diese verborgene Welt sehr zeitraubend und sogar deprimierend gestalten.

Das R8C-Board, das im Heft 12/2005 von Elektor zum ersten Mal vorgestellt wurde, ist reichlich mit guter Peripherie ausgestattet. Deshalb eignet sich die Plattform sehr gut für verschiedene Anwendungen wie z. B. für das in dieser Beschreibung vorgestellte Projekt.

Das Projekt „MusicTree“ ist eine Licht- und Sound-Effekt Schaltung, die in Form von einem Tannenbaum gestaltet wurde. Es sind mehrere Leuchtdioden angeordnet, die nach dem Zufallsprinzip blinken. Desweiteren besitzt der Tannenbaum einen Schallwandler, über den Melodien ausgegeben werden. Der Clou dabei ist, dass diese Melodien im Nokia-Format im Speicher des Mikrocontrollers abgelegt sind, so dass neue Melodien ohne große Probleme hinzugefügt werden können. Schließlich besitzt der Tannenbaum einen lichtempfindlichen Sensor, über den die Art der Lichteffekte abhängig von der Helligkeit im Raum gesteuert wird.

Mit all diesen Eigenschaften eignet sich das Projekt „MusicTree“ sehr gut als das ultimative Weihnachtsgeschenk, das jede Eltern, Verwandte und Freunde mit Sicherheit begeistern wird.

Inhaltsverzeichnis

1	Einleitung	1
2	Beschreibung	3
2.1	Technische Daten	3
2.2	Mechanischer Aufbau	3
2.3	Ansteuerung der LEDs	4
2.4	Melodien	5
2.5	Helligkeitssensor	6
2.6	Menüführung	6
2.7	Schaltplan	8
3	Ausblick	10
A	Bilder	11
B	Melodien	16

Kapitel 1

Einleitung

Jedes Elektronik-Projekt hat seine eigene Geschichte, die meistens nur der Autor selbst kennt. Der Leser erfährt nur sehr selten die Hintergründe zum Projekt. Deshalb möchte ich in diesem Abschnitt die Entstehungsgeschichte des Projektes kurz schildern.

Das Projekt „MusicTree“ entstand in der Vorweihnachtszeit im Dezember 2005, als in Elektor das R8C-Board vorgestellt wurde. Ich kann mich noch an die Werbung in Elektor erinnern. Die Besonderheit war dabei, dass es ein R8C/13-Modul kostenlos als Beilage im Heft gab. Da das Heft sehr gefragt war, musste ich extra zum nächsten Hauptbahnhof fahren. Geizig wie Ich bin, habe ich mir gleich drei Hefte besorgt, um weitere Exemplare zu ergattern, falls Mal was schief laufen sollte. Zuhause ausgepackt, hatte ich nun drei tolle R8C/13-Module auf dem Tisch liegen. Nachdem ich den Artikel in Elektor gut studiert habe, wollte ich dann richtig zur Sache gehen und endlich irgendwas mit dem magischen R8C anstellen.

Das einzige was noch fehlte, war eine Anwendung, um das R8C-Board testen zu können. Was macht man, wenn man überhaupt keine Ahnung hat, was man basteln soll? Natürlich, man holt sich die Ideen aus dem Alltag. Da Weihnachten vor der Tür stand, habe ich mich für eine kleine Licht-Effekt-Schaltung entschieden. LED-Blink-Schaltungen sind bei jedem Elektronik-Freund beliebt. Ab diesem Zeitpunkt an stand die Anwendung fest.

Allerdings war mir die Schaltung für eine anspruchsvolle Anwendung noch zu wenig. Schließlich gab es Schaltungen mit LEDs wie Sand am Meer. Wieso sollte ich also so eine Schaltung aufbauen? Deshalb musste was Neues her, was die Eigenschaften des Mikrocontrollers R8C zwar nicht ganz, aber mehr ausnutzt. Da die Peripherie des Mikrocontrollers bei weitem mehr als nur digitale Ein- und Ausgänge bietet, kamen mit der Zeit immer mehr Kleinigkeiten dazu.

In meiner Bastelkiste habe ich mehrere große Duo-LEDs gefunden. Weiterhin dachte ich mir, dass eine Textanzeige sehr nützlich wäre. Deshalb mussten noch vier 7-Segment Anzeigen her. Zwar sind 7-Segment Anzeigen nur für Zahlen gedacht, aber mit ein bisschen Fantasie kann man darauf auch Buchstaben darstellen. Ich habe extra keine alphanumerischen Displays genommen, weil ich unbedingt Leuchtdioden haben wollte, die man auch im Dunklen gut erkennt. Und die Stromversorgung war nicht das Problem. Ein Netzteil und ein 5V-Festspannungsregler würden alle Anforderungen erfüllen, um eine digitale Schaltung versorgen zu können.

So wurde neben den mehrfarbigen Leuchtdioden und einer vierstelligen 7-Segment Anzeige ein Fotowiderstand für die Helligkeitsmessung, 4 Taster zum Bedienen und ein Piezo-Wandler hinzugefügt. Das Design wurde dem „Tannenbaum-Look“ angepasst. Daraus entstand eine sehr interessante Anwendung, die jeder Geschenkidee standhalten konnte. Was daraus geworden ist, kann man in der Abbildung 1.1 betrachten.

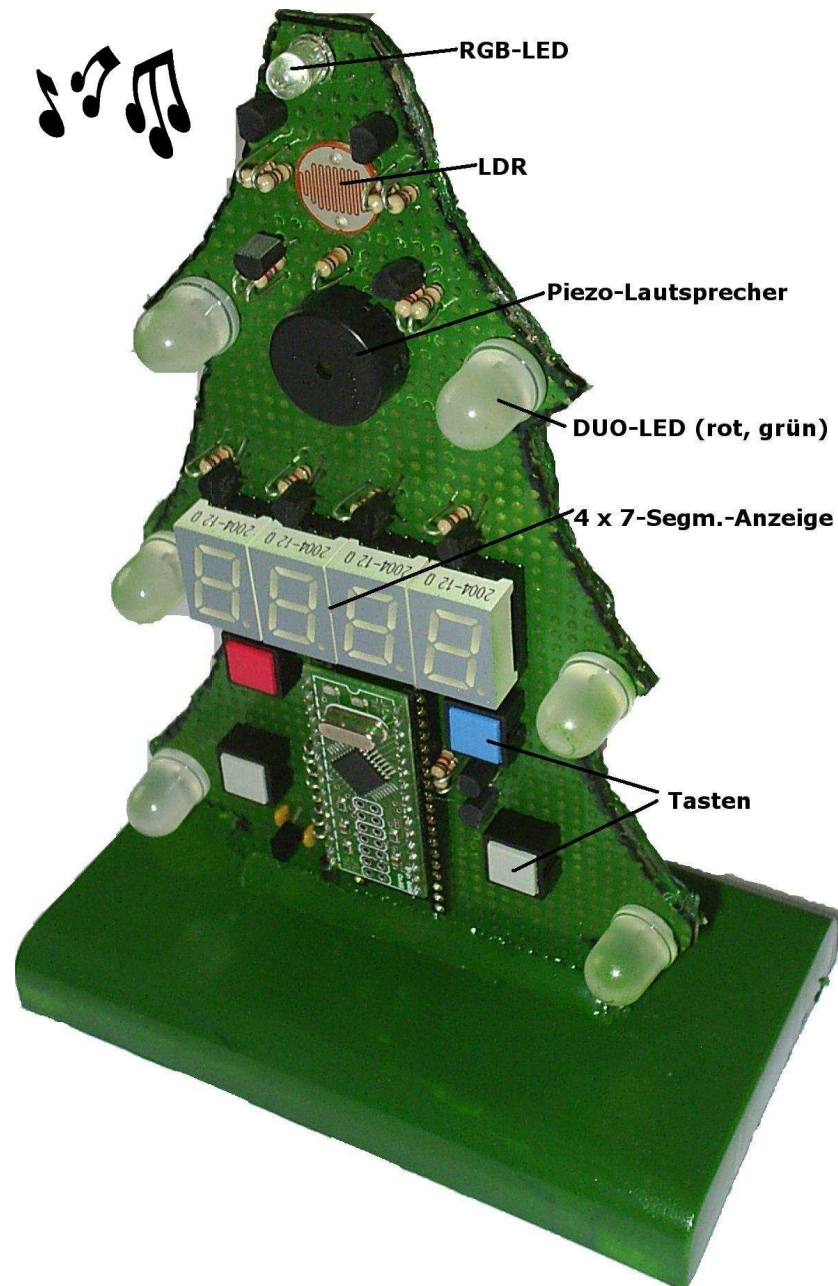


Abbildung 1.1: Übersicht MusicTree.

