



第5部

Sound ライブラリ編

この章では、サウンドを扱うライブラリ全般について、今まで説明しきれなかった部分（サウンド機能のハードウェア仕様や、サウンドデータを作成するツールの使用法等）をまとめて解説します。

目次

1	Dreamcast のサウンドシステム	4
1.1	Dreamcast サウンド基本仕様	5
1.2	サウンドの基礎知識	6
2	Dreamcast のサウンド概要	9
2.1	サウンドの種類	9
2.2	サウンドライブラリで使用するファイルフォーマット	10
3	サウンドライブラリについて	12
3.1	サウンドライブラリ の概念	12
3.1.1	プラットフォーム構造	12
3.2	サウンドライブラリ上での基礎概念	13
3.2.1	データ (Data)	13
3.2.2	バンク (Bank)	13
3.2.3	マルチユニット (Multi Unit)	14
3.2.4	マルチユニットファイル (Multi Unit File) とバンクファイル (Bank File) の個々の使い分けについて	14
3.3	ホストコマンド (Host Command) とは	17
3.3.1	Host Command の流れ	17
4	Dreamcast で利用できる音源形式	19
4.1	再生方法の種類	19
4.2	MIDI 音源	20
4.2.1	MIDI 音源を利用するときに使用するバンクの種類	21
4.2.2	MIDI 音源での再生	22
4.3	One Shot 音源	23
4.3.1	One Shot 音源を利用するときに使用するバンクの種類	23
4.3.2	One Shot 音源でコントロール可能なパラメータ	24
4.3.3	One Shot 音源の再生	25
4.4	PCM Stream 音源	26
4.4.1	PCM Stream 音源で使用可能な PCM Encode 形式	26
4.4.2	PCM Stream 音源でコントロール可能なパラメータ	27
4.4.3	PCM Stream 音源での再生	28
4.5	GD-DA 音源	31
4.5.1	GD-DA 音源でコントロール可能なパラメータ	31
4.5.2	GD-DA 音源を利用するときに使用するファイルの種類	32
4.5.3	GD-DA 音源の再生	32
5	FX モジュール	34
5.1	FX とは	34
5.2	FX の種類	34
5.3	FX を利用する際に使用するバンクの種類	35
5.3.1	FX Program Bank	35
5.3.2	FX Output Bank	35
5.4	発音中の FX Control の注意事項	35
5.5	GD-DA 音源に対しての FX Control	35
6	その他の注意事項	36
6.1	ボリュームの上げすぎによる音割れの発生	36
6.2	無駄な Host Command をなるべく減らす	37

6.2.1 <i>sdMidiPlay()</i> 関数・ <i>sdShotPlay()</i> 関数などで発音するごとに一定のボリューム・Panpot のバランスの設定を行わない	37
6.2.2 音源のパラメータのリセットは <i>Reset</i> 関数を利用する	37
6.3 Stereo と Monaural 設定での注意事項	38
6.3.1 なぜ Stereo モードと Monaural モードで音量のバランスが変化するか？	38
6.3.2 実際の対処方法	39
6.4 発音中のバンク・マルチユニットのダウンロードについて	39
6.4.1 発音中のブロックは操作しない	39
6.5 Memory Block 転送について	40
6.5.1 CPU 上での CPU 転送と DMA 転送の方法	40
6.5.2 DMA を利用した Memory Block 転送の注意点	40
6.6 メモリマッピングについて	41
6.6.1 ミドルウェア使用時のメモリマッピング	41
6.6.2 可能な限り Sound Memory Mapping は Clear にする	41
7 付録	42
7.1 Dreamcast シーケンサ仕様	42
7.1.1 Dreamcast MIDI メッセージリスト	42
7.2 関数一覧	49
用語集	53

1 Dreamcast のサウンドシステム

Dreamcast のサウンド再生は、大きく分けて 4 つの方法があります。

出力は、最終的にスピーカから発声されますが、その再生の処理経路を図にまとめると、次のようになります。

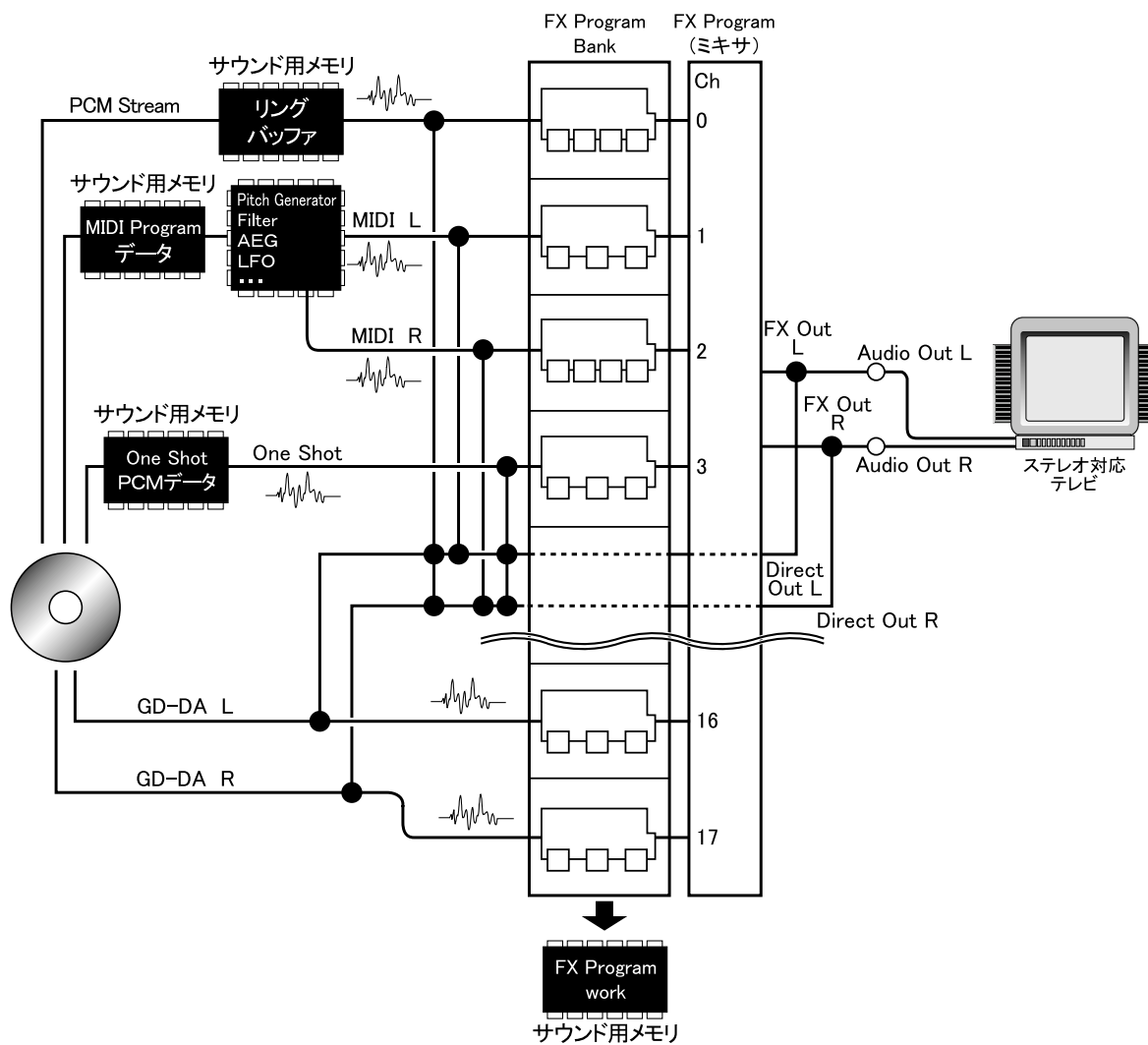


図 1 - 1 スピーカからの再生の処理経路

1.1 Dreamcast サウンド基本仕様

Dreamcast のサウンドの基本仕様は、以下のとおりです。

使用チップ AICA (AdvancedIntegratedCapableAudio)
 同時発音数 64 スロット
 DSP ProgramCode128step
 音色データ 1 音色当り 65534 サンプルまで

1 スロット当り

- ADPCM デコード可能
- DSR エンベロープの設定が可能
- TVF (TimeVariantFilter) 使用可能

音源の特徴

- DA
- 44.1KHz
- 16Bit ステレオ
- リトルエンディアン
- PCM
- DSP
- DSP MicroProgram AICA チップ内
- DSP CoefcientTable AICA チップ内
- DSP WorkArea SoundRAM 内

• ADPCM (4 モード) の使用内容

	OneShot	PCMStream
Linear		
ADPCM Loop		×
ADPCM NonLoop		×
ADPCM	×	

再生の種類

- MIDI48 音
- OneShot 8 音
- PCMStream 8 音
- GD-DA 別扱い

上記の各値は初期値として設定されていますが、ライブラリにより変更が可能です。ただし、同時発音数 64 音を超えることはできません。

1.2 サウンドの基礎知識

(1) サウンドのスペックを決める

サウンドの種類にはそれぞれに適した再生方法があり、どの音をどの方法で鳴らすかということ、あらかじめ十分に検討した上で制作に入らなければなりません。その場合、基本的に考えなければならないことを簡単にまとめます。

スペック決定は、サウンドの担当者だけでなく、企画担当や、プログラム、グラフィックの担当者などプロジェクトスタッフ全員で、それぞれの再生方法の特徴や短所などを踏まえながら決めていきます。状況に応じて、最も適した再生方法を選択して下さい。

(2) ゲームやコンテンツのジャンル

例えば、アドベンチャーゲームなのか、アクションゲームなのかといったような、ジャンルの持つ特質です。

アクションの場合は、何種類もの SE をランダムに高速に再生しなければなりません。そこで、サウンド用メモリの多くを SE 用に割り振る必要があります。

一方アドベンチャーの場合は、高音質で長い SE を多用する機会が多くなります。そこで、サウンド用のメモリを BGM 用に多く割り振り、プレイ中のシーンに必要な SE などは、メモリに読み込まないようにするなどの配慮が必要です。

(3) 音質と容量

音質と容量とは非常に密接な関係があります。ここでいう容量とは、GD-DA 自体の容量と、サウンド用メモリの容量のどちらも考慮する必要があります。一般に、音質を良いものにしていくとそれだけ容量を必要とします。GD-DA の場合は最高の音質ですが、時間的な制限があり、一曲の長さや曲数を考えなければなりません。

MIDI や One Shot の場合は、サウンド用メモリの使い方が重要になってきます。詳しくは後述しますが、サンプリングした波形データをサウンド用メモリに読み込まなければならないため、同時に再生すべき波形データ（SE や楽器音）の数や長さによって、音質を下げていかなくてもなりません。逆に、音質を上げるためにサウンド用メモリに読み込む波形データを最低限にするような考慮も必要です。

(4) 異なる再生方法の組み合わせ

「BGM を再生しながら複数の SE を再生する」「BGM を再生しながら長いナレーションを再生する」など、ゲームによっていろいろなシチュエーションが考えられます。これらは、再生方法をうまく組み合わせることによって同時再生の問題が解決できます。

再生方法の向き不向きに加え、その再生方法による CPU の負荷や、他のデータ（グラフィックなどを含む）へのアクセス、同時に鳴らさなければならない曲数、音数など、プログラムの要素も組み合わせを考える上で重要なことです。

それでは、具体的な組み合わせの例をいくつか挙げてみます。

a. シーンとシーンのつながりをスムーズにさせたい。

例えば、あるシーンから次のシーンに移るときを考えてみます。「サウンドのデータを読み込む時間をできるだけ減らしてシーンのつながり目をスムーズにさせたい」といった場合、MIDI の波形データとシーケンスデータをすべて入れ替えるには時間が掛かり過ぎてしまいます。こういった場合、次のような方法が考えられます。

- BGM を GD-DA にする
- BGM を PCM Stream にする
- 1 回のデータ転送で再生できる曲数を増やす
- メモリ上の 1 曲分のデータだけを入れ替える

大量のメモリを必要とする One Shot にも同様のことがいえます。

b. BGM と効果音をループさせたい。

BGM に加えて波の音や機械音などの環境音（SE）を BGM のようにループさせて使用する場合を考えてみます。

長さの違う複数の BGM を同時再生する場合に、PCM Stream を利用するのはふさわしくありません。

環境音は PCM Stream、BGM は MIDI というように切り分けたり、両方を MIDI にするなどの工夫をする必要があります。

c. ソフトのメイン処理にとっても時間が掛かる。

3D グラフィックの計算やリアルタイムシミュレーションなど、メインの処理が非常に大変な場合を考えます。

このようなときには、なるべくメイン CPU に負荷をかけないように、サウンドプロセッサが独立して発生できる音を選ぶ必要があります。このような場合、PCM Stream は使用せず、GD-DA や MIDI を用いるようにします。

d. GD-ROM の読み込みが頻繁にある。

ムービーの再生が頻繁に行われる場合や、グラフィックのデータをこまめに GD-ROM から読み込むようなプログラムの場合を考えます。

GD-DA や PCM Stream の再生は、頻繁に GD-ROM をアクセスしますので、メインプログラムと GD-ROM の読み込みで競合が起こることが考えられます。このような場合は、MIDI を用いるようにするとよいでしょう。

- e. GD-ROM に納めるべきデータがたくさんある。

ムービーや画像データが非常に多く、GD-ROM の容量を多く必要とするプログラムの場合を考えます。

この場合、GD-DA やサンプリングレートの高い PCM Stream は、より多くの GD-ROM の容量を必要としますので、サンプリングレートを下げたり、短めの曲にするようにします。

2 Dreamcast のサウンド概要

ここでは、Dreamcast 用アプリケーションのサウンドを作成する場合の概論などについて説明します。

2.1 サウンドの種類

Dreamcast 用のゲームやマルチメディアコンテンツを制作するにあたって、そこで使われるであろう音源全般に関して、用語の統一を兼ねて説明します。

サウンドは大きく分けると、「BGM」「SE (Sound Effect) 」「Voice」の3つに分けられます。

サウンドの関連に関しては、以下の図を参照して下さい。

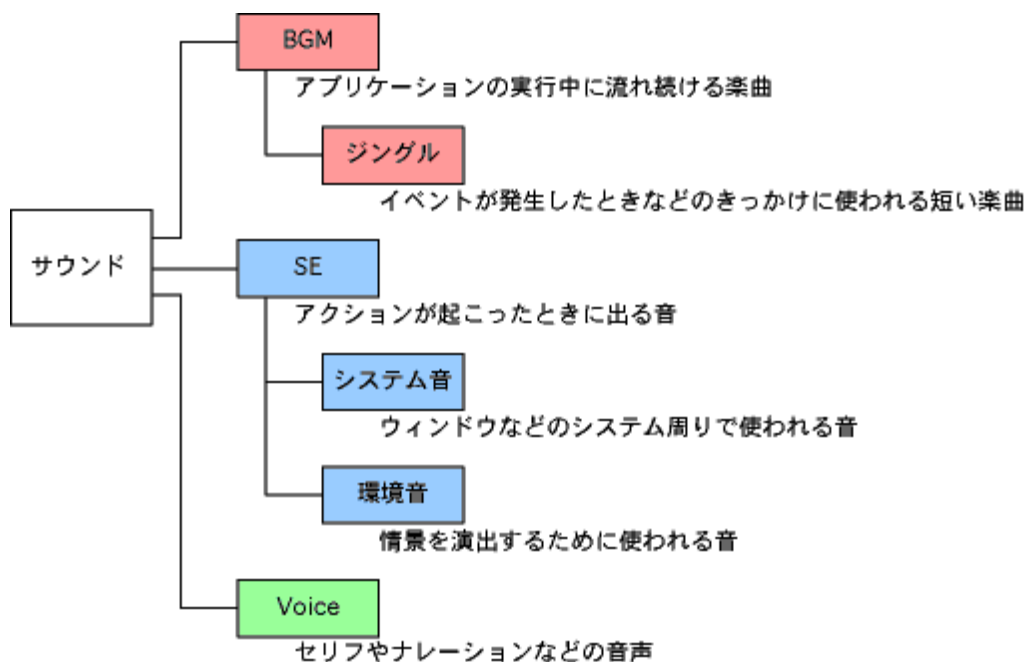


図 2 - 1 サウンドの種類の関連図

- 「BGM」
サウンドのなかで、不可欠なのが「BGM」です。BGM は基本的にアプリケーションの実行中に再生され流れ続けます。
緊迫感や安心感などの臨場感を高める働きをしたり、グラフィックの表現力を高める手助けになるものです。一般には、長めの楽曲が使用されることになります。
- 「ジングル」
BGM の中で、何らかのイベントやアクションに伴なって再生される、比較的短くて単発的なものを「ジングル」と一般的に呼ばれておりますので、ここでも呼ぶことにします。
例えば何かが始まるときや、アクションゲームでアイテムを取得したときなどに鳴らすファンファーレ的なものです。

