

3.2 MSX-MUSIC 拡張BASIC

3.2 MSX-Music Extended BASIC

3.2 拡張BASIC

3.2 Extended BASIC

3.2.1 概要

3.2.1 Summary

MSX-MUSICには、各機能を簡単に使用できるように、MSX-MUSIC拡張BASICが用意されています。使い方はCALL MUSICのように拡張ステートメントの形式です。CALLは_(アンダーバー)で代用できます。

MSX-MUSICの拡張BASICはMSX-AUDIO拡張BASICのサブセット版です。MSX-AUDIO用のPCM関連コマンドなどは使用できません。また、音色もできるだけMSX-AUDIOの音色に近づけてありますが、多少異なりますし、ユーザー音色の制限(MSX-MUSICでは1音色だが、MSX-AUDIOでは9音色)もあります。使えないコマンドについては、「3.2.5 MSX-AUDIOのステートメント」を参照して下さい。

MSX-MUSIC extension BASIC is prepared for MSX-MUSIC so that each function can be used easily. Usage is the form of an extended statement like CALL MUSIC. _(underbar) can be substituted for CALL.

MSX-MUSIC extended BASIC is a subset of MSX-AUDIO extended BASIC. PCM related commands for the MSX-AUDIO and cannot be used. And tone might be as close as possible MSX-AUDIO sound but slightly different and there is also a limit of 1 user tone at MSX-MUSIC, against MSX-AUDIO 9 tones. Please see "3.2.5 Statement of MSX-AUDIO" for the commands that cannot be used.

3.2.2 この章の表記法

3.2.2 Notation for this chapter

<deleted image>

3.2.3 拡張BASICコマンド一覧

1. 拡張ステートメント(CALL文と共に使用します)

コマンド名	機能	ページ
AUDREG	音源LSIのレジスタに値を書き込みます。 Write a value to the register of the sound source LSI.	228
BGM	バックグラウンド処理を行うかどうかを指定します。 Select whether to perform background processing.	228
MUSIC	MSX-MUSICシステムを初期化します。 Initialize the MSX-MUSIC system.	229
PITCH	FM音源の楽音の音高(ピッチ)を与えます。 Set the FM synthesis sound pitch (pitch).	230
PLAY	音楽をミュージックマクロランゲージにしたがって演奏します。 Play music according to the macro language to music.	231
PLAY	PLAY文が音楽を演奏中かどうかを返します。 Return whether the PLAY statement is playing music.	235
STOPM	バックグラウンドで実行中のPLAY文の演奏を停止します。 Stop playing the PLAY statement running in the background.	235
TEMPER	音律(テンペラメント)を与えます。	236

	Set the temperament.	
TRANPOSE	FM音源の楽音に対してセント単位で移調を与えます。 Set the transposition in cents for the musical sound of FM synthesis.	237
VOICE	FM音源の各チャンネルに音色(ボイス)を直接に指定します。 directly sets a tone (Voice) for each channel of the FM sound.	237
VOICE COPY	音色パラメータデータの転送を行います。 Instrument parameter data copy.	242

3.2.4 拡張BASICの解説

CALL AUDREG

機能

Function

音源LSIのレジスタに値を書き込みます。
Write a value to a register of the sound source LSI.

書式

Syntax

CALL AUDREG(<レジスタ番号>, <値> [, <チャンネル番号>])
CALL AUDREG (<Register number>, <Value> [<Channel number>])

解説

Explanation

音源LSIのレジスタに対する書き込みを行います。システムソフトウェアが割り込みなのでひんぱんに書き込んでいるレジスタには、効果がない場合やシステムの再立ち上げが必要な場合があります。

<チャンネル番号>は0、または省略しなければなりません。これはMSX-AUDIO拡張BASICとの互換性をとるために用意されているものでMSX-MUSICでは意味を持ちません。

Write to the register of the sound source LSI. The register that is writing frequently, you may re-boot the system and if there is no effect of the required system software because the interrupt.

Executes a writing to a sound module LSI register. The system software works on the interrupts so if the register is being frequently written, this command may not be effective or a restarting of the system may become necessary.

<Channel number> must be omitted or 0. This does not have a meaning in MSX-MUSIC with those that are provided for compatibility with MSX-AUDIO Extended BASIC.

文例

Example

CALL AUDREG(&H20,0)

CALL BGM

機能

Function

バックグラウンド処理を行うかどうかを指定します。
Selects whether to perform background BGM processing.

書式
Syntax

CALL BGM(<変数>)
CALL BGM(<Variable>)

解説
Explanation

<変数>は0または1の値をとり、次のような意味を持ちます。
0 PLAY文のバックグラウンド処理を行わない。1 PLAY文のバックグラウンド処理を行う。
0を与えることでフォアグラウンド処理とすることができます。
<Variable> The value of 0 or 1 is taken and it has the following meanings.
0 Don't perform background processing of a PLAY sentence.
1 Perform background processing of a PLAY sentence.
It can be considered as foreground processing by giving 0

文例
Example

CALL BGM(0)
PLAY文のバックグラウンド処理を行わない。
CALL BGM(1)
PLAY文のバックグラウンド処理を行う。
CALL BGM(0)
PLAY Without background processing of the statement.
CALL BGM(1)
PLAY With background processing of the statement.