Kapitel 2

Beschreibung

Nachdem in der Einleitung die Entstehungsgeschichte geschildert wurde, kommen wir nun zur Beschreibung des Projektes. Aber zunächst einmal ein paar Daten zum Projekt „MusicTree“.

2.1 Technische Daten

Abmessungen: Europa-Platine 160 x 100 mm (ausgesägt)

Mikrocontroller: R8C/13 (16k)

Stromversorgung: Netzteil 9-12V (5V-Festspannungsregler)

Sensoren:

- 4 Taster
- LDR (Fotowiderstand)

Aktoren:

- Piezo-Lautsprecher
- 4 x 7 Segm. Anzeigen
- 1 RGB-LED (rot, grün, blau)
- 6 DUO-LEDs (rot, grün)

Anzahl Melodien: ca. 60 (im Nokia-Format)

2.2 Mechanischer Aufbau

Im Wesentlichen besteht der „Tannenbaum“ aus einer ausgesägten Lochrasterplatine, auf der alle elektronischen Bauteile aufgelötet sind. Damit die Platine nicht umfällt, habe ich eine Halbschale eines Gehäuses als Standfuss verwendet.

Es ist auch denkbar ein Layout für das Projekt herzustellen, doch fand ich den Aufbau auf einer Lochrasterplatine relativ einfach. Die elektronischen Bauteile werden zuerst ganz normal mit Schalt Draht verlötet. Da die Rückseite mit all den vielen



Abbildung 2.1: Lochrasterplatine mit Muster.

Drähten nicht besonders schön aussieht und die Kontakte von außen zugänglich sind, habe ich danach die Rückseite mit einer Heißklebepistole versiegelt. Wenn der Kleber getrocknet ist, kann dann die Rückseite lackiert werden. Der einzige Nachteil dieser Methode ist, dass man die Leiterbahnen nicht nachträglich verändern kann. Also muss man erst ganz sicher gehen, dass die Schaltung funktioniert, bevor man die Drähte unwiderruflich versiegelt.

Damit das ganze äußerlich ansprechend wirkt, habe ich sowohl die Platine als auch die Rückseite und den Standfuss passend dazu mit grüner Sprühfarbe, die man für ein Paar Euro in einem Baumarktgeschäft kaufen kann, lackiert. Die Ränder habe ich mit einem Eding nachgezogen.

Auf der Platine sind unter anderem Leuchtdioden, ein Piezo-Lautsprecher, ein Fotowiderstand und Taster zu finden. Die Stromversorgung kommt von einem handelsüblichen Steckernetzteil. Die Spannung wird mit einem Festspannungsregler auf 5V runtergeregelt.

2.3 Ansteuerung der LEDs

Die Leuchtdioden auf der Platine werden im Multiplexbetrieb angesteuert. Dabei kann entweder eine 7-Segment-Anzeige, linke DUO-LED Reihe oder rechte DUO-LED Reihe gleichzeitig angesteuert werden. Da das Auge des Menschen sehr träge ist, merkt es nicht, dass nur einzelne Baugruppen zu einem Zeitpunkt angesteuert werden. Die Voraussetzung ist allerdings, dass man zwischen den einzelnen Baugruppen sehr schnell ($> 50\text{Hz}$) umschaltet.

Da der R8C/13 Mikrocontroller noch freie Ports zur Verfügung hatte, wird die RGB-LED ohne Multiplexbetrieb angesteuert. Es wurde außerdem für jede einzelne LED im RGB-LED-Gehäuse jeweils ein Treibertransistor spendiert. Für die restlichen Leuchtdioden wurden außer für den Multiplexbetrieb keine zusätzlichen Treiber eingesetzt, da der R8C/13 Mikrocontroller Ports mit Treibern besitzt, die man per Software hinzuschalten kann.

2.4 Melodien

Der größte Teil des Speichers vom R8C/13 wurde dazu verwendet, um Melodien abzuspeichern. Insgesamt haben ca. 60 Melodien in dem 16k großen Programmspeicher Platz gefunden. Das Format der Melodien basiert auf dem Nokia Format, das von Nokia-Handys verwendet wird. Damit ist es relativ einfach neue Melodien zu importieren, da das Format sehr verbreitet ist und überall im Internet zu finden ist. Darüber hinaus kann man anhand von Noten eine Melodie abspeichern, wenn man ein wenig das Format verstanden hat.

	C1 - H1	C2 - H2	C3 - H3	C4 - F4
	f (Hz=1/s)	f (Hz=1/s)	f (Hz=1/s)	f (Hz=1/s)
C	261,63	523,25	1046,50	2093,00
Cis/Des	277,18	554,37	1108,73	2217,46
D	293,66	587,33	1174,66	2349,32
Dis/Es	311,13	622,25	1244,51	2489,02
E	329,63	659,26	1318,51	2637,02
F	349,23	698,46	1396,91	2793,83
Fis/Ges	369,99	739,99	1479,98	
G	392,00	783,99	1567,98	
Gis/As	415,30	830,61	1661,21	
A	440,00	880,00	1760,00	
Ais/B	466,16	932,33	1864,66	
H	493,88	987,77	1975,53	

Nokia-Format

Im Folgenden ist ein Beispiel für eine Melodie aufgeführt. Es sind zwei Konstanten, durch die eine Melodie beschrieben wird. Zum einen wird für jede Melodie eine Zahl für das Tempo ($BPM = \text{beats per minute}$) abgespeichert, die die Geschwindigkeit bestimmt, mit der die Melodie abgespielt wird. Weiterhin wird zu jeder Melodie ein String mit den Pseudo-Noten abgespeichert. Viele Melodien sind im Internet zu finden, besonders zu nennen ist die Homepage

<http://nokia.nigelcoldwell.co.uk/tunes.html>.

Wie das Format aufgebaut ist, findet man in dem Manual „Nokia 3210 PDF User's Guide“.

Melodietitel: Abba, Mamma Mia
Tempo (BPM): 40
String: "32f2 32#d2 32f2 8#d2 32#d2 32#d2 32f2 32g2 32f2
 16.#d2 32- 16f2 8#d2 16#g2 32#g2 32#g2 32#g2 16g2
 16.#d2 32- 8#a2 32#a2 32#a2 16#a2 16f2 16g2 8#g2
 16g2 16g2 32g2 16g2 16d2 16#d2 8f2 16f2 8#d2 16#g2
 32#g2 32#g2 32#g2 32g2 32#d2 32f2 16#d2"

Die einzelnen Noten werden durch Leerzeichen voneinander getrennt. Vor der Note steht die Dauer z. B. 16d für 1/16. Durch „-“ wird die Dauer um die Hälfte verlängert. Pausen werden durch „-“ angegeben. Die Oktave wird hinter der Note angegeben. Mit „#“ wird die Note erhöht z. B. steht 16#c1 für „cis“.