- 「SE (Sound Effect) 」
画面上のオブジェクト (物や人や道具や背景) それぞれに対して、それらが変化するとき (移動したり出現したり、色が変わったり爆発したりするかもしれません)、それに付随して鳴らす音のことを「SE (Sound Effect) 」 (効果音) と呼びます。
- 「システム音」と「環境音」
「SE」の中には、自動車が走ったり扉が開いたり、モンスターが鳴いたりといった、それら自身を持つ固有の音の他に、「システム音」と「環境音」があります。システム音は、ユーザーがカーソルを動かしたり、選択肢を決定したりといった、ゲーム操作上の行動に対して鳴らす音です。メーカーによっては「ウィンドウ音」とも呼ばれる場合もあります。一方、環境音は、特に動きが与えられていなくても背景に描かれていたり、見えていなくても状況を説明するような音 (風の音、川のせせらぎ、雑踏音など) のことをいいます。
- 「ボイス」
BGM と SE 以外の音源として、キャラクタの台詞や物語のナレーションなどの人間の声を使うものを、特に「ボイス」と呼んでいます。
位置付けとしては SE のひとつとしてもとらえられますが、ゲームシステム上、大抵は普通の SE と別の再生方式が使われるのが一般的なので、あえて別のものと考えことにします。
- 「ループ」の概念
これは「一つの波形データを切れ間なく再生し続ける」という、サンブラ機などで使われている言葉です。
ゲーム音楽の中では、繰り返し再生するという意味で使われます。ゲームでは、会話をしたり戦っている時間は、ユーザー次第でいつまで続くかわかりません。そのために、BGM や環境音を終わらせないで永遠に切れ間なく繰り返し再生する必要があります。楽曲や SE がループできるかどうかは、再生方法によって異なることを覚えておいて下さい。

2.2 サウンドライブラリで使用するファイルフォーマット

ここでは、サウンドライブラリで使用するファイルフォーマットについて、簡単に説明します。

テクニカルサポートセンターに対して質問する場合も、この名称で現状の説明をして下さい。

- DRV File (sound DRiVer)
サウンドドライバのファイルです。
サウンドドライバをサウンドメモリ上にダウンロードして実行しなければ、何も音を鳴らすことはできません。ファイルの拡張子は.DRV です。
- MLT File (MuLTi unit)
マルチユニットファイルです。
おそらく1つのプロジェクトに最低1つのファイルが存在すると思われます (サウンドメモリのマッピングはマルチユニットファイルを利用しないとできないため)
ファイルの用途は単にサウンドメモリのマッピングの為だけ、1ステージ分の音声や効果音など用途はさまざまです。ファイルの拡張子は.MLT です。
- MPB File (Midi Program Bank)
MIDI Program バンクファイルです。
MIDI 音源で音声を再生する場合には必ず必要になります。通常は実際の曲や効果音の演奏情報が入った MIDI Sequence Bank も組みになって渡されます。ファイルの拡張子は.MPB です。

- **MDB File (Midi Drum Bank)**
MIDI Drum バンクファイルです。
MIDI Program Bank File と同様に、MIDI 音源の音色のファイルです。Drum や Percussion など音を鳴らしたら途中で再生を止めたりしない用途の楽器音や効果音に使用します。ファイルの拡張子は.MDB です。
- **MSB File (Midi Sequence Bank)**
MIDI Sequence Bank File です。
MIDI 音源での演奏情報が入っています。通常、必要な音色を含んだマルチユニットファイルもしくは MIDI Program Bank File も渡されます。ファイルの拡張子は.MSB です。
- **OSB File (One Shot Bank)**
One Shot Bank File です。One Shot 音源で音声を再生する場合はこのファイルが必要です。
MIDI 音源と違い、One Shot 音源はこのファイル単体で音声の再生をすることが可能です。ファイルの拡張子は.OSB です。
- **P16・P08・P04 File**
PCM Stream 用のヘッダー無しの Raw 波形ファイルです。それぞれ 16 bit Linear Encode 形式・8 bit Linear Encode 形式・4 bit ADPCM Encode 形式になります。SCAT (Sound Creator's Assistant Tools) で作成できるのは Monaural 音声のみです。ファイルの拡張子はそれぞれ.P16、.P08、.P04 です。
- **FPB File (Fx Program Bank)**
FX Program Bank File です。
FX を利用するには、後述のマルチユニットファイルとして渡されることが多いでしょう。サウンドメモリ上には1つしか存在できません。ファイルの拡張子は.FPB です。
- **FOB File (Fx Output Bank)**
FX Output Bank File です。
DSP の演算結果の Digital Audio としての出力パラメータ (DSP の演算結果に対する Volume や Panpot の設定群) が格納されています。これも FPB File と同様にマルチユニットファイルで渡されることが多いでしょう。
サウンドメモリ上には1つしか存在できません。ファイルの拡張子は.FOB です。
- **DA File (Digital Audio)**
Digital Audio File です。
S ヘッダー無しの Raw 波形ファイルです。GD Workshop で GD Emulation を利用するときに使います。CD Craft では WAV File が必要になりますので、同等の波形の WAV File も用意してもらわなければなりません。ファイルの拡張子は.DA です。
- **WAV File (microsoft WAVe)**
一般的な Microsoft 社のファイル形式です。
内部的には Stereo (16 bit Linear Encode 形式で、サンプリングレートは 44.1KHz) です。つまり CD の音質のものを使用します。GD-DA Track を CD Craft にて作成する場合に必要なファイルです。ファイルの拡張子は.WAV です。

3 サウンドライブラリについて

ここでは、サウンドを扱うライブラリを全般にわたり説明します。

3.1 サウンドライブラリの概念

3.1.1 プラットフォーム構造

サウンドライブラリのプラットフォーム構造について説明します。

(1) ハードウェアの構成

Dreamcast 本体でのサウンド関連のハードウェア部分は、下記のように構成されます。

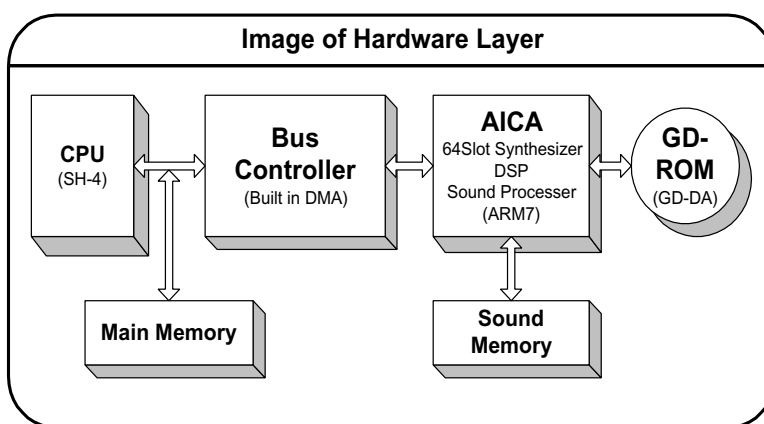


図 3 - 1 ハードウェア構成図

(2) ソフトウェアの構成

ソフトウェアの構造を大きく分けると、サウンド関連の全ハードウェアリソースを管理するサウンドドライバと、それをコントロールするサウンドライブラリの2つのレイヤーに分かれます。GD-DAの再生では、一部GD-FSライブラリを利用します。

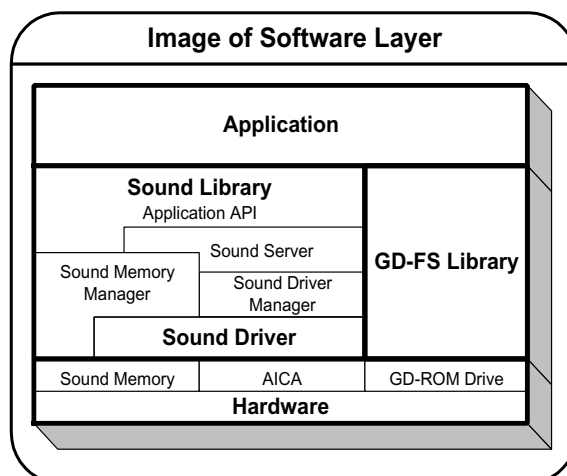


図 3 - 2 ソフトウェア構成図

アプリケーションと各ライブラリはCPU上で、サウンドドライバはサウンドプロセッサ (ARM7) 上で動作することになります。

3.2 サウンドライブラリ上での基礎概念

サウンドライブラリでは、下記のような概念を定義しています。

3.2.1 データ (Data)

データは1オブジェクトのことで、サウンドの最小単位になります。

例えば、1音色(波形とパラメータ群)はOne Shot Data やMIDI Program Data といい、1曲(MIDI シーケンス)はMIDI Sequence Data という概念の定義です。

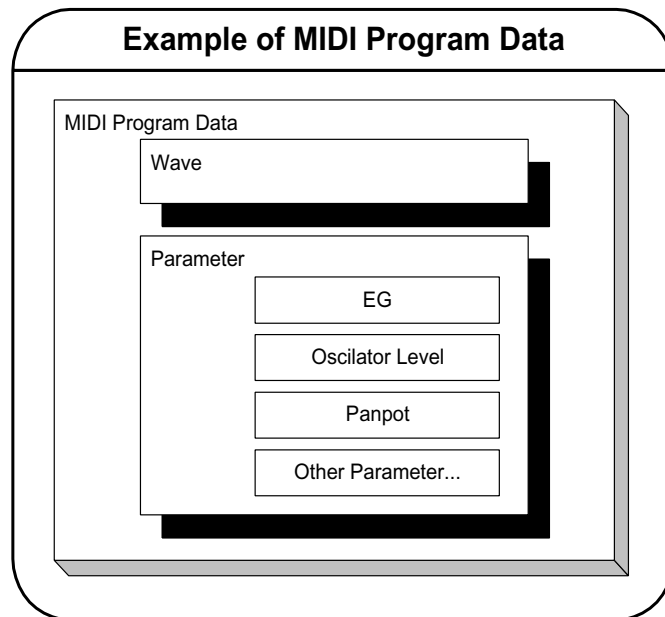


図 3 - 3 MIDI Program Data 構成図

3.2.2 バンク (Bank)

1つ以上のデータの集まりがバンクという概念です。

MIDI Program Bank は、1つ以上の MIDI Program Data の集まりをいい、つまり1つの音色セットになることを意味します。

MIDI Sequence Bank は、1つ以上の MIDI Sequence Data の集まりという概念定義になります。また、バンクはプログラマが扱うファイルの最小単位でもあります。

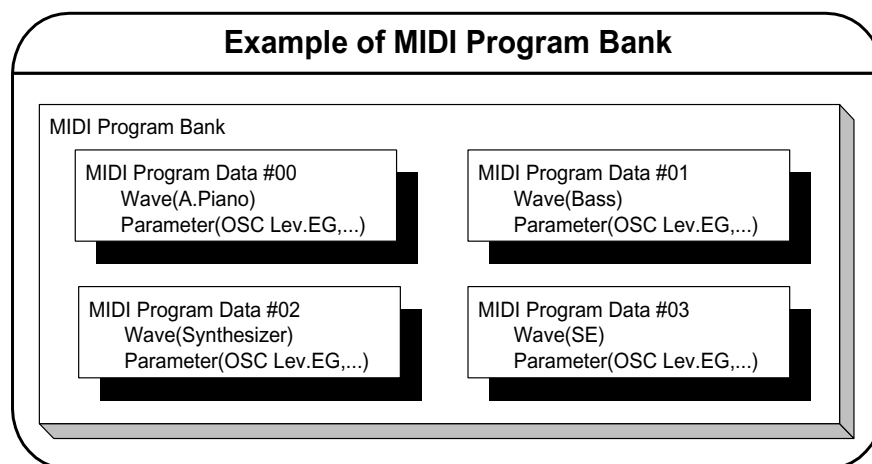


図 3 - 4 MIDI Program Bank 構成図

3.2.3 マルチユニット (Multi Unit)

マルチユニットは、Sound Memory Mapping 情報やサウンドメモリへのダウンロード情報、そしてダウンロードする必要のあるバンクの内容といった3つの情報を、記録することができます。

3つの情報を全て含めても構いませんし、ダウンロードするバンクの内容とそのダウンロード先の情報だけ、または Sound Memory Mapping 情報のみにしても存在することが可能です。

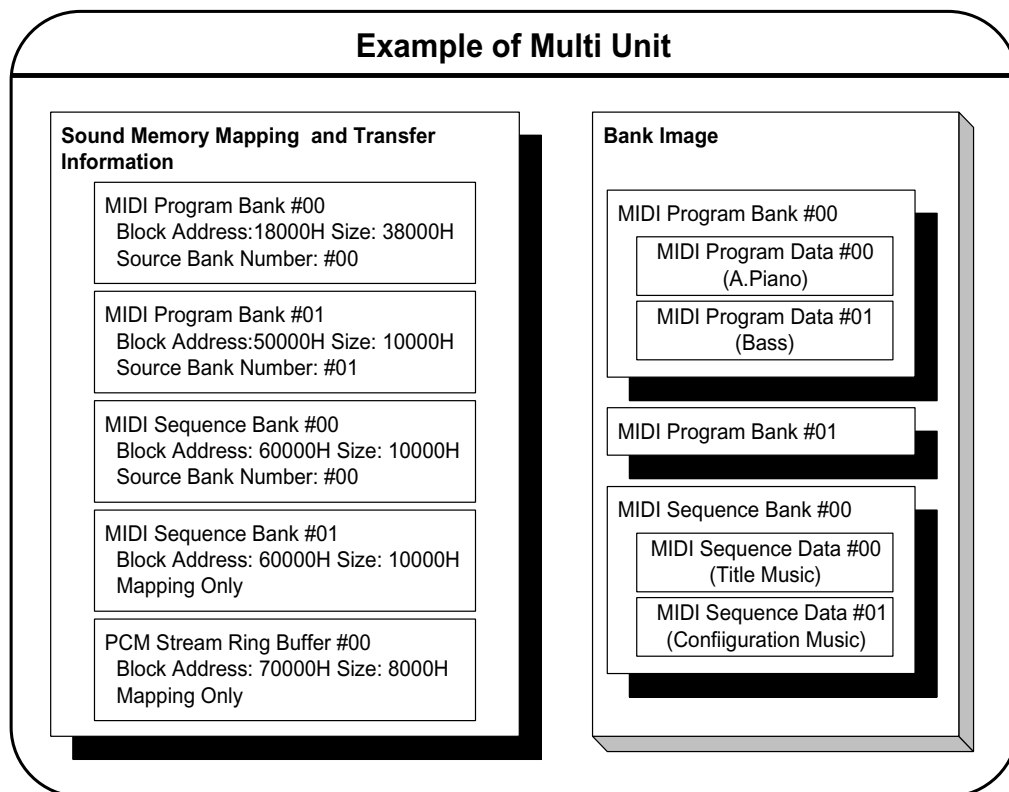


図 3 - 5 Multi Unit 構成図

3.2.4 マルチユニットファイル (Multi Unit File) とバンクファイル (Bank File) の個々の使い分けについて

(1) マルチユニット形式でファイルを持つメリット

マルチユニットファイルは、複数のバンクをまとめてファイルにすることができ、かつファイル内に実際のダウンロード先のブロック情報を、記録することができます。

例えば、MIDI 音源を利用する場合は MIDI Program Bank と MIDI Sequence Bank をダウンロードする場合が大半を占めますが、バンク形式でファイルを持っている場合は2回ファイルをダウンロードしなくてはなりません。

こういった場合でも、マルチユニット形式でファイルを持っておくと、プログラム制御でダウンロード先を指定しなくても、内部に記録されている情報をもとにしてバンクの内容をダウンロードする方法が可能になります。

(2) バンク形式でファイルを持つメリット

バンク形式でファイルを持つことのメリットは、ダウンロード先のブロック番号をプログラムで制御し指定できることです。

例えば、格闘ゲームでキャラクタごとの音声や効果音の One Shot Bank を作成しておき、ゲームの進行状況に応じてプレイヤー 1・プレイヤー 2 のキャラクタに対しそれぞれに対応するブロックへダウンロードする...といった方法が可能になります。

マルチユニット形式でファイルを持つ場合は、バンクのダウンロード先のブロック番号はマルチユニットファイル内に指定されているので、バンクファイル形式のときのようなプログラムで制御する指定ができません。

- (3) マルチユニットファイルとバンクファイルを要求仕様に応じて使い分ける
状況に応じて、マルチユニットファイルとバンクファイルを使い分けることが可能です。

例えば、格闘ゲームの場合は最低限でも下記のようなアイテムが含まれるファイルが必要になります。

- プレイヤー 1 用音声・効果音（キャラクタごとに異なる）
- プレイヤー 2 用音声・効果音（キャラクタごとに異なる）
- その他効果音（ポーズ音など）
- BGM（ステージごとに異なる）

この場合、各キャラクタの声や効果音は One Shot 音源を利用することにします。そしてファイルは One Shot Bank 形式で、サウンドクリエイターから受け取ります。ゲーム中は、プレイヤーの選択により対応するキャラクタの One Shot Bank ファイルをプレイヤー 1・2 に対応するブロックにダウンロードします。

バンクファイルにする利点は、このような実際に使用するバンクのダウンロード先のブロックが固定ではないときに、任意に指定できることです。

また、BGM のようにあらかじめダウンロード先のブロック番号が決まっている場合は、マルチユニットファイルにしておくとし、ステージに適したマルチユニットファイルのファイル名だけをサウンドクリエイターから教えてもらえばよいことになります。

