```
os series V
GEOFF COX
 ************************************************************************
 *                                                                      *
 *       Check SPEECH System                                            *
 *                                                                      *
 ************************************************************************

                          ;X=&10
 DB11    BIT      &FE40   ;if bit 7 low then we have speec system fitted
 DB14    BMI      &DB27   ;else goto DB27

 DB16    DEC      &027B   ;(027B)=&FF to indicate speech present

 DB19    LDY      #&FF    ;Y=&FF
 DB1B    JSR      &EE7F   ;initialise speech generator
 DB1E    DEX              ;via this
 DB1F    BNE      &DB19   ;loop
                          ;X=0
 DB21    STX      &FE48   ;set T2 timer for speech
 DB24    STX      &FE49   ;


 *********** SCREEN SET UP *********************************************
                          ;X=0
 DB27    LDA      &028F   ;get back start up options (mode)
 DB2A    JSR      &C300   ;then jump to screen initialisation

 DB2D    LDY      #&CA    ;Y=&CA
 DB2F    JSR      &E4F1   ;to enter this in keyboard buffer
                          ;this enables the *KEY 10 facility

 ********* enter BREAK intercept with Carry Clear **********************

 DB32    JSR      &EAD9   ;check to see if BOOT address is set up if so
                          ;JMP to it

 DB35    JSR      &F140   ;set up cassette options
 DB38    LDA      #&81    ;test for tube to FIFO buffer 1
 DB3A    STA      &FEE0   ;
 DB3D    LDA      &FEE0   ;
 DB40    ROR              ;put bit 0 into carry
 DB41    BCC      &DB4D   ;if no tube then DB4D
 DB43    LDX      #&FF    ;else
 DB45    JSR      &F168   ;issue ROM service call &FF
                          ;to initialise TUBE system
 DB48    BNE      &DB4D   ;if not 0 on exit (Tube not initialised) DB4D
 DB4A    DEC      &027A   ;else set tube flag to show its active


 DB4D    LDY      #&0E    ;set current value of PAGE
 DB4F    LDX      #&01    ;issue claim absolute workspace call
 DB51    JSR      &F168   ;via F168
 DB54    LDX      #&02    ;send private workspace claim call
 DB56    JSR      &F168   ;via F168
 DB59    STY      &0243   ;set primary OSHWM
 DB5C    STY      &0244   ;set current OSHWM
 DB5F    LDX      #&FE    ;issue call for Tube to explode character set etc.
 DB61    LDY      &027A   ;Y=FF if tube present else Y=0
 DB64    JSR      &F168   ;and make call via F168

 DB67    AND      &0267   ;if A=&FE and bit 7 of 0267 is set then continue
 DB6A    BPL      &DB87   ;else ignore start up message
 DB6C    LDY      #&02    ;output to screen
 DB6E    JSR      &DEA9   ;'BBC Computer ' message
 DB71    LDA      &028D   ;0=warm reset, anything else continue
```

```
DB74    BEQ     &DB82   ;
DB76    LDY     #&16    ;by checking length of RAM
DB78    BIT     &028E   ;
DB7B    BMI     &DB7F   ;and either
DB7D    LDY     #&11    ;
DB7F    JSR     &DEA9   ;finishing message with '16k' or '32k'
DB82    LDY     #&1B    ;and two newlines
DB84    JSR     &DEA9   ;

*********: enter BREAK INTERCEPT ROUTINE WITH CARRY SET (call 1)
DB87    SEC             ;
DB88    JSR     &EAD9   ;look for break intercept jump do *TV etc
DB8B    JSR     &E9D9   ;set up LEDs in accordance with keyboard status
DB8E    PHP             ;save flags
DB8F    PLA             ;and get back in A
DB90    LSR             ;zero bits 4-7 and bits 0-2 bit 4 which was bit 7
DB91    LSR             ;may be set
DB92    LSR             ;
DB93    LSR             ;
DB94    EOR     &028F   ;or with start up options which may or may not
DB97    AND     #&08    ;invert bit 4
DB99    TAY             ;Y=A
DB9A    LDX     #&03    ;make initialisation call if Y=0 on entry
DB9C    JSR     &F168   ;RUN, EXEC or LOAD !BOOT file
DB9F    BEQ     &DBBE   ;if a ROM accepts this call then DBBE
