LASER v2.3

SISTEMA DI SVILUPPO PER MARCATURA LASER

MANUALE DI PROGRAMMAZIONE

ELETTRONICA VALSERIANA srl
EVLASER©
produzione di laser industriali e chirurgici
P.IVA e C.F. n. IT00759540164 - CCIAA n. 170839
via S. Carlo, 45/47 - 24020 Casnigo (BG) - Italia
tel. 035/726301 - fax 035/740758
E-mail: evlaser@cyberg.it

Indice

1. Regole di programmazione

- su ogni linea di programma può essere scritto un solo comando:

```
LET %aa 5                  && ok
* %bb %aa %aa              && ok
STR $ss %bb     PRINT $ss       && ERRORE!!!
```

- ogni comando può essere abbreviato ad almeno quattro lettere:

```
STRING $ss 5 "#"                && crea "#####"
STRIN  $ss 7 "s"                && crea "sssssss": il comando è STRING
STRI   $ss 8 "M"                && crea "MMMMMMMM": il comando è STRING
STR    $ss 5 "#"                && ERRORE!!!: il comando è STR
STR    $ss 5               && crea "5": il comando è STR
```

- il comando è sempre all'inizio della linea ed i parametri lo seguono; le variabili che vengono assegnate per prime, costanti o variabili che vengono solo lette per ultime
- nelle sintassi dei comandi i parametri spiegati dopo l'uguale (=) vengono assegnati (e quindi devono essere variabili), gli altri solamente letti (e quindi possono essere variabili o costanti):

```
STR $xyzr 5.12             && ok
STR $xyzr %x               && ok
STR "abc" 5                     && ERRORE!!!
STR "123" %x               && ERRORE!!!
```

- ogni comando ha una sintassi ben precisa da rispettare ed i parametri non sono mai opzionali:

```
STRING $s 5 "%"            && ok
STRING %s 5 "%"            && ERRORE!!!
STRING $s "5" "%"               && ERRORE!!!
STRING $s 5                     && ERRORE!!!
```

- coordinate e dimensioni sono sempre espresse in mm, gli angoli in gradi (°) con lo zero ad est, positivi in senso antiorario:

```
LOCATE 10.5 12             && posizionamento in 10.5mm, 12mm
HEIGHT 3.1                 && altezza testo = 3.1mm
SROTATION 90               && direzione del testo 90° (nord)
PRINT "EVLASER"            && stampa di "EVLASER"
```

- i parametri dei comandi possono essere di diversi tipi:

costanti numeriche:	3.14	5		3000
costanti stringa:	"Year: 95"		"S/N"	"EVLASER"
variabili numeriche:	%posx	%timer		%zaxis
variabili stringa:	$name	$logo		$answer
etichette	:mask	:init		:subroutine

- i nomi di variabili o etichette possono contenere lettere, cifre e qualsiasi carattere speciale; non vi è sensibilità sulle lettere minuscole o maiuscole; fare molta attenzione a non avere nello stesso script etichette con lo stesso nome (l'errore può capitare soprattutto quando si includono sorgenti):

```
LET $name "EVLASER"        && var. stringa di nome name
LET $NAME "ITALY"               && var. stringa di nome name
LET $NaMe "LASER"               && var. stringa di nome name
LET $name1 "Nd:Yag"        && var. stringa di nome name1
LET $<234!> "CO²"               && var. stringa di nome <234!>
LET $$ "Shutter"                && var. stringa di nome $
LET %pi 3.14159            && var. numerica di nome pi
```

```
LET %2pi 6.28318                && var. numerica di nome 2pi
LET %rock'n'roll 5.5       && var. numerica di nome rock'n'roll
LET %$ 123.456             && var. numerica di nome $
GOTO :inizio               && etichetta di nome inizio
GOTO :123                  && etichetta di nome 123
GOTO :#12                  && etichetta di nome #12
GOTO :::                   && etichetta di nome ::
GOTO :-1-2-                     && etichetta di nome -1-2-
```

- le variabili o costanti numeriche sono sempre in singola precisione:

```
LET %pi 3.14159265358979323
STR $pi %pi
PRINT $pi       && viene stampato 3.141593
```

- le variabili o costanti stringa possono contenere al massimo 31 caratteri:

```
LET $nam "Donald Duck"          && ok
LET $nam "The report of the week is very important"
                           && ERRORE!!!
ADDSTR $s "The report of the week" " is very important"
                           && ok ma $s conterrà solamente i primi 31 caratteri:
                           && "The report of the week is very "
```

- è possibile inserire nelle costanti stringa tutti i codici ASCII (meno lo 0) precedendo il codice decimale a tre cifre con un backslash (\):

```
LET $deg "45\248"                 && 45°
MSG "Nome: \034Anna\034" "" "" ""  && "Anna"
LET $a "ø"
IF= $a "\237"                   && c'è uguaglianza
```

- il numero massimo di variabili numeriche è 256
- il numero massimo di variabili stringa è 100
- le variabili o costanti numeriche possono essere testate con IF=, IF<>, IF>, IF>=, IF<, IF<=
- le variabili o costanti stringa possono essere testate solo con IF= e IF<>
- IF= e IF<> pretendono come primo parametro sempre una variabile stringa o numerica
- i commenti possono essere inseriti utilizzando un'apostrofo (') all'inizio della linea:

```
LOCATE 0 0
'LINE 40.4 -53.44               linea ignorata
LINE 20 56
'+------------------+      linea ignorata
'¦ Inizio procedura ¦      linea ignorata
'+------------------+      linea ignorata
HPGLCONV "evlaserr.plt"
LOCATE 0 0
EXTLOGO "evlaserr" 0
```

... oppure sulla stessa linea di un comando utilizzando la doppia "e" commerciale (&&):

```
LET %alt 4                 && altezza carattere
LET %x -20                 && ascissa
LET %y 10                  && ordinata
LET $s "EVLASER"                && stringa

HEIGHT %alt
LOCATE %x %y
PRINT $s                   && stampa di "EVLASER"
```

- ogniqualvolta s'incontri un flag numerico normalmente 0 indica il no, la disabilitazione mentre l'1 indica il si, l'abilitazione:

```
LAMP 1                    && accensione lampada
WAIT 500                  && attesa di mezzo secondo
LAMP 0                    && spegnimento lampada
```

- tra i comandi principali si trovano:

LOCATE	posizionamento del raggio laser
PRINT	marcatura di un testo
LINE	marcatura di una linea
AXIS	posizionamento degli assi lineari (o rotanti)
+ - * /	operatori matematici
SIN COS	operatori trigonometrici
TAN ATAN	operatori trigonometrici
INKEY	lettura della tastiera
START	lettura del pulsante di START
SDEF	definizione dello stile di testo ed i correlati: FNAME, HEIGHT, WIDTH, ITALIC, SROTATION
SNAME SNUM	gestione di tabelle di stili di testo
WDEF	definizione stile di marcatura ed i correlati: SPEED, CURRENT, FREQUENCY, WDIMENSION, FROTATION
WNAME WNUM	gestione di tabelle di stili di marcatura
EXTLOGO	marcatura di un logo
HPGLCONV	conversione di un file HPGL in logo
IF, GOTO	controllo del flusso di programma
$INCLUDE	caricamento di librerie di routines
GOSUB	richiamo di una routine

- ogni script può essere interrotto premendo Shift, Control e Alt contemporaneamente; l'utilità si riscontra soprattutto quando lo script non prevede un'uscita o non esce a causa di un errore di programmazione; questa combinazione di tasti non ha effetto quando compaiono a video le seguenti richieste: GETNUM, GETSTR, FIELD, FLIST, MSG, ERRMSG, YESNO, TABVOB, TABSTILI, HPGLCONV, AXSET, MENU, SHOWILK; ATTENZIONE: questa interruzione forzata non chiude eventuali file aperti!

2. Elenco alfabetico dei comandi con argomenti

$INCLUDE "source"	inclusione sorgenti o librerie
* %product %fact1 %fact2	prodotto di due numeri
+ %sum %addend1 %addend2	somma di due numeri
- %difference %minuend %subtrahend	differenza tra due numeri
/ %quotient %divisor %dividend	divisione tra due numeri
ABS %absolute %signed	assoluto di un numero
ABSOLUTE	selezione coordinate assolute
ADDSTR $all $string1 $string2	concatenazione di stringhe
AND %and %byte1 %byte2	prodotto logico
ASC %code $character	codice da carattere ASCII
ATAN %angle %deltay %deltax	arcotangente di un numero
AUTOILK %auto	abilitazione maschera interlock automatica
AXIS %newx %newy %newz %newr%delay %wait %laser	movimentazione assi lineari e/o rotanti
AXRESET	reset assi lineari e/o rotanti
AXSET	definizione parametri assi lineari e/o rotanti
BOX %x %y	marcatura rettangolo
BRESET %result %byte %bit	impostazione a 0, singoli bit
BSET %result %byte %bit	impostazione a 1, singoli bit
BTOGGLE %result %byte %bit	inversione singoli bit
CHGXY %exchange	scambio canali X/Y
CHR $character %code	carattere da codice ASCII
CHS %inverted %value	inversione segno di un numero
CLIP %clip	abilitazione finestra di taglio
COPY $from $to	copia file
COS %cosine %angle	coseno di un angolo
CURRENT %current	impostazione corrente o potenza
DATE $date	data corrente
DEFSEG %segment	segmento di memoria corrente
EDIT $file %protection	modifica file di testo
ELAPSED %secs	secondi dopo mezzanotte
ELSE	condizione IF inversa
ENCODER %speed	encoder volantino elettronico
END	fine script
ENDIF	fine struttura IF
ERRMSG $line1 $line2 $line3 $line4	messaggio di errore
EXIST %exist $file	esistenza file
EXTLOGO $logo %table	esecuzione logo
FCLOSE %handle	chiusura file
FDATE %date %time %handle	data e ora di un file
FFIND %position %handle $text %record	ricerca in un file
FFIRST $found $pattern	ricerca di un file
FIELD %code $field $default %x %y %length %video %number %foreground %background	editing controllato a video
FLIST $file $title $pattern %new $init	lista di selezione file
FLNREAD $line %handle	lettura linea da un file
FLNWRITE $line %handle	scrittura linea in un file
FLOCATE %handle %pos	posizionamento in un file
FNAME $font	selezione font corrente
FNEXT $found	continuazione ricerca di un file
FOPEN %handle $file $mode	apertura di un file
FPOS %position %handle	posizione corrente in un file
FREAD $read %handle %number	lettura da un file
FREQUENCY %frequency	impostazione frequenza di modulazione
FROTATION %frequency	impostazione frequenza di ripetizione pattern
FWRITE %handle $text	scrittura in un file
GETAXIS %positionx %positiony %positionz %positionr	posizione corrente assi lineari e/o rotanti
GETCFG $value $item	lettura file di configurazione
GETNUM %value $prompt %default %length	richiesta di un numero

GETRTCTIME %time %precision	ora corrente per cronometraggio
GETSTR $text $prompt $default %length	richiesta di una stringa
GETSYSN %value %code	lettura variabili di sistema
GOSUB :label	esecuzione routine
GOTO :label	salto di programma
HEIGHT %height	altezza del testo
HPGLCONV $file	conversione da HPGL a logo
IF< %value1 %value2	condizione IF
IF<= %value1 %value2	condizione IF
IF<> variable value	condizione IF
IF= variable value	condizione IF
IF> %value1 %value2	condizione IF
IF>= %value1 %value2	condizione IF
IN %byte %port	lettura da una porta
INGLOGO %lowestx %lowesty %highestx %highesty %development $logo %pen	lettura ingombri logo
INKEY $key	lettura della tastiera
INSTR %position $text $find %start	ricerca in una stringa
INT %integer %real	troncamento decimali
ITALIC %angle	italicità del testo
KILL $file	cancellazione di un file
LAMP %lamp	accensione della lampada
LCASE $lowercase $text	trasformazione in minuscole
LDIMENSION %dimx %dimy	dimensione del logo
LEFT $left $text %number	parte sinistra di una stringa
LEN %length $text	lunghezza di una stringa
LET variable value	assegnamento di una variabile
LINE %x %y	marcatura linea
LOCATE %x %y	posizionamento
LOF %lengthoffile %handle	lunghezza di un file
LPRINT $text	scrittura su stampante
LROTATION %angle	rotazione logo
LTRIM $lefttrim $text	eliminazione spazi iniziali
MENU %item $item $file $title	menu di selezione
MID $middle $text %position %number	estrazione di una stringa
MIRROR %x %y	specularità sugli assi
MKDIR $dir	creazione directory
MSG $line1 $line2 $line3 $line4	messaggio
NAME $old $new	rinomina di un file
NOT %not %byte	inversione logica
OR %or %byte1 %byte2	somma logica
OUT %port %byte	scrittura su una porta
PEEK %byte %address	lettura dalla memoria
POKE %address %byte	scrittura sulla memoria
PRINT $text	marcatura di un testo
PUTCFG $item $value	scrittura file di configurazione
PUTSYSN %code %value	scrittura variabili di sistema
REDRAW	ridisegno griglia di simulazione
RELATIVE	selezione coordinate relative
RESOLUTION %resolution	risoluzione di marcatura
RETURN	ritorno al GOSUB
RIGHT $right $text %number	parte destra di una stringa
RITAXIS %delay	ritardo di assestamento assi
RITLONC %delay	ritardo dopo accensione laser CW
RITLONQ %delay	ritardo dopo accensione laser QS
RITSIM %delay	ritardo in simulazione
RMDIR $dir	cancellazione directory
ROTATE %centerx %centery %angle	rotazione area di marcatura
RTRIM $righttrim $text	eliminazione spazi finali
RUN $scx	esecuzione di uno script
SCALE %basex %basey %scalex %scaley	scala area di marcatura
SCREEN %mode	selezione modalità video
SDEFINITION $font %height %width %charspace %linespace %angle %rotation	definizione stile di testo
SETALIM %powersupply	abilitazione alimentatore
SETCO2 %co2	accensione tubo laser CO^2

SETPLC %plc	segnale PLC
SETRELE %rele	segnale relé
SHOWILK %autoexit	visualizzazione maschera interlock
SIN %sine %angle	seno di un angolo
SNAME $file	selezione tabella stili di testo
SNUMBER %style	selezione stili di testo
SOUND %frequency %time	segnale acustico
SPEED %speed	velocità di marcatura
SQR %squareroot %number	radice quadra
SROTATION %angle	rotazione del testo
START %start	lettura pulsante di START
STR $string %number	conversione da numero a stringa
STRING $string %number $character	creazione stringa
TABSTILI $table	gestione tabella stili di testo
TABVOB $table	gestione tabella stili di marcatura
TAN %tangent %angle	tangente di un angolo
TIME $time	ora corrente
TRIM $trim $text	eliminazione spazi iniziali e finali
UCASE $uppercase $text	trasformazione in maiuscole
VAL %number $string	conversione da stringa a numero
VCLS	cancellazione video
VCOLOR %foreground %background	selezione colori a video
VLOCATE %row %column	posizionamento video
VOLCMD $command	comandi volantino elettronico
VOLDATA $text	dati volantino elettronico
VOLKEYS %byte	tasti volantino elettronico
VPRINT $text	stampa a video
WAIT %time	attesa
WCLIP %xmin %ymin %xmax %ymax	definizione finestra di taglio
WDEFINITION %dimension $patternlibrary %number %speed %resolution %patternresolution %frequency %current %diafram %rotationfrequency $type	definizione stile di marcatura
WDIMENSION %dimension	dimensione pattern
WIDTH %width	larghezza percentuale del testo
WNAME $file	selezione tabella stili di marcatura
WNUMBER %style	selezione stili di marcatura
XOR %xor %byte1 %byte2	OR esclusivo logico
YESNO %answer $line1 $line2 $line3 $line4	richiesta di conferma

3. Lista comandi per argomento

3.1. Matematici

*	prodotto
+	somma
-	differenza
/	divisione
ABS	valore assoluto
CHS	cambio di segno
INT	intero
SQR	radice quadra

3.2. Trigonometrici

ATAN	arcotangente
COS	coseno
SIN	seno
TAN	tangente

3.3. Booleani

AND	prodotto logico
BRESET	settaggio singolo bit a 0
BSET	settaggio singolo bit a 1
BTOGGLE	inversione singolo bit
NOT	inversione di bit
OR	somma logica
XOR	or esclusivo

3.4. Gestione file system

COPY	copia file
EXIST	esistenza file
FFIRST	ricerca file o directory con pattern
FNEXT	continuazione ricerca file o directory
KILL	cancellazione file
MKDIR	creazione directory
NAME	rinomina file
RMDIR	cancellazione directory
FLIST	selezione file

3.5. Gestione file a basso livello

FCLOSE	chiusura file
FDATE	data e ora del file
FFIND	ricerca nel file
FLNREAD	lettura linea da file
FLNWRITE	scrittura linea su file
FLOCATE	posizionamento nel file
FOPEN	apertura file
FPOS	posizione corrente nel file
FREAD	lettura da file
FWRITE	scrittura su file
LOF	lunghezza del file

3.6. Gestione stringhe

ADDSTR	concatenazione stringhe
GETNUM	input controllato numerico
GETSTR	input controllato di testo
INSTR	ricerca nella stringa
LCASE	conversione in minuscolo
LEFT	parte iniziale della stringa
LEN	lunghezza stringa
LTRIM	eliminazione spazi iniziali
MID	estrazione parte della stringa
RIGHT	parte finale della stringa
RTRIM	eliminazione spazi finali
STRING	creazione stringa ripetitiva
TRIM	eliminazione spazi iniziali
UCASE	conversione in maiuscolo

3.7. Flusso del programma

$INCLUDE	inclusione sorgenti
ELSE	condizione inversa
END	fine script
ENDIF	fine blocco condizione
GOSUB	richiamo routine
GOTO	salto a etichetta
IF	condizione
LET	assegnamento variabili
RETURN	ritorno alla chiamata
RUN	esecuzione script

3.8. Output a video

FIELD	input controllato
REDRAW	ridisegno griglia di simulazione
SCREEN	selezione modatità video
VCLS	cancellazione schermo
VCOLOR	definizione colore
VLOCATE	posizionamento
VPRINT	stampa

3.9. Interfaccia utente

EDIT	editor testi
ERRMSG	messaggio di errore
FIELD	input controllato
FLIST	selezione file
GETNUM	input controllato numerico
GETSTR	input controllato di testo
HPGLCONV	conversione file HPGL
MENU	selezione da menu
MSG	messaggio
YESNO	richiesta di conferma

3.10. Dialog

AXSET	settaggi assi
HPGLCONV	conversione file HPGL
TABSTILI	tabella stili di testo
TABVOB	tabella stili di marcatura

3.11. Gestione assi

AXIS	movimentazione assi
AXRESET	reset assi
AXSET	settaggi assi
GETAXIS	posizione assi
RITAXIS	attesa assestamento assi

3.12. Conversione

ASC	codice dal carattere
CHR	carattere dal codice
STR	stringa del valore

VAL	valore della stringa

3.13. Marcatura e posizionamento

ABSOLUTE	coordinate assolute
BOX	rettangolo
EXTLOGO	logo
LINE	linea
LOCATE	posizionamento
PRINT	testo
RELATIVE	coordinate relative

3.14. Volantino elettronico

ENCODER	lettura encoder
VOLCMD	comando
VOLDATA	dati
VOLKEYS	lettura tasti

3.15. Stile di marcatura

CURRENT	corrente o potenza
FREQUENCY	frequenza
FROTATION	frequenza di rotazione pattern
RESOLUTION	risoluzione
SPEED	velocità di traslazione
WDEFINITION	definizione stile
WDIMENSION	dimensione pattern
WNAME	caricamento tabella
WNUMBER	caricamento gruppo

3.16. Stile di testo

FNAME	nome font
HEIGHT	altezza
ITALIC	italicità
SDEFINITION	definizione stile
SNAME	caricamento tabella
SNUMBER	caricamento gruppo
SROTATION	rotazione
WIDTH	larghezza percentuale
PRINT	marcatura testo

3.17. Ritardi

RITAXIS	attesa assestamento assi
RITLONC	attesa dopo accensione in continua
RITLONQ	attesa dopo accensione ad impulsi
RITSIM	ritardo simulazione
WAIT	attesa

3.18. Temporizzazioni

DATE	data corrente
TIME	ora corrente
ELAPSED	secondi dopo mezzanotte
GETRTCTIME	orologio in tempo reale
WAIT	attesa

3.19. Logo

EXTLOGO	marcatura
INGLOGO	ingombri e sviluppo
LDIMENSION	dimensione
LROTATION	rotazione

3.20. Generali

CHGXY	scambio assi
CLIP	abilitazione finestra di taglio
MIRROR	specularità
ROTATE	rotazione
SCALE	scala
WCLIP	definizione finestra di taglio

3.21. Interlock

AUTOILK	gestione maschera automatica
SHOWILK	visualizzazione maschera interlock

3.22. Comunicazione e I/O

DEFSEG	selezione segmento memoria
IN	input da porta
OUT	output su porta
PEEK	lettura da memoria
POKE	scrittura in memoria
LAMP	lampada di marcatura in corso
SETALIM	abilitazione alimentatore
SETCO2	accensione tubo CO^2
SETPLC	gestione PLC
SETRELE	gestione relé
START	lettura pulsante di START

3.23. Avanzato

GETCFG	lettura configurazione corrente
GETSYSN	lettura valori di sistema
PUTCFG	settaggio configurazione corrente
PUTSYSN	settaggio valori di sistema

3.24. Altro

INKEY	lettura buffer tastiera
LPRINT	stampa su LPT1
SOUND	segnalazione acustica

4. Riferimenti a tabelle

5. Sintassi dei comandi

$INCLUDE "source"

source.....file sorgente (default .inc)

Il file indicato verrà inserito in quel punto durante la compilazione.