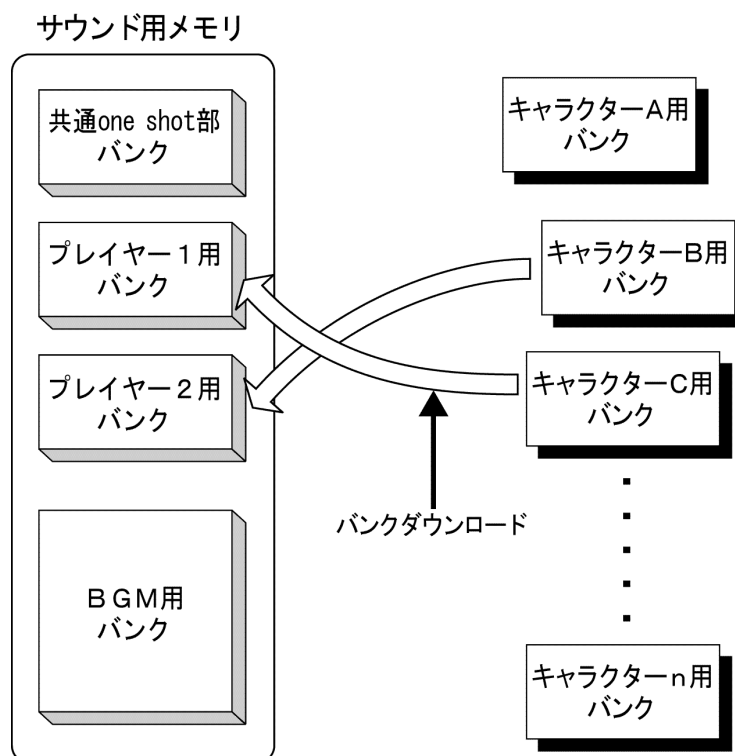


図 3 - 6 使い分けの例

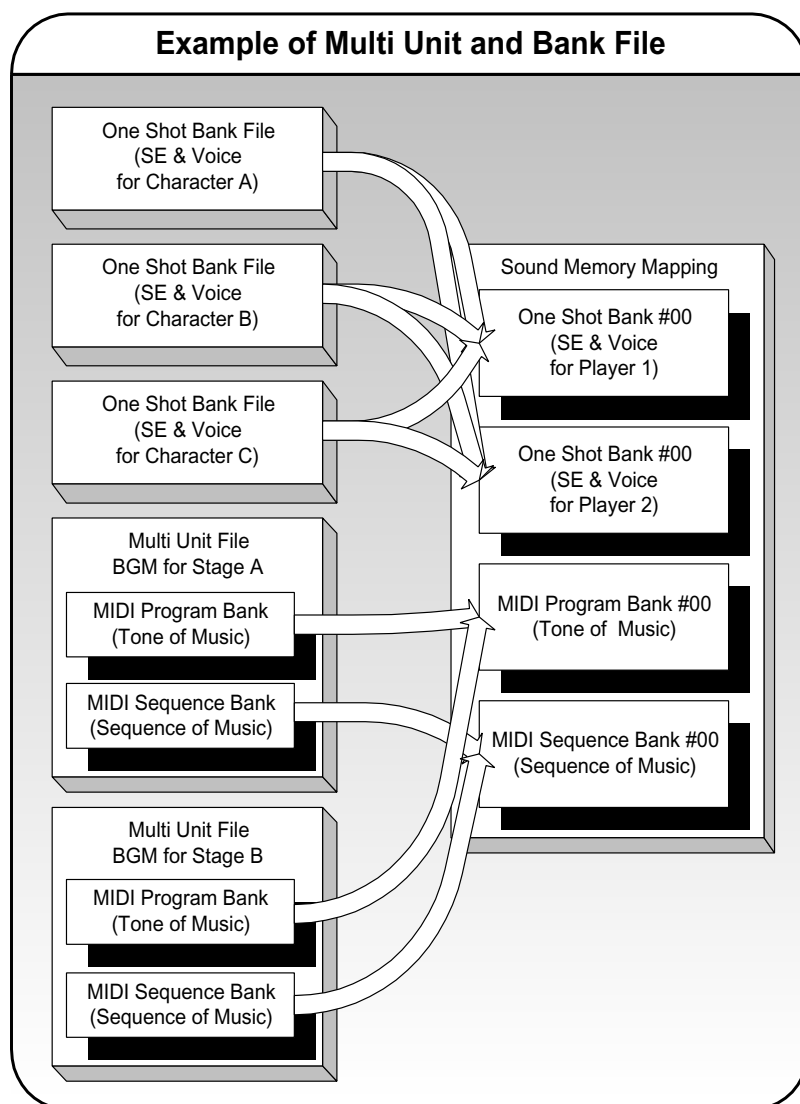


図 3 - 7 マルチユニットファイルとバンクファイルの使い分け

3.3 ホストコマンド (Host Command) とは

Host Command とは、サウンドドライバへの命令のパケットのことを指します。

注意 これを送らないと音声を再生することも、Volume・Panpotなどのコントロールを行うこともできません。

3.3.1 Host Command の流れ

Host Command はアプリケーションからサウンドライブラリ内の関数を実行することによって発生し、サウンドライブラリ内の Work Memory (Host Command Buffer) でいったんバッファリングします。そしてサウンドライブラリ内のサーバによってまとめてサウンドドライバに送信します。

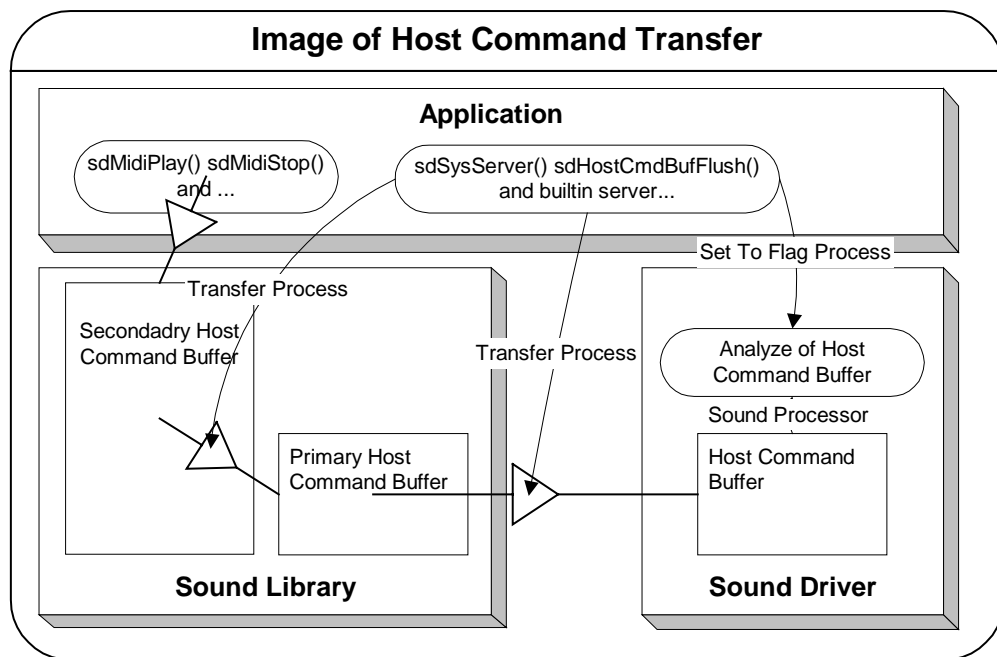


図 3 - 8 Host Command Transfer のフロー

Host Command Buffer はサウンドライブラリ内に 2 種類あり、サウンドドライバに 1 種類あります。

(1) Host Command Buffer について

このうちサウンドライブラリ内にある 2 種類の Host Command Buffer のそれぞれの説明を下記に記述します。

● Secondary Host Command Buffer

アプリケーションからサウンドライブラリの関数を実行することで発生した Host Command は、いったんこのバッファに貯えられます。発生する Host Command の個数はサウンドライブラリの関数によって異なりますが、Host Command が発生する関数のほとんどが、1 つだけ Host Command を発行します。

このバッファで貯えられる Host Command の個数は、サウンドライブラリのマクロ定数として `SDD_HOST_CMD_2ND_BUF_NUM` という名称で定義されています。

● Primary Host Command Buffer

Secondary Host Command Buffer に貯えられた Host Command をサウンドドライバに送信するときに、いったん整理するために使用されるバッファのことです。

アプリケーションから、サウンドライブラリの関数を実行することによって発生した Host Command は、下記の順序で処理されます。

- (2) サウンドライブラリのサーバによって、Secondary Host Command Buffer にバッファリングされた Host Command を Primary Host Command Buffer に送信します。
 - a. 次にサウンドドライバに全ての（または送信可能な量だけ）Host Command をサウンドドライバに送信します。
 - b. サウンドドライバへ送信が完了したら、サウンドドライバに Host Command の解析要求を出します。
 - c. サウンドドライバは、4ms のタイミングで Host Command Buffer の解析要求を監視しています。監視中に Host Command の解析要求があった場合は、Host Command の解析処理に入ります。

注 意 Host Command の種類や量によってサウンドドライバの負荷が変わり、場合によっては次の Host Command の解析要求に対し応じられなかったり、実際の演奏に支障が出る場合があります。

4 Dreamcast で利用できる音源形式

Dreamcast 上では、次の 4 種類の音源形式が利用できます。

- MIDI 音源
- One Shot 音源
- PCM Stream 音源
- GD-DA 音源

GD-DA 音源を除いた残りの 3 つの音源は、同時に複数個の存在が可能で 1 音源を 1 ポートと数えます。

それぞれの音源は、最大で MIDI 音源は 8 個、One Shot 音源は 64 個、PCM Stream 音源は 16 個のポートを持つことが可能です。

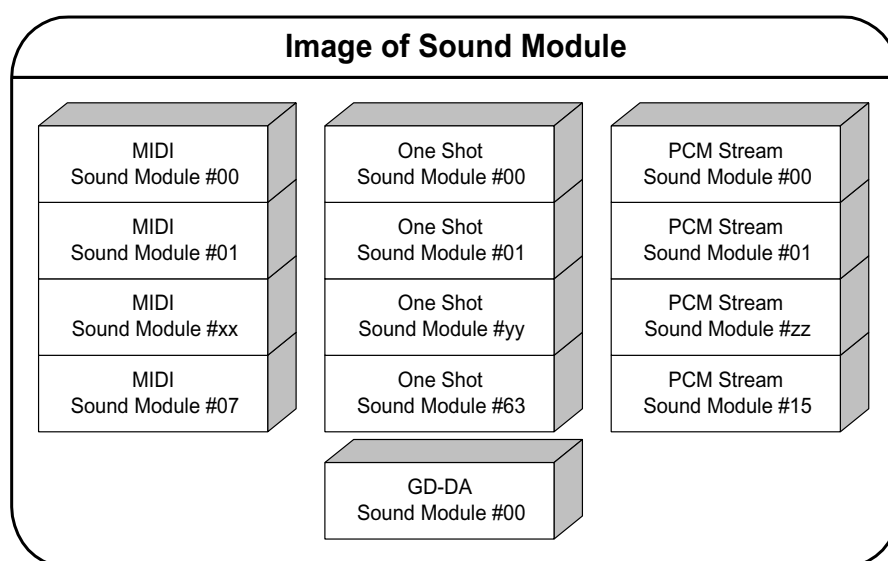


図 4 - 1 モジュール関連図

4.1 再生方法の種類

音声の再生方法は、作曲の方法やデータの作り方に関わる大切な要素です。

再生方法の種類により長所や短所がありますので、それらを把握した音作りが必要です。

名 称	解 説	用 途
GD-DA	GD-ROM の GD-DA を再生する	「BGM」や「ナレーション」などに利用
One Shot	サウンド用メモリに One Shot Bank をロードし、ループなしで再生する	「ジングル」や「SE」などで利用
PCM Stream	サウンド用メモリにリングバッファ (RingBuffer) を確保し、PCM や ADPCM データをロードして再生する	「ボイス」や長めの「ジングル」「BGM」などに利用
MIDI	サウンド用メモリに MIDI Program Bank (音色データ) と MIDI Sequence Bank (シーケンスデータ) をロードし再生する	「BGM」や「ジングル」などに利用

再生方法	再生時間	音質	メインCPUへの負荷	サウンドCPUへの負荷	使用メモリ場所		ループ再生	再生中に変換可能なパラメータ				用途				
					メインメモリ	サウンドメモリ		音声	ピッチ	パン	テンポ	BGM	SE	ジングル	環境音	ボイス
G D - D A	最長112分	CD音質	無	無	×	×	×	○	×	○	×	○	×	×	×	○
ワンショット	最長6秒	CD音質からAMラジオ音質	無	中	×	○	○	○	○	○	×	×	○	○	×	△
PCMストリーム	無制限	CD音質からAMラジオ音質	大	無	○	△	○	○	○	○	×	○	○	×	○	○
M I D I	無制限	CD音質よりは悪い	無	大	×	○	○	○	○	○	○	○	○	○	△	×

注意：上記の場合は、プログラムの種類や音の作り方により、異なる場合があります。

4.2 MIDI 音源

パソコンなどの DTM (Desk Top Music) 上で使用されている音源と同様のものです。

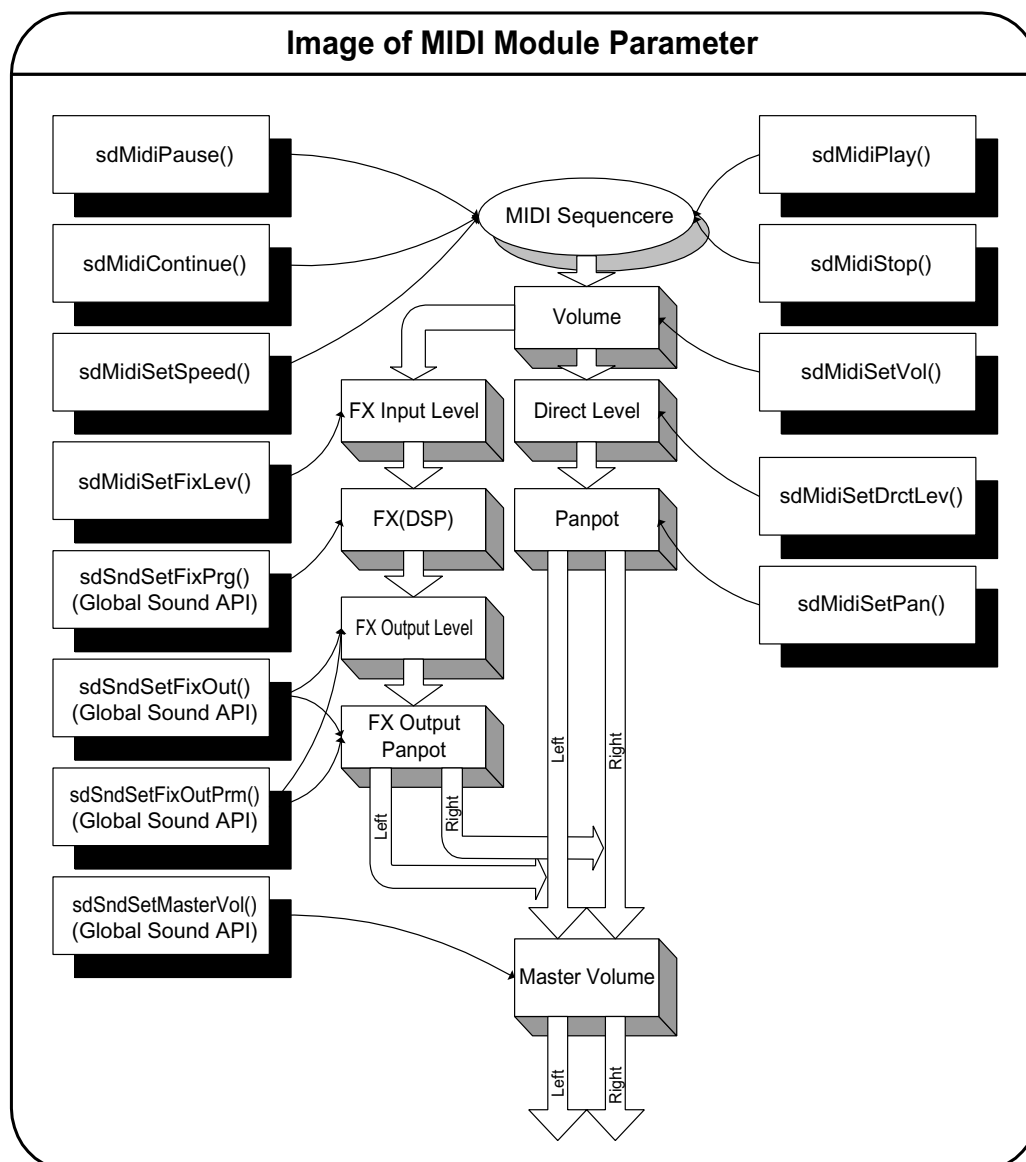


図 4 - 2 MIDI 音源で操作可能なパラメータと信号の流れ

4.2.1 MIDI 音源を利用するときに使用するバンクの種類

MIDI 音源を利用するときに通常では、次の3種類のバンクを扱うことになります。

(1) MIDI Program Bank

MIDI 音源で使用する音色の Bank 形式です。波形と音の減衰具合などのパラメータをもっています。

1つの音色のことを MIDI Program Data といい、1つの MIDI Program Bank 内に最大 128 の音色を持つことが可能で、サウンドメモリ上で最大 16 種類の MIDI Program Bank を存在させることが可能です。