2.5 Helligkeitssensor

Mit dem Fotowiderstand kann die Helligkeit im Raum gemessen werden. Der Wert wird mit einem AD/Wandler des R8C/13 Mikrocontrollers gemessen. Dabei wurde der Wertebereich auf 0-1024 beschränkt. Die Null steht für ganz „dunkel“ und die 1024 für ganz „hell“. In dem Helligkeitsmodus wird der Wert entweder als eine Dezimalzahl oder als ein Balken ausgegeben. Mit diesem Sensor wird es erst möglich, dass z. B. die LEDs erst in der Dämmerung blinken.



Abbildung 2.2: Helligkeit als Wert.

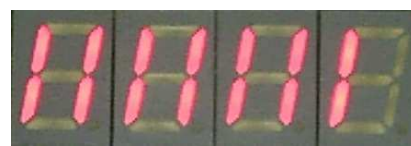


Abbildung 2.3: Helligkeit als Balken.

2.6 Menüführung

Durch die vier Tasten können die unten aufgeführten Modi ausgewählt werden. Es sind keine externen Pull-Up Widerstände für die Taster notwendig, da der R8C/13 Mikrocontroller interne Widerstände besitzt, die man zuschalten kann. Die 7-Segment Anzeige dient zur Navigation durch die Menüs und zur Anzeige von Informationen, wie z. B. Melodienummer, Helligkeit, Text, usw.

Music-Modus (T1)

Melodien werden nach Zufall abgespielt. Mit jedem weiteren Druck auf Taste T1 wird die aktuelle Melodie abgebrochen und eine andere Melodie zufällig ausgewählt und abgespielt.



Abbildung 2.4: Animation „Smile“.

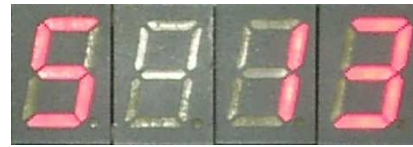


Abbildung 2.5: Nummer der Melodie.

Lauflicht-Modus (T2)

Verschiedene Lichtmuster werden nach Zufall ausgegeben. Die Geschwindigkeit ist abhängig von der Helligkeit, die gerade vom Fotowiderstand (LDR) gemessen wird. Je dunkler die Umgebung ist, desto schneller blinken die Leuchtdioden.

Test-Modus (T3)

Lichtmuster und alle Melodien können mit einem Tastendruck manuell abgerufen werden. Mit jedem Tastendruck wird zuerst entweder ein bestimmtes Lichtmuster oder eine bestimmte Melodie ausgewählt.

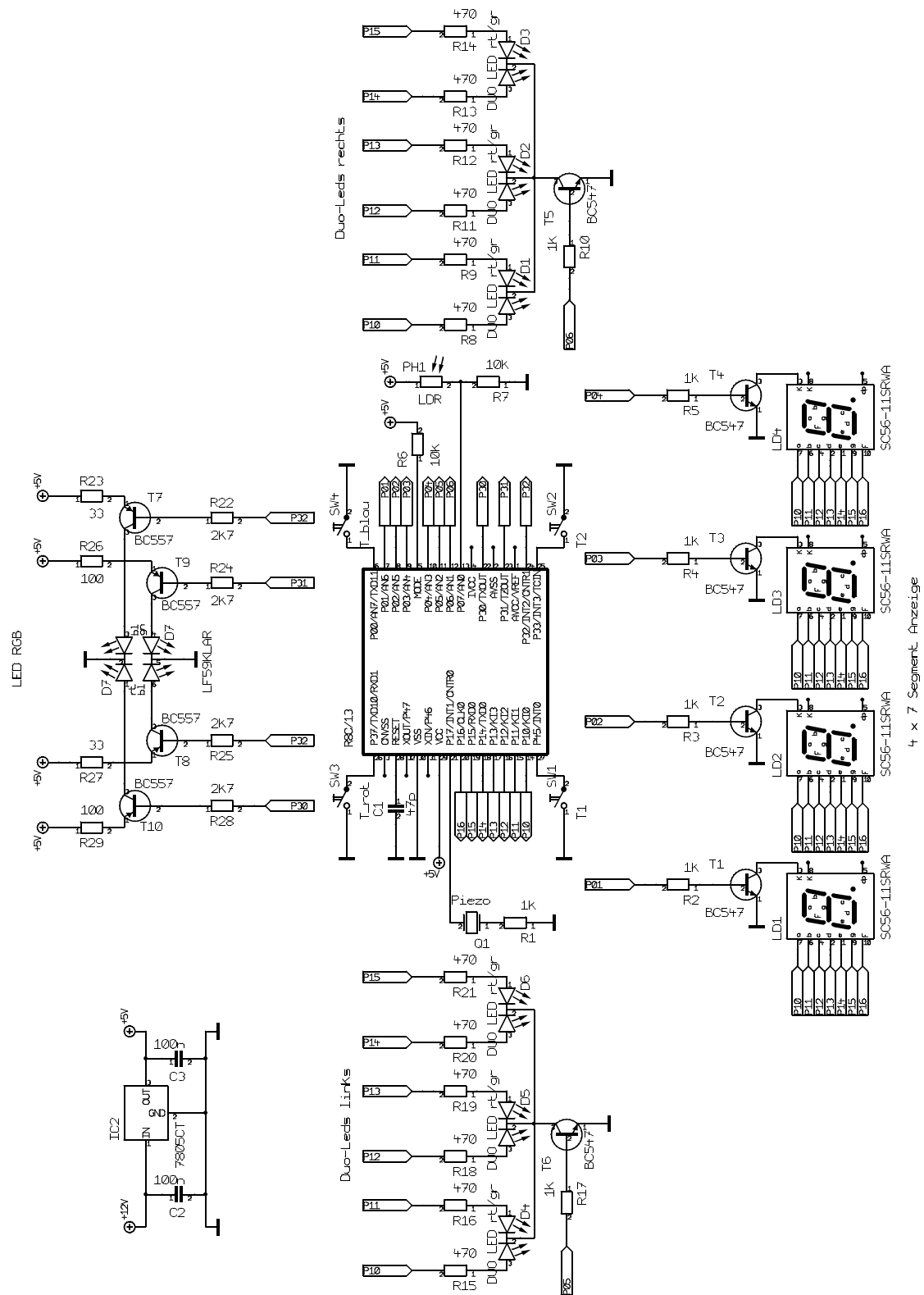
Helligkeits-Modus (T4)

Helligkeit wird mit Hilfe von einem Fotowiderstand (LDR) gemessen. Dabei wird die Helligkeit entweder als Dezimalzahl von 0 (dunkel) bis 1024 (hell) oder als ein Balken mit einer Auflösung von 8 Strichen angezeigt. Mit jedem Tastendruck auf T4 wird zwischen den beiden Modi gewechselt.

