Sim68K Text I/O

TRAP #15 is used for I/O. Put the task number in D0.

Task	
0	Display n characters of string at (A1), n is D1.W (stops on NULL or max 255) with CR, LF. (see task 13)
1	Display n characters of string at (A1), n is D1.W (max 255) without CR, LF. (see task 14)
2	Read string from keyboard and store at (A1), NULL (0) terminated, length returned in D1.W (max 80)
3	Display signed number in D1.L in decimal in smallest field. (see task 15 & 20)
4	Read a number from the keyboard into D1.L.
5	Read single ASCII character from the keyboard into D1.B.
6	Display single ASCII character in D1.B.
7	Check for keyboard input. Set D1.B to 1 if keyboard input is pending, otherwise set to 0. Use task 2 or 5 to read pending key.
8	Return time in hundredths of a second since midnight in D1.L.
9	Terminate the program. (Halts the simulator)
10	Print the NULL terminated string at (A1) to the default printer. (Always send a Form Feed character to end printing. See below.)
11	Cursor Position, Set/Get or Clear Screen Set cursor position: The high byte of D1.W holds the COL number (0-255), The low byte holds the ROW number (0-128). 0,0 is top left. Out of range coordinates are ignored. Get cursor position: Set D1.W to $00FF Returns COL number, ROW number in high and low byte of D1.W respectively. Clear Screen : Set D1.W to $FF00. (Task 95 supports exact pixel placement of text)
12	Keyboard Echo. D1.B = 0 to turn off keyboard echo. D1.B = non zero to enable. (default). Echo is restored on 'Reset' or when a new file is loaded.
13	Display the NULL terminated string at (A1) with CR, LF.
14	Display the NULL terminated string at (A1) without CR, LF.
15	Display the unsigned number in D1.L converted to number base (2 through 36) contained in D2.B. For example, to display D1.L in base16 put 16 in D2.B Values of D2.B outside the range 2 to 36 inclusive are ignored.
16	Adjust display properties D1.B = 0 to turn off the display of the input prompt. D1.B = 1 to turn on the display of the input prompt. (default) D1.B = 2 do not display a line feed when Enter pressed during Trap task #2 input D1.B = 3 display a line feed when Enter key pressed during Trap task #2 input (default) Other values of D1 reserved for future use. Input prompt display is enabled by default and by 'Reset' or when a new file is loaded.
17	Combination of Trap codes 14 & 3. Display the NULL terminated string at (A1) without CR, LF then Display the decimal number in D1.L.
18	Combination of Trap codes 14 & 4. Display the NULL terminated string at (A1) without CR, LF then Read a number from the keyboard into D1.L.
19	Returns current state of up to 4 specified keys or returns key scan code. Pre: D1.L = four 1-byte key codes Post: D1.L contains four 1-byte Booleans. $FF = corresponding key is pressed, $00 = corresponding key not pressed. Pre: D1.L = $00000000 Post: D1.L upper word contains key code of last key up. D1.L lower word contains key code of last key down. Example: `MOVE.B #19,D0`

	`MOVE.L #'A'<<24+'S'<<16+'D'<<8+'F',D1 ; check for keypress (a,s,d,f)`
	`TRAP #15`
	`BTST.L #24,D1 ; test for 'a'`
	`IF <NE> THEN ; if 'a'`
	`{a code}`
	`ENDI`
	`BTST.L #16,D1 ; test for 's'`
	`IF <NE> THEN ; if 's'`
	`... etc`

