

C64 native tracker

# SID-Wizard Manual

Japanese version

SID-Wizard 1.4

マニュアル日本語版



Written by Mikael Norrgård (WitchMaster)  
Mihály Horváth (Hermit)

Translated and Edited by Takashi Kawano (Akaobi)

9. Feb 2013 Ver. 1.0

# 目次

<b>SID-WIZARD マニュアル日本語版への序文</b> .....	5
<b>PREFACE FOR SID-WIZARD MANUAL JAPANESE VERSION</b> .....	6
<b>1. 前書き</b> .....	8
1.1 SID-WIZARD とは何か? .....	8
1.2 SID-WIZARD の機能.....	10
SID-WIZARD VERSION 1.2 で追加された新機能/改善事項.....	11
SID-WIZARD VERSION 1.4 で追加された新機能/改善事項.....	12
今後追加を予定している機能.....	12
各プレイヤーの性能比較 (LIGHT / MEDIUM / FULL / EXTRA) .....	14
1.3 サウンド合成の原理 .....	14
1.4 SID (6581/8580) チップ.....	15
波形のタイプ: 三角波.....	18
波形のタイプ: ノコギリ波.....	18
波形のタイプ: パルス波.....	18
波形のタイプ: ノイズ波.....	19
HARD SYNC [ハード・シンク] .....	19
RING MODULATION [リング・モジュレーション] .....	19
FILTER [フィルター] .....	19
1.5 16 進法.....	20
ニブル.....	20
1.6 ショートカット・キーの利用について一言.....	21
1.7 コモドール 64 実機で SID-WIZARD を実行する.....	21
X1541 ケーブル+ソフトウェア .....	21
SD2IEC (μIEC/MMC2IEC) .....	21
1541 ULTIMATE.....	22
TURBO CHAMELEON 64 .....	22
1.8 VICE (コモドール 64 エミュレータ) で SID-WIZARD を使う .....	23
ダウンロード.....	23
VICE の設定.....	23
全てをシンプルに (.PRG ファイルを使う場合) .....	24
全てをシンプルに (.D64 ディスク・イメージを使う場合) .....	25
<b>2. ユーザー・インターフェイス</b> .....	26
2.1 スタートアップ・メニュー .....	26
2.2 ユーザー・インターフェイス概要 .....	27
INFO AREA [インフォ・エリア] .....	27
PATTERN EDITOR [パターン・エディタ] .....	28
ORDER LIST [オーダーリスト] .....	28
INSTRUMENT EDITOR [インストゥルメント・エディタ] .....	28
CHORD TABLE [コード・テーブル] .....	28
TEMPO-PROGRAM TABLE [テンポプログラム・テーブル] .....	28

SUBTUNE AND TEMPO [サブチューンとテンポ] .....	28
<b>3. インストゥルメント・エディタ</b> .....	<b>29</b>
3.1 波形 .....	29
WF カラム [波形カラム] .....	30
波形／第 1 ニブルを制御する .....	30
波形／第 2 ニブルを制御する .....	31
ARP カラム [アルペジオカラム] .....	31
DT カラム [デチューンカラム] .....	32
3.2 パルス幅 .....	32
パルス値 (1 つ目と 2 つ目のカラム) .....	33
3.3 ADSR エンヴェロープ .....	33
3.4 ヴィブラート .....	34
3.5 フィルター .....	35
3.6 他のインストゥルメント・パラメータ .....	37
1 フレーム目の波形レジスタ .....	37
HARD-RESTART ADSR [ハードリスタート ADSR] .....	38
HARD-RESTART TIMER / TYPE [ハードリスタート・タイマー／タイプ] .....	38
DEFAULT ARPEGGIO TABLE SPEED [デフォルト・アルペジオ・テーブル・スピー ド] .....	38
DEFAULT CHORD [デフォルト・コード] .....	39
OCTAVE-TRANSPOSE AMOUNT [オクターヴ移調量] .....	39
INSTRUMENT NAME [インストゥルメント名] .....	39
3.7 インストゥルメントの保存と読込 .....	39
<b>4. コード・テーブル</b> .....	<b>41</b>
<b>5. インストゥルメント作成例</b> .....	<b>42</b>
5.1 ベース .....	42
5.2 スネア .....	43
5.3 アルペジオ .....	44
<b>6. パターン・エディタ</b> .....	<b>45</b>
6.1 ノートカラム・エフェクト .....	47
6.2 インストゥルメントカラム・エフェクト .....	47
6.3 エフェクトカラム スモール・エフェクト .....	48
6.4 エフェクトカラム ビッグ・エフェクト .....	49
6.5 パターン作成例 .....	51
<b>7. オーダーリスト</b> .....	<b>52</b>
7.1 オーダーリスト・エフェクト .....	53
<b>8. テンポとマルチスピード</b> .....	<b>54</b>
8.1 テンポプログラム・テーブル .....	54
8.2 マルチスピード .....	55

<b>9. メイン・メニューとファイル操作</b> .....	57
9.1 簡単な設定 .....	57
9.2 曲（ワークチェーン）またはインストゥルメントを読み込む／保存する .....	59
ファイル・ダイアログで使用可能なキー .....	60
SWM ファイルフォーマット .....	60
9.3 曲やインストゥルメントを初期化／消去／削除する .....	61
<b>10. SID-MAKER（制作した曲のエクスポート）</b> .....	61
10.1 SWMCONVERT を使って.S00 ファイルを.SID ファイルへ変換する .....	63
10.2 SID ファイルのメタデータを編集する .....	65
10.3（プログラムに埋め込まれている）プレイヤー・ルーチンについてのアプリケーション・ノート .....	66
<b>11. HERMIDI インターフェイス（拡張 MIDI インプット・ハードウェア）</b> ....	66
<b>12. 使用キー一覧表</b> .....	68
キーボード・レイアウト一覧 .....	69
12.1 一般使用キー一覧 .....	70
12.2 パターン・エディタ 使用キー一覧 .....	71
12.3 オーダーリスト 使用キー一覧 .....	72
12.4 インストゥルメント・エディタ 使用キー一覧 .....	73
12.5 コード・テーブル／テンポ・テーブル 使用キー一覧 .....	73
12.6 10 進－16 進変換表 .....	74
12.7 ARP カラム音階対応表 .....	76
12.8 ATTACK /DECAY / RELEASE タイミング .....	76

# SID-WIZARD マニュアル日本語版への序文

これは、コモドール 64 のコーダー／ミュージシャン／グラフィシャンである Hermit によって制作されたコモドール 64 (以下、C64) ネイティブの SID トラック、SID-Wizard Version 1.4 のマニュアルの日本語版です。このドキュメントは、下記の 2 種のマニュアルのほぼ全訳になります。

底本 : Creating Chip-Tunes with SID-Wizard 1<sup>st</sup> ed. By WitchMaster  
SID-Wizard 1.4 User Manual by Hermit<sup>1</sup>

Hermit と WitchMaster からは、翻訳の許可のみならず、多数のアドバイスをいただきました。最大限の配慮を尽くして下さった両者に深く感謝いたします。

このドキュメントは、WitchMaster のマニュアルを基に、Hermit の記述を補い、それに加えて新たな脚注も追加しています (WitchMaster の脚注には[W]と表記しました。それ以外は訳者による脚注になります)。

このマニュアルを翻訳紹介しようと思った経緯ですが、C64 やそのサウンド・チップたる SID、さらにはトラックという種の音楽ソフトウェアに対して、私たちが知りたいと思うであろうことを丁寧に解説した WitchMaster のマニュアルのスタイルの異例さに驚き、感銘を受けたからです。デモシーンに由来するソフトウェアに、リリースからこんなにも速く、またかくも詳細な解説書が現れることは、まずないと言ってよいでしょう。

さらには SID-Wizard が、総合的な制作環境と、過去の四半世紀に及ぶトラックのなかで抜きんできた完成度を実現していたことを挙げなくてはなりません。Hermit の高レベルのモチベーションと思慮深さから私は多くを学びました。

たしかに、日本に住む私たちにとって、C64 実機での音楽制作を掲げる「ネイティブ」のツールは敷居が高いように思われるかもしれません。しかし、このハードルも乗り越えられる種のものであり、また乗り越える価値のある力強いリリースだと信じて疑いません。

付け加えて、このマニュアルは 80 年代後期の Ultimate Soundtracker の登場より、西欧の主要な音楽制作ソフトウェアの一つであり続けている様々な「トラック」の利用と理解にも役立つでしょう。

---

<sup>1</sup> SID-Wizard の制作者。 [W]

SID-Wizard という素晴らしい贈り物によって、日本でも SID による楽曲制作が活発化することを訳者として願ってやみません。

本マニュアル公開にあたり、翻訳及び語句の確認の申し出を受けて下さった、さざむしさん、K->さん、osoumen さんに感謝いたします。

Hermit の意向に敬意を表し、このドキュメントもまた、パブリック・ドメインのもとに公開します。

この翻訳を同じ SID トラッカーである GoatTracker で音楽制作をされていた Little-Nero さんに捧げます。

## PREFACE FOR SID-WIZARD MANUAL JAPANESE VERSION

This is a Japanese version of manual for a C64 native Music Tracker SID-Wizard 1.4 coded by Hermit, a C64 coder/musician/graphician. This document is almost full translation of the existing two manuals for the tool;

Creating Chip-Tunes with SID-Wizard 1st Ed. By WitchMaster  
SID-Wizard 1.4 User Manual by Hermit

Hermit and WitchMaster gave me permission of translation and a lot of advice as well. I deeply thank them very much for their maximum support. This document is mainly based on WitchMaster's one and I attached Hermit's description to it. Besides, I added some footnotes (if [W] is shown at the end of footnote, it is written by WitchMaster, and the rest is by me).

I was much amazed and impressed at the singular style of the manual by WitchMater, who told politely us about C64, SID, and music trackers we wanted to know. Therefore I introduced and translated this. There are few detailed manuals for "brand-new" softwares originated in the demoscene

Besides, of course I must mention SID-Wizard achieves the comprehensive composing development environment and high level of completion. I have learned high-motivation and deep thought from Hermit. Certainly for us, we, people live in Japan, may feel difficulty of operating "the native tool," which is supposed to be used on real C64. But it can be surely gotten over. And I believe SID-Wizard is a strong release worth getting over the barrier.

Moreover, this manual is usable for using and understanding "music tracker," which has been one of main music tools since the appearance of Ultimate Soundtracker at the late '80s.

Due to the wonderful gift of SID-Wizard, I hope SID composing would be activated in Japan.

Dear mz3364, K->, and osoumen, thanks for receiving my offer upon the publishing of the document. They check and gave me some advice about my translation.

As a mark of respect for Hermit, I publish this document under public domain.

I would like to devote this document to Little-Nero, who is an early GoatTracker user/musician in Japan.

河野崇 Takashi KAWANO / 赤帯 Akaobi

Japan

akaobi0827@livedoor.com

<http://akaobi.wordpress.com/>

# 1.前書き

コモドール 64 で動作する卓越したミュージック・トラッカー、SID-Wizard<sup>23</sup>のマニュアルの増補版へようこそ。これから、このトラッカーの全面にわたって詳細に述べて行くつもりですが、それだけではなく、素晴らしき SID チップの諸機能についても説明していこうかと考えています。私 (WitchMaster) が Mihály の書いたマニュアルを増補するに至ったのは、SID グル<sup>4</sup>には周知であっても、ビギナーには未知のことが [そのマニュアルでは] 色々と省かれていたからです。このマニュアルに後々また追記していくでしょうが、不明瞭だとあなたが思う点や、もっとよりよく説明できそうな点については、お構いなくコメントして下さい。私の E-mail アドレスは [mikael.norrgard@gmail](mailto:mikael.norrgard@gmail) です。私の CSDb (Commodore Scene Database) <sup>5</sup>でのユーザー名は、幸いにももうじき明らかになるでしょうが、*WitchMaster* です。この E ブックの新版は私のホームページ <http://www.witchmastercreations.com> でのリリースを予定しています。

続く 2 節 (1.1 節および 1.2 節) は、SID-Wizard User Manual からの抜粋であり、全て Mihály Horváth (SID-Wizard の制作者) の執筆になります。また他の節は、私と Mihály の記述のミックスです。[Hermit 執筆の] SID-Wizard User Manual と全く同様と言って良い箇所もあります。自分が本当に理解できなかった機能について、私は彼にメールを送り、繰り返し質問してきました。彼の忍耐と助力にこの場を借りて感謝を述べます。

あなたが SID トラッカー未経験者なら、SID-Wizard の機能リストといくらにらめっこしていても進みません。いずれはそこに書かれている全てを理解し始めることでしょう。

## 1.1 SID-WIZARD とは何か?<sup>6</sup>

やあ、皆さん、Hermit です。SID-Wizard とは何かを知りたい方のために、私は簡潔な答を用意しています。これは 80 年代の名高いパーソナル・コンピュータである、

---

<sup>2</sup> 本マニュアル執筆時の SID-Wizard のバージョンは 1.2。[W]

<sup>3</sup> WitchMaster のマニュアルを翻訳中に Hermit よりバージョン 1.4 リリースの告知があった。

<sup>4</sup> SID guru。SID に精通している人。

<sup>5</sup> C64 シーンでは最大規模のプロダクトのアーカイブ・サイトであり、シーナーの主要な意見交換場所。 <http://noname.c64.org/csdb/>

<sup>6</sup> 当節は SID-Wizard 1.2 のマニュアルからの抜粋。[W]



コモドール 64 で動作する、もう一つの [新たな] トラック・アプリケーションです。より明確に述べるならば、**SID (Sound Interface Device)** と呼ばれている、コモドール 64 のサウンド・チップのためのミュージック・エディタです。ご存じの方もいらっしゃるでしょうが、このマシンの背後には、今なお前進を続けるカルチャーと技術的背景が存在します——すなわち、現在もデモ・ミュージシャンたちは進んでシーンに参加し、素晴らしい **SID** チューンを作曲しているのです。**SID-Wizard** はリリース当初からオープンソースのプロジェクトです。私がコメントを付け加えているソースコードの、更なる発展の手助けをどなたでも歓迎します。

当初より私の目的は、**SID** ミュージック制作のための総合的な **C64** ネイティブのツールを創造することにあります。実に山ほどあるエディタには優れた面もある一方で、その内のほとんどは本当に基本的な機能（例えば、インストゥルメントを個別に保存／読み込む機能、マルチスピード・サポート、デチューン、ジャミング、キーボード・トラッキング、等々）を欠いていたためです。

そうは言っても、私が多分にインスピレーションを得た、重要な機能をほぼ全て備えている、名を挙げておくべきトラックもいくつか存在します。私のフェイバリットは次のトラックです——より機能性が高いと思われるものから順に並べていきます——**GoatTracker**<sup>7</sup>（エミュレートした **SID** サウンドを用いるクロスプラットフォームのトラック）、**SDI [SID Duzz' It]**、**X-SID**、**SID-Factory**、**JCH Editor**、**DMC [Demo Music Creator]**、**Hardtrack Composer**。これらのエディタに関する比較表を作成したこともあります。そして自分は、必要不可欠な重要な機能と、発明的／革新的な機能をさらに備えた **C64** をベースとしたネイティブのエディタをコードする必要があると悟りました。多少私は自分のやり方が分かりました。莫大な新機能を備えていても簡潔さ、使いやすさを実現するため、共通の覆いの下に [*under common hood*] 様々な機能の連結を行いました<sup>8</sup>。次節では **SID-Wizard** でできることをざっと見てみましょう……。

---

<sup>7</sup> 2001 年にフィンランド在住の **Cadaver (Lasse Öörni, 当時 23 歳)** が発表した初のクロスプラットフォームの **SID** トラック。Dag Lem の開発した **ReSID** エンジンを使用。これにより Windows 上での曲作成と、エミュレータ及び実機での再生が可能になった。現在は彼の所属するグループ、**Covert Bitops** からリリースされている。サポートも継続中。<http://covertbitops.c64.org/>

<sup>8</sup> *under common hood*。用途によって使い分けが可能な 4 つの異なるプレイヤーが、1 つのプログラムに統合されていることを指す。

## 1.2 SID-WIZARD の機能<sup>9</sup>

- 最大で 50 個のインストゥルメント、100 個の 250 バイトのロング・パターン、トータルで 16 (\$0~\$F) 個のサブチューン<sup>10</sup>。これらの仕様は後に変更される可能性あり
- ラスタータイム<sup>11</sup>の最適化 (およそ\$1C) 及び音楽データの利用によりスペースを割くための、小さなコード・サイズ (エディタのコードとグラフィックを合わせても上限 14k バイト。プレイヤーのコード・サイズは、2k バイトをやや上回る程度)
- ゼロページを実質的に利用しないので、ミュージック・ルーチンがプログラムの他の部分に干渉を起こすことがない<sup>12</sup>
- 3 X 25 行<sup>13</sup>のパターン編集——スクリーンを中心に可能な限りの音楽データが表示される見通しの良さ
- GoatTracker ライクのパターン編集をコンセプトに据えており、キーボード操作は GoatTracker 利用者に負担の少ないように設計されている
- 作曲のワークフロー軽減のために、一画面 (メニュー・スクリーンを除く) に全てを収める。端のスペースを広範囲に利用
- 繋ぎ合わせ [dovetailing]<sup>14</sup>によるポリフォニック・ジャム・モード (ヴォイス 1~3 を順に切り替える) ——Release が長い音に最適
- パターン・データのリアルタイム圧縮——メモリにゼロが現れることがないので、情報価値のあるデータに、より多くのメモリを割ける (XM フォーマットのパターン・データ圧縮に類似しているが、さらなる有効利用が可能)
- オーダーリスト/シーケンス内での「他のサブチューンへジャンプ」コマンド。サブチューンを一つの塊と捉えた曲の打ち込みが可能。オーダーリストのシーケンス・コピー/ペースト機能でまとめられる
- 全てのインストゥルメントに別々のテーブルとインストゥルメント名を備える——テーブルポインタを必要とせず、インストゥルメンを別個に読込/保存が可能
- 全てのインストゥルメントにハードリスタート用の ADSR、ハードリスタート用の時間調整、1 フレーム目の波形設定を備える
- 1 行毎のデチューンとアルペジオ、パルス幅、フィルター・インストゥルメント・テーブルのキーボード・トラッキング

<sup>9</sup> 当節は SID-Wizard 1.2 マニュアルからの抜粋。[W]

<sup>10</sup> subtunes。1 つのファイルに格納されている個別の曲のこと。songs と表記される場合もある。

<sup>11</sup> CPU タイムあるいは CPU サイクルに相当するものを、C64 シーンでは慣例的に raster time あるいは raster line と表現し、プログラム実行の効率性を測る。

<http://noname.c64.org/forums/?roomid=14&topicid=77957&showallposts=1>

<sup>12</sup> ゼロページとは、コモドール 64 の 64k バイトのメモリ内、最初から 0~255 バイトを指す。このメモリ・ロケーションを用いることで、より高速かつ消費メモリを抑えたオペレーションが可能となるが、SID-Wizard では制作された曲 (ミュージック・プレイヤー) がデモのプログラムと干渉を起こすことを避けるため、ゼロページを\$FEと\$FFでしか利用していない。

<sup>13</sup> このマニュアルでは column に「カラム」、row に「行」の訳語を当てた。

<sup>14</sup> dovetailing。ドイツで生まれたアメリカの作曲家、サミュエル・アドラー (1928 年生~) の提示した交響曲のオーケストレーション方法に由来する概念。ジャミング・モードではメロディのノートがチャンネル 1~3 に分散される時、次のノートがゲート・オンになり、最初のノートが減衰を始めても、そのチャンネルに割り当てられた次のノートがゲート・オンになるまで「カット」されない。こうしてノートが次々と「繋ぎ合わされて」いくことで、リヴァーブに相当する音響効果が生成される。

