TR-82-019

繪圖系統之介面常式
使用手冊

中央研究院

資訊科學研究所

電子計算機中心

劉興祥

中華民國 七十一 年 十一 月 十八 日

致　謝　詞

本手册能順利完成，得助於鄭國揚老師的指導，計算機中心工作人員：邱淑美、李昌芹的協助，陳瓊音、方鳳梅的幫忙打字，以及邱顯華的製版油印及裝訂，在此，我衷心地感謝他（她）們。

目　　錄

(五)輸出裝置的選擇(Output device selection)

```
CALL  YWDV (NDEV) ! WRITE_DEVICE
NDEV : 'PLotter'
   or  'RAmtek'
   or  'TEktronix'  (default)
   or  'KEYboard'
```

(六)儲存圖形

```
CALL  YPUSH  (FILENAME)
```

其中SCGS會取FILENAME 中所放的字，直到不爲'a'-'z','A'-'Z','0'-'9' 及'.','-',';' 等字元才停止，然後將往後的圖形儲存在此檔案裡。

(七)關閉儲存圖形的檔案

```
CALL  YCLOSE  (FILENAME)
```

(八)取用儲存於檔案中之圖形

```
CALL  YPOP  (FILENAME)
```

緒　言

本使用手册乃針對使用本中心 PDP-11/70 機型的電腦繪圖工作者所編寫的。本中心所提供之電腦繪圖之週邊裝置如下：

(1) CALCOMP 600 Digitizer （數位轉化器）

(2) CALCOMP 960 Plotter （繪圖機）

(3) TEKTRONIX 4054, 4010 graphic terminal （圖形終端機）

(4) RAMTEK 9300 color graphic terminal （彩色圖形終端機）

SCGS 即爲上列設備之介面常式，對於初學者，使用起來非常方便。

SCGS之英文全名爲 SIMPLE CORE GRAPHICS SYSTEM 乃由 CORE GRAPHICS SYSTEM （簡稱 CGS）抽取常用之程式，滙集而成一套電腦繪圖設備之介面程式，其所使用之電腦語言爲 FORTRAN 。

倘使你所需求之繪圖功能，超乎 SCGS所提供者，可進一步以 CGS 來達成其他更複雜的電腦製圖。

一、控制常式 (Control Routine)

㈠起始常式

```
CALL  YINI  ! INITIALIZATION
```

此常式用來定一些初值，應出現於其他常式之前，如果在呼叫 YEXIT 後還想畫另一圖時，仍須再呼叫此常式，呼叫後 CP (current point) 會歸於原點。

㈡終止常式

```
CALL  YEXIT  !  TERMINATION
```

完成各種繪圖功能之後，應使用此常式，結束一切動作。

㈢偵錯常式

```
CALL  YCTB (LCTB)  !  CONTROL_TRACEBACK
```

此常式用來幫助程式之偵錯，當 LCTB='YES' 時會將錯誤產生的路徑（主、副程式名稱）列出以便於查出錯誤的所在，當 LCTB= 'NO' 時，則不印出，機定值 (Default) 爲 'NO'。

㈣輸入裝置的選擇 (Input device Selection)

```
CALL  YGDV(NDEV)  ! GET_DEVICE
```

```
NDEV : 'KEyboard'
   or  'TEktronix'
   or  'RAmtek'
   or  'DIgitizer'   (default)
```

裝置的名稱，可僅標示出前面 2 個字母。

二、輸入常式

㈠輸入點座標

```
CALL  YGPT (X, Y, IFLAG)  ! GET_POINT_2
```

須要由輸入裝置（如上述）輸入點座標時，可呼叫此常式。以接收 X,Y 座標，且機器自動會傳回按鍵之值 (IFLAG), 可忽略不用，或當做自己程式中的指標。

（註：IFLAG 爲一Byte 的變數）

㈡ RAMTEK 輸入游標的選擇

```
CALL  YSRCS (IRCS)  !  SET_CURSOR_INDEX
```

IRCS = 1（選用游標 1 號）

0（選用游標 0 號，default）

三、輸出常式

㈠顯示出 window 之方框

CALL YWWD ! WRITE_WINDOW

此時 CP (Current Point) 不變。

㈡移動 CP

CALL YWMV (X,Y) ! MOVE_ABS_2

將CP 移動到點 (X,Y)，座標值 X,Y 爲實變數。

㈢畫直線

CALL YWLN (X,Y) ! LINE_ABS_2

由CP 畫一直線到點(X,Y)，而直線的形態可由另一常式 YSLS 來決定爲實線或虛線；此常式執行後CP成爲 (X,Y) 。

㈣畫圓弧

CALL YWCC (CENTER, P_0, DEGREE) ! ARC_ABS_2

其中變數型態爲 REAL CENTER(2),P_0(2);DEGREE 爲一實變數（度度量）。

此常式以 CENTER座標爲圓心，P_0 爲起點，繞著圓心走DEGREE° 畫一圓弧，弧的型態可由常式 YSCCS 來決定爲實線弧或虛線弧；此常式執行後CP會移到圓弧的末端處。

㈤畫多邊線

CALL YWPLN(V_x, V_y, N_v) ! POLYLINE_ABS_2

畫出多邊線，如下圖：

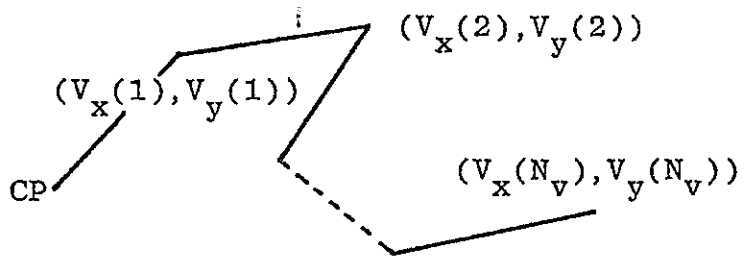


圖 3-1 多邊線

其中 V_x, V_y 均爲一維的實變數列陣，N_v 爲正整數表多邊線的邊數。

此常式執行完後，CP 變爲 $(V_x(N_v), V_y(N_v))$，起點與終點並不做連線。

(六)畫多邊形

CALL YWPG (V_x, V_y, N_v) ! POLYGON_ABS_2

畫出多邊形如下圖

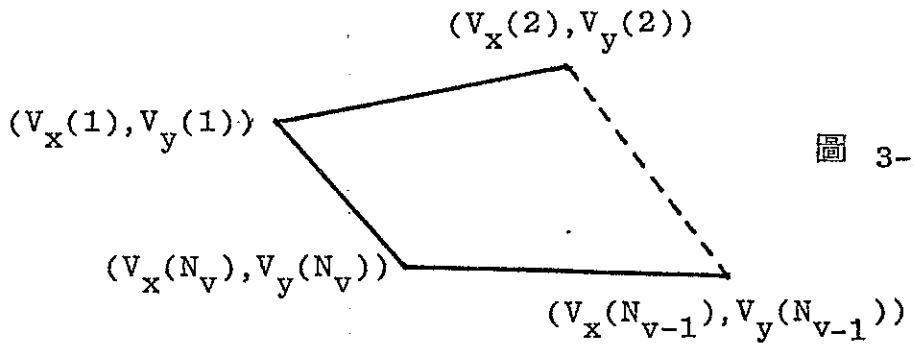


圖 3-2 多邊形

此常式執行後，CP 成爲 $(V_x(1), V_y(1))$ 起點與終點會做成連線。又若輸出裝置爲 RAMTEK 時，會將其內部塗上由常式 YSPGC 所指定的顏色（連邊線亦然）。