20	Display signed number in D1.L in decimal in field D2.B columns wide.
21	Set font properties where: D1.L is color as 0x00BBGGRR BB is amount of blue from 0x00 to 0xFF GG is amount of green from 0x00 to 0xFF RR is amount of red from 0x00 to 0xFF D2.L Low word is style by bits 0 = off, 1 = on bit0 is Bold bit1 is Italic bit2 is Underline bit3 is StrikeOut High word (low byte) is Size in points (High word = 0, keep current font) 8, 9, 10, 11, 12, 14, 16, 18 (not all sizes are valid for all fonts) Font sizes in multiples of valid sizes (size * n) results in a scaled appearance. For example: in Fixedsys font sizes of 9*2, 9*3, ..9*n will result in larger characters but the characters will have pixelated edges. High word (high byte) is Font 1 - Fixedsys (size: 9) 2 - Courier (sizes: 10, 12, 15) 3 - Courier New (sizes 8,9,10,11,12,14,16,18) 4 - Lucida Console (sizes 8,9,10,11,12,14,16,18) 5 - Lucida Sans Typewriter (sizes 8,9,10,11,12,14,16,18) 6 - Consolas (sizes 8,9,10,11,12,14,16,18) 7 - Terminal (sizes 9,12,14) Example: D2.L = $01090005 is Fixedsys, 9 point, Bold Underline
22	Read char at Row,Col of text screen. Pre: D1.L = High 16 bits = Row Low 16 bits = Column Post: D1.B contains ASCII code of character.
23	Delay n/100 of a second Pre: D1.L = n as unsigned number 0 through $FFFFFFFF Delay releases CPU which reduces power consumption.
24	Text I/O control Pre: D1.L = 0, Enable simulator shortcut keys. (default) D1.L = 1, Disable simulator shortcut keys. All key codes are made available for 68000 program read using task 19. all other values reserved. Shortcuts are restored by Rewind or Reload.
25	Scroll Text Rectangle Pre: D1.L = High 16 bits = Top row (0 to 128) Low 16 bits = Left column (0 to 255) D2.L = High 16 bits = Height in rows (Top + Height max 128) Low 16 bits = Width in columns (Left + Width max 255) D3.W = 0 scroll up D3.W = 1 scroll down D3.W = 2 scroll left D3.W = 3 scroll right all other values reserved

Task numbers 0 - 9 and 11 - 12 are compatible with the Teesside simulator.

The following control characters are supported. The labels shown (BEL, BS, HT, etc.) are not predefined in EASy68K. Placing the code in your program will equate the labels with the control characters.

```
BEL EQU  $07   Bell
BS  EQU  $08   Backspace
HT  EQU  $09   Tab (horizontal 5 characters)
LF  EQU  $0A   Line Feed
```

```
VT   EQU  $0B   Vertical tab (4 lines)
FF   EQU  $0C   Form Feed (Always end printing with a Form Feed.)
CR   EQU  $0D   Carriage Return
```

Sim68K Environment

TRAP #15 is used for I/O. Put the task number in D0.

Task	
30	Clear the Cycle Counter
31	Return the Cycle Counter in D1.L Zero is returned if the cycle count exceeds 32 bits.
32	Hardware/Simulator D1.B = 00, Display hardware window D1.B = 01, Return address of 7-segment display in D1.L D1.B = 02, Return address of LEDs in D1.L D1.B = 03, Return address of toggle switches in D1.L D1.B = 04, Return Sim68K version number in D1.L Version 3.9.10 is returned as 0003090A D1.B = 05, Enable exception processing. Exceptions will be directed to the appropriate 68000 exception vector. This has the same effect as checking Enable Exceptions in the Options menu. D1.B = 06, Set Auto IRQ D2.B = 00, disable all Auto IRQs or Bit 7 = 0, disable individual IRQ Bit 7 = 1, enable individual IRQ Bits 6-0, IRQ number 1 through 7 D3.L, Auto Interval in milliseconds, 1000 = 1 second D1.B = 07, Return address of push button switches in D1.L

33	Get/Set Output Window
	D1.L High 16 bits = Set width in pixels, min = 640
	Low 16 bits = Set height in pixels, min = 480
	D1.L = 0, Get current output window resolution in D1.L as
	High 16 bits = Width in pixels
	Low 16 bits = Height in pixels
	D1.L = 1, Set windowed mode
	D1.L = 2, Set full screen mode
	
	Example:
	`MOVE.B #33,D0`
	`MOVE.L #1024*$10000+768,D1  Set screen to 1024 x 768`
	`TRAP #15`

Sim68K Serial I/O

TRAP #15 is used for I/O. Put the task number in lower 8 bits of D0.

The success of the Serial I/O calls is returned in D0.W as follows:
0 = Success
1 = Invalid port identifier (PID)
2 = Error
3 = Port not initialized (Tasks 41, 42, 43 only)
4 = Timeout (Tasks 42, 43 only)

A maximum of 16 communications ports are supported.

