NRTDRV 仕様

メモリマップ

アドレス	内容
0000-17FFH	IPL 起動用プログラム (IPLPATCH.BIN) / <u>NRDSEL</u> (NRDSEL2D.BIN) / <u>OPM 音色エディタ</u> (IPLOPMED.BIN)
1800-3DFFH	ドライバ本体 (NRTDRV.BIN)
3E00-3EFFH	OPM#1 仮想レジスタ (<u>NRDSEL</u> 、 <u>OPM 音色エディタ</u> で使用)
3F00-3FFFH	OPM#2 仮想レジスタ (<u>NRDSEL</u> 、 <u>OPM 音色エディタ</u> で使用)
4000-FEFFFH	曲データ (*.NRD)
FF00-FFFFH	IPL 起動用ワークエリア / スタック領域

NRTDRV 内ではスタックポインタを設定していないため、呼び出すプログラム側で SP レジスタに適切な値を設定しておく必要がある (X1 の場合は 0000H)。

エントリ

01800H (DRVINI)

ドライバ初期化。最初に実行する必要がある。

01803H (MPLAY)

再生。EI 状態になり、任意のアドレスから演奏する。

01806H (MSTOP)

停止。EI 状態になる。

01809H (MFADE)

フェードアウト。

0180FH (MTEST)

テスト音を発音。

01812H (MPLAYD)

デフォルトアドレス再生。EI 状態になり、04000H から演奏する。

01815H (VINIT)

簡易演奏モニタ用の画面初期化。PCG の定義などを行う。

01818H (VMAIN)

簡易演奏モニタを1フレーム描画する。

0181BH (INVREG)

NRDSEL/OPM 音色エディタ用の仮想 OPM レジスタを無効にします。

外部参照用ワークエリア

01831H (VER)

内部バージョン番号

- ・ 0 : 2016/10/28 版まで
- ・ 1 : 2016/11/01 版から
- ・ 2 : 2017/03/03 版から

この内部バージョン番号は、ユーザーアプリケーション側でドライバと曲データの不整合を防ぐために使う。

実際にバージョン番号が実装されているのは2016/12/03版からとなる。それ以前のバージョンで上記のワークエリアを読むと「0」が返る。

2016/12/03版以前では演奏開始した時点で値が変化してしまう(トラック1のワークエリアと重複している)ため、演奏開始前に値を取得しておく必要がある。

データ

4000H から配置。リロケータブルなので別の場所にも置けるが、その場合はドライバ側の BGMADR ワークを書き換える(曲データを再アセンブルする必要はなく、再生エントリをコールする前にワークを書き換えるだけで OK)。

曲データ内で使用されるアドレスは、すべて曲データ開始アドレスからみたオフセット値。たとえば、曲データが 04A00H ~ で、トラック A の実際のアドレスが 06809H ~ なら、曲データ内に記録されているトラック A の開始アドレスは 1E09H となる。

- ・ 1 バイト目
 - ・ CTC0 の初期分周値 (#TEMPO ヘッダで設定)
- ・ 2 バイト目
 - ・ CTC3 の初期分周値 (#TEMPO ヘッダで設定)

早送りの時は上記2バイトに早送り時の分周値が入る。

iコマンドの実体は tコマンド。tコマンドで分周値を再設定することによって、本来のテンポに戻している。

上記2バイトは必ず確保されているので #TEMPO ヘッダは曲データのサイズに影響しない。#TEMPO ヘッダを全く使わなくても上記には BPM120 相当の分周値が入る。

逆に、tコマンドや iコマンドは使うたびに3バイト消費する。

よって、テンポの変わらない曲では #TEMPO ヘッダだけを使うのがサイズの的に望ましい。

- ・ 3 バイト目

- ・ビット 0：OPM にウェイトを掛けるかどうかのフラグ
 - ・ビット 1：キーオンフラグ（このフラグを立てるとキーオンしない。テンポ再設定でキーオン復活）
 - ・ビット 2：q コマンド未使用フラグ（このフラグを立てると q コマンド関連の演算を一切行わなくなる）
 - ・ビット 3～6：予約
 - ・ビット 7：内部バージョン番号が 42 バイト目に存在するかどうか
- ・ 4 バイト目～41 バイト目
 - ・ OPM#1 の 8 チャンネル分の各トラック先頭アドレス (16 バイト)
 - ・ OPM#2 の 8 チャンネル分の各トラック先頭アドレス (16 バイト)
 - ・ PSG の 3 チャンネル分の各トラック先頭アドレス (6 バイト)
 - ・ 42 バイト目
 - ・ 存在しない=2016/10/28 版まで
 - ・ 1=2016/11/01 版から
 - ・ 2=2017/03/03 版から