DBA1    TYA             ;else put Y in A
DBA2    BNE     &DBB8   ;if Y<>0 DBB8
DBA4    LDA     #&8D    ;else set up standard cassete baud rates
DBA6    JSR     &F135   ;via &F135

DBA9    LDX     #&D2    ;
DBAB    LDY     #&EA    ;
DBAD    DEC     &0267   ;decrement ignore start up message flag
DBB0    JSR     OSCLI   ;and execute */!BOOT
DBB3    INC     &0267   ;restore start up message flag
DBB6    BNE     &DBBE   ;if not zero then DBBE

DBB8    LDA     #&00    ;else A=0
DBBA    TAX             ;X=0
DBBB    JSR     &F137   ;set tape speed

******** Preserve current language on soft RESET ************************

DBBE    LDA     &028D   ;get last RESET Type
DBC1    BNE     &DBC8   ;if not soft reset DBC8

DBC3    LDX     &028C   ;else get current language ROM address
DBC6    BPL     &DBE6   ;if +ve (language available) then skip search routine



*************************************************************************
*                                                                       *
*       SEARCH FOR LANGUAGE TO ENTER (Highest priority)                 *
*                                                                       *
*************************************************************************

DBC8    LDX     #&0F    ;set pointer to highest available rom

DBCA    LDA     &02A1,X ;get rom type from map
DBCD    ROL             ;put hi-bit into carry, bit 6 into bit 7
DBCE    BMI     &DBE6   ;if bit 7 set then ROM has a language entry so DBE6

DBD0    DEX             ;else search for language until X=&ff
DBD1    BPL     &DBCA   ;
```

```
*************** check if tube present ***********************************

DBD3    LDA     #&00    ;if bit 7 of tube flag is set BMI succeeds
DBD5    BIT     &027A   ;and TUBE is connected else
DBD8    BMI     &DC08   ;make error

********* no language error ***********************************************

DBDA    BRK             ;
DBDB    DB      &F9     ;error number
DBDC    DB      'Language?'     ;message
DBE5    BRK             ;


DBE6    CLC             ;


*************************************************************************
*                                                                       *
*       OSBYTE 142 enter Language ROM at &8000                          *
*                                                                       *
*       X=rom number C set if OSBYTE call clear if initialisation       *
*                                                                       *
*************************************************************************

DBE7    PHP             ;save flags
DBE8    STX     &028C   ;put X in current ROM page
DBEB    JSR     &DC16   ;select that ROM
DBEE    LDA     #&80    ;A=128
DBF0    LDY     #&08    ;Y=8
DBF2    JSR     &DEAB   ;display text string held in ROM at &8008,Y
DBF5    STY     &FD     ;save Y on exit (end of language string)
DBF7    JSR     OSNEWL  ;two line feeds
DBFA    JSR     OSNEWL  ;are output
DBFD    PLP             ;then get back flags
DBFE    LDA     #&01    ;A=1 required for language entry
DC00    BIT     &027A   ;check if tube exists
DC03    BMI     &DC08   ;and goto DC08 if it does
DC05    JMP     &8000   ;else enter language at &8000


*************************************************************************
*                                                                       *
*       TUBE FOUND enter tube software                                  *
*                                                                       *
*************************************************************************

DC08    JMP     &0400   ;enter tube environment


*************************************************************************
*                                                                       *
*       OSRDRM entry point                                              *

*                                                                       *