Task	
	Initialize communications port. This must be called once for each port before any other serial I/O function. The port defaults to 9600 baud, 8 data bits, no parity, one stop bit. Pre: High 16 bits of D0 contain port identifier (PID), (0 through 15). The PID is used to identify the port in all other serial I/O tasks. It is not used to specify the COM port number, (i.e. A PID of 4 does not mean the same thing as COM4)

40	(A1) address of serial port name as null terminated string (e.g. PORT DC.B 'COM4',0)
	Post:
	D0.W is 0 on success, 1 on invalid PID, 2 on error
	
	Example:
	`; initialize serial port`
	`move.l #1<<16+40,d0 ; PID 1, task 40`
	`lea PORT,A1 ; name of port`
	`trap #15`
	
	Visit www.EASy68K.com for examples.

41	Set port parameters. Pre: High 16 bits of D0 contain port identifier (PID) D1.L Bits 0-7 (D1.B) Baud rate: 0=9600(default), 1=110, 2=300, 3=600, 4=1200, 5=2400, 6=4800, 7=9600 8=19200, 9=38400, 10=56000, 11=57600, 12=115200, 13=128000, 14=256000. Bits 8-9 Parity: 0=no, 1=odd, 2=even, 3=mark Bits 10-11 Number of data bits: 0=8 bits, 1=7 bits, 2=6 bits Bit 12 Stop bits: 0=1 stop bit, 1=2 stop bits The higher bits of D1.L are reserved for future use. Post: D0.W is 0 on success, 1 on invalid PID, 2 on error, 3 on port not initialized
42	Receive string. Pre: High 16 bits of D0 contain port identifier (PID) (A1) buffer address. D1.B max number of characters to receive. The port will wait sufficient time for the requested number of bytes to be received. A timeout will occur if the number of bytes received is less than the requested number. For faster response use a smaller number in D1.B. Post: D0.W is 0 on success, 1 on invalid PID, 2 on error, 3 on port not initialized, 4 on timeout D1.B number of characters received. (A1) null terminated string of characters received.
43	Send string. Pre: High 16 bits of D0 contain port identifier (PID) (A1) buffer address. D1.B number of characters to send. The port will wait sufficient time for the specified number of bytes to be sent. A timeout will occur if the number of bytes sent is less than the requested number. Post: D0.W is 0 on success, 1 on invalid PID, 2 on error, 3 on port not initialized, 4 on timeout D1.B number of characters sent.

Sim68K File I/O

TRAP #15 is used for I/O. Put the task number in D0.

The success of the file handling calls is returned in D0.W as follows:
0 = success
1 = EOF encountered
2 = Error
3 = File is Read Only (Task 51 & 59 only)

A maximum of 8 files may be open at any one time.

Task	
50	Close all files. It is recommended to use this at the start of any program using file handling.
51	Open existing file. Pre: (A1) null terminated file name. Post: D1.L contains the File-ID (FID).
52	Open new file.. As above except the file is created if not found. If the file exists it will be overwritten.
53	Read file Pre: File-ID in D1.L as returned from 51 or 52, (A1) buffer address, D2.L number of bytes to read. Post: D2.L holds number of bytes actually read, EOF may cause a shortened read.
54	Write file As above except D2.L holds number of bytes to write (unaltered upon return).
55	Position file Sets the file position where the next read/write will take place. Files begin at byte 0. Pre: File-ID in D1.L as returned from 51 or 52, D2.L the file position to be set.
56	Close file Pre: File-ID in D1.L as returned from 51 or 52.
57	Delete file. Pre: (A1) null terminated file name.
58	Display File Dialog. Pre: D1 = 0 to display 'Open' dialog D1 = 1 to display 'Save' dialog (A1) Points to a null terminated string that will be used as the request title string (max 256) or A1 = $00000000 to use the default 'Open' or 'Save' depending on D1 (A2) Points to a null terminated string (max 256) containing semicolon separated list of file extensions for use in dialog, e.g. '*.txt;*.pcb',0 or A2 = $00000000 for all file types. (A3) Points to a buffer of sufficient size (max 256) to hold the zero terminated full file path and name. On calling, if this contains a file path and name this will be used as the original file path and name. If the user exits via the cancel button this buffer will remain unchanged. Post: D1 = 0 if user cancelled or just closed D1 = 1 if user hit open/save
59	File operations. Pre: (A1) null terminated file name. D1.L = 0 does file exist? Post: D0.W 0 = file exists 2 = Error 3 = file exists and is Read Only Other values of D1 reserved for future use.