3 バイト目の最上位ビットが立っている場合は内部バージョン番号がここに入る。

- ・ 42/43 バイト目以降
 - ・ 曲名 /ANK (#TITLE ANK ヘッダ)
 - ・ 曲名 /Shift_JIS (#TITLE ヘッダ)
 - ・ 作曲者名 (#COMPOSER ヘッダ)
 - ・ プログラマ名 (#PROGRAMMER ヘッダ)
 - ・ メモ (#MEMO ヘッダ)

3 バイト目の最上位ビットが立っている場合は 43 バイト目から、立っていない 42 バイト目から、上記 5 項目の曲情報が入る。可変長で、各項目の終端は 0。

- ・ それ以降

シーケンスデータ

シーケンスデータ仕様

コマンド番号はすべて 10 進数で記載する。[????.w] の場合は 2 バイト値。

[00]+[count] (+[count]+[count]...)

休符。[count] はカウント数。もしカウント数が 255 なら、その次の 1 バイトもカウント数として扱う。

[01]+[times]+[register]+[data] (+[register]+[data]...)

z コマンド。一気に複数のレジスタへと書き込める。[times] は書き込むレジスタの個数。

[02]+[PAN/ALG/FB]+[voice_address+2.w]

軽量音色設定。PAN/ALG/FB をドライバ側で計算せずに決め打ちする。さらに、オペレータマスクは指定できない(前回の設定値を引き継ぐ)。[voice_address+2.w] は音色データが置かれているメモリの相対アドレス +2 バイト。

[03]+[volume]

ボリューム値をワークエリアに書き込む。レジスタへの書き込みは行わない。音色切替と音量 コマンドが連続したときに TL レジスタを 2 回叩くのを防ぐために使用。[volume] はボリューム値。

[04]+[OP1 の DT2/MUL]+[OP1 の TL]...[OP4 の DT2/MUL]+[OP4 の TL]

全オペレータの MUL、DT2、TL を一括で書き換える。CSM 音声合成に使用可能。

[05]+[type/sync]

ソフトウェア PM- タイプ再設定 (シンクフラグ設定用)

[06]+[OP/sync]

OPM トラックではソフトウェア AM-OP 再設定(シンクフラグ設定用) / PSG トラックでは予約。

[07] ~ [11]

予約。

[12]+[port]+[register]+[data]

ポート指定時の y コマンド。[port] は OPM のポートアドレスの下位バイト (0=PSG、1=0701H、9=0709H)。[register] はレジスタ番号、[data] は書き込む値。

[13]+[release_rate]+[release_level]

PSG リリース設定。PSG パートでキーオフした際のリリースレートとリリースレベルを設定する。

[14]+[voice_address.w]

音色設定。[voice_address.w] は音色データが置かれているメモリの相対アドレス。

[15]+[detune]

デチューン。[detune] はデチューン値。2 の補数でマイナスを表現するので、値のもつ範囲は -128 ~ 127 となる。

[16]+[macro.address.w]

マクロ。[macro.address.w] はマクロデータの相対アドレス。

[17]

レガート ON。

[18]

レガート OFF。

[19]+[volume]

ボリューム。[volume] はボリュームの値。PSGトラックでも 0 ~ 127 となっているため、ドライバ側で 3 ビット右シフトし、0 ~ 15 に丸め込まれる。

[20]+[number]

OPMトラックではパン。[number] はパンの値。PSGトラックではトーン・ノイズ指定。[number] のビット 0 を立てるとトーン有効、ビット 1 を立てるとノイズ有効 (0= ミュート 1= トーン 2= ノイズ 3= 両方)。

[21]+[times]

リピート開始。[times] はリピート回数。

[22]

リピート終了。

[23]

リピート脱出。

[24]+[register]+[data]

y コマンド。[register] はレジスタ番号、[data] は書き込む値。

[25]+[number]

q コマンド。

[26]+[number]

Q コマンド。

[27]+[number]

OPMトラックではハードウェア LFO リセット兼シンク。[number] は機能選択 (0=リセット 1=シンク解除 2=シンク)。PSGトラックではノイズ周波数。PSGレジスタ #6 の値。

[28]+[key_shift]

移調。[key_shift] はキーシフト値。2 の補数。

[29]+[pitch]/[pitch.d]+[pitch]

ポルタメント。[pitch] の値が割り込み 1 周期ごとにレジスタに加算される。

OPMトラックでは [pitch] は 0 ~ 255 で、この値がポルタメント ON/OFF のフラグも兼ねる。

PSGトラックでは [pitch.d]+[pitch] の 2 バイトとなり、[pitch.d] は小数点以下 2 桁を 100 倍したものとなる。また、[pitch] は 0 ~ 127 で、最上位ビットはポルタメント ON/OFF のフラグとして扱う (つまり、この 2 バイトで 0.01 ~ 127.99 の精度となる)。

[30]+[CTC0]+[CTC3]

