

AP-ETIS: IL CONTROLLORE PROGRAM-
MABILE E I PROGRAMMI DI GESTIONE

S. Cerri

B4-41 (1987)

Sirio Cerri

Istituto di Elaborazione della Informazione del CNR - Pisa

Il package AP-ETIS e' una procedura elaborativa progettata e realizzata nell'ambito di una collaborazione tecnico-scientifica tra la Societa' Aeritalia G.U.T. e l'Istituto di Elaborazione della Informazione per lo studio di sistemi orientati al controllo non distruttivo di componentistica aeronautica in materiale composito.

Il package opera nel controllore programmabile PC costituito dal personal computer MINUS KYBER con S.O. CP/M e fa parte integrante del sistema di sviluppo ETIS-1 la cui architettura e' sinteticamente descritta in fig.1

Il package software di seguito descritto e' relativo alla procedura di acquisizione di dati digitali ricavati dal controllore programmato CU. I dati vengono ricavati mediante il campionamento e la quantizzazione del segnale analogico relativo alle misure di assorbimento eseguite sul materiale in esame utilizzando tecniche ad ultrasuoni [1].

Il sistema di gestione AP-ETIS prevede la definizione dei parametri di acquisizione (coordinate di posizionamento iniziale, area di acquisizione, passo di campionamento); la configurazione della porta di I/O; l'invio dei comandi operativi; l'acquisizione dei dati; la loro preelaborazione per la riduzione del rumore; infine la rappresentazione dei dati su monitor TVC e l'archiviazione su memoria di massa.

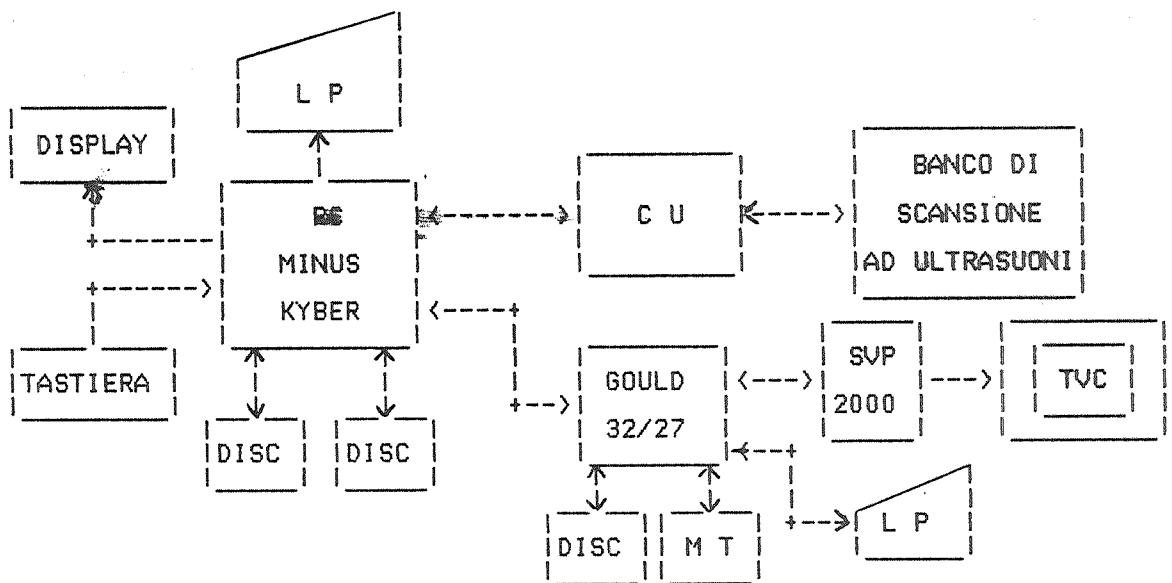


Fig. 1

Il PC MINUS KYBER: caratteristiche

- CPU: 280
- RAM: 64K
- EPROM: 4K
- Tastiera alfanumerica
- Video alfanumerico
- 2 disc drive da 5"
- 1 porta parallela per stampante
- 1 porta seriale RS-232 collegata ad un host computer (GOULD MPX-32)
- 1 porta seriale TTL modificata in RS 232 per migliorare l'affidabilità di collegamento) con il controllore programmato.

Il sistema operativo di cui e' dotato il PC e' il CP/M. Il programma di gestione e' scritto in Basic (data la semplicita' di apportare modifiche in fase di implementazione e messa a punto del sistema ETIS-1), che pero' e' interfacciato a routines scritte in Assembler per quanto riguarda il pilotaggio delle due porte seriali [2].

Dopo avere attivato il CP/M per dare il controllo al programma MENU e' sufficiente digitare il seguente comando:

```
SUBMIT MENU (.SUB)
```

dove SUBMIT e' il comando di sistema che attiva un file di comandi e MENU e' il file dei comandi cosi' strutturato:

```
MBASIC52 MENU.BAS /M:&HBFFF
```

dove:

- MBASIC52 e' l'interprete Basic che viene attivato e prende quindi il controllo sul CP/M.
- MENU.BAS e' un programma che verra' a sua volta attivato dall'interprete basic e presentera' sul display le varie operazioni eseguibili mostrate dalla tabella in fig.2.
- /M:&HBFFF e' la richiesta all'interprete basic di utilizzare la memoria fino alla locazione BFFF esadecimale, poiche' dalla locazione successiva verra' caricata la libreria utilizzata dai programmi selezionati da MENU.BAS.

Sul terminale video verranno presentate le varie operazioni eseguibili (v. fig. 2), si potra' quindi selezionare quella desiderata premendo il codice numerico corrispondente.

MENU' DI GESTIONE

- Acquisizione dati -

- 1 - Trasferimento dati su SVP-200
- 2 - Trasferimento dati su disco Gould 32/27
- 3 - Trasferimento dati su nastro magnetico

- Utilita' -

- 4 - Diagnostico del sistema di scansione
- 5 - Caricamento e configurazione driver DART
- 6 - Terminale Gould (MPX)

- Uscita dal menu' -

- 7 - Terminale Minus (BASIC)
- 8 - Terminale Minus (CFM)

Fig. 2 - Menu' disponibile sul controllore programmabile PC

I programmi di acquisizione differiscono tra loro solo nella parte relativa alla scelta della periferica di uscita:

- 1 - AT/SVP permette l'uscita dei dati sulla memoria video del sistema SVP-2000
- 2 - AT/DSK permette l'uscita dei dati su file di disco (GOULD)
- 3 - AT/NM permette l'uscita dei dati su nastro magnetico (GOULD)

Alla selezione di una delle tre operazioni segue il caricamento della libreria con l'inizializzazione delle porte seriali. Ovviamente si presuppone che le "periferiche" collegate siano pronte ad essere utilizzate; e' bene quindi assicurarsi che il controllore programmato e il S.O. MPX-32 siano operanti prima di attivare la richiesta.

Nella tabella che segue sono riportati i parametri relativi all'area da acquisire dipendenti dalla periferica scelta.

	INIZIO	FINE	PASSO	NX	NY
SVP	>=1 - <=fine	>=inizio - <=1000	>=1	>=1 - <=512	>=1 - <=512

	INIZIO	FINE	PASSO		
DSK/MT	>=1 - <=fine	>=inizio - <=1000	>=1	>=1 - <=1000	>=1 - <=1000

Una volta caricata la libreria sul display comparira' un nuovo menu' (fig. 3) che presenta una serie di comandi, selezionabili mediante codici, per attivare il processo di scansione e acquisizione.

- 1 - reset carrelli X,Y a 0,0
 - 2 - posizionamento carrello X
 - 3 - posizionamento carrello Y
 - 4 - acquisizione
 - 5 - disattivazione ATXXX GOULD
- CTRL-G - interruzione richiesta in corso su una delle linee seriali
CTRL-C - stop programma

Fig. 3

Selezionando 1 verra' richiesto al controllore programmato di posizionare i carrelli X e Y di scansione della testa di acquisizione, nella posizione 0,0 cioe' all'origine del sistema di riferimento. Una volta effettuata l'operazione verra' nuovamente presentato il menu' di fig. 3 per l'esecuzione di nuovi eventuali comandi.

Scegliendo 2, richiesta di posizionamento carrello X, fara' seguito la richiesta della coordinata X a cui si desidera posizionare il carrello X nel range tra 1 e 1000.

Selezionando 3, richiesta di posizionamento carrello Y, fara' seguito la richiesta della coordinata Y a cui si desidera posizionare il carrello Y nel range tra 1 e 1000.

Queste due operazioni sono utili per determinare la posizione assoluta del campione da esaminare in fase di acquisizione: cio' per

ridurre la superficie di scansione in fase di acquisizione e quindi il numero dei dati da manipolare.

Selezionando 4: Acquisizione, l'operatore dovrà inserire le coordinate di scansione nell'ordine di presentazione sul monitor:

Scansione X (inizio, fine, passo)

dove inizio è la coordinata X del primo punto da acquisire; fine è la coordinata dell'ultimo punto da acquisire; passo è l'incremento di posizionamento relativo.

Scansione Y (inizio, fine, passo)

dove inizio è la coordinata Y del primo punto da acquisire; fine è la coordinata dell'ultimo punto da acquisire; passo è l'incremento di posizionamento relativo.

ed infine Accelerazione X definisce lo spazio di accelerazione e decelerazione costante.

Tutti i parametri verranno controllati per assicurare che rientrino nei ranges prestabiliti come segue:

INIZIO	FINE	PASSO
$\geq 1 - \leq \text{fine}$	$\geq \text{inizio} - \leq 1000$	≥ 1

Se è stato scelto un trasferimento sulla memoria video del sistema di rappresentazione SVP 2000 il numero di punti definiti nella matrice di acquisizione durante il caricamento dei parametri non può superare le dimensioni di 512x512 punti; una volta controllati, i parametri vengono inviati al controllore programmato CU.

Fino a questo punto abbiamo istruito il controllore programmato che rimarrà in attesa di uno start per iniziare l'acquisizione dei dati; dobbiamo quindi ora configurare la periferica di uscita: andrà quindi attivato il programma che gestisce la periferica selezionata sotto il S.O. MPX-32.

Automaticamente il programma AT/XXX:

- effettua il LOG-IN sul S.O. MPX-32 operante sul sistema Gould 32/27, spedendo il carattere di "attention ESC";
- risponde con lo "user's name" MINUS e "Key" MINUS;
- controlla di essere in ambiente TSM, attiva il batch BATXXX e quindi esegue LOG-OUT.

BATXXX schedula il programma ATXXX di gestione della porta seriale collegata con il MINUS e della periferica di uscita selezionata. Tale programma rimarra' in attesa di comandi da parte dell'AT/XXX che, come prima verifica, spedira' la richiesta di "STATO" per assicurarsi che il collegamento sia attivo. Verra' stampato quindi su display, se tutto e' in ordine, il messaggio "Programma ATXXX attivo (BATXXX sotto TSM MPX-32 GOULD)" altrimenti verra' segnalato errore [3].

Puo' essere ora configurata la periferica di uscita selezionata.

SVP

"Reset del sistema SVP 2000 (SI/NO)?"

Rispondendo SI il sistema SVP 2000 verra' inizializzato come in fase di accensione, e cioe' le tre memorie verranno poste ad un valore costante, le tavole di look-up RGB verranno caricate con una rampa e verra' presentata la memoria 1 sul banco 1.