N.B.:

- il nome di file dev'essere obbligatoriamente una costante stringa ed esistere in fase di compilazione;
- assicurarsi di non utilizzare variabili od etichette con lo stesso nome di altre già presenti nel file di include per evitare comportamenti strani o errati;
- normalmente un file di include fa subito un salto alla fine dello stesso in modo da rendere disponibili le routine presenti senza eseguirle al punto dell'inclusione;
- mentre ad ogni modifica di un .scr la ricompilazione è automatica, modificando un file di include questo non succede; per forzare la ricompilazione di tutti gli script che includono un sorgente modificato è sufficiente cancellarne i .scx

Vedere anche: GOSUB

Esempio:

```
+-TEST.SCR-----------------------------------+ +-MATH.INC-----------+
|                                            | |                     |
| $INCLUDE "math"                            | | GOTO :endpit        |
|                                            | | :pitagora           |
| GETNUM %c1 "Cateto 1?" 0 10                | |    * %_tmp %c1 %c1  |
| GETNUM %c2 "Cateto 2?" 0 10                | |    * %ip %c2 %c2    |
| GOSUB :pitagora                            | |    + %ip %_tmp %ip  |
| STR $c1 %c1                                | |    SQR %ip %ip      |
| STR $c2 %c2                                | | RETURN              |
| STR $ip %ip                                | | :endpit             |
| MSG "Cateto 1, 2 e ipotenusa:" $c1 $c2 $ip | +---------------------+
+--------------------------------------------+

+-TEST.SCR appena prima della compilazione ----+
|                                              |
| GOTO :endpit                                 |
| :pitagora                                    |
|    * %_tmp %c1 %c1                           |
|    * %ip %c2 %c2                             |
|    + %ip %_tmp %ip                           |
|    SQR %ip %ip                               |
| RETURN                                       |
| :endpit                                      |
|                                              |
```

```
| GETNUM %c1 "Cateto 1?" 0 10                 |
| GETNUM %c2 "Cateto 2?" 0 10                 |
| GOSUB :pitagora                             |
| STR $c1 %c1                                 |
| STR $c2 %c2                                 |
| STR $ip %ip                                 |
| MSG "Cateto 1, 2 e ipotenusa:" $c1 $c2 $ip  |
+---------------------------------------------+
```

L'esempio dimostra l'utilizzo di una piccola libreria contenente la funzione per il calcolo dell'ipotenusa fornendo i cateti di un triangolo rettangolo utilizzando il teorema di pitagora. La libreria viene scritta separatamente nel file MATH.INC e contiene la routine 'pitagora'. Il sorgente che vuole utilizzare la routine, in questo caso TEST.SCR include la libreria con $INCLUDE "math" ed esegue la routine con GOSUB :pitagora dopo aver chiesto i parametri %c1 e %c2 (i cateti); infine viene utilizzato il valore ritornato %ip cioè l'ipotenusa. Prima della compilazione il sorgente viene 'assemblato'. Scrivere e testare routine complesse separatamente diminuisce la complessità di programmi più grossi.

* %product %fact1 %fact2

%product = prodotto
%factor1 - fattore 1
%factor2 - fattore 2

Il comando trova il prodotto dei due fattori.

Vedere anche: +, -, /, SQR

Esempio:

```
GETNUM %k "°K" 0 10
- %c %k 273.15        && °C = °K - 273.15
* %f %c 9             &&             9
/ %f %f 5             && °F = °C · --- + 32
+ %f %f 32            &&             5
STR $k %k
STR $c %c
STR $f %f
MSG "°K -> °C, °F" $k $c $f
```

L'esempio permette la conversione dei gradi kelvin in celsius e fahrenheit dimostrando l'uso delle quattro operazioni.

+ %sum %addend1 %addend2

%sum =.....somma
%addend1.....addendo 1
%addend2.....addendo 2

Il comando trova la somma dei due addendi.

Vedere anche: *, -, /, ABS

Esempio:

```
GETNUM %k "°K" 0 10
- %c %k 273.15        && °C = °K - 273.15
* %f %c 9             &&            9
/ %f %f 5             && °F = °C · --- + 32
+ %f %f 32            &&            5
STR $k %k
STR $c %c
STR $f %f
MSG "°K -> °C, °F" $k $c $f
```

L'esempio permette la conversione dei gradi kelvin in celsius e fahrenheit dimostrando l'uso delle quattro operazioni.

- %difference %minuend %subtrahend

%difference =.....differenza
%minuend.....minuendo
%subtrahend.....sottraendo

Il comando trova la differenza tra minuendo e sottraendo.

Vedere anche: *, +, /, CHS

Esempio:

```
GETNUM %k "°K" 0 10
- %c %k 273.15        && °C = °K - 273.15
* %f %c 9             &&            9
/ %f %f 5             && °F = °C · --- + 32
+ %f %f 32            &&            5
STR $k %k
STR $c %c
```

```
STR $f %f
MSG "°K -> °C, °F" $k $c $f
```

L'esempio permette la conversione dei gradi kelvin in celsius e fahrenheit dimostrando l'uso delle quattro operazioni.

/ %quotient %divisor %dividend

%quotient =.....quoziente
%divisor.....divisore
%dividend.....dividendo (<>0)

Il comando trova il quoziente tra divisore e dividendo.

Vedere anche: *, +, -

Esempio:

```
GETNUM %k "°K" 0 10
- %c %k 273.15        && °C = °K - 273.15
* %f %c 9             &&             9
/ %f %f 5             && °F = °C · --- + 32
+ %f %f 32            &&             5
STR $k %k
STR $c %c
STR $f %f
MSG "°K -> °C, °F" $k $c $f
```

L'esempio permette la conversione dei gradi kelvin in celsius e fahrenheit dimostrando l'uso delle quattro operazioni.

ABS %absolute %signed

%absolute =.....valore assoluto
%signed.....valore con segno

Il comando trova il valore assoluto.

Vedere anche: CHS, INT

Esempio:

```
GETNUM %n "Spostamento? (±)" 0 10
ABS %tmp %n
STR $n %tmp
IF< %n 0
     MSG "Indietro di" $n "" ""
ELSE
     IF= %n 0
          MSG "Fermo" "" "" ""
     ELSE
          MSG "Avanti di" $n "" ""
     ENDIF
ENDIF
```

L'esempio controlla il segno del valore ma lo presenta in assoluto.

ABSOLUTE

Il comando fa si che tutte le coordinate successive vengano interpretate in modo assoluto. Alla partenza lo script è sempre in assoluto.

Vedere anche: RELATIVE, LOCATE, LINE, BOX

Esempio:

```
LOCATE -20 20
GOSUB :greca
LOCATE -20 0
GOSUB :greca
LOCATE -20 -20
GOSUB :greca
END

:greca
     RELATIVE
     LET %n 10
     :loop
     IF> %n 0
          LINE 3 3
          LINE 3 -3
          - %n %n 1
          GOTO :loop
     ENDIF
     ABSOLUTE
RETURN
```

L'esempio mostra come eseguire delle greche eseguite in RELATIVE rispetto alla posizione ABSOLUTE iniziale utilizzando LOCATE e LINE. E` possibile marcare o simulare.

ADDSTR $all $string1 $string2

$all = ...stringa completa
$string1.....prima stringa
$string2.....seconda stringa

Il comando restituisce la concatenazione di due stringhe.

Vedere anche: LEFT, LEN, LTRIM, MID, RIGHT, RTRIM, STRING, TRIM

Esempio:

```
GETNUM %n " Numero iniziale progressivo?" 0 5
+ %l %n 10
:loop
IF< %n %l
      STR $n %n
      STRING $z 6 "0"
      ADDSTR $n $z $n
      RIGHT $n $n 6
      MSG " Numero seriale:" $n "" ""
      + %n %n 1
      GOTO :loop
ENDIF
```

L'esempio mostra la creazione di dieci numeri seriale progressivi a sei cifre sempre riempiti di zeri a partire da quello specificato.

AND %and %byte1 %byte2

%and =.....AND logico
%byte1.....primo byte (0÷255)
%byte2.....secondo byte (0÷255)

Il comando esegue l'AND logico (detto anche 'prodotto logico') tra i due byte, bit per bit.

```
TABELLA DELLA VERITÀ
+---------------------+
| bit 1 | bit 2 | AND |
|-------+-------+-----|
|   0   |   0   |  0  |
|   0   |   1   |  0  |
|   1   |   0   |  0  |
|   1   |   1   |  1  |
+---------------------+
```

Vedere anche: BRESET, BSET, BTOGGLE, NOT, OR, XOR

Esempio:

```
GETNUM %y "Anno?" 1996 4

let %m 4
GOSUB :multiplo
let %m4 %m

let %m 100
GOSUB :multiplo
let %m100 %m

let %m 400
GOSUB :multiplo
let %m400 %m

NOT %nm100 %m100 && \
OR %tmp %nm100 %m400       && +- bis = m4 AND [(NOT m100) OR m400]
AND %bis %m4 %tmp && /
STR $y %y
IF= %bis 0
     MSG "L'anno" $y "non è bisestile" ""
ELSE
     MSG "L'anno" $y "è bisestile" ""
ENDIF
END

:multiplo                  && ritorna %m = 1 se multiplo di %m altrimenti 0
     / %tmp %y %m
     INT %tmp1 %tmp
     IF= %tmp %tmp1
          LET %m 1
     ELSE
          LET %m 0
     ENDIF
RETURN
```

L'esempio mostra l'utilizzo della logica booleana trovando se l'anno fornito è bisestile o no; un anno è bisestile se è multiplo di 4; se però è multiplo anche di 100 allora non è bisestile; se, ancora, è multiplo anche di 400 allora torna ad essere bisestile.

ASC %code $character

%code =.....codice ASCII (1÷255)
$character.....carattere ASCII (\001÷\255)

Il comando trova il codice del primo carattere ASCII. Il codice ASCII 0 non è valido.
Di seguito la tabella ASCII:

Caratteri di controllo

Dec	Esa	Car	Name	Ctrl	Dec	Esa	Car	Name	Ctrl
000	00		NUL	^@	016	10		DLE	^P
001	01		SOH	^A	017	11		DC1	^Q
002	02		STX	^B	018	12		DC2	^R
003	03		ETX	^C	019	13		DC3	^S
004	04		EOT	^D	020	14		DC4	^T
005	05		ENQ	^E	021	15		NAK	^U
006	06		ACK	^F	022	16		SYN	^V
007	07		BEL	^G	023	17		ETB	^W
008	08		BS	^H	024	18		CAN	^X
009	09		HT	^I	025	19		EM	^Y
010	0A		LF	^J	026	1A		SUB	^Z
011	0B		VT	^K	027	1B		ESC	^[
012	0C		FF	^L	028	1C		FS	^\
013	0D		CR	^M	029	1D		GS	^]
014	0E		SO	^N	030	1E		RS	^^
015	0F		SI	^O	031	1F		US	^_

Punteggiatura, Cifre, Lettere maiuscole

Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car
032	20		048	30	0	064	40	@	080	50	P
033	21	!	049	31	1	065	41	A	081	51	Q
034	22	"	050	32	2	066	42	B	082	52	R
035	23	#	051	33	3	067	43	C	083	53	S
036	24	$	052	34	4	068	44	D	084	54	T
037	25	%	053	35	5	069	45	E	085	55	U
038	26	&	054	36	6	070	46	F	086	56	V
039	27	'	055	37	7	071	47	G	087	57	W
040	28	(	056	38	8	072	48	H	088	58	X
041	29	)	057	39	9	073	49	I	089	59	Y
042	2A	*	058	3A	:	074	4A	J	090	5A	Z
043	2B	+	059	3B	;	075	4B	K	091	5B	[
044	2C	,	060	3C	<	076	4C	L	092	5C	\
045	2D	-	061	3D	=	077	4D	M	093	5D	]
046	2E		062	3E	>	078	4E	N	094	5E	^
047	2F	/	063	3F	?	079	4F	O	095	5F	_

Lettere minuscole, Varie

Dec	Esa	Car	Dec	Esa	Car
096	60	`	112	70	p
097	61	a	113	71	q
098	62	b	114	72	r
099	63	c	115	73	s
100	64	d	116	74	t
101	65	e	117	75	u
102	66	f	118	76	v
103	67	g	119	77	w
104	68	h	120	78	x
105	69	i	121	79	y
106	6A	j	122	7A	z
107	6B	k	123	7B	{
108	6C	l	124	7C	|
109	6D	m	125	7D	}
110	6E	n	126	7E	~
111	6F	o	127	7F	

Internazionali, Caratteri grafici 1

Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car
128	80		144	90		160	A0		176	B0	
129	81		145	91		161	A1		177	B1	
130	82		146	92		162	A2		178	B2	
131	83		147	93		163	A3		179	B3	
132	84		148	94		164	A4		180	B4	
133	85		149	95		165	A5		181	B5	
134	86		150	96		166	A6		182	B6	
135	87		151	97		167	A7		183	B7	
136	88		152	98		168	A8		184	B8	
137	89		153	99		169	A9		185	B9	
138	8A		154	9A		170	AA		186	BA	
139	8B		155	9B		171	AB		187	BB	
140	8C		156	9C		172	AC		188	BC	
141	8D		157	9D		173	AD		189	BD	
142	8E		158	9E		174	AE		190	BE	
143	8F		159	9F		175	AF		191	BF	

Caratteri grafici 2, Simboli

Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car
192	C0		208	D0		224	E0		240	F0	
193	C1		209	D1		225	E1		241	F1	
194	C2		210	D2		226	E2		242	F2	
195	C3		211	D3		227	E3		243	F3	
196	C4		212	D4		228	E4		244	F4	
197	C5		213	D5		229	E5		245	F5	
198	C6		214	D6		230	E6		246	F6	
199	C7		215	D7		231	E7		247	F7	
200	C8		216	D8		232	E8		248	F8	
201	C9		217	D9		233	E9		249	F9	
202	CA		218	DA		234	EA		250	FA	
203	CB		219	DB		235	EB		251	FB	
204	CC		220	DC		236	EC		252	FC	
205	CD		221	DD		237	ED		253	FD	
206	CE		222	DE		238	EE		254	FE	
207	CF		223	DF		239	EF		255	FF	

Vedere anche: CHR

Esempio:

```
VLOCATE 1 1
VPRINT "Premere dei tasti, "
VPRINT "ESC per uscire"

:loop
INKEY $t
IF= $t ""
GOTO :loop
ENDIF
ASC %t $t
VLOCATE 5 1
VPRINT $t
IF= %t 27 &&    ESC
     END
     ELSE
```

```
      IF< %t 48 &&    0
           VPRINT ": carattere speciale"
      ELSE
           IF<= %t 57 &&   9
                VPRINT ": cifra"
           ELSE
                IF< %t 65 &&    A
                     VPRINT ": carattere speciale"
                ELSE
                     IF<= %t 90 &&Z
                          VPRINT ": lettera maiuscola"
                     ELSE
                          IF< %t 97 &&  a
                               VPRINT ": carattere speciale"
                          ELSE
                               IF<= %t 122 &&  z
                                 VPRINT ": lettera minuscola"
                               ELSE
                                 VPRINT ": carattere speciale"
                               ENDIF
                          ENDIF
                     ENDIF
                ENDIF
           ENDIF
      ENDIF
ENDIF
VPRINT " "
GOTO :loop
```

L'esempio visualizza, in base al tasto premuto, se si tratta di una lettera minuscola, maiuscola o cifra. Alla pressione di Escape lo script termina. In simulazione non è possibile vedere il risultato.

ATAN %angle %deltay %deltax

%angle =.....angolo [°]
%deltay.....cateto verticale (ordinata)
%deltax.....cateto orizzontale (ascissa)

Il comando trova l'angolo compreso tra cateto orizzontale e ipotenusa di un triangolo rettangolo fornendo i due cateti (arcotangente del rapporto tra ordinata e ascissa).

Vedere anche: COS, SIN, TAN

Esempio:

```
GETNUM %a "Angolo?" 0 10
SIN %s %a
COS %c %a
TAN %t %a
STR $a %a
STR $s %s
```

```
STR $c %c
MSG "Angolo, seno, coseno:" $a $s $c
STR $t %t
MSG "Angolo, tangente:" $a $t ""
ATAN %a1 %s %c
STR $a1 %a1
MSG "Angolo in base a seno e coseno:" $a1 "" ""
```

L'esempio mostra l'utilizzo delle funzioni trigonometriche.

AUTOILK %auto

%auto.....maschera interlock automatica? (0=no, 1=si)

Il comando permette l'abilitazione o la disabilitazione della maschera interlock automatica; quando abilitata essa compare e scompare immediatamente in base allo stato degli interlock; quando la maschera è a video il tasto ESC blocca e termina lo script.
Quando disabilitata la gestione va fatta da script:

- controllare lo stato degli interlock con il comando "GETSYSN %ilk 88"
- avvertire l'utente con un messaggio di interlock presenti o assenti
- vietare la marcatura che comunque non avverrebbe
- visualizzare se richiesto la maschera con il comando SHOWILK (in questo caso il tasto ESC fa scomparire la maschera ma senza bloccare o terminare lo script)

N. B.:
Per prendere possesso della gestione degli interlock sin dall'inizio la linea "AUTOILK 0" deve essere la prima in assoluto dello script. Infatti la maschera automatica si abilita solo dal secondo comando.

Vedere anche: SHOWILK, GETSYSN

Esempio:

```
AUTOILK 0
VCLS
VLOCATE 1 1
VPRINT "ESC per uscire "
VLOCATE 10 10
VPRINT "Interlock:"
LET %ilk -1               && forza il primo refresh video
:loop
INKEY $t
IF= $t "\027"         && premuto ESCAPE?
     END
ENDIF
GETSYSN %tmp 88
IF<> %ilk %tmp        && interlock cambiati?
     let %ilk %tmp
     ELAPSED %t
     VLOCATE 10 21
     IF= %ilk 0
          VPRINT "assenti "
```

```
      ELSE
           VPRINT "PRESENTI"
      ENDIF
ENDIF
IF<> %ilk 0               && interlock presenti?
      ELAPSED %tmp
      - %tmp %tmp %t
      IF> %tmp .5         && interlock presenti da più di .5s?
           SHOWILK 1
      ENDIF
ENDIF
GOTO :loop
```

L'esempio controlla e visualizza lo stato degli interlock. Se durano per più di mezzo secondo compare anche la maschera degli interlock. In simulazione non è possibile controllare il funzionamento dell'esempio ed in esecuzione si deve poter intervenire su un interlock reale (per esempio spegnendo la lampada o aprendo le portelle testa).

AXIS %newx %newy %newz %newr %delay %wait %laser

%newx.....nuova posizione assoluta asse X [mm]
%newy.....nuova posizione assoluta asse Y [mm]
%newz.....nuova posizione assoluta asse Z [mm]
%newr.....nuova posizione assoluta asse R [°]
%delay.....ritardo di movimentazione assi (4÷10000)
%wait.....attendere movimentazione? (0=no, 1=si)
%laser.....(*)

(*) funzione futura

Il comando muove fino a quattro assi contemporaneamente. Di solito l'asse X è l'asse orizzontale piano, l'Y il verticale piano, lo Z quello verticale per la messa a fuoco (supporto della testa laser) e l'R quello di rotazione (mandrino, tavola rotante, ecc...). L'asse Z si muove comunque separatamente: per primo se si alza (incremento) per ultimo se si abbassa (decremento). Le coordinate sono, nel caso vi sia uno zero software, relative a quello. Nel caso una movimentazione vada oltre il finecorsa massimo impostato o sotto lo zero comparirà una finestra di avvertimento e conferma. Confermare la movimentazione può ovviamente causare collisioni meccaniche o perdite di posizione degli assi (in quest'ultimo caso si dovrà eseguire un reset). Se la seriale è disabilitata (0) il comando non ha effetto. Gli assi disabilitati non si muovono in ogni caso; se il tipo di EPROM è quella a tre assi l'asse R, anche se abilitato, in ogni caso non si muove.

Il valore del ritardo è inversamente proporzionale alla velocità (espressa in m/min) di movimentazione:

$$\text{ritardo} = 100 / \text{velocità} - 28$$

dove $0.010 \leq \text{velocità} \leq 3.125$ [m/min]
quindi $10000 \leq \text{ritardo} \leq 4$

Normalmente è meglio attendere il termine della movimentazione per assicurarsi di non eseguire delle operazioni errate mentre il sistema è ancora in movimento (marcature, altre movimentazioni).

