

M E M O R A N D U M

A : Utilisateurs software des consoles (CO list 7a)

De : F. Perriollat, A. Gagnaire

Concerne : Modules P-plus pour les consoles

Veillez trouver ci-contre une version mise à jour de la note PS/CO/Note 81-37 / Rev. 4.

F. Perriollat

A. Gagnaire

List 7a) Software Information

V. Adorni, G. Baribaud, G.P. Benincasa, M. Bennett,
A. Branz, R. Cailliau, C. Cate-Bettels, B. Carpenter,
V. Chohan, G. Cuisinier, J. Cupérus, A. Daneels,
G. Daems, R. Debordes, R. Delgado, A. Gagnaire,
F. Giudici, D. Hallberg, W. Heinze, P. Heymans,
J. Kenaghan, B. Kuiper, J. Kupiec, M. Lelaizant,
J. Lewis, E. Malandain, M. Martini, L. Mérard,
J. Navratil, K. Osen, F. Perriollat, R. Pluta,
C. Poinard, J.P. Potier, G. Quickfall, E. Ratcliff,
W. Remmer, J. Redard, F. Rohner, Ch. Serre,
C.H. Sicard, P. Skarek, Ch. Steinbach, N. Vogt-Nilsen

M E M O R A N D U M

A : Utilisateurs software des consoles (CO list 7a)

De : F. Perriollat, A. Gagnaire

Concerne : Modules P-plus pour les consoles

Veillez trouver ci-contre une version mise à jour de la note PS/CO/Note 81-37 / Rev. 4.

F. Perriollat

A. Gagnaire

List 7a) Software Information

V. Adorni, G. Baribaud, G.P. Benincasa, M. Bennett,
A. Branz, R. Cailliau, C. Cate-Bettels, B. Carpenter,
V. Chohan, G. Cuisinier, J. Cupérus, A. Daneels,
G. Daems, R. Debordes, R. Delgado, A. Gagnaire,
F. Giudici, D. Hallberg, W. Heinze, P. Heymans,
J. Kenaghan, B. Kuiper, J. Kupiec, M. Lelaizant,
J. Lewis, E. Malandain, M. Martini, L. Mérard,
J. Navratil, K. Osen, F. Perriollat, R. Pluta,
C. Poinard, J.P. Potier, G. Quickfall, E. Ratcliff,
W. Remmer, J. Redard, F. Rohner, Ch. Serre,
C.H. Sicard, P. Skarek, Ch. Steinbach, N. Vogt-Nilsen

(*

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH
GENEVA, SWITZERLAND

PS/CO/NOTE 81-37/REV.4
11-NOV-1982

P-PLUS MODULES FOR CONSOLE ENVIRONMENT

F. PERRIOLLAT, A. GAGNAIRE

*)

(*

TABLE OF CONTENTS

- 1 - GENERAL XIP CONSOLE FACILITIES (XIP_TOOLS)
- 2 - DEDICATED MIP CONSOLE FACILITIES (MIP_TOOLS)
- 3 - SPECIAL GFA AND BRIGHT-UP FACILITIES FOR MIP (GFA_TOOLS)
- 4 - DEDICATED SIP CONSOLE FACILITIES (SIP_TOOLS)
- 5 - NON INTERACTIVE PROGRAMS CONSOLE FACILITIES (SRT_TOOLS)
- 6 - GENERAL GRAPHIC CONSOLE FACILITIES (GRAPHIC_FUNCTIONS)
- 7 - DEDICATED MIP GRAPHIC CONSOLE FACILITIES (MIP_GRAPHIC)
- 8 - LOAD-TIME (RT-LOADER) INFORMATIONS

*)

```
(*****  
(*)  
(* 1 - GENERAL XIP CONSOLE FACILITIES ( XIP_TOOLS ) *)  
(*)  
(*) GENERAL XIP CONSOLE FACILITIES *)  
(*) AND VIP & LIP SPECIAL FEATURES *)  
(*)  
(*****)
```

MODULE PROVIDING XIP_TOOLS AS LIBRARY:

PROVIDE ALL WITH

```
(* 1.1 CONSTANT AND TYPE DEFINITION *)  
(*) ===== *)
```

```
(* TARGET SOFTWARE-TRIGGER IDENTIFICATION FOR TRIG CALL *)  
(*) ----- *)
```

TYPE TRIGGER:

```
(DUMMY_TRIGGER, (* MUST NEVER BE USED *)  
TRGTIP, (* TRIGGER TO TIP *)  
TRGMIP, (* TRIGGER TO MIP *)  
TRGVIP, (* TRIGGER TO VIP *)  
TRGLIP, (* TRIGGER TO LIP *)  
TRGSIP); (* TRIGGER TO SIP *)
```

```
(* GROUP NUMBER FOR EVENT IDENTIFICATION *)  
(*) ----- *)
```

```
(* FOR MWAIT AND BUTS CALL *)
```

CONSTANT

```
TRIGER : 255; (* SOFTWARE TRIGGER GROUP *)  
EVGRUP : 254; (* PROCESS-EVENT GROUP *)
```

(* VIDEO SCREEN CONTROL CHARACTERS DEFINITION *)
(* ----- *)