"Memoria (1-3), Banco (1-4)?" - Viene selezionata e visualizzata la memoria ed il banco di memoria desiderata.

"Cancellazione memoria (SI/NO)?" - A seconda della risposta, la memoria attualmente selezionata verra' cancellata.

"Coordinate iniziali SVP 2000(X,Y)?" - I valori selezionati x e y definiscono il vertice in alto a sinistra della matrice che verra' acquisita e ricostruita sul video pittorico. Tenere presente che questi valori sommati rispettivamente alle dimensioni X,Y della matrice da acquisire non devono superare 512: se cio' avviene verra' ripetuta la richiesta.

Con questa ultima definizione la configurazione e' completata e inizia quindi la fase di acquisizione: sul display alfanumerico comparira' il numero di riga che di volta in volta verra' letta dal sistema di scansione e contemporaneamente sul monitor pittorico questa stessa riga verra' scritta ricostruendo l'immagine. Completata la scansione verra' nuovamente presentato il menu' di fig. 2.

DSK

"Nome file (max 6 caratteri)" - Viene richiesto il nome del file che verra' creato sul disco Gould e dove verranno immagazzinati i dati da acquisire. Il formato del file e' compatibile a quello GE.FI. utilizzato sotto MPX nello standard che e' stato scelto. Se in fase di creazione avviene un errore, questo verra' denunciato con il messaggio "Errore in creazione file" seguito dal numero di errore.

Se tutto e' corretto, verra' iniziata la scansione ed ogni riga acquisita verra' trasferita sul file. Completata la scansione il file verra' chiuso e sul monitor saranno riportate le caratteristiche del file stesso, quindi per proseguire premere return.

NM

"Montare nastro con anello di scrittura su unita' 0, premere return quando pronto"

"Numero di file da saltare" - Specifica il numero dei file da saltare prima di scrivere quello che verra' acquisito.

A questa risposta segue la scansione e quindi le righe acquisite verranno scritte in maniera sequenziale sulla banda magnetica; a fine scansione verra' scritto un end-of-file e sul monitor verranno visualizzate le caratteristiche del file. Per continuare premere return.

La scelta dell'opzione 5 "Disattivazione ATXXX (GOULD)" verra' selezionata nel caso che non si desideri effettuare altre acquisizioni: essa si preoccupa di far terminare il programma batch sul S.O. MPX-32. L'opzione CTRL-G verra' utilizzata nel caso di gravi errori di trasferimento sulle porte seriali. Essa terminera' il programma AT/XXX, lasciando attivo pero' il programma ATXXX (GOULD). L'ultima opzione, CTRL-C, verra' usata per terminare il programma AT/XXX, e come la precedente, lascerà attivo il programma ATXXX su Gould.

Seguono le liste dei programmi implementati:

MENU	- gestore
AT/SUPE	- acquisizione su SVP-2000
AT/DSKE	- acquisizione su file di disco (Gould)
AT/NME	- acquisizione su file di nastro magnetico (Gould)
DGSUPE	- diagnostico del sistema di scansione
CONF	- configurazioe delle porte seriali del PC
MPXE	- emulatore del terminale sotto MPX-32.

RIFERIMENTI

- [1] L. Azzarelli, M. Chimenti: "Progetto del sistema ETISR per il controllo di qualita` ad ultrasuoni". Rapporto Aeritalia, 1986
- [2] S. Cerri: "Libreria per la gestione delle porte seriali sotto CP/M e manipolazione dati". Nota B4-40 IEI-CNR, 1987.
- [3] S. Cerri, E. Fantini: "Gestione di periferiche sotto sistema operativo MPX-32 (Gould) in collegamento con personal computer" Nota B4-42 IEI-CNR, 1987.

```

10 REM *** PISA 31/3/87
20 REM *** MENU ***
30 PRINT CHR$(12);TAB(10)"*****"
40 PRINT TAB(10)"*TAB(70)*"
50 PRINT TAB(10)"*TAB(28)Menu' di gestione ETIS-2"TAB(70)*"
60 PRINT TAB(10)"*TAB(70)*"
70 PRINT TAB(10)"*****"
80 PRINT:PRINT TAB(30)"- Acquisizione dati -"
90 PRINT:PRINT TAB(10)"1 - Trasferimento dati su SVP-2000"
100 PRINT TAB(10)"2 - Trasferimento dati su disco Gould 32/27"
110 PRINT TAB(10)"3 - Trasferimento dati su nastro magnetico"
120 PRINT:PRINT TAB(35)"- Utilita' -"
130 PRINT:PRINT TAB(10)"4 - Diagnostico del sistema di scansione"
140 PRINT TAB(10)"5 - Caricamento e configurazione driver DART"
150 PRINT TAB(10)"6 - Terminale Gould (MPX)"
160 PRINT:PRINT TAB(30)"- Uscita dal menu' -"
170 PRINT:PRINT TAB(10)"7 - Terminale Minus (BASIC)"
180 PRINT TAB(10)"8 - Terminale Minus (CP/M)"
190 PRINT:PRINT TAB(10);:INPUT K
200 IF K<1 OR K>8 THEN 20
210 PRINT CHR$(12)
220 ON K GOTO 230,250,270,290,310,330,350,370
230 CHAIN "AT/SVPE.BAS"
240 STOP
250 CHAIN"AT/DSKE.BAS"
260 STOP
270 CHAIN "AT/MTE.BAS"
280 STOP
290 CHAIN"DGSVPE.BAS"
300 STOP
310 CHAIN"CONFE.BAS"
320 STOP
330 CHAIN "MPXE.BAS"
340 STOP
350 PRINT"BASIC":NEW:END
360 STOP
370 PRINT "CP/M":SYSTEM
380 STOP
390 END

```

```

:0 REM *** PISA 31/3/87
20 REM *** AT/SVPE : Acquisizione su SVP2000
30 GOTO 690
40 REM*****
50 REM
60 REM *** carica driver da file DARTE.COM e inizializza canali A/B (RS232-C)
70 REM
80 REM*****
90 OPEN "R",#1,"DARTE.COM" : FIELD #1,128 AS A$: LA=&HC000
100 FOR I=1 TO 7 : GET #1,I : FOR K=1 TO 128 : POKE LA,ASC(MID$(A$,K,1))
110 LA=LA+1:NEXT K:NEXT I:CLOSE #1
120 ST%=-1:CALL INIZ%(CHCA%,ST%):IF ST%(<0) THEN PRINT"Err INIZ A:st=";ST%:STOP
130 ST%=-1:CALL INIZ%(CHCB%,ST%):IF ST%(<0) THEN PRINT"Err INIZ B:st=";ST%:STOP
140 RETURN
150 REM*****
160 REM
170 CALL BRWS%(CWA%,A$,ST%) : RETURN : REM *** ch A write A$ (CTRL-G)
180 CALL BRWS%(CWB%,A$,ST%) : RETURN : REM *** ch B write A$ (CTRL-G)
190 REM
200 CALL BRWS%(CWA%,A$,ST%) : RETURN : REM *** ch A write A$ (T.O.)
210 CALL BRWS%(CWB%,A$,ST%) : RETURN : REM *** ch B write A$ (T.O.)
220 REM
230 CALL BRV%(CWA%,B%(0),STAT%(0)) :RETURN : REM *** ch A write B% (CTRL-G)
240 CALL BRV%(CWB%,B%(0),STAT%(0)) :RETURN : REM *** ch B write B% (CTRL-G)
250 REM
260 CALL BRV%(CWA%,B%(0),STAT%(0)) : RETURN : REM *** ch A write B% (T.O.)
270 CALL BRV%(CWB%,B%(0),STAT%(0)) : RETURN : REM *** ch B write B% (T.O.)
280 REM
290 A$=SPACE$(255) : CALL BRWS%(CWA%,A$,ST%) : RETURN : REM *** ch A read A$ (CTRL-G)
300 A$=SPACE$(255) : CALL BRWS%(CWB%,A$,ST%) : RETURN : REM *** ch B read A$ (CTRL-G)
310 REM
320 A$=SPACE$(255) : CALL BRV%(CWA%,A$,ST%) : RETURN : REM *** ch A read A$ (T.O.)
330 A$=SPACE$(255) : CALL BRV%(CWB%,A$,ST%) : RETURN : REM *** ch B read A$ (T.O.)
340 REM
350 CALL BRV%(CWA%,B%(0),STAT%(0)) : RETURN : REM *** ch A read B% (CTRL-G)
360 CALL BRV%(CWB%,B%(0),STAT%(0)) : RETURN : REM *** ch B read B% (CTRL-G)
370 REM
380 CALL BRV%(CWA%,B%(0),STAT%(0)) : RETURN : REM *** ch A read B% (T.O.)
390 CALL BRV%(CWB%,B%(0),STAT%(0)) : RETURN : REM *** ch B read B% (T.O.)
400 REM*****
410 REM

```

```

420 REM *** log-in GOULD
430 REM
440 REM*****
450 F=0
460 A$="STOP"+CR$+"X"+CR$+"X"+CR$+CR$+CR$
470 GOSUB 170
480 A$=CHR$(27);B%(0)=2000 : GOSUB 170
490 GOSUB 380:IF STAT%(1)=0 THEN 490
500 A$="MINUS"+CR$ : GOSUB 170 : REM *** nome
510 GOSUB 380:IF STAT%(1)=0 THEN 510
520 GOSUB 170 : REM *** chiave
530 GOSUB 380:IF STAT%(1)=0 THEN 530
540 IF STAT%(1)<7 THEN 500
550 IN=INT(STAT%(1)/2)-2
560 B%(IN)=4:IF (STAT%(1) MOD 2)(>0 THEN B%(IN)=5
570 B$="":CALL SPAK%(B%(IN),A%(0),ST%)
580 FOR I=A%(0)-3 TO A%(0):B$=B$+CHR$(A%(I)):NEXT I:REM *** Es=ultimi 4 caratt. di B%
590 IF B$="TSM)" THEN RETURN
600 F=F+1 : IF F(<=3 THEN 460
610 RETURN
620 REM*****
630 REM
640 REM*****
650 REM
660 REM *** inizio programma
670 REM
680 REM*****
690 DIM B%(1000),STAT%(1),A%(1000)
700 BRW$=&HCOEE : BRWV%=&HCO59 : ST%=0 : CR$=CHR$(13) : B%(0)=2000
710 PAK%=&HC27E:SPAK%=&HC2AF:INV%=&HC2E2:INIZ%=&HC000:CHCA%=&H42:CHCB%=&H43
720 CWRSA%=&H40:CWRBX%=&H41:CWMAX%=&H140:CWNB%=&H141
730 CWRTOA%=&H240:CWRTOB%=&H241:CWNTOA%=&H340:CWNTOB%=&H341
740 PRINT CHR$(12):INPUT"Caricamento driver ?":A$
750 IF A$="no" OR A$="NO" THEN CALL INIZ%(CHCB%,ST%):GOTO 790
760 IF A$(">"SI" AND A$("<"s)" THEN GOTO 740
770 GOSUB 90
780 REM PRINT "STATO INIZ":ST%:STOP
790 PRINT CHR$(12);TAB(10)"*****"
800 PRINT TAB(10)"*TAB(70)"*

