

ArduinoとExcelで作る 電子オルゴール

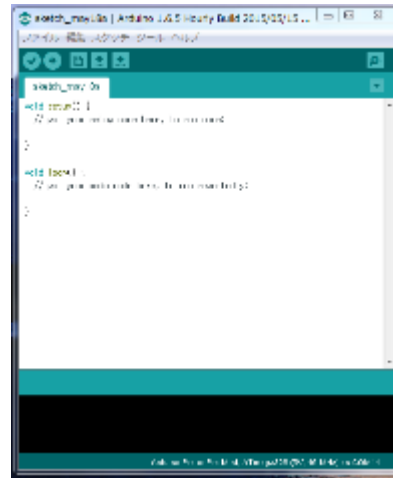


熊本高等専門学校
2015-08-13 update
森下功啓

電子オルゴールに必要なもの



MacやPC(パソコン)



Arduino IDE

(上の図はWindows用)



Arduino
(アルデュイーノ)



<http://akizukidenshi.com/catalog/g/gC-07605/>

通信ケーブル



圧電スピーカー

音を出す原理

- 音は、空気の振動です
- 圧電スピーカーは電圧をかけると変形する素子です
- 1は250 Hzで電圧を変化させると250 Hzの音が出ます
 - ヘルツHzは、1秒間に繰り返される現象の回数
- Arduinoには、そのための機能が備わっています

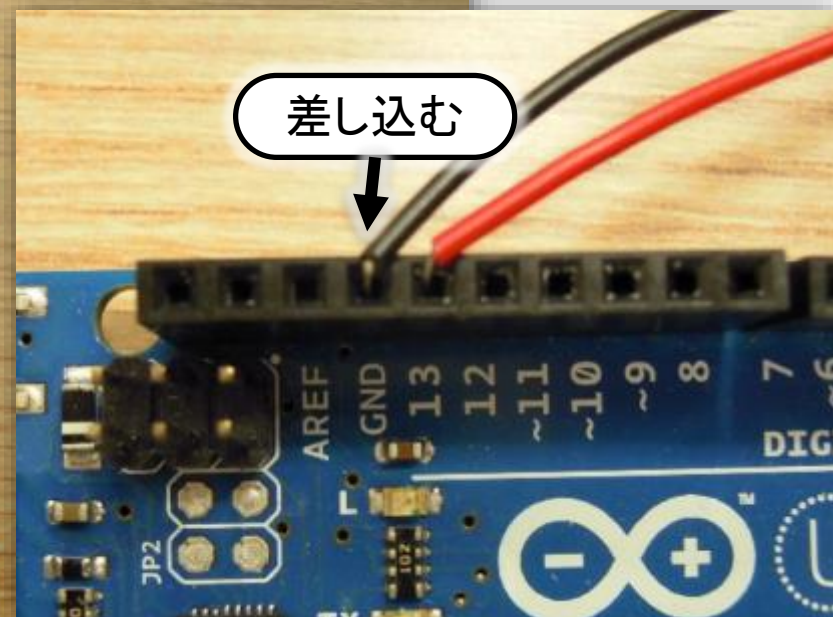
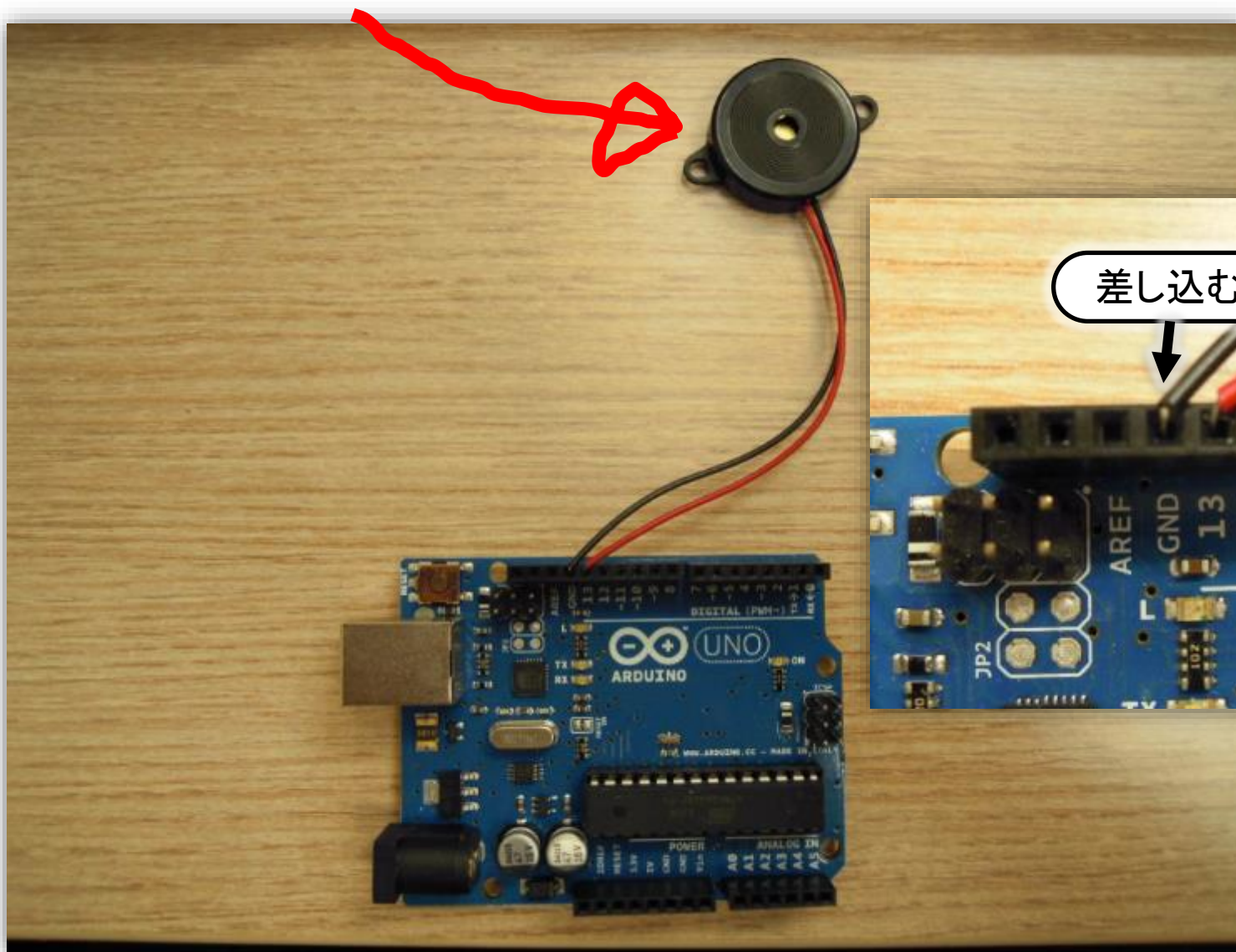