CONSTANT

RED : CHAR(0); (* RED COLOUR *)
GREEN : CHAR(1); (* GREEN COLOUR *)
BLUE : CHAR(2); (* DARK BLUE COLOUR *)
WHITE : CHAR(3); (* WHITE COLOUR *)
LBLUE : CHAR(4); (* LIGHT BLUE COLOUR *)
MAGENT : CHAR(5); (* MAGENTA COLOUR *)
YELLOW : CHAR(6); (* YELLOW COLOUR *)
BLACK : CHAR(7); (* BLACK COLOUR *)

ERASE : CHAR(8#14#); (* ERASE AND RESET SCREEN *)
INVERT : CHAR(8#17#); (* SWAP BACKGROUND AND FOREGROUND COLOUR *)
BACKGR : CHAR(8#20#); (* BACKGROUND COLOUR INTO NEXT CHARACTER *)
DUMMY_CHAR : CHAR(8#33#); (* DUMMY CHARACTER (NON-PRINTABLE) *)

(* PROSSESS-EVENT TYPE DEFINITION FOR EVENT CALL *)
(* ----- *)

TYPE

MACHINE: (BOOSTER,PS,DUMMY_1,DUMMY_2);

PULSE: (DUMMY_PULSE,
PULSE_1,
PULSE_2,
PULSE_3,
PULSE_4,
BX_WBC,
BX_STB,
BX_RBI,
PX_WTR,
PX_RSC,
PX_STC,
PX_ELFT,
PX_EPC,
PULSE_13,
PULSE_14,
PULSE_15,
PULSE_16);

(* TOUCH-PANEL PAGE RECORD DEFINITION *)
(* ----- *)

CONSTANT

LEGEND_SIZE : 24; (* 3 LINES OF 8 CHARACTERS *)
TITLE_SIZE : 52; (* 48 PRINTABLE CHARACTERS AND 4 INVERT CONTROL *)

RANGE

BUTTON_NUMBER : 1..16;

TYPE

BUTTON_LEGEND : RECORD
 B_ENABLE : BOOLEAN; (* FALSE = NO BUTTON *)
 (* TRUE = BUTTON ENABLE AND VISIBLE *)
 B_INVERT : BOOLEAN; (* FALSE = NORMAL BLACK ON WHITE LEGEND *)
 (* TRUE = INVERTED LEGEND VIDEO : WHITE ON BLACK *)
 B_LEGEND : STRING [LEGEND_SIZE];
END RECORD;

BUTTON_LEGEND_ARRAY : ROW [BUTTON_NUMBER] OF BUTTON_LEGEND;

BUTTON_PAGE : RECORD

 P_ENABLE : BOOLEAN; (* TRUE = NEW PAGE OF BUTTONS *)
 (* FALSE = ERASE THE TOUCH-PANEL *)
 P_TITLE : STRING [TITLE_SIZE];
 P_BUTTON : BUTTON_LEGEND_ARRAY;
END RECORD;

(* NOTICE :

 THE PROCEDURE BTINV WRITES THE LEGEND IN VIDEO STATUS
 INVERTED ACCORDING TO THE B_INVERT BOOLEAN.
 FOR EXAMPLE : THE LEGEND OF A BUTTON DEFINED
 WITH B_INVERT = FALSE
 IS OVERRITTEN IN INVERTED VIDEO
 (WHITE ON BLACK LEGEND) .

*)

(* 1.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(*
FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
)

PROCEDURE BTINV (RO PAGE: BUTTON_PAGE;
RO BUTTON_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'VNITB';

PROCEDURE BTLEG (RO BUTTON: BUTTON_LEGEND;
RO BUTTON_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'GELTB';

FUNCTION BUTS (RO GROUP: INTEGER;
WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'IBUTS';

FUNCTION COLOUR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'COLOR';

FUNCTION COLUMN (RO COLUMN_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER) STRING[2]
ENTRY 'COL';

FUNCTION COMPUTER_NUMBER (RO COMP_NB: COMPUTER) INTEGER
ENTRY 'CMPNB';

FUNCTION EVAL_STRING (RO EXPR: STRING;
WO ERROR_CODE: INTEGER) REAL
ENTRY 'XEVAL';

FUNCTION EVENT (RO MACH: MACHINE;
RO PULSA: PULSE;
RO PLS_LINE: INTEGER;
WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'TNEVE';

FUNCTION GRAGR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'GRGRA';

PROCEDURE LEGEND (RO BUTTON_NUMBER: INTEGER;
RO LABEL: STRING;
WO ERROR_CODE: INTEGER)
ENTRY 'NEGEL';

FUNCTION LEGLUN (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'LEGLN';

