

**Linguaggio PASCAL+
Manuale di Consultazione**

olivetti

Type

4 char = packed array [1..4] of char;

Param = core integer of

0: (arr = 4 char);

1: (long = long integer);

end.

intre and code

SKIP FFI = 3

skeep di 1 o più file Mark

WRITE FFI = 4

scrive 1 o più File Mark

SKIP RECORD = 5

skeep di uno o più record (avanti o indietro)

SET MODE = 7

setta velocità e bloccaggio

ERASE = 8

erare del nastro

OPEN = 9

setta bloccaggio default, setta banda velocità, rewind

CLOSE (REWIND) = 10

esegue un rewind (nastro a BOT)

SEARCH FOR = 11

cerca la prima posizione libera nel nastro.

Modalità di chiamata

reply := SystemFunction (FFIId, param. arr, code)

intre si ottiene facendo una Connect o una get a

/DEV/PTX

parametri a inter

~~SKIP FFI~~ SKIP FFI : rappresenta il numero di FFI da skippare (preceduto da segno negativo esce del skip back).

Write FFI : rappresenta il numero di FFI da scrivere.

SKIP RECORD : ~~skip~~ skip di uno o più record avanti o indietro a seconda del segno

SET MODE = YY XX dove YY = 25 e XX = bloccaggio
dove YY può variare da 4..16384

PREFAZIONE

Questo manuale e' una guida per programmatori ed analisti interessati allo sviluppo di programmi in PASCAL+.

Il PASCAL+ e' una estensione del PASCAL standard ISO. Questo manuale non presuppone una specifica conoscenza di linguaggi di programmazione.

Oltre ad essere un manuale di consultazione su argomenti specifici, puo' anche essere utilizzato come Guida Utente per la risoluzione di problemi di programmazione.

SOMMARIO

Il manuale e' diviso in cinque parti:

- La prima parte puo' essere considerata un Manuale introduttivo: dando una visione generale del linguaggio e trattando la filosofia del linguaggio e la strutturazione del programma. Essa deve essere letta dagli utenti che si accostano per la prima volta ad un linguaggio ad alto livello.
- La seconda parte puo' essere considerata un Manuale di Consultazione: descrive in dettaglio le caratteristiche del linguaggio dal punto di vista sintattico e semantico.
- La terza parte puo' essere considerata una Guida Utente: fornisce ulteriori consigli per scrivere programmi piu' complessi.
- La quarta parte tratta le estensioni del linguaggio, principalmente la programmazione concorrente e le librerie di risorse istanziabili.
- La quinta parte descrive in dettaglio le caratteristiche del linguaggio esteso, espandendo ulteriormente soggetti precedentemente trattati nella seconda parte.

RIFERIMENTI

In precedenza leggere ...

Introduzione al Sistema Operativo MOS - Code 4002120 F (vol. 2)

Per ulteriori informazioni leggere...

PASCAL+ - Preparazione ed Esecuzione dei Programmi - Code 4002470 J

Lista dei codici di errore con spiegazione sintetica

- 9 L'opzione di debug deve essere riportata prima della prima linea di programma
- 10 Troppe cifre
- 11 Cifra attesa dopo il . in numero reale
- 12 Overflow di intero
- 13 Cifra attesa ad esponente
- 14 Fine di linea incontrato nella costante di tipo string
- 15 Carattere illegale in input
- 16 Cifra ottale attesa
- 17 Incontrata inattesa fine di file (eof)
- 18 Linea di "include file" non corretta
- 19 Troppi livelli di inclusione di file
- 20 Simbolo illegale
- 21 Errore nel simple type
- 22 Errore nella parte di dichiarazione
- 23 Errore nella lista dei parametri
- 24 Errore nella costante
- 25 Errore nel tipo
- 26 Errore nella lista dei campi
- 27 Errore nel factor
- 28 Errore nella variabile
- 29 Atteso un identificatore
- 30 Atteso un integer
- 31 Atteso '('

32 Atteso ')'

 33 Atteso '['

 34 Atteso ']'

 35 Atteso ':'

 36 Atteso ';'

 37 Atteso '='

 38 Atteso ','

 39 Atteso '*'

 40 Atteso ':='

 41 'PROGRAM' atteso

 42 'OF' atteso

 43 'BEGIN' atteso

 44 'END' atteso

 45 'THEN' atteso

 46 'UNTIL' atteso

 47 'DO' atteso

 48 'TO'/'DOWNTO' atteso

 49 'FILE' atteso

 50 'IF' atteso

 51 '.' atteso

 52 'FROM' atteso

 100 Identificatore dichiarato due volte

 101 Identificatore non della classe appropriata

 102 Identificatore non dichiarato

 103 Segno non permesso

- 104 Numero atteso
- 105 Limite inferiore oltrepassa limite superiore
- 106 Tipi subrange incompatibili
- 107 Il tipo di costante deve essere integer
- 108 Il tipo non deve essere real
- 109 Il tagfield deve essere scalar o subrange
- 110 Tipo incompatibile col tipo del tagfield
- 111 Il tipo dell'index non deve essere real
- 112 Il tipo dell'index deve essere scalar o subrange
- 113 Il tipo dell'index non deve essere integer
- 114 Forward reference non soddisfatta
- 115 Tipo identificatore di forward reference nella dichiarazione delle variabili
- 116 Dichiarato forward : ripetizione della lista di parametri non permessa
- 117 Funzione dichiarata forward : ripetizione del result type non permessa
- 118 Il tipo della funzione deve essere scalar, subrange, o pointer
- 119 Parametro file value non permesso
- 120 Omesso il tipo della funzione nella dichiarazione della funzione
- 121 F-format solo per reali
- 122 Errore nel tipo di parametro della funzione standard
- 123 Errore nel tipo di parametro della procedura standard
- 124 Numero di parametri non in accordo con la dichiarazione
- 125 Sostituzione illegale di parametri
- 126 Tipo del risultato di funzione parametrica non in accordo con la dichiarazione

- 127 L' espressione non e' di tipo set
- 128 Test permessi solo su uguaglianze
- 129 Inclusione stretta non permessa
- 130 Confronto di file non permesso
- 131 Tipo di operando(i) non legale
- 132 Il tipo di operando deve essere boolean
- 133 Il tipo dell'elemento del set deve essere scalar o subrange
- 134 Tipi di elementi del set non compatibili
- 135 Il tipo di variabile non e' array o string
- 136 Il tipo dell'index non e' compatibile con la dichiarazione
- 137 Il tipo della variabile non e' record
- 138 Il tipo della variabile deve essere file o pointer
- 139 Tipo illegale per la variabile di controllo del loop
- 140 Tipo illegale di espressione
- 141 Assegnamento di file non permesso
- 142 Tipo della label non compatibile con l'espressione selezionante
- 143 I limiti del subrange devono essere scalari
- 144 Conflitto di tipi fra gli operandi
- 145 Assegnamento a funzione standard non permesso
- 146 Assegnamento a funzione permesso solo nel corpo della funzione
- 147 Nessun campo simile in questo record
- 148 Errore di tipo in lettura
- 149 Il parametro attuale deve essere una variabile
- 150 Case label definita piu' di una volta

