

Python

サウンド・オーディオ プログラミング



MIDIと
WAVEサウンドの
プログラミング

日向俊二●著



■ サンプルファイルのダウンロードについて

本書掲載のサンプルファイルは、下記 URL からダウンロードできます。

<https://----->

- 本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送または E-mail）でお送りください。
- 電話によるお問い合わせはお受けできません。
- 本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
- 匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および ® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

はじめに

現代ではさまざまな場面でサウンドが使われています。音楽や作曲のアプリやゲームはもちろん、Web アプリや Web ブラウザに表示される一般的なサイトでもサウンドが使われることがよくあります。

本書は、コンピューターにおけるサウンドの扱い方と、Python を使ったサウンドプログラミングのやさしい解説書です。

第 1 章と第 2 章では、コンピューターで扱うサウンドと Python の基本的なことについて概説します。

第 3 章から第 5 章では、標準化された規格である ^{ミディ}MIDI を活用するサウンドプログラミングについて解説します。

第 6 章と第 7 章では、任意の波形のオーディオを生成する ^{ウェーブフォーム}Waveform オーディオプログラミングについて説明します。

本書ではプログラミング言語として Python を使います。Python は豊富なライブラリを比較的容易に使えるプログラミング言語です。また、プログラミングの初心者でも読み進めるように配慮してあります。

本書の表記

- > Windows のコマンドプロンプトを表します。
- \$ Linux や WSL など UNIX 系 OS のコマンドプロンプトを表します。
- >>> Python のインタラクティブシェル（インタプリタ）のプロンプト（一次プロンプトともいう）を表します。Python のインタラクティブシェルでの実行例で「>>>」が記載されていても、それは入力しません。
- ... Python のインタラクティブシェル（インタプリタ）の行の継続を表します。Python のインタラクティブシェルでの実行例で「...」が掲載されていても、それは入力しません。
- () ひとまとまりの実行可能なコードブロックである関数であることを示します。たとえば、main という関数を表すときに、「main という名前の関数」や「関数 main()」と表記しないで、単に「main()」と表記することがあります。
- abc* 斜体で表す部分は、そこに具体的な文字や数値、変数、式などが入ることを表します。たとえば「Python *m.n*」は、*m* や *n* に具体的な値が入ること、例えば「Python 3.11」などとなることを表します。
- abc** 太字で表す部分は、ユーザー（プログラマ）が入力する式や値などであることを表します。
- 0xn 0x で始まる表記は 16 進数表現の整数であることを表します。たとえば、0x41 は 10 進数で 65 であることを表します。この表記は主に Python のプログラムの中で使います。
- nmH 数値の最後に H が付く数は 16 進数表現であることを表します。たとえば、41H は 10 進数で 65 であることを表します。この表記は主に値を説明するときに使います。
- 0bn 0b で始まる表記は 2 進数表現の整数であることを表します。たとえば、0b111 は 10 進数で 7 であることを表します。
- [X] キーボードのキーを押すことを示します。たとえば、[F5] は「F5」キーを押すことを意味します。
- [S] + [X] キーボードの「S」キーを押したまま「X」キーを押すことを示します。[Ctrl] + [F5] は、「Ctrl」キーを押したまま「F5」キーを押すことを意味します。
- Note** 本文を補足するような説明や、知っておくとよい話題です。

ご注意

- 本書の内容は本書執筆時の状態で記述しています。Python やプログラムが使用するモジュールのバージョンによっては本書の記述と実際とが異なる結果となる可能性があります。
- 本書は、Python、サウンド、オーディオ、MIDI などについてすべてを完全に解説するものではありません。必要に応じて他のリソースを参照してください。
- 本書のサンプルは、プログラミングを理解するために掲載するものです。実用的なプログラムとして提供するものではありませんので、ユーザーのエラーへの対処やセキュリティ、その他の面で省略してあるところがあります。
- 音が鳴るプログラムを実行する際には、音量にご注意ください。

プログラムの実行を確認した環境

Python 3.11.3、pygame 2.5.0、PyAudio 0.2.13

本書に関するお問い合わせについて

本書に関するお問い合わせは、sales@cutt.co.jp にメールでご連絡ください。

なお、お問い合わせは本書に記述されている範囲に限らせていただきます。特定の環境や特定の目的に対するお問い合わせ等にはお答えできませんので、あらかじめご了承ください。特に、特定の環境における特定の開発ツールのインストールや設定、使い方、読者固有の環境におけるエラーなどについてご質問いただいてもお答えできませんのでご了承ください。