*       get byte from PHROM or page ROM                                 *
*       Y= rom number, address is in &F6/7                              *
*************************************************************************

DC0B    LDX     &F4     ;get current ROM number into X
DC0D    STY     &F4     ;store new number in &F4
DC0F    STY     &FE30   ;switch in ROM
DC12    LDY     #&00    ;get current PHROM address
DC14    LDA     (&F6),Y ;and get byte
```

```
******** Set up Sideways Rom latch and RAM copy ************************
                ;on entry X=ROM number

DC16    STX     &F4     ;RAM copy of rom latch
DC18    STX     &FE30   ;write to rom latch
DC1B    RTS             ;and return



********************************************************************************
********************************************************************************
**                                                                            **
**      MAIN IRQ Entry point                                                  **
**                                                                            **
**                                                                            **
********************************************************************************
********************************************************************************
;ON ENTRY STACK contains        STATUS REGISTER,PCH,PCL                      ;

DC1C    STA     &FC     ;save A
DC1E    PLA             ;get back status (flags)
DC1F    PHA             ;and save again
DC20    AND     #&10    ;check if BRK flag set
DC22    BNE     &DC27   ;if so goto DC27
DC24    JMP     (&0204) ;else JMP (IRQ1V)


********************************************************************************
*                                                                              *
*               BRK handling routine                                           *
*                                                                              *
********************************************************************************

DC27    TXA             ;save X on stack
DC28    PHA             ;
DC29    TSX             ;get status pointer
DC2A    LDA     &0103,X ;get Program Counter lo
DC2D    CLD             ;
DC2E    SEC             ;set carry
DC2F    SBC     #&01    ;subtract 2 (1+carry)
DC31    STA     &FD     ;and store it in &FD
DC33    LDA     &0104,X ;get hi byte
DC36    SBC     #&00    ;subtract 1 if necessary
DC38    STA     &FE     ;and store in &FE
DC3A    LDA     &F4     ;get currently active ROM
DC3C    STA     &024A   ;and store it in &24A
DC3F    STX     &F0     ;store stack pointer in &F0
DC41    LDX     #&06    ;and issue ROM service call 6
DC43    JSR     &F168   ;(User BRK) to roms
                        ;at this point &FD/E point to byte after BRK
                        ;ROMS may use BRK for their own purposes

DC46    LDX     &028C   ;get current language
DC49    JSR     &DC16   ;and activate it
DC4C    PLA             ;get back original value of X
DC4D    TAX             ;
DC4E    LDA     &FC     ;get back original value of A
DC50    CLI             ;allow interrupts
DC51    JMP     (&0202) ;and JUMP via BRKV (normally into current language)


********************************************************************************
*                                                                              *
*       DEFAULT BRK HANDLER                                                    *
*                                                                              *
```

```
*************************************************************************

DC54    LDY     #&00    ;Y=0 to point to byte after BRK
DC56    JSR     &DEB1   ;print message

DC59    LDA     &0267   ;if BIT 0 set and DISC EXEC error
DC5C    ROR             ;occurs
DC5D    BCS     &DC5D   ;hang up machine!!!!