- 151 Omissa la corrispondente dichiarazione di parte variante
- 152 Tagfield real o string non permessi
- 153 Una precedente dichiarazione non era stata resa forward
- 154 La sostituzione di proc/func standard non e' permessa
- 155 Label definita piu' di una volta
- 156 Label dichiarata piu' di una volta
- 157 Label non definita
- 158 Label non dichiarata
- 159 Value parameter atteso
- 160 Variante del record multidefinita
- 161 File non ammesse qui
- 162 Direttiva al compilatore sconosciuta (ne' 'external' ne' 'forward')
- 163 Una variabile non puo' essere un packed field
- 164 L'insieme dei reali non e' permesso
- 165 I campi di packed record e gli elementi di packed array non devono essere parametri "var" o argomenti usati per lettura
- 166 L'espressione case selector deve essere scalar o subrange
- 167 Il parametro attuale deve essere una costante
- 200 Il file nel programma non e' nel header ne' dichiarato come file globale
- 201 Identificatore esportato non dichiarato come routine globale
- 202 La procedura esportata da accesso a un tipo pointer
- 203 Il tipo del risultato della funzione esportata da accesso a un tipo pointer
- 204 Nessuna definizione per procedure o funzioni importate
- 205 Il descrittore da accesso a un tipo pointer

- 206 Restrizione nell'implementazione : gli export devono apparire prima della procedura
- 207 Forza il descrittore a fornire accesso ad un puntatore nel risolvere un descrittore dichiarato forward
- 208 Direttiva AT permessa solo per variabili globali
- 209 Una condizione non puo' essere passata come valore
- 210 Tentativo di fornire accesso a puntatore ad una procedura importata
- 211 Nessuna siffatta routine in questo monitor o modulo
- 212 Il tipo del risultato della procedura importata da accesso a puntatore
- 213 Sezioni di import ed export permesse solo a livello globale
- 214 Definizione di procedura che non e' importata
- 215 Non si puo' rendere forward o external una procedura importata
- 216 Non si puo' fornire un corpo ad una procedura importata
- 217 File globali solo per il programma, non per monitor o moduli
- 218 Costante set fuori dal range da 0 a 127
- 219 Array conformi non possono essere confrontate in questo modo
- 220 Le liste di argomenti di procedure formali e procedure attuali non combaciano
- 221 I tipi dei risultati di procedure formali e procedure attuali non combaciano
- 222 Non si puo' passare una procedura standard come parametro di procedura
- 223 Procedure e funzioni non possono essere scambiate l'una per l'altra
- 224 Stringa nome-file richiesta per reset e rewrite
- 225 Wait e signal richiedono condizioni come argomenti
- 225 Wait e signal disponibili solo nei monitor
- 226 Attesa un'array

- 227 Packed array non permesso in questo contesto
- 228 Tipi incompatibili con il tipo dell'index dell'array
- 229 Packed array richiesta in questo contesto
- 230 Elementi dell'array di tipi non compatibili
- 231 Atteso descrittore
- 232 Atteso puntatore
- 233 La procedura precedente contiene un goto ad una statement strutturata o un goto a una label inesistente
- 234 La variabile del for deve essere dichiarata localmente
- 235 Variabile di controllo di un for loop assegnata o letta all'interno del for loop
- 236 Chiamate permesse solo per procedure sottoposte a fork
- 237 Fork non permesso per procedure formali o standard
- 238 Il numero di un segmento per un indirizzo assoluto deve essere in 0..127
- 239 Usare := negli assegnamenti
- 240 Non possono essere copiate condizioni in statement di assegnamento
- 241 Attesa costante di tipo string per la versione
- 300 Troppi record scope nidificati
- 301 Limiti del set fuori del range
- 302 Limiti di stringa fuori del range
- 303 Troppe procedure/function nidificate
- 304 Troppi registri richiesti, espressione troppo complicata
- 305 Troppi registri di floating point richiesti
- 306 Troppi registri dispari richiesti
- 307 Non disponibile coppia di registri per DIV o MOD
- 1000 Errore del compilatore, riferire al system manager

* comandi spal *

a :	<nome variabile>	:	indirizzo di una variabile.
b :	block 'statement number	:	set breakpoint.
c :	" "	:	remove breakpoint.
d :	segment / offset := data	:	data in
e :		:	context change.
g :	segment / offset	:	word contents.
h :		:	elenco procedure attive
i :		:	
l :		:	lista unità di compilazione.
		:	lista breakpoint.
m :	<nome> := expression	:	modifica il contenuto di una variabile.
n :		:	elenco dei nomi visibili.
p :	<nome>	:	stampa il contenuto di variabili o costanti
q :	(quit)	:	uscita / end - debugger.
r :		:	run - (lancio del programma).
s :		:	step - run.
t :	{y} {o} {p}	:	visualizzazione unità prima della esecuzione
u :	{s} {o} {p} {o} {p}	:	passo di step: single ; procedure ; level.
w :		:	stampa nome istruzione.

Compilazione:

ZPC '-r -d' FileName

Esecuzione:

SPD

- sorgente .macol
- symbol [.sym]
- Exec [.out]

Key Name	WITHOUT SHIFT KEY PRESSED			WITH SHIFT KEY PRESSED		
	FUNCTION	CODING	HEX. VALUE	FUNCTION	CODING	HEX. VALUE
F1/F9	F1	ESC C	1B 43	F9	ESC ;	1B 3B
F2/F10	F2	ESC B	1B 42	F10	ESC :	1B 3A
F3/F11	F3	ESC A	1B 41	F11	ESC 9	1B 39
F4/F12	F4	ESC @	1B 40	F12	ESC 8	1B 38
F5/F13	F5	ESC ?	1B 3F	F13	ESC 7	1B 37
F6/F14	F6	ESC >	1B 3E	F14	ESC 6	1B 36
F7/F15	F7	ESC =	1B 3D	F15	ESC 5	1B 35
F8/F16	F8	ESC <	1B 3C	F16	ESC 4	1B 34
EXIT	EXIT	ESC H	1B 48	EXIT	ESC /	1B 2F
SEND/ERASE	SEND	ESC M	1B 4D	ERASE	ESC [	1B 5B
CL.ER	BACKSPACE	-	00 0B	CLEAR ERROR	ESC Q	1B 51
SKIP	FIELD SKIP	ESC N	1B 4E	FIELD SKIP	ESC N	1B 4E
ENTER	LINE FEED	-	00 0A	LINE FEED	-	00 0A
P1	P1	ESC *	1B 2A	P1	ESC %	1B 25
P2	P2	ESC)	1B 29	P2	ESC \$	1B 24
P3	P3	ESC (	1B 28	P3	ESC #	1B 23
P4	P4	ESC '	1B 27	P4	ESC "	1B 22
P5	P5	ESC &	1B 26	P5	ESC !	1B 21
IC/IL	INSERT CHAR.	ESC P	1B 50	INSERT LINE	ESC J	1B 4A
DC/CL	DELETE CHAR.	ESC O	1B 4F	DELETE LINE	ESC I	1B 49
←	LEFT MARGIN	ESC W	1B 57	LEFT MARGIN	ESC W	1B 57
→	RIGHT MARGIN	ESC V	1B 56	RIGHT MARGIN	ESC V	1B 56