2.7 Schaltplan

Stückliste

	Anzahl	Wert
Widerstand (R)	7	1k
Widerstand (R)	4	2k7
Widerstand (R)	2	10k
Widerstand (R)	2	33
Kondensator (C)	1	47p
Kondensator (C)	2	100n
Widerstand (R)	2	100
Widerstand (R)	12	470
Halbleiter IC	1	7805CT
Transistor (T)	6	BC547
Transistor (T)	4	BC557
DUO LED	6	rt/gr
LDR (PH)	1	
RGB-LED (D)	1	LF59KLAR
Piezo-Wandler (Q)	1	
Halbleiter IC	1	R8C/13-Modul
7-Segment-Anzeige (LD)	4	SC56-11SRWA
Taster (SW)	4	$T_1, T_2, T_{blau}, T_{rot}$



Kapitel 3

Ausblick

Das Projekt „MusicTree“ ist aus einer Geschenkidee entstanden, kann aber für Übungszwecke oder als eine schöne Lichteffekt- und Soundeffekt-Schaltung zum Nachbauen interessant sein. Vieles am Projekt kann verbessert und optimiert werden. Dem Leser selbst sei es überlassen, ob das Projekt genau nach der Bauanleitung nachgebaut wird oder gewisse Teile modifiziert werden.

Wenn man nur Melodien abspielen möchte, so reicht es aus nur einen Piezo-Wandler anzuschließen. Die LEDs müssen mit entsprechender Modifikation in der Software nicht unbedingt bestückt werden.

Anhang A

Bilder



Abbildung A.1: MusicTree im Betrieb.



Abbildung A.2: MusicTree von vorne.



Abbildung A.3: MusicTree von der Seite.



Abbildung A.4: Rückseite MusicTree.

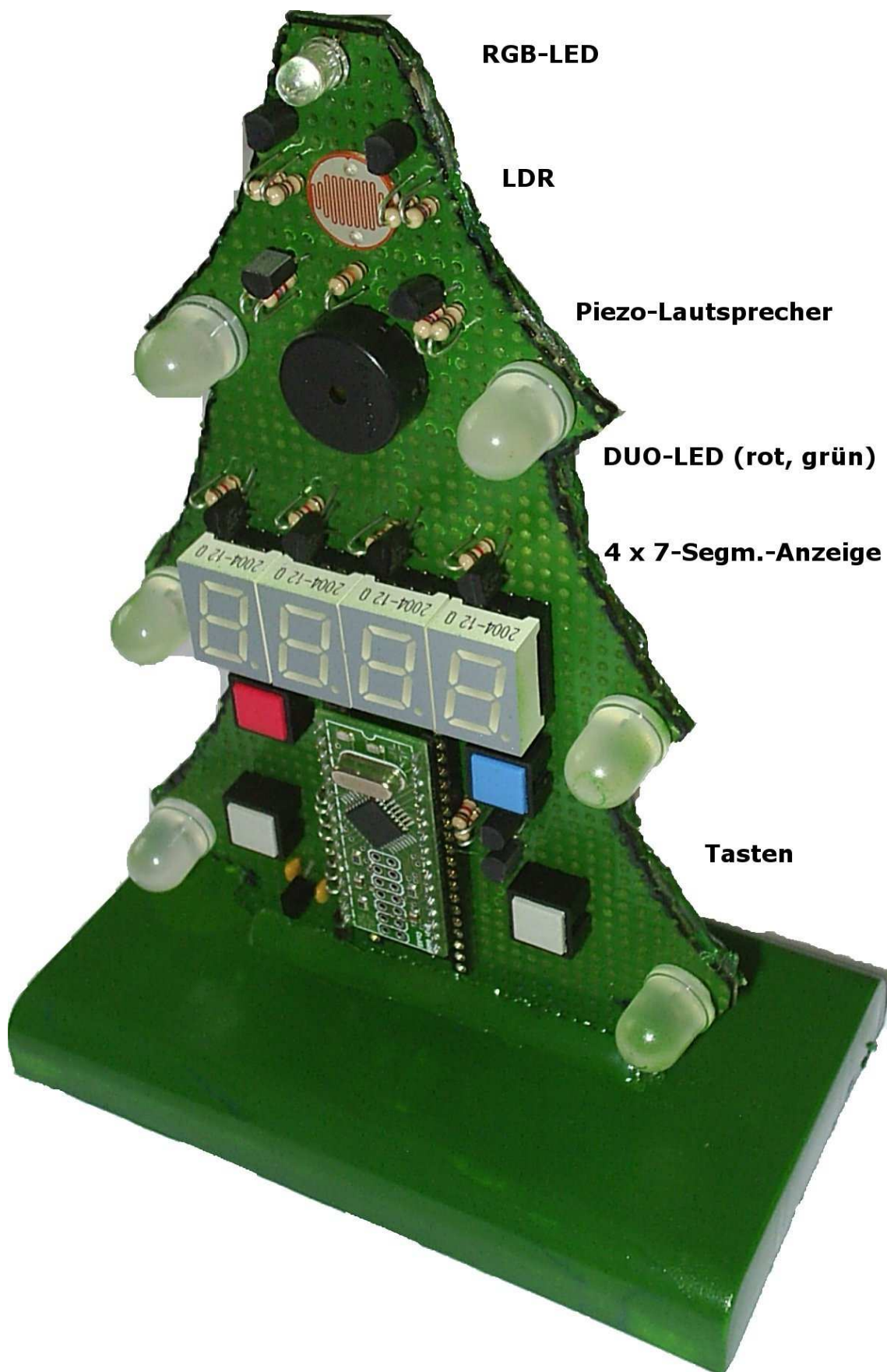


Abbildung A.5: MusicTree.

Anhang B

Melodien

Folgende Melodien sind standardmäßig gespeichert. Diese sind von 1 bis 64 durchnummeriert, wobei manche Melodien doppelt vorhanden sind. Die Zeile `song_str` enthält den String für die Melodie im Nokia-Format und die Zeile `song_bpm` die Geschwindigkeit, mit der die Melodie abgespielt wird.

1 Jingle Bells

```
const char* song_str_371 = "8a2 8a2 4a2 8a2 8a2 4a2 8a2 8c3 8.f2  
16g2 2a2 8#a2 8#a2 8.#a2 16#a2 8#a2 8a2 8.a2 16a2 8a2 8g2 8g2  
8a2 4g2 4c3";  
const unsigned char song_bpm_371 = 112;
```

2 Rudolf

```
const char* song_str_376 = "8g1 4a1 8g1 4e1 4c2 4a1 2.g1 8.g1  
16a1 8.g1 16a1 4g1 4c2 1b1 8f1 4g1 8f1 4d1 4b1 4a1 2.g1 8.g1  
16a1 8.g1 16a1 4g1 4d2 2.c2";  
const unsigned char song_bpm_376 = 160;
```

3 We Wish you a Merry Christmas v1.0

```
const char* song_str_380 = "4d1 4g1 8g1 8a1 8g1 8#f1 4e1 4c1 4e1  
4a1 8a1 8b1 8a1 8g1 4#f1 4d1 4d1 4b1 8b1 8c2 8b1 8a1 4g1 4e1 8d1  
8d1 4e1 4a1 4#f1 2g1";  
const unsigned char song_bpm_380 = 140;
```

4 Auld Lang Syne

```
const char* song_str_15 = "4g1 4.c2 8c2 4c2 4e2 4.d2 8c2 4d2 8e2 8d2  
4.c2 8c2 4e2 4g2 2.a2 4a2 4.g2 8e2 4e2 4c2 4.d2 8c2 4d2 8e2 8d2 4.c2  
8a1 4a1 4g1 2c2";  
const unsigned char song_bpm_15 = 100;
```

5 Axel F

```
const char* song_str_17 = "4g2 8.#a2 16g2 16- 16g2 8c3 8g2 8f2 4g2  
8.d3 16g2 16- 16g2 8#d3 8d3 8#a2 8g2 8d3 8g3 16g2 16f2 16- 16f2 8d2  
8a2 2g2";  
const unsigned char song_bpm_17 = 100;
```