(2) MIDI Drum Bank

MIDI 音源で使用する音色の Bank 形式です。

MIDI Program Bank との違いは、ドラムやパーカッションなどの音を鳴らしたときに、自然に減衰する音に特化したパラメータに調整されています。

サウンドメモリ上には、最大 2 種類の MIDI Drum Bank を存在させることが可能です。

(3) MIDI Sequence Bank

鍵盤を“押した・放した・音色を切り替えた”といった情報 (MIDI Event) と、それをいつ行ったかという時間の情報を併せて保存した Bank 形式です。これは、Windows などでもよく使われている SMF (Standard MIDI File) と同様の役目を持ちます。

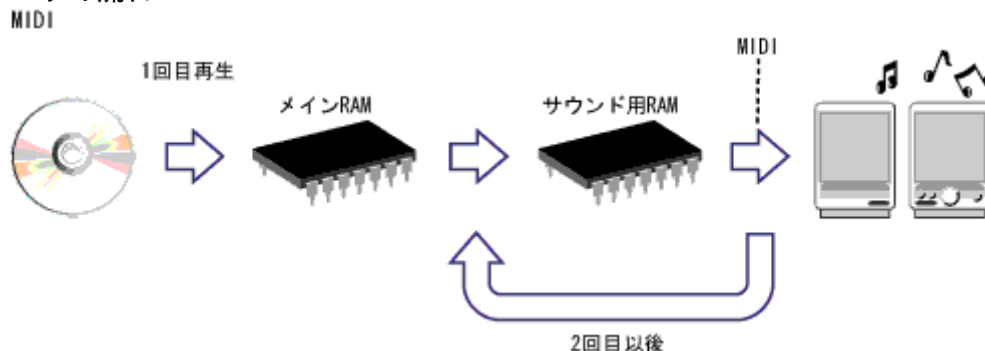
SMF との違いについては、組み込み機器に特化した SEGA 独自の圧縮などが施されている点です。

MIDI 音源は、主に曲として使用することが多いのですが、音程感のある効果音 (例えば、メニュー画面のカーソル音やレベルアップ時のファンファーレなど) や長い周期で音色の変化がある音 (例えば海辺で聞こえる波の音) にも使用することが可能ですが、One Shot 再生と組み合わせで使用する場合は、設計をきちんと行う必要があります。

ここでは、1つの演奏情報を MIDI Sequence Data といい、1つの MIDI Sequence Bank 内に最大 128 種類の MIDI Sequence Data 持つことが可能で、サウンドメモリ上には最大 16 種類の MIDI Sequence Bank を存在させることが可能です。

4.2.2 MIDI 音源での再生

● データの流れ



サウンド用メモリに MIDI の音色セット (MIDI Program Bank) と MIDI シーケンスセット (MIDI Sequence Bank) を読み込み、メインプログラムからシーケンス開始指示を受け取ると再生されます。

● 特徴

再生時間	短いものから長いものまであります。
音質	工夫次第ですが、一般的にはそれほど音質は高くはありません。
データ量	PCM Stream や GD-DA に比べ、非常に少ないデータ量になります。
CPU の負荷	メイン CPU にはほとんど負荷は掛かりませんが、サウンドプロセッサにはかなりの負荷が掛かります。
必要なメモリ量	メインメモリはほとんど必要としませんが、サウンドメモリをかなり必要とします。
同時再生	最大 64 音、8 曲までの同時再生が可能です。
ループ再生	切れ目のないループ再生が可能です。
再生レスポンス	再生開始までのタイムラグは、ほぼありません。
再生中に変更可	音量、ピッチ、パン、テンポ、FX、ダイレクトレベル
制作方法	作曲 サンプルング データ処理
適した用途	BGM、効果音

4.3 One Shot 音源

比較的簡単な音の発音をするための音源で、音色にはどれ位の長さで発音するかなど、時間軸に対してのパラメータも所持しています。

用途として、比較的単純な効果音の再生に使用されることが多いです。

例えば、サイレンの音を 10 秒間鳴らし続けるといった効果音などは、One Shot 音源が適しています。

4.3.1 One Shot 音源を利用するときに使用するバンクの種類

(1) One Shot Bank

One Shot 音源の音色で、波形や音の減衰具合などのパラメータを持っています。また MIDI Program Data と違い、どれ位の時間分再生するかといった簡単な時間情報のパラメータを含んでいます。

1 つの音色を One Shot Data といいます。

1 つの One Shot Bank 内に最大 128 種類の音色持つことが可能で、サウンドメモリ上には最大 16 種類の One Shot Bank を存在させることが可能です。

4.3.2 One Shot 音源でコントロール可能なパラメータ

One Shot 音源では、下記の図のようなパラメータが操作できます。

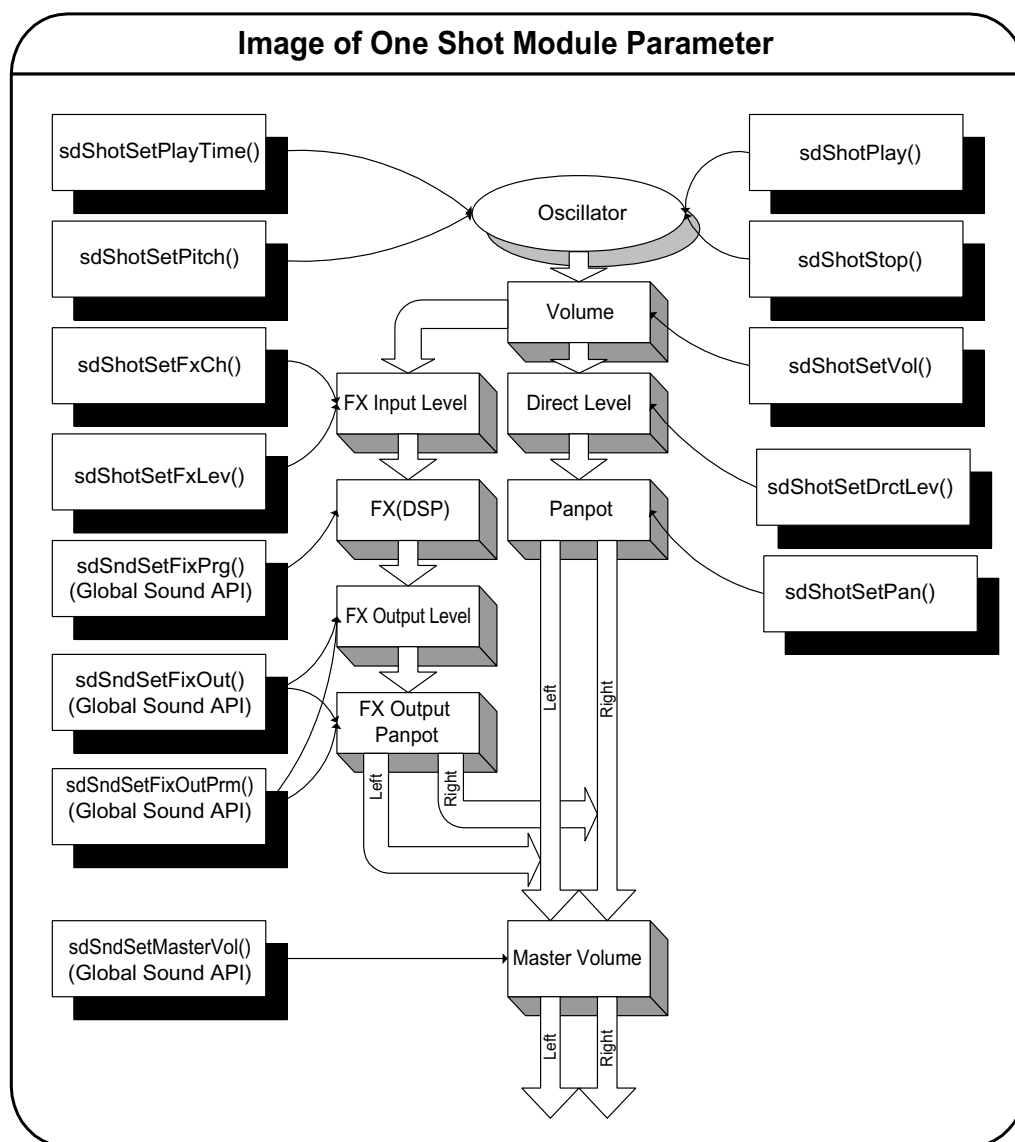


図 4 - 3 One Shot 音源で操作可能なパラメータと信号の流れ

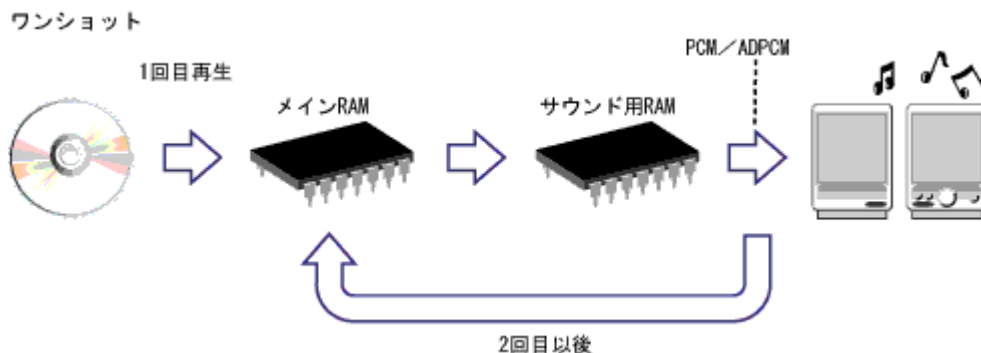
実際にサウンドライブラリで操作できるのは、サウンドクリエイターが設定した値に対しての相対値になります。

例えば、サウンドライブラリで Panpot を 0 に設定しても、サウンドクリエイターが One Shot Data のパラメータ上で Panpot を左端に設定していた場合は、出力される音声は左端に寄った音になってしまいます。

4.3.3 One Shot 音源の再生

- データの流れ

One Shot は、あらかじめサウンド用メモリに読み込まれている PCM / ADPCM 波形（One Shot Data）を再生します。One Shot Bank はサウンド用メモリに読み込まれるため、グラフィックやプログラムの容量に影響されることはありません。サウンドメモリの使用配合はサウンドクリエイターが行います。



次の図のように、あらかじめサウンド用メモリに読み込まれているデータがあります。メインプログラムから再生開始の指示を受けると、そのデータが再生されるようになっています。データの最後まで再生されると、再生完了となります。

- 特徴

再生時間	サンプリングレート	44.1kHz 約 1.5 秒以内	22kHz 約 3 秒以内	11kHz 約 6 秒以内
		65534 サンプル以下		
音質	可変します。CD 音質の 44kHz から、AM ラジオ並みの 11kHz まで。			
データ量	量子化ビット数と PCM/ADPCM により可変します。			
		16 ビット PCM 約 128K バイト	8 ビット PCM 約 64K バイト	4 ビット ADPCM 約 32K バイト
	一つの音に対して、最大で 220KB まで（サウンドメモリの約 1/10）です。			
CPU の負荷	主にサウンドプロセッサに負荷が掛かる。			
必要なメモリ	サウンドメモリのみ必要です。			
同時再生	最大 64 音までの発音が可能です。			
ループ再生	切れ目のないループ再生が可能です。			
再生レスポンス	再生開始までのタイムラグはほぼありません。			
再生中に変更可	音量、ピッチ、パン、FX、ダイレクトレベル			
制作方法	サンプリング データ処理			
適した用途	効果音、ボイス（音声）			

再生時間については、上記の時間よりも仕様の実際は長くなります。上記の時間は現実的な範囲を提示しているもので、8kHz などを設定することも可能です。ただし、音質がかなり低くなるため使用する際には、その点に注意して使用しなければなりませんので、注意して下さい。

4.4 PCM Stream 音源

サウンドメモリ上のバッファ（PCM Stream Ring Buffer）に順次波形を転送することにより、サウンドメモリに収まらない（場合によってはメインメモリにも収まらない）大きなサイズの波形を再生するときに使用します。

例えば、歌入りの BGM やナレーションまたは歓声などです。

サウンドメモリ上には、最大 16 種類の PCM Stream Ring Buffer を存在させることが可能です。

4.4.1 PCM Stream 音源で使用可能な PCM Encode 形式

Dreamcast 上では、次の 3 種類の PCM Encode 形式が使用可能です。

(1) 16 bit Linear Encode 形式

16 bit で音声を PCM 符号化したものです。1 つの符号化された波形で 2 Bytes のサイズになります。

(2) 8 bit Linear Encode 形式

8 bit で音声を PCM 符号化したものです。1 つの符号化された波形で 1 Byte のサイズになります。

(3) 4 bit ADPCM Encode 形式

YAMAHA が開発した PCM の圧縮形式です。

一つの符号化された波形で 4 Bit のサイズ（16 bit Linear Encode 形式から比較して 1/4 のサイズ）になります。

ノート ミドルウェアライブラリを利用することにより、ADX や MPEG-Audio などの圧縮形式も扱うことが可能になります。

4.4.2 PCM Stream 音源でコントロール可能なパラメータ

PCM Stream 音源では、下記の図のようなパラメータが操作できます。

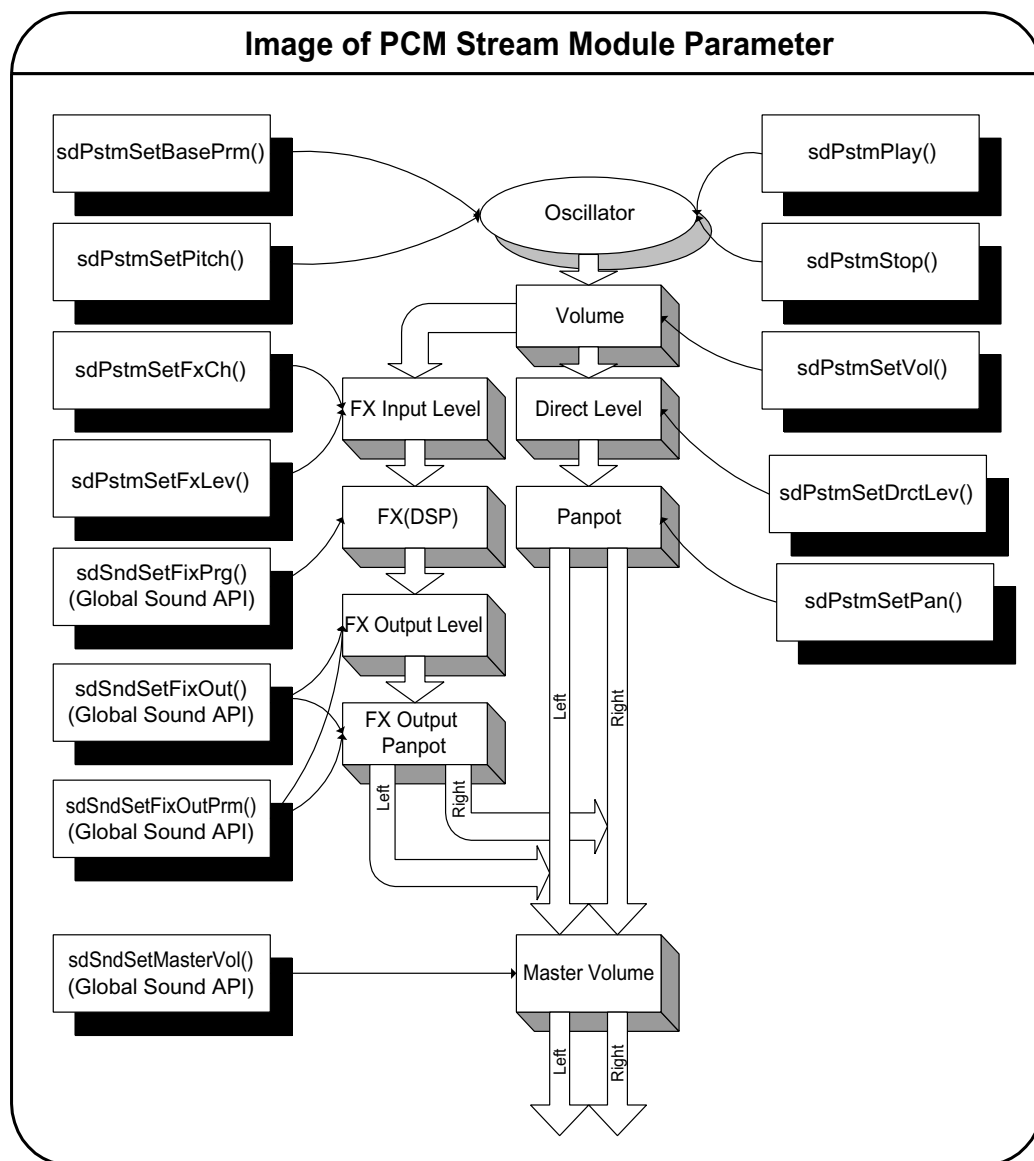


図 4 - 4 PCM Stream 音源で操作可能なパラメータと信号の流れ

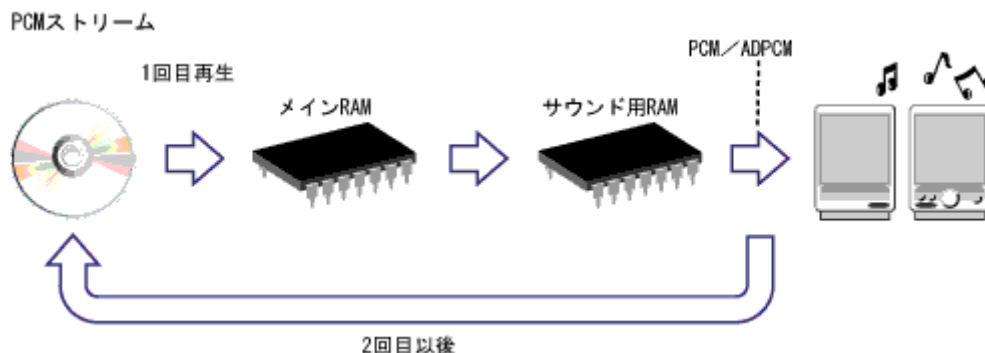
サウンドライブラリでは、2段階のパラメータ操作に分かれます。

基本値を絶対値で設定する `sdPstmSetBasePrm()`関数と各種パラメータを相対値で設定する `sdPstmSetVol()`関数や `sdPstmSetPan()`関数などです。