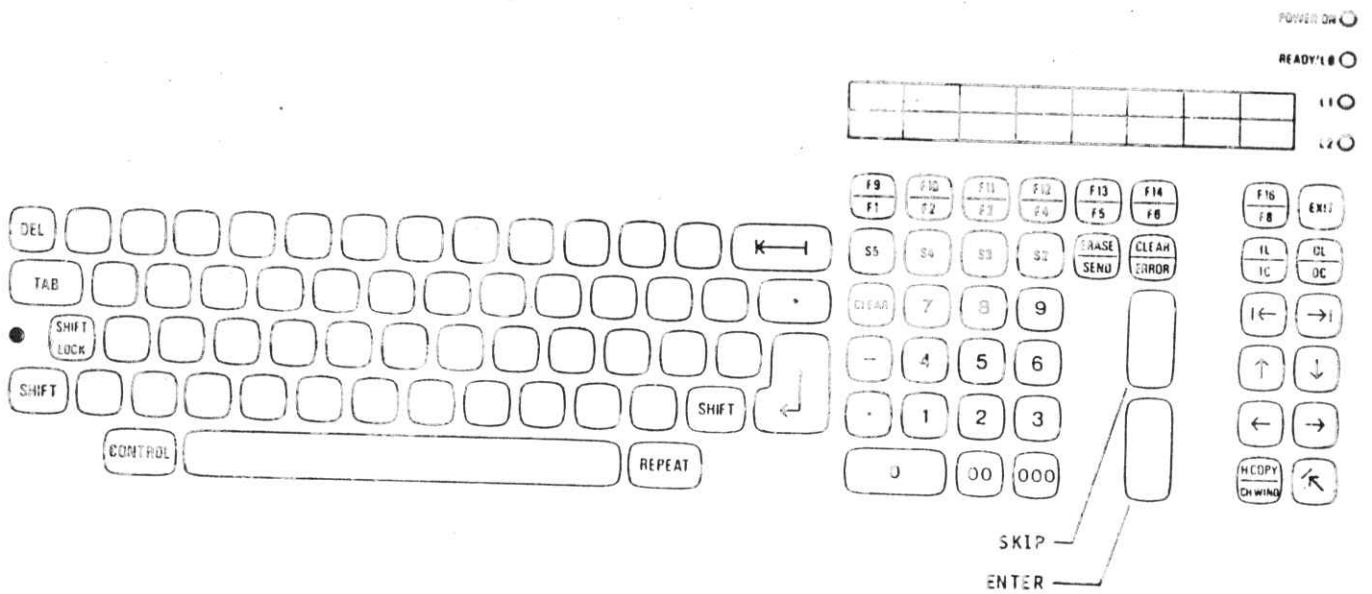
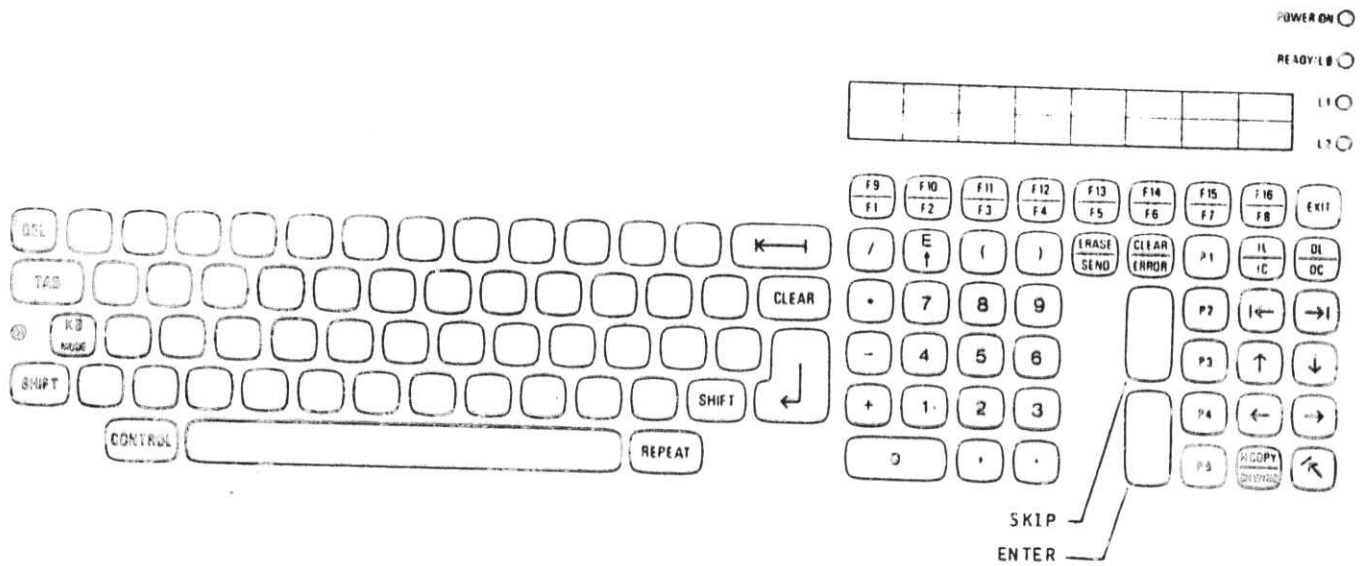


Fig. 6. 1 - Scientific Keyboard and DP + BC Keyboard

↑	MOVE UP	ESC R	1B 52	MOVE UP	ESC O	1B 30
↓	MOVE DOWN	ESC S	1B 53	MOVE DOWN	ESC 1	1B 31
←	MOVE LEFT	ESC U	1B 55	MOVE LEFT	ESC 3	1B 33
→	MOVE RIGHT	ESC T	1B 54	MOVE RIGHT	ESC 2	1B 32
CH.WIND/HC	CHANGE WINDOW	ESC L	1B 4C	HARD COPY	ESC K	1B 4B
↶	HOME	ESC Z	1B 5A	HOME	ESC Z	1B 5A

Tab. 6. 2 - Coding for the Function Keys of the Scientific Keyboard
Numeric and Function Section

Key Name	WITHOUT SHIFT KEY PRESSED			WITH SHIFT KEY PRESSED		
	FUNCTION	CODING	HEX. VALUE	FUNCTION	CODING	HEX. VALUE
DEL	DELETE	-	00 7F	DELETE	-	00 7F
TAB	TABULATION	-	00 09	TABULATION	-	00 09
←	BACKSPACE	-	00 08	BACKSPACE	-	00 08
CLEAR	CLEAR	ESC Y	1B 59	CLEAR	ESC Y	1B 59
↵	LINE FEED	-	00 0A	LINE FEED	-	00 0A

Tab. 6. 3 - Coding for the Function Keys of the Scientific Keyboard
Alphanumeric Section

Key Name	WITHOUT CONTROL KEY PRESSED			WITH CONTROL KEY PRESSED		
	FUNCTION	CODING	HEX. VALUE	FUNCTION	CODING	HEX. VALUE
F1/F9	F1	ESC C	1B 43	F9	ESC ;	1B 3B
F2/F10	F2	ESC B	1B 42	F10	ESC :	1B 3A
F3/F11	F3	ESC A	1B 41	F11	ESC 9	1B 39
F4/F12	F4	ESC @	1B 40	F12	ESC 8	1B 38
F5/F13	F5	ESC ?	1B 3F	F13	ESC 7	1B 37
F6/F14	F6	ESC >	1B 3E	F14	ESC 6	1B 36
F8/F16	F8	ESC <	1B 3C	F16	ESC 4	1B 34
EXIT	EXIT	ESC X	1B 48	EXIT	ESC /	1B 2F
S5	S5	ESC D	1B 44	S5	ESC +	1B 2B
S4	S4	ESC E	1B 45	S4	ESC ,	1B 2C
S3	S3	ESC F	1B 46	S3	ESC -	1B 2D
S2	S2	ESC G	1B 47	S2	ESC .	1B 2E
SEND/ERASE	SEND	ESC M	1B 4D	ERASE	ESC [	1B 5B
CL.ER	BACKSPACE	-	00 08	CLEAR ERROR	ESC Q	1B 51
IC/IL	INSERT CHR.	ESC P	1B 50	INSERT LINE	ESC J	1B 4A
DC/DL	DELETE CHR.	ESC O	1B 4F	DELETE LINE	ESC I	1B 49
CLEAR	CLEAR	ESC Y	1B 59	CLEAR	ESC Y	1B 59
SKIP	FIELD SKIP	ESC N	1B 4E	FIELD SKIP	ESC N	1B 4E
ENTER	LINE FEED	-	00 0A	LINE FEED	-	00 0A
	LEFT MARGIN	ESC W	1B 57	LEFT MARGIN	ESC W	1B 57
	RIGHT MARGIN	ESC V	1B 56	RIGHT MARGIN	ESC V	1B 56
	MOVE UP	ESC R	1B 52	MOVE UP	ESC O	1B 30
	MOVE DOWN	ESC S	1B 53	MOVE DOWN	ESC 1	1B 31