お問い合わせの際には下記事項を明記していただきますようお願いいたします。

- 氏名
- 連絡先メールアドレス
- 書名
- 記載ページ
- お問い合わせ内容
- 実行環境

目次

はじめに

iii

第 1 章 コンピューターとサウンド

1

1.1 音について	2
1.1.1 音	2
1.1.2 音の波	2
1.1.3 音の性質と波形	3
1.1.4 倍音と音色	4
1.1.5 アナログとデジタル	4
1.1.6 オーディオとサウンド	5
1.2 MIDI サウンド	6
1.2.1 MIDI	6
1.2.2 MIDI デバイス	7
1.2.3 MIDI メッセージ	7
1.2.4 MIDI メッセージの実体	7
1.2.5 ノートナンバー	8
1.2.6 デルタタイム	10
1.2.7 可変長表現	10
1.2.8 GM	10
1.2.9 標準 MIDI ファイル	11
1.3 Waveform オーディオ	11
1.3.1 オーディオの形式	11
1.3.2 オーディオの生成と合成	12

第2章 Pythonの基礎 13

2.1 Python との対話	14
2.1.1 インタラクティブシェル	14
2.1.2 プロンプト	15
2.1.3 単純な加算	15
2.1.4 print() を使った出力	16
2.1.5 行継続文字	18
2.2 スクリプトファイル	19
2.2.1 ファイルの作成	19
2.2.2 ファイルの保存	20
2.2.3 スクリプトの実行	20
2.3 実行方法の比較	21
2.3.1 インタラクティブシェル	21
2.3.2 スクリプトファイル	22

第3章 MIDIの出力と入力 23

3.1 MIDI メッセージ	24
3.1.1 ノートオン	24
3.1.2 ノートオフ	24
3.1.3 オールノートオフ	24
3.1.4 オールサウンドオフ	25
3.1.5 ピッチベンド	25
3.1.6 プログラムチェンジ	25
3.1.7 コントロールチェンジ	25
3.1.8 SysEx メッセージ	26
3.1.9 メタイベント	26
3.2 単純な出力	27
3.2.1 pygame の準備	27
3.2.2 単音の出力	27
3.2.3 和音の出力	30
3.3 出力の応用	31
3.3.1 メロディーの出力	31

3.3.2	ピッチベンド	32
3.3.3	音色の設定	34
3.3.4	打楽器	40
3.3.5	音程のある打楽器	43
3.3.6	効果音	44
3.4	MIDI 入力	46
3.4.1	MIDI 入力デバイス	46
3.4.2	MIDI 入出力	48

第 4 章 Mido 51

4.1	Mido の基礎	52
4.1.1	Mido とは	52
4.1.2	Mido のインストール	52
4.2	Mido による単純な出力	53
4.2.1	単音の出力	53
4.2.2	和音の出力	55
4.2.3	音色の設定	56
4.3	Mido による出力の応用	58
4.3.1	メロディーの出力	58
4.3.2	ピッチベンド	59
4.3.3	打楽器	61
4.3.4	音程のある打楽器	62
4.3.5	効果音	64
4.4	Mido による MIDI 入力	66
4.4.1	MIDI 入力デバイス	66
4.4.2	MIDI 入出力	67

第 5 章 SMF 69

5.1	MIDI ファイル	70
5.1.1	SMF の構造	70
5.1.2	SMF ファイルの内容	72
5.1.3	3トラックの SMF の例	74

5.2 SMF の作成	76
5.2.1 Mido による SMF の生成	76
5.2.2 メロディーの SMF の作成	77
5.3 SMF の再生	79
5.3.1 Mido による SMF の再生	79

第 6 章 Waveform オーディオ 83

6.1 基礎知識	84
6.1.1 PyAudio	84
6.1.2 波形データの生成	84
6.2 サイン波	87
6.2.1 sin のグラフ	87
6.2.2 サイン波を鳴らすプログラム	89
6.2.3 振幅の変更	92
6.3 さまざまな波形	95
6.3.1 矩形波	95
6.3.2 ノコギリ波	98
6.4 サイン波の合成	101
6.4.1 倍音の合成	101
6.4.2 低音の合成	104
6.4.3 3 種類のサイン波の合成	108