(七)清除畫面

CALL YWER ! ERASE

(八)輸出字元串 (String)

CALL YWTEXT (STRING, N_c) ! WRITE_TEXT

由CP 開始畫出一共有N_c 個字元的字元串。該字元串之大小與方向，可由另一常式 YSTEXT 來決定，執行後，CP 保持不變。

(九)輸出一特殊符號（由 YPMARK指定者）

CALL YWMARK ! MARKER_ABS_2

以CP 爲中心，輸出一已由常式 YPMARK 選擇之特殊符號，CP 不變。

(十) 輸出一串特殊符號

CALL YMARKS (V_x,V_y,N_v) ! POLYMARKER_ABS_2

先由常式YPMARK 定一特殊符號後，呼叫此常式即可在 (V_x(1), V_y(1)), (V_x(2),V_y(2)),....(V_x(N_v),V_y(N_v)) 各點座標上輸出該已定特殊符號。執行後，CP保持不變。

(十一) 輸出一圓區域

CALL YFCC (CENTER,RADIUS) ! FILL_CIRCCE

其中變數型態爲 REAL CENTER (2), RADIUS 分別表圓心及半徑，CP 保持不變。

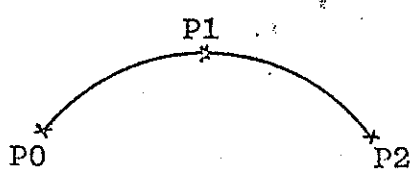
(十二) 利用弧上三點畫出一圓弧

CALL YWCC3 (P0,P1,P2)

其中 P0,P1,P2 之變數形態爲

REAL P0(2),P1(2),P2(2)

此常式會畫出如下之一圓弧



四、有關畫面控制 (View Operation) 的常式

㈠定圖形之投影框 (View port)

CALL YDVP (Vxmin, Vxmax, Vymin, Vymax) ! DEFINE_VIEW_POR

㊀此常式用以定出其圖形的顯示位置。

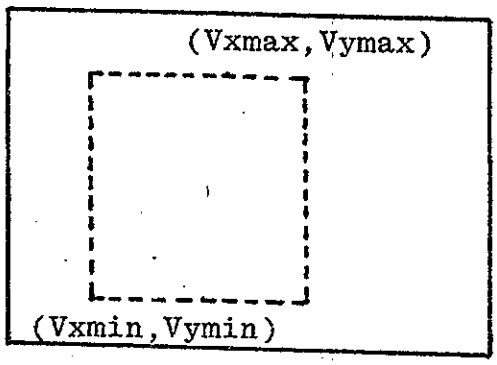


圖 4-1 投影框

Normalized device coordinate

㊁此處 argument 之值採用的是標準化的裝置座標一般的範圍爲

$0 \leq X \leq 1.25$, $0 \leq y \leq 1$

但系統並不做此檢查（理由：一般吾人使用繪圖機時，常僅使用其中的一小部分，故若加以限制，會使得繪圖機的使用格外不方便）。

㊂若以繪圖機爲輸出裝置，在上述座標範圍所得畫面大小與 TEKTRONIX 4054 相同。

㈡定圖形之透視窗 (Window)

CALL YDWD (Wxmin,Wxmax,Wymin,Wymax) ! DEFINE_WINDOW

見下圖，設 A 爲全部圖形所成之實物座標系，若欲顯示出其中某範圍下之部分圖形，可利用此常式定出 WINDOW.

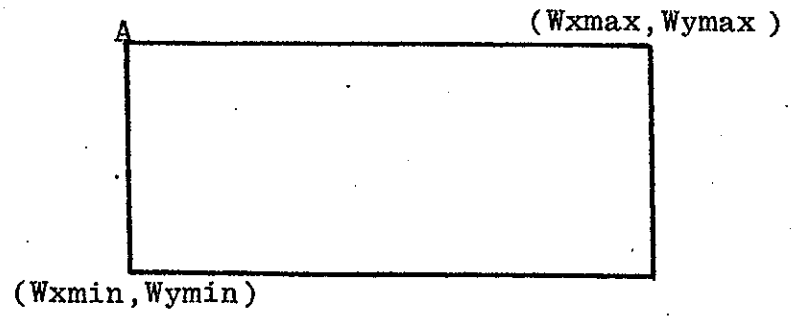


圖 4-2 透視窗 (Window)

此系統會將WINDOW 內的畫對應至 VIEWPORT 上再畫出來，

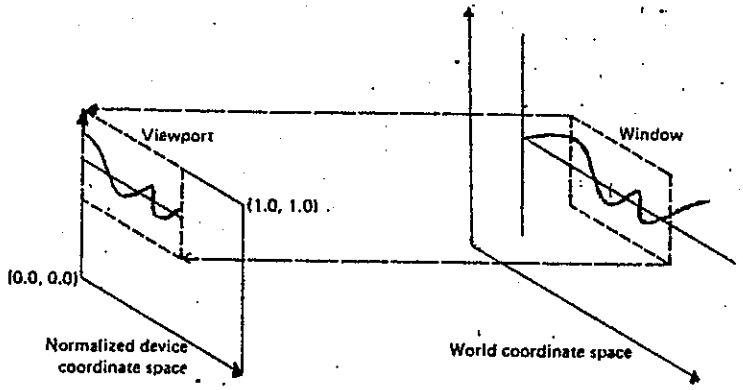
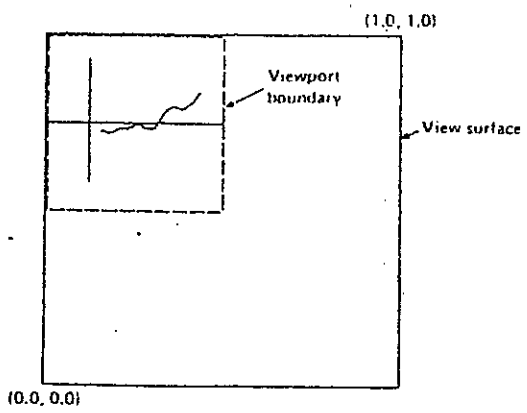


圖 4-3 透視窗與投影框的關係

註： Default :