PROCEDURE LGPAGE (RO PAGE: BUTTON_PAGE;
WO ERROR_CODE: INTEGER)
ENTRY 'GAPGL';


```

FUNCTION LINE      (RO LINE_NUMBER: INTEGER;
                   WO ERROR_CODE: INTEGER ) STRING[2]
                   ENTRY 'LYNE';

PROCEDURE MPXVID   (RO SOURCE: INTEGER;
                   RO SCREEN: INTEGER;
                   WO ERROR_CODE: INTEGER)
                   ENTRY 'VIMPX';

PROCEDURE MWAIT    (WO BUTVAL: INTEGER;
                   WO GROUP_NUMBER: INTEGER;
                   WO ERROR_CODE: INTEGER)
                   ENTRY 'WAITM';

PROCEDURE NEWS     (RO TEXT: STRING;
                   WO ERROR_CODE: INTEGER)
                   ENTRY 'SWEN';          (* USES ICCI REMOTELY ON MCR *)

PROCEDURE PLS_TELEGRAM (RO MACH: MACHINE;
                       WO TELEGRAM: ROW[L..H:INTEGER] OF INTEGER;
                       WO ERROR_CODE: INTEGER);

FUNCTION POSIT     (RO X_VALUE,Y_VALUE: INTEGER;
                   WO ERROR_CODE: INTEGER) STRING[4]
                   ENTRY 'TISOP';

FUNCTION SYSGR     (WO ERROR_CODE: INTEGER) INTEGER
                   ENTRY 'GRSYS';

FUNCTION TPGR      (WO ERROR_CODE: INTEGER) INTEGER
                   ENTRY 'GRTP';

PROCEDURE TRIG     (RO VALUE: INTEGER;
                   RO THIS_TRIGGER: TRIGGER;
                   WO ERROR_CODE: INTEGER)
                   ENTRY 'ITRIG';

PROCEDURE VIDEO    (RO LUN: INTEGER;
                   RO TEXT: STRING;
                   WO COCO: INTEGER)
                   ENTRY 'IVIDE';

FUNCTION VIDGR     (WO ERROR_CODE: INTEGER) INTEGER
                   ENTRY 'GRVID';

```

```
(* 1.3 END OF XIP_TOOLS *)  
(* ===== *)
```

```
END XIP_TOOLS.
```

```
(*****  
(*  
(* 2 - DEDICATED MIP CONSOLE FACILITIES ( MIP_TOOLS )  
(*  
(* SPECIAL MIP CONSOLE FACILITIES  
(* EXECPTED GFA & BRIGHT-UP FEATURES  
(*  
(*****
```

MODULE PROVIDING MIP_TOOLS AS LIBRARY:

PROVIDE ALL WITH

```
(* 2.1 CONSTANT AND TYPE DEFINITION *)  
(* ===== *)
```

```
(* KNOBS FACILITY *)  
(* ----- *)
```

```
TYPE  
  KNOB_MODE: (CUMNOV,INCNOV,CUMVAL,INCVAL);  
  KNOB_INIT: RECORD  
    INITIAL_VALUE: REAL;  
    KNOB_NUMBER: INTEGER[1..4];  
    MODE: KNOB_MODE;  
    BBW,EDPT: INTEGER;  
    MAX_VALUE,MIN_VALUE: REAL;  
    UNST: STRING[4];  
    TIST: STRING[24]  
  END RECORD;
```

```
(* TV-EDITING FACILITY *)  
(* ----- *)
```

```
TYPE TV_WINDOW: RECORD  
  LUN: INTEGER;  
  X_LINE,Y_COLUMN: INTEGER; (* DEFAULT = -1 *)  
  WINDOW: INTEGER (* DEFAULT = -1 *)  
END RECORD;
```

(* 2.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(*
FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
*)

- FUNCTION BALLGR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'BALGR';
- FUNCTION BALLST (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'STBAL';
- FUNCTION BANDW (RO INDEX: INTEGER;
WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'WANDB';
- FUNCTION GRAPH (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'HPARG';
- PROCEDURE IMAGTR (RO PICTURE: ROW[L..H:INTEGER] OF INTEGER;
WO CONVERTED_PICTURE: ROW[LO..HI:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'TRIMG';
- PROCEDURE KAPPEA (RO LUN: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'KAPEA';
- PROCEDURE SET_KCURC (RO VALUE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'SCURC';
- FUNCTION READ_KCURC (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'RCURC';
- PROCEDURE SET_KCURL (RO VALUE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'SCURL';
- FUNCTION READ_KCURL (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'RCURL';
- PROCEDURE KDSABL (RO LUN: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'DSABK';
- PROCEDURE KENABL (RO LUN: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'ENABK';
- PROCEDURE KINIT (RO INIT_VALUE: KNOB_INIT;
WO ERROR_CODE: INTEGER)
ENTRY 'INITK';

FUNCTION KNOB (RO KNOB_NUMBER: INTEGER;
 WO ERROR_CODE: INTEGER) REAL
 ENTRY 'IKNOB';

FUNCTION KNVGR (WO ERROR_CODE: INTEGER) INTEGER
 ENTRY 'GRKNV';

PROCEDURE KOVERW (RO VALUE: REAL;
 RO KNOB_NUMBER: INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'OVERK';

PROCEDURE KPURGE (RO KNOB_NUMBER: INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'PURKB';

FUNCTION KSTATU (RO KNOB_NUMBER: INTEGER;
 WO ERROR_CODE: INTEGER) INTEGER
 ENTRY 'STATK';

PROCEDURE KWRITE (RO TEXT: STRING;
 RO KNOB_NUMBER: INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'WRKNB';

PROCEDURE LGREAD (RO BUTTON_NUMBER: INTEGER;
 WO LABEL: STRING;
 WO ERROR_CODE: INTEGER)
 ENTRY 'REDLG';

PROCEDURE PIXIN (RO LUN: INTEGER;
 WO PICTURE: ROW[L..H:INTEGER] OF INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'INPIX';