テンポ。[CTC0] と [CTC3] は各 CTC チャンネルの分周値。

[31]+[delay]+[pitch]+[steps/flag]+[type/sync]

ソフト PM。[steps/flag] が動作フラグも兼ねている (0 以外で動作)。[type/sync] のビット 7 はシンクフラグ。

[32]+(OPM=[delay]+[depth]+[steps/flag]+[OP/sync] / PSG=[shape])

OPMトラックではソフト AM。[steps/flag] が動作フラグも兼ねている (0 以外で動作)。[OP/sync] のビット 7 はシンクフラグ。PSGトラックでは s コマンド (ハードウェアエンベロープ設定)。

[33]

強制キーオフ。

[34]+[number]

トラック再開。[number] はトラック番号 (トラック A ~ P が 1 ~ 16、トラック 1 ~ 3 が 17 ~ 19 になる)。

[35]+[incliment]+[start_pitch.w]

グライド。[incliment] は増分兼フラグ (0 でオフ)。[start_pitch] はグライド開始ピッチ。2 の補数。

[36]+(OPM=[volume]+[OP] / PSG=[period.w])

OPM トラックでは OP 指定型ボリューム。[volume] はボリュームの値。[OP] はオペレータ指定。PSG トラックでは m コマンド (ハードウェアエンベロープ周期)。

[37]+[fade_speed]

フェードアウト。[fade_speed] はフェードアウトの速度。0 ~ 15 で、0 が最速。

[38] ~ [124]

予約。内部処理は [126] のトラック終了と同等。

[125]

トラック一時停止。トラック再開コマンドで再開できる。

[126]

トラック終了。トラック再開コマンドで再開できない。

[127] (+[loop_address.w])

トラック / マクロ終了。トラック終了の場合は [loop_address.w] がループ地点の相対アドレス。

[128 ~ 247]+[count] (+[count]+[count]...)

音符。[128]=o0c、[247]=o9b。[count] はカウント数。もしカウント数が 255 なら、その次の 1 バイトもカウント数として扱う。

割り込み仕様

2本の割り込みを駆動し、テンポとエフェクトに利用している。

CTC0+CTC3

この割り込みに演奏ルーチンを接続。割り込み周期自体を可変させてテンポを変更する。

$$\text{周期} = 937500 / (\text{BPM} * (\text{TimeBase} / 4))$$

X1のクロックは4MHzなので、1クロックあたりの時間は250ns。

しかし、プリスケアラを256倍にしているため、割り込み周期の最小単位は64ms(250ns*256)となる。

[例] BPM=120、TimeBase=192の場合

$$937500 / (120 * (192 / 4)) * 250 \text{ns} * 256 = 10.4166666\dots$$

つまり、BPM120の割り込み1周期あたりの時間は約10.42ミリ秒。

CTC1

この割り込みにエフェクトルーチンを接続。常に1周期16.384msで処理されるため、各種エフェクト(ポルタメント、グライド、ソフトウェアLFO)に関してはテンポの影響を受けない。タイムモードでタイムコンスタント値は0(256)、プリスケアラは1(256倍)。

$$\text{周期} = 16.384 \text{ms} = 16384000 \text{ns} = 256(\text{タイムコンスタント}) * 256(\text{プリスケアラ}) * 250 \text{ns}(4 \text{MHz})$$

音色データ仕様

OPM 音色データ

下記の30バイトからなる。

```
PAN/FB/ALG,      OM,
DT1/ML, DT1/ML, DT1/ML, DT1/ML,
VOL,      VOL,      VOL,      VOL,
KS /AR, KS /AR, KS /AR, KS /AR,
AM /DR, AM /DR, AM /DR, AM /DR,
DT2/SR, DT2/SR, DT2/SR, DT2/SR,
SL /RR, SL /RR, SL /RR, SL /RR,
TL,      TL,      TL,      TL
; (OP1)  (OP3)  (OP2)  (OP4)
```

VOLはコンパイラ側で音量コマンドの値を反映するように加工したTL値。そのため、同じ音色で音量だけが違う場合、データ内には複数個の音色データが定義される(音色定義時にドライバ側でTLを再計算させるのを避けて負荷軽減するため)。

PSG 音色

次の値が可変長で並ぶ。

[00] ~ [15] 音量
[16] ~ [47] ノイズ周波数指定 (-16 した値を PSG レジスタ #6 に)
[48] ~ [51] トーン / ノイズ指定
[52] ~ [67] ハードエンベロープ指定 (-52 した値を PSG レジスタ #13 に)
[68]+[env_period.w] ハードエンベロープ周期指定
[255]+[loop_offset] 音色データ終了

発音が最後まで到達すると、発音位置から loop_offset の値を引く。loop_offset は 1 バイトなので、255 バイト以上前にループさせることはできない。