

User Callable Kernal Routines

Name	Address	Function
	Hex Decimal	
ACPTR	\$FFA5 65445	Input byte from serial port.
CHKIN	\$FFC6 65478	Open channel for input.
CHKOUT	\$FFC9 65481	Open channel for output.
CHRIN	\$FFCF 65487	Input character from channel.
CHROUT	\$FFD2 65490	Output character to channel.
CIOUT	\$FFAB 65448	Output byte to serial port.
LINT	\$FF81 65409	Initialize screen editor.
CLALL	\$FFE7 65511	Close all channels and files.
CLOSE	\$FFC3 65475	Close a specified logical file.
CLRCHN	\$FFCC 65484	Close input and output channels.
GETIN	\$FFE4 65508	Get character from keyboard buffer.
IOBASE	\$FFF3 65523	Return base address of I/O devices.
IOINIT	\$FF84 65412	Initialize input/output.
LISTEN	\$FFB1 65457	Command devices on serial bus to LISTEN.
LOAD	\$FFD5 65493	Load RAM from a device.
MEMBOT	\$FF9C 65436	Read/set bottom of memory.
MEMTOP	\$FF99 65433	Read/set top of memory.
OPEN	\$FFC0 65472	Open a logical file.
PLOT	\$FFF0 65520	Read/set X,Y cursor position.
RAMTAS	\$FF87 65415	Initialize RAM, reset tape buffer.
RDTIM	\$FFDE 65502	Read realtime clock.
READST	\$FFB7 65463	Read I/O status word.
RESTOR	\$FF8A 65418	Restore I/O default vectors.
SAVE	\$FFDE 65496	Save RAM to device.
SCNKEY	\$FF9F 65439	Scan keyboard.
SCREEN	\$FFED 65517	Return X,Y organization of screen.
SECOND	\$FF93 65427	Send secondary address after LISTEN.
SETLFS	\$FFBA 65466	Set logical, first, and second address.
SETMSG	\$FF90 65424	Control Kernal messages.
SETNAM	\$FFBD 65469	Set filename.
SETTIM	\$FFDB 65499	Set realtime clock.
SETTMO	\$FFA2 65442	Set time-out on serial bus.
STOP	\$FFE1 65505	Check for STOP key.
TALK	\$FFB4 65460	Command serial bus device to TALK.
TKSA	\$FF96 65430	Send secondary address after TALK.
UDTIM	\$FFEA 65514	Increment realtime clock.
UNLSN	\$FFAE 65454	Command serial bus to UNLISTEN.
UNTLK	\$FFAB 65451	Command serial bus to UNTALK.
VECTOR	\$FF8D 65421	Read/set vectored I/O.

Here is a brief summary of each routine with examples:

ACPTR is used to get data off the serial bus. TALK and TKSA must be called first.

```
;Get a byte from the serial bus.  
JSR ACPTR  
STA $0800  
;This example only shows the end result; call TALK and TKSA first.
```

CHKIN is used to define any OPENed file as an input file. OPEN must be called first.

```
;Define logical file #2 as an input channel.  
LDX #2  
JSR CHKIN  
;The X register designates which file #.
```

CHKOUT. Just like CHKIN, but it defines the file for output. OPEN must be called first.

```
;Define logical file #4 as an output file.  
LDX #4  
JSR CHKOUT  
;Once again the X register defines the file #.
```

CHRIN will get a character from the current input device. Calling OPEN and CHKIN can change the input device.

```
;Store a typed string to the screen.  
LDY #$00  
LOOP JSR CHRIN  
STA $0800,Y  
INY  
CMP #$0D  
BNE LOOP  
RTS  
;This example is like an INPUT statement. Try running it.
```

CHROUT. Load the accumulator with your number and call. OPEN and CHKOUT will change the output device.

```
;Duplicate the command of CMD 4:PRINT "A";  
LDX #4  
JSR CHKOUT  
LDA #'A  
JSR CHROUT  
RTS  
;The letter A is printed to the screen; call OPEN first for the printer.
```

CIOUT will send data to the serial bus. LISTEN and SECOND must be called first. Call UNLSN to finish up neatly.

```
;Send the letter X to the serial bus.  
LDA #'X
```

```
JSR CIOUT
RTS
;The accumulator is used to transfer the data.
```