N. B.:
Vi sono alcuni problemi di comunicazione ancora non individuati che bloccano a volte la movimentazione; se lo script dovrà eseguire la movimentazione degli assi si consiglia vivamente di iniziare lo script con i seguenti comandi al fine di ridurre o eliminare i problemi suddetti:

```
GETAXIS %posx %posy %posz %posr
AXIS %posx %posy %posz %posr 20 1 0
```

Vedere anche: AXRESET, AXSET, GETAXIS, RITAXIS

Esempio:

```
GETAXIS %posx %posy %posz %posr
AXIS %posx %posy %posz %posr 20 1 0
SDEFINITION "iso9" 2 100 0 0 0 270

LET %posx 0
:loop
AXIS %posx %posy %posz %posr 20 1 0
GOSUB :linea
+ %posx %posx 1
IF<= %posx 200
     GOTO :loop
ENDIF
END

:linea
     LOCATE 0 0
     / %tmp %posx 10
     INT %tmp1 %tmp
     IF= %tmp %tmp1
          LINE 0 -5            && linea dei cm
          LOCATE -1 -5.5
          STR $x %posx
          PRINT $x
     ELSE
          / %tmp %posx 5
          INT %tmp1 %tmp
          IF= %tmp %tmp1
               LINE 0 -4       && linea del mezzo cm
          ELSE
               LINE 0 -3       && linea dei mm
          ENDIF
     ENDIF
RETURN
```

L'esempio permette la marcatura di un righello di precisione di 200mm utilizzando la movimentazione degli assi.

AXRESET

Il comando esegue il reset degli assi (ricerca dello zero meccanico o origine) e l'eventuale posizionamento sugli zeri software. Se la seriale è disabilitata (0) il comando non ha effetto.

Vedere anche: AXIS, AXSET, GETAXIS, RITAXIS

Esempio:

```
GETAXIS %posx %posy %posz %posr
AXIS %posx %posy %posz %posr 20 1 0

VCLS
VPRINT "* MOVIMENTAZIONE ASSI *\013\013"
VPRINT "ESC = uscire, * = reset, "
VPRINT "S = settaggi\013"
VPRINT "X/Y/Z/R = posizione assi:"
GOSUB :posassi

:loop
INKEY $t
IF= $t ""
     GOTO :loop
ENDIF
UCASE $t $t
IF= $t "\027" && ESC
     END
ENDIF
IF= $t "*" && *
     AXRESET
ENDIF
IF= $t "S" && S
     AXSET
ENDIF
IF= $t "X"
     GETNUM %posx "Posizione asse X?" %posx 10
ENDIF
IF= $t "Y"
     GETNUM %posy "Posizione asse Y?" %posy 10
ENDIF
IF= $t "Z"
     GETNUM %posz "Posizione asse Z?" %posz 10
ENDIF
IF= $t "R"
     GETNUM %posr "Posizione asse R?" %posr 10
ENDIF
GOSUB :posassi
GOTO :loop

:posassi
     AXIS %posx %posy %posz %posr 20 1 0
     GETAXIS %posx %posy %posz %posr
     VLOCATE 4 27
     STR $p %posx
     VPRINT $p
     VPRINT ", "
     STR $p %posy
     VPRINT $p
     VPRINT ", "
     STR $p %posz
     VPRINT $p
     VPRINT ", "
     STR $p %posr
     VPRINT $p
     VPRINT " "
RETURN
```

L'esempio permette il settaggio, il reset ed il posizionamento degli assi in modo semplice e guidato.

AXSET

Il comando permette la modifica dei parametri di gestione degli assi aprendo la dialog relativa. I parametri sono:

- seriale in uso (0 per disabilitare la gestione, 1 o 2)
- velocità (in realtà è un ritardo: rit = 100 / vel[m/min] - 28)
- tipo di EPROM (per 3 o 4 assi)

e per ciascun asse:

- abilitazione
- passi per millimetro (o per grado)
- finecorsa software (se 0 è disabilitato)
- zero software

Vedere anche: AXIS, AXRESET, GETAXIS, RITAXIS, HPGLCONV, TABSTILI, TABVOB

Esempio:

```
GETAXIS %posx %posy %posz %posr
AXIS %posx %posy %posz %posr 20 1 0

VCLS
VPRINT "* MOVIMENTAZIONE ASSI *\013\013"
VPRINT "ESC = uscire, * = reset, "
VPRINT "S = settaggi\013"
VPRINT "X/Y/Z/R = posizione assi:"
GOSUB :posassi

:loop
INKEY $t
IF= $t ""
     GOTO :loop
ENDIF
UCASE $t $t
IF= $t "\027" && ESC
     END
ENDIF
IF= $t "*" && *
     AXRESET
ENDIF
IF= $t "S" && S
     AXSET
ENDIF
IF= $t "X"
     GETNUM %posx "Posizione asse X?" %posx 10
ENDIF
IF= $t "Y"
     GETNUM %posy "Posizione asse Y?" %posy 10
ENDIF
IF= $t "Z"
     GETNUM %posz "Posizione asse Z?" %posz 10
ENDIF
IF= $t "R"
     GETNUM %posr "Posizione asse R?" %posr 10
ENDIF
GOSUB :posassi
GOTO :loop

:posassi
     AXIS %posx %posy %posz %posr 20 1 0
     GETAXIS %posx %posy %posz %posr
     VLOCATE 4 27
```

```
      STR $p %posx
      VPRINT $p
      VPRINT ", "
      STR $p %posy
      VPRINT $p
      VPRINT ", "
      STR $p %posz
      VPRINT $p
      VPRINT ", "
      STR $p %posr
      VPRINT $p
      VPRINT " "
RETURN
```

L'esempio permette il settaggio, il reset ed il posizionamento degli assi in modo semplice e guidato.

BOX %x %y

%x.....ascissa spigolo opposto [mm]
%y.....ordinata spigolo opposto [mm]

Marcatura di un rettangolo dal punto corrente a quello specificato intesi come spigoli opposti. Il punto corrente non varia.

Vedere anche: LINE, LOCATE, PRINT, EXTLOGO, ABSOLUTE, RELATIVE

Esempio:

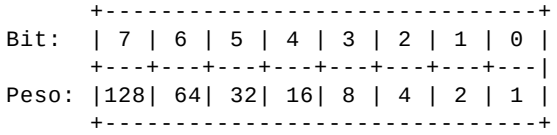
```
LET %lato 60
:loop
IF> %lato 0
      LOCATE %lato %lato
      CHS %lato
      BOX %lato %lato
      CHS %lato
      - %lato %lato 2
      GOTO :loop
ENDIF
END
```

L'esempio marca 30 quadrati concentrici. Il comando CHS inverte e ripristina il valore del lato ad ogni quadrato.

BRESET %result %byte %bit

%result =.....risultato (0÷255)
%byte.....byte (0÷255)
%bit.....bit da porre a 0 (0÷7)

Il comando permette di porre a 0 (resettare) un bit di un byte.



Vedere anche: BSET, BTOGGLE, AND, NOT, OR, XOR

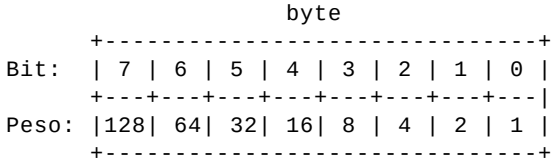
Esempio:

```
BRESET %r 255 3
```

BSET %result %byte %bit

%result =.....risultato (0÷255)
%byte.....byte (0÷255)
%bit.....bit da porre a 1 (0÷7)

Il comando permette di porre a 1 (settare) un bit di un byte.



Vedere anche: BRESET, BTOGGLE, AND, NOT, OR, XOR

Esempio:

```
BSET %r 0 2
```

BTOGGLE %result %byte %bit

%result =.....risultato (0÷255)
%byte.....byte (0÷255)
%bit.....bit da invertire (0÷7)

Il comando permette di invertire un bit di un byte.

```
                  byte
      +-------------------------------+
Bit:  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
      +---+---+---+---+---+---+---+---|
Peso: |128| 64| 32| 16| 8 | 4 | 2 | 1 |
      +-------------------------------+
```

Vedere anche: BRESET, BSET, AND, NOT, OR, XOR

Esempio:

```
BTOGGLE %r 255 7
```

CHGXY %exchange

%exchange.....scambio dei canali X e Y? (0=no, 1=si)

Il comando permette lo scambio dei canali X e Y. Utilizzandolo assieme al comando MIRROR si possono ottenere rotazioni di tutta l'immagine multiple di 90° ed, eventualmente, speculari. In base alla configurazione potrebbe a 1 sin dall'inizio dello script (utilizzare "GETSYSN %xyf 53" per conoscerne il valore). Le tabelle seguenti indicano la rotazione (e l'eventuale specularità) che si danno all'immagine come la si trova all'inizio).

```
Ammettendo che CHGXY all'inizio abbia valore 0:
+-------------------------------------------------------------+
```

```
| MIRROR X  |  0  |  0  |  0  |  0  |  1  |  1  |  1  |  1  |
| MIRROR Y  |  0  |  0  |  1  |  1  |  0  |  0  |  1  |  1  |
| CHGXY     |  0  |  1  |  0  |  1  |  0  |  1  |  0  |  1  |
+-----------+-----+-----+-----+-----+-----+-----+-----+-----|
| ROTAZIONE |  0  | 270s| 180s|  90 |  0s | 270 | 180 | 90s |
+-------------------------------------------------------------+

Ammettendo che CHGXY all'inizio abbia valore 1:
+-------------------------------------------------------------+
| MIRROR X  |  0  |  0  |  0  |  0  |  1  |  1  |  1  |  1  |
| MIRROR Y  |  0  |  0  |  1  |  1  |  0  |  0  |  1  |  1  |
| CHGXY     |  0  |  1  |  0  |  1  |  0  |  1  |  0  |  1  |
+-----------+-----+-----+-----+-----+-----+-----+-----+-----|
| ROTAZIONE | 270s|  0  |  90 | 180s| 270 |  0s | 90s | 180 |
+-------------------------------------------------------------+
```

Vedere anche: MIRROR, ROTATE, SCALE

Esempio:

```
CHGXY 1
```

CHR $character %code

$character = ...carattere ASCII (\001÷\255)
%code.....codice ASCII (1÷255)

Il comando trova il carattere ASCII in base al codice. Il codice ASCII 0 non è valido.
Di seguito la tabella ASCII:

```
Caratteri di controllo
+-----------------------------------------------+
| Dec Esa Car Name Ctrl | Dec Esa Car Name Ctrl |
+-----------------------+-----------------------|
| 000  00     NUL   ^@  | 016  10     DLE   ^P  |
| 001  01     SOH   ^A  | 017  11     DC1   ^Q  |
| 002  02     STX   ^B  | 018  12     DC2   ^R  |
| 003  03     ETX   ^C  | 019  13     DC3   ^S  |
| 004  04     EOT   ^D  | 020  14     DC4   ^T  |
| 005  05     ENQ   ^E  | 021  15     NAK   ^U  |
| 006  06     ACK   ^F  | 022  16     SYN   ^V  |
| 007  07     BEL   ^G  | 023  17     ETB   ^W  |
| 008  08     BS    ^H  | 024  18     CAN   ^X  |
| 009  09     HT    ^I  | 025  19     EM    ^Y  |
| 010  0A     LF    ^J  | 026  1A     SUB   ^Z  |
| 011  0B     VT    ^K  | 027  1B     ESC   ^[  |
| 012  0C     FF    ^L  | 028  1C     FS    ^\  |
| 013  0D     CR    ^M  | 029  1D     GS    ^]  |
| 014  0E     SO    ^N  | 030  1E     RS    ^^  |
```

015 0F SI ^O	031 1F US ^_

Punteggiatura, Cifre, Lettere maiuscole

Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car
032	20		048	30	0	064	40	@	080	50	P
033	21	!	049	31	1	065	41	A	081	51	Q
034	22	"	050	32	2	066	42	B	082	52	R
035	23	#	051	33	3	067	43	C	083	53	S
036	24	$	052	34	4	068	44	D	084	54	T
037	25	%	053	35	5	069	45	E	085	55	U
038	26	&	054	36	6	070	46	F	086	56	V
039	27	'	055	37	7	071	47	G	087	57	W
040	28	(	056	38	8	072	48	H	088	58	X
041	29	)	057	39	9	073	49	I	089	59	Y
042	2A	*	058	3A	:	074	4A	J	090	5A	Z
043	2B	+	059	3B	;	075	4B	K	091	5B	[
044	2C	,	060	3C	<	076	4C	L	092	5C	\
045	2D	-	061	3D	=	077	4D	M	093	5D	]
046	2E		062	3E	>	078	4E	N	094	5E	^
047	2F	/	063	3F	?	079	4F	O	095	5F	_

Lettere minuscole, Varie

Dec	Esa	Car	Dec	Esa	Car
096	60	`	112	70	p
097	61	a	113	71	q
098	62	b	114	72	r
099	63	c	115	73	s
100	64	d	116	74	t
101	65	e	117	75	u
102	66	f	118	76	v
103	67	g	119	77	w
104	68	h	120	78	x
105	69	i	121	79	y
106	6A	j	122	7A	z
107	6B	k	123	7B	{
108	6C	l	124	7C	|
109	6D	m	125	7D	}
110	6E	n	126	7E	~
111	6F	o	127	7F	

Internazionali, Caratteri grafici 1

Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car	Dec	Esa	Car
128	80		144	90		160	A0		176	B0	
129	81		145	91		161	A1		177	B1	
130	82		146	92		162	A2		178	B2	
131	83		147	93		163	A3		179	B3	
132	84		148	94		164	A4		180	B4	
133	85		149	95		165	A5		181	B5	
134	86		150	96		166	A6		182	B6	
135	87		151	97		167	A7		183	B7	
136	88		152	98		168	A8		184	B8	
137	89		153	99		169	A9		185	B9	
138	8A		154	9A		170	AA		186	BA	
139	8B		155	9B		171	AB		187	BB	
140	8C		156	9C		172	AC		188	BC	
141	8D		157	9D		173	AD		189	BD	
142	8E		158	9E		174	AE		190	BE	

143 8F	159 9F	175 AF	191 BF

Caratteri grafici 2, Simboli

Dec Esa Car	Dec Esa Car	Dec Esa Car	Dec Esa Car
192 C0	208 D0	224 E0	240 F0
193 C1	209 D1	225 E1	241 F1
194 C2	210 D2	226 E2	242 F2
195 C3	211 D3	227 E3	243 F3
196 C4	212 D4	228 E4	244 F4
197 C5	213 D5	229 E5	245 F5
198 C6	214 D6	230 E6	246 F6
199 C7	215 D7	231 E7	247 F7
200 C8	216 D8	232 E8	248 F8
201 C9	217 D9	233 E9	249 F9
202 CA	218 DA	234 EA	250 FA
203 CB	219 DB	235 EB	251 FB
204 CC	220 DC	236 EC	252 FC
205 CD	221 DD	237 ED	253 FD
206 CE	222 DE	238 EE	254 FE
207 CF	223 DF	239 EF	255 FF

Vedere anche: ASC

Esempio:

```
VCLS
VPRINT " Caratteri ASCII:\013"
LET %c 1
:loop
IF<= %c 255
     CHR $c %c
     VPRINT $c
     + %c %c 1
     GOTO :loop
ENDIF
MSG "" "" "" ""
```

L'esempio stampa a video tutti i caratteri ASCII da 1 a 255. In simulazione non è possibile vedere il risultato.

CHS %inverted %value

%inverted =.....valore invertito
%value .....valore da invertire

Il comando inverte il segno di un valore.

Vedere anche: -, ABS

Esempio:

```
LET %lato 60
:loop
IF> %lato 0
     LOCATE %lato %lato
     CHS %lato
     BOX %lato %lato
     CHS %lato
     - %lato %lato 2
     GOTO :loop
ENDIF
END
```

L'esempio marca 30 quadrati concentrici. Il comando CHS inverte e ripristina il valore del lato ad ogni quadrato.

(*) CLIP %clip

%clip.....abilitazione finestra di taglio? (0=no, 1=si)

(*) funzione futura

Il comando permette di attivare o disattivare la finestra di taglio (clipping window). Per default è inattiva. Le coordinate della finestra di taglio possono essere definite con il comando WCLIP (v.). Questo potente comando limita la marcatura ad una finestra mentre tutto ciò che sta intorno viene, appunto, tagliato ed omesso. Un'applicazione interessante è quella della marcatura di pezzi cilindrici su tutto lo sviluppo ottenuta ruotando il cilindro un po' per volta marcando nelle varie posizioni la finestra interessata del logo o della scritta completa. Invece che avere molti disegni diversi e preparati spezzati si può marcare sempre tutto il disegno cambiandolo però di posizione e ridefinendo opportunamente la finestra di taglio.

Vedere anche: WCLIP

Esempio:

```
CLIP 1
```

COPY $from $to

$from .....file sorgente (<>"")
$to .....file destinazione (<>"")

Il comando esegue la copia di un file.

Vedere anche: NAME, KILL, MKDIR, RMDIR

Esempio:

```
COPY "laser2.cfg" "c:\092backup.cfg"
```

COS %cosine %angle

%cosine = ...coseno dell'angolo (-1÷1)
%angle.....angolo in gradi [°]

Il comando trova il coseno dell'angolo specificato.

Vedere anche: SIN, TAN, ATAN

Esempio:

```
GETNUM %a " Angolo?" 0 10
SIN %s %a
COS %c %a
TAN %t %a
STR $a %a
STR $s %s
STR $c %c
MSG " Angolo, seno, coseno:" $a $s $c
STR $t %t
MSG " Angolo, tangente:" $a $t ""
ATAN %a1 %s %c
STR $a1 %a1
MSG " Angolo in base a seno e coseno:" $a1 "" ""
```

L'esempio mostra l'utilizzo delle funzioni trigonometriche.

CURRENT %current

%current .....corrente o potenza [%] (0÷100)

Il comando permette di impostare la corrente o potenza dello stile di marcatura. Se il valore è nullo il laser non verrà mai acceso durante la marcatura (può essere utile per eliminare parti di logo associabili a penne - e quindi stili - di una tabella stili di marcatura). Se il valore è superiore a zero il laser verrà acceso durante la marcatura e, se presente nell'hardware della marcatrice l'opzione "corrente/frequenza automatiche", la corrente verrà anche impostata sull'alimentatore, viceversa dovrà essere cambiata manualmente tramite potenziometro.

Vedere anche: FREQUENCY, FROTATION, RESOLUTION, SPEED, WDEFINITION, WDIMENSION, WNAME, WNUMBER

Esempio:

```
CURRENT 10
```

DATE $date

$date = ...data di sistema [gg/mm/aaaa] (01/01/1980÷31/12/2099)

Il comando restituisce la data di sistema.

Vedere anche: TIME, GETRTCTIME, FDATE, ELAPSED

Esempio:

```
DATE $d
```

DEFSEG %segment

%segment.....segmento di memoria (0÷65535)

Il comando seleziona il segmento di memoria dove i comandi PEEK e POKE devono lavorare. Da notare che, per sicurezza, POKE non lavora se il segmento è fuori dall'intervallo C800h÷EFC0h (51200÷61376).

Vedere anche: PEEK, POKE, IN, OUT

Esempio:

```
DEFSEG 53248 && D000h
```

EDIT $file %protection

$file .....file testo (<>"")
%protection.....protezione: 0=no (modifica), 1=si (visualizzazione)

Il comando permette la visualizzazione o la modifica di un file di testo.

Vedere anche: MSG, ERRMSG, GETNUM, GETSTR, FIELD

Esempio:

```
EDIT "axes.scr" 1
```

ELAPSED %secs

%secs =.....secondi dalla mezzanotte

Il comando restituisce i secondi passati dopo la mezzanotte. La precisione è di 1/18 di secondo. Non è possibile utilizzare questo comando per cronometrare le marcature perchè in quel caso l'interrupt che aggiorna l'orologio è inattivo. Utilizzare invece GETRTCTIME, l'orologio completamente indipendente.

Vedere anche: DATE, TIME, GETRTCTIME, WAIT

Esempio:

```
ELAPSED %now
```

ELSE

Vedere struttura IF

ENCODER %speed

%speed =.....velocità di rotazione [%] (-100÷100)

Il comando restituisce la velocità di rotazione dell'encoder del volantino elettronico; il segno indica la direzione: positivo se oraria, negativo se antioraria. Lo zero indica ovviamente il volantino fermo.

Vedere anche: VOLCMD, VOLDATA, VOLKEYS

Esempio:

```
ENCODER %es
```

END

Il comando termina lo script.

Vedere anche: RUN, GOSUB, RETURN

Esempio:

```
EXIST %e "evlaserr.log"
IF= %e 0
     HPGLCONV "evlaserr.plt"
ENDIF
GOSUB :scritta
GOSUB :logo
END

:scritta
     LOCATE 0 20
     PRINT "EVLASER"
RETURN

:logo
     LOCATE 0 0
     EXTLOGO "evlaserr" 0
RETURN
```

L'esempio esegue scritta, logo e termina. Se non vi fosse l'END verrebbe marcata ancora la scritta e vi sarebbe un errore dovuto alla presenza di un RETURN pur non essendovi stato un GOSUB.