```

```

810 PRINT TAB(10)*"TAB(28)"Acquisizione su SVP2000"TAB(70)*"
820 PRINT TAB(10)*"TAB(70)*"
830 PRINT TAB(10)*****"Reset carrelli (X,Y=0)"*****"
840 PRINT:PRINT TAB(10)"1 - Reset carrelli (X,Y=0)"
850 PRINT TAB(10)"2 - Posizionamento X"
860 PRINT TAB(10)"3 - Posizionamento Y"
870 PRINT TAB(10)"4 - Acquisizione"
880 PRINT TAB(10)"5 - Stop programma ATSVP Gould"
890 PRINT:PRINT TAB(10)"CTRL-G - Break della richiesta in corso su RS232/TTL"
900 PRINT:PRINT TAB(10)"CTRL-C - Stop programma"
910 PRINT:PRINT TAB(10);:INPUT K
920 IF K<1 OR K>5 THEN GOTO 910
930 ON K GOTO 940,1160,1270,1400,1090
940 REM *
950 REM * RESET
960 REM *
970 A$="R"
980 GOSUB 180:REM *** Richiesta di reset carrelli x-y atius
990 GOSUB 300:REM *** aspetto K
1000 IF STX=0 AND A$="K" THEN PRINT "Rispoosta su reset :";A$ : GOTO 840
1010 IF STX<>7 THEN PRINT"Errore di reset":STOP
1020 REM *
1030 REM *** Richiesta di abort
1040 REM *
1050 A$="H":REM *** richiesta di abort
1060 GOSUB 180:GOSUB 300 :IF A$="K" THEN GOTO 840
1070 PRINT "Errore su risposta a H":STOP
1080 REM *
1090 REM *** Disattivazione prog. ATSVP
1100 REM *
1110 A$="STOP"
1120 GOSUB 170 : REM *** manda "STOP"
1130 GOSUB 290 : REM *** Aspetta "K"
1140 IF A$="K" THEN PRINT"ATSVP Gould inattivo":GOTO 840
1150 PRINT"Errore su richiesta di stop a ATSVP":GOTO 840
1160 REM *
1170 REM *** Posizionamento X
1180 REM *
1190 INPUT"Coordinata X :";X
1200 N1=INT(X/256):N2=X-(N1*256)
1210 A$="X"+CHR$(N2)+CHR$(N1)

```

```

1220 GOSUB 180:REM *** richiesta posiz. X
1230 GOSUB 300:REM *** aspetto K/E
1240 IF ST%=0 AND A#="K" THEN GOTO 840
1250 IF ST%>7 THEN PRINT"Errore posiz. X":STOP
1260 PRINT"CTRL-G":STOP
1270 REM *
1280 REM *** Posizionamento Y
1290 REM *
1300 INPUT"Coordinata Y :";Y
1310 Y=Y*5
1320 N1=INT(Y/256):N2=Y-(N1*256)
1330 A#="Y"+CHR$(N2)+CHR$(N1)
1340 GOSUB 180:REM *** richiesta posiz. Y
1350 GOSUB 300:REM *** aspetto K/E
1360 IF ST%=0 AND A#="K" THEN GOTO 840
1370 IF ST%>7 THEN PRINT"Errore posiz. Y":STOP
1380 PRINT "CTRL-G":STOP
1390 REM **
1400 REM ***** COORDINATE DI SCANSIONE
1410 REM **
1420 INPUT"Scansione X (inizio, fine, passo)";XY(1),XY(4),XY(9)
1430 INPUT"Scansione Y (inizio, fine, passo)";XY(5),XY(8),XY(10)
1440 INPUT"Rincorsa su X ";IS
1450 IF IS(1 OR IS)1000 THEN GOTO 1440
1460 IF XY(1)<IS OR XY(4)>1000-IS OR XY(1)>XY(4) THEN GOTO 1420
1470 IF XY(5)<1 OR XY(8)>1000 OR XY(5)>XY(8) THEN GOTO 1420
1480 IF XY(9)<1 OR XY(10)<1 THEN GOTO 1420
1490 XY(2)=XY(1)-IS
1500 XY(3)=XY(4)+IS
1510 XY(6)=XY(5) : XY(7)=XY(8)
1520 FOR I=5 TO 8:XY(I)=XY(I)*5:NEXT I:XY(10)=XY(10)*5
1530 NX=INT((XY(4)-XY(1))/XY(9))+1
1540 NY=INT((XY(8)-XY(5))/XY(10))+1
1550 IF NX>512 OR NY>512 THEN PRINT"NX o NY >512":GOTO 1470
1560 HEY=INT(NX/256):LBY=NX-(HEY*256)
1570 A#="N"+CHR$(LBY)+CHR$(HEY)
1580 GOSUB 180
1590 GOSUB 300
1600 IF ST%=0 AND A#="K" THEN GOTO 1630

```



```

1220 GOSUB 180:REM *** richiesta posiz. X
1230 GOSUB 300:REM *** aspetto K/E
1240 IF ST%=0 AND A$="K" THEN GOTO 840
1250 IF ST%>7 THEN PRINT"Errore posiz. X":STOP
1260 PRINT"CTRL-G":STOP
1270 REM *
1280 REM *** Posizionamento Y
1290 REM *
1300 INPUT"Coordinata Y ";Y
1310 Y=Y*5
1320 N1=INT(Y/256):N2=Y-(N1*256)
1330 A$="Y"+CHR$(N2)+CHR$(N1)
1340 GOSUB 180:REM *** richiesta posiz. Y
1350 GOSUB 300:REM *** aspetto K/E
1360 IF ST%=0 AND A$="K" THEN GOTO 840
1370 IF ST%>7 THEN PRINT"Errore posiz. Y":STOP
1380 PRINT "CTRL-G":STOP
1390 REM **
1400 REM **** COORDINATE DI SCANSIONE
1410 REM **
1420 INPUT"Scansione X (inizio, fine, passo)";XY(1),XY(4),XY(9)
1430 INPUT"Scansione Y (inizio, fine, passo)";XY(5),XY(8),XY(10)
1440 INPUT"Rincorsa su X ";IS
1450 IF IS(1 OR IS)1000 THEN GOTO 1440
1460 IF XY(1)(IS OR XY(4))1000-IS OR XY(1))XY(4) THEN GOTO 1420
1470 IF XY(5)(1 OR XY(8))1000 OR XY(5))XY(8) THEN GOTO 1420
1480 IF XY(9)(1 OR XY(10)(1 THEN GOTO 1420
1490 XY(2)=XY(1)-IS
1500 XY(3)=XY(4)+IS
1510 XY(6)=XY(5) : XY(7)=XY(8)
1520 FOR I=5 TO 8:XY(I)=XY(I)*5:NEXT I:XY(10)=XY(10)*5
1530 NX=INT((XY(4)-XY(1))/XY(9))+1
1540 NY=INT((XY(8)-XY(5))/XY(10))+1
1550 IF NX>512 OR NY>512 THEN PRINT"NX o NY >512":GOTO 1470
1560 HEY=INT(NX/256):LBY=NX-(HEY*256)
1570 A$="N"+CHR$(LBY)+CHR$(HEY)
1580 GOSUB 180
1590 GOSUB 300
1600 IF ST%=0 AND A$="K" THEN GOTO 1630

```

```

1610 IF ST%( ) > 7 THEN PRINT "Errore su NX":STOP
1620 PRINT "CTRL-G":STOP
1630 FOR I=1 TO 10
1640 HEY=INT(XY(I)/256):LEY=XY(I)-(HEY*256)
1650 A#=CHR$(I+48)+CHR$(LEY)+CHR$(HEY)
1660 IF I=9 THEN A#="x"+CHR$(LEY)+CHR$(HEY)
1670 IF I=10 THEN A#="y"+CHR$(LEY)+CHR$(HEY)
1680 GOSUB 180
1690 GOSUB 300
1700 IF ST%( ) < 0 OR A#( )="K" THEN GOTO 1730
1710 PRINT"OK ";I;XY(I)
1720 NEXT I:GOTO 1750
1730 PRINT "Errore su ";I; ST = ";ST%"; ricevuto ";A#
1740 STOP
1750 INPUT"Attivazione ATSVP(Gould) ASI/NOU :";SVP#
1760 IF SVP#="NO" OR SVP#="no" THEN GOTO 1860
1770 IF SVP#( )="SI" AND SVP#( )="si" THEN GOTO 1750
1780 PRINT CHR#(&HC):REM *** cancella lo schermo
1790 FOR I=1 TO 10:PRINT CHR#(10);NEXT I
1800 PRINT "Log-in Gould MINUS/MINUS : attendere"
1810 GOSUB 450: REM *** Log-in Gould
1820 PRINT CHR#(12):IF F>3 THEN PRINT"Collegamento fallito.":END
1830 A#="BATCH BATSVP"+CR#+X"+CR#
1840 GOSUB 170:REM *** Attivazione in batch del programma ATSVP e log-off Gould
1850 GOSUB 380:REM *** Lettura log-off e attesa del T.O.
1860 A#="STAT":GOSUB 170:GOSUB 290:IF ST%=7 THEN STOP:REM ***Rich. di stato in A#
1870 IF A#( )="ATSVP" THEN PRINT"Errore ATSVP Gould":STOP
1880 PRINT"Programma ";A#;" attivo (batch BATSVP sotto TSM MPX-32 Gould)"
1890 INPUT"Reset del sistema SVP-2000 (si/no) :";A#
1900 IF A#="NO" OR A#="no" THEN 1950
1910 IF A#( )="SI" AND A#( )="si" THEN 1890
1920 A#="SUB"+CHR#(1):GOSUB 170:REM *** Richiesta di reset su SVP2000
1930 GOSUB 290:REM *** Lettura in A#, se=K ok
1940 IF A#( )="K" THEN PRINT "Errore":STOP
1950 PRINT"Configurazione SVP-2000"
1960 INPUT"Memoria (1-3), Banco (1-4) :";M,B
1970 IF M(1 OR M)3 OR B(1 OR B)4 THEN 1960
1980 A#="SUB"+CHR#(2):GOSUB 170:REM *** BANCO
1990 GOSUB 290:REM *** Aspetto K
2000 IF A#( )="K" THEN PRINT"Errore":STOP