Sim68K Peripheral I/O

TRAP #15 is used for I/O. Put the task number in D0.

The output screen resolution is adjustable via Task 33. The top left corner of the window is position 0,0

Task	
60	Enable/Disable mouse IRQ An IRQ is created when a mouse button is pressed or released in the output window. D1.W High Byte = IRQ level (1-7), 0 to turn off D1.W Low Byte = 1 in the corresponding bit to indicate which mouse event triggers IRQ where: Bit2 = Move, Bit1 = button Up, Bit0 = button Down (Example D1.W = $0107, Enable mouse IRQ level 1 for Move, button Up and button Down) (Example D1.W = $0002, Disable mouse IRQ for button Up)

61	Mouse Read Pre: D1.B = 00 to read current state of mouse = 01 to read mouse up state = 02 to read mouse down state Post: The mouse data is contained in the following registers: D0 as bits = Ctrl, Alt, Shift, Double, Middle, Right, Left Left is Bit0, Right is Bit 1 etc. 1 = true, 0 = false Shift, Alt, Ctrl represent the state of the corresponding keys. D1.L = 16 bits Y, 16 bits X in pixel coordinates. (0,0 is top left)
62	Enable/Disable keyboard IRQ An IRQ is created when a key is pressed or released in the output window. D1.W High Byte = IRQ level (1-7), 0 to turn off D1.W Low Byte = 1 in the corresponding bit to indicate which keyboard event triggers IRQ where: Bit1 = key Up, Bit0 = key Down (Example D1.W = $0103, Enable keyboard IRQ level 1 for key Up and key Down) (Example D1.W = $0002, Disable keyboard IRQ for key Up) Read the last key down or key up with trap task #19.

Sim68K Sound

TRAP #15 is used for I/O. Put the task number in D0.

Task	
70	Play the WAV file using the standard player. Only one sound may be played at a time. Pre: (A1) null terminated path address. Invalid file names are ignored. Post: D0.W = 0 if player is busy, sound is not played. D0.W = non zero if sound played.
71	Load a WAV file into sound memory (not 68000 memory). Pre: (A1) null terminated path address. Invalid file names are ignored. D1.B reference number to use for sound 0-255. A maximum of 256 sounds may be loaded at any one time. Reusing a reference number will replace the current sound.
72	Play sound from sound memory loaded with task 71 using standard player. Only one sound may be played at a time. Pre: D1.B must contain sound reference number used in task 71. Post: D0.W = 0 if player is busy, sound is not played. D0.W = non zero if sound played.
73	Play the WAV file using DirectX player, if available. Multiple sounds may be played at the same time. Pre: (A1) null terminated path address. Invalid file names are ignored. Post: D0.W = 0 if DirectX player not available, sound is not played. D0.W = non zero if sound played.
74	Load a WAV file into DirectX sound memory (not 68000 memory). A maximum of 256 sounds may be loaded at any one time. Reusing a reference number will replace the current sound. Pre: (A1) null terminated path address. Invalid file names are ignored. D1.B reference number to use for sound 0-255. Post: D0.W = 0 if DirectX player not available. D0.W = non zero if sound loaded.
75	Play sound from DirectX sound memory loaded with task 74. Pre: D1.B must contain sound reference number used in task 74. Post: D0.W = 0 if DirectX player not available, sound is not played. D0.W = non zero if sound played.
76	Control Standard player Sounds must be in memory loaded with task 71. Only one sound may be played at a time. Pre: D1.B contains sound reference number used in task 71. D2.L = 0, play sound once (this is the same as task 72) D2.L = 1, play sound in loop, returns error if sound currently playing. D2.L = 2, stop D1.B referenced sound, returns error on bad reference number D2.L = 3, stop all sounds, returns success (D1.B ignored) D2.L = other values reserved Post: D0.W = 0 on error. D0.W = non zero on success.