PROCEDURE PIXOUT (RO LUN: INTEGER;
 RO PICTURE: ROW[L..H:INTEGER] OF INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'OUIPIX';

PROCEDURE SDMIN (RO LUN: INTEGER;
 WO SYMBOLS: ROW[L..H:INTEGER] OF INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'INSDM';

PROCEDURE SDMOUT (RO LUN: INTEGER;
 RO SYMBOLS: ROW[L..H:INTEGER] OF INTEGER;
 WO ERROR_CODE: INTEGER)
 ENTRY 'OUSDM';

FUNCTION TVASK (RO TV_PART: TV_WINDOW;
 WO ERROR_CODE: INTEGER) REAL
 ENTRY 'ASKTV';

FUNCTION TVEDIT (RO TV_PART: TV_WINDOW;
 WO ERROR_CODE: INTEGER) STRING[64]
 ENTRY 'EDITV';

```
FUNCTION TVREAD (RO LUN: INTEGER;  
                RO LINE: INTEGER;  
                WO ERROR_CODE: INTEGER) STRING[64]  
                ENTRY 'REDTV';
```

```
(* 2.3 END OF MIP_TOOLS *)  
(* ===== *)
```

```
END MIP_TOOLS.
```

```
(*****  
(*  
(* 3 - SPECIAL GFA AND BRIGHT-UP FACILITIES FOR MIP (GFA_TOOLS) *)  
(*  
(* GFA AND BRIGHT-UP TOOLS FOR MIP CLASSE PROGRAMS *)  
(*  
(*****
```

MODULE PROVIDING GFA_TOOLS AS LIBRARY:

PROVIDE ALL WITH

```
(* 3.1 CONSTANT AND TYPE DEFINITION *)  
(* ===== *)
```

TYPE BRIGHT_MODE: (BRIGHT_OFF,BRIGHT_ON);

TYPE GFA_BR_MODE: (BR_OFF,BR_ONE,BR_TWO,BR_THREE);

(* 3.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(*
FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
*)

PROCEDURE SET_BRITRG (RO VALUE: REAL;
RO SCOPE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'SBRTG';

FUNCTION READ_BRITRG (RO SCOPE: INTEGER;
WO ERROR_CODE: INTEGER) REAL
ENTRY 'RBRTG';

PROCEDURE GFA (RO THIS_GFA: INTEGER;
RO GFA_DATA: ROW[L..H:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'ICGFA';

PROCEDURE SET_GFABTG (RO VALUE: INTEGER;
RO GFA, MARQUEUR: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'SGFBG';

FUNCTION READ_GFABTG (RO GFA: INTEGER;
RO MARQUEUR: INTEGER;
WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'RGFBG';

PROCEDURE GFAMOD (RO GFA: INTEGER;
RO CLOCK: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'GFAMD';

PROCEDURE SET_GFATRG (RO VALUE: REAL; (* CARDINAL *)
RO GFA: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'SGFTG';

FUNCTION READ_GFATRG (RO GFA: INTEGER;
WO ERROR_CODE: INTEGER) REAL (* CARDINAL *)
ENTRY 'RGFTG';

PROCEDURE MODBRI (RO MODE: BRIGHT_MODE;
RO SCOPE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'BRMOD';

PROCEDURE MOGFAB (RO MODE: GFA_BR_MODE;
RO GFA: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'BGFAM';

```
PROCEDURE MPXBRI      (RO PULSA: INTEGER;  
                      RO SCOPE: INTEGER;  
                      RO SIGNAL_TYPE: INTEGER;  
                      WO ERROR_CODE: INTEGER)  
                      ENTRY 'BRMPX';
```

```
PROCEDURE MPXGFA      (RO PULSA: INTEGER;  
                      RO SCOPE: INTEGER;  
                      RO SIGNAL_TYPE: INTEGER;  
                      WO ERROR_CODE: INTEGER)  
                      ENTRY 'GFMPX';
```

(* 3.3 END OF GFA_TOOLS *)
(* ===== *)

END GFA_TOOLS.

```
(*****  
(*  
(* 4 - DEDICATED SIP CONSOLE FACILITIES ( SIP_TOOLS )  
(*  
(* SPECIAL CONSOLE FACILITIES FOR SIP  
(*  
(*****
```

MODULE PROVIDING SIP_TOOLS AS LIBRARY:

PROVIDE ALL WITH

```
(* 4.1 CONSTANT AND TYPE DEFINITION *)  
(* ===== *)
```

(* 4.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(*
FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
*)

FUNCTION AN1LUN (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'AN1LN';

FUNCTION AN2LUN (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'AN2LN';

FUNCTION ANAGR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'GRANA';

PROCEDURE ANALEG (RO BUTTON_NUMBER: INTEGER;
RO LABEL: STRING;
WO ERROR_CODE: INTEGER)
ENTRY 'LEGAN';

PROCEDURE DTMTRG (RO VALUE: REAL;
RO SCOPE: INTEGER;
RO DELAY_TYPE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'TGDTM';

PROCEDURE MPXTRG (RO PULSA: INTEGER;
RO SCOPE: INTEGER;
RO SIGNAL_TYPE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'TRMPX';