電子オルゴールで利用する主な関数

- `tone(ポート番号, 周波数)`
 - 特定の周波数の矩形信号を出力する
- `delay(待ち時間[ms])`
 - 引数で指定した時間だけ待つ

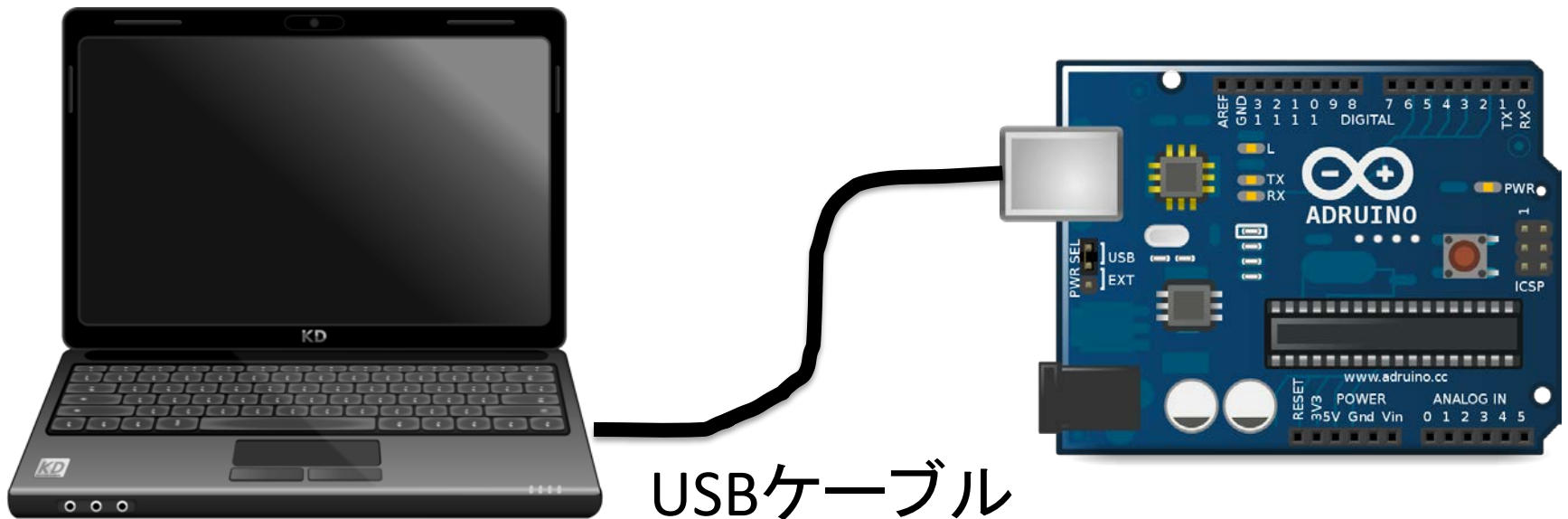
オルゴール の作り方

圧電スピーカを接続する



ArduinoとPCをUSBで接続する

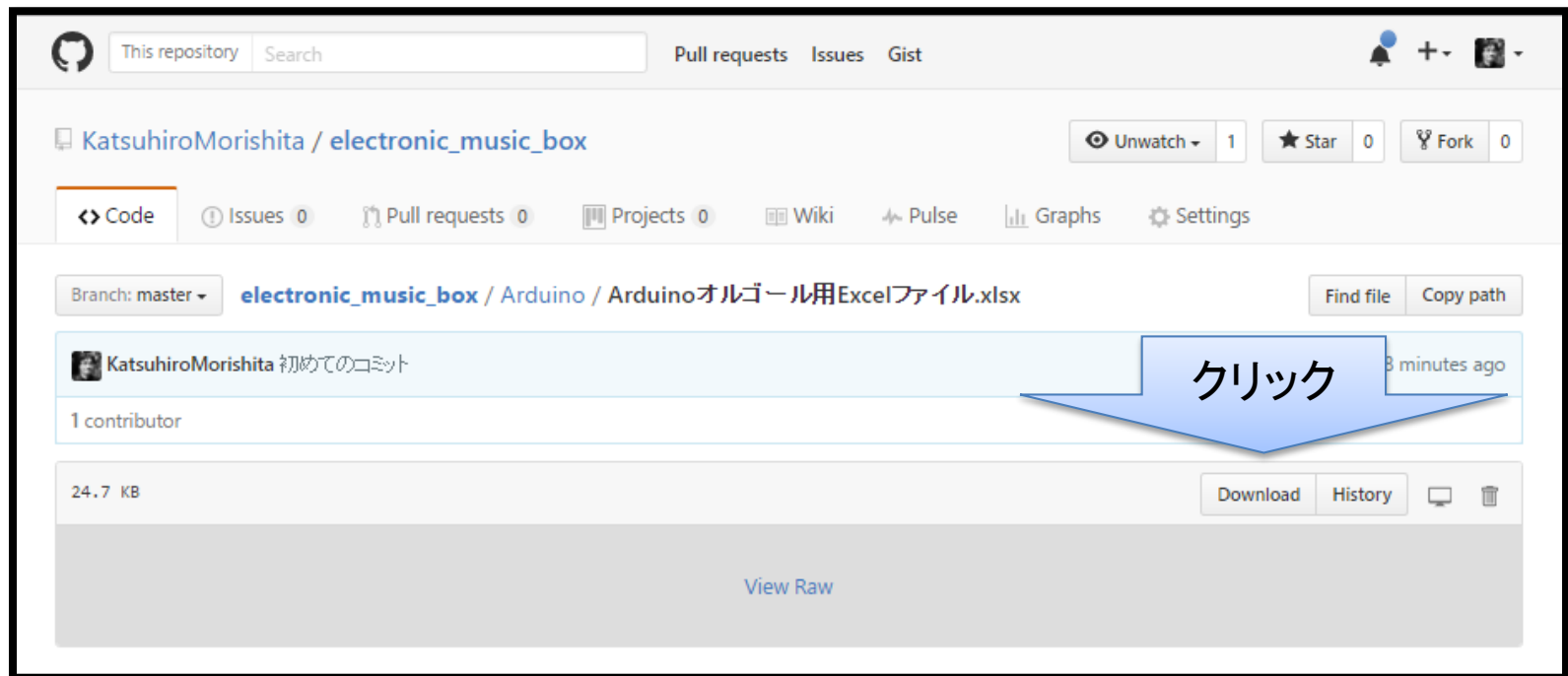
- 電源をArduinoに供給しつつ、プログラムを書き込んだり、Arduinoとシリアル通信するためにUSBケーブルで接続します



スケッチ作成補助ツールを入手

下記リンクよりダウンロードできます

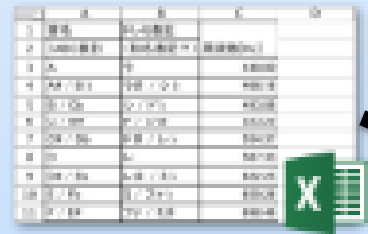
https://github.com/KatsuhikoMorishita/electronic_music_box/blob/master/Arduino/Arduino%E3%82%AA%E3%83%AB%E3%82%B4%E3%83%BC%E3%83%AB%E7%94%A8Excel%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB.xlsx