CINT resets the 6567 video controller chip and the screen editor.

```
;Reset the 6567 chip and the 6566 VIC chip.
JSR CINT
RTS
;Basically, just like pressing the STOP and RESTORE keys.
```

CLALL really does what its name implies-it closes all files and resets all channels.

```
;Close all files.
JSR CLALL
RTS
;The CLRCHN routine is called automatically.
```

CLOSE. This routine will CLOSE any logical file that has been OPENed.

```
Close logical file #2.
LDA #2
JSR CLOSE
;The accumulator designates the file #.
```

CLRCHN resets all channels and I/O registers - the input to keyboard and the output to screen.

```
;Restore default values to I/O devices.
JSR CLRCHN
RTS
;The accumulator and the X register are altered.
```

GETIN will get one piece of data from the input device. OPEN and CHKIN can be used to change the input device.

```
;Wait for a key to be pressed.
WAIT JSR GETIN
    CMP #0
    BEQ WAIT
;If the serial bus is used, then all registers are altered.
```

IOBASE returns the low and high bytes of the starting address of the I/O devices in the X and Y registers.

```
;Set the Data Direction Register of the user port to 0 (input).
JSR IOBASE
STX POINT
STY POINT+1
```

```
LDY #2
LDA #0
STA (POINT),Y
;POINT is a zero-page address used to access the DDR indirectly.
```

IOINIT initializes all I/O devices and routines. It is part of the system's powering-up routine.

```
;Initialize all I/O devices.
JSR IOINIT
RTS
;All registers are altered.
```

LISTEN will command any device on the serial bus to receive data.

```
;Command device #8 to listen.
LDA #8
JSR
LISTEN
;The accumulator designates the device #.
```

LOAD. The computer will perform either the LOAD or the VERIFY command. If the accumulator is a 1, then LOAD; if 0, then verify.

```
;Load a program into memory.
LDA #$08
LDX #$02
LDY #$00
JSR SETLFS
LDA #$04
LDX #L,NAME
LDY #H,NAME
JSR SETNAM
LDA #$00
LDY #$20
JSR LOAD
RTS
NAME .BY 'FILE'
;Program 'FILE' will be loaded into memory starting at 8192 decimal, X being the low byte and
Y being the high byte for the load.
```

MEMBOT. If the carry bit is set, then the low byte and the high byte of RAM are returned in the X and Y registers. If the carry bit is clear, the bottom of RAM is set to the X and Y registers.

```
;Move bottom of memory up one page.
SEC
JSR MEMBOT
INY
CLC
```

```
JSR MEMBOT
RTS
;The accumulator is left alone.
```

MEMTOP. Same principle as MEMBOT, except the top of RAM is affected.

```
;Protect 1K of memory from BASIC.
SEC
JSR MEMTOP
DEY
CLC
JSR MEMTOP
;The accumulator is left alone.
```

OPEN. After SETLFS and SETNAM have been called, you can OPEN a logical file.

```
;Duplicate the command OPEN 15,8,15,'I/O'
LDA #3
LDX #L,NAME
LDY #H,NAME
JSR SETNAM
LDA #15
LDX #8
LDY #15
JSR SETLFS
JSR OPEN
RTS
NAME .BY 'I/O'
;OPEN opens the current name file with the current LFS.
```

PLOT. If the carry bit of the accumulator is set, then the cursor X,Y is returned in the Y and X registers. If the carry bit is clear, then the cursor is moved to X,Y as determined by the Y and X registers.

```
;Move cursor to row 12, column 20 (12,20).
LDX #12
LDY #20
CLC
JSR PLOT
;The cursor is now in the middle of the screen.
```

RAMTAS is used to test RAM, reset the top and bottom of memory pointers, clear \$0000 to \$0101 and \$0200 to \$03FF, and set the screen memory to \$0400.

```
;Do RAM test.
JSR RAMTAS
RTS
;All registers are altered.
```

RDTIM. Locations 160-162 are transferred, in order, to the Y and X registers and the accumulator.

;Store system clock to screen.

JSR RDTIM

STA 1026

STX 1025

STY 1024

;The system clock can be translated as hours/minutes/ seconds.

READST. When called, READST returns the status of the I/O devices. Any error code can be translated as operator error.

;Check for read error.