CALL MUSIC

機能
Function

MSX-MUSICシステムを初期化します。
Initializes the MSX-MUSIC system.

書式
Syntax

CALL MUSIC{([<モード>],[0],[<PLAY文第1文字列へのチャンネル数>],[<PLAY文第2文字列へのチャンネル数>],[...],[<PLAY文第9文字列へのチャンネル数>]]]}
CALL MUSIC{([Mode] [, [0] [, [number of channels for the PLAY statement 1 string] [, [number of channels for the PLAY sentence 2 string] [,... [, [number of channels for the PLAY statement 9 string]]]]]}
CALL MUSIC{([Mode] [, [0] [, [number of channels for the PLAY statement 1 string] [, [number of channels for the PLAY sentence 2 string] [,... [, [number of channels for the PLAY statement 9 string]]]]]}

解説
Explanation

音源LSIの初期化とともに9個のFM音源のチャンネルをどのように使用するかを指定します。MUSIC文により初期化を行うまでは、すべての拡張BASICステートメントは使用できません。<モード>は0か1で、0はリズム音を使用しないモードで、1はリズム音を使用するモードです。リズム音を使用する場合にはチャンネル7、8、9を使用しますので、楽音に使えるのは残り6チャンネルになります。したがって、PLAY文で使用するチャンネル数はリズム使用時には6以下、リズムを使わないときは9以下である必要があります。チャンネルの使用割り当ては、PLAY文では、チャンネル番号の小さい方(1,2,3...)から割り当て

ます。

パラメータを1つ以上指定した場合、他のパラメータの省略時の値は0となります。PLAY文の文字列へのチャンネル数を、0に設定したり省略したりすることはできません。次の例を参照して下さい。

Specifies how to use the 9 channels of FM sound along with the initialization of the sound source LSI. No extended BASIC statements will be available until it initializes by a MUSIC sentence. The <mode> is 0 or 1, 0 is the mode which does not use rhythm sound, and 1 is the mode which uses Rhythm sound. Rhythm sounds use use channel 7, 8, 9, so 6 channels are left for musical score. Therefore, the number of channels used in a PLAY sentence needs to be nine or less, when not using rhythm and 6 or less when rhythm is used.

The use of channel assignment, the PLAY statement, assignment from (1, 2, 3, ...) the smaller channel number. If you specify one or more of the parameters, the default value of other parameters is 0.

You will not be able to or omitted, set to 0, the number of channels to a string of PLAY statement. Refer to the following examples.

```
CALL MUSIC(0, 0, 0, 5, 0)
           ↑      ↑
           0を設定してはいけない(Illegal function callになる)
           Do not set to 0 (Results in Illegal function call)
```

```
CALL MUSIC(0, 0, 1, , 0)
           ↑
           省略してはいけない(Syntax errorになる)
           Do not omit (Results in Syntax error)
```

パラメータなしで使われた時には、

When it is used without parameters, it will be the same as

```
CALL MUSIC(1, 0, 1, 1, 1)
```

と同じになります。すなわち、

- FM音源のチャンネル1をPLAY文の最初の文字列に割り当てる。
- FM音源のチャンネル2をPLAY文の2番目の文字列に割り当てる。
- FM音源のチャンネル3をPLAY文の3番目の文字列に割り当てる。
- リズム音をPLAY文の4番目の文字列に割り当て使用する。
- PLAY文の5番目以降7番目までの文字列はPSG音源の制御に割り当てる。

という意味になります。

MUSIC文を実行すると、システムの割り込みのフックがMSX-MUSICのシステムソフトウェアにリンクされるので割り込み処理ルーチンのオーバーヘッドが増え、システムのスループットが低下します。また、MUSIC文はワークエリア確保のために内部でCLEAR文に相当することを行っているので、HIMEM(CLEAR文の第2パラメーターに相当します)が807バイト小さく再設定され、変数はすべてクリアされます。

That is,

- The channel 1 of an FM sound is assigned to the character string of the beginning of a PLAY sentence.
- The channel 2 of an FM sound is assigned to the 2nd character string of a PLAY sentence.
- The channel 3 of an FM sound is assigned to the 3rd character string of a PLAY sentence.
- Rhythm sound is assigned to the 4th character string of a PLAY sentence, and is used.
- From the fifth PLAY sentence until the seventh string assign PSG sound source control.

The overhead of interrupt handling routine increases when you run the statement MUSIC, hooks interrupts because the system is linked to the system software of MSX-MUSIC, the throughput of the system is reduced. In addition, when the MUSIC statement is executed it wil run the CLEAR

statement internally to reserve work area space, is re-set to a small 807 byte (equivalent to the second parameter of the CLEAR statement) HIMEM, and all variable will be cleared.

