

【パソコン特集】

パソコン入出力支援ツール

Application of Personal Computer to Input and Output Tool System

北 島 彰 夫 *
Akio KITAJIMA

1. まえがき

最近、パーソナルコンピュータ（略称、パソコン）が急速に普及しているが、これはパソコンの16ビット化にともない漢字処理が容易にできるようになったこと、安価な割には性能が向上したこと、入力の場合などでその優れた対話処理能力を活用できること等によるものである。またパソコン用のアプリケーションプログラムが普及し始めており、コンピュータに素人の人でも、実業務をこなせるといったことも、その普及に拍車をかけていると思われる。

パソコンの32ビット化、強力なマイクロプロセッサの出現といったものが直前に控えており、現在の大型機なみの性能を有するパソコンの出現も近いと予測されている。これからパソコンが我々の身近な道具として、生活の一部に組み込まれていくのは明らかである。

一方、パソコンのプログラム開発に携っている人達にとっては、パソコンが優れた対話処理能力を有しているが故に、パソコン画面への入出力を行なうプログラムを作成するのに意外に苦労している。一概にはいえないものの、パソコンの入出力プログラムを作成するのに高度の知識は unnecessary ですが、時間と労力がかかっているのが実状であり、そのプログラム開発の合理化をはかり、創造性あるいは吟味検討が必要な処理ルーチンの開発等多くの時間を振り向けられないか。このような考えから生まれたのが、「パソコン入出力支援ツール・IOTOOL (INPUT, OUTPUT TOOL)」である。IOTOOLは入出力プログラムを自動生成するプログラムジェネレータであり、以下、その概要を紹介する。

2. IOTOOLの機能

今までの入出力プログラムはパソコン画面をみたてた25行×80カラム（パソコンにより異なることがある）のグラフ用紙に、文字、表枠、図形等の配置を計画し、PRINT文、INPUT文、あるいはグラフィック命令文等を用いてプログラムコーディングを行ない、そのプロ

グラムをパソコンに入力して、デバッグを行なうという手順で作られている。コーディングを行なう際には、LOCATE文に用いる行、カラム番号の抽出、グラフィック文に用いる座標値の抽出といったことに注意を払う必要があり、また入力文については入力者が入力ミスをするとの前提に立って、入力データの修正ルーチンも組込むといったことにも配慮しなければならない。

IOTOOLはこのような複雑なコーディングをいっさい省略して、パソコン自身でプログラムを作ろうとしたもので、その開発にあたって基本的にめざした点は以下のものである。

- ① IOTOOLの管理化のもとに会話処理により、入出力プログラムで実現させたい画面を直接作る。会話処理により画面に作るものは文字、表枠、図形、入出力変数（数値型、文字型）の表示位置と、そのエリア（表示桁数と小数点位置、文字スペース等）とする。
- ② 画面上に定義した文字、図形、変数エリアは変更、修正、削除、行換え、移動等が自由に行なえる。
- ③ 画面作成が終了したら、その画面を実現できるBASICSプログラムを自動生成（ジェネレート）する。作られたプログラムには入力修正ルーチン、入力データのファイル保存、読み出し等ができる基本ルーチンをすべて組込むことができる。
- ④ ジェネレートされたプログラムをIOTOOLに再び呼び込み、追加、修正、削除、入力順の変更を行ない、新たなプログラムが作られる。
- ⑤ 作成したプログラムは従来、人手によりコーディング作成されたものと何等変わらず、直接、人手により手直しすることもできる。