$(V_{xmin} \quad V_{xmax} \quad V_{ymin} \quad V_{ymax}) := (0\ 1\ 0\ 1)$

$(W_{xmin} \quad W_{xmax} \quad W_{ymin} \quad W_{ymax}) := (0\ 1\ 0\ 1)$



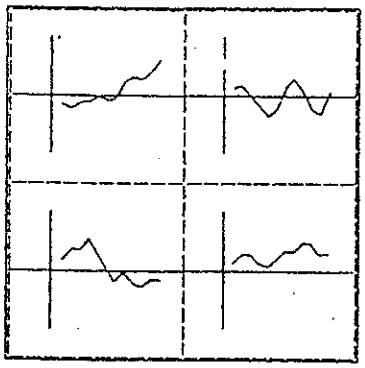


圖 4-4 投影面上，左上角的投影框　圖 4-5 四個投影框的實例圖

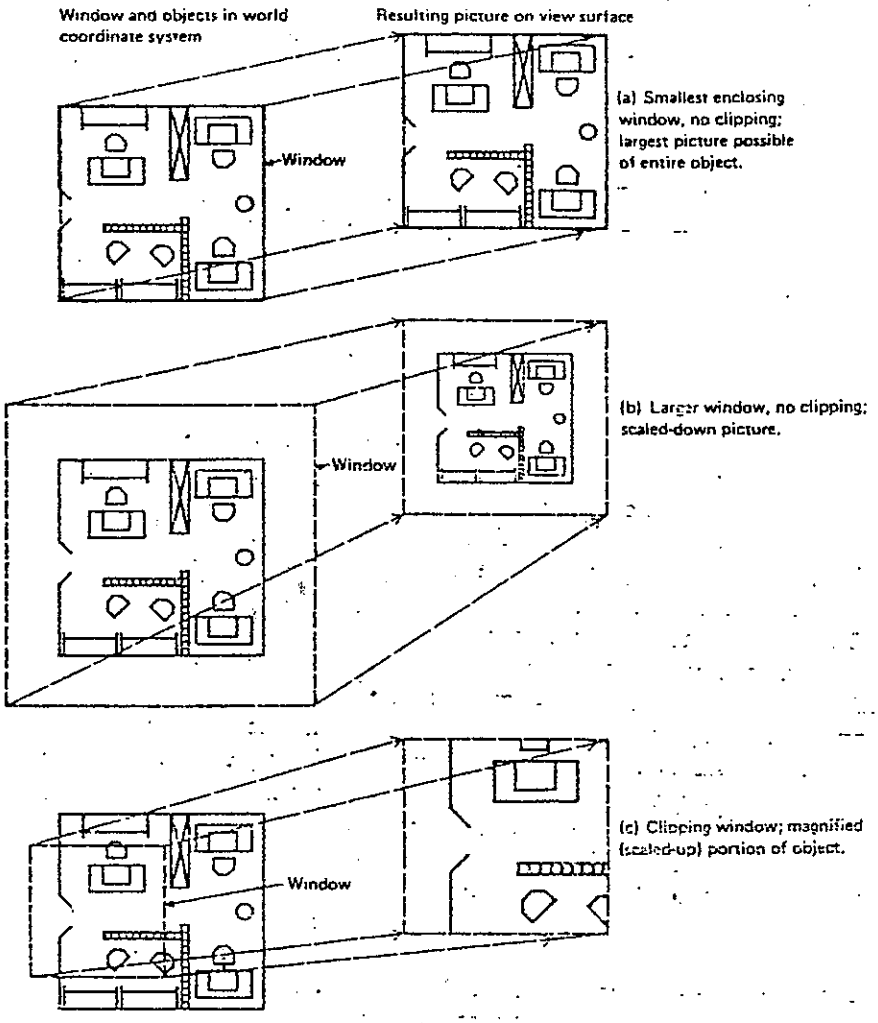
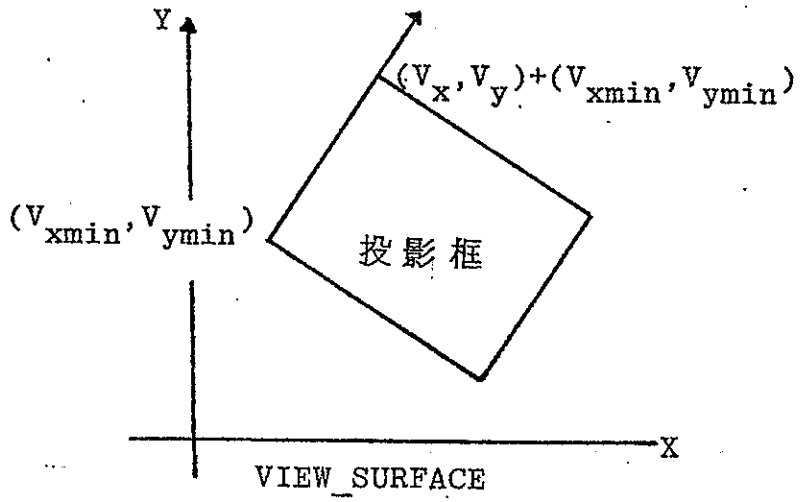


圖 4-6 實物座標、透視窗、投影面及投影框等之關係

㈢定出投影框的新Y 軸，因而使得投影框在投影面上，能隨意旋轉。

CALL YSVUP(V_x,V_y) ! SET_VIEW_PORT_UP_2

新的投影框位置如下

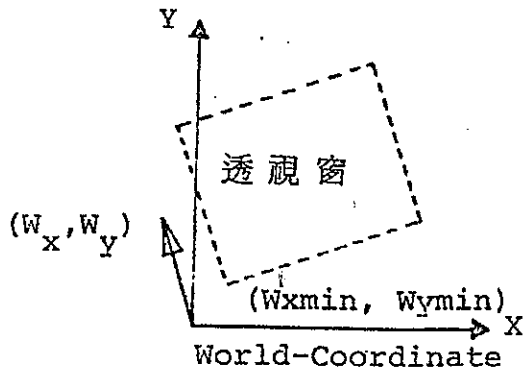


Default: (V_x,V_y) := (0,1)