文 例

Example

CALL MUSIC

デフォルトの設定をする。

For the default settings.

CALL MUSIC(0,0,1,1,1,1,1,1,1,1,1)

1チャンネルずつPLAY文の文字列に割り当てる。

Assign one string of PLAY to each channel.

CALL PITCH

機 能

Function

FM音源の楽音の音高(ピッチ)を与えます。

Set the pitch of a musical tone of FM sound.

書 式

Syntax

CALL PITCH(<ピッチ1>[,<ピッチ2>])

CALL PITCH(<pitch 1>, [<pitch 2>])

解 説

Explanation

FM音源で発生する楽音の音高を指定します。<ピッチ>の範囲は410~459で単位は[Hz]です。中央のCのすぐ上のAの音(a2)の周波数で音高を表します。トランスポーズとは独立に設定でき、デフォルトの値は440です。

ピッチ(またはトランスポーズ値)を変えるとリズム音や音程を持たない音を除くFM音の音の高さが増減します。PSG音源には作用しないので注意して下さい。

ピッチ2は指定しても無視されます。これはMSX-AUDIO拡張BASICとの互換性をとるために用意されているものでMSX-MUSICでは意味を持ちません。

トランスポーズについてはTRANSPOSEの項を参照して下さい。

I specify the pitch of the musical tone generated by the FM sound. The range of <pitch> in the 410-459 unit is [Hz]. I represent the pitch at the frequency of (a2) the sound of A just above middle C. Can be set independently of the transpose, the default value is 440.

Pitch of the FM sound except for the sound that does not have the pitch and rhythm sounds change as you change the (transpose value) or pitch. Please keep in mind that it does not act on the PSG sound source.

Pitch 2 is ignored if specified. This does not have a meaning in MSX-MUSIC with those that are provided for compatibility with MSX-AUDIO Extended BASIC.

Refer to the section of TRANSPOSE for transposing.

文 例

Example

CALL PITCH(440)

PLAY

機能 Function

音楽をミュージックマクロランゲージ(MML)にしたがって演奏します。
Plays music according to the music macro language (MML).

書式 Syntax

PLAY[#<モード>,<文字列1>[,<文字列2>[,<文字列3>]...[,<文字列13>]]
PLAY[#<Mode> <string 1> <string 2>, <string 3>] ... [, <string 13>]

解説 Explanation

PLAY文は音楽を演奏する物で、FM音源(9)、従来のPSG音源(3)の最大12声まで同時発声が可能です。<文字列>に書かれたミュージックマクロランゲージ(MML)にしたがって演奏します。他の拡張命令と異なりCALL文は必要ありません。

<モード>は0から3までの値をとり、PLAY文の音源や動作モードを次のように設定します。

- 0や省略されたときはPSGのみが音源となり、文字列は最大3つまでとなります。従来のPLAY文と互換性があります。
- 1のとき、「Illegal function call」となります。これはMSX-AUDIO拡張BASICとの互換性をとるために用意されているもので、MSX-MUSICでは意味を持ちません。
- 2または3のとき、FM音源、リズム音、PSG音源を使用できます。(2のときと3のときで動作に違いはありません)。

<文字列>と音源との関係は始めから順に、

<FM音源用文字列1>,...,<FM音源用文字列n>,<リズム音用文字列>,<PSG音源用文字列1>,<PSG音源用文字列2>,<PSG音源用文字列3>

となります。nはMUSIC文で設定されたミュージックマクロランゲージの個数です。

MUSIC文でリズム音を使用しないモードに設定した場合は、リズム音用文字列をカンマと共に省略しなければいけません。

例として初期設定のMUSIC文に対する文字列の配列をあげると次のようになります。

<FM音源用文字列1>,<FM音源用文字列2>,<FM音源用文字列3>,<リズム音用文字列>,<PSG音源用文字列1>,<PSG音源用文字列2>,<PSG音源用文字列3>

Those that play music, can be up to 12 concurrent speech voice maximum FM sound source (9), PSG sound traditional (3) PLAY statement. Plays music macro language they are written in the <string> according to (MML). Unlike the other extension instruction CALL statement is not required. <mode> takes the values from 0 to 3, and sets up the sound source and operational mode of a PLAY sentence as follows.

- 0 and when it is omitted, only PSG serves as a sound source, and a character string is set to a maximum of three. There are a conventional PLAY sentence and compatibility.
- When 1, it will be "Illegal function call". The "1" is provided for compatibility with MSX-AUDIO extended BASIC, which has no meaning in MSX-MUSIC.
- When 2 or 3, it can use FM sound, rhythm sounds, the PSG sound source. (There is no difference in behavior when three in the case of 2).