3. プログラム構成

IOTOOLは図-1に示すルーチン群によって構成されている。

(1) 基本入力ルーチン

新規作成、あるいは修正するプログラムの収容ファイ

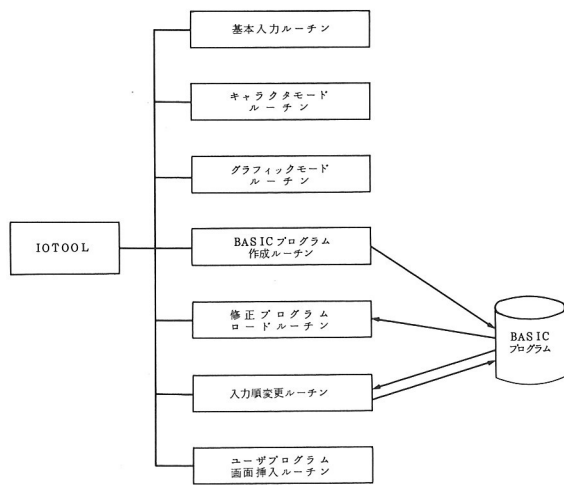


図-1 プログラム構成

ル名，作成するプログラムの開始文番号とそのピッチ，あるいは作成するプログラムの命令型式等を指定入力するルーチンである。

(2) キャラクタモードルーチン

パソコンキーの↑↓←→等のキーを用いてカーソルをパソコン画面内で自由に移動させ，好きな位置に文字，入出力変数のフィールドを割りあて，変数名を定義するルーチンである。また消去，挿入，行換えができるエディット機能も備っている（4-1節参照）。

(3) グラフィックモードルーチン

直線，円，円弧，寸法線，グラフィック文字等を画面の好みの位置に定義するルーチンである。グラフィックコマンドが選択されると画面左上スミに小さな円のマークが現われ，Ten Keyを用いてこのマークを8方向に自由に移動させることができ，その移動速度も速い，遅いの2者をキー操作で選択できる。マークをコマンドに応じて画面上にセットすることにより，所定の図形を定義できる。また一度定義した図形を消去したり，画面の一部を拡大してその拡大状態で図形を定義するといった機能も有している（4-2節参照）。

(4) BASIC プログラム作成ルーチン

画面上に作られた文字，図形状態を解析して，プログラムをジェネレートするルーチンである。作成されたプログラムには入力データ修正ルーチン，入力データのファイル保存，読み出しルーチンを組込むことができる。

(5) 修正プログラムのロードルーチン

(4)で作成したプログラムを修正するために，そのプロ

グラムのロードを行ない，その命令を解釈してIOTOOL下の文字，図形状態に再現するルーチンである。再現された画面状態は今まで会話操作によって作ってきたものであるとみなして，そこから追加，修正，削除といった作業に入ることができる。

(6) 入力順変更ルーチン

(4)で作成されたプログラムの入力命令に関する順序は画面位置において上から下へ，左から右へ定義された入力変数の順序に従う。図形をみながら入力したいようなプログラムの場合，データ入力順に工学的な意味合いをもたせて入力したいといった要求が出てくるが，このような場合，本ルーチンを利用すると，入力命令の順序を好みの順番に変更することができる。

(7) ユーザプログラムの画面挿入ルーチン

1つの画面を複数のプログラムで実現させたい（例えば画面上の一部図形を取り換えながら画面表示したい）場合，プログラム作成単位を分ける必要がある。既作成済みのプログラム（ユーザプログラム）を参照図形として画面に表示し，この図形に重ねて新たな文字，図形等を定義できる。ただし参照図形はプログラム作成の対象とはならない。

4. コマンドの種類

コマンドの種類はキャラクタモードで9，グラフィックモードで22ある。画面の25行目をガイダンスエリアにとり，このエリアにコマンドの種類がエンドレスに表示されるので，そのコマンド番号を選択し，キー入力することにより所定のコマンドを選択できる。

4-1 キャラクタモードにおけるコマンド

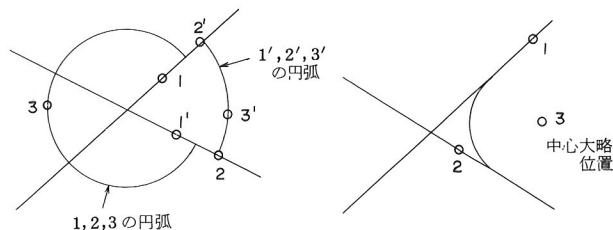
- ① 文字プリント・・・画面上の任意位置に文字（英数字，カナ文字，英記号，漢字）を定義する。漢字を定義する場合はカナ漢字変換機能を用いる。
- ② 変数プリント・・・出力変数名の定義と，そのフィールド指定を行なう。数値型変数の場合は###・##，文字型変数の場合は&&&&のように，その画面表示位置に#，&等の英記号を定義する。
- ③ 変数インプット・・・入力変数名の定義と，そのフィールド指定を行なう。フィールドの指定は②と同じである。
- ④ エディット・・・①で設定した文字，②，③で設定した変数フィールドの消去，挿入，行換え，文字新