スケッチ作成補助ツールを起動

アイコンをダブルクリックします

起動



Arduinoオルゴール用Excelファイル.xlsx

左側には何も入力しないで下さい										右側の白欄に入力してください									
番号	音名	音高	音量	音長	音速	音色	音調	音種	音域	番号	音名	音高	音量	音長	音速	音色	音調	音種	音域
1	ド	261.6	100	1.0	1.0	1.0	1.0	1.0	1.0	1	ド	261.6	100	1.0	1.0	1.0	1.0	1.0	1.0
2	レ	196.0	100	1.0	1.0	1.0	1.0	1.0	1.0	2	レ	196.0	100	1.0	1.0	1.0	1.0	1.0	1.0
3	ミ	329.6	100	1.0	1.0	1.0	1.0	1.0	1.0	3	ミ	329.6	100	1.0	1.0	1.0	1.0	1.0	1.0
4	ファ	391.9	100	1.0	1.0	1.0	1.0	1.0	1.0	4	ファ	391.9	100	1.0	1.0	1.0	1.0	1.0	1.0
5	ソ	440.0	100	1.0	1.0	1.0	1.0	1.0	1.0	5	ソ	440.0	100	1.0	1.0	1.0	1.0	1.0	1.0
6	ラ	495.0	100	1.0	1.0	1.0	1.0	1.0	1.0	6	ラ	495.0	100	1.0	1.0	1.0	1.0	1.0	1.0
7	シ	587.3	100	1.0	1.0	1.0	1.0	1.0	1.0	7	シ	587.3	100	1.0	1.0	1.0	1.0	1.0	1.0
8	ハ	659.3	100	1.0	1.0	1.0	1.0	1.0	1.0	8	ハ	659.3	100	1.0	1.0	1.0	1.0	1.0	1.0
9	ニ	734.4	100	1.0	1.0	1.0	1.0	1.0	1.0	9	ニ	734.4	100	1.0	1.0	1.0	1.0	1.0	1.0
10	ホ	880.0	100	1.0	1.0	1.0	1.0	1.0	1.0	10	ホ	880.0	100	1.0	1.0	1.0	1.0	1.0	1.0
11	ヘ	989.6	100	1.0	1.0	1.0	1.0	1.0	1.0	11	ヘ	989.6	100	1.0	1.0	1.0	1.0	1.0	1.0
12	ト	1108.7	100	1.0	1.0	1.0	1.0	1.0	1.0	12	ト	1108.7	100	1.0	1.0	1.0	1.0	1.0	1.0
13	チ	1244.5	100	1.0	1.0	1.0	1.0	1.0	1.0	13	チ	1244.5	100	1.0	1.0	1.0	1.0	1.0	1.0
14	リ	1396.9	100	1.0	1.0	1.0	1.0	1.0	1.0	14	リ	1396.9	100	1.0	1.0	1.0	1.0	1.0	1.0
15	フ	1568.0	100	1.0	1.0	1.0	1.0	1.0	1.0	15	フ	1568.0	100	1.0	1.0	1.0	1.0	1.0	1.0
16	ビ	1760.0	100	1.0	1.0	1.0	1.0	1.0	1.0	16	ビ	1760.0	100	1.0	1.0	1.0	1.0	1.0	1.0
17	ブ	1960.0	100	1.0	1.0	1.0	1.0	1.0	1.0	17	ブ	1960.0	100	1.0	1.0	1.0	1.0	1.0	1.0
18	パ	2189.9	100	1.0	1.0	1.0	1.0	1.0	1.0	18	パ	2189.9	100	1.0	1.0	1.0	1.0	1.0	1.0
19	ド	2449.9	100	1.0	1.0	1.0	1.0	1.0	1.0	19	ド	2449.9	100	1.0	1.0	1.0	1.0	1.0	1.0
20	レ	2750.0	100	1.0	1.0	1.0	1.0	1.0	1.0	20	レ	2750.0	100	1.0	1.0	1.0	1.0	1.0	1.0
21	ミ	3099.6	100	1.0	1.0	1.0	1.0	1.0	1.0	21	ミ	3099.6	100	1.0	1.0	1.0	1.0	1.0	1.0
22	ファ	3498.8	100	1.0	1.0	1.0	1.0	1.0	1.0	22	ファ	3498.8	100	1.0	1.0	1.0	1.0	1.0	1.0
23	ソ	3956.5	100	1.0	1.0	1.0	1.0	1.0	1.0	23	ソ	3956.5	100	1.0	1.0	1.0	1.0	1.0	1.0
24	ラ	4479.9	100	1.0	1.0	1.0	1.0	1.0	1.0	24	ラ	4479.9	100	1.0	1.0	1.0	1.0	1.0	1.0
25	シ	5080.0	100	1.0	1.0	1.0	1.0	1.0	1.0	25	シ	5080.0	100	1.0	1.0	1.0	1.0	1.0	1.0
26	ハ	5779.1	100	1.0	1.0	1.0	1.0	1.0	1.0	26	ハ	5779.1	100	1.0	1.0	1.0	1.0	1.0	1.0
27	ニ	6564.4	100	1.0	1.0	1.0	1.0	1.0	1.0	27	ニ	6564.4	100	1.0	1.0	1.0	1.0	1.0	1.0
28	ホ	7454.4	100	1.0	1.0	1.0	1.0	1.0	1.0	28	ホ	7454.4	100	1.0	1.0	1.0	1.0	1.0	1.0
29	ヘ	8469.6	100	1.0	1.0	1.0	1.0	1.0	1.0	29	ヘ	8469.6	100	1.0	1.0	1.0	1.0	1.0	1.0
30	ト	9629.8	100	1.0	1.0	1.0	1.0	1.0	1.0	30	ト	9629.8	100	1.0	1.0	1.0	1.0	1.0	1.0
31	チ	10959.9	100	1.0	1.0	1.0	1.0	1.0	1.0	31	チ	10959.9	100	1.0	1.0	1.0	1.0	1.0	1.0
32	リ	12479.9	100	1.0	1.0	1.0	1.0	1.0	1.0	32	リ	12479.9	100	1.0	1.0	1.0	1.0	1.0	1.0
33	フ	14299.9	100	1.0	1.0	1.0	1.0	1.0	1.0	33	フ	14299.9	100	1.0	1.0	1.0	1.0	1.0	1.0
34	ビ	16439.9	100	1.0	1.0	1.0	1.0	1.0	1.0	34	ビ	16439.9	100	1.0	1.0	1.0	1.0	1.0	1.0
35	ブ	18999.9	100	1.0	1.0	1.0	1.0	1.0	1.0	35	ブ	18999.9	100	1.0	1.0	1.0	1.0	1.0	1.0
36	パ	21999.9	100	1.0	1.0	1.0	1.0	1.0	1.0	36	パ	21999.9	100	1.0	1.0	1.0	1.0	1.0	1.0
37	ド	25499.9	100	1.0	1.0	1.0	1.0	1.0	1.0	37	ド	25499.9	100	1.0	1.0	1.0	1.0	1.0	1.0
38	レ	29499.9	100	1.0	1.0	1.0	1.0	1.0	1.0	38	レ	29499.9	100	1.0	1.0	1.0	1.0	1.0	1.0
39	ミ	34099.9	100	1.0	1.0	1.0	1.0	1.0	1.0	39	ミ	34099.9	100	1.0	1.0	1.0	1.0	1.0	1.0
40	ファ	39399.9	100	1.0	1.0	1.0	1.0	1.0	1.0	40	ファ	39399.9	100	1.0	1.0	1.0	1.0	1.0	1.0

スケッチ作成補助ツールの説明

Arduinoオルゴール用Excelファイル.xlsx - Excel

左側には何も入力しないで下さい

右側の白欄に入力してください

入力欄はN, O, R, Sの4列

音名	ドレミ表記	周波数[Hz]	番号	音程クラス	音符	音符記号	周波数	音符	符丁	基本周波数	最終周波数	長さ	付点	係数	コード
(ABC表記)	(和名表記?)		1	-1	ラ	ラ-1	220.0	ラ-1		220.0	220.0	4			tone(loudspeaker_pin, 220);delay(base_time / 4);
A	ラ	440.00	2	-1	ラ#	ラ#-1	233.1	ラ#-1		246.9	246.9	4			tone(loudspeaker_pin, 247);delay(base_time / 4);
A# / Bb	ラ# / シb	466.16	3	-1	シ	シ-1	246.9	シ-1		261.6	261.6	4			tone(loudspeaker_pin, 262);delay(base_time / 4);
B / Cb	シ / シb	493.88	4	0	ド	ド0	261.6	ド0		293.7	293.7	4			tone(loudspeaker_pin, 294);delay(base_time / 4);
C / B#	ド / シ#	523.25	5	0	ド#	ド#0	277.2	ド#0		329.6	329.6	4			tone(loudspeaker_pin, 330);delay(base_time / 4);
C# / Db	ド# / レb	554.37	6	0	レ	レ0	293.7	レ0		349.2	349.2	4			tone(loudspeaker_pin, 349);delay(base_time / 4);
D	レ	587.33	7	0	レ#	レ#0	311.1	レ#0		392.0	392.0	4			tone(loudspeaker_pin, 392);delay(base_time / 4);
D# / Eb	レ# / ミb	622.25	8	0	ミ	ミ0	329.6	ミ0		440.0	440.0	4			tone(loudspeaker_pin, 440);delay(base_time / 4);
E / Fb	ミ / ファb	659.26	9	0	ファ	ファ0	349.2	ファ0	#	493.9	523.3	4			tone(loudspeaker_pin, 523);delay(base_time / 4);
F / E#	ファ / ミ#	698.46	10	0	ファ#	ファ#0	370.0	ファ#0	b	523.3	493.9	4			tone(loudspeaker_pin, 494);delay(base_time / 4);
F# / Gb	ファ# / ソb	739.99	11	0	ソ	ソ0	392.0	ソ0	#	493.9	523.3	4		* 3 / 2	tone(loudspeaker_pin, 523);delay(base_time * 3 / 2 / 4);
G	ソ	783.99	12	0	ソ#	ソ#0	415.3	ソ#0	b	523.3	493.9	4		* 3 / 2	tone(loudspeaker_pin, 494);delay(base_time * 3 / 2 / 4);
G# / Ab	ソ# / ラb	830.61	13	0	ラ	ラ0	440.0	ラ0		587.3	587.3	16			tone(loudspeaker_pin, 587);delay(base_time / 16);
Ab	ラ	880.00	14	0	ラ#	ラ#0	466.2	ラ#0		659.3	659.3	8			tone(loudspeaker_pin, 659);delay(base_time / 8);
参考サイト:	http://asrite.blog.fc2.com/blog-entry-229.html		15	0	シ	シ0	493.9	シ0		698.5	698.5	4			tone(loudspeaker_pin, 698);delay(base_time / 4);
音符の長さ:	http://www.animato-jp.net/~se/onpuhyou.html		16	1	ド	ド1	523.3	ド1		784.0	784.0	2			tone(loudspeaker_pin, 784);delay(base_time / 2);
			17	1	ド#	ド#1	554.4	ド#1		880.0	880.0	1			tone(loudspeaker_pin, 880);delay(base_time / 1);
			18	1	レ	レ1	587.3	レ1		987.8	987.8	16		* 3 / 2	tone(loudspeaker_pin, 988);delay(base_time * 3 / 2 / 16);
			19	1	レ#	レ#1	622.3	レ#1		1046.5	1046.5	8		* 3 / 2	tone(loudspeaker_pin, 1047);delay(base_time * 3 / 2 / 8);
			20	1	ミ	ミ1	659.3	ミ1		1174.7	1174.7	4		* 3 / 2	tone(loudspeaker_pin, 1175);delay(base_time * 3 / 2 / 4);
			21	1	ファ	ファ1	698.5	ファ1		1318.5	1318.5	2		* 3 / 2	tone(loudspeaker_pin, 1319);delay(base_time * 3 / 2 / 2);
			22	1	ファ#	ファ#1	740.0	ファ#1		1296.9	1296.9	2		* 3 / 2	tone(loudspeaker_pin, 1297);delay(base_time * 3 / 2 / 2);
			23	2	レ	レ2	1174.7	レ2		#N/A	#N/A	4			#N/A
			24	2	レ#	レ#2	1244.5	レ#2		440.0	440.0	4			tone(loudspeaker_pin, 440);delay(base_time / 4);
			25	2	ミ	ミ2	1318.5	ミ2		#N/A	#N/A	4			#N/A
			26	2	ファ	ファ2	1396.9	ファ2		440.0	440.0	4			tone(loudspeaker_pin, 440);delay(base_time / 4);
			27	2	ファ#	ファ#2	1480.0	ファ#2		#N/A	#N/A	4			#N/A
			28	2	ソ	ソ2	1568.0	ソ2		440.0	440.0	4			tone(loudspeaker_pin, 440);delay(base_time / 4);
			29	2	ソ#	ソ#2	1661.2	ソ#2		#N/A	#N/A	4			#N/A