Example

```
#include 'systypes.i'
type dataItem = char; (* this type is user-defined *)
#include 'term_t.i'
#include 'term_p.i'

var wsId      : T_systemId;
    wsWriteInfo : T_wsWriteInfo;
    reply      : T_reply;

(* type
   T_physSize = ( S520, S1000, S2000 );
   T_crtMode = ( HIDE_NORMAL, HIDE_REVERSE, NOHIDE_NORMAL, NOHIDE_REVERSE );
   T_defineKey = record
       section : ( ALPHA, FUNCT );
       code    : char
   end;
   T_wsWriteInfo = record
       physSize : T_physSize;
       crtMode  : T_crtMode;
       processId : T_systemId;
       code      : integer;
       breakKey  : T_defineKey;
       changeKey : T_defineKey
   end;

   function WriteInfoWs( wsId : T_systemId;
                        var wsWriteInfo : T_wsWriteInfo ) : T_reply;

   external; *)

begin
.
.
with wsWriteInfo do
begin
    physSize := S520;
    crtMode  := HIDE_NORMAL;
    processId := (* process identifier *);
    code := 1;
    breakKey.section := ALPHA;
    breakKey.code := 'A';
    changeKey.section := ALPHA;
    changeKey.code := 'B'
end;
reply := WriteInfoWs ( wsId, wsWriteInfo );
.
.
end.
```

←	MOVE LEFT	ESC U	1B 33	MOVE LEFT	ESC 3	1B 33
→	MOVE RIGHT	ESC T	1B 54	MOVE RIGHT	ESC 2	1B 32
CH.WIND/HC	CHANGE WINDOW	ESC L	1B 4C	HARD COPY	ESC K	1B 4B
↵	HOME	ESC Z	1B 5A	HOME	ESC Z	1B 5A

Tab. 6. 4 - Coding for the Function Keys of the DP Keyboard
Numeric and Function Section

Key Name	WITHOUT CONTROL KEY PRESSED			WITH CONTROL KEY PRESSED		
	FUNCTION	CODING	HEX. VALUE	FUNCTION	CODING	HEX. VALUE
DEL	DELETE	-	00 7F	DELETE	-	00 7F
TAB	TABULATION	-	00 09	TABULATION	-	00 09
←	BACKSPACE	-	00 08	BACKSPACE	-	00 08
* (1)	F7	ESC =	1B 3D	F15	ESC 5	1B 35
↵	LINE FEED	-	00 0A	LINE FEED	-	00 0A

Tab. 6. 5 - Coding for the Function Keys of the DP Keyboard
Alphanumeric Section

(1) the second level of this key is obtained by pressing the SHIFT key instead of the CONTROL key.

Erase	ESC "row" "column"	1B 5F	All the characters in the data buffer from the active position up to and including the position pointed to by "row" "column" are substituted by null characters. See note 2.
Erase Line	ESC Y	1B 59	The screen is erased from the active position up to the end of the current row.
Erase Screen	ESC X	1B 58	The screen is cleared from the active position up to the end of the screen.
Form Feed	FF	0C	A line feed is performed.
Home	ESC Z	1B 5A	The active position of the cursor is set to row one, column one.
Horizontal Tab	HT	09	The active position is set to the next eighth position of the current row.
Move Down	ESC S	1B 53	The cursor is moved one step down in wrap around mode.
Move Left	ESC U	1B 55	The cursor is moved one step left in wrap around mode.
Move Right	ESC T	1B 54	The cursor is moved one step right in wrap around mode.
Move Up	ESC R	1B 52	The cursor is moved one step up in wrap around mode.
New line	LF	0A	If the active position is at the end of the scrolling region and the scroll is set, a scroll is performed and the active position is set to the first position of the line. Otherwise the active

* CARATTERI DI CONTROLLO VIDEO *

All the tty like commands are alphsbetically grouped and described in the following table:

TTY LIKE COMMANDS			
Command			Description
Its Name	Its Physical Name	Its hexadecimal value	
Back Space	BS	08	It erases the character preceding the cursor and the cursor is moved one space left until the first position of the screen. At that point, no action is performed on the cursor.
Bell	BEL	07	The bell is activated.
Carriage Return	CR	0D	The active position is set at the beginning of the current line.
Clear	ESC [	1B 5B	The active position is set to Home. The data buffer and the attributes buffer are cleared. The scrolling region is set to the entire screen. The scroll is enabled (data and attributes).
Cntl led	ESC e "led command"	1B 65	The "led command" identifies the led itself and the action to be performed on the led. See note 1.
Enter Alpha	ESC f	1B 5E	The terminal is set to alpha mode: the following items will be interpreted as defined for the alpha vocabulary.
Enter Graphics	ESC g	1B 67	The terminal is set to graphic mode: the following items will be interpreted as items of the graphic vocabulary.

be written in the attribute buffer, starting from the active position up to the position indicated by "row" "column". See note 7.

Write ESC f 18 66
Background "attribute"
"row" "column"

"attribute" specifies which colors are to be set for the background from the active position up to and including the position "row" "column". See note 9.

Write ESC d "row" 18 64
Data "column"

All the "items" following this command are interpreted as characters to be written in the data buffer of the screen. Using this command, it is possible to display data in transparent mode. See note 8.

notes

- 1 The led command is made up of the following values:
 - 00: led READY on
 - 01: led READY off
 - 02: led 1 on
 - 03: led 1 off
 - 04: led 2 on
 - 05: led 2 off.
- 2 If "row" "column" points to a position preceding the one pointed to by the active position, then just the character pointed to by the active position is cleared. If the parameter "row" is equal to zero, the parameter "column" is then interpreted as the number of positions to erase. If the parameter "column" is equal to zero, no action is performed. An attempt to bypass the end of the screen has no effect.

position is moved to the first position of the next line up to the end of the screen. An attempt to bypass the end of the screen has no effect.

See note 3, substituting "set" with "reset".

The scroll mode is reset.

The row pointed to by the active position, the row indicated by the parameter "row" and the number of columns declared in "columns" define a section of the screen which is scrolled up one line by this command. See note 4.

The active position is set to the position corresponding to "row" "column". See note 10.

See note 3.

The scrolling region is set from the row corresponding to the active position up to and including the parameter row. See note 5.

This command enables the scroll every time a line feed is executed. See note 6.

A line feed is performed.

Every "item" in the string following this command is interpreted as a byte to

Reset Attribute	ESC a "attribute" "row" "column"	1B 61
Reset Scroll	ESC A	1B 5E
Scroll Up	ESC W "row" "columns" "scroll pattern"	1B 57
Set Active Position	ESC / "row" "column"	1B 5C
Set Attribute	ESC ' "attribute" "row" "column"	1B 60
Set Region	ESC c "row"	1B 63
Set Scroll	ESC] "scroll pattern"	1B 5D
Vertical Tab	VT	0B
Write Attribute	ESC b "row" "column"	1B 62

- 5 The command has no effect if "row" points to a position preceding the one pointed to by the active position. If "row" points to the position pointed to by the active position, the scrolling region is just one line long. The scrolling region is scrolled if scroll is set and if:

- a character is written in the last position of the scrolling region or
- a line feed command is performed when the active position is on the last line of the scrolling region.

When the scrolling region scrolls, a new line is inserted at the end of the scrolling region and the first one is lost.

