

```

1: *
2: * TMS 9914 Auxiliary commands
3: *
0000 4: LSWRST EQU $00 Clear software reset
0080 5: HSWRST EQU $80 Software reset
0001 6: LDACR EQU $01 Release ACDS holdoff
0081 7: HDACR EQU $81 Release ACDS holdoff (ADD. Pass)
0002 8: RHDF EQU $02 Release ready for data holdoff.
0003 9: LHDF EQU $03 Ready for data holdoff
0083 10: HHDF EQU $83 Clear ready for data holdoff
0004 11: LHDFE EQU $04 Holdoff on end
0084 12: HHDFE EQU $84 Holdoff on end
0005 13: NBFA EQU $05 Set new byte available false
0006 14: LFGET EQU $06 Force group execute pulse
0086 15: HFGET EQU $86 Force group execute level
0007 16: LRTL EQU $07 Return to local (temporary)
0087 17: HRTL EQU $87 " " " (held untill LRTL sent)
0008 18: FEQI EQU $08 Force end or identify
0009 19: LLON EQU $09 Clear listen only
0089 20: HLOn EQU $89 Set listen only
000A 21: LTON EQU $0A Clear talk only
008A 22: HTON EQU $8A Set talk only
000B 23: GTS EQU $0B Go to standby
000D 24: TCS EQU $0D Take control synchronously
000C 25: TCA EQU $0C Take control asynchronously
000E 26: LRPP EQU $0E Clear request parallel poll
008E 27: HRPP EQU $8E Set request parallel poll
000F 28: LSIC EQU $0F Clear interface clear
008F 29: HSIC EQU $8F Set interface clear
0010 30: LSRE EQU $10 Clear remote enable
0090 31: HSRE EQU $90 Send remote enable
0011 32: RQC EQU $11 Request control of bus
0092 33: RLC EQU $92 Release control of bus
0013 34: LDAI EQU $13 Release disable of interrupts
0093 35: HDAI EQU $93 Disable all interrupts
0014 36: PTS EQU $14 Pass through next secondary ADD
0015 37: LSTDLEQU $15 Set T1 delay 10 clock cycles
0095 38: HSTDLEQU $95 Set T1 delay 6 clock cycles
0016 39: LSHDW EQU $16 Release shadow handshake
0096 40: HSHDW EQU $96 Set shadow handshake
41: *
42: *
43: * TMS 9914 register addresses
44: *
EE00 45: INTSTO EQU $EE00 Path 0 Interrupt status (read)
EE00 46: INTMKO EQU $EE00 Path 0 Interrupt mask (write)
EE01 47: INTST1 EQU $EE01 Path 1 Interrupt status (read)
EE01 48: INTMK1 EQU $EE01 Path 1 Interrupt mask (write)
EE02 49: ADDSTS EQU $EE02 Address status (read only)
EE03 50: BUSSTS EQU $EE03 Bus status register (read)
EE03 51: AUXCMD EQU $EE03 Auxillary command (write)
EE08 52: ADDSWT EQU $EE08 Device address switches
EE04 53: ADDRST EQU $EE04 Device address store
EE05 54: SERPOL EQU $EE05 Serial poll (write only)
EE06 55: CMDPAS EQU $EE06 Command pass through (read)
EE06 56: PARPOL EQU $EE06 Parallel poll (write only)
EE07 57: DATAIN EQU $EE07 Data recieved (read)