4.4.3 PCM Stream 音源での再生

One Shot は、最長でも約 6 秒間の再生しかできませんが、PCM Stream は One Shot を拡張して更に長い PCM や ADPCM のデータを再生することが可能です。

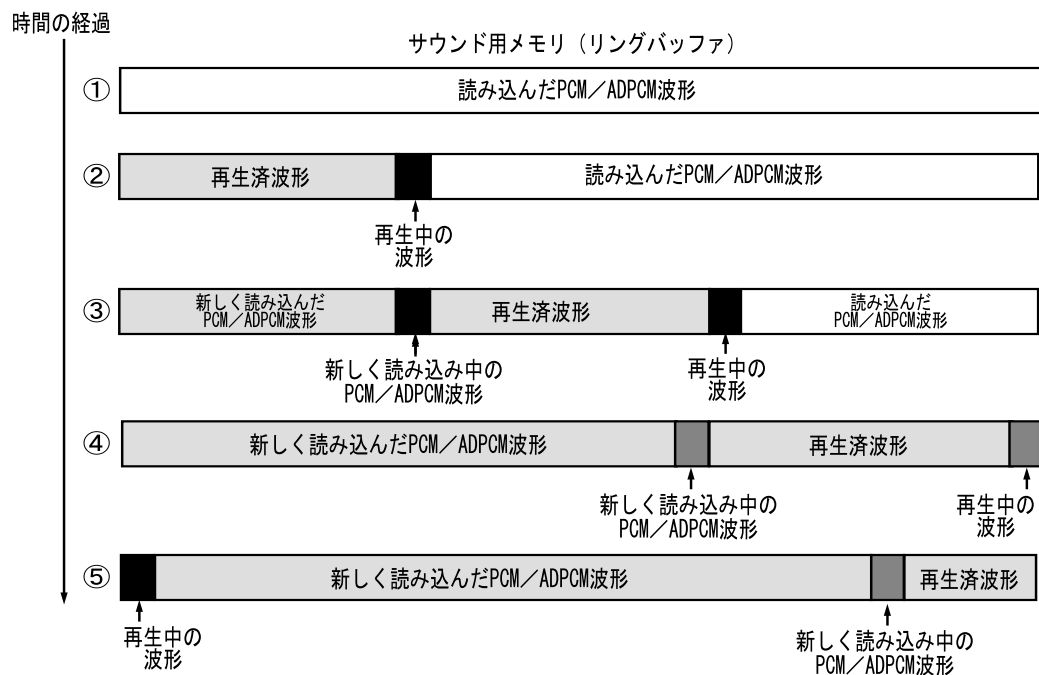
- データの流れ



- リングバッファについて

長いデータを再生するための仕組みが、サウンド用メモリに確保するリングバッファと呼ばれるものです。リングバッファは、次の図のように機能します。

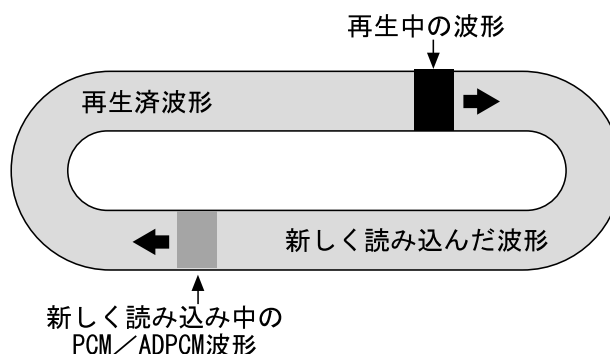
図は上から下に向って時間が経過することを表しています。



- (1) GD-ROM から読み込んだ PCM / ADPCM 波形は、サウンド用メモリにあるリングバッファいっぱいまで読み込まれます。
- (2) 再生が始まります。
- (3) ある程度再生が終わると、再生が終わった波形領域で読み込んだ続きの波形がすでに読み込まれています。
- (4) バッファの最後まで再生が終わります。バッファの先頭には、続きの波形が読み込まれます。
- (5) 最後まで再生が終わると、再びバッファの先頭から再生が始まります。

PCM / ADPCM 再生中にも、常に続きの波形をリングバッファに読み込んで行くため、少ないメモリで長いデータを再生できます。

この動作は、リングの形をしたバッファの波形を再生中にも、続きの波形が次々読み込まれるような動作をするため、リングバッファと呼ばれます。



ノート リングバッファへの波形転送は、CPU が専用 DMA が行います。このために、3D のグラフィック処理などに時間が掛かって、音楽処理で時間を欠けられないようなプログラムの場合、データの読み込みが再生処理に追い付かない場合があります。

上図で解説すると「新しく読み込み中の波形」の領域を「再生中の波形」が追い越してしまっ、すでに再生済みのデータが突然再生されるという症状が起こります。

これを回避するためには、リングバッファのサイズを大きくします。サイズを大きくすることで、こまめにリングバッファの続きのデータを読み込まなくても、CPU の処理が軽いときにリングバッファへたくさんのデータを書き込むことが可能になり、より安定してデータを再生できます。

● 特徴

再生時間	音質にもよりますが制限はほとんどありません。			
音質	任意に変換します。CD 音質の 44kHz から、AM ラジオなみの 11kHz までです。			
データ量	音質に比例してデータ量は増します。CD 音質では、GD-DA と同じです。1 秒間あたりのデータ量は、次の以下のとおりです。			
	サンプリング レート	16 ビット PCM 約 86K バイト	8 ビット PCM 約 43K バイト	4 ビット ADPCM 約 21K バイト
	44k	約 43K バイト	約 21K バイト	約 11K バイト
	22k 11k	約 21K バイト	約 11K バイト	約 5K バイト
CPU の負荷	主にメイン CPU に負荷が掛かります。			
必要なメモリ量	同時に再生する発音数分のリングバッファ（大きさ可変）が必要です。			
同時再生	最大 16 音まで発音可能です。			
ループ再生	切れ目のないループ再生が可能です。			
再生レスポンス	再生開始まで若干のタイムラグがあります。			
再生中に変更可	音量、ピッチ、パン			
制作方法	作曲 レコーディング サンプリング データ処理			
適した用途	BGM、効果音、ボイス（音声）			

4.5 GD-DA 音源

GD-ROM ドライブのデジタルオーディオ部分を音源として利用します。

実際には、GD-ROM ドライブのコントロールを GD-FS ライブラリで行い、Volume や Panpot のコントロールをサウンドライブラリで行うことになります。

ノート DA は、Digital Audio の省略形です。

4.5.1 GD-DA 音源でコントロール可能なパラメータ

GD-DA 音源では、下記の図のようなパラメータが操作できます。

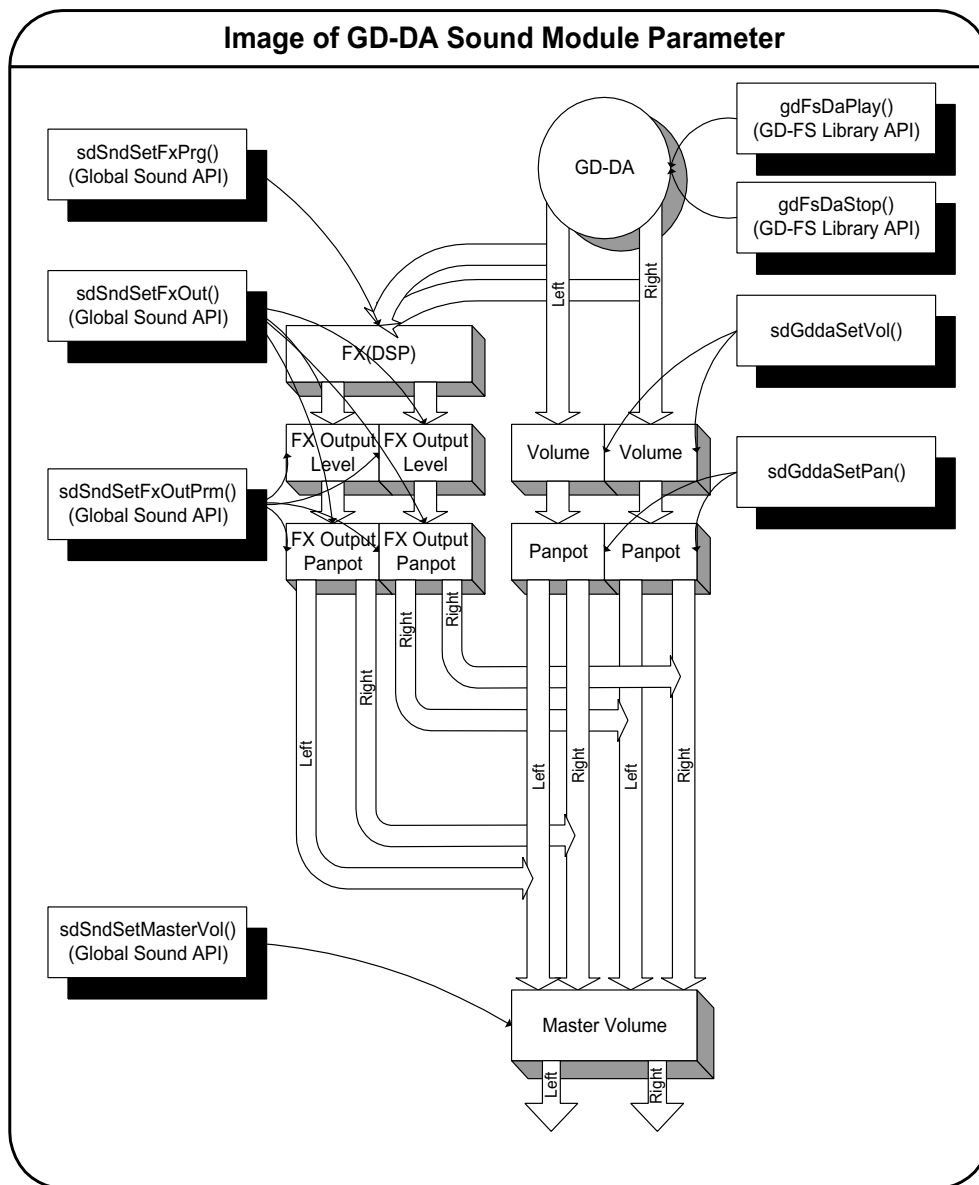


図 4 - 5 GD-DA 音源で操作可能なパラメータと信号の流れ

GD-DA 音源では FX Input Channel は固定で、常に DSP には専用の Input Channel がアサインされています。

FX Input Level は、FX Program Data に指定されたパラメータに対し固定となります。つまりプログラムからは FX Program Data の入れ換え以外は、FX に対してのコントロールはできません。

4.5.2 GD-DA 音源を利用するときに使用するファイルの種類

GD-DA 音源を利用したアプリケーションの開発を行うときは、下記の 2 種類のファイルが必要になります。

(1) DA 形式 File

GD Workshop で GD をエミュレーションするとき 사용합니다。

フォーマットとしては、16 bit Linear Encode PCM 形式 (Little Endian) を Stereo で、1つの Sample ごとに左 Channel 用波形・右 Channel 用波形が対になったものです。Sample は必要分連続して 1つのファイルにします。1つの Track につき 1つの DA 形式のファイルを用意します。

ノート SEGA からは、SCAT (Sound Creator Assistant Tool : サウンドクリエイターが使用するツール) で DA 形式のファイルを作成できる環境を提供しています。

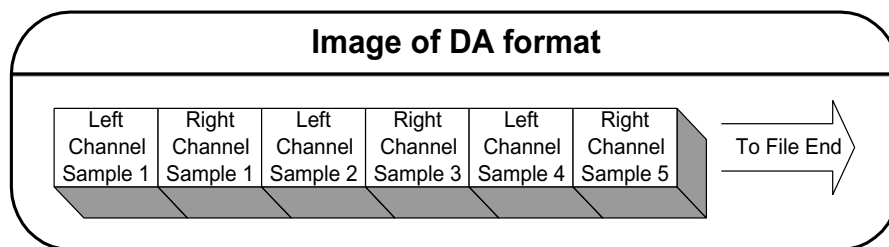


図 4 - 6 DA データイメージ

(2) WAV 形式 File

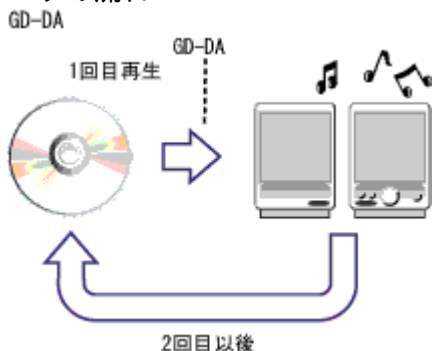
CD Craft などの GD-DA 形式で記録する GD-R を作成するとき 사용합니다。Microsoft Windows で標準で使われている形式のファイルで、Stereo の 16 bit Linear Encode PCM 形式のものを利用します。

通常は、波形が DA 形式と同等のものを対にして用意します。

4.5.3 GD-DA 音源の再生

GD-ROM に記録されている GD-DA を再生します。Dreamcast 本体の起動メニューの GD プレイヤ上で再生可能です。

● データの流れ



ノート 一般の CD プレイヤでは、GD-DA は再生できません。

ただし、内周の単密エリアは CD-DA と同等なので、一般の CD プレイヤ上での再生が可能です。通常、この単密エリアには「これは、Dreamcast 専用のディスクです」という旨の CD-DA データを書き込むために利用します。

● 特徴

再生時間	最大で約 112 分まで再生が可能です。
音質	市販の CD と同じ音質です。 (サンプリングレート 44.1kHz / 量子化ビット数 16 ビット)
データ量	データ量は、多めになります。
CPU の負荷	サウンドプロセッサ・メイン CPU とともに負荷は非常に小さいです。
必要なメモリ	サウンドメモリ・メインメモリともに必要としません。
同時再生	GD-DA 1 曲のみです。ただし、One Shot や MIDI との同時再生は可能です。 GD-DA の再生中は、GD-ROM の読み込みや PCM Stream の再生などの際に、GD にアクセスすることができません。
ループ再生	切れ目のないループの再生はできません。
再生レスポンス	再生開始まで、若干のタイムラグがあります。
再生中に変更可	音量のみです。
制作方法	作曲 レコーディング サンプリング
得意なもの	「BGM」「ナレーション」など

