

Sound

by poem

Modified by Alphar

Introduction

- Sound
 - Playing **sound** at **specific frequencies** and **duration** via the speaker on the motherboard.

Function

- Functions

- `bool Beep(int freq, int duration)`
 - Headfile: `windows.h`
 - Turning the PC speaker on at the **specified frequency**.
 - Human's ears: able to hear sound with freq between 20 Hz ~ around 20 KHz.
 - **freq**: The frequency of the sound, in hertz. This parameter must be in the range 37 through 32,767 (0x25 through 0x7FFF).
 - **duration**: The duration of the sound, in milliseconds.
 - Example: `Beep(262,500); //Playing C4 for 0.5 sec`

* Ref: <https://docs.microsoft.com/en-us/windows/win32/api/utilapiset/nf-utilapiset-beep>

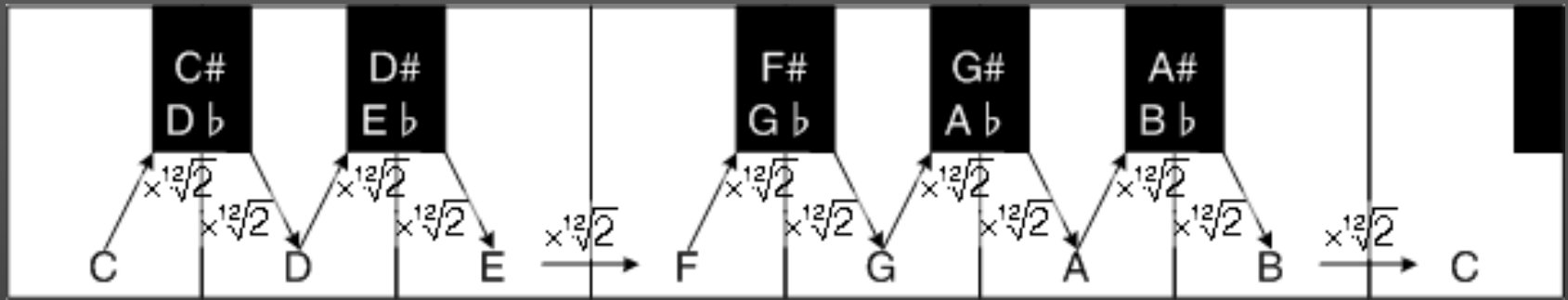
Pitch

- Pitch and Frequency (used in `sound()`)

Name	Freq	Name	Freq	Name	Freq	Name	Freq
C (Do)	131	C	262	C	523	C	1047
D (Re)	147	D	294	D	587	D	1175
E (Mi)	165	E	330	E	659	E	1319
F (Fa)	175	F	349	F	698	F	1397
G (Sol)	196	G	392	G	784	G	1568
A (La)	220	A	440	A	880	A	1760
B (Ti)	247	B	494	B	988	B	1976

Pitch (cont)

- Pitch and Frequency (cont)
 - In music,



- `Beep(262, 500)`
 // playing the middle C with 0.5 sec
- `Beep((int)(262 * pow(2, 1./12)), 500)`
 // playing C#

Pitch (cont)

■ List of frequencies with 88 piano notes.

B0	31	C1	33	CS1	35	D1	37	DS1	39
E1	41	F1	44	FS1	46	G1	49	GS1	52
A1	55	AS1	58	B1	62	C2	65	CS2	69
D2	73	DS2	78	E2	82	F2	87	FS2	93
G2	98	GS2	104	A2	110	AS2	117	B2	123
C3	131	CS3	139	D3	147	DS3	156	E3	165
F3	175	FS3	185	G3	196	GS3	208	A3	220
AS3	233	B3	247	C4	262	CS4	277	D4	294
DS4	311	E4	330	F4	349	FS4	370	G4	392
GS4	415	A4	440	AS4	466	B4	494	C5	523
CS5	554	D5	587	DS5	622	E5	659	F5	698
FS5	740	G5	784	GS5	831	A5	880	AS5	932
B5	988	C6	1047	CS6	1109	D6	1175	DS6	1245
E6	1319	F6	1397	FS6	1480	G6	1568	GS6	1661
A6	1760	AS6	1865	B6	1976	C7	2093	CS7	2217
D7	2349	DS7	2489	E7	2637	F7	2794	FS7	2960
G7	3136	GS7	3322	A7	3520	AS7	3729	B7	3951
C8	4186	CS8	4435	D8	4699	DS8	4978		

Function (cont)

- Functions

- `void _sleep(unsigned seconds)`
 - Suspending execution for interval (**second**).
 - Can be used as a **rest**.