規配置等が自由にできる。

- ⑤ 変数名変更・・・カーソルを#あるいは&の定義位置に合せることにより、ユーザが定義した変数名をガイダンスエリアに表示し、その変数名変更、入出力変数の交換等ができる。
- ⑥ 同一文字・・・画面上に同じ文字列をいくつか定義したい場合、このコマンドを用いる。カナ漢字変換を何度も行なうよりも効率的に同一文字列が定義できる。
- ⑦ カラム表示・・・24行×80カラムの枠表示、消去をコマンド選択ごとに繰返す。
- ⑧ 作 表・・・セミグラフィックコードを用いて、画面任意位置に任意の大きさの表(縦、横線からなる)を効率よく定義できる。
- ⑨ シフト・・・1行分の画面スクロールを行なう。25行目に文字等を定義したい場合に用いる(25行目をガイダンスエリアとして使用しているため)。

4-2 グラフィックモードにおけるコマンド

- ① 直線・・・3章の(3)で説明したマーカーを画面上に2点セットすることにより、マーカーを結ぶ直線が定義される。
- ② 連続直線・・・マーカーの連続セットによる。
- ③ 四角形・・・対角の2点セットによる。
- ④ 円・・・円中心と半径上の1点セット。
- ⑤ グラフィック文字・・・文字描画原点、描画方向、横縦倍率により文字の大きさを指定し、描画文字列をキー入力する。
- ⑥ 作 表・・・グラフィック命令(LINE命令)での作表ができる。作表位置はカーソル移動により定める。
- ⑦ 寸法線・・・水平、鉛直寸法線のみ。寸法引出線の足の部分2点と、寸法位置を定める1点の合計3点のマーカーセットによる。引出線も同時に定義される。
- ⑧ 円 弧・・・2直線にはさまれる円弧、2直線に接する円弧等、3種類の円弧定義が可能。マーカーセットの要領を図-2に示す。
- ⑨ 消 去・・・①～⑧(⑥は除く)で定義した図形の消去。消去したい図形上にマーカーセットを行なう。マーカー位置に2つの図形が重なった場合、後で定義されたものが消去される。
- ⑩ 拡 大・・・対角の2点をマーカーセットするこ



(注) 番号はセットの順番を示す。

図-2 円弧のマーカーセット法

とによりできる四角形内に入っている図形を画面一杯に表示する。この状態で①～⑨(⑥を除く)のコマンドを選択できる。キャラクタモードで定義した文字等は拡大されない。

- ⑪ 原図復帰・・・拡大モードの解消を行なう。
- ⑫ 再表示・・・マーカーの走行によりつぶれた図形をきれいに再表示する(マーカーのオン、オフを行なうことによってマーカーにかかった図形の一部が消されるため)。
- ⑬ 線種設定・・・実線、点線、1点鎖線、2点鎖線の設定。現在設定されている線種は画面上右スミに常に表示されている。
- ⑭ カラー設定・・・図形の設定色、ペイントをする際の領域色の2色の設定。現在設定されている2色は画面上スミに短形表示されている(カラーの場合)。
- ⑮ ペイント・・・領域内の一点のマーカーセットを行なうことにより、⑭で設定した領域色で、図形設定色の境界内をペイントする。
- ⑯ 図形移動・・・対角の2点を与えることによりできる四角形の中に定義された図形を任意位置に平行移動する。
- ⑰ 図形コピー・・・対角の2点を与えることによりできる四角形の中に定義された図形を任意位置にコピーする(同じ図形を同一画面上に複数定義したい場合に用いる)。
- ⑱ 文字マーカー・・・キャラクタモードにおける1文字分の文字の大きさの短形マークを画面任意位置に設定できる。拡大時にこのマーカーも拡大表示されるので、文字位置との関係をみながら図形定義が可能。作成されるプログラムとは無関係である。
- ⑲ カラム表示・・・24行×80カラムの枠の表示、消去。拡大モード時にも、拡大されて表示される。