第 7 章 Waveform ファイル 113

7.1 WAV ファイル	114
7.1.1 WAV ファイルの構造	114
7.1.2 WAV ファイルの内容	115
7.2 ファイルの生成	118
7.2.1 wave モジュール	118
7.2.2 WAV ファイルを生成するプログラム	118
7.2.3 ノコギリ波の WAV ファイル生成	120

7.3 ファイルの再生	121
7.3.1 WAV ファイルを再生するプログラム	121
7.4 MP3 への変換	123
7.4.1 MP3 ファイル	123
7.4.2 FFmpeg	123
7.4.3 MP3 への変換プログラム	124
付録 A Python のインストール	129
付録 B トラブル対策	133
付録 C FFmpeg	137
付録 D 参考リソース	141
索引	143

第 1 章

コンピューターとサウンド

この章では、サウンドの実体についてと、コンピューターで扱う代表的なサウンドの種類について概説します。

1.1

音について

コンピューターでは、短い警告音から長大な音楽までさまざまな音を扱います。

1.1.1 音

音の実体は空気の振動です。音は空気中を疎密波として伝わります。疎密波は、空気の疎ら^{まば}なところと、密なところが繰り返して存在する波です。

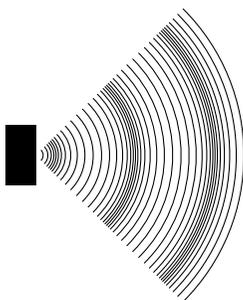


図 1.1 ● 音が伝搬するイメージ

疎と密を繰り返す頻度が多ければ高い音になり、繰り返す頻度が少なければ低い音になります。また、空気の圧力が大きければ音は大きく、小さければ音は小さくなります。

1.1.2 音の波

音は粗密波として伝わりますが、この音の波は縦波です。そのため、距離に応じて振動の頻度が変わります。疎であったり密である空気の振動が拡散してゆくといっても良いでしょう。

しかし、これはイメージしにくいので、慣用的に音の波は、振動した強さ（縦軸）と時間（横軸）の関係を表した図 1.2 のようなものとして表します。

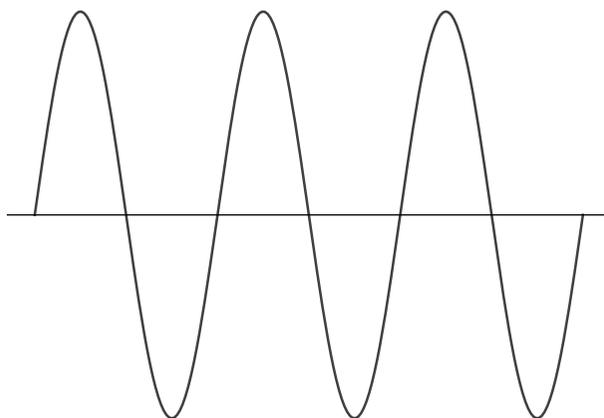


図 1.2 ● 音の波の表現

このように表現したときには、一定の時間間隔の中で振動を繰り返す頻度（周波数）が多ければ高い音になり、繰り返す頻度が少なければ低い音になります。また、波の大きさ（振幅）が大きければ音は大きく、小さければ音は小さくなります。

1.1.3 音の性質と波形

図 1.2 では、音の波形としてサイン（sin）のグラフに似た一定の規則で順次変化する波を示しました。このように振動する波をサイン波と言います。サイン波は単純な響きの音として耳に聴こえます。

人の声や物音、楽器の音など、普段耳にするさまざまな音は、さらに複雑な波形であるのが普通です。

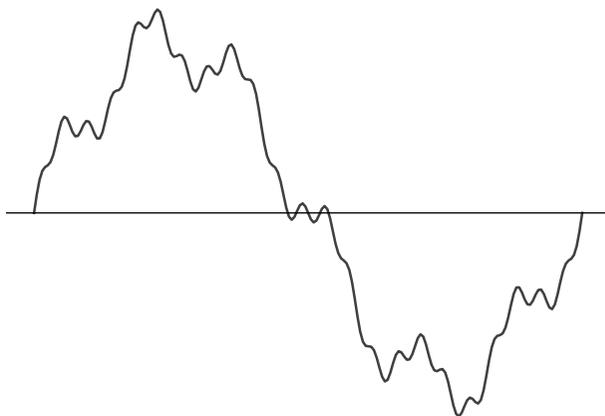


図 1.3 ● 複雑な波形の例

一般に、波形がシンプルであれば澄んだ音に聞こえ、波形が複雑になればなるほど豊かな響きの音と感じます。たとえば音叉の音色の波形は比較的単純で澄んだ音色に聴こえ、一方、クラリネットやオーボエの音色の波形は複雑で豊かな響きに聴こえます。