DC5F    JSR     OSNEWL  ;else print two newlines
DC62    JSR     OSNEWL  ;
DC65    JMP     &DBB8   ;and set tape speed before entering current
                        ;language

DC68    SEC             ;set carry
DC69    ROR     &024F   ;and rotate right to set RS423 busy flag
DC6C    BIT     &0250   ;check bit 7 of current ACIA control register
DC6F    BPL     &DC78   ;if interrupts NOT enabled DC78
DC71    JSR     &E741   ;else E741 to check if serial buffer full
DC74    LDX     #&00    ;
DC76    BCS     &DC7A   ;if carry set goto DC7A to transfer data

DC78    LDX     #&40    ;X=&40
DC7A    JMP     &E17A   ;and transfer data

DC7D    LDY     &FE09   ;read serial data from ACIA
DC80    AND     #&3A    ;and %0011 1010
DC82    BNE     &DCB8   ;if no 0 then DCB8

DC84    LDX     &025C   ;read RS423 input suppression flag
DC87    BNE     &DC92   ;if not 0 then DC92 ignore RS423 input
DC89    INX             ;else X=X+1
DC8A    JSR     &E4F3   ;put byte in buffer
DC8D    JSR     &E741   ;count buffer
DC90    BCC     &DC78   ;and if carry clear (buffer not full) back to DC78
DC92    RTS             ;else return
                        ;
*************************************************************************
*                                                                       *
*       Main IRQ Handling routines, default IRQIV destination           *
*                                                                       *
*************************************************************************


DC93    CLD             ;clear decimal flag
DC94    LDA     &FC     ;get original value of A
DC96    PHA             ;save it
DC97    TXA             ;save X
DC98    PHA             ;
DC99    TYA             ;and Y
DC9A    PHA             ;
DC9B    LDA     #&DE    ;A=&DE
DC9D    PHA             ;store it
DC9E    LDA     #&81    ;save &81
DCA0    PHA             ;store it (an RTS will now jump to DE82)
DCA1    CLV             ;clear V flag
DCA2    LDA     &FE08   ;get value of status register of ACIA
DCA5    BVS     &DCA9   ;if parity error then DCA9
DCA7    BPL     &DD06   ;else if no interrupt requested DD06

DCA9    LDX     &EA     ;read RS423 timeout counter
DCAB    DEX             ;decrement it
DCAC    BMI     &DCDE   ;and if <0 DCDE
DCAE    BVS     &DCDD   ;else if >&40 DCDD (RTS to DE82)
DCB0    JMP     &F588   ;else read ACIA via F588
```

```
DCB3    LDY     &FE09   ;read ACIA data
DCB6    ROL             ;
DCB7    ASL             ;
DCB8    TAX             ;X=A
DCB9    TYA             ;A=Y
DCBA    LDY     #&07    ;Y=07
DCBC    JMP     &E494   ;check and service EVENT 7 RS423 error

DCBF    LDX     #&02    ;read RS423 output buffer
DCC1    JSR     &E460   ;
DCC4    BCC     &DCD6   ;if C=0 buffer is not empty goto DCD6
DCC6    LDA     &0285   ;else read printer destination
DCC9    CMP     #&02    ;is it serial printer??
DCCB    BNE     &DC68   ;if not DC68
DCCD    INX             ;else X=3
DCCE    JSR     &E460   ;read printer buffer
DCD1    ROR     &02D2   ;rotate to pass carry into bit 7
DCD4    BMI     &DC68   ;if set then DC68
DCD6    STA     &FE09   ;pass either printer or RS423 data to ACIA
DCD9    LDA     #&E7    ;set timeout counter to stored value
DCDB    STA     &EA     ;
DCDD    RTS             ;and exit (to DE82)


                        ;A contains ACIA status
DCDE    AND     &0278   ;AND with ACIA bit mask (normally FF)
DCE1    LSR             ;rotate right to put bit 0 in carry
DCE2    BCC     &DCEB   ;if carry clear receive register not full so DCEB
DCE4    BVS     &DCEB   ;if V is set then DCEB
DCE6    LDY     &0250   ;else Y=ACIA control setting
DCE9    BMI     &DC7D   ;if bit 7 set receive interrupt is enabled so DC7D

DCEB    LSR             ;put BIT 2 of ACIA status into
DCEC    ROR             ;carry if set then Data Carrier Detected applies
DCED    BCS     &DCB3   ;jump to DCB3

DCEF    BMI     &DCBF   ;if original bit 1 is set TDR is empty so DCBF