PROCEDURE PLSTRG (RO SCOPE: INTEGER;
RO LINE_NB: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'TRPLS';

PROCEDURE TIMODE (RO SCOPE: INTEGER;
RO TIME_BASE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'MODTI';

FUNCTION TIMGR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'GRTIM';

PROCEDURE SET_TIMTRG (RO VALUE: REAL; (* CARDINAL *)
RO SCOPE, DELAY_TYPE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'STMTG';

FUNCTION READ_TIMTRG (RO SCOPE, DELAY_TYPE: INTEGER;
WO ERROR_CODE: INTEGER) REAL (* CARDINAL *)
ENTRY 'RTMTG';

(* 4.3 END OF SIP_TOOLS *)
(* ===== *)

END SIP_TOOLS.

```
(*****  
(*  
(* 5 - NON INTERACTIVE PROGRAMS CONSOLE FACILITIES ( SRT_TOOLS ) *)  
(*  
(* CONSOLE FACILITIES FOR NON INTERACTIVE PROGRAMS *)  
(*  
(*****
```

MODULE PROVIDING SRT_TOOLS AS LIBRARY:

PROVIDE ALL WITH

```
(* 5.1 CONSTANT AND TYPE DEFINITION *)  
(* ===== *)
```

```
(* TARGET SOFTWARE-TRIGGER IDENTIFICATION FOR TRIG CALL *)  
(* ----- *)
```

```
TYPE TRIGGER:  
  (DUMMY_TRIGGER,          (* MUST NEVER BE USED *)  
   TRGTIP,                 (* TRIGGER TO TIP *)  
   TRGMIP,                 (* TRIGGER TO MIP *)  
   TRGVIP,                 (* TRIGGER TO VIP *)  
   TRGLIP,                 (* TRIGGER TO LIP *)  
   TRGSIP);               (* TRIGGER TO SIP *)
```

```
(* VIDEO SCREEN CONTROL CHARACTERS DEFINITION *)  
(* ----- *)
```

CONSTANT

```
  RED      : CHAR(0);      (* RED COLOUR *)  
  GREEN    : CHAR(1);      (* GREEN COLOUR *)  
  BLUE     : CHAR(2);      (* DARK BLUE COLOUR *)  
  WHITE    : CHAR(3);      (* WHITE COLOUR *)  
  LBLUE   : CHAR(4);      (* LIGHT BLUE COLOUR *)  
  MAGENT   : CHAR(5);      (* MAGENTA COLOUR *)  
  YELLOW   : CHAR(6);      (* YELLOW COLOUR *)  
  BLACK    : CHAR(7);      (* BLACK COLOUR *)  
  
  ERASE    : CHAR(8#14#);  (* ERASE AND RESET SCREEN *)  
  INVERT   : CHAR(8#17#);  (* SWAP BACKGROUND AND FOREGROUND COLOUR *)  
  BACKGR   : CHAR(8#20#);  (* BACKGROUND COLOUR INTO NEXT CHARACTER *)  
  DUMMY_CHAR : CHAR(8#33#); (* DUMMY CHARACTER (NON-PRINTABLE) *)
```

```
(* PROSCESS-EVENT TYPE DEFINITION FOR EVENT CALL *)  
(* ----- *)
```

```
TYPE  
MACHINE: (BOOSTER,PS,DUMMY_1,DUMMY_2);
```

```
PULSE: ( DUMMY_PULSE,  
PULSE_1,  
PULSE_2,  
PULSE_3,  
PULSE_4,  
BX_WBC,  
BX_STB,  
BX_RBI,  
PX_WTR,  
PX_RSC,  
PX_STC,  
PX_ELFT,  
PX_EPC,  
PULSE_13,  
PULSE_14,  
PULSE_15,  
PULSE_16);
```

```
(* INPUT/OUTPUT DIRECTION FOR RESERVATION *)  
(* ----- *)
```

```
TYPE IOFLAG:  
(INP,OUTP);
```


(* 5.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(*
FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
*)

FUNCTION BANDW (RO INDEX: INTEGER;
WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'WANDB';

FUNCTION COLOUR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'COLOR';

FUNCTION COLUMN (RO COLUMN_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER) STRING[2]
ENTRY 'COL';

FUNCTION COMPUTER_NUMBER (RO COMP_NB: COMPUTER) INTEGER
ENTRY 'CMPNB';

FUNCTION EVAL_STRING (RO EXPR: STRING;
WO ERROR_CODE: INTEGER) REAL
ENTRY 'XEVAL';

FUNCTION GRAPH (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'HPARG';

FUNCTION KSTATU (RO KNOB_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'STATK';

PROCEDURE KWRITE (RO TEXT: STRING;
RO KNOB_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'WRKNB';

FUNCTION LINE (RO LINE_NUMBER: INTEGER;
WO ERROR_CODE: INTEGER) STRING[2]
ENTRY 'LYNE';

PROCEDURE NEWS (RO TEXT: STRING;
WO ERROR_CODE: INTEGER)
ENTRY 'SWEN'; (* USES ICCI REMOTELY ON MCR *)

PROCEDURE PLS_TELEGRAM (RO MACH: MACHINE;
WO TELEGRAM: ROW[L..H:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER);