また、単純な波形に倍音を重ねると複雑な波形になります（後の章で実際の例をみます）。

1.1.4 倍音と音色

多くの音には、基礎となる周波数に加えて、周波数が基礎となる周波数の2倍の音、3倍の音など、基礎となる周波数の倍数の音が含まれています。整数倍のこのような音を倍音といいます。さまざまな倍音を複雑に重ね合わせると、音の響きも複雑になります。

倍音を含まない単純な波形の音は純音と呼びます。

1.1.5 アナログとデジタル

一般に、音は図 1.2 や図 1.3 に示したように滑らかな曲線を描くように変化する波として表現することができます。言い換えると、音の波は一般的にはアナログです。

Note ただし、第6章で示す矩形波やノコギリ波のように、滑らかな曲線のように変化しない波も存在します。

一方、コンピュータなど電子機器が扱う値はデジタルです。そこで、コンピュータなど電子機器で音を扱うときには、アナログの波形をデジタルに変換します。デジタルに変換する一般

的な方法は、波を非常に短い時間で区切って、区切った個々の大きさを数値にして表現する方法です。図 1.4 で、破線はアナログの波形であり、短い時間で区切った個々の値から構成される、連続する棒グラフのような表現がデジタルの音の表現です。

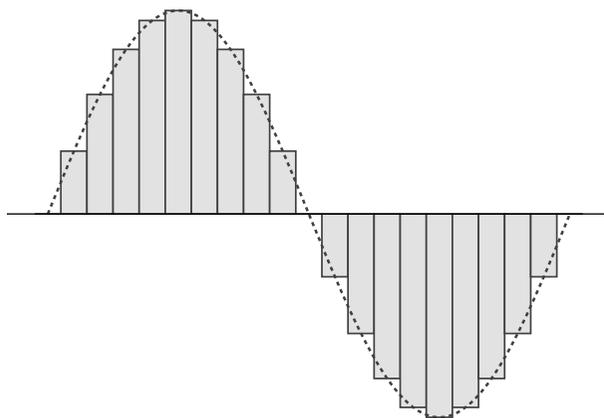


図 1.4 ● アナログ波形のデジタル化

本書では、以降、音の波を表現するときに、原則的には音の波はアナログであるかのように表現しますが、本書はプログラミングについての書籍なので、実際にコンピューターが扱う波はデジタルの波になります。

1.1.6 オーディオとサウンド

コンピューターやそのほかの電子機器で扱う音を表現するひとつの方法は、上記のようにデジタル化した波形で表す方法です。このような波形 (Waveform) で表す音を ウェーブフォーム Waveform オーディオといいます。

Waveform オーディオでは、音はデジタル化した波形で表されますが、一曲の音楽のような音のすべてをデジタル化した波形で表現すると、データの量がとても大きくなってしまいます。たとえば、4分ほどの曲を Waveform オーディオデータにすると、40 MB 程度になるでしょう (具体的なサイズは曲の内容によって異なります)。圧縮という技術を使ってデータを減らしても 4 MB ぐらいは必要になります。

コンピューターやそのほかの電子機器で音を扱う方法には、ほかの方法もあります。

鳴らす音をいくつかの情報 (音色、音の高さ、音の大きさなど) で指定するとオーディオを鳴らせるデバイスを用意しておいて、そのデバイスに鳴らす音の情報を送ると希望する音を

鳴らせるようにすれば、保存したりデバイスに送るデータ量はとでも少なくなります。このような目的で標準化されたインターフェースの仕様として MIDI (Musical Instrument Digital Interface) があります。このような音を MIDI サウンドということがあります。

音を扱うプログラミングでは、オーディオとサウンドという 2 種類の言葉を良く使います。オーディオ (audio) は音のうち可聴周波のという意味を持ち、そのような音を扱う装置のことを指すこともあり、一方、サウンド (sound) は音や音響を表しますが、いずれもコンピューターでは音を扱うときに使われます。