5 FX モジュール

ここでは、それぞれの FX モジュールの特徴やパラメータなどを説明します。

5.1 FX とは

FX とは、簡単にはサウンド専用を用意された DSP を利用し、各音源の設定で DSP に入力された音をリアルタイムに加工し、出力された音のことを指します。

参 照 DSP の説明は、用語集を参照して下さい。

5.2 FX の種類

DSP に対しての入力の設定を FX Input といい、DSP の動作を決めるものを FX Program といいます。

DSP の出力値のコントロールを行う部分を、FX Output といいます。

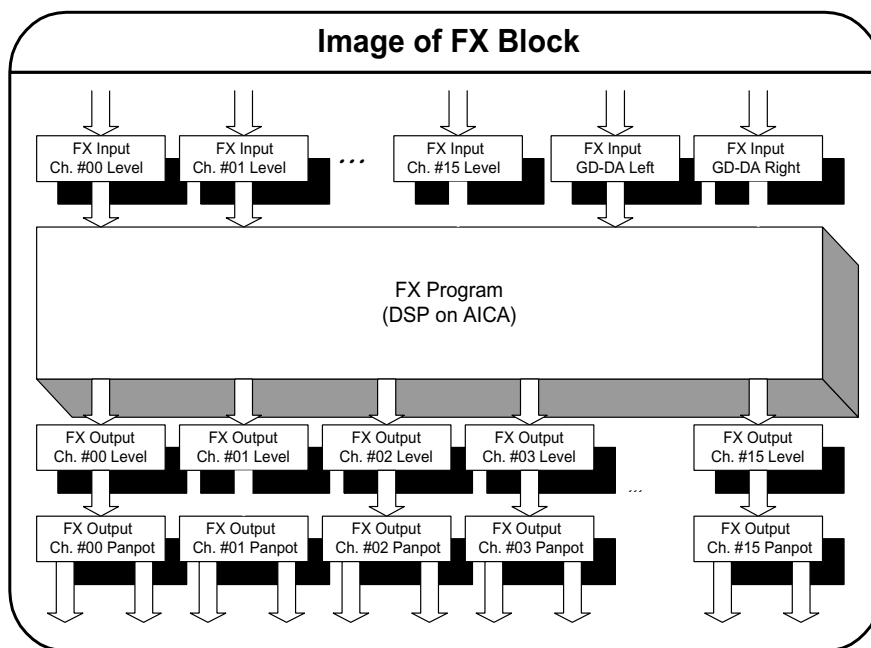


図 5 - 1 FX モジュール関連図

5.3 FX を利用する際に使用するバンクの種類

5.3.1 FX Program Bank

DSP のマイクロコードを FX Program Data といいます。

1 つの FX Program Bank に最大 128 種類の FX Program Data を持つことが可能で、サウンドメモリ上には 1 種類のバンクのみを存在させることが可能です。

5.3.2 FX Output Bank

DSP の演算結果の出力は複数のチャンネルがあり、それぞれのチャンネルを FX Output Channel といいます。

FX Output Channel は 0 から 15 までの 16 のチャンネルがあり、チャンネルごとに出力するレベルを決める FX Output Level と最終的な定位を決める FX Output Panpot がパラメータとしてあります。16 の FX Output Channel の設定をひとまとめにしたものが FX Output Data です。

1 つの FX Output Bank に最大 128 種類の FX Output Data 持つことが可能で、サウンドメモリ上には 1 種類のバンクのみを存在させることが可能です。

5.4 発音中の FX Control の注意事項

基本的に何か音声を再生しているときには、FX Program の変更はしないようにする必要があります。理由は、MIDI 音源でのクリックノイズの項目と同じようなレベル変化が起こったり、場合によっては DSP の誤動作によるノイズが発生する可能性があるからです。

5.5 GD-DA 音源に対しての FX Control

GD-DA 音源で FX を利用する場合、他の音源よりいくつかの制限が付きます。GD-DA 音源はハードウェア的に DSP に入力するチャンネルやレベル (FX Input Level) が固定で、FX Program Data にあらかじめ組み込まれた状態のままの使用が前提になります。

DSP による処理自体は他の音源と同様に扱われ、サウンドクリエータが指定の FX Output Channel に接続されます。よって `sdSndSetFxOutPrm()` 関数を利用して FX Output Level と FX Output Panpot のみリアルタイムにコントロールすることができます。

6 その他の注意事項

6.1 ボリュームの上げすぎによる音割れの発生

Dreamcast では、1つのスロットは 16 bit の振幅幅を持っていて、スロット数が 64 スロットあり DSP を利用した FX を使用することでエフェクト音を重ねることも可能です。また、GD-DA から音を再生することができます。

しかし、これらをミキシングした最終的な出力も 16 bit の振幅幅しか持っていないので、簡単に音割れが起きてしまいます。

下記の図は、2 スロットを同時に鳴らした場合の例です。

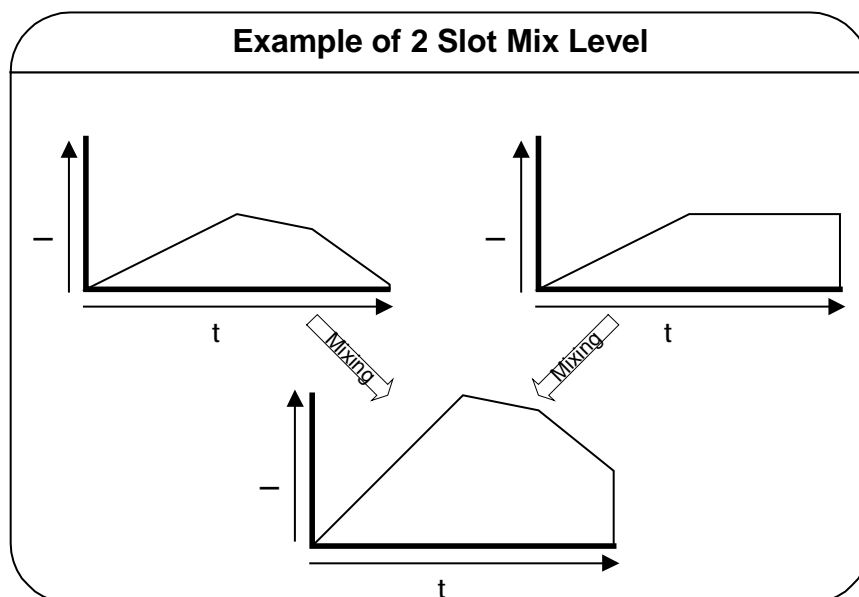


図 6 - 1 2 スロットを同時に鳴らした場合

上図のレベルを `sdXXXXSetVol()` 関数などでボリュームを上げると、下記の図のようになります。

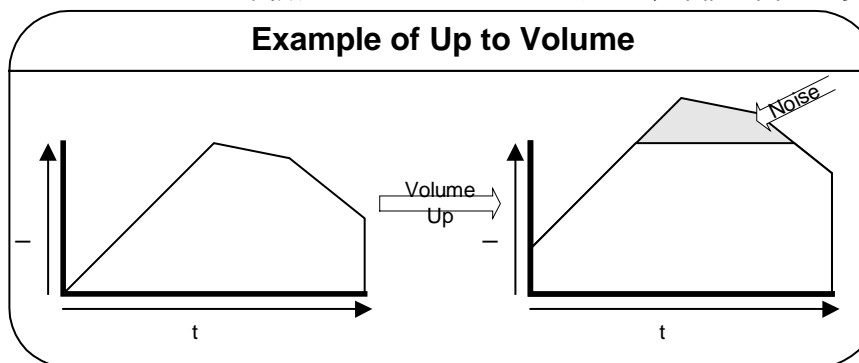


図 6 - 2 ボリュームを上げた場合

注意 上図のようにボリュームを上げすぎるとノイズになる可能性がありますので、注意が必要です。

6.2 無駄な Host Command をなるべく減らす

Host Command を送信することは、多少なりとも CPU 及びサウンドプロセッサに負荷を掛けます。そのため、なるべく無駄な Host Command はサウンドドライバに流れないようにする必要があります。

6.2.1 sdMidiPlay()関数・sdShotPlay()関数などで発音することに一定のボリューム・Panpot のバランスの設定を行わない

サウンドライブラリ上で音量や左右のバランスなどを操作することは、サウンドクリエイターが調整したパラメータを変化させたいときに行うことを理想としています。

つまり発音ごとに、常に一定の値でボリュームや Panpotなどを同時に行うのは、Host Command の無駄に繋がります。

ノート 実際の各パラメータのバランス調整についてはできる限りサウンドクリエイターに行ってもらうようにして下さい。

6.2.2 音源のパラメータのリセットは Reset 関数を利用する

プログラムのコーディングの効率を上げようとするために、再生前に一律でパラメータをリセットし、それぞれのパラメータに対し常に 0 を入れるケースをよく見かけます。これはその関数分（場合によってはそれ以上）の Host Command が送信されることを意味します。

そういった場合は、リセットを行う関数（sdMidiReset、sdShotReset、sdPstmReset など）を利用するほうが、間違いなく確実に全パラメータのリセット処理が行われ、Host Command の発行についても最小限に減らすことができます。

6.3 Stereo と Monaural 設定での注意事項

Dreamcast に関わらず、一般的にステレオで調整された音声を Monaural で再生すると、音量などに変化が起こります。しかし、Dreamcast 上では音量などの変化だけではなく、音割れなどの不都合が発生する可能性があります。

6.3.1 なぜ Stereo モードと Monaural モードで音量のバランスが変化するのか？

Dreamcast ではハードウェア上の制限から Stereo モードで調整された音声を Monaural モードで再生した場合、音が割れる可能性があります。

(1) Panpot について

下記に Stereo 設定時の Panpot (パラメータ) の移動による Volume (パラメータではなく実際に出る音と解釈して下さい) と Monaural 設定時の Panpot (パラメータ) の違いを図で表します。

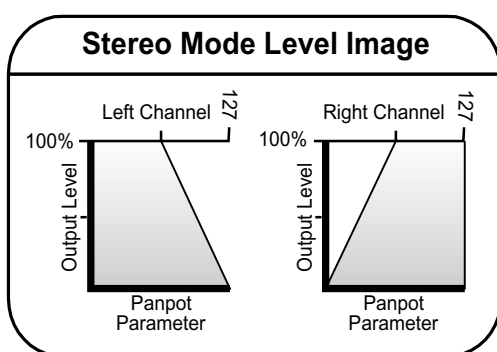


図 6 - 3 ステレオ時

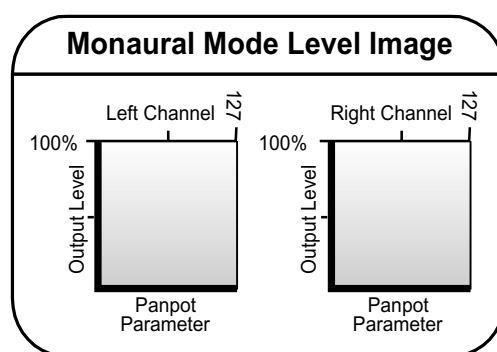


図 6 - 4 モノラル時

例えば Panpot が 0 の場合、Stereo に設定したときは左のスピーカーからは 100%の出力を行い、右のスピーカーからは 0%の出力を行います。

Panpot を 64 に設定していた場合、Stereo や Monaural の関係なく左右ともに出力レベルは 100%です。

Panpot を 127 の場合、Stereo に設定したときは右のスピーカーからは 100%の出力を行い、左のスピーカーからは 0%の出力を行います。

これが Monaural に設定した場合では、それぞれ 100%で出力されます。つまり、左の図の白い部分に相当する Panpot Parameter のときに Monaural に設定した場合は、出力が大きくなることになります。

下の表は、仮に Panpot を 0 と 127 にした音を 2 音重ねて発音した場合の例です。

Example of Level of Stereo Mode		
Parameter of Panpot	Level of Left Speaker	Level of Right Speaker
0	100%	0%
127	0%	100%
Total	100%	100%

図 6 - 5 Panpot 設定（ステレオ時）

Example of Level of Monaural Mode		
Parameter of Panpot	Level of Left Speaker	Level of Right Speaker
0	100%	100%
127	100%	100%
Total	200%	200%

図 6 - 6 Panpot 設定（モノラル時）

音を左右に振っている音ほど Stereo モードのときと、Monaural モードのときに音量差が大きくなるので、Monaural モードにしたときに音量が上がってしまう傾向がある（つまり割れやすい）ということを意味しています。

6.3.2 実際の対処方法

(1) Master Volume を 2 下げる

Master Volume を 2 下げる（つまり最大で 13 までにする）ことで、出力を 50%に抑えることができます。その場合の問題点は、Panpot が 64 で設定されていた場合（Stereo でも Monaural でも出力は 100%になります）は、ボリュームを下げなくてもいいのですが、この場合も音量が下がってしまいます。Monaural にするとき全体的に音量が下がった感じになる可能性があるかもしれません。

(2) 音を左右に振っている音ほど Volume を下げる

実際に、プログラマでも調整はかなり困難で、MIDI 音源に関してはほぼ不可能といってもいいでしょう。これはサウンドクリエイターにて、調整をしてもらわなければなりません。

6.4 発音中のバンク・マルチユニットのダウンロードについて

6.4.1 発音中のブロックは操作しない

例えば、再生中の MIDI Sequence Bank のブロックに全く別の MIDI Sequence Bank をダウンロードするとサウンドドライバ内の MIDI Sequencer が混乱したりする可能性があります。

使用中の MIDI Program Bank のブロックに、全く別の MIDI Program Bank をダウンロードすると AICA が予想もつかない音を発生するかもしれません。

発音が無い場合でも、FX を ON にした状態でマルチユニットファイルのダウンロードを行わないで下さい。（OFF：FX プログラムがクリアされた状態）

6.5 Memory Block 転送について

ここでは Dreamcast 上での、メインメモリからサウンドメモリへの Memory Block の転送方法について説明します。

6.5.1 CPU 上での CPU 転送と DMA 転送の方法

(1) CPU を用いた Memory Block 転送

CPU を用いた Memory Block 転送は、メインメモリからサウンドメモリへの特定のメモリー領域 (Memory Block) を CPU で転送することを表します。実際にはサウンドメモリへのアクセスはかなり低速なので Wait が入ります。一般的に CPU は転送終了まで占有されるので、負荷は高くなります。

(2) DMA を用いた Memory Block 転送

DMA を用いた Memory Block 転送は、メインメモリからサウンドメモリへ特定のメモリー領域 (Memory Block) のサウンド専用 DMA を使用して、転送することを表します。

DMA を用いると DMA の実行以外の CPU は開放されますので、CPU の負荷については CPU 転送に比べると、はるかに低くなります。

(3) DMA 転送を行う場合

DMA 転送を行う場合は、転送終了まで CPU を占有する (サウンドライブラリ内で Wait をかける) 完了復帰と、サウンドライブラリ内で最低限の手続きのみを行い関数から戻る即時復帰の 2 種類があります。

a. 完了復帰

完了復帰とは、Memory Block の転送をすべて終了したときに、関数から戻ります。

b. 即時復帰

即時復帰とは、DMA を用いた Memory Block の転送のことです。これは、サウンドライブラリでは転送情報 (転送元と転送先の情報やアドレス・サイズなど) を記憶するだけで関数を抜けて (キューイング)、実際の転送については CPU の動作と並列で DMA が行うことになります。

つまり、サウンドメモリへの Memory Block の転送の間に CPU を占有されることがありません。更に複数の転送要求を Queue に溜めて、順次転送を行うことも可能です。

(4) DMA を利用する利点

実際には DMA を使ったほうが、転送時間は長く掛かります。しかし、CPU 転送の場合では転送終了まで CPU を占有するので、即時復帰を利用できる DMA 転送は結果的に CPU パワーを節約できます。

6.5.2 DMA を利用した Memory Block 転送の注意点

(1) DMA を利用した転送の場合は、すぐに転送を開始しないことがある

DMA を利用した Memory Block の転送はすぐに開始しないことがあります。

理由は、Queue に転送情報を保存するだけでサウンドライブラリから戻る場合があるからです。Queue を利用するのは、2 度以上の連続転送 (例えば、ステレオの楽曲の PCM Stream) を簡単に実現するためです。

(2) 二重の転送を行わない

例えば、ステレオの波形を用いて PCM Stream 音源で再生を行う場合、一つの Memory Block Handle で、左右の波形を転送するプログラミングを見かけることがあります。しかし、DMA を利用する場合にはコーディングできません。

理由は、DMA を利用した際に転送を実行する場合、即時復帰命令で戻るようにコーディングする場合が多いため、この場合左右の波形を転送した瞬間に右用の波形の転送を要求することになります。

転送処理中の Memory Block Handle に対して、新しいアドレスやサイズなどの設定と転送の実行はできません（二重転送エラーになる）ので、この場合は2つの Memory Block Handle が必要になります。

6.6 メモリマッピングについて

6.6.1 ミドルウェア使用時のメモリマッピング

ミドルウェアは、1音に対してサウンドメモリの後ろから 4040H Bytes を確保します。

例えば、モノラル音声を1つに再生するには 4040H Bytes、ステレオ音声を1つに再生するには 4040H Bytes のサウンドメモリを空けておく必要があります。

6.6.2 可能な限り Sound Memory Mapping は Clear にする

現在のサウンドライブラリでは、複数のブロックが重なってもエラーにはなりません。

例えば、異なる Sound Memory Mapping 情報が入った複数のマルチユニットファイルを次々にダウンロードしていくと、それぞれの Sound Memory Mapping の情報が重なった状態になってしまいます。

この時に例えば、FX Program Work（DSP の作業に必要な領域のこと）と MIDI Program Bank のブロックがたまたま重なってしまった場合には、DSP によって MIDI Program Bank の内容が破壊されてしまうことが考えられます。

ノート こういったトラブルを避ける為に配慮された Sound Memory Mapping の重ね書きをするとき以外には、あらかじめ `sdSndMemClearBankStatAll()` 関数で Sound Memory Mapping 情報を初期化しておくようにすることです。