👉 Dev-Beep.cpp

Function (cont)

- Functions

- `bool PlaySound(char *pszSound, int hmod, int fdwsound)`
 - Headfile: **windows.h**
 - Plays a sound specified **waveform sound (.wav)** by the given file name.
 - **pszSound**: A string that specifies the sound to play. If this parameter is **NULL**, any currently playing waveform sound is stopped.
 - **hmod**: Handle to the executable file that contains the resource to be loaded. This parameter must be **NULL** unless **SND_RESOURCE** is specified in *fdwSound*.
 - **fdwsound**: Flags for playing the sound. Please refer to the following URL for detail. [https://docs.microsoft.com/en-us/previous-versions/dd743680\(v=vs.85\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/dd743680(v=vs.85)?redirectedfrom=MSDN)

Function (cont)

- Functions

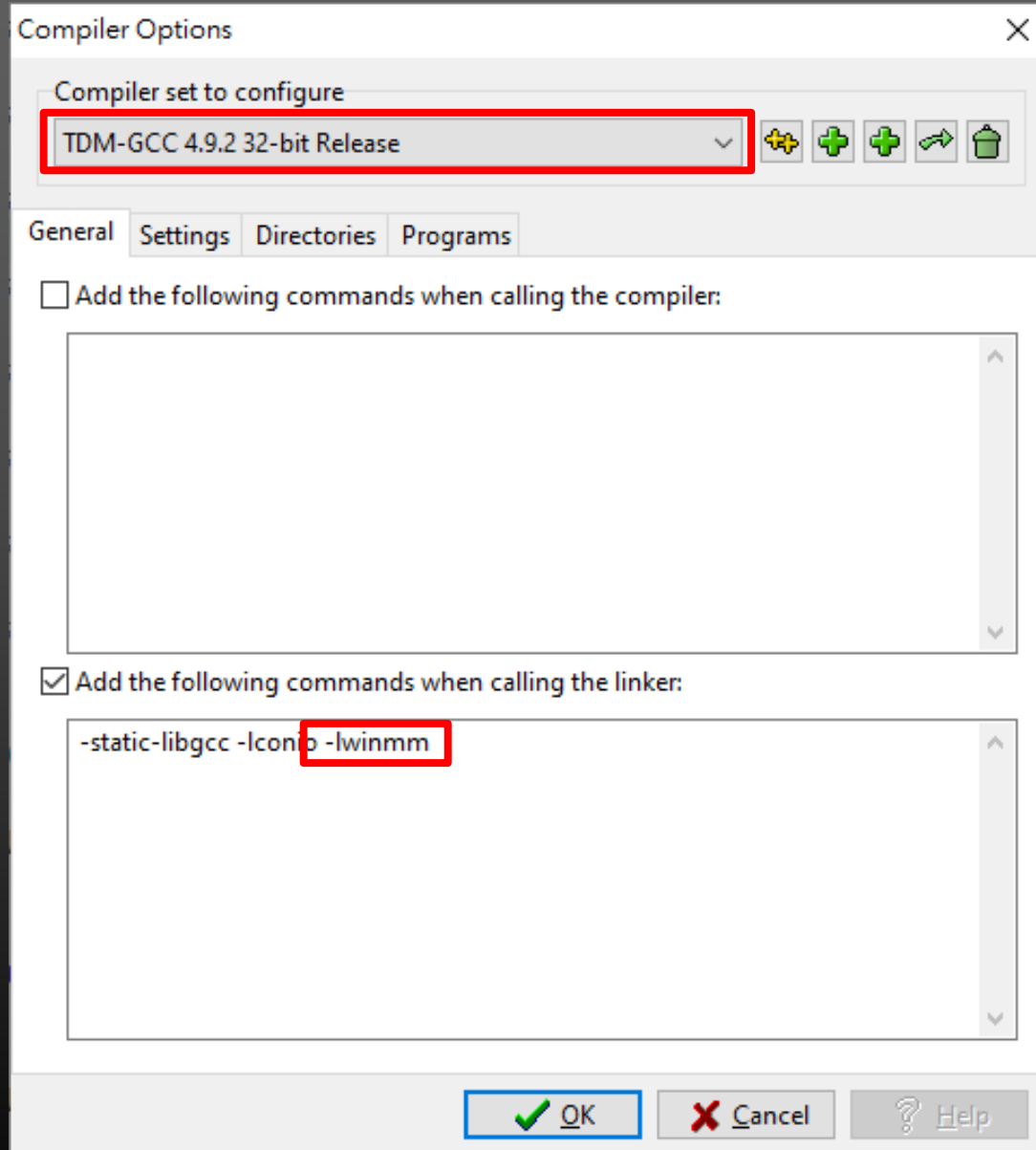
- `bool PlaySound(char *pszSound, int hmod, int fdwsound)`

- Adding linker

- In Dev-C++ compiler
 - Tools → Compiler Options
 - Compiler set to configure: **TDM-GCC 4.7.1 32-bit Release**
 - Type “**-lwinmm**” in the dialog box below

☞ Dev-Playsond.cpp

Introduction (cont)



Exercise

- Exercise
 - Writing a program to play a diatonic scale from middle C (that is, Do, Re, Mi, Fa, So, La, Ti, Do). The length of each note is 500 msec. You have to use loop and cannot use array.

Advanced Method for Playing a Song

- Array
 - Use array to store all pitches and pauses of a song
 - Then, use loops to read this array
- File
 - Use file to store all pitches and pauses of a song
 - Then, read this file to play this song
 - More flexible