The relation between a <character string> and a sound source becomes order from the start with <the character string 1 for FM sounds>, --, the <character string n for FM sounds>, <the character string for rhythm sound>, the <character string 1 for PSG sound sources>, the <character string 2 for PSG sound sources>, and the <character string 3 for PSG sound sources>.

n is the number of the music macroscopic language set up in the MUSIC sentence.

When it sets to the mode which does not use rhythm sound in a MUSIC sentence, you have to omit the character string for rhythm sound with a comma.

It is as follows if the array of the character string to the MUSIC sentence of initial setting is raised as an example.

<The character string 1 for FM sounds>, the <character string 2 for FM sounds>, the <character string 3 for FM sounds>, <the character string for rhythm sound>, the <character string 1 for PSG sound sources>, the <character string 2 for PSG sound sources> <character string 3 for PSG sound sources>

文 例
Example

PLAY#2,"CD","EF","GA"

ミュージックマクロランゲージ(MML)の仕様
Specification of the Music Macro Language (MML)

FM音源、PSG音源MMLの仕様
Specifications for FM sound source, PSG sound source MML

表7.23 FM音源、PSG音源用MMLの仕様一覧
Table 7.23 Specification list of FM synthesis, PSG sound source for MML

文 字 Character	意 味 Meaning	値のとり範囲 Range	初期設定値 default value
Mn	エンベロープ周期の設定 Set the envelope period	$1 \leq n \leq 65535$	M255
Sn	エンベロープ形状の設定 Set the envelope shape	$0 \leq n \leq 15$	S0
Vn	音量の設定 Set the volume	$0 \leq n \leq 15$	V8
Ln	長さの設定 Length setting	$1 \leq n \leq 64$	L4
Qn	音の長さの割合 Ratio of the length of the sound	$1 \leq n \leq 8$	Q8
On	オクターブの設定 Set the octave	$1 \leq n \leq 8$	O4
>	オクターブを1つ上げる Raise one octave		
<	オクターブを1つ下げる Lower one octave		
Tn	テンポの設定 Setting the tempo	$32 \leq n \leq 255$	T120
Nn	nで指定された高さの音を発生する The sound of the height specified by n is generated.	$0 \leq n \leq 96$	
Rn	休符の設定 play a pause	$1 \leq n \leq 64$	R4
A~G	音程の発生 Occurrence of pitch		
+,#	音を半音上げる Raise the sound semitone		
-	音を半音下げる Lower the sound semitone		
.(ピリオド)	音符や休符の長さを1.5倍する The length of a note or a rest is increased 1.5 times		
=x;	パラメータnを変数xで設定する Set the parameter n in the variable x		
Xx;	文字変数xに入っているMMLを演奏する(*1) Plays MML contained in the string variable x (*1)		
&	タイ、前後の音をつなぐ Connects two notes		
{ } n	連符、n分音符を { } の中の音程の個数で等分にした音を発生する、 $1 \leq n \leq 64$ And generates a sound that was in the time being by the number of intervals in { } tuplet, the n minutes note, $1 \leq n \leq 64$ This is tuplet, and generates sounds of which a nth note is	Lnで設定された値	

	equally divided by the quantity of pitches that are written between {}. $1 \leq n \leq 64$		
@n	n番の音色に切り替える Changes to the tone of No. n.	$0 \leq n \leq 63$	
@Vn	音量を細かく設定する fine-tune the volume	$0 \leq n \leq 127$	
@Wn	nで指定された長さだけ状態を継続する The length specified in the n simply continues to state Continues the state by the length specified by n.	$1 \leq n \leq 64$	Lnで設定された値
Yr,d	音源LSIのレジスタrにdを書き込む d is written in the register r of sound generator LSI.		
Zd	「Illegal function call」になる(*2) Results in "Illegal function call" (*2)		

*1 このマクロを指定した場合、このマクロ以降に何かマクロを書くことはできません。書いた場合はエラーとなります。

*2 MSX-AUDIOの場合、MIDIポートへの出力という意味ですが、MSX-MUSICの場合はエラーとなります。

*1 If you use this macro, you can not write a macro after this macro. It results in error if you do it.

*2 In MSX-AUDIO, it is meant as the output to a MIDI port, but in MSX-MUSIC, it results in error.

リズム音用MMLの仕様

Specification of MML for rhythm sound

リズム音の場合、1つのMMLで同時にいくつかの音を発生するために音楽用とは異なった記述様式をとります。まずならしたい楽器を並べてその後に長さを記述します。

In the case of rhythm sound, since some sound is simultaneously generated in one MML, a different description style from the object for music is taken. Describes the length and then side-by-side instrument you want to play first.