```

```

EE07      58: DATOUT EQU  $EE07      Data transmitted (write)
          59: *
EA02      60: ACIAS EQU  $EA02
EA03      61: ACIAD EQU  $EA03
          62: *
          63: *
          64: * Interface control commands
          65: *
0014      66: DCL   EQU  $14         Device clear
0018      67: SPE   EQU  $18         Serial poll enable
0019      68: SPD   EQU  $19         Serial poll disable
003F      69: UNL   EQU  $3F         Unlisten all devices
005F      70: UNT   EQU  $5F         Untalk
          71: *
FB94      72: PDATA1 EQU $FB94      MONITOR PRINT STRING
D283      73: WSTART EQU $D283     DISK WARM START
FB64      74: BADDR EQU  $FB64     GET BOTH LISTEN AND TALK
          75: *
          76: *
          77:      OPT  NOG,NOS
          78: *
1000      79:      ORG  $1000
          80: *
1000 20 49 81: START BRA  SOFTRT     GO ROUND ALL SCRATCH
          82: *
1002      83: DELT1 RMB  1
1003      84: DELT2 RMB  1
1004      85: DELT3 RMB  1
1005      86: LISTAD RMB  1         Listen address for unit
1006      87: TALKAD RMB  1         Talk address for unit
1007      88: TEMPE RMB  2
          89: *
1009 0A   90: INTRO FCB  $A,$D
100B 20   91:      FCC  / ENTER LISTEN THEN TALK ADDRESS /
102B 00   92:      FCB  0,0,4
          93: *
102E 0A   94: ERROR FCB  $A,$D
1030 48   95:      FCC  /HPIB PORT SELECT ERROR !/
1048 00   96:      FCB  0,0,4

```

```

98: *
99: * Software reset for the TMS 9914A
100: * AND RESET THE ACIA WITH RTS OFF (0)
101: * Set own address and clear interface
102: *
104B CE 1009 103: SOFTRT LDX £INTRO POINT AT START MESSAGE
104E BD FB94 104: JSR PDATA1 GO PRINT IT
1051 BD FB64 105: JSR BADDR GO BUILD THE ADDRESS
1054 FF 1005 106: STX LISTAD SAVE BOTH LISTEN AND TALK
1057 B6 80 107: LDA A £HSWRST Pickup Reset hardware
1059 B7 EE03 108: STA A AUXCMD Put out to 9914 command reg
105C B6 EE08 109: LDA A ADDSWT Pick up DIL switch data
105F B7 EE04 110: STA A ADDRES Put in 9914 address reg
1062 B6 8F 111: LDA A £HSIC Set interface clear
1064 B7 EE03 112: STA A AUXCMD Put out to 9914 command reg
1067 B6 00 113: LDA A £LSWRST Remove software reset
1069 B7 EE03 114: STA A AUXCMD Put out to 9914A (4 command reg
115: *
116: * SET UP SECOND ACIA FOR 8 BIT 1 STOP
117: *
106C B6 03 118: LDA A £#03 LOAD ACIA RESET
106E B7 EA02 119: STA A ACIAS PUT INTO ACIA
1071 B6 15 120: LDA A £#15 LOAD 8 BIT 1 STOP
1073 B7 EA02 121: STA A ACIAS PUT IN ACIA STATUS
122: *
123: * Delay for 500 microseconds for 9914
124: * to settle
125: *
1076 B6 03 126: LDA A £#03
1078 C6 FF 127: LDA B £#FF
107A 4A 128: DLOOP DEC A
107B 26 FD 129: BNE DLOOP
107D 5A 130: DEC B
107E 26 FA 131: BNE DLOOP Delay complete
132: *
133: * Remove interface clear and send
134: * remove remote enable
135: *
1080 B6 0F 136: LDA A £LSIC Remove interface clear
1082 B7 EE03 137: STA A AUXCMD Put out to 9914 command reg
1085 B6 90 138: LDA A £HSRE Load remote enable
1087 B7 EE03 139: STA A AUXCMD Put out to 9914 command reg
140: *
141: * Setup now complete self in controller
142: * active state having sent IFC, remote
143: * enable has been sent to all devices.
144: *
145: *
146: * Address unit to a listen device and
147: * wait for acknowledge
148: *
108A B6 EE00 149: LDA A INTSTO Clear out interrupt reg
108D F6 1005 150: LDA B LISTAD Load listen address
1090 F7 EE07 151: STA B DATOUT Put out to the unit