本書では、第 3 章から第 5 章で MIDI サウンドについて、第 6 章と第 7 章で Waveform オーディオのプログラミングについて解説します。

1.2

MIDI サウンド

MIDI は、MIDI インターフェースを備えた楽器や音源、ミュージックシーケンサーなどを制御するため標準インターフェースです。

1.2.1 MIDI

MIDI (Musical Instrument Digital Interface) は、MIDI インターフェースと呼ぶ標準化されたインターフェースを備えた楽器や音源、ミュージックシーケンサーなどを制御するため標準インターフェースです。MIDI サウンドのデータの実体は、MIDI のイベントやその他の制御情報のシーケンス (メッセージが連続したもの) です。

MIDI のイベントメッセージには、たとえば、「中央ドの音を 60 の強さでチャンネル 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」など、一般に人間が楽器などを操作して音楽を表現する際に行う出来事 (イベント) のメッセージがあります。また、「鳴らす楽器の種類を変える」、「テンポの基準を変える」などの制御情報も MIDI のメッセージに含まれます。

サウンドとしての MIDI は、このような MIDI のメッセージのシーケンスを記録したデータであるということもできます。

MIDI サウンドのプログラムは、PC やその他の制御装置と MIDI デバイスとの間で MIDI メッセージをやり取りすることでサウンドを鳴らしたり、サウンドの情報を装置が受け取れるようにします。

Note MIDI も 1.3 節で紹介する WAV（または WAVE）も、使われる状況やコミュニティーなどによって定義や表現はさまざまです。

1.2.2 MIDI デバイス

MIDI という規格に従って音を鳴らすための情報を送るとオーディオを鳴らせるデバイスを MIDI デバイスといいます。また、MIDI デバイスの中には MIDI キーボードのように PC やその他の制御装置に MIDI データを送ることができるものもあります。

ハードウェアの MIDI デバイスには音源やハードウェアシーケンサーと呼ばれる音を鳴らす機器のほかに、MIDI を使って照明などを制御する装置なども含まれます。ハードウェアの MIDI デバイスにプログラムがアクセスするためには、通常、接続状態に対応したデバイスドライバが必要です（OS のような基本ソフトウェアがサポートしている場合もあります）。

プログラミングの観点からは、MIDI デバイスは、必ずしもハードウェアではなく、MIDI の情報を送受できるようにしたソフトウェアも含まれます。

ソフトウェアの MIDI デバイスとして一般に良く使われるものは、MIDI で送られた情報をもとにして適切な音を生成して再生するミュージックシーケンサー（Music Sequencer）で、ソフトウェア音源ともいいます。

一般的に言って、ハードウェアの MIDI 出力デバイスはサウンドの質が高いうえに、PC などの制御装置は MIDI メッセージを送受するだけでよいので制御装置のリソース（CPU やメモリ）をあまり消費しません。一方、ソフトウェアの MIDI 出力デバイスはサウンドの質が低い傾向があるうえに PC などの制御装置のリソースを多く消費するという問題もあります。

1.2.3 MIDI メッセージ

すでに説明したように、MIDI サウンドのデータの実体は、MIDI のイベントなどのメッセージのシーケンス（連続したもの）です。

MIDI の情報は、プログラムと MIDI デバイスの間でメッセージとして送受されます。

最も基本的な使い方は、プログラムから MIDI 出力デバイスに MIDI メッセージを送って音を鳴らすことと、MIDI キーボードのような MIDI 入力デバイスから MIDI 情報を送ってプログラムがその情報を受け取ってなんらかの処理を行うということです。

1.2.4 MIDI メッセージの実体

MIDI サウンドのデータの実体は、MIDI のイベントやその他の情報のシーケンスなので、MIDI デバイスを直接制御したいときには、MIDI メッセージをバイト列として MIDI デバイスに送ります。