- 6 This command enables the scroll either every time a line feed is executed with the active position on the last line of the scrolling region or every time a character is written in the last position of the scrolling region.

"Scroll pattern" specifies which attributes must be scrolled (see note 3 for the "attribute" mask):

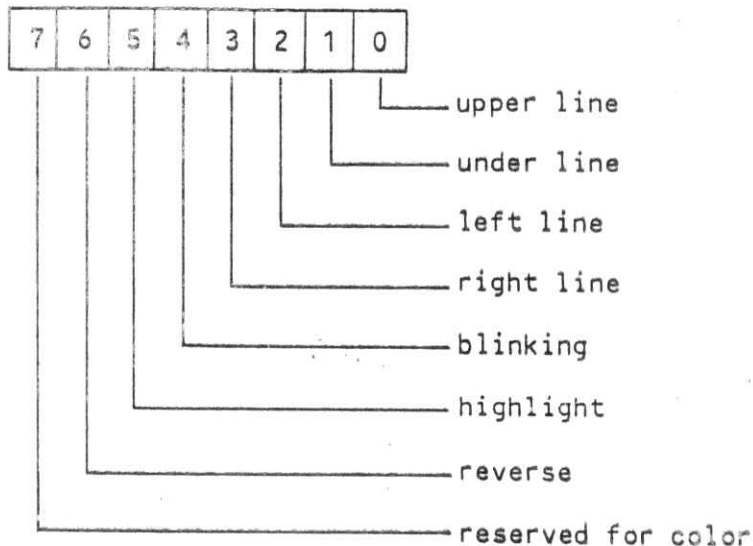
- 00 hex means no attributes;
- FF hex means all attributes.

- 7 The active position is not affected. If "row" "column" points to a position preceding the one pointed to by the active position, just the position pointed to by the active position is modified. If the parameter "row" is equal to zero, then the parameter "column" is interpreted as the number of items to be horizontally written. If the parameter "row" is less than zero, then the parameter "column" is interpreted as the number of "items" to be vertically written.

- 8 Every "item" is interpreted as a character. If the parameter "row" is equal to zero, the parameter "column" is interpreted as the number of items which are to be written from left to right. If the parameter "row" is less than zero, the parameter "column" is interpreted as the number of items which are to be written from right to left. In the latter case the scrolling region scrolls when a character is written in the first position of the last row.

- 3 "Attribute" is a mask whose bits specify which attributes are to be set from the active position up to and including the position pointed to by "row" "column".

The "attribute" mask is a byte within which the bits have the following meanings:



If "row" "column" points to a position preceding the one pointed to by the active position, then the specified attributes are set in the position pointed to by the active position. If the parameter "row" is equal to zero, the parameter "column" is interpreted as the number of positions to be horizontally set. If the parameter "row" is less than zero, then the parameter "column" is interpreted as the number of positions to be vertically set. When "row" is less than zero, setting the first column of the screen to reverse results in the loss of any Right Line attributes for the last column.

- 4 The "scroll pattern" also identifies which attributes must be scrolled (see note 3 for the "attribute" mask):
- 00 hex means no attribute
 - FF hex means all attributes.

The content of "row" is lost. The command is always executed, even if the scroll is reset ("reset scroll"). The value of "row" must be less than or equal to the position pointed to by the active position. If it is equal, it scrolls one line.

- 9 If "row" "column" points to a position which precedes the one pointed to by the active position, then the specified attributes are set in the position pointed to by the active position. If the parameter "row" is equal to zero, then the parameter "column" is interpreted as the number of positions to be horizontally set. If the parameter "row" is less than zero, then the parameter "column" is interpreted as the number of positions to be vertically set.
"Attribute" is a mask described in note 3.
- 10 If either "row" or "column" is set outside the window, the cursor is switched off. The cursor is switched on if either "row" or "column" is set to zero.

Example

```

#include 'systypes.i'
type dataItem = char; (* this type is user-defined *);
#include 'term_t.i'
#include 'term_p.i'

var wsId      : T_systemId;
    data      : packed array [1..20] of char;
    startIndex : integer;
    dataSize  : integer;
    writeMode : T_tmWriteMode;
    reply     : T_reply;

(* type T_tmWriteMode = ( NORMAL, PROMPT );
    dataItem = char;

function WriteTm(
    var wsId : T_systemId;
    data : packed array [min..max : integer]
        of dataItem;
    startIndex : integer;
    dataSize : integer;
    writeMode : T_tmWriteMode ) : T_reply;

external; *)

begin
.
.
reply := WriteTm ( wsId, data, 1, 15, NORMAL );
.
end.
```

Type

4 char = packed array [1..4] of char;

param = case integer of

0: (arr : 4 char);

1: (long : long integer);

end.

name and code

SKIP FT = 3

WRITE FT = 4

SKIP RECORD = 5

SET MOD = 7

ERASE = 8

OPEN = 9

CLOSE (number) = 10

SEARCH MOD = 11

skip di 1 o più file mark

write 1 o più file mark

skip di uno o più record (avanti o indietro)

setta delimito e bloccaggio

erare del modulo

setta bloccaggio default, setta banda delimito, reset

erare un record (modulo a BOT)

cerca la prima posizione libera nel
modulo

Modulo di Chiamata

reply := SystemFunction (modId, param, arr, code)

intesa si ottiene facendo una connect o un get a

/DEV/PTX

parametri a int

~~SKIP~~ SKIP FT : rappresenta il numero di FT da skipare (preceduto da segno negativo esce dal skip block).

WRITE FT : rappresenta il numero di FT da scrivere

SKIP RECORD : skip di uno o più record avanti o indietro a seconda del segno

SET MOD = YY XX dove YY = 25 e XX = bloccaggio
1..16 che può variare da 1..16384

Type

4 char = packed array [1..4] of char;

long = core integer of
 0 (an 4 char);
 1 (long - long integer);
 end.

name and code

SKIP FN	= 3	skip di 1 o più file Mark
WRITE FN	= 4	write 1 o più File Mark
SKIP RECORD	= 5	skip di uno o più record (avanti o indietro)
SET MODE	= 7	setta sebche e bloccaggio
ERASE	= 8	erase del mark
OPEN	= 9	setta bloccaggio default, setta bano sebche, revind
CLOSE (REWIND)	= 10	eroge un rewind (mark a BOT)
SEARCH EOD	= 11	cerca la prima porzione libera nel mark

Modello di chiamata

reply := SystemFunction (PID, param. ar, code)

intuita si ottiene facendo una Connect o una open a

/DEV/PTX

parametri a inter

~~SKIP~~ SKIP FN : rappresenta il numero di FN da skippare (preceduto da segno negativo esce del skip back).

Write FN : rappresenta il numero di FN da scrivere

SKIP RECORD : skip di uno o più record avanti o indietro a conto del segno

SET MODE = YYXX dove YY = 25 e XX = bloccaggio
 1...16384

Type

4 char = packed array [1..4] of char;

param = case integer of

0: (an 4 char);

1: (long - long integer);

end.

intu and code

SKEEP FM	= 3	skip di 1 o più file mark
WRITE FM	= 4	write 1 o più File Mark
SKEEP RECORD	= 5	skip di uno o più record (avanti o indietro)
SET MODA	= 7	setta delcche e bloccaggio
ERASE	= 8	erore del mark
OPEN	= 9	setta bloccaggio default, setta banda delcche, rewind
CLOSE (REWINO)	= 10	esegue un rewind (mark a BOT)
SEARCH EOD	= 11	cerca la prima porzione libera nel mark

Modello di chiamata

reply = SystemFunction (pid, param, code)

intu si ottiene facendo una Connect o una open a

/DEV/PTX

parametri a intu

~~SKEEP~~ SKEEP FM : rappresenta il numero di FM da skipare (preceduto da segno negativo esce del skip back).