```



```
1093 BD 11D1 152:      JSR  AKWAIT      Wait for data accepted
153: *
154: * Set TMS 9914 to talk only
155: *
1096 F6 EE08 156:      LDA  B ADDSWT    Get 9914 address
1099 C8 40 157:      EOR  B f$40     Add in talk bit
109B F7 EE07 158:      STA  B DATOUT   Put out to the unit
109E BD 11D1 159:      JSR  AKWAIT     Wait for data accepted
10A1 C6 8A 160:      LDA  B fHTON   Load self to talk
10A3 F7 EE03 161:      STA  B AUXCMD  Put out 9914 command reg
10A6 C6 0B 162:      LDA  B fGTS   Release ATN line
10A8 F7 EE03 163:      STA  B AUXCMD  Put out 9914 command reg
10AB BD 11D1 164:      JSR  AKWAIT     Wait for data accepted
10AE 39 165:      RTS          RETURN
```

```

167: *
168: * WAIT FOR SERVICE REQUEST THEN READ UNIT
169: *
170: *
10AF B6 EE01 171: SERVRO LDA A INTST1      LOAD INTERRUPT REG 1
10B2 B4 02   172:          AND A £#02      MASK OUT ALL BUT SRQ BIT
10B4 27 F9   173:          BEQ  SERVRO      GO AND WAIT FOR SERVICE
174: *
175: * SERVICE REQUESTED GO AND READ, NO POLL
176: * AS WE ONLY HAVE ONE ON THE BUS
177: *
178: * SET UNIT TO TALK SELF LISTEN
179: *
10B6 B6 0C   180:          LDA A £TCA      LOAD TAKE CONTROL OF BUS
10B8 B7 EE03 181:          STA A AUXCMD     PUT INTO THE COMMAND REG:
10BB BD 11D1 182:          JSR  AKWAIT     WAIT FOR DATA ACCEPTED
10BE B6 3F   183:          LDA A £UNL      LOAD UNLISTEN COMMAND
10C0 B7 EE07 184:          STA A DATOUT     PUT OUT TO ALL DEVICES
10C3 BD 11D1 185:          JSR  AKWAIT     WAIT FOR DATA READY
10C6 B6 0A   186:          LDA A £LTON     REMOVE SELF TALK ONLY
10C8 B7 EE03 187:          STA A AUXCMD     PUT INTO THE 9914A
10CB B6 1006 188:          LDA A TALKAD    GET UNIT TALK ADDRESS
10CE B7 EE07 189:          STA A DATOUT     PUT OUT TO THE UNIT
10D1 BD 11D1 190:          JSR  AKWAIT     WAIT FOR ACKNOWLEDGE
10D4 B6 EE08 191:          LDA A ADDSWT    LOAD SELF ADDRESS SWITCHS
10D7 B8 20   192:          EOR A £#20     MAKE INTO A READ ADDRESS
10D9 B7 EE07 193:          STA A DATOUT     PUT OUT ADDRESS OVER BUS
10DC BD 11D1 194:          JSR  AKWAIT     WAIT FOR ACKNOWLEDGE
10DF B6 B9   195:          LDA A £HLON    LOAD SELF TO LISTEN
10E1 B7 EE03 196:          STA A AUXCMD     PUT INTO THE 9914A
197: *
198: * SEND SERIAL POLL ENABLE
199: *
10E4 B6 18   200:          LDA A £SPE     LOAD SERIAL POLL ENABLE
10E6 B7 EE07 201:          STA A DATOUT     PUT OUT TO UNITS
10E9 BD 11D1 202:          JSR  AKWAIT     WAIT FOR THE OK
10EC B6 0B   203:          LDA A £GTS     LOAD GO TO STANDBY
10EE B7 EE03 204:          STA A AUXCMD     PUT IN 9914A
10F1 F6 EE07 205:          LDA B DATAIN   READ DATA LINES (SERVICE REQU:)
10F4 B6 0D   206:          LDA A £TCS     LOAD TAKE CONTROL OF BUS
10F6 B7 EE03 207:          STA A AUXCMD     PUT IN TO THE 9914A
10F9 BD 11D1 208:          JSR  AKWAIT     WAIT FOR OK
10FC B6 19   209:          LDA A £SPD     LOAD SERIAL POLL DIABLE
10FE B7 EE07 210:          STA A DATOUT     PUT OUT TO THE UNITS
1101 BD 11D1 211:          JSR  AKWAIT     WAIT FOR OK
1104 B6 0B   212:          LDA A £GTS     LOAD RELEASE CONTROL OF BUS
1106 B7 EE03 213:          STA A AUXCMD     PUT INTO THE 9914A