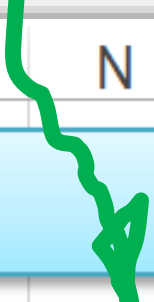
```
// sample code
const int loudspeaker_pin = 2;
const int base_time = 1000;

void setup()
{
  pinMode(loudspeaker_pin, OUTPUT);
}

void loop()
{
  tone(loudspeaker_pin, 220);delay(base_time);
  tone(loudspeaker_pin, 247);delay(base_time);
  tone(loudspeaker_pin, 262);delay(base_time);
  tone(loudspeaker_pin, 294);delay(base_time);
  tone(loudspeaker_pin, 330);delay(base_time);
  tone(loudspeaker_pin, 349);delay(base_time);
  tone(loudspeaker_pin, 392);delay(base_time);
  tone(loudspeaker_pin, 440);delay(base_time);
  tone(loudspeaker_pin, 523);delay(base_time);
  tone(loudspeaker_pin, 494);delay(base_time);
  tone(loudspeaker_pin, 523);delay(base_time * 3 / 2 / 4);
  tone(loudspeaker_pin, 494);delay(base_time * 3 / 2 / 4);
  tone(loudspeaker_pin, 587);delay(base_time / 16);
  tone(loudspeaker_pin, 659);delay(base_time / 8);
  tone(loudspeaker_pin, 698);delay(base_time / 4);
  tone(loudspeaker_pin, 784);delay(base_time / 2);
  tone(loudspeaker_pin, 880);delay(base_time / 1);
  tone(loudspeaker_pin, 988);delay(base_time * 3 / 2 / 16);
  tone(loudspeaker_pin, 1047);delay(base_time * 3 / 2 / 8);
  tone(loudspeaker_pin, 1175);delay(base_time * 3 / 2 / 4);
  tone(loudspeaker_pin, 1319);delay(base_time * 3 / 2 / 2);
  tone(loudspeaker_pin, 1297);delay(base_time * 3 / 2 / 2);
}
```

N列

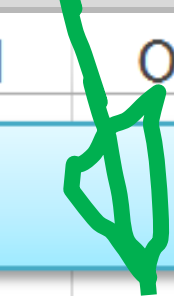
- ドレミファなどの音階を入力



M	N	O	P	Q	R	S
こちら側の白欄は						
	音符	符丁	基本周波数	最終周波数	長さ	付点
	ラ-1		220.0	220.0	4	
	シ-1		246.9	246.9		
	ド0		261.6	261.6		
	レ0		293.7	293.7	4	

0列

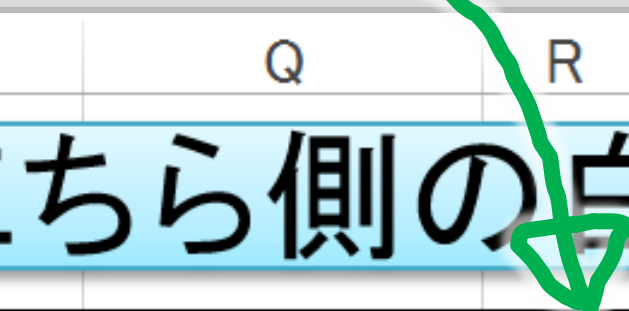
- #や♭を入力



M	N	O	P	Q	R	S
こちら側の白欄は						
	音符	符丁	基本周波数	最終周波数	長さ	付点
	ラ-1		220.0	220.0	4	
	シ-1		246.9	246.9		
	ド0		261.6	261.6		
	レ0		293.7	293.7	4	

R列

- 音の長さを入力
-4分音符なら"4"

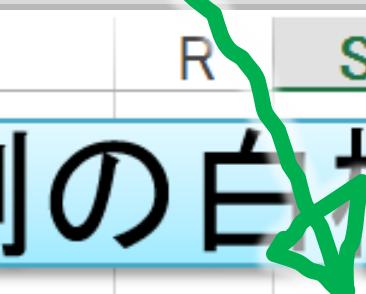


こちら側の白欄は

	N	O	P	Q	R	S
	音符	符丁	基本周波数	最終周波数	長さ	付点
	ラ-1		220.0	220.0	4	
	シ-1		246.9	246.9		
	ド0		261.6	261.6		
	レ0		293.7	293.7	4	

S列

- 付点(音の長さが1.5倍)を入力

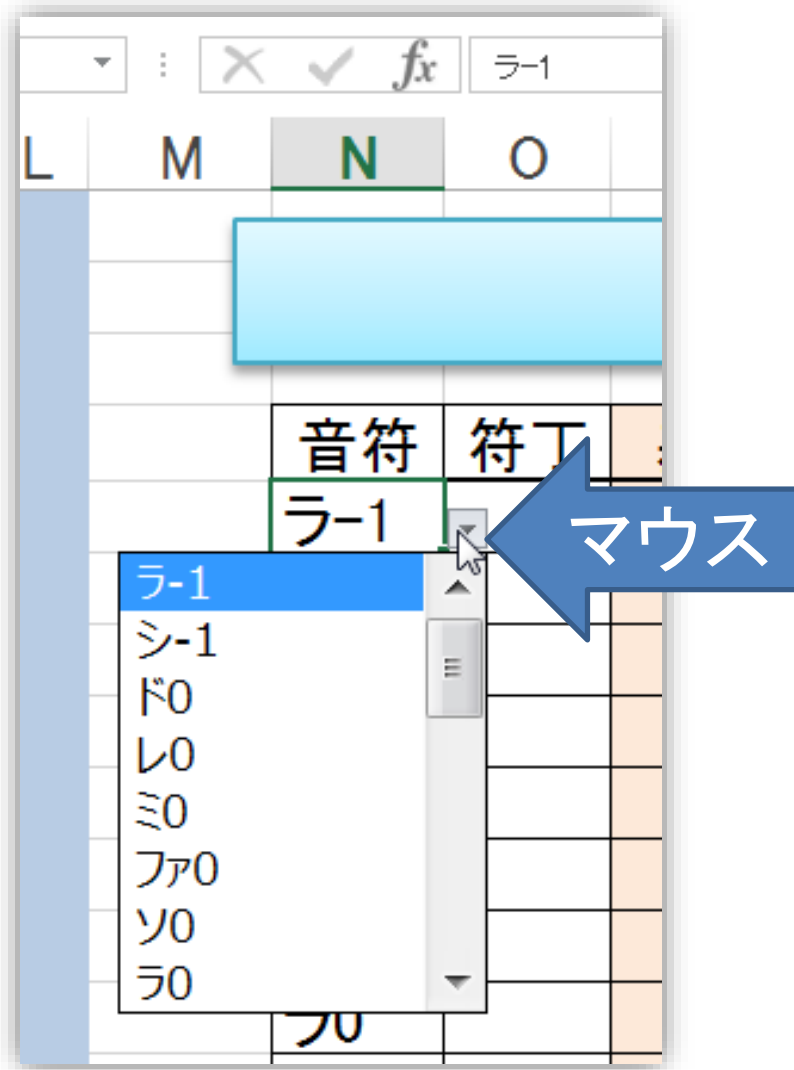


こちら側の白欄は

	N	O	P	Q	R	S	
	音符	符丁	基本周波数	最終周波数	長さ	付点	
	ラ-1		220.0	220.0	4		
	シ-1		246.9	246.9			
	ド0		261.6	261.6			
	レ0		293.7	293.7	4		