77	Control DirectX player, if available Sounds must be in DirectX memory loaded with task 74. Multiple sounds may be played at the same time. Pre: D1.B contains sound reference number used in task 74. D2.L = 0, play sound once (this is the same as task 75) D2.L = 1, play sound in loop. The same sound may be played multiple times. D2.L = 2, stop D1.B referenced sound, returns error on bad reference number D2.L = 3, stop all sounds (D1.B ignored) D2.L = other values reserved Post: D0.W = 0 if DirectX player not available. D0.W = non zero on success.

Sim68K Graphics

TRAP #15 is used for I/O. Put the task number in D0.

The output screen resolution is adjustable via Task 33. The top left corner of the window is position 0,0
Drawing outside the screen area is ignored.
The screen may be cleared with task 11.

Task	
80	Set pen color where D1.L is color as $00BBGGRR BB is amount of blue from $00 to $FF GG is amount of green from $00 to $FF RR is amount of red from $00 to $FF
81	Set fill color where D1.L is color as $00BBGGRR
82	Draw pixel in pen color at X,Y where X = D1.W & Y = D2.W The drawing mode is not used (see 92) . The drawing point is not moved (see 86).
83	Get pixel color at X,Y and place in D0.L where X = D1.W & Y = D2.W.
84	Draw line using pen color from X1,Y1 to X2,Y2 where X1 = D1.W, Y1 = D2.W, X2 = D3.W, Y2 = D4.W
85	Draw line using pen color to X,Y where X = D1.W & Y = D2.W
86	Move to X,Y where X = D1.W & Y = D2.W (Task 84 & 85 also move drawing point)
87	Draw rectangle defined by (Left X, Upper Y, Right X, Lower Y). where LX = D1.W, UY = D2.W, RX = D3.W, LY = D4.W The rectangle is drawn using pen color and filled using fill color.
88	Draw ellipse bounded by the rectangle (Left X, Upper Y, Right X, Lower Y) where LX = D1.W, UY = D2.W, RX = D3.W, LY = D4.W The ellipse is drawn using pen color and filled using fill color. A circle is drawn if the bounding rectangle is a square.
89	Flood Fill the area at X, Y with the fill color where X = D1.W & Y = D2.W
90	Draw unfilled rectangle defined by (Left X, Upper Y, Right X, Lower Y). where LX = D1.W, UY = D2.W, RX = D3.W, LY = D4.W The rectangle is drawn using pen color and is not filled.
91	Draw unfilled ellipse bounded by the rectangle (Left X, Upper Y, Right X, Lower Y) where LX = D1.W, UY = D2.W, RX = D3.W, LY = D4.W The ellipse is drawn using pen color and is not filled. A circle is drawn if the bounding rectangle is a square.
92	Set drawing mode which affects pen and fill colors. D1.B is mode number as: 0 - Draw color is always black, ignores pen and fill colors 1 - Draw color is always white, ignores pen and fill colors 2 - Move cursor but do not draw 3 - NOT (background color) is drawn 4 - The draw color is drawn (default) 5 - NOT (draw color) is drawn 6 - NOT (background color) OR (draw color) is drawn 7 - NOT (background color) AND (draw color) is drawn 8 - (background color) OR NOT (draw color) is drawn 9 - (background color) AND NOT (draw color) is drawn 10 - (background color) OR (draw color) is drawn 11 - (background color) NOR (draw color) is drawn 12 - (background color) AND (draw color) is drawn 13 - (background color) NOT AND (draw color) is drawn

	14 - (background color) XOR (draw color) is drawn 15 - (background color) NOT XOR (draw color) is drawn| 16 - Turn off double buffering (default) 17 - Enable double buffering. Draw on off screen buffer (Use Task 94 to display off screen buffer) * The background color is whatever is already there. * The draw color is the pen color or the fill color.
93	Set pen width where D1.B is width in pixels.
94	Repaint screen. Copies off screen buffer to visible screen. (Requires drawing mode 17 be set from task 92 above.)
95	Draw the NULL terminated string of text at (A1) to screen location X,Y where X = D1.W & Y = D2.W. The text is drawn as graphics and may not be read using trap task 22. Control characters are ignored. X,Y specifies the location of the top left corner of the text.
96	Get X,Y pen position where D1.W = X, D2.W = Y