FUNCTION POSIT (RO X_VALUE,Y_VALUE: INTEGER;
WO ERROR_CODE: INTEGER) STRING[4]
ENTRY 'TISOP';

PROCEDURE TRIG (RO VALUE: INTEGER;
RO THIS_TRIGGER: TRIGGER;

```
      WO ERROR_CODE: INTEGER)
      ENTRY 'ITRIG';

PROCEDURE RELEAS      (RO LUN: INTEGER;
      RO IO_DIRECTION: IOFLAG;
      WO ERROR_CODE: INTEGER)
      ENTRY 'AELER';

PROCEDURE RESERV      (RO LUN: INTEGER;
      RO IO_DIRECTION: IOFLAG;
      WO ERROR_CODE: INTEGER)
      ENTRY 'RSRV';

PROCEDURE VIDEO      (RO LUN: INTEGER;
      RO TEXT: STRING;
      WO COCO: INTEGER)
      ENTRY 'IVIDE';

PROCEDURE WRESER      (RO LUN: INTEGER;
      RO IO_DIRECTION: IOFLAG;
      WO ERROR_CODE: INTEGER)
      ENTRY 'RESEW';
```

```
(* 5.3  END OF SRT_TOOLS *)  
(* ===== *)
```

```
END SRT_TOOLS.
```

```
(*****  
(*  
(* 6 - GENERAL GRAPHIC CONSOLE FACILITIES ( GRAPHIC_FUNCTIONS ) *)  
(*  
(* BASIC CONSOLE GRAPHIC FACILITIES *)  
(*  
(*****
```

MODULE PROVIDING GRAPHIC_FUNCTIONS AS LIBRARY:

USE TYPE CONSOLE_TYPES \$ TRIPLE_WORD;

PROVIDE ALL WITH

```
(* 6.1 CONSTANT AND TYPE DEFINITION *)  
(* ===== *)
```

```
(* OBJECT DEFINITION TYPES *)  
(* ----- *)
```

TYPE

```
OBJECT_ID: (OBJ0, OBJ1, OBJ2, OBJ3, OBJ4, OBJ5, OBJ6, OBJ7,  
            OBJ8, OBJ9, OBJ10, OBJ11, OBJ12, OBJ13, OBJ14, OBJ15,  
            OBJ16, OBJ17, OBJ18, OBJ19, OBJ20, OBJ21, OBJ22, OBJ23,  
            OBJ24, OBJ25, OBJ26, OBJ27, OBJ28, OBJ29, OBJ30, OBJ31);
```

```
OBJECT_CLASS: (CANVAS_OBJECT, VARIABLE_OBJECT,  
              XXXXXX_OBJECT, EDITABLE_OBJECT);
```

```
OBJECT_DEFN: RECORD  
    OBJECT_NUMBER: OBJECT_ID;  
    OBJECT_TYPE: OBJECT_CLASS;  
    ADDRESS: INTEGER;  
END RECORD;
```

```
(* OBJECT MODIFICATION MARKER *)  
(* ----- *)
```

```
TYPE TRIPLE: RECORD  
    TR: TRIPLE_WORD;  
END RECORD;
```

```
(* GRAPHIC COORDINATE *)  
(* ----- *)
```

```
TYPE CO_ORDINATE: RECORD  
    X_VAL, Y_VAL: REAL;  
END RECORD;
```

```
(* BAR PLOTS AND HISTOGRAMS *)  
(* ----- *)
```

```
TYPE STEPS: RECORD  
    INITIAL_VALUE, STEP: REAL;  
END RECORD;
```

```
(* AXISES *)  
(* ----- *)
```

```
TYPE AXIS_TICK: (DOWN,UP,LEFT,RIGHT);
```

```
  AXIS: RECORD  
    AT_VALUE: REAL;  
    START_VAL,STOP_VAL: REAL;  
    FIRST_TICK,TICK_STEP: REAL;  
    TICK_DIRECTION: AXIS_TICK;  
  END RECORD;
```

(* 6.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(*
FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
*)

PROCEDURE BLINK (RO OBJECT: OBJECT_ID;
WO ERROR_CODE: INTEGER)
ENTRY 'KNILB';

PROCEDURE CHNSTA (RO CHANNEL: INTEGER;
WO DATA: ROW[L..H:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'STCHN';

PROCEDURE CHSIZE (RO CHS: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'SIZCH';

PROCEDURE FLUSH (WO ERROR_CODE: INTEGER)
ENTRY 'HSULF';

PROCEDURE FUNSTA (WO DATA: ROW[L..H:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'STFUN';

PROCEDURE GAPPEA (WO ERROR_CODE: INTEGER)
ENTRY 'EPPAG';

FUNCTION GERMES (RO ERROR: INTEGER;
WO ERROR_CODE: INTEGER) STRING[80]
ENTRY 'EMREG';

PROCEDURE GMODE (RO MODE: INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'MODEG';

FUNCTION GRCHK (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'CHKGR';

FUNCTION GRERR (WO ERROR_CODE: INTEGER) INTEGER
ENTRY 'ERRGR';

PROCEDURE GWRITE (RO TEXT: STRING;
WO ERROR_CODE: INTEGER)
ENTRY 'WRITG';

PROCEDURE IMGRES (RO DATA: ROW[L..H:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER)
ENTRY 'RESIM';