ENDIF

Vedere struttura IF

ERRMSG $line1 $line2 $line3 $line4

$line1.....prima linea del messaggio d'errore
$line2.....seconda linea del messaggio d'errore
$line3.....terza linea del messaggio d'errore
$line4.....quarta linea del messaggio d'errore

Il comando mostra una finestra contenente il messaggio d'errore (da una a quattro linee) ed attende conferma. A differenza del comando MSG in questo caso viene emesso un segnale audio (se abilitato) per attirare l'attenzione. In caso i parametri siano vuoti viene visualizzato il messaggio:

```
   ERRORE NON SPECIFICATO
Premere ENTER per continuare
```

Vedere anche: MSG, YESNO, SOUND

Esempio:

```
ERRMSG "File" $f "non trovato!" ""
```

EXIST %exist $file

%exist =.....file esistente? (0=no, -1=si)
$file .....file da testare (<>"")

Il comando controlla l'esistenza del file specificato. ATTENZIONE: il valore restituito nel caso il file esista è -1 e non 1!

Vedere anche: FFIRST, FNEXT, FLIST

Esempio:

```
EXIST %e "evlaserr.log"
```

EXTLOGO $logo %table

$logo .....nome del logo (.log)
%table.....stili di marcatura da tabella? (0=no, 1=si)

Il comando permette la marcatura di un logo alla posizione, scala (v. LDIM) e rotazione (v. LROTATION) correnti. Si può determinare se il logo deve essere marcato con lo stile di marcatura corrente o se invece si vuole associare ogni penna (o colore, da 1 a 8) allo stile relativo nella tabella stili di marcatura corrente. Assicurarsi dunque che il logo abbia delle penne corrispondenti a stili effettivamente esistenti anche nella tabella. Al termine della marcatura del logo i parametri vecchi vengono ripristinati. La posizione corrente diventa l'ultimo punto posizionato nel logo.

Vedere anche: PRINT, LINE, BOX, LOCATE, ABSOLUTE, RELATIVE, INGLOGO, LDIMENSION, LROTATION

Esempio:

```
EXTLOGO "evlaserr" 0
```

FCLOSE %handle

%handle.....gestore del file (1÷65535)

Il comando chiude un file precedentemente aperto con FOPEN.

Vedere anche: FOPEN, FREAD, FWRITE, FLNREAD, FLNWRITE, FLOCATE, FFIND, FPOS, LOF, FDATE

Esempio:

```
FCLOSE %h
```

FDATE %date %time %handle

%date =.....data del file [aaaammgg] (19800101÷20991231)
%time =.....ora del file [oommss] (0÷235959)
%handle.....gestore del file (1÷65535)

Il comando restituisce data e ora di un file aperto.

Vedere anche: FPOS, LOF, FOPEN, DATE, TIME, GETRTCTIME

Esempio:

```
FDATE %d %t %h
```

FFIND %position %handle $text %record

%position =.....posizione del file trovata (0=non trovata, 1÷65536²-1)
%handle.....gestore del file (1÷65535)
$text.....testo di ricerca (<>"")
%record .....lunghezza del record (1÷65535)

Questo comando permette di eseguire una ricerca, a partire dalla posizione corrente del file, del testo specificato. Ogni volta che il testo non viene trovato il comando salta avanti nel file della quantità di bytes specificati come record; questo per aumentare la velocità nel caso si debba cercare un testo in una posizione ben precisa all'interno di un record e nel caso di un file a lunghezza record fissa. Negli altri casi si può utilizzare il valore 1. Per eseguire la ricerca dall'inizio del file è necessario prima posizionarvisi (FLOCATE %han 1). Il comando ritorna zero quando la ricerca non va a buon fine. Il primo byte del file è il numero 1. Il comando non è molto veloce e non è fatto per lavorare su grossi file.

Vedere anche: FLOCATE, LOF, FPOS, FREAD, FWRITE, FLNREAD, FLNWRITE, FOPEN

Esempio:

```
FFIND %p %h "TESTO=" 1
```

FFIRST $found $pattern

$found = ...file o directory trovata
$pattern .....pattern di ricerca

Il comando permette di cercare il primo file che concorda con il pattern di ricerca. Il pattern può contenere wildcards (? e *) e deve cominciare con "<" per cercare directory piuttosto che file. Se non vengono trovati file o directory viene restituita una stringa vuota. Utilizzare FNEXT per continuare la ricerca.

Vedere anche: FNEXT, EXIST, FLIST

Esempio:

```
FFIRST $f "ax*.sc?"
```

FIELD %code $field $default %x %y %length %video %number %foreground %background

%code =.....codice di uscita (0÷7, 11÷20; v. tabella)
$field = ...contenuto del campo editato
$default .....contenuto iniziale del campo
%x.....colonna iniziale del campo (1÷80)
%y.....riga del campo (1÷25)
%length.....lunghezza campo reale (1÷31)
%video.....lunghezza campo visualizzata (1÷31)
%number .....consentite solo cifre? (0=no, 1=si)
%foreground .....colore di primo piano (0÷31)
%background.....colore di sfondo (0÷7)

Il comando permette l'editing controllato di un campo a video. Alla pressione di uno dei tasti presenti in tabella il comando termina restituendo il relativo codice oltre che all'effettivo contenuto del campo. Lo script poi, in base al codice, opererà opportunamente (per esempio scarterà il campo se il codice era 1=ESC, lo manterrà se il codice era 0=Enter, darà un messaggio di aiuto se il codice era 11=F1). Nel caso la lunghezza a video sia inferiore a quella reale, con le freccie orizzontali, Home, End, digitando lettere o cancellandole il campo potrà scrollare orizzontalmente e non occupare spazio prezioso a video. Il campo non viene cancellato automaticamente dallo schermo. Questo potente comando può venire utile per creare complicate maschere video.

Tasto	%cod
Enter	0
Esc	1
->	2
<-	3

```
| TAB    |   4   |
| Sh-TAB |   5   |
| PageDn |   6   |
| PageUp |   7   |
| F1÷F10 | 11÷20-|
+----------------+
```

Vedere anche: GETSTR, GETNUM, VCLS, VCOLOR, VLOCATE, VPRINT

Esempio:

```
FIELD %c $f "testo123" 10 30 20 15 0 14 1
```

FLIST $file $title $pattern %new $init

$file = ...file selezionato o creato
$title .....titolo della lista
$pattern.....pattern della lista di file
%new.....è possibile creare un file? (0=no, 1=si)
$init.....nome iniziale nella lista

Il comando permette di selezionare da una lista di file quello desiderato oppure crearne uno nuovo.

Vedere anche: FFIRST, FNEXT, MENU

Esempio:

```
FLIST $f "Nome logo?" "*.plt" 0 "evlaserr.plt"
```

FLNREAD $line %handle

$line = ...linea letta
%handle.....gestore del file (1÷65535)

Il comando legge sequenzialmente una linea da un file aperto ovviamente in lettura.
Il codice ASCII 0 non è gestito.
ATTENZIONE: la linea, se con un numero di caratteri maggiore di 31, viene troncata.

Vedere anche: FREAD, FLNWRITE, FWRITE, FLOCATE, FFIND, FOPEN

Esempio:

```
FLNREAD $l %h
```

FLNWRITE $line %handle

$line .....linea da scrivere
%handle.....gestore del file (1÷65535)

Il comando scrive sequenzialmente una linea in un file aperto ovviamente in scrittura (più semplicemente il comando aggiunge automaticamente CR/LF dopo i caratteri specificati).

Vedere anche: FWRITE, FLNREAD, FREAD, FLOCATE, FFIND, FOPEN

Esempio:

```
FLNWRITE "linea di esempio ->" %h
```

FLOCATE %handle %pos

%handle.....gestore del file (1÷65535)
%pos.....posizione nel file (1÷65536²-1)

Il comando permette di posizionarsi all'interno del file per una successiva lettura o scrittura. Il primo byte del file è il numero 1.

Vedere anche: FFIND, FPOS, LOF, FREAD, FWRITE, FLNREAD, FLNWRITE, FOPEN, FCLOSE

Esempio:

```
FLOCATE %h 12000
```

FNAME $font

$font .....font di caratteri (.fon)

Il comando permette di selezionare il font dello stile di testo.
Di seguito la lista dei font normalmente utilizzabili:

Font	Prop?	Descrizione
0961A___	N	Egyptian 505 Light (Windows©)
1176A___	N	Empire (Windows©)
COMPLEX	N	Romanico Complesso 10/23/91 (AutoCAD©)
CYRILLIC	N	Cirillico complesso (AutoCAD©)
CYRILTLC	N	Cirillico complesso (AutoCAD©)
E006004T	N	Englische Schreibschrift T Demi Bold (Windows©)
EKLEKTIN	N	Eklektic Plain (Windows©)
GOTHICE	N	Gotico inglese (AutoCAD©)
GOTHICG	N	Gotico tedesco (AutoCAD©)
GOTHICI	N	Gotico italiano (AutoCAD©)
GREEKC	N	Greco complesso (AutoCAD©)
GREEKS	N	Greco semplice (AutoCAD©)
ISO9	N	Font ISO esteso per ISO 8859-1 (AutoCAD©)
ITALIC	N	Italico (AutoCAD©)
ITALICC	N	Italico complesso (AutoCAD©)
ITALICT	N	Italico triplice (AutoCAD©)
MONOTXT	S	Font TXT monospaziato (AutoCAD©)
ROMANC	N	Romanico complesso (AutoCAD©)
ROMAND	N	Romanico duplice (AutoCAD©)
ROMANS	N	Romanico semplice (AutoCAD©)
ROMANT	N	Romanico triplice (AutoCAD©)
SCRIPTC	N	Script complesso (AutoCAD©)
SCRIPTS	N	Script semplice (AutoCAD©)
SIMPLEX	N	Romanico semplice (AutoCAD©)
SYASTRO(*)	N	Simboli astronomici (AutoCAD©)
SYMAP(*)	N	Simboli mappature (AutoCAD©)
SYMATH(*)	N	Simboli matematici (AutoCAD©)
SYMETEO(*)	N	Simboli meterologici (AutoCAD©)
SYMUSIC(*)	N	Musica ed elettronica (AutoCAD©)
TXT	S	Font standard (AutoCAD©)
WINGDING	N	Wingdings (Windows©)

(*) riferimento altezza sul carattere 241 (±) invece di 65 (A).

Vedere anche: HEIGHT, ITALIC, SDEFINITION, SNAME, SNUMBER, SROTATION, WIDTH, PRINT

Esempio:

```
FNOME "iso9"
```

FNEXT $found

$found = ...file o directory trovata

Il comando permette di continuare la ricerca specificata con FFIRST.

Vedere anche: EXIST, FFIRST, FLIST

Esempio:

```
FNEXT $f
```

FOPEN %handle $file $mode

%handle =.....gestore del file (0=errore, 1÷65535)
$file .....nome del file
$mode .....modo di accesso (i=input, o=output, a=append, r=random)

Il comando permette l'apertura in lettura o scrittura di un file ritornandone il gestore. Il modo 'random' permette sia la lettura che la scrittura.
Utilizzare FREAD o FLNREAD per leggere, FWRITE o FLNWRITE per scrivere, FLOCATE o FFIND per posizionarsi, FPOS, FDATE, LOF per avere informazioni sul file ed FCLOSE per chiudere.
E` possibile aprire molti file contemporaneamente (il numero massimo dipende dalla specifica FILES nel CONFIG.SYS).

N. B.:

- se il gestore ritornato è 0 significa che il file non è stato aperto per qualche problema
- il codice ASCII 0 non è gestito!
- memorizzare con attenzione in gestore del file per poterlo gestire e chiudere successivamente; in particolare dimenticare di chiudere un file aperto in scrittura può significare perdere dei dati o, peggio, rovinare il file

Vedere anche: FCLOSE, FREAD, FWRITE, FLNREAD, FLNWRITE, FLOCATE, FFIND, FPOS, LOF, FDATE

Esempio:

```
FOPEN %h "c:\092config.sys" "i"
```

FPOS %position %handle

%position = ... posizione corrente nel file
%handle..... gestore del file (1÷65535)

Il comando resituisce la posizione corrente all'interno del file.
Questa posizione viene spostata in avanti ad ogni lettura o scrittura, impostata direttamente tramite FLOCA TE, trovata tramite FFIND. Il primo byte del file è il numero 1.

Vedere anche: FLOCATE, LOF, FFIND, FOPEN, FCLOSE

Esempio:

```
FPOS %p %h
```

FREAD $read %handle %number

$read = ... caratteri letti
%handle..... gestore del file (1÷65535)
%number ..... numero di bytes da leggere (0÷31)

Il comando legge sequenzialmente da un file aperto in lettura un numero ben preciso di bytes.
Il codice ASCII 0 non è gestito.

Vedere anche: FWRITE, FLNREAD, FLNWRITE, FOPEN, FCLOSE, FLOCATE, FFIND

Esempio:

```
FREAD $r %h 10
```

FREQUENCY %frequency

%frequency.....frequenza di modulazione [%] (0÷100)

Il comando permette di impostare la frequenza dello stile di marcatura.
Se il valore è nullo la marcatura avverrà in modalità continua.
Se il valore è superiore a zero la marcatura avverrà in modalità impulsata e, se presente nell'hardware della marcatrice l'opzione "corrente/frequenza automatiche", la frequenza verrà anche impostata sul modulatore, viceversa dovrà essere cambiata manualmente tramite potenziometro.

Vedere anche: CURRENT, FROTATION, RESOLUTION, SPEED, WDEFINITION, WDIMENSION, WNAME, WNUMBER

Esempio:

```
FREQUENCY 94
```

FROTATION %frequency

%frequency.....frequenza di rotazione [Hz] (0÷255)

Il comando imposta la frequenza di ripetizione del pattern dello stile di marcatura utilizzando la scheda hardware (si dice anche rotazione in quanto il pattern è normalmente un cerchio).

Vedere anche: CURRENT, FREQUENCY, RESOLUTION, SPEED, WDEFINITION, WDIMENSION, WNAME, WNUMBER

Esempio:

```
FROTATION 50
```

FWRITE %handle $text

%handle.....gestore file (1÷65535)
$text.....testo da scrivere

Il comando scrive sequenzialmente in un file aperto in scrittura il testo specificato.
Il codice ASCII 0 non è gestito.

Vedere anche: FREAD, FLNWRITE, FLNREAD, FLOCATE, FFIND, FPOS, FOPEN

Esempio:

```
FWRITE %h "Acusto-Ottico"
```

GETAXIS %positionx %positiony %positionz %positionr

%positionx =.....posizione corrente dell'asse X [mm]
%positiony =.....posizione corrente dell'asse Y [mm]
%positionz =.....posizione corrente dell'asse Z [mm]
%positionr =.....posizione corrente dell'asse R [mm]

Il comando ritorna le posizioni correnti dei quattro assi.
Utilizzare sempre questo comando invece di prevedere gli spostamenti dopo AXIS o AXRESET in quando potrebbero collidere con finecorsa software o dare errori di comunicazione seriale. I finecorsa meccanici possono invece bloccare gli assi senza che il software se ne possa accorgere: impostare quindi i finecorsa software correttamente e non ignorare errori di movimentazione oltre gli stessi.

Vedere anche: AXIS, AXRESET, AXSET, RITAXIS

Esempio:

```
GETAXIS %posx %posy %posz %posr
AXIS %posx %posy %posz %posr 20 1 0

VCLS
VPRINT "* MOVIMENTAZIONE ASSI *\013\013"
VPRINT "ESC = uscire, * = reset, "
VPRINT "S = settaggi\013"
VPRINT "X/Y/Z/R = posizione assi:"
GOSUB :posassi

:loop
```

```
INKEY $t
IF= $t ""
      GOTO :loop
ENDIF
UCASE $t $t
IF= $t "\027" && ESC
      END
ENDIF
IF= $t "*" && *
      AXRESET
ENDIF
IF= $t "S" && S
      AXSET
ENDIF
IF= $t "X"
      GETNUM %posx "Posizione asse X?" %posx 10
ENDIF
IF= $t "Y"
      GETNUM %posy "Posizione asse Y?" %posy 10
ENDIF
IF= $t "Z"
      GETNUM %posz "Posizione asse Z?" %posz 10
ENDIF
IF= $t "R"
      GETNUM %posr "Posizione asse R?" %posr 10
ENDIF
GOSUB :posassi
GOTO :loop

:posassi
      AXIS %posx %posy %posz %posr 20 1 0
      GETAXIS %posx %posy %posz %posr
      VLOCATE 4 27
      STR $p %posx
      VPRINT $p
      VPRINT ", "
      STR $p %posy
      VPRINT $p
      VPRINT ", "
      STR $p %posz
      VPRINT $p
      VPRINT ", "
      STR $p %posr
      VPRINT $p
      VPRINT " "
RETURN
```

L'esempio permette il settaggio, il reset ed il posizionamento degli assi in modo semplice e guidato.

GETCFG $value $item

$value =.....valore della voce (v. tabella)
$item .....voce (v. tabella)

Il comando permette di conoscere il valore correntemente impostato nel file di configurazione (LASER2.CFG) relativo alla voce specificata.
Le voci disponibili sono le seguenti:

OBJECT_EDGE	lato obiettivo (<>0) [mm]
MONOCHROMATIC	soppressione colori? (TRUE/FALSE)
QUIET_SOUND	soppressione suono? (TRUE/FALSE)
MOUSE_ENABLE	abilitazione mouse? (TRUE/FALSE)
CURRENT_WOBBLE_FILE_NAME	tabella stili di marcatura corrente (.wob)
CURRENT_STYLE_FILE_NAME	tabella stili di testo corrente (.stl)
CURRENT_LIBRARY_FILE_NAME	libreria logo corrente (.lib)
CURRENT_SCR_FILE_NAME	script corrente (.scr)
OFFSET_X	offset X [mm]
OFFSET_Y	offset Y [mm]
HEX_CARRIER_ADDRESS	indirizzo carrier [esa] (0÷FFFF)
MODULE_NUMBER	modulo Burr Brown (0÷3; 0=assente)
HEX_VOBUL_ADDRESS	ind. vobulatore elettronico [esa] (0÷FFFF)
QS_LASER_ON_DELAY	attesa dopo laser on ad impulsi [µs]
CW_LASER_ON_DELAY	attesa dopo laser on in continua [µs]
QS_LASER_OFF_DELAY	attesa prima di laser off ad impulsi [µs]
CW_LASER_OFF_DELAY	attesa prima di laser off in continua [µs]
QS_JUMP_DELAY	attesa dopo salto ad impulsi [µs]
CW_JUMP_DELAY	attesa dopo salto in continua [µs]
QS_EDGE_DELAY	attesa spigolo oltre limite ad impulsi [µs]
CW_EDGE_DELAY	attesa spigolo oltre limite in continua [µs]
CHANGE_FREQUENCY_DELAY	attesa dopo cambio frequenza [ms]
CHANGE_CURRENT_DELAY	attesa dopo cambio corrente [ms]
CHANGE_DIAFRAM_DELAY	attesa dopo cambio diaframma [ms]
CLOSE_A/O_DELAY	attesa chiusura A/O [µs]
OPEN_A/O_DELAY	attesa apertura A/O [µs]
CLOSE_SHUTTER_DELAY	attesa chiusura shutter [µs]
OPEN_SHUTTER_DELAY	attesa apertura shutter [µs]
AXIS_POS_CHANGE_DELAY	attesa dopo movimentazione assi [ms]
X_GALVO_CALIBRATION	calibrazione galvo X [%] (0÷100)
Y_GALVO_CALIBRATION	calibrazione galvo Y [%] (0÷100)
X_CORRECTION_FLAG	correzione galvo X? (TRUE/FALSE)
Y_CORRECTION_FLAG	correzione galvo Y? (TRUE/FALSE)
X_CORRECTION_FACTOR	fattore di correzione galvo X [~mm]
Y_CORRECTION_FACTOR	fattore di correzione galvo Y [~mm]
XY_EXCHANGE	scambio canali X/Y? (TRUE/FALSE)
5µs_LOOPS	loop processore per 5µs
ms_LOOPS	loop processore per 1ms
X_HPGL_CALIBRATION	calibrazione X HPGL [u/mm]
Y_HPGL_CALIBRATION	calibrazione Y HPGL [u/mm]
HPGL_SCALE_FACTOR	scala di conversione HPGL [%]
X_HPGL_OFFSET	offset X di conversione HPGL [u] (±65536)
Y_HPGL_OFFSET	offset Y di conversione HPGL [u] (±65536)
HPGL_SWAP	scambio X/Y in conv. HPGL? (TRUE/FALSE)
LIMIT_ANGLE	angolo limite in conv. HPGL [°] (0÷180)
SIMULATION_DELAY	ritardo di simulazione [~µs/pt]
ELECTR_VOL_DELAY	ritardo scrittura su volantino elettr. [µs]
SIMULATION_VIDEO_MODE	modo video simulazione (CGAL/CGAH/EGA/VGA)
SIMULATION_GRID	griglia in simulazione? (TRUE/FALSE)
SIMULATION_GRID_DIM	dim. griglia di simulazione [mm] (<>0)
SIMULATION_CALIBRATION	calibrazione in simulazione? (TRUE/FALSE)
SIMULATION_CORRECTION	correzione in simulazione? (TRUE/FALSE)
INTERLOCK_ENABLE	abilitazione interlock? (TRUE/FALSE)
MIN_INTERLOCK_LOOPS	numero minimo di interlock consecutivi [#]
SPEED_FACTOR_A	fattore di velocità A
SPEED_FACTOR_B	fattore di velocità B
EV_TYPE	tipo marcatrice (EV50/EV80/EV120/EVCO2)
A/O_INVERTED	segnale A/O invertito? (TRUE/FALSE)
RF_INVERTED	segnale RF invertito? (TRUE/FALSE)
LANGUAGE	linguaggio (Italiano/English/Deutsch/...)
MAX_JUMP	salto dinamico massimo percentuale [%]
JUMP_SPEED	velocità salto dinamico [mm/s]