- (X-SID/SDI<sup>15</sup>のように) スピート値やテンポに影響を及ぼさない上限 8x (400Hz) までのマルチフレームスピード
- 高度な部分コピー/ペーストならびに自動前進間隔 [自動スクロール]、パターン/オーダーリスト内のページ・アップ/ページ・ダウン
- トラック・テンポ/ファンクテンポ<sup>16</sup>/テンポプログラム及びパターンスクロールがトラック毎に個別に設定可能
- ヴァイオリン、ギター等々、様々な楽器の印象表現をシミュレートする 4 タイプのヴィブラート
- 同じパターン行に同時に 3 つまでのパターン・エフェクトをかけることが可能 (ノート FX、インストゥルメント FX、等々)
- オクターヴを通して均等に [自然に] ピッチが変化しているように聴こえる、計算されたヴィブラートとスライド/ポルタメント・スピードの達成
- オシロスコープならびにパルス幅/カットオフ周波数ディスプレイ・バーにより、波形やスイープを可視化
- インストゥルメント・テーブル・ポインターをトリガーとしたゲートオフ・イベント (インストゥルメントの **Release** 状態をコントロール可能)
- エディタの基本的な設定情報 (フレームスピード、カラー・テーマ等、「テンプレート」に相当するもの) は、ワークファイルに保存される
- インストゥルメントのアルペジオ・テーブルまたはパターン FX から呼び出される、コード用に分けられたテーブル概念が存在する。それゆえ別のコードのために新たなインストゥルメント作成する手順が不要
- 曲の保存時にノートカラム内の空き領域を圧縮する (これは「packed rest」/「packed NOP」と呼ばれる)
- 曲を SID フォーマットで保存/エクスポート、PRG フォーマットで (さらにはシンプルな PRG と BIN フォーマットでも) 実行可能
- 「SWMcovert」を利用することで XM/MID フォーマットへ/からエクスポート/インポートが可能。x86 プロセッサ用のツールとの連携を容易にする

## SID-WIZARD VERSION 1.2 で追加された新機能/改善事項

- NTSC マシンのサポート——C64 マシン・タイプを検出し、グラフィックと周波数テーブルを自動的に設定
- 1 フレーム目の波形レジスタは全てのインストゥルメントで、どんな値 (SWM1 フォーマットでは 0 を除く) にも設定可能
- メニューで設定可能な選択項目の追加 (設定はワークファイルに保存される)
  - (定義済みの) カラー・テーマを選択可能
  - チカチカとちらつくラスタ・バーの非表示に対応

<sup>15</sup> X-SID は 2007 年にリリースされた、Camelot/Cosine/Crest/Virus に所属するデンマークのシーナー (scener、demoscene にコミットする人間の総称)、Jeff の開発したトラックャー。SDI [SID Duzz' It] はノルウェーの音楽グループ SHAPE の開発したトラックャー。GoatTracker も含め、SID-Wizard はこれらのツールの仕様や機能の一部を引き継いでいる。

<sup>16</sup> パターンの内、偶数のステップをあるテンポで、奇数のステップを別のテンポで演奏すること。GoatTracker の仕様/術語を引き継いでいると思われる。

- ・インストゥルメントの自動入力（曲の保存時、インストゥルメントの不必要な連続指定 [連続選択] を最適化）
- ・GT [GoatTracker] / DMC [Demo Music Creator] のキーボード・レイアウト [2 つのノート入力方法の切替]、2 行のノート (ASDF…)、「1~8」キーによるオクターヴ選択
- ・デフォルトで追いかけ再生が設定——リアルタイムでの [機能の使用 / 不利用の] 切替も可能（使用キーについて書かれた箇所を参照）
- 再生 [*playback*] 中の曲のラスタertimeの測定。ラスタertimeのピーク / 最大値の計算
- オーダーリストのシーケンス内の（部分）コピー / ペースト。C= + E キーにより空白パターンを検索
- （「←」キーで）早送り再生（通常の 4x のスピード）が可能
- 重要なキーのほとんどを網羅したチート・シート（インスタント・ヘルプ）の実装（メニューポイントの最後に追加）
- インストゥルメント・テーブルとコード・テーブルで、「=」キーを押すことにより値の符号反転が可能（例：\$40→\$C0）
- インストゥルメントのデフォルトのコードナンバー上で Return/Enter キーを押すと、そのコードにフォーカスする
- SID-Maker の改善事項：作者情報を再利用メモリに移動、再配置アドレス設定の速度向上、「exe.prg」エクスポート時にサブチューンの切替をサポート、再生時間とラスタertime使用度（現時とピーク時）の表示
- その他の修正点：テーブル内のカーソル・ポジションの記憶。エディット / ジャム [ジャミング] ・モードの見分けがつきやすいようにした。ミュートされたトラックのフィルター・プログラムの停止。ノートカラム内でカーソルのある場所の値を C= + Delete キーで削除できるようにした。再配置アドレスの範囲の拡張（\$0200~\$FFFF）。シングルスピードの曲用の正常な、垂直同期を保った SID の出力。
- スタートアップ・メニューでプレイヤーを Normal/Light/Medium/Extra から選択可能（SID-Maker でも同様に可能）。Light/Medium ヴァージョンの目的はプレイヤーのコード・サイズを減らし、ラスタertimeの消費を抑えること。Extra ヴァージョンではラスタertimeとメモリをいくらか余計に使用するが、スタンドアローンの音楽作品リリースに適した良好な品質を得る……。全てのヴァージョンで同一の SWM1 モジュール・フォーマットを利用。このフォーマットは、SID サウンド・エンジン（プレイヤー / ドライバルーチフォーマットのこと）とはわずかに異なる

## SID-WIZARD VERSION 1.4 で追加された新機能 / 改善事項

- sng2swm コンバータ。サンプルの追加。「F2」キーによる再生で先行するエフェクトを処理するようになった。パターン・エフェクトの追加。
- スタートアップ・メニューにプレイヤー情報（サイズ、ラスタertime）を表示。26 行目に作者情報を表示。C= + +/- キーによるオクターヴの選択

今後追加を予定している機能

- 進行中のプロジェクト：MIDI キーボードのサポート（モノフォニック）——HerMIDI<sup>17</sup>という名の容易に実装可能なハードウェア
- 「source」フォルダ内の **TODO.txt** を確認すること。SID-Wizard がオープンソースのトラッカーであることをお忘れなく……。

---

<sup>17</sup> 第 11 章参照。[W]

## 各プレイヤーの性能比較 (LIGHT / MEDIUM / FULL / EXTRA)

機能 (source フォルダ内の setting.cfg に準じて、個々の必要に応じた変更とコンパイルを行う)	Light	Medium	Full	Extra
ヴィブラート計算、デチューン、コード・テーブル、移調、インストゥルメントのオクターヴ、WRARP スピード		X	X	X
PW/フィルター・テーブル・リセットのオフ、フィルターのキーボード・トラッキング、11 ビット・フィルター、テンポプログラム		X	X	X
ヴィブラート・タイプ、ハードリスタート・タイプ、1 フレーム目の\$09 波形スイッチ			X	X
パルス幅のキーボード・トラッキング、ノート・オフ・テーブル・ポインタ			X	X
サブチューン・ジャンプ FX (トラックに依存しない)、ゼロページの保存/リストア			X	X
フィルター/パルス幅/WF プログラム/スライドにおけるスキップの不使用、FX を除いたフィルター、ゴースト・レジスタ				X
テンポ (0~2) のフル・サポート、ポルタメント後のヴィブラート復帰、ノート/トラックへのディレイ				X

### 1.3 サウンド合成の原理<sup>18</sup>

アナログ周波数合成の基本原理には、加算と減算の二つのタイプが存在します。加算合成とは、ある楽音 [*musical note*] の基本周波数に有限個の倍音を加算することによって、楽器の音をシミュレートする方法のことです。減算合成方法は、本質的に豊富なスペクトルを有したハーモニック・コンテンツ [倍音の強弱] をともなっている、幾種かの波形 (サイン波、三角波、パルス/矩形波、ノコギリ波、ノイズ波、等々) を生成できるオシレーターを土台としています。波形をミックスさせられますし、望みの音色を形作るために、フィルターもかけられます。

私たちは他にも多数存在する、音の要素を変更できます。特に重要なのは、音のヴォリューム・エンヴェロープ (ADSR) の制御です。これにより、非常に鮮明なインストゥルメントの作成が可能になります。サンプルベースの合成と比較すると、可能性は無限大と言えます。ハードウェアとソフトウェアの制約が少しだけ、そのヴァリエーションの数を限定するだけです。

<sup>18</sup> 当節は SID-Wizard 1.2 マニュアルからの抜粋。 [W]

ここではサウンド合成について、これ以上とやかくは言うつもりはありません。サイバースペースには多くの読み物が見つかるはずですから。

## 1.4 SID (6581/8580) チップ

「6581 Sound Interface Device (SID) はシングル・チップの、65XX や同種のマイクロプロセッサと互換性を有した、3 ヴォイスのエレクトロニック・ミュージック・シンサイザー／サウンド・エフェクト・ジェネレーターです。SID は広域のピッチ（周波数）と高分解能のコントロール、音色（ハーモニック・コンテンツ [倍音の強弱] ）、ダイナミクス（ヴォリューム）を提供します。特別な制御回路がソフトウェア・オーヴァーヘッドを最小化し、アーケード／家庭用ビデオ・ゲームでの使用ならびに低コストでの音楽楽器の利用を容易にします」

コモドール SID 6581 データシートより引用

C64 のサウンド・チップである MOS Technology SID 6581 あるいは 8580 は、1982 年に Bob Yannes が設計した減算合成 [*a subtractive sound-synthesis*] を原理とするチップです。SID は C64 のポピュラリティの説明理由のなかでも最重要事項に入ります。何年間にもわたって生産された SID チップの様々なリビジョンとタイプの間には、たくさんの違いが存在します。切り詰められた開発期間によるものでしょう、このチップの仕様はいくらか未完成でした。

オリジナルである SID 6581 に対する新しい SID モデル 8580 では、非常に重要な変更／改善が見えるのですが、6581 の方が良いと考える人が多数です。8580 ではフィルターのカットオフ周波数のコントロール・カーヴの安定化、レゾナンスの歪みの減少、組み合わせ波形のサポートに改善が見られ、あるいは ADSR エンヴェロープの操作がより意に沿いやすいものになっています。チップが内側で正確には何をしているのか、いくつかの面では未だに謎が存在します。それでも近年のソフトウェアとハードウェア双方のエミュレーションの分野（例えば、ReSid.dll<sup>19</sup>や 1541 Ultimate の SID 再生用の VHDL コード<sup>20</sup>）において、リヴァースエンジニ

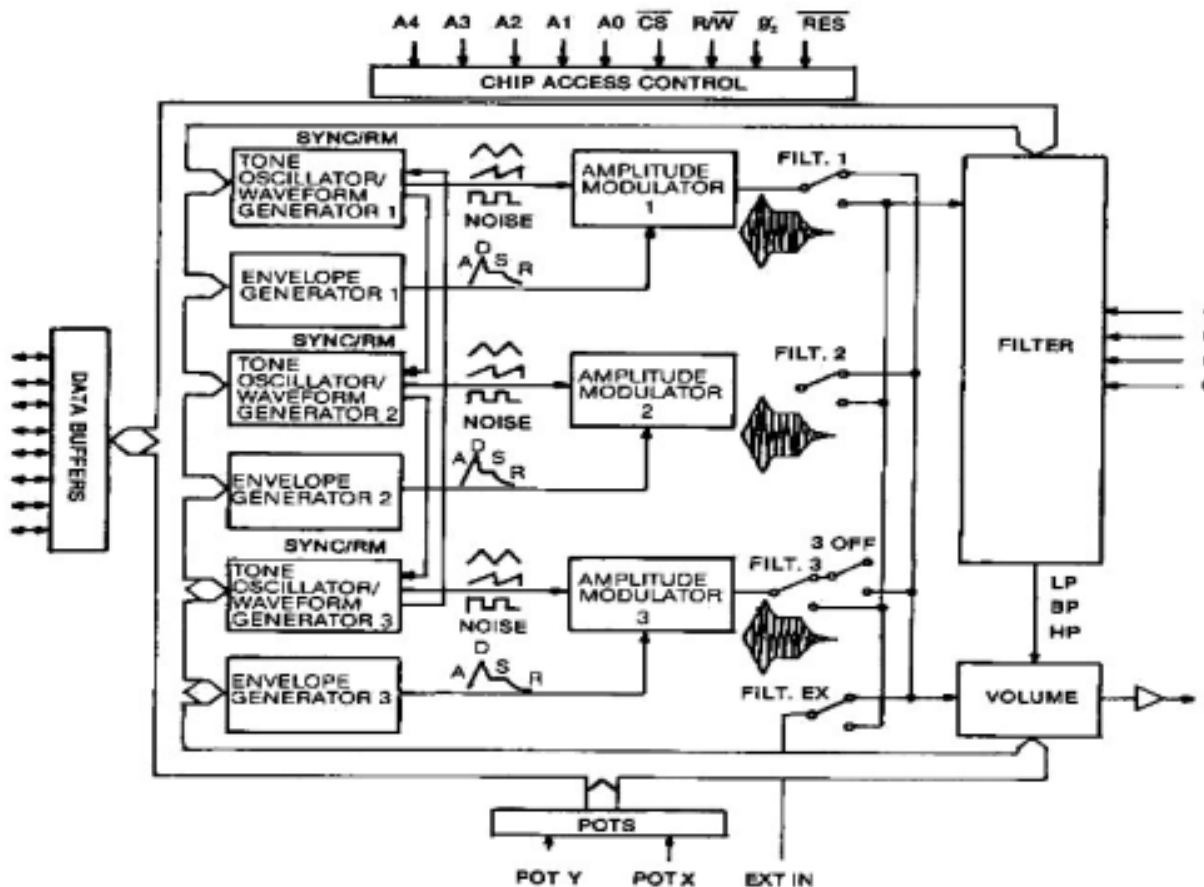
---

<sup>19</sup> 1998 年に Dag Leg が発表した SID エミュレーションを目的としたエンジン。2013 年現在も各種エミュレータや SID プレイヤーに実装されていることを確認できる。

<sup>20</sup> C64/C128 用のストレージ・デバイス 1541Ultimate の後継 1541 Ultimate -II では、ファームウェア 2.1 から最大で左右 8 チャンネルずつ合計で 16 チャンネルの SID エミュレーションに対応。同種のデバイス Turbo Chameleon 64 の SID エミュレーションより優れている (Hermit の教示)。  
[http://www.1541ultimate.net/content/index.php?option=com\\_content&view=article&id=42&Itemid=20](http://www.1541ultimate.net/content/index.php?option=com_content&view=article&id=42&Itemid=20)

アリングのプロジェクトが進みました。Bob Yannes 当人ですら、SID チップの長期にわたるその存続期間や、人々がチップを利用して何をしているかについて、絶えず注意を払ってはいないのです。

SID の構造と挙動をよりよく理解するために、ブロック図を見てみましょう。



SID チップには 3 チャンネルのポリフォニーしかありません（ところで、Yannes の当初のアイデアは遥かに先進的でした。もっと時間 [開発期間] とチップ領域が確保できていたなら、32 チャンネル程度になっていたらと言われていています）。私たちは 3 ヴォイスを扱う必要がありますが、何年もかけて私が (9 ヴォイス [polyphony] の 3SID トラッカー<sup>21</sup>をコードした後で) 学んだのは、智恵の働く音楽家にとっては 3 チャンネルだけでも全く十分だということです。おまけに、これらの制限は作曲家にも影響を与えます。聴き手の興味を惹くために、本当にクールに遣り果せなければならないのです (そんな理由 [制限のなかでの創意工夫] も

<sup>21</sup> 2008 年にリリースされた Hermit 3SID-Tracker を指す。 <http://noname.c64.org/csdb/release/?id=66065>



あって、私はリミックスや **SIDrip Alliance**<sup>22</sup>のバンド・メンバーと **SID** チューンを演奏することが好きなのです）。

**SID** チップには、私 (**Hermit**) の個人的に考える真の限界要因が **1** つ存在します。それはプレーンな波形を形成するためのフィルターが **1** つしかない点です。幸いにもこのフィルターは全てのトラックで個別にスイッチのオン/オフが可能です。特にミュージック・エディタがインストゥルメントのフィルター利用を自動的にトラックしない場合は、作曲時にフィルターの使用設計に関して慎重に進めねばなりません。**SID-Wizard** ではフィルターはトラックで自動的に処理され、フィルターのかかったインストゥルメントを使用するトラックにだけ、フィルター処理がなされ、通常は一番後にフィルターのかかったインストゥルメントが、フィルターのカットオフ周波数をコントロールします。

とはいうものの、**SID** チップには、チャンネル毎のパルス波のデューティ・サイクルがスムーズに変化するという長所があります。この機能はゆっくりとした/速いペースで波形をスイープさせる、クールなエフェクトの生成に用いられます (**VSTi** やシンセサイザー・モジュールの多数で見落されている機能です)。タイミングを合わせる際 [*when it comes to timing*]、**SID** は完璧なデバイスとはいえません。**ADSR** エンヴェロープは時々動かなくなり、正確なヴォリューム・エンヴェロープの転送に失敗します。つまり、全く音が鳴らなくなるのです。ハードリスタートと呼ばれるものに視点を移してみましょう。ハードリスタートとは、レジスタのいくつか音が実際の開始 (ゲート・オン) 前に **1-2** フレーム (**20~40ms**) リセットされ、急な **ADSR** ノートのトリガー・シーケンスに対して、**SID** の反応が固定化してしまう現象を指します。

ここで **SID 6581** の全ての仕様表を繰り返すつもりはありません。さらに先へ進んだ/深い情報を必要としているなら、インターネット上の情報を見てみましょう。優れた概要をお探しなら、こちらを訪ねてみて下さい。

[http://en.wikipedia.org/wiki/MOS\\_Technology\\_SID](http://en.wikipedia.org/wiki/MOS_Technology_SID)

<http://noname.c64.org/csdb/release/?id=57722>

上述の通り、**SID** は **3** ヴォイスによるポリフォニーを有しています (つまり、**3** 音の同時再生が可能です)。これらの音を、波形を組み合わせることによって生成させ

---

<sup>22</sup> デモシーンに出自を持つハンガリーの音楽家 **Hermit**、**NecroPolo**、**Vincenzo**、**Chanbee** により結成されたバンド。「**SID**」の生演奏だけでなく、プロダクトの作成も行う。<http://www.sidripalliance.com/>

ることも、フィルター（ロー・パス／ハイ・パス／バンド・パス）を用いたり、リング・モジュレーションやハード・シンクを用いたり、ヴィブラートを加えたりと、パルス波のパルス幅を変更する種々のエフェクトを用いることもできます。より複雑な音（例えばドラムのような）を作成するための、もう一つの知られた方法は、ウェーブ・テーブルを使うことです。異なった波形を高速で、連続的に変化させるのです。ウェーブ・テーブルを使うことにより、アルペジオの作成も可能になります。

**SID** チップは **3** つの異なった波形（三角波、ノコギリ波、パルス波）を生成可能です。それらを単独で使うことも、組み合わせることもできます。ですが、**4** つ目の波形であるノイズ波だけは、他の波形と組み合わせることができません。**SID 6581** は波形の組み合わせ時にヴォリュームが小さくなるというバグ（あるいは特徴）がある<sup>23</sup>ののですが、**SID 8580** では「その点が改善され」波形を上手く組み合わせることが出来ます。

下記の説明は **SID 6581** データシートより抜粋しています。ハード・シンクとリング・モジュレーションの機能性については、ヴォイス **1** での使用を前提に記述しています。これら両方のエフェクトを使った時、左に位置するヴォイスがこれらのエフェクトに効果を及ぼす「唯一の」ヴォイスになります（例：ヴォイス **1** ⇒ ヴォイス **2**、ヴォイス **2** ⇒ ヴォイス **3**、ヴォイス **3** ⇒ ヴォイス **1**）。