DCF1    BVS     &DCDD   ;if V is set then exit to DE82

DCF3    LDX     #&05    ;X=5
DCF5    JSR     &F168   ;issue rom call 5 'unrecognised interrupt'
DCF8    BEQ     &DCDD   ;if a rom recognises it then exit to DE82
DCFA    PLA             ;otherwise get back DE82 address from stack
DCFB    PLA             ;
DCFC    PLA             ;and get back X, Y, and A
DCFD    TAY             ;
DCFE    PLA             ;
DCFF    TAX             ;
DD00    PLA             ;
DD01    STA     &FC     ;&FC=A
DD03    JMP     (&0206) ;and offer to the user via IRQ2V


*************************************************************************
*                                                                       *
* VIA INTERUPTS ROUTINES                                                *
*                                                                       *
*************************************************************************

DD06    LDA     &FE4D   ;read system VIA interrupt flag register
DD09    BPL     &DD47   ;if bit 7=0 the VIA has not caused interrupt
                        ;goto DD47
DD0B    AND     &0279   ;mask with VIA bit mask
DD0E    AND     &FE4E   ;and interrupt enable register
DD11    ROR             ;rotate right twice to check for IRQ 1 (frame sync)
```

```
DD12    ROR             ;
DD13    BCC     &DD69   ;if carry clear then no IRQ 1, else
DD15    DEC     &0240   ;decrement vertical sync counter
DD18    LDA     &EA     ;A=RS423 Timeout counter
DD1A    BPL     &DD1E   ;if +ve then DD1E
DD1C    INC     &EA     ;else increment it
DD1E    LDA     &0251   ;load flash counter
DD21    BEQ     &DD3D   ;if 0 then system is not in use, ignore it
DD23    DEC     &0251   ;else decrement counter
DD26    BNE     &DD3D   ;and if not 0 go on past reset routine

DD28    LDX     &0252   ;else get mark period count in X
DD2B    LDA     &0248   ;current VIDEO ULA control setting in A
DD2E    LSR             ;shift bit 0 into C to check if first colour
DD2F    BCC     &DD34   ;is effective if so C=0 jump to DD34

DD31    LDX     &0253   ;else get space period count in X
DD34    ROL             ;restore bit
DD35    EOR     #&01    ;and invert it
DD37    JSR     &EA00   ;then change colour

DD3A    STX     &0251   ;&0251=X resetting the counter

DD3D    LDY     #&04    ;Y=4 and call E494 to check and implement vertical
DD3F    JSR     &E494   ;sync event (4) if necessary
DD42    LDA     #&02    ;A=2
DD44    JMP     &DE6E   ;clear interrupt 1 and exit


*************************************************************************
*                                                                       *
*       PRINTER INTERRUPT USER VIA 1                                    *
*                                                                       *
*************************************************************************

DD47    LDA     &FE6D   ;Check USER VIA interrupt flags register
DD4A    BPL     &DCF3   ;if +ve USER VIA did not call interrupt
DD4C    AND     &0277   ;else check for USER IRQ 1
DD4F    AND     &FE6E   ;
DD52    ROR             ;
DD53    ROR             ;
DD54    BCC     &DCF3   ;if bit 1=0 the no interrupt 1 so DCF3
DD56    LDY     &0285   ;else get printer type
DD59    DEY             ;decrement
DD5A    BNE     &DCF3   ;if not parallel then DCF3
DD5C    LDA     #&02    ;reset interrupt 1 flag
DD5E    STA     &FE6D   ;
DD61    STA     &FE6E   ;disable interrupt 1
DD64    LDX     #&03    ;and output data to parallel printer
DD66    JMP     &E13A   ;


*************************************************************************
*                                                                       *
*       SYSTEM INTERRUPT 5   Speech                                     *
*                                                                       *
*************************************************************************

DD69    ROL             ;get bit 5 into bit 7
DD6A    ROL             ;
DD6B    ROL             ;
DD6C    ROL             ;