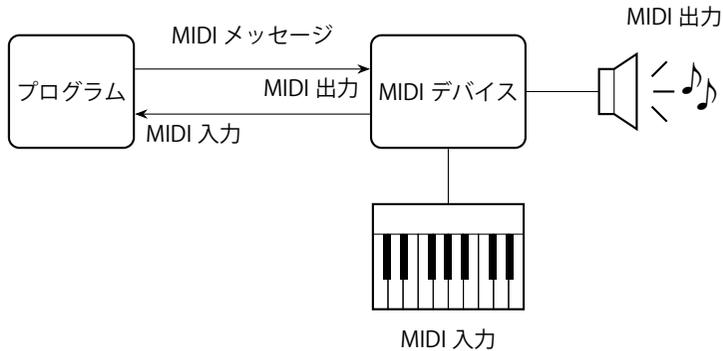


図 1.5 ● MIDI デバイスと MIDI メッセージ

たとえば、音の高さを表すノートナンバーが n の音をチャンネル c で v の強さで鳴らしたいとします。そのときには次の一連のバイトを MIDI 出力デバイスに送ります。

```
9cH, n, v
```

最初の $9c$ の 9 はノートオン（発音）を表します（ H は 16 進数であることを表します）。 c はチャンネル、 n はノートナンバー、 v は音の強さを表します。

1.2.5 ノートナンバー

音の高さを表すノートナンバーは $C-1 \sim G9$ までの音を $0 \sim 127$ までの数値で表し、全体では図 1.6 のように決まっています（音の呼び方はメーカーによっては異なる場合があります）。

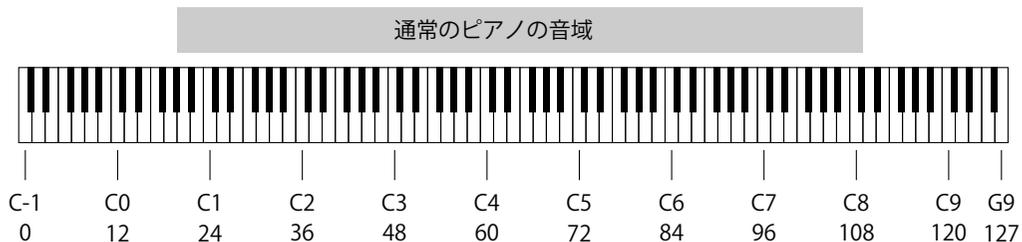
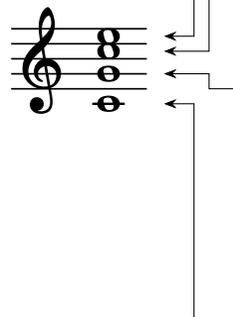


図 1.6 ● MIDI ノートナンバー（全体）

半音を含むオクターブの音は図 1.7 のようになります。



音 (英語)	音	ノートナンバー
E5	ミ	76
D # 5	レ#	75
D5	レ	74
C # 5	ド#	73
C5	ド	72
B4	シ	71
B ♭ 4	シ♭	70
A4	ラ	69
G # 4	ソ#	68
G4	ソ	67
F # 4	ファ#	66
F4	ファ	65
E4	ミ	64
D # 4	レ#	63
D4	レ	62
C # 4	ド#	61
C4	ド	60

図 1.7 ● MIDI ノートナンバー (オクターブ)

ノートナンバーを指定して再生される音は、楽器が音階のある楽器である場合、通常は A = 440 Hz 平均律の音です。鍵盤打楽器以外の打楽器の場合は特定の打楽器の音が鳴ります（例は後の章で示します）。

Note 平均律から外れた高さの音を鳴らすには、あとで紹介するピッチベンドを使います。打楽器の音は一般的には GM 音源と呼ばれる音源の音が基準となります。

たとえば、中央ド（C、ノートナンバー 60）の音をチャンネル 0 で 127（16 進数で 7F）の強さで鳴らしたいときには次のバイトを送ります。

```
90H, 60, 7FH
```

実際にコンピュータで単に MIDI サウンドを鳴らして止めたいときには、ノートオンコマンドで音を鳴らした後に、音を止めたいときにノートオフコマンドのメッセージをデバイスに送ります。

Note ノートオンやノートオフなどの短いメッセージを特にショートメッセージと呼ぶことがあります。ショートメッセージのほかにより長いメッセージもあります。さまざまな MIDI メッセージについては、第3章で説明します。