表7.24 リズム音用MMLの仕様一覧

Table 7.24 Specification of MML for rhythm sound

文字 Character	意味 Meaning	値のとり範囲 Range	初期設定値 default value
B	バスドラム音を発生 Generates bass drum sound		
S	スネアドラム音を発生 Generates snare drum sound		
M	タムタム音を発生 Generates tom-tom sound		
C	シンバル音を発生 Generates cymbal sound		
H	ハイハット音を発生 Generates hi-hat sound		
!	直前の楽器の音量をアクセントボリュームにする Volume of the last musical instrument is made into accent volume.		
n	直前までに書かれた楽音を発生し、n分音符分待つ Generates a tone that was written immediately before, and waits n minutes note content	$1 \leq n \leq 64$	
Vn	アクセントの付いていない楽音の音量を設定する Set the volume of the tone that is not accented	$0 \leq n \leq 15$	8

@An	アクセントの付いている楽音の音量を設定する Set the volume of the tone that is accented	$0 \leq n \leq 15$	
-----	--	--------------------	--

Tn、@Vn、Rn、=x;、Xx;、.はFM音源と同様です。

Tn, @ Vn, Rn, = x;, Xx;., Is similar to the FM sound.

例

Example

"BSH8H8S!H8H8"

- バス、スネア、ハイハットを鳴らし、8分音符分待ちます。
- ハイハットを鳴らし、8分音符分待ちます。
- スネアをアクセント付きでハイハットと鳴らし8分音符分待ちます。
- ハイハットを鳴らし、8分音符分待ちます。
- Plays bass drum, snare, hi-hat, and wait for eighth note minute.
- Plays the hi-hat, and wait eighth note minute.
- Waits for eighth note minute and plays the hi-hat accented snare.
- Plays the hi-hat, and wait eighth note minute.

MMLと各音源との対応

Correspondence between MML and each sound source

表7.25 MMLと各音源との対応一覧

Table 7.25 List of Correspondence between MML and each sound source

文 字 Char	FM音源 FM chip	PSG音源 PSG chip
Mn	*1	✓
Sn	*1	✓
Vn	✓	✓
Ln	✓	✓
Qn	✓	*1
On	✓	✓
>	✓	✓
<	✓	✓
Tn	✓	✓
Nn	✓	✓
Rn	✓	✓
A~G	✓	✓
+,#	✓	✓
-	✓	✓
.	✓	✓
=x;	✓	✓
Xx;	✓	✓
&	✓	✓
{ } n	✓	*3
@n	✓	*1
@Vn	✓	*1
@Wn	✓	*2

Yr,d	✓	*1
Zd	*4	*1

*1 無視されます。

*2 Rnと同等です。

*3 PSG音源に対しては使用できません。使用するとエラーになります。

*4 MSX-MUSICではエラーになります。

*1 Will be ignored

*2 Is equivalent to Rn.

*3 It can not be used against PSG sound source.It will result in error if it is used.

*4 In MSX-MUSIC, it will result in error.

CALL PLAY

機能 Function

PLAY文が音楽を演奏中かどうかを返します。

Checks whether PLAY statement is playing music.

書式 Syntax

CALL PLAY(<PLAY文のSTRING番号>,<変数名>)

CALL PLAY([PLAY statement string number], [variable name])

解説 Explanation

PLAY文のミュージックキューの状態を調べ、各チャンネルが音楽を演奏中かどうかを判断し、演奏中であれば-1、そうでなければ0の値を返します。ただし、STRING番号として0が与えられた場合はいずれかのSTRINGが演奏中であれば-1を、そうでなければ0を返します。

PLAY文のSTRING番号は、MUSIC文で指定したSTRING数+3まで使えます。すなわち、MUSIC文で指定したFM音源に加え3チャンネルのPSG音源について有効です。

Each channel plays music, examine the state of PLAY music queue if while playing, and then determine whether or not in 1, otherwise returns a value of 0. However, if 0 is given as a string number returns 0 if any channel is playing and 1 otherwise.

String number for the PLAY statement can be used up to + 3 string number specified in MUSIC. Namely, in addition to FM MUSIC statement, 3 channel PSG sound source is valid.

文例 Example

CALL PLAY(0,A):PRINT A

CALL STOPM

機能 Function

バックグラウンドで実行中のPLAY文の演奏を停止します。

Stop playing the PLAY statement running in the background.

書式
Syntax

CALL STOPM

解説
Explanation

バックグラウンドで実行中のPLAY文の音楽の演奏を停止します。
Stop playing the PLAY statement running in the background.

CALL TEMPER

機能
Function

音律(テンペラメント)を与えます。
Set the temperament.

書式
Syntax

CALL TEMPER(<音律番号>)
音律番号 0~21
CALL TEMPER (<Temperament number>)
Temperament number 0~21

解説
Explanation

音律を与えるステートメントで、FM音源の楽音の音高に影響を与えます。
音律は1オクターブをどのような比率で12音に分割するかを決めるもので、古典音楽には古典音律が適していると言われます。
デフォルト値は9番の完全平均律です。
A statement that gives the temperament, which in turn affects the pitch of the musical tone of FM sound.
Temperament decides by what kind of ratio to divide 1 octave into 12 sound, and classical music is told that classic temperment is suitable.
The default value is the perfect temperament of No. 9.