Some color values:

```
BLACK   equ   $00000000
MAROON  equ   $00000080
GREEN   equ   $00008000
OLIVE   equ   $00008080
NAVY    equ   $00800000
PURPLE  equ   $00800080
TEAL    equ   $00808000
GRAY    equ   $00808080
RED     equ   $000000FF
LIME    equ   $0000FF00
YELLOW  equ   $0000FFFF
BLUE    equ   $00FF0000
FUCHSIA equ   $00FF00FF
AQUA    equ   $00FFFF00
LTGRAY  equ   $00C0C0C0
WHITE   equ   $00FFFFFF
```

Sim68K Network I/O

EASy68K supports both TCP and UDP communications.

TCP
TCP (Transmission Control Protocol) provides reliable delivery of data. Large messages are automatically broken into smaller IP-sized packets, transmitted, received and reassembled in order by the receiver. Packets that are lost during transmission are automatically retransmitted. The reliability of TCP message delivery can result in long delays (on the order of seconds) if packets are lost and must be retransmitted. For this reason TCP is typically not considered suitable for the delivery of real-time data. TCP establishes a connection between the two devices. This connection is maintained until broken by one or both devices. In EASy68K only one TCP connection is permitted at a time. If the other host breaks the connection an error will result when attempting to send or receive data. Refer to the error codes below.

UDP
UDP (User Datagram Protocol) is a connectionless protocol that does not guarantee data delivery. Data packets may arrive out of order, be duplicated, or not delivered at all. UDP is typically used for time sensitive applications where timely delivery of data is more important than reliability. Because of it's connectionless nature a UDP server may receive transmissions from multiple UDP clients.

EASy68K uses non-blocking sockets which means all of the network trap tasks return immediately. Check the Post register contents to determine the results.

TRAP #15 is used for I/O. Put the task number in lower 8 bits of D0.

Task	
	Configure as network Client.

100

Pre:
D1.L {31.....16}{15.....8}{7.....0}
Bits 0-7, 0 for UDP, 1 for TCP, all other values reserved
Bits 8-15, Reserved for future use
Bits 16-31, Port number
Port numbers 0-1023 are used for well-known services. Port numbers 1024-65535 may be freely used.
(A2) IP address to connect to as null terminated string (e.g. '192.168.1.100',0) or
null terminated domain name (e.g. 'www.easy68k.com',0)
Post:
D0.L is 0 on success, non zero on error
Error codes:
Bits 0-15 Low word of D0
1 - general error
2 - network initialization failed
3 - invalid socket
4 - get host by name failed
5 - bind failed
6 - connect failed
7 - port already in use
8 - domain not found
All other values reserved
Bits 16-31 High word of D0, extended error code (see below)
(A2) IP address connected to as null terminated string

Example:

```
        move.b  #100,d0                 ; create network client
        move.l  #(2048 << 16 + 0),d1    ; port 2048, UDP
        lea     serverIP,a2             ; IP to connect
        trap    #15
        ...
        ...
serverIP dc.b   '192.168.1.101',0
```

Visit www.EASy68K.com for examples.

101

Configure as network Server.
May not be configured as Server and Client at the same time.
Pre:
D1.L {31.....16}{15.....8}{7.....0}
Bits 0-7, 0 for UDP, 1 for TCP, all other values reserved
Bits 8-15, Reserved for future use
Bits 16-31, Port number
Port numbers 0-1023 are used for well-known services. Port numbers 1024-65535 may be freely used.
Post:
D0.L is 0 on success, non zero on error
Error codes:
Bits 0-15 Low word of D0
1 - general error
2 - network initialization failed
3 - invalid socket
4 - get host by name failed
5 - bind failed
6 - connect failed
7 - port already in use
8 - domain not found
All other values reserved
Bits 16-31 High word of D0, extended error code (see below)