Write FM : rappresenta il numero di FM da scrivere

SKEEP RECORD : skip di uno o più record avanti o indietro a conto del segno

SETMOD 2 = yy xx dove yy = 25 e xx = bloccaggio
1...16384

- SYS. LIB -

000000	FF 01 00 00 01 C0 00 00 1F BC 00 05 43 4F 42 4F	COBO
000010	4C 00 00 87 FC 01 08 5A 52 54 5F 44 49 46 46 00	L.....ZRT_DIFF.
000020	00 87 FC 01 06 5A 52 54 5F 49 4E 00 00 87 FC 01	ZRT_IN.....
000030	09 5A 52 54 5F 49 4E 54 45 52 00 00 87 FC 01 09	.ZRT_INTER.....
000040	5A 52 54 5F 4E 55 4C 4C 53 00 00 87 FC 01 09 5A	ZRT_NULLS.....Z
000050	52 54 5F 53 45 54 45 51 00 00 87 FC 01 09 5A 52	RT_SETEQ.....ZR
000060	54 5F 53 45 54 47 45 00 00 87 FC 01 09 5A 52 54	T_SETGE.....ZRT
000070	5F 53 45 54 4C 45 00 00 87 FC 01 09 5A 52 54 5F	_SETLE.....ZRT_
000080	53 45 54 4E 45 00 00 87 FC 01 08 5A 52 54 5F 53	SETNE.....ZRT_S
000090	45 54 52 00 00 87 FC 01 08 5A 52 54 5F 53 53 45	ETR.....ZRT_SSE
0000A0	54 00 00 87 FC 01 09 5A 52 54 5F 55 4E 49 4F 4E	T.....ZRT_UNION
0000B0	00 00 1F BC 01 05 61 5F 61 64 64 00 00 1F BC 01	a_add.....
0000C0	06 61 5F 63 6F 6D 70 00 00 1F BC 01 08 61 5F 64	.a_comp.....a_d
0000D0	66 6C 6F 61 74 00 00 1F BC 01 05 61 5F 64 69 76	float.....a_div
0000E0	00 00 1F BC 01 08 61 5F 64 72 6F 75 6E 64 00 00	a_dround..
0000F0	1F BC 01 08 61 5F 64 74 72 75 6E 63 00 00 1F BC	a_dtrunc....

/IPL/DPC/COMMON/LIB/SYS.LIB MIN 000000 MAX 00BB7F

001500	00 24 3E 01 0B 73 74 64 65 6E 76 5F 71 75 69 74	.\$>..stdenv_quit
001510	00 00 24 3E 01 0C 73 74 64 65 6E 76 5F 72 74 65	..\$>..stdenv_rte
001520	72 72 00 00 24 3E 01 11 73 74 64 65 6E 76 5F 72	rr..\$>..stdenv_r
001530	74 70 68 61 6E 64 6C 65 72 00 00 24 3E 01 12 73	tphandler..\$>..s
001540	74 64 65 6E 76 5F 73 68 6F 72 74 65 70 69 6C 6F	tdenv_shortepilo
001550	67 00 00 24 3E 01 12 73 74 64 65 6E 76 5F 73 68	g..\$>..stdenv_sh
001560	6F 72 74 70 72 6F 6C 6F 67 00 00 3A 6E 01 0C 73	ortprolog...nl.s
001570	79 73 74 65 6D 66 75 6E 63 74 69 00 00 33 CB 00	ystemfuncti.....
001580	05 74 61 62 6C 65 00 00 93 CB 01 0A 74 61 62 6C	.table.....tabl
001590	65 5F 67 72 69 64 00 00 37 30 00 04 74 65 72 6D	e_grid..70..term
0015A0	00 00 5D 40 01 0C 74 72 61 6E 73 66 6F 72 6D 6B	..e\$.transformk
0015B0	65 79 00 00 6B B9 01 0C 74 72 61 6E 73 66 6F 72	ey..k...transfor
0015C0	6D 70 6F 73 00 00 6D 6F 01 10 74 72 68 64 6C 5F	mpos...uo...trhdl_
0015D0	63 6C 6F 73 65 74 72 61 63 65 00 00 6D 6F 01 10	closetrace..uo..
0015E0	74 72 68 64 6C 5F 67 69 76 65 74 61 62 6C 65 73	trhdl_givetables
0015F0	00 00 6D 6F 01 0F 74 72 68 64 6C 5F 6F 70 65 6E	..uo...trhdl_open

/IPL/DPC/COMMON/LIB/SYS.LIB MIN 000000 MAX 00BB7F

IPL / DPC / COMMON / LIB / SYS.LIB

**CODICE DI
RITORNO**

SIGNIFICATO

0	0	Esecuzione corretta
1	...	Il carattere di destra contiene l'identificazione della chiave di terminazione del formato digitata dall'operatore
2	...	Il carattere di destra contiene l'identificazione della chiave di interruzione digitata dall'operatore
3	0	Il valore di timeout e' stato superato
3	1	Fine pagina
4	0	Errore non recuperabile
4	1	Formato non trovato
4	2	Parametro errato
4	4	Sequenza non valida
4	5	Non c'e' spazio per i dati
4	6	Non esiste unita' periferica per il formato
4	8	Sottoformato non trovato
4	9	Non c'e' spazio sull'unita' periferica
4	10	Troppi formati
4	11	Identificatore di formato non valido
4	12	L'area di rilocalazione per la stampante (PRR) e' totalmente utilizzata
4	13	Input-output non definito
4	14	Identificatore di posto di lavoro non trovato

TABELLA DEI CODICI DI RITORNO

Nota

I codici di ritorno sono composti da due caratteri, ad eccezione di quelli dell'interfaccia FL10 che ne hanno uno solo.
Il carattere di sinistra indica il tipo di informazione, quello di destra definisce l'informazione stessa.
Il carattere di sinistra può assumere uno dei seguenti valori:

- 0 : operazione corretta
- 1 : fine input sul sottoformato
- 2 : interruzione dell'input sul sottoformato
- 3 : messaggi VISA di avvertimento
- 4 : errore a livello formato
- 5 : errore a livello programma di validazione
- 6 : errore a livello formato su unità periferica
- 7 : errore a livello ambiente VISA
- 8 : fase di input interrotta dall'elaborazione su un campo

I codici di ritorno FL10 sono descritti a fine tabella.

CODICE DI
RITORNO

SIGNIFICATO

4	15	Dimensioni dello schermo non corrette
4	16	Lo schermo non e' multivalente
4	17	Operazione non valida
4	18	Banconote nel contenitore di consegna
4	19	Operazione di consegna di banconote fallita
5	...	Questi errori sono descritti nel manuale "TFORM-VPL - Strumenti per la Preparazione dei Formati - Manuale di Consultazine"
6	0	Errore non recuperabile
6	1	Unita' spenta o fuori servizio
6	2	Unita' non disponibile
6	3	Risorsa occupata
6	4	Errore di trasmissione di linea
6	5	Documento non inserito correttamente (bourrage) Per il distributore automatico di banconote: cassetta vuota, cassetta non presente o in stato di lift-down.
6	6	Documento non presente
6	7	Coperchio stampante aperto
6	8	Unita' periferica commutata in locale
6	9	Errore di lettura o di controllo
6	10	Striscia vergine (non contiene dati)
6	11	Testina di stampa bloccata
6	12	Fine input: e' stato digitato /CTRL/ /D/
6	13	Badge inserito
6	14	Inserzione non valida