#### 波形のタイプ：三角波

三角波は倍音が小さく、メロウな、フルートのような音質をしています。

#### 波形のタイプ：ノコギリ波

ノコギリ波は偶数倍音と奇数倍音が含まれていて、キレのある、ブラスのような音質をしています。

#### 波形のタイプ：パルス波

この波形のハーモニック・コンテンツ「倍音の強弱」はパルス幅のレジスタによって調整が可能です。キレと窪みのある矩形の波から、鼻にかかったような細く鋭い波動にまで及ぶ、音の質を生成します。リアルタイムでパルス幅を動かすと、音に動きの感覚を加えるダイナミックな「フェージング」エフェクトが生じ、異なったパルス幅間をすばやく移動させれば、面白いハーモニック・シーケンスが生じます。

---

<sup>23</sup> この現象は **ADSR bug** という名で呼ばれている。

## 波形のタイプ：ノイズ波

この出力では、ランダム信号によってオシレーターの周波数が変化します。オシレーターの周波数レジスタによって、低くガラガラと鳴る音からシューツというホワイト・ノイズまで、音質は変化に富みます。スネア・ドラムやシンバルだけでなく、爆発、銃撃、ジェット・エンジン、風、波しぶきといったピッチのない音を作成するのにノイズは便利です。選択されたノイズでオシレーターの周波数をスイープさせると、派手なラッシング・エフェクトが生じます。

## HARD SYNC [ハード・シンク]

オシレーター1 とオシレーター3 の基本周波数を同期させると、「ハード・シンク」エフェクトが生じます。オシレーター3 に関してオシレーター1 の周波数を変化させるとオシレーター3 の周波数でヴォイス 1 から、広域の倍音構造が生じます。シンクがオシレーター3 で生じるためには、ゼロとは別の、しかしなるべくならオシレーター1 の周波数よりは低い周波数がセットされる必要があります。ヴォイス 3 のパラメータがシンクに効果を及ぼすことはありません。

## RING MODULATION [リング・モジュレーション]

リング・モジュレーションとは、オシレーターの三角波形出力をオシレーター1 とオシレーター3 を「リング・モジュレート」された一つの組み合わせに置き換えることです。オシレーター3 に関してオシレーター1 の周波数を変化させると、ベルやゴーンと鳴る音を作成するのに向いた、特殊な広域の非倍音構造が生じます。リング・モジュレーションが聴こえるためには、オシレーター1 に三角波が選択され、かつオシレーター3 の周波数がゼロでない [よりも大きい] 必要があります。ヴォイス 3 のパラメータがリング・モジュレーションに効果を及ぼすことはありません。

## FILTER [フィルター]

SID チップに 1 つだけ存在するフィルターは、3 つのヴォイスで共有しなければなりません。サポートされたフィルターの種類は、ロー・パス、ハイ・パス、バンド・パス [の 3 タイプ] です。これらのフィルターを組み合わせたり、カットオフ周波数やレゾナンスの強さを変化させたりできます。

## 1.5 16 進法

「数学やコンピュータ科学において確立している、16 を底、すなわち基数とした 16 進 (base16 あるいは hex) 数字システム。16 の別個のシンボルを用いるが、zero から nine の値の表象に 0 から 9 のシンボルが、ten から fifteen の値の表象に A、B、C、D、E、F (あるいは a から f) のシンボルが大抵の場合用いられる」

<http://en.wikipedia.org/wiki/Hexadecimal>

16 進法について基本的な理解を得たいのなら、別の場所で [もっと違った] 文書に目を通すべきなのですが、怠惰な人たちのために、私 (WitchMaster) はこのマニュアルの最後に 10 進数と 16 進数の変換表を収めました。符号付き値と符号なし値のものと、表を見れば両方の値を 1 バイトの値に変換することができます。0 から 255 までの符号なし値から -128 から 127 までの符号付き値を範囲とした表になります。SID-Wizard で用いられるほとんどの値は、12 ビット数で示される (\$000 ~ \$FFF) パルス幅を例外として、この範囲を超え出ることはありません。

SID-Wizard には、符号付き値 [入力] を手助けする優れた機能が存在します。この機能はあなたに自動的に反転値を返します。値 -30 が必要な場合、HEX フォーマットで正值 (\$1E) を記入し、「=」キーを押せば、値は反転され、-30 (\$E2) に変わります。

HEX 番号は、値の前にドル記号 [\$] を置きます。例えば、HEX 値 \$A は通常の 10 進法における 10 と等しくなります。

### ニブル

私たちは後の章でニブル [nibble] という語を用います。1 つのニブルは 4 ビットの集合体、すなわちオクテット [octet] の半分に対応します。1 つのニブルが 4 ビットの情報を含む時、16 の可能値が存在します。つまり、それは単独の 16 進の [アラビア] 数字に相当するのです。フル・バイト (オクテット) は 2 つの 16 進の数字で表現されます。従って、1 バイトの情報が 1 つのニブルで表示されるのはよくあることです。HEX 値 \$4F を例にとれば、この場合 1 つ目の [上位] ニブルは 4、2 つ目の [下位] ニブルは F になります。

## 1.6 ショートカット・キーの利用について一言

このマニュアルで言及されているショートカット・キーは、**C64** キーボードの配列に基づいています。**C64** キーボードと **C64** エミュレータ (**Vice**) のキー [の配列] は同じではありません。以下、ショートカット・キーについては括弧内に記します。

## 1.7 コモドール 64 実機で SID-WIZARD を実行する

筆筒のなかにコモドール 64 を仕舞っているなら、取り出して使うのにまたとない機会です。**C64** エミュレータで **SID-Wizard** を動作させることもできます。近頃は **SID** チップのエミュレーションが可能ではあるものの、それでも本物の音質 (例: 温かみやフィルターの歪み) には未だにかないません。また、[実機を利用すれば] ジャミングの際にディレイ (遅延) が生じることもほとんどありません。

**SID-Wizard** をコモドール 64 に転送するには、かなり多くの選択肢が存在します。試しにそのうちのいくつかを列挙してみます。選択肢の網羅的リストは、下記のリンクを確認してみてください。

[http://www.c64-wiki.com/index.php/Data\\_transfer\\_hardware](http://www.c64-wiki.com/index.php/Data_transfer_hardware)

### X1541 ケーブル+ソフトウェア

**C64** フロッピー・ディスク・ドライブを **PC** へと接続し、相互のファイル転送を可能にするケーブルのシリーズです。これらのケーブルを使うには **Star Commander** や **OpenCBM** のようなソフトウェアが必要になります。このケーブルには様々なバージョンが存在し (**X1541**、**XA1541**、**XM1541**、**XMP1541**)、利用にはパラレル・ポートを要します。**USB** 版 (**XU1541**) すら存在するのですが、うまく動くかどうか私にはわかりません。あなたの使おうとしているソフトウェアで動作するケーブルのバージョンを確認しておくことです。

コスト: ~10-20 ユーロ

組立の必要: 既製品のケーブルを買わない場合、あり

### SD2IEC ( $\mu$ IEC/MMC2IEC)

**IEC** バスへとインターフェイス接続する、**SD** をメモリに用いるマス・ストレージのハードウェア・デバイスです。**SD2IEC** は **C64** 用のディスク・ドライブである **Commodore-1541** のエミュレーション/代替品であることを意味しています。**.D64** イメージを **SD** カードへと転送し、普通のフロッピー・ディスクのように

ファイルを用いることができます。これは本当に実用的で、家庭のコモドール 64 実機で、曲を使って作業することすらできますし、（旅行中ラップトップで）エミュレータを立ち上げて.D64 イメージを利用することもできます。

コスト：～50 ユーロ

組立の必要：多少

## 1541 ULTIMATE

1541 Ultimate はコモドール 64（及び他のモデル）のためのストレージ・ソリューションです。このカートリッジには 2 つのバージョン（I と II）があります。後者ではスティック・タイプの USB ドライブさえサポートしています。

コスト：～130 ユーロ

組立の必要：なし

## TURBO CHAMELEON 64

Turbo Chameleon 64 はコモドール 64 用の、利用の簡単な多目的用途のカートリッジです。MMC64 互換の MMC/SD カードスロットが 1 つと、種々様々の機能の加えられた 1541 エミュレーションを備えています。スタンドアローンのユニットとして、Turbo Chameleon 64 を操作することすら可能です（基本的には、本当にクールな C64 のカートリッジ・ケースにおける、C-One や Minimig<sup>24</sup>のような FPGA コンピュータなのです）

コスト：～200 ユーロ

組立の必要：なし

---

<sup>24</sup> <http://en.wikipedia.org/wiki/Minimig> [W]

## 1.8 VICE（コモドール 64 エミュレータ）で SID-WIZARD を使う

コモドール 64 を所有することに恵まれていない人々のために、ここでは Vice エミュレータで SID-Wizard を利用する方法を説明しましょう。[ただし、] エミュレータ使用時にジャミング・モードを利用すると、実機と比較して音の遅延がかなり生じます。以下の手順は Windows を前提にしたものですが、Linux や Mac OS X でも同じように使えるでしょう。

### ダウンロード

1. Vice64 をダウンロードし、インストールします

<http://www.viceteam.org/#download>

2. SID-Wizard をダウンロードします

<http://noname.c64.org/csdb/release/?id=112716>

もしくは、

<http://sourceforge.net/projects/wid-wizard/>

[各種プログラムが収録されたディスク・イメージである] SID-Wizard 1.4 disk.d64 をダウンロードするのも良いですし、Vice のシームレスなフォルダ統合環境<sup>25</sup>を利用するつもりなら、(SID-Wizard と SID-Maker の) PRG ファイルを個別にダウンロードしても良いでしょう。このマニュアルでは後者の方法を推奨し、使用を想定し説明しています<sup>26</sup>。

### VICE の設定<sup>27</sup>

1. Vice を起動します
2. ツールバーの Options から Double size をチェックし、有効化します
3. Options の内、Warp mode の無効化と、Sound playback の有効化を確認します
4. ツールバーの Settings から SID settings を立ち上げ、エミュレートする SID チップを選択します。ReSid エンジンを選ぶことで最も精度の高いエミュレーション

---

<sup>25</sup> 直に prg ファイルを読み込むと、保存時に prg ファイルの存在するフォルダが用いられます。この仕様は本当に実用的です。[W]

<sup>26</sup> シームレスなフォルダ統合環境下では、.S00 ファイルを .SID ファイルへ変換するために SWMconvert を使用する必要があります。このツールは Linux や Windows で動作するので、他の OS で D64 イメージを利用し、またそのディスク・イメージから SID ファイルを取り出すことができます。[W]

<sup>27</sup> 既に Vice を起動したことがあり、Joystick settings でキーを Joystick 用にあらかじめ割り当てているようなら 2 つのポートのどちらも None に選択を入れ、必ず Joystick を不使用にしておくこと。

ョンを得ることができますが、処理能力を余分に必要とします。エミュレートする SID チップの選択は好みの問題です。目安として、8580 は 6581 より効果的なフィルターを有していますが、やせた音がします。また、高ピッチと薄い音で良い音質を得るには、Sample method [サンプル方法] を interpolating に変更した方が良いでしょう。違ったバージョンの SID チップでは、あなたの音楽も違った風に鳴るであろうという点を、忘れないようにして下さい

5. このステップは飛ばしても構いませんが、ユーザー・インターフェイスに箔がつかますし、グラフィックや文字が見やすくなるでしょう（ユーザー・インターフェイスは PAL マシンのブラー・エフェクトのスムージングを活用するように設計されています）。Options から Pal Emulation をチェックし、有効化して下さい
6. [最後に] Settings から Save current settings をクリックします

全てをシンプルに（.PRG ファイルを使う場合）

1. SID-Wizard 用のフォルダを新規作成し、その場所に .PRG ファイルをコピーします（曲やインストゥルメントもこのフォルダに保存されることになります）
2. デスクトップ（あるいはあなたの望むところへ）に Vice へのショートカットを作成します
3. ショートカットの行き先を次のように変更します（右クリック後、プロパティを選択）

“Vice フォルダ内の x64.exe へのパス” -autostart “SID-Wizard-1.4.prg” の存在するパス”

これによって SID-Wizard を自動開始すれば、シームレスなフォルダ統合環境が利用可能になります。すなわち、あなたの制作した曲やインストゥルメントは SID-Wizard-1.4.prg と同じディレクトリに保存されます

4. ショートカットを SID-Wizard へリネームします
5. ショートカットから Vice を実行すれば、自動的に SID-Wizard が立ち上がります。ALT + D でフルスクリーン・モードの切替ができます



全てをシンプルに (.D64 ディスク・イメージを使う場合)

1. ツールバーの **Settings** から **Drive settings...** をクリックし、**Drive 9** のドライブ・タイプを **1541** に指定します
2. **Settings** から **Save current settings** をクリックします
3. ツールバーの **File** から **Attack disk image** を選択し、**Drive 8** (または **9**) をクリック。(あなたの音楽とインストールメントを保存するための) **ブランク・ディスク** を作成します
4. デスクトップに (あるいはどこでも良いので) **Vice** へのショートカットを作成します
5. ショートカットの行き先を次のように変更します (右クリック後、プロパティを選択)

“Vice フォルダの **x64.exe** へのパス” **-autostart “SID-Wizard 1.4 disk.d64**  
の存在するパス:**sid-wizard-1.4” -9 “Path to savedisk.d64” -attach9rw**

6. ショートカットを **SID-Wizard** にリネームします
7. ショートカットから **Vice** を実行すれば、自動的に **SID-Wizard** が立ち上がります。**ALT + D** でフルスクリーン・モードの切替ができます

## 2. ユーザー・インターフェイス

### 2.1 スタートアップ・メニュー

SID-Wizard 起動時に、最初に見ることになるのがスタートアップ・メニューです。ここでは好みのプレイヤーを選択することができます。Light/Medium ヴァージョンの目的は、プレイヤーのコード・サイズとラスタertimeの消費量を抑えることにあります。Extra ヴァージョンはラスタertimeの消費とメモリ使用が多くなりますが、スタンドアローンの音楽のリリースに適した好ましいクオリティを得られます。全てのヴァージョンで、SWM1 という同一のモジュール・フォーマットを利用するのですが、これは僅かばかり SID のサウンド・エンジン（プレイヤー/ドライバルーチンのようなもの）と異なります。メニューにポインタを合わせている時、それぞれのプレイヤーの機能を見ることができます。スタートアップ後にダークグレーで表示されているのが、選択されたプレイヤーでは無効化されている機能/テーブルです。



図 1 : SID-Wizard 1.4 スタートアップ・メニュー

## 2.2 ユーザー・インターフェイス概要

SID-Wizard のユーザー・インターフェイスは直感的に理解がしやすく、7つのエリアに区切られた1スクリーンに、全てが表示されています。図2をご覧ください。それぞれの領域と、操作エリアを切り替えるためのショートカット・キーを示しています。キーボードだけを用いて、このユーザー・インターフェイスの内側を動き回ることになります。ほとんどのパラメータと値は、16進フォーマット（これについては前章を読むこと）が与えられています。SID-Wizard のキーボード・ショートカットの完璧なリストは、マニュアル最後の使用キー一覧表を参照して下さい。

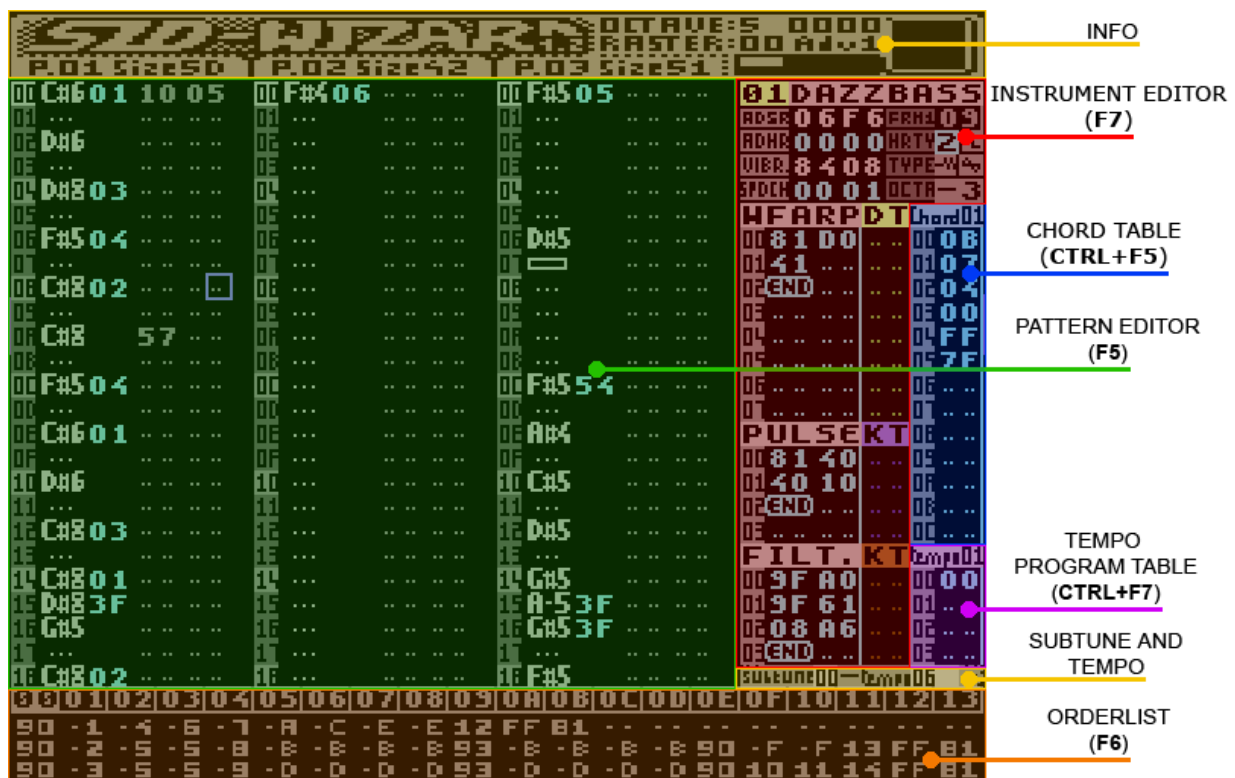


図2：メイン・スクリーン概観

まずは7つのエリアそれぞれを簡単に紹介します。次章からもっと徹底的に説明していきます。

### INFO AREA [インフォ・エリア]

インフォ・エリアには、ロゴ、選択されたパターンに関する情報、ジャミング・モード時のオクターヴ、ラスタertime、再生時間、自動前進モード/間隔 [auto advanced mode / amount]、パルス幅メーター、フィルター・カットオフ・メーター、オシロスコープが表示されています。

## PATTERN EDITOR [パターン・エディタ]

ショートカット・キー：F5

曲 [*the actual music*] を書く場所です。SID チップの 3 チャンネルに対応する、3 つのパターンそれぞれを一度に編集可能です。

## ORDER LIST [オーダーリスト]

ショートカット・キー：F6

曲の全体配置（パターンの順番等）を管理する場所です。

## INSTRUMENT EDITOR [インストゥルメント・エディタ]

ショートカット・キー：F7

ここで曲のインストゥルメントを作成します。このエリアはメイン・エリア、波形アルペジオ・テーブル (WFARP)、パルス・テーブル (PULSE)、フィルター・テーブル (FILT.)、以上 4 つのセクションから構成されています。

## CHORD TABLE [コード・テーブル]

ショートカット・キー：C= + F5 (CTRL + F5)

このテーブルではコードの編集ならびに保管 [リストア] を行えます。インストゥルメントにデフォルトで設定されるコードの選択ができます。使用するコードはパターン・エディタで変更できます。インストゥルメントとは別々にコードを保管できることによって、柔軟さがもたらされます（別のコードを使用する度に別のインストゥルメントを作成する手間が要りません）。

## TEMPO-PROGRAM TABLE [テンポプログラム・テーブル]

ショートカット・キー：C=+F7 (CTRL + F7)

複雑なテンポ設定が必要な時にのみこのテーブルを利用します。テンポに関するさらなる情報は該当箇所 [第 8 章] を参照して下さい。

## SUBTUNE AND TEMPO [サブチューンとテンポ]

このエリアには有効になっているサブチューンと、曲のデフォルトのテンポ/ファンクテンポが表示されています。

## 3. インストゥルメント・エディタ

インストゥルメント・エディタは 4 つの別個のセクションから構成されています。1 つ目のセクションはインストゥルメントに共通するパラメータを含んでいます。2 つ目のセクションは波形とアルペジオを編集するための場所で、3 つ目のセクションはパルス幅テーブル、4 つ目のセクションはフィルター・テーブルです。+/-キーでインストゥルメントを変更します。またユーザー・インターフェイスのどこでも、コード・テーブルやテンポプログラム・テーブル内でさえ、Shift + 「+」または Shift + 「-」キーで、インストゥルメントの変更が可能であることも覚えておいた方が良いでしょう。CTRL (TAB) キーでセクション間を移動可能です。また Shift + CTRL (Shift + TAB) キーを押せば前のセクションに戻ることができます。

これから各エリアについて、私が最も論理的だと考える順番で説明していきます。まず波形アルペジオ・テーブルから説明を始めます。というのは、インストゥルメントを作成するのに、これが最も基本的な部分だからです。直ちに全てが理解的できなくても安心して下さい。第 5 章では何個かのインストゥルメント作成を通して解説します。その時、この章に記された情報の意味も明瞭になってくるでしょう。

### 3.1 波形

利用可能な波形は三角波、ノコギリ波、パルス波、ノイズ波 [の 4 タイプ] です。これら波形については、前置きで紹介した通りです。波形アルペジオ (WFARP) ・テーブルでは波形を選択します。1 つのインストゥルメント [を作成するため] には、少なくともこのテーブルで 1 行 [one row] が必要になります。シンプルなインストゥルメントを作成するのなら、1 行だけでも十分です (波形は時間が経つにつれて変化することはありません)。さらに行を使用する時は、テーブルはフレーム毎に次行に進みます。行を新たに挿入する場合は Shift + DEL (Insert) キーを、行を削除する場合は DEL/Pound (Delete/Backspace) キーを押します。

	WF	ARP	DT
00	\$1	00	..
01	\$1	..	..
02	END	..	..
03	..	..	..
04	..	..	..
05	..	..	..
06	..	..	..
07	..	..	..