JSR READST

CMP #16

BEQ ERROR

;In this case, if the accumulator is 16, a read error occurred.

SCREEN returns the number of columns and rows the screen has in the X and Y registers.

;Determine the screen size.

JSR SCREEN

STX MAXCOL

STY MAXROW

RTS

;SCREEN allows further compatibility between the 64, the VIC-20, and future versions of the 64.

SECOND. After LISTEN has been called, a SECONDAry address may be sent.

;Address device #8 with secondary address #15.

LDA #8

JSR LISTEN

LDA #15

JSR SECOND

;The accumulator designates the address number.

SETLFS stands for SET Logical address, File address, and Secondary address. After SETLFS is called, OPEN may be called.

;Set logical file #1, device #8, secondary address of 15.

LDA #1

LDX #8

LDY #15

JSR SETLFS

;If OPEN is called, the command will be OPEN 1,8,15.

SETMSG. Depending on the accumulator, either error messages, control messages, or neither is printed.

```
;Turn on control messages.  
LDA #$40  
JSR SETMSG  
RTS  
;A 128 is for error messages; a zero, for turning both off.
```

SETNAM. In order to access the OPEN, LOAD, or SAVE routines, SETNAM must be called first.

```
;SETNAM will prepare the disk drive for'FILE#1'.  
LDA #6  
LDX #L,NAME  
LDY #H,NAME  
JSR SETNAM  
NAME.BY 'FILE#1'  
;Accumulator is file length, X is low byte, and Y is high byte.
```

SETTIM is the opposite of RDTIM: it SETs the system clock instead of ReaDing it.

```
;Set system clock to 10 minutes =3600 jiffies.  
LDA #0  
LDX #L,3600  
LDY #H,3600  
JSR SETTIM  
;This allows very accurate timing for many things.
```

SETTMO is used only with an IEEE add-on card to access the serial bus.

```
;Disable time-outs on serial bus.  
LDA #0  
JSR SETTMO  
;To enable time-outs, set the accumulator to a 128 and call SETTMO.
```

STOP will set the Z flag of the accumulator if the STOP key was pressed.

```
;Check for STOP key being pressed.  
WAIT JSR STOP  
BNE WAIT  
RTS  
;STOP must be called if the STOP key is to remain functional.
```

TALK. This routine will command a device on the serial bus to send data.

```
;Command device #8 to TALK.  
LDA #8
```

```
JSR TALK
RTS
;The accumulator designates the file number.
```

TKSA is used to send a secondary address for a TALK device. TALK must be called first.

```
;Signal device #4 to talk with command #7.
LDA #4
JSR TALK
LDA #7
JSR TKSA
RTS
;This example will tell the printer to print in uppercase.
```

UDTIM. If you are using your own interrupt system, you can update the system clock by calling UDTIM.

```
;Update the system clock.
JSR UDTIM
RTS
;It is useful to call UDTIM before calling STOP.
```

UNLSN commands all devices on the serial bus to stop receiving data.

```
;Command the serial bus to UNLiSteN.
JSR UNLSN
RTS
;The serial bus can now be used for other things.
```

UNTLK. All devices previously set to TALK will stop sending data.

```
;Command serial bus to stop sending data.
JSR UNTLK
RTS
;Sending UNTLK commands all talking devices to get off the serial bus.
```

VECTOR. If the carry bit of the accumulator is set, the start of a list of the current contents of the RAM vectors is returned in the X and Y registers. If the carry bit is clear, there the user list pointed to by the X and Y registers is transferred to the system RAM vectors.

```
;Change the input routines to new system.
SEC
JSR VECTOR
LDA #L,MYINP
STA USER+10
LDA #H,MYINP
STA USER+11
LDX #L,USER
```



```
LDY #H,USER
CLC
JSR VECTOR
RTS
USER .DE 26
```

;The new input list can start anywhere. USER is the location for temporary strings, and 35-36 is the utility pointer area.

Error Codes

If an error occurs during a Kernal routine, then the carry bit of the accumulator is set and the error code is returned in the accumulator.

Number	Meaning
0	Routine ended by the STOP key.
1	Too many files open.
2	File already open.
3	File not open.
4	File not found.
5	Device not present.
6	File is not an input file.
7	File is not an output file.
8	File name is missing.
9	Illegal device number
240	Top-of-memory change RS-232 buffer allocation