番号 Number	音律 Temperament
0	ピタゴラス Pythagoras
1	ミーントーン Meantone
2	ヴェルクマイスター Velcumeister
3	ヴェルクマイスター(*) Velcumeister (*)
4	ヴェルクマイスター(別) Velcumeister (other)
5	キルンベルガー Kirnberger
6	キルンベルガー(*) Kirnberger (*)

7	ヴァロッティ・ヤング Vallotti, Young
8	ラモー Rameau
9	完全平均律(デフォルト) Completely equal temperament (default)
10	純正律cメジャー(aマイナー) Just intonation c major (a minor)
11	純正律cisメジャー(b) Just intonation major cis (b)
12	純正律dメジャー(h) Just intonation major d (h)
13	純正律esメジャー(c) Just intonation major es (c)
14	純正律eメジャー(cis) Just intonation e major (cis)
15	純正律fメジャー(d) Just intonation major f (d)
16	純正律fisメジャー(es) Just intonation major fis (es)
17	純正律gメジャー(e) Just intonation major g (e)
18	純正律gisメジャー(f) Just intonation gis major (f)
19	純正律aメジャー(fis) Just intonation a major (fis)
20	純正律bメジャー(g) Just intonation major b (g)
21	純正律hメジャー(gis) Just intonation major h (gis)

(*)は平島達司氏による。

(*) according to Mr. Tatsuji Hirashima.

文例

Example

CALL TEMPER(0)

CALL TRANSPOSE

機能

Function

FM音源の楽音に対してセント単位で移調を与えます。

Set the transposition in cents for the musical sound of FM synthesis.

書式

Syntax

CALL TRANSPOSE(<トランスポーズ値1>[, <トランスポーズ値2>])

CALL TRANSPOSE(<Transpose value 1> [, <transpose value 2>])

解説

Explanation

移調を行うためのステートメントで、単位はセントです。これは、半音を100とした移調の単位で、1オクターブ上げるには、+1200を与えます。

トランスポーズ値として許される範囲は、±12799以内ですが、実際にはFM音源の音色によってある高さの範囲以外は制限されます。音高制度はLSIの制限により、±2セントていどです。

トランスポーズはピッチとは独立して設定できます。MUSIC文による初期化の値は0です。ピッチについてはPITCH文を参照してください。

トランスポーズ値2は指定されても無視します。これはMSX-AUDIO拡張BASICとの互換性をとるために用意されているものでMSX-MUSICでは意味を持ちません。

It is a statement for transposing and a unit is a cent.

This gives +1200, in order to be a unit of the transposition set to 100 and to raise 1 octave of seminotes.

Range for allowable transpose value is within ± 12799 , but outside the range of a certain height by the sound of FM synthesis is limited in practice. Due to limitations of LSI, the pitch system is two cents about \pm .

Transpose can be set independently of the pitch. Value of the initialization by MUSIC statement is 0. Please refer to the PITCH sentence for pitch.

The transposition value 2 is ignored even if specified. This provided in order to keep compatibility with MSX-AUDIO extension BASIC, and it does not have a meaning in MSX-MUSIC.

文 例 Example

```
CALL TRANSPOSE(0) CALL TRANSPOSE(0,700)
```

CALL VOICE

機 能 Function

FM音源の各チャンネルに音色（ボイス）を直接設定します。

Directly sets the tone (Voice) for each channel of the FM sound.

書 式 Syntax

```
CALL VOICE([<チャンネル1用のボイス>],[<チャンネル2用のボイス>],...[<チャンネル9用のボイス>])
```

ボイス=@+数値 または、配列変数名

```
CALL VOICE([<Voice for channel 1>], [<voice for channel 2>], ..., [<voice for channel 9>])
```

Voice = @+ numerical value, or array variable name

解 説 Explanation

音源LSIの9チャンネルあるFM音源のそれぞれに音色を設定します。

音色の設定方法には2つあります。

システムに備えられている音色ライブラリを使う場合には、0~63の音色の番号を数式により指定します。この場合には数式の前に@記号をつけて次の配列変数名と区別します。

プログラムにより音色パラメータを与えて設定する場合には、配列変数に音色パラメータを入れてその配列変数名を指定します。

音色パラメータのフォーマットの詳細はVOICE COPY文の解説を参照してください。

パラメータを省略したチャンネルの音色は変更されません。

音色ライブラリ（表7.26 音色ライブラリ一覧表参照）のうち*印が付いていない音色や、配列変数で設定する場合は同時に1音色しか設定できません。複数設定しようとしたときはパラメータ列の一番右側のパラメータが最後に実行したCALL VOICE文の設定のみが有効となります。

Sets the tone for each of the FM signal source from channel 9 of the sound source LSI.

There are two ways to set the tones.

