Commodore 64 Single Servo Control

The first part of the Commodore 64 source listing for a single servo control shows how the interrupt vectors (locations 788 and 789) are altered. This cannot be done using BASIC as an interrupt may occur during this alteration, causing the system to crash. Notice that the interrupts are turned off (SEI) while the alteration is made, and are re-enabled using CLI. The rest of the code is the interrupt handling routine for a single servo control.

The BASIC Calling program shows everything required to load the machine code routines, set up the user port, and then POKE values into memory location $3000 (12288) according to which key (1 to 9) is pressed. A motor connected to the user port should then move to a position proportional to the key value. Pressing E ends the session.

If you have an assembler, type in the source listing and assemble it into an object file that can be subsequently loaded by the BASIC Calling program. Alternatively, type in the BASIC Loader for the machine code and run this to load the code into memory. Type NEW before loading and running the BASIC Calling program. If you use the BASIC Loader, then lines 30 and 40 can be omitted.

Note: It is extremely important to note that if anything is wrong in a program that uses interrupts, the whole system can very easily become totally corrupted. This does not damage the computer, but you will probably have to switch the machine off and back on again to recover. Therefore, it is imperative to SAVE the program before RUNning it

Source Code

```
;++++++++++++++++++++++
;++++++++++++++++++++++
;++                    ++
;++ CBM SINGLE SERVO   ++
;++       DRIVER       ++
;++                    ++
;++++++++++++++++++++++
;++++++++++++++++++++++
;
PORT = 56579 ; USER PORT DATA REG
ANGLE=12288  ; ANGLE VALUE LOCATION
;
*=$0334
;
      SEI          ;INTERRUPTS OFF
      LDA $0314    ;EXXISTING IRQ VECOTOR
      LDX $03C4
      STA $03C4
      STA $0314
      LDA $0315
      LDX $03C5
      STA $03C5
      STX $0315
;
      CLI          ;INTERRUPTS BACK ON
      RTS
;
;++++ EVENT HANDLER ++++
;
      PHP
      PHA
      TYA          ;SAVE REGISTERS
      PHA          ;ON STACK
      TXA
      PHA
      LDA #$FF
      STA PORT
      LDY #$FF
LOOP
      DEY          ;DELAY LOOP
      BNE LOOP     ;APPROX 1MSEC
;
      LDY ANGLE
LOOP1
      DEY          ;COUNT OUT PULSE
      BNE LOOP1
;
      LDA #$00
      STA PORT     ;ZERO DATA REGISTER
;
      PLA
      TAX          ;RESTORE REGISTER
      PLA          ;VALUES
      TAY
      PLA
      PLP
;
      JMP $EA31
```

BASIC Loader Program

```
10 REM **** BASIC LOADER FOR ***
*
20 REM **** SINGLE SERVO PROG***
*
30 :
40 FOR I=820 TO 882
50 READ A:POKE I,A
60 CC=CC+A
70 NEXT I
80 READ CS:IF CC<>CS THENPRINT
"CHECKSUM ERROR":STOP
100 DATA120,173,20,3,174,196,3,1
41,196
110 DATA3,141,20,3,173,21,3,174,
197,3
120 DATA141,197,3,142,21,3,88,96
,8,72
130 DATA152,72,138,72,169,255,14
1,3
140 DATA221,160,255,136,208,253,
172,0
150 DATA48,136,208,253,169,0,141
,3,221
160 DATA104,170,104,168,104,40,7
6,49
170 DATA234
180 DATA7170:REM*CHECKSUM*
```

BASIC Calling Program

```
10 REM **** SINGLE SERVO ****
20 :
30 DN=8:REM IF CASSETTE THEN
DN=1
40 IF A=0 THEN A=1:LOAD"SINGSERV
.HEX",8,1
50 POKE 964,79: POKE965,3:REM
POINT TO IRQ HANDLER
60 DDR=56577: POKE DDR,255:
REM ALL OUTPUT
70 MC=820:SYS MC: REM SET IRQ
VECTOR
80 POKE 53265,PEEK(53265)AND239:
REM BLANK SCREEN
90 :
100 GET K$:IF K$="" THEN100:REM
AWAIT KEYPRESS
110 REM ** ALTER MOTOR POSITION
**
120 IF ASC(K$)>48 AND ASC(K$)<58
 THEN POKE 12288,VAL(K$)*20
130 IF K$<>"E" THEN 80:REM 'E'
TO EXIT
140 END
```