PROCEDURE IMGSAV (WO DATA: ROW[L..H:INTEGER] OF INTEGER;
WO ERROR_CODE: INTEGER)

```

        ENTRY 'SAVIM';
PROCEDURE LISTOB      (WO DATA:      ROW[L..H:INTEGER] OF INTEGER;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'OBLIS';
PROCEDURE MARKER     (RO MARQUEUR: INTEGER;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'EKRAM';
PROCEDURE MOVE       (RO POINT: CO_ORDINATE;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'EVOM';
PROCEDURE OBJALC     (RO SYZE: INTEGER;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'ALOBJ';
PROCEDURE OBJCLS     (WO ERROR_CODE: INTEGER)
                      ENTRY 'CLOBJ';
PROCEDURE OBJECT     (RO THIS_OBJECT: OBJECT_DEFN;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'CEJBO';
PROCEDURE OBJERS     (RO OBJECT: OBJECT_ID;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'ERSOB';
PROCEDURE OBJEXT     (RO OBJECT: OBJECT_ID;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'EXTOB';
PROCEDURE OBJMDF     (RO OBJECT: OBJECT_ID;
                      RO TICKV: TRIPLE;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'MDFOB';
PROCEDURE OBJMOV     (RO OBJECT: OBJECT_ID;
                      RO DISPLACEMENT: CO_ORDINATE;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'MOVOB';
PROCEDURE OBJREF     (RO OBJECT: OBJECT_ID;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'REFOB';
PROCEDURE OBJRES     (RO OBJECT: OBJECT_ID;
                      RO DATA:      ROW[L..H:INTEGER] OF INTEGER;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'RESOB';
PROCEDURE OBJSAV     (RO OBJECT: OBJECT_ID;
                      WO DATA:      ROW[L..H:INTEGER] OF INTEGER;
                      WO ERROR_CODE: INTEGER)
                      ENTRY 'SAVOB';
FUNCTION  OBJSZ      (RO OBJECT: OBJECT_ID;

```

```

      WO ERROR_CODE: INTEGER) INTEGER
      ENTRY 'SIZOB';

FUNCTION OBJSPAC      WO ERROR_CODE: INTEGER) INTEGER
      ENTRY 'SPAOB';

PROCEDURE OBJSTA      (RO OBJECT: OBJECT_ID;
      RO CHANNEL: INTEGER;
      WO DATA:      ROW[L..H:INTEGER] OF INTEGER;
      WO ERROR_CODE: INTEGER)
      ENTRY 'STAOB';

PROCEDURE POINT      (RO WHERE: CO_ORDINATE;
      WO ERROR_CODE: INTEGER)
      ENTRY 'TNIOP';

PROCEDURE POST      (RO OBJECT: OBJECT_ID;
      WO ERROR_CODE: INTEGER)
      ENTRY 'TSOP';

PROCEDURE REDCUR      (WO WHERE: CO_ORDINATE;
      WO ERROR_CODE: INTEGER)
      ENTRY 'CURED';

PROCEDURE SCREEN      (RO X1,X2,Y1,Y2: REAL
      WO ERROR_CODE: INTEGER)
      ENTRY 'NERCS';

FUNCTION TICKV      (WO ERROR_CODE: INTEGER) TRIPLE
      ENTRY 'VKCIT';

PROCEDURE UBLINK      (RO OBJECT: OBJECT_ID;
      WO ERROR_CODE: INTEGER)
      ENTRY 'NILBU';

PROCEDURE UNPOST      (RO OBJECT: OBJECT_ID;
      WO ERROR_CODE: INTEGER)
      ENTRY 'SOPNU';

PROCEDURE VECTOR      (RO WHERE: CO_ORDINATE;
      WO ERROR_CODE: INTEGER)
      ENTRY 'OTCEV';

PROCEDURE XAXIS      (RO WHAT: AXIS;
      WO ERROR_CODE: INTEGER)
      ENTRY 'AXISX';

PROCEDURE WINDOW      (RO X1,X2,Y1,Y2: REAL;
      WO ERROR_CODE: INTEGER)
      ENTRY 'ODNIW';

PROCEDURE XBAR      (RO XARRAY:      ROW [L..H: INTEGER] OF REAL;
      RO YARRAY:      ROW[LO..HI:INTEGER] OF REAL;
      RO X_VALUE: REAL;
      WO ERROR_CODE: INTEGER)
      ENTRY 'BARX';

PROCEDURE XGRAF      (RO XARRAY:      ROW [L..H: INTEGER] OF REAL;