ELECT_CURR_ADJ_DELAY	ritardo scrittura potenziometro elettr. [µs]
ELECT_CURR_ENABLE	abil. potenziometro elettr.? (TRUE/FALSE)
AXIS_COM	porta seriale mov. assi (0/1/2; 0=disab.)
X_AXIS_ENABLE	abilitazione asse lineare X? (TRUE/FALSE)
Y_AXIS_ENABLE	abilitazione asse lineare Y? (TRUE/FALSE)
Z_AXIS_ENABLE	abilitazione asse lineare Z? (TRUE/FALSE)
R_AXIS_ENABLE	abilitazione asse rotante R? (TRUE/FALSE)
X_AXIS_STEP_MM	calibrazione asse lineare X [pp/mm]
Y_AXIS_STEP_MM	calibrazione asse lineare Y [pp/mm]
Z_AXIS_STEP_MM	calibrazione asse lineare Z [pp/mm]
R_AXIS_STEP_MM	calibrazione asse rotante R [pp/°]
X_AXIS_POSITION	posizione asse lineare X [mm]
Y_AXIS_POSITION	posizione asse lineare Y [mm]
Z_AXIS_POSITION	posizione asse lineare Z [mm]
R_AXIS_POSITION	posizione asse rotante R [°]
X_AXIS_MAX	finecorsa sw asse lineare X [mm] (0=disab.)
Y_AXIS_MAX	finecorsa sw asse lineare Y [mm]
Z_AXIS_MAX	finecorsa sw asse lineare Z [mm]
R_AXIS_MAX	finecorsa sw asse rotante R [°]
X_AXIS_ZERO	zero software asse lineare X [mm]
Y_AXIS_ZERO	zero software asse lineare Y [mm]
Z_AXIS_ZERO	zero software asse lineare Z [mm]
R_AXIS_ZERO	zero software asse rotante R [°]
AXIS_SPEED	ritardo mov. assi [100/(m/min)-28]
AXIS_NUMBER	tipo EPROM controllore assi (3/4=num. assi)
FREQ_MIN	frequenza corrispondente a 0% [KHz]
FREQ_MAX	frequenza corrispondente a 100% [KHz]
CURR_MIN	corrente corrispondente a 0% [A/%/W/...]
CURR_MAX	corrente corrispondente a 100% [A/%/W/...]
CURR_UNIT	unità di misura della corrente [A/%/W/...]
TABVOB_FIELDS_ENABLE	abil. PRFZ tab. stili marc.? (TRUE/FALSE)
DEBUG_LEVEL	livello di debugging [#] (0/...)
JUMP_COLOR	colore salti in simulazione (0÷15;0=disab.)
CO2_FREQ	laser CO^2 in frequenza? (TRUE/FALSE)

Se la descrizione è interrogativa la voce restituisce solo TRUE o FALSE (vero o falso).

Vedere anche: GETSYSN, PUTCFG, PUTSYSN

Esempio:

```
GETCFG $tabvob "CURRENT_WOBBLE_FILE_NAME"
```

GETNUM %value $prompt %default %length

%value =.....valore restituito
$prompt.....descrizione della richiesta
%default.....valore proposto in partenza
%length.....numero massimo di cifre (1÷31)

Il comando richiede tramite una finestra di dialogo un valore.

Vedere anche: GETSTR, FIELD, YESNO, EDIT, FLIST, MENU

Esempio:

```
GETNUM %spd "Velocità [mm/s]?" 300 5
```

GETRTCTIME %time %precision

%time =.....ora corrente in secondi (fino a 3 decimali)
%precision.....precisione richiesta (0=±.5s, 1=±31.5ms)

Il comando permette di ottenere l'ora corrente del Real Time Clock con precisione bassa (ma immediato) o con precisione elevata (ma con attesa da 0 ad 1 secondo). Nel secondo caso l'ora restituita è quella all'inizio del comando che prima di continuare attende lo scadere del secondo successivo. Per una precisa e corretta cronometraggio di un evento si dovrà operare nel seguente modo:

- memorizzare con alta precisione l'ora iniziale
- eseguire l'operazione da cronometrare
- memorizzare con alta precisione l'ora finale
- se l'ora iniziale ha decimali toglierli ed aggiungere 1 secondo
- la differenza tra tempo finale ed iniziale è il tempo impiegato

Vedere anche: DATE, TIME, WAIT, FDATE, ELAPSED

Esempio:

```
LOCATE 0 0
GETRTCTIME %t0 1
EXTLOGO "evlaserr" 0
GETRTCTIME %t1 1
INT %tmp %t0
IF<> %tmp %t0
     + %t0 %tmp 1
ENDIF
- %tmp %t1 %t0
STR $t %tmp
MSG " Tempo impiegato [s]:" $t "(errore max: ±31.5ms)" ""
```

L'esempio mostra il cronometraggio della marcatura di un logo.

GETSTR $text $prompt $default %length

$text = ...testo restituito
$prompt.....descrizione della richiesta
$default.....testo proposto in partenza
%length.....lunghezza massima del testo (1÷31)

Il comando richiede tramite una finestra di dialogo un testo.

Vedere anche: GETNUM, FIELD, YESNO, EDIT, FLIST, MENU

Esempio:

```
GETSTR $s "Come ti chiami?" "MARIO" 31
```

GETSYSN %value %code

%value =.....valore della variabile di sistema
%code.....codice della variabile di sistema (0÷255; v. tabella)

Il comando permette di leggere il valore di alcune variabili di sistema.
Di seguito i codici relativi alle varie variabili di sistema:

0 OFFSETX	64 D	128 BPP	192 NILOCKS
1 OFFSETY	65 SCALEX	129 RT	193 LOCX
2 SROTATION	66 SCALEY	130 RATIO	194 LOCY
3 DEFSEG	67 BASESX	131 RATIOP	195 ATTRIB
4 CARRIER	68 BASESY	132 ANG	196 CFACT
5 MODULE	69 ROTAZ	133 AI	197 RITEDGEC
6 VOBULADD	70 BASERX	134 AF	198 OBJ
7 ---	71 BASERY	135 PER	199 MONO
8 KSPEEDB	72 N	136 LARCO	200 RITEDGEQ
9 RITLONQ	73 NP	137 DELTAANG	201 O1
10 RITLONC	74 DELTAX	138 AX8	202 O2
11 RITLOFFQ	75 DELTAY	139 BX8	203 O3
12 RITLOFFC	76 PASSOX	140 CX8	204 O4
13 RITJUMPQ	77 PASSOXP	141 DX8	205 ---
14 RITJUMPC	78 PASSOY	142 EX8	206 ---
15 RITFREQ	79 PASSOYP	143 ---	207 ---
16 RITCURR	80 DIST	144 ---	208 ---
17 RITDIAF	81 E	145 ---	209 ---
18 RITCAO	82 EP	146 ---	210 ---
19 RITCSH	83 ---	147 P0	211 ---
20 RITAAO	84 ---	148 P1	212 ---
21 RITASH	85 FUORIX	149 P2	213 ---
22 RITAXIS	86 FUORIY	150 P3	214 ---
23 ---	87 LAMP	151 P4	215 ---
24 ---	88 ILOCKS	152 P5	216 ---
25 ---	89 LMERR1	153 ---	217 ---
26 CALX	90 TIPO	154 ---	218 ---
27 CALY	91 ---	155 ALLIN	219 ---
28 CXFL	92 EXITFLAG	156 ALLINFIT	220 ---
29 CYFL	93 LASER	157 ORFLAG	221 ---

```
| 30 CORRX     |  94 ---      | 158 ---        | 222 ---      |
| 31 CORRY     |  95 AO       | 159 LASTX      | 223 ---      |
| 32 MICRO     |  96 SH       | 160 LASTY      | 224 ---      |
| 33 MILLI     |  97 XV       | 161 XSIM       | 225 ---      |
| 34 ---       |  98 YV       | 162 YSIM       | 226 ---      |
| 35 RITSIM    |  99 ---      | 163 SIMXS      | 227 ---      |
| 36 VMODE     | 100 ---      | 164 SIMYS      | 228 ---      |
| 37 MINILOCKS | 101 ---      | 165 SIMXO      | 229 ---      |
| 38 KSPEEDA   | 102 ---      | 166 SIMYO      | 230 ---      |
| 39 PDIM      | 103 ---      | 167 SIMRT      | 231 ---      |
| 40 PATNUM    | 104 ---      | 168 ---        | 232 ---      |
| 41 VEL       | 105 ---      | 169 ---        | 233 ---      |
| 42 RISOL     | 106 ---      | 170 MARCARE    | 234 ---      |
| 43 PRISOL    | 107 X        | 171 ---        | 235 ---      |
| 44 FREQ      | 108 Y        | 172 ---        | 236 ---      |
| 45 CURR      | 109 ---      | 173 ---        | 237 ---      |
| 46 DIAF      | 110 ---      | 174 ---        | 238 ---      |
| 47 FROTAZ    | 111 ---      | 175 ---        | 239 ---      |
| 48 ALT       | 112 ---      | 176 SVX        | 240 ---      |
| 49 LARG      | 113 XCEN     | 177 SVY        | 241 ---      |
| 50 CSPACE    | 114 XCENP    | 178 SVPASSOX   | 242 ---      |
| 51 LSPACE    | 115 YCEN     | 179 SVPASSOY   | 243 ---      |
| 52 ITALIC    | 116 YCENP    | 180 SVN        | 244 ---      |
| 53 CHGXY     | 117 RELS     | 181 SVDELTAANG | 245 ---      |
| 54 VEL1      | 118 ---      | 182 SVAI       | 246 ---      |
| 55 FREQ1     | 119 ---      | 183 SVRATIO    | 247 ---      |
| 56 ---       | 120 ---      | 184 SVE        | 248 ---      |
| 57 FSTATE    | 121 TMP      | 185 SVA        | 249 ---      |
| 58 ERAON     | 122 TMP1     | 186 SVB        | 250 ---      |
| 59 ERAONP    | 123 TMPX     | 187 SVXCEN     | 251 ---      |
| 60 ORIGINX   | 124 TMPY     | 188 SVYCEN     | 252 ---      |
| 61 ORIGINY   | 125 A        | 189 PATFLAG    | 253 ---      |
| 62 MIRRX     | 126 AP       | 190 BEGFLAG    | 254 ---      |
| 63 MIRRY     | 127 B        | 191 NSTART     | 255 ---      |
+-------------------------------------------------------------+
```

Vedere anche: GETCFG, PUTCFG, PUTSYSN

Esempio:

```
GETSYSN %txth 48 && ALT
```

GOSUB :label

label .....nome etichetta

Il comando fa si che il programma salti momentaneamente ad una routine e, al termine della stessa, continui all'istruzione successiva. La routine deve iniziare con il nome dell'etichetta e terminare con un RETURN.
I GOSUB possono essere nidificati fino ad un massimo di 40.

Il numero massimo di etichette è circa 500.

Vedere anche: RETURN, GOTO, $INCLUDE, RUN

Esempio:

```
        ABSOLUTE                && 1 = prima chiamata
        LOCATE 10 0         && 2 = primo ritorno
  +---- GOSUB :3ang         && 3 = seconda chiamata
  | +-> LOCATE 20 0         && 4 = secondo ritorno
1 | |   GOSUB :3ang ---+
  | |   END <-----+    | 3
  | |             |    |
  +-|-> :3ang <---|----+
    | RELATIVE    |
  2 | LINE 10 10  | 4
    | LINE 0 -20  |
    | LINE -10 10 |
    | ABSOLUTE    |
    +-- RETURN ---+
```

L'esempio marca due triangoli allineati orizzontalmente; le frecce indicano il flusso del programma, i numeri le sequenze.

GOTO :label

label .....nome etichetta

Il comando fa si che il programma salti all'etichetta indicata.
Il numero massimo di etichette è circa 500.

Vedere anche: GOSUB

Esempio:

```
LET %num 0
+-> :inizio <---------+
|   START %st         |
|   IF= %st 0         |
|   GOTO :inizio -----+
|   ENDIF
|   LOCATE -10 10
|   PRINT " Numero seriale: "
|   LOCATE -5 5
|   STR $s %num
|   PRINT $s
|   + %num %num 1
+-- GOTO :inizio
```

Il programma d'esempio marca un numero seriale consecutivo ogni volta che viene premuto il pulsante o pedale di START.

HEIGHT %height

%height.....altezza dei caratteri [mm] (<>0)

Il comando imposta l'altezza dello stile di testo.

Vedere anche: FNAME, ITALIC, SDEFINITION, SNAME, SNUMBER, SROTATION, WIDTH, PRINT

Esempio:

```
HEIGHT 3.2
```

HPGLCONV $file

$file .....file HPGL (.plt)

Il comando permette di eseguire una conversione dal file HPGL (.plt) al file logo (.log). L'estensione è obbligatoria. Nel caso il parametro sia vuoto viene presentata una lista di selezione.
Prima di eseguire la conversione conviene controllare se il .log esiste già ed ha data ed ora uguali al .plt: in questo caso la conversione è ovviamente inutile.

Vedere anche: FLIST, AXSET, TABSTILI, TABVOB

Esempio:

```
HPGLCONV "evlaserr.plt"
```

IF< %value1 %value2

...

```
        ...         (blocco istruzioni THEN)
        ...
[ELSE]
        [...]
        [...]       (blocco opzionale istruzioni ELSE)
        [...]
ENDIF
```

%value1.....primo valore numerico
%value2.....secondo valore numerico

Il comando permette di confrontare due valori numerici o due stringhe. Nel caso il primo numero sia minore del secondo o la prima stringa venga alfabeticamente prima della seconda verranno eseguite le istruzioni del blocco THEN altrimenti quelle del blocco ELSE.
Il blocco di istruzioni ELSE è opzionale.
Ogni IF deve essere chiuso da un ENDIF.
Gli IF possono essere nidificati senza apparente limite.
Le stringhe non possono essere testate con questo tipo di IF.

Vedere anche: IF>, IF<=, IF>=, IF<>, IF=, ELSE, ENDIF

Esempio:

```
IF< %età 18
      MSG "Mi spiace: solo per adulti" "" "" ""
ENDIF
```

IF<= %value1 %value2

```
        ...
        ...         (blocco istruzioni THEN)
        ...
[ELSE]
        [...]
        [...]       (blocco opzionale istruzioni ELSE)
        [...]
ENDIF
```

%value1.....primo valore numerico
%value2.....secondo valore numerico

Il comando permette di confrontare due valori numerici o due stringhe. Nel caso il primo numero sia minore o uguale al secondo o la prima stringa venga alfabeticamente prima della seconda o sia uguale verranno eseguite le istruzioni del blocco THEN altrimenti quelle del blocco ELSE.
Il blocco di istruzioni ELSE è opzionale.
Ogni IF deve essere chiuso da un ENDIF.
Gli IF possono essere nidificati senza apparente limite.
Le stringhe non possono essere testate con questo tipo di IF.

Vedere anche: IF>=, IF<, IF>, IF=, IF<>, ELSE, ENDIF

Esempio:

```
IF<= %n 0
     MSG "Richiesto valore positivo!" "" "" ""
ENDIF
```

IF<> variable value

```
        ...
        ...         (blocco istruzioni THEN)
        ...
[ELSE]
        [...]
        [...]       (blocco opzionale istruzioni ELSE)
        [...]
ENDIF
```

variable .....variabile numerica (%) o stringa ($)
value .....costante o variabile; numerica (%) o stringa ($)

Il comando permette di confrontare due valori numerici i due stringhe. Nel caso di non uguaglianza verranno eseguite le istruzioni del blocco THEN altrimenti quelle del blocco ELSE.
Il blocco di istruzioni ELSE è opzionale.
Ogni IF deve essere chiuso da un ENDIF.
Gli IF possono essere nidificati senza apparente limite.
Con questo tipo di IF possono essere testate anche le stringhe.

Vedere anche: IF=, IF<, IF<=, IF>, IF>=, ELSE, ENDIF

Esempio:

```
IF<> $sesso "M"
     VPRINT "Femmina"
ENDIF
```

IF= variable value

```
        ...
```

```
        ...         (blocco istruzioni THEN)
        ...
[ELSE]
        [...]
        [...]       (blocco opzionale istruzioni ELSE)
        [...]
ENDIF
```

variable .....variabile numerica (%) o stringa ($)
value .....costante o variabile; numerica (%) o stringa ($)

Il comando permette di confrontare due valori numerici i due stringhe. Nel caso di uguaglianza verranno eseguite le istruzioni del blocco THEN altrimenti quelle del blocco ELSE.
Il blocco di istruzioni ELSE è opzionale.
Ogni IF deve essere chiuso da un ENDIF.
Gli IF possono essere nidificati senza apparente limite.
Con questo tipo di IF possono essere testate anche le stringhe.

Vedere anche: IF<>, IF<, IF<=, IF>, IF>=, ELSE, ENDIF

Esempio:

```
IF= %exist 0
     VPRINT "inesistente..."
ENDIF
```

IF> %value1 %value2

```
...
...         (blocco istruzioni THEN)
...
[ELSE]
        [...]
        [...]       (blocco opzionale istruzioni ELSE)
        [...]
ENDIF
```

%value1.....primo valore numerico
%value2.....secondo valore numerico

Il comando permette di confrontare due valori numerici o due stringhe. Nel caso il primo numero sia maggiore del secondo o la prima stringa venga alfabeticamente dopo la seconda verranno eseguite le istruzioni del blocco THEN altrimenti quelle del blocco ELSE.
Il blocco di istruzioni ELSE è opzionale.
Ogni IF deve essere chiuso da un ENDIF.
Gli IF possono essere nidificati senza apparente limite.
Le stringhe non possono essere testate con questo tipo di IF.

Vedere anche: IF<, IF>=, IF<=, IF<>, IF=, ELSE, ENDIF

Esempio:

```
IF> %donne 2
     VPRINT "Posso sempre provarci..."
ENDIF
```

IF>= %value1 %value2

```
...
...        (blocco istruzioni THEN)
...
[ELSE]
[...]
[...]      (blocco opzionale istruzioni ELSE)
[...]
ENDIF
```

%value1.....primo valore numerico
%value2.....secondo valore numerico

Il comando permette di confrontare due valori numerici o due stringhe. Nel caso il primo numero sia maggiore o uguale al secondo o la prima stringa venga alfabeticamente dopo la seconda o sia uguale verranno eseguite le istruzioni del blocco THEN altrimenti quelle del blocco ELSE.
Il blocco di istruzioni ELSE è opzionale.
Ogni IF deve essere chiuso da un ENDIF.
Gli IF possono essere nidificati senza apparente limite.
Le stringhe non possono essere testate con questo tipo di IF.

Vedere anche: IF<=, IF>, IF<, IF=, IF<>, ELSE, ENDIF

Esempio:

```
IF>= %temp 100
     MSG "L'acqua bolle:" "butta la pasta!" "" ""
ENDIF
```

IN %byte %port

%byte =.....byte letto (0÷255)
%port .....porta di input (0÷65535)

Il comando legge un byte dalla porta di input.

Vedere anche: OUT, PEEK, POKE, DEFSEG, START

Esempio:

```
IN %porta 1552 && 610h
```

INGLOGO %lowestx %lowesty %highestx %highesty %development $logo %pen

%lowestx = ...ascissa minima [mm]
%lowesty =.....ordinata minima [mm]
%highestx =.....ascissa massima [mm]
%highesty = ...ordinata massima [mm]
%development =.....sviluppo totale [mm]
$logo .....logo da esaminare (.log)
%pen.....penna da esaminare (1÷8)

Il comando restituisce gli ingombri del logo specificato solo per la penna selezionata e lo sviluppo totale.

