

VII. Making music

The NXT has a built-in speaker that can play tones and even sound files. This is in particular useful when you want to make the NXT tell you that something is happening. But it can also be funny to have the robot make music or talk while it runs around.

Playing sound files

BricxCC has a built-in utility to convert .wav files into .rso files accessible via menu Tools → Sound conversion.

Then you can store .rso sound files on NXT flash memory using another utility, the NXT memory browser (Tools → NXT explorer) and play them with the command

```
PlayFileEx(filename, volume, loop?)
```

Its arguments are sound filename, volume (a number from 0 to 4), and loop: this last argument is set to 1 (TRUE) if you want the file to be looped or 0 (FALSE) if you want to play it only once.

```
#define TIME 200
#define MAXVOL 7
#define MINVOL 1
#define MIDVOL 3
#define pause_4th Wait(TIME)
#define pause_8th Wait(TIME/2)
#define note_4th \
    PlayFileEx("! Click.rso",MIDVOL,FALSE); pause_4th
#define note_8th \
    PlayFileEx("! Click.rso",MAXVOL,FALSE); pause_8th

task main()
{
    PlayFileEx("! Startup.rso",MINVOL,FALSE);
    Wait(2000);
    note_4th;
    note_8th;
    note_8th;
    note_4th;
    note_4th;
    pause_4th;
    note_4th;
    note_4th;
    Wait(100);
}
```

This nice program first plays the startup tune you might know already; then uses the other standard click sound to play “Shave and a haircut” jingle that made Roger Rabbit go crazy! The macros are really useful in this case to simplify notation in the main task: try modifying the volume settings to add accents in the tune.

Playing music

To play a tone, you can use the command `PlayToneEx(frequency, duration, volume, loop?)`

It has four arguments. The first is the frequency in Hertz, the second the duration (in 1/1000 of a second, like in the wait command), and the last are volume a loop as before. `PlayTone(frequency, duration)` can also be used; in this case the volume is the one set by NXT menu, and loop is disabled.

Here is a table of useful frequencies:

Sound	3	4	5	6	7	8	9
-------	---	---	---	---	---	---	---

B	247	494	988	1976	3951	7902	
A#	233	466	932	1865	3729	7458	
A	220	440	880	1760	3520	7040	14080
G#		415	831	1661	3322	6644	13288
G		392	784	1568	3136	6272	12544
F#		370	740	1480	2960	5920	11840
F		349	698	1397	2794	5588	11176
E		330	659	1319	2637	5274	10548
D#		311	622	1245	2489	4978	9956
D		294	587	1175	2349	4699	9398
C#		277	554	1109	2217	4435	8870
C		262	523	1047	2093	4186	8372

As with the case of `PlayFileEx`, the NXT does not wait for the note to finish. So if you use multiple tones in a row then you had better add (slightly longer) wait commands in between. Here is an example:

```

#define VOL 3

task main()
{
    PlayToneEx(262,400,VOL,FALSE); Wait(500);
    PlayToneEx(294,400,VOL,FALSE); Wait(500);
    PlayToneEx(330,400,VOL,FALSE); Wait(500);
    PlayToneEx(294,400,VOL,FALSE); Wait(500);
    PlayToneEx(262,1600,VOL,FALSE); Wait(2000);
}

```

You can create pieces of music very easily using the Brick Piano that is part of the BricxCC.

If you want to have the NXT play music while driving around, better use a separate task for it. Here you have an example of a rather stupid program where the NXT drives back and forth, constantly making music.

```
task music()
{
  while (true)
  {
    PlayTone(262,400); Wait(500);
    PlayTone(294,400); Wait(500);
    PlayTone(330,400); Wait(500);
    PlayTone(294,400); Wait(500);
  }
}

task movement()
{
  while(true)
  {
    OnFwd(OUT_AC, 75); Wait(3000);
    OnRev(OUT_AC, 75); Wait(3000);
  }
}

task main()
{
  Precedes(music, movement);
}
```

Summary

In this chapter you learned how to let the NXT play sounds and music. Also you saw how to use a separate task for music.