Send data. **(Deprecated, use 106)**
Pre:
D1.L {31.....16}{15.....0}
Bits 0-15, Number of bytes to send
Bits 16-31, Reserved for future use
(A1) data to send

102	If server (A2) IP address of client as null terminated string (e.g. '192.168.1.100',0) Post: D0.L is 0 on success, non zero on error. Success does not indicate data was sent. Bits 0-15 Low word of D0 1 - send failed All other values reserved Bits 16-31 High word of D0, extended error code (see below) D1.L number of bytes sent. 0 if no data was sent. Unchanged on error.
103	Receive data. **(Deprecated, use 107)** Pre: D1.L {31.....16}{15.....0} Bits 0-15, Number of bytes to receive. Bits 16-31, Reserved for future use. (A1) received buffer, must be large enough to hold D1.W bytes. Post: D0.L is 0 on success, non zero on error. Success does not indicate data was received. Bits 0-15 Low word of D0 1 - receive failed All other values reserved Bits 16-31 High word of D0, extended error code (see below) D1.L number of bytes received. 0 if no data has been received. Unchanged on error. (A2) IP address of sender as null terminated string.
104	Close connections Pre: If TCP server D1.L IP address of connection to close, 0 to close all. Post: D0.L is 0 on success, non zero on error Bits 0-15 Low word of D0 1 - close failed All other values reserved Bits 16-31 High word of D0, extended error code (see below)
105	Get local IP address Pre: The network has been initialized with task 100 or 101 Post: D0.L is 0 on success, non zero on error Bits 0-15 Low word of D0 1 - get local IP failed All other values reserved Bits 16-31 High word of D0, extended error code (see below) (A2) local IP address as null terminated string. Max size 16 characters including null.
106	Send data on specified port. Pre: D1.L {31.....16}{15.....0} Bits 0-15, Number of bytes to send Bits 16-31, Port number Port numbers 0-1023 are used for well-known services. Port numbers 1024-65535 may be freely used. (A1) Data to send If server (A2) IP address of client as null terminated string (e.g. '192.168.1.100',0) Post: D0.L {31.....16}{15.....0} Bits 0-15, 0 on success, non zero on error. Success does not indicate data was sent. 1 - Send failed All other values reserved Bits 16-31 Extended error code (see below) D1.L {31.....16}{15.....0} Bits 0-15, Number of bytes received. 0 if no data has been received. Unchanged on error. Bits 16-31, Reserved for future use
	Receive data and port number Pre:

107	D1.L {31.....16}{15.....0} Bits 0-15, Number of bytes to receive. Bits 16-31, Reserved for future use. (A1) received buffer, must be large enough to hold D1.W bytes. Post: D0.L {31.....16}{15.....0} Bits 0-15, 0 on success, non zero on error. Success does not indicate data was received. 1 - receive failed All other values reserved Bits 16-31 High word of D0, extended error code (see below) D1.L {31.....16}{15.....0} Bits 0-15, Number of bytes received. 0 if no data has been received. Unchanged on error. Bits 16-31, Port number of received data (A2) IP address of sender as null terminated string.

Extended Error Codes in high word of D0

$2714 - A blocking operation was interrupted
$271D - Socket access permission violation
$2726 - Invalid argument
$2728 - Too many open sockets
$2735 - Operation in progress
$2736 - Operation on non-socket
$2737 - Address missing
$2738 - Message bigger than buffer
$273F - Address incompatible with protocol
$2740 - Address is already in use
$2741 - Address not valid in current context
$2742 - Network is down
$2743 - Network unreachable
$2744 - Connection broken during operation
$2745 - Connection aborted by host software
$2746 - Connection reset by remote host
$2747 - Insufficient buffer space
$2748 - Connect request on already connected socket
$2749 - Socket not connected or address not specified
$274A - Socket already shut down
$274C - Operation timed out
$274D - Connection refused by target
$274E - Cannot translate name
$274F - Name too long
$2750 - Destination host down
$2751 - Host unreachable
$276B - Network cannot initialize, system unavailable
$276D - Network has not been initialized
$2775 - Remote has disconnected

(Refer to Winsock file winerror.h for unlisted error codes)