図 3 : 波形アルペジオ (WFARP) ・テーブル

どの行にも 3 つのカラムがあります。1 カラム目 (WF) で波形を設定し、2 カラム目 (ARP) でピッチを変更し、3 カラム目は波形のデチューン (DT) に用いられます。

### WF カラム [波形カラム]

**\$00~\$0F** アルペジオ/デチューン行に限定して 1~16 フレームの間繰り返す  
(波形変化なし)

**\$10~\$FD** 単なる波形/制御レジスタ値の設定

**\$FE** テーブル・ポジションへジャンプ (ジャンプ先は 2 つ目の (ARP) カラムで指定する。値が \$40 以上の場合は、それ自身へとジャンプする)

**\$FF** テーブル終端。この場所でテーブル実行は終了する。この値が入力されなかった時は、テーブル終端が END と見なされる

### 波形/第 1 ニブルを制御する

波形カラムの値のうち、1 つ目のニブルで波形 (あるいは波形の組み合わせ) を設定します。波形を組み合わせたい場合は、ある波形の値を、別の波形の値に加算します。すなわち、ノコギリ波とパルス波を組み合わせる場合、値は  $\$2 + \$4 = \$6$  となります。ノイズ波だけは他の波形と組み合わせることができないことを忘れずに。波形ごとの値は次の通りです。

**\$1** 三角波

**\$2** ノコギリ波

- \$4**           パルス波  
パルス波が聴こえるためには、パルス幅をセットするパルス・テーブルに少なくとも 1 行挿入する必要がある
- \$8**           ノイズ波

## 波形／第 2 ニブルを制御する

2 つ目のニブル（制御レジスタ）では波形のための様々なオプションを設定します。これらの値は、同時に使用したいオプションの値を加算することで、ここでもまた組みわせるが可能です。（通常通り）ゲート・オンとリング・モジュレーションを有効にする場合、その結果は  $\$1 + \$4 = \$5$  という風になります。

- \$1**           ゲート・オン  
波形が聞こえるためにはゲートがオンである必要がある
- \$2**           シンク  
前書きでも説明した「ハード・シンク」エフェクトを生じさせる
- \$4**           リング・モジュレーション  
前書きの説明を参照
- \$8**           テスト  
この値が消去されるまで、オシレーターをゼロにリセットし、ロックをかける

## ARP カラム [アルペジオカラム]

- \$00**           ゼロ・ピッチ・シフト（最重要、「0」インターヴァル [音程]）——元のノート・ピッチを再取得する
- \$01～\$5F**   半音で上方に相対ピッチ・シフト（ポジティブ [正]・インターヴァル）
- \$7F**           デフォルトの、またはパターン FX で設定したコードへジャンプする。該当コードに依存するので、波形テーブルの実行がコードの後にも続くこともある。このコマンド使用時、（WF）カラムでスピード値（\$00～\$0F の範囲で値を設定する必要がある）を設定することで、デフォルトのコード・スピードを無視することもできる。
- \$80**           処理なし [No process]（ピッチやデチューンの操作を行わず、そのままの波形を出力する際に役立つ）

**\$81~\$DF** 絶対ピッチの設定（周波数はノート C1~A8 と同一。マニュアル最後の表を参照）

**\$E0~\$FF** 負の方向へ、下方に相対ピッチ・シフト（ネガティブ [負] ・インターヴァル）

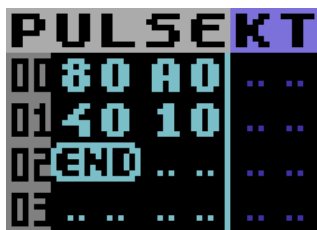
### DT カラム [デチューンカラム]

**\$00~\$FE** 細かなデチューン量の設定。値が増えるほど、ピッチは鋭くなる

**\$FF** デチューン・テーブルで何も処理を行わない [*No process*]（前に設定されたデチューン値を保持する）

## 3.2 パルス幅

パルス幅の設定には、パルス [PULSE] ・テーブルを利用します。変更できるのはパルス波の音色のみなので、WFARP [波形アルペジオ・] テーブルでパルス波を設定しておく必要があります。パルス幅は固定されている必要はなく、時間変化させることができます。これがいわゆる **Pulse Width Modulation** [パルス幅変調] (PWM) といって音の表現を豊かにするもので、SID の楽曲ではありふれた手法です。WFARP テーブルと同じく、行を新たに挿入するには **Shift + DEL (Insert)** キーを、行を削除するには **DEL/Pound (Delete/Backspace)** キーを使います。



PULSEKT					
00	80	A0	..	..	
01	40	10	..	..	
02	END	..	..	..	..
03	..	..	..	..	..

図 4 : パルス・テーブル

1 行毎に 3 つのカラムがあります。最初の 2 つのカラムはパルス幅とモジュレーションをコントロールし、3 つ目のカラムはキーボード・トラッキング [KT] を設定します。このテクニックは KORG のシンセサイザー (MS20、707、等々) でも知られています。キーボード・トラッキングは、パルス幅を音程 [*note-pitch*] に従属させます。この設定を使えば、音はよりいきいきと、変化に富んだものになります。この値が大きくなればなるほど、隣り合った音色の音程の差が大きくなります。 .



指数スケールにおける符号付き値が、キーボード・トラッキングの強さになります。  
\$00～\$7F の間では、高音になるに従ってパルス幅は増加します。負値\$FF～\$80  
(-1～-127) でも同様のことが行えますが、劇的変化は望めません（言い換えるなら、それは弱い、あるいは勾配の緩いエフェクトです）。

パルス幅を設定／変調させる時、ユーザー・インターフェイスのインフォ・エリアにあるパルス幅が非常に役に立ちます。一番上から、バーはヴォイス 1、ヴォイス 2、ヴォイス 3 を示しています。上から 4 番目のバーはフィルターのカットオフ周波数を示しています。



図 5 : パルス幅／フィルター・カットオフ・バー

パルス値 (1 つ目と 2 つ目のカラム)

**\$8x～\$Fx xx** パルス波のパルス幅を上位ニブルと下位バイトで設定。値を\$000  
または\$FFF に設定した場合は一定時間 DC 出力（静音）が生じる。  
\$800 に設定した場合は矩形波が生じる。上位ニブルの値はパルス  
幅に影響しないが、\$8～\$F の範囲で設定する必要がある

**\$00～\$7F xx** 符号付き値 (0～\$7F / \$80～\$FF) を現在のパルス幅に 0～127  
(\$00～7F) 回加算／減算する。この値によってパルス幅を動かせる。  
正值を負値に変換するには「=」キーを押すか、マニュアル最後の、  
符号付き 10 進値 (-128～127) 用の変換表を参照

**\$FE** テーブル内のあるポジションにジャンプする（ジャンプ先を 2 カラム  
目で指定し、同じポジションにもジャンプできる）

**\$FF** テーブル終端。この場所でテーブル実行は終了する。この値が入力  
されなかった時は、テーブル終端が END と見なされる

### 3.3 ADSR エンヴェロープ

ほぼ全てのシンセサイザーに ADSR エンヴェロープは共通して存在します。ADSR  
とは、アタック [*Attack*]、ディケイ [*Decay*]、サステイン [*Sustain*]、リリ  
ーズ [*Release*] を [各頭文字をとって] 表しています。エンヴェロープを用いて、  
時間経過とともにインストゥルメントの振幅がいかに変化するかをコントロールで

きます。4 つのパラメータの内、3 つは時間値設定のパラメータ (A、D、R) で、後 1 つはサステイン (S) 状態における振幅値のパラメータを設定します。

下図 6 をご覧下さい。ノート再生時、アタック・タイム (A) はヴォリュームが最大に達するまでにかかる長さを決定します。振幅が要求されたサステイン・レベルに達するのにかかる時間を意味するディケイ・タイム (D) が、この後振幅に作用します。ノートオフ・コマンド送信 (キー・リリース) 時には、リリース・タイム (R) が、振幅がゼロ (静音) に到達するのにかかる長さを決定します。

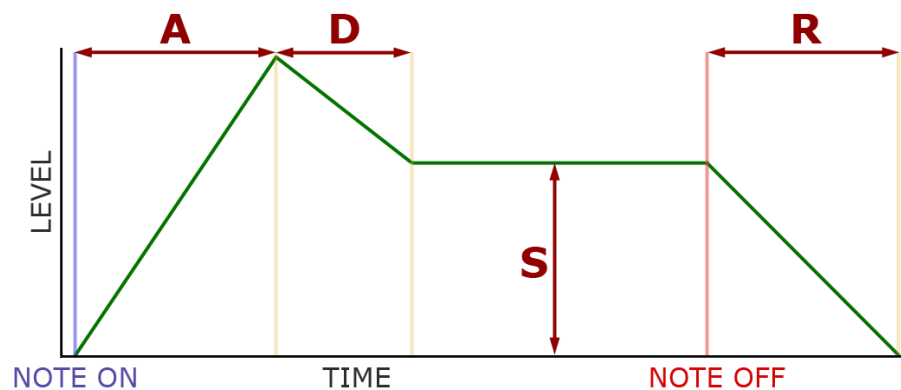


図 6 : ADSR エンヴェロープ

SID-Wizard では 4 つの HEX 値でこれらのパラメータを制御します。値の範囲は全て \$0 ~ \$F です。

**ADSR 0 6 A A**

図 7 : ADSR 値

マニュアル最後の表 [12.8 節] でおおよそのタイミングを見ることができます。アタック・タイムには 2ms (\$0) から 8s (\$F) が、リリース・タイムには 6ms から 24s が費やされます。マニュアルで Attack、Decay、Release のおおよそのタイミングを確認してみましょう。

### 3.4 ヴィブラート

SID チューンは、ヴィブラートなしにはありえません。SID-Wizard ではヴィブラートを次の 4 タイプから選択します。

TYPE-W 

ディレイあり 上昇・下降

TYPE-W 

ディレイあり 下降のみ

TYPE-W 

ディレイあり 上昇のみ

TYPE-W 

速度増加（ヴァイオリン・スタイル）

4 つの HEX 値でヴィブラートのパラメータを設定します。下図 8 をご覧下さい。

VIBR 1 8 8 0

図 8 : ヴィブラート・パラメータ

左から見ていきます。1 つ目の数字で振幅を、2 つ目数字で周波数（速度）を設定します。ディレイありか、速度増加のヴィブラート・タイプを選択している場合、右半分（残り 2 つの空色の数字）でディレイ（ヴィブラートの有効化にかかる時間）を設定します。

### 3.5 フィルター

前書きで説明した通り、SID チップには 3 つのチャンネルで共有されるフィルターが 1 つだけ存在します。ノート開始時には、チャンネルそれぞれにフィルターがかかります。フィルターがかかるのは、フィルター・テーブルの 1 行目が値ゼロのチャンネルだけです。値\$00 を有するインストゥルメントにのみ、フィルターがかかるのですが、フィルターのコントロールは行いません。1 行目がゼロでない場合、インストゥルメントはフィルター（バンド、レゾナンス、カットオフ周波数）のコントロールを行います。常に（フィルターのかけたインストゥルメントの）最も新しいノートが共有されたフィルター・バンド／レゾナンス／カットオフ周波数のコントロールを引き継ぎます（JCH エディタで聴くことのできる、ポリフォニックなジャミングに似ています）。フィルターのかけたインストゥルメントのノートが、同時に 2 つまたは 3 つ開始される時は、一番左のトラックのインストゥルメントがフィルターを制御します。

フィルターの編集にはフィルター（FILT.）・テーブルを用います。このテーブルの機能は WFARP テーブルとそっくりです。WFARP テーブルと同様、フィルター・

テーブルでも行を新たに挿入する時は **Shift + DEL (Insert)** キーを、行を削除する時は **DEL/Pound (Delete/Backspace)** キーを押します。

FILT.		KT
00	9F C0	.. ..
01	9F B0	.. ..
02	30 E8	.. ..
03	END	.. ..

図 9 : フィルター・テーブル

**\$8r~\$Fr xx** 第 1 ニブルでフィルター・バンドを設定。可能値は、**\$8** でフィルター無効化、**\$9** でロー・パス、**\$A** でバンド・パス、**\$B** でロー・パス+バンド・パス、**\$C** でハイ・パス、**\$D** でロー・パス+ハイ・パス [ノッチ・パス]、**\$E** でバンド・パス+ハイ・パス、**\$F** なら全てのモードを同時利用

第 2 ニブル (r) でレゾナンスを設定。**\$0~\$F** まで、どの値でも可能。当然**\$F** にすれば最も強いレゾナンス効果を得る。バンドが**\$F** (全てのモードを同時利用) に設定されている時は、r [レゾナンス値] を**\$F** に設定することはできない (その値がテーブル終端 [のシグナル] を示す**\$FF** 用に予約されるため)

第 3、第 4 ニブル (xx) でカットオフ周波数を設定。有効値は**\$00~\$FF**。おおよそのカットオフ周波数の範囲は **30Hz~10KHz**

\* **xx** が**\$80~8F** の間にある時は、第 2 ニブルでチャンネルのフィルター・スイッチを設定する [キーボード・トラッキング用]

**\$00~\$7F xx** 符号付き値 **xx** (**0~\$7F/\$80~\$FF**) をカットオフ周波数へ **0~127** (**\$00~7F**) 回加算/減算する。言い換えるなら、フィルター・スイープを生じさせる (スイープは **SID** のカットオフ周波数レジスタの **11** ビットを全て利用する)。正值を符号反転するには「**=**」キーを利用するか、マニュアル最後の、符号付き **10** 進値 (**-128~127**) 用の変換表を参照

**\$FE** テーブルのあるポジションにジャンプする (ポジションが **2** カラム目の場合は、それ自身へとジャンプ)

**\$FF** テーブル終端。この場所でテーブル実行は終了する。この値が入力されなかった時は、テーブル終端が **END** と見なされる

3つ目のカラムでキーボード・トラッキングのカットオフ周波数の強さを設定します。この設定を使うことにより、カットオフ周波数を音程に従属させることができます。このエフェクトはパルス幅テーブルでも同様に動作します。よりいきいきとした、広がりのある音色を作りだせるようになり、かくしてそれらは常に生楽器の印象を与える楽音の周波数付近で共鳴を起こします [resonate]。

キーボード・トラッキングの強さは、指数スケールにおける符号付き値になり、\$00~\$7F でカットオフ周波数 [およびパルス幅] は増加し、ノートは高音で再生されます。負値\$00~\$7F (-1~127) で同様のことが行えますが、劇的変化は望めません (言い換えるなら、それは勾配の緩いエフェクトです)。

### 3.6 他のインストゥルメント・パラメータ



図 10: インストゥルメント・パラメータ

#### 1 フレーム目の波形レジスタ

(1) Return (Enter) キーを押して有効/無効を切り替えます。有効時に値を変更することができます。デフォルトでこの設定は有効になっています。設定有効時にある値 (波形) を設定すると、その値は 1 フレーム目で再生される音の最初の波形になります (マルチスピードが設定されていようといなかろうと、いずれにしろ一番初めのフレームになります)。インストゥルメント・パラメータの主要な機能は GoatTracker と同様です。音が開始する前に波形 \$09 (テストビット設定) をともなったオシレーターをテスト/リセットしておけば、音は短く「セクシーに」鳴り始めます<sup>28</sup> (WFARP テーブルの 1 行目は短く鳴り、オールドスクールの曲のようにはなりません<sup>29</sup>)。

<sup>28</sup> 音の出力に対し「セクシー」という表現を用いることは、おそらく Cadaver の発言を踏まえている。  
<http://noname.c64.org/forums/?roomid=14&topicid=27005&showallposts=1>

<sup>29</sup> これは Hermit が実際に言っていたことです。[W]

バージョン 1.2 までは\$09 がハードウェアにより実現されていましたが、修正を加えることもできます。ですが、互換性を保つために、0 が予約されます。

## HARD-RESTART ADSR [ハードリスタート ADSR]

(2) ハードリスタート専用の ADSR 値で、ADHR とも呼ばれます（新しいノートがトリガーされる前に ADSR レジスタに 1~2 フレーム読み込まれます）。ノート・リリース後のアタック [のタイミング] について言うなら、SID はまったく正確性を欠いています。この問題を引き起こしているのは、ノートの前の ADSR レジスタのリセットです。「0F00」や「F800」が [ハードリスタートの] よく知られた値です。

SID チップの ADSR バグは多くの人々にとって（私も CSDb フォーラムでバグに関する書き込みを読んできましたが）謎です。該当スレッドから、CSDb 登録ユーザーの Frantic のコメントを抜粋します<sup>30</sup>。

「伝統的に言うなら……使用されているプレイヤー内部でハードリスタートにアクセスすること——あるいは実際に ADSR バグならびにハードリスタートがいかにか作用しているかを理解すること——は SID の世界において、エリートと聖人から下層階級のクズども [lowlife scum] やダサイ奴らを区別することにあたる。これゆえに、ハードリスタートを用いることは、たとえ秩序が保たれていとしても、[それぞれが当然のごとく預かることできる] 分け前からは吹き飛ばされていなければならないシーンの秘蹟なのだ。私はこの儀礼的実践の継続に一票を投じる」