入力方法

1. セルを選択
2. 右側に出る▼をクリック
3. 選択肢が表示される
4. 選択肢をクリック



自動生成されるスケッチ

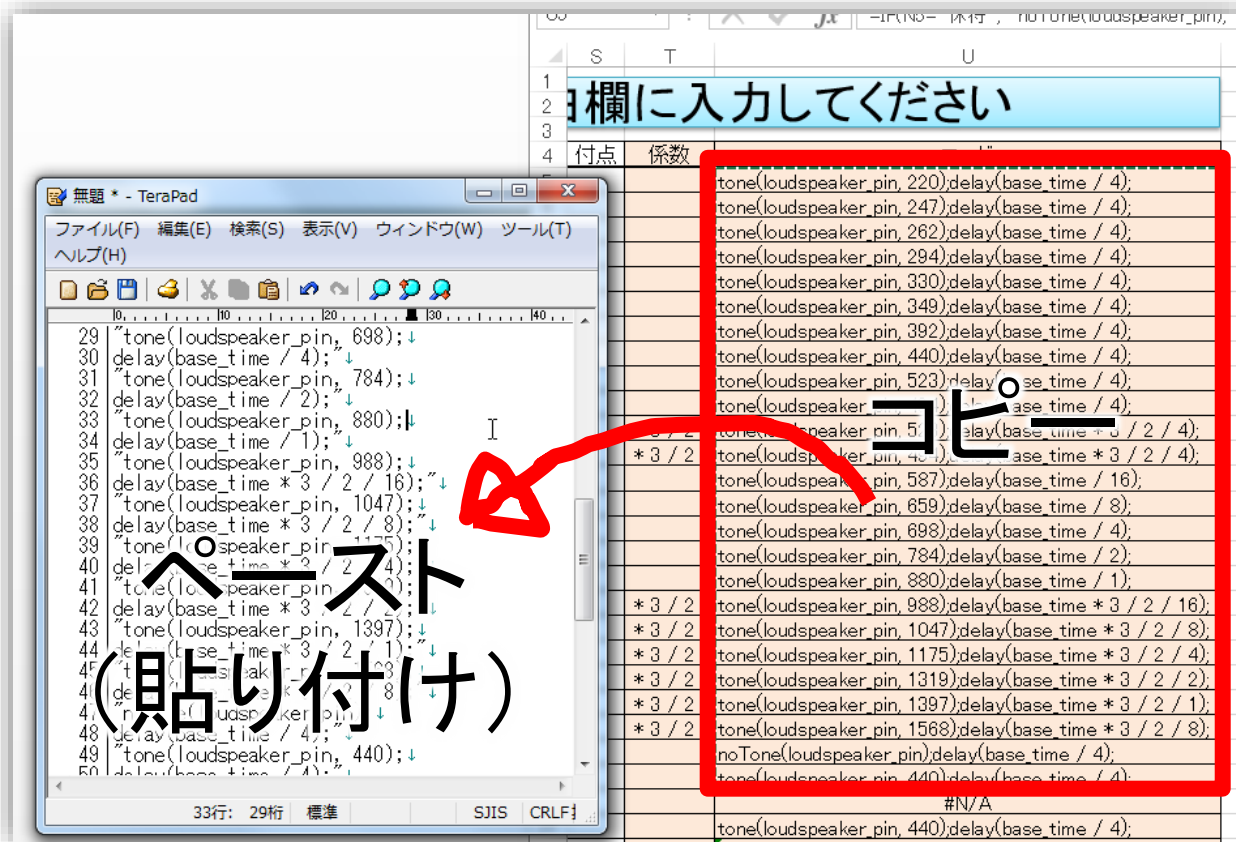
- U列に出力される
- Excelが演奏部分だけ自動生成する

T	U
	tone(loudspeaker_pin, 220);delay(base_time / 4);
	tone(loudspeaker_pin, 247);delay(base_time / 4);
	tone(loudspeaker_pin, 262);delay(base_time / 4);
	tone(loudspeaker_pin, 294);delay(base_time / 4);
	tone(loudspeaker_pin, 330);delay(base_time / 4);
	tone(loudspeaker_pin, 349);delay(base_time / 4);
	tone(loudspeaker_pin, 392);delay(base_time / 4);
	tone(loudspeaker_pin, 440);delay(base_time / 4);
	tone(loudspeaker_pin, 523);delay(base_time / 4);
	tone(loudspeaker_pin, 494);delay(base_time / 4);
* 3 / 2	tone(loudspeaker_pin, 523);delay(base_time * 3 / 2 / 4);
* 3 / 2	tone(loudspeaker_pin, 494);delay(base_time * 3 / 2 / 4);
	tone(loudspeaker_pin, 587);delay(base_time / 16);
	tone(loudspeaker_pin, 659);delay(base_time / 8);
	tone(loudspeaker_pin, 698);delay(base_time / 4);
	tone(loudspeaker_pin, 784);delay(base_time / 2);
	tone(loudspeaker_pin, 880);delay(base_time / 1);
* 3 / 2	tone(loudspeaker_pin, 988);delay(base_time * 3 / 2 / 16);
* 3 / 2	tone(loudspeaker_pin, 1047);delay(base_time * 3 / 2 / 8);
* 3 / 2	tone(loudspeaker_pin, 1175);delay(base_time * 3 / 2 / 4);
* 3 / 2	tone(loudspeaker_pin, 1319);delay(base_time * 3 / 2 / 2);
* 3 / 2	tone(loudspeaker_pin, 1397);delay(base_time * 3 / 2 / 1);
* 3 / 2	tone(loudspeaker_pin, 1568);delay(base_time * 3 / 2 / 8);
	tone(loudspeaker_pin, 440);delay(base_time / 4);
	#N/A