㈣定出透視窗的新Y 軸，因而使得透視窗得以隨意旋轉

CALL YSWUP (W_x,W_y) ! SET_VIEW_UP_2

新的透視窗位置如下：



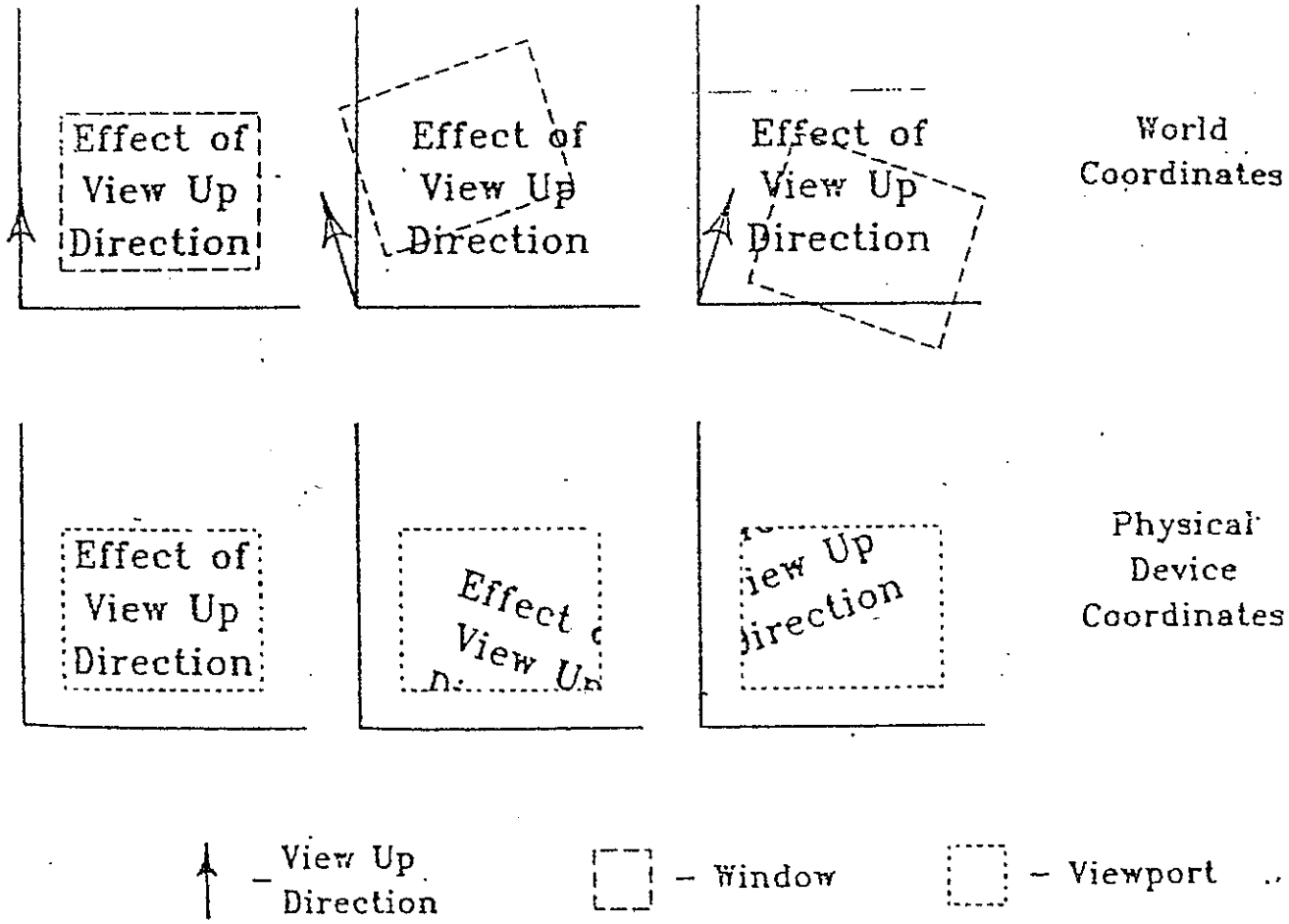


圖 4-7 : Effect of View Up Direction

五、屬性 (Attributes) 常式

㈠線的形態

```
CALL  YSLNS (LS)  !  SET_LINE_STYLE
```

LS : 'DAshed'　　——虛線

or 'SOlid'　　——實線 (default)

㈡弧的形態

```
CALL  YSCCS (LS) !  SET_ARC_STYLE
```

LS : 'DAshed'　　——虛線弧

or 'SOlid'　　——實線弧 (default)

㈢弧的精密度

```
CALL  YSCCP (PL)  !  SET_ARC_PRECISION
```

PL為一大於 0 之實數表構成弧的線段長，可由常式 YSPLC 定其所指的長度單位乃屬於World coordinate 或 Normalized device coordinate.

PL 的機定值為 0.01。

㈣虛線的精密度

```
CALL  YSDLP (PL) !  SET_DASHED_LINE_PRECISION
```

PL表構成虛線之線段長度，與上述同。

㈤定精密度之座標系

```
CALL  YSPLC (NC) !  SET_PRECISION_COORDINATE
```

NC : 'WOrld'　　(以實物座標系為準)

or 'NDc'　　(以輸出裝置之標準化座標系為準)

(六)線的顏色

CALL YSLNC (LLNC) ! SET_LINE_COLOR

當輸出裝置爲 Plotter 或 RAMTEK時，此常式方有作用：

1) Plotter : LLNC = 1 : PEN 1 (default)

LLNC = 2 : PEN 2

2) RAMTEK :

0≤ LLNC ≤ 192

default LLNC = 7 (白色) 。可由常式YDCI 指定與調配顏色。

(七)定多邊形的內部之塗色

CALL YSPGC (LLNC) ! SET_POLYGON_FILL_COLOR

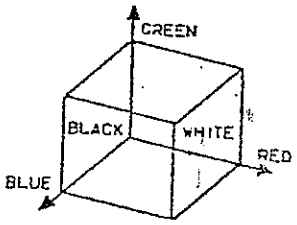
LLNC 的意義同(六)。

(八)定基色的類別

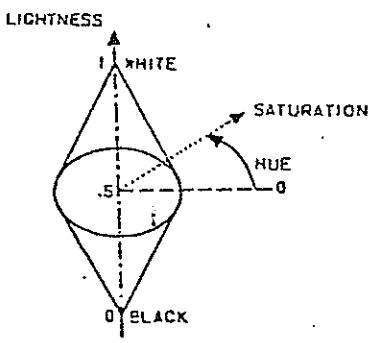
CALL YSCM (MODEL)

MODEL = 'RGB' (default)

or 'HLS'



5.1 RGB 基色與指標之關係



5.1.1 HLS 基色與指標之關係

(九) RGB 顏色的調配

CALL YDCI (ICLR, R, G, B)

其中 $0 \leq$ ICLR ≤ 192 ; $0 \leq R,G,B \leq 1$

ICLR : 表顏色的指標，供往後取用。

R, G, B 分別表 RED, GREEN, BLUE 之成分。

(十) HLS 顏色的調配

CALL YDCI (ICLR, H, L, S)

其中 H, L, S 均爲實數

且 $0 \leq$ ICLR ≤ 192;其意義與(九)同

$0 \leq H \leq 360$ 表 HUE (色彩)

$0 \leq L \leq 1$ 表 LIGHTNESS (明亮度)

$0 \leq S \leq 1$ 表 SATURATION (飽和度)

且 $S + L \leq 1$ (見圖 5.1.1)

系統指標 (USER可重定)	色澤	基色類別 (R G B)	(H L S)
0	BLACK	(0 0 0)	(0 0 0)
1	RED	(1 0 0)	(120 .5 1)
2	GREEN	(0 1 0)	(240 .5 1)
3	YELLOW	(1 1 0)	(180 .5 1)
4	BLUE	(0 0 1)	(0 .5 1)
5	MAGENTA	(1 0 1)	(60 .5 1)
6	CYAN	(0 1 1)	(300 .5 1)
7-192	WHITE	(1 1 1)	(0 1 0)

表5-1 色彩的調配與系統自定的顏色指標

(十) 定字元串的大小與書寫角度

CALL YSTEXT (XL, YL, DEGREE)

在呼叫 YWTEX 前，可利用此常式定出即將輸出的字元串或文字的大小與書寫角度，其中 XL, YL 爲字的長、寬（單位與實物座標相同），DEGREE 爲實數之度度量。Default : (0.01,0.01,0)