```

```

2010 A#=CHR$(B)+CHR$(M):GOSUB 170:REM *** B=banco
2020 GOSUB 290:REM ***Aspetto K
2030 IF A#(">"K) THEN PRINT "Errore":STOP
2040 A#="SUB"+CHR$(3):GOSUB 170:REM *** INTM
2050 GOSUB 290
2060 IF A#(">"K) THEN PRINT"Errore":STOP
2070 A#=CHR$(M):GOSUB 170:REM *** M=memoria
2080 GOSUB 290:REM *** Aspetto K
2090 IF A#(">"K) THEN PRINT"Errore":STOP
2100 INPUT"OK ";A#
2110 IF A#="ND" OR A#="no" THEN 1960
2120 IF A#(">"SI" AND A#(">"si" THEN 2100
2130 INPUT"Cancelazione memoria (si/no) ";A#
2140 IF A#="ND" OR A#="no" THEN 2220
2150 IF A#(">"SI" AND A#(">"si" THEN 2130
2160 A#="SUB"+CHR$(6):GOSUB 170:REM *** MEMC
2170 GOSUB 290:REM *** Aspetto K
2180 IF A#(">"K) THEN PRINT"Errore":STOP
2190 A#=CHR$(M):GOSUB 170:REM *** M=memoria
2200 GOSUB 290:REM *** Aspetto K
2210 IF A#(">"K) THEN PRINT"Errore":STOP
2220 INPUT"Coordinate iniziali SVP (X,Y) ";IX,IY
2230 IX=IX-1:IY=IY-1
2240 IF IX+NX>512 OR IY+NY>512 THEN GOTO 2220
2250 A#="SUB"+CHR$(14)
2260 GOSUB 170:REM SCRIVE A#
2270 GOSUB 290:REM *** Aspetto K
2280 IF A#(">"K) THEN PRINT"Errore":STOP
2290 B%(0)=9:B%(1)=IY:B%(2)=IY+NY-1:B%(3)=IX:B%(4)=IX+NX-1:B%(5)=M
2300 GOSUB 230: REM SCRIVE B% (PARAMETRI AREA)
2310 GOSUB 290: REM ASPETTO K
2320 IF A#(">"K) THEN PRINT"Errore":STOP
2330 FOR RY=1 TO NY
2340 B%(0)=NX+1
2350 A#="S":PRINT"Start riga RY = ";RY:GOSUB 180
2360 GOSUB 360:REM *** Lettura riga
2370 IF STAT%(0)(">0 THEN PRINT"Errore su riga RY = ";RY:STOP
2380 IF RY(">NY THEN GOTO 2410
2390 GOSUB 300:REM *** aspetta K
2400 IF STAT%(0) OR A#(">"K) THEN PRINT"Err. fine scansione":STOP

```

```
2410 B%(0)=NX:IF (RY MOD 2)=0 THEN CALL INV%(B%(0),ST%)
2420 PRINT"Scrivo su SVP2000 riga n. ";RY
2430 GOSUB 230:REM Scrittura riga su SVP
2440 GOSUB 290:REM *** Aspetto K
2450 IF A#()="K" THEN PRINT"Errore scrittura SVP":STOP
2460 NEXT RY
2470 GOTO B40
```

```

.0 REM *** PISA 3/4/87
20 REM *** AT/DSKE : Acquisizione su file di disco (Gould)
30 GOTO 700
40 REM*****
50 REM
60 REM *** carica driver da file DARTE.COM e inizializza canali A/B (RS232-C)
70 REM
80 REM*****
90 OPEN "R",#1,"DARTE.COM" : FIELD #1,128 AS A$ : LA=&HCOOO
100 FOR I=1 TO 7 : GET #1,I : FOR K=1 TO 128 : POKE LA,ASC(MID$(A$,K,1))
110 LA=LA+1:NEXT K:NEXT I:CLOSE #1
130 ST%=1:CALL INIZ%(CHCA%,ST%):IF ST%<>0 THEN PRINT"Err INIZ A:st=";ST%:STOP
140 ST%=-1:CALL INIZ%(CHCB%,ST%):IF ST%<>0 THEN PRINT"Err INIZ B:st=";ST%:STOP
150 RETURN
160 REM*****
170 REM
180 CALL BRW$(CWA%,A$,ST%) : RETURN : REM *** ch A write A$ (CTRL-G)
190 CALL BRW$(CWB%,A$,ST%) : RETURN : REM *** ch B write A$ (CTRL-G)
200 REM
210 CALL BRW$(CWA%,A$,ST%) : RETURN : REM *** ch A write A$ (T.O.)
220 CALL BRW$(CWB%,A$,ST%) : RETURN : REM *** ch B write A$ (T.O.)
230 REM
240 CALL BRW$(CWA%,B%(0),STAT%(0)) :RETURN : REM *** ch A write B% (CTRL-G)
250 CALL BRW$(CWB%,B%(0),STAT%(0)) :RETURN : REM *** ch B write B% (CTRL-G)
260 REM
270 CALL BRW$(CWA%,B%(0),STAT%(0)) : RETURN : REM *** ch A write B% (T.O.)
280 CALL BRW$(CWB%,B%(0),STAT%(0)) : RETURN : REM *** ch B write B% (T.O.)
290 REM
300 A#=SPACE$(255) : CALL BRW$(CWA%,A$,ST%) : RETURN : REM *** ch A read A$ (CTRL-G)
310 A#=SPACE$(255) : CALL BRW$(CWB%,A$,ST%) : RETURN : REM *** ch B read A$ (CTRL-G)
320 REM
330 A#=SPACE$(255) : CALL BRW$(CWA%,A$,ST%) : RETURN : REM *** ch A read A$ (T.O.)
340 A#=SPACE$(255) : CALL BRW$(CWB%,A$,ST%) : RETURN : REM *** ch B read A$ (T.O.)
350 REM
360 CALL BRW$(CWA%,B%(0),STAT%(0)) : RETURN : REM *** ch A read B% (CTRL-G)
370 CALL BRW$(CWB%,B%(0),STAT%(0)) : RETURN : REM *** ch B read B% (CTRL-G)
380 REM
390 CALL BRW$(CWA%,B%(0),STAT%(0)) : RETURN : REM *** ch A read B% (T.O.)
400 CALL BRW$(CWB%,B%(0),STAT%(0)) : RETURN : REM *** ch B read B% (T.O.)

```

```

410 REM*****
420 REM
430 REM *** log-in GOULD
440 REM
450 REM*****
460 F=0
470 A$="STOP"+CR$+"X"+CR$+"X"+CR$+CR$+CR$
480 GOSUB 180
490 A#=CHR$(27)+B%(0)=2000 : GOSUB 180
500 GOSUB 390:IF STAT%(1)=0 THEN 500
510 A#="MINUS"+CR$ : GOSUB 180 : REM *** nome
520 GOSUB 390:IF STAT%(1)=0 THEN 520
530 GOSUB 180 : REM *** chiave
540 GOSUB 390:IF STAT%(1)=0 THEN 540
550 IF STAT%(1) < 7 THEN 610
560 IN=INT(STAT%(1)/2)-2
570 B%(IN)=4:IF (STAT%(1) MOD 2) < 0 THEN B%(IN)=5
580 B$="":CALL SPAK%(B%(IN),A%(0),ST%)
590 FOR I=A%(0)-3 TO A%(0):B$=B$+CHR$(A%(I)):NEXT I:REM *****ultimi 4 caratt. di B%
600 IF B$="TSM)" THEN RETURN
610 F=F+1 : IF F <=3 THEN 470
620 RETURN
630 REM*****
640 REM
650 REM*****
660 REM
670 REM *** inizio programma
680 REM
690 REM*****
700 DIM B%(1000),STAT%(1),A%(1000)
710 BRWS%=&HC0EE : BRWV%=&HC059 : ST%=0 : CR$=CHR$(13) : B%(0)=2000
720 PAK%=&HC27E:SPAK%=&HC2AF:INV%=&HC2E2:INIZ%=&HC000:CHCA%=&H42:CHCB%=&H43
730 CWRA%=&H40:CWRB%=&H41:CWRA%=&H140:CWRB%=&H141
740 CWRTOA%=&H240:CWRTOB%=&H241:CWNTOA%=&H340:CWNTOB%=&H341
750 PRINT CHR$(12):INPUT"Caricamento driver ?";A$
760 IF A$="no" OR A$="NO" THEN GOTO 800
770 IF A$(">SI" AND A$(">si" THEN GOTO 750
780 GOSUB 90
790 REM PRINT "STATO INIZ";ST%:STOP
800 PRINT CHR$(12);TAB(10)"*****"

```

```

810 PRINT TAB(10)"*"TAB(70)"*"
820 PRINT TAB(10)"*"TAB(14)"Acquisizione su file di disco (Gould)";TAB(70)"*"
830 PRINT TAB(10)"*"TAB(70)"*"
840 PRINT TAB(10)"*****"
850 PRINT:PRINT TAB(10)"1 - Reset carrelli (X,Y=0)"
860 PRINT TAB(10)"2 - Posizionamento X"
870 PRINT TAB(10)"3 - Posizionamento Y"
880 PRINT TAB(10)"4 - Acquisizione"
890 PRINT TAB(10)"5 - Stop programma ATDSK Gould"
900 PRINT:PRINT TAB(10)"CTRL-G - Break della richiesta in corso su R6232/TTL"
910 PRINT:PRINT TAB(10)"CTRL-C - Stop programma"
920 PRINT:PRINT TAB(10);:INPUT K
930 IF K(1 OR K)5 THEN GOTO 920
940 ON K GOTO 950,1170,1290,1400,1100
950 REM *
960 REM * RESET
970 REM *
980 A$="R"
990 GOSUB 190:REM *** Richiesta di reset carrelli x-y atius
1000 GOSUB 310:REM *** aspetto K
1010 IF STK=0 AND A$="K" THEN PRINT "Risposta su reset ";A$ ; GOTO 950
1020 IF STK<>7 THEN PRINT"Errore di reset":STOP
1030 REM *
1040 REM *** Richiesta di abort
1050 REM *
1060 A$="H":REM *** richiesta di abort
1070 GOSUB 190:GOSUB 310 :IF A$="K" THEN GOTO 850
1080 PRINT "Errore su risposta a H":STOP
1090 REM *
1100 REM *** Disattivazione prog. ATDSK Gould
1110 REM *
1120 A$="STOP"
1130 GOSUB 180 : REM *** manda "STOP"
1140 GOSUB 300 : REM *** Aspetta "K"
1150 IF A$="K" THEN PRINT"ATDSK Gould disattivo":GOTO 850
1160 PRINT"Errore su richiesta di stop a ATDSK":GOTO 850
1170 REM *
1180 REM *** Posizionamento X
1190 REM *
1200 INPUT"Coordinata X ";X

```

```

1210 N1=INT(X/256):N2=X-(N1*256)
1220 A#="X"+CHR$(N2)+CHR$(N1)
1230 GOSUB 190:REM *** richiesta posiz. X
1240 GOSUB 310:REM *** assetto K/E
1250 IF ST%=0 AND A#="K" THEN GOTO 850
1260 IF ST%>7 THEN PRINT"Errore posiz. X":STOP
1270 PRINT"CTRL-G":STOP
1280 REM *
1290 REM *** Posizionamento Y
1300 REM *
1310 INPUT"Coordinata Y ";Y
1320 Y=Y*5
1330 N1=INT(Y/256):N2=Y-(N1*256)
1340 A#="Y"+CHR$(N2)+CHR$(N1)
1350 GOSUB 190:REM *** richiesta posiz. Y
1360 GOSUB 310:REM *** assetto K/E
1370 IF ST%=0 AND A#="K" THEN GOTO 850
1380 IF ST%>7 THEN PRINT"Errore posiz. Y":STOP
1390 PRINT"CTRL-G":STOP
1400 REM **
1410 REM **** COORDINATE DI SCANSIONE
1420 REM **
1430 REM
1440 INPUT"Scansione X (inizio, fine, passo)";XY(1),XY(4),XY(9)
1450 INPUT"Scansione Y (inizio, fine, passo)";XY(5),XY(8),XY(10)
1460 INPUT"Rimcorsa su X ";IS
1470 IF IS(1 OR IS)1000 THEN GOTO 1460
1480 IF XY(1)(15 OR XY(4))1000-IS OR XY(1))XY(4) THEN GOTO 1430
1490 IF XY(5)(1 OR XY(8))1000 OR XY(5))XY(8) THEN GOTO 1430
1500 IF XY(9)(1 OR XY(10)(1 THEN GOTO 1430
1510 XY(2)=XY(1)-IS
1520 XY(3)=XY(4)+IS
1530 XY(6)=XY(5) : XY(7)=XY(8)
1540 FOR I=5 TO 8:XY(I)=XY(I)*5:NEXT I:XY(10)=XY(10)*5
1550 NX=INT((XY(4)-XY(1))/XY(9))+1
1560 NY=INT((XY(8)-XY(5))/XY(10))+1
1570 IF NX>1024 OR NY>1024 THEN PRINT"Errore : NX o NY >1024":GOTO 1440
1580 HBY=INT(NX/256):LBY=NX-(HBY*256)
1590 A#="N"+CHR$(LBY)+CHR$(HBY)
1600 GOSUB 190