## HARD-RESTART TIMER / TYPE [ハードリスタート・タイマー／タイプ]

(3) 1 つ目の箱（数値）は、「ハードリスタート・タイマー」で、SID の ADSR レジスタがリセットされるノートまで何フレームかかるかを設定します。2 つ目の箱はハードリスタート・タイプ（通常のハードリスタートまたはスタックカート／集合モード [aggregated mode] ）です。有効化（Return/Enter キーで切替）されているなら、ハードリスタートの際にはテスト・ビットも同時にリセットされます（その後続くノートに 1-2 フレーム分の際立った間隙を加えます）。

## DEFAULT ARPEGGIO TABLE SPEED [デフォルト・アルペジオ・テーブル・スピード]

<sup>30</sup> <http://noname.c64.org/csdb/forums/?roomid=14&topicid=92196&firstpost=2> [W]

(4) インストゥルメント・テーブルとコード・テーブルの実行速度を設定します。これはマルチスピードによって WF テーブルの実行をスロー・ダウンさせる必要がある場合、特に役立ちます。この値が\$40 より大きい時、パルス幅テーブルはマルチスピードになります。値が\$80 より大きい時、マルチスピードはインストゥルメント用のパルス幅テーブルとフィルター・テーブル双方で有効化されます。この値を\$00 に設定したままで、WFARP テーブルでスピードを設定できることも覚えておいて下さい (ARP カラムについて書かれた 3.1 節を参照)。

#### DEFAULT CHORD [デフォルト・コード]

(5) 指定したコードがインストゥルメントにデフォルトで結びつけられます。それだけでなく FX パターンでもコードは変化します。ここにはコードのナンバー (コード・テーブルから) を記入します。

#### OCTAVE-TRANPOSE AMOUNT [オクターヴ移調量]<sup>31</sup>

(6) ベースと高ピッチのサウンドを簡単に合体させることができます——ジャミング時にオクターヴを行ったり来たりジャンプする必要はありません。オクターヴ移調を行うには、ある番号を入力し、(X オクターヴ上へ、または下へ移調するには) + と - キーで切替を行える番号 (あるいは番号の前の箱) にカーソルを合わせて、Return (Enter) キーを押します。

#### INSTRUMENT NAME [インストゥルメント名]

インストゥルメント名を変更するには Shift + N キーを押します。名前を保存し編集を終了する場合は Return (Enter) キーを、名前を保存しないまま編集を終える場合は RUN/STOP (ESC) キーを押します。

### 3.7 インストゥルメントの保存と読込

インストゥルメントの保存と読込ができるのは、SID-Wizard の実に優れた機能だと言えます。この機能によって、他の曲で自分のインストゥルメントの再利用がしやすくなります。インストゥルメントを保存するには、保存したいインストゥルメントに変更後、F8 キーを押してメイン・メニューに移動します。これで「SAVE INSTRUMENT」が指定されたので、そのインストゥルメントに名前をつけ終わったら、Return (Enter) キーを押します。

---

<sup>31</sup> オクターヴ・シフトとも呼ばれる。

読込も保存の作業と似たようなものです。指定された曲のインストゥルメント番号は読み込んだ曲の番号に置き換わることを忘れずに。

インストゥルメントとパターンは、保存・読込時にバッファされています。ある曲のインストゥルメントを別の曲でも利用したい場合は、インストゥルメント・エディタで **C= + C (CTRL + C)** キーを押してインストゥルメントをバッファにコピーします。別の曲を読み込んだら、**C= + V (CTRL + V)** キーを押してバッファからインストゥルメントを貼り付けます。貼り付けができるのは、インストゥルメント・エディタ内だけです。

ファイル操作とメイン・メニューについては、第 9 章でさらに説明します。



## 4. コード・テーブル<sup>32</sup>

インストゥルメント・テーブルの右に位置する細長いテーブルには、指定されているコードのピッチ一覧が表示されています。SID-Wizardにおいてコードとは、本質的にアルペジオを指しているのですが、コードがインストゥルメントに依存することはありません。つまりどのインストゥルメントを選ぼうが、どのコードも使えるのです。コード利用の主眼は、1つのチャンネルでインストゥルメントを別のコードで鳴らすために、同じタイプのインストゥルメントを作成する冗長性／数を減らすことです。



Chord	01
00	0B
01	07
02	04
03	00
04	FF
05	7F

**C= + F5 (CTRL + F5)** キーを押せば、いつでもコード・テーブルに移動できます。利用方法はとても簡単で、**+/-**キーを押して、あるいはテーブルを呼び出さずとも **Shift + K/L** キーを押せばどこでもコードの選択が可能です。規定のタイミング外でピッチを揺らす命令を下したいなら、そのカラムで半音単位で数えられた相対ピッチを（16進フォーマットで）入力します。

**Insert/Delete** キーでコードのサイズ／複雑性を増加／減少できます。「7F」はコードのループを、「7E」はコードが呼び出された波形アルペジオ [WFARP] ・テーブルへの復帰と、次行での波形アルペジオ・テーブルの実行継続を意味してしま

<sup>32</sup> 当節は SID-Wizard 1.2 マニュアルの抜粋。 [W]

## 5. インストゥルメント作成例

インストゥルメント作成の基礎を知る手はずが整いました。いくつかの例を挙げながら、ゆっくりと説明していきます。<sup>33</sup>

### 5.1 ベース

まずは「標準的な」SID のベース [bazz] ・サウンドを作成してみましょう。インストゥルメント・エディタの WFARP テーブルに移り、行を新規挿入したら、1 カラム目に「41」と入力します。これでパルス波とゲート・オンが設定されます。

パルス波を選択したので、パルス幅テーブルでパルス幅を設定しなければなりません。パルス幅テーブルに移動したら、行を新規挿入し、1 カラム目に「82」と入力します。この状態で Z あるいは Q キーを押せば、インストゥルメントを試しに聴けます（キー入力時のノートは C です）。既定のオクターヴはベース・サウンドには高過ぎるので、オクターヴ移調量を-3 に変更したら、再度 Z キーを押してみましょう。

出力された音は少しばかり退屈なものです。PULSE テーブルに移り、最初に制作した行の後にもう 1 行を加えます。新しく追加した行の最初の 2 つのカラムに「40 20」と入力します。これでパルス幅は\$20、\$40 回まで増加しました。作成した音をもう一度試してみましょう。インストゥルメントを試しに鳴らす間、ユーザー・インターフェイスのインフォ・エリアにあるパルス幅に注目してみてください。それでは KT（キーボード・トラッキング）カラムの両方の行に「10」という値を挿入してみましょう。キーボード・トラッキングを用いる際は通常、どの行にも同じ値を設定します。できたらもう一度音を鳴らしてみてください。演奏するノートに従属して、パルス幅が変わるのがわかります。

ADSR エンヴェロープも少しひねってやるなら、値を「089A」に変更し、音を鳴らしてみましょう。リリースが長くなっているのに気づくでしょう。

---

<sup>33</sup> オクターヴ移調量の入力に関しては 3.6 節を参照。

こうしてできた音は聞けない音ではありませんが、ヴィブラートも少し加えてみます。ヴィブラート・パラメータを「**3720**」に設定し、ヴィブラート・タイプを「**ディレイあり 上昇・下降**」（デフォルトで選択されているはずです）に設定します。

格好良いベース・サウンドができたなら、締め括りにローパス・フィルター [**\$9**] を加えてみます。フィルター・テーブルに移り、**2** 行を新規挿入したら、**1** 行目に「**9B 45**」と入力します。これでカットオフ周波数が**\$45**、レゾナンスが**\$B** のローパス・フィルター**\$9** を作りだせます。インストゥルメントを鳴らすと、かなり暗く聞こえます。**2** 行目に、「**40 05**」と入力します。これでカットオフ周波数に**\$40** 回**\$05** を加算します（音が段々と明るくなっていくようになるでしょう）。さあ、これで本当に格好良いベース・サウンドが出来ました。

このインストゥルメントを保存し、次に進みましょう。インストゥルメントは後にパターン例を作成する際に使用します。

## 5.2 スネア

スネア・サウンドはウェーブ・テーブルを作成することで作ることができます。空インストゥルメントを用意したら、**WFARP** テーブルに移り、**5** 行を新規挿入します。**1** 行目に「**81 CE**」と入力すれば、正確なピッチ **CE**（本マニュアルの最後の表では **F-7**）でノイズ波（ゲート・オン）が生じます。

続いて**2** 行目に「**41 AD**」、**3** 行目に「**41 AC**」と入力します。これでピッチ **G#-4** と **G-4** で**2** フレーム分のパルス波が生成されます。スネアらしい音が聞こえるでしょう。

最後の**2** 列 [**4** 行目と **5** 行目] に「**80 C4**」、「**80 C5**」と記入すれば、ノイズ波がさらに**2** フレーム生成され、ゲートがオフに設定されることに気がつくでしょう。つまり **ADSR** の **R** 値にしたがって、たとえノート・オフを送らなくても（あるいはジャミングしている際にキーを離しても）、音が**4** 行目でフェイド・アウトし始めるのです。

パルス波を使用しているので、パルス幅を設定する必要があります。**PULSE** テーブルに移り、**1** 行挿入したら**1** カラム目に「**88**」と記入します。出来たスネアを試し

に聴いてみましょう。ブツッと音が途切れますから、ADSR エンヴェロープを編集しなくてはなりません。

Decay を「D」に、Sustain レベルを「E」に、Release を「9」に変更します。このリリース・パラメータによって、途轍もなく大きな変化が現れます。スネアは美しいノイズへとスムーズにフェイド・アウトしていくでしょう。スネア・ドラムが作成できました。WFARP テーブルで正確なピッチを使用している [厳密に指定されたただ一つのピッチしか指定していない] ので、どのノートでも同じように音は鳴ります。

### 5.3 アルペジオ

最後のインストゥルメント実例として、アルペジオを作成してみましょう。まずはインストゥルメントを再び消去し、WFARP テーブルに 1 行新規挿入したら、1 カラム目に「11 7F」と記入し、三角波を生成します。すると 2 カラム目の値によってインストゥルメントのデフォルト・コードが鳴り始めます。

次にコード・テーブルに移り (C= + F6 または CTRL + F6 キーを押します)、6 行を新たに挿入します。(1 行目から)「00」、「03」、「07」、「0C」、「07」、「03」と入力して行きます。7 行目には今まで通り、コードのループを指示するために「7F」と示すべきところです。編集したコードが、インストゥルメントのデフォルト・コードと同じ番号かどうかを確認して下さい。

Z<sup>34</sup>キーまたは Q キーを押してインストゥルメントを試しに聴いてみましょう。そうするのが手っ取り早いからです。インストゥルメントの修正には二つ方法があります。一つはインストゥルメント・パラメータでコード・スピードを高値 (試しに「02」と記入してみてください) に変更する方法です。もう一つは、コード・スピードを「00」のままにして、先ほど挿入した「7F」の左に位置する 1 カラム目に、ある値 (例として再び\$02) を入力することによって、スピードを無視するという選択肢もあります。<sup>35</sup>

---

<sup>34</sup> HEX 値の範囲にないノート・キーのどれでもインストゥルメントをプレビューすることができます (1~8 および A~F) 。 [W]

<sup>35</sup> この 2 つ目の方法については、誤記の可能性があるので (osoumen 氏の教示) 。

Attack を「8」に、Release を「B」に設定してスムーズなアタックや長いリリースに変更するという方法も良いでしょう。

## 6. パターン・エディタ<sup>36</sup>

スクリーンの大部分を占めるのは、パターン・エディタ・ウィンドウです。ここでは SID ルーチンが再生可能な 3 つのトラック／チャンネルに対応するパターンを、確認しながら編集することができます。それぞれのトラックには 4 つのカラムがあります。1 カラム目 (1) は、ノート／ピッチのコントロールとノート・オン／ノート・オフ (7) と (更に加えられた他の) エフェクトを示してします。2 カラム目 (2) の主な機能は、インストゥルメントや「レガート」の選択です。ここにもスモール・パターン・エフェクトが含まれることがあります。3 カラム目 (3) と 4 カラム目 (4) は明確にパターン・エフェクト用にデザインされています。ただし、4 カラム目が必要になるのは、「ビッグ・エフェクト」を利用する時だけです。このエフェクト利用時には、カラムにはエフェクト値が含まれることとなります。



図 11 : パターン・エディタ

F5 キーを押せばいつでもパターン・エディタ・ウィンドウに移動します。カーソル・キーと「CTRL」キー (Vice エミュレータでは TAB キー)、さらにはページ・アップ／ページ・ダウン (C64 ではスラッシュ [/] /Shift + スラッシュ)、Home キー等々、見た目に分かりやすい他のキーを用いて、容易に 3 つのトラックの間を行き来できます。カーソルがインストゥルメント番号のポジションに据えられている時、Return (Enter) キーを押せば対応するインストゥルメントのエディ

<sup>36</sup> 当節の大部は SID-Wizard 1.2 マニュアルからの抜粋。[W]

タ・パネルに飛びます。パターンの最終行（「-end-」のシグナルが表示されているところ）で **Insert/Delete** キーを押すことで、パターンの長さ／サイズを自由に増減可能です。パターン番号とパターンのサイズはインフォ・エリア（**(5)** および **(6)**）に表示されます。

また「自動前進 [スクロール]」の間隔 [*amount*] を変更するには、**Shift + A**（増やす）または **Shift + Z**（減らす）を使用します。ノートカラムであるノートを記入後には、指定された行間隔分、カーソルがジャンプします。**C= + B** キーで「連動モード」に切り替えれば、トラック／パターンそれぞれを独立してスクロールさせることも、他のトラック／パターンと一緒に移動／スクロールさせることもできます（スクロール可能な程の長さがあるならずっと）。ミュート／ソロにしたいトラックがある場合は、トラック番号に応じて **Shift + 1/2/3** キーを押します。パターン・エディタ・ウィンドウにおける完璧な使用キーの概要をお探しなら「キーボード・レイアウト」の節 [実際には **12.2 節**] を確認するのも良いでしょう。

ノートを挿入するには、ノートを置きたいと思う行の **1** カラム目にカラムを移動させてから、挿入したいノートに対応するキー（下記、**6.1 節**参照）を押します。上手くいかない場合は、**SPACE** キーを押してエディット・モードを有効化しておく必要があります（**SPACE** キーでエディット・モードとジャミング・モードを切り替えます。カーソルがゆっくり点滅しているならジャミング・モードになっている証拠です）。ノート・オフを挿入するには、ノート・オフを置きたいと思う行で **Return** (**Enter**) キーを押します。**1** または **A** キーでカーソル・ポジションにあるノートを消去できます。パターン・エディタで使用するキーを知るには、このマニュアルの最後のキーボード・レイアウトを確認してみてください。

## 6.1 ノートカラム・エフェクト

ノートカラム (1) ではピッチ/ゲーティングに関連したエフェクトを保持することができます。これらの状態はノートカラムにグラフィカルに表示され、前述したキーボードの組み合わせによるパターンに置き換えることができます。組み合わせのほとんどは、簡単なスイッチです（シンクのオン/オフ、リング・モジュレーションのオン/オフ、トーン・ポルタメントのオン）。ただヴィブラートだけは、振幅値を規定するニブルを 1 つ (0~F) 有しています。

ノート	<b>2 3 5 6 7 9 0</b>	(黒鍵—上位オクターヴ)
	<b>Q W E R T Y U I O P</b>	(白鍵—上位オクターヴ)
	<b>S D G H J L</b>	(黒鍵—下位オクターヴ)
	<b>Z X C V B N M , .</b>	(白鍵—下位オクターヴ)

ノート・オフ	Return (Enter)
オートポルタメント	Shift + P
ヴィブラート	Shift + V
リング・モジュレーション オン	Shift + R
リング・モジュレーション オフ	C= + R (CTRL + L)
シンク オン	Shift + S
シンク オフ	C= + S (CTRL + S)

## 6.2 インストゥルメントカラム・エフェクト

インストゥルメント・カラム (2) ではスモール・エフェクトを適用することができます。カラムの第 1 ニブルでエフェクト・タイプを、第 2 ニブルで対応する値 (0~15 または \$0~\$F) を設定します。これらスモール FX 番号のほとんどは、覚えやすいように、エフェクトカラムのスモール FX 番号とビッグ FX 番号と同じになっています。

**\$01~\$3E** ノート編集に使用するインストゥルメントの変更。[再度] このコマンドで別のインストゥルメントに変更するまで、選択は解除されない。またこのコマンドは、**C= + P (CTRL + P)** キーまたは **C= + F (CTRL + F)** を押してオフ状態なり再スタートしたテーブルを

リセットする。この場合は同じインストゥルメントの連続指定 [再選択] にも意味があるが、ただし「**auto.instr** [自動インストゥルメント入力]」がオンになっているなら場合は利用できない

- \$3F** ノートの連結 [**Tied note**] (正確な意味でのレガート。インストゥルメントが新たに開始することではなく、単に音程だけを変える [キーオンせずに、ピッチ変更の場合、エンヴェベロープ等は再スタートされずに継続される])
- \$40~\$4F** 波形 (SID レジスタ 4) のニブル調整——後に続いて実行される WF テーブルでの波形変更が優先される
- \$50~\$5F** ADSR の **Sustain** のニブル調整 (スモール FX)
- \$60~\$6F** ADSR の **Release** (SID レジスタ 6) のニブル調整 (スモール FX)
- \$70~\$7F** インストゥルメントの用のコード (デフォルト・コードに優先する) を選択する (コード毎にインストゥルメント作成が不必要である代わりに、[最低でも] 一つのインストゥルメント作成が必要。ARP テーブルで**\$7F** を入力してコードを呼び出したら、このパターン FX でコードを選択すること)

### 6.3 エフェクトカラム スモール・エフェクト

スモール・エフェクト (3) の第 1 ニブルでオペレーションのタイプを規定します。第 2 ニブルではオペレーションに対応するエフェクト値/量を設定します。覚えやすいように、スモール FX 番号**\$4~\$7** は、同範囲のビッグ・エフェクト番号と機能が同じになっています (さらに言うなら、エフェクトの HEX 番号は SID レジスタやエフェクト名と関連性があります)。「Extra」プレイヤー・バージョンを利用しない限り、1 ノート毎に許可されているスモール FX の ADSR 設定は 1 つのみです。

- \$20~\$2F** ADSR の **Attack** ニブル調整 (スモール FX)
- \$30~\$3F** ADSR の **Decay** ニブル調整 (スモール FX)
- \$40~\$4F** 波形 (SID レジスタ 4) のニブル調整——後に続いて実行される WF テーブルでの波形変更が優先される
- \$50~\$5F** ADSR の **Sustain** ニブル調整 (スモール FX)
- \$60~\$6F** ADSR の **Release** (SID レジスタ 6) ニブル調整 (スモール FX)



- \$70~\$7F**      インストゥルメントの用のコード（デフォルト・コードに優先する）を選択する（コード毎にインストゥルメント作成が不必要である代わりに、[最低でも]一つのインストゥルメント作成が必要。ARP テーブルで\$7F を入力してコードを呼び出したら、このパターン FX でコードを選択すること）
- \$80~\$8F**      ヴィブラートの振幅の調整——周波数を変えない
- \$90~\$9F**      ヴィブラート周波数の調整——振幅は変わらない
- \$A0~\$AF**      メイン・ヴォリューム 調整（\$D418 [SID のレジスタ・アドレス] の下位ニブル）
- \$B0~\$BF**      フィルター・バンド（ロー・パス/バンド・パス/ハイ・パス/フィルター・オフ）（フィルター・テーブルの実行で書き換え可能）
- \$C0~\$CF**      コード・スピード（メリハリの利いたアルペジオを利用している場合はアルペジオ・スピード）の調整
- \$D0~\$DF**      一定量で編集集中のノートにデチューンをかける
- \$E0~\$EF**      テスト・ビット/リング・ビット/シンク・ビット/ゲート・ビットの有効化/無効化（波形テーブル優先）
- \$F0~\$FF**      フィルター・レゾナンス（強さ）のニブル設定（後に続くフィルター・テーブル・コマンドが優先される）