```

```

215: *
216: * TELL UNIT TO SEND AND SELF TO LISTEN
217: *
1109 B6 0C 218: LDA A £TCA LOAD TAKE CONTROL OF THE BUS
110B B7 EE03 219: STA A AUXCMD PUT IT IN THE 9914A
110E BD 11D1 220: JSR AKWAIT WAIT FOR OK
1111 B6 0A 221: LDA A £LTON LOAD REMOVE SELF TALK
1113 B7 EE03 222: STA A AUXCMD PUT IN THE 9914A
1116 B6 1006 223: LDA A TALKAD LOAD TALK ADDRESS(UNIT)
1119 B7 EE07 224: STA A DATOUT PUT OUT TO THE UNITS
111C BD 11D1 225: JSR AKWAIT WAIT FOR THE OK
111F B6 EE08 226: LDA A ADDSWT PICK UP SELF ADDRESS
1122 B8 20 227: EOR A £*20 TURN INTI A READ ADDRESS
1124 B7 EE07 228: STA A DATOUT PUT OUT TO THE UNIT
1127 BD 11D1 229: JSR AKWAIT WAIT FOR THE OK
112A B6 B9 230: LDA A £HLON LOAD SELF TO LISTEN
112C B7 EE03 231: STA A AUXCMD PUT INTO THE 9914A
112F B6 0B 232: LDA A £GTS LOAD RELEASE THE BUS
1131 B7 EE03 233: STA A AUXCMD PUT INTO THE 9914A
1134 39 234: RTS RETURN
235: *

```

```

237: *
238: *
239: * AFTER THE SERIES OF BYTES HAVE BEEN RECEIVED
240: * TERMINATED BY WHAT EVER RUN INTI THIS PROG
241: * TO REVERSE THE BUS IE SELF BACK TO TALK
242: * AND UNIT BACK TO LISTEN.
243: *
1135 B6 0C 244: CHBACK LDA A £TCA          LOAD TAKE CONTROL
1137 B7 EE03 245:          STA A AUXCMD        PUT INTO THE 9914A
113A BD 11D1 246:          JSR  AKWAIT        WAIT FOR OK
113D B6 EE08 247:          LDA A ADDSWT        LOAD SELF ADDRESS SWITCH
1140 B8 40 248:          EOR A £*40          TURN SELF TO TALK
1142 B7 EE07 249:          STA A DATOUT        PUT OUT TO UNIT
1145 BD 11D1 250:          JSR  AKWAIT        WAIT FOR IT
1148 B6 1005 251:          LDA A LISTAD        LOAD UNIT LISTEN ADDRESS
114B B7 EE07 252:          STA A DATOUT        PUT OUT TO ALL UNITS
114E BD 11D1 253:          JSR  AKWAIT        YES THAT AGAIN
1151 B6 8A 254:          LDA A £HTON        LOAD SELF TO TALK
1153 B7 EE03 255:          STA A AUXCMD        PUT IT IN THE 9914A
1156 B6 0B 256:          LDA A £GTS          LOAD GO TO STANDBY
1158 B7 EE03 257:          STA A AUXCMD        PUT IT IN THE 9914A
115B BD 11D1 258:          JSR  AKWAIT        IF YOU DONT KNOW BY NOW
115E 39 259:          RTS              RETURN
260: * GO READ THE UNIT AGAIN

