Commodore 64 Multiple Servo Control

Source Code

```
1000 ;++++++++++++++++++
1010 ;++++++++++++++++++
1020 ;++                ++
1030 ;++ CBM MULTIPLE  ++
1040 ;++ SERVO CONTROL++
1050 ;++                ++
1060 ;++++++++++++++++++
1070 ;++++++++++++++++++
1080 ;
1090 PORT = 56577 ;USER PORT DATA REGISTER
1100 ANGLE=12288  ;ANGLE VALUE LOCATION
1110 ZPAGE=$FB    ;0 PAGE POINTER TO TABLE
1120 ;
1130 *=$0334
1140 ;
1150       SEI          ;INTERRUPTS OFF
1160       LDA $0314 ;EXISTING IRQ VECTOR
1170       LDX $03C4
1180       STA $03C4
1190       STX $0314
1200       LDA $0315
1210       LDX $03C5
1220       STA $03C5
1230       STX $0315
1240 ;
1250 ;++++ INITIALISE TABLE ++++
1260 ;
1270       LDA #$FF
1280       LDY #$00
1290 TABLE
1300       STA (ZPAGE),Y
1310       DEY
1320       BNE TABLE
1330       CLI          ;INTERRUPTS ON
1340       RTS
1350 ;
1360 ;++++ EVENT HANDLER ++++
1370 ;
1380       PHP
1390       PHA          ;SAVE REGISTERS
1400       TYA          ;ON STACK
1410       PHA
1420       TXA
1430       PHA
1440 ;
1450 ;++ START PULSE, FOR SOME MOTORS IT
1460 ;MAY BE POSSIBLE TO START BEFORE FILLING
1470 ;TABLE AND SO REDUCE WAIT LOOP BELOW ++
1480 ;
1490       LDA #$FF
1500       STA PORT
1510 ;++ FILL TABLE WITH EXCEPTIONS ++
1520       LDX #$07
1530       LDA #$FF
1540       CLC
1550 EXCEPT
1560       ROR A
1570       PHA          ;BIT PATTERN
1580       LDY ANGLE,X ;GET MOTOR X OFFSET
1590       AND (ZPAGE),Y ;KEEP EXISTING PATTERN
1600       STA (ZPAGE),Y ;BUT MODIFIED FOR MOTOR X
1610       PLA
1620       DEX
1630       BPL EXCEPT
1640 ;++ TABLE IS NOW LOADED ++
1650       LDY #$30
1660 WAIT
1670       DEY          ;FILL IN SOME
1680       BNE WAIT     ;TIME
1690 ;
1700       LDA #$FF     ;ALL PULSES ON
1710       LDY #$00
1720 LOOP
1730       AND (ZPAGE),Y ;BUT MASK OFF WITH EACH
1740       STA PORT     ;TABLE ELEMENT IN TURN
1750       INY
1760       BNE LOOP
1770 ;
1780       LDX #$07
1790       LDA #$FF
1800 CLEAR
1810       LDY ANGLE,X  ;CLEAR ALL EXCEPTIONS
1820       STA (ZPAGE),Y
1830       DEX
1840       BPL CLEAR
1850 ;++ ALL PULSES SHOULD NOW BE FINISHED ++
1860       PLA
1870       TAX
1880       PLA          ;RESTORE REGISTERS
1890       TAY
1900       PLA
1910       PLP
1920       JMP $EA31
```

BASIC Loader Program

```
10 REM **** BASIC LOADER FOR ***
*
20 REM **** MULTIPLE SERVOS  ***
*
30 :
40 FOR I=820 TO 922
50 READ A:POKE I,A
60 CC=CC+A
70 NEXT I
80 READ CS:IF CS<>CC THEN PRINT"
CHECKSUM ERROR":STOP
100 DATA120,173,20,3,174,196,3,1
41,196
110 DATA3,142,20,3,173,21,3,174,
197,3
120 DATA141,197,3,142,21,3,169,2
55,160
130 DATA0,145,251,136,208,251,88
,96,8
140 DATA72,152,72,138,72,169,255
,141,1
150 DATA221,162,7,169,255,24,106
,72
160 DATA188,0,48,49,251,145,251,
184
170 DATA202,16,243,160,48,136,20
8,253
180 DATA169,255,160,0,49,251,141
,1,221
190 DATA200,208,248,162,7,169,25
5,188
200 DATA0,48,145,251,202,16,248,
104
210 DATA170,104,168,104,40,76,49
,234
220 DATA13072:REM*CHECKSUM*
```

BASIC Calling Program

```
10 REM **** MULTIPLE SERVO ****
20 :
30 DN=8:REM IF CASSETTE THEN DN=
1
40 IF A=0 THEN A=1:LOAD"MULTISER
V.HEX*",DN,1
50 POKE 778,88: POKE 779,3:REM P
OINTER TO EVENT HANDLER
60 POKE 251,0: POKE 252,48:REM S
ET UP ZERO PAGE PTR
70 DDR= 56579:POKE DDR,255:REM A
LL OUTPUT
80 MC=826: SYS MC
90 :
100 GET K$:IF K$="" THEN 100:REM
 AWAIT KEY
110 REM ** ALTER MOTOR POSITION
**
115 AK=ASC(K$)
120 IF AK>48 AND AK<58 THEN POKE
12288+SERVO,VAL(K$)*20
130 IF AK>32 AND AK<40 THEN SERV
O=ASC(K$)-35
140 IF K$<>"E" THEN 100:REM 'E'
TO EXIT
```