- ② 点表示・・・・・・定義された直線の両端, 円の中心, 円弧の中心と始終点には点が必ず定義されるようになっている。このコマンドを選択すると, このような点の表示, 消去が交互に繰返される。点はマーカーと同じ大きさの円として表示される。
- ③ 厳密直線・・・・点表示状態にして, 表示点にマーカーをセットすると, その2点を結ぶ厳密な直線が定義できる。これによりペイントを行なう場合の色モレ等を防止できる。
- ④ フィレット・・・・定義直線の一部を削除する。円弧定義等により不要となった直線の一部を削除する。

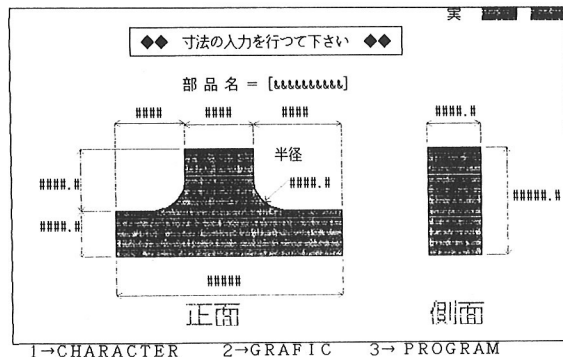


図-3 プログラム作成直前の画面例

5. 作成例

図-3はプログラム作成直前の IOTOOLによってつくった画面のハードコピーの一例である。

図-4はその作成されたプログラム, 図-5はこのプログラムを実行することによって得られた画面のハードコピーである。プログラムは文番号100から始まって10ステップごとにつくられ, プログラムが長くなるので省略して示している。文番号の末尾に5がついているものは, 作られたプログラムに後で追加を行なった命令である。

図-3に示した画面に入力変数が定義されている場合, 点線, 1点鎖線, 2点鎖線の直線, 円, 円弧等が定義されている場合, 寸法線が定義されている場合には, でき上がったプログラムを実行する際に, 専用の外部サブルーチンと一緒に実行しなければならない。

文番号240のサブルーチン65400は寸法線を描画する機能を有している。文番号710サブルーチン65000は入力データの汎用的処理を行なう機能を有し

```

100  "SAMPLE PROGRAM
110  CLS
120  VIEW (0,0)-(639,399):WINDOW (0,0)-(639,459)
130  REM "GRAPHIC START
140  LINE (120,320)-(120,286),1
150  LINE (200,186)-(280,186),1
160  LINE (380,266)-(380,326),1
170  LINE (160,226)-(320,226),1
180  LINE (120,286)-(160,286),1
190  LINE (280,186)-(320,226),1
200  LINE (320,286)-(380,286),1
210  LINE (226,254)-(320,254),1
220  LINE (320,254)-(322,1,224,1),5
230  LINE (222,1,254,1)-(232,1,248,1),5
240  LINE (292,1,254,2)-(298,1,254,2),5
250  LINE (292,1,254,2)-(298,1,254,2),5
260  SYMBOL (200,386), "正面",2,2,5,0
270  SYMBOL (480,386), "側面",2,2,5,0
280  REM "GRAPHIC END
290  NDATF=0:NDATR=0:PAGS="1":NLOOP=2:MODETP(1)=1:MODETP(2)=3
300  FOR SINDEX=1 TO NLOOP
310  MODE=MODETP(SINDEX)
320  ON MODE GOTO 640,540,710,710,710
330  LOCATE 18,2:PRINT "◆◆ 寸法の入力を行って下さい ◆◆"
340  LOCATE 18,3:PRINT "部品名=[ ]"
350  LOCATE 18,4:PRINT "部品名=[ ]"
360  LOCATE 26,6:PRINT "部品名=[ ]"
370  MTP=2:MOJISU=10:CY=6:IF MODE=1 THEN GOSUB 65000:NAMES=MOJIS ELSE MOJ
380  IS=NAMES:GOSUB 65000:NAMES=MOJIS
390  MTP=1:MOJISU=4:KETSU=0:CY=8:IF MODE=1 THEN GOSUB 65000:LH1=SUJ1 ELSE
400  SUJ1=LH1:GOSUB 65000:LH1=SUJ1
410  MTP=1:MOJISU=4:KETSU=0:CY=8:IF MODE=1 THEN GOSUB 65000:LH2=SUJ1 ELSE
420  SUJ1=LH2:GOSUB 65000:LH2=SUJ1
430  MTP=1:MOJISU=4:KETSU=0:CY=40:IF MODE=1 THEN GOSUB 65000:LH3=SUJ1 ELSE
440  SUJ1=LH3:GOSUB 65000:LH3=SUJ1
450  TLV=LH1+LV2
460  LOCATE 73,14:PRINT USING "####.#":TLV
470  TLH=LH1+LH2+LH3
480  LOCATE 29,20:PRINT USING "####.#":TLH
490  IF MODE=1 OR MODE=3 THEN 830
500  IF MODE=2 AND SINDEX=NLOOP AND MODETP(SINDEX)=3 THEN 890 ELSE 880
510  LOCATE 15,25:PRINT "データ修正しますか ? (1→いいえ 2→はい) [1]"
520  LOCATE 70,25:INPUT "OKSW"
530  IF OKSW=1 OR OKSW=2 THEN 830
540  IF OKSW=1 THEN 880 ELSE 870
550  IF MODE=1 OR MODE=2 THEN SINDEX=SINDEX+1:GOTO 880 ELSE 880
560  IF MODE=1 OR MODE=2 THEN 880 ELSE 630
570  NEXT SINDEX
580  RETURN