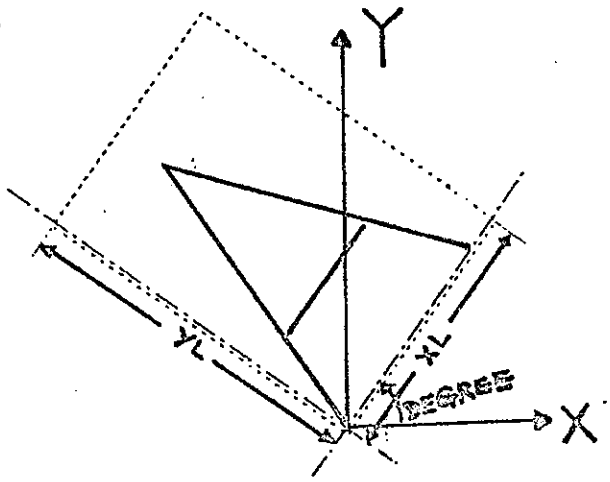


圖 5-2 大寫字母的機定位置

(十一) 定特殊符號的大小與書寫角度

CALL YSMARK (XL, YL, DEGREE) ! SET_MARK_CONDITION

XL, YL, DEGTEE 與上述常式 YSTEXT 相同，特殊符號乃指 YPMARK 所選用者。

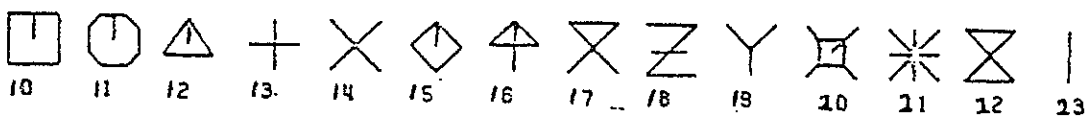
(十二) 選擇特殊符號

CALL YPMARK (INDEX) ! PICK_MARK_INDEX

此常式用於選用特殊符號（常式 YWMARK 或 YMARKS）前。

爲一整數，範圍在 1 與 32 之間。目前 INDEX = 10~23 時定義如下：(: Default =, INDEX := 10)

圖 5-3 10 至 23 的特殊符號



其餘 (1~9, 24~32) 未定義者，皆視同 INDEX = 10 之符號。

此外，輸出符號之大小、方向等可由常式YSMARK 來控制。

(四)定 RAMTEK 螢幕之底色

```
CALL  YSBC (ICLR)  !  SET_BACKGROUND_COLOR
```

ICLR 爲一顏色的指標，可由常式 YDCI 來定義其顏色。Default ICLR := 7（白色）。

(五)定多邊形之邊線的類別

```
CALL  YSPES (LPES)  !  SET_POLYGON_EDGE_STYLE
      LPES := 'SOLID' (default)
           or 'INTERIOR'
```

當 LPES := 'SOLID'時，多邊形輸出時以線的顏色表現，否則以其內部的顏色表現。

(六)當輸出設置不爲 RAMTEK 時，一般情形多邊形的輸出僅以其邊線來表示，假若要求以多邊形的內部亦須填滿時，則

```
CALL  YSPF (LPF)  !  SET_POLYGON_FILL
      LPF := 'NO' (default)
          or 'YES'
```

(七)決定多邊形內部的塗色類別（僅用於 RAMTEK）

```
CALL  YSPIS (IPIS)  !  SET_POLYGON_INTERIOR_STYLE
      IPIS := 'PLANE' (default)
           or 'SHADED'
           or 'PATTERN'
```

(i) IPIS = 'PLANE' 則多邊形輸出時，以 YSPGC 所定的顏色塗滿。

(ii) IPIS = 'SHADED' 則依照所定的多邊形各個頂點的顏色（由 YSVI 指定）。

將內部各點的顏色用內插法逐次算出並顯像之。

（SCGS 是由一條和 Y 軸平行的線逐次地由左向右掃）。

(iii) IPIS = 'PATTERN' 則依照事先所定的色塊，將之一塊一塊地，塗抹在多邊形內部（色塊的定法，參見 YSPPO 與 YSPA）。

(六) 定多邊形頂點的塗色

CALL YSVI (LVI, N_V) ! SET_VERTEX_INDEX

表示多邊形的頂點從1 至 N_V 將分別以 LVI(1) 至 LVI(N_V) 等色定之。

(七) 定色塊的基準點

CALL YSPPO (X0,Y0) ! SET_PIXEL_ORIGIN_2

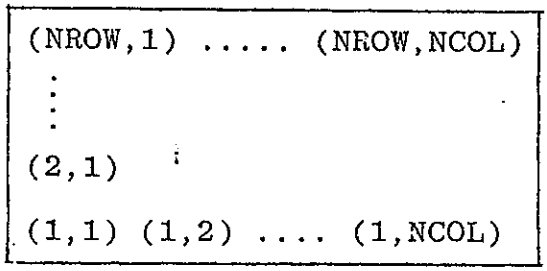
default(X0,Y0) := (0, 0)

往後色塊在輸出時，將以 (X0,Y0) 爲基準點（原點），在適當處顯像。

(八) 定色塊的顏色

CALL YSPA (IPA,NCOL,NROW) ! SET_PIXEL_ARRAY

色塊的安排如下圖



IPA(1..NROW,1..NCOL)

㈤設置範圍的製定

```
Call  YDVSZ  (XMIN,XMAX,YMIN,YMAX) ! DEVICE_SIZE_LIMIT
Default : (XMIN,XMAX,YMIN,YMAX) : =
          (0,1.35,0,1) for RAMTEK,
      or  (0,1.31,0,1) for TEKTRONIX,
      or  (-32767,32767,-32767,32767) for PLOTTER,
      or  (-32767,32767,-32767,32767) for KEYBOARD.
```

六、後記

㈠使用SCGS各常式前，必先呼叫YINI，且呼叫YEXIT 以爲終止。

㈡在呼叫 YEXIT 後若須要的話，可再呼叫 YINI重新開始另一迴路，此時一切的值均會自動回到機定值（如輸出設置爲TEKTRONIX)

㈢如果你認爲SCGS中所用的常式名稱，不好使用，或難以記憶，可以下列方式組織一個名稱的介面常式：

```
SUBROUTINE MOVE (A1, A2)
CALL YWMV (A1, A2)
RETURN
ENTRY LINE (A1, A2)
CALL YWLN (A1, A2)
RETURN
ENTRY PLYGON (A1, A2, A3)
CALL YWPG (A1, A2, A3)
END
```

於是 CALL MOVE (A1,A2)即等於 CALL YWMV (A1,A2),以此類推。

㈣當輸出裝置被指定爲 RAMTEK時，即 CALL YWDV ('RAmtek') 顏色指標 0-7 會自動產生，但仍可用YDCI 常式重新設定。

㈤ YGDV, YWDV 及…等各常式所使用的引數(argument) 只有前 2 個文字有效。故 CALL YGDV ('RA') 即等於 CALL YGDV ('RAMTEK') 以此類推。

㈥ SCGS Library 存在於 DR0 : [1,1] SCGS.OLB

(七) SCGS 所預留之 logical unit no. 如下：

6 : plotter

7,8 : RAMTEK

9,12,13 : System

10 : Digitizer

11 : TEKTRONIX

14 : 保留（往後若增加其他繪圖的功能時會用到）。

(八) SCGS 使用的 COMMON name 爲

```
BITS
DEVCHR
INTERN
L4COM
OPTS
OVRLAY
PARMS
PLTCOM
STDCOL
TEST
VLTDAT
WORK
XBUFF
XNUMBE
XPDMP
XPLOT
XSYMBO
YPLT2
YPLT3
```

(九) 實例：

1) Source file (EXAMP.FTN)

```
PROGRAM EXAMP
BYTE IFLAG
CALL YINI
CALL YCTB ('Y')
CALL YGDV ('RA')
CALL YGPT (X0, Y0, IFLAG)
CALL YGPT (X1, Y1, IFLAG)
CALL YDVP (X0, X1, Y0, Y1)
CALL YWDV ('PL')
CALL YSLS ('DA')
CALL YWWD
CALL YEXIT
END
```