```

```

262: *
263: * THIS IS FOR USE WITH READ FROM HP1B UNIT
264: * THIS READS THE DATA IN READY FLAG
265: * IT IS THE SAME AS AKWAIT BUT READS THE
266: * INPUT FLAG.
267: *
115F 37 268: DAWAIT PSH B          SAVE "B"
1160 C6 FF 269:          LDA B £$FF      LOAD TIME
1162 F7 1002 270:          STA B DELT1
1165 F7 1003 271:          STA B DELT2
1168 B6 05 272:          LDA A £$05
116A F7 1004 273:          STA B DELT3
116D F6 EE00 274: DAWA1 LDA B INTST0      READ STATUS REG: 0
1170 C4 20 275:          AND B £$20      STRIP ALL BUT READ FLAG
1172 27 02 276:          BEQ  DALOP      WAIT TILL SET
1174 33 277:          PUL B          RESTORE THE STACK
1175 39 278:          RTS          RETURN
279: *
1176 7A 1002 280: DALOP DEC  DELT1
1179 26 F2 281:          BNE  DAWA1
117B 7A 1003 282:          DEC  DELT2
117E 26 ED 283:          BNE  DAWA1
1180 7A 1004 284:          DEC  DELT3
1183 26 EB 285:          BNE  DAWA1
1185 33 286:          PUL B
1186 CE 102E 287:          LDX  £ERROR      POINT AT ERROR
1189 BD F894 288:          JSR  PDATA1     PRINT IT
118C 7E D283 289:          JMP  WSTART     GO BACK TO DOS
290: *

```



```

292: *
293: *
294: * SUB FOR SENDING STRING TO HPIB LENGTH
295: * DETERMINED BY THE FIRST CHARACTER
296: * ENTER WITH "X" POINTING AT STRING.
297: *
298: *
118F 37 299: SENDIT PSH B
1190 E6 00 300: LDA B 0,X          Set up byte count
1192 27 0D 301: BEQ PRINDT       IF ZERO END
1194 08 302: INX              Bump pointer past number
1195 A6 00 303: PRINTL LDA A 0,X    Pick up character
1197 08 304: INX
1198 B7 EE07 305: STA A DATOUT      Put out to the unit
119B BD 11D1 306: JSR AKWAIT      Wait for data accepted
119E 5A 307: DEC B
119F 26 F4 308: BNE PRINTL
11A1 33 309: PRINDT PUL B
11A2 39 310: RTS
311: *
312: * SEND BYTE TO THE 250, ENTER WITH "X"
313: * POINTING AT THE STRING TO BE SENT, THE
314: * FIRST BYTE BEING THE LENGTH OF STRING
315: *
11A3 37 316: SENDBT PSH B          SAVE "B"
11A4 E6 00 317: LDA B 0,X          PICK UP LENGTH
11A6 27 0A 318: BEQ SENDT
11A8 08 319: INX              POINT AT THE NEXT
11A9 A6 00 320: SENDB1 LDA A 0,X    PICK UP BYTE TO BE SENT
11AB 08 321: INX              POINT AT THE NEXT
11AC BD 11B4 322: JSR OUTEED      SEND IT DOWN THE RS232
11AF 5A 323: DEC B          DEC; COUNTER
11B0 26 F7 324: BNE SENDB1      GO BACK TILL DONE
11B2 33 325: SENDT PUL B        RESTORE THE STACK
11B3 39 326: RTS

```

```

328: *
329: * TRANSMIT CH; IN "A" ON RS232
330: *
11B4 37 331: OUTEEE PSH B          SAVE "b"
11B5 F6 EA02 332: OT3 LDA B ACIAS      READ STATUS
11B8 57 333: ASR B              SHIFT IN FLAG
11B9 57 334: ASR B              SHIFT IN FLAG
11BA 24 F9 335: BCC OT3
11BC B7 EA03 336: STA A ACIAD        PUT OUT THE DATA
11BF 33 337: PUL B              RESTORE "b"
11C0 39 338: RTS
339: *
340: *
341: * LOAD SEND DATA INTO THE ACIA REG;
342: * THEN READ ANY DATA SENT FROM THE HP250
343: * IF THE RECEIVED BYTE IS "05" ANSWER WITH "06"
344: *
11C1 B6 EA02 345: INEEE LDA A ACIAS      READ STATUS REG
11C4 47 346: ASR A
11C5 24 FA 347: BCC INEEE          IF NOT READY RETURN
11C7 B6 EA03 348: LDA A ACIAD        READ DATA
11CA 84 7F 349: AND A £#7F
11CC 81 7F 350: CMP A £#7F          SEE IF VALID
11CE 27 F1 351: BEQ INEEE          IF 7F GO AND READ AGAIN
11D0 39 352: RTS              RETURN FROM SUB
    
```