6 Beatles: Yellow Submarine

```
const char* song_str_26 = "8#a1 8b1 2#c2 8#a1 8#g1 8#a1 2#f1 8#a1  
8#a1 8#g1 8#f1 2#d1 8#a1 8#a1 1#g1 4#c2 4#c2 4#c2 8#c2 8#d2 8#g1  
8#g1 8#g1 8#g1 2#g1 8#g1 8#g1 4#g1 2#g1 8#f1 8#f1 8#f1 8#f1 2#f1  
4#c2 4#c2 4#c2 8#c2 8#d2 8#g1 8#g1 8#g1 8#g1 2#g1 8#g1 8#g1 4#g1  
2#g1 8#f1 8#f1";  
const unsigned char song_bpm_26 = 125;
```

7 Beethoven: Für Elise

```
const char* song_str_28 = "8e2 8#d2 8e2 8#d2 8e2 8b1 8d2 8c2 4a1  
8- 8c1 8e1 8a1 4b1 8- 8e1 8#g1 8b1 4c2 8- 8e1 8e2 8#d2 8e2 8#d2 8e2  
8b1 8d2 8c2 4a1 8- 8c1 8e1 8a1 4b1 8- 8e1 8c2 8b1 4a1";  
const unsigned char song_bpm_28 = 125;
```

8 Beverly Hills Cop

```
const char* song_str_29 = "4f1 4- 4#g1 8- 4f1 8f1 4#a1 32- 4f1 4#d1  
4f1 4- 4c2 8- 4f1 8f1 4#c2 32- 4c2 4#g1 4f1 4c2 4f2 8f1 4#d1 8#d1  
32- 4c1 4g1 1f1";  
const unsigned char song_bpm_29 = 225;
```

9 Birdy Song

```
const char* song_str_32 = "16g2 16g2 16a2 16a2 16e2 16e2 8g2 16g2  
16g2 16a2 16a2 16e2 16e2 8g2 16g2 16g2 16a2 16a2 16c3 16c3 8b2 8b2  
8a2 8g2 8f2 16f2 16f2 16g2 16g2 16d2 16d2 8f2 16f2 16f2 16g2 16g2  
16d2 16d2 8f2 16f2 16f2 16g2 16g2 16a2 16b2 8c3 8a2 8g2 8e2 4c2";  
const unsigned char song_bpm_32 = 100;
```

10 Black Bear

```
const char* song_str_33 = "4#d2 4#d2 8g2 16#d2 8#a2 16g2 4#d2 4#d2  
8g2 16#d2 8#a2 16g2 4f2 8c2 16b1 4c2 8f2 16#d2 8d2 16#d2 8c2 16d2  
8#a1 16c2 8d2 16#a1 4#d2 4#d2 8g2 16#d2 8#a2 16g2 4#d2 4#d2 8g2 16#d2  
8#a2 16g2 4f2 4f2 4f2 8g2 16f2 4#d2 4g2 2#d2";  
const unsigned char song_bpm_33 = 180;
```

11 Bolero

```
const char* song_str_37 = "4c3 8c3 16b2 16c3 16d3 16c3 16b2 16a2 8c3  
16c3 16a2 4c3 8c3 16b2 16c3 16a2 16g2 16e2 16f2 2g2 16g2 16f2 16e2  
16d2 16e2 16f2 16g2 16a2 4g2 4g2 16g2 16a2 16b2 16a2 16g2 16f2 16e2  
16d2 16e2 16d2 8c2 8c2 16c2 16d2 8e2 8f2 4d2 2g2";  
const unsigned char song_bpm_37 = 80;
```

12 Colonel Bogey

```
const char* song_str_45 = "8g2 8e2 4- 8- 8e2 8f2 8g2 4e3 4e3 2c3 8g2  
8e2 2- 8e2 8f2 8e2 4g2 4g2 2f2 8f2 8d2 4- 8- 8d2 8e2 8f2 8g2 8e2 4-  
8- 8e2 8#f2 8e2 8d2 8g2 8- 8e2 8#f2 8d2 8- 8a2 8.g2 16#f2 8g2 8a2 8g2
```

```
8f2 8e2 8d2 8c2";  
const unsigned char song_bpm_45 = 140;
```

13 Verve

```
const char* song_str_357 = "4E1 4G1 4E1 4F1 4D1 2F1 4#A1 4F1  
2#A1 4A1 4F1 2A1 4- 4E1 4G1 4E1 4F1 4D1 2F1 4#A1 4F1 2#A1 4A1  
4F1 2A1 4-";  
const unsigned char song_bpm_357 = 180;
```

14 Dallas

```
const char* song_str_51 = "8e2 4a2 8e2 4e3 8a2 4#c3 8b2 8#c3 4a2 4e2  
4a2 4#f3 4e3 8#c3 8d3 2e3 8- 8e2 4a2 4#f3 4e3 8#c3 8d3 4e3 8b2 8#c3  
4a2 4e2 4a2 8#c3 8d3 4b2 8a2 2a2";  
const unsigned char song_bpm_51 = 125;
```

15 Eiffel 65: Blue v1

```
const char* song_str_67 = "4a1 4#a1 8g1 8#a1 8c2 8f1 8a1 4#a1 8g1 8#a1  
8d2 4#d2 8d2 8c2 4#a1 8g1 8#a1 8c2 8f1 8a1 4#a1 8g1 8#a1 8d2 4#d2 8d2  
8c2 4#a1 8g1 8#a1 8c2 8f1 8a1 4#a1 8g1 8#a1 8d2 4#d2 8d2 8c2 4#a1 8g1  
8#a1 8a1 8f1 8f1 2g1";  
const unsigned char song_bpm_67 = 140;
```

16 SuperMan

```
const char* song_str_352 = "8G1 8G1 8G1 4C2 8C2 2G2 8- 8G2 8A2  
16G2 8F2 1G2 8- 8G1 8G1 8G1 4C2 8C2 2G2 8- 8G2 8A2 16G2 8F2 8A2  
2G.2 4- 8C2 8C2 8C2 2B.2 4G2 8C2 8C2 8C2 2B.2 4G2 8C2 8C2 8C2  
8B2 8A2 8B2 2C3 8C2 8C2 8C2 8C2 8C2 2C2";  
const unsigned char song_bpm_352 = 180;
```

17 Entertainer

```
const char* song_str_74 = "8d2 8#d2 8e2 4c3 8e2 4c3 8e2 2c3 8c3 8d3  
8#d3 8e3 8c3 8d3 4e3 8b2 4d3 2c3 4- 8d2 8#d2 8e2 4c3 8e2 4c3 8e2 2c3  
8- 8a2 8g2 8#f2 8a2 8c3 4e3 8d3 8c3 8a2 2d3";  
const unsigned char song_bpm_74 = 140;
```

18 Europe: The Final Countdown

```
const char* song_str_75 = "4- 8- 16b2 16a2 4b2 4e2 4- 8- 16c3 16b2 8c3  
8b2 4a2 4- 8- 16c3 16b2 4c3 4e2 4- 8- 16a2 16g2 8a2 8g2 8#f2 8a2 4g2  
8- 16#f2 16g2 4a2 8- 16g2 16a2 8b2 8a2 8g2 8#f2 4e2 4c3 2b2 4- 16b2  
16c3 16b2 16a2 1b2";  
const unsigned char song_bpm_75 = 125;
```

19 Flintstones

```
const char* song_str_81 = "4#g2 4#c2 8- 4#c3 8#a2 4#g2 4#c2 8- 4#g2  
8#f2 8f2 8f2 8#f2 8#g2 4#c2 4#d2 2f2 2- 4#g2 4#c2 8- 4#c3 8#a2 4#g2
```

```
4#c2 8- 4#g2 8#f2 8f2 8f2 8#f2 8#g2 4#c2 4#d2 2#c2";  
const unsigned char song_bpm_81 = 200;
```