1.2.6 デルタタイム

デルタタイムは、前のイベントからそのイベントまでの時間を表します。デルタタイムはこの後で説明する可変長の数値で表します。

リアルタイムで MIDI デバイスと制御装置の間でデータを送受する場合には、デルタタイムは事実上無視されることがあります。

1.2.7 可変長表現

状況によって、数値がゼロや一桁の数値であったり、100000 を超えるような大きな数値である場合があります。このようなときに固定長で表現すると、たとえば 4 バイトなら 1 を表現するために「00 00 00 01」と表現しなければならず、3 バイトは無駄に長い情報であるといえます。そこで、数値が小さいとき（127 以下）には 1 バイトで表現し、それ以上の場合はより長いバイト数で表現する方法を数値の可変長表現といいます。

数値の可変長表現は 1 バイトの最上位ビットを除く 7 ビットを数値を表現するのに利用し、最上位ビットは次のバイトもデータバイトである場合に 1 にします。

たとえば、デルタタイムが 2 バイトで「8f00」とであるとすると、8FH は 2 進数で「10001111」なので、これは値が 0b1111（10 進数で 15）で次のビットもデータバイトであることを表します。つまり、デルタタイムの値は「0F00」になります。

1.2.8 GM

MIDI は、インターフェースやメッセージについて規定されていますが、どのような情報を送るとどのような音が鳴るのか（あるいはどのように制御されるのか）という点は具体的に定めていません。そのため、接続した MIDI デバイスによってなる音の種類が変わってしまいます。たとえば、トランペットの音を鳴らそうとしているのに、特定の MIDI デバイスではトランペット以外の音が鳴ってしまうことがあります。

基本的な音色マップとコントロールチェンジなどを規定した MIDI の統一規格として ^{ジェネラル}General MIDI ^{ミディ}があり、広く使われています。General MIDI は通常 GM と省略して表記し、たとえば GM 音源というような使い方をします。本書の後の章のプログラムは、GM に準拠した音源（GM 音源）で音を鳴らすことを前提としています。

1.2.9 標準 MIDI ファイル

MIDI の情報をファイルに保存するときには、通常、標準化された MIDI ファイル形式である標準 MIDI ファイル (Standard MIDI File; SMF) として保存します。

SMF のフォーマットには SMF0、SMF1、SMF2 の 3 種類があります。

- SMF0 (Format 0) は、すべてのチャンネルのデータを 1トラックにすべて保存する形式です。
- SMF1 (Format 1) は、チャンネルデータを複数のトラックに保存する形式です。
- SMF2 (Format 2) は、チャンネルデータを複数のトラックに保存し、複数曲または複数のシーケンスパターンを 1ファイルに収めることができる形式です。

通常使われるのは SMF0 か SMF1 のいずれかです。

SMF についてより詳しいことは第 5 章で説明します。

1.3 Waveform オーディオ

Waveform オーディオはアナログ波形をデジタル化した情報です。

1.3.1 オーディオの形式

音楽や音声などのアナログを、アナログ→デジタル変換によってデジタルに変換することで作成されるオーディオが一般に Waveform オーディオと呼ばれるものです。代表的なものとして、Windows で良く使われる WAV 形式があります (WAVE とも呼びます)。

Windows の Waveform オーディオは RIFF Waveform Audio Format という形式で保存されます。そのうちデータを圧縮していないものを一般に WAV または WAVE といい、この形式のファイルは拡張子を .wav にします。また、このファイルを WAV ファイルまたは WAVE ファイルと呼ぶことがあります。

Waveform オーディオのデータを非可逆圧縮した形式のひとつが MP3 です。圧縮することによってデータ量が大幅に少なくなりますが、Wave を MP3 に変換すると情報の一部が失われるので、変換された MP3 形式を再び WAV 形式に変換しても元の WAV 形式のデータと同じにはなりません。MP3 のファイルの拡張子は .mp3 です。

圧縮されたオーディオ形式には、ほかに Windows Media Audio (WMA) や AIFF (Audio Interchange File Format) などがあります。

Note より詳しくいうと、これらのオーディオの形式は、保存形式や圧縮形式などの組み合わせです。各形式の正確な定義はそれぞれの仕様書を参照してください。

1.3.2 オーディオの生成と合成

プログラムで任意の Waveform オーディオを生成することができます。Waveform オーディオを生成するには、希望する周波数、波形、振幅の波のデータを作り、ストリームとして出力に送ります。複数の波形を合成することもできます。

Waveform オーディオの生成や合成については第 6 章と第 7 章で詳しく取り上げます。