DD6D    BPL     &DDCA   ;if not set the not a speech interrupt so DDCA

DD6F    LDA     #&20    ;clear interrupt flag
DD71    LDX     #&00    ;
```

```
DD73    STA     &FE4D   ;
DD76    STX     &FE49   ;and zer0 hi byte of T2 Timer
DD79    LDX     #&08    ;&FB=8
DD7B    STX     &FB     ;
DD7D    JSR     &E45B   ;and examine buffer 8
DD80    ROR     &02D7   ;shift carry into bit 7
DD83    BMI     &DDC9   ;and if set buffer is empty so exit
DD85    TAY             ;else Y=A
DD86    BEQ     &DD8D   ;
DD88    JSR     &EE6D   ;control speech chip
DD8B    BMI     &DDC9   ;if negative exit
DD8D    JSR     &E460   ;else get a byte from buffer
DD90    STA     &F5     ;store it to indicate speech or file rom
DD92    JSR     &E460   ;get another byte
DD95    STA     &F7     ;store it
DD97    JSR     &E460   ;and another
DD9A    STA     &F6     ;giving address  to be accessed in paged ROM
DD9C    LDY     &F5     ;Y=&F5
DD9E    BEQ     &DDBB   ;and if =0 then DDBB
DDA0    BPL     &DDB8   ;else if +ve DDB8
DDA2    BIT     &F5     ;if bit 6 of F5 =1 (&F5)>&40
DDA4    BVS     &DDAB   ;then DDAB
DDA6    JSR     &EEBB   ;else continue for more speech processing
DDA9    BVC     &DDB2   ;if bit 6 clear then DDB2
DDAB    ASL     &F6     ;else double address in &F6/7
DDAD    ROL     &F7     ;
DDAF    JSR     &EE3B   ;and call EE3B
DDB2    LDY     &0261   ;get speech enable/disable flag into Y
DDB5    JMP     &EE7F   ;and JMP to EE7F

DDB8    JSR     &EE7F   ;Call EE7F
DDBB    LDY     &F6     ;get address pointer in Y
DDBD    JSR     &EE7F   ;
DDC0    LDY     &F7     ;get address pointer high in Y
DDC2    JSR     &EE7F   ;
DDC5    LSR     &FB     ;
DDC7    BNE     &DD7D   ;
DDC9    RTS             ;and exit
                        ;


*************************************************************************
*                                                                       *
*       SYSTEM INTERRUPT 6 10mS Clock                                   *
*                                                                       *
*************************************************************************

DDCA    BCC     &DE47   ;bit 6 is in carry so if clear there is no 6 int
                        ;so go on to DE47
DDCC    LDA     #&40    ;Clear interrupt 6
DDCE    STA     &FE4D   ;

;UPDATE timers routine, There are 2 timer stores &292-6 and &297-B
;these are updated by adding 1 to the current timer and storing the
;result in the other, the direction of transfer being changed each
;time of update.  This ensures that at least 1 timer is valid at any call
;as the current timer is only read.  Other methods would cause inaccuracies
;if a timer was read whilst being updated.

DDD1    LDA     &0283   ;get current system clock store pointer (5,or 10)
DDD4    TAX             ;put A in X
DDD5    EOR     #&0F    ;and invert lo nybble (5 becomes 10 and vv)
DDD7    PHA             ;store A
DDD8    TAY             ;put A in Y

                        ;Carry is always set at this point
```

```
DDD9    LDA     &0291,X ;get timer value
DDDC    ADC     #&00    ;update it
DDDE    STA     &0291,Y ;store result in alternate
DDE1    DEX             ;decrement X
DDE2    BEQ     &DDE7   ;if 0 exit
DDE4    DEY             ;else decrement Y
DDE5    BNE     &DDD9   ;and go back and do next byte

DDE7    PLA             ;get back A
DDE8    STA     &0283   ;and store back in clock pointer (i.e. inverse previous
                        ;contents)
DDEB    LDX     #&05    ;set loop pointer for countdown timer
DDED    INC     &029B,X ;increment byte and if
DDF0    BNE     &DDFA   ;not 0 then DDFA
DDF2    DEX             ;else decrement pointer
DDF3    BNE     &DDED   ;and if not 0 do it again
DDF5    LDY     #&05    ;process EVENT 5 interval timer
DDF7    JSR     &E494   ;

DDFA    LDA     &02B1   ;get byte of inkey countdown timer
DDFD    BNE     &DE07   ;if not 0 then DE07
DDFF    LDA     &02B2   ;else get next byte
DE02    BEQ     &DE0A   ;if 0 DE0A
