＜BASIC言語の書式＞
BASICプログラムの記述は、行番号式、ラベル式、その混合、いずれの方法でも構いません。以下、仕様について述べます。

＜利用可能な変数型＞
利用できる変数の型は、32ビット符号付整数(-2147483648 以上 +2147483647 以下)と、文字列型の２種類です。文字列の末端部には0x00が付加されます。

A-Zの２６個の整数型変数が利用可能です。文字列として扱う場合はA$のように記述します。ただし、A（整数型）とA$（文字列型）を同時に使用することは出来ません。

整数型の定数は、１０進法で記述します。１６進法を使う場合、「$1200」のように、頭に「$」を付加するか、「0x1200」の様に表記して下さい。

文字列型の定数は、「″」で囲って記述してください。「″」を使用する場合は、「CHR$($22)」のように記述することが出来ます。

＜命令＞
以下、x, y, z等は整数値を、x$, y$, z$は文字列を指します。xxx, yyy, zzz, www等は任意のステートメントを指します。[ ]は省略可能である事を示します。

命令同士を「:」で区切ることにより、一行で複数のコマンドを処理すること
が出来ます。

BGCOLOR r,g,b
　　背景色指定。
CLEAR
　　すべての文字列型変数と整数型配列を破棄し、整数値を０とする。また、
　　PCGの使用をやめ、表示キャラクターをリセットする。
CLS
　　スクリーン消去。
COLOR x
　　テキスト色指定。
CURSOR x,y
　　カーソル位置指定。
DATA x[,y[,z[...]]]
　　データー列を整数値で指定する。
DIM xxx [, yyy [, zzz [, ...]]]
　　整数型の一次元配列を割り当てる。
　　xxx,yyy,zzzは、例えば「A(10)」のように記述する。この場合、A(0)から
　　A(10)までの１１個の整数型変数が確保される。
DRAWCOUNT
　　DRAWCOUNT値を指定する。DRAWCOUNT値に付いては、DRAWCOUNT()関数を
　　参照。
END
　　BASICプログラムを停止する。

EXEC x[,y[,z[...]]]

機械語を実行する。ただし、x,y,z は 32 ビット整数値。

FOR x=yyy TO zzz [STEP www]

NEXT

yyy で示された計算結果を x に代入し、x の値が zzz になるまで次の NEXT 文までのステートメントを、繰り返し実行する。繰り返しのたび、x の値は www ずつ増加する（省略された場合は１ずつ）。「NEXT」の次に何も記述しないことに注意。

GOSUB xxx

現在の実行位置を記憶し、xxx 行目(もしくはラベル)に移動する。

GOTO xxx

xxx 行目(もしくはラベル)に移動する。

IF x THEN yyy [ELSE zzz]

x が０以外のとき、yyy を、０のとき zzz を実行

LABEL xxx

GOTO/GOSUB のジャンプ先を登録する。xxx は、英数字６文字以内の文字列。

[LET] x=yyy

y で示された計算結果を、x に代入する。「LET」は省略可。

[LET] x$=yyy

yyy で示された文字列（もしくは連結結果;連結演算子は「+」）を、x$に代入する。「LET」は省略可。

MUSIC x$

BGM を演奏する。詳細は、下記<MUSIC>の項を参照。

PALETTE n,r,g,b

パレット指定。

PCG x,y,z

ASCII コードが x の文字の表示キャラクターを変更する。y,z は、キャラクターデーター。詳細は、下記<PCG>の項を参照。

POKE x,y

x で示される物理的アドレスに、y で示される値(１バイト値)を書き込む。

PRINT [x または x$ [,または; [y または y$ [...]]]]

ディスプレイに、整数値または文字列を表示する。「;」を使用した場合、次の表示が続けて行われる。「,」を使用した場合、１０文字ずつに区切って表示される。どちらも使用しない場合、次の表示は行を変えて行われる。

REM xxx

何も実行しない

RESTORE xxx

DATA 読み出し開始位置を指定。xxx は行番号もしくはラベル。

RETURN

最後に実行された GOSUB 文の次のステートメントに移動する。戻り値を指定することががができる。この場合の戻り値は GOSUB()関数にて取得が可能。

SCROLL x,y

画面を横方向、もしくは縦方向(斜めも可)に動かす。動かす方向と大きさは、x, y でそれぞれ、横方向の移動度、縦方向の移動度として指定する。

SOUND xxx

効果音を再生する。詳細は、下記<SOUND>の項を参照。xxx は行番号もしくはラベル。

USEPCG [x]