Vedere anche: EXTLOGO, LDIMENSION, LROTATION

Esempio:

```
INGLOGO %x0 %y0 %x1 %y1 %sv "EVLASERR" 5
```

INKEY $key

$key =.....tasto premuto ([ ]\001÷\255; v. tabella)

Il comando legge dalla tastiera l'eventuale tasto premuto senza attendere. Se non è stato premuto alcun tasto la variabile è nulla.
Di seguito i codici possibili:

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
Normale	*59	*60	*61	*62	*63	*64	*65	*66	*67	*68	*133	*134
Shift	*84	*85	*86	*87	*88	*89	*90	*91	*92	*93	*135	*136
Control	*94	*95	*96	*97	*98	*99	*100	*101	*102	*103	*137	*138
Alt	*104	*105	*106	*107	*108	*109	*110	*111	*112	*113	*139	*140

	<-	->			Home	End	PgUp	PgDn	Ins	Del	Tab	BkSp
Normale	*75	*77	*72	*80	*71	*79	*73	*81	*82	*83	9	8
Shift	*75	*77	*72	*80	*71	*79	*73	*81	*82	*83	*15	8
Control	*115	*116	*141	*145	*119	*117	*132	*118	*146	*147	*148	127
Alt	*155	*157	*152	*160	*151	*159	*153	*161	*162	*163	*165	*14

	aA	bB	cC	dD	eE	fF	gG	hH	iI	jJ	kK	lL
Normale	97	98	99	100	101	102	103	104	105	106	107	108
Shift	65	66	67	68	69	70	71	72	73	74	75	76
Control	1	2	3	4	5	6	7	8	9	10	11	12
Alt	*30	*48	*46	*32	*18	*33	*34	*35	*23	*36	*37	*38

	mM	nN	oO	pP	qQ	rR	sS	tT	uU	vV	wW	xX
Normale	109	110	111	112	113	114	115	116	117	118	119	120
Shift	77	78	79	80	81	82	83	84	85	86	87	88
Control	13	14	15	16	17	18	19	20	21	22	23	24
Alt	*50	*49	*24	*25	*16	*19	*31	*20	*22	*47	*17	*45

	yY	zZ	0)	1!	2@	3#	4$	5%	6^	7&	8*	9(
Normale	121	122	48	49	50	51	52	53	54	55	56	57
Shift	89	90	41	33	64	35	36	37	94	38	42	40
Control	25	26			*3				30			
Alt	*21	*44	*129	*120	*121	*122	*123	*124	*125	*126	*127	*128

	`~	-_	=+	|	[{	]}	;:	'"	,<	.>	/?	
Normale	96	45	61	92	91	93	59	39	44	46	47	
Shift	126	95	43	124	123	125	58	34	60	62	63	
Control		31		28	27	29						
Alt	*41	*130	*131	*43	*26	*27	*39	*40	*51	*52	*53	

	Paus	Esc	Spaz	Ent	PrSc	g/	g*	g-	g+	gEnt		
Normale	paus	27	32	13		47	42	45	43	13		
Shift	paus	27	32	13		47	42	45	43	13		
Control	*32	27	32	10	*114	*149	*150	*142	*144	10		
Alt	paus	*1	32	*28		*164	*55	*74	*78	*166		

	g0In	g.Dl	g1En	g2	g3PD	g4<-	g5	g6->	g7Hm	g8	g9PU	
Normale	48	46	49	50	51	52	53	54	55	56	57	
Shift	*82	*83	*79	*80	*81	*75	*76	*77	*71	*72	*73	
Control	*146	*147	*117	*145	*118	*115	*143	*116	*119	*141	*132	
Alt												

L'asterisco significa che il codice è a due caratteri invece che uno ed il primo è uno spazio. La 'g' indica tasti 'grey' sul tastierino numerico.

La tabella si riferisce ad una tastiera americana estesa con Caps Lock spento e Num Lock acceso. Accendendo il Caps Lock si invertono i codici delle lettere alfabetiche tra Normali e Shift, spegnendo Il Num Lock si invertono i codici gr0÷gr9 e gr. tra Normali e Shift.
Attenzione: ^d e ^Brk non sono momentaneamente distinguibili.

Vedere anche: START

Esempio:

```
INKEY $k
```

INSTR %position $text $find %start

%position = ... posizione trovata (0=non trovata, 1÷31)
$text ..... testo in cui cercare
$find ..... testo da cercare
%start ..... partenza della ricerca (1÷31; 1=primo carattere)

Il comando permette di eseguire la ricerca di una testo all'interno di un altro ritornandone la posizione.

Vedere anche: LEFT, LEN, MID, RIGHT, FFIND

Esempio:

```
INSTR %p "386 or 486?" "86" 3
```

INT %integer %real

%integer = ... numero intero
%real ..... numero reale

Il comando trova la parte intera di un numero specificato senza arrotondamenti (troncamento).

Vedere anche: ABS

Esempio:

```
INT %i 3.141593
```

ITALIC %angle

%angle.....inclinazione o italicità dei caratteri [°] (-80÷80)

Il comando imposta l'inclinazione dei caratteri dello stile di testo.

Vedere anche: FNAME, HEIGHT, SDEFINITION, SNAME, SNUMBER, SROTATION, WIDTH, PRINT

Esempio:

```
ITALIC -10
```

KILL $file

$file .....nome del file

Il comando permette la cancellazione del file specificato.

Vedere anche: COPY, NAME, RMDIR, MKDIR, EXIST, FFIRST, FNEXT, FLIST

Esempio:

```
KILL "*.bak"
```

LAMP %lamp

%lamp.....stato della lampada? (0=off, 1=on)

Il comando permette l'accensione o lo spegnimento della lampada.
Questo segnale solitamente segnala la marcatura in corso accendendo una lampada ma può essere anche semplicemente un uscita per un interfacciamento personalizzabile.

Vedere anche: SETALIM, SETCO2, SETPLC, SETRELE, START, POKE, OUT

Esempio:

```
LET %n 10
:loop
WAIT 500
LAMP 1
WAIT 500
LAMP 0
IF> %n 0
      - %n %n 1
      GOTO :loop
ENDIF
```

Il programma di esempio fa lampeggiare la lampada 10 volte per un qualche avvertimento.

LCASE $lowercase $text

$lowercase =.....testo minuscolo
$text.....testo

Il comando trasforma tutti i caratteri alfabetici in minuscoli e lascia invariati gli altri.

Vedere anche: UCASE, STRING, ADDSTR, LEFT, LTRIM, MID, RIGHT, RTRIM, TRIM

Esempio:

```
LCASE $lc "Acusto-Ottico"
```

LDIMENSION %dimx %dimy

%dimx.....dimensione X percentuale [%]
%dimy.....dimensione Y percentuale [%]

Il comando imposta la dimensione del logo.

Vedere anche: EXTLOGO, INGLOGO, LROTATION

Esempio:

```
LDIMENSION 50 50
```

LEFT $left $text %number

$left = ...caratteri a sinistra
$text.....testo
%number.....numero di caratteri (0÷31)

Il comando preleva i caratteri specificati alla sinistra del testo.

Vedere anche: RIGHT, MID, LTRIM, TRIM, RTRIM, ADDSTR, INSTR, LCASE, UCASE, LEN, STRING

Esempio:

```
LEFT $s "1234567" 4
```

LEN %length $text

%length =.....lunghezza del testo (0÷31)
$text .....testo

Il comando restituisce la lunghezza del testo.

Vedere anche: ADDSTR, STRING, INSTR, LEFT, MID, RIGHT, LTRIM, RTRIM, TRIM

Esempio:

```
LEN %l "Rossi Mario"
```

LET variable value

variable =.....nome variabile numerica (%) o stringa ($)
value .....valore numerico o stringa (costanti o variabili)

Il comando assegna alla variabile il valore specificato.
Le variabili numeriche sono sempre in singola precisione.
Le variabili stringa possono contenere al massimo 31 caratteri.
E` possibile inserire nelle costanti stringa caratteri non presenti sulla tastiera utilizzando il backslash seguito da tre cifre relative al codice ASCII in decimale (p.es.: \014).
Il codice ASCII 0 non è valido.
Il numero massimo di variabili numeriche è 256.
Il numero massimo di variabili stringa è 100.

edere anche: IF, STR, VAL

Esempio:

```
LET $data " Giovedì 6 Maggio 1996"
LET %pigreco 3.141593
LET $nome "PAOLO"
LET %eta 33
LET %peso 71.5
LET $sesso "M"

LOCATE -5 10
PRINT "Oggi: "
PRINT $data
LOCATE -5 7
PRINT "Valore di π: "
```

```
STR $tmp %pigreco
PRINT $tmp
LOCATE -5 4
PRINT "Nome: "
PRINT $nome
LOCATE -5 1
PRINT "Età [anni]: "
STR $tmp %eta
PRINT $tmp
LOCATE -5 -2
PRINT "Peso [Kg]: "
STR $tmp %peso
PRINT $tmp
LOCATE -5 -5
PRINT "Sesso: "
PRINT $sesso
```

Il programma di esempio assegna alcune variabili e le marca.

LINE %x %y

%x.....ascissa di arrivo [mm]
%y.....ordinata di arrivo [mm]

Il comando traccia una linea dalla coordinata corrente a quella specificata.
Il punto di arrivo diventa quello corrente.

Vedere anche: LOCATE, BOX, PRINT, EXTLOGO, ABSOLUTE, RELATIVE

Esempio:

```
LOCATE -20 20
GOSUB :greca
LOCATE -20 0
GOSUB :greca
LOCATE -20 -20
GOSUB :greca
END

:greca
     RELATIVE
     LET %n 10
     :loop
     IF> %n 0
          LINE 3 3
          LINE 3 -3
          - %n %n 1
          GOTO :loop
     ENDIF
     ABSOLUTE
RETURN
```

L'esempio mostra come eseguire delle greche eseguite in RELATIVE rispetto alla posizione ABSOLUTE iniziale utilizzando LOCATE e LINE.
E` possibile marcare o simulare.

LOCATE %x %y

%x.....ascissa [mm]
%y.....ordinata [mm]

Il comando imposta il nuovo punto corrente.

Vedere anche: ABSOLUTE, RELATIVE, LINE, BOX, EXTLOGO, PRINT

Esempio:

```
LOCATE -20 20
GOSUB :greca
LOCATE -20 0
GOSUB :greca
LOCATE -20 -20
GOSUB :greca
END

:greca
     RELATIVE
     LET %n 10
     :loop
     IF> %n 0
          LINE 3 3
          LINE 3 -3
          - %n %n 1
          GOTO :loop
     ENDIF
     ABSOLUTE
RETURN
```

L'esempio mostra come eseguire delle greche eseguite in RELATIVE rispetto alla posizione ABSOLUTE iniziale utilizzando LOCATE e LINE.
E` possibile marcare o simulare.

LOF %lengthoffile %handle

%lengthoffile =.....dimensione del file (0÷65535²-1)
%handle.....gestore del file (1÷65535)

Il comando restituisce la dimensione del file aperto senza spostare il puntatore.

Vedere anche: FPOS, FDATE, FLOCATE, FFIND, FOPEN

Esempio:

```
LOF %l %h
```

LPRINT $text

$text .....testo da mandare alla stampante

Il comando manda alla stampante il testo specificato.
Per i caratteri di controllo della specifica stampante consultare il relativo manuale e ricordare che tutti i codici ASCII (a parte lo 0) possono essere inviati utilizzando lo slash e le tre cifre del codice richiesto all'interno del testo (p.es.: \014).
Di seguito una tabella dei codici standard Epson più usati:

Decimale	ASCII	Descrizione
27 64	ESC @	inizializzazione spampante
10	LF	salto linea (line feed)
12	FF	salto pagina (form feed)
27 50	ESC 2	6 linee per inch
27 48	ESC 0	8 linee per inch
27 51 n	ESC 3 n	altezza linea in 1/216 di inch (n=0÷255)
27 67 48 n	ESC C 0 n	lunghezza pagina in inches (n=1÷22)
27 67 n	ESC C n	lunghezza pagina in linee (n=1÷127)
27 108 n	ESC l n	margine sinistro (n=colonna)
27 81 n	ESC Q n	margine destro (n=colonna)
15	SI	modo condensato
18	DC2	modo non condensato
27 80	ESC P	larghezza pica (10 cpi)
27 77	ESC M	larghezza elite (12 cpi)
27 87 49	ESC W 1	modo doppia larghezza
27 87 48	ESC W 0	modo non doppia larghezza
27 45 49	ESC - 1	modo sottolineato
27 45 48	ESC - 0	modo non sottolineato
27 69	ESC E	modo evidenziato
27 70	ESC F	modo non evidenziato

27 71	ESC G	modo doppia passata
27 72	ESC H	modo non doppia passata
27 83 48	ESC S 0	modo apice
27 83 49	ESC S 1	modo pendice
27 84	ESC T	modo nè apice nè pendice
27 52	ESC 4	modo italico
27 53	ESC 5	modo non italico

Vedere anche: PRINT, VPRINT

Esempio:

```
LPRINT "Normale\015Condensato\018Normale"
```

LROTATION %angle

%angle.....angolo [°]

Il comando imposta la rotazione dei logo.

Vedere anche: LDIMENSION, EXTLOGO, INGLOGO

Esempio:

```
LROTATION 90
```

LTRIM $lefttrim $text

$lefttrim =.....testo senza spazi a sinistra
$text.....testo iniziale

Il comando elimina tutti gli spazi consecutivi iniziali (a sinistra).

Vedere anche: RTRIM, TRIM, LEFT, RIGHT, MID

Esempio:

```
LTRIM $lt "   Jim Morrison & the Doors"
```

MENU %item $item $file $title

%item =.....numero della voce selezionata (³0; 0=nessuna)
$item = ...testo della voce selezionata
$file .....file dati contenente le voci del menu
$title .....titolo del menu

Il comando permette la visualizzazione di un menu e la selezione di una voce. La definizione del menu dev'essere contenuta nel file di testo specificato, una voce per linea. Selezionando una voce essa verrà riportata sia come numero della voce che essa stessa. Uscendo dal menu senza selezionare nulla il numero della voce risulterà 0 e la voce nulla.

Vedere anche: FLIST, HPGLCONV, YESNO

Esempio:

```
MENU %i $i "menu.dat" "SELEZIONE OPERAZIONE"
```

MID $middle $text %position %number

$middle = ...Testo estratto
$text.....testo iniziale
%position.....posizione iniziale (1÷31)
%number.....numero di caratteri (0÷31)

Il comando permette l'estrazione di una parte di un testo specificando posizione iniziale e numero di caratteri.

Vedere anche: LEFT, RIGHT, LTRIM, TRIM, RTRIM

Esempio:

```
MID $m "The Dark Side if the Moon" 5 9
```

MIRROR %x %y

%x.....mirror asse X? (0=no, 1=si)
%y.....mirror asse Y? (0=no, 1=si)

Il comando attiva/disattiva la specularità orizzontale e/o verticale.
All'inizio lo script ha sempre a zero sia la specularità orizzontale sia la verticale, ciò non avviene con lo scambio canali CHGXY.
Le tabelle seguenti indicano la rotazione (e l'eventuale specularità indicata con 's') che si danno all'immagine come la si trova all'inizio.

```
Ammettendo che CHGXY all'inizio abbia valore 0:
+-------------------------------------------------------------+
| MIRROR X  |  0  |  0  |  0  |  0  |  1  |  1  |  1  |  1  |
| MIRROR Y  |  0  |  0  |  1  |  1  |  0  |  0  |  1  |  1  |
| CHGXY     |  0  |  1  |  0  |  1  |  0  |  1  |  0  |  1  |
+-----------+-----+-----+-----+-----+-----+-----+-----+-----|
| ROTAZIONE |  0  | 270s| 180s|  90 |  0s | 270 | 180 | 90s |
+-------------------------------------------------------------+

Ammettendo che CHGXY all'inizio abbia valore 1:
+-------------------------------------------------------------+
| MIRROR X  |  0  |  0  |  0  |  0  |  1  |  1  |  1  |  1  |
| MIRROR Y  |  0  |  0  |  1  |  1  |  0  |  0  |  1  |  1  |
| CHGXY     |  0  |  1  |  0  |  1  |  0  |  1  |  0  |  1  |
+-----------+-----+-----+-----+-----+-----+-----+-----+-----|
| ROTAZIONE | 270s|  0  |  90 | 180s| 270 |  0s | 90s | 180 |
+-------------------------------------------------------------+
```

Vedere anche: CHGXY, ROTATE, SCALE

Esempio:

```
MIRROR 1 0
```

MKDIR $dir

$dir .....nome directory

Il comando permette la creazione della directory specificata.

Vedere anche: RMDIR, KILL, NAME, COPY, FFIRST, FNEXT

Esempio:

```
MKDIR "c:\092l2temp"
```

MSG $line1 $line2 $line3 $line4

$line1.....prima linea del messaggio
$line2.....seconda linea del messaggio
$line3.....terza linea del messaggio
$line4.....quarta linea del messaggio

Il comando mostra una finestra contenente il messaggio (da una a quattro linee) ed attende conferma.
In caso i parametri siano vuoti viene visualizzato il messaggio:

```
Premere ENTER per continuare
```

Vedere anche: ERRMSG, YESNO

Esempio:

```
MSG "Lo sviluppo del logo" $logo "è pari a mm" $sv
```

NAME $old $new

$old .....vecchio nome del file (<>"")
$new - nuovo nome del file (<>"")

Il comando permette di rinominare un file.

Vedere anche: COPY, KILL, EXIST, FFIRST, FNEXT, MKDIR, RMDIR, FLIST

Esempio:

```
NAME "acme.cfg" "acme.old"
```

NOT %not %byte

%not =.....byte invertito (0÷255)
%byte.....byte iniziale (0÷255)

Il comando inverte tutti gli 8 bit del byte.

```
TABELLA DELLA VERITA`
   +-----------+
   ¦ bit ¦ NOT ¦
   ¦-----+-----¦
   ¦  0  ¦  1  ¦
   ¦  1  ¦  0  ¦
   +-----------+
```

Vedere anche: AND, OR, XOR, BRESET, BSET, BTOGGLE

Esempio:

```
GETNUM %y "Anno?" 1996 4

let %m 4
GOSUB :multiplo
let %m4 %m

let %m 100
GOSUB :multiplo
let %m100 %m
```

```
let %m 400
GOSUB :multiplo
let %m400 %m

NOT %nm100 %m100 && \
OR %tmp %nm100 %m400       && +- bis = m4 AND [(NOT m100) OR m400]
AND %bis %m4 %tmp && /
STR $y %y
IF= %bis 0
     MSG "L'anno" $y "non è bisestile" ""
ELSE
     MSG "L'anno" $y "è bisestile" ""
ENDIF
END

:multiplo                  && ritorna %m = 1 se multiplo di %m altrimenti 0
     / %tmp %y %m
     INT %tmp1 %tmp
     IF= %tmp %tmp1
          LET %m 1
     ELSE
          LET %m 0
     ENDIF
RETURN
```

L'esempio mostra l'utilizzo della logica booleana trovando se l'anno fornito è bisestile o no; un anno è bisestile se è multiplo di 4; se però è multiplo anche di 100 allora non è bisestile; se, ancora, è multiplo anche di 400 allora torna ad essere bisestile.

OR %or %byte1 %byte2

%or = ...OR logico (0÷255)
%byte1.....primo byte (0÷255)
%byte2.....secondo byte (0÷255)

Il comando restituisce l'OR logico (detto anche 'somma logica') tra i due byte, bit per bit.

TABELLA DELLA VERITA`

bit 1	bit 2	OR
0	0	0
0	1	1
1	0	1
1	1	1

Vedere anche: AND, NOT, XOR, BRESET, BSET, BTOGGLE

Esempio:

```
GETNUM %y "Anno?" 1996 4

let %m 4
GOSUB :multiplo
let %m4 %m

let %m 100
GOSUB :multiplo
let %m100 %m

let %m 400
GOSUB :multiplo
let %m400 %m

NOT %nm100 %m100 && \
OR %tmp %nm100 %m400       && +- bis = m4 AND [(NOT m100) OR m400]
AND %bis %m4 %tmp && /
STR $y %y
IF= %bis 0
     MSG "L'anno" $y "non è bisestile" ""
ELSE
     MSG "L'anno" $y "è bisestile" ""
ENDIF
END

:multiplo                  && ritorna %m = 1 se multiplo di %m altrimenti 0
     / %tmp %y %m
     INT %tmp1 %tmp
     IF= %tmp %tmp1
          LET %m 1
     ELSE
          LET %m 0
     ENDIF
RETURN
```

L'esempio mostra l'utilizzo della logica booleana trovando se l'anno fornito è bisestile o no; un anno è bisestile se è multiplo di 4; se però è multiplo anche di 100 allora non è bisestile; se, ancora, è multiplo anche di 400 allora torna ad essere bisestile.

OUT %port %byte

%port .....porta di output (0÷65535)
%byte .....byte da presentare (0÷255)

Il comando manda sulla porta di output il byte indicato.

Vedere anche: IN, POKE, PEEK, DEFSEG, SETPLC, SETRELE, SETCO2, LAMP, SETALIM

Esempio:

```
OUT 800 127 && 320h
```

PEEK %byte %address

%byte =.....byte letto (0÷255)
%address .....indirizzo (0÷65535)

Il comando restituisce il byte letto dalla memoria all'indirizzo specificato del segmento corrente.

Vedere anche: DEFSEG, IN, OUT, POKE, START

Esempio:

```
PEEK %b 513 && 201h
```

POKE %address %byte

%address .....indirizzo (0÷65535)
%byte.....byte da scrivere (0÷255)

Il comando permette di scrivere il byte all'indirizzo specificato del segmento corrente. Per sicurezza POKE non lavora se il segmento corrente è fuori dall'intervallo C800h÷EFC0h (51200÷61376).