20 Greensleaves

```
const char* song_str_95 = "4g2 2#a2 4c3 4.d3 8#d3 4d3 2c3 4a2 4.f2 8g2  
4a2 2#a2 4g2 4.g2 8f2 4g2 2a2 4f2 2d2 4g2 2#a2 4c3 4.d3 8e3 4d3 2c3 4a2  
4.f2 8g2 4a2 4.#a2 8a2 4g2 4.#f2 8e2 4#f2 2g2 2.g2";  
const unsigned char song_bpm_95 = 140;
```

21 Imperial

```
const char* song_str_105 = "4a1 4a1 4a1 4f1 16c2 4a1 4f1 16c2 2a1 4e2  
4e2 4e2 4f2 16c2 4#g1 4f1 16c2 2a1 4a2 4a1 16a1 4a2 4#g2 16g2 16#f2  
16f2 4#f2 8#a1 4#d2 4d2 16#c2 16c2 16b1 4c2 8f1 4#g1 4f1 16#g1 4c2 4a1  
16c2 2e2 4a2 4a1 16a1 4a2 4#g2 16g2 16#f2 16f2";  
const unsigned char song_bpm_105 = 112;
```

22 Indiana Jones v2

```
const char* song_str_107 = "4e1 16f1 8g1 2c2 4d1 16e1 1f1 4g1 16a1 8b1  
2f2 4a1 16b1 4c2 4d2 4e2 4e1 16f1 8g1 1c2 4d2 16e2 2f2 4g1 16g1 4e2 4d2  
16g1 4e2 4d2 16g1 4f2 4e2 16d2 2c2";  
const unsigned char song_bpm_107 = 125;
```

23 Knight Rider

```
const char* song_str_117 = "16e2 32f2 32e2 8b2 16e3 32f3 32e3 8b2 16e2  
32f2 32e2 16b2 16e3 4d3 8- 4- 16e2 32f2 32e2 8b2 16e3 32f3 32e3 8b2  
16e2 32f2 32e2 16b2 16e3 4f3 4-";  
const unsigned char song_bpm_117 = 63;
```

24 La Cucaracha

```
const char* song_str_120 = "4c1 4c1 4c1 8f1 8a1 4- 4c1 4c1 4c1 8f1 8a1  
4- 4f1 4f1 4e1 4e1 4d1 4d1 8c1 4- 4- 4c1 4c1 4c1 8e1 8g1 4- 4c1 4c1 4c1  
8e1 8g1 4- 4c2 4d2 4c2 4#a1 4a1 4g1 8f1";  
const unsigned char song_bpm_120 = 225;
```

25 Looney Tunes

```
const char* song_str_125 = "16f3 16f3 8e3 8d3 8c3 4a2 8- 8c3 16f3 16f3  
8e3 8d3 8#d3 4e3 8- 8e3 8e3 8c3 8d3 8c3 16e3 16e3 8c3 8d3 8a2 8c3 8g2  
8#a2 8a2 8f2";  
const unsigned char song_bpm_125 = 160;
```

26 Macarena

```
const char* song_str_128 = "4f2 8f2 8f2 4f2 8f2 8f2 8f2 8f2 8f2 8f2 8f2  
8a2 8c2 8c2 4f2 8f2 8f2 4f2 8f2 8f2 8f2 8f2 8f2 8f2 8d2 8c2 4- 4f2 8f2  
8f2 4f2 8f2 8f2 8f2 8f2 8f2 8f2 8f2 8a2 4- 2.c3 4a2 8c3 8a2 8f2 4- 2-";  
const unsigned char song_bpm_128 = 180;
```

27 Michael Jackson: Smooth criminal

```
const char* song_str_143 = "8a1 16a1 16a1 16g1 16a1 8b1 8b1 8- 16a1  
16b1 8c2 8c2 8- 16b1 16c2 8b1 4g1 8a1 8- 8a1 16a1 16a1 16g1 16a1 8b1  
8b1 8- 16a1 16b1 8c2 8c2 8- 16b1 16c2 8b1 4g1";  
const unsigned char song_bpm_143 = 125;
```

28 Minuet

```
const char* song_str_144 = "4d2 8g1 8a1 8b1 8c2 4d2 4g1 4g1 4e2 8c2 8d2  
8e2 8f2 4g2 4g1 4g1 4c2 8d2 8c2 8b1 8a1 4b1 8c2 8b1 8a1 8g1 4#f1 8g1  
8a1 8b1 8g1 4b1 4a1";  
const unsigned char song_bpm_144 = 125;
```

29 Mission Impossible

```
const char* song_str_145 = "16g2 8- 16g2 8- 16f2 16- 16#f2 16- 16g2 8-  
16g2 8- 16#a2 16- 16c3 16- 16g2 8- 16g2 8- 16f2 16- 16#f2 16- 16g2 8-  
16g2 8- 16#a2 16- 16c3 16- 16#a2 16g2 2d2 32- 16#a2 16g2 2#c2 32- 16#a2  
16g2 2c2 16- 16#a1 16c2";  
const unsigned char song_bpm_145 = 100;
```

30 Monty Python v1

```
const char* song_str_146 = "8- 8c3 8a2 8- 8a2 8a2 8#g2 8a2 8f3 8- 8c3  
8c3 8- 8a2 8#a2 8- 8#a2 8#a2 8- 8c3 2d3 8- 8#a2 8g2 8- 8g2 8g2 8#f2  
8g2 8e3 8- 8d3 8d3 8- 8#a2 8a2 8- 8a2 8a2 8- 8#a2 2c3";  
const unsigned char song_bpm_146 = 140;
```

31 Mozart

```
const char* song_str_149 = "16#f1 16e1 16#d1 16e1 4g1 16a1 16g1 16#f1  
16g1 4b1 16c2 16b1 16#a1 16b1 16#f2 16e2 16#d2 16e2 16#f2 16e2 16#d2  
16e2 4g2 8e2 8g2 32d2 32e2 16#f2 8e2 8d2 8e2 32d2 32e2 16#f2 8e2 8d2  
8e2 32d2 32e2 16#f2 8e2 8d2 8#c2 4b1";  
const unsigned char song_bpm_149 = 63;
```

32 Muppet Show

```
const char* song_str_152 = "4c3 4c3 4a2 4b2 8a2 4b2 4g2 4- 4c3 4c3 4a2  
8b2 8a2 8- 4g2 4- 4e2 4e2 4g2 4f2 8e2 4f2 8c3 8c2 8d2 4e2 8e2 8e2 8-  
8e2 4g2";  
const unsigned char song_bpm_152 = 225;
```

33 Nacht

```
const char* song_str_153 = "2#g1 8#d1 2#g1 8#d1 8#g1 8#d1 8#g1 8c2 2#d2  
2#c2 8#a1 2#c2 8#a1 8#c2 8#a1 8g1 8#a1 2#d1 4#g1 2#g1 8c2 8#a1 8#g1  
32#a1 32#g1 32#a1 32#g1 8g1 2g1 8#a1 8#c2 8g1 8#a1 8#g1 2#g1 8c2 8#a1  
8#g1 32#a1 32#g1 32#a1 32#g1 8g1 2g1 8#a1 8#c2 8g1 8#g1 8#g1 16#g1";  
const unsigned char song_bpm_153 = 125;
```