2) Task builder command file (EXAMP.CMD)

```
EXAMP/FP=EXAMP,LB:[1,1]SCGS/LB
/
UNITS=13
ACTFIL=6
ASG=TT2:6,RM:7:8,SY:9:12:13
//
```

3) Operation procedures:

```
> F4P EXAMP=EXAMP
> TKB @EXAMP
> RUN EXAMP
```

七、測試程式：

LB:[1,54]SCGS.TSK 爲一用來試驗 SCGS 中各常式的 task，如果你想熟悉各常式的功能，可 RUN $SCGS，如下：

```
>RUN $SCGS
Routine Name?[A6]: YINI
Routine Name?[A6]: YWWD
Routine Name?[A6]: YGDV
Input Device Name?[A2](DI): TE
Routine Name?[A6]: YCTB
YCTB---Control Trace Back?[Y/N](Y): Y
Routine Name?[A6]: YGDV
Input Device Name?[A2](DI): keyboard
 Illegal input device name.
  *** The traceback route is:
TT5  --  ERROR 73
FLOATING ZERO DIVIDE
AT PC = 031640
  IN    "YERR   " AT OR AFTER 5
  FROM "YGDV   " AT OR AFTER 15
  FROM "SCGS0  " AT OR AFTER 22
  FROM "SCGS   " AT OR AFTER 2


Routine Name?[A6]: YWDV
Output Device Name?[A2](TE): PL
Routine Name?[A6]: YWWD
Routine Name?[A6]: YWDV
Output Device Name?[A2](TE): KE
Routine Name?[A6]: YWWD
Draw (  0.00000,   0.00000) to (   1.00000,   0.00000)
Draw (   1.00000,   0.00000) to (   1.00000,   1.00000)
Draw (   1.00000,   1.00000) to (   0.00000,   1.00000)
Draw (   0.00000,   1.00000) to (   0.00000,   0.00000)
Routine Name?[A6]: YEXIT
Routine Name?[A6]: STOP
TT5  --  STOP *** SCGS  test program stop!!!
```

Note: After run $SCGS, a question "$Routine Name?" is given, then you can type any routine's name supplied by SCGS, or name "HELP", "EXIT" and "STOP".