PCG を使用、もしくは使用停止する。x=0 で使用停止、x=1 で使用、x=2 でキャラクターをリセットして使用。x を省略した場合は、x=1 と同じ。

WAIT x

x で示された時間、プログラムの実行を停止する。x が 60 の場合、約１秒間停止。

＜関数＞

以下、x, y, z は整数値を、x$, y$, z$は文字列を指します。[ ]は省略可能である事を示します。

ABS(x)

x の絶対値を返す。

ASC(x$)

文字列の最初の一文字の、アスキーコードを返す。

DRAWCOUNT()

DRAWCOUNT 値を得る。DRAWCOUNT は１６ビット整数値で、1/60 秒ごとに１ずつ増える。

GOSUB(xxx)

GOSUB 命令と同じだが、戻り値(RETURN を参照)を得ることが出来る。xxx は、ラベルもしくは行番号。

INKEY([x])

x を指定しない場合、現在押されているキーの ASCII 値を返す。押されていない場合は、０。ASCII 値で x を指定した場合、そのキーが押されているかどうかを返す。

KEYS([x])

キー入力を得る。x の値は以下の通り。x を指定しない場合は、x=63 と同じ。

```
KEYUP:    1
KEYDOWN:  2
KEYLEFT:  4
KEYRIGHT: 8
KEYSTART: 16
KEYFIRE:  32
```

LEN(x$)

文字列の長さを返す。

MUSIC()

BGM の演奏の残り数を返す。

NOT(x)

x=0 の場合に１を、そうでない場合に０を返す。

PEEK(x)

x で示される物理アドレスから１バイト読み取り、返す。

READ()

DATA 文の後から、一つずつデーター（整数値）を読み出す。

RND()
　　0から32767までの擬似乱数を返す。
SGN(x)
　　xの符号(-1, 0, または1)を返す。
STRNCMP(x$,y$,z)
　　２つの文字列のうちz文字分を比較し、結果を返す。同じ文字列の場合は０。
TVRAM([x])
　　ビデオRAMのx番目の内容を、バイト値で返す。xを省略した場合、ビデオ
　　RAMの開始位置の物理アドレスを返す。
VAL(x$)
　　１０進数もしくは１６進数文字列としてのx$の値を、整数値で返す。
A$(x [,y])など
　　xの値が０の場合、文字列全体を返す。
　　xの値が正の場合、xで示される位置より右側の文字列を返す。
　　xの値が負のとき、文字列の右側x文字を返す。
　　yが指定された場合、y文字分の文字列を返す。
CHR$(x)
　　xをアスキーコードとする文字を返す。
DEC$(x)
　　xの値を、１０進数の文字列として返す。
HEX$(x [,y])
　　xの値を、１６進数の文字列として返す。yが指定された場合、yバイト長の
　　文字列になる。
INPUT$()
　　文字列入力状態になり、入力が終了すると(Enterが押されると)文字列を返す。

＜演算子＞
-x
　　符号を反転
x + y
　　整数加算
x - y
　　整数減算
x * y
　　整数乗算
x / y
　　整数除算
x % y
　　整数剰余
x = y
　　２つの整数値が等しい場合に１、そうでないときに０
x != y
　　２つの整数値が等しい場合に０、そうでないときに１
x < y
　　xがyより小さい場合に１、そうでないときに０
x <= y

　　ｘがｙより小さいか等しい場合に１、そうでないときに０
x > y
　　ｘがｙより多きい場合に１、そうでないときに０
x >= y
　　ｘがｙより多きいか等しい場合に１、そうでないときに０
x AND y
　　ｘとｙの値のビットごとの AND（論理積でないことに注意）
x OR y
　　ｘとｙの値のビットごとの OR
x XOR y
　　ｘとｙの値のビットごとの XOR
x$ + y$
　　文字列の連結

なお、整数演算子の優先順位は、優先度が高いものから以下の順です。

+ -（単項演算子）
* / %
+ -（加算・減算）
< <= > >=
= !=
XOR
AND
OR

＜MUSIC＞
MUSIC 命令では、BGM 用のデーターを文字列で指定します。文字列の書式は、ABC notation に準拠しています。ただし、すべての記法が使えるわけではありません。
なお、キーや速度などのデフォルト設定値は以下の通りです。

Q: 1/4=90
L: 1/8
K: C

BGM 演奏時に一度に設定できる音の数は、31 迄です。これを超えて音楽を再生したい場合は、MUSIC()関数の戻り値を調べ、その値が十分小さくなってから、次の MUSIC 命令を実行するようにします。