CODICE DI
RITORNO

SIGNIFICATO

6	15	Fine nastro
6	16	Terminale spento o terminale inattivo
6	17	Riscontrato errore di diagnostica durante l'accesso ad un governo periferico
6	18	Errore nella ricezione da una 'unita' periferica
6	19	Malfunzionamento hardware
6	20	Errore di sistema: malfunzionamento software
6	21	Overflow di tabella
6	22	Non c'e' piu' spazio sul disco
6	23	Errore non recuperabile: precede normalmente un guasto con conseguente caduta del sistema
6	24	Errore in segmento "read-only"
6	25	Errore nel segmento kern
6	26	Errore nella lunghezza segmento
6	27	Errore in segmento protetto
6	28	Errore xqt
6	29	Istruzione non valida
6	30	Una istruzione privilegiata non puo' essere eseguita in ambiente utente
6	31	Errore non recuperabile nel segmento
6	32	Non c'e' procedura: il driver di periferica non e' presente
6	33	No routing
6	34	I risultati possono non essere corretti
6	35	Operazione abortita
6	36	Record troppo lungo
6	37	Operazione non valida (la send-cassette non e' stata eseguita prima dell'operazione di distribuzione)
6	38	Identificatore non valido

CODICE DI
RITORNO

SIGNIFICATO

6	39	Il parametro richiesto dalla primitiva non e' corretto
6	40	Errore nella stringa dati e/o nei comandi inviati su video o su stampante. Questo errore puo' essere causato da errori del programma applicativo; fra i piu' comuni: . dispositivo non presente . il comando di alimentazione documento non e' presente sul dispositivo . attributo visivo di stampante non valido . spaziatura verticale e/o orizzontale non valida . la stampa avviene al di fuori del documento.
6	41	Violazione della protezione del flusso
6	42	Servizio non ancora implementato
6	43	Campo di validita' non corretto: il parametro deve essere locale o globale alla procedura
6	44	Stato non corretto di un processo o famiglia
6	45	Indice non corretto
6	46	Canale tra le famiglie distrutto
6	47	Canale tra le famiglie occupato
6	48	Il processo ha uno o piu' figli attivi
6	49	Oggetto di sistema non esistente
6	50	Non vi sono processi attivi in grado di eseguire la scrittura nel canale specificato
7	0	Errore non recuperabile
7	1	Terminale spento o inattivo
7	2	Riscontrato errore di diagnostica durante l'accesso ad un governo periferico

CODICE DI
RITORNO

SIGNIFICATO

7	3	Errore nella ricezione da una unita' periferica
7	4	Malfunzionamento hardware
7	5	Errore di sistema: malfunzionamento software
7	6	Overflow di tabella
7	7	Non c'e' piu' spazio sul disco
7	8	Errore non recuperabile: precede normalmente un guasto con conseguente caduta del sistema
7	9	Errore in segmento "read-only"
7	10	Errore nel segmento kern
7	11	Errore nella lunghezza segmento
7	12	Errore in segmento protetto
7	13	Errore xqt
7	14	Istruzione non valida
7	15	Una istruzione privilegiata non puo' essere eseguita in ambiente utente
7	16	Errore non recuperabile nel segmento
7	17	Non c'e' procedura: primitive di MONITOR non disponibili
7	18	No routing
7	19	I risultati possono non essere corretti
7	20	Operazione abortita
7	21	Record troppo lungo
7	22	Operazione non valida
7	23	Identificatore non valido
7	24	Il parametro richiesto dalla primitiva non e' corretto

**CODICE DI
RITORNO****SIGNIFICATO**

7	42	Processo o funzione interrotta
7	43	Processo terminato senza eseguire un "HALT" (il processo non esiste piu')
7	44	Processo VISA distrutto
7	45	Processo sospeso
7	46	Interruzione (punto di alt) riscontrata nel processo
7	47	L'area indirizzi non e' vuota (non esiste piu')
7	48	Overflow nella memoria fisica disponibile
7	49	Troppi processi concorrenti
7	50	Lunghezza eccessiva della matrice contenente la risposta da inviare al canale
7	51	L'intervallo di timeout e' scaduto
7	52	Record non accessibile: e' stata specificata la clausola LOCK
7	53	Unita' non pronta (non collegata, collegata non correttamente, spenta ecc.)
7	54	Pathname non corretto
7	55	Il nome del flusso, la directory o il volume esistono gia'
7	56	\$VISA, \$VMN o \$CONFVISA non trovato
7	57	La directory non puo' essere cancellata perche' non vuota
7	58	Fine flusso
7	59	Posizionamento fuori dei limiti di un flusso byte-stream o di un flusso sequenziale
7	60	Record non trovato
7	61	Il record esiste gia'
7	62	Chiave non trovata
7	63	Chiave gia' esistente.

CODICE DI
RITORNO

SIGNIFICATO

7	25	Errore nella stringa dati e/o nei comandi inviati su video o su stampante. Questo errore puo' essere causato da errori del programma applicativo; fra i piu' comuni: . dispositivo non presente . il comando di alimentazione documento non e' presente sul dispositivo . attributo visivo di stampante non valido . spaziatura verticale e/o orizzontale non valida . la stampa avviene al di fuori del documento.
7	26	Violazione della protezione del flusso
7	27	Servizio non ancora implementato
7	28	Campo di validita' non corretto: il parametro deve essere locale o globale alla procedura
7	29	Stato non corretto di un processo o famiglia
7	30	Indice non corretto
7	31	Canale tra le famiglie distrutto
7	32	Canale tra le famiglie occupato
7	33	Il processo ha uno o piu' figli attivi
7	34	Oggetto di sistema non esistente
7	35	Non vi sono processi attivi in grado di eseguire la scrittura nel canale specificato
7	36	Conflitto di memoria (conflitto tra segmenti di MONITOR o di VP)
7	37	Identificatore di flusso non valido
7	38	Tipo di segmento e attributi di segmento non compatibili
7	39	Formato di modulo caricabile non valido
7	40	Dimensioni del segmento non valide
7	41	Numero del segmento non valido

CODICE DI
RITORNO

SIGNIFICATO

8 XX L'operatore ha chiuso un campo dichiarato
 processabile con TFORM.
 XX identifica il campo come posizione del campo
 nell'ambito della struttura della tabella dei simboli
 del formato.

CODICI DI RITORNO FLIO

0 La routine e' stata eseguita correttamente

2 Parametro errato

5 La lunghezza dei dati eccede "field-data"

9 La lunghezza del campo eccede lo spazio disponibile
 sulla riga di schermo

255 Errore non recuperabile.

type

byte = -128..127;

T_class = (CORRECT, SYSTEM, DEBUG,
PMMWARNING, PMMERROR,
FSWARNING, FSERROR,
WSWARNING, WSERROR,
RUNTIME, USRCLASS,
LMWARNING, LMERROR,
THWARNING, THERROR,
FRWARNING, FRERROR);

T_correctCode = (OKAY);

T_sysErrCode = (TERMINALDOWN, DIAGNOSTICERROR,
RECEIVEERROR, HARDWAREERROR, SYSTEMERROR,
TABLEOVERFLOW, OUTOFDISKSPACE,
FATALERROR, ROFAULT,
KERNSEGFault, SEGLENFault, INHIBSEGFault,
XQTFault, INVALIDINST,
PRIVILEGEDINST, FATALSEGFault, NOPROCEDURE,
NOROUTING, RESULTS DOUBTFUL, OPERATIONABORTED);

T_dbgErrCode = (RECORDTOOLONG, INVALIDOPERATION,
INVALIDID, INVALIDPARAMETERS,
ERRORINTHESTRING, SECURITYVIOLATION,
NOTYETIMPLEMENTED, BADSCOPE,
BADSTATE, ILLEGINDEX,
CHANDESTROY, CHANBUSY,
CHILDALIVE, NOOBJECT, NOWRITERS);