```

図-4 作成されたBASICプログラム

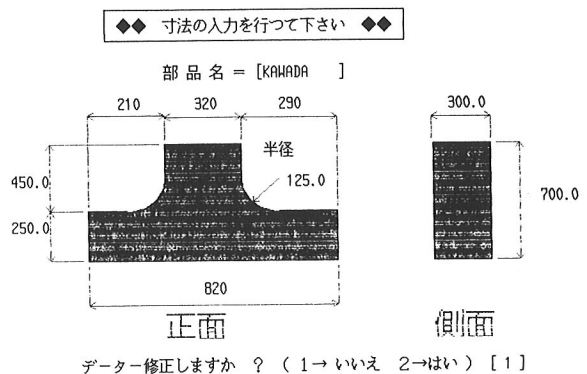


図-5 作成プログラムの実行による画面

ている。この他に点線表示等に係わるサブルーチンが用意されている。

文番号600の命令は IOTOOL の標準形としてつくられ, この命令を変更することによって, 入力データの処理手順を変えることができる。文番号620で設定しているMODEは1が設定されるとキー入力・入力データ画面表示がサブ65000の中で行なわれる。2の場合, 入力データの単独表示, 3の場合入力データの修正, 4の場合入力データのファイル保存(インデックス・クエンシャルファイルを用いている)。5の場合

合入力データのファイル読み出しが、それぞれサブルーチン 65000 中に行なえるようになっている。

標準型の場合はキー入力・データ表示、修正であるが、NLOOP=3としてMODETPを1, 3, 4とすればデータのファイル保存まで、NLOOP=4でMODETPを5, 2, 3, 4とすればデータのファイル読み出し、表示、修正、ファイル保存といった手順でデータの処理ができる。

作成されたプログラムにはデータ修正に関するガイダンス命令も自動的に組込まれている。

6. あとがき

IOTOOLを用いれば定型的な画面のものであれば、1/3から1/5程度にプログラム作成時間が短縮される。これはIOTOOLで作った画面はプログラムになっ

た場合も必ずその画面が保障され、コーディング、デバックといった作業が一切、不必要だからである。しかしながら画面入出力プログラムは必ずしも定型的なものばかりではなく、分岐文、繰返し文等を用いたい場合も多く、このような場合、IOTOOLの部分使用を行なっていくのが特策と考えられる。

IOTOOLはプログラム開発の合理化、標準化に大きく貢献すると考えられ、また最近、大型機用のデータエントリーとしてパソコンを利用する傾向が強くなっていることから、この種のプログラムの社会的必要性は今後、増々強まることが予想される。

IOTOOLはユニパックのパソコンUPIO-E用に開発したもので、使用言語はすべてBASICである。今後、他機種にも使用できるようにその移植を計っていきたいと考えている。