Single Servo Motor Control

```
 755 REM ************************
 756 REM  Assemble the machine code
 757 REM ************************
10000 DEF PROCinitial
10010 DIM space% 200
10020 FOR I=0 TO 2 STEP 2
10030  portb=&FE60
10040  P%=space%
10050  angle=P%
10060  P%=P%+1
10070  [OPT I
10080  .eventhandler
10090  \save registers first
10100  PHP:PHA:TYA:PHA:TXA:PHA
10110  LDA &604
10120  LDX &timer
10130  LDY &timer
10140  JSR &FFF1 \reset timer
10150  LDA &FF
10160  STA portb
10170  \wait approx. 1msec
10180  LDY &FF
10190  .LOOP DEY
10200       BNE LOOP
10210  \end countout pulse
10220  LDY angle
10230  .LOOP1 DEY
10240       BNE LOOP1
10250  \stop all output pulses
10260  LDA &0
10270  STA portb
10280  PLA:TAX:PLA:TAY:PLA:PLP
10290  RTS
10300  ]
10310  NEXT I
10320 REM point to eventhandler
10330 !&220=eventhandler OR !&220 AND &FFFF0000
10340 ENDPROC
```

BBC Micro Listings

Common Initial Routine

```
10 MODE 0
15 REM ****************************
16 REM     Set up the timer etc
17 REM ****************************
20 osbyte=&FFF4
30 A%=&9C :X%=&62 :Y%=&FF
40 CALL osbyte:REM set up port B for output
50 CLS
60 DIM p% 600
70 DIM timer% 12 ,read% 12
80 xtimer=timer% MOD 256
90 ytimer=timer% DIV 256
100 xread=read% MOD 256
110 yread=read% DIV 256
120 PROCinitial
130 FOR I%=angle TO angle+8:angle?I%=128:NEXT
140 t=.02  :REM sec between pulses
150 time%=&FFFFFFFF -(t*100) +1
160 timer%?4=&FF    :REM load highest byte
170 !timer%=time%   :REM set up timer, enable events
180 A%=14 :X%=timer :Y%=ytimer :CALL &FFF1
195 REM ****************************
196 REM     The BASIC controller
197 REM ****************************
200 CLS
210 PRINT"PRESS  SHIFT + NUMBER to select motor"
220 PRINT"PRESS  NUMBER to select angle"
225 motor=1
230 REPEAT
240   A=GET
250   IF A%&2F AND A<&3A THEN a=(A-&30)*10+(255/90):angle?
      (motor-1)=a:PRINTTAB(1,motor+1);"motor "motor" angle "a*(90/255)
260   IF A%&20 AND A<&2A THEN motor=A-&20
270   UNTIL 0
280 END
750
```

Multiple Servo Motor Control

```
755 REM ****************************
756 REM Assemble the machine code
757 REM ****************************
760 DEF PROCinitial
770 DIM space% 600
780 FOR I=0 TO 3 STEP 3
790   zeropage=&70   :REM free for users
800   portb=&FE60 :osword=&FFF1
810   P%=space%
820   angle=P% :P%=P%+8  :REM potentially 8 motors
830   table=P% :P%=P%+256  :REM 256 possible pulse lengths
835   FOR I%=table TO table+&100:?I%=&FF:NEXT
840   lowtable%=table MOD 256
850   hightable%=table DIV 256
860   ?zeropage=lowtable% :zeropage?1=hightable%
870   [OPT I
880   .eventhandler
890   PHP:PHA:TYA:PHA:TXA:PHA
900   LDA &604
910   LDX &timer
920   LDY &timer
930   JSR osword
940   \Start pulse, for some motors it may be possible to start
1000  \before filling table and so reduce the wait loop below
1010  LDA &FF :STA portb
1020  \fill table with exceptions
1030  LDX &7 :LDA &FF :CLC \set up bit pattern
1040  .exceptions
1050       ROR A :PHA  \bit pattern
1060       LDY angle,X \get offset corresponding to angle of motor X
1070       AND (zeropage),Y \keep existing bit pattern
1080       STA (zeropage),Y \but modified for motor X
1090       PLA :DEX
1100       BPL exceptions
1110  \table is now loaded, fill in some time
1120  LDY &A0
1130  .wait DEY
1140       BNE wait
1150  LDA &FF  \all pulses on
1160  LDY &0
1170  .loop AND (zeropage),Y  \but mask off with each table
1180       STA portb  \element in turn
1190       INY
1200       BNE loop
1201  LDX &7 :LDA &FF
1202  .clear
1205       LDY angle,X  \clear all the exceptions again
1207       STA (zeropage),Y
1208  DEX
1209  BPL clear
1210   \all pulses should now be finished
1220  PLA:TAX:PLA:TAY:PLA:PLP
1230  RTS
1240  ]
1250  NEXT I
1260 !&220=eventhandler OR !&220 AND &FFFF0000
1270 ENDPROC
```