

将来にわたって使えるスキルを身につけるために

オブジェクト指向で はじめる プログラミング

日向 俊二
HYUGA, Shunji

第10回

C#でMIDIを扱う (その1)

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
.NET Framework SDK

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥CSHARPディレクトリに収録しています。

¥MIDIKB
キーボードプログラム

※この記事では.NET Framework SDKを利用して解説していますが、C#Builder、SharpDevelopなどでも利用できます(C#コンパイラならすべて利用できます)。



MIDIとは?

MIDI (Musical Instrument Digital Interface) は、MIDIインターフェイスを備えた楽器や音源、ミュージックシーケンサなどを制御するための標準インターフェイスです。

サウンドを扱うソフトウェアでは、WAVEやMP3 (MPEG-1 Audio Layer-3) もよく使われますが、MIDIは音の情報や、作/編曲された既存の音楽を保存し再生するだけでなく、音源や音響装置を制御したり、任意の高さと音色の音を必要ときに鳴らしたいような場合にも使うことができます。特に、音楽を耳に聞こえる音の波形情報としてではなく、一連の音 (さまざまな高さ、長さ、大きさの音) の情報として保存する場合にはMIDIを使うのが一般的で最適です。

今回はC#でMIDIを利用するプログラムについて解説します。



MIDIファイル

MIDIファイルのデータの実体は、MIDIのイベントのシーケンス (イベントが連続したもの) です。MIDIのイベントには、たとえば、「中央ドの音を80の強さで楽器番号0番の音色で鳴らす」とか、「現在鳴っている音をすべて止める」「テンポの基準を変える」など、一般に人間が楽器などを使って音楽を表現する際に行なう出来事 (イベント) があります。つまり、サウンドとしてのMIDIは、このようなMIDIのイベントのシーケンスを記録したデータです。この情報を保存したファイルを「標準MIDIファイル」と呼びます。標準MIDIファイルの一般的な拡張子は“.MID”または“.RMI”です。

インターフェイスとしてのMIDIも、MIDIサウンドも、コンピュータの世界では広く普及しています。実際、MIDIは主にコンピュータミュージックで使われますが、MIDIインターフェイスを備えた他のデバイスの制御に利用する

オブジェクト指向で はじめる プログラミング

こともできます。

インターフェイスとしてのMIDIの仕様は、ハードウェアとソフトウェアの両方で構成されています。ハードウェアの仕様には、たとえば、MIDIケーブルやそのコネクタなどが既定されています。しかし、アプリケーションやGUIコンポーネントなどのプログラミングでは、ハードウェアに近い低レベルの領域は通常扱いません。一般的なMIDIのプログラミングでは、抽象化されたソフトウェアオブジェクトである「MIDIデバイスオブジェクト」をプログラムコードで制御します*1。

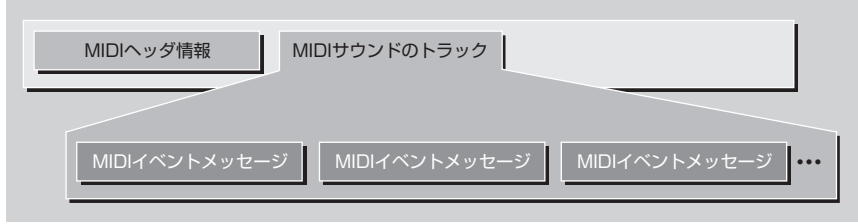
MIDIファイルの中身

標準MIDIファイル*2は、ヘッダ情報、トラック、チャンネル、イベント（メッセージ）などで構成されます（図1）。

標準MIDIファイルフォーマットには、すべての楽器（チャンネル）のMIDIイベントデータを1個のトラックに保存する「フォーマット0（ゼロ）」と、楽器ごとに別のトラックに保存する「フォーマット1」の2種類があります。いずれのフォーマットでも、トラックには「MIDIイベントメッセージ」が含まれています。そして、フォーマットの種類にかかわらず、MIDIでは音とその関連情報をMIDIイベントで表現します。また、MIDIデバイスで単に音を鳴らしたり止めたりしたいような場合には、そのデバイスに適切なMIDIイベントメッセージを送ることで実行できます。

- ★1 アプリケーションレベルでは、一般的には、音源を制御するデバイスドライバ、もしくはソフトウェア音源に実装されていて、システムにインストールされているMIDIインターフェイスを使ってMIDIを制御したり音を鳴らします。
- ★2 Standard MIDI File。略して「SMF」と表記することもあります。

図1：MIDIデータの典型的な構造



MIDIメッセージ

MIDIメッセージには、数バイトのデータで構成される「MIDIショートメッセージ」と、1回に数バイト以上のデータを送ることができる「MIDIロングメッセージ」の2種類があります。

コンピュータで単にMIDIサウンドを鳴らしたいときには、ノートオンコマンドやノートオフコマンドのようなMIDIショートメッセージをデバイスに送ることで音源を鳴らします。また、特定のシステムに固有の（システムエクスクルーシブ）情報や、楽器名と歌詞のようなテキストなどは、MIDIロングメッセージをデバイスに送ることでMIDIデバイスに伝えます。

MIDIの主なショートメッセージとロングメッセージを表1に示します。MIDI

のフォーマットやMIDIメッセージの詳細については、記事末の参考URLなどを参照してください。

MIDIとWindows

MIDIを扱うWindowsプログラムを作る場合、現在のところ、「winmm.dll」の中にあるMIDI関数を使うのが最も一般的な方法です。winmm.dllは従来のWindowsのAPIであるWin32 APIの一部で、DirectXの機能を実際に提供する要素のひとつでもあります。

一方、C#のWindowsプログラミングでは、原則的に直接Win32 APIを使うのではなく、.NET Frameworkのクラスを使います。そうするほうがプログラムの一貫性を保ちやすく、プログラミングやデバッグも容易だからです。

表1：MIDIの主なイベントメッセージの例

種類	コマンド	メッセージ (例)
ショート	ノートオン	9n, pp, vv (nはチャンネル、ppはピッチ、vvは音の強さ)
	ノートオフ	8n, pp, vv
	オールノートオフ	Bn, 7B, 00
	コントロールチェンジ	Bn, cc, dd (ccはコントロールナンバー、ddは制御データ)
ロング	システムエクスクルーシブ	F0, ID, データ, FF
	テキスト	FF, 01, データ長, テキストデータ
	楽器名	FF, 04, データ長, テキストデータ
	歌詞	FF, 05, データ長, テキストデータ