34 Phantom Of The Opera

```
const char* song_str_169 = "4e1 4a1 4e1 4g1 8f1 2f1 4d1 4g1 8d1 1e1 4e1  
4a1 4e1 4g1 8f1 2f1 4d1 4g1 8d1 1e1 4e1 4a1 4c2 4e2 8d2 2d2 4d2 4g2 8d2  
1e2 4e2 1a2 8g2 8f2 8e2 8d2 8c2 8b1 8a1 1#g1 4f1 4f1 8e1 1e1";  
const unsigned char song_bpm_169 = 100;
```

35 Pink Panther

```
const char* song_str_172 = "8#g1 2a1 8b1 2c2 8#g1 8a1 8b1 8c2 8f2 8e2  
8a1 8c2 8e2 2#d2 16d2 16c2 16a1 8g1 1a1 8#g1 2a1 8b1 2c2 8#g1 8a1 8b1  
8c2 8f2 8e2 8c2 8e2 8a2 1#g2 8#g1 2a1 8b1 2c2 16#g1 8a1 8b1 8c2 8f2  
8e2 8a1 8c2 8e2 2#d2 8d2 16c2 16a1";  
const unsigned char song_bpm_172 = 100;
```

36 Popcorn

```
const char* song_str_173 = "16g2 16f2 16g2 16d2 16#a1 16d2 8g1 16g2  
16f2 16g2 16d2 16#a1 16d2 8g1 16g2 16a2 16#a2 32g2 16#a2 32g2 16#a2  
16a2 32f2 16a2 32f2 16a2 16g2 16f2 16d2 16f2 8g2";  
const unsigned char song_bpm_173 = 70;
```

37 Robert Miles: Children

```
const char* song_str_183 = "8- 4f3 1- 4#g3 8g3 4#d3 1- 4#g3 8g3 4c3 1-  
4#g3 8g3 4#g2 1- 16f2 16g2 16#g2 16c3 4f3 1- 4#g3 8g3 4#d3 1- 16#c3  
16c3 4#c3 8c3 4#g2 2- 4g2 4#g2 8c3 4f2";  
const unsigned char song_bpm_183 = 112;
```

38 Rule B.

```
const char* song_str_184 = "4e2 8e2 8f2 4f2 8e2 8f2 16e2 8d2 16c2 2b1  
4g2 4f2 16e2 16c2 16f2 16d2 8g2 8f2 4e2 8d2 16c2 4c2";  
const unsigned char song_bpm_184 = 100;
```

39 Saints

```
const char* song_str_185 = "8c1 8e1 8f1 2g1 8c1 8e1 8f1 2g1 8c1 8e1  
8f1 4g1 4e1 4c1 4e1 2d1 8- 8e1 8d1 4c1 8c1 4e1 4g1 8g1 2f1 8a1 8a1  
4g1 4e1 4c1 4d1 2c1";  
const unsigned char song_bpm_185 = 125;
```

40 Simpsons v1

```
const char* song_str_193 = "4c2 4e2 4#f2 8a2 4.g2 4e2 4c2 8a1 8#f1  
8#f1 8#f1 2g1 4- 8#f1 8#f1 8#f1 8g1 4#a1 8c2 8c2 8c2 4c2";  
const unsigned char song_bpm_193 = 160;
```

41 Simpsons v2

```
const char* song_str_194 = "8c2 8a1 4- 8c2 8a2 4- 8c2 8a1 8c2 8a1 8c2  
8a2 4- 8- 8c2 8d2 8e2 8- 8e2 8f2 8g2 4- 8d2 8c2 4d2 8f2 4#a2 4a2 2c3";  
const unsigned char song_bpm_194 = 160;
```


42 Smoke on the Water

```
const char* song_str_196 = "4c1 4#d1 4.f1 4c1 4#d1 8#f1 4f1 4- 4c1  
4#d1 4.f1 4#d1 4c1 2- 8- 4c1 4#d1 4.f1 4c1 4#d1 8#f1 4f1 4- 4c1 4#d1  
4.f1 4#d1 4c1 4-";  
const unsigned char song_bpm_196 = 112;
```

43 Starwars

```
const char* song_str_203 = "8#c1 8#c1 16#c1 2#f1 2#c2 8b1 16#a1  
8#g1 2#f2 4#c2 8b1 16#a1 8#g1 2#f2 4#c2 8b1 16#a1 8b1 2#g1 8#c1  
8#c1 16#c1 2#f1 2#c2 8b1 16#a1 8#g1 2#f2 4#c2 8b1 16#a1 8#g1 2#f2  
4#c2 8b1 16#a1 8b1 2#g1 4#c1 16#c1 2#d1 8#c2 8b1 8#a1 8#g1 8#f1  
16#f1 8#g1 16#a1 4#g1";  
const unsigned char song_bpm_203 = 100;
```

44 Starwars: Imperial March

```
const char* song_str_204 = "4e1 4e1 4e1 8c1 16- 16g1 4e1 8c1 16-  
16g1 4e1 4- 4b1 4b1 4b1 8c2 16- 16g1 4#d1 8c1 16- 16g1 4e1 8-";  
const unsigned char song_bpm_204 = 100;
```

45 Surfari: Wipeout

```
const char* song_str_207 = "8g1 8#a1 8b1 8c2 8c2 8c2 8#a1 8g1 8g1  
8#a1 8b1 8c2 8c2 8c2 8#a1 8g1 8g1 8#d2 8e2 8f2 8f2 8f2 8#d2 8c2  
8c2 8#d2 8e2 8f2 8f2 8f2 8#d2 8c2 8c2 8#a1 8b1 8c2 8c2";  
const unsigned char song_bpm_207 = 160;
```

46 Susanna

```
const char* song_str_208 = "16f2 16g2 8a2 8c3 8c3 16d3 8c3 8a2  
8f2 16g2 8a2 8a2 8g2 8f2 4g2 8- 16f2 16g2 8a2 8c3 8c3 16d3 8c3  
8a2 8f2 16g2 8a2 8a2 8g2 8g2 4f2";  
const unsigned char song_bpm_208 = 100;
```

47 Venga Boys: Going to Ibiza

```
const char* song_str_223 = "2f1 8a1 8#a1 8c2 8f1 2#a1 16a2 16f2  
16a2 16f2 16a2 16f2 16a2 16f2 2f1 8a1 8#a1 8c2 8f1 2#a1 16a2 16f2  
16a2 16f2 16a2 16f2 16a2 16f2 4f1 8- 8f1 8f1 8f1 8g1 16g1 8g1  
16- 4f1";  
const unsigned char song_bpm_223 = 112;
```

48 JamesBondTheme

```
const char* song_str_255 = "8E1 16#F1 16#F1 8#F1 4#F1 8E1 8E1  
8E1 8G1 16G1 16G1 8G1 4G1 8#F1 8#F1 8#F1 8E1 16#F1 16#F1 8#F1  
4#F1 8E1 8E1 8E1 8G1 16G1 16G1 8G1 4G1 8#F1 8#F1 8#F1";  
const unsigned char song_bpm_255 = 125;
```

49 Zelda: Original Theme

```
const char* song_str_242 = "2#a1 2f1 4- 8#a1 8c2 8d2 8#d2 2f2 2-  
4f2 4f2 8#f2 8#g2 2#a2 2- 4#a2 8#a2 8- 8#g2 8#f2 4#g2 8#f2 2f2  
2- 2f2 4#d2 8#d2 8f2 2#f2 2- 4f2 4#d2 4#c2 8#c2 8#d2 2f2 2- 4#d2  
4#c2 4c2 8c2 8d2 2e2 2- 2g2 1f2";  
const unsigned char song_bpm_242 = 200;
```