## 6.4 エフェクトカラム ビッグ・エフェクト

ビッグ・エフェクトとは、エフェクトカラム (3) 内で利用可能なエフェクト・タイプです。一つのパターンの一番最後のカラム (4) にビッグ・エフェクトの 1 バイト（\$0~\$FF または-\$7F~ + \$80）の値を置きます（\$4~\$7 までのエフェクトは、覚えやすいように、同じ範囲のスモール・エフェクトのタイプと同一です）。

- \$01**            上方にピッチ・スライド（GoatTracker と同一の FX 番号）
- \$02**            下方にピッチ・スライド（GoatTracker と同一の FX 番号）
- \$03**            （一定のスピードでの）トーン・ポルタメント（GoatTracker と同一の FX 番号）
- \$04**            波形制御レジスタ（\$D404 等）の設定（または\$F0 以上で WFARP テーブルに移動する）
- \$05**            **ATTACK/DECAY**（SID レジスタ 5）の 1 バイト調整

<b>\$06</b>	<b>SUSTAIN/RELEASE</b> (SID レジスタ 6) 1 バイト調整
<b>\$07</b>	インストゥルメントに用いるコードの選択 (デフォルト・コードを上書きする)
<b>\$08</b>	ヴィブラートの振幅と周波数の調整 (インストゥルメントのデフォルトのヴィブラートに優先する)
<b>\$09</b>	現在選択されているインストゥルメントの波形アルペジオ・プログラムのテーブル・インデックスに分岐する
<b>\$0A</b>	現在選択されているインストゥルメントのパルス幅プログラムのテーブル・インデックスを調整する (テーブル・ポジションに移動する)
<b>\$0B</b>	現在選択されているインストゥルメントのフィルター・プログラムのテーブル・インデックスに分岐する
<b>\$0C</b>	コード・スピード (メリハリの利いたアルペジオを利用している際はアルペジオ・スピード) の調整
<b>\$0D</b>	編集集中のノートに一定量のデチューンをかける
<b>\$0E</b>	パルス波用のシンプルなパルス幅の設定 (インストゥルメントの PW テーブルの変更が優先される)
<b>\$0F</b>	シンプルなフィルターのカットオフ周波数の設定 (キーボード・トラッキングを考慮したエフェクト)
<b>\$10</b>	メイン (サブチューンの基本となる) ・シングルテンポの設定
<b>\$11</b>	メイン (サブチューンの) ・ファンクテンポの設定。左/右のニブルに、1 番目 (偶数行) のテンポと 2 番目 (奇数行) のテンポを表示
<b>\$12</b>	全てのサブチューンに適応されるメイン・テンポ・プログラムの設定 (それぞれの行にどんなスピードも定められる)
<b>\$13</b>	トラック個々のシングルテンポの設定
<b>\$14</b>	トラックのファンクテンポの設定 (左/右のニブルに偶数行と奇数行のテンポを表示)
<b>\$15</b>	トラック個々のテンポプログラムの設定
<b>\$16~\$1C</b>	ヴィブラート・タイプの選択 (可能値: \$00、\$10、\$20、\$30)
<b>\$17~\$1B</b>	後の拡張用に保持

- \$1C**            フィルターのカットオフ周波数の上位バイトシフト（フィルター周波数に加えられる）。曲の初期化 [**tune-init**] によってのみ、リセットが行われる<sup>37</sup>
- \$1D**            \$00～\$FF（20ms）フレーム単位でトラックにディレイをかける（ディレイがかかるのは「Extra」バージョンのみ）
- \$1E**            フレーム単位で編集集中のノートにのみディレイをかける（最大値はテンポから 3 を引いた値になる）
- \$1F**            フィルター・スイッチ（外部フィルターも）とレゾナンスのダイレクト設定（レジスタ\$D417）

## 6.5 パターン作成例

この節では、簡単なパターンの作成方法を解説します。[事前に] 手頃なパターン・エディタのキーボード・ショートカットの一覧を用意しておくといいでしょう。**SPACE** キーでエディット・モードとジャミング・モードが切り替わることを覚えておいて下さい（カーソルがゆっくりと点滅しているなら、ジャミング・モードが有効になっている証拠です。ノートが書き換わることはありません）。

まず、第 5 章で保存した「ベース」サウンドを読み込みましょう。保存していないのなら、新たに作成します。

そしてパターン・サイズを「40」まで上げます（最初のトラックのパターン終端に移動し、番号「3F」の表示が見える最終行の手前に至るまで、**Shift + DEL**（**Insert**）キーを押し続けます）。

1 行目にノート **C-6** を挿入します（カーソルを 1 カラム目に置き、**Q** キーを押してジャミングのオクターヴを 5 に設定します）。値 **C-6** のすぐ右横に移動したら、ベース・インストゥルメントの番号を入力します。

**\$10** 行目で **G#5** (**H**) を、**\$20** 行目で **F-5** (**V**) を、最後に**\$30** 行目に **G-5** (**B**) を挿入します。インストゥルメント変更時にインストゥルメント番号の設定が必要

---

<sup>37</sup> 音楽ルーチンにおいてサブチューンの初期化へジャンプする時（通常は **jsr \$1000** によって）、命令 **tune-init** は **SID** のリセットを行う。このリセットにより、ヴォリュームの設定ならびに曲の始まりのポインタ、変数の初期化が行われる。

なだけで、ノート毎に番号を設定する必要がない点を覚えておいて下さい。パターンを演奏／ループするには **F3** キーを押します。

それではポルタメントとピッチ・スライドを試してみましょう。**\$10** 行目に移動したら、スピード「**25**」で、ビッグ・エフェクト、トーン・ポルタメント (**\$03**) を挿入します。**\$10** 行目の最後 2 つのカラムには「**03 25**」と記入します。**\$37** 行目に値「**20**」でビッグ・エフェクトのピッチ・スライド・ダウンを挿入します。**\$37** 行目の最後 2 つのカラムに「**02 20**」と記入します。そして**\$3C** 行目でノートオフ・コマンド (1 カラム目にカーソルを合わせ **Return/Enter** キーを押します) を挿入したら、もう一度パターンを再生してみましょう。

スロー・ポルタメント (10 行目のノート **G#5**) を使っている時は、ノートが少し早目に再生されるように調整した方が良いでしょう。ノートが遅れて始まるようなら、うまく鳴らないからです。**\$10** 行目で **DEL (Backspace)** キーを押し、記入値を削除したら、ノート **G#5** を挿入し、代わりに**\$0C** 行目にポルタメント効果を挿入します。

## 7. オーダーリスト<sup>38</sup>

スクリーン下端の最下位にあるオーダーリストを紹介します。オーダーリストでは、それぞれ後にパターンの通し番号を置くことで、曲の全体的配置を制御できます。オーダーリストの再生されている最中は、曲の再生ポジションが点滅します。あなたの音楽のタイムラインの現在地を追うのに役立ちます。

**F6 (Shift + F5)** キーで、いつでもオーダーリストに移動できます。すぐ目につくところにナビゲーション・キーが存在します。カーソルのある場所で **Insert/Delete** キーを押すことで、個々のトラック・シーケンスの長さの増減が可能です。パターン・エディタ・ウィンドウに戻る際に、オーダーリストのパターン番号の上で **Return (Enter)** キーを押すことで、対応するトラックのパターンが選択されます。**Shift + Return (Shift + Enter)** キーで、編集中のオーダーリストのカーソル・ポジションのパターン 3 つ全てを選択できます。その時に再生されているパターンを選択するには、随時 **C= + Return (CTRL + Enter)** キーを押します (他の

---

<sup>38</sup>当節は **SID-Wizard 1.2** マニュアルの抜粋。[W]

トラックと異なったスピードや長さを用いている場合に〔パターンに追従させるのに〕便利です)。

\* **Version 1.2** で加わった新機能について：オーダーリストで **C= + E** を押すことで、存在するオーダーリストの後の一番始めにある空（未使用）パターンを検索することができます。パターン終端の徴に **\$FE/\$FF** を使用していた場合は、オーダーリストの終端に自動的に空パターン番号が挿入されます。**C= + E** キーを押すことで番号は増加し、新しく編集するパターンに移動する度にリフレッシュされます。これで以前よりもずっと早く曲を発展させることができます。

値「**FE**」は定められたトラックの再生終端を示しています。値「**FF**」は、その値の後に続く（書かれた）番号で与えられる、パターンシーケンス内のあるポジションにジャンプします（値「**00**」は「処理なし」のために予約されているため、「**00**」をパターン番号として用いることはできません<sup>39</sup>。〔従って〕パターンの最小値は「**01**」になります）。これらが **16** 進数であることに気をつけて下さい。例えば、「**09**」の後には「**10**」ではなく「**0A**」が続きます。

\* 転調やヴォリューム／テンポ変更のようなオーダーリストから操作されるエフェクトもいくつか存在します。

## 7.1 オーダーリスト・エフェクト

<b>\$00～\$7F</b>	オーダーリストの標準パターン番号のために保持されている値
<b>\$80～\$8F</b>	キー・ダウン移調（例： <b>\$8F</b> =半音下げる、 <b>\$8E</b> =全音下げる）
<b>\$90</b>	移調オフ（オリジナル・キーに戻す）
<b>\$91～\$9F</b>	キー・アップ移調（例： <b>\$91</b> =半音上げる、 <b>\$92</b> =全音上げる）
<b>\$A0～\$AF</b>	メイン・ヴォリュームの設定（ <b>\$0</b> ～ <b>\$F</b> ）
<b>\$B0～\$FD</b>	対応するトラックのトラック・テンポの設定（テンポ <b>\$00</b> ～ <b>\$5D</b> と同等の範囲）
<b>\$FE～\$FF</b>	シーケンス・フロー制御のために予約されている値。 <b>\$FE</b> で再生停止し、ポジション <b>\$FF</b> に到達するとループする。 <b>\$FF</b> の後に <b>\$80</b> を下回るポジション番号が記入された場合、編集集中のサブチューンの規定のポジションまでループする。あるトラックのポジション番号

<sup>39</sup> 「**FF**」の後に「**00**」を記入すると「-」と表示されるので、パターン番号「**00**」がただ存在しないだけでも考えられる（osoumen氏の示唆）。

が\$80を上回る場合は、規定のサブチューンの対応するトラック／シーケンス（番号から\$80を引いたポジション）にジャンプする（デモ・パートに使用される曲等、サイズの小さい曲の制作に有用）

サブチューン・ジャンプを完了するためには、全てのトラックにこのコマンドを入力する必要があります。トラックごとに個別に操作が行われるためです。サブチューン・ジャンプでは、対象となるサブチューンのオーダーリスト／パターンにテンポ変更エフェクトが適用されない限り、現在のテンポが保たれます……（デフォルトのサブチューン・テンポは新規に曲を作成し始めた時にだけ有効です）。

\*注記：オーダーリスト・エフェクトはループ・シグナル\$FFの寸前に置かなくても構いません。エフェクトが生じれば、オーダーリストFXとループとの間の素早いジャンプに起因するフリーズを防止するため、ループ・コマンドは無視されるからです。

## 8. テンポとマルチスピード

曲のデフォルトのテンポは「Shift + .」と「Shift + ,」で変更できます。C= + Tを押すことでファンクテンポとの切替が可能です。ファンクテンポの2つ目のテンポを変更するにも、「Shift + .」と「Shift + ,」キーを利用します。

ファンクテンポでは、パターンがステップを踏む度に2つのテンポが入れ替わります。ファンクテンポが有効化されているなら、パターン内で一方の行は1つ目のテンポで、片方の行は2つ目のテンポで再生されます。

テンポはパターン・エフェクトやオーダーリスト・エフェクトでも変更できます（第6章、第7章を参照）。

### 8.1 テンポプログラム・テーブル

スクリーンの右隅のテーブルには選択されたテンポプログラムが表示されています。ここでは極端なパターン・タイミング／スウィング／ファンクテンポ用の入り組んだテンポプログラムを設定できます。パターンFX指定後にテンポプログラムが選択

される場合、行全ては [パターン FX に優先して] テンポプログラムが適用される結果生じるパターンの行のタイミングに対応します。通常は複雑なテンポプログラムもファンクテンポも必要ありませんが、特殊なリズムが頻繁に現れる曲には利用価値があるでしょう……。

**C= + F7 (CTRL + F7)** キーを押せばいつでもテンポプログラム・テーブルに移動できます。テンポプログラムを選択するには随時 **Shift + T/Y** キーを押すか、テンポプログラム・テーブル内で **+/-** キーを押します。\$7F (「6」) よりも大きい番号が許可されないことをのぞけば、編集方法はコード・テーブルと同じです。テンポプログラムは自動的に一番最後の値の後に転がり込みます。特別なジャンプ・コマンドは必要ありません。

## 8.2 マルチスピード

マルチスピードとは、PAL 50Hz (または NSTC 60Hz) の 1 フレーム内で複数回レジスタを変更することで、SID のきめ細かい/高速制御を可能にするテクニックのことです。[PAL マシンの場合] 8x マルチスピードでは、SID は周波数 400Hz に変更され、波形/アルペジオ・テーブルは高速度で実行されます……。そうはいつでも、2x スピード (100Hz) でもかなりの効果を期待できます (例えば、Jeroen Tel の作曲した Rubicon のタイトル・テーマは 2x です。50Hz の 1x のシングルスピードでは実現不可能なドラムを使用しています)。

プログラムが違えばマルチスピードの扱い方も違ってきます。X-SID や SID-Wizard では、マルチスピードで実行されるのは (波形/パルス/フィルター) テーブルだけです。テンポやヴィブラート/[ピッチ・] スライドの操作は 1 フレームに一度だけ、プレイヤーがシングルスピードの個所でのみ呼び出されます。GoatTracker や全てのプレイヤーが呼び出される他のトラックとは全く反対です。どちらの解決策にも長所と短所があります。SID-Wizard が採った解決策の唯一の短所は、きめ細かいスケールで曲のテンポを設定できない点です。その一方でマルチスピードを用いても、テンポ/ハードリスタート/ヴィブラート/[ピッチ・] スライドの設定が無効化されない長所もあります。

SID-Wizard は 8x (400Hz) マルチフレームスピードまでサポートしています。

(Jeff の X-SID のように) マルチスピードがスピード値やテンポに作用することはありません。この設定はメイン・メニュー (次章参照) から変更できます。



## 9. メイン・メニューとファイル操作

前章で言及したように、F8（もしくは Shift + F7）キーを押すことでメイン・メニューを呼び出せます。メニューを閉じる際は、もう一度 F8 キーを押します。カーソル・キーの↑/↓でオプションを選択し、Return (Enter) キーで設定に移ります。デフォルトのパターン長さを選択するメニュー項目「DEF.PATT.LEN」だけが唯一の例外で、+/-キーで編集の曲のデフォルトのパターン長さを指定します。

```
*****  
* - MAIN MENU - *  
* ----- *  
* SAVE TUNE *  
* LOAD TUNE *  
* SAVE INSTRUMENT *  
* LOAD INSTRUMENT *  
* SELECT DRIVE:08 *  
* COLOR: DEFAULT *  
* NOTE-KEYS: 2ROW *  
* AUTO-INSTR: OFF *  
* AUTOFOLLOW: OFF *  
* HIDE RBARS: OFF *  
* FRAMESPEED: 1X *  
* DEF.PATT.LEN:20 *  
* KILL TUNE&TEMPO *  
* KILL INST&CHORD *  
* INSTANT HELP! *  
*****
```

図 12: メイン・メニュー

メニューから何らかのアクションを選択中であっても（再生やインストゥルメント変更に関係する）キー動作のいくつかは受け付けてくれます（メニューがパターン再生ディスプレイの上に表示されている間は、ディスプレイがリフレッシュされることはありません）。ファイル操作やメニューを終了した後もなお、最後に利用したメニュー項目は記憶されています。

### 9.1 簡単な設定

#### SELECT DRIVE

ファイル操作を行うアクティブ状態のディスク・ドライブを選択します（Vice でシームレスなフォルダ統合環境を利用している場合は、特に気にする必要はありません）。

#### COLOR

ユーザー・インターフェイスのカラー・テーマを変更します。

#### NOTE-KEYS [ノートの入力方法選択]

2 行 (GoatTracker/FastTracker) モードと DMC/GMC<sup>40</sup>ライクのピアノ鍵モードの切替。SID-Wizard ではデフォルトで 2 行モードが設定されています。

#### AUTO-INSTR [インストゥルメントの自動入力]

この設定がオンになっていると、パターン・エディタでノートを入力時に現在指定されているインストゥルメントが自動的に追加されます。保存時にオンのままにしておくと、インストゥルメントの不要な連続指定が消去され、保存される曲のサイズを減らしてくれます。オフの時 (C64 エディタでは標準的な方式です) は、手動でインストゥルメント変更「コマンド」を記入しなくてはなりません。この場合は、ファイルの保存時にインストゥルメントの連続指定のコマンドの最適化もなされません。

#### AUTOFOLLOW [再生ポジションの追跡]

設定をオンにすると、カーソルは自動的に再生ポジションを追いかけます。

#### HIDE RBARS [ラスター・バーを隠す]

ユーザー・インターフェイスの横に位置するラスター・バーを隠します。

#### FRAMESPEED

マルチフレームスピード [*multi-framespeed*] を設定します (8.2 節参照)。

#### DEF.PATT.LEN [デフォルト・パターン・レングス]

パターンの長さの標準値 [*Default Pattern Length*] を設定します。

#### KILL TUNE AND TEMPO [曲とテンポの消去]

編集集中の曲を消去し、テンポをデフォルトに設定し直します。曲を消去する前には、SID-Wizard に曲を初期化後デフォルトで使用するパターンの長さを指示するためには、メニューポイントの「**default pattern length** [パターンの長さの標準値]」で +/- キーを使ったほうが良いでしょう。パターンの長さの標準値は **.SWM**

---

<sup>40</sup> GMC = Game Music Creator。Graffity のミュージック・エディタ DMC は当初 GMC としてリリースされた。

のワークファイルに情報として保存されるので、後に曲をロードした時は、空パターンは自動的にこの値にサイズ調整されます（その後サイズを手動で設定することもできます）。

## KILL INST&CHORD [インストゥルメントとコードの消去]

全てのインストゥルメントとコードを消去します。

## INSTANT HELP!

ええと、何のことだったかな？

## 9.2 曲（ワークチューン）またはインストゥルメントを読み込む／保存する

「SAVE」や「LOAD」の項目を選ぶと、PC でよく見るファイル・ダイアログが起動するはずですが。ここでは直に希望するファイル名を入力したり、ディスク・ディレクトリからそのファイルを選択したりする（そしてさらには編集する）ことができます（インストゥルメントの読込や保存がしたい時には、あらかじめエディタ／メニューで+/-キーを押して、インストゥルメント（読込／保存操作の対象となる）に変更されているかどうかを確認して下さい）。

ディスク・ドライブが読み込まれると、3 つのカラムには [指定の拡張子で] マスクされたファイル名が配置されます。スクリーンの一番下でディスク名とディスクの空き容量（キロバイト単位）を確認することもできます。ファイル名には 12 文字まで使用できます。「.SWM」あるいは「.SWI」（「M」は「module」を、「I」は「instrument」を意味しています）といった拡張子が後から自動的に付いてくるためです。SID-Wizard ではディスク・ディレクトリ表示の際に、それらの拡張子でファイル名のフィルタリングを行います。デフォルトのディスク・ドライブはメニューに既に選択されている（デバイス 8～15 の範囲で）はずですが。

ファイル・ダイアログの左上角に見えるのは、エディタ・プレイヤー・バージョンの番号です。この先（あるいはことによると）より新しい／別のモジュール・バージョンを読み込みたい場合は、この番号についてよく知っておいて下さい。なぜ