If you use the built-in sound library system, specify a number of 0~63. In this case, @ sign is attached before expression and it distinguishes from the following array variable name.

If you want to set the tone parameters given by the program, specify the array variable name to put the sound parameters to the array variable.

Please refer to the description of the VOICE COPY statement details of the format of tone data parameters. Does not change the timbre of a channel whose parameter is omitted.

When setting up with the tone to which * mark is not attached among tone libraries (Table 7.26 refer to tone library table), and an array variable, only one tone can be set up simultaneously.

When it is going to carry out multidata input, only a setup of the CALL VOICE sentence performed at the parameter or the last on the rightmost side of a parameter sequence becomes effective.

文 例
Example

CALL VOICE(@26,@27)

チャンネル1の音色は2と同じ27番の音色になります。

The tone of the channel 1 turns into the same tone of No. 27 as channel 2.

CALL VOICE(@20,@10)

CALL VOICE(,@21)

チャンネル1の音色は3と同じ21番となります。

The tone of the channel 1 will be same tone 21 as channel 3.

CALL VOICE(A,,B)

チャンネル1の音色は3と同じBという配列変数で設定される音色となります。

Tone of channel 1 will be tone set by the array variable named the same as the B 3.

The instrument of channel 1 will be same with channel 3, which is the instrument set by the array variable called B.

文 例
Example

CALL VOICE(@0,@0,@0,,, @7,@7,@7) CALL VOICE(@0,@0)

略号は音色の名前としてライブラリに登録されているものです。

Abbreviations are those that are registered in the library as the name of the tone.

表7.26 音色ライブラリー一覧表

Table 7.26 Tone library list

音色番号 Tone number	音色名 Voice Name	略号 Abbreviation
0	Piano 1*	Piano 1
1	Piano 2	Piano 2
2	Violin*	Violin
3	Flute 1*	Flute
4	Clarinet*	Clarinet
5	Oboe*	Oboe
6	Trumpet*	Trumpet
7	Pipe Organ 1	PipeOrgn
8	Xylophone	Xylophon
9	Organ*	Organ
10	Guitar*	Guitar
11	Santool 1	Santool
12	Electric Piano 1*	Elecpian
13	Clavicode 1	Clavicod
14	Harpsicode 1*	Harpsicd
15	Harpsicode 2	Harpscd2
16	Vibraphone*	Vibraphn
17	Koto 1	Koto
18	Taiko	Taiko

19	Engine 1	Engine
20	UFO	UFO
21	Synthesizer bell	SynBell
22	Chime	Chime
23	Synthesizer bass*	SynBass
24	Synthesizer*	Synthsiz
25	Synthesizer Percussion	SynPercu
26	Synthesizer Rhythm	SynRhyth
27	Harm Drum	HarmDrum
28	Cowbell	Cowbell
29	Close Hi-hat	ClseHiht
30	Snare Drum	SnareDrm
31	Bass Drum	BassDrum
32	Piano 3	Piano 3
33	Electric Piano 2*	Elecpia2
34	Santool 2	Santool2
35	Brass	Brass
36	Flute 2	Flute 2
37	Clavicode 2	Clavicd2
38	Clavicode 3	Clavicd3
39	Koto 2	Koto 2
40	Pipe Organ 2	PipeOrg2
41	PohdsPLA	PohdsPLA
42	RohdsPRA	PohdsPRA
43	Orch L	Orch L
44	Orch R	Orch R
45	Synthesizer Violin	SynViol
46	Synthesizer Organ	SynOrgan
47	Synthesizer Brass	SynBrass
48	Tube*	Tube
49	Shamisen	Shamisen
50	Magical	Magical
51	Huwawa	Huwawa
52	Wander Flat	WnderFlt
53	Hardrock	Hardrock
54	Machine	Machine
55	Machine V	MachineV
56	Comic	Comic
57	SE-Comic	SE-Comic
58	SE-Laser	SE-Laser
59	SE-Noise	SE-Noise
60	SE-Star 1	SE-Star
61	SE-Star 2	SE-Star2
62	Engine 2	Engine 2
63	Silence	Silence

音色パラメータデータのフォーマット

音色パラメータは次の表のように1音色あたり32バイトのデータを使います。
オペレータ0とオペレータ1のデータは同じものの繰り返しになっています。

Instrument parameters data format

A tone parameter uses 32 bytes per one tone of data, as shown in the next table.
The data of the operator 0 and the operator 1 is a repetition of the same thing.