7 付録

7.1 Dreamcast シーケンサ仕様

ここでは、サウンドドライバがサポートしている MIDI メッセージの仕様について説明します。

7.1.1 Dreamcast MIDI メッセージリスト

数値は 10 進表記と 16 進表記をあわせて掲載しています。() で囲まれた数値は 16 進数表記です。

(1) ノートオン

Status	1st Data	2nd Data
144 + n (9nH)	Note	Velocity

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Note	ノート番号 0 ~ 127 (00H ~ 7FH)
Velocity	ベロシティ値 0 ~ 127 (00H ~ 7FH)

(2) ノートオフ

Status	1st Data	2nd Data
128 + n (8nH)	Note	Velocity
144 + n (9nH)	Note	00 (0 に限る)

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Note	ノート番号 0 ~ 127 (00H ~ 7FH)
Velocity	ベロシティ値 0 ~ 127 (00H ~ 7FH)

(3) プログラムチェンジ

Status	Data
192 + n (CnH)	Program

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Program	MIDI プログラム番号 0 ~ 127 (00 ~ 7FH)

(4) ピッチベンド

Status	1st Data	2nd Data
224 + n (EnH)	ll	mm

n	MIDI チャンネル番号 0 ~ 127 (00H ~ 0FH)
ll mm	ピッチベンド値・8192 (00H 00H) ~ 0 (40H 00H) ~ 8191 (7FH 7FH)

上位 8 ビットのみ有効

(5) モジューション

Status	1st Data	2nd Data
176 + n (BnH)	1 (01H)	Depth

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Depth	モジューションの深さ 00H、10H、20H、30H、40H、50H、60H、70H

Dreamcast の場合、モジューション値として指定した値を 16 で割った数が使われます。つまり、モジューションは 8 段階のみとなります。

(6) データエントリ

Status	1st Data	2nd Data
176 + n (BnH)	6 (06H)	Value

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Value	データエントリ値 0 ~ 127 (00H ~ 7FH)

(7) チャンネルボリューム

Status	1st Data	2nd Data
176 + n (BnH)	7 (07H)	Volume

n	MIDI チャンネル番号 (00H ~ 0FH)
Volume	ボリューム値 0 ~ 127 (00H ~ 7FH) デフォルトは 64H

チャンネルボリュームは、各パートの音量バランスを調整する目的で使用されます。

(8) パンポット

Status	1st Data	2nd Data
176 + n (BnH)	10 (0AH)	Pan

n	MIDI チャンネル番号 (00H ~ 0FH)
Pan	パン値 0 ~ 127 (00H ~ 7FH)

デフォルト値は 64 (40H) ですが、実際の定位は MIDI Program Editor での設定の内容に依存します。

(9) エクスプレッション

Status	1st Data	2nd Data
176 + n (BnH)	11 (0BH)	Expression

N	MIDI チャンネル番号 (00H ~ 0FH)
Expression	エクスプレッション値 0 ~ 127 (00H ~ 7FH)

パートの音量を調節します。チャンネルボリュームメッセージとは独立して使用できます。エクスプレッションメッセージはエクスプレッションペダル、クレッシェンド、デクレッシェンドなど、演奏中の抑揚表現に使用します。デフォルト値は 127 (7FH) です。

(10) フィルタスタートレベル

Status	1st Data	2nd Data
176 + n (BnH)	20 (14H)	ll
176 + n (BnH)	52 (34H)	mm

n	MIDI チャンネル番号 (00H ~ 0FH)
ll mm	フィルタスタートレベル値 0 ~ 16383 (00H 00H ~ 7FH 7FH)、デフォルトは 8192 (40H 00H)

(11) フィルタアタックレベル

Status	1st Data	2nd Data
176 + n (BnH)	21 (15H)	ll
176 + n (BnH)	53 (35H)	mm

n	MIDI チャンネル番号 (00H ~ 0FH)
ll mm	フィルタアタックレベル値 0 ~ 16383 (00H 00H ~ 7FH 7FH)、デフォルトは 8192 (40H 00H)

(12) フィルタディケイ 1 レベル

Status	1st Data	2nd Data
176 + n (BnH)	22 (16H)	Ll
176 + n (BnH)	54 (36H)	Mm

n	MIDI チャンネル番号 (00H ~ 0FH)
ll mm	フィルタディケイ 1 レベル値 0 ~ 16383 (00H 00H ~ 7FH 7FH)、デフォルトは 8192 (40H 00H)

(13) フィルタディケイ 2 レベル

Status	1st Data	2nd Data
176 + n (BnH)	23 (17H)	Ll
176 + n (BnH)	55 (37H)	Mm

n	MIDI チャンネル番号 (00H ~ 0FH)
ll mm	フィルタディケイ 2 レベル値 0 ~ 16383 (00H 00H ~ 7FH 7FH)、デフォルトは 8192 (40H 00H)

キーオンに変更すると、その時点のカットオフ周波数を指定できます。

その場合、フィルタアタックレート、フィルタディケイ 1 レート、フィルタディケイ 2 レートの値が 0 以外に設定されていなければなりません。

(14) フィルタリリースレベル

Status	1st Data	2nd Data
176 + n (BnH)	24 (18H)	ll
176 + n (BnH)	56 (38H)	mm

n	MIDI チャンネル番号 (00H ~ 0FH)
ll mm	フィルタリリースレベル値 0 ~ 16383 (00H 00H ~ 7FH 7FH)、デフォルトは 8192 (40H 00H)

(15) フィルタアタックレート

Status	1st Data	2nd Data
176 + n (BnH)	25 (19H)	Rate

n	MIDI チャンネル番号 (00H ~ 0FH)
Rate	フィルタアタックレート値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(16) フィルタディケイ1レート

Status	1st Data	2nd Data
176 + n (BnH)	26 (1AH)	Rate
n	MIDI チャンネル番号 (00H ~ 0FH)	
Rate	フィルタディケイ1レート値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)	

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(17) フィルタディケイ2レート

Status	1st Data	2nd Data
176 + n (BnH)	27 (1BH)	Rate
n	MIDI チャンネル番号 (00H ~ 0FH)	
Rate	フィルタディケイ2レート値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)	

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(18) フィルタリリースレート

Status	1st Data	2nd Data
176 + n (BnH)	28 (1CH)	Rate
n	MIDI チャンネル番号 (00H ~ 0FH)	
Rate	フィルタリリースレート値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)	

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(19) バンクチェンジ

Status	1st Data	2nd Data
176 + n (BnH)	32 (20H)	Bank#
n	MIDI チャンネル番号 (00H ~ 0FH)	
Bank#	MIDI Program Bank No 0 ~ 127 (00H ~ 7FH)	

指定した MIDI Program Bank No.に対応するバンクが存在しない場合は、プリセット音 (ビープ) が設定されます。プログラムチェンジが送信されるまではバンクチェンジの処理は保留されず。一度に送信できる MIDI Program Bank は 16 なので、実際の Bank#の有効範囲は 0 ~ 15 です。

(20) ホールド

Status	1st Data	2nd Data
176 + n (BnH)	64 (40H)	ll
n	MIDI チャンネル番号 00H ~ 0FH	
ll	ホールド値 0 ~ 127 (00H ~ 7FH)、64 (40H) ~ 127 (7FH) のときにはホールドオン、デフォルトは 0 (00H) (off)	

(21) フィルタ Q (レゾナンス)

Status	1st Data	2nd Data
176 + n (BnH)	71 (47H)	ll
176 + n (BnH)	85 (55H)	ll

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
ll	Q の値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)

フィルタの Q を音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(22) リリースタイム

Status	1st Data	2nd Data
176 + n (BnH)	72 (48H)	Rate

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Rate	EG のリリースレートの値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。127 に設定するとリリースタイムが最長になります。GM 準拠のため、このコントロールチェンジでは、MIDI Program Editor での「Release Rate」と増減の向きが逆になっています。

(23) アタックタイム

Status	1st Data	2nd Data
176 + n (BnH)	73 (49H)	Rate

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Rate	EG のアタックレートの値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。127 に設定するとアタックタイムが最長になります。GM 準拠のため、このコントロールチェンジでは、MIDI Program Editor での「Attack Rate」と増減の向きが逆になっています。

(24) カットオフフリークエンシ

Status	1st Data	2nd Data
176 + n (BnH)	74 (4AH)	Data

n	MIDI チャンネル 0 ~ 15 (00H ~ 0FH)
Data	カットオフ値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(25) ディケイレベル

Status	1st Data	2nd Data
176 + n (BnH)	88 (58H)	Level

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
Level	EG のディケイの値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。

(26) ディケイ1レート

Status	1st Data	2nd Data
176 + n (BnH)	89 (59H)	Rate
n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)	
Rate	EG のディケイレート 1 の値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)	

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。127 を設定すると、ディケイ 1 の変化量が最大になります。

(27) ディケイ2レート

Status	1st Data	2nd Data
176 + n (BnH)	90 (5AH)	Rate
n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)	
Rate	EG のディケイレート 2 の値 0 ~ 127 (00H ~ 7FH)、デフォルトは 64 (40H)	

音色データ設定に対する相対値 (-64 ~ 0 ~ +63) で指定します。127 を設定すると、ディケイ 2 の変化量が最大になります。

(28) Q サウンドコントロール

Status	1st Data	2nd Data
176 + n (BnH)	80 (50H)	Ll
n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)	
ll	0 ~ 127 (00H ~ 7FH)	

Q サウンドモジュールを使用しているときに効果が得られます。MIDI チャンネルが Q サウンドモジュールの入力チャンネルに対応しています。Q サウンドコントロールメッセージの場合、MIDI チャンネル番号 (n) は Q サウンドモジュールの入力番号になります。例えば、4 チャンネルの Q サウンドモジュールで、出力のチャンネル 0 が左、チャンネル 1 が右の場合、ll の値は、0 (00H) で左、64 (40H) で中央、127 (7FH) で右となります。

(29) FX センドレベル

Status	1st Data	2nd Data
176 + n (BnH)	91 (5BH)	Level
n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)	
Level	FX 送信レベル 0 ~ 127 (00H ~ 7FH)	

エフェクトを使用しているときに、FX モジュールへの出力レベルを設定します。

(30) オールサウンドオフ

Status	1st Data	2nd Data
176 + n (BnH)	120 (78H)	0 (00H)
n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)	

n チャンネルで発音中の音をすべて消音します。

(31) リセットオールコントローラ

Status	1st Data	2nd Data
176 + n (BnH)	121 (78H)	0 (00H)

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
---	-----------------------------------

n チャンネルの次の値がリセットされます。

コントローラ	リセット値
モジュレーション	0 (00H)
ボリューム	100 (64H)
エクスプレッション	127 (7FH)
ホールド	0 (00H)
パンポット	64 (40H)
RPN	Null
Q	64 (40H)
ピッチコース	64 (40H)
ピッチファイン	64 (40H)
フィルタアタックレート	8192 (40H 00H)
フィルタディケイ 1 レート	8192 (40H 00H)
フィルタディケイ 2 レート	8192 (40H 00H)
フィルタリリースレート	8192 (40H 00H)
フィルタスタートレベル	8192 (40H 00H)
フィルタアタックレベル	8192 (40H 00H)
フィルタディケイ 1 レベル	8192 (40H 00H)
フィルタディケイ 2 レベル	8192 (40H 00H)
フィルタリリースレベル	8192 (40H 00H)
アタックタイム	8192 (40H 00H)
リリースタイム	8192 (40H 00H)
ディケイ 1 レート	8192 (40H 00H)
ディケイ 2 レート	8192 (40H 00H)
サスティンレベル	8192 (40H 00H)
カットオフフリークエンシ	0 (00H)
ピッチベンドセンシティビティ	MIDI Program Bank での設定値
FX センドレベル	MIDI Program Bank での設定値
ピッチベンド	8192 (40H 00H)

(32) オールノートオフ

Status	1st Data	2nd Data
176 + n (BnH)	123 (7BH)	0 (00H)

n	MIDI チャンネル番号 0 ~ 15 (00H ~ 0FH)
---	-----------------------------------

n チャンネルでノートオンしているノートをすべてオフにします。ただし、ホールドがオンの場合は、ホールドがオフになるまでノートオフされません。

7.2 関数一覧

Sound ライブラリでは、以下の関数がサポートされています。
詳しい内容については、関数リファレンスをご参照下さい。

関数名	機 能
sdBankDownload	Bank の転送
sdBankDownloadEx	Bank の分割転送
sdBankDownloadSetDivSz	Bank の DMA 転送時の分割サイズの指定
sdMltDownload	MultiUnit の転送
sdMltDownloadEx	MultiUnit の分割転送
sdSndClearFxPrg	FXProgram のクリア
sdSndGetFxOut	カレント FXOutputData 番号の取得
sdSndGetFxPrg	カレント FXProgramData 番号の取得
sdSndGetPanMode	現在のパンボットのモードの取得
sdSndSetFxOut	FXOutputData の設定
sdSndSetFxOutPrm	FXOutputData の Channel 毎のパラメータの設定
sdSndSetFxPrg	FXProgramData の設定
sdSndSetMasterVol	マスターボリュームの設定
sdSndSetPanMode	パンボットのモードの設定
sdSndSetQsndPos	Q サウンドのポジションの設定
sdSndStopAll	すべての音声の再生停止
sdMemBlkCreate	メモリブロックハンドルの取得
sdMemBlkDestroy	メモリブロックハンドルの破棄
sdMemBlkDestroyAll	全ての MemoryBlockHandle の破棄
sdMemBlkGetStat	メモリブロックのステータスの取得
sdMemBlkSetPrm	メモリブロックハンドルのパラメータ設定
sdMemBlkSetTransferMode	メモリブロックの転送モードの設定
sdMemBlkTransfer	メモリブロックの転送
sdSndMemBlockClear	1 つのブロックの内容を消去
sdSndMemBlockClearAll	全てのブロックの内容を消去
sdSndMemBlockFree	1 つのメモリマップ情報の初期化
sdSndMemBlockFreeAll	全てのメモリマップ情報の初期化
sdSndMemBlockGetMap	サウンドメモリ上のブロックのマッピング情報の取得
sdGddaGetStat	GD-DA ポートのステータスの取得
sdGddaResetPrm	GD-DA ポートのパラメータのリセット
sdGddaSetPan	GD-DA ポートのパンボットの設定
sdGddaSetVol	GD-DA ポートのボリュームの設定
sdMidiClose	1 つの MIDI ポートのアクセス権の放棄
sdMidiCloseAll	全ての MIDI ポートのアクセス権の放棄
sdMidiGetCtlValue	MIDI 音源のコントロールチェンジで設定された値の取得
sdMidiGetStat	MIDI ポートのステータスの取得
sdMidiOpen	MIDI ポートのアクセス権の取得
sdMidiResetAllPrm	全ての MIDI ポートのパラメータのリセット
sdMidiResetPrm	MIDI ポートのリセット
sdMidiSendMes	MIDI ポートに MIDI メッセージを送信
sdMidiSeqChainLoopOut	MIDI ポートの複数 MIDISequenceData 再生をシームレスに行う
sdMidiSeqContinue	一時停止されている MIDISequenceData 再生の再開
sdMidiSeqGetCurAdr	再生中の MIDISequenceData のアドレスの取得
sdMidiSeqGetTotalBeatTime	再生中の MIDISequenceData のカレント演奏拍数の取得

関数名	機 能
sdMidiSeqPause	MIDISequenceData の再生の一時停止
sdMidiSeqPlay	MIDISequenceData の再生
sdMidiSeqSetBeatReso	MIDISequenceData の再生における 1 拍の設定を行う
sdMidiSeqStop	MIDISequenceData の再生の停止
sdMidiSeqStopAll	全ての MIDISequenceData の再生を停止
sdMidiSetDrctLev	MIDI ポートのダイレクトレベルの設定
sdMidiSetFxInLev	MIDI ポートの FXInputLevel の設定
sdMidiSetGmMode	MIDI ポートの GM モードの設定
sdMidiSetMes	MIDI ポートに送信する MIDI メッセージを作成
sdMidiSetMuteCh	MIDI 音源の Channel 単位の Mute(消音)
sdMidiSetPan	MIDI ポートのパンポットの設定
sdMidiSetPitch	MIDI ポートの再生ピッチの設定
sdMidiSetReleaseMode	MIDI ポートの音声の再生停止(もしくは一時停止)時のリリースの設定
sdMidiSetSeqSpeed	MIDI ポートのスピードの設定
sdMidiSetVol	MIDI ポートのボリュームの設定
sdPstmClose	1 つの PCMStream ポートのアクセス権の放棄
sdPstmCloseAll	全ての PCMStream ポートのアクセス権の放棄
sdPstmGetCurAdr	PCMStream 再生のカレントアドレスの取得
sdPstmGetCurTransferWaveDataAdr	全ての PCMStreamPortHandle の開放
sdPstmGetStat	PCMStream ポートのステータスの取得
sdPstmGetTotalSmpFrame	PCMStream の総サンプルフレームの取得
sdPstmIsTransferWaveData	PCMStream ポートにサンプルフレームを転送可能 / 不可の問合せ
sdPstmOnMemGetCurTransferAdr	OnMemoryPCMStream のメインメモリ上の現在の転送元アドレスの取得
sdPstmOnMemPlay	オンメモリ PCMStream の再生
sdPstmOnMemSetWaveData	オンメモリ PCMStream の再生を行う波形の設定
sdPstmOpen	PCMStream ポートのアクセス権の取得
sdPstmPlay	PCMStream の再生
sdPstmResetAllPrm	全ての PCMStream ポートのパラメータのリセット
sdPstmResetPrm	PCMStream ポートのリセット
sdPstmSetBasePrm	PCMStream ポートのベースボリュームの設定
sdPstmSetDrctLev	PCMStream ポートのダイレクトレベルの設定
sdPstmSetFxInCh	PCMStream ポートの FXInputChannel の設定
sdPstmSetFxInLev	PCMStream ポートの FXInputLevel の設定
sdPstmSetPan	PCMStream ポートのパンポットの設定
sdPstmSetPitch	PCMStream ポートのピッチの設定
sdPstmSetVol	PCMStream ポートのボリュームの設定
sdPstmStop	PCMStream の再生の停止
sdPstmStopAll	全ての PCMStream の再生を停止
sdPstmTransferWaveData	PCMStream ポートにサンプルデータを転送
sdShotClose	1 つの OneShot ポートのアクセス権の放棄
sdShotCloseAll	全ての OneShot ポートのアクセス権の放棄
sdShotGetCurAdr	再生中の OneShotData のアドレスの取得
sdShotGetStat	OneShot ポートのステータスの取得
sdShotGetTotalSmpFrame	OneShot の総サンプルフレーム数の取得
sdShotOpen	OneShot ポートのアクセス権の取得
sdShotPlay	OneShotData の再生