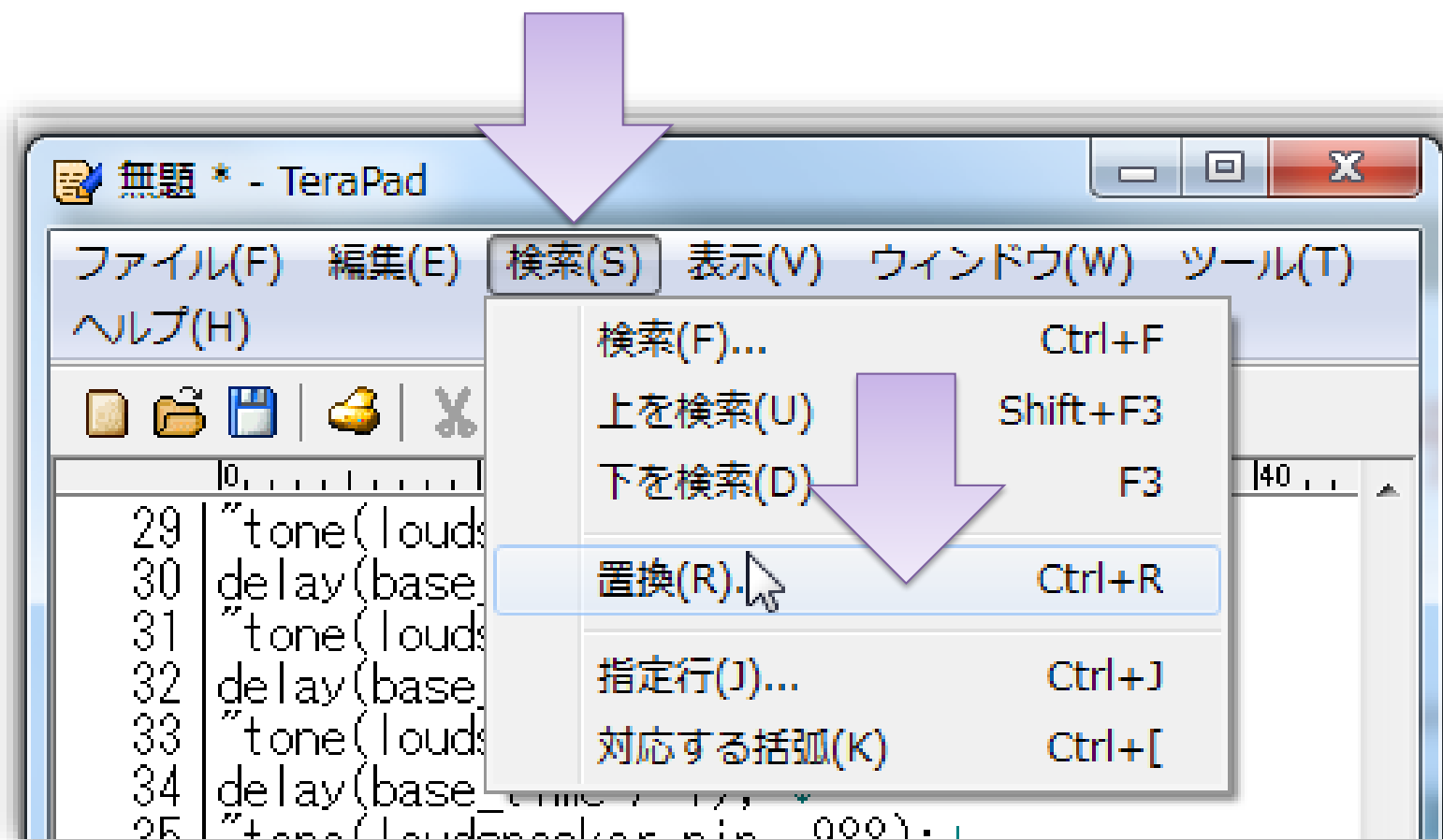
ツールを使う流れ

1. 楽譜を読む（音階と長さ）
2. Excelに音符を入力
3. 生成されたコードをテキストエディタにコピー
 1. TeraPadを推奨
 2. ダブルコーテーション「”」を置換機能で削除
 3. 再びコピー
4. Arduino IDEにペースト
 1. loop()関数内に貼り付ける

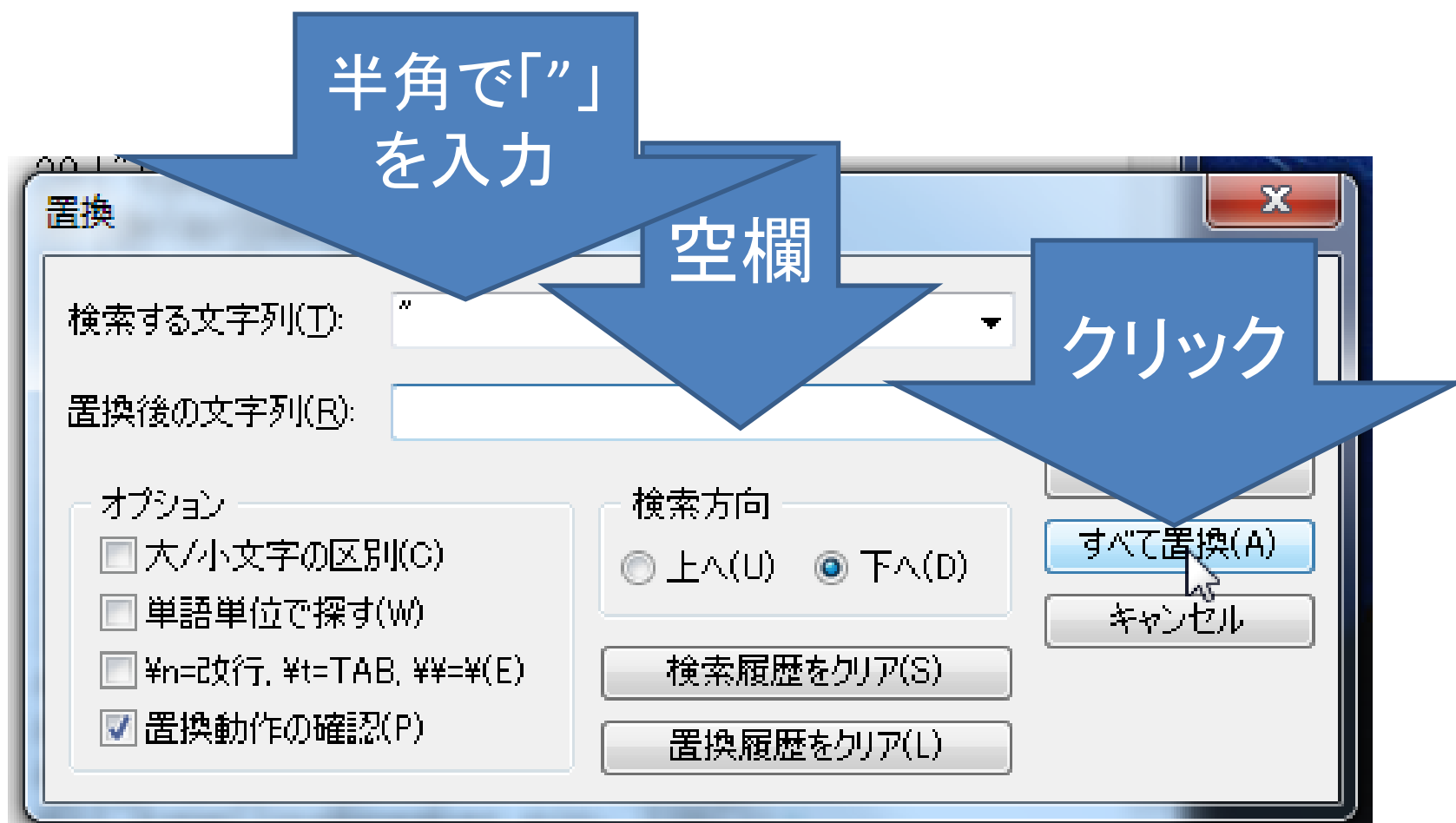
生成されたコードをテキストエディタに コピー



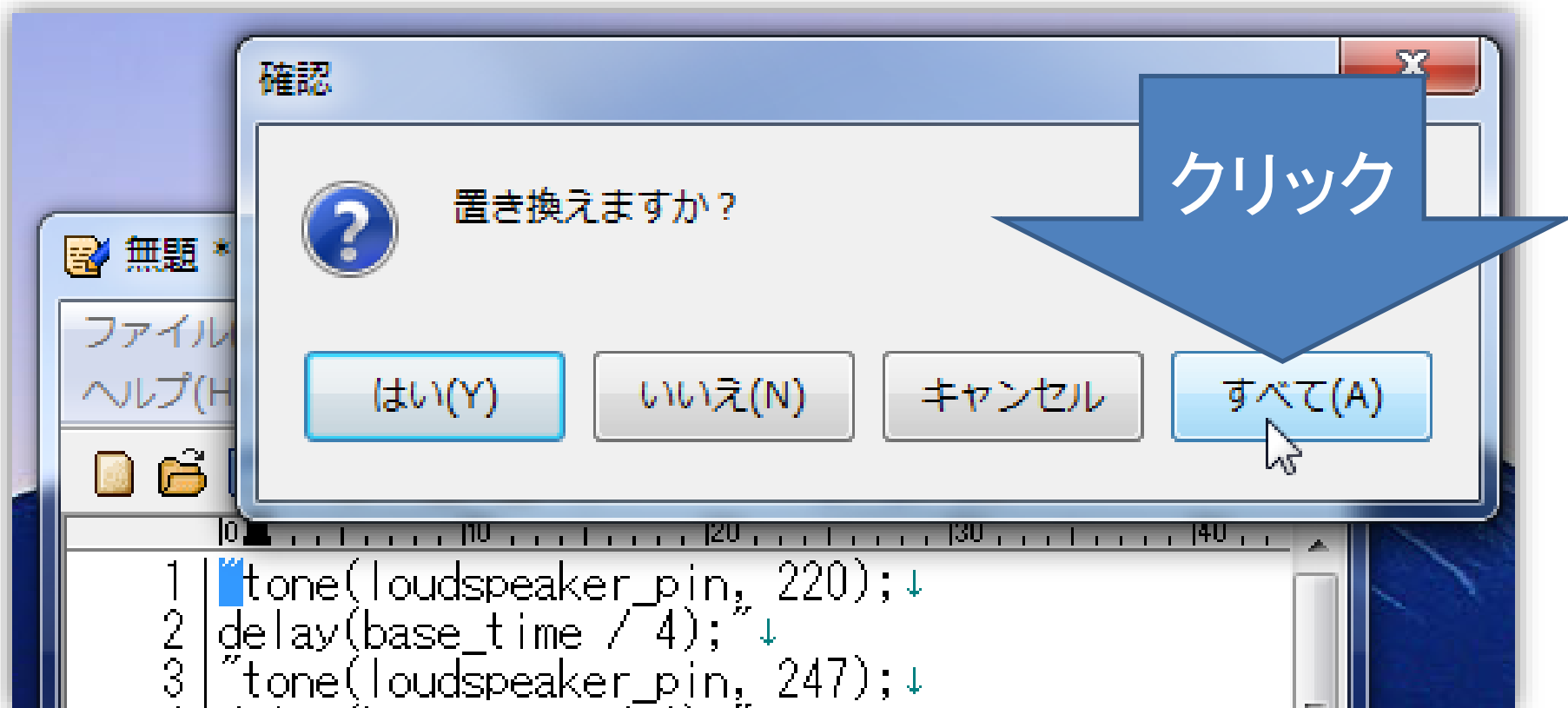
「”」を置換する(置換機能の起動)