Vedere anche: DEFSEG, IN, OUT, PEEK, SETPLC, SETRELE, SETALIM, LAMP, SETCO2

Esempio:

```
POKE 128 64 && 80h
```

PRINT $text

$text .....testo da stampare

Il comando permette la stampa del testo utilizzando lo stile di testo corrente e partendo dal punto corrente. Il punto corrente viene posto al termine del testo stampato.

Vedere anche: LOCATE, EXTLOGO, LINE, BOX, FNAME, HEIGHT, ITALIC, SDEFINITION, SNAME, SNUMBER, SROTATION, WIDTH, VPRINT, LPRINT

Esempio:

```
PRINT "Simbolo del grado: \248"
```

PUTCFG $item $value

$item .....voce (v. tabella)
$value.....valore da attribuire (v. tabella)

Il comando permette di attribuire alla voce del file di configurazione (LASER2.CFG) il valore specificato.
Le voci disponibili sono le seguenti:

OBJECT_EDGE	lato obiettivo (<>0) [mm]
MONOCHROMATIC	soppressione colori? (TRUE/FALSE)
QUIET_SOUND	soppressione suono? (TRUE/FALSE)
MOUSE_ENABLE	abilitazione mouse? (TRUE/FALSE)
CURRENT_WOBBLE_FILE_NAME	tabella stili di marcatura corrente (.wob)
CURRENT_STYLE_FILE_NAME	tabella stili di testo corrente (.stl)
CURRENT_LIBRARY_FILE_NAME	libreria logo corrente (.lib)
CURRENT_SCR_FILE_NAME	script corrente (.scr)
OFFSET_X	offset X [mm]
OFFSET_Y	offset Y [mm]
HEX_CARRIER_ADDRESS	indirizzo carrier [esa] (0÷FFFF)
MODULE_NUMBER	modulo Burr Brown (0÷3; 0=assente)
HEX_VOBUL_ADDRESS	ind. vobulatore elettronico [esa] (0÷FFFF)
QS_LASER_ON_DELAY	attesa dopo laser on ad impulsi [µs]
CW_LASER_ON_DELAY	attesa dopo laser on in continua [µs]
QS_LASER_OFF_DELAY	attesa prima di laser off ad impulsi [µs]
CW_LASER_OFF_DELAY	attesa prima di laser off in continua [µs]
QS_JUMP_DELAY	attesa dopo salto ad impulsi [µs]
CW_JUMP_DELAY	attesa dopo salto in continua [µs]
QS_EDGE_DELAY	attesa spigolo oltre limite ad impulsi [µs]
CW_EDGE_DELAY	attesa spigolo oltre limite in continua [µs]
CHANGE_FREQUENCY_DELAY	attesa dopo cambio frequenza [ms]
CHANGE_CURRENT_DELAY	attesa dopo cambio corrente [ms]

CHANGE_DIAFRAM_DELAY	attesa dopo cambio diaframma [ms]
CLOSE_A/O_DELAY	attesa chiusura A/O [µs]
OPEN_A/O_DELAY	attesa apertura A/O [µs]
CLOSE_SHUTTER_DELAY	attesa chiusura shutter [µs]
OPEN_SHUTTER_DELAY	attesa apertura shutter [µs]
AXIS_POS_CHANGE_DELAY	attesa dopo movimentazione assi [ms]
X_GALVO_CALIBRATION	calibrazione galvo X [%] (0÷100)
Y_GALVO_CALIBRATION	calibrazione galvo Y [%] (0÷100)
X_CORRECTION_FLAG	correzione galvo X? (TRUE/FALSE)
Y_CORRECTION_FLAG	correzione galvo Y? (TRUE/FALSE)
X_CORRECTION_FACTOR	fattore di correzione galvo X [~mm]
Y_CORRECTION_FACTOR	fattore di correzione galvo Y [~mm]
XY_EXCHANGE	scambio canali X/Y? (TRUE/FALSE)
5µs_LOOPS	loop processore per 5µs
ms_LOOPS	loop processore per 1ms
X_HPGL_CALIBRATION	calibrazione X HPGL [u/mm]
Y_HPGL_CALIBRATION	calibrazione Y HPGL [u/mm]
HPGL_SCALE_FACTOR	scala di conversione HPGL [%]
X_HPGL_OFFSET	offset X di conversione HPGL [u] (±65536)
Y_HPGL_OFFSET	offset Y di conversione HPGL [u] (±65536)
HPGL_SWAP	scambio X/Y in conv. HPGL? (TRUE/FALSE)
LIMIT_ANGLE	angolo limite in conv. HPGL [°] (0÷180)
SIMULATION_DELAY	ritardo di simulazione [~µs/pt]
ELECTR_VOL_DELAY	ritardo scrittura su volantino elettr. [µs]
SIMULATION_VIDEO_MODE	modo video simulazione (CGAL/CGAH/EGA/VGA)
SIMULATION_GRID	griglia in simulazione? (TRUE/FALSE)
SIMULATION_GRID_DIM	dim. griglia di simulazione [mm] (<>0)
SIMULATION_CALIBRATION	calibrazione in simulazione? (TRUE/FALSE)
SIMULATION_CORRECTION	correzione in simulazione? (TRUE/FALSE)
INTERLOCK_ENABLE	abilitazione interlock? (TRUE/FALSE)
MIN_INTERLOCK_LOOPS	numero minimo di interlock consecutivi [#]
SPEED_FACTOR_A	fattore di velocità A
SPEED_FACTOR_B	fattore di velocità B
EV_TYPE	tipo marcatrice (EV50/EV80/EV120/EVCO2)
A/O_INVERTED	segnale A/O invertito? (TRUE/FALSE)
RF_INVERTED	segnale RF invertito? (TRUE/FALSE)
LANGUAGE	linguaggio (Italiano/English/Deutsch/...)
MAX_JUMP	salto dinamico massimo percentuale [%]
JUMP_SPEED	velocità salto dinamico [mm/s]
ELECT_CURR_ADJ_DELAY	ritardo scrittura potenziometro elettr. [µs]
ELECT_CURR_ENABLE	abil. potenziometro elettr.? (TRUE/FALSE)
AXIS_COM	porta seriale mov. assi (0/1/2; 0=disab.)
X_AXIS_ENABLE	abilitazione asse lineare X? (TRUE/FALSE)
Y_AXIS_ENABLE	abilitazione asse lineare Y? (TRUE/FALSE)
Z_AXIS_ENABLE	abilitazione asse lineare Z? (TRUE/FALSE)
R_AXIS_ENABLE	abilitazione asse rotante R? (TRUE/FALSE)
X_AXIS_STEP_MM	calibrazione asse lineare X [pp/mm]
Y_AXIS_STEP_MM	calibrazione asse lineare Y [pp/mm]
Z_AXIS_STEP_MM	calibrazione asse lineare Z [pp/mm]
R_AXIS_STEP_MM	calibrazione asse rotante R [pp/°]
X_AXIS_POSITION	posizione asse lineare X [mm]
Y_AXIS_POSITION	posizione asse lineare Y [mm]
Z_AXIS_POSITION	posizione asse lineare Z [mm]
R_AXIS_POSITION	posizione asse rotante R [°]
X_AXIS_MAX	finecorsa sw asse lineare X [mm] (0=disab.)
Y_AXIS_MAX	finecorsa sw asse lineare Y [mm]
Z_AXIS_MAX	finecorsa sw asse lineare Z [mm]
R_AXIS_MAX	finecorsa sw asse rotante R [°]
X_AXIS_ZERO	zero software asse lineare X [mm]
Y_AXIS_ZERO	zero software asse lineare Y [mm]
Z_AXIS_ZERO	zero software asse lineare Z [mm]
R_AXIS_ZERO	zero software asse rotante R [°]
AXIS_SPEED	ritardo mov. assi [100/(m/min)-28]
AXIS_NUMBER	tipo EPROM controllore assi (3/4=num. assi)
FREQ_MIN	frequenza corrispondente a 0% [KHz]
FREQ_MAX	frequenza corrispondente a 100% [KHz]

CURR_MIN	corrente corrispondente a 0% [A/%/W/...]
CURR_MAX	corrente corrispondente a 100% [A/%/W/...]
CURR_UNIT	unità di misura della corrente [A/%/W/...]
TABVOB_FIELDS_ENABLE	abil. PRFZ tab. stili marc.? (TRUE/FALSE)
DEBUG_LEVEL	livello di debugging [#] (0/...)
JUMP_COLOR	colore salti in simulazione (0÷15;0=disab.)
CO2_FREQ	laser CO^2 in frequenza? (TRUE/FALSE)

Se la descrizione è interrogativa la voce accetta solo TRUE o FALSE (vero o falso).
Modificare il file di configurazione significa rendere attiva la modifica solo alla PROSSIMA esecuzione del programma di marcatura.
Utilizzare con estrema attenzione questo comando che può stravolgere anche la marcatura e le calibrazioni di fabbrica.
Si consiglia di tenere una copia di sicurezza di LASER2.CFG.

Vedere anche: GETCFG, GETSYSN, PUTSYSN

Esempio:

```
PUTCFG "OBJECT_EDGE" "180"
```

PUTSYSN %code %value

%code.....codice variabile di sistema (0÷255; v. tabella)
%value.....valore per la variabile di sistema

Il comando permette di porre nella variabile di sistema il valore specificato.
Utilizzare con estrema attenzione questo comando che può stravolgere o addirittura bloccare il sistema.
Di seguito i codici relativi alle varie variabili di sistema:

0 OFFSETX	64 D	128 BPP	192 NILOCKS
1 OFFSETY	65 SCALEX	129 RT	193 LOCX
2 SROTATION	66 SCALEY	130 RATIO	194 LOCY
3 DEFSEG	67 BASESX	131 RATIOP	195 ATTRIB
4 CARRIER	68 BASESY	132 ANG	196 CFACT
5 MODULE	69 ROTAZ	133 AI	197 RITEDGEC
6 VOBULADD	70 BASERX	134 AF	198 OBJ
7 ---	71 BASERY	135 PER	199 MONO
8 KSPEEDB	72 N	136 LARCO	200 RITEDGEQ
9 RITLONQ	73 NP	137 DELTAANG	201 O1
10 RITLONC	74 DELTAX	138 AX8	202 O2
11 RITLOFFQ	75 DELTAY	139 BX8	203 O3
12 RITLOFFC	76 PASSOX	140 CX8	204 O4
13 RITJUMPQ	77 PASSOXP	141 DX8	205 ---
14 RITJUMPC	78 PASSOY	142 EX8	206 ---
15 RITFREQ	79 PASSOYP	143 ---	207 ---
16 RITCURR	80 DIST	144 ---	208 ---
17 RITDIAF	81 E	145 ---	209 ---
18 RITCAO	82 EP	146 ---	210 ---
19 RITCSH	83 ---	147 P0	211 ---
20 RITAAO	84 ---	148 P1	212 ---

21 RITASH	85 FUORIX	149 P2	213 ---
22 RITAXIS	86 FUORIY	150 P3	214 ---
23 ---	87 LAMP	151 P4	215 ---
24 ---	88 ILOCKS	152 P5	216 ---
25 ---	89 LMERR1	153 ---	217 ---
26 CALX	90 TIPO	154 ---	218 ---
27 CALY	91 ---	155 ALLIN	219 ---
28 CXFL	92 EXITFLAG	156 ALLINFIT	220 ---
29 CYFL	93 LASER	157 ORFLAG	221 ---
30 CORRX	94 ---	158 ---	222 ---
31 CORRY	95 AO	159 LASTX	223 ---
32 MICRO	96 SH	160 LASTY	224 ---
33 MILLI	97 XV	161 XSIM	225 ---
34 ---	98 YV	162 YSIM	226 ---
35 RITSIM	99 ---	163 SIMXS	227 ---
36 VMODE	100 ---	164 SIMYS	228 ---
37 MINILOCKS	101 ---	165 SIMXO	229 ---
38 KSPEEDA	102 ---	166 SIMYO	230 ---
39 PDIM	103 ---	167 SIMRT	231 ---
40 PATNUM	104 ---	168 ---	232 ---
41 VEL	105 ---	169 ---	233 ---
42 RISOL	106 ---	170 MARCARE	234 ---
43 PRISOL	107 X	171 ---	235 ---
44 FREQ	108 Y	172 ---	236 ---
45 CURR	109 ---	173 ---	237 ---
46 DIAF	110 ---	174 ---	238 ---
47 FROTAZ	111 ---	175 ---	239 ---
48 ALT	112 ---	176 SVX	240 ---
49 LARG	113 XCEN	177 SVY	241 ---
50 CSPACE	114 XCENP	178 SVPASSOX	242 ---
51 LSPACE	115 YCEN	179 SVPASSOY	243 ---
52 ITALIC	116 YCENP	180 SVN	244 ---
53 CHGXY	117 RELS	181 SVDELTAANG	245 ---
54 VEL1	118 ---	182 SVAI	246 ---
55 FREQ1	119 ---	183 SVRATIO	247 ---
56 ---	120 ---	184 SVE	248 ---
57 FSTATE	121 TMP	185 SVA	249 ---
58 ERAON	122 TMP1	186 SVB	250 ---
59 ERAONP	123 TMPX	187 SVXCEN	251 ---
60 ORIGINX	124 TMPY	188 SVYCEN	252 ---
61 ORIGINY	125 A	189 PATFLAG	253 ---
62 MIRRX	126 AP	190 BEGFLAG	254 ---
63 MIRRY	127 B	191 NSTART	255 ---

Vedere anche: GETCFG, GETSYSN, PUTCFG

Esempio:

```
PUTSYSN 70 10 && BASERX
```

REDRAW

Il comando cancella e riscrive la griglia in fase di simulazione.

Vedere anche: FIELD, SCREEN, VCLS, VCOLOR, VLOCATE, VPRINT

Esempio:

```
REDRAW
```

RELATIVE

Il comando fa si che tutte le coordinate seguenti vengano interpretate in modo relativo ad ogni ultimo punto.
Alla partenza lo script è sempre in assoluto.

Vedere anche: ABSOLUTE, LOCATE, LINE, BOX

Esempio:

```
LOCATE -20 20
GOSUB :greca
LOCATE -20 0
GOSUB :greca
LOCATE -20 -20
GOSUB :greca
END

:greca
     RELATIVE
     LET %n 10
     :loop
     IF> %n 0
          LINE 3 3
          LINE 3 -3
          - %n %n 1
          GOTO :loop
     ENDIF
     ABSOLUTE
RETURN
```

L'esempio mostra come eseguire delle greche eseguite in RELATIVE rispetto alla posizione ABSOLUTE iniziale utilizzando LOCATE e LINE.
E` possibile marcare o simulare.

RESOLUTION %resolution

%resolution.....risoluzione [pt/mm] (>0)

Il comando imposta la risoluzione dello stile di marcatura.

Vedere anche: CURRENT, FREQUENCY, FROTATION, SPEED, WDEFINITION, WDIMENSION, WNAME, WNUMBER

Esempio:

```
RESOLUTION 40
```

RETURN

Il comando termina la routine invocata riportando il flusso del programma all'istruzione immediatamente successiva a quella di chiamata.

Vedere anche: GOSUB, GOTO, $INCLUDE, RUN

Esempio:

```
ABSOLUTE             && 1 = prima chiamata
        LOCATE 10 0        && 2 = primo ritorno
  +---- GOSUB :3ang        && 3 = seconda chiamata
  | +-> LOCATE 20 0        && 4 = secondo ritorno
1 | |   GOSUB :3ang ---+
  | |   END <-----+    | 3
  | |             |    |
  +-|-> :3ang <---|----+
    | RELATIVE    |
  2 | LINE 10 10  | 4
    | LINE 0 -20  |
    | LINE -10 10 |
    | ABSOLUTE    |
    +-- RETURN ---+
```

Il programma d'esempio marca due triangoli allineati verticalmente; le frecce indicano il flusso del programma.

RIGHT $right $text %number

$right = ... caratteri a destra
$text ..... testo
%number ..... numero di caratteri (0÷31)

Il comando preleva i caratteri specificati alla destra del testo.

Vedere anche: LEFT, MID, LTRIM, TRIM, RTRIM, LEN

Esempio:

```
GETNUM %n "Numero iniziale progressivo?" 0 5
+ %l %n 10
:loop
IF< %n %l
      STR $n %n
      STRING $z 6 "0"
      ADDSTR $n $z $n
      RIGHT $n $n 6
      MSG "Numero seriale:" $n "" ""
      GOTO :loop
ENDIF
```

L'esempio mostra la creazione di dieci numeri seriale progressivi a sei cifre sempre riempiti di zeri a partire da quello specificato.

RITAXIS %delay

%delay ..... ritardo assestamento [ms]

Il comando modifica il tempo di attesa automatico dopo ogni movimentazione degli assi affinchè il sistema si assesti.

Vedere anche: RITLONC, RITLONQ, RITSIM, WAIT, AXRESET, AXSET, GETAXIS, RITAXIS, AXIS

Esempio:

```
RITAXIS 1000 && 1s
```

RITLONC %delay

%delay.....ritardo di apertura [µs]

Il comando imposta l'attesa DOPO l'apertura del laser in continua.

Vedere anche: RITAXIS, RITLONQ, RITSIM, WAIT

Esempio:

```
RITLONC 2000 && 2ms
```

RITLONQ %delay

%delay.....ritardo di apertura [µs]

Il comando imposta l'attesa DOPO l'apertura del laser ad impulsi.

Vedere anche: RITAXIS, RITLONC, RITSIM, WAIT

Esempio:

```
RITLONQ 1500 && 1.5ms
```

RITSIM %delay

%delay.....ritardo di simulazione [»µs/pt]

Il comando imposta la velocità di simulazione a video.

Vedere anche: RITAXIS, RITLONC, RITLONQ, WAIT

Esempio:

```
RITSIM 1000 && ≈1ms/pt
```

RMDIR $dir

$dir .....nome directory

Il comando permette la rimozione della directory specificata.
Assicurarsi però prima che la directory non contenga file o il comando non avrà effetto.

Vedere anche: MKDIR, KILL, COPY, NAME, FFIRST, FNEXT

Esempio:

```
RMDIR "c:\092l2temp"
```

ROTATE %centerx %centery %angle

%centerx.....ascissa centro di rotazione [mm]
%centery.....ordinata centro di rotazione [mm]
%angle.....angolo di rotazione [°]

Il comando ruota tutta l'area di marcatura attorno alla coordinata fornita dell'angolo specificato.

Vedere anche: CHGXY, MIRROR, SCALE

Esempio:

```
ROTATE 0 0 45
```

RTRIM $righttrim $text

$righttrim =.....testo senza spazi a destra
$text.....testo iniziale

Il comando elimina tutti gli spazi consecutivi finali (a destra).

Vedere anche: LTRIM, TRIM, RIGHT, LEFT, MID

Esempio:

```
RTRIM $title "Wish You Were Here    "
```

RUN $scx

$scx .....nome script compilato da eseguire (.scx)

Il comando carica ed esegue il script specificato.
Lo script dev'essere stato precedentemente compilato senza errori.

Vedere anche: $INCLUDE, GOSUB

Esempio:

```
RUN "fuoco"
```

SCALE %basex %basey %scalex %scaley

%basex.....ascissa base della scala [mm]
%basey.....ordinata base della scala [mm]
%scalex .....scala orizzontale [%]
%scaley.....scala verticale [%]

Il comando scala tutta l'area di marcatura sulla base della coordinata fornita della scala specificata.

Vedere anche: ROTATE, MIRROR, CHGXY

Esempio:

```
SCALE 0 0 99 50
```

SCREEN %mode

%mode.....modo video (v. tabella)

Il comando permette di passare alla modalità video richiesta secondo la tabella:

CODICE	MODO	COLONNExRIGHE	PIXELS	COLORI
0	TESTO	80x25	-	16
1	GRAFICA CGAL	40x25	320x200	4
2	GRAFICA CGAH	80x25	640x200	2
9	GRAFICA EGA	80x25	640x350	16
12	GRAFICA VGA	80x30	640x480	16

ATTENZIONE: alcuni comandi - ad esempio TABVOB, EDIT - funzionano solamente in modalità testo (0).

Vedere anche: VLCS, FIELD, REDRAW, VCOLOR, VLOCATE, VPRINT

Esempio:

```
SCREEN 12
```

SDEFINITION $font %height %width %charspace %linespace %angle %rotation

$font .....font o tipo di carattere (.fon)
%height.....altezza del testo [mm] (<>0)
%width.....larghezza percentuale [%] (-400÷400)
%charspace .....(*)
%linespace .....(*)
%angle.....inclinazione o italicità dei caratteri [°] (-80÷80)
%rotation .....rotazione del testo [°]

(*) funzione futura

Il comando permette di definire completamente lo stile di testo.

Vedere anche: FNAME, HEIGHT, ITALIC, SNAME, SNUMBER, SROTATION, WIDTH, PRINT, WDEFINITION

Esempio:

```
SDEFINITION "simplex" 3 100 0 0 20 90
```

SETALIM %powersupply

%powersupply.....stato alimentatore? (0=off, 1=on)

Il comando permette di spegnere e disabilitare l'alimentatore oppure di riabilitarlo (ma non di riaccenderlo).