附錄二：SCGS 自我測試的主程式

```
      PROGRAM SCGS
      CALL SCGS0
      END
      SUBROUTINE SCGS0          ! MAIN TEST ROUTINE FOR SCGS
      BYTE LCTB,LCTB2,STRING(80),IFLAG
      INTEGER PX(256),PY(256)
      REAL*4 CENTER(2),P0(2),VX(64),VY(64),P1(2),P2(2)
      REAL*8 NRTN
      DATA MODEL/'RG'/
9999  CALL PUSH1 ('$Routine Name?[A6]: /\')
      READ(5,20,END=9999) NRTN
20    FORMAT(A6)
22    FORMAT(A2)
25    FORMAT(80A1)
      IF (NRTN .NE. 'YINI  ') GOTO 1000
      CALL YINI
      GOTO 9999
1000  IF (NRTN .NE. 'YTEST ') GOTO 1010
      CALL PUSH1 ('$TEST_ID := /\')
      ACCEPT*,ID
      CALL YTEST (ID)
      GOTO 9999
1010  IF (NRTN .NE. 'YEXIT ') GOTO 1020
      CALL YEXIT
      GOTO 9999
1020  IF (NRTN .NE. 'YCTB  ') GOTO 1030
1022  CALL PUSH1 ('$YCTB---Control Trace Back?[Y/N](Y): /\')
      READ(5,25,END=1022) LCTB
      IF (LCTB .EQ. ' ') LCTB = 'Y'
      CALL YCTB (LCTB)
      GOTO 9999
1030  IF (NRTN .NE. 'YFCC  ') GOTO 1040
      CALL PUSH1 ('$RADIUS := /\')
      ACCEPT*,RADIUS
      TYPE*,'CENTER:='
      CALL YGPT (CENTER(1), CENTER(2), IFLAG)
      CALL YFCC (CENTER, RADIUS)
      GOTO 9999
1040  IF (NRTN.NE.'YPOP' .AND. NRTN.NE.'YPUSH') GOTO 1050
      CALL PUSH1 ('$Filename := /\')
      ACCEPT25,(STRING(I1),I1=1,30)
      IF (NRTN .EQ. 'YPOP  ') CALL YPOP (STRING)
      IF (NRTN .EQ. 'YPUSH ') CALL YPUSH(STRING)
      GOTO 9999
1050  IF (NRTN .NE. 'YCLOSE') GOTO 1060
      CALL YCLOSE
      GOTO 9999
1060  IF (NRTN .NE. 'YWCC3 ') GOTO 2000
      CALL YGPT (P0(1), P0(2), IFLAG)
      CALL YGPT (P1(1), P1(2), IFLAG)
      CALL YGPT (P2(1), P2(2), IFLAG)
      CALL YWCC3 (P0, P1, P2)
      GOTO 9999
```

```
2000    IF (NRTN .NE. 'YGDV  ') GOTO 2010
2001    CALL PUSH1 ('$Input Device Name?[A2](DI): /\')
        READ(5,22,END=3001) NGDV
        IF (NGDV .EQ. '  ') NGDV = 'DI'
        CALL YGDV (NGDV)
        GOTO 9999
2010    IF (NRTN .NE. 'YGPT  ') GOTO 2020
        CALL YGPT (X, Y, IFLAG)
2011    WRITE(5,2012) X,Y,IFLAG,IFLAG,IFLAG
2012    FORMAT(' YGPT---X,Y,IFLAG(D8,I8,A1)='2F10.6,5X,D8,I8,2X,A1)
        GOTO 9999
2020    IF (NRTN .NE. 'YGPTS ') GOTO 2030
        CALL PUSH1 ('$TIME LIMIT IN SECOND= /\')
        ACCEPT*,SECLIM
        CALL YGPTS (X, Y, IFLAG, SECLIM)
        GOTO 2011
2030    IF (NRTN .NE. 'YSVUP ') GOTO 2040
        CALL PUSH1 ('$UPX,UPY= /\')
        ACCEPT*,UPX,UPY
        CALL YSVUP (UPX, UPY)
        GOTO 9999
2040    IF (NRTN .NE. 'YSPES ') GOTO 2050
        CALL PUSH1 ('$IPES= /\')
        ACCEPT22,IPES
        CALL YSPES (IPES)
        GOTO 9999
2050    IF (NRTN .NE. 'YSCM  ') GOTO 2060
2051    CALL PUSH1 ('$MODEL FOR COLOR = /\')
        ACCEPT22,MODEL
        IF (MODEL.NE.'RG' .AND. MODEL.NE.'HL') GOTO 2051
        CALL YSCM (MODEL)
        GOTO 9999
2060    IF (NRTN .NE. 'YSPF  ') GOTO 2070
        CALL PUSH1 ('$FILL?[Y/N](Y) = /\')
        ACCEPT25,NFILL
        IF (NFILL .EQ. ' ') NFILL = 'Y'
        CALL YSPF (NFILL)
        GOTO 9999
2070    IF (NRTN .NE. 'YSPIS ') GOTO 2090
        CALL PUSH1 ('$IPIS(POLYGON_INTERIOR_STYLE) = /\')
        ACCEPT22,IPIS
        CALL YSPIS (IPIS)
        GOTO 9999
2090    IF (NRTN .NE. 'YSVI  ') GOTO 2100
        CALL PUSH1 ('$NO_OF_VERTEX= /\')
        ACCEPT*,N
        IF (N .LE. 1) GOTO 9999
        CALL PUSH1 ('$VERTEX_INDEX = /\')
        ACCEPT*,(PX(I1),I1=1,N)
        CALL YSVI (PX, N)
        GOTO 9999
```

```
2100    IF (NRTN .NE. 'YSPPO ') GOTO 2110
        CALL PUSH1 ('$ORIGIN(XO,YO) = /\')
        ACCEPT*,XO,YO
        CALL YSPPO (XO, YO)
        GOTO 9999
2110    IF (NRTN .NE. 'YSPA  ') GOTO 3000
        CALL PUSH1 ('$SET_PIXEL_ARRAY:NCOL,NROW = /\')
        ACCEPT*,NCOL,NROW
        IF (NCOL.LE.0 .OR. NROW.LE.0) GOTO 9999
        NUMBER = NCOL * NROW
        CALL PUSH1 ('$PIXEL_ARRAY(1st col., then 2nd col., ... etc.) = /\'
        ACCEPT*,(PX(I1),I1=1,NUMBER)
        CALL YSPA (PX, NCOL, NROW)
        GOTO 9999
3000    IF (NRTN .NE. 'YWDV  ') GOTO 3010
3001    CALL PUSH1 ('$Output Device Name?[A2](TE): /\')
        READ(5,22,END=3001) NWDV
        IF (NWDV .EQ. '  ') NWDV = 'TE'
        CALL YWDV (NWDV)
        GOTO 9999
3010    IF(NRTN.NE.'YDVP'.AND.NRTN.NE.'YDVSZ'.AND.NRTN.NE.'YDWD')GOTO3020
        CALL YGPT (VX0, VY0, IFLAG)
        CALL YGPT (VX1, VY1, IFLAG)
        IF (NRTN .EQ. 'YDVP  ') CALL YDVP (VX0, VX1, VY0, VY1)
        IF (NRTN .EQ. 'YDVSZ ') CALL YDVSZ(VX0, VX1, VY0, VY1)
        IF (NRTN .EQ. 'YDWD  ') CALL YDWD (VX0, VX1, VY0, VY1)
        GOTO 9999
3020    IF (NRTN .NE. 'YWWD  ') GOTO 3040
        CALL YWWD
        GOTO 9999
3040    IF (NRTN .NE. 'YWMV  ') GOTO 3050
        CALL YGPT (X, Y, IFLAG)
        CALL YWMV (X, Y)
        GOTO 9999
3050    IF (NRTN .NE. 'YWLN  ') GOTO 3060
        CALL YGPT (X, Y, IFLAG)
        CALL YWLN (X, Y)
        GOTO 9999
3060    IF (NRTN .NE. 'YWCC  ') GOTO 3070
        TYPE*,'Input Center:'
        CALL YGPT (CENTER(1), CENTER(2), IFLAG)
        TYPE*,'Input PO:'
        CALL YGPT (PO(1), PO(2), IFLAG)
3064    CALL PUSH1 ('$Degree= /\')
        READ(5,*,END=3064) DEGREE
        CALL YWCC (CENTER, PO, DEGREE)
        GOTO 9999
```

```
3070    IF(NRTN.NE.'YWPL' .AND. NRTN.NE.'YWPG' .AND. NRTN.NE.'YMARKS'
        1       .AND. NRTN.NE.'YWPLN') GOTO 3080
3071    CALL PUSH1 ('$No. of vertex?(<=64): /\')
        READ(5,*,END=3071) NV
        DO 3076 I1 = 1 , NV
          CALL YGPT (VX(I1), VY(I1), IFLAG)
3076    CONTINUE
        IF (NRTN .EQ. 'YWPL  ') CALL YWPL  (VX, VY, NV)
        IF (NRTN .EQ. 'YWPLN ') CALL YWPLN (VX, VY, NV)
        IF (NRTN .EQ. 'YWPG  ') CALL YWPG  (VX, VY, NV)
        IF (NRTN .EQ. 'YMARKS') CALL YMARKS (VX, VY, NV)
        GOTO 9999
3080    IF (NRTN .NE. 'YSBC  ') GOTO 3090
3081    CALL PUSH1 ('$Back Ground Color index= /\')
        READ(5,*,END=3081) IBGC
        CALL YSBC (IBGC)
        GOTO 9999
3090    IF (NRTN .NE. 'YWER  ') GOTO 3110
        CALL YWER
        GOTO 9999
3110    IF (NRTN .NE. 'YWTEXT') GOTO 3120
3112    CALL PUSH1 ('$TEXT_STRING=(<=80): /\')
        READ(5,25,END=3112) STRING
D       TYPE3113,STRING
D3113   FORMAT(1X,80A1)
        DO 3115 I1 = 80 , 1 , -1
          IF (STRING(I1) .NE. ' ') GOTO 3117
3115    CONTINUE
        GOTO 3112
3117    NTEXT = I1
D       TYPE*,'SCGS---NTEXT=',NTEXT
        CALL YWTEXT (STRING, NTEXT)
D       TYPE*,'SCGS---EXIT FROM YTEXT.'
        GOTO 9999
3120    IF (NRTN .NE. 'YWMARK') GOTO 4000
        CALL YWMARK
        GOTO 9999
4000    IF (NRTN.NE.'YSCCP ' .AND. NRTN.NE.'YSDLP ') GOTO 4020
4012    CALL PUSH1 ('$Precision length(>0)= /\')
        READ(5,*,END=4012) PL
        IF (NRTN .EQ. 'YSCCP ') CALL YSCCP (PL)
        IF (NRTN .EQ. 'YSDLP ') CALL YSDLP (PL)
        GOTO 9999
4020    IF (NRTN .NE. 'YSLNC ') GOTO 4030
4022    CALL PUSH1 ('$Linecolor= /\')
        READ(5,*,END=4022) LLNC
        CALL YSLNC (LLNC)
        GOTO 9999
4030    IF (NRTN .NE. 'YSPGC ') GOTO 4040
4032    CALL PUSH1 ('$Polygon Fill Color index? /\')
        READ(5,*,END=4032) LPGC
        CALL YSPGC (LPGC)
        GOTO 9999
```

```
4040    IF (NRTN .NE. 'YDCI  ') GOTO 4050
4041    IF (MODEL .EQ. 'RG') CALL PUSH1 ('$YDCI---ICLR,R,G,B= /\')
        IF (MODEL .NE. 'RG') CALL PUSH1 ('$YDCI---ICLR,H,L,S= /\')
        READ(5,*,END=4041) ICLR,R,G,B
        CALL YDCI (ICLR, R, G, B)
        GOTO 9999
4050    IF (NRTN .NE. 'YSTEXT') GOTO 4060
4051    CALL PUSH1 ('$YSTEXT---XL,YL,DEGREE= /\')
        READ(5,*,END=4051) CHRX,CHRY,DEGREE
        CALL YSTEXT (CHRX, CHRY, DEGREE)
        GOTO 9999
4060    IF (NRTN .NE. 'YSMARK') GOTO 4070
        CALL PUSH1 ('$YSMARK---XL,YL,DEGREE= /\')
        ACCEPT*,SYMX,SYMY,DEG
        CALL YSMARK (SYMX, SYMY, DEG)
        GOTO 9999
4070    IF (NRTN.NE.'YSLNS ' .AND. NRTN.NE.'YSCCS ') GOTO 4080
        CALL PUSH1 ('$LS= /\')
        ACCEPT22,LS
        IF (NRTN .EQ. 'YSLNS ') CALL YSLNS (LS)
        IF (NRTN .EQ. 'YSCCS ') CALL YSCCS (LS)
        GOTO 9999
4080    IF (NRTN .NE. 'YSMARK') GOTO 4090
        CALL YWMARK
        GOTO 9999
4090    IF (NRTN .NE. 'YPMARK') GOTO 4100
        CALL PUSH1 ('$Mark-symbol index= /\')
        ACCEPT*,MARK
        CALL YPMARK (MARK)
        GOTO 9999
4100    IF (NRTN .NE. 'YSPLC ') GOTO 4110
        CALL PUSH1 ('$Which coordinate[WORLD/NDC](NDC): /\')
        ACCEPT22,NC
        IF (NC .EQ. '  ') NC = 'ND'
        CALL YSPLC (NC)
        GOTO 9999
4110    IF (NRTN .NE. 'YSRCS ') GOTO 5000
        CALL PUSH1 ('$LRCS= /\')
        ACCEPT*,LRCS
        CALL YSRCS (LRCS)
        GOTO 9999
5000    IF (NRTN .EQ. 'EXIT  ') RETURN
5010    IF (NRTN .EQ. 'STOP  ') STOP '*** SCGS  test program stop!!
6000    IF (NRTN .NE. 'HELP  ') GOTO 7000
        CALL YHELP
        GOTO 9999
7000    TYPE*,'Illegal routine name, input again.'
        GOTO 9999
        END
```