置換条件の設定



置換の実行



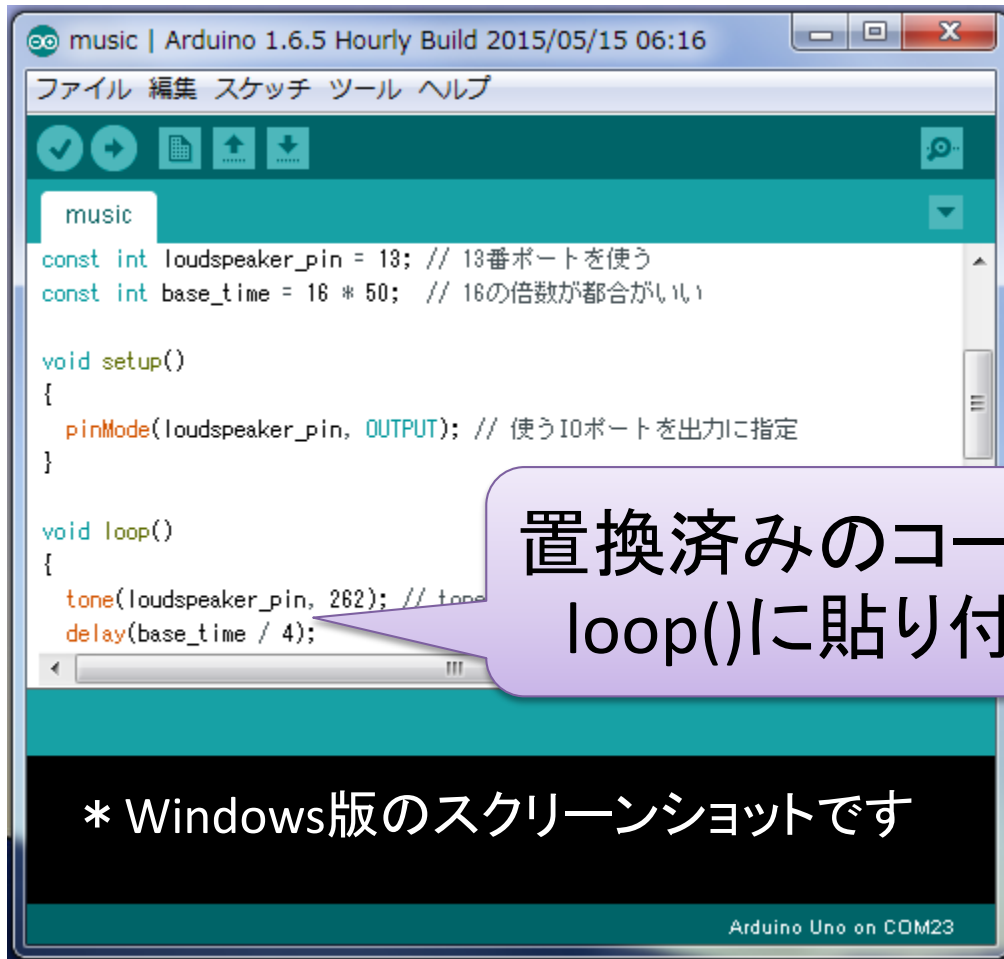
「”」が取れた(^o^)/



The screenshot shows a TeraPad text editor window titled "無題 * - TeraPad". The menu bar includes "ファイル(F)", "編集(E)", "検索(S)", "表示(V)", "ウィンドウ(W)", "ツール(T)", and "ヘルプ(H)". The toolbar contains icons for file operations (new, open, save, print, copy, paste, undo, redo) and search (find, replace, find all). The text area contains a C program snippet with line numbers 1 through 9. The code uses the `tone` and `delay` functions. The cursor is positioned at the end of line 9.

```
1 tone(loudspeaker_pin, 220); ↓
2 delay(base_time / 4); ↓
3 tone(loudspeaker_pin, 247); ↓
4 delay(base_time / 4); ↓
5 tone(loudspeaker_pin, 262); ↓
6 delay(base_time / 4); ↓
7 tone(loudspeaker_pin, 294); ↓
8 delay(base_time / 4); ↓
9
```

IDE起動！ー＞プログラムを書く



サンプル入手先

https://github.com/KatsuhiroMorishita/electronic_music_box/tree/master/Arduino/code/music

```

const int loudspeaker_pin = 13;
const int base_time = 16 * 50; // 16の倍数が良い

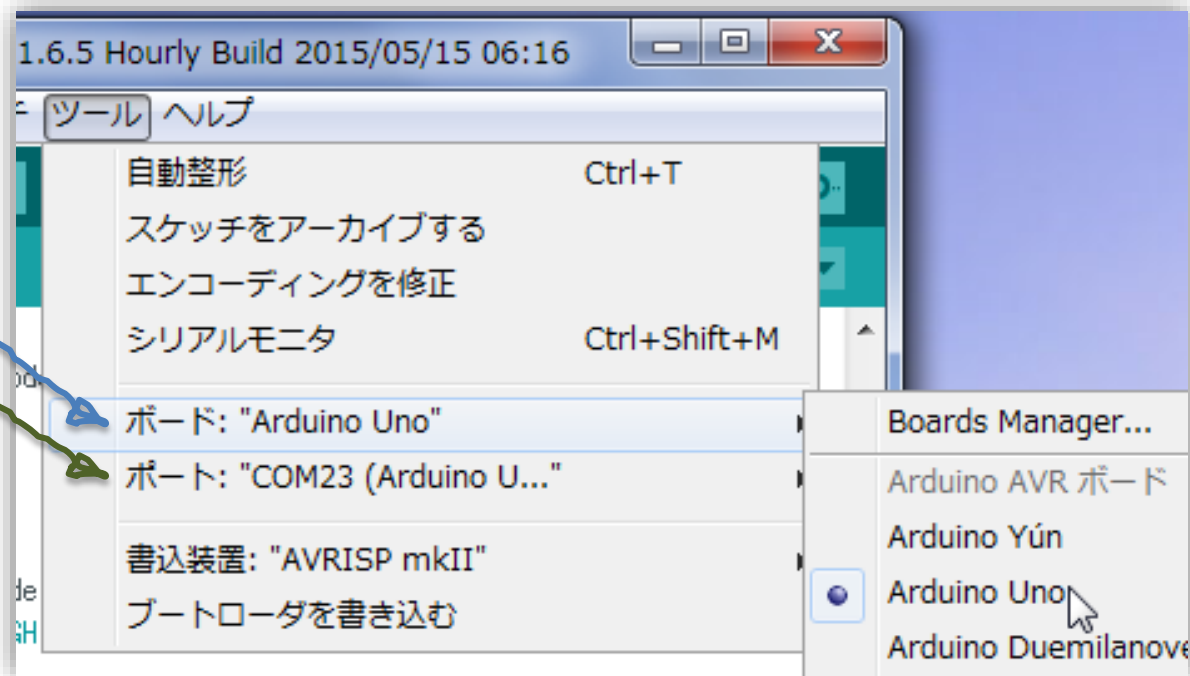
// 起動後に1度だけ実行される
void setup()
{
    pinMode(loudspeaker_pin, OUTPUT);
}

// 繰り返し実行される
void loop()
{
    tone(loudspeaker_pin, 262);
    delay(base_time / 4);
    tone(loudspeaker_pin, 294);
    delay(base_time / 4);
    tone(loudspeaker_pin, 330);
    delay(base_time / 4);
    tone(loudspeaker_pin, 349);
    delay(base_time / 4);
    tone(loudspeaker_pin, 392);
    delay(base_time / 4);
    tone(loudspeaker_pin, 440);
    delay(base_time / 4);
    tone(loudspeaker_pin, 494);
    delay(base_time / 4);
    tone(loudspeaker_pin, 523);
    delay(base_time / 4);
}