```



```

1610 GOSUB 310:REM *** Num. punti su una riga
1620 IF ST%=0 AND A$="K" THEN GOTO 1640
1630 IF ST%>7 THEN PRINT "Errore su NX":STOP
1640 FOR I=1 TO 10
1650 HEY=INT(XY(I)/256):LBY=XY(I)-(HEY*256)
1660 A#=CHR$(I+48)+CHR$(LBY)+CHR$(HEY)
1670 IF I=9 THEN A#="x"+CHR$(LBY)+CHR$(HEY)
1680 IF I=10 THEN A#="y"+CHR$(LBY)+CHR$(HEY)
1690 GOSUB 190
1700 GOSUB 310
1710 IF ST%<0 OR A$(">") THEN PRINT "Errore su ";I;" ST=";STX:STOP
1720 PRINT"OK ";I;XY(I)
1730 NEXT I
1740 INPUT "Attivazione ATDSK(Gould) ASI/NOU :";SVP$
1750 IF SVP$="NO" OR SVP$="no" THEN GOTO 1850
1760 IF SVP$(">")"SI" AND SVP$("<")"si" THEN GOTO 1740
1770 PRINT CHR$(8HC):REM *** cancella lo schermo
1780 FOR I=1 TO 10:PRINT CHR$(10);:NEXT I
1790 PRINT "Log-in Gould MINUS/MINUS : attendere"
1800 GOSUB 460: REM *** Log-in Gould
1810 PRINT CHR$(12):IF F>3 THEN PRINT "Collegamento fallito.":END
1820 A#="BATCH BATDSK"+CR$+"X"+CR$
1830 GOSUB 180:REM *** Attivazione in batch del programma ATDSK e log-off Gould
1840 GOSUB 390:REM *** Lettura log-off e attesa del T.O.
1850 A#="STAT":GOSUB 180:GOSUB 300:IF STX=7 THEN STOP:REM ***Rich. di stato in A$
1860 IF A$(">")"ATDSK" THEN PRINT "Errore":STOP
1870 PRINT "Programma ";A#;" attivo (batch BATDSK sotto TSM MPX-32 Gould)"
1880 INPUT "Nome file (max 6 caratteri) :";NOME$
1890 IF LEN(NOME$)<1 OR LEN(NOME$)>6 THEN 1880
1900 A#="SUB"+CHR$(50):GOSUB 180:REM *** crea file
1910 GOSUB 300:REM *** aspetto K
1920 IF A$(">")"K" THEN PRINT "Errore in creazione":STOP
1930 FOR I=LEN(NOME$)+1 TO 6:NOME$=NOME$+CHR$(32):NEXT I
1940 A#="NOME$":GOSUB 180:REM *** nome file
1950 B%(0)=4:B%(1)=NX:B%(2)=NY:REM *** lung. e num. record
1960 GOSUB 240
1970 GOSUB 300:REM *** leggi A$(CTRL-G)
1980 IF A#="K" THEN 2040:REM *** creazione riuscita
1990 PRINT "Errore in creazione file :";ASC(MID$(A$,1,1));ASC(MID$(A$,2,1))
2000 INPUT "riprovi (si/no) :";A$
2010 IF A#="SI" OR A#="si" THEN 1880

```

```

2020 IF A#("&NO" AND A#("&no" THEN 2000
2030 STOP
2040 REM
2050 FOR Y=1 TO NY
2060 B%(0)=NX+1
2070 A#="S":GOSUB 190
2080 GOSUB 370:REM *** lettura riga Y
2090 IF STAT%(0)<0 THEN PRINT"Errore su riga ";Y:STOP
2100 IF Y<NY THEN GOTO 2130
2110 GOSUB 310:REM *** aspetta K
2120 IF ST%(0) OR A#("&K" THEN PRINT"Errore fine scansione":STOP
2130 B%(0)=NX:IF (Y MOD 2)=0 THEN CALL INV%(B%(0),ST%)
2140 A#="SUB"+CHR$(53):GOSUB 180:REM *** Richiesta scrittura record
2150 GOSUB 300:REM *** ASPETTA K
2160 IF A#("&K" THEN PRINT"Errore":STOP
2170 C%(0)=4:C%(1)=NX:C%(2)=Y:CALL BRV%(CWA%,C%(0),STAT%(0)):REM *** lung. e num rec. da scrivere
2180 GOSUB 300:REM *** aspetto K
2190 IF A#("&K" THEN PRINT "Errore":STOP
2200 GOSUB 240:REM *** manda i valori
2210 GOSUB 300:REM *** aspetto K
2220 PRINT"SPEDITO RECORD ";Y
2230 IF A#("&K" THEN PRINT"Errore trasf. dati ":";ASC(MID$(A#,1,1));ASC(MID$(A#,2,1)):STOP
2240 NEXT Y
2250 A#="SUB"+CHR$(51)
2260 GOSUB 180:REM *** chiude file
2270 GOSUB 300:REM *** aspetto K
2280 IF A#("&K" THEN PRINT"Errore chiusura file ":";ASC(MID$(A#,1,1));ASC(MID$(A#,2,1)):STOP
2290 GOTO 800
2300 END

```

```

.0 REM *** PISA 3/4/87
20 REM *** AT/MTE : Acquisizione file su nastro magn. (Gould)
30 GOTO 690
40 REM*****
50 REM
60 REM *** carica driver da file DARTE.COM e inizializza canali A/B (RS232-C)
70 REM
80 REM*****
90 OPEN "R", #1, "DARTE.COM" : FIELD #1, 128 AS A$ : LA=&HC000
100 FOR I=1 TO 7 : GET #1, I : FOR K=1 TO 128 : POKE LA, ASC(MID$(A$, K, 1))
110 LA=LA+1: NEXT K: NEXT I: CLOSE #1
120 STX=-1: CALL INIZ$(CHCA%, STX): IF STX<>0 THEN PRINT"Err INIZ A:st="; STX: STOP
130 STX=-1: CALL INIZ$(CHCB%, STX): IF STX<>0 THEN PRINT"Err INIZ B:st="; STX: STOP
140 RETURN
150 REM*****
160 REM
170 CALL BRWS$(CWMA%, A$, STX) : RETURN : REM *** ch A write A$ (CTRL-G)
180 CALL BRWS$(CWMB%, A$, STX) : RETURN : REM *** ch B write A$ (CTRL-G)
190 REM
200 CALL BRVS$(CWMTOA%, A$, STX) : RETURN : REM *** ch A write A$ (T.O.)
210 CALL BRVS$(CWMTOB%, A$, STX) : RETURN : REM *** ch B write A$ (T.O.)
220 REM
230 CALL BRVW$(CWMWA%, B%(0), STAT%(0)) : RETURN : REM *** ch A write B% (CTRL-G)
240 CALL BRVW$(CWMWB%, B%(0), STAT%(0)) : RETURN : REM *** ch B write B% (CTRL-G)
250 REM
260 CALL BRVW$(CWMTOA%, B%(0), STAT%(0)) : RETURN : REM *** ch A write B% (T.O.)
270 CALL BRVW$(CWMTOB%, B%(0), STAT%(0)) : RETURN : REM *** ch B write B% (T.O.)
280 REM
290 A$=SPACE$(255) : CALL BRWS$(CWRMA%, A$, STX) : RETURN : REM *** ch A read A$ (CTRL-G)
300 A$=SPACE$(255) : CALL BRWS$(CWRMB%, A$, STX) : RETURN : REM *** ch B read A$ (CTRL-G)
310 REM
320 A$=SPACE$(255) : CALL BRVS$(CWRTOA%, A$, STX) : RETURN : REM *** ch A read A$ (T.O.)
330 A$=SPACE$(255) : CALL BRVS$(CWRTOB%, A$, STX) : RETURN : REM *** ch B read A$ (T.O.)
340 REM
350 CALL BRVW$(CWRMA%, B%(0), STAT%(0)) : RETURN : REM *** ch A read B% (CTRL-G)
360 CALL BRVW$(CWRMB%, B%(0), STAT%(0)) : RETURN : REM *** ch B read B% (CTRL-G)
370 REM
380 CALL BRVW$(CWRTOA%, B%(0), STAT%(0)) : RETURN : REM *** ch A read B% (T.O.)
390 CALL BRVW$(CWRTOB%, B%(0), STAT%(0)) : RETURN : REM *** ch B read B% (T.O.)
400 REM*****
410 REM

```

```

420 REM *** log-in GOULD
430 REM
440 REM*****
450 F=0
460 A$="STOP"+CR$+"X"+CR$+CR$+CR$
470 GOSUB 170
480 A$=CHR$(27):B$(0)=2000 : GOSUB 170
490 GOSUB 380:IF STAT%(1)=0 THEN 490
500 A$="MINUS"+CR$: GOSUB 170 : REM *** nome
510 GOSUB 380:IF STAT%(1)=0 THEN 510
520 GOSUB 170 : REM *** chieve
530 GOSUB 380:IF STAT%(1)=0 THEN 530
540 IF STAT%(1) < 7 THEN 600
550 IN=INT(STAT%(1)/2)-2
560 B$(IN)=4:IF (STAT%(1) MOD 2) < 0 THEN B$(IN)=5
570 B$="":CALL SPAK%(B$(IN),A$(0),ST%)
580 FOR I=A$(0)-3 TO A$(0):B$=B$+CHR$(A$(I)):NEXT I:REM ***b$=ultimi 4 caratt. di B$
590 IF B$="TSM)" THEN RETURN
600 F=F+1 : IF F <=3 THEN 460
610 RETURN
620 REM*****
630 REM
640 REM*****
650 REM
660 REM *** inizio programma
670 REM
680 REM*****
690 DIM B$(1000),STAT%(1),A$(1000)
700 BRWS%=&HC0EE : BRWV%=&HC059 : ST%=0 : CR$=CHR$(13) : B$(0)=2000
710 PAK%=&HC27E:SPAK%=&HC2AF:INV%=&HC2E2:INIZ%=&HC000:CHCA%=&H42:CHCB%=&H43
720 CWRAX%=&H40:CWRB%=&H41:CWRWA%=&H140:CWRWB%=&H141
730 CWRTOA%=&H240:CWRTOB%=&H241:CWRTOA%=&H340:CWRTOB%=&H341
740 PRINT CHR$(12):INPUT"Caricamento driver ?":A$
750 IF A$="no" OR A$="NO" THEN GOTO 790
760 IF A$(">")"SI" AND A$(">")"si" THEN GOTO 740
770 GOSUB 90
780 REM PRINT "STATO INIZ":ST%:STOP
790 PRINT CHR$(12);TAB(10)"*****"
800 PRINT TAB(10)"*TAB(70)"*