LDAA #6
~~BEQ~~

CMPA #6
BEQ

LDAA #15
STAA ACIAS

CMPA #5 IS IT ENQ
BEQ
~~PSH B~~
LDAB #55
STAB ACIAS
PULB

```

354: *
355: *
356: * Subroutine to wait for byte out accepted
357: *
11D1 36      358: AKWAIT PSH A           Save the register
11D2 B6 FF   359:      LDA A £$FF       SET UP COUNTER
11D4 B7 1002 360:      STA A DELT1      PUT IN COUNT
11D7 B7 1003 361:      STA A DELT2      PUT IN COUNT
11DA B6 05   362:      LDA A £$5        FIVE LOOPS
11DC B7 1004 363:      STA A DELT3
11DF B6 EE00 364: AKLOOP LDA A INTST0    Get interupt status
11E2 B4 10   365:      AND A £$10       Mask out all but out bit
11E4 27 02   366:      BEQ  AKLP1       If not accepted read again
11E6 32      367:      PUL A           Restore the reg
11E7 39      368:      RTS           Return from sub
369: *
11E8 7A 1002 370: AKLP1 DEC  DELT1      DEC COUNTER
11EB 26 F2   371:      BNE  AKLOOP      GO AND TRY AGAIN
11ED 7A 1003 372:      DEC  DELT2      DEC SECOND COUNTER
11F0 26 ED   373:      BNE  AKLOOP      GO TRY AGAIN
11F2 7A 1004 374:      DEC  DELT3      DEC LAST COUNT
11F5 26 E8   375:      BNE  AKLOOP      IF NOT LAST GO BACK
11F7 32      376:      PUL A           RESTORE STACK
11F8 CE 102E 377:      LDX  £$ERROR     POINT AT ERROR MSG:
11FB BD F894 378:      JSR  PDATA1      GO PRINT THE STRING
11FE 7E D283 379:      JMP  WSTART      GO BACK TO DOS

```

```

381: *
382: * PATCH TO READ THE 250 AND WHEN A DC1 ARRIVES
383: * THEN ANSWER WITH A PREDETERMINED ANSWER
384: *
1201 BD 11C1 385: PATCH0 JSR INEEE READ THE 250
1204 B1 11 386: CMP A £#11 IS IT DC1
1206 26 F9 387: BNE PATCH0 WAIT TILL DC1
1208 39 388: RTS RETURN
389: *
1209 BD 1201 390: PTCHIN JSR PATCH0 WAIT FOR DC1
120C CE 1237 391: LDX £PAUSE1 POINT AT FIRST STRING
120F BD 11A3 392: JSR SENDBT SEND IT TO THE 250
393: *
1212 BD 1201 394: JSR PATCH0 READ 250
1215 CE 1237 395: LDX £PAUSE1 YES THE SAME AS B4
1218 BD 11A3 396: JSR SENDBT SEND IT TO THE 250
397: *
121B BD 1201 398: JSR PATCH0 WAIT FOR THE DC1
121E CE 123B 399: LDX £PAUSE2 POINT AT STRING
1221 BD 11A3 400: JSR SENDBT SEND IT TO THE 250
401: *
1224 BD 1201 402: JSR PATCH0 WAIT YET AGAIN FOR THE DC1
1227 CE 1241 403: LDX £PAUSE3 POINT AT ANOTHER STRING
122A BD 11A3 404: JSR SENDBT SEND THE STRING TO THE 250
405: *
122D BD 1201 406: JSR PATCH0 WAIT FOR DC1
1230 CE 1249 407: LDX £PAUSE4 POINT AT THE STRING
1233 BD 11A3 408: JSR SENDBT SEND THE STRING
1236 39 409: RTS RETURN
410: *
411: *
1237 03 412: PAUSE1 FCB 3, '0, #D, #A
413: *
123B 05 414: PAUSE2 FCB 5, '9, '2, '8, #D, #A
415: *
1241 07 416: PAUSE3 FCB 7, '7, '2, '2, '0, 'A, #D, #A
417: *
1249 03 418: PAUSE4 FCB 3, '8, #D, #A
    
```