附錄三：建附錄二的 OVERLAY DESCRIPTION LANGUAGE

```
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;                                                       ;;;;;;;;;;;;;
;;;;;;;;;;;;;   L B 0 : [ 1 , 1 ] S C G S . O D L                   ;;;;;;;;;;;;;
;;;;;;;;;;;;;                                                       ;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
        .ROOT LB:[1,1]SCGS/LB:SCGS:SCGS0-M23-MAIN-HARD
MAIN:   .FCTR LB:[1,1]SCGS/LB:YTINI:YINI:YWMV:YERR:YROOT:YPOP:YWCC3-TEST
M23:    .FCTR LB:[1,1]SCGS/LB:PUSH1:PUSH
TEST:   .FCTR LB:[1,1]SCGS/LB:YTEST:YTEST0:YTEST1:YTEST2:YTEST3
SOFT:   .FCTR LB:[1,1]SCGS/LB:YCTB:YSLNS:YDVP:YDWD:YDVSZ:YHELP-SOFT1
SOFT1:  .FCTR LB:[1,1]SCGS/LB:YSPGC:YSPLC:YSCCS:YPMARK:YSCCP:YSDLP:YSRCS-SOFT2
SOFT2:  .FCTR LB:[1,1]SCGS/LB:YSTEXT:YSVUP:YSWUP:YSPES:YSCM:YSPF-SOFT3
SOFT3:  .FCTR LB:[1,1]SCGS/LB:YFILE:YPUSH:YCLOSE:YDEG:YCTR:YLEQ2-SOFT4
SOFT4:  .FCTR LB:[1,1]SCGS/LB:YSPPO:YSPA:YSPIS:YSVI-SOFT5
SOFT5:  .FCTR LB:[1,1]SCGS/LB:YWCCN:YCDEV:YSMARK:YVCLIP-LB:[1,1]SCGS/LB
HARD:   .FCTR LB:[1,1]SCGS/LB:YGPT:YWLN:YWCC:YWPL:YWPLN:YWTEXT-H23-HARD1
H23:    .FCTR LB:[1,1]SCGS/LB:VEQU:VADD:VMULNO
HARD1:  .FCTR LB:[1,1]SCGS/LB:YWER:YWWD:YWMARK:YMARKS:YWPG:YPGF:YCUTC1-HARD2
HARD2:  .FCTR LB:[1,1]SCGS/LB:YCHAR:YGPTS:YWDLN:YWSLN:YWDR:YWCDR-HARD3
HARD3:  .FCTR LB:[1,1]SCGS/LB:YGDV:YWDV:YLNSH0:YLNSH2:YLNPL:YSLNC-HARD4
HARD4:  .FCTR LB:[1,1]SCGS/LB:YFCC:YNDR:-*(DEVDI,DEVPL,DEVTE,DEVRAM,SOFT)
DEVDI:  .FCTR LB:[1,1]SCGS/LB:GETDIT-LB:[1,1]SCGS/LB
DEVPL:  .FCTR LB:[1,1]SCGS/LB:YEXIT-P23-LB:[1,1]SCGS/LB
P23:    .FCTR LB:[1,1]SCGS/LB:XPLOTS:XPLOT:XFACTO:XNEWPE
DEVTE:  .FCTR LB:[1,1]SCGS/LB:YTEKLN-T23-LB:[1,1]SCGS/LB
T23:    .FCTR LB:[1,1]SCGS/LB:ERASE:POINT:CRHAIR:RCHAIR:GRAMO
DEVRAM: .FCTR *(YOVRA0,DEVDR)
YOVRA0: .FCTR LB:[1,1]SCGS/LB:YINIRM:YRCS:YSBC:YDCI0:YDCI:YDCIR-YOVRA1
YOVRA1: .FCTR *(LB:[1,1]SCGS/LB:YHLRG:YVAL:YFCI,DEVRA)
DEVDR:  .FCTR LB:[1,1]SCGS/LB:YCLIP:YTWV:YNEAR:YWDMP:YWDMP0:YMDP:YSWAP
DEVRA:  .FCTR RTO-*(MINST,NINST)
RTO:    .FCTR LB:[1,1]SCGS/LB:PLOTS:PLOT:ERROR
MINST:  .FCTR LB:[1,1]SCGS/LB:NEWPEN:NEWP:IMAGEC-MINST0
MINST0: .FCTR LB:[1,1]SCGS/LB:WINERS-*(RINST,OINST)
RINST:  .FCTR LB:[1,1]SCGS/LB:INST:RCSMDF:ERS-RINST0
RINST0: .FCTR LB:[1,1]SCGS/LB:SET:LAM:WV:READ-INSTR
INSTR:  .FCTR *(INST0,INST1,INST2)
INST0:  .FCTR LB:[1,1]SCGS/LB:BUILD:CLRPRM
INST1:  .FCTR LB:[1,1]SCGS/LB:RMWD:RMRD
INST2:  .FCTR LB:[1,1]SCGS/LB:CLRPRE:SETPRE:CHECK:RMWGPC
OINST:  .FCTR LB:[1,1]SCGS/LB:FGD:BGD:WIN:COP
INST3:  .FCTR LB:[1,1]SCGS/LB:IMAGEC:WINERS:BGD:NEWP:COP:WIN:FGD
NINST:  .FCTR LB:[1,1]SCGS/LB:INITL3:CLRPRM:INITL4:RMWST:CHECK:RMWGPC
        .END
```