T_pmmWarnCode = (MEMCONFLICT, NEFILE,
TYPECONFLICT, FORMATERROR,
ILLEGSEGLEN, INVALIDSEGNUM,
SOFTINTERRUPT, NOHALT,
PROCDESTROY, PROCSUSPEND,
BREAKPT, NOTEMPTY);

T_pmmErrCode = (MEMOVF, CONCERROR, TOOLONG);

T_fsWarnCode = (BYTESTREAMOPERATION, POSITIONALOPERATION,
INCONSISTENTDATABASE, NOPRIMARYINDEX,
DUPLICATEDKEY, INCOMPLETERECORD);

T_fsErrCode = (TIMEOUT, RECORDLOCKED, DEVICENOTREADY,
INVALIDPATHNAME, NAMEALREADYEXISTS,
NAMENOTFOUND, NOTEMPTYDIRECTORY,
ENDOFFILE, OUTOFBOUNDS,
RECORDNOTFOUND, RECORDALREADYEXISTS,
KEYNOTFOUND, KEYALREADYEXISTS,
OPERATORINTERVENTION);

T_wswarnCode = (ENDOFFPAGE, DOCUMENTTOOFARIN,
OPERATORREQUEST, NOCHARS,
BREAKOCCURRED, TERMINALON);

```

T_wsErrCode = (UNITDOWN, UNITNOTAVAILABLE,
RESOURCEBUSY,
LINEERROR, BOURRAGE,
NODOCUMENT, CARTERISOPEN,
LOCAL, VERIFYORREADERROR,
VIRGINSTRIPE, PRINtheadBLOCKED,
ENDOFINPUT, BADGEINSERTED,
INVALIDINSERTION, ENDOFTAPE,
OFFOCCURRED, PRTIMEOUT);

```

```

T_rtCode = (RTTODEFINE);

```

```

T_usrCode = (USRTODEFINE);

```

```

T_lmWarnCode = (LMENDOFFILE);

```

```

T_lmErrCode = (CONNFAILED);

```

```

T_thWarnCode = (THWTODEFINE);

```

```

T_thErrCode = (THETODEFINE);

```

```

T_frWarnCode = (FRTIMEOUT, FRTERMCHAR);

```

```

T_frErrCode = (FRLINEERROR, SYSTEMFRERROR,
PERIPHERALNOTOPENED, PERIPHERALOPENED,
PERIPHERALDOWN, INVALIDFRID,
FRUNITNOTAVAILABLE,
INVALIDFRPARAMETERS);

```

```

T_reply = record
    case class : T_class of
        CORRECT      : (correctCode : T_correctCode);
        SYSTEM       : (sysErrCode  : T_sysErrCode );
        DEBUG        : (dbgErrCode  : T_dbgErrCode );
        PMMWARNING    : (pmmWarnCode : T_pmmWarnCode);
        PMMERROR      : (pmmErrCode  : T_pmmErrCode );
        FSWARNING     : (fswWarnCode : T_fswWarnCode );
        FSERROR       : (fsErrCode   : T_fsErrCode );
        WSWARNING     : (wswWarnCode : T_wswWarnCode );
        WSERROR       : (wsErrCode   : T_wsErrCode );
        RUNTIME       : (rtCode       : T_rtCode );
        USRCLASS      : (usrCode      : T_usrCode );
        LMWARNING     : (lmWarnCode  : T_lmWarnCode );
        LMERROR       : (lmErrCode   : T_lmErrCode );
        THWARNING     : (tnWarnCode  : T_thWarnCode );
        TERROR        : (thErrCode   : T_thErrCode );
        FRWARNING     : (frWarnCode  : T_frWarnCode );
        FRERROR       : (frErrCode   : T_frErrCode );
    end;

```

```

R_id = record
end;

```

```

T_id = ^R_id;

```

```

T_systemId = record
    net : T_id;
    local : T_id;
end;

```

```

INPUT
mtu.obj
/IPL/DPC/COMMON/INTERF/LPRT.obj
LIBRARY
/IPL/DPC/COMMON/LIB/P_PAU_ui.lib
/IPL/DPC/COMMON/LIB/rtp_ui.lib
/IPL/DPC/COMMON/LIB/ems_ui.lib
OUTPUT mtu.out
MAP mtu.map
OPTIMIZE
Type 2, Attributes 1 <<59>>%0000 PROG_p *_[pP]
Type 5, Attributes 0 <<60>>%0000 PROG_d *_[dD] *_k
Type 6, Attributes 32 <<63>>%0000 *_[sS]
Type 5, Attributes 0 <<61>>%0000 *_[h1] <<61>>%4000 *_e
ENTRY Pmain

```

mtu-link

File commandi per zloc :

T> zloc < mtu-link

- INCL -

AMS.d	AMSBS.d	AMSK.d	AMSP.d
AUX.d	CHAIN_p.i	DIFAS_p.i	DIFAS_t.i
GTS_p.i	GTS_t.i	LMinter_p.i	LMinter_t.i
NAME.d	PGU.d	PM.d	VISA.d
badg_p.i	badg_t.i	cadp_p.i	cadp_t.i
cal.d	clnd.d	fstypes.i	getp.d
ln_p.i	ln_t.i	micr_p.i	micr_t.i
mws.d	pinp_p.i	pinp_t.i	port_c.i
port_p.i	port_t.i	prin_p.i	prin_t.i
putp.d	systypes.i	term_p.i	term_t.i
usr_eru_p.i	eru_t.i	ker_b.i	ker_bob_p.i
ker_buf_p.i	ker_buf_t.i	ker_c.i	ker_cns_c.i
ker_dkcmd.i	ker_erlog.i	ker_p.i	ker_t.i
ker_sct_t.i	MTS_p.i	MTS_t.i	LMsnaWInfo_t.i
LMsnaRInfo_t.i	LMbscInfo_t.i	LM3270info_t.i	LMlmInfo_t.i
LMgenInter_t.i	LMlmdumT_t.i	LMgenRInfo_t.i	LMsnadebug_t.i
LM2848info_t.i	LMRwtrace_t.i	LMdeb_t.i	LMsnatrace_t.i
LMdumtrace_t.i	CM_p.i	CM_t.i	suconst.p
sutype.p	suproc.p	FTRtypes.i	FTRproc.i
FTCtypes.i	8CONFL.d	urmlog.d	usrlog.d
dump.d	VISAP.d	RTGSP.d	BLOG.d
TOTAL 84 FILES			

- INTERF -

ALIAS	BDG_ui.obj	CAD_ui.obj	CONVERT
LM_ui.obj	LPRT.obj	MCR_ui.obj	MKBOOT
MKENV	MKVOL	PATHNAME	PPD_ui.obj
PRT_ui.obj	PRY	P_PAU_ui.obj	P_PGU_ui.obj
SHDIR	TUB_ui.obj	WSM_ui.obj	ams_ui.obj
aux_ui.obj	calendar.obj	exec.obj	exec_PM.obj
ker_ui.obj	key_ui.obj	name.obj	opc_ui.obj
pm_ui.obj	port_ui.obj	pos_ui.obj	putp.obj
rpc_ui.obj	rtp_ui.obj	visa.obj	MTS.obj
MTS_ui.obj	MTS_BATCH.obj	CHAIN_DML.obj	CM_ui.obj
MSW_ui.obj	slam.obj	snamsoft.obj	readAS.obj
readGL.obj	area.obj	getReqId.obj	pInit.obj
FTR.obj	FTFserver.obj	RTGSP_ui.obj	bconfl.obj
get_path.obj	lms_ui.obj	PM_dump.obj	visap.obj
blog.obj	eru_ui.obj	set_mach.obj	trasc.obj
confiles.obj			
TOTAL 61 FILES			