~~CMPA #B05 IS IT ENQ~~
~~BEQ FRSTAK SEND ECK~~
~~* send ACK to 250~~
~~FRSTAK LDA #B06~~
~~JSR OUTEE~~
~~BRA PATCH#0~~

```

420: *
421: * START OF THE MAIN PROGRAM TO READ
422: * THE ENQ AND REPLY WITH AKN.
423: *
124D BD 104B 424: MAIN JSR SOFTRT GO SET UP THE I O's
1250 BD 1209 425: JSR PTCHIN GO READ ALL THE HP250 DATA
426: *
427: * MAIN PROGRAM FOR READING DATA AND REPLY
428: * WITH "06 WHEN BUFFER CLEAR.
429: *
1253 CE 12C8 430: GOLOP1 LDX £BUFFER POINT AT THE BUFER
1256 5F 431: CLR B SET BYTE COUNT TO ZERO
1257 BD 11C1 432: GOLOP2 JSR INEEE READ HP250 DATA
125A B1 11 433: CMP A £$11 IS IT A DC1
125C 27 36 434: BEQ GOLOP7 IF YES READ THE LAST AND REPLY
125E B1 05 435: CMP A £$05 IS IT ENQ
1260 27 0B 436: BEQ GOLOP3 IF YES PROCESS
1262 A7 00 437: STA A 0,X PUT INTO THE BUFFER
1264 08 438: INX POINT AT THE NEXT
1265 5C 439: INC B INC: BYTE COUNT
1266 8C 13C8 440: CPX £BUFFER+256 IS IT THE END OF SPACE
1269 26 EC 441: BNE GOLOP2 IF NO GO READ MORE
126B 20 16 442: BRA GOLOP5
443: *
126D CE 12C8 444: GOLOP3 LDX £BUFFER POINT AT THE START OF BUFFER
1270 C1 00 445: GOLOP4 CMP B £0 IS IT THE FIRST BYTE
1272 27 18 446: BEQ GOLOP6 GO REPLY WITH 06
1274 A6 00 447: LDA A 0,X PICK UP FROM THR BUFFER
1276 B7 EE07 448: STA A DATOUT PUT INTO THE HPIB
1279 BD 11D1 449: JSR AKWAIT WAIT FOR JOB COMPLETED
127C 5A 450: DEC B DEC BYTE COUNT
127D 08 451: INX POINT AT THE NEXT
127E 8C 13C8 452: CPX £BUFFER+256 IS IT THE LAST
1281 26 ED 453: BNE GOLOP4 IF NOT GO AGAIN
1283 CE 102E 454: GOLOP5 LDX £ERROR OINT AT ERROR MESSAGE
1286 BD F894 455: JSR PDATA1 GO PRINT IT
1289 7E D283 456: JMP WSTART GO BACK TO MONITOR
457: *
128C 86 06 458: GOLOP6 LDA A £$06 LOAD AKN
128E BD 11B4 459: JSR OUTEEO SEND IT TO THE HP250
1291 7E 1253 460: JMP GOLOP1 GO FOR A BIT MORE
461: *
1294 8C 12C8 462: GOLOP7 CPX £BUFFER IS IT THE START
1297 27 26 463: BEQ GOLOP9 IF YES MISS ALL
1299 09 464: DEX GO BACK ONE
129A 8C 12C8 465: CPX £BUFFER AND ANOTHER
129D 27 20 466: BEQ GOLOP9 IS THE START
129F 09 467: DEX AND ANOTHER
12A0 8C 12C8 468: CPX £BUFFER IS IT THE START OF BUFFER
12A3 27 1A 469: BEQ GOLOP9 AND ANOTHER
12A5 09 470: DEX AND ANOTHER
12A6 8C 12C8 471: CPX £BUFFER IS IT THE START OF BUFFER
12A9 27 14 472: BEQ GOLOP9 IF YES CLEAN UP
12AB FF 1007 473: STX TEMPE SAVE THE TEMP END
    
```