```

```

810 PRINT TAB(10)*"TAB(14)"Acquisizione file su N.M. (Gould)";TAB(70)*"
820 PRINT TAB(10)*"TAB(70)*"
830 PRINT TAB(10)"*****"
840 PRINT:PRINT TAB(10)"1 - Reset carrelli (X,Y=0)"
850 PRINT TAB(10)"2 - Posizionamento X"
860 PRINT TAB(10)"3 - Posizionamento Y"
870 PRINT TAB(10)"4 - Acquisizione"
880 PRINT TAB(10)"5 - Stop programma ATMT Gould"
890 PRINT:PRINT TAB(10)"CTRL-G - Break della richiesta in corso su RS232/TTL"
900 PRINT:PRINT TAB(10)"CTRL-C - Stop programma"
910 PRINT:PRINT TAB(10);:INPUT K
920 IF K(1 OR K)5 THEN GOTO 910
930 ON K GOTO 940,1150,1270,1390,1090
940 REM *
950 REM * RESET
960 REM *
970 A$="q"
980 GOSUB 180:REM *** Richiesta di reset carrelli x-y atius
990 GOSUB 300:REM *** aspetto K
1000 IF STX=0 AND A$="K" THEN PRINT "Risposta su reset ";A$ : GOTO 840
1010 IF STX(>7 THEN PRINT"Errore di reset":STOP
1020 REM *
1030 REM *** Richiesta di abort
1040 REM *
1050 A$="H":REM *** richiesta di abort
1060 GOSUB 180:GOSUB 300 :IF A$="K" THEN GOTO 840
1070 PRINT "Errore su risposta a H":STOP
1080 REM *
1090 REM *** Disattivazione prog. ATMT Gould
1100 REM *
1110 A$="STOP"
1120 GOSUB 170 : REM *** manda "STOP"
1130 GOSUB 290 : REM *** Aspetta "K"
1140 IF A$="K" THEN PRINT"ATMT Gould disattivo":GOTO 840
1150 PRINT"Errore su richiesta di stop a ATMT":GOTO 840
1160 REM *
1170 REM *** Posizionamento X
1180 REM *
1190 INPUT"Coordinata X :";X
1200 N1=INT(X/256):N2=X-(N1*256)

```

```

1210 A#="X"+CHR$(NZ)+CHR$(N1)
1220 GOSUB 180:REM *** richiesta posiz. X
1230 GOSUB 300:REM *** aspetto K/E
1240 IF ST%=0 AND A#="K" THEN GOTO 840
1250 IF ST%>7 THEN PRINT"Errore posiz. X":STOP
1260 PRINT"CTRL-G":STOP
1270 REM *
1280 REM *** Posizionamento Y
1290 REM *
1300 INPUT"Coordinata Y :";Y
1310 Y=Y*5
1320 N1=INT(Y/256):N2=Y-(N1*256)
1330 A#="Y"+CHR$(NZ)+CHR$(N1)
1340 GOSUB 180:REM *** richiesta posiz. Y
1350 GOSUB 300:REM *** aspetto K/E
1360 IF ST%=0 AND A#="K" THEN GOTO 840
1370 IF ST%>7 THEN PRINT"Errore posiz. Y":STOP
1380 PRINT"CTRL-G":STOP
1390 REM **
1400 REM **** COORDINATE DI SCANSIONE
1410 REM **
1420 REM
1430 INPUT"Scansione X (inizio, fine, passo)";XY(1),XY(4),XY(9)
1440 INPUT"Scansione Y (inizio, fine, passo)";XY(5),XY(8),XY(10)
1450 INPUT"Rincorsa su X ";IS
1460 IF IS<1 OR IS>1000 THEN GOTO 1450
1470 IF XY(1)<IS OR XY(4)>1000-IS OR XY(1)>XY(4) THEN GOTO 1420
1480 IF XY(5)<1 OR XY(8)>1000 OR XY(5)>XY(8) THEN GOTO 1420
1490 IF XY(9)<1 OR XY(10)>1 THEN GOTO 1420
1500 XY(2)=XY(1)-IS
1510 XY(3)=XY(4)+IS
1520 XY(6)=XY(5) : XY(7)=XY(8)
1530 FOR I=5 TO 8:XY(I)=XY(1)*5:NEXT I:XY(10)=XY(10)*5
1540 NX=INT((XY(4)-XY(1))/XY(9))+1
1550 NY=INT((XY(8)-XY(5))/XY(10))+1
1560 IF NX>1024 OR NY>1024 THEN PRINT"Errore : NX o NY >1024":GOTO 1430
1570 HEY=INT(NX/256):LBY=NX-(HEY*256)
1580 A#="N"+CHR$(LBY)+CHR$(HEY)
1590 GOSUB 180
1600 GOSUB 300:REM *** Num. punti su una riga

```

```

1610 IF STX=0 AND A#="K" THEN GOTO 1630
1620 IF STX<>7 THEN PRINT"Errore su NX":STOP
1630 FOR I=1 TO 10
1640 HBY=INT(XY(I)/256):LBY=XY(I)-(HBY*256)
1650 A#=CHR$(I+48)+CHR$(LBY)+CHR$(HBY)
1660 IF I=9 THEN A#="x"+CHR$(LBY)+CHR$(HBY)
1670 IF I=10 THEN A#="y"+CHR$(LBY)+CHR$(HBY)
1680 GOSUB 180
1690 GOSUB 300
1700 IF STX<>0 OR A#("<")"K" THEN PRINT"Errore su ";I;" ST=";STX:STOP
1710 PRINT"OK ";I;XY(I)
1720 NEXT I
1730 INPUT"Attivazione ATMT (Gould) ASI/NDU :";SVB#
1740 IF SVB#="NO" OR SVB#="no" THEN GOTO 1840
1750 IF SVB#("<")"SI" AND SVB#("<")"s" THEN GOTO 1730
1760 PRINT CHR$(8)<":REM *** cancella lo schermo
1770 FOR I=1 TO 10:PRINT CHR$(10);NEXT I
1780 PRINT "Log-in Gould MINUS/MINUS : attendere"
1790 GOSUB 450: REM *** Log-in Gould
1800 PRINT CHR$(12):IF F>3 THEN PRINT"Collegamento fallito.":END
1810 A#="ZATCH BATMT"+CR#+"X"+CR#
1820 GOSUB 170:REM *** Attivazione in batch del programma ATMT e log-off Gould
1830 GOSUB 380:REM *** Lettura log-off e attesa del T.O.
1840 A#="STAT":GOSUB 170:GOSUB 290:IF STX=7 THEN STOP:REM ***Rich. di stato in A#
1850 IF A#("<")"ATMT" THEN PRINT"Errore":STOP
1860 PRINT"Programma ";A#;" attivo (batch BATMT sotto TSM MPX-32 Gould)"
1870 PRINT"Montare nastro magnetico su unita' 0 (Gould):print
1880 PRINT:PRINT"Premere return quando pronto"
1890 A#=INKEY#:IF A#=" " THEN GOTO 1890
1900 IF A#("<")CHR$(13) THEN GOTO 1870
1910 A#="SUB"+CHR$(100):GOSUB 170:REM *** rewind
1920 GOSUB 290:REM *** aspetta K
1930 IF A#("<")"K" THEN PRINT "Errore in rewind":STOP
1940 INPUT"Numero di files da saltare";NF
1950 IF NF<0 OR NF>255 THEN GOTO 1940
1960 A#="SUB"+CHR$(101):GOSUB 170: REM *** richiesta salto file
1970 GOSUB 290: REM ***aspetta K
1980 IF A#("<")"K" THEN PRINT"Errore su richiesta salto file":STOP
1990 A#=CHR$(NF):GOSUB 170: REM *** salta NF file
2000 GOSUB 290 : REM ** aspetta K

```

```

2010 IF A$(0)"K" THEN PRINT"Errore su NF":STOP
2020 REM
2030 FOR Y=1 TO NY
2040 B$(0)=NX+1
2050 A#="S":GOSUB 180
2060 GOSUB 360:REM *** lettura riga Y
2070 IF STAT%(0)<>0 THEN PRINT"Errore su riga ";Y:STOP
2080 IF Y<>NY THEN GOTO 2110
2090 GOSUB 300:REM *** aspetta K
2100 IF ST%(0) OR A$(0)"K" THEN PRINT"Errore fine scansione":STOP
2110 B$(0)=NX:IF (Y MOD 2)=0 THEN CALL INV%(B$(0),ST%)
2120 A#="SUB"+CHR$(106):GOSUB 170:REM *** Richiesta scrittura record
2130 GOSUB 250:REM *** ASPETTA K
2140 IF A$(0)"K" THEN PRINT"Errore":STOP
2150 C$(0)=4:CX(1)=NX:CX(2)=Y:CALL BRW%(CWMAX,C$(0),STAT%(0)):REM *** lung. e num rec. da scrivere
2160 GOSUB 290:REM *** aspetta K
2170 IF A$(0)"K" THEN PRINT "Errore":STOP
2180 GOSUB 230:REM *** manda i valori
2190 GOSUB 290:REM *** aspetta K
2200 PRINT"SPEDITO RECORD ";Y
2210 IF A$(0)"K" THEN PRINT"Errore trasf. dati ":STOP
2220 NEXT Y
2230 A#="SUB"+CHR$(108)
2240 GOSUB 170:REM *** chiude file con e.o.f.
2250 GOSUB 290:REM *** aspetta K
2260 IF A$(0)"K" THEN PRINT"Errore chiusura file ":STOP
2270 GOTO 840
2280 END