DE04    DEC     &02B2   ;decrement 2B2
DE07    DEC     &02B1   ;and 2B1

DE0A    BIT     &02CE   ;read bit 7 of envelope processing byte
DE0D    BPL     &DE1A   ;if 0 then DE1A
DE0F    INC     &02CE   ;else increment to 0
DE12    CLI             ;allow interrupts
DE13    JSR     &EB47   ;and do routine sound processes
DE16    SEI             ;bar interrupts
DE17    DEC     &02CE   ;DEC envelope processing byte back to 0

DE1A    BIT     &02D7   ;read speech buffer busy flag
DE1D    BMI     &DE2B   ;if set speech buffer is empty, skip routine
DE1F    JSR     &EE6D   ;update speech system variables
DE22    EOR     #&A0    ;
DE24    CMP     #&60    ;
DE26    BCC     &DE2B   ;if result >=&60 DE2B
DE28    JSR     &DD79   ;else more speech work

DE2B    BIT     &D9B7   ;set V and C
DE2E    JSR     &DCA2   ;check if ACIA needs attention
DE31    LDA     &EC     ;check if key has been pressed
DE33    ORA     &ED     ;
DE35    AND     &0242   ;(this is 0 if keyboard is to be ignored, else &FF)
DE38    BEQ     &DE3E   ;if 0 ignore keyboard
DE3A    SEC             ;else set carry
DE3B    JSR     &F065   ;and call keyboard
DE3E    JSR     &E19B   ;check for data in user defined printer channel
DE41    BIT     &FEC0   ;if ADC bit 6 is set ADC is not busy
DE44    BVS     &DE4A   ;so DE4A
DE46    RTS             ;else return
                        ;

**********************************************************************
*                                                                    *
*       SYSTEM INTERRUPT 4 ADC end of conversion                     *
*                                                                    *
**********************************************************************


DE47    ROL             ;put original bit 4 from FE4D into bit 7 of A
DE48    BPL     &DE72   ;if not set DE72
```

```
DE4A    LDX     &024C   ;else get current ADC channel
DE4D    BEQ     &DE6C   ;if 0 DE6C
DE4F    LDA     &FEC2   ;read low data byte
DE52    STA     &02B5,X ;store it in &2B6,7,8 or 9
DE55    LDA     &FEC1   ;get high data byte
DE58    STA     &02B9,X ;and store it in hi byte
DE5B    STX     &02BE   ;store in Analogue system flag marking last channel
DE5E    LDY     #&03    ;handle event 3 conversion complete
DE60    JSR     &E494   ;

DE63    DEX             ;decrement X
DE64    BNE     &DE69   ;if X=0
DE66    LDX     &024D   ;get highest ADC channel preseny
DE69    JSR     &DE8F   ;and start new conversion
DE6C    LDA     #&10    ;reset interrupt 4
DE6E    STA     &FE4D   ;
DE71    RTS             ;and return


*************************************************************************
*                                                                       *
*       SYSTEM INTERRUPT 0 Keyboard                                     *
*                                                                       *
*************************************************************************

                        ;
DE72    ROL             ;get original bit 0 in bit 7 position
DE73    ROL             ;
DE74    ROL             ;
DE75    ROL             ;
DE76    BPL     &DE7F   ;if bit 7 clear not a keyboard interrupt
DE78    JSR     &F065   ;else scan keyboard
DE7B    LDA     #&01    ;A=1
DE7D    BNE     &DE6E   ;and off to reset interrupt and exit

DE7F    JMP     &DCF3   ;


************** exit routine ***********************************************

DE82    PLA             ;restore registers
DE83    TAY             ;
DE84    PLA             ;
DE85    TAX             ;
DE86    PLA             ;
DE87    STA     &FC     ;store A


*************************************************************************
*                                                                       *
*       IRQ2V default entry                                             *
*************************************************************************

DE89    LDA     &FC     ;get back original value of A
DE8B    RTI             ;and return to calling routine