```

設定して

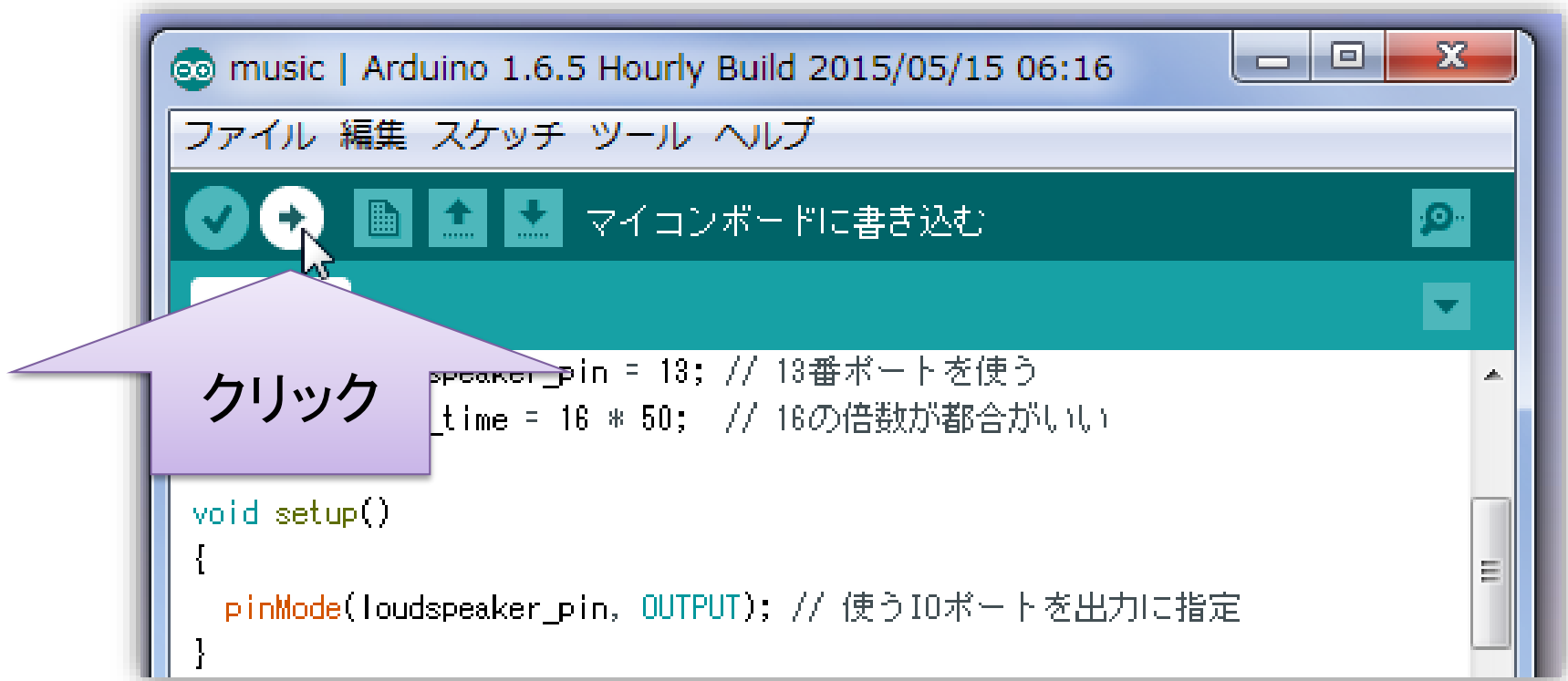
- ボードのシリーズを選ぶ
- ポートのマイコンor電圧を選ぶ



* Windows版のスクリーンショットです

書き込んで

- コンパイル & 書き込み (エラーが出たら修正)

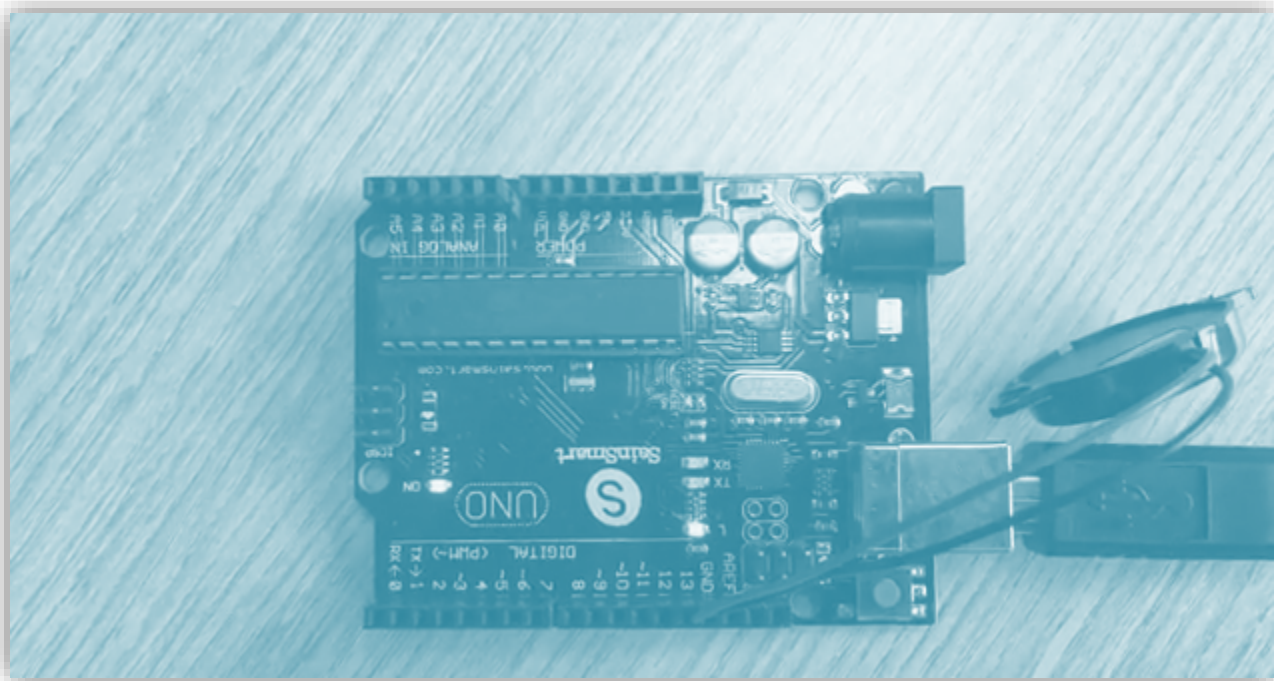


* Windows版のスクリーンショットです

完成！！

- 動作の例

- <https://www.youtube.com/watch?v=UCOcOTA026Q>















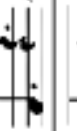



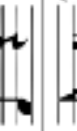















自分のオリジナル曲を 作ってみよう



*複数台を連携させれば複雑な音楽が実現するかも！？

参考資料

2.音符と休符の長さと割合

音符休符の形と名前		長さの割合			
	全 休 符 全 音 符				4拍
	付点二分休符 付点二分音符				3拍
	二分休符 二分音符				2拍
	付点四分休符 付点四分音符				1½拍
	四分休符 四分音符				1拍
	付点八分休符 付点八分音符				¾拍
	八分休符 八分音符				½拍
	十六分休符 十六分音符				¼拍