```



```

10 REM *** PISA 18/3/87
20 REM *** DGSVPE - Diagnostico ETIS-2 + uscita su SVP + stampa a TTY/LP
30 GOTO 690
40 REM*****
50 REM
60 REM *** carica driver da file DARTE.COM e inizializza canali A/B (RS/TTL)
70 REM
80 REM*****
90 OPEN "R", #1, "DARTE.COM" : FIELD #1, 128 AS A$ : LA=&HC000
100 FOR I=1 TO 7 : GET #1, I : FOR K=1 TO 128 : POKE LA, ASC(MID$(A$, K, 1))
110 LA=LA+1:NEXT K:NEXT I:CLOSE #1:CHCAX=&H42:CHCER=&H43
120 STX=-1:CALL INIZX(CHCAX, STX):IF STX(<)0 THEN PRINT"Err INIZ A:st=";STX:STOP
130 STX=-1:CALL INIZX(CHCER, STX):IF STX(<)0 THEN PRINT"Err INIZ B:st=";STX:STOP
140 RETURN
150 REM*****
160 REM
170 CALL BRWSX(CWMAX, A$, STX) : RETURN : REM *** ch A write A$ (CTRL-G) RS232
180 CALL BRWSX(CWMBX, A$, STX) : RETURN : REM *** ch B write A$ (CTRL-G) TTL
190 REM
200 CALL BRWSX(CWWT0AX, A$, STX) : RETURN : REM *** ch A write A$ (T.O.) RS232
210 CALL BRWSX(CWWT0BX, A$, STX) : RETURN : REM *** ch B write A$ (T.O.) TTL
220 REM
230 CALL BRVX(CWMAX, B%(0), STATX(0)) :RETURN : REM *** ch A write B% (CTRL-G) RS232
240 CALL BRVX(CWMBX, B%(0), STATX(0)) :RETURN : REM *** ch B write B% (CTRL-G) TTL
250 REM
260 CALL BRVX(CWWT0AX, B%(0), STATX(0)) : RETURN : REM *** ch A write B% (T.O.) RS232
270 CALL BRVX(CWWT0BX, B%(0), STATX(0)) : RETURN : REM *** ch B write B% (T.O.) TTL
280 REM
290 A$=SPACE$(255) : CALL BRWSX(CWRAX, A$, STX) : RETURN : REM *** ch A read A$ (CTRL-G) RS232
300 A$=SPACE$(255) : CALL BRWSX(CWRBX, A$, STX) : RETURN : REM *** ch B read A$ (CTRL-G) TTL
310 REM
320 A$=SPACE$(255) : CALL BRVX(CWRTOAX, A$, STX) : RETURN : REM *** ch A read A$ (T.O.) RS232
330 A$=SPACE$(255) : CALL BRVX(CWRTOBX, A$, STX) : RETURN : REM *** ch B read A$ (T.O.) TTL
340 REM
350 CALL BRVX(CWRAX, B%(0), STATX(0)) : RETURN : REM *** ch A read B% (CTRL-G) RS232
360 CALL BRVX(CWRBX, B%(0), STATX(0)) : RETURN : REM *** ch B read B% (CTRL-G) TTL
370 REM
380 CALL BRVX(CWRTOAX, B%(0), STATX(0)) : RETURN : REM *** ch A read B% (T.O.) RS232
390 CALL BRVX(CWRTOBX, B%(0), STATX(0)) : RETURN : REM *** ch B read B% (T.O.) TTL

```

```

400 REM*****
410 REM
420 REM *** log-in GOULD
430 REM
440 REM*****
450 F=0:WU=27:REM *** wakeup char.
460 A$="STOP"+CR$+"X"+CR$+"X"+CR$+CR$+CR$
470 GOSUB 170
480 A$=CHR$(WU):B$(0)=2000:GOSUB 170
490 GOSUB 380:IF STAT%(1)=0 THEN 490
500 A$="MINUS"+CR$:GOSUB 170:REM *** nome
510 GOSUB 390:IF STAT%(1)=0 THEN 510
520 GOSUB 170:REM *** chieve
530 GOSUB 380:IF STAT%(1)=0 THEN 530
540 IF STAT%(1) < 7 THEN 600
550 IN=INT(STAT%(1)/2)-2
560 B$(IN)=4:IF (STAT%(1) MOD 2) < 0 THEN B$(IN)=5
570 B$="":CALL SPAK%(B$(IN),A$(0),ST%)
580 FOR I=A$(0)-3 TO A$(0):B$=B$+CHR$(A$(I)):NEXT I:REM *** B$=ultimi 4 caratt. di B%
590 IF B$="TSM)" THEN RETURN
600 F=F+1:IF F <=3 THEN 460
610 RETURN
620 REM*****
630 REM
640 REM*****
650 REM
660 REM *** inizio programma
670 REM
680 REM*****
690 DIM B$(1000),STAT%(1),A$(1000)
700 BRWS%=&HC0EE:BRWV%=&HC059:ST%=0:CR$=CHR$(13):B$(0)=2000
710 PAK%=&HC27E:SPAK%=&HC2AF:INV%=&HC2E2:INIZ%=&HC000
715 CHCA%=&H42:CHCB%=&H43
720 CWRA%=&H40:CWRB%=&H41:CWVA%=&H140:CWVB%=&H141
730 CWRTOA%=&H240:CWRTOB%=&H241:CWWTDA%=&H340:CWWTDB%=&H341
740 PRINT CHR$(12):INPUT"Caricamento driver ?";A$
750 IF A$="no" OR A$="NO" THEN CALL INIZ%(CHCB%,ST%):GOTO 790
760 IF A$(">"SI" AND A$("<"SI" THEN GOTO 740
770 GOSUB 90
780 REM PRINT "STATO INIZ";ST%:STOP
790 PRINT CHR$(12):PRINT TAB(10)"*****"
800 PRINT TAB(10)"*TAB(70)"*

```

```

810 PRINT TAB(10)"*"TAB(28)"Diagnostico ETIS-2"TAB(70)"*"
820 PRINT TAB(10)"*"TAB(70)"*"
830 PRINT TAB(10)"*****"
840 PRINT:PRINT TAB(10)"1 - Reset (X,Y=0)"
850 PRINT TAB(10)"2 - Richiesta disattivazione ATSP/GOULD"
860 PRINT TAB(10)"3 - Posizionamento X"
870 PRINT TAB(10)"4 - Posizionamento Y"
880 PRINT TAB(10)"5 - Invio coordinate di scansione"
890 PRINT TAB(10)"6 - Start scansione"
900 PRINT TAB(10)"7 - Lettura dati in loop"
910 PRINT:PRINT TAB(10)"CTRL-G - Break della richiesta in corso su RS232/TTL"
920 PRINT:PRINT TAB(10)"CTRL-C - Stop programma"
930 PRINT:PRINT TAB(10)":INPUT K
940 IF K<1 OR K>7 THEN GOTO 930
950 ON K GOTO 960,1100,1180,1290,1420,1740,2860
960 REM *
970 REM * RESET
980 REM *
990 A#="R"
1000 GOSUB 180:REM *** Richiesta di reset carrelli x-y atius
1010 GOSUB 300:REM *** aspetto K/E
1020 IF STX=0 AND A#="K" THEN PRINT "Risposta su reset ":"A# : GOTO 840
1030 IF STX<>7 THEN PRINT"Errore di reset":STOP
1040 REM *
1050 REM *** Richiesta di abort
1060 REM *
1070 A#="H":REM *** richiesta di abort
1080 GOSUB 180:GOSUB 300 :IF A#="K" THEN GOTO 840
1090 PRINT "Errore su risposta a H":STOP
1100 REM *
1110 REM *** Disattivazione prog. ATSPV
1120 REM *
1130 A#="STOP"
1140 GOSUB 170 : REM *** manda "STOP"
1150 GOSUB 290 : REM *** Aspetta "K"
1160 IF A#="K" THEN PRINT"K da ATSPV":GOTO 840
1170 PRINT"Errore su richiesta di stop a ATSPV":GOTO 840
1180 REM *
1190 REM *** Posizionamento X
1200 REM *

```

```

1210 INPUT"Coordinata X ":";X
1220 N1=INT(X/256):N2=X-(N1*256)
1230 A#="X"+CHR$(N2)+CHR$(N1)
1240 GOSUB 180:REM *** richiesta posiz. X
1250 GOSUB 300:REM *** aspetto K/E
1260 IF STX=0 AND A#="K" THEN GOTO 840
1270 IF STX<>7 THEN PRINT"Errore posiz. X":STOP
1280 PRINT"CTRL-G":STOP
1290 REM *
1300 REM *** Posizionamento Y
1310 REM *
1320 INPUT"Coordinata Y ":";Y
1330 Y=Y*5
1340 N1=INT(Y/256):N2=Y-(N1*256)
1350 A#="Y"+CHR$(N2)+CHR$(N1)
1360 GOSUB 180:REM *** richiesta posiz. Y
1370 GOSUB 300:REM *** aspetto K/E
1380 IF STY=0 AND A#="K" THEN GOTO 840
1390 IF STY<>7 THEN PRINT"Errore posiz. Y":STOP
1400 PRINT"CTRL-G":STOP
1410 REM **
1420 REM *** COORDINATE DI SCANSIONE
1430 REM **
1440 INPUT"Coord. da DATA";A#
1450 IF A#="SI" OR A#="SI" THEN RESTORE:FOR J=1 TO 10:READ XY(J):NEXT J:GOTO 1580
1460 IF A#<>"NO" AND A#<>"no" THEN GOTO 1440
1470 INPUT"Scansione X (inizio, fine)";XY(1),XY(4)
1480 INPUT"Scansione Y (inizio, fine)";XY(5),XY(8)
1490 INPUT"Passo scansione X,Y";XY(9),XY(10)
1500 INPUT"Rincorsa su X ";IS
1510 IF IS<1 OR IS>1000 THEN GOTO 1500
1520 IF XY(1)<15 OR XY(4)>1000-15 OR XY(1)>>XY(4) THEN GOTO 1440
1530 IF XY(5)<1 OR XY(8)>1000 OR XY(5)>>XY(8) THEN GOTO 1440
1540 IF XY(9)<1 OR XY(10)<1 THEN GOTO 1440
1550 XY(2)=XY(1)-15
1560 XY(3)=XY(4)+15
1570 XY(6)=XY(5) : XY(7)=XY(8)
1580 FOR I=5 TO 8:XY(I)=XY(I)*5:NEXT I:XY(10)=XY(10)*5
1590 FOR I=1 TO 10
1600 HBY=INT(XY(I)/256):LBY=XY(I)-(HBY*256)
1610 A#=CHR$(I+48)+CHR$(LBY)+CHR$(HBY)

```

```

620 IF I=9 THEN A#="X"+CHR$(LBY)+CHR$(HBY)
1630 IF I=10 THEN A#="Y"+CHR$(LBY)+CHR$(HBY)
1640 GOSUB 180
1650 GOSUB 300
1660 IF ST<>0 OR A#(">K" THEN GOTO 1690
1670 PRINT"OK ";I;XY(I)
1680 NEXT I:GOTO 840
1690 PRINT "Errore su ";I;" ST = ";ST%;" ricevuto ";A#
1700 STOP
1710 REM **
1720 REM ** Start scansione
1730 REM **
1740 NX=INT((XY(4)-XY(1))/XY(3))+1
1750 NY=INT((XY(8)-XY(5))/XY(10))+1
1760 HBY=INT(NX/256):LBY=NX-(HBY*256)
1770 A#="N"+CHR$(LBY)+CHR$(HBY)
1780 GOSUB 180
1790 GOSUB 300
1800 IF ST=0 AND A#="K" THEN GOTO 1830
1810 IF ST<>7 THEN PRINT"Errore su NX":STOP
1820 PRINT "CTRL-G":STOP
1830 INPUT"Uscita dati : 0-menu 1-svp 2-stampante 3-ITY";IU
1840 IF IU=3 THEN GOTO 2990
1850 IF IU=2 THEN GOTO 2610
1860 IF IU=0 THEN GOTO 840
1870 IF IU<>1 THEN GOTO 1830
1880 IF NX>512 OR NY>512 THEN PRINT"NX o NY >512":GOTO 1830
1890 INPUT"Attivazione ATSVp(Gould) ASI/NOU :";SVP#
1900 IF SVP#="ND" OR SVP#="no" THEN GOTO 2000
1910 IF SVP#(">SI" AND SVP#(">si" THEN GOTO 1890
1920 PRINT CHR$(&HC):REM *** cancella lo schermo
1930 FOR I=1 TO 10:PRINT CHR$(10):NEXT I
1940 PRINT "Log-in Gould MINUS/MINUS : attendere"
1950 GOSUB 450:REM *** Log-in Gould
1960 PRINT CHR$(12):IF F>3 THEN PRINT"Collegamento fallito.":END
1970 A#="BATCH BATSVP"+CR#+X"+CR#
1980 GOSUB 170:REM *** Attivazione in batch del programma ATSVp e log-off Gould
1990 GOSUB 380:REM *** Lettura log-off e attesa del T.O.