添付の music.bas に、使い方に関するサンプルがありますので、参考にして下さい。

＜SOUND＞
SOUND 命令では、DATA 列のデーターを、行番号もしくはラベルで指定します。SOUND 命令による効果音再生中は、BGM は再生されません。また、前の効果音が終わる前に次の SOUND 命令を実行すると、前の効果音の再生は停止し、新しい効果音がすぐに再生されます。

DATA 列では、３２ビット整数値として、交換音を表現します。この整数値の下位１６

ビットは周波数の指定です。2048 が 440Hz(ラの音)に対応します。値が大きくなるほど、より低い音が出ます。上位１６ビットは、音の長さです。1 が、1/60 秒に相当します。最後に、65535 以下の値で、効果音の繰り返し回数を指定します。これらのデーターの数は、32 を超えないようにして下さい。

添付の sound.bas に、使い方に関するサンプルがありますので、参考にして下さい。

＜PCG＞
PCG(Programmable Character Generator)を用いると、ASCII コードごとにフォントを指定して、疑似グラフィックスとして表示させることが出来ます。使用する場合は、まず

USEPCG

とします。フォントの変更は、PCG ステートメントを用いて、

PCG 0x80,0x80402010,0x08040201

の様に設定します。この例では、ASCII コード 0x80 の文字のフォントを設定していて、バックスラッシュの様な記号(左上から右下に向かう斜め線)が表示されるようになります。PCG の利用を停止し、オリジナルのフォントに戻す場合は、

USEPCG 0

とします。再度 PCG を使用したい場合は、

USEPCG

として下さい。先に設定したフォントデーターが、復活します。なお、先に設定したフォントデーターを破棄して PCG の使用を始めたい場合は、

USEPCG 2

として下さい。

＜ヒント＞
FOR 文では、TO ステートメントの次に書かれた値に合致する場合（超えた時ではなく）に、ループから抜ける仕様です。また、FOR-NEXT ループの途中で、GOTO 文でループの外に飛んだり、RETURN 文を実行したりすると、予期せぬ結果（機器のリセット等）を引き起こします。ただし、GOSUB 文でサブルーチンを呼んだり、別の FOR-NEXT をネストして使う事は可能です。

ON GOTO 分や ON GOSUB 文はサポートしていません。ただし、例えば次のように記述することで、同様の動作をさせることは可能です。

```
 GOSUB 10000+A
 ....
 10000 PRINT "A=0" : RETURN
```

```
10001 PRINT "A=1" : RETURN
10002 PRINT "A=2" : RETURN
```

一行中で連続して文字列を扱うと、"String too complexed"というエラーがでて、停止することがあります。この場合は、文字列を扱う命令を独立した行するか、文字列関連の演算を幾つかのステップに分けて、それぞれ１行ずつの記述にして試してみて下さい。

＜バージョン履歴＞

・KM-1121 2016年７月公開。
  １．プログラム実行中のキーボード読み取りが欠落する場合がある問題を修正。
  ２．乱数のアルゴリズムを修正。
  ３．最終行に改行がなくても実行可能とする修正。
  ４．DATA文のRESTORE初期化の問題を修正。

・KM-1120 2016年５月バグ修正版。
  １．INPUT$()関数呼び出し時に、キー入力バッファが空になるようにした。
  ２．文字列操作時に、特殊なケースに於いて内容が変更されてしまう不具合を解消。
  ３．RESTOREが正常に行なわれない場合があるバグを修正。

・KM-1120 2016年４月バグ修正版。
  １．数値$80000000の表示修正

・KM-1120 2016年2月公開。
  １．PCG機能を追加。
  ２．SCROLL命令を追加。
  ３．WAIT命令を追加。
  ４．「Ctrl+Break」キーによる実行停止に対応。
  ５．FOR無しでNEXTを実行した場合、GOSUB無しでRETURNを実行した場合にエラーを表示して停止するようにした。

・Ver 1.1.0 (KM-1110) 2015年12月公開。
  １．2015年11月21日に変更されたPIC32TVGSの仕様に対応。
  ２．PS/2キーボードに対応。
  ３．INKEY() INPUT$() VAL() DEC$() の４つの関数を追加。
  ４．TVRAM() ASC() PEEK()が、0x80-0xFFの値に関して負の数を返していた不具合を修正。
  ５．DIMステートメントにより配列を定義した際、すべての要素がゼロになるようにした。
  ６．単項演算子、「-」「+」を追加。
  ７．LABEL定義されていない飛び先にGOTOするとリセットされる不具合を修正。
  ８．同一のLABELを複数回使用している場合に、コンパイルエラーが出るように修正。
  ９．引数を持たないPRINT文に対応。

・Ver 1.0.5 (KM-1100) 最初の正規公開バージョン。