50 2Unlimited: Get Ready

```
const char* song_str_250 = "8.g2 8g2 16- 8g2 8c2 8c2 8d2 8d2  
8.g2 8g2 16- 8g2 8#a2 8#a2 8c3 8c3 8.g2 8g2 16- 8g2 8c2 8c2 8d2  
8d2 8.g2 8g2 16- 8g2 8#a2 8#a2 8c3 8d3";  
const unsigned char song_bpm_250 = 125;
```

51 TakeOnMe

```
const char* song_str_256 = "8#F2 8#F2 8#F2 8D2 8- 8B1 8- 8E2 8-  
8E2 8- 8E2 8#G2 8#G2 8A2 8B2 8A2 8A2 8A2 8E2 8- 8D2 8- 8#F2 8-  
8#F2 8- 8#F2 8E2 8E2 8#F2 8E2 4F#2 8#F2 8#F2 8D2 8- 8B1 8- 8E2 8-  
8E2 8- 8E2 8#G2 8#G2 8A2 8B2 8A2 8A2 8A2 8E2 8- 8D2 8- 8#F2 8-  
8#F2 8- 8#F2 8E2 8E2";  
const unsigned char song_bpm_256 = 160;
```

52 Abba: Money Money Money v1

```
const char* song_str_1 = "8e3 8e3 8e3 8e3 8e3 8e3 16e2 16a2 16c3 16e3  
8#d3 8#d3 8#d3 8#d3 8#d3 8#d3 16f2 16a2 16c3 16#d3 4d3 8c3 8a2 8c3 4c3  
2a2 32a2 32c3 32e3 8a3";  
const unsigned char song_bpm_1 = 112;
```

53 Aha: Take on me

```
const char* song_str_6 = "8- 16#a1 16#a1 16#a1 8#f1 8#d1 8#g1 8#g1  
16#g1 16c2 16c2 16#c2 16#d2 16#c2 16#c2 16#c2 8#g1 8#f1 8#a1 8#a1  
16#a1 16#g1 16#g1 16#a1 16#g1 16#a1 16#a1 16#a1 8#f1 8#d1 8#g1 8#g1  
16#g1 16c2 16c2 16#c2 16#d2 16#c2 16#c2 16#c2 8#g1 8#f1 8#a1 8#a1";  
const unsigned char song_bpm_6 = 100;
```

54 Aqua: Barbie Girl

```
const char* song_str_11 = "8#g2 8e2 8#g2 8#c3 4a2 4- 8#f2 8#d2 8#f2  
8b2 4#g2 8#f2 8e2 4- 8e2 8#c2 4#f2 4#c2 4- 8#f2 8e2 4#g2 4#f2";  
const unsigned char song_bpm_11 = 125;
```

55 AdamsFamily

```
const char* song_str_257 = "8C2 8F2 8A2 8F2 8C2 4B1 2G2 8F2 4E2 8G2  
4E2 8E1 4A1 2F2 8C2 4F2 8A2 4F2 8C2 4B1 2G2 8F2 4E2 8C2 4D2 8E2 1F2  
8C2 8D2 8E2 8F2 1- 8D2 8E2 8#F2 8G2 1- 8D2 8E2 8#F2 8G2 4- 8D2 8E2  
8#F2 8G2 4- 8C2 8D2 8E2";  
const unsigned char song_bpm_257 = 160;
```

56 Agadoo

```
const char* song_str_258 = "8B2 8#G2 4E2 8E2 8E2 4E2 8E2 8E2 8E2 8E2  
8#D2 8E2 4#F2 8A2 8#F2 4#D2 8#D2 8#D2 4D#2 8#D2 8#D2 8#D2 8#D2 8#C2  
8#D2 4E2";  
const unsigned char song_bpm_258 = 125;
```

57 DualingBanjos

```
const char* song_str_285 = "8#C2 8D2 4E2 4#C2 4D2 4B1 4#C2 4#D1 4B1  
4- 16#C3 16- 16D3 16- 8E3 8- 8#C3 8- 8D3 8- 8B2 8- 8#C3 8- 8A2 8- 4B2  
4- 4A1 4A1 4B1 4#C2 4A1 4#C2 4B1 4- 8A2 8- 8A2 8- 8B2 8- 8#C3 8- 8A2  
8- 8#C3 8- 8B2";  
const unsigned char song_bpm_285 = 200;
```

58 TheGoodTheBadAndTheUgly

```
const char* song_str_299 = "32C2 32F2 32C2 32F2 4C2 8#G1 8#A1 4F1 8-  
32C2 32F2 32C2 32F2 42C2 4#G1 4#A1 4#D2 8- 32C2 32- 32#D2 4F2 16#G2  
32- 16G2 32- 32F2 32- 4#D2 8- 32C2 32F2 32C2 32F2 4C2 8#A1 4F1";  
const unsigned char song_bpm_299 = 125;
```

59 Itchy

```
const char* song_str_309 = "8C2 8A1 4- 8C2 8A2 4- 8C2 8A1 8C2 8A1 8C2  
8A2 4- 8- 8C2 8D2 8E2 8- 8E2 8F2 8G2 4- 8D2 8C2 4D2 8F2 4#A2 4A2 2C3";  
const unsigned char song_bpm_309 = 160;
```

60 Wannabe

```
const char* song_str_346 = "16G1 16G1 16G1 16G1 8G1 8A1 8G1 8E1 8-  
16C1 16D1 16C1 8D1 8D1 8C1 4E1 4- 8G1 8G1 8G1 8A1 8G1 8E1 8- 4C2  
8C2 8B1 8G1 8A1 16B1 16A1 4G1";  
const unsigned char song_bpm_346 = 125;
```

61 Birdy Song

```
const char* song_str_32 = "16g2 16g2 16a2 16a2 16e2 16e2 8g2 16g2  
16g2 16a2 16a2 16e2 16e2 8g2 16g2 16g2 16a2 16a2 16c3 16c3 8b2 8b2  
8a2 8g2 8f2 16f2 16f2 16g2 16g2 16d2 16d2 8f2 16f2 16f2 16g2 16g2  
16d2 16d2 8f2 16f2 16f2 16g2 16g2 16a2 16b2 8c3 8a2 8g2 8e2 4c2";  
const unsigned char song_bpm_32 = 100;
```

62 Entertainer

```
const char* song_str_74 = "8d2 8#d2 8e2 4c3 8e2 4c3 8e2 2c3 8c3 8d3  
8#d3 8e3 8c3 8d3 4e3 8b2 4d3 2c3 4- 8d2 8#d2 8e2 4c3 8e2 4c3 8e2 2c3  
8- 8a2 8g2 8#f2 8a2 8c3 4e3 8d3 8c3 8a2 2d3";  
const unsigned char song_bpm_74 = 140;
```

63 Indiana Jones v2

```
const char* song_str_107 = "4e1 16f1 8g1 2c2 4d1 16e1 1f1 4g1 16a1 8b1
```

```
2f2 4a1 16b1 4c2 4d2 4e2 4e1 16f1 8g1 1c2 4d2 16e2 2f2 4g1 16g1 4e2 4d2
16g1 4e2 4d2 16g1 4f2 4e2 16d2 2c2";
const unsigned char song_bpm_107 = 125;
```

64 Looney Tunes

```
const char* song_str_125 = "16f3 16f3 8e3 8d3 8c3 4a2 8- 8c3 16f3 16f3
8e3 8d3 8#d3 4e3 8- 8e3 8e3 8c3 8d3 8c3 16e3 16e3 8c3 8d3 8a2 8c3 8g2
8#a2 8a2 8f2";
const unsigned char song_bpm_125 = 160;
```