関数名	機 能
sdShotResetAllPrm	全ての OneShot ポートのパラメータのリセット
sdShotResetPrm	OneShot ポートのリセット
sdShotSetDrctLev	OneShot ポートのダイレクトレベルの設定
sdShotSetFxInCh	OneShot ポートの FXInputChannel の設定
sdShotSetFxInLev	OneShot ポートの FXInputLevel の設定
sdShotSetPan	OneShot ポートのパンポットの設定
sdShotSetPitch	OneShot ポートのピッチの設定
sdShotSetPlayTime	OneShot の再生時間の設定
sdShotSetReleaseMode	OneShot ポートの OneShotData の再生停止時のリリースの設定
sdShotSetVol	OneShot ポートのボリュームの設定
sdShotStop	OneShot 再生の停止
sdShotStopAll	全ての OneShot 再生の停止
sdDrvCheckExecute	SoundDriver の動作確認
sdDrvGetBuildInImage	SoundDriver のバージョンの取得
sdDrvGetErr	SoundDriver のエラー情報の取得
sdDrvGetExecCounter	SoundDriver の実行カウント数の取得
sdDrvGetVer	SoundDriver のバージョンの取得
sdDrvHostCmdIsScan	SoundDriver が HostCommand を解析中かどうか調べる
sdDrvInit	SoundDriver の初期化
sdLibGetVer	SoundLibrary のバージョンを取得
sdLibInit	サウンドライブラリの初期化
sdSysFinish	サウンドシステムの終了
sdSysHostCmdBufFlush	SoundLibrary でバッファリングされている HostCommand をフラッシングします
sdSysHostCmdBufIsLock	SoundLibrary ・ SoundDriver 両方の HostCommandBuffer のロック状態を調べる
sdSysServer	VSync で動作させるサーバー関数
sdSysSetErrCallback	エラーコールバック関数の指定
sdSysSetMidiIfMode	SET5 の MIDI 入力端子への MIDI イベント入力の受け付けの有無の設定
sdSysSetSlotNum	各音源ごとの使用可能な発音数の設定

用語集

Sound ライブラリ編

日本語	英 語	内 容
AD 変換	analog to digital translation	アナログデータをデジタルデータに変換すること
ADPCM	Adaptive Differential PCM	アナログ波形の連続的に変化する性質を利用し、AD 変換したデータとの差分を記録することで、デジタル波形のデータ量を減らすデジタル音声記録方式のこと Dreamcast 上では、YAMAHA 独自のフォーマットの ADPCM を使用することができる
AICA	Advanced Integrated Capable Audio	Dreamcast 本体や Sound Box に実装されているサウンド専用の LSI。一つの LSI で波形データの再生、MIDI シーケンスの再生、エフェクト機能、ミキシング機能を持つ
AIFF	Audio Interchange File Format	波形データファイルの形式の一つ。Macintosh 上でのオーディオファイルの標準形式
ARM7		ARM 社提供の 32bit RISC Processor のことで、サウンド部分をコントロールするもの
アタックレート (AR)	Attack Rate	音が立ち上がる時の、EG の変化の度合い (レート) を設定するパラメータ
バンク	Bank	データ (Data) を一つ以上集めてまとめたもの。 サウンドメモリ上で、サウンドデータの差し替えが可能な最小単位であり、プログラマが扱えるサウンドファイルの最小単位でもある
Bend Range		ピッチベンドでどれくらいの音程 (ピッチ) がシフトするかを表す数値あるいはパラメータ。MIDI Program Layer では、ベンドレンジは半音単位で表され、1 半音、12 で 1 オクターブとなる
ブロック	Block	サウンドメモリを分割した時の 1 区分のこと
Control Change		MIDI メッセージのうち、鍵盤以外のコントローラの動作や MIDI 音源が持つ機能を、コントロールするために使用されるもの
DA file		GD-DA 用のサウンドデータファイル。拡張子は「.da」
DA 変換	digital-to-analog translation	デジタルデータをアナログデータに変換すること
データ	Data	1 のオブジェクトのことで、サウンドアイテムを扱う上での最小単位。MIDI Sequence なら 1 曲、MIDI Program なら 1 音色。
Decay Level		鍵盤を押さえている間、持続する音量を EG のコード上位 5 ビットで指定し設定するパラメータのこと
Decay1 Rate		音が立ち上がってから Decay Level に至るまでの、EG の変化の度合い (レート) を指定するパラメータのこと
Decay2 Rate		Decay Level から鍵盤をリリースするまでの、EG の変化の度合い (レート) を指定するパラメータのこと
DMA	Direct Memory Access	コンピュータにおけるメモリアクセスに関する方式。周辺装置がメモリを参照する際に CPU を経由せずに直接アクセスする
Direct Level		スロットから FX の掛からない音を出力する際の出力レベル
DSP	Digital Signal Processor	積和演算に特化している高速処理が可能な LSI。AICA には DSP が内蔵されており、ミキシングやエフェクト処理を行う
Fine Tune		ベースノートを微調整するもの。1/64 半音単位の精度で設定が可能
FX		Dreamcast 内部の DSP で処理されるエフェクト
FX Input Channel		FX モジュールへの入力チャネル
FX Input Level		スロットから FX モジュールに送られる際の出力レベル
FX Level		FX 処理された音が、アウトプットモジュールから出力される時の

日本語	英語	内容
		音量
FX Output Bank		FX のアウトプット設定用バンク。一つのマップに一つしか持てない
FX Output Channel		FX の出力用チャンネル。0～15 までである
FX Output Data		FX の出力音量と定位をチャンネルごとに設定したデータ。一つの FOB に 16 個まで持てる
FX Panpot		FX 処理された音がアウトプットモジュールから出力される時の定位
FX Program		DSP を使用して音色に FX をかける処理を記述したプログラム
FX Program Bank		DSP のプログラム用のバンク。一つのマップに一つしか持てない
FX Program Data		FX のプログラムデータ。一つの FPB に 16 個まで持てる
FX Program Editor		FX Program Data ファイルを作成するためのサウンド開発用ツール。ベースは YAMAHA 社が作成し、サポートは SEGA が行っている。現在は、Macintosh 版のみリリースされている
FX Program Work		FX 用のプログラムが使用するワークブロック。一つのマップに 16 個まで持てる
GD-DA		GD-ROM ドライブのデジタルオーディオ部分を音源として利用する再生方法。市販の CD と同じクォリティで曲を再生することが可能
GD-ROM	GIGA-BYTE Disc	Dreamcast 上で使用される CD サイズのディスク。通常の CD-ROM の容量は 640M バイトだが、GD-ROM は特殊な製造方法により、約 2 倍の 1G (ギガ) バイトの容量がある
GM	General MIDI	最低限必要な音色番号を統一し、メーカーや機種に依存されずに同じ音色が出るように決められた標準の MIDI 規格
Input Channel		入力チャンネル
Linear		非圧縮
LFO	Low Frequency Oscillator	低周波発振器。ここで発生させた低周波を音程や音量にかけ変調することで、ビブラート (周波数変調) やトレモロ (振幅変調) が実現できる。Dreamcast では、LFO の波形としてノコギリ波、矩形波、三角波が用意されている
MIDI	Musical Instruments Digital Interface	電子楽器間や電子楽器とコンピュータとの間で、音程や音の強さ、音の長さ、発音のタイミング、音色などの信号データを送受信するための国際規格。端子には「IN」「OUT」「THRU」がある
MIDI Drum Bank		ドラム用のバンク。レイヤーは一つしか持てない。一つのマップに対して二つまで持てる
MIDI Drum Data		MIDI チャンネルごとにアサインする単位ドラム
MIDI Program Bank		MIDI 音源用の音色として使用するバンク。16 個までサウンドメモリ上に持てる
MIDI Program Data		MIDI チャンネルごとにアサインする単位音色。MIDI Program Bank の中に 128 個まで持てる。 Program Change で切り替えられることから、「プログラム」と呼ぶこともある
MIDI Sequence Bank		MIDI シーケンス用のバンク。最大 16 個までサウンドメモリ上に持てる
MIDI Sequence Data		SMF (スタンダード MIDI ファイル) をコンバートしたもの。MIDI Sequence Bank 内に 128 個まで持てる
マルチユニット	Multi Unit	でのマッピング情報や Bank の内容、及びその Bank のダウンロード先情報などを記録できる。メモリマッピング情報のみを含ん

日本語	英 語	内 容
		だものや、Bank の中身のみ記述されたものなどがある
Oscillator Level		波形を再生した時の出力レベル
One Shot		One Shot データタイプ全体を指す
One Shot Bank		One Shot 用のバンク。マップ上に 16 個まで持てる
One Shot Data		発音させる基本波形にパラメータを付加した Data。一つの One Shot Bank 上に 128 個まで持てる
p04 ファイル		PCM Stream 再生用波形ファイル。ADPCM でヘッダ情報のないモノラルファイルで、拡張子は「.p04」
p08 ファイル		PCM Stream 再生用波形ファイル。コンバートタイプが 8 ビットで、ヘッダ情報のないモノラルファイルで拡張子は「.p08」
p16 ファイル		PCM Stream 再生用波形ファイル。コンバートタイプが 16 ビットでヘッダ情報のないモノラルファイルで拡張子は「.p16」
PCM	Pulse Code Modulation	アナログ波形を一定時間ごとに AD 変換し、デジタルデータに置き換えること（サンプリング）、もしくはその結果で得られる波形データのことを指す
PCM Stream		Dreamcast で使用されている再生方法の一つ。メイン側からサウンドメモリ上の PCM Stream Ring Buffer に波形をストリーミングして再生する方法で、波形データ量の多いものを再生したり、曲をシームレスに連続再生することが可能
PCM (Encode) Type		波形をコンバートする際のエンコードタイプを設定するパラメータ
PCM Stream Ring Buffer		PCM Stream を再生する際に必要になるメモリ。PCM Stream を安定して再生するためには、PCM Stream Ring Buffer のサイズを大きくする
Queue		待ち行列
SDII		波形データファイルの形式の一つ。AIFF と並び Macintosh の標準的なファイル形式
SMF	Standard MIDI File	MIDI シーケンスデータの標準ファイルで、ほとんど全てのシーケンサやシーケンスソフトで読み書きが可能
SCAT	Sound Creators Assistant Tools	SEGA から提供している、Dreamcast 用サウンド開発ツールの総称
Sound Box		Dreamcast のサウンド機能のみを切り出したハードウェア
Sound Driver		サウンド部分の制御プログラム。当然、一つしか持つことができない。ワークを含めて 18000h のサイズを保有する
サウンドシステム	Sound System	Dreamcast の中でサウンドを処理する部分の総称。AICA+サウンドメモリを示す
SMF	Standard MIDI File	MIDI による演奏データを記録する際の標準ファイルフォーマットで、機能が異なるシーケンサの間で演奏データをやり取りする際に利用される
Target		PC や Macintosh と接続して、現在のデータをやり取りしている開発機材を指す
WAV		波形データファイルの形式の一つ。Windows の標準的なファイル形式
エンベロープジェネレータ (EG)	Envelope Generator	音の時間的な変化のようすを折れ線グラフ状に表したものをエンベロープといい、音の立ち上がり方や減衰の仕方を制御する機能のことをいう。 制御できるパラメータとしては、アタックレート（音が出初めに規定の音量に達するまでの変化率）、ディケイレート（増加のあとの変化率）、サスティンレベル（持続音量）、リリースレート（ノートオフ後の変化率）などがある

日本語	英語	内容
カットオフ周波数 (カットオフフ リークエンシ)	cut-off frequency	フィルタなどで指定する周波数。 ローパスフィルタの場合は、カットオフ周波数以上の周波数成分がカットされ、ハイパスフィルタであればカットオフ周波数以下の周波数成分がカットされる。バンドパスフィルタの場合は、カットオフ周波数付近の周波数成分だけを通過させ、残りをカットする。
クリックノイズ		「プイッ」「プリッ」という音のノイズのこと
サウンド用メモリ	Sound Memory	Dreamcast 本体や Sound Box に実装されているサウンド専用のメモリ 標準で 2 M バイトが実装されており、MIDI の音色データ (MIDI Program Bank)、シーケンスデータ (MIDI Sequence Bank)、One Shot 用データや PCM Stream 用のリングバッファなどに利用する
サスティン	Sustain	音の出力中で (ノートオン状態) 一定音量を維持している状態を意味する。この時の音量レベルをサスティンレベルという
サンプリングレ ート	Sampling rate	サンプリングにおいて、1 秒間に何回波形を読み取るか (サンプリングするか) を示す。 22.05kHz の場合は 1 秒間に約 22050 回、44.1kHz の場合は 1 秒間に 44100 回のサンプリングを行う。 同じ量子化ビット数の場合、サンプリングレートが高いほど音質が良くなるが、多くのメモリを必要とする
スプリット	Split	レイヤーの中での階層のこと
デルタタイム		サウンドドライバの MIDI メッセージ送信の最小分解能は 4ms。 ただし、ライブラリは通常 16.7ms ごとにサウンドドライバにコマンドを発行するため、プログラムで直接 MIDI メッセージを送る場合には注意が必要
トレモロ	Tremolo	周期的に音量が上下する効果
波形		音の形
ハイパスフィルタ	Highpass Filter	ローパスフィルタの逆で、カットオフフリークエンシより高い周波数帯域を通過させる。一般に高次倍音だけの硬い音色になる
パン (定位)	Pan	L/R のステレオで、各楽器を左右のどの位置に置くかという設定のこと。各パートの音の分離を良くしたり、ニュアンスを出すことができる
ピッチベンド	Pitch Bend	音程を連続的に変化させるコントローラ。データバイトは、2 バイトで各 7 ビット有効の最大 14 ビット (16384 段階)
フィードバック	Feedback	一般的には出力の一部を入力に戻すこと。ひずみやノイズの改善、ゲインの制限が可能。エフェクトの場合、一度出力された音に再びエフェクトをかけること繰り返し、より深い効果を得ることができる
プログラムチェ ンジ	Program Change	MIDI メッセージなどで現在の音色を別の音色に切り替えること (ここでいうプログラムとは音色のことを指す)
分解能		シーケンサは、あくまでもデルタタイムを基準にして動作するので、分解能は可変です。テンポの早い曲では分解能は荒くなり、遅い曲では細くなる
ベロシティ	Velocity	音の強弱を表すために使う数値。音の出ない状態を 0、最大の音を 127 で表す
ホストコマンド	Host Command	Sound Driver への命令のパケット
量子化ビット数		サンプリングしたデータの精度を決める (サンプリングした音声は、サンプリングレートと量子化ビット数で決まる)。 Dreamcast では、量子化ビット数 8 ビットと 16 ビットの PCM データが利用できる。同じサンプリングレートの場合は、量子化ビ

日本語	英 語	内 容
		ット数が多い方が音質は良くなる。 オーディオ CD（CD-DA）では、サンプリングレート 44.1kHz、量子化ビット数 16 ビットを採用している
PCM Stream リングバッファ	PCM Stream Ring Buffer	PCM Stream などを利用する一時的なメモリ。 PCM データをリングバッファに読み込みながら、逐次再生するために、直線的なメモリがあたかも輪の形状をしているように機能することからリングバッファと呼ぶ
ループ再生		ループ終了地点まで再生を終了した後で、再度ループ開始地点まで戻り再生を行うこと
レイヤー	Layer	積み重ねる層のこと
ローパスフィルタ	Lowpass Filter	ハイパスフィルタの逆で、カットオフフリースクエンシより低い周波数帯域を通過させる