ならモジュールのフォーマット・バージョンは常にエディタのエクスポーター<sup>41</sup>のバージョンと一致している必要があるからです。

PC で.D64 イメージを使用している場合は、ファイルの上書きができません。Vice のシームレスなフォルダ統合環境を利用者に注記しておきます。同名の一つのファイルである場合を除いて、拡張子 P00 ならびに S00 のファイルはどちらも上書きはされませんが、[新たに] 拡張子 (.P01、.P02) が作成されます。既存のファイルと同名にしてファイルを保存しないようにして下さい。

### ファイル・ダイアログで使用可能なキー

F1	ディスク・ディレクトリを再読込、ディレクトリの先頭に戻る
F3/↑ (F3/Page-Down)	ディレクトリ内のページ・ダウン (1 スクリーンに収まりきらない場合)
CTRL (TAB)	ファイル名入力/作者情報入力/ファイル名セレクトの切替 (作者情報は曲の保存時に編集可能)
↑/↓/←/→	言わずもがな、入力域/ファイル・セレクト領域におけるカーソル操作を行う
Return (Enter)	ディスク操作を行う (または作者情報エディタを後にする)。問題発生時 (ファイルが存在する/見つからない) には通知が出る
F7/STOP (F7/F8/Esc)	ファイル操作を行わず、ファイル・ダイアログからエディタに復帰する

### SWM ファイルフォーマット

.SWM フォーマットは SID-Wizard の最適化された/圧縮された独自フォーマットであり、他のいかなるシステムとも互換性はありません。ですが必ずこの先の SID-Wizard の [新しい] バージョンとは、ずっと互換性は保たれます (不適格なバージョンかどうかエディタに通知するための、「SWM1」というファイルタイプ、バージョン文字列を含みます)。

(Shift + I キーの同時押しかメニューで) 「インストゥルメント自動入力」モードが選択されている場合、パターン・サイズの減少のために、保存前にインストゥルメントの不必要な連続指定が削除/最適化されます。

.SWI フォーマットは、指定されているインストゥルメント (空/未使用領域を除いた) のコピーです。

---

<sup>41</sup> エディタ : Sid-Wizard、エクスポーター : SID-Maker [W]

## 9.3 曲やインストゥルメントを初期化／消去／削除する

2つのメニュー項目でサブチューン（パターン、テンポ）やインストゥルメント／コードを個別に消去／初期化できます。曲を消去する前に、メニュー項目の「default pattern-length」で+/-キーを押すことで、その曲の初期化時に **SID-Wizard** がデフォルトで指定するパターンの長さを決めることもできます。パターンの長さは **.SWM** ワークファイルに譲歩として保存され、後で曲を読み込んだ時に、空パターンは自動的にその値に調整されます（ですが、サイズは後で手動で設定することもできます）。

## 10. SID-MAKER（制作した曲のエクスポート）<sup>42</sup>

このプログラム [**SID-Maker**] は **SID-Wizard** ディスクに含まれている（もしかしたら別個にダウンロードされている）、単独実行可能なアプリケーション・ファイルです。なぜ別のファイルに分かれているのかといえば、**SID-Wizard** エディタのコードに要する領域を記憶するために、プログラム再配置がほとんど実現不可能なことによります。このプログラムを起動するのが、曲をリリースする用意が整った時点であり、おそらくは一番最後の段階であると考えられることから、私はこのことをそれ程大きな問題だとは捉えていません。

デバイス（ディスク・ドライブ）には、読み込まれた **SID-Maker** が一つあるはずですが、**SID-Maker** は同時に 4 種のファイル出力を行うことができます。まずは **C64** ネイティブの「元から存在する」3つのフォーマット（「**.C64.PRG**」、**.BIN.SEQ**、そして実行可能フォーマット「**.EXE.PRG**」）と、**Star Commander**<sup>43</sup>や **Total Commander**<sup>44</sup>の **D64** プラグインのような、クロスプラットフォームのファイルコピー・ツールでファイルを転送した際に、クロスプラットフォーム環境での再生を可能にする「**.SID.SEQ**」（**SID**）です（**.PRG** や **.SEQ** は **CBM-DOS** で付加された標準的な拡張子です。**SID** ファイルを作成したい時は——各自コピー・ツールを使用して——、エクスポートされた曲の拡張子「**.sid.seq**」を「**.sid**」にリネームしなくてはなりません）。**Vice** のシームレスなフォルダ統合

<sup>42</sup> テキストの大部は **SID-Wizard 1.2** マニュアルからの抜粋。[W]

<sup>43</sup> **C64** のイメージ・ファイルを管理可能な **DOS** シェル。 <http://sta.c64.org/sc.html>

<sup>44</sup> **Windows OS** 上で動作するシェアウェアのファイル・マネージャ。プラグインによって対応ファイルを増やすことができる。 <http://www.ghisler.com/>

環境で SID-Maker を利用する場合は、.S00 ファイルを.sid ファイルへと変換するために SWMconvert を使う必要があります。

```
* SID-MAKER FOR HERMIT SID-WIZARD 1.4 *
-----
PLEASE SELECT OUTPUT FORMAT WITH CURSOR:
  C64/PRG   BIN/SEQ   EXE/PRG   SID/SEQ
(NORMAL) (RAW DATA) (RUNNABLE) (PC-SID)

PLAYER-TYPE VIA CURSOR LEFT/RIGHT: NORMAL
MACHINE-TYPE WITH CURSOR UP/DOWN: PAL

RELOCATION ADDRESS VIA +/- KEYS: $1000
```

図 13 : SID-Maker

1. SID-Maker を起動し、変換したい曲を選択します
  2. 変換先のフォーマットを選択します。見ての通り、SID ファイルを作成するには、SID/SEQ を選択します
  3. プレイヤータイプとマシンタイプ（自動検出されます）を選択します。プレイヤータイプは曲の作成に使用し続けてきたものと一致している必要があります。
  4. 再配置アドレスについてよく分からなければ、デフォルトの値のままにしておいて下さい（Return/Enter キーを押すだけです）
  5. SID モデルを選択します（自動検出されます）。あなたの [使用・所持する] C64 か、あるいは編集曲があなたの耳に入るのと似たように聞こえるように、Vice で選択したバージョンと同じものを選んで（必ずというわけではありません）下さい
  6. ファイル名を記入したら、Return/Enter キーを押します。これであなたの SID ファイル（または別のファイル）は保存されます
- SID フォーマットのエクスポート時、作者情報に「.」（ピリオド）を用いると、SID のヘッダが自動的に分割され、「.」の前は作者名情報に、「.」の後には曲名情報に割り当てられます。
  - SID-Wizard と同様、SID-Maker でもモジュールのバージョンがチェックされます。SWM モジュールのバージョンは、SID-Maker のソフトウェア・バージョンと合致しておく必要があります。

- ▶ おそらくは **1541** のような互換ディスク・ドライブが見つからなかったため、エクスポーターが **SEQ** ファイルを書き込めない時、(CBM-DOS のエラー・コードと一緒に) 「**SAVE ERROR**」の文字が表示されます。とにかく互換モードにしたいなら、再配置アドレスを変更せず(一番始めに位置している) **C64/PRG** フォーマットでエクスポートすることにより、標準的な互換性のある「保存」プロセスを行うことができます。

**PRG** または **SID** フォーマットに再配置アドレスを設定している場合、エラー発生時にはエクスポーターは自動的に互換保存モードに切り替わりますが、エクスポートしようとしている曲の読込アドレスを強制的にデフォルトの**\$1000** (**SID** フォーマットの場合は**\$0F82**) にしてしまいます——ですが、コードそれ自体は再配置されます。こういった時、プログラム内に **PRG** フォーマットを含んでいるなら、希望する読込アドレスを強制し、手動で再配置された曲を読み込む必要があります。(SID フォーマットの場合は **SID** との互換性を得るために **HEX** エディタによって読込アドレスを **2** バイト削除しても良いでしょう)。

## 10.1 SWMCONVERT を使って .S00 ファイルを .SID ファイルへ変換する

以下のインストラクションはマイクロソフト **Windows** での利用を想定して書かれたものです (Linux 用の **SWMconvert** のバイナリも存在します)。まず **SWMconvert.exe** をディレクトリ **Windows/System32** へとコピーして下さい。こうすることによって、コマンド・プロンプトからプログラムを容易に起動できます。

1. コマンド・プロンプトを開き、**.S00** ファイルの存在するフォルダへ移動します。**Windows 7** では、**Windows** エクスプローラのフォルダ名 (あるいはフォルダのどこか) の上で **Shift** キーを押しながら右クリックし、「コマンド・ウィンドウを開く」を選ぶことで、簡単に同じことができます。
2. ウィンドウで次のように記入するだけです。  
**swmconvert** (インプット・ファイル名) **.S00** (アウトプット・ファイル名) **.sid**

(アウトプット・ファイル名) の他のパラメータは任意です。 **.S00** は自動的に **sid** フォーマットに変換されるためです。

**SWMconvert** では、他にも **SID-Wizard** の **SWM** フォーマットと **XM** または **MID** フォーマット間の変換ができます。

私 (**Hermit**) はコマンドラインで **SWM** フォーマットと **XM** または **MID** フォーマットを相互変換可能な **PC** 実行形式 [ファイル] を制作しました<sup>45</sup> (**GUI** のフロントエンドを計画していますが、今のところはインプット/アウトプットするモジュール・ファイル名を手動で入力しなくてはなりません)。使用方法と [コマンドの] 構文は簡単で、コマンドライン・パラメータや「**-help**」パラメータなしに実行すれば、**SWMconvert** の実行形式自らによって説明がなされます (アドバイス: **SWMconvert** はシステム・ディレクトリに置いておくのが良いでしょう)。

異種フォーマット間では、移すことの不可能なエフェクトもある点をお断りしておきます。例えば、**XM** はテンポプログラムや個々別々のパターン長さをサポートしていませんし、**SWM** は 3 チャンネル以上を含むこともできません。**SWM** へ一度に変換できるのは、**XM** または **MID** フォーマットの 1~3 チャンネル/トラックのみです。それゆえ前もって **XM/MID** ファイルを編集しておく必要があります。変換を上手くいかせるためには 3 トラック全てをモノフォニックにしておいて下さい。そうしなければ、ポリフォニック・トラックのほんの少しのノートしか変換されないでしょうから。

他方で、**SWM** フォーマットのコードは、変換時 **XM/MIDI** のトラックより大きく膨れ上がります。**XM** パターンが長すぎる場合、**SWM** フォーマットで最大限許された長さまで切り落とされます。

**MIDI**—**SWM** 変換は **MIDI** ファイルの拍子記号を加工します。**MIDI** ファイルに拍子記号が含まれない場合、デフォルトで **4/4** に設定されます。**MIDI** ファイル内に繰り返し部分がある場合、コンヴァータは出来る限りパターンの再利用を行います。

**SWM**—**MIDI** 変換はトラックのテンポ変更 (オーダーリストに由来するものです) を処理します。3 トラックを有する **SWM** フォーマットは、そのまま 3 トラッ

---

<sup>45</sup> ここより **SID-Wizard 1.2** マニュアルの抜粋。



クの MIDI フォーマット (MIDI フォーマット 1) に変換されます。しかしながら、インストゥルメントそれぞれは別々の MIDI チャンネル (1~16) に広がっています。MIDI シーケンサーを使えばそれらを切り離すこともできるでしょう…… (例: Muse なら可能ですし、SEQ24 なら 3 トラックしかないようにファイル进行处理します……<sup>46</sup>)。

更に、SWMconvert を使って「.swm.prg」から「swm.P00」へ変換することも、その反対への変換も可能です。また VICE フォーマット「.S00」で保存された SID ファイルを、.sid フォーマットへ変換することもできます (その場合に限り、インプット・ファイルが容易しなくてはなりません)。

```
=====
SID-Wizard SWM module v1.0 converter
=====
Converts to/from XM or MIDI formats,
between .prg & .P00, and S00 to SID.
Usage syntax is pretty simple:
SWMconvert <inputfile> [outputfile]
-----
(Formats will depend on extensions.)
If you don't specify an output-file,
name of the input file will be used
as output-filename. Existing files
will be overwritten automatically.
-----
2012 Hermit Soft. (Mihaly Horvath)
=====
```

「application」フォルダには「sng2swm」という名前のもう一つのコンヴァータがあります。GoatTracker の「.sng」ファイルを SID-Wizard の「.swm」ファイルへ高精度に変換できます。構文は SWMconvert と同じです。

## 10.2 SID ファイルのメタデータを編集する

SID ファイルのメタデータ (曲名、作者名、リリース日時、等々) を編集するには、下記リンクよりダウンロード可能な、SIDEedit を利用して下さい。

[http://www.transbyte.org/SID/SIDedit\\_download.html](http://www.transbyte.org/SID/SIDedit_download.html)

---

<sup>46</sup> Muse、SEQ24。MIDI シーケンサー。

### 10.3 (プログラムに埋め込まれている) プレイヤー・ルーチンについてのアプリケーション・ノート

プレイヤーは内部的に使用された 2 つのゼロページ・バイト (デフォルトでは \$FE と \$FF) の保存とリストアを行います。実質的にゼロページがルーチンの外側を侵されることがないので、どのプログラムでも包含が容易です (コンパイルする前に、`setting.cfg` 内のエクスポートの「`PLAYERZP_VAR`」が 0 以外 [`nonzero`] の値に設定されていない限り)。「Normal/Full」プレイヤー・ルーチンの要求する最大限のラスタertimeはおおよそ \$1A~\$1F ラスター行です。わずかにエフェクトや同時に使用されているテーブルコマンドの数に依存しています (「Light」バージョンでは最大 \$14~\$19 ラスター行です)。……音楽で使用されるトラックが 3 トラックより少なければ、プレイヤー・ルーチンのラスタertimeは著しく減少します。初期化ルーチンのコール・アドレスはアキュムレータ内の現在のサブチューン・ナンバーを (いつものように他のルーチンとともに) 要求するベース・アドレス/ロード・アドレス (例: \$1000) と同じです。

シングルスピードの再生アドレスは先頭アドレス・プラス・3 になります (例: \$1003)。X-SID/SID と同じく (GoatTracker とは異なり)、マルチスピードでの再生も同じやり方で処理されます。もちろんラスター行 (といわんやラスタertimeの使用 [消費]) に関してはマルチスピード・ルーチンのコールは、シングルスピードと比較した時、先頭アドレス・プラス・6 で実行されます (例: \$1006)。曲のヴォリュームを外部から変更するには、アキュムレータに希望ヴォリューム (0~F) を置き、先頭アドレス・プラス・9 でヴォリューム設定ルーチンをコールします (例: \$1009)

## 11. HERMIDI インターフェイス (拡張 MIDI イン プット・ハードウェア) <sup>47</sup>

MIDI キーボードや MIDI コントローラーで SID-Wizard を制御したい方のために、私は解決策を用意しています (あいにく未完成ではありますが、少なくとも進行中です)。

---

<sup>47</sup> 第 11 章の記述は全て Hermit。

てっとり早くチープな複製可能な解決を望んでいる人たちのために、容易に実装可能な C64 用の MIDI インターフェイスを、私は開発しているということです

(MSSIAH<sup>48</sup>を私は所有していますが、それはオープンソースではありませんし、私の知る限り、それは他のプログラムでは使用不可能です……)。ユーザーポート・コネクタのような入手困難なパーツについて心配する必要も、拡張ポートの使用に気を揉む必要さえありません。HerMIDI は C64 の背後のシリアル・ポート (あるいはデジチェーンの 1541 ドライブ) に接続するもので、通常のドライブと同様のふるまいをします。

C64 (あるいは他の IEC/シリアル互換製品) が TALK コマンドでデバイス 15 をコールする際、HerMIDI は自身のインプットからやって来る MIDI データを C64 に転送する時間を察知し、C64 のリクエストの用意が整ったサインとして LED が光ります。

C64 は DATA ライン<sup>49</sup>をプル・ダウンすることにより、いつでもデータのリクエストが可能です。HerMIDI はシリアル・ポートを通して 1 バイトを置きます。この時送信されるのは 1 バイトだけです。これにはノート/ノート・オフや、あるいはインストゥルメント変更のような他の異なったメッセージを含んでいるでしょう (追々、HerMIDI のビルディング/プログラミング・マニュアルにおいて、これらのメッセージの詳細な説明を行うつもりです)。

転送に ATN ライン<sup>50</sup>を使用するファストローダー・ルーチンを所有しているのなら、データ干渉防止のため、HerMIDI のプラグを抜く [*unplugged*] か、スイッチをオフにする必要があります (MIDI ケーブルを通して 5V が流れ込むため)<sup>51</sup>。

基板に組み込まれている PIC12F509 と回路図面は「HerMIDI」フォルダ下の SID-Wizard ソース・ディレクトリに含まれています。

---

<sup>48</sup> MSSIAH。8 bit ventures の開発した C64 用の MIDI カートリッジ。専用のシーケンサーが付属する。  
<http://www.8bitventures.com/mssiah/>

<sup>49</sup> DATA-line。信号線的一种。

<sup>50</sup> ATN-line。信号線的一种。

<sup>51</sup> 具体的には HerMIDI に接続されているデバイス (MIDI キーボードやコントローラ) のスイッチを切るか、C64 に接続されている HerMIDI のプラグを直接引き抜くことを指す。実際には、ATN ラインを使用するファストローダーを使用する場合であっても、デモやゲームに、全く同一の長い文字列が現れるようなことはめったに生じないので、こうした事態が訪れる心配はない。

先立って HerMIDI のプロトタイプが実装、組立時にどんな風になるのかお見せしましょう。



HerMIDI は未完成であり、SID-Wizard 1.5 での正式公開を予定しています（つまり、HerMIDI を必要としないユーザーのためにそれを取り除いたのがバージョン 1.4 の位置づけになります）。ここまでの記述には著しく変更が加えられることがあります。