```



```

      RO VALUES: STEPS;
      WO ERROR_CODE: INTEGER)
      ENTRY 'GRAFX';

PROCEDURE XMARK (RO XARRAY:      ROW [L..H: INTEGER] OF REAL;
                RO VALUES: STEPS;
                WO ERROR_CODE: INTEGER)
                ENTRY 'MARKX';

PROCEDURE XYGRAF (RO XARRAY:      ROW [L..H: INTEGER] OF REAL;
                 RO YARRAY:      ROW[LO..HI:INTEGER] OF REAL;
                 WO ERROR_CODE: INTEGER)
                 ENTRY 'GRFXY';

PROCEDURE XYMARK (RO XARRAY:      ROW [L..H: INTEGER] OF REAL;
                 RO YARRAY:      ROW[LO..HI:INTEGER] OF REAL;
                 WO ERROR_CODE: INTEGER)
                 ENTRY 'MRKXY';

PROCEDURE YAXIS (RO WHAT: AXIS;
                WO ERROR_CODE: INTEGER)
                ENTRY 'AXISY';

PROCEDURE YBAR (RO YARRAY:      ROW [L..H: INTEGER] OF REAL;
               RO XARRAY:      ROW[LO..HI:INTEGER] OF REAL;
               RO Y_VALUE: REAL;
               WO ERROR_CODE: INTEGER)
               ENTRY 'BARY';

PROCEDURE YGRAF (RO YARRAY:      ROW [L..H: INTEGER] OF REAL;
                RO VALUES: STEPS;
                WO ERROR_CODE: INTEGER)
                ENTRY 'GRAFY';

PROCEDURE YMARK (RO YARRAY:      ROW [L..H: INTEGER] OF REAL;
                RO VALUES: STEPS;
                WO ERROR_CODE: INTEGER)
                ENTRY 'MARKY';

```

(* 6.3 END OF GRAPHIC_FUNCTIONS *)
(* ===== *)

END GRAPHIC_FUNCTIONS.

```
(*****  
(*  
(* 7 - DEDICATED MIP GRAPHIC FACILITIES ( MIP_GRAPHIC ) *)  
(*  
(* GRAPHIC CONSOLE FACILITIES FOR MIP INTERACTION *)  
(*  
(*****
```

MODULE PROVIDING MIP_GRAPHIC AS LIBRARY:

```
USE TYPE GRAPHIC_FUNCTIONS $ OBJECT_ID,  
GRAPHIC_FUNCTIONS $ CO_ORDINATE;
```

PROVIDE ALL WITH

```
(* 7.1 CONSTANT AND TYPE DEFINITION *)  
(* ===== *)
```

```
(* RECOGNITION WINDOW VISIBILITY *)  
(* ----- *)
```

```
TYPE ON_OFF: (OFF_WINDOW,ON_WINDOW);
```

(* 7.2 FUNCTIONS AND PROCEDURES IN ALPHANUMERICAL ORDER *)
(* ===== *)

(* FOR DOCUMENTATION REFERE TO PS/CO/NOTE 81-11
)

PROCEDURE DELAST (RO POINT1,POINT2: CO_ORDINATE;
WO ERROR_CODE: INTEGER)
ENTRY 'LASDE';

PROCEDURE ELASTI (RO POINT: CO_ORDINATE;
WO ERROR_CODE: INTEGER)
ENTRY 'LASTE';

PROCEDURE GERASE (WO ERROR_CODE: INTEGER)
ENTRY 'ASGER';

FUNCTION GRASK (RO OBJECT: OBJECT_ID;
WO ERROR_CODE: INTEGER) REAL
ENTRY 'ASKGR';

FUNCTION GREDIT (RO OBJECT: OBJECT_ID;
WO ERROR_CODE: INTEGER) STRING[80]
ENTRY 'EDIGR';

FUNCTION IDFOBJ (WO ERROR_CODE: INTEGER) OBJECT_ID
ENTRY 'OBIDF';

PROCEDURE MONOPL (WO ERROR_CODE: INTEGER)
ENTRY 'PONOM';

PROCEDURE OBJBAL (RO OBJECT: OBJECT_ID;
WO ERROR_CODE: INTEGER)
ENTRY 'BALOB';

PROCEDURE OBJIDF (RO OBJECT: OBJECT_ID;
RO X1,X2,Y1,Y2: REAL;
WO ERROR_CODE: INTEGER)
ENTRY 'IDFOB';

PROCEDURE OBJNID (RO OBJECT: OBJECT_ID;
WO ERROR_CODE: INTEGER)
ENTRY 'NIDOB';

PROCEDURE OBJWIN (RO OBJECT: OBJECT_ID;
RO WINDOW_STATUS: ON_OFF;
WO ERROR_CODE: INTEGER)
ENTRY 'WINOB';

(* 7.3 END OF MIP_GRAPHIC *)
(* ===== *)

END MIP_GRAPHIC.

(*****)
(* *)
(* 8 - LOAD-TIME (RT-LOADER) INFORMATIONS *)
(* *)
(* NOTES FOR TARGET CONSOLE COMPUTER SEGMENT-LOAD MODE-FILE *)
(* *)
(*****)

- (*
- 1 - THE LIBRARY FILE : (CONS-NEW)CONS-PPL-LIBR:BRF
MUST BE LOADED AFTER USER BRF FILES.
 - 2 - FIX-SYMBOLS 57 MUST BE DONE IN ORDER TO
USE SEGMENTS NAMES.
 - 3 - FOR ANY MODULE USED :
EXPLICITE SEARCH FOR SEGMENT CONF1 MUST BE DONE.
 - 4 - IF MODULES GRAPHIC_FUNCTIONS OR MIP_GRAPHIC ARE USED :
EXPLICITE SEARCH FOR SEGMENT GRAF1 MUST ALSO BE DONE.
 - 5 - IF MODULES GFA_TOOLS OR SIP_TOOLS ARE USED :
EXPLICITE SEARCH FOR SEGMENT CONF2 MUST ALSO BE DONE.

*)