表7.27 音色パラメータのフォーマット
Table 7.27 Format of a tone parameter

オフセット Offset	内容 Content	
ヘッダ Header		
0~7	音色名 Voice Name	
8~9	ボイス移調 Voice transposition	
10	bit0	アルゴリズム* Algorithm *
	bit1~3	フィードバック Feedback
	bit4	固定ピッチ* Fixed pitch *
	bit5	AMD/PMDロード可能* AMD/PMD load possible *
	bit6	PMD*
	bit7	AMD*
11~15	予約 Reserved	
オペレータ0 Operator 0		
16	bit0~3	MULT
	bit4	KSR
	bit5	EG
	bit6	PM
	bit7	AM
17	bit0~5	トータルレベル Total level
	bit6~7	レベルキースケール Key scale level
18	bit0~3	ディケイレート Decay rate
	bit4~7	アタックレート Attack Rate
19	bit0~3	リリースレート Release Rate
	bit4~7	サスティンレベル Sustain level
20	bit4~7	ベロシティセンシビリティ(0~8)* Velocity sensibility (0~8)*
21~23	予約 Reserved	

オペレータ1		
24	bit0~3	MULT
	bit4	KSR
	bit5	EG
	bit6	PM
	bit7	AM
25	bit0~5	トータルレベル* Total level
	bit6~7	レベルキースケール Key scale level
26	bit0~3	ディケイレート Decay rate
	bit4~7	アタックレート Attack Rate
27	bit0~3	リリースレート Release Rate
	bit4~7	サスティンレベル Sustain level
28	bit4~7	ベロシティセンシビリティ(0~8)* Velocity sensibility (0~8)*
29~31	予約	

*はMSX-AUDIOとの互換性のためにあるものです。MSX-MUSICでは意味を持ちません(無視されます)。

* For compatibility with MSX-AUDIO. In MSX-MUSIC, it does not have a meaning (ignored).

CALL VOICE COPY

機能 Function

音色パラメータデータの転送を行います。
Do the instrument parameter data transfer

書式 Syntax

CALL VOICE COPY(<パラメータ1>,<パラメータ2>)
パラメータ1=@+数式または配列変数名
パラメータ2=@+数式(結果は63でなければなりません)または配列変数名

Parameter 1=@+ expression or an array variable name.

Parameter 2=@+ expression (a result will not become, if it is not 63), or an array variable name

解説 Explanation

配列と音色ライブラリ(0~63)の間でのデータの転送を行います。パラメータ1の音色パラメータをパラメータ2に転送します。@と数式が指定されたときは、その数値の結果で指定される音色番号の音色データが対象となります。

The data between arrangement and a tone library (0?63) is transmitted. The tone parameter of the parameter 1 is transmitted to the parameter 2. When @ and expression are specified, the tone data of the timbre number specified by the result of the figure is applicable.

ソース(パラメータ1)に指定できる音色番号は0~63のうち*印の付いていない音色の番号ですが、ディスティネーション(パラメータ2)に指定できる音色番号は63に限ります。ソース(パラメータ1)に*印の付いている音色番号を指定すると「Illegal function call」となります。

Although the timbre number which can be specified as sauce (parameter 1) is a number of the tone to which * mark is not attached among 0?63, the timbre number which can be specified as a destination (parameter 2) is restricted to 63. If the timbre number to which * mark is attached to sauce (parameter 1) is specified, it will become "Illegal function call."

@記号がない場合の変数名は配列変数とみなされ、その内容が転送の対象になります。1つの音色パラメータは32バイトの長さがあります。音色パラメータの詳細については、「表7.27 音色パラメータデータのフォーマット」を参照してください。

It is considered that a variable identifier in case there is no @ sign is an array variable, and the contents are the targets of transmission. One tone parameter has a length of 32 bytes. Please refer to "Table 7.27 Format of tone parameter data" for the details of a tone parameter.

<p style="text-align: center;">文 例 Example</p>
--

CALL VOICE COPY(@17,@63)

17番の音色データを63番に転送する。

The tone data of No. 17 is transmitted to No. 63

DIM A%(16):CALL VOICE COPY(@28,A%)

28番の音色データを配列変数A%に転送する。

The tone data of No. 28 is transmitted to A% of an array variable.

CALL VOICE COPY(A%,@63)

配列変数A%の音色データを63番に転送する。

The tone data of A% of an array variable is transmitted to No. 63.

3.2.5 MSX-AUDIOのステートメント

3.2.5 MSX-AUDIO Statements

以下のMSX-AUDIOのステートメントは全て「Illegal function call」となります。

文法上の間違いがあっても同様です。

All of the statement of the following MSX-AUDIO become "Illegal function call." It is the same even if the mistake on grammar occurs.

ADPCM/PCM関連のステートメント

[ADPCM/PCM-related statements](#)

CONVA, CONVP, COPY PCM, LOAD PCM, PCM FREQ, PCMVOL, PLAY PCM, REC PCM, SAVE PCM, SET PCM

インスツルメント関連のステートメント

[Instrument related statements](#)

INMK, KEY ON/KEY OFF, MK PCM, MK TEMPO, MK VEL, MK VOICE, MK VOL

MK記録関連のステートメント

[MK record statements](#)

APPEND MK, CONT MK, MK STAT, PLAY MK, REC MK RECMOD

外部プログラムの呼び出し

[Call of an external program](#)

SYNTHE, APEEK, APOKE