*************************************************************************
*                                                                       *
*       OSBYTE 17 Start conversion                                      *
*                                                                       *
*************************************************************************

                        ;
```

```
DE8C    STY     &02BE   ;set last channel to finish conversion
DE8F    CPX     #&05    ;if X<4 then
DE91    BCC     &DE95   ;DE95
DE93    LDX     #&04    ;else X=4

DE95    STX     &024C   ;store it as current ADC channel
DE98    LDY     &024E   ;get conversion type
DE9B    DEY             ;decrement
DE9C    TYA             ;A=Y
DE9D    AND     #&08    ;and it with 08
DE9F    CLC             ;clear carry
DEA0    ADC     &024C   ;add to current ADC
DEA3    SBC     #&00    ;-1
DEA5    STA     &FEC0   ;store to the A/D control panel
DEA8    RTS             ;and return
                        ;

DEA9    LDA     #&C3    ;point to start of string @&C300
DEAB    STA     &FE     ;store it
DEAD    LDA     #&00    ;point to lo byte
DEAF    STA     &FD     ;store it and start loop@

DEB1    INY             ;print character in string
DEB2    LDA     (&FD),Y ;pointed to by &FD/E
DEB4    JSR     OSASCI  ;print it expanding Carriage returns
DEB7    TAX             ;store A in X
DEB8    BNE     &DEB1   ;and loop again if not =0
DEBA    RTS             ;else exit


********** OSBYTE 129 TIMED ROUTINE ****************************
;ON ENTRY TIME IS IN X,Y

DEBB    STX     &02B1   ;store time in INKEY countdown timer
DEBE    STY     &02B2   ;which is decremented every 10ms
DEC1    LDA     #&FF    ;A=&FF
DEC3    BNE     &DEC7   ;goto DEC7



*******************************************************************************
*******************************************************************************
**                                                                           **
**      OSRDCH Default entry point                                           **
**                                                                           **
**      RDCHV entry point       read a character                             **
**                                                                           **
*******************************************************************************
*******************************************************************************

DEC5    LDA     #&00    ;A=0

DEC7    STA     &E6     ;store entry value of A
DEC9    TXA             ;store X and Y
DECA    PHA             ;
DECB    TYA             ;
DECC    PHA             ;
DECD    LDY     &0256   ;get *EXEC file handle
DED0    BEQ     &DEE6   ;if 0 (not allocated) then DEE6
DED2    SEC             ;set carry
DED3    ROR     &EB     ;set bit 7 of CFS active flag to prevent  clashes
DED5    JSR     OSBGET  ;get a byte from the file
DED8    PHP             ;push processor flags to preserve carry
DED9    LSR     &EB     ;restore &EB
DEDB    PLP             ;get back flags
DEDC    BCC     &DF03   ;and if carry clear, character found so exit via DF03
```

```
DEDE    LDA     #&00    ;else A=00 as EXEC file empty
DEE0    STA     &0256   ;store it in exec fil;e handle
DEE3    JSR     OSFIND  ;and close file via OSFIND

DEE6    BIT     &FF     ;check ESCAPE flag if bit 7 set Escape pressed
DEE8    BMI     &DF00   ;so off to DF00
DEEA    LDX     &0241   ;else get current input buffer number
DEED    JSR     &E577   ;get a byte from keyboard buffer
DEF0    BCC     &DF03   ;and exit if valid character found

DEF2    BIT     &E6     ;(E6=0 or FF)
DEF4    BVC     &DEE6   ;if entry was OSRDCH not timed keypress go back and
                        ;do it again i.e. perform GET function
DEF6    LDA     &02B1   ;else check timers
DEF9    ORA     &02B2   ;
DEFC    BNE     &DEE6   ;and if not zero go round again
DEFE    BCS     &DF05   ;else exi
```