Vedere anche: LAMP, SETCO2, SETPLC, SETRELE, START, POKE, OUT

Esempio:

```
SETALIM 0
```

SETCO2 %co2

%co2.....stato del tubo laser CO^2? (0=spento, 1=acceso)

Il comando permette, utilizzando una marcatrice CO^2, di accendere o spegnere il tubo laser. Ad ogni esecuzione di script viene comunque acceso per default. Visto che il tubo impiega circa un secondo ad accendersi bisogna assicurarsi di non marcare immediatamente all'entrata di uno script e di attendere un tempo opportuno ad ogni riaccensione quando gestita direttamente. Assicurarsi di riaccendere il tubo dopo ogni spegnimento in quanto è possibile marcare anche a tubo spento senza avvertimenti e non avere quindi risultati reali.

Vedere anche: LAMP, SETALIM, SETPLC, SETRELE, START, POKE, OUT

Esempio:

```
SETCO2 1
```

SETPLC %plc

%plc.....stato PLC? (0=off, 1=on)

Il comando permette l'accensione o lo spegnimento del PLC.
Questo segnale solitamente viene utilizzato per gestire un'apparecchiatura esterna che lavora in concomitanza della marcatura.
Questo segnale è un uscita per un interfacciamento personalizzabile.

Vedere anche: LAMP, SETALIM, SETCO2, SETRELE, START, POKE, OUT

Esempio:

```
SETPLC 1
```

SETRELE %rele

%rele .....stato relé esterno? (0=off, 1=on)

Il comando permette l'accensione o lo spegnimento del relè esterno.
Questo segnale è un uscita per un interfacciamento personalizzabile.

Vedere anche: LAMP, SET ALIM, SETCO2, SETPLC, START, POKE, OUT

Esempio:

```
SETRELE 1
```

SHOWILK %autoexit

%autoexit.....uscita automatica? (0=no, 1=si)

Il comando mostra la maschera degli interlock.
Se l'uscita è automatica la maschera sparisce immediatamente al ripristino degli interlock, diversamente si deve uscire con ESC.
In caso sia già abilitata la maschera automatica (v. AUTOILK) essa compare e scompare indipendentemente.

Vedere anche: AUTOILK, GETSYSN

Esempio:

```
SHOWILK 0
```

SIN %sine %angle

%sine =.....seno dell'angolo (-1÷1)
%angle.....angolo in gradi [°]

Il comando trova il seno dell'angolo specificato.

Vedere anche: COS, TAN, ATAN

Esempio:

```
GETNUM %a "Angolo?" 0 10
SIN %s %a
COS %c %a
TAN %t %a
STR $a %a
STR $s %s
STR $c %c
MSG "Angolo, seno, coseno:" $a $s $c
STR $t %t
MSG "Angolo, tangente:" $a $t ""
ATAN %a1 %s %c
STR $a1 %a1
MSG "Angolo in base a seno e coseno:" $a1 "" ""
```

L'esempio mostra l'utilizzo delle funzioni trigonometriche.

SNAME $file

$file .....tabella stili di testo (.stl)

Il comando carica una nuova tabella stili di testo.
Nessuno stile viene però immediatamente attivato.

Vedere anche: FNAME, HEIGHT, ITALIC, SDEFINITION, SNUMBER, SROTATION, WIDTH, PRINT

Esempio:

```
SNAME "table2"
```

SNUMBER %style

%style .....stile di testo da attivare (0÷255)

Il comando attiva lo stile di testo della tabella corrente.
Assicurarsi che lo stile esista effettivamente in tabella.

Vedere anche: FNAME, HEIGHT, ITALIC, SDEFINITION, SNAME, SROTATION, WIDTH, PRINT

Esempio:

```
SNUMBER 5
```

SOUND %frequency %time

%frequency.....frequenza [Hz]
%time.....durata [1/18 s]

Il comando emette un segnale acustico di frequenza e durata specificate (il computer della marcatrice contiene un altoparlante?).

	Frequenze per Ottave				
Nota	I	II	III	IV	V
DO	131	262	523	1046	2093
DO#/REb	139	277	554	1108	2217
RE	147	294	587	1174	2349
RE#/MIb	156	311	622	1244	2489
MI	165	330	659	1318	2637
FA	175	349	698	1397	2794
FA#/SOLb	185	370	740	1480	2960
SOL	196	392	784	1568	3136
SOL#/LAb	208	415	831	1662	3322
LA	220	440	880	1760	3520
LA#/SIb	233	466	932	1864	3729
SI	248	494	988	1976	3951

Vedere anche: ERRMSG

Esempio:

```
SOUND 392 18 && SOL - II ottava
```

SPEED %speed

%speed.....velocità di marcatura [mm/s] (>0)

Il comando modifica la velocità di traslazione in marcatura dello stile di marcatura.

N. B.:
In base alla risoluzione corrente ed alla velocità della CPU ci sarà un limite massimo non superabile. Valori superiori di velocità non avranno effetto.

Vedere anche: CURRENT, FREQUENCY, FROTATION, RESOLUTION, WDEFINITION, WDIMENSION, WNAME, WNUMBER

Esempio:

```
SPEED 320
```

SQR %squareroot %number

%squareroot =.....radice quadrata
%number.....valore numerico (³0)

Il comando trova la radice quadrata di un valore.

Vedere anche: *, +, -, /

Esempio:

```
LET %cateto1 35
LET %cateto2 21
* %tmp1 %cateto1 %cateto1
* %tmp2 %cateto2 %cateto2
```

```
+ %tmp1 %tmp1 %tmp2
SQR %ipotenusa %tmp1
```

Il programma di esempio trova l'ipotenusa utilizzando i cateti ed il teorema di pitagora.

SROTATION %angle

%angle.....angolo di rotazione del testo [°]

Il comando imposta l'angolo di rotazione del testo nello stile di testo.

Vedere anche: FNAME, HEIGHT, ITALIC, SDEFINITION, SNAME, SNUMBER, WIDTH, PRINT

Esempio:

```
SROTATION 180
```

START %start

%start =.....stato pulsante di START? (0=non premuto, -1=premuto)

Il comando pone nella variabile lo stato del pulsante o pedale di START.
ATTENZIONE: il valore restituito nel caso il pulsante sia premiuto è -1 e non 1!
Questo segnale è un ingresso per un interfacciamento personalizzabile.

Vedere anche: LAMP, SETALIM, SETCO2, SETPLC, SETRELE, IN, PEEK

Esempio:

```
:riprova
START %s
IF= %s 0
     GOTO :riprova
ENDIF
```

```
PRINT "123456"
GOTO :riprova
```

Il programma di esempio attende la pressione del pulsante di START per marcare.

STR $string %number

$string =.....stringa
%number.....valore numerico

Il comando permette di convertire un valore numerico in stringa.

Vedere anche: VAL, CHR, ASC

Esempio:

```
LET %pi 3.14159
* %valore %pi 2
STR $stringa %valore
PRINT $stringa     && verrà stampato "6.28318"
```

STRING $string %number $character

$string =.....stringa risultante
%number.....numero di ripetizioni (1÷31)
$character.....carattere da ripetere (\001÷\255)

Il comando permette di costruire una stringa ripetendo più volte il carattere specificato.

Vedere anche: ADDSTR

Esempio:

```
GETNUM %n "Numero iniziale progressivo?" 0 5
+ %l %n 10
```

```
:loop
IF< %n %l
     STR $n %n
     STRING $z 6 "0"
     ADDSTR $n $z $n
     RIGHT $n $n 6
     MSG "Numero seriale:" $n "" ""
     GOTO :loop
ENDIF
```

L'esempio mostra la creazione di dieci numeri seriale progressivi a sei cifre sempre riempiti di zeri a partire da quello specificato.

TABSTILI $table

$table .....tabella stili di testo (.stl)

Il comando permette la modifica interattiva della tabella stili di testo specificata.

Vedere anche: TABVOB, AXSET, HPGLCONV

Esempio:

```
TABSTILI "table0"
```

TABVOB $table

$table .....tabella stili di marcatura (.wob)

Il comando permette la modifica interattiva della tabella stili di marcatura specificata.

Vedere anche: TABSTILI, AXSET, HPGLCONV

Esempio:

```
TABVOB "ev1"
```

TAN %tangent %angle

%tangent =.....tangente dell'angolo
%angle.....angolo in gradi [°] (<>90; <>270)

Il comando trova la tangente dell'angolo specificato.

Vedere anche: ATAN, SIN, COS

Esempio:

```
GETNUM %a "Angolo?" 0 10
SIN %s %a
COS %c %a
TAN %t %a
STR $a %a
STR $s %s
STR $c %c
MSG "Angolo, seno, coseno:" $a $s $c
STR $t %t
MSG "Angolo, tangente:" $a $t ""
ATAN %a1 %s %c
STR $a1 %a1
MSG "Angolo in base a seno e coseno:" $a1 "" ""
```

L'esempio mostra l'utilizzo delle funzioni trigonometriche.

TIME $time

$time .....ora di sistema [oo:mm:ss] (00:00:00÷23:59:59)

Il comando restituisce l'ora di sistema.

Vedere anche: DATE, GETRTCTIME, WAIT, FDATE, ELAPSED

Esempio:

```
TIME $t
```

TRIM $trim $text

$trim = ...testo senza spazi a sinistra nè a destra
$text .....testo iniziale

Il comando elimina tutti gli spazi consecutivi iniziali (a sinistra) e finali (a destra).

Vedere anche: LTRIM, RTRIM, LEFT, MID, RIGHT

Esempio:

```
TRIM $tr "     Bruce Springsteen     "
```

UCASE $uppercase $text

$uppercase =.....testo maiuscolo
$text .....testo

Il comando trasforma tutti i caratteri alfabetici in maiuscoli e lascia invariati gli altri.

Vedere anche: LCASE

Esempio:

```
UCASE $u "attenzione"
```

VAL %number $string

%number = ...valore numerico
$string.....stringa

Il comando converte una stringa in un valore numerico. Se non vi sono cifre nella stringa il risultato sarà 0.

Vedere anche: STR, ASC, CHR

Esempio:

```
VAL %g "9.80665 m/s²"
```

VCLS

Cancellazione del video.

Vedere anche: REDRAW, SCREEN, VCOLOR, VLOCATE, VPRINT, FIELD

Esempio:

```
VCLS
```

VCOLOR %foreground %background

%foreground.....primo piano (0÷31; v. tabella)
%background.....sfondo (0÷7; v. tabella)

Il comando imposta i colori di stampa a video secondo le seguenti tabelle:

```
+--------------------------++-------------------------+
¦          COLORI          ¦¦        ATTRIBUTI        ¦
¦ 0 - nero      4 - rosso   ¦¦  +8 - doppia luminosità ¦
¦ 1 - blu       5 - magenta ¦¦ +16 - lampeggio         ¦
¦ 2 - verde     6 - marrone ¦+-------------------------+
¦ 3 - azzurro   7 - bianco  ¦
+--------------------------+
```

NOTE:

- la doppia intensità e il lampeggio sono riservate al colore di primo piano
- il nero a doppia luminosità diventa grigio mentre il marrone diventa giallo
- sui monitor monocromatici i colori appaiono come sfumature di grigi
- se in configurazione sono disabilitati i colori vengono automaticamente trasformati, se diversi dal nero, in bianco, mantenendo gli attributi.

Vedere anche: VPRINT, VLOCATE, VCLS, FIELD, REDRAW, SCREEN

Esempio:

```
VCOLOR 14 1
```

VLOCATE %row %column

%row.....riga (1÷25)
%column.....colonna (1÷80)

Il comando posiziona alle coordinate specificate il cursore per la stampa a video. Lo spigolo in alto a sinistra è in 1, 1 mentre quello in basso a destra è in 25, 80.

Vedere anche: VPRINT, VCOLOR, VCLS, FIELD, REDRAW, SCREEN

Esempio:

```
VLOCATE 20 60
```

VOLCMD $command

$command.....comandi per volantino elettronico (v. tabella)

Il comando permette di mandare comandi al display del volantino elettronico.

```
Display Toshiba TLC-501:
+--------------------------------++----------------------------+
¦ Lista dei comandi principali:  ¦¦ Caratteristiche:           ¦
+--------------------------------¦+----------------------------¦
¦ FunctionSet = 56               ¦¦ Data Length = 8 bit        ¦
¦ DispOn,CursOff,BlinkOff = 12   ¦¦ Lines = 2x20 (2x40 virtual) ¦
¦                                ¦¦ Font = 5x7 dots            ¦
¦ Cls.........................1 ¦+----------------------------+
¦ Home........................2 ¦
¦ Entry Mode to Left..........4 ¦
¦    To Right.....: +2          ¦
¦    Shift Display: +1          ¦
¦ Display, Cursor & Blink OFF...8 ¦
¦    Display ON...: +4          ¦
¦    Cursor ON....: +2          ¦
¦    Blink ON.....: +1          ¦
¦ Shift Cursor Left...........16 ¦
¦    Right........: +4          ¦
¦    Screen.......: +8          ¦
¦ Function Set................32 ¦
¦    (data=4,lines=1,font=5x7)  ¦
¦    Data=8.......: +16         ¦
¦    Lines=2......: +8 (TLC-501) ¦
¦    Font=5x10....: +4          ¦
¦ Set DD RAM Address (locate).128 ¦
¦    2nd Line.....: +64         ¦
¦    Position.....: +0÷39       ¦
+--------------------------------+
```

Vedere anche: ENCODER, VOLDATA, VOLKEYS

Esempio:

```
VOLCMD "\056\013"
```

VOLDATA $text

$text .....testo per volantino elettronico

Il comando permette di visualizzare sul display del volantino elettronico il testo specificato. Per le modalità e le posizioni vedere il comando VOLCMD.

N. B.:
In questo solo caso il codice ASCII 232 viene interpretato come il codice ASCII 000, normalmente non disponibile.

Vedere anche: ENCODER, VOLCMD, VOLKEYS

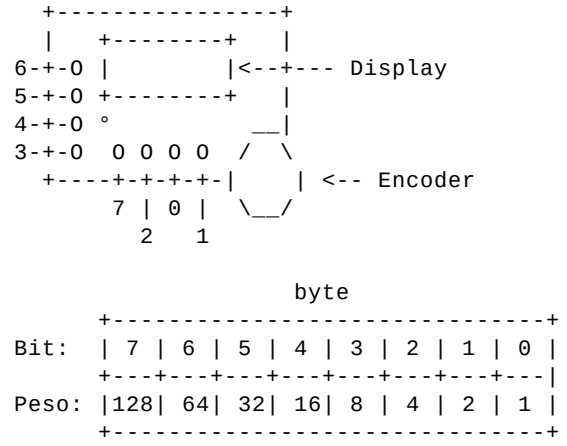
Esempio:

```
VOLDATA "Toshiba TLC-501"
```

VOLKEYS %byte

%byte =.....stato degli otto pulsanti (0÷255)

Il comando restituisce lo stato degli otto pulsanti (il 7 in realtà è un'interrutore elettronico).
All'accensione normalmente il pulsante 7 è acceso (codice: 128).
Di seguito la mappatura dei pulsanti relativi ai bit del byte:



	byte						
Bit:	7	6	5	4	3	2	1	0
Peso:	128	64	32	16	8	4	2	1

Vedere anche: ENCODER, VOLCMD, VOLDATA

Esempio:

```
VOLKEYS %k
```

VPRINT $text

$text.....testo da stampare a video

Il comando stampa a video il testo specificato alla posizione corrente del cursore con i colori correnti.

Vedere anche: VLOCATE, VCOLOR, VCLS, FIELD, REDRAW, SCREEN

Esempio:

```
VPRINT "* MENU PRINCIPALE *"
```

WAIT %time

%time.....attesa [ms]

Il comando permette l'attesa del tempo indicato.
L'utilizzo è fatto ad esempio per attendere la sincronizzazione di altre periferiche.

Vedere anche: RITAXIS, RITLONC, RITLONQ, RITSIM, DATE, GETRTCTIME, TIME, ELAPSED

Esempio:

```
WAIT 3500 && 3.5s
```

(*) WCLIP %xmin %ymin %xmax %ymax

%xmin.....ascissa minima [mm]
%ymin.....ordinata minima [mm]

%xmax.....ascissa massima [mm]
%ymax.....ordinata massima [mm]

(*) funzione futura

Il comando permette di definire la finestra di taglio (clipping window) fornendone le coordinate. Per default, all'esecuzione dello script, le coordinate sono impostate come l'area dell'obiettivo, offset compresi. Per abilitare la finestra di taglio utilizzare il comando CLIP (v.). Questo potente comando limita la marcatura ad una finestra mentre tutto ciò che sta intorno viene, appunto, tagliato ed omesso. Un'applicazione interessante è la marcatura di oggetti cilindrici su tutto lo sviluppo ruotando per gradi l'oggetto e marcandolo ogni volta con la parte del disegno interessata. Invece che dover preparare le parti di disegno staticamente separate esso può essere eseguito interamente demandando la selezione della parte interessata proprio alla finestra di taglio.

Vedere anche: CLIP

Esempio:

```
WCLIP 10 10 20 70
```

WDEFINITION %dimension $patternlibrary %number %speed %resolution %patternresolution %frequency %current %diafram %rotationfrequency $type

%dimension.....dimensione del pattern [mm] (³0)
$patternlibrary.....porre a "" (*)
%number.....porre a 0 (*)
%speed.....velocità di traslazione [mm/sec] (>0)
%resolution.....risoluzione [pt/mm] (>0)
%patternresolution.....(*)
%frequency.....frequenza di modulazione [%] (0÷100)
%current .....corrente o potenza [%] (0÷100)
%diafram .....(*)
%rotationfrequency.....frequenza di rotazione [Hz] (0÷255)
$type.....porre ad "H" (*)

(*) funzione futura

Il comando permette di definire completamente lo stile di marcatura.

Vedere anche: CURRENT, FREQUENCY, FROTATION, RESOLUTION, SPEED, WDIMENSION, WNAME, WNUMBER, SDEFINITION

Esempio:

```
WDEFINITION 2 "" 0 50 20 0 50 80 0 255 "H"
```

WDIMENSION %dimension

%dimension.....dimensione pattern di vobulazione [mm] (³0)

Il comando imposta la dimensione del pattern di vobulazione (diametro del tratto) dello stile di marcatura considerando.

Vedere anche: CURRENT, FREQUENCY, FROTATION, RESOLUTION, SPEED, WDEFINITION, WNAME, WNUMBER

Esempio:

```
WDIMENSION .8
```

WIDTH %width

%width.....larghezza del testo [%] (-400÷400)

Il comando imposta la larghezza percentuale (relativa all'altezza) dello stile di testo.

Vedere anche: FNAME, HEIGHT, ITALIC, SDEFINITION, SNAME, SNUMBER, SROTATION, PRINT

Esempio:

```
WIDTH 200
```

WNAME $file

$file .....tabella stili di marcatura (.wob)

Il comando carica una nuova tabella di stili di marcatura. Nessuno stile viene però immediatamente attivato.

Vedere anche: CURRENT, FREQUENCY, FROTATION, RESOLUTION, SPEED, WDEFINITION, WDIMENSION, WNUMBER, SNAME

Esempio:

```
WNAME "evlaser"
```

WNUMBER %style

%style .....stile di marcatura da attivare (0÷255)

Il comando attiva lo stile di marcatura della tabella corrente. Assicurarsi che lo stile esista effettivamente in tabella.

Vedere anche: CURRENT, FREQUENCY, FROTATION, RESOLUTION, SPEED,WDEFINITION, WDIMENSION, WNAME, SNUMBER

Esempio:

```
WNUMBER 8
```

XOR %xor %byte1 %byte2

%xor = ...XOR logico (0÷255)
%byte1.....primo byte (0÷255)
%byte2.....secondo byte (0÷255)

Il comando esegue lo XOR (OR esclusivo) logico tra i due byte, bit per bit.

```
 TABELLA DELLA VERITA`
+---------------------+
¦ bit 1 ¦ bit 2 ¦ XOR ¦
¦-------+-------+-----¦
¦   0   ¦   0   ¦  0  ¦
¦   0   ¦   1   ¦  1  ¦
¦   1   ¦   0   ¦  1  ¦
```

```
¦   1   ¦   1   ¦  0  ¦
+--------------------+
```

Vedere anche: AND, NOT, OR, BRESET, BSET, BTOGGLE

Esempio:

```
XOR %b 125 3
```

YESNO %answer $line1 $line2 $line3 $line4

%answer = ... risposta? (0=si, 1=no, 2=cancella)
$line1 ..... prima linea della domanda
$line2 ..... seconda linea della domanda
$line3 ..... terza linea della domanda
$line4 ..... quarta linea della domanda

Il comando mostra una finestra contenente la domanda (da una a quattro linee) ed attende conferma. In caso i parametri siano vuoti viene visualizzato il messaggio:

```
Richiesta generica di conferma
```

Vedere anche: MSG, ERRMSG, FLIST, GETNUM, GETSTR, MENU

Esempio:

```
YESNO %a "Cancellazione logo" $logo "Conferma?" ""
```