Handwritten annotations:

- Red arrows pointing to lines 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 458, 459, 462, 463, 464, 465, 466, 467, 468, 469, 471, 472, 473.
- Red text annotations:
 - Line 433: *CMPA # \$11*
 - Line 434: *BEQ GOLOP7*
 - Line 435: *CMPA # \$05*
 - Line 436: *BEQ GOLOP3*
 - Line 437: *STA A 0,X*
 - Line 438: *INX*
 - Line 439: *INC B*
 - Line 440: *CPX £BUFFER+256*
 - Line 441: *BNE GOLOP2*
 - Line 442: *BRA GOLOP5*
 - Line 445: *CMP B £0*
 - Line 446: *BEQ GOLOP6*
 - Line 447: *LDA A 0,X*
 - Line 448: *STA A DATOUT*
 - Line 449: *JSR AKWAIT*
 - Line 450: *DEC B*
 - Line 451: *INX*
 - Line 452: *CPX £BUFFER+256*
 - Line 453: *BNE GOLOP4*
 - Line 454: *GOLOP5 LDX £ERROR*
 - Line 455: *JSR PDATA1*
 - Line 456: *JMP WSTART*
 - Line 458: *GOLOP6 LDA A £\$06*
 - Line 459: *JSR OUTEEO*
 - Line 462: *GOLOP7 CPX £BUFFER*
 - Line 463: *BEQ GOLOP9*
 - Line 464: *DEX*
 - Line 465: *CPX £BUFFER*
 - Line 466: *BEQ GOLOP9*
 - Line 467: *DEX*
 - Line 468: *CPX £BUFFER*
 - Line 469: *BEQ GOLOP9*
 - Line 471: *CPX £BUFFER*
 - Line 472: *BEQ GOLOP9*
 - Line 473: *STX TEMPE*
- Other red text annotations:
 - CMPA # \$1B*
 - BEQ GOLOP6*
 - CMPA # \$1B IS FF ESC*
 - BEQ GOLOP6*
 - BNE*
 - DEC B INX*
 - BEQ GOLOP9*
 - STAB*
 - STAB TEMPB*
 - CMPB # 0*
 - BEQ GOLOP9*
 - CMPA # \$1B*
 - BNE*
 - LDA # \$06*
 - JSR OUTEEO*
 - JSR INEEE*
 - CMPA # \$11*
 - BNE*
 - BRA GOLOP1*
 - CMPA # \$*
 - BEQ GOLOP2*

12AE	CE	12C8	474:	LDX	£BUFFER	POINT AT THE START OF BUFFER
12B1	A6	00	475:	GOLOP8	LDA A 0,X	PICK UP FROM BUFFER
12B3	B7	EE07	476:	STA	A DATOUT	PUT INTO THE HP1B
12B6	BD	11D1	477:	JSR	AKWAIT	WAIT FOR READY
12B9	08		478:	INX		POINT AT THE NEXT
12BA	BC	1007	479:	CPX	TEMPE	IS IT THE END OF DATA
12BD	26	F2	480:	BNE	GOLOP8	GO READ ANOTHER
12BF	CE	123B	481:	GOLOP9	LDX £PAUSE2	POINT AT 928 CR LF
12C2	BD	11A3	482:	JSR	SENDBT	GO SEND TO HP250
12C5	7E	1253	483:	JMP	GOLOP1	GO BACK TO THE START
			484:	*		
12C8			485:	BUFFER	RMB 256	
124D			486:	END	MAIN	

NO ERROR(S) DETECTED