## 12. 使用キー一覧表

次ページ以降参照……。

# キーボード・レイアウト一覧

## コモドール 64 US キーボード



## US インターナショナル・キーボード



## コモドール 64 US キーボードのキー配列がマップされた、Vice エミュレータ・キーボード



## 12.1 一般使用キー一覧

C64 のキー	VICE エミュレータのキー (US インターナショナル)	カテゴリー	キーの働き
F5	F5	移動	パターン・エディタ・ウィンドウへ移動
Shift + F5	F6	移動	オーダーリストへ移動
F7	F7	移動	インストゥルメント・エディタ・パネルへ移動
C= + F5	CTRL + F5 または F6	移動	コード・テーブルへ移動
C= + F7	CTRL + F7	移動	テンポプログラム・テーブルへ移動
Shift+F7	F8	移動	メイン・メニューへ移動
CURSOR-down/right および Shift	矢印カーソル ↑/↓/←/→	移動	行ける範囲まで上/下/左/右に 1 ステップずつカーソルを移動
CTRL または Shift + CTRL	TAB または Shift + TAB	移動	トラックまたはインストゥルメント・テーブルを順に切り替える
/ または Shift+/, ↑* または RESTORE	/ または Shift + /, Page-Down*/Page-Up	移動	(パターン/オーダーリスト等々の) ウィンドウ内で上/下/左/右に 4 または 8 ステップずつカーソルを移動
HOME	Home	移動	スクリーンの開始ポジション、テーブル/パターン内の絶対開始ポジションにカーソルを移動
Return	Enter	移動	行の始めに戻る。あるいはカーソルの下のパターンまたはインストゥルメントに移動
F1 または C= + F1	F1 または CTRL + F1	再生	始めから曲を再生/始めから曲を追いかけ再生
Shift + F1 または C= + Shift + F1	F2 または CTRL + F2	再生	マーカ・ポジション [position(s)] から曲を再生/マーカ・ポジションから曲を追いかけ再生
F3 または C= + F3	F3 または CTRL + F3	再生	選択されたパターンを再生/選択されたパターンを追いかけ再生
Shift + F3 または STOP	F4 または ESC	再生	曲/パターン再生時の停止/再生 [停止解除] の切替
Shift + SPACE	Shift + SPACE	再生	カーソル・ポジションから選択されたパターンを再生
←	‘ (TAB の上)	再生	早送り (通常の 4x スピード)
Shift + ← C= + ←	Shift + ‘ (TAB の上) CTRL + ‘	再生/設定	追いかけ再生モード/通常再生の切替。デフォルトの追いかけ再生のオン/オフを設定する
Shift+ 1 / 2 / 3	Shift + 1 / 2 / 3	再生/設定	チャンネル 1 ~ 3 のミュート/ソロの切替
Shift + A/Z	Shift + A/Z	設定	ノート入力後の自動前進する (ステップを踏む) 間隔を増やす/減らす
Shift + I	Shift + I	設定	ノート [編集時] のインストゥルメント自動入力切替
Shift + D	Shift + D	設定	ジャミング・モード時のモノフォニック [単声] + レガートとポリフォニック [多声] + 繋ぎ合わせ [dovetailing] の切替
Shift + F/G	Shift + F/G	設定	フレームスピードを下げる/上げる (1x シングルスピード ~ 8x マルチスピード)
Shift + H/J	Shift + H/J	設定	パターン・エディタ・ウィンドウ内で強調したいステップの密度 [比重] を下げる/上げる <sup>52</sup>
C= + B	CTRL + B	設定	パターン/トラックの運動のオン/オフの切替。パターンをまとめて/個別にスクロールさせることが可能
C= + T	CTRL + T	設定	サブチューンの全体通常テンポ/ファンクテンポ・モードの切替
Shift + colon/dot , Shift + : または ;	Shift + colon/dot , Shift + ; または '	設定	サブチューンの全体通常テンポ/2 番目のファンクテンポを上げる/下げる
C= + colon / dot	CTRL + colon / dot	選択	サブチューンの変更 (オーダーリスト)。パターンはサブチューン間で共有される
C= + 1~8 または +/-	CTRL + 1~8 または +/-	選択	編集時のまたはジャミング時の音階のオクターヴの選択 (1 ~ 8)。 (DMC モードでは C= キーを押す必要なし)
Shift + PLUS または Shift + MINUS	Shift + PLUS または Shift + MINUS	選択	ジャミング/編集に用いるインストゥルメントの変更
Shift + T/Y	Shift + T/Y	選択	テンポプログラムの選択 (テンポプログラム番号の増減)
Shift + K/L	Shift + K/L	選択	編集するコードの指定 (コード番号の増減)
SPACE	SPACE	編集	キーボード・ジャミング・モードと編集モードの切替
Shift + N	Shift + N	編集	選択中のインストゥルメントのリネーム。最大 8 文字、Esc/Stop キーで編集を中断する
DEL/Pound または Shift + DEL	Delete/Backspace または Insert	編集	カーソル・ポジションにある値の削除/新規挿入またはエンド・ポジションにおけるパターン/テーブルのサイズ増減

\*Linux では VICE のビルドは、Page-Down キーが ↑ [上矢印] キーに対応させられているようです。デフォルトでは「Del」キーのほうです (それで問題ありません。というのも、これによって Linux では、Page-Up キーの下にある、Page-Down (上矢印) キーが正しい場所にあることになるからです。これは C64 のキー配置のリストアを目的とした関連付けなのです)。私見では Linux では「Del」キーを Backspace キーとして用いるのが良いように思います。

\*注記: 追いかけ再生モードでは、曲の再生中の不意の編集を防止するために、パターン・エディタで使用するキーは無効化されます

<sup>52</sup> 拍子の変化等による曲中のアップビートとダウンビートの割合の変化を目立たせ、分かりやすくするための機能。

## 12.2 パターン・エディタ 使用キー一覧

C64 のキー	VICE エミュレータのキー (US インターナショナル)	カテゴリー	キーの働き
CTRL または Shift + CTRL	TAB または Shift + TAB	移動	前／後ろヘトラックを順に切り替える
Return または Shift + Return	Enter または Shift + Enter	移動	次行に移動し、ノートのオン／オフを置く。またはカーソル下のインストゥルメント／コードの変更
Z X C V B N M , . (DMC モードでは A~L)	Z X C V B N M , . (DMC モードでは A~L)	編集／ジャム	白鍵：低オクターヴ（全音階・Cメジャー [ハ長調]）。ジャミング時はレガートが有効
Q,W,E,R,T,Y,U,I,O,P (DMC モードでは不使用)	Q,W,E,R,T,Y,U,I,O,P (DMC モードでは不使用)	編集／ジャム	白鍵：高オクターヴ（全音階・Cメジャー [ハ長調]）。
S,D, G,H,J, L , 2,3, 5,6,7, 9,0 (DMC モードでは不使用)	S,D, G,H,J, L , 2,3, 5,6,7, 9,0 (DMC モードでは不使用)	編集／ジャム	黒鍵：低・高オクターヴ（半音）。DMC モードでは QWERTY 行が鍵盤に割り当てられる！
C= + 1~8 (DMC モードでは 1~8)	CTRL + 1~8 (DMC モードでは 1~8)	編集／ジャム	編集またはジャミング時音階が生じている場のオクターヴの設定 (1~8)
A, 1, C=+Del (DMC モードでは Q)	A, 1, CTRL + Del (DMC モードでは Q)	編集	空ノート（パターン・ポジション内のノートカラムのノートを、残りのノートを移動させず [それだけを] 削除する）
1~9, A~F (16 進数の入力)	1~9, A~F (16 進数の入力)	編集	インストゥルメント／エフェクトカラムで値を記入する。ノートカラムでノートまたはヴィブラートの振幅を設定する
DEL/Pound または Shift + DEL	Delete/Backspace または Insert	編集	カーソル・ポジションで削除／新規挿入またはパターン終端部においてパターン・サイズの増減
C= + DEL	CTRL + Delete/Backspace	編集	カーソル・ポジションを基準とした、編集中のパターン行のノートまたはインストゥルメント＋エフェクトカラムの削除
C= + Shift + DEL	CTRL + Shift + Del./Backspace	編集	編集中のトラックの全てのパターン行（ノート、インストゥルメント、エフェクト）の削除
Shift + Q/W	Shift + Q/W	編集	カーソル・ポジションの後に来る編集パターンの移調。半音ずつ上げる／下げる
C= + Q/W	CTRL + Q/W	編集	カーソル・ポジションの後に来る編集パターンの移調。オクターヴずつ上げる／下げる
Shift + R または C= + R	Shift + R または CTRL + R	編集	リングモジュレーション・エフェクトのオン／オフを現在のカーソル・ポジションのカラムのノートへ置く
Shift + P	Shift + P	編集	自動ポルタメント・エフェクトを現在のカーソル・ポジションのカラムのノートへ置く
Shift + S または C= + S	Shift + S または CTRL + S	編集	シンクビット・エフェクトのオン／オフを現在のカーソル・ポジションのカラムのノートへ置く
Shift + V	Shift + V	編集	ヴィブラート・エフェクトを現在のカーソル・ポジションのカラムのノートへ置く。振幅を編集可能。
C= + X	CTRL + X	編集	終端までのパターン内容を切り取り／削除し、パターンバッファ／クリップボードへコピーする
C= + C	CTRL + C	編集	パターンバッファ／クリップボードへ、カーソル・ポジションからパターン終端までのパターン内容をコピーする
Shift + C	Shift + C	編集	カーソル・ポジションに合わせて、バッファへコピーされる（または切り取られる）データの範囲を限定する。バッファ終端の定義。
C= + V	CTRL + V	編集	カーソル・ポジションからパターン終端まで、パターンバッファ内容を貼り付ける

## 12.3 オーダーリスト 使用キー一覧

C64 のキー	VICE エミュレータのキー (US インターナショナル)	カテゴリー	キーの働き
Return または Shift + Return	Enter または Shift + Enter	移動	オーダーリストにおいてカーソル・ポジションの下にある (複数の) パターンを選択する。パターン・エディタにおいて編集集中のトラックへジャンプする
C= + Return	CTRL + Enter	移動	現在再生されているパターンの選択。パターン・エディタにおいて編集集中のトラックへジャンプする
Shift + SPACE	Shift + SPACE	再生	全てのトラックのカーソル・ポジションへ、オーダーリストの再生開始マーカーを設定 (F2 キーで再生)。ポジション番号は反転する
C= + SPACE	CTRL + SPACE	再生	全てのトラックの個々の再生ポジションへ、オーダーリストの再生開始マーカーを設定。1トラック目のポジション番号は反転する
1~9, A~F (16 進数の入力)	1~9, A~F (16 進数の入力)	編集	<b>16 進値</b> でパターン番号およびエフェクト/ジャンプ番号を入力する
DEL/Pound または Shift + DEL	Delete/Backspace または Insert	編集	カーソル・ポジションにおいて値の削除/新規挿入を行う。カーソルがシーケンスの終端/ループ・ポジションにある場合は、同時にオーダーリストのシーケンス・サイズの増減を行う
C= + C	CTRL + C	編集	カーソル・ポジションからバッファへオーダーリストのシーケンスをコピーする
Shift + C	Shift + C	編集	バッファ内にコピーされたデータの終端を設定。(バッファ制限)
C= + V	CTRL + V	編集	カーソル・ポジションへバッファを貼り付ける。既存のデータに追加する
C= + E	CTRL + E	編集	<b>最初の空</b> (未使用) パターン番号の入力。キーを押す度にパターン番号は増加し、エンド・シグナル [ \$FF ] で作用する



## 12.4 インストゥルメント・エディタ 使用キー一覧

C64 のキー	VICE エミュレータのキー (US インターナショナル)	カテゴリー	キーの働き
CTRL または Shift + CTRL	TAB または Shift + TAB	移動	インストゥルメント設定パネルとインストゥルメント・テーブルの切替
Return または Shift + Return	Enter または Shift + Enter	移動	テーブルの次行始めに戻る (または主なインストゥルメント・パラメータの切替 [toggle/cycle])
+ / -	+ / -	選択	インストゥルメントの変更。パターン/オーダーリスト/インストゥルメント・ウィンドウと同様、コード/テンポ・テーブルでは Shift キー同時押しが必要
Return	Enter	選択	ハードリスタートのタイプ、ヴィブラートのタイプ、オクターヴ・シフトの信号等のインストゥルメントの主なパラメータの切替。またはカーソルの下の番号 [コードネーム] をデフォルト・コードに指定する。
1~9, A~F (16 進数の入力) ならびに F~Z	1~9, A~F (16 進数の入力) ならびに F~Z	編集	インストゥルメントのデータ領域へ <b>16 進値</b> の入力。またはインストゥルメント名の入力 (Shift キーを必要としないシンボル、. [ピリオド] や/ [スラッシュ] 等、一部の記号も使用可能)
=	=	編集	値の符号反転 (例: \$40 は \$C0、つまり -\$40 に変換される)
Shift + N	Shift + N	編集	選択中のインストゥルメントのリネーム。最大 8 文字。中断するには Esc/Stop キーを押す
DEL/Pound または Shift + DEL	Delete/Backspace または Insert	編集	カーソル・ポジションの値の削除/新規挿入。テーブルサイズの縮小/拡大
Shift + Space	Shift + Space	編集	カーソル下にある編集中のテーブル行のゲートオフ・インデックスの設定または消去
C= + C	CTRL + C	編集	選択された全てのインストゥルメントのコピー
C= + V	CTRL + V	編集	インストゥルメント (クリップボードにコピーされた変更されたインストゥルメントに関する内容) の貼付。全てのインストゥルメントは上書きされる
C= + P または C= + F	CTRL + P / F	編集	ノート開始地点のパルス幅/フィルター・テーブルの再設定を交互に切替。(再設定するインストゥルメントの選択)

## 12.5 コード・テーブル/テンポ・テーブル 使用キー一覧

C64 のキー	VICE エミュレータのキー (US インターナショナル)	カテゴリー	キーの働き
CTRL または Shift + CTRL	TAB または Shift + TAB	移動	パターン・エディタまたはインストゥルメント・エディタに移動
Return または Shift + Return	Enter または Shift + Enter	移動	テーブルの次行始めに戻る (または主なインストゥルメント・パラメータの切替)
+ / -	+ / -	選択	コード/テンポプログラムの選択。他のウィンドウでは Shift + K/L か Shift+T/Y の同時押しが必要
Shift + PLUS または Shift + MINUS	Shift + PLUS または Shift + MINUS	選択	ジャミング/編集に用いるインストゥルメントの変更
1~9, A~F (16 進数の入力)	1~9, A~F (16 進数の入力)	編集	<b>16 進値</b> やコードのループ/繰り返し回数をコード・テーブル/テンポプログラム・テーブルへ入力
=	=	編集	値の符号反転 (例: \$04 は -4 -2s から成り立つ値 \$FB に変換される)
DEL/Pound または Shift + DEL	Delete/Backspace または Insert	編集	カーソル・ポジションの値の削除/新規挿入。テーブルサイズの縮小/拡大

## 12.6 10 進－16 進変換表<sup>53</sup>

次ページ参照……。

---

<sup>53</sup> UNSIG.=Unsigned Value (符号なし値)、SIG.=Signed Value (符号付き値)。

# DECIMAL TO HEXADECIMAL CONVERSION TABLE

Mikael Norrgård 2012  
 (<http://www.witchmastercreations.com>)

UNSIG.	SIG.	HEX
0	0	00
1	1	01
2	2	02
3	3	03
4	4	04
5	5	05
6	6	06
7	7	07
8	8	08
9	9	09
10	10	0A
11	11	0B
12	12	0C
13	13	0D
14	14	0E
15	15	0F
16	16	10
17	17	11
18	18	12
19	19	13
20	20	14
21	21	15
22	22	16
23	23	17
24	24	18
25	25	19
26	26	1A
27	27	1B
28	28	1C
29	29	1D
30	30	1E
31	31	1F
32	32	20
33	33	21
34	34	22
35	35	23
36	36	24
37	37	25
38	38	26
39	39	27
40	40	28
41	41	29
42	42	2A
43	43	2B
44	44	2C
45	45	2D
46	46	2E
47	47	2F
48	48	30
49	49	31
50	50	32
51	51	33
52	52	34
53	53	35
54	54	36
55	55	37
56	56	38
57	57	39
58	58	3A
59	59	3B
60	60	3C
61	61	3D
62	62	3E
63	63	3F

UNSIG.	SIG.	HEX
64	64	40
65	65	41
66	66	42
67	67	43
68	68	44
69	69	45
70	70	46
71	71	47
72	72	48
73	73	49
74	74	4A
75	75	4B
76	76	4C
77	77	4D
78	78	4E
79	79	4F
80	80	50
81	81	51
82	82	52
83	83	53
84	84	54
85	85	55
86	86	56
87	87	57
88	88	58
89	89	59
90	90	5A
91	91	5B
92	92	5C
93	93	5D
94	94	5E
95	95	5F
96	96	60
97	97	61
98	98	62
99	99	63
100	100	64
101	101	65
102	102	66
103	103	67
104	104	68
105	105	69
106	106	6A
107	107	6B
108	108	6C
109	109	6D
110	110	6E
111	111	6F
112	112	70
113	113	71
114	114	72
115	115	73
116	116	74
117	117	75
118	118	76
119	119	77
120	120	78
121	121	79
122	122	7A
123	123	7B
124	124	7C
125	125	7D
126	126	7E
127	127	7F

UNSIG.	SIG.	HEX
128	-128	80
129	-127	81
130	-126	82
131	-125	83
132	-124	84
133	-123	85
134	-122	86
135	-121	87
136	-120	88
137	-119	89
138	-118	8A
139	-117	8B
140	-116	8C
141	-115	8D
142	-114	8E
143	-113	8F
144	-112	90
145	-111	91
146	-110	92
147	-109	93
148	-108	94
149	-107	95
150	-106	96
151	-105	97
152	-104	98
153	-103	99
154	-102	9A
155	-101	9B
156	-100	9C
157	-99	9D
158	-98	9E
159	-97	9F
160	-96	A0
161	-95	A1
162	-94	A2
163	-93	A3
164	-92	A4
165	-91	A5
166	-90	A6
167	-89	A7
168	-88	A8
169	-87	A9
170	-86	AA
171	-85	AB
172	-84	AC
173	-83	AD
174	-82	AE
175	-81	AF
176	-80	B0
177	-79	B1
178	-78	B2
179	-77	B3
180	-76	B4
181	-75	B5
182	-74	B6
183	-73	B7
184	-72	B8
185	-71	B9
186	-70	BA
187	-69	BB
188	-68	BC
189	-67	BD
190	-66	BE
191	-65	BF

UNSIG.	SIG.	HEX
192	-64	C0
193	-63	C1
194	-62	C2
195	-61	C3
196	-60	C4
197	-59	C5
198	-58	C6
199	-57	C7
200	-56	C8
201	-55	C9
202	-54	CA
203	-53	CB
204	-52	CC
205	-51	CD
206	-50	CE
207	-49	CF
208	-48	D0
209	-47	D1
210	-46	D2
211	-45	D3
212	-44	D4
213	-43	D5
214	-42	D6
215	-41	D7
216	-40	D8
217	-39	D9
218	-38	DA
219	-37	DB
220	-36	DC
221	-35	DD
222	-34	DE
223	-33	DF
224	-32	E0
225	-31	E1
226	-30	E2
227	-29	E3
228	-28	E4
229	-27	E5
230	-26	E6
231	-25	E7
232	-24	E8
233	-23	E9
234	-22	EA
235	-21	EB
236	-20	EC
237	-19	ED
238	-18	EE
239	-17	EF
240	-16	F0
241	-15	F1
242	-14	F2
243	-13	F3
244	-12	F4
245	-11	F5
246	-10	F6
247	-9	F7
248	-8	F8
249	-7	F9
250	-6	FA
251	-5	FB
252	-4	FC
253	-3	FD
254	-2	FE
255	-1	FF

## 12.7 ARP カラム音階対応表

Value	Note	Value	Note	Value	Note	Value	Note
81	C-1	99	C-3	B1	C-5	C9	C-7
82	C#	9A	C#	B2	C#	CA	C#
83	D	9B	D	B3	D	CB	D
84	Eb	9C	Eb	B4	Eb	CC	Eb
85	E	9D	E	B5	E	CD	E
86	F	9E	F	B6	F	CE	F
87	F#	9F	F#	B7	F#	CF	F#
88	G	A0	G	B8	G	D0	G
89	G#	A1	G#	B9	G#	D1	G#
8A	A	A2	A	BA	A	D2	A
8B	Bb	A3	Bb	BB	Bb	D3	Bb
8C	B	A4	B	BC	B	D4	B
8D	C-2	A5	C-4	BD	C-6	D5	C-8
8E	C#	A6	C#	BE	C#	D6	C#
8F	D	A7	D	BF	D	D7	D
90	Eb	A8	Eb	C0	Eb	D8	Eb
91	E	A9	E	C1	E	D9	E
92	F	AA	F	C2	F	DA	F
93	F#	AB	F#	C3	F#	DB	F#
94	G	AC	G	C4	G	DC	G
95	G#	AD	G#	C5	G#	DD	G#
96	A	AE	A	C6	A	DE	A
97	Bb	AF	Bb	C7	Bb	DF	Bb
98	B	B0	B	C8	B		

## 12.8 ATTACK / DECAY / RELEASE タイミング

HEX	Attack Rate	Decay / Release Rate
0	2 ms	6 ms
1	8 ms	24 ms
2	16 ms	48 ms
3	24 ms	72 ms
4	38 ms	114 ms
5	56 ms	168 ms
6	68 ms	204 ms
7	80 ms	240 ms
8	100 ms	300 ms
9	250 ms	750 ms
A	500 ms	1.5 s
B	800 ms	2.4 s
C	1 s	3 s
D	3 s	9 s
E	5 s	15 s
F	8 s	24 s