```

```

2000 A#="STAT":GOSUB 170:GOSUB 290:IF STX=7 THEN STOP:REM ***Rich. di stato in A#
2010 IF A#(">")ATSV" THEN PRINT"Errore":STOP
2020 PRINT"programma ";A#;" attivo (batch BATSVP sotto TSM MPX-32 Gould)"
2030 INPUT"Reset del sistema SVP-2000 (si/no) ";A#
2040 IF A#="NO" OR A#="no" THEN 2030
2050 IF A#(">")SI" AND A#(">")si" THEN 2030
2060 A#="SUB"+CHR$(1):GOSUB 170:REM *** Richiesta di reset su SVP2000
2070 GOSUB 290:REM *** Lettura in A#, se=K ok
2080 IF A#(">")K" THEN PRINT "Errore":STOP
2090 PRINT"Configurazione SVP-2000"
2100 INPUT"Memoria (1-3), Banco (1-4) ";M,B
2110 IF M(1 OR M)3 OR B(1 OR B)4 THEN 2100
2120 A#="SUB"+CHR$(2):GOSUB 170:REM *** BANCO
2130 GOSUB 290:REM *** Aspetto K
2140 IF A#(">")X" THEN PRINT"Errore":STOP
2150 A#="CHR$(B)+CHR$(M):GOSUB 170:REM *** B=banco
2160 GOSUB 290:REM ***Aspetto K
2170 IF A#(">")K" THEN PRINT "Errore":STOP
2180 A#="SUB"+CHR$(3):GOSUB 170:REM *** INTM
2190 GOSUB 290
2200 IF A#(">")K" THEN PRINT"Errore":STOP
2210 A#="CHR$(M):GOSUB 170:REM *** M=memoria
2220 GOSUB 290:REM *** Aspetto K
2230 IF A#(">")K" THEN PRINT"Errore":STOP
2240 INPUT"OK ";A#
2250 IF A#="NO" OR A#="no" THEN 2100
2260 IF A#(">")SI" AND A#(">")si" THEN 2240
2270 INPUT"Cancelazione memoria (si/no) ";A#
2280 IF A#="NO" OR A#="no" THEN 2350
2290 IF A#(">")SI" AND A#(">")si" THEN 2270
2300 A#="SUB"+CHR$(5):GOSUB 170:REM *** MEMC
2310 GOSUB 290:REM *** Aspetto K
2320 IF A#(">")K" THEN PRINT"Errore":STOP
2330 A#="CHR$(M):GOSUB 170:REM *** M=memoria
2340 GOSUB 290:REM *** Aspetto K
2350 IF A#(">")K" THEN PRINT"Errore":STOP
2360 INPUT"Coordinate iniziali SVP (X,Y) ";IX,IY
2370 IF IX+NX-1>512 OR IY+NY-1>512 THEN GOTO 2360
2380 A#="SUB"+CHR$(14)
2390 GOSUB 170:REM SCRIVE A#
2400 GOSUB 290:REM *** Aspetto K

```

```

2410 IF A#(>)"K" THEN PRINT"Errore":STOP
2420 B%(0)=9:B%(1)=IY:B%(2)=IY+NY-1:B%(3)=IX:B%(4)=IX+NX-1:B%(5)=M
2430 GOSUB 230: REM SCRIVE B% (PARAMETRI AREA)
2440 GOSUB 290: REM ASPETTO K
2450 IF A#(>)"K" THEN PRINT"Errore":STOP
2460 B%(0)=NX+1
2470 FOR RY=1 TO NY
2480 B%(0)=NX+1
2490 A#="S":PRINT"Start riga RY = ";RY:GOSUB 180
2500 GOSUB 360:REM *** Lettura riga
2510 IF STAT%(0)<>0 THEN PRINT"Errore su riga RY = ";RY:STOP
2520 IF RY(>NY THEN GOTO 2550
2530 GOSUB 300:REM *** aspetta K
2540 IF ST%(0)OR A#(>)"K" THEN PRINT"Err. fine scansione":STOP
2550 B%(0)=NX:IF (RY MOD 2)=0 THEN CALL INV%(B%(0),ST%)
2560 GOSUB 230: REM Scrittura riga su SVP
2570 GOSUB 290:REM *** Aspetto K
2580 IF A#(>)"K" THEN PRINT"Errore scrittura SVP":STOP
2590 NEXT RY
2600 GOTO 840
2610 LPRINT:LPRINT:COORDINATE DI SCANSIONE (INIZIO, FINE, PASSO) : X =";XY(1);XY(4);XY(9);" Y*5 =";
2620 FOR RY=1 TO NY
2630 B%(0)=NX+1:A#="S":PRINT "Start riga RY=";RY;NX;" valori":GOSUB 180
2640 GOSUB 360:REM lettura riga RY
2650 IF STAT%(0)<>0 THEN PRINT"Errore su riga ";RY:STOP
2660 LPRINT:LPRINT"-----":LPRINT "Grafico riga n. ";RY:LPRINT"-----"
2670 K1=1:K2=INT(STAT%(1)/2):K3=1:KN=1
2680 FOR K=K1 TO K2 STEP K3
2690 LN=B%(K) AND 255: HN=INT(B%(K)/256) AND 255
2700 L1=LN: IF LN>100 THEN L1=100
2710 H1=HN: IF HN>100 THEN H1=100
2720 LPRINT KN;LN;TAB(12);:FOR J=1 TO L1:LPRINT"*";:NEXT J:LPRINT
2730 IF KN+1<=NX THEN LPRINT KN+1;HN;TAB(12);:FOR J=1 TO H1:LPRINT"*";:NEXT J:LPRINT
2740 KN=KN+2:K4=K1:K1=K2:K2=K4:K3=-K3
2750 NEXT K
2760 REM** INPUT"premi return per continuare";A#
2770 NEXT RY
780 GOSUB 300:REM *** ASPETTA K
2790 IF ST%=0 AND A#="K" THEN PRINT"FINE SCANSIONE OK":GOTO 840

```

```

2800 PRINT "Errore su fine scansione"
2810 DATA 200,100,600,500,100,50,300,150,7,7
2820 STOP
2830 REM ***
2840 REM *** Lettura dati in 1000
2850 REM ***
2860 B%(0)=2 : GOSUB 360
2870 IF STAT%(0) > 0 THEN PRINT "ERRORE IN LETTURA":STOP
2880 REM PRINT "VALORI LETTI =";STAT%(1)
2890 PRINT B%(1) AND &HFF;INT(B%(1)/256) AND &HFF;
2900 GOTO 2860
2910 KK=1
2920 FOR K=1 TO INT(STAT%(1)/2)
2930 LN=B%(K) AND 255 : HN=INT(B%(K)/256) AND 255
2940 PRINT KK;LN;TAB(12);:FOR J=1 TO LN:PRINT "*";:NEXT J:PRINT
2950 PRINT KK+1;HN;TAB(12);:FOR J=1 TO HN:PRINT "*";:NEXT J:PRINT
2960 KK=KK+2
2970 NEXT K
2980 GOTO 2860
2990 PRINT"Coord. (inizio, fine, passo) :X ";XY(1);XY(4);XY(9)
3000 PRINT"
Y ";XY(5);XY(8);XY(10)
3010 FOR RY=1 TO NY
3020 FOR I=1 TO NX:B%(I)=255:NEXT I
3030 B%(0)=NX+1:A#="S":PRINT"Start riga RY ";RY;NX;" valori richiesti"
3040 GOSUB 180:GOSUB 360: REM ***** lettura riga RY
3050 IF STAT%(0) > 0 THEN PRINT"ERRORE DI LETTURA":STOP
3060 B%(0)=B%(0)-1:REM *** ELIMINA IL CR(13 O ODH) DAL VETTORE DEI DATI
3070 IF INT(RY/2)=RY/2 THEN CALL INV%(B%(0),ST%)
3080 CALL SPAK%(B%(0),A%(0),ST%)
3090 FOR I=1 TO A%(0) STEP 10
3100 PRINT I;TAB(7) "-----";I2=I+9:IF I2>A%(0)-1 THEN I2=A%(0)
3110 FOR I1=I TO I2:PRINT A%(I1);:NEXT I1:PRINT
3120 NEXT I
3130 PRINT:INPUT"PREMI RETURN PER CONTINUARE";H
3140 NEXT RY
3150 GOSUB 300:IF ST%=0 AND A#="K" THEN PRINT"fine scansione OK":GOTO 840
3160 PRINT"Errore fine scansione":STOP

```



```

10 REM *** PISA 31/3/87
20 REM *** MPXE ***
30 GOTO 160
40 REM*****
50 REM
60 REM *** carica driver da file DARTE.COM e inizializza canale A (RS232)
70 REM
80 REM*****
90 OPEN "R",#1,"DARTE.COM" : FIELD #1,128 AS A$ : LA=&HC000
100 FOR I=1 TO 7 : GET #1,I
110 FOR K=1 TO 128 : POKE LA,ASC(MID$(A$,K,1)) : LA=LA+1 :NEXT K
120 NEXT I : CLOSE #1 : CALL INIZ(CAN#,ST%):IF ST%=0 THEN RETURN
130 PRINT:PRINT*** Errore di inizializzazione driver ***:STOP
140 REM*****
150 REM
160 REM*****
170 REM
180 REM *** inizio programma
190 REM
200 REM*****
210 REM
220 INIZ%=&HC000 : CAN#=&H42 :ST%=-1
230 PRINT CHR$(12):PRINT TAB(23)"*****"
240 PRINT TAB(23)"*"
250 PRINT TAB(23)"* Collegamento Minus-Gould *"
260 PRINT TAB(23)"* MPX-32 System *"
270 PRINT TAB(23)"*"
280 PRINT TAB(23)"*****"
290 PRINT:PRINT"Configurazione Minus linea RS232-C : "
300 PRINT:PRINT TAB(10)"Half/duplex"
310 PRINT TAB(10)"Baud 1200"
320 PRINT TAB(10)"Tx/Rx 8 bit"
330 PRINT TAB(10)"parita' no"
340 PRINT TAB(10)"Stop/bit 1"
350 PRINT:PRINT"Assicurarsi che la configurazione Gould linea RS232-C sia compatibile"
360 PRINT:PRINT"Return per continuare"
370 PRINT"Ctrl-G per tornare al menu'":PRINT"Ctrl-C per stop programma"
380 A$=INKEY$:IF A$="" THEN 380
390 IF ASC(A$)=13 THEN 420
400 IF ASC(A$)=7 THEN 470
410 GOTO 380

```

```
420 GOSUB 90 : REM *** carica e inizializza driver
430 POKE &HC31D,&H60:CALL INIZ%(CAN%,ST%):REM *** configura per 1200 baud
440 PRINT:PRINT"Ring in for service"
450 EMU%=&HC23E
460 CALL EMU%
480 END
```

